Game Development and AI

Analysis and educational strategies on artificial intelligence systems used in game development.

Author Peduzzi Pietro

Submitted: 02.10.2024

MA Research Thesis
Lucerne School of Film, Art and Design
Master of Arts in Design: Digital Ideation

Research Supervisor	Professor, Axel, Vogelsang
Second Reader	Professor, Hans Kaspar, Hugentobler
Design Supervisors	Professor, Gabriele, Fackler
	UX designer and researcher, Allison, Crank
	Professor, Marcel, Uhr

HSLU Hochschule Luzern

Master Design

Abstract

The video game industry is rapidly growing, with global revenue projected to rise from $262 billion in 2023 to $312 billion by 2027. At the same time artificial intelligence (AI) research has become widely discussed, many companies have started exploring and implementing AI technologies for performance and profitability. The increasing use of the term “AI” deriving from these processes has led to confusion, especially in the game design industry, where now the use of the term is often inconvenient and can lead to struggles in communications due to its morphing interpretation.This lack of clarity does not only affect game developers but also other stakeholders, delaying communication and slowing progress.

This research addresses these issues by exploring how game developers can better understand and apply AI systems in their work. Rather than offering a fixed solution, the study proposes strategies for analysing and mapping these algorithms, focusing on how information can be taught and shared. The document will start with an overview of AI development in general as well as in game design specifically, followed by an analysis of AI algorithms and the potential approaches to education. By helping developers navigate and understand AI technologies, this work aims to strengthen their creative freedom while maintaining high-quality experiences for players.

Keywords:

AI system, video games, game development, data visualisation, library

Acknowledgement

I would like to thank Axel Vogelsang for his support throughout the course of this research. His feedback helped me to better understand the scope of this thesis and highlighted new and interesting paths to investigate on the topic.

I would like to thank Allison Crank for her in-depth insights on the topic of AI in relation to the gaming industry, her availability of offering support and her invaluable help to find a better focus on what has later become the main topic of this work

I would like to show appreciation to the Master Design team for the coaching offered and my colleagues for the open conversations that brought new and diverse points of view on the research, allowing me to understand the importance of clarity and considerably enhanced the research with new materials and feedback.

Furthermore, I am thankful to Goran Sarich, René Bauer, Sebastian Burkhardt, Ryan Brand and Martin Hodler for their availability, insights and helpful feedback that contributed immensely to the contents and refinement of this paper.

Just as importantly, I am truly thankful for the support from Elisa Blatter, my girlfriend, and my family whose feedback and open ears helped me understand what I really wanted to research in this academic quest.

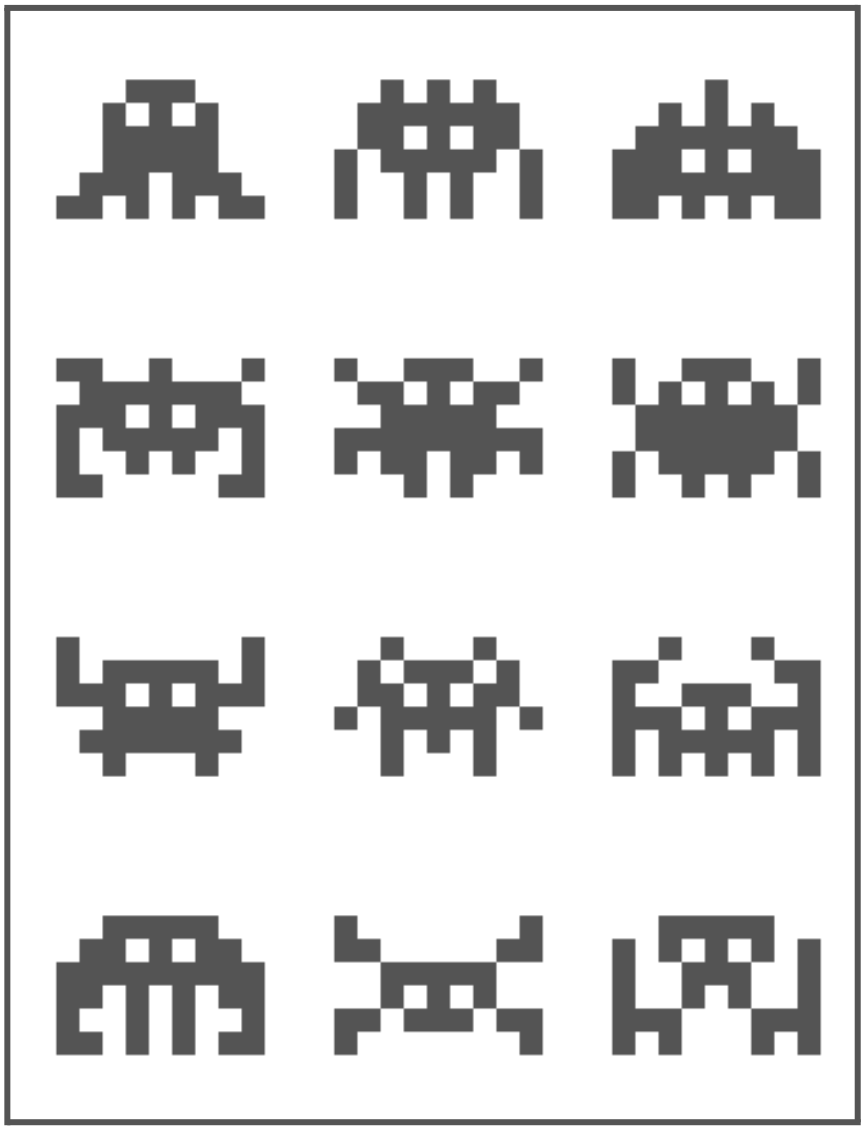
Affidavit

I hereby declare that I have independently prepared this research thesis. Concepts and ideas from outside sources are indicated as such. When writing the thesis, I used the AI-powered writing tools Copilot from Microsoft and ChatGPT from OpenAI for inspiration and as a tool between others to find fitting articles or media for my research. The work has not previously been presented to another examination authority nor otherwise published either in the same or in similar form.

_____

Pietro Peduzzi

Date:



1. Introduction

The video game industry is a giant in the global entertainment and media industry, and is experiencing an unprecedented surge in popularity and profitability. The total video game revenue is expected to increase from $262 billion in 2023 to $312 billion in 2027. New titles on multiple consoles and platforms are generated every day from all around the globe. However game developers still represent a niche compared to the broader software development field.
Research on AI algorithms on a mass level is also very common and talked about nowadays offering great opportunities where almost every company is discussing possible applications of machine learning systems for performance and monetary gain.
The increasing usage of the term “artificial intelligence” or “AI” is starting to generate issues especially when it is not clear what the expression itself is referring to. This leads game developers, especially new or independent developers, to be often confused on what artificial intelligence in video games is and on the qualities different AI systems have when applied to the game environment. This issue is not limited only to game developers but all the stakeholders in the industry, hindering communications between parties and development in general. Almost every internet search using the term “AI” will now lead to a machine learning project of sorts, but it wasn’t like this in the past.
To cover these issues, multiple strategies have already taken place helping creators and programmers to better understand these systems of automation. While the current state represents a promising start into the issues it is still a very chaotic structure, often leading to time and resources consumed into research. Information is scattered through the internet and context is often misleading or too specific for application. Given the various roles a game designer undertakes during development, especially in smaller teams and tight schedules, it’s crucial to allocate resources effectively, prioritising creative tasks that will help the game to be unique and the experience pleasing.
This research aims at addressing these issues while examining possible solutions to help game developers in their journey, ideally by offering a strategy to map and analyse AI systems to better understand them, while opening a conversation on how information can be taught. The objective is therefore not to offer a ready made solution, but to discuss the different variables that have to be considered when approaching the problem and offer possible directions to be taken.
Starting with a quick introduction on understanding AI systems and what they are used for in the field of game development as well as the state of the art. The focus will then shift on a more detailed analysis of the algorithms themselves leading to a proposal on possible approaches a designer can take to tackle these issues. Concluding there will be a chapter showcasing a proof of concept, a prototype that will then continue during the following design or practical thesis.

Although game design seems a niche in the design and software development industry, we have seen that it has a huge impact as a media. Helping game designers to develop games closer to their vision will grant them more creative freedom and space to convey their message, while keeping a high quality and experience for gamers and enthusiasts.

2. Definition of the problem

In this chapter I will address the current state on the topic of the thesis. Explain what facts lead to it and analyse stakeholders and state of the art. The chapter will terminate into the research questions for this study.

2.1. What is game development

Game development, or gamedev, is a multidisciplinary practice aimed at ideating, developing and distributing games. Tasks like programming, art, sound design, storytelling and many others can be often assigned to one or more individuals related to the team size. It can therefore happen that a game designer does not only work on tasks related to the programming of the game but also has to cover other tasks such as art, storytelling or sound design and vice versa.
Depending on the scope of the project, founding or even a publisher might also be involved, often adding restrictions and regulations for a more professional and marketable product. Games that do not rely on external funding are known as independent or indie. Game designers might start from scratch or rely on pre existing software specialised in game development, a game engine, such as Unity (Unity, n.d.), Unreal (Epic Games, n.d.) or Godot (Godot, n.d.).
When developing a game therefore a game designer has to take into account multiple variables between art, software and economics.

2.2. History of artificial intelligence

The concept and significance of the term artificial intelligence or AI has changed in the 21st century, a niche that was before only interesting to specific groups of researchers and developers is now common knowledge (even if by a limited amount).
The rise of computers started in the 1940s where they were already able to perform very complex calculations. The discoveries between the 1930s and 1950s at the neurological level showed that brains functioned as a network of neurons communicating through electrical pulses and given the similarities with these computational machines multiple researchers and scientists in various fields such as mathematics, psychology and engineering started to discuss the possibility of “machine intelligence” (Hodges, 2004). The term “artificial intelligence” was later coined by Alan Turing in 1950. In his article he suggested the possibility for machines to approach problems in the same way as humans do, by analysing the information following logic (Turing, 1950). AI therefore became a term to represent a machine that is able to take decisions or perform tasks autonomously.
The following years saw an increased research and development of more complex computational systems, machines able to prove geometry theorems or speak rudimentary English are just some examples. Due to the fast development and never before seen results, optimism started to rise and expectations raised with it (McCorduck, 2004, p. 218). Funding by large companies and even government agencies started to flow and multiple universities started to set up their own laboratories to research the topic. Following the success of the 1950s research came to a halt in the 1970s where expectations exceeded technological progression. Due to an understatement of the problems encountered the promised results did not occur prompting critiques and financial setbacks from the general public. The 1970s and following years, until 1990, were later defined as an “AI winter”, a definition derived by comparing the current state of research with a “nuclear winter”, where setbacks started from the public would later derive in less monetary inflow and later in a end of major research (Crevier, 1993). This momentary setback ended in the 1980s where a new form of system arised: “expert systems” (Jackson, 1999, p. 2). Programs that are able to solve problems only in specific fields (Jackson, 1999). Rules and knowledge is specifically gathered from experts on the domain and added to an existing program base, allowing flexibility and ease of use. For the first time AI systems proved to not only be interesting but also practical (Crevier, 1993). After the 1980s multiple corporations started developing and implementing AI systems to cut costs and improve performance. Around them hardware and software companies started to develop an industry that grew exponentially allowing huge cuts that gained millions of dollars (Crevier, 1993). By the start of the 21st century computers and machine learning techniques were powerful enough to assimilate large amounts of data (or “big data”).

Companies that until that point had all this data stored were able to quickly analyse it and extrapolate information increasing innovation, competition and productivity. By the 2010s big data models and AI related products reached into more fields such as ecology (Hampton et al., 2013, pp. 156-162) and many others. AI technology and advancements in deep learning lead to great progress in image and video analysis and generation as well as text or speech recognition. After the huge progress in large language models showing human-like behaviour and thinking and the public release of such powerful systems to the public such as ChatGPT defined the start of what we could call "the start of a new era" (Gates, 2023).

2.3. AI and game development

In this chapter I will dig a bit further in how artificial intelligence is connected with game development. Starting from the history and later describing how algorithms have been and are being used for or inside video games production.

In game design it has always been very common to talk about artificial intelligence, when referring to any system in games that is able to take decisions or act without the need of any user input. It is important to understand that the term AI in this context covers a lot of many different systems that follow the same simple principle of being autonomous. One of the most common uses of the term is when referring to NPCs (non player characters) and their behaviours, and already the workings and systems behind this kind of characters could vary significantly. Artificial intelligence is used in many more interesting ways during the game development process as a tool for game developers to speed up processes to autonomous sorting of players in online games during gameplay.
When talking about AI in video game development however we rarely refer to machine learning algorithms or neural networks, the focus often lies on more simple, small and light algorithms that are able to perform specific tasks aimed at a better player experience (cf. §3.2.1). It is not always necessary to have a very complex algorithm trained on huge amounts of data to move a point between two coordinates, very often it is enough to tell the point to update his position every frame in the direction of the desired destination.

Evolution of ai systems with time
The first instance of an AI algorithm in a digital display electronic game was in 1950 with the computational machine Bertie the Brain (Simmons, 1975, p. 17). This computer was the first one to be programmed to play games, more specifically tic-tac-toe against visitors of the Canadian national exhibition taking place in Toronto. Soon after followed the Nimrod (Ferranti International PLC, 1951) a computer aimed at showcasing the technological development in calculating power. The machine was used to play the game of Nim against the audience and it was programmed to, not only display the game state itself with the help of small LEDs, but to slow down the calculations and processes of the machine so that they could be explained to the audience. Despite the origin of the machine being showcasing its calculating skills, the audience seemed way more interested in playing the game. In the following years multiple systems arised showcasing the computational power of machines by playing checkers and other logic based games. The video game industry was however more focused on more simple logic and on a local multiplayer base such as pong (Atari, 1972) or Spacewar! (Russell et al., 1962), having 2 or more players playing the game removed the need for more complex calculations or systems such as AI to challenge the users.
In the 1970s some single player games started to feature rudimentary AI enemies, not controlled by other players and based mostly on predefined patterns stored in the code of the machine. Thanks to the technological advancements into microprocessors arcade games such as Qwak! (Atari, 1974) or Speed Race (Taito, 1974) or computer games such as Hunt the Wumpus (Yob, 1973) were able to create more complex and random patterns to the experience. Thanks to the success of Space Invaders (Taito, 1978), with its enemy patterns, increasing difficulty and random events, the industry started to understand the potential of AI driven enemies and characters and the impact they could have on the game experience. Adding enemies that are not reliant on player inputs and have behaviours that might change due to randomness added a layer of challenge and replayability to the game. In the following years the complexity and tasks that AI could perform increased significantly. Pac-man introduced different enemy

behaviours (Namco, 1980), as new genres of games appeared also the algorithms started to evolve and specialise on specific tasks. Dragon Quest IV introduced a system where the player was able to alter the AI behaviour to better fit their tactics (Chunsoft, 1990), while sport games such as John Madden Football (Antonick et al., 1989) or Tony La Russa Baseball (Beyond Software, 1991) started to develop systems that tried to replicate the real personality of the character they were based on.
From the 1990s with the rise of multiple new genres and hardware possibilities AI systems started to gain even more complexity, concepts like pathfinding, behaviour trees or finite state machines to control characters and environments in the games.
In the last decade game developers started to experiment with more advanced or complex AI systems such as neural networks like the one found in Supreme Commander II (Gas Powered Games, 2010), Dota 2 (Valve Corporation, 2013) or Battlefield V (Digital Illusions Creative Entertainment, 2018), chatBots as in event[0] (Ocelot Society, 2016) or even generative AI for creating infinite worlds in games such as No Man's Sky (Hello Games, 2016).

What is an AI system today in game development
The use of the term "artificial intelligence" is often confusing or can vary its definition from developer to developer. There is therefore a need for a clearer structure that could help creators to understand each other without terminology as a drawback in the discussion.

The constant evolution and progress on artificial intelligence in general as well as the continuous search for new and better experiences in the environment of game development made it so that the term "AI" itself got very awkward to define even between developers. When asking game developers (cf. §2.5.2.) to define what "AI" represents for them the answers were often contrasting or at least dubious on certain systems and application areas. When talking about machine learning and other recent big data trained systems nobody discusses the group belonging to artificial intelligence. But when discussing algorithms in the symbolic systems opinions start to vary depending on complexity and practice (Bourg & Seemann, 2004, p.22). As we have seen by definition any system that showcases autonomy in solving problems or tasks given to it can be defined as artificial intelligence, however in application this definition feels redundant and the need for a clearer structure is needed. An interesting view on the topic can be found in the book *AI for Game Developers:*

> From another perspective it's appropriate to think of AI as the intelligent behavior exhibited by the machine that has been created, or perhaps the artificial brains behind that intelligent behavior. But even this interpretation is not complete. To some folks, the study of AI is not necessarily for the purpose of creating intelligent machines, but for the purpose of gaining better insight into the nature of human intelligence. Still others study AI methods to create machines that exhibit some limited form of intelligence.
> This begs the question: "what is intelligence?" To some, the litmus test for AI is how close it is to human intelligence. Others argue that additional requirements must be met for a machine to be considered intelligent. Some people say intelligence requires a conscience and that emotions are integrally tied to intelligence, while others say the ability to solve a problem requiring intelligence if it were to be solved by a human is not enough; AI must also learn and adapt to be considered intelligent.
> AI that satisfies all these requirements is considered strong AI. Unlike strong AI, weak AI involves a broader range of purposes and technologies to give machines specialized intelligent qualities. Game AI falls into the category of weak AI (Bourg & Seemann, 2004, p.22).

The division between strong and weak AI grants a new point of view on the matter, a very useful addition to the other already existing categories. In game design this variable can be added easily to the existing categories, helping even more into the necessary definition of the multitude of systems. Very common is the division of algorithms for the genre they serve (Millington, 2019, pp.959-969), dividing the systems by the game genre helps with the creation of smaller more understandable groups of algorithms without overlooking existing interactions and combinations between them. Similarly dividing the systems by type or task the focal point moves into seeing the different possibilities to solve

the same problem(Millington, 2019, pp.39-800). This subdivision in system type can be also focused on the approach of the game developer creating the algorithm. Allowing therefore an additional way of categorisation of AI as techniques (Sweetser & Wiles, 2002).

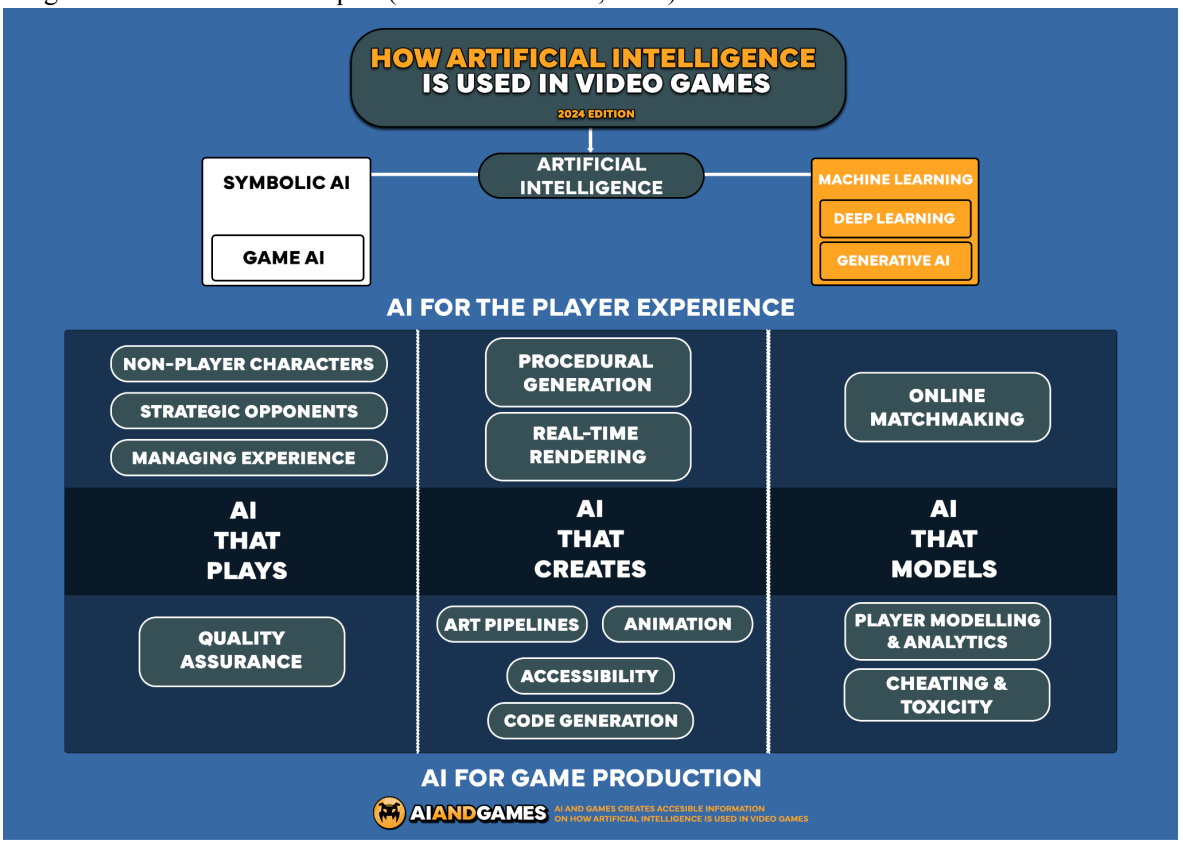


Figure 1 - Graph showing distribution of AI systems and their role in game development.

Symbolic systems in games

The main type of artificial intelligence algorithms found in game design can be referred to as symbolic AI.Due to the reputation it has and the long time it has been part of game development it is sometimes also referred as “classical AI” (Thompson, 2024).

It is possible to find examples of symbolic systems already in the first games as Pac-man with his enemy behaviours (Namco, 1980). Although at first glance the ghosts behaviour in the game might seem randomised it follows a specific set of rules, with a slight variation for each character the game triggering a sense of intellect (Retro Game Mechanics Explained, 2019). Symbolic systems cover to this date the majority of AI algorithms used in video games, due to their ease of use and logical construction they are very straightforward to handle. Due to their structure being built on rules and logical behaviours this kind of systems are also easier to understand than the complex behaviours developed in machine learning algorithms, where often the patterns are generated automatically by being trained on examples.

The importance of the distinction between symbolic and machine learning systems is connected to the previously mentioned shift in the meaning of the term “AI” for the general public. Due to symbolic systems being limited by the rules and logic they follow, even if they can be very complex on their own, it is often challenged whether simpler systems are still “intelligent”. Fact that per definition is true when recognised that all the symbolic AIs per nature are weak AI systems.

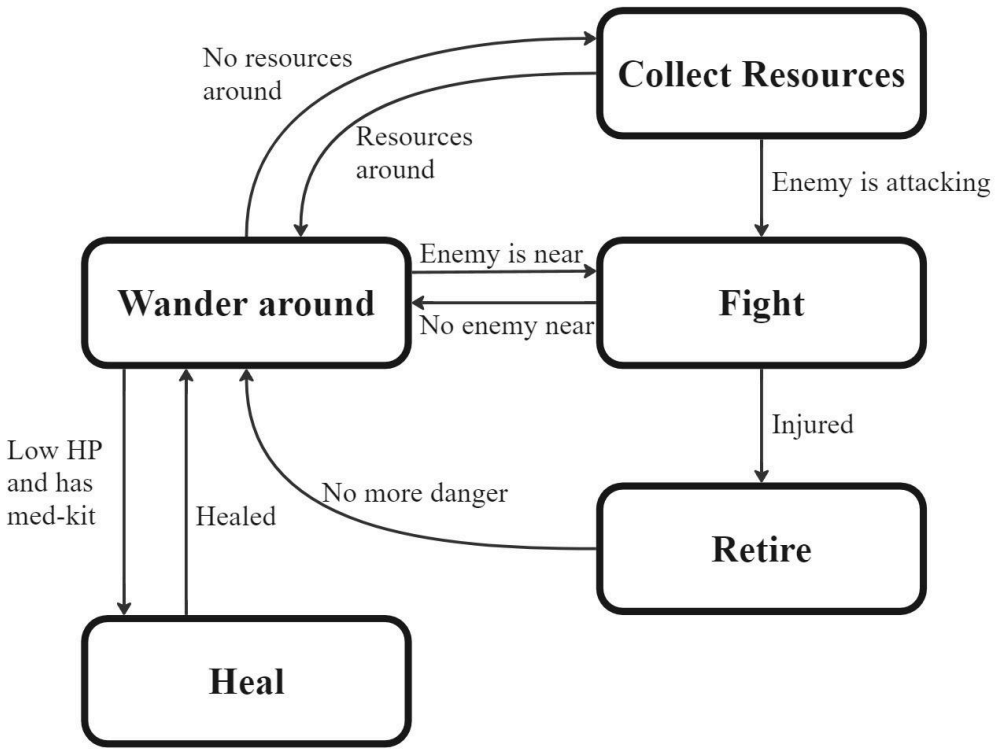


Figure 2 - Graph showcasing a simple state machine. Actions are influenced by the states around the character.

Machine learning and game development

Recent years saw the rise of machine learning algorithms as tools in the game design business. Differently to symbolic algorithms, machine learning can be more expensive to develop and often is used as a tool during development or to enhance existing algorithms.

Machine learning tools are still very new to the game design industry, although some common tools like Large Language Models (LLMs), image generation softwares and other similar algorithms can be easily used to produce some quick prototypes for assets and stories, the still unsettled rules regarding ownership and rights render the usage of such systems unreliable, especially on commercial projects.
Although there have been some experiments with big data algorithms such as ChatCPT-like systems as seen in the game Suck Up! (Proxima, 2023) these games are still subject to limitations, the costs of training and running the game is still way too high, especially for big numbers of people accessing it at the same time, as often seen in gaming.
Game designers express the interest for such systems as an helping tool to understand in an easier way the approach they want to take on a project but, due to these systems being very generic in nature and without "creativity" the preferred approach (especially in smaller studios) is still the classical artistic approach. Having an artist to discuss and create the assets still grants way more freedom and defines a very important level of uniqueness for the project itself. An interesting fact is the bigger interest and speed of code generation or coding assistants for many languages used in game development. Generic LLM tools like ChatGPT and copilot, or more engine specific tools like Muse and Unreal Assistant, are very often used to get a grasp on the possible logic behind the game behaviour as well as a help for faster coding thanks to their autofill options. The tools still require a very good knowledge of the programming language to get a consistent and performant result but are still reliable enough that it is pretty common to see content creators pushing the limits of the systems themselves to see what they can generate just by themselves and some prompts from the developer (candlesan, 2023).

Machine learning for creation

Machine learning tools offer flexibility during development and allow to cut costs and reduce time for production if used in the correct way.

Using AI for development and asset generation is common especially for tedious and time consuming tasks. An example is the game Mass Effect: Legendary Edition (BioWare, 2021) where the development studio BioWare used AI upscaling tools on the game's original textures (BioWare, 2007) to create better textures and material for the remastered version (a version updated for new consoles and devices).
This kind of process has been used too by third parties creating mods for already existing games, normally old or with low resolution textures. Thanks to the ease of use and speed of these upscaling technology powered by AI it is possible to find mods for games like Morrowind, Doom II and Max Payne with enhanced graphics and a completely refreshing and updated look.
Soon it will be even possible to use the power of GPUs (graphic cards in computers and consoles) to enhance texture and materials using AI in game. Thanks to the improvements in the NVIDIA GPUs (Nvidia, n.d.) the calculating power and speed of such cards is very high. Developers could therefore work with lower resolution textures to then upscale during playtime directly, taking less time to create, less space to store and less power to display on screens. These advancements are of course not limited to gaming, spreading to any type of design and rendering related task for computer devices.

Figure 3 - Comparison of original textures and AI enhanced textures in Mass Effect Legendary Edition.

Machine learning to create characters or managing a game
Machine learning can be used as a tool or base to create and manage multiple complex behaviours in games. Often based on more classic systems an AI is then trained to use the set of rules available to become better at its task.

Machine learning, or more precisely reinforcement learning, is a very powerful tool to enhance NPCs' behaviours. An example are competitive or sports games, where the player will have to face off with the computer's controlled characters for a long period of time, if not for the whole duration of the game. Of course multiplayer nowadays is taking a lot of these requirements away but, in some cases like a single player campaign or a situation where the opponent players are not enough, the computer needs to take control of the missing characters or teams. We can find reinforcement learning in games like FIFA 23 (EA Sports, 2022) and Gran Turismo Sport (Fuchs et al., 2021) In these cases the computers are given the same set of rules that the player is given, the characters can move only in the way the real player would be able to move them and so on. The computer is then trained against himself in multiple

generations. Differently from the classical methods of machine learning the computer is not given any external data other than the rules he has to follow and will have to generate the data by himself through trial-and-error. Every generation starts with the same set of rules and tries to achieve a task, achieving the task is seen by the computer as a threat while a bad performance is punished. The best result from one generation will be set as a basis for the next with the addition of a small randomness factor in the variables allowing the software to evolve and get better from generation to generation. This method gives the game developers a lot of freedom into creating the characters and thanks to the different generations it is possible to set the game difficulty for the player making them face newer or older versions of the system (the newer the more difficult the game). A big issue for this system lies on time and updatability. When training a simple NPC without many rules it takes some time to get good results . When training a very complex system where the situations and rules are many more it can take way more time and calculating power to train. Updating the system for balancing the game or adding new content can also be very time consuming, since any changes that influence the AI too much might break it and make it so that it requires new training.

Machine learning to manage the player experience
When talking about multiplayer games, a lot of time and resources has to go into management of the interactions between players to ensure the experience is pleasing for everyone. Thanks to the use of specialised algorithms it is easier to find or recognise toxic or damaging behaviours and act on solving them faster.
AI doesn't necessarily need to take the place of a player or direct action on the difficulty of a game, a common use of AI in online games for example is the multiple systems used for cheat recognition and toxicity. Due to the huge amounts of players online at any moment in games it is very difficult for moderators to check and ban manually cheating or toxic players that are ruining the fun of others. On this topic the workings of the systems are often kept from public view to prevent cheaters circumventing the system knowing how it works. However developers at Valve for example have showcased and discussed their usage of deep learning to identify cheaters in the popular game CS:GO (Kombucha, 2018).

Games as training tools for AI
On top of enhancing the game experience I feel it is important to showcase how in the opposite direction games are a very good platform to test and train AI models in a secure and cheaper environment.
Similarly to the situation with reinforcement learning it is possible to create game environments that are simulating the context where we want to apply our system into, simulation and testing of the algorithm in a digital world is way faster and can be tracked with multiple variables allowing for tweaks and adjustments without the costs of building a physical version . This type of training is used by BostonDynamics to train and evaluate the robots capabilities without risking damaging the real product (BostonDynamics, n.d.). While other projects and researches have a more direct approach embracing the world offered by games to test the AIs in a more direct way like in Project Malmo testing the system reactions in the Minecraft environment (Microsoft, n.d.) and research on autonomous driving using Grand Theft Auto V as an environment to learn about stop signs (Filipowicz et al., 2017).

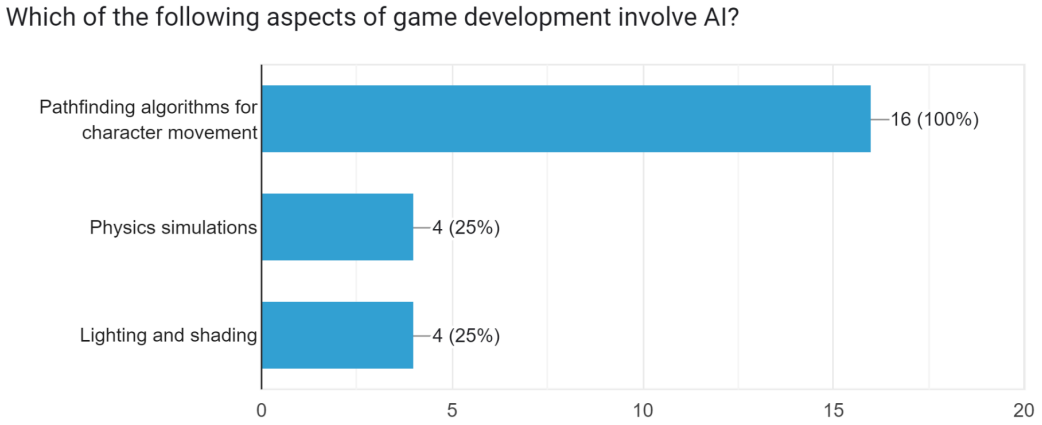
2.4. Addressing the stakeholders

A survey directed to game designers and enthusiasts was conducted during the Summer of 2024, although still very general in the topic of AI the investigation was aimed at a better understanding of the following topics:

- Current opinion on artificial intelligence in game development.
- Position of developers specifically regarding machine learning algorithms.
- Testing and understanding the needs of the stakeholders on the topic of learning and understanding AI systems.

