



Impact of AI on Game Development

Opium des Volkes or Panacea

Evgenii Vetoshkin

Ilia Rozhnov

Bachelor's thesis

June 2024

Business Information technology

Evgenii Vetoshkin & Ilia Rozhnov

**Impact of AI on Game Development
Opium des Volkes or Panacea**

Jyväskylä: Jamk University of Applied Sciences, June, 2024, 67 pages

Degree Programme in Business Information Technology. Bachelor's thesis.

Permission for open-access publication: Yes

Language of publication: English

Abstract

This article examines the transformative impact of generative artificial intelligence (AI) on the video game industry, which is a leading sector of technological innovation. The main goal is to explore how artificial intelligence technologies such as ChatGPT, MidJourney and GitHub Copilot change the development process or change the execution speed and price of a project.

The study used a mixed approach combining qualitative interviews with industry experts and quantitative analysis of the use of artificial intelligence tools. The main findings show that generative AI significantly speeds up development by creating a wide range of diverse game resources, speeding up information retrieval and automating tedious tasks.

However, the integration of AI also comes with challenges, such as ethical considerations related to copyright infringement, potential bias in AI-generated materials, and the possibility of replacing human labor. The study suggests that while generative AI offers unprecedented opportunities, a balanced approach is necessary to address these challenges responsibly.

Future applications of the findings may include the wider adoption of AI tools in independent game development, enhanced player engagement through customized gameplay experiences, and the ongoing improvement of AI technologies to support increasingly complex and innovative game projects.

This study provides crucial insights into the evolving relationship between AI and the gaming industry, emphasizing both the opportunities and the importance of careful implementation.

Keywords/tags (subjects)

Game design, Game programming, Generative AI, Chat-GPT, Gemini, DALL-E, MidJourney

Contents

1	Introduction	3
1.1	Purpose of the Study	4
1.2	Knowledge Base	5
1.2.1	Information Retrieval.....	5
1.2.2	Terminology	6
1.2.3	Generative AI	7
1.2.4	Short History of OpenAI.....	7
1.2.5	Chat-GPT and Gemini.....	7
1.2.6	MidJourney and DALL-E.....	8
1.2.7	Current Researches.....	8
1.3	Acknowledgements.....	9
2	Topic Overview.....	10
2.1	Justification	10
2.2	Delineation	11
2.3	Definition.....	12
2.4	Hypotheses.....	13
3	History of the Gaming Industry	13
3.1	Structure of Modern Games	15
4	Overview of Artificial Intelligence in the Gaming Industry	18
4.1	Classification of AI	18
4.2	The Generative AI Tools for Game Production per Specialty Field.....	20
5	Overview of the Research Methodology.....	21
6	The Impact of AI in Various Fields of Game Production.....	23
6.1	The Influence of AI on Programming in Games	23
6.1.1	Coding with Gemini	23
6.1.2	Copilot.....	26
6.1.3	Machine Learning Enhancement	29
6.2	The Influence of AI on Narrative Design	31
6.2.1	Narrative Using Generative AI	31
6.2.2	Adaptive Storylines and Content Personalization	33
6.3	The Impact of AI on 3D and 2D design.....	38
6.3.1	Mid-Journey, Character Creation with Kemppainen Jussi-Petteri	38
6.3.2	Differences Between 3d and 2d design, New Midjourney Feature	44

6.3.3	Glimpse in the Background Images	45
6.3.4	Chapter Summary	46
6.4	Generative AI From the Perspective of a Game Designer	47
7	Ethical Repercussions of Using Generative AI	48
7.1	Possible Unemployment	49
8	Ethics.....	52
9	Reliability	52
10	Results.....	53
11	Conclusion	54
12	Recommendations for Game Developers Based on Research Results.....	55
	References	56
	Appendices	60
	Appendix 1. Fishing Game with Gemini	60
	Appendix 2. Plot by ChatGPT	65

Images

Image 1	GitHub Copilot offers StopFishing Method	28
Image 2	Story development without AI	37
Image 3	Story development with AI	37
Image 4	Generated Character First Draft	41
Image 5	Generated Character Second Draft.....	43
Image 6	Generated Character Finale	44
Image 7	Infographic “Job landscape changes”	51

1 Introduction

The rapid advancement of artificial intelligence has revolutionized numerous industries, with the video game industry at the forefront. Generative AI, a subset of AI technologies that creates new content and data, has the potential to transform game development. This paper explores the multifaceted impact of this technology on the gaming sector, examining its applications, advantages, and challenges in various domains (Christofferson, James, Rowland, & Rey, 2023; Vionix Studio, 2023).

The gaming industry is a leader in technological innovation, always pushing the boundaries of interactive entertainment. Generative AI provides developers with unprecedented opportunities to enhance creativity and streamline development processes. This technology allows for the autonomous generation of diverse and intricate game assets, including characters, environments, dialogues, and storylines, resulting in more immersive gaming experiences for players (Mind Studios, 2023). One of the most significant areas where generative AI is making progress is in the field of programming. Tools powered by AI, such as GitHub Copilot, are transforming the way developers write code. These tools offer intelligent suggestions, automatic completion, and the ability to generate entire functions based on context, significantly accelerating the development process and empowering developers to tackle complex tasks more efficiently (KPMG, 2023).

Furthermore, generative AI is revolutionizing the creation of 3D models and virtual environments. Through the use of AI algorithms, developers can create realistic and visually impressive assets in a shorter period of time than would be required using traditional methods. This reduces production costs and allows for more experimentation and iteration during the design process, leading to better results (Vionix Studio, 2023).

Narrative design, an essential component of game development, has also been enhanced by the use of generative AI. AI-powered tools such as ChatGPT can produce dynamic dialogue, adaptive plotlines, and customized content, creating more immersive and engaging experiences for players. This offers new opportunities for interactive storytelling and greater player involvement in shaping the story (Mind Studios, 2023).

The impact of AI extends to the realm of game and system design, revolutionizing how developers conceptualize and create interactive experiences. AI-powered tools can assist in generating level layouts, balancing game mechanics, and even predicting player behavior to optimize gameplay. This allows designers to iterate faster, explore a wider range of possibilities, and create more engaging and challenging experiences for players. However, it also raises questions about the role of human creativity and the potential for AI to homogenize game design (Bungie, 2023; Christofferson et al., 2023).

However, integrating generative AI into game development also poses challenges and ethical concerns. Issues such as copyright infringement, bias in AI-generated material, and the possible

displacement of human labor need careful consideration. As AI technology continues to advance, it is essential for the gaming industry to balance innovation with responsible implementation (GeekWire, 2023).

This research paper seeks to provide a thorough examination of the implications of generative AI for the gaming industry. Through a review of relevant literature, an analysis of industry professionals' experiences, and a consideration of specific use cases, this study aims to illuminate the potential impact of generative AI on future interactive entertainment (Mind Studios, 2023; Vionix Studio, 2023).

1.1 Purpose of the Study

This research aims to examine the transformative potential of generative artificial intelligence within the gaming industry. Although AI has been an integral component of game development for many years, recent advances in generative AI technologies, such as ChatGPT for text generation and Midjourney for image generation, have presented new opportunities and challenges.

This study aims to explore the current state of these tools, their impact on various stages of game production, and how they influence programming, art design, narrative design, and overall game design. A key focus of the research is understanding how generative AI is reshaping the way games are created.

There are several questions that this paper is focusing on:

- Overall change of game development process under Generative AI influence
- How introducing the generative AI to certain field e.g. programming will influence production time and cost?
- What tasks are generative AI good for, and which tasks it struggles at?
- What are the downsides of using generative AI tools e.g. copyrighted content?

The question of whether these tools truly accelerate development, as proponents claim, or if there are hidden complexities and limitations that need to be addressed will be investigated. Through real-world case studies and interviews with industry experts, evaluation of the practical benefits and challenges of integrating generative AI into the game development process will be conducted.

Another significant aspect of this study is to explore the ethical implications of generative artificial intelligence in the video game industry. As AI-generated content becomes increasingly prevalent, questions regarding intellectual property rights, potential biases in algorithmic systems, and the impact on employment opportunities in creative fields arise. This research will delve into these ethical concerns, aiming to understand the possible consequences of the widespread adoption of AI in game development. Additionally, this research will investigate the broader implications of generative artificial intelligence for the video game industry as a whole. How will these

technologies shape the future of game design, narrative, and player experience? Will they facilitate the democratization of game development, making it more accessible to smaller teams and independent developers, or will they give rise to new forms of creative expression?

Through an examination of the current landscape and possible future trajectories, this study seeks to provide valuable insights into the evolving relationship between AI and the video game industry.

1.2 Knowledge Base

This chapter serves as a foundational knowledge base. It will review the key terminology related to the field in order to gain a comprehensive understanding of the concepts and techniques underlying the integration of generative AI into games. The main terms and their definitions will be reviewed, along with relevant research that has been published in the area. This will establish a necessary foundation, from which it will be easier to analyze and comprehend the main research conducted within this work.

1.2.1 Information Retrieval

The literature review for this study will be conducted primarily through the Jamk University of Applied Sciences library, focusing on scholarly articles, technical documents, industry reports, and books related to AI and game development. The literature review will aim to synthesize existing knowledge, identify research gaps, and provide a theoretical foundation for the study.

The knowledge base for this study will be established through a combination of academic resources and online sources. In addition to the literature review conducted through the school library, the research will utilize the Yandex browser to access open-source information, including articles, blog posts, and research papers available on platforms like YouTube and LinkedIn. These sources will provide valuable insights into the practical applications of AI in game development, industry trends, and the perspectives of developers and researchers.

To collect up-to-date information, the newest research in the field had higher priority, aiming to focus on data collected in the past 5 years to exclude irrelevant and outdated information. The research will utilize a combination of keywords and phrases to access relevant information from various sources. These keywords include "artificial intelligence," "generative AI," "game development," "ChatGPT," "Midjourney," and specific terms related to AI applications in programming, 3D modeling, narrative design, and game design. These terms will be used to search academic databases, industry reports, online forums, and social media platforms to identify relevant research articles, blog posts, and discussions. Additionally, the search will include variations and combinations of these keywords to ensure a comprehensive exploration of the topic.

1.2.2 Terminology

This chapter aims to provide a foundational understanding of key AI concepts and technologies relevant to this study, establishing a knowledge base for subsequent discussions and analyses.

Artificial Intelligence (AI): A broad field encompassing the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. (Russell & Norvig, 2021).

Machine Learning (ML): A subset of artificial intelligence that involves the use of algorithms and statistical models to enable computers to learn from and make predictions or decisions based on data. (Mitchell, 1997)

Generative AI: A specialized branch of machine learning that focuses on creating new content, such as images, text, or music, that is similar in style and structure to existing data. Generative AI models are trained on massive datasets and can then generate original content based on the patterns and relationships they have learned. (Goodfellow et al., 2014)

Gemini: Google's next-generation AI model that excels in logical reasoning, coding, and creative collaboration, with a vast context window.

ChatGPT: An AI chatbot developed by OpenAI that uses natural language processing (NLP) to understand and respond to user prompts in a conversational manner.

Midjourney: An AI program that generates images from textual descriptions, enabling the creation of unique artwork and visual assets.

DALL-E: Another AI image generator developed by OpenAI, similar to Midjourney, with its own artistic style and capabilities.

Co-pilot: An AI-powered coding assistant that helps programmers write code faster and more efficiently by suggesting code snippets and completing lines of code.

Workflow: The sequence of steps involved in completing a task or project, from concept to execution. In the context of game development, it includes the design, creation of assets, programming, testing, and deployment.

These terms are essential to understanding the influence of generative AI on the gaming industry. Generative AI, powered by machine learning and neural networks, is enabling game developers to automate tasks, create personalized experiences, and generate vast amounts of unique content, leading to more efficient and innovative game development processes.

1.2.3 Generative AI

According to the definition provided by Google Cloud, generative AI refers to the use of artificial intelligence to create new content such as text, images, music, audio, and video. This type of AI works by analyzing patterns in large datasets and learning from them.

The neural network, a type of machine learning algorithm, is trained on specific datasets created by humans with relevant labels. Through numerous iterations, the network learns to recognize patterns and generate new content based on those patterns. Only neural networks that have been trained on sufficient data to generate content independently will be considered in this process.

1.2.4 Short History of OpenAI

OpenAI is a company best known for its groundbreaking work in generative artificial intelligence (AI). It was the pioneer of this field, and its efforts have had a significant impact on the industry.

The OpenAI Lab was founded in 2015 to advance AI for the benefit of all humanity. The company has taken a unique approach, publishing its research in open repositories to promote collaboration and transparency.

In 2019, OpenAI registered a subsidiary, OpenAI LP, which allows for commercial activities and profit generation while maintaining the organization's commitment to public good. This move has allowed the company to attract necessary investments for project development and future returns. In 2018, OpenAI unveiled its first major Generative Pre-trained Transformer (GPT) language model, which revolutionized the natural language processing field by demonstrating the capacity to generate coherent, natural text from minimal input. This breakthrough, along with other research and development efforts, sparked innovation in the area and opened new avenues for utilizing AI in various sectors, contributing to the emergence of numerous generative AI solutions for diverse purposes.

OpenAI presently offers various generative AI models, including ChatGPT for text generation, DALL-E for image synthesis, and Codex for code creation. The company is currently testing SORA, a generative video synthesis model.

1.2.5 Chat-GPT and Gemini

ChatGPT and Gemini are two generative artificial intelligence models designed to understand and generate natural language. ChatGPT, developed by OpenAI, functions primarily as a conversational AI capable of understanding and generating human-like text. It is trained on a diverse dataset, which includes licensed data, data generated by human instructors, and publicly available data.

Gemini, on the other hand, is another generative AI model developed by Google DeepMind. It is designed to improve and extend existing language models, such as ChatGPT, by combining advances in machine learning. Gemini aims to provide more accurate and context-sensitive responses. The key distinction of this model lies in its foundation on a novel architecture presented in Ashish Vaswani's paper "Attention Is All You Need", which enables the ability to weigh the significance of individual words within a sentence in relation to one another. Gemini employs reinforcement learning (RL), wherein the agent learns to make choices by taking actions within an environment with the aim of maximizing cumulative rewards. Additionally, Gemini benefits from its integration with Google's ecosystem and available resources.

1.2.6 MidJourney and DALL-E

Midjourney and DALL-E are both AI models designed for generating images from textual descriptions

Midjourney is developed by an independent research lab under the same name, and there is no information regarding the architecture. Midjourney excels in creating stylized and imaginative images, often with a focus on artistic elements. Its applications are primarily in creative fields, such as digital art, graphic design, and advertising, where users generate art, illustrations, concept designs, and other visually compelling content.

Dall-E is developed by OpenAI and adopted transformer architecture from GPT. It generates images from textual descriptions, emphasizing a broad range of realistic and imaginative visuals. Due to its architecture, DALL-E can produce diverse and complex scenes that closely match the given prompts.

The key differences between Midjourney and DALL-E lie in their development and focus, architectural approach, output style, and user base. Midjourney, developed by an independent lab, focuses on artistic and creative image generation. On the other hand, DALL-E, developed by OpenAI, emphasizes a broad range of realistic and imaginative images based on textual descriptions, focusing on generating images that closely match the textual prompts.

1.2.7 Current Researches

The topic of generative artificial intelligence is relatively new, and therefore there are currently very few studies that focus exclusively on this topic. Additional information can be found in publicly available sources, such as media articles and various online platforms.

For instance, JAMK University student Antton Paananen conducted a study examining how ChatGPT impacts programming. While the conclusions were inconclusive, ChatGPT did spend more time on certain tasks, such as in cases involving interaction with previous code and architectural elements, where it struggled and increased development times.

Another study, "Examining the Use of Generative AI by Early Professionals in the Game Development Process," published in 2023, indicates that there is significant skepticism surrounding the use of generative AI. The quote from the article states next:

"A lot of the current AI image generation tools kind of pool in their data just from scraping the internet; scraping Art Station, scraping Sketchfab for art and models and stuff like that, and they use the art. That's the art that they used to create the images, and we just feel uncomfortable using other artists work for our own benefit."

