Piniverse: A Versatile AI-Powered Framework for Streamlining Digital Content Interaction and Empowering User Creativity

A Major Qualifying Project Proposal Outline
submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
degree of Bachelor of Science and Bachelor of Arts

by
Alexander Kwan
Gabriel Buonomano
Shiming De
Yihong Xu[1]
Yongcheng Liu

Date:
April 27th, 2023

Advised by:
Professors Gillian Smith and Tian Guo

[1] Yihong Xu is a significant contributor but is not officially submitting this proposal for his degrees' requirements.

Authorship

Section Number/Name	Primary Author	Primary Project Contributor	Reviewer / Editor
Abstract	Yongcheng	N/A	Alex
Acknowledgements	Alex	N/A	Gabriel
1. Introduction	Yongcheng	N/A	Alex
2. Background	Yongcheng	N/A	Alex
3.1 Implementation (Professional)	Yongcheng	Yongcheng	Gabriel
3.2 Implementation (Casual)	Gabriel, Shiming	Gabriel, Alex, Shiming	Alex
4.1 User Study (Professional)	Yongcheng	Yongcheng	Gabriel
4.2 User Study (Casual)	Alex, Gabriel	Gabriel, Alex, Shiming	Shiming
5.1 Evaluation (Professional)	Yongcheng	N/A	Alex
5.2 Evaluation (Casual)	Alex, Gabriel	N/A	Shiming
6. Conclusion & Future Work	Gabriel	N/A	Alex
References	All	N/A	Gabriel

Abstract

The increasing importance of digital content creation and design in modern communication, learning, and interaction calls for more accessible and versatile tools. Although artificial intelligence (AI) has expanded possibilities in this domain, high entry barriers, technical expertise requirements, and reusability difficulties still exist. Piniverse is a novel framework designed to address these challenges and empower users to create and interact with digital content more easily through system-level support.

The Piniverse framework consists of presentation, design, semantics, and machine abstraction layers, making it adaptable for various types of users. Two proof-of-concept implementations, one for casual and one for professional users, demonstrate the framework's feasibility, usability, and effectiveness in diverse interactive applications. The project also presents a roadmap for the framework's future evolution, positioning it as a steppingstone towards the next generation of digital experiences. By targeting the broad applicability of digital content and focusing on user empowerment, Piniverse lays the foundation for a more accessible and innovative future of digital media experiences.

Acknowledgements

We would like to put forth our gratitude to our advisors at the Worcester Polytechnic Institute, Professor Gillian Smith and Professor Tian Guo, for providing essential input and ideas that were instrumental in the completion of this project.

Table of Contents

List of Figures

List of Tables

1. Introduction

The rapid growth of digital technologies has transformed the way we communicate, learn, and entertain in the modern world. They have also led to an increased demand for intuitive creation, editing, and management of digital contents, which spans a wide range of formats, from images and videos to complex interactive experiences like games and simulations. However, the complexity of specialized tools and the technical expertise required to use them often pose significant barriers for aspiring individuals who wish to express their ideas digitally.

The concept of artificial intelligence (AI) has long been seen as a potential solution to bridge this gap. Since John McCarthy first introduced the term [1] in 1955, AI has captivated the imagination of those seeking to utilize its power for automatic and smart data processing, as well as digital communications. Recently, there had been a booming availability in AI Generated Content (AIGC) solutions. Thanks to the latest advancements, we are now closer to achieving the dream of fluid interaction between humans and machines. Nowadays, people can describe their ideas in natural language to create digital images [2], [3], 3D models [4], [5], and even interactive experiences like games [6], [7] in a matter of minutes.

Inspired by personal experiences witnessing the people struggling with learning specialized technologies, watching the booming availability of AI solutions, and recognizing the potential of AI in enabling users to create and interact with digital content in more intuitive ways, such as through chatting with AI and receiving real-time visual feedback, we initiated Project: Piniverse at Worcester Polytechnic Institute (WPI) as a Major Qualifying Graduation Project (MQP). Our goal is to integrate these novel solutions under a unified framework, allowing users to access and utilize them out of the box regardless of their technical expertise. Moreover, we aim to develop a versatile AI-powered framework at the system level, ensuring easy customization and seamless integration with future technologies by providing a clear structure and guidelines for future technology integrations.