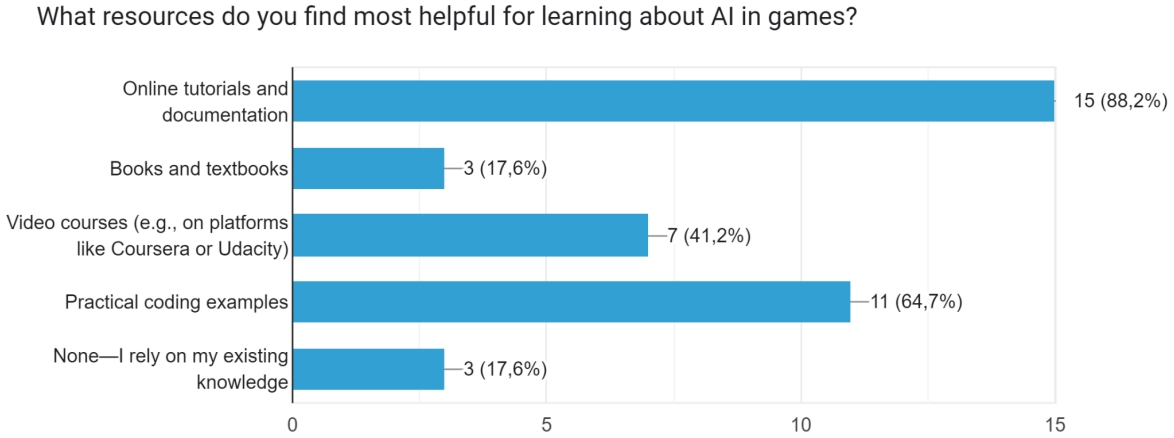
The survey itself was not detailed enough to already give substantial results on the main thesis question but had enough content to confirm or eliminate eventual biases and highlight patterns or pain points on the topic.

Although the majority of participants agreed on the role of AI systems for controlling and managing NPCs, for other tasks the opinions were more divergent, highlighting a discrepancy between different developers' view on AI and what it is defined from. Interestingly it seems that the systems that are less recognised as artificial intelligence are the ones that often are offered by third parties or come in ready made packages. An example being in the area of “Lighting and shading” a minority of developers recognised that AI can be used to help with the task. Although lighting is a very important aspect in game development it is often offered in a ready-made state from game engines and its components are rarely modified.



Tab 1 - Form on AI in games, showing what is perceived as AI in the game development environment.

Other interesting results cycle around the method of research, seeing books or research articles as the least used tool for research. The most common approach was often connected to an internet search, with online tutorials and documentation as the most recurrent.



Tab 2 - Form on AI in games, showing what tools are most used for research by game developers.

Overall almost every game developer is familiar with the term “AI”, although it can be interpreted in multiple ways, all agree it is an important aspect of the development process and it can be used to enhance the player's immersion in a game.
None of the people who filled the form were against a library for AI systems used in video games, although some were not sure about its utility, often depending on the approach or structure a platform of this type could have.

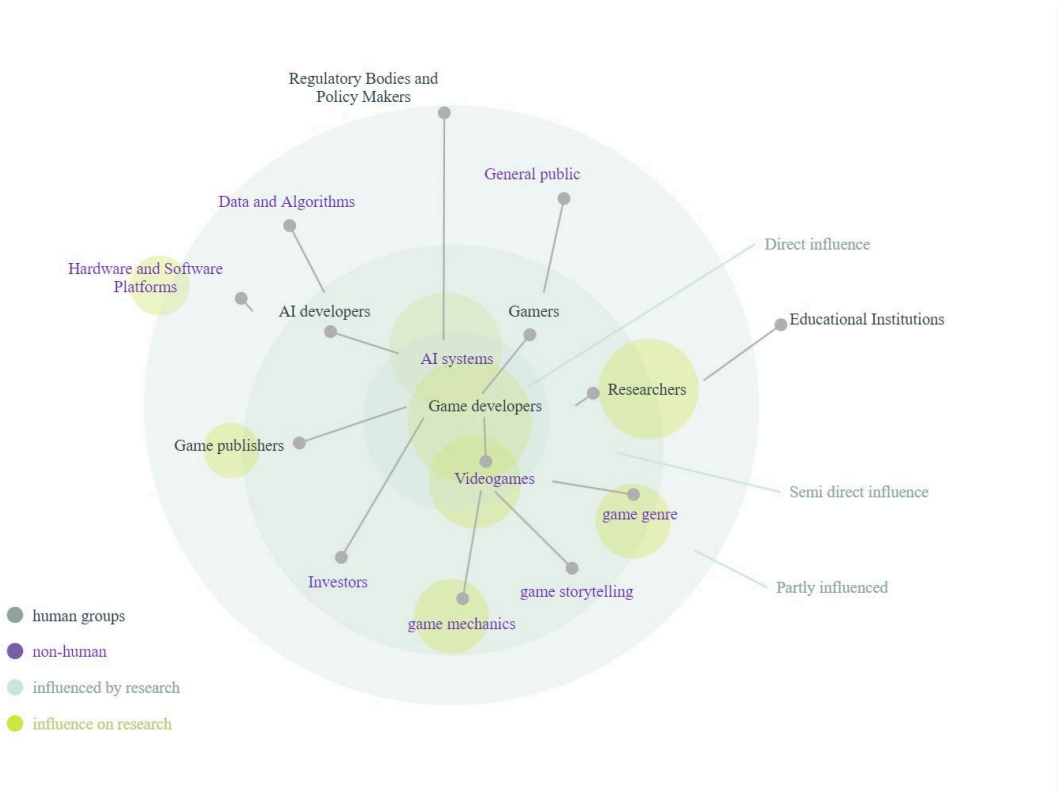


Image 4 - Stakeholder map showing human and non-human stakeholders and their relations.

The stakeholder map showcases all human stakeholders as well as their relations. In focus and centre of the map we can find the game developers, as direct users and creators of the ai systems themselves their inclusion is imperative for the research. Still in the centre of the map we also find the main non-human stakeholders, any change or help given to game designers will significantly influence the two groups videogames and AI systems. Other human stakeholders like gamers, investors, publishers or researchers are all important parts of the game journey and therefore have an influence on the final result. This research, although will be heavily focused on the point of view of a game designer, will take in consideration all these secondary stakeholders and the influence they have on the primary. Due to the broad topic revolving around AI for the general public, focusing on one target group will allow for a more in depth research on solutions. I believe that allowing the central stakeholder group to better understand and use AI systems can indirectly benefit all other stakeholder groups due to the influence of the first on the others. Game developers work in very close contact with this type of algorithms and due to the surrounding environment the approach on the topic is very playful, allowing space for creativity, while other branches often focus more on productivity or performance. This unique setting is a great start to help in clarifying the role of AI in everyday life in a lighter manner.
Highlighted in green we can also see the stakeholders that have an influence on the research itself, it is important to notice that even if a stakeholder can heavily influence the research it doesn't mean it will be influenced itself by the thesis and subsequent prototypes.

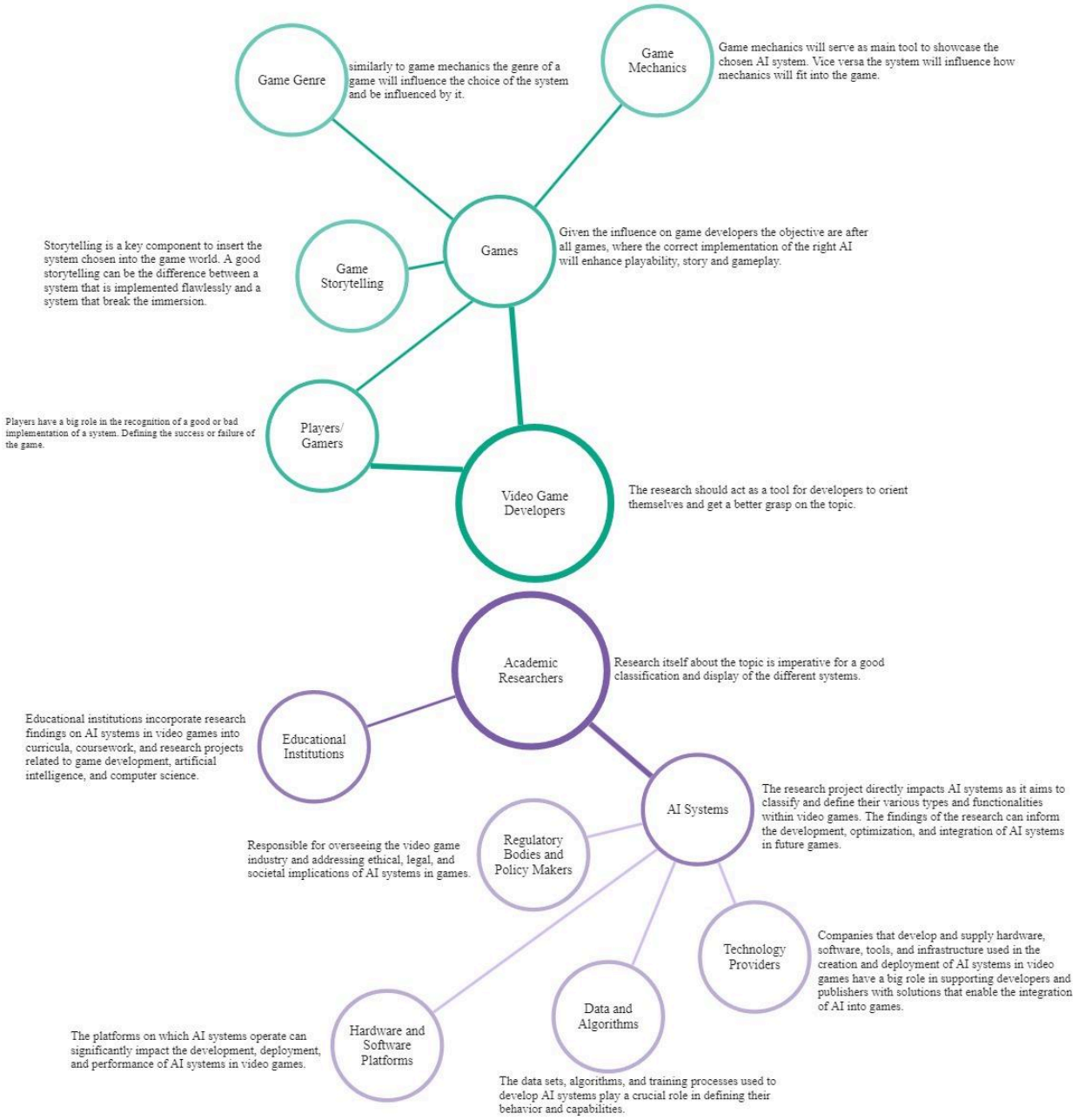


Image 5 - Relations map showing direct relations between stakeholders.

This map is more focused on stakeholders relations, the map is divided into two main topics: direct approach on primary stakeholders (green) and academic research influence on the thesis (purple). To achieve a significant result it is necessary to understand and consider all the researches and existing categorisations of AI systems to then be able to apply the knowledge directing the focus at helping the game developers themselves and anything connected. The duality of this topic, applied and research, can be bridged to connect knowledge and know-how allowing game developers or enthusiasts to express their creativity without setbacks from technological limitations or theoretical approaches.

2.5. State of the art

In this chapter I will discuss how it is currently possible to find and research information concerning AI systems when creating games. I will also examine the materials found with their pros and cons and highlight the opinions of game developers on what is missing or could be improved.

2.5.1. Media

The fastest and most consistent way to find information on AI systems nowadays is the internet. The majority of game developers, when in doubt, start with a quick internet search about the issues they are encountering or the system they would like to understand better. Keywords and searching directly for encountered problems are the fastest way to find solutions (cf. §2.4.). Although limited by the search engine algorithm this method of research offers a quick dive into a topic without the risk or time needed in other approaches. Development blogs like gitHub (GitHub, 2008) as well as video tutorials, engine specific platforms or even libraries can be found with this method. For experienced developers a quick internet search is often the best option since the context and best practices are already clear. This method can be however dangerous or time consuming in the case of more specific and unique questions the developer might have. Not knowing on which blog or platform to look or only getting examples of similar problems but not exactly fitting solutions is frustrating and can lead to errors in the project.

Websites

Websites are often focused on explaining a game engine or a programming language in general and not focused on specific AI systems per se. It is useful because often discussions open up multiple solutions on one problem and it is easier to share your own experience with others that understand the topic. In addition if the developers don't have a strict schedule it is still possible to open new topics of discussion where different opinions and solutions might arise, even on very fresh topics that haven't yet been covered on other types of media. Game engine platforms such as unity are well organised to cover all topics related to them and offer extensive guides, pictures and sometimes examples on various topics. These webpages are rarely focused on creation of artificial intelligence systems, but materials for implementation or use of ready made solutions can be found. This offers some flexibility but is still limited in application. Social media platforms like YouTube or reddit are also great sources of information and exchange. Video essays on many topics ranging from very specific tutorials (Brackeys, n.d.) to more logic and structure based explanations (Lague, n.d.) can help a lot due to the dynamic structure they offer as well as the audio explanation supported by the visuals requiring less energy to follow than reading. Threads on the other hand offer a platform where to exchange and discuss with other experts and enthusiasts, the live and open nature of these networks mean that the topics touched are always up to date and the possibility to exchange will always bring new and fresh perspective to the problems discussed, eliminating biases that might have arised.

Books

When the objective is learning, books cover the most topics in a structured and complete way. Introductory books explaining the basics of simple game algorithms like *AI for Game Developers* (Bourg & Seemann, 2004) are a very good start for beginner developers and due to the generic overview approach the content remains significant even after multiple years. For more experienced developers there still are books that are a bit more challenging in the reading like *AI for Games: Third Edition* (Millington, 2019) where more practical examples and even pseudo code can be found explaining the concepts in detailed approach maintaining enough level of abstraction to avoid becoming obsolete. The biggest issue for books is the prices and time necessary to invest into them. This factor makes them the best choice in a learning environment, while during development and especially inside a schedule an internet search often offers the best return on investment.

Research articles

A similar case can be said about academic research, articles and books aimed at specific systems. Although they often have great and interesting content. Reaching it is often tedious and expensive (especially for non-students), so that it is rare to see a game designer looking into academic research for his own research. In the case of system specific books the situation might vary. When a developer is not able to find answers or wants to learn more on determined methods they often offer great contents and examples, but not knowing exactly what to look for might be an investment not all developers can afford, especially in the indie community.

Game design school
In an educational environment, school or courses, it is possible for students to learn about AI systems, used for video games and not, in the most direct way. Exploring the connections between creativity and technology the students are able to learn on both theoretical and practical approaches, allowing a better perception on how artificial intelligence algorithms and systems can enhance gameplay, performance and overall quality of the final product if done correctly. While in design related institutions it is possible to learn about games in a wider space, connecting gameplay with all the other artistic and technical aspects, in informatik or software engineering associated studies it is possible to really experiment with machine learning, neural networks and symbolic systems. Observing how different systems can influence solutions in a variety of ways. In both cases the direct support offered by the schools or professors leads to a safe environment of experimentation where students can collaborate in hands-on projects, bringing the theory to practice while learning.
Events and conferences
Being digital or in real life events can be very important for a game designer. Offering a place of direct encounter with other professionals as well as an opportunity to playtest games. In huge fairs like Comic-Con (San Diego Comic Convention, n.d.) or smaller ones like FantasyBasel (*Fantasy Basel*, n.d.) or Pop-Con (Amazing Event AG, n.d.) the affluence of people interested in games offers a great occasion to learn more and create connections in the industry. While on a digital side the interactions are often topic related, in real life discussions might lead to ideas or problems that would otherwise be missed. Moreover this type of events host conferences bringing on the stage veterans and experts in the industry to discuss various topics, also about AIs (AI and Games et al., 2024).

2.5.2. Experts opinion
During research I had the opportunity to contact and talk to multiple professionals and experts in the game design industry. In this chapter I will try to summarise my findings as well as highlight their opinion on AI and proposals and ideas on what it can be done to help game designers and others within the topic of learning and knowing about artificial intelligence in video games. All interviews transcriptions can be found in the appendix (cf. §10.).

The opinions on what “AI” stands for were very mixed. Interestingly everyone agreed that machine learning algorithms, or other systems based on big data, are in fact artificial intelligence. While some developers had an opinion similar to the one we have seen by Millington (cf. §2.3.), where AI can be subdivided into machine learning and others, other developers were hesitant to consider anything that wasn’t a machine learning algorithm as artificial intelligence due to the lack of complexity in some systems. This to the point where even algorithms like chatGPT could be possibly excluded from the definition of AI since “I think the idea is that it’s intelligent in a way a human is. But these kinds of AI's are not really, they are just copying stuff.” (S. Burkhardt). Symbolic algorithms or classical AI was almost never considered if not indirectly by defining as artificial intelligence any system that can be seen as “a coded behaviour of an entity inside a software structure” (G. Sarich).
Generally everyone agreed that the meaning of artificial intelligence shifted dramatically with the upsurge of LLMs and other machine learning algorithms. Right now when talking about autonomous behaviour and patterns in a games code developers are hesitant to use the word “AI” to avoid misunderstandings on the complexity of a system by the interlocutor and prefer to use other terms or even straight avoiding the word completely by describing the algorithm by the task it accomplishes.
An interesting point was brought up when discussing what AI is used for. When creating a game or testing it is very common to see systems based on machine learning technologies. Asset production, storytelling and coding all can have some form of support by tools and softwares with machine learning AI features. When looking at games themselves however it is still very rare to see mechanics or systems based on this kind of technology, this because classical AI still offers the most flexibility for adjustments and control on the player experience.
This duality of AI in game development highlights the strength of the different types of algorithms and shows that one is not taking the place of the other, but they can be combined to increase performance, decrease production time and cost and still bring unique and memorable experiences to the players.
Opinions on the use of machine learning softwares were, although very clear, on the fact that as tools they should not or cannot completely substitute the artistry and creativity of the game designer or artists

collaborating on a project especially when talking about asset creation and art. This for similar reasons to the ones we have seen previously (cf. §2.3.), where any algorithm that is trained on a great amount of data will always generate generic results, and the action from the designer is necessary to add changes and tweak variables to get a result that is unique and original. These systems can therefore help to speed up the process but it is really important to keep control on the outcome and when necessary take the time to mould it to the desired shape, thus avoiding the fall into a generic loop of projects that all look and feel bland and uninteresting. A similar view is on chatbots or any behavioural algorithm, while it is getting easier to set up said algorithms with machine learning it can be problematic to get them to behave in the desired way for storytelling. Having a completely autonomous system also means that there is no control over it and this can work for certain projects where it is not important how information is given to the player but it can be harmful in other cases.
It is therefore very important to help and guide game designers to understand the strengths of these systems and not fear the change. Showing how machine learning tools can be used for prototyping and production and simpler systems can be combined or modified to allow creativity and identity to show through the games.

In the second part of the interviews, methods of research and learning were discussed. How can a game designer learn information during development, how experimentation and exploration influences games. The most common method for research for all developers was internet research. All necessary information can be often found on websites, online videos or blogs. Although articles and academic research was nominated it is still not accessible enough for it to be worth the time invested.
In the following paragraphs I want to list the most requested and looked for tools and methods during research that became apparent during the interviews.

A good description

Not surprisingly when researching a very important aspect is to find a description of the system, whether is a video, a graph or text it is helpful to understand the main rules a system operates in. Often imagery is appreciated helping the reader to visualise the logic and workings for an algorithm. Due to the differences in game engines and coding languages a general description is preferred, so that the developer can later translate the knowledge in a system without having to understand any other programming language or software. This abstraction can be helped with examples.

Tutorials

Having a guide that helps to get a first understanding on the concepts is always a good start, a very common starting point was the YouTube platform, where a lot of information and material can be found. Due to their accessibility and ease of sharing knowledge, through visuals and audio, videos are common for this part of research. Another option can be found on platforms like Git-Hub or other blogs where step by step instructions can be found and easily followed. Although tutorials are often specific to a game engine or programming language they offer a lot of information and are therefore appreciated.

Examples

Examples are often talked about, the definition or level of depth may vary. Having a very abstract example can help with the understanding of the logic of a system, while seeing a practical example or an application of a system can help understand the feeling and experience it can transmit.

Opportunity to practice

Practice occurred to be a preferred and encouraged action to take when learning on anything and therefore AI systems too. Having an environment where it is possible to experiment and test ideas and at the same time learn the behaviours and applications of a system will help developers to get a complete understanding, theoretical and practical, on it.

Exchange and discussions

Having someone to discuss issues or even findings with was a frequent suggestion too. While the environment is not directly important, if in person or digitally, having someone with more, similar or less knowledge to discuss with can help highlighting parts that were not clear or answering questions that arise during research in a direct way.

The end of all interviews introduced the main concept for what will later be opportunities of the design thesis, or for a practical strategy to the issue. All experts were asked how they would like to learn more

on systems and algorithms used for and in game design as well as their opinion on a first draft of the current approach that we will discuss when talking about prototyping later (cf. §5.).
Costs were often brought up during this section, when talking about consulting opportunities the main concerns were around the fact that independent or solo developers do not have the monetary means to afford consulting experts while bigger companies often have someone working for them already that covers that role. Real time events were also a proposition that came up, with the example of GDC (GDC, n.d.) as a good practice and helpful event where developers don't only have the opportunity to learn on new progresses in the industry but also get to know other professionals and enthusiasts.
Concluding with the most supported method of an online platform. Cost, accessibility, space for improvement and opportunities for a community are all favourable elements of the web based solution.
Based on examples of a similar project Locomotionvault (Di Luca et al., 2021) the feedback was positive, where a digital library for AI systems used in games has potential to help and connect game developers.

2.5.3. Good practices

Results from the interview and open form highlighted certain practices that are preferred by game developers during research. In this chapter I will talk about the most prominent ones in more detail and explain how they can support learning of an AI system.

Generalising systems

When talking about AI systems in the context of game development, it is always best to keep a level of abstraction. This is due to the multiple game engines in use today, such as Unity, Unreal Engine, and Godot, as well as custom engines created by game developers. These engines can differ significantly in the programming languages used or internal structure of the software, leading to inconsistencies or differences in how AI systems can be implemented. As a result, having a specific description of an AI system for only one engine can limit its relevance and understanding for developers who are not familiar with the single platform presented.
For a developer it is more important to understand logic and overall structure of an AI system, rather than getting lost in engine-specific details. By focusing on abstract concepts developers get the freedom and flexibility to implement these systems in any language or engine they choose, without having to translate or from one programming language to another. This abstraction therefore brings a clearer understanding of AI structures, allowing developers to reproduce these systems in the environment or programming language they are most comfortable with. If AI systems are explained with too much focus on specific engines, developers would first have to learn the engine or at least the logic behind it before they can understand how the AI works, which can cause confusion and slow things even more. By focusing on the overall structure and main ideas, we can make learning easier and help developers apply AI techniques in different engines and tools.

Pseudo code

Using pseudo code is a very effective method for illustrating the structure and logic of an AI system. Having access to a structure where no specific programming language is used allows developers to focus on the core functionality of the AI system without getting restrained by the syntax and specifics of any particular language or software. Similarly to the previous point on generalising a system description, having the code to be more abstract gives developers the flexibility to adapt and implement these systems in the programming language or game engine they are most comfortable with. This approach also helps with a better understanding of the AI's structure, since the developers will have to translate the logic into their own code, supporting both comprehension and practical application. Additionally having a generalised programming language will help with communication and discussion between individuals that may use different languages or tools. Contributing to a common ground where the focus is on the problem-solving approach and structure of the AI system, rather than on specific technical details. Point especially valuable in game development, where the same AI principles can be applied across a wide range of game engines or platforms. Pseudo code is therefore an ideal tool for showcasing AI systems in an accessible, flexible, and adaptable way. Giving developers a template that can be easily customised to suit their needs and fits in their programming environment.

```
class Kinematic:
    # ... Member data as before ...

    function update(steering: SteeringOutput, time: float):
        # Update the position and orientation.
        half_t_sq: float = 0.5 * time * time
        position += velocity * time + steering.linear * half_t_sq
        orientation += rotation * time + steering.angular * half_t_sq

        # and the velocity and rotation.
        velocity += steering.linear * time
        rotation += steering.angular * time
```

Image 6 - Image showing an example of the pseudo code presented in AI for Games (Millington, 2019, p.49).

Abstract examples

Following the two previous points examples can also help with understanding of a system. Similarly to pseudo code, abstract examples can focus on the main structure of an algorithm in a visual manner. Helping further game developers into the logic and limitations of a system. Small animations and illustrations using simple geometric shapes will focus the attention on the understanding of the AI system itself, without any distraction that other variables like fancy visuals could bring in, empowering the developer to make better and unbiased decisions. Abstract examples give developers a point of reference. Showing how an AI algorithm might handle different situations without getting too technical.

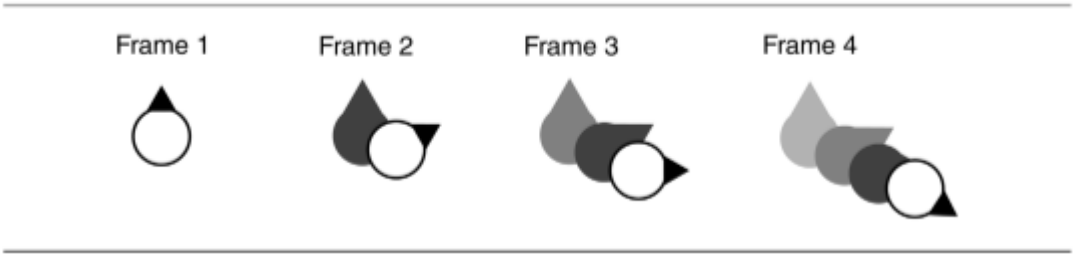


Image 7 - Abstract example of an AI behaviour presented in AI for Games (Millington, 2019, p.49).

Implemented examples

While abstract examples help with an insight on the AI system itself it is not necessary to limit examples to that. Showing an implementation or finished project can allow a developer to understand how the AI system will behave and feel during gameplay. These examples will show how an algorithm can interact with the rest of the game and outside of a “sterile” testing environment. Seeing and playing a game that uses a certain AI system will help with understanding his strong and weak point and maybe inspire the game developer into additions or modifications he feels could be fitting or are missing in the example.

Leaving space for discussion

Having a space to discuss, compare and showcase AI systems seemed a very important point for a lot of game developers. When learning or experimenting on algorithms it is helpful to get external feedback from other developers. From the ones that have more experience and can help us understand what could and could not work to less experienced developers or external viewers that could bring in new ideas or ways to render an AI system more simple or feel better.

Giving the tools to experiment

“Best way to learn is to do” (Halmos et al., 1975, p.466), giving the users a space to experiment and test what the possibilities are with an AI system is a very good way to teach and help to understand the algorithm better. Ranging to games that are made to teach (Platformer Toolkit, GMTK, 2022) to game jams focused on AI algorithms restrictions or ready made AI systems to practice with (AI and Games

Jam 2023, AI and Games, 2023). By allowing developers to test out a system in a more controlled or risk free environment we can create a great space for experimentation and, in the case of game jams for example, generate a good amount of different ideas that can later be inspiration or be combined by developers into other projects.

2.6. Research question and hypothesis

We have seen that the topic of AI systems in game development can be complex. New technologies entering the market generate more confusion on requirements and complexity removing space and freedom from developers to concentrate on creative tasks. Interactions between different developers and other stakeholders are also getting more time consuming and ambiguous to the point where older or more classical AI systems are not recognised as artificial intelligence anymore to avoid further complications in the exchange. This issue does not only hinder the development from a communication point of view but heavily influences research time and costs of development.

Throughout this thesis, I aim to address these challenges and propose a potential strategy by exploring the following questions:

> [Q1]
> What can be done to empower game designers in deciding or evaluating AI systems during their development journey?
>
> [Q2]
> What structure can be adopted for AI systems used in/for game development so that they form a knowledge base for anyone to start from?

3. Generating a structure to map and analyse AI

In this fundamental chapter I will explain my approach at the research thesis focusing mostly on the listing and mapping of existing algorithms. Following the challenges and complexity of the problem we have underlined in the past sections the aim of the chapter is to concentrate on the second research question (cf. §2.6.):

> What structure can be adopted for AI systems used in/for game development so that they form a knowledge base for anyone to start from?

Suggesting a consistent structure that can be used to organise or list any new system and map it to the rest. Due to the fast pace in development of AI systems in general and the multiple possibilities they offer when combined with creativity, the final list will not contain all the possible systems or combinations but should showcase enough material and examples to understand the underlying rules for this grouping task.

3.1. Analysis of the systems

My approach on the analysis of artificial intelligence aims at creating an environment that is precise enough for the subdivision and grouping of systems but not too detailed to completely disconnect automations that are deemed too simple to be seen as artificial intelligence. The main objective is therefore to create a procedure that can be applied to any type of automation used during game development or inside games. A good starting point is the recognition of the difference between “weak” and “strong” (cf. §2.3.) AI algorithms and what their implications are on a game development level. Due old definition of the term by Bourg where game AI systems are recognised to be only under the “weak” identifier (Bourg & Seemann, 2004, p.22) I feel that, due to the development of the last 2 decades, a more interesting subdivision is the one discussed by Thompson (Thompson, 2024) where are defined “symbolic” and “machine learning” AI algorithms or systems.
A symbolic or classical artificial intelligence system is built on symbols. Natural language, programming and other elements based on communication through symbols are combined with common logic. These systems are created on rules that people can recognise and understand. Every rule is rooted into logic and has been built manually by developers to analyse a problem and divide the data into states. The algorithm can then select a correct sequence of actions to move into new states and solve a specific problem.
Machine learning on the other hand, although it has existed since a long time, has only recently started to take the spotlight seeing fast development and starting to appear in a multitude of fields as well as in game development. The design of machine learning is structured around algorithms that learn to find a solution by looking at existing data of a certain problem. Following this premise, machine learning algorithms can reach the desired goal in different ways, with or without supervision of a designer, by mirroring existing previous cases or improving by trial and error as we have seen when talking about reinforcement learning techniques.
The idea is to divide and classify AI systems by applying measurable or distinguishable variables to them. When every algorithm is assigned all the values for every variable it will only be a matter of filtering, grouping or layering different variables to showcase and find connections and distinctions.

3.2. Defining variables

As we have seen in past chapters there are already many techniques and classifications that subdivide and distinguish systems in groups. The objective is to gather these classification strategies and assign them measurable values or distinguishable naming patterns. Variables are defined by any value, element or factor that might change between algorithms, this process should yield a concrete and consistent list of features for all the systems analysed allowing a procedural and automatic grouping or mapping of all systems. Filtering systems can also be easily applied. The majority of variables will divide AI systems by specific groups similar to what we have seen with symbolic AI and machine learning AI, while other variables will be defined by values selected from a spectrum and applied depending on the level of compatibility of a system with said variable.

Variables are defined by a name and a trait that will be assigned to it. When analysing systems all traits for all variables will have to be set accordingly. Variable traits might be subject to slight variations in time due to the appearance of new systems and thanks to feedback and discussion.
A big number of variables and values will be assigned by reference and combination of data from previous analysis and suggestions brought from researches such as *AI for Games* (Millington, 2019), *How AI is Actually Used in the Video Games Industry* (Thompson, 2024) and C*urrent AI in games : a review* (Sweetser & Wiles, 2002).
A trait is the value or family that can be assigned to a variable, due to the different methods of groupings that can be used to define and map algorithms traits can hold different types of data as follows:
Text traits are a data type defined by a string or text that is assigned to the variable, often following existing grouping patterns like a genre or family the AI system belongs to. An example could be for a "Seek" algorithm the variable "AI system type for task" ; its trait will be "movement".
Binary traits are a similar data type that can only be assigned one of two values "true" or "false". Following the previous example for the "Seek" algorithm the variable "can be modified and adjusted" will hold "true".
Value traits are assigned when the variable requires a numerical value assigned to it. For sake of simplicity all numerical values will be assigned on a scale from 1 to 10 where 1 is the least applicable to the current system and 10 is the most. With "Seek" we would therefore have a variable like "memory performance" as "1" since it requires low memory calculations to run.
Sometimes a combination or multiple traits can be assigned to a single variable, in this case each data will be divided by a comma.