Additionally, according to the study results, many respondents expressed strong resistance to the use of generative AI, a disparity in perceptions of the benefits of the technology, and sensitivity regarding ethical implications. However, there is a potential downside to the use of generative AI. According to a recent survey conducted by Tommy Thompson and published on December 21st, 2023, approximately 46% of respondents used generative AI for professional purposes either as a requirement from their superiors or by personal choice.

At the same time, while around 13% of those surveyed acknowledged the simplification of their workflow and noted the apparent impact of generative AI on their development process, approximately 50% expressed interest in new technologies. This suggests that while only a fraction of people believe in some changes due to AI impact, there are a lot of individuals who are enthusiastic about exploring this technology.

In order to gain a more comprehensive understanding of the situation, it would be beneficial to conduct interviews with individuals who have directly utilized generative AI within the development process. These interviews will allow to identify any potential challenges or limitations associated with this technology. Additionally, existing studies on the topic will be thoroughly reviewed as part of our ongoing research, so our findings with existing literature can be compared on-the-spot and provide a deeper understanding of the subject.

1.3 Acknowledgements

The completion of this thesis would not have been possible without the support and contributions of several individuals and groups. Without their support and encouragement, this project would not have been possible.

Foremost gratitude is extended to Koskenkorva Risto for his invaluable guidance, support, and constructive feedback throughout this research. His expertise and insights have been instrumental in shaping this thesis.

Special thanks belong to all the participants of my interviews. Your willingness to share your experiences and insights was crucial for my research. Your contributions provided a depth of

understanding and perspective that significantly enriched the quality of this study. This gratitude stands for Häußler Alexander, Klar Jonas and Elmasri Nasser.

Deep appreciation is also extended to people, who gathered or published crucial information. Special thanks to Kemppainen Jussi-Petteri, who runs a blog, describing the process of creating games using generative AIs, and Tommy Thompson, for various research in AI in games, and for sharing the findings.

2 Topic Overview

2.1 Justification

The gaming industry, a multibillion-dollar sector, is constantly evolving, driven by technological advances and consumer demand for more exciting and dynamic experiences. According to logrus IT (2023), the computer games market has 3.38 billion users, which is 6.3% more than in 2022, while the market is estimated at \$184 billion. According to the same study, the video game market will only grow and the number of users will reach 3.79 billion people by 2026. Meanwhile, according to the Mordor Intelligence report, the market size is projected to be in the range of \$273 billion by the end of 2024, and \$426 billion by the end of 2029. Therefore, research in the field of AI interaction with the gaming industry is important.

In recent years, generative artificial intelligence has evolved into a transformative technology capable of revolutionizing various aspects of game development and gameplay. The rapid development of AI will affect the interaction of players and game developers with games. Developers will receive tools that will allow them to create more content and a unique gaming experience, and users will be able to evaluate the result of working with these tools.

The following aspects may be affected due to the introduction of AI into the gaming industry:

Innovations in game design and development. Generative AI has introduced new game development methodologies that allow developers to create complex and procedurally generated content that was previously impractical or impossible. This technology allows to create vast game worlds, diverse character models and intricate storylines, increasing the depth and variety of the gaming experience.

Improved gaming experience. AI-based content creation can significantly increase player engagement and satisfaction by offering a personalized and adaptive gaming experience.

Economic and market implications. The introduction of generative AI into the gaming industry has economic consequences that affect the cost of production, time to market and overall competitiveness in the market.

Ethical and social considerations. Integrating generative AI into games raises important ethical and social issues. Issues such as intellectual property rights require careful study.

Technological advances and future trends. As generative AI continues to evolve, it is extremely important to understand the current state of the technology and its further development trajectory. This article will review the latest developments in the field of artificial intelligence algorithms and their application in games, as well as provide a forecast of possible future developments.

Studying the impact of generative AI on the gaming industry is an important and timely study. When considering real cases of using AI in various fields, it will be possible to find out certain trends that exist now, as well as try to predict the future of the gaming industry and its interaction with AI in particular.

2.2 Delineation

This study focuses on the application and impact of generative AI in the gaming industry. It covers a wide range of topics such as the overall acceleration of the process, narrative design based on AI, personalized gaming experience, as well as the economic and ethical implications of AI integration. The research will mainly focus on current developments and case studies, highlighting key technological advances and their practical applications.

In particular, this study will consider:

The use of AI in programming. How Programmers interact and which tools they prefer in the development process.

AI-driven narrative Design. The role of generative AI in creating adaptive and branching storylines that respond to player choices and actions, increasing the depth and replayability of games.

Personalized gameplay. The use of artificial intelligence to adapt content to the individual preferences and behavior of players. This section will explore the possibilities of AI for such interaction, as well as examples of existing similar solutions.

Economic Implications. A study of how generative AI affects the economics of game development, including cost reduction, time savings, and market competitiveness. This will include case studies of gaming companies that have successfully integrated AI into their development processes.

Ethical and social aspects. The study will address ethical issues such as intellectual property rights, potential distortions in AI-generated content, and the impact of AI on employment in the gaming industry. It will examine the balance between innovation and ethical responsibility.

In order to limit the study to non-verbal things, this study is excluded:

Applications that do not use generative AI: Basic AI used to control the behavior of non-player characters, enemy AI in games, and other applications that do not use generative AI are outside the scope of this study.

Detailed Technical analysis: Although the study will discuss the technologies and methods used, it will not delve into the detailed technical foundations of artificial intelligence algorithms, instead focusing on their application and impact.

Non-entertainment sectors: The scope of application is limited to commercial video games and does not apply to serious gaming or gamification in non-entertainment sectors such as education, healthcare, or vocational training.

2.3 Definition

Generative artificial intelligence refers to a subset of artificial intelligence technologies designed to create new content and data that mimic patterns observed in existing datasets. In the context of the gaming industry, generative AI involves the use of advanced machine learning techniques such as neural networks and deep learning to autonomously create diverse and complex game resources.

Key concepts include:

Neural networks and deep learning: These are artificial intelligence methods in which artificial neural networks based on the human brain learn from large amounts of data. In games, neural networks can be trained on various types of game content to create new, original content that matches the learned patterns. For example, in the Echoes of somewhere project, the main character and the environment were created based on images generated by AI.

Artificial intelligence-driven narrative design: It involves the use of artificial intelligence to create adaptive and branching storylines that change depending on the player's decisions and actions. Artificial intelligence systems can dynamically generate dialogues, plot twists, and character development, ensuring that each play through is unique.

An important clarification is that generative AI differs from traditional AI used in games such as enemy behavior algorithms or pathfinding systems in that it focuses on creating new content rather than optimizing existing game mechanics. Despite this, this paper will consider the possibilities of generative AI for creating traditional ones, to emphasize the possibilities and provide an alternative vision for creating such content, but this is a fundamental topic that deserves a separate study.

2.4 Hypotheses

This study aims to explore the impact of generative artificial intelligence on the gaming industry through a number of hypotheses. These hypotheses will serve as a guide for research and a basis for analyzing data and drawing conclusions about the impact of generative AI on game development. Based on a preliminary review of the literature and familiarization with the capabilities of AI and trends in the industry, the following hypotheses are proposed:

- Generative AI significantly improves the efficiency of game development by reducing the time and resources needed to create complex game resources.
- Games that use generative AI provide a higher degree of personalization and adaptability, which leads to increased player engagement and satisfaction.
- The impact of generative AI on the gaming industry will lead to lower production costs and increased profits for developers and publishers.
- Integrating generative AI into game design raises ethical issues, including concerns about intellectual property, algorithm bias, and the potential bias of creative workers.
- The continuous development of artificial intelligence technologies will enhance the effect of the first four hypotheses.

These hypotheses will be tested using qualitative research methods, mainly based on interviews with industry experts. The results obtained will provide an insight into the transformative potential of generative AI in the gaming industry.

3 History of the Gaming Industry

The gaming industry has undergone significant evolution since its emergence in the mid-20th century. From its humble beginnings as basic programs developed by computer scientists, the software industry has evolved into a lucrative sector that generates billions of dollars and produces highly intricate software. This paper examines the historical development of game production, providing a comprehensive analysis of the industry's evolution and delving into the complex structure of contemporary video games. The various components of games, such as the game engine, code, audio, graphics, and narrative, are analyzed to gain a comprehensive understanding of how they collectively contribute to the gaming experience. (Wolf, M. J. P., 2008)

The origins of the gaming industry can be traced back to the early 1950s, when scholars started conducting experiments with basic computer games. An early example of a tic-tac-toe game was

"Bertie the Brain," which was created by Josef Kates for the 1950 Canadian National Exhibition. In 1952, A.S. Douglas developed "OXO," a tic-tac-toe game, as a component of his doctoral dissertation at the University of Cambridge. (Wolf, M. J. P., 2008)

In the 1970s, the emergence of arcade games laid the foundation for the commercial gaming industry. "Pong," which was developed by Nolan Bushnell, one of the co-founders of Atari, in 1972, is widely recognized as the inaugural video game that achieved significant commercial success. The success of this game popularized arcade gaming and led to the development of other iconic games such as "Space Invaders" (1978) and "Pac-Man" (1980). (Wolf, M. J. P., 2008)

In the late 1970s and early 1980s, the first home video game consoles were introduced. The Magnavox Odyssey, which was released in 1972, was the first home console. However, it was the Atari 2600, released in 1977, that brought about a significant revolution in home gaming. The cartridge-based system of the Atari 2600 enabled players to effortlessly switch games, which greatly contributed to its immense popularity. (Wolf, M. J. P., 2008)

In the early 1980s, the gaming industry experienced a significant setback known as the video game crash of 1983. Consumer interest and confidence experienced a significant decline due to a saturated market and the presence of low-quality games. Companies such as Atari experienced significant financial setbacks, leading some to question the longevity of the video game industry. (Wolf, M. J. P., 2008)

The 1990s were characterized by intense rivalry among major console manufacturers, notably Nintendo and Sega. The introduction of the 16-bit Sega Genesis in 1989 and the Super Nintendo Entertainment System (SNES) in 1990 ignited what became known as the "console wars." During this period, well-known franchises such as "Sonic the Hedgehog" and "The Legend of Zelda" emerged, with each platform competing for market dominance. (Wolf, M. J. P., 2008)

During the mid-1990s, there were notable technological advancements that emerged, particularly in the realm of 3D graphics. The PlayStation, which was released in 1994, and the Nintendo 64, which was released in 1996, utilized 3D graphics to enhance the level of immersion in gaming. Games such as "Final Fantasy VII" and "Super Mario 64" demonstrated the capabilities of 3D gaming and established groundbreaking benchmarks for game design and narrative. (Wolf, M. J. P., 2008)

During the late 2000s and 2010s, there was a significant rise in the popularity of mobile gaming. In 2008, the introduction of Apple's App Store revolutionized smartphones, turning them into convenient gaming devices. Games such as "Angry Birds" and "Candy Crush Saga" have gained popularity among a wide range of players, increasing the accessibility of gaming. By the mid-2010s, the mobile gaming market experienced significant growth, surpassing revenues generated by traditional console and PC gaming. (Wolf, M. J. P., 2008)

3.1 Structure of Modern Games

The different elements of a game, such as the game engine, code, graphics, audio, and narrative, interact in intricate and dynamic ways to produce a cohesive and engaging gaming experience. It is crucial for both developers and players to have a comprehensive understanding of how these elements interact.

Game Engine as the Central Hub

The game engine functions as the central hub that integrates all other components. The necessary infrastructure is provided for rendering graphics, processing audio, and executing the game's logic. Major game engines such as Unreal Engine and Unity provide a variety of built-in tools and features that support this integration. (Taken from JAMK University of Applied Sciences studies, 2020-2024)

1. **Graphics Rendering:** The game engine utilizes its rendering system to transform graphical data, such as textures, models, and lighting, into the visual elements displayed on the screen. This requires the use of intricate algorithms to guarantee accurate and efficient display of graphics, while upholding optimal frame rates and visual quality.
2. **Audio Processing:** The audio engine, which is a component of the game engine, handles various aspects of audio such as sound effects, background music, and voice acting. It guarantees the synchronization of audio cues with visual events and player actions. For instance, the sound of footsteps varies depending on the surface the player character is walking on, which enhances immersion.
3. **Physics and Animation:** Game engines frequently incorporate physics engines that replicate lifelike movements and interactions within the game world. This is essential for animations, as they depend on these simulations to create natural movements for characters and objects. Physics engines are responsible for managing collisions, gravity, and various other physical interactions to ensure a consistent and realistic behavior.

Code: The Foundation of Game Logic

The code forms the foundation of the entire game, establishing the rules, mechanics, and interactions. The code, written in languages such as C++, C#, or Python, is responsible for:

1. **Game Mechanics:** The core gameplay mechanics, including movement, combat, and item usage, are implemented using code. This logic determines how players engage with the game world and how the game reacts to their actions.
2. Code also regulates the behavior of non-player characters (NPCs) and enemies through AI algorithms. These algorithms are responsible for determining the reactions of non-player characters (NPCs) to player actions, their ability to navigate the environment, and their interactions with other elements within the game.

3. Scripting: Scripting languages, commonly found within the game engine, enable designers to create and modify game events, cutscenes, and interactive elements without making changes to the core codebase. This flexibility allows for quick iteration and precise adjustment of gameplay elements.

Graphics: Visual Fidelity and Immersion

Graphics play a crucial role in game development, covering a wide range of elements such as character design, environment creation, and special effects. These visual elements have various interactions with the game engine and code.

1. Asset Integration: Artists generate assets such as textures, models, and animations, which are then imported into the game engine. The engine's rendering pipeline processes these assets, applying lighting, shading.
2. Real-time Rendering: Contemporary game engines facilitate real-time rendering, enabling dynamic lighting, shadows, and reflections. The graphical effects are dynamically calculated, taking into account the current state of the game and the actions of the players.
3. Optimization is a crucial aspect in ensuring smooth gameplay. Techniques like level of detail (LOD), occlusion culling, and texture streaming are employed to effectively manage the rendering workload and uphold optimal performance.

Audio: Enhancing Atmosphere and Feedback

The role of audio design is crucial in creating an immersive experience. The atmosphere is greatly enhanced, and players receive crucial feedback through the use of high-quality sound effects, music, and voice acting.

1. 3D Audio: Numerous games employ 3D audio techniques to establish a perception of space and direction. The positioning of sounds in the game enables players to accurately determine the origin of various sounds, such as footsteps or gunfire, in relation to their own position. This promotes a sense of immersion and aids players in navigating the game world.
2. Dynamic Music: Music in games frequently adjusts to the player's actions and the events of the game. As an illustration, the background music's intensity may rise during combat and subside during exploration. The dynamic adjustment is controlled by the game engine, which activates specific audio tracks depending on the game's state.
3. Sound Effects and Feedback: Sound effects offer instant feedback for player actions, such as the sound of a weapon firing or an object being picked up. The effects are synchronized with the game's events through the audio engine to ensure they align with the visual and gameplay cues.

Narrative: Storytelling and Player Engagement

The narrative effectively integrates all the game elements, offering a comprehensive context and driving the player's actions. The narrative interacts with other components in the following manner:

1. **Scripted Events:** Plot points and cutscenes are carefully planned and activated at specific moments during gameplay. These events typically incorporate a blend of animations, dialogue, and special effects, all carefully coordinated by the game engine.
2. **Dialogue Systems:** Dialogue between characters is facilitated by branching dialogue systems, which enable player choices and offer various outcomes. These systems are seamlessly integrated into the game's code, allowing them to have a profound impact on the storyline and game world in accordance with player decisions.
3. **Environmental Storytelling:** The story is conveyed through a combination of visual and audio elements within the environment. For instance, a dilapidated structure accompanied by haunting melodies and scattered documents can evoke a feeling of historical occurrences without the need for direct storytelling.