The Piniverse framework comprises four interconnected layers: presentation, design, semantics, and machine abstraction. These layers work synergistically to streamline the digital content creation and interaction process, making it adaptable for various user types. The presentation layer handles the display and interaction of content, ensuring a seamless experience across various devices. The design layer compartmentalizes various concerns during the design process, enabling users to focus on specific aspects without needing to comprehend the entire system, thus simplifying the design experience. The semantics layer stores and delivers digital content in both semantic and digital formats, facilitating more efficient content usage and management. Lastly, the machine layer addresses technical details, allowing users to concentrate on content creation without being burdened by the underlying technology. Upon successful implementation, this approach will empower any user to use, share, or customize any digital content via shared solutions under the same framework, effectively lowering the barriers to entry and fostering a more inclusive and collaborative digital content ecosystem.

Through multiple proof-of-concept implementations, the Piniverse framework has demonstrated its feasibility, usability, and effectiveness in diverse interactive applications. By focusing on user empowerment and the broad applicability of digital content, Piniverse lays the foundation for a more accessible and innovative future of digital media experiences.

The following chapters of this report detail the Project Piniverse, its development, and its findings. Chapter 2 provides the problem statement and an overview of the current approaches and technologies employed in intelligent digital content creation. Chapter 3 illustrates the project timeline and delves into the origins and motivations behind the project. Chapter 4 covers the minimal implementation achievable within our time constraints and system implementations. Chapter 5 presents the user testing methods and procedures. Chapter 6 evaluates the data collected from the user study. Chapter 7 analyzes the feasibility, usability, and effectiveness of the two implementations and the proposed framework. Finally, Chapter 8 concludes the findings and outlines detailed plans for the next steps to further develop the Piniverse framework and revolutionize digital content experiences for users, from casual content creators to professional developers and system administrators.

2. Background

Under the drive of advancement in technology and demand of artificial intelligence from various industries, the landscape of intelligent digital content systems is evolving rapidly. In this chapter, we identified the common challenges in content creation, as well as the state-of-the-art approaches to support intuitive digital content interactions. This chapter aims to provide an overview of the current approaches and technologies employed in intelligent digital content systems, highlighting their benefits and drawbacks, while emphasizing the need for a unified, systematic framework to address the complexities of digital content creation.

2.1 Problem Statement

The initial inspiration for project Piniverse emerged from numerous discussions, experiences, and projects within the Worcester Polytechnic Institute's Interactive Media and Game Development Department (WPI IMGD). Our team observed several common challenges faced by students during game development. Interestingly, a quick review of the literature, like [8]–[10] produced coherent, relevant, yet surprising findings, indicating that those challenges are prevalent in professional content creation as well. Some challenges were specific to content design [11], while others were exclusive to content implementation [12], [13]. We have summarized the identified common challenges that we try to solve in Table 2.1.

No.	Challenge	Description
1	Production Management	Both students and professionals often face challenges managing projects alongside other responsibilities (like other classes). Inexperience in production management and inadequate planning, tracking, and monitoring can complicate the process, leading to project delays and waste of work or resources. Scope creep is common, which could lead to loss of focus on original goals or unfulfillment of the original concept, even lead to project failures.

2	Team Dynamics and Understanding of Individual Responsibilities and Authority	Effective communication and collaboration are crucial for successful content production. Miscommunication and low engagement often lead to knowledge inconsistencies, duplicate work, and conflicts within the team. Members need to understand their responsibilities and authority to move forward with the team.
3	Clear and Coherent Vision	Having a well-defined vision and strategy is essential for guiding the development process. However, this is often missing in teams, leading to disagreements on quality expectations, inconsistencies between design and development, and even team conflicts.
4	Design Agreement	Lack of unified or agreed usage of design languages can lead to misunderstandings, inconsistencies, and even conflicts in the project goals and objectives, ultimately resulting in a loss of competitiveness or even project failures.
5	Technical Difficulties	Content creation involves various tools, software, and programming languages. Team members often come from different backgrounds, resulting in inconsistencies in technical knowledge and skills. Unclear metrics and the need to keep up with the rapidly changing industry can create difficulties in coordinating and integrating work, leading to project failures or huge overheads.
6	Balancing creativity and feasibility	Creators must strike a balance between their innovative ideas and practical limitations, such as hardware capabilities, software restrictions, and target audience preferences. Ignorance of these factors can lead to scope creep, unrealistic expectations, and project difficulties.
7	Templates and Asset Management	A lack of rudimentary templates and procedures to manage assets can lead to repetitive work, incompatibilities, and difficulties in integrations. A lack of effective asset management tool can lead to overlapped or loss of work or chaotic collaborations. Tremendous of time may be needed to setup bare minimal skeletons of the project repeatedly without templates.