System name	Variable name	Variable trait
Seek	Description	An algorithm used to move characters or objects trying to match the position of a target object. It is able to find the direction to the target and tries to reach it in the fastest way possible.
	AI system type for task	Movement (text trait)
	Sub type for task	Steering behaviour (text trait)
	Can be modified and adjusted	True (binary trait)
	Memory performance	1 (numerical trait)
	Platforms supported	PC windows, iOS, mobile, PS consoles, Xbox consoles, others (multiple text traits within one variable)

Tab 3 - Examples on different kinds of variables possible and respective traits.

As previously stated the main source of information and data is existing literature, basing variables on groups of AI systems that have already been established will grant a good level of consistency with other researches and the current state of knowledge. It is important to not work on generating a completely new categorisation system but to start on the existing state of the art and better define strategies to support it. Other methods for generating variables will be applied to cover or add to missing or lacking aspects of the existing groupings. I therefore propose to organise AI systems by complexity to further refine the categorization process. Although assigning a specific value of complexity can be challenging it can allow us to organise them on a spectrum and therefore enabling us for a more flexible analysis. Computational power required, adaptability, infrastructure needed can all be taken into account for this ranking method, allowing developers to better understand whether AI systems could fit or not their needs in the project.
It is also important to recognise that AI systems are rarely applied in isolation. In game design AI algorithms are often combined and integrated with other systems and technologies to achieve more complex tasks. For instance an NPC may rely on multiple systems to manage its movement, behaviour

and intractability. Moreover other connections can be identified by selecting a game genre, technique employed to generate the algorithm or task the algorithm is able to solve.
For example, a stealth game might combine pathfinding and movement AI with detection algorithms to create more immersive gameplay, while a strategy game may rely on decision-making AI alongside resource management systems. Understanding these relationships is essential for creating a complete and coherent work where AI systems are collaborating and functioning as one.

Another important source of inspiration on methods and approach is rooted in the research article and project *Locomotion Vault* (Di Luca et al., 2021), where a similar grading, filtering and mapping of systems was applied to research methods of locomotion into virtual reality applications. A very interesting combination of methods is explained in how they generated variables or "attributes" and assigned values during research, as well as a very interesting platform showcasing all results from the research allowing developers to find the best technique to fit their needs.

In the following subchapters I will explain and showcase the existing and my proposals for approaches on subdividing artificial intelligence algorithms. These processes will then provide us with the tools to create impactful and compatible variables to be applied.

3.2.1. System type

Subdivision of AI systems by "type" is a straightforward approach. Although the category might vary depending on the features analysed when defining it the process once this is clarified is more of assigning each system the correct trait.
One of the main subdivisions can be set on a topic of whether the AI algorithm uses a classical (symbolic) or machine learning structure. This can translate to a variable "AI system type for structure" and it will have a special kind of binary trait allowing only "symbolic" or "machine learning" as entries. These categories can often be further subdivided in smaller categories that will determine new variables originating from the main category. As an example "AI system type for structure" will need a sub variable "Sub type for structure" that will accept traits such as "deep learning" or "generative AI" in the case the main variable is set as "machine learning". Having these variables as separate entries will allow for a more controlled filtering.

Each AI system can also be defined by its employment during development as displayed in the article and connected video *How AI is actually used in video games* (Thompson, 2024). By checking these aspects of a system we can create new variables "AI system employment" and traits such as "AI that plays", "AI that creates" and "AI that models" and all the deriving sub variables and traits.

Variable name	Possible traits	
AI system type for structure	Symbolic/Classic	Machine learning
Sub type for structure or by technique	State machine, Decision tree, Behaviour trees, Rule-based systems, Pathfinding systems	Neural network, Generative AI, Reinforcement learning,

AI system employment	AI that plays	AI that creates	AI that models
Sub system employment	Non-player characters, Strategic opponents, Managing experience	Procedural generation, Real-time rendering	Online matchmaking

Tab 4 - Variable list from AI system type for structure, technique and employment.

Complexity or “performance” of a system can define a variable by itself with a numerical trait and can support a decision where multiple systems cover the same or similar tasks. *AI for Games* (Millington, 2019, pp. 14-15) proposes a good example on how variable traits can be applied when analysing each system and I will therefore base my standards on its example. A similar formula is applied for the variable referring to a system type for the task it performs, “AI system type for task”, following the great work conducted in the book it is possible to assign traits for common tasks performed by specific systems to new variable classes.

Variable name	Possible traits
Performance	Numerical value from 1 to 10 where 1 is very performant and 10 is the least.
AI system type for task	Movement, Pathfinding, Decision making, Tactical and strategic, Learning, Procedural content generation (PCG)
Sub type for task	Basic movement algorithm, Kinematic movement algorithm, Steering behaviour, Combining steering behaviour, Predicting physics, Jumping, Coordinated movement, Motor control, Dijkstra, A*, Hierarchical pathfinding, Pathfinding others, Decision trees, State machines, Behaviour trees, Fuzzy logic, Markov systems Goal-oriented behaviour, Rule-based systems, Blackboard architecture, Action execution, Waypoint tactics, Tactical analysis, Tactical pathfinding, Coordinated action, Action prediction, Decision learning, Naive Bayes classifiers, Landscape generation, Dungeon and maze generation

Tab 5 - Variable list from AI system type for task and sub type for task.

3.2.2. System genre

A game genre will often define what AI algorithms will be used for its development, rarely a game genre is defined by the system used since the genre is often one of the first choices a game developer makes when working on a new idea. Division in genres for games is already a common practice for gamers to find similar games to their likings and distinguish the experience they can expect from them. In the same way assigning this attribute to an AI system can help filtering out all others that would not fit into the desired experience and highlight the other algorithms that could be combined. While it is common to have certain AI systems fit perfectly into a game genre, like AI for resource management

would fit in a real-time strategy game, it can also happen to have AI systems that are flexible enough to be used in multiple games of different genres. The variable for “genre” can therefore be assigned multiple traits for the same system. For reasons of simplicity and to have a recognisable base to work with the game genres we are going to refer to are the same categories used by Steam, a very well known distributing platform.

Genre	Action
Sub genres	Arcade & Rhythm, Fighting and Martial Arts. First-Person Shooter, Hack & Slash, Platformer & Runner, Third-Person Shooter, Shmup
Genre	Role-Playing
Sub genres	Action RPG, Adventure RPG, JRPG (Japanese role play game), Party-Based, Rogue-Like, Strategy RPG, Turn-Based
Genre	Strategy
Sub genres	Card & Board, City & Settlement, Grand & 4X, Military, Real-Time Strategy, Tower Defense, Turn-Based Strategy
Genre	Adventure
Sub genres	Adventure RPG, Casual, Hidden Object, Metroidvania, Puzzle, Story-Rich, Visual Novel
Genre	Simulation
Sub genres	Building & Automation, Dating, Farming & Crafting, Hobby & Job, Life & Immersive, Sandbox & Physics, Space & Flight
Genre	Sports & Racing
Sub genres	All Sports, Fishing & Hunting,

	Individual Sports, Racing, Racing Sim, Sports Sim, Team Sports

Tab 6 - Variable list from AI system genre and sub genres.

The process of assigning a genre will also expose relations between multiple genres or systems that are not easily identifiable on a first look. Allowing the process of choosing the right system for a project and, vice-versa, getting the right idea to use a system in the best way possible.
Due to the continuous development of both AI systems and games with hybrid genres keep on adding new and inspiring combinations a clear mapping on this aspect does not only allow to identify traditional methods but will highlight opportunities for innovative mechanics.

3.2.3. System requirements

The requirements of a system to run can also influence the choice. Whether a system is able to run on multiple consoles or only on specific ones and if it needs backend structures to run or internet connection are all important aspects to consider. A variable such as “AI system requirements” can be therefore included and with it all its miscellaneous traits. This variable can be treated specially holding multiple binary traits that are automatically seen as true if present and as false if not present in the description of a system. Note that multiple traits can be true at the same time depending on the system analysed.

Variable name	Possible traits
System requirements	Is run locally, Is run on a server, Needs internet connection, Runs on PC, Runs on Playstation, Runs on Xbox, Runs on mobile, Runs on Handheld devices

Tab 7 - Variable list from AI system requirements.

3.2.4. Player immersion and level of control

A final point I find is important to help with understanding and choice of an AI algorithm is its level of immersion in a game. While an algorithm can be very good at his task it doesn't mean it will be “fun” for a player to play against. Although “fun” can be subjective it is important to understand that the algorithms used in games are often created with flaws on purpose to give the player more playability and a better experience. Having a very intricate AI driving the opponents cars in a racing game could result in the player always losing because of the perfect actions taken by the other vehicles and therefore not having fun. Having a variable that somewhat can show how disruptive a system can be in a game can be an opportunity for a developer to rethink or focus more on a certain system so that it is enhancing the player experience and not restricting it.

Player immersion	Numerical value from 1 to 10 where 1 is very disruptive and 10 is the least.

Tab 8 - Variable showcase on assigning AI system immersion.

3.3. Mapping the systems

The approach will result in a growing list of different AI systems that have all variables assigned with their traits. Thanks to this mapping, filtering or grouping these systems becomes a matter of automation,

each variable can be used to identify specific systems and combinations of variables can create maps or graphs for a more straightforward recognition. Learning about systems and their relation becomes easier since information becomes a clear structured library. In addition to explaining and assigning variables of course each system can be assigned other characteristics like the ones we discussed in a previous chapter (cf. §2.5.3.). Addition of description, examples and structural illustrations will complete the research circle, allowing the user to find, learn, understand and apply new knowledge on AI systems in an easier and more structured approach.
Of course how the information is shown and the user experience when searching for this information is a very important aspect as well that will be investigated in the following chapters focused on sharing knowledge (cf. §4.) and practical approach in the prototype (cf. §5.) or design thesis (cf. §7.).

3.4. Space for improvement

As stated previously the task of gathering all these elements as well as assigning variables and other characteristics can be a lengthy process, if not indefinite, due to the continuous development of new systems and alterations on the existing ones. Therefore a great focus on this approach should be on empowering anyone in the community to understand these principles and be able to participate and cooperate on the creation of this structure.
Assigned traits and variables might also vary through time with the introductions of better technologies, faster calculating systems. These variables should be checked and reassigned regularly to ensure the information is up to date. It is also possible for new variables to be added in case the proposed ones do not cover all useful groupings and traits a game developer could look for in a system. All these concepts will be discussed in the prototype and design approach where the methods can be tested and feedback implemented in a more concrete way.

4. Sharing knowledge

In this chapter I want to discuss the different methods viable for sharing the knowledge gathered addressing the first research question (cf. §2.6.):

> What can be done to empower game designers in deciding or evaluating AI systems during their development journey?

Based on results from research, form and interviews I have isolated 3 main approaches on sharing knowledge. I will then explain the benefits and drawbacks for each of them and relations between them. This chapter will act as a prologue to the prototype and later the design thesis, justifying the initial choices and methods taken.

4.1. Open-source library

An online platform or library offers a straightforward and efficient way to share the knowledge on AI systems to and between game developers. One of the strongest reasons for it is the accessibility it provides, both in terms of availability and costs. Game development requires a good investment of time and capital, an open-source library allows developers to access a good amount of information, at any time with little to no costs of time or resources.
The low cost of an open source platform can stimulate participation and promote it in the community. Developers or enthusiasts are able to access, comment or even modify the contents sharing their own personal experiences. Everything without the worry of licensing fees related to custom tools. This environment allows therefore more inclusion and a space for developers to find and innovate on AI systems and game mechanics even where budgets are low or even unexisting. Small indie teams or hobbyists can learn and experiment on new technologies cultivating creativity and novelty in the industry.
Besides the lower costs, benefits and availability an open-source library offers a great platform to build a community around it. Once a community starts to use, contribute and adapt the library and AI systems within it becomes a self-sustaining organism. The collective knowledge of all contributors can support the library, keeping it updated and adding to it over time. Developers can share improvements, optimizations and new ideas easily. Everything encourages transparency and educational growth. Developers of any level of expertise can learn from the examples, documentation and discussions. This learning environment is therefore able to support any type of developer, or any interested party, and offers a great opportunity for anyone to get resources.
Once a community has formed the sustainability of the platform is in an advantage too, the library would not rely solely on one entity for its maintenance or expansion. Contributions from a big range of users will take care of updates, additions of new AI systems and other tasks, ensuring the platform is up-to-date and functional. The distribution in responsibilities and maintenance means that the library can grow and improve without requiring a significant financial investment from any individual or specific group. Furthermore these circumstances allow for a broader range of ideas and approaches on AI systems themselves, the vision of multiple developers can be combined instead of being held back from the perception of a single one.
In conclusion, an open-source library can be an ideal approach for sharing knowledge in the game development environment due its low costs, potential for a growing community and self-sustaining nature deriving from it. It serves as a flexible, educational and evolving foundation to support game developers of all levels in learning, implementing or sharing knowledge on AI systems.

4.2. Peer-to-peer approach

A peer-to-peer approach for learning and sharing knowledge on AI systems offers a collaborative approach, accentuating the strengths of direct interaction, shared experiences and hands-on practice. In a peer-to-peer environment everyone is able to contribute, learn and share experiences. Practices like game development yield a loot of positive aspects from direct interactions. Feedback and testing is easier and more direct as well as new approaches and ideas. Developing and working on video games, with AI systems in particular, can present different challenges where the solutions are not always

possible to be found in books or tutorials. However developers that have already faced similar challenges and have found solutions or workarounds can share their experiences and developers who do not have any experience can bring a new perspective to the problems suggesting alternatives. Problem-solving, innovative approaches and workarounds are all subject and results of trial and errors, an environment that encourages conversation and exchange of these experiences will avoid developers to commit errors others already did and find solutions they wouldn't think about.
Peer-to-peer collaboration also encourages a sense of community, which can alleviate the more daunting or troublesome tasks and render game design more approachable and fun. Working with or close to other developers can alleviate the weight on someone's shoulders and support with encouragement from the community, helping with motivation. These circumstances do not only support newcomers and inexperienced developers to start their own journey but also more experienced ones with new ideas and updates on technology development.
A very good example of a working peer-to-peer environment taken from my personal experience is the Swiss Game Hub (Stark & Schwab, n.d.), a coworking space where developers can stay while working on their own projects. Every month playtesting events are organised inviting anyone to bring their project to test and get feedback from. Other events also take place to support networking and discussion between game developers.
These kinds of spaces really help with collaboration and support developers throughout their journey. Other than the technical learnings these environments encourage the development of other soft skills such as communication, teamwork, expressing ideas and learning. These competences can later be of great support when working in bigger teams where tasks are divided by one or more members. Peer-to-peer allows developers to learn these skills in a stress free environment allowing them to learn how to collaborate or discuss crucial points in case of future stressful events.
Finally, the casual nature of these interactions creates an environment where experimenting is encouraged. Due to the collaborative and non-hierarchical setting, developers can share ideas, test new approaches and receive feedback without the fear of judgement by the other parties. In game design events like Game Jams offer a playground where developers can create new ideas and in a later moment share and learn what others did differently. It is always amazing how many different ideas can be even from starting from a common topic or theme. Game Jams also show how peer-to-peer interactions can take place in person as well as in an online setting regardless.
In conclusion, a peer-to-peer approach, digital or analog, to learn AI systems in video game development offers significant benefits, collaboration, shared experiences and practical problem-solving as most prominent. By working together game designers can learn from each other and approach challenges in a more effective way.

4.3. Teaching and conferences

A pedagogic environment, teaching, workshops or conferences, offer a structured approach on learning AI systems in video game development. Experts and professionals are able to share knowledge and latest advancement in the field.
In a formal teaching environment, such as courses or workshops, students can benefit from guidance and direct instructions on concepts and logic behind the AI techniques. The experience from the instructor guides the learners into understanding the basics and essential topics with a clear and coherent structure that makes sure no topic is left unclear or uncovered. The students do not only learn about the systems but are able to understand the best approaches in a clearer and comprehensive environment.
Conferences, on the other hand, provide a complete experience where developers can learn from pioneering research, case studies and real world examples brought by researchers and professionals. Events like the Game Developers Conference (GDC) showcase a practical application of the systems in successful and well known games. Offering insights and inspiration to participants to innovate in their own projects. Workshops, demos and networking events related to these conferences also offer a great opportunity for hands-on learning and dialogue.
Both teaching and conferences encourage active participation and conversation. In a classroom setting, students can ask questions on topics they are not confident about and receive direct feedback, while conferences offer Q&A sessions with experts where more complex or specific questions can be answered. These environments encourage professionals to share their experiences, offering examples of

how AI systems have been successfully implemented in various games. These interactions offer a lot of possibilities, showing how AI systems can work in a real-world scenario and highlights potential areas for improvement or adaptation.
A community aspect in both these approaches is very well present. Connections between teachers and students or professionals during a conference can lead to new collaborations and learning. By keeping connections developers can continue to exchange ideas and stay informed on the latest developments in the field.
In short, teaching or conferences are a good environment for learning about AI systems in game development, they offer direct connections with experts and researchers, real world examples and networking opportunities that can lead to innovation and better understanding of the matter.

5. Prototyping

In this chapter I will explain and showcase a prototype that has been constructed to further test or confirm results from the past research. The prototype objective was to have a tangible example for developers to express their opinions, ideas and concerns.

The prototype was constructed on the basis of knowledge and contents that were found in *AI for games: third edition* (Millington, 2019) and will embody the base of what later will be the design segment for this thesis. The prototype will act as a material tool to showcase structures and practices that can help users to understand the topic of AI in video games and gain knowledge in the process. Due to his nature the prototype was created with limitations, other than having a low fidelity wireframe structure, so to highlight the contents and not the visual nature of it, the systems showcased are in a very limited amount and did not get all variables applied. This allowed for a better flow and a more direct focus on structure than features.
The testing was performed during a playtesting event at the Swiss Game Hub (Stark & Schwab, n.d.) in the city of Zurich. During the event multiple developers were able to showcase their own projects and get direct feedback from peers.

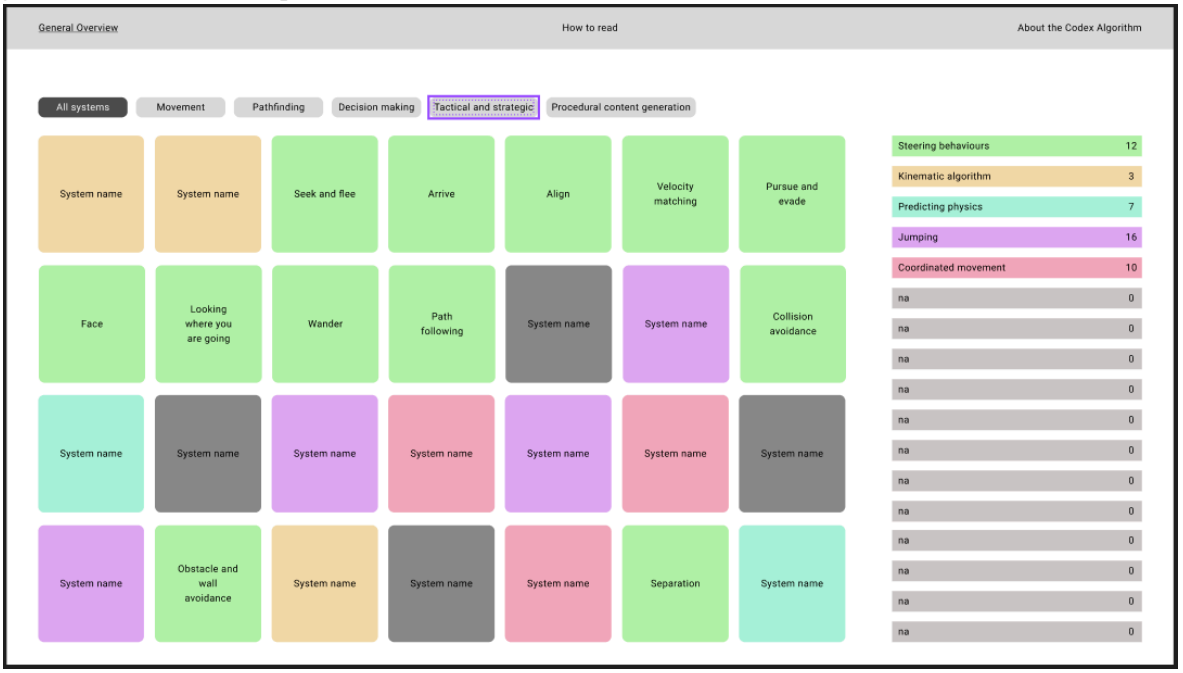


Image 8 - General Overview page of the prototype. More can be found in the appendix (cf. §10.3.).

The outcome from testing was overall positive, having a prototype to exhibit acted as a steady base to an open discussion on AI systems and the utility of such a service like an online library. This test did confirm a lot of the aspects that have been discussed in the previous chapters (cf .§2.5.2. and §2.5.3.).
An open-source approach, where anyone is able to participate, learn and add to the contents of a community, seems to be the best received. Especially in an indie or hobbyist environment.
The prototype underlined the importance of a good structure in the library, assigning variables and allowing for multiple filtering options, allowing users to generate examples by following a template to keep consistency, leaving space for rating or discussing AI algorithms. Many more aspects are also important and all can define if the platform can be useful or not in the game development environment.

6. Conclusion

In this chapter I will summarise the findings of this research and highlight the key discoveries from the study on AI systems used in video game development. I will reflect on how academic research, interviews and prototyping contributed to a better understanding of the concept of "AI" in this specific environment and its role in enhancing gameplay and experience for the players. Furthermore I will discuss how the proposed methods of subdivision of AI systems can reveal interesting patterns and assist the comprehension of the complex structures that can be found in games.

6.1 Insights

The current situation, following rescind discoveries on AI technologies, in the field of machine learning and big data structures, heavily influenced our perception on what an artificial intelligence implies. More people are now familiar with the concept of AI, often referring to large language models algorithms such as ChatGPT or generative models such as Midjourney. This resulted in a fast spread of the word "AI" modifying its perception on multiple disciplines, among them in game design.
Artificial intelligence has been around for a long time already in niche environments, often related to software development or data analysis. Although the complexity of AI systems have evolved with time, the past decades brought the most disruption into the recognition of such systems as "intelligence".
We have noticed this in the game design domain, where algorithms that were recognised in the past as AI are now appointed as a "simpler" type of algorithm that doesn't represent artificial intelligence behaviours.
This sudden change has brought a lot of confusion between developers, often struggling to be clear on what is the subject of a discussion when referring to "AI" as a term, leading some to even avoid the word completely to circumvent misunderstandings. It is therefore necessary to work on a better structure to fill what the term "artificial intelligence" holds in its meaning. Smaller groups and systems that all are recognised as AI but can differ in complexity, techniques or even structure between each other.
Categorisations, naming and grouping of these systems are already part of the game development culture due to a long development history concerning this type of algorithms and in general due to games becoming more varied following opportunities offered from the increasingly more complex technologies. Although these organisational methods are a good basis on understanding the underlying systems they often lack completeness to cover every possible variation and system.
Multiple approaches are also possible on facing this issue. Platforms like AI and Games (AI and Games, n.d.) started to offer an in depth analysis of systems with insightful videos and blogs. Research and books such as *AI for Games: third edition* (Millington, 2019) or *AI for game developers* (Bourg & Seemann, 2004) suggest a good structure to base everything on. While events like GDC (GDC, n.d.) or AI and Games Conference (AI and Games et al., 2024) started to offer an opportunity to learn more on the topic.
Information, although present, is still scattered in a multitude of platforms, researches and events. Making it very hard or even impossible to grasp on the whole picture for most of the community.

6.2 Conclusion

The increase of confusion due to overuse of the term "AI" and scattered information asks for a change. Due to the size of the topic it is a task that cannot be solved in one simple action, but with multiple projects already working for a better condition it is possible to act on it and offer a hand on approaching this issue.
To empower game designers (cf. §2.6. [Q1]), and others, into deciding or evaluating AI systems during development there are measures to be taken. Distribution of knowledge can be through direct conversation in a peer-to-peer approach where developers can directly exchange experiences and learnings from past events while discussing other possibilities and future technologies. Having the opportunity for experts in research or professionals to share their knowledge in classes or at conferences can also help with the spread of know-how and help with a more harmonious structure of knowledge as well as create communities that can support each other through the sharing of this knowledge.

Finally an open space that is accessible to anyone, independently of their current knowledge or economic status, where the community can share and learn from others will support creativity and negate unnecessary limitations.
On the note of the structure that can be applied to render the knowledge of AI systems as a good base to start from (cf. §2.6. [Q2]) the solution relies on a cohesive collection of the numerous already existing methods. The structure should then be flexible enough to be applied in any circumstance, but strict enough to keep consistency even if the person applying it might change. Community collaboration will also be an important aspect due to the amount of material and knowledge that can be gathered from multiple people in comparison to what an individual is capable of.
The methods suggested in this thesis as well as the variables and traits related should act as an inspiration for further development on the topic. Different approaches on assigning variables will generate more opportunities to recognise, research or get inspired by existing games and systems. Using variables as a method to filter in or out systems in a complex map can allow for new approaches on game development and research in general. Systems that have always been seen as relying on specific genres or useful to solve only specific tasks could be rediscovered and assigned new openings in interesting mechanics. The limited time for this research did not allow further exploration on the opportunities the method could offer, although further development will follow in an outlook on the design thesis (cf. §7.).

7. Outlook and Design thesis

This research offers a lot of insights into game development, the objective was mostly to address the challenges faced on behalf of the AI topic and suggest different possible measures to be taken in order to clear out the current state. In this chapter I want to discuss lacking points and what my proposition will be regarding the practical approach of this thesis.

7.1. Outlook

This thesis has only scratched the surface of the complexities of AI technologies, especially in the context of game development. While we've uncovered some interesting aspects and structures, fully diving into this topic would take a lot more time and resources. Additional variables and traits would need to be considered and explored to truly capture the full picture of AI systems in gaming. The clearer we can make the distinctions between different systems and algorithms, the easier it becomes to create a solid, functional framework for understanding them.
Involving more experts and professionals in the conversation would go a long way in making the research more credible and reliable. And we can't overlook the power of community. Bringing together people who share a passion for AI and game development could have a huge impact, as collaboration often sparks new ideas, insights, and solutions. Building or working with a community could be a key step in moving forward.
This thesis can serve as a catalyst for that process by identifying gaps in the current state of research, creating connections for developers, and offering a solid platform where experts can come together to share their ideas and knowledge.

7.2 Design thesis

The objective of the design thesis is to address the desired impact found during the research thesis. Through prototypes and research the aim is to experiment multiple options to reach the desired goal and choose one or various that seems the most promising and reliable. Following what was discussed in the state of the art it is important to create something that is really needed and new. The needs and opinions from experts will act as a base to work the prototypes on, while personas and user journey will help in focusing the project on a minimal viable/likeable product (MVP MLP). The design thesis wants to experiment and find possible solutions but it is plausible that it will not be a definitive one, since multiple approaches and methods could be used to address the problem. The aim is to show the opportunities one method could create.
Combining the theories from the research thesis and findings from prototypes it should be possible to find a way to share the knowledge gathered with the stakeholders, answering the research questions, both focused on human stakeholders and non-human ones, will surely improve the chances of the task.

7.2.1. Design methods

ISO
The primary design method was mostly based on the ISO design method. Coming from the International Organisation for Standardization, from where it takes its name, this method is very flexible for any kind of project and team size. Using the ISO method is pretty straight forward: first it is needed to identify the problem or objective of the development process, after assessing that we have the necessary resources to approach the project it becomes more of a circular structure where research on the context is followed by user requirements, a design solution is then proposed and subsequently evaluated against the requirements. If the design solution does not meet the requirements the new context is then reevaluated and the circle continues. This until a system that satisfies the user and organisational requirements is found. Thanks to the looping structure of the method it is possible to easily go back and forth between research and prototyping, very useful in a starting phase where ideas are not yet clear and a schedule is not yet set or too tight.Thanks to the ISO design method I was able to find an area of interest and later pinpoint the focus of the current thesis.
Due to the circular structure the ISO method is however dangerous on a schedule, since it is not possible to calculate the length of a loop in advance or how many loops are needed to get the final system it is impossible to strategize on a tight timeline. With the topic and research question being defined for my

thesis and the requirements for the design aspect set, in addition to the computational nature of the project, I switched to a more compact and structured method that still leaves some space for experimentation: a modified version of the Scrum method.

Scrum modified

The scrum method, often used by teams, helps to structure and manage a project by creating a backlog to reach a desired product goal. This backlog is then subdivided into smaller packages that are run through "sprints". Sprints are a set amount of time that is given to reach the package goal, if at the end of the sprint the goal is not reached an increment can be added. On completion of a package the next is set in a sprint and so on until the whole product backlog is completed and the main goal is reached. During a sprint the situation might be very similar to an ISO or circular method but thanks to the specific task and time limit there is way less danger that development might be delayed.

What I plan to do is to adapt the scrum method to have an adapting backlog. Keeping the sprint time around 1 to 2 weeks can help me to define the state of development pretty often and decide on the next package to approach and in case add, modify, switch or even remove packages from the backlog.

Prototyping and testing

Prototyping will mostly be influenced by a combination of the shown design methods and will be influenced by constant testing and the consequent feedback. Testing will be a key component of this stage allowing evaluation and modification of structures when needed. The prototyping method will follow the described scrum method with the possible addition of a bottom up approach led by research in case of new challenges allowing to test multiple possibilities before focusing on the best choice.

7.2.2. Personas

Personas are generated by taking into account the form and interviews gathered during research. These primary and secondary personas might be subject to variations during the development of the design thesis.

Indie game developer

Game developers are the primary persona and main user and target group. This persona should represent what its attributes and needs could be.

Don, 27

- Worked in the game design industry for 3 years
- Is interested in learning new ways to control NPCs in his games.
- Knows coding but often takes other tasks in the development team.
- Doesn't have much time for research since the project has a tight schedule due to a collaboration with a producer.

Student
This persona should represent anyone that could have an interest in the topic, such as students, enthusiasts, gamers, producers and so on. These groups might have less experience in game development but still an interest in the structure of games.

Alex, 17
- Plays video games everyday after school.
- Is interested in the video game environment and wants to learn how to develop games.
- Has a bit of knowledge of coding due to a class in highschool.
- Doesn't have any experience on any game engine.

7.2.3. User journey

Development Start
The journey begins when a developer or a team decides to start a new project or develop a new game. They know they'll need AI systems or algorithms but haven't defined the details yet.

Getting an Idea
The team brainstorms various ideas for new mechanics. For instance, they might want to create NPCs with advanced decision-making capabilities, or a procedural level generator. They settle on a rough idea based on the game's needs.

Researching Possibilities
Next, the team dives into research, looking into existing AI techniques that could fit their concept. They read papers, and review what has been used successfully in similar games. During this time they stumble on the online AI library (project of Design thesis). In the library they easily find interesting materials on AI systems they could use to achieve the desired mechanics.

Testing Possibilities
Once a few AI techniques are identified, the team tests different methods. They might implement basic prototypes or use ready-made AI frameworks from the game engine of choice to experiment and see how each one works in the context of their game.

Finding the Best Fit
After testing, the team identifies which AI system works best for their game. They choose the one that balances efficiency, complexity, and meets the gameplay needs.

Extra Steps:

Discussing Better Ways
The team evaluates their choice, asking if there are better approaches. They brainstorm possible improvements or alternative AI systems that may enhance performance. In this stage they might go back to the AI library to research other possible options.

Fine Tune and Playtest
They fine-tune the AI system based on internal testing and feedback from playtesters. Adjustments are made to ensure the AI behaves in a way that enhances the player experience.

Improve Performance
With feedback in hand, they work on optimising the chosen AI, making sure it runs efficiently, fits within the game's constraints, and delivers the desired experience.

Share Findings
Finally, the team documents their process and shares their findings directly into the AI library to help other developers benefit from their experience.

7.2.4. Design intervention

The main objective of the design thesis will be focused on showing a proof of concept or MVP for what can become an AI library later. The project should show the opportunities that such a platform can offer to the users and a viable structure to build upon. Findings from the thesis will be a starting point to define the necessary features of the platform as well as additions and modifications that will arise during development due to testing and feedback. Starting from the prototype (cf. §5.) the project should grow and get a more structured shape in its contents and UI design. User research will be a constant during development to guarantee the usability of the final product.
The platform is going to showcase one of the multiple possibilities on confronting the topic of AI in gaming and other similar or existing possibilities will be analysed to gather as much data as possible on how the task can be achieved.