Example 1: Combat Mechanics

1. The game engine is responsible for detecting player input, such as pressing a button to initiate an attack, and then processing the corresponding action.
2. The code executes the attack logic, determining the damage dealt by considering the player's stats and the enemy's defenses.
3. **Graphics:** Illustrates the attack animation and visual effects, including sparks or blood splatter.
4. The audio component of the game includes the sound effect of the weapon making contact with the enemy, as well as any corresponding changes in the background music.
5. **Narrative:** Provides updates to the storyline if the combat is part of a larger quest or mission.

Example 2: Exploration and Puzzle Solving

1. The game engine is responsible for managing player movement and interaction with objects in the environment.
2. **Graphics:** The environment is updated in real-time as the player explores, with changes in lighting and dynamic object movement.
3. **Graphics:** The environment is updated in real-time as the player explores, with changes in lighting and dynamic object movement.
4. **Audio:** Offers ambient sounds and music that dynamically adjust according to the player's location and actions.
5. **Narrative:** Incorporates clues and story elements into the environment to effectively guide the player through the narrative.

4 Overview of Artificial Intelligence in the Gaming Industry

4.1 Classification of AI

Artificial intelligence can be categorized according to its levels of complexity and functionality, as well as the methods used for its implementation. Organized according to varying degrees of intricacy and practicality:

1. Limited or Inadequate Artificial Intelligence: Engineered for specialized functions, such as facial identification, voice-activated virtual assistants (e.g., Siri, Alexa), and personalized suggestion algorithms (e.g., Netflix, Amazon) (Brownlee, J., 2019a).
2. General or Strong AI refers to a theoretical form of artificial intelligence that has the ability to carry out any intellectual work that a human is capable of. This particular form of artificial intelligence has not yet been developed, although it is the ultimate objective of ongoing AI research. (Bostrom, N., 2014)
3. Superintelligent AI refers to artificial intelligence that exceeds human intelligence in every domain. Currently, this is a theoretical concept that gives rise to a multitude of ethical and philosophical inquiries. (Tegmark, M., 2017)

Based on the methods used to put a plan into action:

1. Rule-based artificial intelligence relies on pre-established rules and logical structures to make decisions. Illustrations encompass proficient systems that offer medical guidance derived from a repository of knowledge. (Russell, S., & Norvig, P., 2021)
2. Machine Learning: - Algorithms acquire knowledge from data in order to generate predictions or make judgments without the need for explicit programming for each specific activity. There are various subcategories that fall under this category (Alpaydin, E., 2020):
 - a. Supervised Learning: Algorithms acquire knowledge from data that has been tagged.
 - b. Unsupervised Learning: Algorithms detect patterns in data that does not have any labels.
 - c. Reinforcement Learning: Algorithms acquire knowledge by engaging in interactions with the environment, where they get either rewards or penalties.
3. Deep Learning: - A form of machine learning that employs neural networks with multiple layers. Some examples of applications include the ability to identify and analyze images, as well as the ability to understand and comprehend human language. (Goodfellow, I., Bengio, Y., & Courville, A., 2016)
4. Hybrid Systems: These systems integrate many methodologies and technology to accomplish more intricate objectives. Autonomous vehicles employ machine learning, sensors, rules, and deep learning to facilitate navigation and decision-making. (Gaur, L., Singh, A., & Khari, M., 2021)

According to applications and fields of use:

1. Natural Language Processing (NLP) refers to the field of artificial intelligence that focuses on the interaction between computers and human language. It encompasses tasks such as translation, text analysis, and the development of chatbots. (Jurafsky, D., & Martin, J. H., 2021).

2. Computer Vision: - Artificial intelligence used to identify and comprehend visual data, such as in medical diagnosis or self-driving cars. (Szeliski, R., 2022)
3. Robotics: - Artificial intelligence used to operate robots capable of executing physical tasks. (Siciliano, B., & Khatib, O. (Eds.), 2016)
4. Expert Systems: - Programs that simulate the decision-making abilities of experts in specialized domains, such as medicine or finance. (Jackson, P., 1998)

Generative AI is categorized as a subtype of both machine learning and deep learning. Its main objective is to generate novel and authentic data that closely mimic the training data.

Technological advancements and methodologies:

GANs (Generative Adversarial Networks) (Brownlee, J., 2019b):

1. The user did not provide any text. The system consists of two neural networks, namely a generator and a discriminator, that collaborate with each other. The generator produces novel data, while the discriminator assesses their veracity. Both networks undergo simultaneous training, resulting in enhanced quality of the generated data.
2. Autoencoders employ neural networks to condense data into a concise representation (encoding) and then restore the data from this representation (decoding). Variational Autoencoders (VAEs) are a sophisticated form of autoencoders that have the ability to generate novel data.
3. Transformers are extensively utilized in the field of natural language processing. Illustrations encompass GPT (Generative Pre-trained Transformer) models, which possess the capability to produce text, facilitate language translation, and execute various other text-oriented assignments.

Tasks:

1. One use of Generative AI is text generation, where the AI system is able to generate written content. Models like as GPT have the ability to generate text by using provided input. They may produce articles, poems, stories, and also respond to inquiries.
2. Image Creation: - GANs have the ability to produce lifelike representations of individuals, objects, or environments that do not actually exist in the real world. Notable examples include DeepArt and DALL-E.
3. Music Generation: - AI has the ability to generate original musical compositions in different genres by analyzing existing music.
4. Speech Synthesis: - Generative models have the ability to produce voice messages that seem natural, which are commonly employed in voice assistants and text-to-speech (TTS) systems.
5. Code Generation: - Models such as GitHub's Copilot employ transformers to automatically produce code by utilizing comments and incomplete code snippets.

4.2 The Generative AI Tools for Game Production per Specialty Field

Generative AI techniques are becoming essential in game production, allowing for quick creation of diverse and high-quality content. These tools encompass a wide range of specialized areas, including as game design, character design, animation, audio production, narrative development, and coding. Below is a summary of how generative AI is utilized in each of these domains:

Game design and the creation of game content:

1. Tools for generating procedural content (PCG):

Houdini is a robust program used for procedural creation, enabling developers to generate intricate and elaborate worlds, landscapes, and levels. The system employs algorithms to automatically generate diverse content, so improving the potential for replayability and minimizing the need for manual design work. (Houdini., n.d.)

L-systems are a mathematical framework employed to generate plants, trees, and other organic structures by procedural methods. It is especially beneficial for constructing realistic surroundings. (Antoniuk, I., 2019)

2. Level Design:

Dungeon Alchemist: Utilizes artificial intelligence to create complex and captivating dungeon arrangements. The platform provides designers with tools to precisely define characteristics and limits, enabling the construction of distinct and customized level designs. (Dungeon Alchemist., n.d.)

Creation of characters and assets:

1. GANs (Generative Adversarial Networks):

Artbreeder is a tool that uses GAN technology to create high-quality character portraits and concept art by combining numerous photos. It facilitates the rapid creation of various character designs. (Artbreeder., n.d.)

DeepArt is a Generative Adversarial Network (GAN) tool that generates artistic styles and textures. It is helpful for producing distinctive visual assets. (DeepArt., n.d.)

2. 3D Model Generation:

NVIDIA GauGAN is AI technology that transforms basic sketches into highly realistic images. These images can serve as accurate visual references for the creation of 3D models. (NVIDIA., n.d.)

Animation and motion capture:

1. Animation powered by artificial intelligence (AI):

Mixamo: Offers computer-generated animations for three-dimensional (3D) figures. The platform provides an extensive collection of pre-existing animations and the capability to apply them to personalized characters, simplifying the animation procedure. (Mixamo., n.d.)

DeepMotion utilizes artificial intelligence to generate authentic character animations based on basic input gestures or visual references, hence boosting the naturalistic movements of characters within video games. (DeepMotion., n.d.)

2. Procedural Animation:

Cascadeur: An artificial intelligence tool designed to facilitate the creation of lifelike and dynamic movements by animators. It achieves this by modeling physical interactions and forces. (Cascadeur., n.d.)

Sound engineering and composition:

1. Tools for creating music through generative processes:

AIVA (Artificial Intelligence Virtual Artist) is capable of generating creative music by utilizing specified genres and moods. It is a valuable tool for producing distinctive and flexible soundtracks. (AIVA., n.d.)

2. Sound Effect Generation:

Sonniss employs artificial intelligence to generate and manipulate sound effects, offering a diverse selection of top-notch audio resources that can be customized to suit different gaming settings and situations. (Sonniss., n.d.)

Generation of narratives and dialogues:

1. Text Generation and Interactive Narratives:

ChatGPT is a sophisticated language model developed by OpenAI. It has the ability to produce dialogue, stories, and interactive text-based content. It is especially beneficial for constructing interactive and diverging storylines in video games. (OpenAI., n.d.)

2. Quest and Mission Generation:

Promethean AI: A sophisticated artificial intelligence technology that aids in the development of narrative content, such as quests and missions, by producing concepts and frameworks that align with the game's story and design limitations. (Promethean AI., n.d.)

Programming and Development Support:

1. Code Generation and Assistance:

GitHub Copilot is a code completion tool that utilizes artificial intelligence, developed collaboratively by GitHub and OpenAI. It aids developers by providing suggestions for complete lines or blocks of code while they are typing, using context and previously acquired coding patterns. Copilot offers extensive support for several programming languages and frameworks, resulting in a substantial acceleration of the development process and a reduction in the cognitive burden placed on developers. (GitHub Copilot., n.d.)

2. The Gemini software developed by Tabnine:

A coding assistant powered by artificial intelligence that employs deep learning models to anticipate and provide code completions. It enhances productivity by offering precise and context-sensitive code recommendations, assisting developers in composing more concise and optimized code. (Tabnine Gemini., n.d.)

5 Overview of the Research Methodology

This study employs a mixed-methods approach, combining qualitative and quantitative research methodologies to investigate the impact of generative AI on the gaming industry, with an

emphasis on a qualitative approach. Empirical research is also used in this work according to Mapping Research Methods by Lähdesmäki, T. (2010). The data were obtained through interviews with specialists in various areas of game development. There were few interviews, four in total, one for each direction, making the research methodology more suitable for a case study. This methodology involves investigating one case in-depth to produce detailed and intensive information. In this paper, the cases are the experiences of using generative AI by professionals.

The qualitative component involves in-depth interviews with industry professionals across various domains, including programming, 3D design, narrative design, and game design. These interviews aim to gather firsthand insights into the practical applications, benefits, challenges, and ethical considerations associated with using generative AI in game development. The methodologies are slightly different depending on the field studied, therefore it is better to look into each field to determine the approach.

Programming: The research methodology for this section involves a semi-structured interview with a professional game developer specializing in mobile game programming who utilizes AI tools regularly. The interview will explore the specific AI tools used, the perceived impact on productivity and workflow, and the challenges and limitations encountered. Additionally, the findings from the interview will be compared with existing research papers on the influence of AI on programming productivity to assess the generalizability of the developer's experiences.

3D Design: This section will focus on a case study of a 3D artist who actively uses AI in game development. The methodology involves analyzing the artist's blog posts, presentations, and other relevant materials to understand their workflow and the specific AI tools they employ. The study will compare the artist's AI-assisted workflow with traditional 3D design processes to identify differences, advantages, and disadvantages. The aim is to assess the impact of AI on efficiency, creativity, and the overall quality of 3D assets.

Narrative Design: The research methodology for this section involves a semi-structured interview with a developer of chat-based games who has recently integrated AI into their projects. The interview will explore the specific AI tools and techniques used, the impact on narrative design and player experience, and the challenges and opportunities associated with AI-driven storytelling. The study will also examine the differences in workflow before and after the adoption of AI tools to assess their impact on efficiency and creativity.

Game Design: This section will investigate the use of AI in generating references and design documents for game development. The methodology involves reviewing relevant literature and case studies to understand how AI tools are being used to assist game designers in conceptualizing and documenting game mechanics, levels, and overall design. The study will also explore the potential benefits and limitations of using AI in this context, like working with references and utilizing them for final result.

Social and Ethical Influence: The research methodology for this section involves analyzing an open source research information and social media discussions to gauge public sentiment and identify potential ethical concerns related to AI adoption in the gaming industry both from the perspective of players, developers and companies.

By employing this multifaceted approach, this study aims to provide a comprehensive and nuanced understanding of the impact of generative AI on the gaming industry, encompassing technical, creative, social, and ethical dimensions.

6 The Impact of AI in Various Fields of Game Production

6.1 The Influence of AI on Programming in Games

In everyday life, there is a lot of conflicting information about AI in the field of programming. Some people argue that AI can write code, explain it, and more. Others disagree, saying that AI has not yet reached the level of maturity needed for programming, and that it does not know how to think like a human programmer, so what it produces may not work and, if it does work, it will need to be corrected.

To better understand this topic, Häußler Alexander, a programmer with 15 years of experience in the video game industry, was interviewed. Häußler A. learned about AI from social media, friends, and colleagues around the time it started to become popular, and recently started using it regularly. In this interview, Häußler A. discussed two specific AI tools, Gemini and Copilot.

6.1.1 Coding with Gemini

Gemini is similar to Chat-GPT in several ways, but there are some key differences. When comparing the latest versions as of the time of this writing, these are ChatGPT 4o and Gemini Ultra 1.0. Gemini has built-in tools for checking AI responses, which make it more convenient for various studies. ChatGPT has more capabilities for text generation, and it can also generate images, making it more creative. Therefore, it may be more beneficial to use Gemini when working with code. During the interview, was decided to create a fishing mini-game with mechanics similar to those found in Stardew Valley for the sake of the experiment. The dialogue and generated solution will be included in the Appendix 1 for the project.

It is important to note that, due to the fact that the AI database is constantly being updated, specific fishing mechanics have not been prescribed. Instead, a request was submitted stating that a fishing feature in our game, that is similar to the fishing minigame found in Stardew Valley, using C# and the Unity development platform is required. In other words, the reference material was a strict guideline for the game. Fortunately, Gemini was able to provide the necessary information to support our request. The first attempt was not successful. Instead of generating AI code, the

system generated a detailed description. However, this has already proved useful. First, it is possible now to determine exactly how well AI understands the reference. Second, this reference is now in dialogue with AI, meaning that AI has easy access to this information, and it is no longer necessary to manually prescribe mechanics.

The second question in this dialogue with AI was: "Can you provide me with a code snippet?". The AI response included the following:

- A warning that providing full code for the entire game would be extensive
- An example C# script to get started with the basic mechanics
- A code for the basic mechanics, and a summary of the mechanics.

This part was commented.

Okay, he says that this is quite extensive in order to create the entire game. Therefore, it is consistent with these core mechanics. I suppose it starts from a certain point. In that point, it makes sense, because there is no way you can simply copy and paste everything, which is great because it also teaches you exactly where everything is.

However, why can't code that the AI provides be copied in the project? The main reason for this is as follows: The AI does not understand the context and style of programming the game. For example, Alexander typically uses private variables when creating a game, but if these need to be modified in Unity, they are marked as serialized fields. This is a common practice, but the AI initially did not use this style. After submitting another request as part of ongoing dialogue with the AI, "In our codebase, the inspector variables are private and serialized fields. Please adjust the code accordingly." Gemini was then able to update the variables. It's a simple example, but it illustrates the amount of effort that needs to be put into this kind of code.

As a software developer or, let's say, a senior developer on a project, I would not want all developers to write code in their own way. Typically, we agree on a specific coding style to ensure that everyone feels comfortable in any part of the codebase. However, this is something that AI cannot provide, unless it is connected to your codebase and given the appropriate context. For example, GitHub's AutoPilot does a good job of this.

During our discussion, two key points were agreed on. If a user is unsure how to start writing code, Gemini can provide significant assistance. The generated code structure proved to be satisfactory, and, in general, if one is constantly making edits, the AI will be able to generate code that can be reused in the future. However, the code, particularly when multiple individuals are contributing to it, should be written in a straightforward manner. Häußler A. compares the code with English, where when familiarizing yourself with the code, it immediately becomes clear what its purpose is.. With this methodology, including all necessary edits for Gemini, Häußler A. estimates that the

time required to write the code itself will be comparable to the time spent correcting an erroneous query.