8	User Experience & Audience Engagement	Understanding and addressing user needs, accessibility, inclusivity, and user community engagement are essential for content success. A lack of systematic approaches can lead to content that users won't buy or use, resulting in significant monetary or effort loss after substantial investment.

Table 2.1 Common Challenges in Content Creation Processes

2.2 Emerging Solutions

In this section, we explore the novel opportunities presented by emerging trends and solutions in addressing the common challenges inherent in the content creation process, as identified in the problem statement (Section 2.1). We have selected representative innovations, including AIGC, Web Assembly, Multi-Agent Systems, and Beginner-friendly Design Interfaces, for in-depth discussion. By examining their contributions, potential, and novelty, we aim to demonstrate their relevance and connection to Project Piniverse, highlighting their transformative impact on the digital content ecosystem.

2.2.1 AI Generated Content

AI Generated Content (AIGC) refers to the production of digital content with AI (Artificial Intelligence) systems or algorithms. With the breakthroughs in recent AIGC technologies, there have been a substantial number of works, such as [2]–[7], [14]–[17], showcasing the potential of using AIGC to automate various aspects of content creation. We believe that this potential for content creation automation will increase accessibility and reduce the need for specialized knowledge or training. Furthermore, [18], [19] have demonstrated its versatility and applicability to a wide range of scenarios.

From these examples, we can see that AIGC has the capability or potential to provide out-of-the-box content creation solutions for any user, as it is commonly easy to start using them. Users can use natural language to convey their designs, and the various AIGC solutions can provide content in appropriate formats with good quality in a relatively short amount of time [18], [19].

Therefore, they can help resolve many challenges in Table 2.1, such as reducing technical difficulties by automating specialized tasks or facilitating communication by providing visual concept art after natural language input. We believe that automating the technically advanced aspects of content creation will allow individuals to focus on their strengths and the creative aspects of their projects. We imagine that this type of automation would lead to more efficient workflows and improved learning outcomes.

In our project, we extensively used AIGC technologies to demonstrate the practicality of streamlining the content creation process with AIGC workers. Specifically, we employed AI agents like 3D model generators, image generators, coders, summarizers, and more. Those workers allowed users to get desired digital content within minutes with only natural language input, effectively lowering the barriers to entry for creating digital content and fostering a more inclusive digital content ecosystem. Additionally, by abstracting each agent as an individual callable API in the system, we also demonstrated the feasibility of building a scalable framework to accommodate diverse AIGC solutions at scale to meet different needs. We believe that the integration of AIGC into the Piniverse framework showcases the transformative impact of AI-generated content on the digital content creation process and its potential to empower users of all skill levels, enabling more inclusive and collaborative content creation experiences across various domains.

2.2.2 Web Assembly

First introduced in 2015 [20], WebAssembly (WASM) is a low-level virtual machine that aimed to support code execution in a universal runtime, primarily web browsers. Ideally, it can run any programming language binaries on any platform [21]. As WebAssembly is in a binary format, it offers a potential for integrating works from individuals with diverse technical backgrounds by merging their work in the form of WASM. Examples from multiple industries, such as game distribution [22], [23], IoT execution [24], [25], high-resolution 3D rendering [26], and many more, have shown its capabilities for providing unified virtual experiences using different technology stacks.

By providing a universal runtime as a platform for seamless integration, collaboration, and testing, WebAssembly enables team members to work together, regardless of their individual technical backgrounds. This helps address challenges in Table 2.1, such as production management, team dynamics, and technical difficulties. For instance, WebAssembly can facilitate efficient workflows by enabling the integration of code written in different programming languages, reducing the need for specialized technical expertise and simplifying collaboration among team members with diverse backgrounds.