9. References

9.1. Glossary

AI: artificial intelligence
NPC: non playable character
Mod: game modification, normally created by fans of a game it is an extra unofficial content that can be added to a game's file. It can vary from different graphics, extra missions and much more.
AAA or triple-A: is a term used in the video game environment to define projects, games or companies with extremely large budgets, marketing and developers teams. Generally AAA projects are aimed at getting huge revenues over time.
Indie developer: a term referring to a person or group of people working on a game without being backed by ventures or investors. Typically working on projects with a smaller budget and a more creative control over their product.
Indie game: an independent video game is a game developed by indie developers, often focused on innovation, experimentation and other risks that triple A companies cannot afford.
Asset: any art, 3D model, sound, music or artistic content used for the development of a game.
Game Jam: an event or time frame where multiple developers have to follow a certain amount of rules and create a game from scratch. Normally a theme or keywords are requisites and have to be implemented into the mechanics or story of the game.
Game engine: a software environment prevalently created for video games production, it often contains useful and commonly used libraries to help with development.
Bottom up approach: in game design it often refers to an open approach where multiple mechanics or features are created and tested in a sandbox environment. This allows the developers to quickly test what works well together without a specific structure yet.
Pseudo code: a coding structure that is not limited to one programming language, often aims at showcasing the logic needed for the code to work.
Machine learning: a subfield of artificial intelligence that allows an algorithm to improve their performance autonomously by being fed big data to train on.
Big data: large and diversified amount of information that grows at a high rate. Normally defined by the speed this data grows, the size of the total amount that exceeds normal means of storage and the diversity it includes (varying from text, images, videos …).
Supervised learning: training a model using labelled data to make predictions.
Neural network: a computational model inspired by the human brain, often used for pattern recognition and prediction.
Deep learning: a sub field of machine learning using neural networks to analyse data.
TTs: text to speech, an algorithm able to convert written text to artificially spoken audio.
LLMS: large language models, machine learning algorithms trained on big data specifically to replicate text and human writing.
Reinforcement learning: a sub field of machine learning that focuses on training the software into taking the optimal decisions for a given condition. The training consists in a trial-and-error type of simulation.

9.2. List of images

Image 6 - Image showing an example of the pseudo code presented in AI for Games (Millington, 2019, p.49).
Image 7 - Abstract example of an AI behaviour presented in AI for Games (Millington, 2019, p.49).
Image 8 - General Overview page of the prototype. More can be found in the appendix (cf. §10.3.).

9.3. List of tables

Tab 1 - Form on AI in games, showing what is perceived as AI in the game development environment.
Tab 2 - Form on AI in games, showing what tools are most used for research by game developers.
Tab 3 - Examples on different kinds of variables possible and respective traits.
Tab 4 - Variable list from AI system type for structure, technique and employment.
Tab 5 - Variable list from AI system type for task and sub type for task.
Tab 6 - Variable list from AI system genre and sub genres.
Tab 7 - Variable list from AI system requirements.
Tab 8 - Variable showcase on assigning AI system immersion.

9.4. Bibliography

AI and Games. (n.d.). *Welcome to the AI and Games Jam 2023*. itch.io. Retrieved September 11, 2024, from https://itch.io/jam/aiandgames2023

AI and Games, Half-Space Consulting, & Goldsmiths University. (2024). *AI and Games Conference*. AI and Games Conference: Home. Retrieved September 23, 2024, from https://www.aiandgamesconference.com/

Amazing Event AG. (n.d.). *Zurich Pop Con*. Zurich Pop Con. Retrieved September 23, 2024, from https://zurichpopcon.ch/en

Antonick, R., Friedman, J., Johnson, R., & Bethesda Softworks. (1989). *John Madden Football* [Apple II, MS-DOS, Commodore 64/128].

Atari. (1972). *Pong* [Arcade machine, later ported to the Home Pong console.].

Atari. (1974). *Qwak!* [Arcade].

Beyond Software. (1991). *Tony La Russa Baseball* [Commodore 64, Sega Genesis, MS-DOS].

BioWare. (2007, November 20). *Mass Effect* [Xbox 360, Microsoft Windows, PlayStation 3].

BioWare. (2021, April 30). *Mass Effect: Legendary Edition* [PlayStation 4, Xbox Series X/S, Xbox One, Microsoft Windows, Project xCloud, GeForce Now].

BostonDynamics. (n.d.). *Starting on the Right Foot with Reinforcement Learning*. Boston Dynamics. Retrieved August 5, 2024, from https://bostondynamics.com/blog/starting-on-the-right-foot-with-reinforcement-learning/

Bourg, D. M., & Seemann, G. (2004). *AI for Game Developers*. O'Reilly.

Brackeys. (n.d.). *Brackeys* [YouTube channel that offers video tutorials on game design practices]. Brackeys. Retrieved 09 27, 2024, from https://www.youtube.com/@Brackeys

Buchanan, B. G. (2005). A (Very) Brief History of Artificial Intelligence. *AI Magazine*, *26*(4), 53-60.

candlesan. (2023, March 23). *Can AI code Flappy Bird? Watch ChatGPT try* [video]. YouTube. Retrieved August 06, 2024, from https://www.youtube.com/watch?v=8y7GRYaYYQg

Chunsoft. (1990). *Dragon Quest IV* [Famicom/NES].

Crevier, D. (1993). *AI : the tumultuous history of the search for artificial intelligence*. Basic Books.

Digital Illusions Creative Entertainment. (2018). *Battlefield V* [PlayStation 4, Microsoft Windows, Xbox One, Project xCloud, GeForce Now].

Di Luca, M., Seifi, H., Egan, S., & Gonzalez-Franco, M. (2021). *Locomotion Vault*. Locomotion Vault. Retrieved September 10, 2024, from https://locomotionvault.github.io/

EA Sports. (2022, July). *Pitch notes FIFA 23 gameplay deep dive*. ea.com. Retrieved August 05, 2024, from https://www.ea.com/it-it/games/fifa/fifa-mobile/news/pitch-notes-fifa-23-gameplay-deep-dive

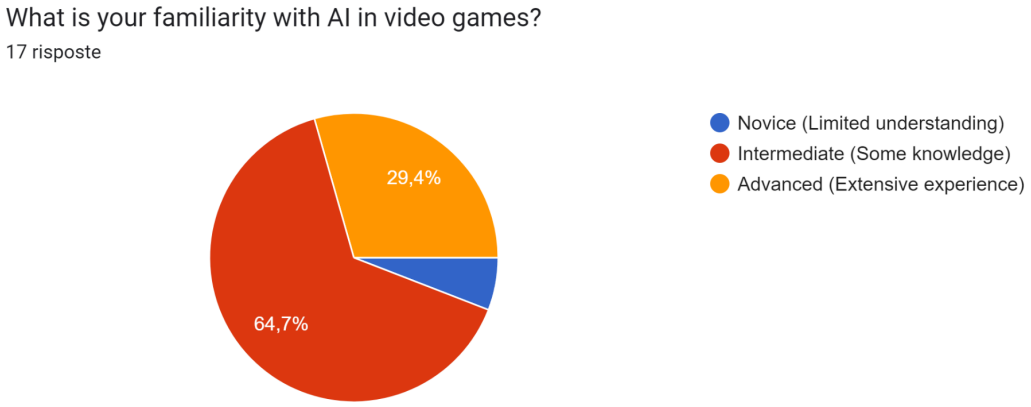
Epic Games. (n.d.). *Unreal Engine*. The most powerful real-time 3D creation tool. Retrieved September 10, 2024, from https://www.unrealengine.com/en-US

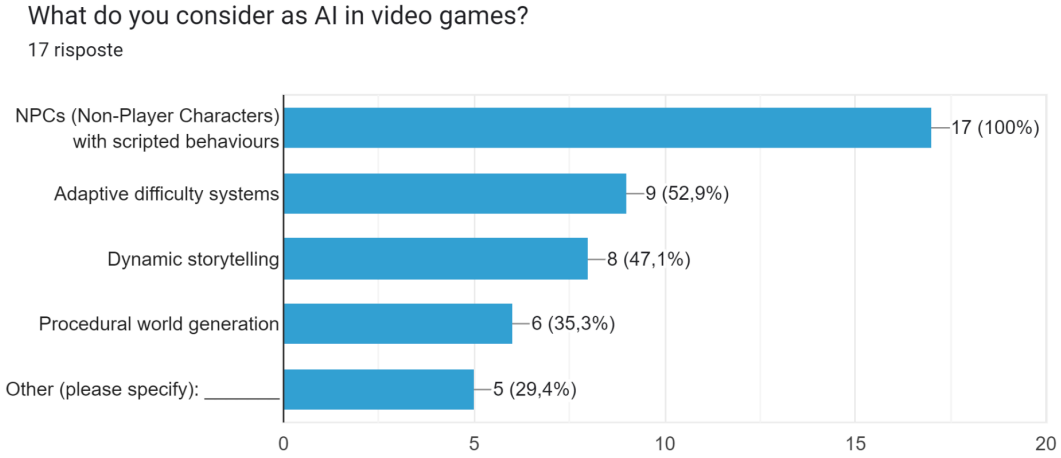
Fantasy Basel. (n.d.). Fantasy Basel. Retrieved September 23, 2024, from https://fantasybasel.ch/it
Ferranti International PLC. (1951). *Nim* [Nimrod (computer)].
Filipowicz, A., Liu, J., & Kornhauser, A. (2017, January). *Learning to Recognize Distance to Stop Signs Using the Virtual World of Grand Theft Auto 5*.
Fuchs, F., Song, Y., Kaufmann, E., Scaramuzza, D., & Dürr, P. (2021, February). *Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning* (preprint version). IEEE ROBOTICS AND AUTOMATION LETTERS.
Gas Powered Games. (2010). *Supreme Commander II* [GeForce Now, Microsoft Windows, Xbox 360, macOS, Mac OS Classic].
Gates, B. (2023, December 21). *This year signaled the start of a new era*. linkedin. Retrieved July 06, 2024, from https://www.linkedin.com/pulse/year-signaled-start-new-era-bill-gates-qbpfc/
GDC. (n.d.). *Game Developers Conference*. Game Developers Conference (GDC) | The Game Industry's Premier Professional Event. Retrieved September 10, 2024, from https://gdconf.com/
GitHub. (2008). *GitHub*. GitHub: Let's build from here · GitHub. Retrieved September 23, 2024, from https://github.com/
GMTK. (2022). *Platformer Toolkit by Game Maker's Toolkit*. Game Maker's Toolkit. Retrieved September 11, 2024, from https://gmtk.itch.io/platformer-toolkit
Godot. (n.d.). *Godot*. Godot Engine - Free and open source 2D and 3D game engine. Retrieved September 10, 2024, from https://godotengine.org/
Halmos, P. R., Moise, E. E., & Piranian, G. (1975, May). The problem of learning to teach. *The American Mathematical Monthly*, *82*(5), 466-476. 10.2307/2319737
Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., Duke, C. S., & Porter, J. H. (2013, March 12). Big data and the future of ecology. *Frontiers in ecology and the environment*, *11*(3), 156-162. 10.1890/120103
Hello Games. (2016). *No Man's Sky* [PlayStation 5, Nintendo Switch, PlayStation 4, macOS].
Hodges, A. (2004). *The Essential Turing* (B. J. Copeland, Ed.). Clarendon Press.
Jackson, P. (1999). *Introduction to Expert Systems*. Addison-Wesley.
Kombucha. (2018, November 13). *Valve using machine learning and deep learning to catch cheaters on CS:GO (794 words) - Technology and Operations Management*. Digital Data Design Institute at Harvard. Retrieved August 5, 2024, from https://d3.harvard.edu/platform-rctom/submission/valve-using-machine-learning-and-deep-learning-to-catch-cheaters-on-csgo-794-words/
Lague, S. (n.d.). *Sebastian Lague* [YouTube channel that offers insights on complex coding projects related to game design.]. Sebastian Lague. Retrieved 09 27, 2024, from https://www.youtube.com/@SebastianLague
McCorduck, P. (2004). *Machines Who Think: A Personal Inquiry Into the History and Prospects of Artificial Intelligence*. Taylor & Francis.
Microsoft. (n.d.). *Project Malmo*. Microsoft. Retrieved August 5, 2024, from https://www.microsoft.com/en-us/research/project/project-malmo/
Millington, I. (2019). *AI for Games*. CRC Press, Taylor & Francis Group.
Namco. (1980). *Pac-man* [Arcade].
Nvidia. (n.d.). *GeForce RTX | Ultimatives Raytracing und KI*. NVIDIA. Retrieved September 23, 2024, from https://www.nvidia.com/de-de/geforce/rtx/
Ocelot Society. (2016). *event [0]* [Microsoft Windows, Mac OS Classic].
Proxima. (2023). *Suck Up!* Suck Up! Retrieved July 19, 2024, from https://www.playsuckup.com/
Retro Game Mechanics Explained. (2019, July 13). *Pac-Man Ghost AI Explained* [video]. Retrieved August 21, 2024, from https://www.youtube.com/watch?v=ataGotQ7ir8
Russell, S., Graetz, M., Wiitanen, W., Saunders, B., & Piner, S. (1962). *Spacewar!* [PDP-1].
San Diego Comic Convention. (n.d.). *Homepage*. Comic-Con.org. Retrieved September 23, 2024, from https://www.comic-con.org/cc/
Simmons, M. (1975, October 8). Bertie the Brain programmer heads science council. *Ottawa Citizen*, 17.
Stark, D., & Schwab, P. (n.d.). *Swiss Game Hub*. Swiss Game Hub: Home. Retrieved September 19, 2024, from https://www.swissgamehub.com/

Sweetser, P., & Wiles, J. (2002). Current AI in games : a review. In *Australian Journal of Intelligent Information Processing Systems* (Vol. 8, Issue 1, pp. 24-42).
Taito. (1974). *Speed Race* [Arcade].
Taito. (1978). *Space Invaders* [Arcade, later multiple Atari consoles].
Thompson, T. (2024, February 28). *How AI is Actually Used in the Video Games Industry*. AI and Games. Retrieved July 20, 2024, from https://www.aiandgames.com/p/how-ai-is-actually-used-in-the-video
Turing, A. (1950). *COMPUTING MACHINERY AND INTELLIGENCE*.
Unity. (n.d.). *Unity*. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. Retrieved September 10, 2024, from https://unity.com/
Valve Corporation. (2013). *Dota 2* [macOS, Linux, GeForce Now, Microsoft Windows, Classic Mac OS].
Weizenbaum, J. (n.d.). *AI Timeline: Key Events in Artificial Intelligence from 1950-2024*. The AI Navigator. Retrieved July 5, 2024, from https://www.theainavigator.com/ai-timeline
Yob, G. (1973). *Hunt the Wumpus* [PC].

10. Appendix

10.1. AI in video games form results



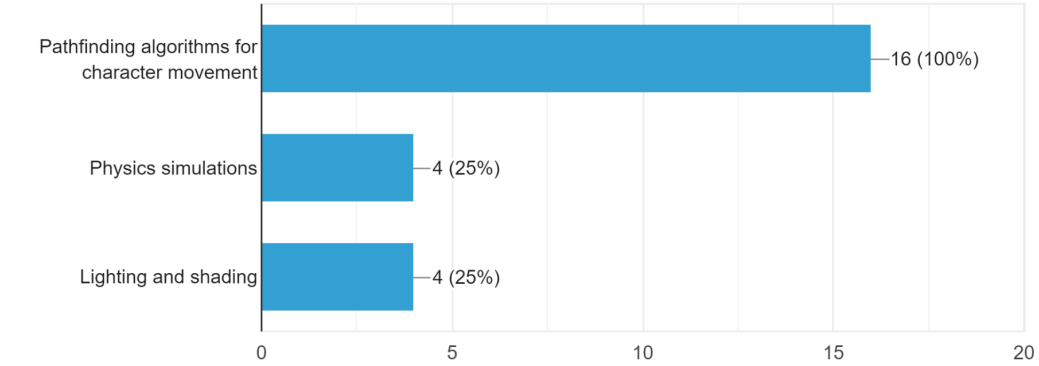


If selected others please specify

- Opponent behaviour in eg strategy games that aren't attached to characters in particular.
- AI Director, Machine Learning (e.g. Generative dialogue)
- Generative AI models such as LLM (Large language models) and SD (Stable diffusion) which are used to generate text and images, good example would be Chat GPT
- The other options may be, if machine learning is used
- Usage of LLMs and other Machine Learning Tools for development

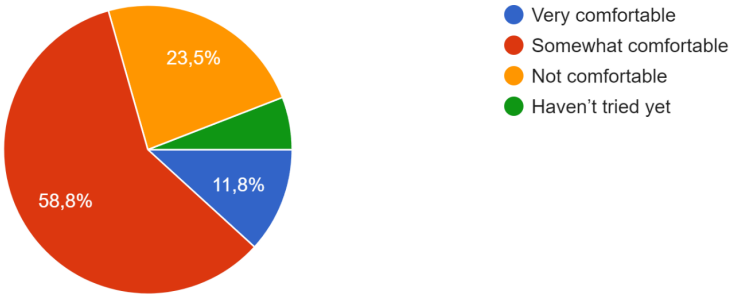
Which of the following aspects of game development involve AI?

16 risposte



How comfortable are you implementing AI systems in your games?

17 risposte



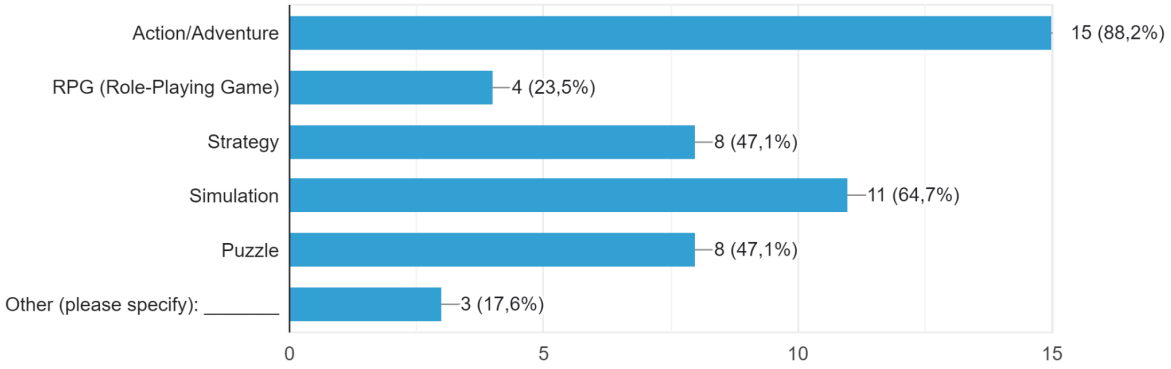
Could you explain why?

- If the game engine provides nice structures it is quite easily feasible. If not, it is more of a hustle.
- I've done it a bunch of times but it's always hard to make an AI system that works well and doesn't have exploitable issues or weird behaviour.
- well I can't really code ^^
- I don't code nowadays, but back when I did, I used to write simple NPC/enemy behaviours that I'd consider "AI", albeit a very, very basic version
- I haven't tried implementing AI systems many times.
- I have done AI for games as part of my professional experience
- Well I don't feel any "discomfort", either I know how to or l research how to and then do it. Quite straightforward.
- I'm a researcher not a game developer
- Creating an AI for a character in a video game is a rather simple process, people often might believe that making an NPC to do something is a huge undertaking when in reality it often involves a few lines of code or nodes thrown together to make a logic. Heck even if we take a Cube and write code where it is always moving towards the player location we now have an AI which probably would take a few minutes to do and now you have the entire enemy system of games like Vampire survivors implemented. It really depends on the scope of the project.

- Definition "NPC behaviour": I mean yeah obviously, Definition "Machine Learning": If it isn't a substitute for content that could be built better by a human
- Too controversial in some cases
- I mostly work on the design / art side of games, which is why I don't have much experience in implementing AI in my games
- Using AI meaning some kind of bots/ NPC systems and similar and not meaning anything Machine Learning related: I have written a lot of different AI Systems over the years.
- Limited knowledge with some basic starting points.

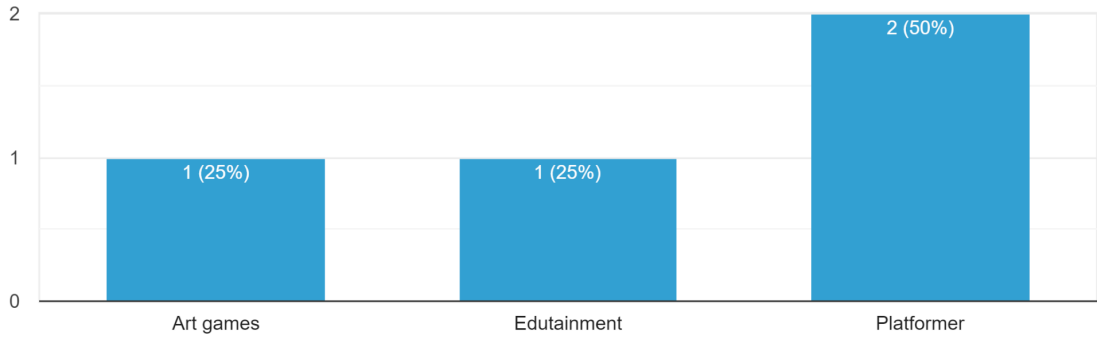
Which game genres have you worked on? (Select all that apply)

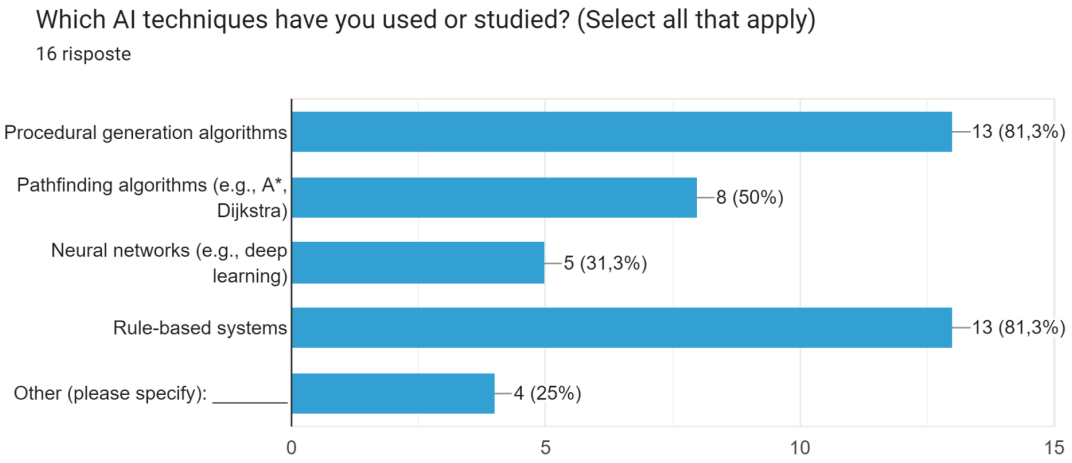
17 risposte



If selected others please specify

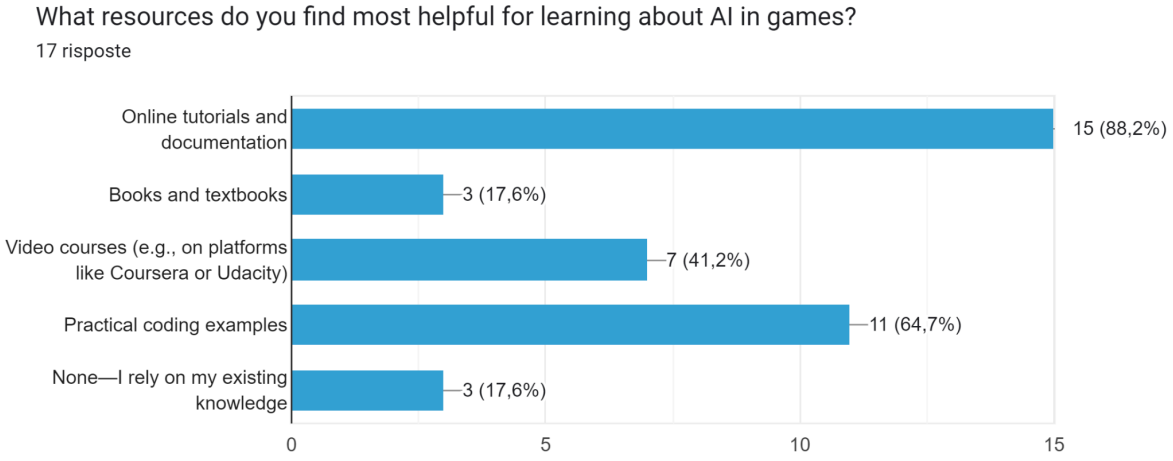
4 risposte

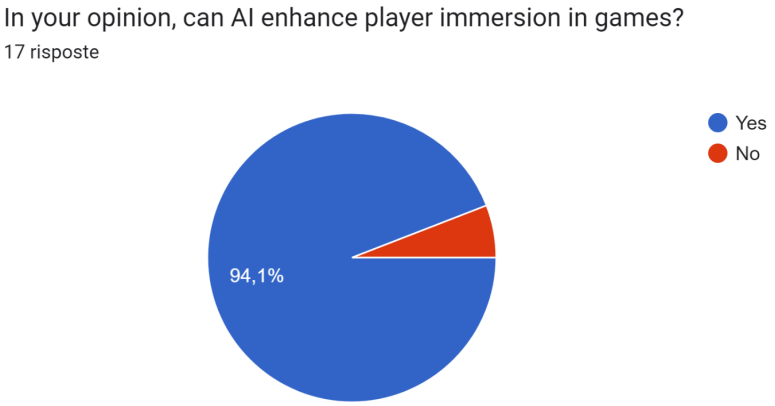




If selected others please specify

- Genetic algorithms, Utility Systems
- Symbolic AI e.g. BT, State Machines
- When making games there is little to no need for actual algorithms as almost every game engine has built-in NPC tools. I personally also believe that algorithms and neural networks etc. take it to such a high level that even triple A game studios most likely aint implementing massive systems for AI logic.
- State Machines, Tree Behaviours, Task Systems



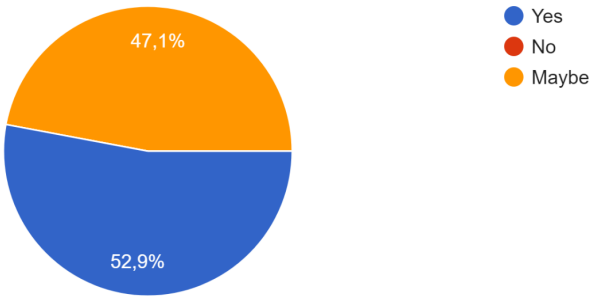


Could you explain why?

- It adapts to the player so it has a strong effect on immersion
- Even the conventional idea of AI in games already offers different "paths" for different playthroughs.The more of those the better, so it doesn't feel like a scripted experience.
- Making the world react to player behaviour in a sophisticated way makes it feel more alive.
- If it is about creating an experience that feels more rich, unique to players or real, I think it can help. Everything that feels scripted is not good for player immersion.
- Essentially, anything that simulates "something going on" has the potential to make games interesting and more realistic and immersive. NPC behaviour is definitely one of those things
- It broadens the variety of reactions to player input and thus makes players feel like they have more or more meaningful agency.
- Complex behaviour adds interest for players
- Game is all about systems interacting with other systems and the players, AI or "autonomous systems", can dynamically react by setting rules in both symbolic fashion or with machine learning.
- Better AI means that it's more human-like, and I think systems that behave like a human are more immersive. They're not always more fun though.
- Without AI the world would feel empty, think games like GTA, every vehicle on the street is an AI running on some code on the CPU. The player can interact with those and sometimes even forget that it is indeed a piece of code with some behaviour traits and yet it feels like we are interacting with something that feels real.
  One great example is Red Dead Redemption 2 where you can help a random person to learn how to hunt in a game... just think about it for a second... a real human being is teaching a piece of NPC code how to hunt in a game which sounds absurd on paper, but it actually makes the player feel like "I'm doing good" and "I can feel good for myself", that is the power of AI being able to enhance the experience.
- Bad NPC behaviour breaks immersion
- Adaptability and UX adjustments (difficulty, accessibility)
- AI can allow certain elements in a game to behave in a less predictable, more complex manner which can help with player immersion.
- Again, just meaning the classical definition, not anything ML related: yes, it's a often basic feature of single player games and greatly enhances immersion and the power of the magic circle if players feel like they engage with a kind of „intelligence“
- I think AI can provide different elements and interesting permutations that keep the players engaged.

Do you think a platform that maps AI systems for game development, with examples and relations between each others could be beneficial for game development?

17 risposte

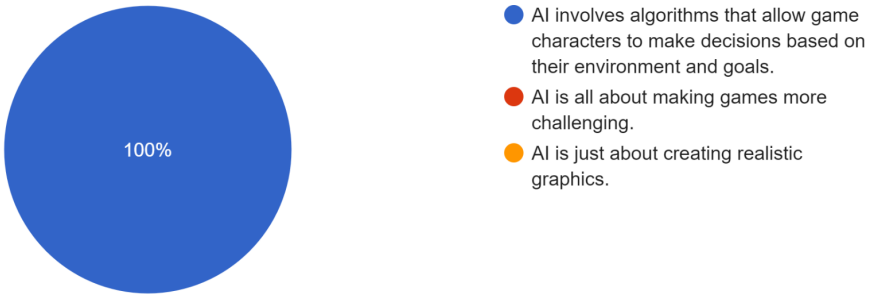


Could you explain why?

- Well I know some of the tool I use are based on AI but I don’t have a overview knowledge about how it really works
- It could be interesting to get an overview. But depending on what kind of game I am developing I would choose a specific AI system and would mainly research this.
- Having an overview of what techniques are out there, and what they're suitable for, would be useful.
- having an overview of possibilities is always good
- generally yes, because I'm sure some people would benefit from easy ways of learning about this stuff, but it wouldn't necessarily be useful/relevant to me nowadays due to my shifted job focus
- I think the information already exists, in books and articles
- Not sure HOW and WHY the mapping platform will be implemented, you only specified that"if it does" i.e. WHAT it does.
- Basically platform that does map AI systems (the "what"), and is implemented in such a way that its purpose (it's WHY) and implementation (the HOW) is well executed then yes of course; In the end if the developers gets another tool that helps with the workflow from early concept and feature testing (exploring new ideas), later concrete implementation, debug & testing and lastly maintenance and extensions/iterations then it would clearly be beneficial.
- Am not sure i understand what is being asked, but if by this you mean that we can use automated tools to quickly build an AI system out of nothing with little effort, then that is how game engines already work. UnrealEngine for example provides a drag n drop volumetric nav mesh. By simply dropping a cube into your game the map gets fully calculated for the AI to travel and there are other tools that make NPC/AI creation for a game extremely easy.
- More Resources = lower barrier of entry
- As a Designer, my knowledge of AI tools for games is very limited, so I would definitely find a comprehensive overview helpful
- Depends on your definition of the term AI and level of abstraction talked about; e.g are we talking about programming patterns? design choices? philosophy of ai behaviour in games? I think it would need a lot of depth to actually help or just focus on one of the mentioned topics.
- Such a platform could help with understanding and learning, especially for people not in the field or limited experience and knowledge.

How would you explain AI to someone who is new to game development?

14 risposte



Do you have any other comments you would like to add?

- Really cool topic. I'm actually curious to know more about it :)
- I wrote that when I say AI in a games context, I mainly mean things like NPC behaviour. But really, I think that what is/isn't AI in programming is super vague and can include everything and nothing. Personally I wouldn't use the term AI to talk about procedural level generation, but in the end that's also just an algorithm that's supposed to help create something that's interesting to play/spend time in, same as with NPC behaviour.
-
- I think it's impossible to find an "objective" answer to what is/isn't AI in games programming, and it's fine that people have different definitions of that.
- In the previous explanation, I would say that AI is not just about characters but about any actor or system that has to make decisions.
- Regarding the previous question, none of them. AI is just another interacting system in a game that is full of other interacting systems with the purpose of making it interesting and fun for the players and helping with the development for the developer.
- Well for the "How would i explain AI to someone", AI is just a piece of code made to do "something" and we can build onto that "something" as much as we want depending on the scope of the project, without AI our games would just be lifeless.
- Ai is also for skipping some part of creative process
- This has been somewhat confusing to me because I did not fully get whether you mean bots and similar or the trendy ML term. I hope you still can use my data :)

10.2. Expert interviews transcriptions

Name of interviewed person: Allison Crank
Profession: UX Designer and Researcher
Date of interview: 20.08.2024
Duration: 32 min

Interviewer:
Right now I'm doing research on AI systems used during game development and I noticed that when developing games, we are often confronted with the need of a system for specific tasks that they could do autonomously. And I also noticed that the world “artificial intelligence”, especially in game development right now, is getting used for more and more of this type of systems to the point where even between game developers we don't have the understanding of what this world is anymore. So just for introduction I would like to ask you how you would define artificial intelligence or AI? When I tell you “AI”, what is the first thing that comes to mind?