To date, Gemini has garnered mixed reviews. On the one hand, it may assist in initiating the coding process, but its actual utility is questionable. Nevertheless, Häußler A. has shared insights into the utilization of AI and his own approach to utilizing it. There are certain tasks that even an experienced professional may not always be aware of. For instance, an efficient sorting algorithm for a two-dimensional array, in which the highest values are located in the center. The Gemini provided a concise description of the algorithm and the actual code when generating a response to this query. Gemini and other AI systems do not verify the code, therefore it may contain errors or not function at all, yet even a description of the algorithm may assist to quickly find a solution.

Another instance of utilizing Gemini is to stay abreast of current trends. Sometimes, it is quicker to ask Gemini for the fastest approach to locate an array element at present than to search for an answer on Stack Overflow, as Häußler A. noted. Another instance is working with SQL. Although Häußler A. did not frequently work with it, such a Gemini request as "How would you accomplish such a task" may provide an idea, reminding of various techniques, and so forth. These issues do not arise frequently on a daily basis. On the contrary, daily activities are often more automated.

During the Interview the following hypothesis was proposed: "AI can be beneficial for novice programmers or individuals who are not directly involved in programming, as even despite its shortcomings, if the code functions with adequate efficiency, it may be utilized, as the end-user is not concerned about the code as they will never see it."

Häußler A. partially concurred with this hypothesis, but offered the following comment:

That is a common scenario if you are an independent developer or a student, and all you are concerned about is the final outcome. If you do not intend to return to the project after it has been completed, it might be okay. However, if you wish to maintain and continue to develop your project, AI can cause issues, as you are combining different elements without a clear structure. AI cannot provide you with an overarching architectural design. It may provide guidelines on coding principles and other aspects, but it does not produce a complete architecture for you.

Let's say, for example, you create a fishing game using only Gemini and other relevant tools. After playing the game, you notice that every time you catch a fish, it jumps out of the water twice, which is not intended. This could be due to an issue with the AI implementation. You are not aware of the reason and the likelihood of being unaware of the reason is higher if you have used AI to write everything, as you have never had a human write any line of code, so the intent is not in the code. You cannot even consult anyone on your team, as nobody knows where to begin looking, since no one has ever written a jump-starting code. And then, you must examine your entire code to find this error. If the bug occurs within the same script, the AI may be

able to find and fix it on its own. However, if the AI generates two different scripts that must interact according to the game's logic, problems may arise. This is because the AI does not adhere to the architecture of the game and script interaction can be disrupted.

A classic example is the interaction between a red barrel and a bullet. According to the game logic, the red barrel should explode if the bullet's velocity exceeds a certain threshold. Because the AI generates scripts for the bullet and red barrel separately, their architectures are different, and even if they work independently, due to this architecture issue, the interaction between the two objects may be incorrect or nonfunctional.

In summary, based on this subchapter, such conclusions can be made: If unsure about what to do, which is not often the case, guidance can be sought, but due to the company's code style and the context of the codebase, it is not usually beneficial to request specific code. Instead, it may be faster to write your own code and modify it as needed.

In such instances, which occur infrequently, Gemini can provide with an idea or a starting point for your work. Additionally, Gemini can offer up-to-date information on the principles of operation for things rarely interacted with, such as another programming language, or provide reminders of those principles. If the code is generated entirely by AI, it may work, however, due to issues with the architecture, errors could occur. Finding and fixing these errors could take a significant amount of time, as no one has written the code, and even identifying the problem could be time-consuming. Antton Paananen conducted research on the impact of chat-GPT on programmer productivity. The results of his study were as follows: "The research did not reveal a clear increase in productivity when dealing with code-related issues. Rather, limited access of ChatGPT to the context surrounding the code may have led to a decrease in productivity.

However, there was a possible minor increase when developing a new feature that did not require ChatGPT to be aware of prior code." This connection can be confirmed by the words of an interviewee, who stated that without the Gemini context, chat-GPT can provide working parts of code, but without the project's context and architecture, the result may not be compatible with the rest of the project and may require more time for fixing than writing the code manually.

6.1.2 Copilot

In the previous chapter, copilot was repeatedly mentioned, what is it? GitHub Copilot has emerged as a groundbreaking tool that is reshaping how programmers conceptualize, write, and refine code. Leveraging the power of AI, Copilot acts as a virtual pair programmer, offering intelligent suggestions, completing lines of code, and even generating entire functions based on context and the developer's intent.

Due to the non-disclosure agreement, Häußler A. could not show the real project, so the code generated by Gemini in the previous chapter was used to demonstrate the capabilities of Copilot. The code was placed in Visual Studio connected to Unity, thus the real development process was simulated.

Immediately after the code migration, Visual Studio applied color palettes to the text, and it was noticed that one logical if operation did not have a parameter. Instead, there was a comment.

```
// Check if fish is caught (logic based on fish icon position and
bar value)

if (/* Check if fish icon is within green bar and progress bar is
full*/)
{

//Fish caught (implement logic for success) isFishing = false;
escapeTimer = 0f;

}
```

Häußler A. erased this comment, after that the logical operation had no input data. At this point, the Copilot analyzed the code and, based on the comment on the previous line, suggested the following solution:

```
if (Vector3.Distance(bobber.position, fishIcon.position) < 0.5f &&
progressBar.value >= 1f)
```

In addition, Copilot noticed that the program contains a "StartFishing" method and assumed that the user would want to have a method "StopFishing" in its program and suggested automatically generated method.

And that's what you want as a coder. You don't want to go to a different window. Do these kinds of things, you want it right in the working environment while you're working on it

Image 1 GitHub Copilot offers StopFishing Method

```

public void StartFishing()
{
    isFishing = true;
    escapeTimer = 0f;
    progressBar.value = 0f; // Reset progress bar
    // Reset fish icon position (optional)
}

public void StopFishing()
{
    isFishing = false;
    escapeTimer = 0f;
    progressBar.value = 0f; // Reset progress bar
    // Reset fish icon position (optional)
}

```

The main advantage of copilot is its awareness of the structure and architecture of the code. AI suggestions are often useful. Häußler A. argues that although it is difficult to estimate in numbers from memory, it feels like 90% of AI assumptions are successful to appear in the code. At the same time, the analysis of proposals occurs seamlessly, the process does not distract the programmer from writing code, and if the solution proposed by the AI turns out to be incorrect, it can always be discarded.

For example, you have an A variable that equals one. And variable B equals two, and you say It's a "sum" equals ... A plus B! So it figures these things out by the way, because I named it "sum", right, I could have called it "dif" and it would give A minus B. So that is a very small example, but that works very well even on bigger cases, I feel that had the biggest impact on my coding style in the past two years.

In addition to the atomic generation of code signatures, copilot has a function similar to Gemini, a dialog window where AI requests can be sent. So, for example, in this window, the AI can be asked to analyze the code and provide an explanation for this file. Copilot copes with this task because, as mentioned above, it has access to the project and is aware of the architecture and content. In this dialog box, Alexander requested a sorting algorithm similar to what he previously requested from gemini, the result of which is a two-dimensional array should contain large values closer to the center. Copilot described the basics of the algorithm, as well as offered several ready-made solutions to choose from.

Choosing one of the proposed solutions, it did not impress Häußler A., this was due to the fact that the function seemed strange, but not because of the content, but the "Style" of writing. During the interview the code was not tested.

If I don't care about this function, I can just ask the Copilot to write a unit test for that function, and if it does exactly what I want to do. I can be happy with it, right? Otherwise, it's still important that at least a human brain went over it and understood what is actually going on.

Returning to the function of automatic code writing, Häußler A. clarified that this function has the greatest impact when a lot of the same type of actions need to be performed in the code, for example, the same addition, but with 10 variables or test units for several functions, AI understands such parts of the code very quickly and often offers exactly what is needed, therefore, saves a lot of time. As a reminder, according to preliminary estimates, Copilot correctly guesses the context in 90% of cases.

The cleaner the code copilot interacts with and the clearer the architecture of the project, the more often the copilot will provide useful parts of the code. *"It is surprisingly good getting context in the area you're working"*, comments Häußler A..

At the end of the interview, Häußler A. noted that in some situations the copilot does not cope in the best way, saving 10% of the time, Sometimes when a lot of the same type of code needs to be written, it copes better and saves 60% of the time. The average score given by the interviewer was 20%. On average, the speed of writing code has accelerated by 20%, which is a good result. Remember that AI can recall various algorithms, help in finding solutions, analyze and explain the work of the code, and the advantages of AI become obvious.

6.1.3 Machine Learning Enhancement

This chapter is one step away from the generative AI to discuss the potential of machine learning in the game industry that Open AI showed.

In addition to directly programming games, artificial intelligence has the potential to enhance the user experience in existing games. To support this assertion, two projects, OpenAI (2019) that is known as "Emergent Tool Use from Multi-Agent Interaction", and OpenAI (2017a), dedicated to "Dota 2" will be examined.

Starting with "Emergent Tool Use from Multi-Agent Interaction." OpenAI, (2019). Two teams, Red and Blue, each consisting of two characters, play the game according to the following rules:

1. The agents can move in the x, y, and z directions, as well as rotate around the z-axis.
2. Agents can see objects within their field of vision and a frontal cone.
3. Agents can detect distance to objects, walls, and other agents using a sensor similar to lidar.
4. Agents can grab and move objects in front of them.

Agents can lock objects, which can only be unlocked by the team that has locked them. In addition, the blue team has a strategic advantage. During the initial few seconds, the red team is stationary, allowing the blue team time to prepare their strategy.

In this study, the specifics of the training process will not be delved into, but rather focus on the results achieved. The agents have learned various strategies to employ. Once the blues have found a means to evade the reds' detection, the reds have developed countermeasures to counteract this tactic. The blue agents have learned how to construct shelters, while the red agents have mastered methods to bypass these defenses. This approach to AI training has potential applications in PvE (Player vs Environment) content, where the AI-controlled opponent can engage the player on equal footing.

Moving to the next OpenAI project. DOTA 2 is a highly complex and competitive multiplayer online battle arena (MOBA) game where two teams of five players each strategize and battle to destroy the opposing team's Ancient while defending their own. According to OpenAI (2017a), the bot played against professional players in a 1v1 match according to standard rules. The article described what had been achieved and stated that the bot had learned the game through self-play without using imitation learning or tree search.

The learning process was lengthy and required significant resources. According to OpenAI (2017b), which provides further details, the training schedule for the bot and its description are disclosed. Here is a short part of the article:

To provide context, approximately 15% of players have a MMR (Matchmaking Rating) below 1.5K, while 58% are below 3K and 99.99% below 7.5K.

On March 1st, we achieved our first reinforcement learning results in a simplified Dota environment. A Drow Ranger learned to kite an Earthshaker that was hardcoded into the system. May 8th: The 1.5k MMR player reported that he was improving faster than the bot.

- *Early June: Beat the 1.5k MMR opponent.*
- *June 30th: Won most games against the 3k MMR player.*
- *July 8th: Barely won the first game against the semi-pro 7.5k player.*
- *August 7th: Defeated Blitz (6.2k ex-pro) 3-0, Pajkatt (8.5k professional) 2-1, and CC&C (8.9k professional) 3-0. All agreed that Sumail could find a way to beat the bot.*
- *August 9th: Beat Arteezy (10k professional, top player) 10-0. He said Sumail could figure it out.*
- *August 10th: Defeat Sumail (8.3k professional, top 1v1 player) 6-0, who said it was unbeatable, but played the same bot from August 9 and lost 2-1.*
- *August 11th: Beat Dendi (7.3k ex-professional, former world champion and crowd favorite of the old school) 2-0. The bot had a 60% win rate against the August 10 bot.*

Based on this data, the bot is able to defeat more than 99% of opponents after 5 months of training. What practical applications of this method of AI usage can be identified? Unlike other

examples of AI in this paper, this is not a cognitive tool designed to improve the development process. Rather, it is a neural network that, after undergoing numerous iterations of self-learning, is able to compete with human players. Similar applications of AI could be used to generate PVE (player versus environment) events of varying levels of difficulty or bots for solo play. The outcome would depend on the specific game and the goals to be achieved, but such an approach has the potential to enhance player engagement with the game experience

6.2 The Influence of AI on Narrative Design

6.2.1 Narrative Using Generative AI

In this chapter the question how using ChatGPT can be an important tool in creating exciting scenarios and narratives for the gaming industry is considered. Using the GPT Chat to develop interesting and interactive game plots, as well as creating dialogues will be focused. This approach allows to automatically create storylines, characters and dialogues in game content using artificial intelligence. The use of GPT Chat not only helps to explore the possibilities of artificial intelligence in the gaming industry, but also provides a unique opportunity to interact with advanced technologies. It is hoped that the demonstration of the use of GPT Chat will serve as an example of how artificial intelligence is actively being introduced into various aspects of the game world, including the creation of plots and narratives..

Starting with the Narrative part. Assume that the game is being developed based on a real historical figure. In this case, chat-GPT can answer your questions and help find important information. Elmasri N., the man who developed the Mesagisto text game, shared information shedding light on this topic. For more information about this person's work, see the chapter on text games. Now it is important to understand that it took the Elmasri N. development team about 6 days to find and track the activities of a certain person before AI, but after the start of using the GPT chat, the team began to take half as much, that is, 3 days. In addition, Elmasri N. argues that the more knowledgeable a person is in history, the more content dedicated to him there is on the Internet, as well as "alternative history", which are easier to search with AI, since it not only finds the necessary information, but also can generate content itself. By "alternative history," offshoots from the real story that did not happen in real life but could have occurred are meant. Turning to examples, the Wolfenstein series of games unfolds in an alternate version of history, in which the Nazis won World War II. With proper use of GPT chat, narrative designers can not only get inspired using AI, but also check certain facts and find the necessary information faster.

Chat-GPT were asked to provide a beginning of the plot and setting for game. The Dialogue with Chat-GPT is in the Appendix 2.

Of course, this is just the beginning, but for 5 minutes of time spent, this is a good result. Chat-GPT analyzed the principles by which the regime works and described the moral dilemmas that may underlie the story.

Speaking of chat-GPT and its capabilities, it is worth considering that this is a powerful tool, thanks to which the genre of text games has been revived, In such games the plot is the basis of gameplay and is generated by AI. The GPT chat not only controls the plot, but also communicates directly with the player, to which an entire chapter is devoted.

However, in large games such as AA and AAA projects, even if chat-GPT can be a useful tool, technical limitations cannot allow AI to think through all the details of the plot and offshoots. These parts are developed by a person, and sometimes by a group of people. According to the credits, at least 20 narrative designers worked on Baldur's gate 3. In addition, each starting character had its own writer assigned to it, who was responsible only for this character, So that the writer could get into it and give out more realistic reactions and interactions with the world, as was said on one of the company's streams during open testing of the game

However, chat-GPT, despite its limitations, can be useful in other aspects. Ordinary people, merchants, and "extras" in games can be significantly improved with the help of AI. A mod for Skyrim VR was recently released, which allowed players to interact with NPC`s. In 2023 User "Art from the Machine" on reddit shared a mod that integrates chat-GPT into the game, thus allowing to freely communicate with any NPC. A Skyrim fan has used the Whisper speech recognition system in his project, which converts voice into text. This allows the player to ask the characters questions of interest, rather than typing them using the keyboard. In addition, the Modder used the xVASynth program, which uses AI algorithms to synthesize the desired voice, converting it from text. Due to this, the characters respond to the player with their voice. A video appeared on the YouTube channel with the same name as redditor`s name, "Art from the Machine", which is also linked from reddit. This video demonstrates that characters like merchants understand what time it is. To the question "what time is it now?", the trader correctly indicates the right time. To the question "is the store open?" (This question was not raised in the video, but from the context of the answer, it can be assumed that this question was asked), the merchant replies that yes, the store is open, and also explains that the store is open, it will be open for several more hours, and also reminds the player of the schedule the store's work. The merchant can also identify the item that the player picked up. When asked "what do you think about this blade?", the merchant correctly explained that the player had picked up an iron sword, as well as told about the enchantment that is applied to the blade. However, if a merchant is encountered in a bar, he will correctly recognize the location, and also clarify that the bar is a good place to take a break from working in the store.