In our project, we used WebAssembly to present a generated game in the professional demo using the Pyxel engine [27]. The game was generated based on user input and showcased the potential of providing seamless integration without worrying about compatibility issues between various platforms and programming languages. This implementation demonstrates the advantages of WebAssembly in streamlining the content creation and interaction process, ultimately making it more accessible for users with diverse skill sets and backgrounds.

2.2.3 Multi Agent Systems

Multi-agent system (MAS) is a computational paradigm that involves the interaction of multiple autonomous agents to achieve specific goals. This approach has demonstrated excellence in decomposing and executing tasks in several industries, such as robotics engineering [20].

Recently, MAS has also shown potential in the content creation domain, offering more abstracted and user-friendly interfaces for usage without the need to worry about the underlying automation technologies. A good example is [21], which allows creators to use formal semantic predicate logic, similar to natural languages, to develop arbitrary 2D games. This innovative approach showcased the effectiveness of using MAS to automate and abstract content creation processes. Additionally, recent experiments with GPT-4 and MAS, like [22], [23], have demonstrated the potential to build intelligent automation pipelines that can handle complex tasks, such as multi-step project management or automated travelling scheduling with bookings , further highlighting the value of MAS in streamlining content creation processes.

In our project, we explored the potential of integrating multi-agent systems into the framework to automate and streamline the processes while abstracting the detailed execution, simplifying the system to users. Specifically, we allowed users to develop any general intelligent assistant agent with domain knowledge, policies, and persona settings. With a chain of such workers, users can create more efficient and intelligent workflows to tackle various content creation challenges. We also provided a retro game development assistant demo that chains several potential game workers, such as receptionist, concept document worker, game designer worker, producer, and coder, to streamline the creation of retro games. This approach demonstrates the versatility and adaptability of multi-agent systems in addressing the common challenges inherent in the content creation process, as identified in Table 2.1.

2.2.4 Game Portability and Intuitive Design Interfaces

[24] emphasized that the idea of design independence from game engines and technologies can support greater compatibility and flexibility for creative designers or creators. An intuitive design interface is critical for creators to effectively translate their ideas to digital content.

One intuitive design interface can be chatting, or natural language-based interactions, which have demonstrated their effectiveness in [6], [18], [19], [21]. Another emerging solution is Low-Code or No-Code platforms, which allow users with no prior coding experience to develop and build complex and engaging applications with real-time visual feedback [25]–[27].

In our project, we adopted those approaches to provide more intuitive design interfaces. We allowed users to use natural language as the primary means of interaction with the system and receive visual feedback. Additionally, we utilized the open-source low-code platform Appsmith [28] in conjunction with our system to build a real-time dynamic survey in just one day. This demonstrated the efficiency and effectiveness of combining natural language-based interactions and low-code solutions in streamlining the content creation process, making it more accessible and user-friendly for individuals with diverse backgrounds and skill levels. By integrating these intuitive design interfaces into the Piniverse framework, we aim to empower users to create and interact with digital content seamlessly, fostering a more inclusive and collaborative digital content ecosystem.

2.3 Project Goals

The goal of this project is to propose a versatile framework that integrates AI and human workers into a content creation pipeline. The framework should be able to support a wide range of content types and formats while allowing for rapid prototyping and visualization. To achieve this goal, we developed a user-friendly content creation engine that facilitates collaboration among content creators and consumers.

One important aspect of this project is the establishment of a collaborative platform for open-source sharing and exploring of content. This fosters a community of content creators and consumers and saves time by enabling content to be reused and repurposed. Furthermore, the framework ensures compatibility with existing development tools and platforms, providing seamless integration with popular technologies. By discussing the pros and cons and future of the pipeline, we can identify what is needed to achieve our goal and the benefits that will be realized. Finally, we implemented two proof-of-concepts to assess the feasibility and effectiveness of the proposed framework.

2.4 Backbone Technologies

Section 2.3 delves into the core technologies that are used or referenced in the project and introduces how they are used both in the content creation industry and in our project.

2.4.1 Autonomous Intelligence

Autonomous intelligence refers to computational systems that can perform tasks and make intelligent decisions like a human with minimal human intervention. Such a system can support the processing and handling of user requests with good quality at any time, as shown by MAS and [22], [23]. Two primary approaches can be used to build such systems: embedding intelligence directly into digital content using a universal language (represented by Semantic Web in Section 2.3.1.1), or training human-like intelligent agents with domain-specific knowledge to generate context-aware solutions (represented by Machine Learning or Artificial Intelligence in 2.3.1.2).