A:
It's either an LLM or some sort of model that creates a sense of… like a simulation of some sort that is fed by a data pool basically where it will make those inferences and then create an output based on the data that it's fed. So it kind of goes through a process of translating that data, whether it's text, image, else work and usually using a neural network of some sort, then creating an output there, which could be creative output. It also could be an output that's used in a software or something internal. But there's a lot of maths involved, obviously. And there's a lot of computation. So, usually it requires quite a bit of computer power or resources there. But normally we have like this large data lake of some sort pulling from that. And from that, we create an output informed by the data that we've presented it.

Interviewer:
Thank you. Yes, I find this question always very interesting because the answers vary a lot. And I feel your answer is also very interesting here, a very technical answer.

A:
Well, because it can be lots of different things, right? You can be doing simulations, you can be doing visual things, you can be doing text-based things. So it's kind of like an umbrella thing. What the core thing is actually…the principle behind it. So it is, and it's simulating, it is a type of intelligence, but it is intelligence informed by a digital setup of some sort.

Interviewer:
So now I wanted to discuss a bit about the impact of these new systems and new large language models like what that you talked about so for example the development of LLMs and big ai models is quickly changing the perception of what AI is, do you feel this is impacting or will impact the current state of game development?

A:
Yes, it is. And it's already been working on it, but it's going to take some time. It's not going to be something that's overnight. Everything's going to be replaced by that. But we already see a lot of procedural-based games where level design is kind of procedural open-world environments as well. A lot of these kinds of games are actually very much built on algorithms. And I mean, even a video game at its core has some basic algorithms and artificial intelligence, such as enemies being spawned and so forth. So actually it's not a fact of whether this is going to be a part of it. It's actually a huge part of the

landscape as it is. We're now just seeing it a little bit differently because we have more abilities, perhaps, to do, or it's probably just more accessible to do and also easier to compute. So you don't need to have something super crazy to run it. But it's happening already, and it will continue to happen. The important thing is just still at its core in terms of gameplay and so forth. To know how it is being used… It's not just if it's just there for like, you know, a value that's cool, like, you know, No Man's Sky, that one which is all procedure generated, but not many people play it in the end because, you know, that's the really cool, interesting thing about it, but it's not enough in terms of the gameplay. So it didn't have that, even though it's such an incredible promise, you know, it doesn't quite deliver that. So I think, but I do think it's going to become more and more relevant because it's going to help make these games more repeatable and serve the economic model a little bit better.

Interviewer:
Good. And I think this translates very well to the next question. You kind of already touched the points a bit. Newer systems offer a lot of flexibility and opportunities at the cost of performance sometimes, especially when talking about large language models or machine learning models, because they need an external server, they need to train these systems. And this performance is way higher compared to symbolic AI, so these simple algorithms that normally just follow certain rules. Do you think one of these two will disappear at some point? So machine learning or symbolic algorithms, do you think one of the two will prevail in the development?

A:
I think both will probably find their place. I mean it's not that there's a new form of... It's not like when cinema came out, theatre was obsolete, right? We have lots of different ways of doing storytelling. And video games is a type of storytelling as well. So I don't think one replaces the other. Maybe one will be more used or more commonly accepted for other reasons or for economic reasons or just in terms of distribution or whatever. But I wouldn't say it's, you know, black and white. There's lots of different types of games, different types of consoles, different types of things. If you're doing something with Arduino based, you know, so but, you know, it's just another we're now just exploring another realm of video games that we can push forward and see where the industry goes. But it's not necessarily going to make other things obsolete. It's just going to add the complexity of it all and, you know, open it up to more fields, to be honest.

Interviewer:
Thank you. And a bit to the side of the hardware. When doing my research, I noticed that the hardware seems to get faster and better every year but the performance of games seems to be always a problem when developing a game a lot of discussions goes on performance how we get a better performance out of this how we can reduce the costs in these algorithms. How do you feel about this tendency of pushing systems to the limits and do you think this has some impact on the player experience?

A:
If you think back to video games from the 80s right? We've come a long way and a lot of that way is hardware um some of the core interactions though you know are still relevant today and some of those core video games like tetris or whatever is still very playable and very fun today. Now with a lot more gimmicks and stuff going on there, much better graphics, but, you know, or at least we can debate whether they're better or not. It might be a preference. But I would say that the idea of, um, sorry, you were mentioning the, I just lost my train of thought there.

Interviewer:

Mostly why, like the fact that people try to use all this performance while not really focusing on simpler systems or trying not to push this performance. Do you feel that this is a-

A:
I feel like there's, you know, the industry is huge. So I wouldn't say that's a big generalisation to say. Now, granted, there are a lot of like: really bad games out there, ones that are full of bells and whistles and of course not optimised. And of course, if you're trying to get it across different consoles and you have to make it compatible, Windows, mobile, things like this, there's a huge amount of complexity because each of these mediums are being evolved together but kind of independently from each other. So there's room for tons of bugs and everything like this. So to make something super scalable, we know it's going to have a big issue. And in the past, of course, we would just have one type of console which would be made for it, right? So now because we're trying to get a larger target audience and, you know, get a wider spread of things, you know, it has much more complexities to work on these things. So there is a question about that. However, I do think pushing for this kind of innovation is necessary in terms of computer graphics, you know, where we can go. You know, a lot of the research that's being done, also like NVIDIA, those game cards and everything that's going into that, that's helping. And then, of course, I mean, there is a lot of work being done as well as how do we optimise the code? How do we make sure it's more efficient? But sometimes when you have something like a game that's already been built, and then they put more layers and layers and layers onto it, in some way, later on, you realise actually would have been better to have done it another way but you know it's also, you know, it takes several years to create one of these games and by the time the game comes out maybe the technology that was running at the time is not the most the state of the art anymore maybe some things are actually a bit of obsolete and then you have to do modifications and versions and renderings and of, this introduces bugs and things that are complex, but it's hard to go in another way because it takes time. And the industry is moving so fast and technology is moving so fast. So the only thing to help with that is to make sure that the concept and the gameplay and the mechanics and everything at its core is simple. And making sure that that simplicity at its core is working very well. And even on a very small scale, the idea without all the bells and whistles works and the user is really engaged and it does that. And then from there on, OK, whatever you add and everything, you're pretty safe. But if it's something that's really heavily reliant on something, you know, outside the game experience with something really code-based, that's when you run into problems. But there's some games that are focused on that, but other ones that are focused on this. So I would say it's too wide of a field to make a general specification for that.

Interviewer:
And so when talking about picking the right system, do you think that being able to choose on the right system to use for a task is often a challenge? I mean it is often a challenge, how would you approach this issue?

A:
I think what's really important is even before the game is designed, the story is made, you know what your target audience is first so who is this for? Do they have the medium? If your target audience is somebody who wouldn't have a PS5, but would have a mobile or a computer, and because that medium is the message, how you interact with a computer versus your mobile versus console versus a VR headset, totally different interaction. And they're also usually very different target audience groups who have different wants and needs in terms of story, in terms of how they play, whether it's passively or more like, you know, puzzle-based solving things versus like shooter uppers type of thing. So that needs to be the first thing that's decided upon before even a concept comes in. Once that's decided on, then when the concept is, you know, brought in and the interactions are really meant to work for that medium

and for that. And of course, by keeping it simple, you might be able to translate those interactions on different console types, but that needs to be decided at the very beginning. Because, you know, you can have an amazing idea and everything, but then when you put it in a certain device or something, it just doesn't translate. So you have to use the medium for what its advantages are. And if there's a great idea and it's actually better in a different medium for some reason because the targeted audience is there or the story is told better in that format, then it's better to shift to that than try to force it um down at that device. Because in the end we still have this interface that we have to experience it with and if it does if there's a disconnect between the interface the story and the user it'll it won't work.

Interviewer:
You have a lot of experience, you have already made some games and what I wanted to ask you is: maybe you can share an example where you had an AI feature that significantly improved the player engagement when added?

A:
Yeah. I would say probably the mixed reality game that I'm working on now, which is featuring in Venice, the film festival. It's in competition. It's called Impulse and it's on MetaQuest 3. And it's a game, it's also a story too though. So there we were working with the mesh and the environment and we tried to come up with interactions that would help the story that has not been done before. So again with that we knew our context was mixed reality on that headset device and so we had to understand exactly what constraints, what tools we had to use and expect. We did a lot of experiments in mixed reality and in the end we chose this interaction where actually we suck up the mesh of the room and then you can project it and shoot it into some other space so you know using mixed reality for the purpose of mixed reality not just to have like “oh this isn't a mixed reality environment” but actually using that in the core gameplay and transforming it so really trying to create an interaction that uh can only be done in this medium and therefore it's really engaging and special for that. So that's one idea there we also have a small game in there which is actually: spoiler alert … you can't win the game. Because it basically at the very beginning it's about like amygdala shutdown when your brain gets overloaded when you have adhd and you're like sucking up the environment and you feel like shoot these targets and you have to create, like, um, you have to get them the right way to make processes of sentences and essentially in very beginning, the first level, you get it, you win. Second level, it gets harder. And then basically by the last level, it's impossible. So you thought that you, you got better at something and then you realise, and so the idea is actually to make you rage quit almost. Cause you're like so frustrated and to simulate basically an amygdala shutdown, which would happen in the brain. So we actually use the gameplay to make you feel like what it would feel, what it feels like when you have this information overload, which is a negative feeling. But it was also part of that process, which is quite innovative in some way. And it's also because this is a story as well. So you want to tell that story. And that's one like recent way. And we used AI because we have to take the mesh of the room, knowing, you know, what's the walls and surfaces and so forth. And in using that to project elsewhere, it's not, you know, every game is different because every room is different. So we have that mesh kind of happening in the space. Now that's more just procedural architecture. It's not like a pure LLM, let's say. But that is one way of using that. And the cool part is that every time you play that game, even in your room, it will always be somewhat different because we're taking in real time information of your space that you're in and using that to translate it. I think, yeah. And then I'm also working in AI right now specifically for, but this is more an app and it's not game based. This is really about networking and so forth, but that's what I am doing right now. I'm doing a lot of research on that, but that's really for not game environments, but there's some gamification for sure. And we're using a lot of user engagement kind of things to gamify things, but it's the software more so than a video game.

Interviewer:
Thank you, I think those were super interesting examples. Now i wanted to to move a bit to what my design thesis will be so my practical approach to this to this issue. When talking about AI systems and information and when I talk about AI systems I talk about machine learning models, I talk about really big models but also symbolic algorithms so those more simple algorithms like state machines. How would you prefer to find this information when developing a game or if you need to research it?

A:
What do you mean by state machines? Like-

Interviewer:
State machines are a simple version of symbolic algorithms that basically takes certain states for a player or normally is used also for NPCs or non-player characters and then it decides what action he needs to take to move from one state to the other.

A:
Like behaviour states?

Interviewer:
Yes. But normally they are based on these rules that a programmer or developer will create, and they are not based on learning from examples like machine learning does, where you just dump a big amount of data and the machine knows what it needs to do and learns from what this is, how to do it, basically.

A:
Well, that's kind of happening on like a very like low level sometimes when a person like for example difficulty right um depending on how well the player is doing they can adjust the difficulty level of like how responsive this npc is or you know how fast it moves and those kinds of things so that's already happening. I think it's just adding more complexity to it, right? So I would say I don't see right now, because I think the thing is, it's not that video games and AI, AI is going to run everything, right? And it's going to replace the entire video… No, there's going to have to be a lot of curation in order to make sure that it really sticks to that. Now, AI can help frame it really well. Also, AI is mostly used to help developers just to do the code, to be honest. That's what it is. But there is someone guiding it in some way. And to have an NPC only working and responding based on pure AI feed, data feed, you're not going to actually have probably a really good output. Because the thing is, you could have, you run it, maybe out of 100 outputs, maybe 10 of them are great. The other one's not so great. So you'd actually want to curate that. So maybe it could be used to help curate certain things. But actually, there will need to be some rules in there that will be put in by the developer. And maybe the an AI can suggest different rules but I still think there will still be some part of finger in there because it's not yeah I mean like in the end it's the creative process… the creative process is using this and a lot of developers are using that to help them code things but you know it's it's like a really great tool more than anything else. But if people are saying, oh, the AI is going to run, no, because also that'd be right now, that's too heavy and expensive to do across every, because when you have a scene, especially, it depends on the graphics, but if there's like a lot of graphics going on, a lot of polygons, just having that data being running in real time in the background, too heavy. You can do something simple like what I mentioned before about the behaviour states, yes or no, this, that. Fine. But that's not a full on training with something in real time. It is also just not efficient. A lot of times players just want to get from point A to point B. But it could be a really helpful tool in figuring out how to do that and then having some variability in there.

Interviewer:
Yeah, I think this discussion on what AI really is is also very hard to tackle because as you said, right now, when you were discussing about AI, I think what you mostly were thinking about is uh machine learning or large language models. While I try normally to divide it in these two groups like machine learning and symbolic algorithms that of course can be combined together um but uh coming back to the the question, do you feel like there would be, like in the perfect world, where you don't know about anything of this, how would you like to find this information? And this could be a professor explaining it to you, I don't know.

A:
About machine learning or ?

Interviewer:
About any type of AI system, so machine learning is one of the types yes and the other one would be these more simple algorithms that as you said it before have been used since a long time like already in the 80s were used

A:
So how would it be best to learn about them?

Interviewer:
Yes, in your personal opinion.

A:
I never had a course that taught me this. Also because of the internet, everything's available there. So I would say actually, I would argue no professor because a lot of times if a professor is teaching that, you know, unless you are going for super high level in your PhD or whatever like this. And OK, that's one thing. But if you just want to learn about it, I would say learn through doing, because there's so many different avenues to go. I mean, it's very complex and huge. And use the Internet. That's what I use. So, I mean, I'm working with AI, doing UX with that and self-teaching, reading, watching tutorials sometimes, also just toying around myself with the AI and trying to see what my inputs and outputs are to understand it better. A lot of video tutorials are probably better to be honest because uh you just because it's changing too much that's the other thing like it's constantly evolving every couple months there's a new update right so you run the risk if you're doing something that's like you know a professor's done a course that they've repeated four times, unless you're like doing basics to computer science or whatever in which case you could do a course on that you don't necessarily need to you know uh but i'm a big advocate for like teaching oneself um because that's the way to learn it's if you're a professor trying to teach you there's it's really difficult because it requires you to you know make those things you need to workshop it and do it yourself.

Interviewer:
And do you think then it would be helpful to have a place where to find this information?

A:
No, online.

Interviewer:

Yeah, I mean, online place, like, I don't know, a website or a platform.

A:
They do. They have Corsica or you can go to MIT and they have a lot of their lectures. Actually, the course is completely online for free. So actually, everyone has access to learn this to be honest to some extent um the question is is do you want to put in the time and energy so when people say oh but i quit because i haven't had the time… well no no no we're we're fortunate to live in a time where this is really accessible um so and and if you reach out to those who work in the field and they know and then you tell them what you express they can also point you in a good direction um but i would say uh like um also i do think that if a person is really interested they'll pursue we all know how to google now right we know how to do search and get information from there so we understand that so just you know go and search and do and you'll find something that might resonate in some way.

Interviewer:
So when you do this kind of research…you talked about the fact that having a practical or the chance to test out things is very important and so what normally are the things that you say I need this to better understand the systems or this whole thing.

A:
It depends on the project and what medium it is. So if it's video games, so I'm working in Unity, I am doing something with AI with procedural architecture and also with behavioural states with a little AI character that will interact with you. And those are like branch tree states and everything like this. I'm using some YouTube tutorials that I found on just like graph theory and things like this. So I'm also reading theory, finding that stuff online, and then finding examples of other people who have achieved something similar. Then I usually analyse those and understand, okay, what's going on, what's working really well. And then also what's the mechanic behind this how is this running and once so i usually also benchmarking is really important because uh it's a way for you to understand and then it scopes things really clearly and then you can actually it's like having a machine and then you know deconstructing it and then rebuilding it again right and so i would say that would be a better place to go because it can go anywhere and everything so if you have that, then it also helps you centre and like, well, what's the purpose? Why am I doing this? What am I interested in? And so benchmarking is quite nice, actually, I would say. And then also knowing, okay, well, what tools went into that? How do they make that? And it's like, oh, that's through TensorFlow or this and that. And you're like, okay, well, maybe I should Google and figure out how to do that kind of thing on TensorFlow. And usually if you Google, quite a few of them are like, how to make this with this. And so actually, you can already find a tutorial that will even help you do that. So I would say that's the best way. That's the way I've learned things, because I usually self-taught on this kind of stuff. And what's nice is that you can go at your own pace, obviously. And also because it is quite cognitive. So, you know, at a time that fits you and, you know, all these things too, right? So I would suggest that.

Interviewer:
Thank you. And that was basically it. I have just one last question. That is, if you have any other comments or things that you have talked about that you didn't really have the chance to to talk about right now because the questions were not on the topic or…

A:
I would say just in general, um it will become a buzzword right now um i'm very much aware of that and there's a lot of companies who are now startups using that and trying to see what they can do. And a

lot of them have a lot of promises that will fail. And, but it's important that we kind of go through this to be able to come back and say, okay, what can we learn and build on for the next? So just because something fails because it uses AI doesn't mean it's bad. But more like, I think it's really interesting and exciting to see what people are doing and how they can see the possibilities of it. And, you know, sometimes you just need to go out there and make stuff. And then afterwards, like reflect and see, oh, wait, did this stick or this didn't stick or whatever. But we're at a moment where everyone's just making things. And that's super exciting. And then some things will definitely stick because there is a huge potential in this. It's not going anywhere. Like we can see that this is taking off really well. And so, you know, be aware of that as well. And that's why it's quite nice, I think, now to at least wrap your head around AI, get to know it, because it's going to be more and more a part of our daily life. And I see one thing when I talk to people and I say I'm working in AI, and they're like, oh, I don't like that and everything like this. I'm like, dude, I love reality. Like I actually do not own, like I don't play video games myself. I just help make them. So I'm all about the "real". However, it's an incredible tool. And it's also, I think it's kind of arrogant when some people say things like "oh, I couldn't because it's too difficult" or whatever. And they don't want to learn. They don't put the time and energy into learning about it. So therefore they hate it and they dismiss it. And it's like, well, no, like that's too short sighted. Yes. Granted, it's not like the easiest thing to start to learn, but you know, it's, you know, they're basic principles. We've got a lake, we've got this output of this right it's like a sentence um you know we've got a bunch of words we put it together there's an output. So um i would say yeah the one only thing would be like the people hopefully should just like get curious and just educate them a little bit so they're not afraid but then also when because there are deep fakes there are a lot of issues and there's going to be a lot of things that pop up with this that are not beautiful because with every new, it's humanity. Every time there's something new or innovative, there's always a dark side that comes with it, right? [...] It also comes with other great things, but we have stuff like this. With that being said, if you're educated or at least you know the subject well, then you won't feel, because I feel like the people who would reject it so violently is because the people just don't know how it works and it scares them. And if you know how the monster is or how it works or what's going on, you can be able to handle it better emotionally and also just be able to have some distance. And even if there's an image that circulates and it's a fake image if you know a bit of AI you can actually probably look for hallucinations you can probably be able to see where, you know, it breaks here and there but if you don't you know you take that as its word and then you distrust every image that you see online right? And although that's going to get better and better, just having that education helps you stay informed so you feel more in control.

Interviewer:
Yeah I think you bring out a very good point here [...]

A:
This is just what progress and technology brings. And you have to adapt, but we've always adapted. Now people are just more vocal about it because we have a platform to be vocal about it. But I mean, it's not like this is an unprecedented thing. Like, oh, my God, now at this point, now the technological revolution, like in the 1900s, you know, changed so much. So, you know, it's kind of funny how people just focus on that when you're like, wait, look back on the pattern of history. This is just how progress works, which means also be flexible, which is why I say, like, learn a bit about it. You don't have to become an expert or anything, but if you're informed by it, you know, you're able to make a more educated guess on certain things or, you know, use it as a tool, really.

Name of interviewed person: Goran Sarich
Profession: Game Designer and Professor
Date of interview: 11.07.2024
Duration: 36 min

G:
Yeah, there's some people in the office too, I hope this is not too disturbing.

Interviewer:
Um, okay. Then I will start with a quick introduction. Okay. Um, I already, uh, wrote it in the email, but what I'm doing right now is, uh, research on AI systems, uh, used for video game development and by AI systems. I don't talk specifically, uh, of machine learning, uh, AI that are the ones that are now commonly, uh, talked about in the last like 10 years.
But overall, any type of AI system. So also what people call symbolic AI. So, you know, the more standard, uh, game development AIs. And, uh, I wanted to see if it was possible to, um, understand the systems have maybe a mapping on how they can be used. And so, um, that's why I'm doing this interview to see a bit, what is the.
What now professionals do when they research on this type of systems and so on. My first question is a bit more open and you just need to tell me the first thing you, uh, you, uh, it comes to your mind when I ask you, uh, how will you define AI?

G:
Um, it's a coded behaviour of an entity, um, inside a software structure.
Maybe

Interviewer:
you, you let me know in the meaning that there is not a really, um, correct answer. But yes, this is a very good answer. Yeah. Um, uh, this question I often ask, just because a lot of people right now tell me, yeah, right now, if you tell me about AI, you know, I start to think about your GPT or things like that, because everyone talks about that while now.
Yeah. But it's super interesting that you have a very structured way of thinking about, uh, artificial intelligence. Um, now we switch a bit to the topic of, um, uh, looking for the right system. So I have a series of questions here. And the first one is, uh, how do you approach, uh, choosing an AI algorithm when you're, uh, for a specific task in your game in like when you're working and what factors influence your decision?
For example, is there, you, you use accuracy or speed, um, performance,

G:
Yeah, as I, as I told you in advance, that's like, I didn't have much experience with, with like artificial behaviour in games per se, but it's, the base factors are like the, the balancing of the game mechanic, the challenge, how difficult it should be for the player and for our game specifically.
Very easy in most cases. So I'm not really sure if like AI is already the right terminology for it, but it is, you know, basically a state machine that switches from behaviours. Um, if it fits certain conditions. So I never did like bigger research on that topic. To figure out, you'll never use the behaviour tree, for example, like a more complex one in games, but if I would need to do stuff like that, I would probably, um, not be worried about performance, I think, because I guess if it's like simple, it is processed quite quickly, but.

More the quality of the behaviours, so that the experience of the player can really, you know, say, Ah, okay, this is made quite smart. It reacted in a smart way to my actions. And it's not like, Oh, very simple. I can always do the same. If I walk to the right side, the enemy always comes from the same perspective and stuff.
Um, I think this would be the criteria if I would now, you know, need to do a, a more complex, um, AI for, for, uh, for entity in a game, does that answer your question?

Interviewer:
Yes. And you also brought out a very interesting point. Maybe we can discuss it later. The fact that right now is not like you say, uh, behaviour tree maybe is not really AI.
Well, maybe other type of systems that are more complex can be seen as AI. I had done a discussion for procedural generation, for example, whether that can be considered AI or simply as something different. So if when you're doing research, how or if you will be doing research on a certain topic or system, or will you arrive at a certain system?
Like what will be your process there? Um, would you research other possibilities on the same solution? And if, if yes, if you, uh, what do you consider in the trade offs between the different algorithms?

G:
I would just try the, you know, since I'm very player oriented or the stuff I design is very player oriented, I would just try the easiest things first.
So if I, you know, look up papers or some stuff in the internet about certain algorithms, I would take the one that is the easiest. The easiest to implement, I guess, and figure out if the tests, if this is already enough, and if I see that there's a lacking of, um, the performance of the algorithm or of, of the system that I implemented, I would dig deeper.
I think I would definitely try to keep it very simple. Um, and you know, based on that, that would go further. I mean, there is like this one, the pathfinding algorithms, we can also ask ourselves, is this. artificial intelligence or it's not, right? It's a bit the similar thing, but it's also if my NPC needs to walk to a certain position, I would just try to do the most simple thing first.
If it doesn't work, a bit like very simple movement. If it doesn't work, I would probably consider to use an A star algorithm, but still, this is still not like artificial intelligence, I would say, but that would do the same approach. Um, if I would need to have like, More artificial intelligence stuff like in The Last Guardian, where you have Tico, the big bird that is very smart that starts to learn from you.
I would definitely, you know, write down the reactions I would like to have from an entity based on player actions. And if I can meet them, look for the easiest solution. If it doesn't work, I would dig deeper into papers, I think.

Interviewer:
And right now you mentioned already the internet, you mentioned papers. Yes.
So when you look for information when developing or doing research, um, where do you look for this information?

G:
I would probably first Google stuff and if I cannot find a lot of things, I would also go into different Discord channels of developers. I would probably approach it. Over the game development, um, scope or like from that, um, scene, I would try to approach it, ask people who did similar things in games first, and try to get recommendations from existing communities in forums, discord, maybe I would also check on Twitter.

If you know, if I have a terminology for a certain thing, if somebody used it. I would also check GitHub, I think, if I need concrete implementations of a certain thing. But, uh, yeah, the first thing would be Google and if, to see if somebody did like a work or a research on it and I would then probably not read the entire thing, but.
You know, the introduction. So to figure out if this is something helpful.

Interviewer:
Okay. And, um, this is, uh, I guess it's really depending on the type of, um, software you use or, um, the game engine, but when creating system, do you rely on preexisting knowledge or like, uh, packages or normally you start from scratch, uh, from coding these behaviours, maybe a mix.

G:
Yeah, I, I used to always start from scratch in the past, but now I try out existing frameworks and, but often you find limitations or it's a big boilerplate. And you just need like tiny features of it. And then I normally, if I don't have like a very, um, if I don't find a solution that fits my needs well, then I start from scratch, but I'm trying to copying things of the existing frameworks.

Interviewer:
Okay. And, um, Are there any techniques that you use for optimising your, or maintain a responsiveness in the AI behaviours?

G:
Uh, not any that comes into my mind right now.

Interviewer:
Okay, thank you. Now we switch a bit of topic and a bit more on the narrative side of AI. So, how do you balance realism and immersion? Um, and fun with when designing these behaviours. I think you already talked a bit about it before. Uh, but, uh, what trade offs do you consider when, uh, creating these systems on this side?

G:
Yeah. So I, you mean hypothetically, if I would create them, I would definitely, you know, like the, the player who plays an experience or he has an experience to question if, um, if the programmer or, uh, did really all these cases when I enter a prompt or if I do something and the NPC or the enemy or whatever it is behaves like in different ways.
So that it feels organic and less mechanic. I think this would be a thing. Um, yeah, which is quite hard to achieve, I think. And in the same way, it still needs to be predictable enough for the player. So that the challenge can be, you know, proceeded so that you can solve a puzzle that you can pass by an enemy or, uh, overcome an enemy.
So I think this balance between believable, what is believable and what And what is still a challenge for the player?

Interviewer:
Um, the next two questions, um, you can also say it's not possible to answer this because of the previous thing, but, uh, um, if you can share an example of an AI feature, uh, of a project you had in the past, also a really small project, I don't know, a game jam or so that when changing it, uh, it's significantly improved.
The, um, experience.

G:
Yeah. I mean, it's a very small example and it's not really a artificial intelligence, but we in far, we had like this buffaloes that were on the road. And when you approach with the, with the vehicle, they slowly move away. And depending on how quick you are, they move away quicker or less quick, so you can not hit them.
Uh, but once you're away, they go back to the street. And if you exit the vehicle, just with the character and you walk to them. They react a little bit differently, so they don't go away that far, but they're still in distance and they look at you and stuff. And before they were just like standing next to the road, but this little interaction, like the behaviour, how they react to when I get closer with like two different, um, two different ways of approaching them, I think made it a way better experience, I think.

Interviewer:
Thank you. This is a very good, uh, very good example. Thank you. And, uh, if you have ever encountered challenges, um, in maintaining consistency between AI actions, Um, or how would you approach the issue to solve it?

G:
I would just, yeah, I think you really need to make them super dumb in the beginning. Try out the gameplay, and if they feel too dumb for the player, make them smarter.
See if the player can still solve the section. And once it's believable enough for like, you know, for the world to immerse yourself, I think you don't have to go further.

Interviewer:
Thank you. Um, so this is the area where I ask a bit more about machine learning and I wanted to know a bit your opinion on this.
Uh, so what are your thoughts on the recent advancement on AI related to video game development? Uh, so such as like, uh, GPT or GPT, like language models or self supervised learning and so on.

G:
Yeah, I think there's like a huge potential there. It all depends a little bit on the data you feed these machines with.
And the internet is a very critical material. Some probably because it's very biased by privileged people. That's why we get weird results when we create pictures of stewardesses, uh, with, with, uh, How's it called? Mid Journey. Maybe you have seen that example in the 20 minutes. They did like 120 examples of stewardess and they all look super weird.

Interviewer:
I haven't seen that one, I saw other many examples of this, this bias, yes.

G:
So the, the, the process data is a very key factor where, uh, probably if we work with systems like that in the future, it would be good to have an own set of data that we can, you know, feed the algorithms with so that we get like.
More the results that we probably seek to have. Um, yeah, it's a huge potential in general. I think what the problem is, is that at the moment, these tools take over their creativity instead of taking us stuff from us that is probably annoying. There was one example, somebody wrote that on LinkedIn that.

You know, you want to paint a picture and not do the laundry and not be able to do the laundry, but you don't have to paint the picture anymore because the, because, uh, the AI is doing it for you. Right. So we don't want to make the stupid stuff because AI is taking the creative stuff from us. And, and I think this is important.
So how can you keep, how can you keep the creative, um, how do you call it? Um, Approach or not approach, um, you know, independency from these things so that you use them to be creative, but not that they do the creative part for you. Yeah. Of course, this is always a mix. It's hard to say because having Photoshop or not having Photoshop, Make, makes your creativity different.
Right. So always when new tools came, I think people were very skeptical if this is, you know, destroying our original hands craft or something. So I'm not so negative about this, but I, I think it's just, the results are so bad. But what we see, especially like visually, I think texts can be good, but it's not profound.
It's just good written texts. But if you look deeper into like the meaning and. And the context, it's not so good often, and that's why I'm not so scared at the moment that these tools will replace humans so quickly. But I think they're, they can be used as, um, as, uh, tools to kind of try out things quicker to iterate a bit quicker, but they cannot replace the final content at all to just supplements to form an opinion.
I think. And maybe it's helpful that they're so bad at the moment because then you can easily say, Oh, okay. I can take this part, but the rest is really crap. If it's different, if it's like three waters are good and one is shit, then it's probably harder to make this decision this way, or you take too many elements from them.
But I think we're still far away from that. Yeah. And when it comes to like, you know, help for coding, like co pilots, GitHub co pilot and stuff, I think this is super nice. And this is not really, you know, taking away the creative work. This is just redundancy. It's removing redundancy you have to do anyways by copying stuff or researching things.
It can help when you have a question, you, you can first ask that GPT if it's just crap, then you, you know, do it manually. I think for this kind of work, it's really helpful and can be helpful for the future, hopefully.

Interviewer:
And the follow up question was if you have ever experimented with these approaches, uh, or

G:
Yes, I did but it's more like, you know We already released the game and I asked MidJurney to create visuals for a concept that sounds like ours to see what the What, uh, the AI would do with it.

And it's interesting. Sometimes it's very similar. Sometimes it's really weird. Um, yeah, I did these experiments more like to re engineer stuff, to figure out how smart is it, but not to proactively create something new. But, uh, since, you know, I do coding, I was actually looking up a lot of code stuff, not visual things.
And it gave me interesting aspects. There's, for example, if you work with physics and engines, it's really hard to move something from A to B if you use forces and not positions, right? If I want a ball or something to move exactly to one position, it's really hard to use forces and I. chatGPT that, and it introduced me to something I never knew.
It's a PID controller, and this is used for drones and self driving cars and these kinds of things. And it really uses physical formulas to move one object to the other place, to a very concrete destination. And, and it was very easy to implement that in Unity. And I never would thought about that idea without using, uh, chatGPT, right?

So I think that can be, it can really help you to probably find spots that you missed. But, but you still need to be a pro. While using these tools, because you need to understand the context. I think like a rookie cannot just use these tools and then, you know, replace somebody who has experience.