In this integration, the introduction of AI into the NPC has significantly improved the player's interaction with the NPC. The NPC has information about the following:

- Time of day
- The role of a character in the world
- Items that the player interacts with
- The location where the character is located
- Character's Environment

In addition, the video shows how the guard, who travels with the protagonist, can play his role, keep up a conversation around the campfire or give battle advice.

Now the mod has technical problems. The voices sound metallic, in addition, the characters can think about the answer for a while. These problems will be solved with the development of artificial intelligence technologies. AI is now available that can change a person's voice to that of a celebrity or anyone else if provided with a sample of the recorded voice. Services such as ElevenLabs allow to synthesize text-to-speech today. Khatsenkova, S. (2023) made news article that scammers are already using this technology, extorting money from the relatives of victims using this service. Therefore, it can be concluded that with proper configuration, the voice of the characters can be voiced using a neural network today. As for the speed of responses. Yes, now AI needs time to recognize what a person is saying and generate an answer, but the longer the AI develops, the less this delay will be, which means that games using these technologies may appear soon.

6.2.2 Adaptive Storylines and Content Personalization

In this subchapter, the development of text talkers using GPT chat technology in the gaming industry, their potential and challenges, as well as the prospects for the development of this area, will be examined.

In the modern gaming industry, text-based "talkers" occupy an important place, offering players a new type of gaming experience based on interactive dialogues and historical contexts. These games differ from traditional forms of entertainment in that they focus on text messages and interaction with virtual characters through them.

Text "talkers" are a genre of games in which the player interacts with the game world and characters through written text. In such games, the player often assumes the role of the main character and interacts with other characters, asking them questions, choosing actions and making decisions that affect the development of the plot and the outcome of the game.

In recent years, interactive text games have become increasingly popular among gamers. This increase in popularity is driven not only by the development of technologies that allow to create more interactive content, but also by the demand for deep and exciting stories. Games of this type provide players with the ability to influence the story and outcome of the game, making their experience more individualized and unique.

Interactive text games play an important role in the development of gaming culture, as they contribute to the formation of literary and interactive skills in players. They stimulate the imagination, develop analytical and critical thinking skills, and promote personal development through engaging with a variety of emotional and moral dilemmas.

The GPT (Generative Pre-trained Transformer) chat technology is an artificial intelligence product developed by OpenAI. It is based on powerful neural networks trained on a huge amount of text data from the Internet. GPT chat is able to generate natural and coherent texts, imitating the style and logic of human communication.

GPT chat technology is actively used in creating text talkers for the gaming industry. It allows developers to create virtual characters that can conduct dialogues with players in natural language. With the flexibility and expressiveness of GPT chat, characters can answer players' questions, keep up conversations, and tell interesting stories, creating a unique gaming experience.

Here are some examples of games that successfully use GPT chat technology to create interactive characters and dialogues:

- "AI Dungeon": This is a text-based adventure game where the player can interact with the virtual world and characters using natural language. AI Dungeon uses GPT chat technology to generate scenarios and character responses to player actions.
- "Spirit AI": This platform provides tools for creating text talkers as virtual assistants, consultants, or game companions. It uses GPT chat technology to create characters with a high degree of realism and interactivity.
- "Siftrun": This is a text-based puzzle game where players interact with a virtual guide as they explore mysterious locations and solve puzzles. The whole game is based on dialogues with the guide, who uses GPT chat technology for his answers and hints.

Trying to predict the advantages and challenges of interactive chats using AI:

Advantages:

1. Interactivity and depth of dialogues: Text talkers using GPT chat allow to create characters that can conduct natural and dynamic dialogues with players. This increases the level of interactivity of the game and makes the game experience deeper and more fun.
2. Flexibility and adaptability: GPT chat technology is highly flexible and adaptable, which allows characters to respond appropriately to various questions and actions of players. This makes the game world more realistic and compelling.
3. Endless content generation possibilities: Thanks to the fact that GPT chat is based on learning from a huge amount of text data, it is able to generate an infinite number of unique dialogues and stories. This makes text talkers almost endless in terms of the variety of content.

Challenges:

- Difficulty creating realistic dialogues: One of the main challenges is creating natural and realistic dialogues that will convincingly convey the characters' personalities and emotions. This requires not only technical precision but also artistic skill in creating content.
- The need for content quality control: Creating text-based talkers requires constant quality control of the content to ensure that the characters' dialogs meet the expectations and standards of the gaming industry. This may require significant resources and effort on the part of developers.

Recently, a person who worked with the character-to-character text dialog division was interviewed. The interview was taken after his speech at the POCKET GAMER CONNECTS HELSINKI 2024 CONFERENCE held in September 2023. Elmasri N. developed the game Mesagisto. At the time of the interview, Elmasri N. had already been part of the gaming industry for 3 years, and Mesagisto had been in development for more than 1 year. In his game, the player communicates with different historical characters, such as Napoleon, and learns various ideologies through dialogue with the characters.

According to Elmasri N., the idea of creating such a game was born even before he got acquainted with AI, but after studying this area, Elmasri N. found out that AI could bring great benefits to his project:

“When I was developing Mesagisto, this is the game we are talking about, I didn't use AI. And the first part of it, it was not AI at all, it was just traditional development.

However, when I started, when people started talking about it, I thought, okay, let me take a look. And so why? Because it's based on dialogue and this game, Mesagisto, that, okay, ChatGPT is dialogue, and my game is dialogue. Let's see if there is any, any things that can be used. And then I found that it greatly improves the development cycle for me. So it was, there was a clear benefit of using it because what I was asking for scriptwriters when I had the idea, was asking them, to translate this decision tree into dialogue. But I found that I can ask the same thing exactly for ChatGPT. And I can ask it and get the result instantaneously. And I can get the result, I can do it iteratively what I don't like, I just get it on another form. And I even was able to make the counterarguments that I needed” An example explaining his words sounded like this: Everyone blames Napoleon for the rapid attack on Russia, which is a strategic mistake. In the case of the GPT chat, I can directly ask why Napoleon did this, and the GPT chat will respond interactively.

In other words, our prediction about Flexibility and interactivity was correct. AIs like ChatGPT are capable of creating an interactive and flexible experience, and since some games generate responses on the go, in real time, the content can be described as unique and endless.

Moving to the challenges. During the interview, the question "How was the program kept in bounds and made to follow the story?" was asked. The answer consisted of 2 parts: Dialog History and Presets.

The dialog history is explained by the fact that chat-GPT inherently does not know how to track the dialog. Each response is based only on the last request. However, for chats to work, the dialog history is sent to chat-GPT with each new request. *"As if you are telling the AI that, okay, this is what we have said before, and you continue on this"* (Elmasri N.)

In addition, there is also a set of presets. At each request, chat-GPT is reminded to *"Answer as if you are Napoleon, in two paragraphs and end your answer with another question"* In this way, ChatGPT can be kept within certain limits, maintain the character, and simultaneously remember the history of conversations.

A little more about the dialog storing system can be found from the video from the Onigiri channel "AI plays mafia" at 19:20-20:10, as well as at 15:15-15:30 solving a problem when the AI was answering for another player, the prompt *"you are such and such a player, answer only from this player's perspective"*, which helped solve the problem (In conjunction with reducing the probability of a line break token in the response). In addition, from 21:20 to 23:45, author talks about reducing the number of tokens. Chat-GPT cannot process an infinite request, and the amount of input data is limited to tokens. Each token is a unique encoding of a word or symbol that chat-GPT can handle. Since chat-GPT, does not remember the history of conversations, and the history of the dialog needs to be reminded, the number of tokens in the "reminder" should be reduced, otherwise, the dialog will break and chat-GPT will not be able to process such a long request. The easiest way to do this is to ask chat-GPT to shorten this "reminder" and it will handle it itself.

These methods also allow responses to be double-checked for inaccuracy and unwanted content to be filtered out before being sent to the user. A different AI or the same chat-GPT can be used for this.

Since text games existed before chat-GPT, it will be interesting to compare how the creation process will change before and after the introduction of AI. Elmasri N. shared the necessary information. For context, it is worth explaining that in this case, only adding a character with a story to a ready-made game is considered. For example, if our chat is already functioning and the player can talk to Robin Hood, Stalin, Oppenheimer, and other personalities, but a dialogue with Napoleon needs to be added, How long it will take before and after and how AI will affect the cost?

Elmasri N. himself answers this as follows:

So what I did, was I compared the development time of the story and how it was reduced. So it was reduced from 42 days to 29 days, so this is a 30 percent change in time. The cost is what I've saved by not hiring, so by reducing the number of people that I'm hiring to do the media, the drawings the videos, and the dialogues. So this is the explanation of the reduction of the cost. And this is the cost per story. So a story,

it took me this time and it cost me this amount and then by using... Chat-GPT and Dali were able to make this reduction.

So, it was found that using chat-GPT not only helps speed up the process but also reduces the cost of production. An important clarification is that the game mesagisto uses not only text but also images for visual interpretation of characters. The Dalle AI was used for the IL media content generation function.

Image 2 Story development without AI

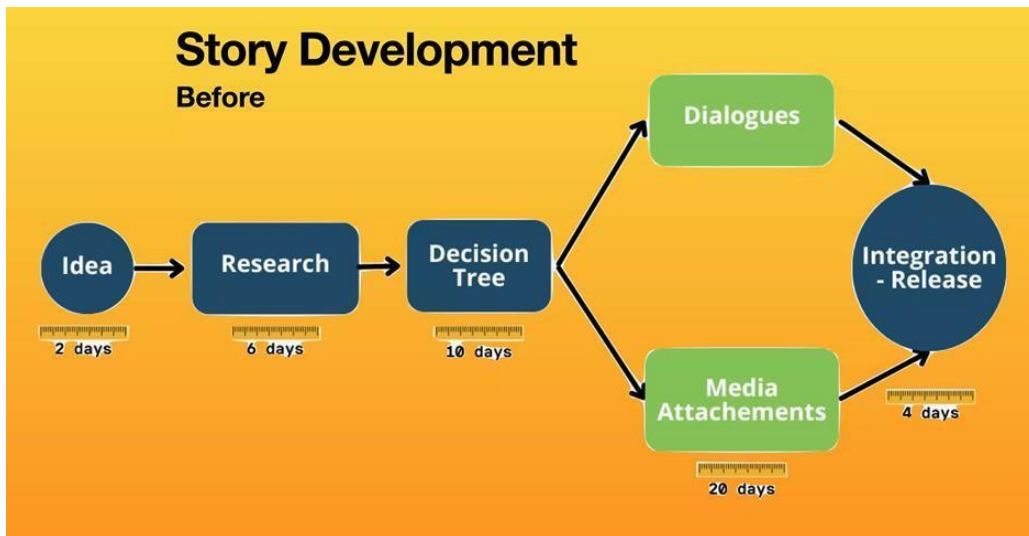
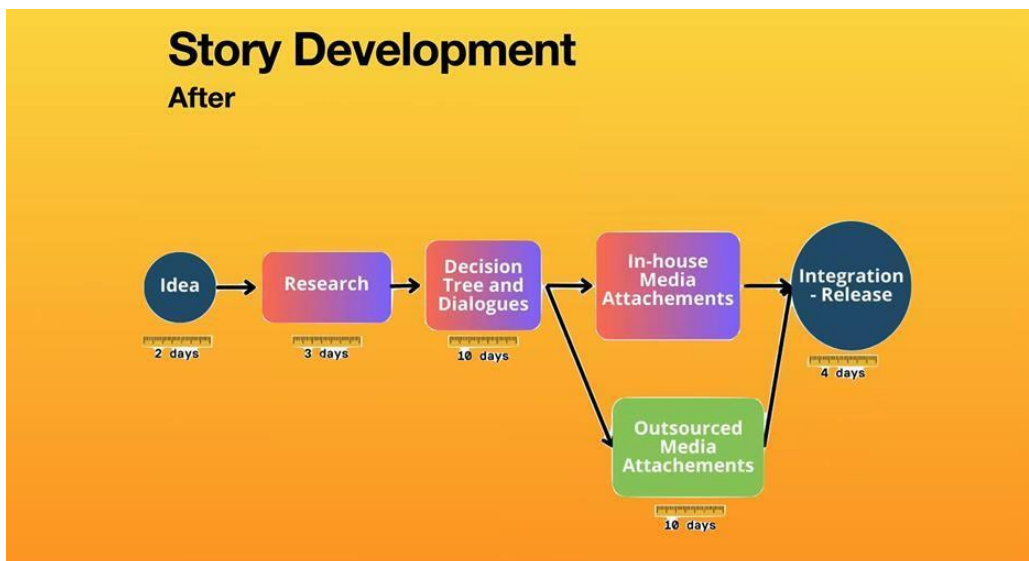


Image 3 Story development with AI



In addition, the development process has also changed. As can be seen from the graphs above, blocks with a red-blue gradient use chat-GPT. The research time has been reduced, because now

information doesn't need to be searched for manually; the chat can be asked directly, and then asked again to check for accuracy. Media production has been halved, because, as the author himself explains, any query can be generated in the desired style, whether oil or pencil, while getting the result instantly and re-generating it if it does not fit for any reason. However, as can be seen, some of the media content remained outsourced. Probably, these are some complex or specialized media parts of the game or UI designers. As the author of UX and UI Designers explains, the neural network cannot replace them yet, however, as can be seen, if earlier artists were in the company, now they went to outsource, which reduced costs by as much as 70%, as shown above.

Summing up, the conclusion of this chapter: Using AI for specific purposes can significantly reduce the cost and time of development, as well as provide the user with a more interactive experience. Of course, additional checks and security are needed, especially when the user is given direct access to AI. Responses need to be checked on the fly, but the neural network itself can also perform this task. Elmasri N. thoughts, based on his hands-on experience with AI, were as follows: the game is still entirely developed by humans, and only a small part relies on AI to develop key decision-making. This part is appreciated by me and is of great value in terms of cost and time. That person also recognized that AI could have a much greater effect in automating the software development process in the future, although this was not the case in his particular situation. The use of AI was primarily intended to automate the content part, such as the development of art and dialogs, which significantly reduced the time spent on these aspects. This person also highlighted the potential for fully integrating AI into the development process in the future, especially in automating software development as a process. The materials provided at the conference and in person emphasize these conclusions.

6.3 The Impact of AI on 3D and 2D design

In this chapter, the impact of AI on the process of creating 3D and 2D content for a game will be examined. Using the example of Kemppainen J., and his blog about creating a point-and-click game called Echoes Of Somewhere. Focusing on the attempt to find out how the process will change under the influence of AI, as well as what advantages and disadvantages will arise when applying this approach.

6.3.1 Mid-Journey, Character Creation with Kemppainen Jussi-Petteri

Before the application of AI is examined as a specific example, assumptions about how the process of creating environments and models will change will be made. Starting with the case of using image generation. The scope of this tool is not limited to themes and styles, which means that anyone can generate any image. However, there is a downside: even if a detailed prompt is sent, the result will be unpredictable. The result cannot be guaranteed to be 100% accurate. This is why some Tools, such as Midjourney, have the option to generate several images at the same time, usually this number is equal to four. For background generation and secondary characters, these disadvantages may not be so important, because image generation is almost instantaneous,

especially when compared with manual sketching and further work with colors. 10 images can be generated, and 1 can be chosen as a compromise, the most suitable one, and if the situation requires, the details can be corrected manually.

Thus, after comparison with real experience, the following conclusion is expected: Accelerated content generation with a slight loss in control of the final result.

Kemppainen J. is a person, who has been working in the gaming industry for 24 years as a game designer, director and 3d artist. Now he is working on the project Echoes Of Somewhere and uses AI in various areas of game creation and has a blog dedicated to the process on his website. In addition to blogs on the channel "Games Now!" the video "Kemppainen J.: Generative AI in Game Development | Games Now!" where the speaker reveals even more details about the creation of the game and, in particular, the impact of AI on the process.