Interviewer:
I agree with that. I also noticed a similar. Uh, thing, uh, during my research

G:
and, you know, what's, what's a big topic right now is for example, if you have very rough animations from a motion capturing AI can smooth it out. So you don't have to do the annoying smoothing out thing. I think this is cool, but probably the content will look very similar for after a while.
Yeah. It'd all be too roundish or.

Interviewer:
I think that was also a topic that people, uh, said, like you said, um, it's very useful for like experimenting fast, but at the same time, it gets very generic, very fast. And then it's, it's yet another game that just will fall in that beat.

G:
Something interesting I just had a recent discussion about is like a lot of people were thinking of moving to Unreal and I don't know what, you know, since Unity has this bad thing planned and Unreal 5 looks amazing.
But what happens with stylized games is they have everything flat shaded and then use the lighting that is hyper realistic, which looks cool, but they're like 10 games out. There, they all look the same now because you know, the, it just quickly looks cool and nice since it's so easy to achieve. It's like the creativity part of making a unique visual stylized look is, um, it's just not done by it.
You cannot, the lighting will not solve the creativity part of having a very unique stylized game. And I think similar things will happen with, uh, with AI tools.

Interviewer:
Um, Now moving a bit more to the topic of my design thesis, I have four questions here. One. Um, how would you like when you're doing research, you talk about using, uh, researching on the internet or so, um, if you could choose the best way, like in a utopical world, the best way to find out about this information on AI system.
And when I talk about AI systems, I also talk about any behaviour trees or procedural generation, any of this type of, uh, automatic behaviours. So how will you prefer to find information?

G:
Yeah, I would actually love to have a conference where people, you know, introduce me to systems without having all the time scalability and mark the market in mind by selling services.
I think I, at the moment, I don't inform myself so much about the systems because I always see, Um, companies offering services, subscriptions so that I don't care. I'm more interested in, in the way how things work, what is possible, but instead I have to always consider it as to include it into my business plan and how I do things.
That would be a nice way to have like an honest conference about AI technology in games. Which is not like focused on, uh, making money in first place, I think, or, uh, I don't know, like a blog with a good

newsletter format that is like, you know, approaching the topic in a more academic way, maybe, or in a more, uh, designer developer perspective.
I think this is something I really miss. I, I, I don't know anything about, I don't know one block who's like really talking about these things in a, in an interesting way, instead, I just see all these big companies with their services. I don't care.

Interviewer:
Um, so the follow up question was if, uh, for you, would it be helpful to have a place where to find this information or examples about systems?
And I mean, you already kind of answered this question. Um, So what I will ask you is what are the most health that you would like to see? On if you have a conference or a platform, but it's something you say, I will need something like that in there.

G:
Maybe concrete examples of how it is done in a game. I mean, a GDC conference is probably part of it. They. They cover stuff like that, some of the talks, but to have real life examples, um, where you see the systems, how they work in place, and you also see the real benefits and not just the potential. I think people talk all the time about the potential, but not about real results, but maybe this also takes a bit of time.
So you know, you can produce stuff. So you see real examples. This probably, and. Yeah, because then I can identify and say, okay, we have a similar game. You could use this, right? I think it goes in this direction.

Interviewer:
Um, now I have, uh, something I wanted to show you is a website that, um,
is a website that basically is made to showcase all the ways you can move in VR. And they made this kind of library for game developers or any type of developers that are working in VR. And, uh, I just wanted to ask you, uh, to let me know what you think about it. So the platform is called Locomotion Vault.
I don't know if you ever heard about it. Um, but it basically looked like this and you can see here, uh, you have all the systems they have inside. I think it's around 140 or something like that. And then you have the whole list underneath with examples or videos where they give you the category and so on the requirements.
And then, of course, there is a lot of filters or values that you can also change. For example, they say magical. So this means it feels like, uh. realistic way of moving or more like something like that would be not possible in real life and you can filter it out or if you need to be something like more of a game type of movement and so on and then you can filter them out and see this the whole thing so if you can give me a bit what are your thoughts what and what do you think about this this type of thing Type of platform.

G:
Yeah. It's super interesting. Um, because if I, I hope I never have to develop a VR game because I'm not a big fan, but if I ever have to. Yeah. It would be cool to see, you know, if I have an experience I want to design, and especially VR is difficult because of motion sickness and stuff, I could at least see some possibilities that people did some research on it probably.
And I guess I get some further information once I click on one of these features, right? Yeah, super helpful. I didn't know about this site. Is it like something that is shared in our communities?

Interviewer:
Uh, it is known, uh, more on the game development in France, I think.

G:
Okay. Cause it's really cool.

Interviewer:
But, uh, yeah, when I found it out, I thought it was, and this is like a research, uh, a group of, uh, students did, uh, not really students, they were like in PhDs, but, uh, they did this research.
They created this, um, um, values and then they, uh, search for the systems themselves. No, it's really good. There is a whole like a research paper on it. But, uh,

G:
and I guess your question is, if this, something like that would be helpful for,

Interviewer:
this is an idea. I think the idea of having a, for example, a forum or an event, for example, is also a very good option. Um, when talking with other, uh, developers, something that was common was like, they like to discuss with other professionals about the systems to see what other people did and so on. So this was for them a very important part. Uh, the examples was also an important part. And so I'm trying to understand a bit what can be done and what is really needed for, uh, game developers.
And of course, uh, depending on the solutions, uh, you know, you always have different pros and cons. Like for example, if you will sell a cell. Kind of a service that consults people that could work maybe with big companies But when talking with smaller studios or solo game developers that is not viable because the price is goes very high And it's not really worth it when while they could just search for youtube videos or things like that and get a similar result for the For the project, right?
So I was trying to see a bit what, what is possible and what people really like in there. So that's why I showed this example is, is just something that I found and I wanted to see a bit opinions in, in general on the, on the thing.

G:
I think it would be helpful because I mean, also for the VR kind of ecosystem, I really don't know all the possibilities, all the features, all the work that has been done to, you know, figure out what are possible techniques to use it.
And I think for AI applies the same. There's so many things going on and I have no idea for what I could use it actually. This would at least help to see the possibilities, the range, because this is also really hard to. To understand at the moment, you know, you can have in Adobe just something that fills your background or gaps, but then you can have a.
Entire story written by one tool. So they do different things or somebody, something is just out to completing your code. So they're different fields and it's probably hard to know everything. Mm-Hmm. , I think that will be very helpful.

Name of interviewed person: Martin Hodler
Profession: Game Designer
Date of interview: 06.07.2024
Duration: 32 min

Interviewer:
Um, yeah, just a quick introduction on what I'm doing. I wanted to study a bit. What is the concept of like in the last 10 years we have seen I have seen a huge increase of the use of the term AI or artificial intelligence in in time. And, um, And now a lot of people that are not into programming or, uh, in game development started to use the, um, this this world, right?
And so the first question I wanted to ask you is if I tell you, um, the term AI does it. What do you think? Like, how would you define AI uh, in your personal opinion? Like, what do you think AI is?

M:
That's, that's a good question because I mix stuff up. Also, today it's more like the artificial intelligence or they call it artificial intelligence of algorithms like, like, uh, um, the reinforced learning algorithms that are used right now for chat GPT or for, for, uh, image generation, all this, uh, generative AI stuff. But previously it was more like the, the word of AI was more, uh, or I, I connected it more with, with like faked AI in games, for example, like, like the behaviour of, of an enemy, how it is controlled or, Yeah, just, just behaviour there, like even in, even in German, it's the same thing. If you say, say the KI, Kunstliche Intelligenz, it's, it had, the meaning changed in both because of this popularity of generative AI.

Interviewer:
Um, yeah, I mean, uh, yeah, sorry, I'm not very good in this type of, but, uh, yes, that's true. Like, um, that's what I also wanted to, to address a bit. Because, uh, as I said, like I've seen this this change and and now it's very difficult to discuss, especially when you're talking about making games when you say, do you use a I then people start to think that you use generative a I or this new large language models and so on, while maybe you're just using a more simple, um System that is just, uh, or is moving a character or so, but you didn't really train it on on big data or so.
Um, So the next questions will be a bit more on you. How you do the research when you are creating or working on a game. So the first is about how do you approach choosing an AI algorithm for a specific task? Um, And if there are any factors that influence your decision.

M:
Hmm. Choosing an AI, meaning of generative AI or just overall algorithms?

Interviewer:
Overall algorithm. Let's say in a random case, you want an enemy to move autonomously in the game. Um, how do you choose which algorithm you want to use? There is a lot of different ways you could set up the enemy. So if there is, um, yeah, if there is a influence in the decisions or, um, how you choose it,

M:
it depends on, on what the results needs to be, if it needs to be a complex AI needs to be a simple one that is just like, like, like say for an old Zelda game where it just had enemies, if they see you, they run at you and try to attack you. And if you go out of a certain range, they stop attacking. For, for this degree, you would rather take a simpler approach, like, like just different states that can be implemented more easily.

But if you want to have more like realistic, Then you would more go for a more flexible system where you can change each behaviour in a way or even the overhaul behaviour. There's like something like behaviour trees. So they switch from each state to another. But if you want even more flexibility in what we want the system where you can change out this behaviour tree is completely like for a boss fight and where you have have different stages and for each stage you have have another set of states and actions.

Interviewer:
So you you will like do research or how do you decide on the system do you like check the other possibilities that there are then you take like consider trade offs and take the best one for you or normally you with the first one that you tried at the beginning Kinda worked or

M:
Yeah, I would say I would, it's, it's, it's a research about what would fit the needs.
Exactly. Yeah. Like I said, if something is simpler then, then I'm also going for a simpler solution because getting it done more quickly, is also most likely the better option and if it needs to be more complex, I look into the technology I use, if it, does it have, have anything to offer for that specific case. For example, the engines I use, if it is Unity or if it's Unreal or there, I try to, to check if there is a solution for that and how many of them and what fits best. For example, Unreal Engine has state trees, we have state machines and behaviour trees. So I could even choose from both. And even in these situations, it's like, um, if I don't know the technology, I try it out with a simple example.
Um to to decide if it works or not. So it would I would say it's still part of the the research Like fiddling around with it to see can it do what I want to achieve or not? And if not going for another

Interviewer:
and where normal for this information when you're looking for a system or in or there are like uh positive sides to this, uh, I mean you already talked a bit the positives and Uh in the the research that you you get but Where normally you look for this information?

M:
Um, it also depends on the project. I would say I start from the technology. I I want to use are already using And if this technology offers already some solutions, it's most of the time not The worst, the worst option to, to go for that that is already integrated, but if it doesn't offer that much flexibility, or if you need more features, and then, then I'm looking more openly, um, on less, So I start on the forums for the technology itself, and if that doesn't work out or like I need more flexibility, then I try to search for much broader approaches, how others try to do it, or even how other technologies implemented and then try to re implement it.

Interviewer:
So you normally start from pre existing knowledge or systems that you have already, and try to develop from there and adapt it, and if in the case that it doesn't work, you try to to find another solution that will fit.

M:
Exactly, yes. I'm in the situation that I kind of can work from pre existing knowledge, but if I wouldn't have this, then I would have to go through the different forums, starting with the ones that are closest to the technology stack I'm using.

Interviewer:
Are there any techniques you use or way to optimise the systems to maintain Like responsiveness while creating like this AI behaviors. Are there any like Things that you're you keep in mind while working on on this, uh, systems

M:
The responsiveness from a player perspective,

Interviewer: I think yes, uh, we can also skip this question I think it's kind of We have the question. I'm not sure what I was thinking when I wrote it down. So I'm sorry. I'm more on the narrative. I feel, um, level, um, how you you said sometimes you need something more realistic, something that, uh, but of course it will take, uh, more, um, performance from it.
I will guess. So how you balance this realism a bit, um, or, um, Yeah, between the immersion and the AI behaviours itself and like if there are trade offs you you consider Yeah

M:
Yes, I would say if you try to the problem with trying to get the more realistic Response or an experience with with this artificial intelligence in game It's like it can quickly Become unrealistic if you try to do too much in a way So if you reduce it, they are more used to AI that is simpler.
So that, so it doesn't feel weird, but you can even, you can still add on and enhance it so it, it feels better. If it somehow, if you get too much in this complexity, it's, it starts showing it through and there's like an art, it's called the uncanny valley. And there's also the same thing for, for the, the AI behaviours of Of companions or enemies in games.
Where it feels like, Ah, it could be real, but it is not. And then it feels, it feels kind of scary or weird. So I would, so in this terms, I stick to simpler approaches. And, and, uh, also for more, uh, player oriented approaches. For example, if you have enemies, you don't try to make them realistic if they are standing behind you, because you only have the small space in front of you to see in this game world.
And AI has much more information. If you would do a realistic shooter, you would always lose against enemies standing behind you because you can't see them or can't react as quickly as they can.

Interviewer:
I think you gave already kind of an answer here, but I was wondering if in any of your projects in the past, if you have like an example of where you, um, changed or had an AI feature that improved the, uh, the, um, the whole experience.
Like if you had something you were like, maybe this could I don't know you said like This is a bit too complex. Let's try to make it easier And then you saw that there was like a huge improvement there or or um a similar thing to this

M:
Not not exactly in in in my uh, bachelor's project it was like um, I tried to create an ai that that would You interact with you and see and record what you're doing in this world and try to to, uh, tell other peoples about that, other players that are trying this, this world.
And the interesting thing is the, the more I tried to give them more features, the more they, they, uh, Part from the other players so like you could see this is this is an NPC and this is a player. But as soon as I turned this a bit back so the NPCs had the same features that a human player could they interact through the same interfaces.

I had situations where player thought that the NPC is a real player and try to answer them in chat and react with animations and stuff like that. So I would say yes. This, this was a benefit of having the AI kind of simulating how another player would do in this world.

Interviewer:
And have you ever, like, uh, encountered, like, challenges in maintaining consistency between these systems, uh, or the actions themselves?

M:
Yes, it depends. There's there's some, the issue is it's like it's an algorithm is pretty deterministic most of the time. So, so with the same input, you get the same output. And then you have some And in this case, you have some, uh, yeah, reoccurring behaviour that you see, like humans would not do or not that often, or just repeating that gets robotically.
And this breaks then the immersion, in my opinion. So this is not that easy to break. It's easier if you can hide the NPC, so it only shows up at certain times and then goes away. So you don't see that much, but if you try to follow them around, you see they, at some point, they have a pattern.

Interviewer:
This was something I also found that I found interesting that, uh, uh, many games also calculate how much you see the NPC to see how much complex you need to do the AI.
'cause of course, if you are in a shooter where the enemy, you will see it probably for five to 10 seconds. You don't really need the very complex system just because they're gonna be dead very soon. So I found this very interesting. And now you you talk about this. I think it's a school that many developers think in this in a similar way where you need to consider this this parts.
Um, I wanted to shift now a bit more on the other topic that is about machine learning. I have two questions here and one is, uh, what are your thoughts in the recent advancement in advancements in the AI uh, such as, uh, I don't know reinforcement learning or GPT like language models concerning to uh video games

M:
In production or in in in the game itself or everything

Interviewer:
Anything like if if you think about video games and these systems like there is already big companies That started to use for example I think narrow networks to train um hm enemies or opponents, like especially for driving games, I think it's pretty common right now where they teach the I to drive basically, and then it's a way lighter on the system. Do you have any thoughts on that? Are you completely against it? You feel is something that could Yeah, I don't know if you

M:
Yes, yeah, there's, there's some of a morality question, especially in generative AI when it comes to creating art.
But, but, um, on the other hand, it's, it's, it's a technology that is not temporary. It will stay. And I think games could benefit of it. I wouldn't say always in production wise, because replacing an artist, I wouldn't say, AI will replace a real artist. It will create images that it learned, but it doesn't create something special or new, in my opinion.

But it can shorten times, for example, for finding an idea, in that terms. But working out the time. This is the end product, I would say wouldn't be beneficial, but integrating it in games for AI sounds pretty interesting and also pretty hard because like if you train AI too hard, then it would outmatch every player.
And if you try to have it some errors, then it would still look robotic and not realistic. So I'm not sure if there is a huge benefit to the algorithms now, maybe more for live service games where they have to. To less players and they need like to simulate other players sample in all these these uh battle royale games That could make sense because it's could be easier to to let an ai learn and set us in then then implementing an Algorithm that tries to simulate all these behaviours Um, but yeah, there are already some interesting games out there.
Like, um, I don't know if you saw the game where we try to, where you are a vampire and try to get into houses of the people. And that's an interesting approach for, for creating like responses of, of, of, uh, yeah, inputs. It offers new inputs like text or speech that you can react with, but I'm not sure if, I guess they will come, uh, they will come where they can impersonate the character in a way

Interviewer:
think it was interesting because from from my research what I found is basically that This whole discussion is often taken from big companies because they have the money to invest in this research and very often they invest in a Like in house software so they will not share it with anyone else and because they they know how valuable it is And for example, uh, the the vampire game You I remember it had a maximum amount of time you could play it because of the fact that you will have an external server doing the calculations.
I found that that will could have changed a lot what gaming means, especially when you need to start paying maybe for a subscription or something because the cost of the whole system is higher. While when looking at at indie games or smaller games. Very often I see them using these systems, um, during development but not for the game itself because it will cost too much, uh, keeping up the game.

M:
Yeah. Yeah, especially if you use services like chatGPT to do that or GPT itself. You pay for requests for each. Each interaction in that way. And if you want to do this calculation offline, you could do that because the teaching is the one that takes more, more calculations or longer. But if the model is there, then you could reuse the model and.
Just enter the prompt and get out the result. But I'm not sure how big they are already. Some models are multiple gigabytes large. And putting this in a game that is also pretty large could also be hard to handle. And it depends on what system you're playing in, in the end. For mobile games, I guess you would have to rely on some sort of online service or build up your service online as well.
Even if it's Closed down just for your product

Interviewer:
and the follow up question would have been if you have ever experimented with this kind of approaches in in your, uh, in your own work. Have you ever tried doing development or to implement systems after in a game?

M:
Um, I tried to use it just for some inspirational stuff or, or part I'm lacking sometimes, like, like, yeah, for generative AI, for creating a concept art kind of picture.

It was not exact the way I would want it, but it's still closer than I, when I would have drawn the image in the end. So I could use that to give it to an artist to make something. Even better and and also experimented with some stuff like music generation, but I haven't used it Actively in game projects themselves right now because of the fact that I would have to use the online service and even if I Wanted to release that I would have to keep track on how much it costs Costs me if everyone does this interactions with the ai

Interviewer:
that's true. I felt like When when looking for information a lot of Game developers have the same, uh, similar opinions, and I agree, uh, to the fact that an art like you can always get the kind of idea from or try to represent an idea that you have, but then it's, uh, it's often impossible to get exactly what you want without being very generic because, of course, these systems are basically based on generic, uh, things.

M:
Um. Oh, at one point I started using it, especially for coding. Um, there are like some AI assistants out there, like GitHub Copilot or Codium or whatever. And this, this was what I started using a little bit because like it, it saves a bit more time. You have already IntelliSense or, or how they name it in other IDEs that help you with code that try to give you already answers.
So…

Interviewer:
you can only press one button to. Complete line and started using it for for for my coding projects and it just it came in pretty handy for stuff. I would otherwise had to type out

now. I move a bit to my design thesis or the part where I because at some point I will have to make a practical approach to my problem and so I was wondering if you could decide how you get the information about AI system or how AI system work. Um. , is there a way you would prefer to get this information any way it works?

Like you could say, a teacher that tells you, or anything that comes to your mind that you will, uh, yeah.

M:
So when I could decide how this information gets taught for, for everyone or for me? Specific for,

Interviewer:
I think for you what, how you would like to learn about a new system or it works or, so in the perfect world.

M:
Depends on how big it is. If it's, if it's something that I use in a daily life, like, like, let's say, Chachapiti goes bigger and Devi's phone has kind of such an AI assistance in it. I would feel like, I would like to know that already in the late years of schools. What these systems do and how they work, how it is right now also with, with web and data and, and these, uh, yeah, that you shouldn't trust anything that you find in the network.
And with generative AI you have the same problem. I mean, you teach them data and the text that comes back doesn't mean that it has to be the truth. It's just a probability. The AI works like the next, it calculates the probability of the next few words that. fits the best in context of what's already written.

So I would prefer knowing this information as soon as, yeah, I guess in a teenage age where you start using this technology.

Interviewer:
I'm sorry, maybe I wasn't very clear with the, with the question. I meant in general, any AI system. So also considering like symbolic system or what you talked about before, like behaviour trees or so.
You said or sometimes you have to research a bit more on the on how the system works, or if there is a system that could do this a similar have a similar result to it. And then you said normally you go on on forums specific to the platforms you're using or the. Um, so in this type of research, do you think there is a way that you could like, um.
Something you would prefer than others by talking with other game developers. Some prefer videos, some prefer just to talk with someone, or some prefer to find books. So, something like this.

M:
Ideally some kind of digital version. Yeah, videos would be nice as well. Um, it reminded me of, of uh, game coding patterns, I guess it's called.
It's a open source website where you can click through a collection of different patterns that you could use for game development. And it would be nice to have a collection like that as well. It feels like a Wikipedia for game algorithms or game features. Yeah, it would be pretty nice to have to share stuff and discuss about stuff and enhance stuff.

Interviewer:
So the next question would be like, for you, would it be helpful then to have a place where you find all the information or correlation between systems with examples, maybe even, you know, made up code that is not specific to a language? Basically a map where you could find all the information and possibilities for different systems.

M:
Exactly, like a whole register or a collection of different approaches and solutions.

Interviewer:
And let's say you find a place like this, what are the must haves of this? Um,

M:
To myself, it's, I guess, like, it shouldn't be dependent on, on other, other technologies that you can't use in, in almost any system, more generic approach. Like, you don't have to rely on a technology from this engine or that engine, because that wouldn't be that much helpful. You can go into specifics later if the, if an engine already supports that, but, but, uh, the information shouldn't be like, yeah, you have to use this technology.
And then. It works. It should be more like the explanation from almost like from scratch in a way that you can in the worst case could build up, build it up yourself. And, uh, yeah, understanding how it works, I guess, for more complex systems, it would be helpful to have visualisations for that stuff. Or like you said, like a pseudocode in neutral language that you can like kind of copy.

Interviewer:
and on the social part, I don't know how much you I hear a lot of people. I mean, some developers like to discuss with other developers what they're doing and so on. Do you think is important or is more a

secondary thing where you have already your system or you have a specific question in your system is good to discuss or do you have any opinions on that?

M:
I think it's important because if you have something like that in a library, this, all this code that is written now could be outdated in five years and wouldn't be less, would be less performant or there would be better solutions to, to approach that and to, yeah, to notify this people in a way to, to update this, this whole thing, I guess, discussions.
Should be possible on such platform or such a solution even if there is like an approach and and other people found find like errors or or missing stuff or could enhance it in a way because I think the collaborative work is is still the one that creates the best results than just one posting out his his own experience and everyone has to follow this.

Interviewer:
Um, yes, thank you. That was everything. So the last question is just If you have any other comments or additions that came in your mind during the discussion or

M:
good question,
I guess I tried to try to talk about everything I had in mind in this way.

Interviewer:
I mean, I think you said already a lot of, uh, good information. Don't feel the pressure now to answer this. I just felt like if you didn't have the chance to say something else before, uh, uh, it was, I just added it in, but it's not really needed.

Name of interviewed person: René Bauer
Profession: Professor and Researcher
Date of interview: 26.08.2024
Duration: 59 min

Interviewer:
First thing I wanted to ask is: how would you define artificial intelligence?

R:
For me artificial intelligence is everything that expands how we act and behave and think to a machine or to an algorithm. So for me AI is also a computer program, it's also AI because it's something that is out of my mind and it can act on its own. I can also make a longer definition with steps but…

Interviewer:
No, no, it's enough. So now I wanted to talk a bit about the impact of these new systems I just talked about. So the development of large language models and big AI models in general is quickly changing the perception of AI. Do you feel this is impacting or will impact the current state of game development?

R:
I would make a difference between game development and games. That means in the game development it's possible or we have seen things that are changing and in the game itself I think it's much harder to do something with LLMs in my opinion. So in the development, I think the first thing where we will see things or saw also things is of course in creating assets. I think there you see a lot coming up, not perhaps not with LLMs, but with yeah, yeah, let's first say just content creation. Then one thing that worked also with newer technology is testing and finding bugs. This is something you see a lot, that some AIs are just testing a game and find out things that the normal person would never do and things like this. So there it's interesting because it's another view on the same topic. It's not anymore the human view on the topic. It's a machine view on the topic. And there I would say there are changes and you see it also in the models also in the models let's say in the assets for example you also can use them to generate 3D models, 2D models, also 2D graphics models and of course, also in storytelling. If we are talking about LLMs and things like this, I think adventure AI and things like this show really what you could do in the construction of games. And in programming, you see at the moment, a lot of people using AI, the last version of AI that helps coding, of course. I think this will really change a lot of things so that also more people are able to code without knowing perfectly how to code. I think these are the steps that will happen and in the games themsel…there it's much harder because as a game designer, you want to control the whole thing. So I don't think that we will see a lot of games that are developing together with the game or in the game. I see more that you train things and then afterwards you use these trained models. If we go one step back, you see also with this bachelor project: “me, I and me” or something like from the ZHDK. [...] This was an interesting thing because they also used a little bit of AI and recreated these common line games. That's where you really type in and they made new things that are very interesting so I see a little bit there that something can happen and the first time I saw something about this classic AI we have was in a racing game where all start together without knowing the course at all. And the AI's NPC really learned each round they drove to get better. This was the first time I thought, okay, this new type of AI could be interesting. [...]

Interviewer:

In the meantime I mean I think you talked a bit about it already. Newer systems offer a lot of flexibility and opportunities but normally at the cost of performance or in general costs compared to the normal symbolic AI, so the AI systems used normally in video games where you just create some rules manually. Do you think one of the two will disappear at some point?

R:
You mean like in…

Interviewer:
Let's say Pac-Man uses a symbolic AI where they just create rules where they say, okay, if you see a wall, turn left, and if there is no space left, turn right, and so on. And this type of system have been used for a long time now. But with the introduction of these newer models where you can just train your AI before having the game out. Do you think at some point one of the two will take, one of the two will disappear or they will both stay?

R:
No, I think at least they will both stay because one of the problems of game design is really till now that you can reproduce fun and have control. And one of the problems of machine learning NPCs or so is really that you can't, if they learn during the game, you can't control if you really have fun. So therefore, I think there is not so much change in the NPCs in games. If you look, for example, not a lot of games really use AI in the front end, also in the games. They use more statistics… also they train it and then use it like we talked before. I think one of the problems… also let's go back… If the AIs would become more complicated and interesting, this would be when they could learn like first in the neural network and then they could create out of classic logic. Then something really would change. And I think this will also be the next step of AI if there should be a next step so that they learn to create symbolic rules as you said because this this is what we do also as humans, we have first so this this mood in in our belly and think okay this could be and then we make research to find out what are the rules. And the rules are, of course, from an information economy system, much simpler than this "I have it only in my belly" and I think if this could change, if there's someone would construct an AI that really can learn this and is embedded in the other and would then not go to the belly feeling, but to the rules, then I think everything could change because then you would train something and say, wow, this is a cool thing. And you go up and it brings you some rules like Pac-Man and you say, ah, cool. On the other side, you bring some rules and you go over the modern AIs, and you go back so that you see something. I think this would then change everything. But at the moment, we are not there, and people also don't see at the moment why classic logic is really a funny and cool cool thing, with classic logic as you said before generative visuals for example you use a rule set and then you sit there and think fuck what did the algorithm do this could be interesting this and then you take something that is interesting and at the moment we are in the opposite. The modern AIs are trained by something that is some black box rules and we don't see more. This means at the moment with big data, you train the mainstream, this is one of the big problems. And in generative systems, you can create something that is in the rule you didn't saw, you can change a little bit the parameters. But this is a little problem for creative jobs that we want to see at the end, not only the mainstream, we want to see something new. And therefore, I think if this would happen, like the AIs would generate rules and work with these rules and not anymore with classic neural networks then i think something would really happen also then something could be totally different but at the moment um it could also be a chance for everybody that if once we are talking about the game and then we say, oh, God, this game is so chat GPT, which means is so mainstream. And this is not something good. Then you could say, OK, I want to have something new. So the discussion with

LLMs is really this, what is new? What can be created as something totally different than it's in the mainstream? So therefore, at the moment, I wouldn't expect some very good new game mechanics and so on from LLMs and deep learning. So I think there will be an industry where some tools work with today's AI and others will not.
I take something else. I was once in a social media game company and the guy told me what he does every day. And he said, you know, it was a Facebook game. I get an Excel list with every item we sold and then what we do is, of course, the items we sold, we make them better. So we sell more. And then I asked him, okay, you do only this? This is kind of an optimization, although it's an optimization like every AI are an optimization to their big data their data. And then he said no of course I decide at least one or two times a day against this Excel data against the statistics, to find something new that is really new because otherwise it's an optimization to the optimization and there's no creativity anymore inside.

Interviewer:
This is a very interesting point of view and um what you talked about before I feel like because right now we are starting to see some big companies for example creating the opposite as you said symbolic systems and then they use AI to train on it. So basically they have a certain amount of rules and then they use an AI to train uh to use these rules the best and i feel super interesting the fact that you the possibility of having an AI creating the rules so having the opposite uh direction.
[...]
The next question is a bit more on the hardware because the hardware seems to get faster and better every year but uh performance of games uh seems to be always a problem like they always talk about performance so you can get the game to run better. How do you feel about this tendency of pushing systems to the limit and do you think this has some impact on the player experience?

R:
My worst answer would be that there was the Game Boy and the Game Boy was, in its time, a really old-fashioned thing that was really, you know... There was a system from Atari, Lynx, that was much better but the thing was and it was coloured and so on and the thing was that at the end people played Gameboy. And I was always like god “how the hell they could do it” of course the problem of the better system was was that the battery um also colours used a lot more batteries so you couldn't play as long but at the end nintendo had the much better concepts, much better game mechanics that were fun. If you look at the market, and then you see, of course, what Nintendo did with the Switch was not to push the limits more and more. It was the opposite. It was like, okay, we make something that you can have in hand every day or you can put it in the stationary and then it has 50% better graphics. So they really make now a difference between “we go for fun and things” and the others go more for for for “performance and pushing the the whole thing a lot further”. Let's go one step back, of course AI also will change their performance systems, because not only in the production, because also things they will implement in the future is where you look in the next second. So render more in VR what you will probably look at. Things like this. I think there will happen a lot so that the same performance will lead to more experience and of course games wherever if it comes to creativity or this new factor “what is new” is very important for games, you can go to the mechanics or you can go to more graphics and often more graphics is very sold with nerd people. They call me a nerd too. There you can say “oh this polygon” because in this nerd fraction this is data that you can compare and say okay it has so much more million polygons: it's a better game than this other game. So it's a statistic you can bring and if you come from gameplay in comparison it's like oh I like this game more but there is no comparison you see? So this is something else but it's interesting that the game industry is really pushing since 20 years… the producers, because nobody of the simulation people in scientific want to have such good graphics, it's really the game industry. And also more and more the film industry that uses game engines

so this is really a fact that makes it very hard at the moment that game engines are not anymore only for game developers.

Interviewer:
I think it's also interesting because what I've seen is that normally, often it's bigger companies that try to do this performance, and especially on the graphics side, while you get indie developers or so where they still maybe keep a retro look or have this pixel art and so on and they are still selling. I mean people are still interested in lower graphic games.

R:
And retro is got something new our students are now creating retro games as PS2 styled games so it's not only pixels also pixelated things so it's retro yeah I think this is some but also the big companies of course have their problems if you see if they say okay 100 francs is not enough for this big product and as you say in this area AI is really important because you can for example create better 3D models and then crunch them with AI and not anymore with some algorithm of a person that made this. So it's more than manual. And of course, the big AI, the AIs will help the big companies and not so the small companies at the moment.

Interviewer:
I think we can now switch to another topic. During game development, being able to choose the right system when creating a game to use for a task is often a challenge. How would you approach this issue?

R:
The question, if you take Unreal or…?

Interviewer:
Not necessarily the game engine itself, but let's say you need to move a character through space, for example. There are multiple ways you could do it. You can use A-star, you can use state machines. How would you approach this problem of choosing the correct AI for yourself and for the game itself?

R:
I think one of the most important things if you are in a team or in a company is the saying from Sankt Gallen, the HSG…that you see as wide as your staff can see. So means if you are in a company and you don't have the know-how, you are not able to do things. So if you want to really decide, you must have people that could implement different types or you have to search for someone. And in my opinion, the question is always coming from the game mechanics. So what do you really need? Do you really need something that is so complex or less complex? And one of the problems is always, as I see it, of course, if you are coming from the more nerd faction, you want to show what you can do. You want to see or you want to experiment for you, you want to try out. Then this is an aspect that you have in a company. Okay, he wants to try out. And I think the most important thing under this decision is always that you go a step back and say okay what is the most important thing for my project. Also that you come from the project and say what is important. For example, if you take your Pac-Man, and in our days, it's not so easy to create a classic Pac-Man. Because if you make it with A star, for example, then you can say, okay, go there, and it goes there but it's not the mechanic of Pac-Man because Pac-Man is really going… you are in this labyrinth but the NPCs are going from one field to another only the Pac-Man can change in the middle of a tile for example. I think the possibilities you have gets you away from these classic tile based things but if you make a classic tile based in my opinion it's much easier to work with classic programming because it's simpler. But so therefore, it's really complicated. And the more AI things we

use today is of course more with if you have an open world not the raster not the tile and you really see this also if you look at students work. You see nobody anymore makes tile based things and before tile based was the main movement. It has also to do that you click on the background and it goes there so it's an indirect thing and most of the games are not any more direct, also the more PC games let's say like this.

Interviewer:
Yeah because you have the possibility of clicking…

R:
Exactly. And I see this change with the coming up of Windows 95 and PC as a gaming platform. Because there were no computer joysticks or so.

Interviewer:
So talking about this “using the right systems” I don't know if in your experience you have an example of an ai feature that improved the player engagement or the game in general when added or when it was changed?

R:
I think you can look at the strategy game strategy or adventures because there you see really the change. Before the adventures where you had to control with the amiga or something, there you had yourself to go to something and do something else and the mouse control was totally different. It's also the question of power, in the one with the joystick you have the direct power and it thinks it's the indirect power so you can ask you the question are you controlling indirect something and this was really used massively in the strategy games like warcraft and things like this and this really changed something. It's really you can also if you have to go with your figure there, it's a longer way. The other thing was I click here and say go here, and then I can do something else. So the AI really took away from the egocentric to a more bottom-down view. And this is also with the adventures. There is also you do something and then you can do something else. Otherwise, you would be there to steer it there. And it also has some magic, this AI, to find something because it's doing something and you don't have to control everything. On the other side, you can ask, is this disengagement if you are away and not any more direct controlling. So i think this would be one of the first thing that really changed how you you play today and i think it's also something like: if you're going from here to here it's really you click here and this is magically go around, it's really magic, even because then you're saying yourself, okay this avatar NPC is doing something…it's a self although it's not any for me directly.

Interviewer:
That was a good answer thank you.

R:
If you are interested in this, look also at Diablo. I wrote an article about Diablo together with Beat. And the interesting thing is that hack and slash was before to do it like this and so on and Diablo is also a hack and slash but you are only clicking the whole time and there you see a little bit also that this..it's something different. [...]

Interviewer:
I will have a design thesis this means that right now I'm doing the research I don't know how it is in ZHDK but basically we have a semester where we write a research so a paper basically and then a

semester where we have time to do a practical approach to the program and so now the questions will be a bit more related to that… [...]
So the first question is how… like when looking for information about AI systems and when talking about AI systems and talking about machine learning and symbolic systems. How would you prefer to find this information?

R:
Also in games?

Interviewer:
Yes.

R:
Also in games, it's a very difficult thing because also Yannick, the guy who made the master work a little bit, he also researched and what he did is also he tried out what in the article was written. And most of the things were not working. So he tried out really, you know, they said in the paper, it's like this. And he also tried out machine learning and other things and it didn't work at all. So one of the problems is really that even in the articles you often see see how validated these things are because it's it's a new field and so on so this is what… Then there are some examples which are well known where you can look into it for example.. also first thing is of course pac-man, because pac-man is the first game that has these behaviours each character has his own behaviours. I think there you find not scientific articles. This is one of the problems there are not so many scientific articles because these are always companies that are behind these inventions.

Interviewer:
So normally you look for articles or examples and then try to deconstruct.

R:
Exactly, because there is not so much. But what you could do, for example, if you had an interesting game, you could try to find the person who made this over LinkedIn and so on. This works. You know, I'm in a research program about 80 to 2000 games from Switzerland. And what you can do there is really to find the people and to talk with them over LinkedIn. This is the thing that works the most yeah and there was also one thing about your topic is there is in sidler or final fantasy I have also to find out. They for example developed for one… i think it was the 10 or something or the seven, they developed a known artificial intelligence so they gave out the artificial intelligence to someone developed it and then they want to build it into the one year before the release to build it in the system and this didn't work at all so what they did, they scripted the whole thing. This would be a good example to show what are a little bit of problems. [...] [...]

Interviewer:
Do you think it would be helpful to have a place, I mean, anything, a person or a website or a platform where you can find information, where you find correlation, examples about systems and how to use and how you could use them, where you want to focus or the possibilities that you have with these systems in games? Like do you think is because when talking with other game developers they often talked about how normally they look for other games that are similar or look at the tutorial videos on YouTube where they explain how to do things or how systems work then they can replicate it. Do you think it can be useful?

R:
Of course it will be useful I think this is one of the problems of the game industry that there is a lack of informal know-how or a lack of the science of game design, that this is not discussed and there are no works in this area that show you, okay, look at this, what is better for you, because this is one problem of the design of the, let's say, design practice or design jobs, that they always have this feeling we don't need research. So it's my practice “I do” it but that everybody could be interested in this is of course important. [...]

Interviewer:
Coming back to the platform, what are must haves you would like to see in a platform or a place like this?

R:
[...]
There would also be an interesting thing to have a blog where you invite a lot of people who are working in this area and they just put in things. And perhaps the work of you could then also be to make research on the topic, but also to take and take examples and make if you want to go to phd for example could then be okay i take these small examples and i make a theory out of this so i take together these three games and I make an article on it so you see a little bit therefore I would invite some people that are working with AI and research and they just have the possibility to post in there and you have something like wiki or something where they could work on but you are the person that also looks that no things coming. So this was just a side thing to your question…
I would really put in some examples you could also work just with a wiki or like this where you have the tagging system and if you find for example articles of it or even interesting would be people also that you say if you are interesting for example in in some pathfinding system these people if you make it from switzerland these people in Switzerland could be interesting for you or also a personal network and an international network another thing if you are interested in but this is more for older games is to go to Atari age and then really look if there is someone in there from those times. [...]
Same thing is of course with twitter or so you really ask people “hey are you the guy that in those times made this game can you explain to me or are there some articles” [...]
I think there must be I would call them pattern or something like this so that you say AI pattern or something in in game design something like this and then i would go there and say pathways for example or

Interviewer:
Like generalised examples…

R:
Yeah, you know if you start now it's something that's growing up it's something that is things if you want to really bring this in a bigger scale you then also have from time to time make some speeches or presentation at Swiss Game Hub to show people here we have, what could be interesting thing? And in the best thing perhaps also some lecture where you invite people could be an add-on which we could make at the ZHDK or the HSLU if you found some money. But I think this pattern could be an interesting thing.
There's someone who made a book that only is from game mechanic patterns. It's just an idea how you perhaps could organise it. [...] He tried to make topics, then you have just in the “dice” and then you see examples of what you can do with the dice as a game mechanic and it's a collection of game mechanics. [...]

Interviewer:
I feel like something like a blog already where you know if you want to see people that contacted you can go on the website then you don't have to share your email or phone number yeah then you can share it inside there I think it will be already a good option to get this contact. The contact or the opportunity to talk with other developers I think was also very common with other people that I interviewed. And I also feel this is important because, of course, you can always ask chatGPT and see what results it gives you when you need to understand something. But that's way better to have someone that already worked with it and then they can explain you what really works best and what didn't really work in their experience and then you know you have this exchange that you wouldn't have with with an AI.

R:
I tried out so out to use a discord server but perhaps in my case the problem was that it was about that the people are too old or…

Interviewer:
[...] the fact that maybe things already exist but are very difficult to find and when you have a bigger community around it of course it's easier to find because more people talk about it but it's not necessarily the fact that it's not interesting but more the fact that it's harder to find.
That was it. That was all my questions. I have one last question. That is just if you have any other comments, or things you wanted to talk about, but we didn't really talk about yet.

R:
One thing is of course this faking AI this is in my opinion would also be a bullet point on your thing where there is no ai behind but you think it's AI and because uh you can also design things that they look intelligent. This is for me other in games often also something that is interesting think about this i don't know one of the most well-known thing is in deus ex you can go there in every toilet also in the toilet of the woman and if you go in in the toilet of the women and then you go in a briefing the chief there will ask why you were in the toilet of women. This is something that is like intelligence so you think oh god but it's of course so trivial because you just can check up in which room you were. It's tricky. But it looks very, very smart. And this is something that I think is also a topic that is important for AI. This whole thing of faking. Doing like it would be because as humans we always think it must be something very clever behind. Also one problem of AI at the moment is that there are so many things that you can do with classic logic or with an older system like if you have just rules that generate something and people really think then it's really deep deep because they have no idea and the smallest AI in my opinion: is random.

Name of interviewed person: Ryan Brand
Profession: Game Designer
Date of interview: 15.08.2024
Duration: 47 min

Interviewer:
Good. So the first question I wanted to ask you is what is the first thing you think about if I tell you: how would you define AI?

R:
It is like a tool that helps you, well it's artificial intelligence so it's a tool that has some kind of knowledge about something that makes it easier for you to access maybe?

Interviewer:
Okay, so I ask this question just because the term AI is basically used everywhere and especially when referring to machine learning, a lot of people just say AI and then when you talk about AI it's super hard to really understand what you're talking about, especially if you're talking with people that don't do game design or don't develop games because of course for us the term will encapsulate a bit more than just machine learning algorithms.
So when looking for a system, when developing a game, how do you approach choosing an AI algorithm? Like for a specific task in your game and what factors influence your decisions? For example, does accuracy or speed or how interpretable the AI algorithm is influence your decision?

R:
That's a bit difficult. I'm trying to think about when I implement artificial intelligence in games. I mean usually when I do it then it's like enemy behaviours and stuff like this, so NPCs as well. So anything like that needs to have some kind of logic in the game apart from what the player can control themselves. And then I mean, then it's very different. So, I mean, in the end, the whole game is based on formulas that react to certain situations, right? So you can say the whole game, I guess, is like an artificial intelligence that reacts to certain things, a very small one.
But, wait, can you repeat the question again?

Interviewer:
It's just how you approach your choosing of the system.

R:
I try to break it down often to make it as simple as possible. Because, I mean, I'm also not an amazing coder so I try to create like behaviours that are easy to understand and I actually want to often make it so that it's easy to to write the code but I don't want it necessarily to be made so that it's so simple that the player can understand immediately what is happening in the back. So that it should kind of be like this entity that maybe does things and you don't exactly know how it works. Maybe you have to figure out after a while know how it works maybe you figure you have to figure out after a while how it works but um yeah I try to break it down for myself that it's easy to understand but i don't necessarily want it to be very clear for for an outside person what is happening sometimes.

Interviewer:

Okay um so you normally try to get the most simple system possible while still keeping a level of like credibility in the AI

R:
Yeah because i i mean when i use it then it's mostly like npc enemy behaviour and then i want it to seem alive and like it takes decisions based on certain things and you don't know what decisions it's gonna do, because it's like its own entity for me that's but this i'm sure, maybe there are other situations when I use like an artificial, I mean, yeah, it's a bit everywhere. Maybe I'm just thinking about NPCs at the moment. But I mean, I'd say I would also use it in others, like artificial intelligence.

Interviewer:
Yeah, it's a difficult word. But coming back to the topic, basically, how do you arrive at the decision on a system you want to use? You talked about the simplicity of it. Do you still research other possibilities when you find a solution? Like do you consider trade-offs between different algorithms or normally you go with the first one that works?

R:
I often have the problem that when I find something that works, I stick with it. And I will change, I mean, I will add different things to it. And then some things I leave out, you know, like just certain patterns or something. But I basically stack stuff on top of it and then just leave certain things out. When I have something that I think works for a purpose that I feel it's suited for, then I will not look into other ways. I mean sometimes yes, if there are some other difficulties or like problems, bugs or something then yes but if something is working I don't often feel the need to figure out a different way how it could work.

Interviewer:
And you talked about researching, where do you look for the information when developing a system? Are there positives or negatives in this approach in your opinion?

R:
I use uh youtube… I work with unity so the unity api which is like uh just a whole explanation of basically everything that you can do in Unity, all the methods that are free and what I also use is actually ChatGPT often now. And I think for YouTube which is really cool is that you have somebody explaining how certain things work. And this is a person who knows from their own experience what maybe was difficult to understand for them. And they can easily teach how something works. And it's also cool because you can basically follow the whole tutorial. So this is one plus with just watching YouTube tutorials. Minus is that often it's specific to certain things and you're looking for something very accurate to what you want to do and you just can't find it. Often you have to opt out. You have like, okay, this kind of goes in this direction and then you follow certain parts and you're like okay this I don't need anymore and then you kind of adapt what you need based on the videos that you see. You learn certain things you didn't know but often it goes in a direction that you don't necessarily need when you look in chat GPT it's often very clear,if you write a proper prompt, if you kind of know what you're looking for you write a proper prompt, it gives you such a detailed answer of how to do certain things. But what I'm afraid of is often that when I try it, because often I'm researching through ChatGPT, I did it yesterday on how to design certain patterns, I'm looking at it not in front of the computer. I can't test it right away. And I'm afraid because I've had answers from ChatGPT that were just false, completely wrong, you know, sometimes. So there's the difficulty that you don't know often if what ChatGPT is telling you is true, and you still kind of have to test it. And sometimes they're just, it works perfectly, sometimes it

doesn't work at all, which is it's more accurate it tries to answer your questions more accurately but you cannot tell if it works often until you try it yourself and the API is I guess something in between. The unity api it's like a whole database of different explanations for certain functions that unity provides and there you can pretty accurately look for what you want but I often find that the descriptions for me are sometimes … I don't understand what they're saying, if I knew more about coding then I could probably understand very easily just through the API, the Unity API what it means and how to use certain functions but often when I'm reading through I read something and I'm like, oh yeah, this makes sense and then I get to another part and I'm like, okay, I don't know what this does. But those are the three main things, I mean, I also go on Google or Reddit. But this is also something in between YouTube and the API somehow. Because you always find certain people with the same problems that you are facing. But sometimes you find an answer, sometimes you don't. Sometimes the answer is outdated.

Interviewer:
And going back to you talked about Unity API, I know there are some tools Unity gives you. So the next question was like, when creating AI systems, do you rely on pre-existing packages or knowledge or do you start from scratch?

[...]

R:
There is one thing that I almost always use that is pre-existing is the character controller because it's super hard to do those things and it's something that is it it's very important I think for the for the game feel for and if you have a shit controller then yeah the game can be really good but if it's annoying to move around with it then it's just it's not it's a very bad thing to have. That's for somethingI tried to see, like how important it is that I have a very well working system and how difficult it is to do so. If it's like super difficult and it's very important that it works very flawlessly then I usually take an asset if it's something also that would take me a very long time to do myself and there is an asset that I can buy for like five francs or ten francs and I also consider getting this but usually I try to do everything myself.

Interviewer:
Now we switch a bit the topic from the research to the narrative or feel of the game um how do you balance realism or immersion with fun when designing AI behaviours? Like what are the trade offs you consider? You talked a bit about it at the start.

R:
I have to think about how do I balance realism?

Interviewer:
It's a bit more simple than it looks. How do you work on the AI so that it feels the best in game for you?

R:
I think it really depends on what you try to achieve. For me, I often want to create a feeling for the player, how is it to interact with the enemies. And like some games, they're very reactionary. So you have very fast-paced AI like enemies and behaviour and you have to react super quickly but I also like it when it's very slow, for instance in the game little nightmares the second one there is a moment where you have this hammer and these enemies and they have a very simple pattern where they're just following you, and then you can easily detect when they're about to jump, you get out of the way, they jump, and then you just have to time it very, very correctly that as soon as they jump, you turn around

and you hit the hammer on top of them. And the difficulty there is not to really stay alive because you can easily react you always have time but uh more just to time it correctly so there are games where it's difficult to actually react to the enemies and then there's others maybe where you just have to kind of figure out the pattern and when you figured out the pattern then it's pretty easy from there on. And for me I don't need enemies that are super hard to defeat in games. I often like to just have enemies that have some kind of a presence. They work towards the story and the player, maybe they're associated with some emotion, like fear or they're intrigued that they want to know more about this enemy, something like this and actually combat related for me in the enemies is it's often something kind of secondary, I mean it depends on the genre and the game you want to create but for me the enemy has to kind of have maybe a story to tell or just be fun to to look at to see how it how it moves something like this for me this it's more you know it's like you see this this animal or whatever and you're kind of trying to understand how is it thinking, how is it moving and so on that is i'm not sure if i answered that correctly.

Interviewer:
Maybe you can have an example of let's say you were developing a game and you had a certain enemy, a character, anything, a system and it wasn't really working and then you added some changes and then it improved the whole feeling. Do you have maybe an example of something like that?

R:
Yeah I have, I mean, it's, this is kind of a strange one. It's from the Bachelor project. I had these kind-of spider things that would react. They would see the player character only under certain conditions. So when the player character is running, they would kind of hear it, and they would react to the player. If the player character was standing inside a light source, they would see him or her. And if the player character was holding a lamp, so these three conditions, if one of them is met they would see the player character and then there's like a maximum distance. This was quite unclear for people in the beginning so i added kind of a visor that whenever the player is visible by the enemy this visual thing like would lock on to the player and like move towards him and then there's like this a laser beam that that basically goes from the player to the enemy so that the idea was to make the player understand that “this enemy is now seeing you” and that was that was something that I added because people did not understand when you're seen and when not and I wanted to make it clear. Something like “from this point on they see you” and then maybe the player will understand, oh, it happened because you walked into the light, or it happened because he started running, or it happened because he picked up the lamp. But then people interpreted it as being shot by a laser. So like, you want to add something yourself to make your thoughts clear, but still often players interpret it differently. I mean, that's not a bad thing. That goes back into what I said before. I want people to have a feeling like this is an entity, like a living thing. And often you do not know how other living things will react. So this is part of the game too, that it reacts in ways you would not expect. But somehow you still have to, after a while, you should start to understand why it happens and so on. But yeah, that was just a funny thing with the spider thing that people completely misunderstood.

Interviewer:
Like you had this behaviour and I think it was super smart what you did in the fact that you didn't really change the AI of the enemy because it was working but you had to change the feedback that the enemy itself was giving like so that the player will understand.

R:

Yeah but it confused people in other ways. I think i i removed the laser beam and I only added this thing then that follows the player so when this appears they know okay now something is happening but they didn't see which enemy was seeing them so but I think that was something secondary which but yeah

Interviewer:
And have you ever encountered challenges in maintaining consistency between actions taken from AI? Let's say maybe certain AIs are super complex and look very smart and then the next enemy looks dumb and then you have to make the other one more simpler or make the second one more complex so that they fit together.

R:
I'm trying to think of… I actually never made a game with multiple enemies, I think. I had, like, other NPCs in the same game. I had other NPCs. And there, yeah, there was a very... It was very inconsistent. You had these spider things that have kind of a complex behaviour. And then you had these others that would just... Literally, they would just stay at one point and when this spider thing came they would follow it and go somewhere and they weren't really enemies they were mainly there for the story kind of uh the world building and it depends on what is the purpose of the enemy. So if the purpose is, okay, it's really an enemy, and it should, like, provide some kind of a challenge, then an inconsistency between two enemies... I mean, you can also have enemies that are more complex, more difficult. But if you have an enemy that is, you know, very clunky, and then you have a super well, you know, an enemy that has a really good behaviour pattern somehow that seems super realistic, then that must be weird somehow. But I mean, it depends on what it provides to the game, but I never added multiple enemies in one game before.

Interviewer:
I mean I think you've made a really good example here in the fact that you have two these two AI systems in the same game but one is complex because it needs to be complex while the other one you didn't really need it to be. You could have made them super smart and whatever but for what they needed to do it wasn't really the purpose. So I think it's a very good example.
Now I wanted to switch a bit the topic on machine learning and I wanted to ask you a bit what are your thoughts on the recent advancement on AI in this topic, so on machine learning and like related to video games so chat-gpt or also reinforcement learning. All of these new systems that are coming up.

R:
To be honest I mainly use chat-gpt when it comes to coding because for me it worked quite well in multiple situations. I think for coding it's quite good because I remember once I had to do a very small code for a certain feature or something in the game and I could have done it myself but it would have probably taken me like 45 minutes and then I just wrote a prompt of everything that I needed and within two minutes I had it. If it's something simple, then this is a very useful tool to have. I know there is a Unity or Visual Studio asset or like a function that you can add that you can write comments and then it will fill out the code for you. I never used that but I think I would like to try. But again I mean, I can't tell really about this program, if it works well or not. I heard from people that it's pretty good. And I think it's just good to be able to do more things in a shorter amount of time. And it makes the job a bit more creative in a way. I still think coming up with solutions like coding solutions it's also kind of creative in a way but especially like in game design I think coming up with creating the concept is where most creativity is, where there is more room for creativity. And I think where there are those tools that help you spend less time on the technical part of a game then you can spend more time on you know

aspects of game design that are more creative i'd say. I think these tools are pretty good but uh i'm really mostly using chat-gpt and not even that often.

Interviewer:
I'll be a bit more annoying with you. Have you ever like considered using things like image generation for example I saw a lot of games lately have a lot of these assets that are generated by AI, have you ever tried to use this or have you ever tried to use other type of system I don't know add charge gpt to a game or something like that or consider doing that.

R:
I considered adding chat gpt to a game or like just kind of a language processor or whatever like a tool basically for conversations or... Yeah, something like this. I considered it. I don't know how complicated it is to add. But back to the image generations, I actually did use it and I think it's a really useful thing in the early stages of game design. So when you... For concept art and stuff like this, I also don't want to take away from like concept artists because I know for them it's also shit when ai is used but um I used it once for a presentation that I had to… we had to come up with a game concept for a certain topic. And I basically had an AI, an artificial image creator, basically draw a mock-up of the game, what it would look like. And in the end, I had to touch it up a bit in Photoshop because I had to add some things just mechanics related. But the visual style was super impressive for the time that it took to make. I mean, it learns from other people. But I'd say for early stages when it's not something that you generate money from I think it's totally fine to use this. I would not use image creation now for assets in a final game like and ship it. I don't think, I mean you'd probably have some legal issues as well, but I would just not do it because this is just not something that I would like to do. But yeah, for like a small presentation, for a small idea or concept. You can really create an appealing design in 10 minutes, 15 minutes. And if you can explain your idea with this more elaborately, then it's worth a lot, I think.

Interviewer:
We are at my last stretch here. The questions now are gonna be a bit more related to what my design thesis is gonna be so the practical approach and so i would like to know: let's say you are in the best world possible for you how would you prefer to find information about ai systems when creating games?

R:
I think it's very hard to find the suitable AI for a certain thing that you want to do. And at the moment, to be honest, I'm not spending that much time researching what different types of AI there are. So I often stick actually with ChatGPT and that is probably not good. Because I'm sure there are AIs that are a lot better for tasks that I use it for. But I often stick with ChatGPT because like it was all over the media. And I mean, it was one of the first and probably not even one of the first, but the first that really marketed itself well i think um but maybe like some kind of um website or whatever where you can basically like a job search portal where you can write what do you want to achieve and then you a list of AI's, something like this, I don't know. And yeah, so maybe for Unity specifically, you can have like, how is it possible to implement this? Maybe there are links or so with tutorials on how you can actually use a language bot or something in an external program if that is possible. But yeah, like some kind of a hub where you can see what different tools there are. But yea I don’t use chat-gpt that often because there are certain flaws, but maybe there is another AI that I don't know about that would be very suited to what I want to do, actually, or to my needs.

Interviewer:

Okay, and let's say then we have this platform or website you are talking about where you can find all these different systems or explain how they work and what are the must-haves that you would like to see in there like something that without it you wouldn't even use it.
I mean you talked about examples for example before you can specify a bit more what do you mean with that or are there other things that you feel could be interesting to have?

R:
I mean it could be an AI itself, no? It could be like that you have a text field where you can just write in a prompt. I think this would be suited more for people who really have no idea what they're looking for who just write down “hey I have this this project I need to do something in this direction” and then it gives you tools that can help you do this. I mean it should probably also tell you if it's a free to use tool or not but these are like basic things, I find it difficult to come up now with something that really is a must have for me it's it would be mainly good just to see like like you get you tell the website or whatever what you're searching for and then it gives you a list of different options and maybe a small description of what this AI is used for and maybe also where that it gets the information from. I don't know if that's because often it's, it can be a bit, you know, where it learns from it. For instance, if it learns image creation tools, they learn from artists. And if it's something that you don't want to support yourself, then I think this should also be something that is explained. Like lots of people they don't want they don't support that the art of artists that is private is being reused for AI purposes.

Interviewer:
I have two other extra questions: one is if you have any other comments? You feel like you wanted to say something before but I didn't ask the right question if you have anything you want to add.

R:
No, I think you got everything out.

Interviewer:
The second one is that I want to show you an example. So this is a website [showing https://locomotionvault.github.io/] that is called Locomotion Vault and it's basically a library for locomotion techniques or ways to move in VR. They basically researched around gaming and other type of projects for all the methods people have used to move inside of VR and then they made it this library where you have these filters that you can move around you can select them depending depending on what your project is and so on and then it will give you a list of the system that will fit with examples for example so you you can see here you have a left right like two hands so this is the classical approach but then you have other stuff for example like fake running to move or things like that can you go on the there was a overwatch one i want to see how this worked this one but it's not yeah yeah it's it's with a keyboard yeah yeah I think basically you hold the jump button and the longer you hold it the fire the farther you get I think this was the example they took yes so a bit the question I wanted to ask you, do you think something like this could be helpful? You talked a bit more about having AI that helps you through it, like an assistant almost, that helps you going through it do you think with some filters like this could be something.

R:
Basically you want your opinions on this website? I think this is like movement patterns and stuff at the moment, right? I don't know, what else does it show you besides?

Interviewer:

It gives you an explanation of how the character movement works with a video also what the input is or what are the requirements. Yes normally they have an example video how you can move basically with this and at what level. So normally say for example it's magical if it's something that is would not be possible in real life and so on right so they have these different variables and then they define it by these and it's not specific for movement based games but for any VR application so there are some more simple versions where you can just move around or zoom in and zoom out.

R:
I think that could be very interesting to see, because then you can probably see patterns, like, okay, this tool is being used for this purpose very often. And then maybe this could be super interesting that could look like something similar where you have like requirements what do you need a tool for and then you have maybe people that enter different tools well like why they use them and then you can kind of see like which tool is good for what purpose. I think that could give you a lot of options, but this might be something difficult to do.

Name of interviewed person: Sebastian Burkhardt
Profession: Game Designer
Date of interview: 04.07.2024
Duration: 47 min

Interviewer:
So, the interview with S today is the third of July. To start, I think I will ask you, like if I say AI, what do you define it? What do you think AI is? What is the first thing you think about when I say AI right now?

S:
So, right now I think about the generative AI things, you know, the chat GPT things. Even though I know that's only a small part, because it's everywhere in the media, I just think about the image creation AI and that stuff, and not the actual what AI is, is a much broader term in my mind. It's like... And even though artificial intelligence, it's like, is it really intelligent right now? Not yet. I think the idea is that it's, you know, intelligent in a way a human is. But these kinds of AI's are not really, they are just copying stuff.

Interviewer:
Yes, so what I noticed in the past 10 years is something similar to you. Like we hear a lot of talking about this generative AI. And like as a game developer myself, I felt it took away a bit what the... What we meant before us AI, because I mean the term AI has existed for 70 years or more. And it was already used to define behaviours like, I don't know, in Pac-Man or so. So in the first video games. And so I wanted to talk a bit about this topic and how the term changed a bit, how we perceive the different systems and how other systems get a bit forgotten and then more difficult to research or find more about them. So my first question will be about how you approach when creating a game, the choosing of an AI algorithm, so for a specific task. And then what factors influence your decisions?

S:
So when I create a game, I don't start out how I work. It's more experimental. So I don't set out, I want to create, I need this AI. Maybe I want to create a game about using AI, but then I think I would just start out with what's the easiest thing I can do. I need to understand it first, and I think it factors in the accessibility and what I know about it. And then take that and start from there, and not really, I go, I don't know, because I don't know which kind of AI I use. It's the first one I take. It's more convenient, or what you find that already works for what you were thinking, like for your ideas. And then when it works, you just go with it. Yeah, and if I see it, it's something interesting, I can create interesting things with it, or I have fun with it, then I'll continue using it. And if it's like when I work with it, and it's really not intuitive for me, or not, you know, it works against my intuition, or I don't see a value in it, then I don't pursue it. But I think once I settle on, okay, I have this, I want to create this kind of small game loop, then I just try it with different kinds of things.

Interviewer:
Yeah. So do you ever like to research other possibilities when you have a working system? You have this system that works, does this job? Do you ever be like, ah, maybe there is a way to do it faster, or in a better way, or you find out, or maybe just test out if there are some other options and then come back to the first one.

S:

Not really, only when there's a problem that arises. When I see I need more functionality, then sometimes I know, okay, I have to start over, or I have to go through the code again and add some stuff, but not, you know, there are so many things when I create a game, you know, when I create the game on my own, not when I, you know, work with a company, or with other people where there's a clear goal. But when I create the experiment with this kind of stuff, there are many things I start, and then I see which thing or things work, and then when I think it's interesting, yeah, I mean, I'm repeating myself, not. Because it's a big hassle to rework and go back to my own code, it's not really, yeah.

Interviewer:
So you said you have a bit more of an approach where you test out a lot of different things. Do you ever do research on certain systems, or let's say you know already, you want to move a character or something, do you research what already exists, or you always start from zero. And if you do research, do you have already, like an idea, where to look for it? I think first I start when, you know, in Unity, let's say, in the engine, because I work in Unity, and it's the only thing I mentioned I work with right now.

S:
When I have, you know, I want a movement, then I start with what I know already from experience, just try to code something quick. To have a really small prototype, and then usually I forget some small things, and I know, okay, there's a tutorial, or there's that code snippet in that forum, or I save this, and then I, or maybe in another older game, copy and paste it. And usually in the new projects, it's always a bit better than the previous ones. So there's the iteration also already part of the, in this, even when I start. And, because I start with the small prototype, then everything I add just improves it. So I don't, yeah, what was the question?

Interviewer:
Yeah, where you look for information, but, yeah, Internet. Mostly you go on the Internet, and you look on different platforms.

S:
Yeah, and it's not that I think, it's just when I see it, and it makes sense for me, when I see, okay, this, I understand this, or, you know, okay. Or sometimes when it's a bigger, more complex problem, because, you know, I also have the, in Cloppyssimo , more the AI, things, the small NPCs, which are a bit complete, complicated. They're different stuff, and I also scrapped different things and, you know, watched different tutorials, but then I said, okay, that's a bit too complicated, or I just, I'm copying the code. I don't understand it. So it's finding the right stuff until my intuition and the project align. Yeah, it's a finding, but it's not really, I don't know when I see it the first time, or sometimes it's more like a gut feeling.

Interviewer:
And so you said you use unity for developing. There are a lot of tools. I mean, I think there is a navigation tool already. You also experiment with these tools.

S:
Yes, I figure. Yeah.

Interviewer:

So you will say that, I mean, when talking before you said that sometimes you develop a system, and then you just add to it, but maybe in a new project, then you will take the system you have already, and then insert it and maybe adapt it.

S:
Yes. So you will use already existing systems and never restart from zero unless you want to. Sometimes, you know, it depends on what I do when I know, okay, I already did this kind of thing, but I know the code is bad, but I start anyway from scratch, then I start from scratch, but usually they are in old code some things, you know, work that was already done that works that I can just copy, maybe it's just just the structure or a function or a function name. Like, okay, that makes sense that I do this before that, for example, I don't know. Yeah, it depends. But I think usually I am a bit lazy or all the work that I do, I never feel that it's wasted that I can reuse it sometimes. So I'm, I think I'm happier when I can reuse something because then I feel like, okay, that's in the past, even though that wasn't really productive. Now I can use it for something productive. And when I start from scratch, sometimes at the end, the code is actually the same as the other one. So it's. It's kind of a balance there.

Interviewer:
Yeah. So talking about this code, I was wondering like, when do you create this AI behaviour or NPCs or any automatic automation in your game? Do you consider the experience versus the complexity of the system? Like sometimes you see, maybe I'm going too much complex in the system. And I could have a better result or similar result with a more simpler system that would be more performant. Or do you ever do these considerations?