During the conversation at 16: 25-18: 00, the author claims that despite the fact that he set up the prompt in such a way that the character had a color and was generated in a certain style, the author lost control over the generated images.

It's so random what AI gives me for the character. I would do prompts of hundreds of different character variations, and it would never give me the same or similar, and then I would have to choose from the list of characters like which is the one that I like the most and then go with that.

But the author also claims that the character design process was at an automatic level and compares it to a lego constructor. With the use of AI, the author no longer had to think through the proportions of the character and other details. The author also managed to model a character while watching TV, for example. Yes, Kemppainen J. has more than 20 years of experience in this field, so it is obvious that as in any other field, when gaining experience and improving the mastery of a certain skill, the process is optimized and becomes easier, but if the author himself claims that AI was able to make the task easier, the same proportions that are used to improve the quality of from now on, there is no need to think about it, it is worth recognizing that such a conclusion takes place.

According to the same conversation, the process of creating a 3d model was as follows:

- Character prompting
- Character modelling
- Character UV Map
- Character morph-based texture projection
- Character texture post processing
- Character rigging
- Character detail sculpting

According to "Workflows for Creating 3D Game Characters " by Terävä T. (2017), a graduate of Tradenomi (AMK), the process of creating 3d models is as follows:

- Concept art
- 3d modeling
- UV mapping
- Texturing
- Rigging
- Animation

It is noticeable that the process is generally the same, with the exception of 1 difference. In the classic process, there is concept art, and in the new one, using AI, there is character prompting instead. The reasons for this phenomenon are described above, repeating, instead of sketching and thinking through characters, content is generated in the selected direction and regenerated until a satisfactory result is achieved.

The implementation of the first step will be analyzed on the Kemppainen J.'s case. The author did not include the very first generations in his blog and explained the reasons during the conversation. Each character generation is unique, even with the same prompt, the result will always be different, so the author had to find a way to get an image of the same character from different angles. To solve this problem, prompt was written in such a way that the AI gave a character scan, as all 3d artists use in game design and not only.

The next step that the author took was to send the Midjourney prompt Midjourney and get the following result:

Image 4 Generated Character First Draft



Note: By Kempainen J. [Image], From AI assisted graphics: Character modeling. [DevBlog]

Prompt used for Midjourney: cyberpunk point and click adventure game character model sheet turnaround -v 4 -ar 3:2

For the best understanding of what is happening, Analysis of the prompt:

- **cyberpunk point and click adventure game character model sheet turnaround:**
 - **cyberpunk:** This specifies the aesthetic and thematic style of the character, referencing a genre of science fiction that focuses on futuristic, high-tech worlds juxtaposed with societal decay.
 - **point and click adventure game:** This indicates the style of the game for which the character is being designed. Point-and-click games typically involve storytelling, exploration, and puzzle solving.

- **character model sheet turnaround:** This refers to a collection of drawings representing a character from multiple angles (usually front, side, and back). This is used by animators and game designers to maintain consistency in appearance throughout production.
- **-v 4:** This is likely a versioning parameter. In MidJourney, **-v** followed by a number can dictate the version of the model or algorithm used to generate the images. Different versions might produce variations in style or detail.
- **-ar 3:2:** This specifies the aspect ratio of the image. **-ar** stands for "aspect ratio," and **3:2** is a common photographic and print format. This sets the dimensions of the image relative to width and height, making it wider than it is tall

The author comments on his beginning as follows *"The first result was very promising. Naturally it was not good for me as it was in black and white, but model sheets usually are."*

After some minor changes in the prompt , it began to look like this: "cyberpunk point and click adventure game character, full body, model sheet turnaround, full color, two- thirds view, front::4 view and back view -v 4 -ar 3:2". Analysis of new variables:

- **full color:** Indicates that the images should be in color, rather than monochrome or limited palette.
- **Two-thirds view, front::4 view, and back view:**
 - **Two-thirds view:** This might be a typo or a misunderstanding. Typically, "two-thirds view" isn't standard in design terminology; perhaps it meant "three-quarter view" which shows the character slightly turned but not fully sideways, offering a dynamic perspective.
 - **front::4 view:** This appears to be another error or non-standard term. If "front view" was intended, it would mean the character is facing the viewer directly.
 - **back view:** Shows the character from behind, completing the range of primary perspectives.

It can be noted that despite the fact that the model sheet turnaround implies different angles, the author decided to specify which angles should be used and prescribed this in the prompt.

Image 5 Generated Character Second Draft



Note: By Kemppainen J. [Image], From AI assisted graphics: Character modeling. [DevBlog]
Prompt used for Midjourney: cyberpunk point and click adventure game character, full body,
model sheet turnaround, full color, two-thirds view, front::4 view and back view -v 4 -ar 3:2

Above is an example of what the author was able to generate, and below is the one where he decided to stop. To generate the model below, the prompt was also added: very old and weak man.

Image 6 Generated Character Finale



Note: By Kempainen J. [Image], From AI assisted graphics: Character modeling. [DevBlog]

After that, as the author himself explains, the process is identical as in the case without using AI. When analyzing and comparing with the process proposed by Terävä T., this can be confirmed once again.

6.3.2 Differences Between 3d and 2d design, New Midjourney Feature

It is also important to clarify that point and click games often use 2D environments and characters, and the approach to creating a 2D game will be different. This can be justified by the fact that in this case, a 3D model of the character does not need to be made, it will be enough to generate an image and create sprites based on it, and since the image creation process has become easier, It can be concluded that in the case of a 2D game, development time will significantly accelerate. Kempainen J. notes that the process has already become easier, even when creating 3d models, and since the images are already ready thanks to AI, it will not be difficult to modify them to create 2d sprites.

In addition¹², on March 12, 2018, Midjourney introduced a new feature, namely Character Reference. The lack of this feature in 2023 caused some difficulties for Kemppainen J., more can be found in his blog, where he says that to create a portrait of a character, he described his appearance in detail and generated images until the AI gave him the best result, which then had to be corrected photoshop manually in photoshop. But now this feature is present, unfortunately, the author has not yet commented on this, but it is clear that this is a great step in the future. This function, as described above, is used to create alternative images with the character that was generated before. Due to the fact that this feature is new, its potential in the gaming industry has not yet been revealed, but for a visual novel, for example, a genre of games that are popular with players, this is a great help. The same character can be generated in different poses and activities.

6.3.3 Glimpse in the Background Images

Since Kemppainen J. developed the point and click game, in addition to the character, locations are needed, in which these characters are located. In the specific case of the author of the project, namely the creation of a point and click game where a 3d environment is used, Astor used the generated images mainly for reference, but this is still a huge part of the work. By analogy with creating a 3d character, there is an approach of Kemppainen J. for the 3d environment.

- Location prompting
- Location retouching
- Locations camera reverse engineering
- Location modeling
- Locations combining rooms
- Locations camera moves

When generating a location using AI, the author got excellent results. The author explains that such images can already be used as ready-made ones. However, since Kemppainen J. decided to combine 3d and 2d work continued. It is important to note that the author generated locations several times, and as he himself notices, new images were always different, some images had a door, and others did not. When viewing these images, the author thought in his head about how to rebuild his story so that it would fit this location. Kemppainen J. stated that this is far from an inconvenience, if he did not like the alternative history, he just regenerated the image, but sometimes it was possible to get something into the story that was not originally planned, but could complement the plot, so this is more of a positive moment.

During the second stage, Kemppainen J. opened the image in photoshop and thanks to the AI plugin for this program, he could select a part of the image and regenerate it. Thus, if the image was initially good and inspired the author to make interesting plot improvements but did not add a door that is necessary in a particular place, the author could generate it thanks to this plugin.

Moving to items 3 and 4. Modeling a room based on an image is difficult, difficult, and impractical, so Kemppainen J. went for a trick. He modeled the structure of the room using primitives, they

were necessary for the correct operation of the light, but an AI-generated image was projected onto these primitives. Naturally, this approach causes visual artifacts, but if the camera is placed in the same position in space as in the image, then the picture will lie exactly on the 3d environment and there will be no artifacts. To find this point in space and the angle of rotation of the camera, Kemppainen J. used the fspy tool fspy. This is a tool that is used just for this purpose in the film industry. It allows you to perform reverse engineering of the camera and find out the location and angle of rotation of the camera in the image.

As part of the 4th point, it was necessary to manually generate some parts of the location, for example, if the refrigerator was opened, then the image of the open refrigerator and its contents were drawn manually.

By connecting several locations together, all that remains is to adjust the camera's transition from one point to another as the character moves around the scene. During these transitions, artifacts that were mentioned earlier can be seen, because at the moment of transition, the camera does not match the position as in the picture and the space is distorted. For a solution, instant switching of the camera can be used, but Kemppainen J., thought that a smooth transition is better. Artifacts are only visible for a short period of time, and the camera is still in motion, so most people won't notice them. In addition, due to the movement of the camera, the player gets the feeling that this is a real 3D location and the game looks "more advanced on a surface than it would be".

At the end of this chapter, the time it would take to create such locations without AI will be compared with the actual time spent. Here the indicators differ slightly. In his blog , Kemppainen J. writes the following "I spent 18 hours working on the 3D assets and unity scripting for this prototype. 12 hours on the AI character design/model/uv/morph/textures/rigging – 3 hours on the location camera & mesh – and the rest on game scripting – not including any of the AI prompting.". In other words, it took from 3 to 6 hours to create the locations. "On minimum the AI saved me ... 3 days on the location. As per my estimation.". During the conversation, the author claims that creating a simple location, even in a 2D environment, can take up to a week of time, and with AI, this process has been reduced to a couple of hours. It is worth noting that the blog was written before the conversation, which means that the 3-6 hours he mentioned there were spent on developing and creating a working model for building levels. At the time of the conversation, these tools were configured and the author already had an idea and a plan to adhere to, which means the process is optimized and runs faster, therefore, at the first trial, the process had already been reduced from 3 days to 3-6 hours, but it could be reduced from a week to a couple of hours with the process already configured, it is said at his speech.

6.3.4 Chapter Summary

The results are impressive. AI was able to completely replace the sketching phase, reducing the work by several times, potentially from days to hours. At the same time, the developer needs to find out what is more important for him, full control over his character or small compromises, reducing the time of character design. It should be taken into account that when creating 2d

models for a point-and-click game or visual novel, the time to the final result will be even more reduced due to the lack of further transformation of the 2d image into a 3d model.

The results of creating the environment and back are even more impressive. The process, which used to take a week, now takes several hours. This is a significant difference that cannot be ignored. But a specific example was analyzed when an image or several such images are presented as a 3d space using clever manipulations. For many games, this is not necessary. As Kemppainen J. points out. The images generated by the AI can be used as a background immediately after generation. With the correct prompt template and a pre-configured system the process of adding a new location, for example, in such genres as a visual novel, can take several minutes. In addition to hiring an artist who will draw and detail the background for several days, the AI is able to generate many locations almost instantly, and details that will be critical for the plot can be added to the Photoshop add-on during the processing stage, also in a few minutes.

6.4 Generative AI From the Perspective of a Game Designer

For this topic, Klar Jonas, a Game designer with 10+ Years of experience in the gaming industry and approximately 8 years of experience in the field of production and design.

I've been very interested in AI since 2020. When AI was kinda in its early stages, but only since 2022 I used first AI tools for game-design related things, mostly with design documents and stuff.

For production purposes Klar J. used Sapphire Automation, by combining Sapphire Automation with some AI tools like Chat-GPT Klar J. was able to automate some things. The process is not described in detail; in the interview, the focus was mainly on the design part of Jonas's experience.

The relationship of Klar J. with Ai tools in the design field were described as "Fetch me information"

It's mainly the same kind of stuff but more like Fetch me information, so it's been very handy. Finding information with the help of the Ais and summarizing stuff from the internet, especially with the reason models that have access to the Internet.

The most important part for Jonas, as a designer is to know the source of the information that generative AI is providing, that's why the improvements of Chat-GPT and Gemini in the field of browsing are the most important for Jonas.

The design workflow can be described like this:

- Finding references

- Analyzing references
- Create tests, mockups, wireframes or prototypes

Finding references sometimes can be tedious work. Any tedious work can be improved by AI, that's what was learned from Häußler A. in the programming chapter, and Elmasri N. stated that finding information is easier with AI in the chapter dedicated to chat games. That's how the process was described by Jonas:

If the AI tools get the source right, it's very cute, usually because it can actually summarize really well. So for fetching information, it's been really healthy. So it helps with designing documents and putting together research material. It's mainly good for that. Of course, you have to double-check its work, but for that purpose, it just makes it a bit easier. So since you don't have to go scour the web yourself.

Throughout the conversation, some parts of generative AI influence were commented. For example, Klar J. mentioned that for starting a design document generative AI is handy because it can create a foundation, which is edited depending on the needs. Chat-GPT doing a great work there, however creating mockups is a pitfall for some generative AI.

Midjourney and DALL-E as AI have to be trained on some real images, created by human, but since the mockups and wireframes are usually not publicly available AI can't access this information, that's why creating mockups and templates is not efficient using generative AI. That's were our best reasonable suggestion during the interview.

They're not able to do the work yet that you can do in Photoshop, but it's getting there and the moment it's in a point where you can give it clear references, and it can build from those references based on your design document. I'm really waiting for that.

7 Ethical Repercussions of Using Generative AI

It is crucial to follow research ethics in the quickly developing fields of artificial intelligence and game creation. Plagiarism, which is the practice of passing off someone else's words, ideas, or data as one's own without giving due credit, raises serious ethical issues. Plagiarism concerns especially affect the gaming business, which usually depends on innovation and intellectual property. There may be a greater chance of inadvertent plagiarism as generative AI tools proliferate. Because of the remarkable similarity these tools might produce to already published works, it is imperative that developers make sure all sources are properly cited and acknowledged.

By producing new characters, storylines, and even entire game settings, generative AI systems have the potential to revolutionize the game creation industry. But using these instruments brings up important moral and legal issues, especially with regard to copyright.

Because generative AI frequently draws inspiration from preexisting works, concerns over the uniqueness of its outputs and the possibility of copyright violation are raised. Legal issues may arise, for instance, if an AI program creates a gaming character or plot that strongly mimics an already-existing copyrighted work. In order to negotiate these challenges, developers must make sure that AI-generated material respects current copyrights and, when required, obtain the proper licenses.

Taking into account how AI-generated content may affect original authors is a necessary part of using generative AI responsibly. The intellectual property rights of authors, artists, and other content producers whose creations might be used as training data must be respected. Developers should, ethically, make an effort to be transparent about the process of creating AI-generated content and make sure that any repurposing of copyrighted resources is duly approved. In order to foster trust and guard against abuse, the European Commission (2021) highlights the importance of accountability and openness in AI ethics.

7.1 Possible Unemployment

The First Industrial Revolution's revolutionary transformations are frequently likened to the emergence of AI. Significant technical developments that transform economies, cultures, and industries are represented by both periods. Comparing AI's possible effects on employment to the First Industrial Revolution's historical background is necessary to appreciate both the prospects and difficulties.

During the First Industrial Revolution, which took place between the late 1700s and the early 1800s, rural economies gave way to industrialized ones. The steam engine, mechanized textile production, and iron-making techniques were among the major technological advances. The labor market was drastically changed by these developments.

- **Employment Creation and Displacement:** Manufacturing and farm mechanization resulted in a large loss of jobs. The percentage of the workforce employed in agriculture in the United Kingdom decreased from 35% in 1800 to 22% by 1850. (Broadberry, 2010). On the other hand, the industrial sectors flourished. Manufacturing employment increased, and new industries like coal mining and transportation arose.
- **Skill Shifts:** As the need for unskilled manufacturing labor expanded, so did the demand for skilled craftspeople. With time, additional abilities were required, especially those pertaining to the upkeep and operation of machinery.

Similar to the technologies of the First Industrial Revolution, artificial intelligence is expected to have a big impact on the labor market. Automation, robotics, and machine learning are just a few of the AI technologies that are already revolutionizing a number of industries.

- **Automating Routine Tasks:** Jobs that require a lot of repetition are most susceptible to automation. Up to 800 million jobs could be eliminated by automation by 2030, forcing 75 million–375 million workers to change occupational categories (Manyika, J., 2017).
- **Emergence of New Roles:** AI is predicted to give rise to new job categories, just like the First Industrial Revolution did. The demand for professionals with expertise in AI, data science, and AI ethics is rising. According to the World Economic Forum (2020a), automation and artificial intelligence might provide 97 million new employment globally by 2025.
- **Skill Shifts:** The kind of abilities that are in demand are evolving. Digital literacy, sophisticated problem-solving techniques, and interpersonal skills are becoming increasingly important. In a labor market driven by AI, workers will require ongoing education and training to remain relevant.

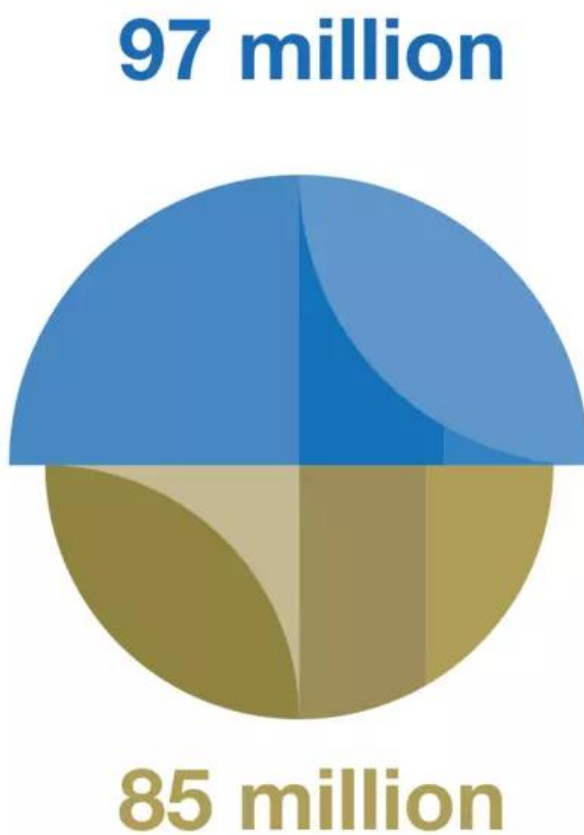
AI's possible impact on employment bears many similarities to the First Industrial Revolution. These times represent significant technology changes that upend current job markets, open up new avenues for opportunity, and demand adaptability. While AI has drawbacks, such as the loss of jobs and the requirement for new skills, it also has advantages in terms of creativity and increased productivity. Together, policymakers, educators, and business executives can make sure that the workforce is prepared for these changes.

Image 7 Infographic “Job landscape changes”



Job landscape

By 2025, new jobs will emerge and others will be displaced by a shift in the division of labour between humans and machines, affecting:



Growing job demand:

1. Data Analysts and Scientists
2. AI and Machine Learning Specialists
3. Big Data Specialists
4. Digital Marketing and Strategy Specialists
5. Process Automation Specialists
6. Business Development Professionals
7. Digital Transformation Specialists
8. Information Security Analysts
9. Software and Applications Developers
10. Internet of Things Specialists

Decreasing job demand:

1. Data Entry Clerks
2. Administrative and Executive Secretaries
3. Accounting, Bookkeeping and Payroll Clerks
4. Accountants and Auditors
5. Assembly and Factory Workers
6. Business Services and Administration Managers
7. Client Information and Customer Service Workers
8. General and Operations Managers
9. Mechanics and Machinery Repairers
10. Material-Recording and Stock-Keeping Clerks

Source: Future of Jobs Report 2020, World Economic Forum.

Note: Future of Jobs Report 2020. (World Economic Forum, 2020b)

8 Ethics

All guidelines and moral principles were closely adhered to when writing this thesis, and only information that the respondents gave their permission to be utilized and published was used. This research has been carried out with the utmost care and accuracy, adhering to the protocols approved by JAMK. Results are accurately recorded and presented in an honest manner. There was strict observance of the confidentiality of the information that the interviewees supplied. Little personal data has been collected, and effort has been taken to adhere to a defined retention policy for personal data. Request for permission was made to utilize the names of the interviewees in the final iteration of the study.

The systematic classification of data was made easier by the use of qualitative software tools for the thematic analysis. The research recognized and tackled possible constraints, so yielding superior understanding of the function of information security in mergers and acquisitions.

Every stage of the research procedure took the JAMK ethical guidelines into account. Keeping the interviewees private and discreet was crucial because they were all bound by non-disclosure agreements. This made sure they couldn't unintentionally talk about any data security breaches they had encountered. It would be improper ethically to intentionally or inadvertently divulge private information. As a result, it was essential to make sure the interview questions were written appropriately and to make it clear to the subjects that the purpose was not to obtain private information.

It was the researcher's responsibility to notify the research subjects of any potential dangers and the precautions taken to lessen any harm if it was thought that they might not be aware of the risks the study could bring. This technique, referred to as "informed consent," entailed disclosing confidentiality policies and ethical guidelines at various phases of the inquiry. The researcher provided an explanation of a number of topics at the outset of the study, including the purpose and context of the investigation, their own motivations, the methodology, and the ethical standards that would be adhered to. (Koskinen, I., Alasuutari, P., & Peltonen, T., 2005).

9 Reliability

The study utilized existing literature and research findings on the research issue that have been published during the last 5 years. Four interviews were undertaken with experts to gain insights into the drawbacks of employing generative artificial intelligence in the gaming sector. Due to the research team's limited experience in conducting interviews and the current surge in interest in generative AI, the research may not be entirely credible. This research depends on the insights of experts in the field and existing research. However, it is important to note that the lack of a substantial evidence base on this topic could potentially impact our findings. However, using the research as a basis, future authors exploring similar topics can collect information regarding the application of generative AI in game development. They can also consider the suggested

recommendations based on the study's findings, which are likely to be confirmed in the near future.

10 Results

Generative AI provides game developers with unparalleled chances to boost creativity and expedite production procedures. Generative AI enhances gaming experiences by enabling the autonomous creation of a wide range of complex game elements, such as characters, environments, dialogues, and stories. Tools like as GitHub Copilot are revolutionizing the coding process for developers by providing intelligent recommendations, automatic code completion, and the capability to construct complete functions based on the given context. This expedites the development process and enables developers to efficiently address intricate jobs.

The utilization of generative AI has a substantial influence on the development of 3D models and virtual worlds. AI algorithms enable developers to efficiently produce lifelike and visually striking assets in a reduced timeframe compared to conventional approaches. This not only decreases manufacturing expenses but also fosters increased exploration and refinement during the design phase, resulting in superior outcomes. Rapidly producing top-notch resources allows developers to concentrate on improving and enriching the entire gaming experience.

AI-driven tools like ChatGPT have transformed narrative design by generating dynamic dialogue, adaptable plotlines, and personalized content. This feature enhances the level of immersion and captivation for players, providing fresh possibilities for interactive narrative. Generative AI has the ability to customize storylines based on the specific decisions made by individual players, hence increasing player engagement and creating a more customized and captivating gaming experience. The capacity to adapt is especially advantageous when it comes to crafting branching narratives and intricate character interactions that react to the choices made by the player.

Generative AI impacts game and system design by aiding in the creation of level layouts, fine-tuning game mechanics, and forecasting user behavior to enhance gameplay. This enables designers to iterate more rapidly, explore a broader spectrum of possibilities, and develop more captivating and demanding experiences for players. Nevertheless, it also prompts inquiries regarding the function of human ingenuity and the capacity of AI to standardize game design. The industry must carefully assess the delicate equilibrium between harnessing AI for improved efficiency and safeguarding human innovation.

Although generative AI offers many benefits, its integration into game production poses several problems and ethical considerations. Matters such as copyright infringement, partiality in AI-generated content, and the potential replacement of human work necessitate meticulous deliberation. With the continuous advancement of AI technology, it is crucial for the gaming business to maintain a balance between innovation and safe deployment. It is essential for ethical

and sustainable growth to guarantee that AI-generated content upholds intellectual property rights and refrains from perpetuating biases.

The research encompasses the examination of case studies and the conduct of interviews with industry experts to assess the tangible advantages and obstacles associated with the incorporation of generative artificial intelligence into the process of game creation. Concrete instances in the real world serve as illustrations of both the capabilities and constraints of these technologies. For example, AI technologies have demonstrated the ability to expedite specific elements of development, but they can also create intricacies that necessitate meticulous handling. Industry experts emphasize the significance of maintaining a collaborative approach, in which AI supplements rather than supplants human innovation and knowledge.

11 Conclusion

The incorporation of generative AI in the gaming sector signifies notable technological progress that has revolutionized multiple facets of game creation. This study has examined the diverse effects of generative AI on game production, encompassing its uses, benefits, and difficulties in areas such as programming, 3D and 2D design, narrative development, and overall game design.

ChatGPT, Midjourney, and DALL-E are advanced artificial intelligence systems that have transformed the process of content production. These tools empower creators to autonomously produce intricate game components, characters, environments, and narratives. These technologies have greatly improved creativity and productivity, enabling more engaging and interactive game experiences.

Programming tools like GitHub Copilot and Gemini offer intelligent code suggestions and automatic code generation, hence expediting the development process. Nevertheless, this study emphasized the significance of comprehending the context and upholding coding standards to guarantee the efficiency of code generated by artificial intelligence.

The utilization of generative AI in narrative design has had a significant impact. AI-powered technologies are able to create dynamic and diverging tales that adapt to player choices, hence boosting player engagement and enriching the depth of storytelling. In addition, the utilization of AI-generated dialogues and content personalization has enhanced the level of interactivity and player-centric experiences.

Although there are many advantages, incorporating generative AI into game production presents various ethical and practical obstacles. Factors such as intellectual property rights, potential biases in AI-generated content, and the effect on employment in creative industries need to be thoroughly examined. The substitution of human labor with AI automation gives rise to

apprehensions regarding job stability and the necessity for game developers to acquire new skill sets.

In the future, the ongoing development of generative AI technologies has the potential for additional progress and breakthroughs in the gaming industry. Developers are advised to appropriately embrace these tools, striking a balance between innovation and ethical issues. By capitalizing on the advantages of generative AI while acknowledging its constraints, the gaming industry can successfully merge AI technologies, resulting in enhanced gaming experiences for gamers and cultivating a more enduring and innovative development atmosphere.

12 Recommendations for Game Developers Based on Research Results

Following guidelines are based on authors opinion after completing research and do not pretend to be objective truth or to foresee the future, just the opinion of two people, after conducting their own research in the field, as well as after interviews with people from the field who have already touched on this topic.

1. **Embrace AI Tools:** Game creators should aggressively embrace and incorporate generative AI tools to augment creativity and optimize development workflows. Utilizing tools such as GitHub Copilot for code assistance and Midjourney for image production can greatly enhance productivity and increase the quality of material.
2. **Uphold Ethical Standards:** It is imperative to acknowledge and deal with ethical issues associated with the integration of AI, such as safeguarding intellectual property rights and addressing algorithmic biases. Developers must prioritize transparency and equity in AI-generated content to establish confidence with players and stakeholders.
3. **Allocate resources to enhance skill development:** With the increasing prevalence of AI tools, it is imperative for developers to allocate resources towards continual learning and skill development in order to proficiently utilize these technologies. Gaining a comprehensive understanding of the capabilities and limitations of AI will empower developers to fully exploit its potential.
4. **Achieve a Harmonious Blend of Automation and Human Creativity:** Although generative AI has the capability to automate numerous facets of game development, it is crucial to strike a delicate equilibrium between AI automation and the ingenuity of human creators. Developers ought to utilize AI as a supplementary instrument that amplifies, rather than substitutes, human input and originality.
5. **Promoting collaboration among AI experts, developers, and designers** can enhance the inventive and efficient utilization of generative AI in game development. Interdisciplinary teams have the ability to incorporate a wide range of ideas and experience, resulting in improved solutions.

Game creators can successfully incorporate generative AI into their workflows by adhering to these suggestions. This will result in the creation of more immersive and captivating gaming experiences, while efficiently managing the complexities and advantages associated with this revolutionary technology.

References

Alpaydin, E. (2020). *Introduction to Machine Learning (4th ed.)*. MIT Press.

American Psychological Association. (2020). *Publication manual of the American Psychological Association (7th ed.)*. American Psychological Association.

Antoniuk, I., Hoser, P., & Strzeciwiłk, D. (2019). *L-system Application to Procedural Generation of Room Shapes for 3D Dungeon Creation in Computer Games*.
https://www.researchgate.net/publication/329606369_L-system_Application_to_Procedural_Generation_of_Room_Shapes_for_3D_Dungeon_Creation_in_Computer_Games

Artbreeder. (n.d.). *Create images with AI*. <https://www.artbreeder.com>

Art from the Machine. (2023a). *ChatGPT in Skyrim VR with lip synced voice generation*. [Online forum post]. Reddit.
https://www.reddit.com/r/singularity/comments/12zveq0/chatgpt_in_skyrim_vr_with_lip_synced_voice

Art from the Machine. (2023b, April 25) *ChatGPT in Skyrim VR | Mantella - Lip Sync & In-Game Awareness Update*. [Video]. YouTube. <https://www.youtube.com/watch?v=Gz6mAX41fs0>

AIVA. (n.d.). *AI-generated music*. <https://www.aiva.ai>

Bostrom, N. (2014). *Superintelligence: Paths, dangers, strategies*. Oxford University Press.

Broadberry, S. (2010). *Agricultural productivity during the industrial revolution: Evidence from annual data*. In S. Broadberry & K. H. O'Rourke (Eds.), *The Cambridge Economic History of Modern Europe* (Vol. 1, pp. 179-208). Cambridge University Press.

Browne, C. (2019). *Artificial Intelligence in Games*. Springer International Publishing.

Brownlee, J. (2019a). *What is narrow, general, and super artificial intelligence*. Machine Learning Mastery.

Brownlee, J. (2019b). *A Gentle Introduction to Generative Adversarial Networks (GANs)*. Machine Learning Mastery. <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>

Cascadeur. (n.d.). *AI-based animation tool*. <https://cascadeur.com>

DeepArt. (n.d.). *Create your own AI-generated art*. <https://www.artvy.ai/ai-tools/deepartio>

DeepMotion. (n.d.). *Bringing Digital Humans to Life With AI*. <https://www.deepmotion.com/>

Donovan, T. (2010). *Replay: The history of video games*. Yellow Ant.

Dungeon Alchemist. (n.d.). *AI-powered dungeon generator*. <https://www.dungeonalchemist.com>

Khatsenkova, S. (2023, March 25). *Audio deepfake scams: Criminals are using AI to sound like family and people are falling for it*. <https://www.euronews.com/next/2023/03/25/audio-deepfake-scams-criminals-are-using-ai-to-sound-like-family-and-people-are-falling-fo>

European Commission. (2021). *Ethics guidelines for trustworthy AI*. <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>

Gaur, L., Singh, A., & Khari, M. (2021). *Autonomous Vehicles: Frameworks, Architectures, and AI-based Approaches*. Springer.

GitHub Copilot. (n.d.). <https://github.com/features/copilot>

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). *Generative Adversarial Networks*. <https://arxiv.org/abs/1406.2661>

Houdini. (n.d.). *Procedural generation software*. <https://www.sidefx.com/products/houdini>

Jackson, P. (1998). *Introduction to Expert Systems (3rd ed.)*. Addison Wesley.

Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing (3rd ed.)*. Prentice Hall.

Kent, S. L. (2001). *The ultimate history of video games: From Pong to Pokémon and beyond—the story behind the craze that touched our lives and changed the world*. Three Rivers Press.

Koskinen, I., Alasuutari, P., & Peltonen, T. (2005). *Laadulliset menetelmät kauppatieteissä*. Vastapaino.