S:
I mean, I'm not a good coder. So I, when I find the solution that is simpler, of course, I try to adapt it. But no, sometimes, you know, when the code works, I'm afraid to change it because sometimes I don't really understand it. So even though I know maybe I have something there that works, usually the game is not that complex yet that I have to create something performant. And when I do then of course, yeah, so I have what that doesn't have to do with AI, but I have a system in the game that analyses something. And I know it's not really performant or it's dumb how it's set up. So this is something I know I have to redo or change. But I'm just not willing to because I think it's boring. I want there are other things that need my attention first, you know, creative things and not technical stuff, which I'm not that interested in. It's more like the technical stuff is just a tool to create something that I think is fun for me. But that could be anything.

Interviewer:
So if I will ask you, is there a moment during development or something when you were working on a system and at some point you changed something that you were like, now it looks way better that I added this change. Yeah. Can you maybe share an example or something that comes to mind?

S:
Well, usually it's the things that, you know, when I create a new game. And so my first game was a 3D platformer. And then I finished that, you know, it was my mature game. After that, I was still motivated to create a game. And I wanted to do a 3D platformer again but better because I already knew much more than before. And I could start from scratch. So for example, the movement was much better than before. So the character moved by, you know, so-called tank controls where you rotate it like this. You don't move where the camera looks. Yeah. You move where the character looks and you have to turn by left and right. And it's really annoying, and then to the other one was that you could move the camera

freely and the character would move there where you look. And also the controls were much better and stuff. You know, not perfect. Or, but in an existing game. Well, so in Kapissimo with AI, with the NPCs, it's not really an improvement. But I had this first part of the game loop, then I tried to merge the NPCs together more but the people were more confused. So I took them out, which made the game better. So, you know, that was an improvement. I'm taking something out. But it's still the idea I didn't want to, I wanted to incorporate them in the game. But I didn't know yet how. So now I'm slowly introducing them in another way. But it's not finished yet. But there I tried to start to make it different. But it's like I started like three or four times and it never really finished. Or I wasn't. Yeah, as I said, I wasn't happy because it wasn't intuitive for me. It was complicated. It wasn't fun. Or I just wasn't motivated to finish it. So now they are just randomly walking around. Which is, you know, better than before. Yeah. In a way. So you feel this change where there are a bit of a number, it still feels better on a plane inside than what it was before. No, no, no. It's in progress. But before the NPCs, they were in the way because they had to build something. You had to wait for them. And it didn't really connect with the gameplay. And now I just want them to be more like inhabitants, which you can mess around with, but they have their own logic and you can, I don't know, maybe guide them or like pick me and say, hey, do that. But you know, it's just in my mind. It's not done. Maybe having another thing that. Yeah, the pirate pigeon things, but you know, the game is not finished.

Interviewer:
So, yeah. The next question will be about, I'm pretty sure in Klopisimo, you have multiple different characters from what I remember and at some point you had interaction between them. Machines that will create materials. And then these piglings that will move the materials around. And then sometimes you had machines that could create piglings or things like that. Did you have or had problems with the consistency of this behaviour? Like sometimes they weren't working in the right way or were not collaborating between each other correctly. And did you manage to approach the issue or. How would you approach this issue?

S:
So, yeah, I had some problems with it. So they are set up in a queue. So they have simple commands that they can. Performed and it's like. They have a queue of things that they can. An array of. Action actions. Yes. And these actions are stored in a script. And then when they receive the script and they're still doing the script, then it just gets put on top of the. But sometimes. It happened that they would just freeze because. So you know there I didn't really understand why maybe they just received two inputs at the same time. And the. The action was not finishable because maybe the action was “go to that place” but that place was destroyed. So they didn't have anything and I didn't code an exception case where, you know, I didn't count for everything because it's so complex because these agents move around and when they are dependent on a thing that is also moving around, it can happen that the other thing just gets deleted. So that was difficult. But the problem was that I also didn't really understand it. So that's why I scratched it or I just moved it aside. And when I work on it the next time I will try to understand it better. I think it's more like what I will do next time. But it's just that I don't understand it better or have a better intuition or you know think about okay this is it: I need to create more null case exceptions. Also something else that I remember is when I learned about events. You know you can subscribe and unsubscribe to events. This was something that I learned and I put that in a lot of places, but it also created many errors but the errors weren't that bad. Then when I worked on another game, I also implemented events systems and somehow it didn't work and I didn't know why. But then I realised I have had to unsubscribe. You know it's something basic but it's a small fix. All this fixes so many problems so there I went back to all the events in all my games where I put them in and did it correctly because I saw a tutorial where he just did it quickly in a quick way. But it was really specific for that

point. There it worked how he did it but you couldn't do it for every case how I did it. So yeah I just put his code so to say or his structure everywhere but that was wrong. So I corrected it. So that was maybe for the other question. Yeah.

Interviewer:
So here now I wanted to switch a bit to topic. We talked a bit before this interview about AI and machine learning models. Coming into the industry I've seen a lot of people starting to generate assets or maybe even sounds with these tools. So do you have any thoughts about this system, in general you have any thoughts on the topic?

S:
I think yeah it's always a difficult question because when it first started I thought it was cool and interesting. But now you know the more I see it, it looks you know… When I think about the images that are generated, you instantly see it's like it has a weird style which looks a bit soulless or a bit generic. So if you use AI you can but I think it's just your fault if you are lazy and just take the thing that the machine spits out. Use it as a tool but the question is how you know the image is, I don't know. For a quick prototype, sure. That's good. Then the other question is the moral question that you know when they steal databases and you know that's another thing I have no answer for. Because yeah I mean it's like if you're an artist and you create art that looks good and you're afraid that you're competing with AI. Yeah I don't want to say it's your own fault if you're afraid because then it means your art was shit. But there are people that are paid to create generic art but it's even art I don't know, something like stock images and so on. But from what I understand is that if you will use these systems it will be more on the prototype inside while for a final product you will go more the traditional way or in a like self or human generated. Because you know if it looks shit you know. Yeah you need to use it as a tool so maybe you want this AI look. But I'm just talking now about the images. So this is also the difficulty of AI. It's how I said in the beginning what we are talking about. Maybe you are also talking about all the things like sounds or it could also be you can automate certain workflows I don't know.

Interviewer:
So do you feel you will have the necessary tools in case you want to do a more complex pipeline or this is not even needed in your opinion? Because from what I've seen, for example, very big companies like game companies are starting to use it. AIs or the also smaller games to train for example the NPCs or you know opponents where you don't have a multiplayer setup and you want opponents very often they use this. Supervised learning or so where they basically learn by themselves or by looking at other players playing and then they can use those you feel there is. Like you ever managed to experiment on something like that or…

S:
I don't have that technical knowledge, it's interesting but yeah but I think I'm still very beginner.

Interviewer:
So you mostly have experimented with image generation or maybe text generations.

S:
Yeah you know just to play around but not really to use it big you know but you know coding okay sometimes I use ChatGPT if I have a code or want to quickly create something and I'm not sure, then it creates something and so for that I used it. Or yes as I said for fun just to see. But the other stuff I haven't really touched because you need to have more knowledge about that, but I would really like to

once use these neural networks. To train, I think that would be interesting.When I want to do something it's because I want to understand it and I also think it has interesting gameplay opportunities or something I can play with images. You can't really play with them, you can create textures for sure and then put them on a model but I mean that's not really a game that just images. But I mean yeah it's AI is not images it's all the other stuff.

Interviewer:
Before you talked a bit about how you do the research and how you find the information about these systems. Is there a preferred way you like to find this information? Because you talked about videos tutorials, you talked about maybe blogs or websites with this information.

S:
No, not really. I mean of course videos if they are short or sometimes they are like the longer videos also you know Code Monkey but it's usually, no no I think those are good because they are like general but also you see how it's used. You know I just google it and then I click on the first thing and think okay that's cool. Maybe I click on another one if I think the video is not good. But yeah good question I don't know where inspiration comes from. This is also something I am trying to find out when I do the masters. Where does the bottom up comes from? It comes from reading, you know the algorithm which is subconsciously teaching me, giving me ideas or if it is from my head I don't know.

Interviewer:
Because I was wondering like for you will be helpful to have one place where you can find like this information like with correlation within different systems or examples and where you would like understand the systems or do for development like, just one place where you find most of this information or you would like more scattered.

S:
I think the scattered thing is better because I know if there's an additional place where no, okay this website is cool, I don't know, stuff about neural networks and it is intuitive that's cool as to have it as an additional resource but not having just one place because I think now modern times everything is, everyone tries to be or every company wants to create something more convenient and something more simpler and something easier and at the end everything is looking at the same. You know the monopolies they are, everything is one company at the end and. It looks the same, it's generic, it's supidified. You know by making something convenient you also lose. You know when you try to find something there's like a path and when you shorten the path the possibility to find something completely different by accident when you shorten it is smaller than if the path is a bit longer. Then you are able to stumble upon something which you never thought about which I think is interesting but now it's like everything you just type something in in chatGPT. You don't even need to use Google and I mean chatGPT is cool and everything but if you just use that then it's like you know you don't think about you don't you you. Yeah you're just getting dumber and dumber in a way or not dumber but. Yeah you're not able to… you know, now you are under the mercy of the big algorithms and not… You're not allowed to breathe for a second and be, you know. [...]

Interviewer:
But from what I've understood, you still want to find this information is just that you want more space to experiment on it basically. Before you said you want to find a video that maybe works and then later find something else that helped you fix what the first video was, why the first video wasn't working and so on. [...]

If you're looking for this information or when you're working on a project and you need it. I need to understand how this works. Is there like something that you normally look for, some stuff you normally look for like I don't know you talked about examples before, you talked about copying code so like code examples, implementation or some games that already use a similar system. Do you have something that you say “to understand this I need to see something like that”? Not that you consciously look for it but when you see it you're “yeah”.

S:
I think I don't understand the question

Interviewer:
If when you look for information yes if there is something normally specific you look for, where lik sometimes you find a blog where it explains all the parts right? But maybe it's not what you want because it's better to see a video of the thing working and you are like “ah it moves like this and exactly what I want” while in the blog you wouldn't see it. So is it other things you look for, you know it will be more interesting for you to understand the system or what you're looking for or. Normally just

S:
Yeah I mean sure yeah if I see the video how it works at the end. But it's so different you know sometimes. Yeah it's, I mean yeah, if it's visualised for me then it's easier to understand it, for me could be an animation video or a gift you know it could be also a website where there's like a website where they talk about tweening. and you know they have these small animations where they show different kinds of tweens and the difference of the curves and you don't need a whole video for that because then you need to scroll back and forth. You know sometimes a good structured website is much better. Because videos then you know they have the intros and advertisements and other stuff and distractions and the code you know then the video is so small and you can't really copy paste it you need to type it out but sometimes that's also good I don't know it's I don't look you know yeah let's say I know 80% videos 20% websites

Interviewer:
And let's say you will have a space where you can ask someone and then they can try to help you.

S:
I mean that's chatGPT.

Interviewer:
Yeah but let's say it's a person.

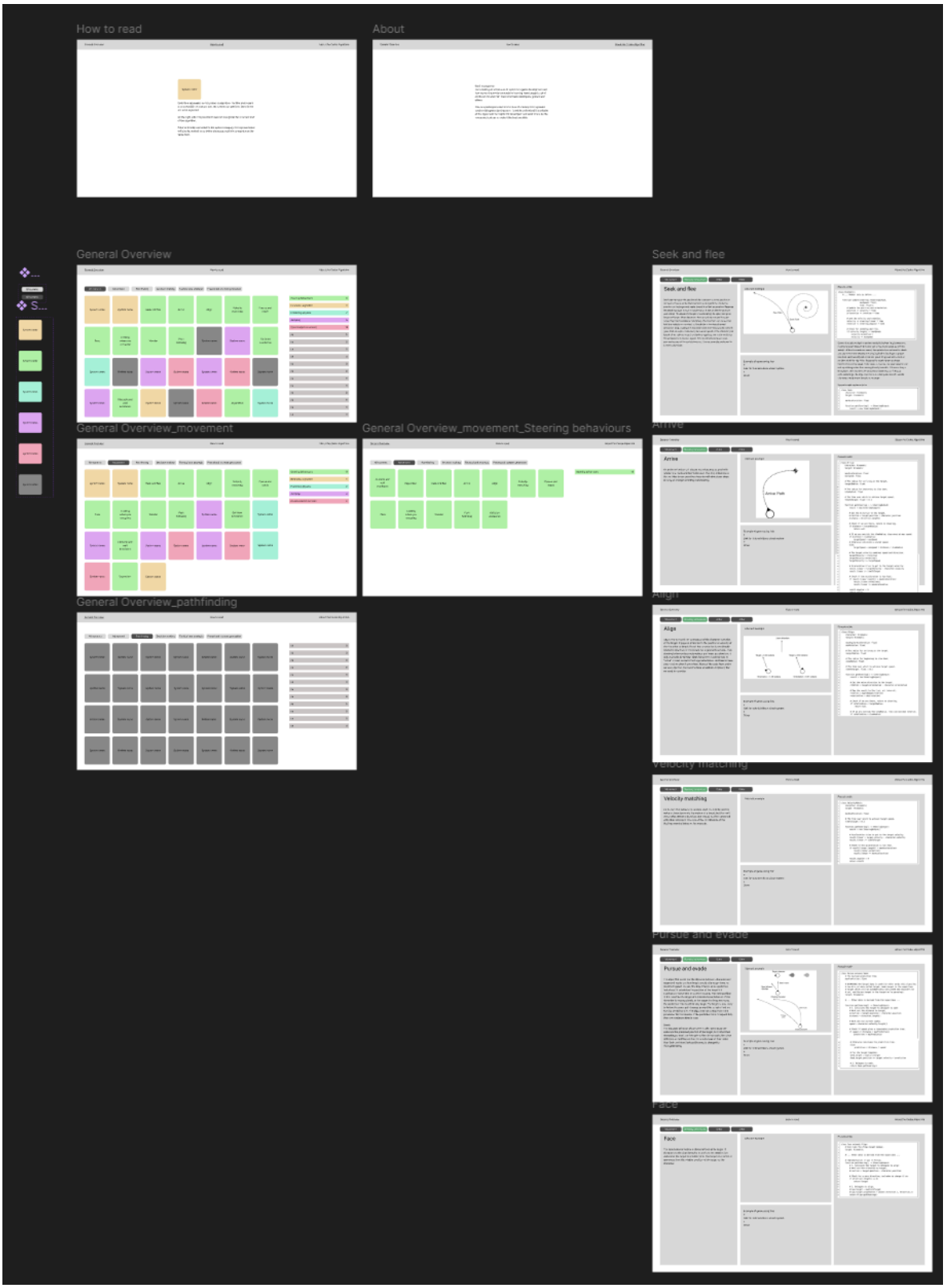
S:
Oh okay yeah.

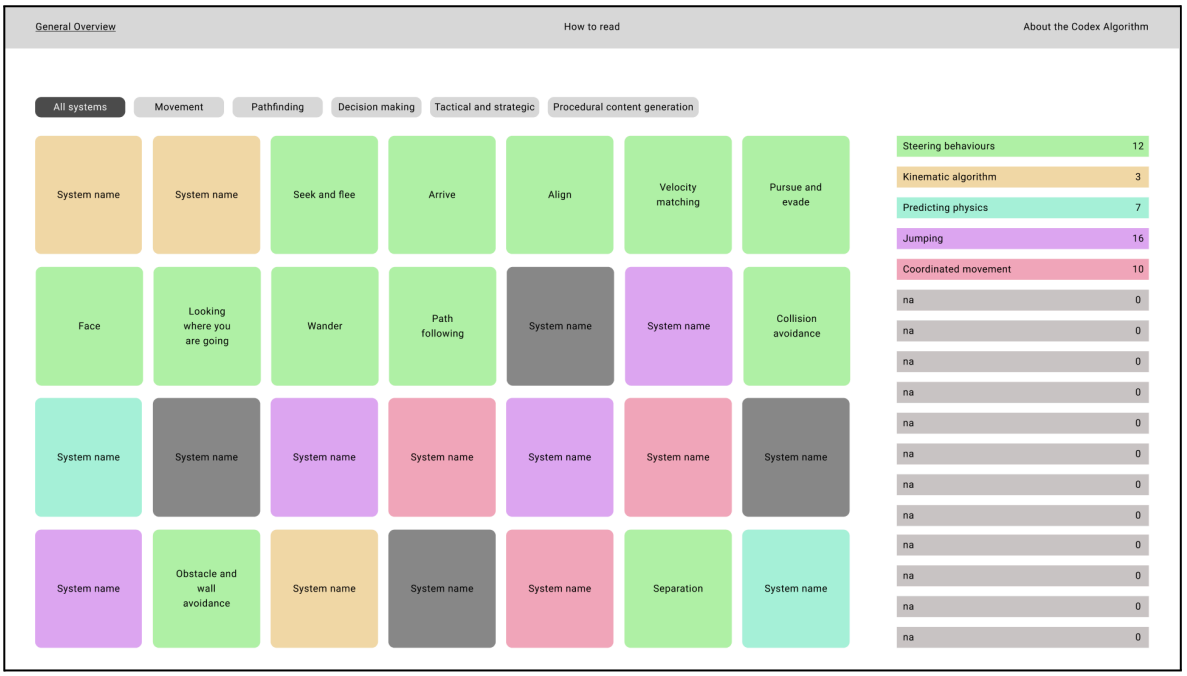
Interviewer:
Like I understand you despise this a bit but if you had access to an expert on a certain topic or so will you prefer that?

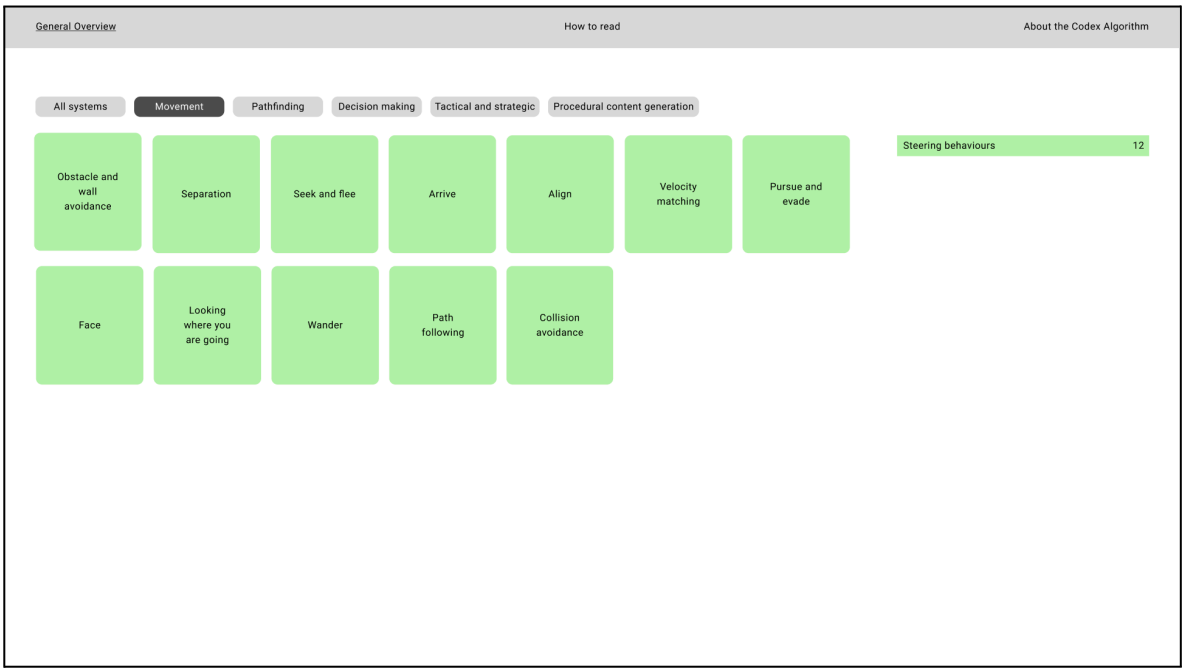
S:
It would be really cool, yes, to have a programming expert that I can just write up and not be guilty about wasting their time you know like Jeff for example but you know I know that other people are, they

don't have much time and they you know, work and whatever. But yeah for sure it would be really cool to have a tutor or a, how do you say, master someone that guides you or you can ask questions yeah and then you can code at the same time but it's I think it's difficult to find a person that thinks the same way as you think. But sometimes you don't even need that yeah for technical stuff you can ask them yeah but I mean I can't afford the person that is there eight hours a day for me or I don't know let's say for the hours a day but they are for sure that would be better because you know you it's a human and you can talk about it and then you get also other ideas, because yeah the computer just gives you back stuff that it knows you know now it's all these algorithms it knows what you want and then it doesn't give you other perspectives and with a human usually they also have other perspectives.

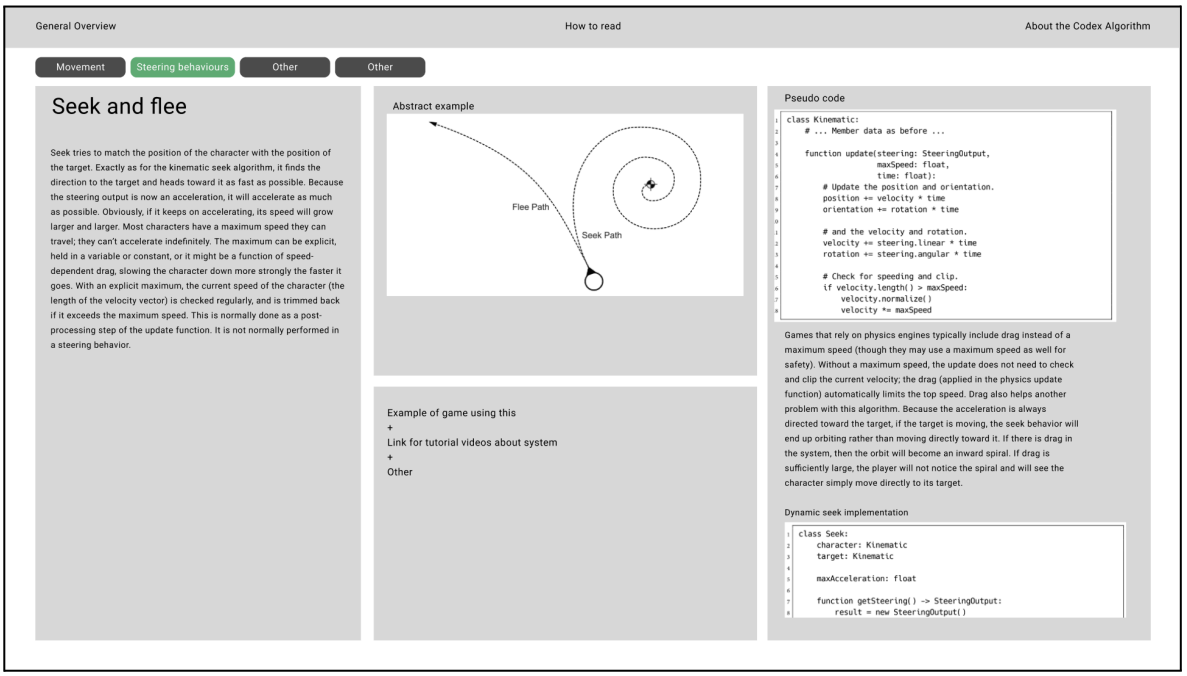
10.3. Prototype structure




General Overview
How to read
About the Codex Algorithm
All systems
Movement
Pathfinding
Decision making
Tactical and strategic
Procedural content generation
System name
System name
Seek and flee
Arrive
Align
Velocity matching
Pursue and evade
Face
Looking where you are going
Wander
Path following
System name
System name
Collision avoidance
System name
System name
System name
System name
System name
System name
System name
System name
Obstacle and wall avoidance
System name
System name
System name
Separation
System name
Steering behaviours 12
Kinematic algorithm 3
Predicting physics 7
Jumping 16
Coordinated movement 10
na 0
na 0
na 0
na 0
na 0
na 0
na 0
na 0
na 0
na 0
na 0
na 0


General Overview
How to read
About the Codex Algorithm
All systems
Movement
Pathfinding
Decision making
Tactical and strategic
Procedural content generation
Obstacle and wall avoidance
Separation
Seek and flee
Arrive
Align
Velocity matching
Pursue and evade
Face
Looking where you are going
Wander
Path following
Collision avoidance
Steering behaviours 12


General Overview
How to read
About the Codex Algorithm
System name
Each item represents an AI system or algorithm. For this prototype it is only possible to sort per task the system can perform. More filters are to be expected.
On the right side it is possible to see sub categories for a certain task of the algorithm.
Color is directly connected to the system category, this representation will also be worked on to better showcase multiple categories on the same item.


General Overview
How to read
About the Codex Algorithm
Movement
Steering behaviours
Other
Other
Seek and flee
Seek tries to match the position of the character with the position of the target. Exactly as for the kinematic seek algorithm, it finds the direction to the target and heads toward it as fast as possible. Because the steering output is now an acceleration, it will accelerate as much as possible. Obviously, if it keeps on accelerating, its speed will grow larger and larger. Most characters have a maximum speed they can travel; they can't accelerate indefinitely. The maximum can be explicit, held in a variable or constant, or it might be a function of speed-dependent drag, slowing the character down more strongly the faster it goes. With an explicit maximum, the current speed of the character (the length of the velocity vector) is checked regularly, and is trimmed back if it exceeds the maximum speed. This is normally done as a post-processing step of the update function. It is not normally performed in a steering behavior.
Abstract example
Flee Path
Seek Path
Example of game using this
+
Link for tutorial videos about system
+
Other
Pseudo code
class Kinematic:
# ... Member data as before ...
function update(steering: SteeringOutput,
maxSpeed: float,
time: float):
Update the position and orientation.
position += velocity * time
orientation += rotation * time
and the velocity and rotation.
velocity += steering.linear * time
rotation += steering.angular * time
# Check for speeding and clip.
if velocity.length() > maxSpeed:
velocity.normalize()
velocity *= maxSpeed
Games that rely on physics engines typically include drag instead of a maximum speed (though they may use a maximum speed as well for safety). Without a maximum speed, the update does not need to check and clip the current velocity; the drag (applied in the physics update function) automatically limits the top speed. Drag also helps another problem with this algorithm. Because the acceleration is always directed toward the target, if the target is moving, the seek behavior will end up orbiting rather than moving directly toward it. If there is drag in the system, then the orbit will become an inward spiral. If drag is sufficiently large, the player will not notice the spiral and will see the character simply move directly to its target.
Dynamic seek implementation
class Seek:
character: Kinematic
target: Kinematic
maxAcceleration: float
function getSteering() -> SteeringOutput:
result = new SteeringOutput()

General Overview | How to read | About the Codex Algorithm

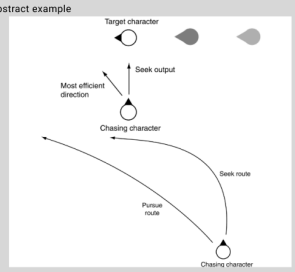
Movement | Steering behaviours | Other | Other

Pursue and evade

The algorithm works out the distance between character and target and works out how long it would take to get there, at maximum speed. It uses this time interval as its prediction lookahead. It calculates the position of the target if it continues to move with its current velocity. This new position is then used as the target of a standard seek behavior. If the character is moving slowly, or the target is a long way away, the prediction time could be very large. The target is less likely to follow the same path forever, so we'd like to set a limit on how far ahead we aim. The algorithm has a maximum time parameter for this reason. If the prediction time is beyond this, then the maximum time is used.

Evade
The opposite behavior of pursue is evade. Once again we calculate the predicted position of the target, but rather than delegating to seek, we delegate to flee. In the code , the class definition is modified so that it is a sub-class of Flee rather than Seek and thus Seek.getSteering is changed to Flee.getSteering.

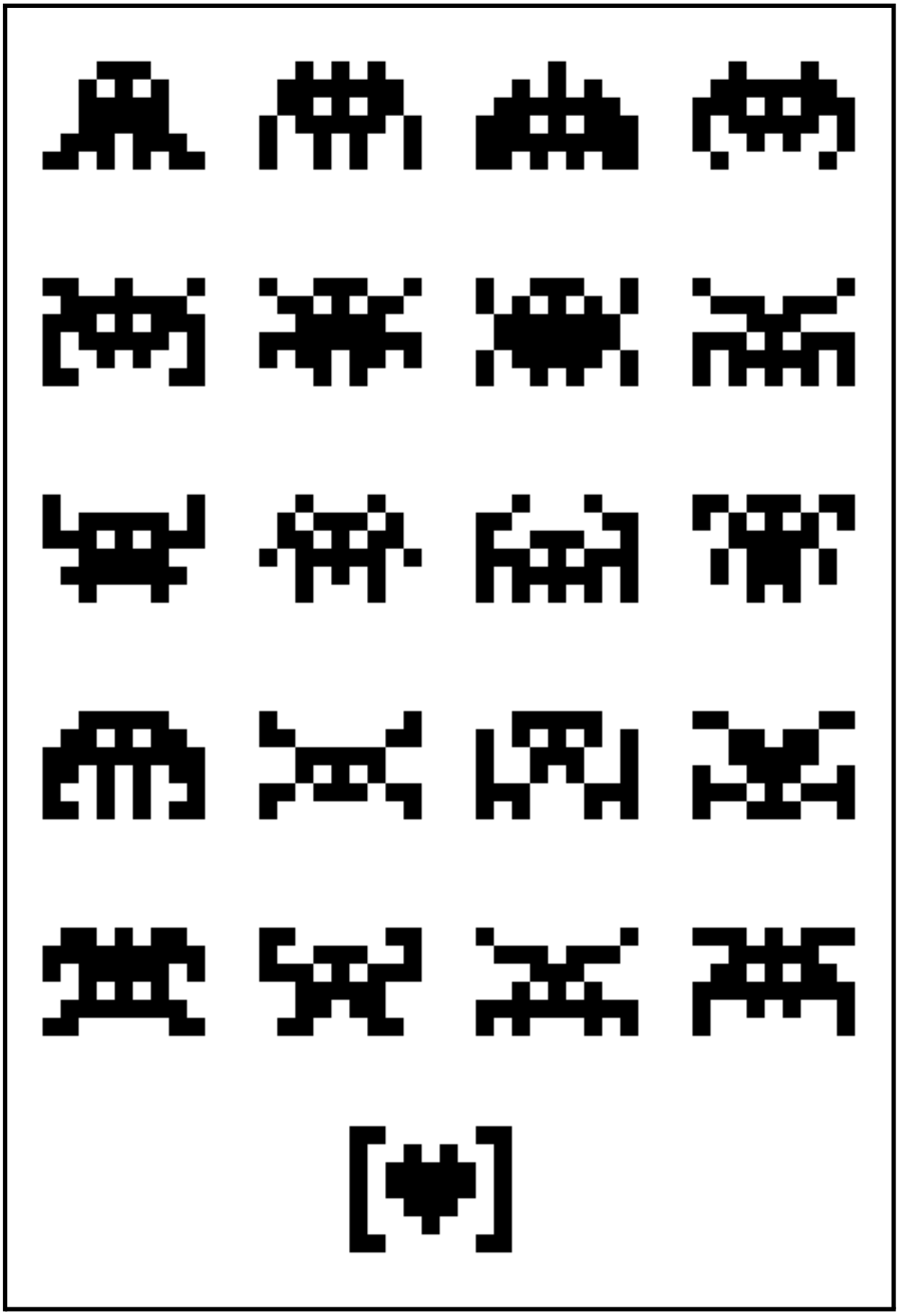
Abstract example



Example of game using this
+
Link for tutorial videos about system
+
Other

Pseudo code

```
class Pursue extends Seek:
    # The maximum prediction time.
    maxPrediction: float

    # OVERRIDES the target data in seek (in other words this class has
    # two bits of data called target: Seek.target is the superclass
    # target which will be automatically calculated and shouldn't be
    # set, and Pursue.target is the target we're pursuing).
    target: Kinematic

    # ... Other data is derived from the superclass ...

    function getSteering() -> SteeringOutput:
        # 1. Calculate the target to delegate to seek
        # Work out the distance to target.
        direction = target.position - character.position
        distance = direction.length()

        # Work out our current speed.
        speed = character.velocity.length()

        # Check if speed gives a reasonable prediction time.
        if speed <= distance / maxPrediction:
            prediction = maxPrediction


        # Otherwise calculate the prediction time.
        else:
            prediction = distance / speed

        # Put the target together.
        Seek.target = explicitTarget
        Seek.target.position += target.velocity * prediction

        # 2. Delegate to seek.
        return Seek.getSteering()
```



Pietro Peduzzi