Kushner, D. (2003). *Masters of Doom: How two guys created an empire and transformed pop culture*. Random House.

Lähdesmäki, T., Hurme, P., Koskimaa, R., Mikkola, L., & Himberg, T. (2010). *Mapping Research Methods: Methods Paths for Humanists*. University of Jyväskylä, Faculty of Humanities. <http://www.jyu.fi/mehu>

Logrus IT. (2023, February 1). *Predictions for the Video Game Industry in 2023*. https://medium.com/@logrusit_40171/predictions-for-the-video-game-industry-in-2023-7961580769e7

Manyika, J., Lund, S., Chui, M., Bughin, J., Woetzel, L., Batra, P., Ko, R., & Sanghvi, S. (2017). *Jobs lost, jobs gained: Workforce transitions in a time of automation*. McKinsey Global Institute. <https://www.mckinsey.com/featured-insights/future-of-work/jobs-lost-jobs-gained-what-the-future-of-work-will-mean-for-jobs-skills-and-wages>

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Mixamo. (n.d.). *3D character animation*. <https://www.mixamo.com>

Modern Language Association (MLA). (2021). *MLA Handbook (9th ed.)*. Modern Language Association of America.

NVIDIA. (n.d.). *NVIDIA GauGAN: AI-powered sketch-to-image technology*. NVIDIA

Onigiri. (2023, May 1). *ИИ играет в мафию [AI plays Mafia]* [Video]. YouTube. <https://www.youtube.com/watch?v=jAHsLJLGfyM>

OpenAI. (n.d.). <https://www.openai.com>

OpenAI. (2017a, August 11). *Dota 2*. <https://openai.com/index/dota-2/>

OpenAI. (2017b, August 16). *More on Dota 2*. <https://openai.com/index/more-on-dota-2/>

OpenAI. (2019, September 17). *Emergent tool use from multi-agent interaction*. <https://openai.com/index/emergent-tool-use/>

Paananen, A. (2023). *ChatGPT and Productivity During Unity Game Development* [Thesis]. <https://urn.fi/URN:NBN:fi:amk-2023052915693>

PlaytestCloud. (n.d.). *PlaytestCloud game testing*. <https://www.playtestcloud.com>

Promethean AI. (n.d.). *AI-assisted game content creation*. <https://www.prometheanai.com>

Rabin, S. (Ed.). (2017). *Game AI pro 3: Collected wisdom of game AI professionals*. A K Peters/CRC Press. <https://www.taylorfrancis.com/books/edit/10.4324/9781315151700/game-ai-pro-3-steve-rabin>

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders. https://books.google.ru/books/about/Andrew_Rollings_and_Ernest_Adams_on_Game.html?id=Qc19ChiOUI4C&redir_esc=y

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach (4th ed.)*. Pearson. <https://aima.cs.berkeley.edu/>

Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer Handbook of Robotics (2nd ed.)*. Springer. <https://link.springer.com/book/10.1007/978-3-319-32552-1>

Sonniss. (n.d.). *AI sound effect generation*. <https://www.sonniss.com>

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications (2nd ed.)*. Springer. <https://szeliski.org/Book/>

Tabnine Gemini. (n.d.). <https://gemini.google.com>

Tegmark, M. (2017). *Life 3.0: Being human in the age of artificial intelligence*. Knopf. [https://aitskadapa.ac.in/e-books/CSE/AI/Life%203.0_%20Being%20Human%20in%20the%20Age%20of%20Artificial%20Intelligence%20\(%20PDFDrive%20\).pdf](https://aitskadapa.ac.in/e-books/CSE/AI/Life%203.0_%20Being%20Human%20in%20the%20Age%20of%20Artificial%20Intelligence%20(%20PDFDrive%20).pdf)

Terävä, T. (2017). *Workflows for Creating 3D Game Characters*. <https://www.theseus.fi/handle/10024/131241>

Testronic. (n.d.). *Automated game testing*. <https://www.testroniclabs.com>

Wolf, M. J. P. (2008). *The video game explosion: A history from PONG to Playstation and beyond*. Greenwood Press. <https://archive.org/details/videogameexplosi0000unse>

World Economic Forum. (2020a). *The Future of Jobs Report 2020*. <https://www.weforum.org/reports/the-future-of-jobs-report-2020>

World Economic Forum. (2020b). *The Future of Jobs Report 2020, Infographics* <https://www.weforum.org/publications/the-future-of-jobs-report-2020/in-full/infographics-e4e69e4de7/>

Yannakakis, G. N., & Togelius, J. (2018). *Artificial Intelligence and Games*. Springer International Publishing. <https://link.springer.com/book/10.1007/978-3-319-63519-4>

Appendices

Appendix 1. Fishing Game with Gemini

User:

I want to create a fishing feature for my game, that is like the fishing minigame in Stardew Valley. I am using C# and Unity.

Gemini:

Building a Stardew Valley-like Fishing Minigame in Unity with C#

Here's a breakdown of the core mechanics you can implement in C# and Unity to create a fishing minigame similar to Stardew Valley:

1. Casting and Fish Catching:
 - a. Casting: Use a UI element (like a button) to trigger casting.
 - b. Simulate casting with an animation or by moving a bobber sprite to the desired location on the water (implement a 2D plane for the water surface).
 - c. Fish Bite: Randomly trigger a fish bite after a set timer or based on specific conditions (like entering a certain area).
 - d. Indicate a bite with a bobber animation or sound effect.
2. Fishing Minigame:
 - a. Green Bar: Create a UI element (a green rectangle) representing the fishing line's range.
 - b. Fish Icon: Introduce a small fish icon that moves up and down based on the fish's movement.
 - c. Catching Mechanic: The player controls the green bar by pressing a designated button (e.g., space bar). Holding the button raises the bar, and releasing it lowers it.
 - d. The goal is to keep the fish icon within the green bar as a progress bar on the right fills up.
 - e. The fish will escape if the icon stays outside the bar for too long (progress bar depletes).
3. Difficulty and Progression:
 - a. Fish Difficulty: Assign different difficulty levels to various fish species. Harder fish have faster and more erratic movements.
 - b. Progression: Increase the green bar's size and decrease fish escape speed as the player levels up their fishing skill. This can be tied to experience gained from catching fish.
4. Additional Features:
 - a. Perfect Catches: Reward players for keeping the fish icon within the green bar throughout the minigame. This could increase quality or experience gain.
 - b. Bait and Tackle: Introduce craftable or purchasable bait/tackle that affects fish behavior (attract rarer fish, slow fish movement).

Implementation Tips:

- Use a coroutine or timer to manage the fish movement and progress bar during the minigame.
- Consider using animation curves to create natural and varied fish movement patterns.
- Utilize sound effects for casting, bites, and catching success/failure.
- Balance the difficulty curve to be challenging but rewarding.

Learning Resources:

- Unity provides good documentation and tutorials for working with UI elements, animations, and coroutines: <https://docs.unity3d.com/Manual/index.html>
- You can find many online resources and communities dedicated to Unity development, which can offer valuable help and inspiration.

This is a basic structure for your fishing minigame. You can customize it further by adding unique fish types, power-ups, and different environments to make it more engaging in your game!

User:

Can you give me code for that?

Gemini:

Providing full code for the entire minigame would be quite extensive. However, here's a C# script example to get you started with the core mechanics:

Code snippet

```
public class FishingMinigame : MonoBehaviour
{
    // Inspector variables
    public Transform bobber; // Transform of the bobber game
    object
    public Transform fishIcon; // Transform of the fish icon game
    object
    public UI.Slider progressBar; // UI Slider for the progress
    bar
    public float fishEscapeTime = 5f; // Time for fish to escape
    (seconds)

    // Internal variables
    private bool isFishing = false;
```

```

private float escapeTimer = 0f;
private float fishSpeed = 1f; // Adjust for difficulty

void Update()
{
    if (isFishing)
    {
        // Move fish icon up and down
        fishIcon.position += new Vector3(0f,
Mathf.Sin(Time.time * fishSpeed) * 0.5f, 0f);

        // Check if fish escapes
        escapeTimer += Time.deltaTime;
        if (escapeTimer > fishEscapeTime)
        {
            // Fish escapes (implement logic for failure)
            isFishing = false;
            escapeTimer = 0f;
        }

        // Check for player input (replace with your button
press logic)
        if (Input.GetKey(KeyCode.Space))
        {
            // Raise the green bar (adjust based on UI element
properties)
            progressBar.value += Time.deltaTime;
        }
        else
        {
            // Lower the green bar
            progressBar.value -= Time.deltaTime;
        }

        // Check if fish is caught (logic based on fish icon
position and bar value)
        if (/* Check if fish icon is within green bar and
progress bar is full*/)
        {
            // Fish caught (implement logic for success)
            isFishing = false;
            escapeTimer = 0f;
        }
    }
}

public void StartFishing()
{
    isFishing = true;
}

```



```

    escapeTimer = 0f;
    progressBar.value = 0f; // Reset progress bar
    // Reset fish icon position (optional)
}
}

```

Explanation:

- This script manages the fishing minigame logic.
- StartFishing function initiates the minigame.
- The Update function constantly checks for fish movement, escape timer, and player input.
- Player input (using Space key here) controls the progress bar.
- Fish movement is simulated with a sine wave pattern.
- The code includes comments for implementing success/failure logic and green bar control based on your UI setup.

Remember:

- This is a basic example. You'll need to integrate it with your UI elements and game logic.
- Add animations, sound effects, and visual cues for a more engaging experience.

User:

In our code base the inspector variable are private and marked as serialized fields. Please update the code for that.

Gemini:

Absolutely, here's the updated code using [SerializeField] for private inspector variables:

Code snippet

```

public class FishingMinigame : MonoBehaviour
{
    // Inspector variables with [SerializeField]
    [SerializeField] private Transform bobber; // Transform of the
bobber game object
    [SerializeField] private Transform fishIcon; // Transform of
the fish icon game object
    [SerializeField] private UI.Slider progressBar; // UI Slider
for the progress bar

```

```

[SerializeField] private float fishEscapeTime = 5f; // Time
for fish to escape (seconds)

// Internal variables
private bool isFishing = false;
private float escapeTimer = 0f;
private float fishSpeed = 1f; // Adjust for difficulty

void Update()
{
    if (isFishing)
    {
        // Move fish icon up and down
        fishIcon.position += new Vector3(0f,
Mathf.Sin(Time.time * fishSpeed) * 0.5f, 0f);

        // Check if fish escapes
        escapeTimer += Time.deltaTime;
        if (escapeTimer > fishEscapeTime)
        {
            // Fish escapes (implement logic for failure)
            isFishing = false;
            escapeTimer = 0f;
        }

        // Check for player input (replace with your button
press logic)
        if (Input.GetKey(KeyCode.Space))
        {
            // Raise the green bar (adjust based on UI element
properties)
            progressBar.value += Time.deltaTime;
        }
        else
        {
            // Lower the green bar
            progressBar.value -= Time.deltaTime;
        }

        // Check if fish is caught (logic based on fish icon
position and bar value)
        if (/* Check if fish icon is within green bar and
progress bar is full*/)
        {
            // Fish caught (implement logic for success)
            isFishing = false;
            escapeTimer = 0f;
        }
    }
}

```

```

    }

    public void StartFishing()
    {
        isFishing = true;
        escapeTimer = 0f;
        progressBar.value = 0f; // Reset progress bar
        // Reset fish icon position (optional)
    }
}

```

Changes made:

- Added [SerializeField] before each private variable declaration.
- This allows these private variables to be shown and edited within the Unity Inspector window.

Note:

- Ensure you have the using UnityEngine.UI; directive at the top of your script to use the UI.Slider class.

Appendix 2. Plot by ChatGPT

User:

I'm developing a game in an alternative reality where nazis won the second war. The story takes place in Saint Petersburg and describes the fate of people there. What were the plans of the nazis regarding Saint-Petersburg and the soviet people? Who lived there and what story might be interesting for players?

ChatGPT:

In an alternative reality where the Nazis won World War II, their plans for Saint Petersburg (then Leningrad) and the Soviet people would likely have been brutal and oppressive. The Nazis had a clear agenda of subjugating and exterminating populations they deemed inferior, particularly Slavic peoples, whom they considered "subhuman" according to their racist ideology.

Here are some potential elements of Nazi plans for Saint Petersburg and the Soviet people:

Forced Labor and Exploitation: The Nazis would likely have implemented policies of forced labor, conscripting able-bodied individuals for work in factories, construction projects, and other labor-

intensive tasks. This could form the backbone of the city's economy under Nazi rule, with harsh penalties for those who resist or fail to meet production quotas.

Ethnic Cleansing and Population Control: The Nazis had a policy of ethnic cleansing, aiming to rid conquered territories of undesirable populations. This could have led to mass deportations, executions, and imprisonment of Soviet citizens deemed undesirable by the Nazi regime. The city's demographics would have shifted dramatically under such policies.

Cultural Suppression and Indoctrination: The Nazis would likely have imposed strict censorship and control over education, media, and cultural institutions in Saint Petersburg. Russian language and culture might have been suppressed in favor of Germanization efforts, with propaganda promoting Nazi ideology and glorifying the regime.

Resistance and Underground Movements: Despite the oppressive regime, there would likely have been pockets of resistance and underground movements within Saint Petersburg. Players could potentially take on the role of resistance fighters, working to sabotage Nazi operations, smuggle information or people out of the city, and rally the oppressed population against their occupiers.

Humanitarian Crisis and Survival Struggles: Under Nazi rule, Saint Petersburg would likely have faced severe shortages of food, medicine, and other essential resources due to Nazi exploitation and the blockade of the city. Players could experience the desperation and hardships faced by the city's inhabitants as they struggle to survive amidst the chaos of war and occupation.

Collaboration and Betrayal: Not everyone would resist the Nazis; some individuals might choose to collaborate with the occupiers for personal gain or out of fear.

Players could encounter characters who are torn between loyalty to their people and the temptation of collaboration, adding moral complexity to their decisions and interactions within the game.

By exploring these themes and scenarios, you could create a compelling narrative that immerses players in the alternate history of Saint Petersburg under Nazi rule, highlighting the resilience of the human spirit in the face of oppression and the power of hope and resistance.

User:

Regarding this setting and the fact, that the main character in this game is just a human, what story can the game tell to the player? What hard decisions can players make in the game, and what morals can the story teach the player? Please give me 2 alternative storylines. One, where

the protagonist is a soviet citizen, and one with a nazi protagonist.

ChatGPT:

Sure, here are two alternative storylines for the game:

Storyline 1: Soviet Citizen Protagonist

The protagonist is a young Soviet citizen living in Saint Petersburg under Nazi occupation. Their family has been torn apart by the war, with some members taken away for forced labor or imprisoned by the Nazis. As the protagonist navigates the city's harsh realities, they become involved in the resistance movement, risking their lives to sabotage Nazi operations and help fellow citizens escape persecution. Along the way, they face difficult moral choices, such as whether to prioritize the safety of their loved ones or the greater good of the resistance. Eventually, they must confront the ultimate decision: whether to sacrifice themselves for the cause or try to escape the city and seek refuge elsewhere. This storyline explores themes of sacrifice, resilience, and the power of ordinary people to make a difference in the face of oppression.

Storyline 2: Nazi Protagonist

The protagonist is a young Nazi officer stationed in Saint Petersburg, tasked with enforcing the regime's oppressive policies. Initially loyal to the Nazi cause, the protagonist begins to question their beliefs and the morality of their actions as they witness the suffering and brutality inflicted on the city's inhabitants. As they interact with Soviet citizens and learn about their experiences, the protagonist grapples with internal conflicts and moral dilemmas. They must decide whether to continue blindly following orders or defy the regime and risk their own life to help those they were once sworn to oppress. Ultimately, the protagonist must choose between loyalty to their country and humanity, facing the consequences of their decisions in a world where even the smallest act of resistance can carry immense weight. This storyline explores themes of moral ambiguity, redemption, and the capacity for change, challenging players to confront the complexities of human nature and the consequences of their choices.