2.4.1.1 Intelligence Through Smart Content: The Semantic Web

The Semantic Web, often considered to be officially started at 2001 with [28], proposed to offer more structured, understandable, and interconnected protocols for digital contents to allow machines to understand the essence of the data and process them intelligently and automatically. Its primary goal is to establish a common framework for sharing and automatic understanding of the data across various applications, platforms, and communities with the help of a unified protocol [28]. Key components of the Semantic Web include the use of ontologies, metadata, and semantic annotations to describe the meaning and relationships between different pieces of information. Although semantic technologies are widely used today to assist effective and efficient data processing with intelligence, like by Google or Wikipedia [29], the formal standard [30] that is intended to shared and used among all machines faces challenges such as poor usability and the need for authoritative organizations to publish agreeable ontologies [29], [31], [32].

```
t:roomBedroom rdfs:subClassOf t:room ;
    t:mandatory t:bed , t:cupboard ;
    t:optional
        t:picture ,
        t:cupboard ,
        t:mirror ,
        t:shelf ,
        t:table ,
        t:chestWood ,
        t:chestRed .
```

Figure 2.1 Using RDF & Ontologies to Generate RPG Game World [33]

Despite these limitations, the Semantic Web has shown potential in specific applications, such as game development. For example, in Figure 4.1, a room can be generated from just a few lines of intuitive descriptions using ontology technologies and Semantic Web languages (e.g., RDF). This approach demonstrates the reusability and generalization of the solution, allowing game developers to create various types of game worlds with different characteristics, ranging from text-based, 2D, to 3D, using the same defined ontology.

In our project, we initially attempted to utilize Semantic Web technologies to streamline the process and make the system less dependent on large language model processing by asking intelligent agents to compose ontologies in Semantic Web format. However, the results were only usable after several rounds of tuning, and the applicability was limited due to the need for specialized parsers.

Therefore, in the final version of the project, instead of directly using Semantic Web technologies, we drew inspiration from the Semantic Web to build the semantic layer in the professional version. We also used large language models to simulate some of the processes involved, which empowered the development of effective generalized intelligent agents.

2.4.1.2 Intelligence Through Smart Workers: Artificial Intelligence

Artificial intelligence (AI) is being used in the gaming industry to increase productivity and create more realistic gameplay. AI can create intelligent opponents that compete with the player in a "brains vs. brains" scenario by producing a level of uncertainty and using searching methods such as A* and D* [34]. Genetic algorithms are also used to enhance the realism of games, as seen in the game Creatures, which uses digital DNA representations to evolve Norns into more intelligent creatures [34]. In "Disappear," a finite state machine is used to determine the actions of enemies in specific scenarios [15]. Deep learning techniques are also used to create infinite content for game levels that can be modified with parameters to adjust difficulty [16]

In addition, AI is being used to generate 3D animations from natural language stories [14]. This technology could be useful in various industries, as it allows for the creation of 3D animations in a passive process compared to reading, which is more active.

2.4.2 Framework Architectures Reference

A versatile system framework is necessary to provide a foundation for integrating the emerging solutions together. Such a system should allow users of the framework to seamlessly access, utilize, and combine various tools and technologies in a unified and coherent manner. Additionally, the generalized access patterns further support the interoperability, generalizability, and, therefore, the reusability of any digital solutions. A versatile digital content system can also resolve the scalability, adaptability, and reproducibility issues of the autonomous intelligence solutions.

We have drawn insights from several candidates that are widely studied or used today, including conceptual virtual architecture, game engines, and content management systems. Such review helped the formation of the proposal and proof-of-concept implementations of the framework in this project.

2.4.2.1 Conceptual Virtual World Architectures

Although there is no generally accepted definition for a virtual world [35], it had been considered as representative frameworks for versatile digital content systems for a long time since last 1970s to 1990s [36]–[38]. Conceptual virtual world architectures are high-level, abstract frameworks that facilitate the design, development, and management of virtual environments. These architectures should theoretically allow the creation of any virtual experiences [35], [36].

Recently, [35] had tried to give a formal definition of a virtual world and required components. The discussion helped us to identify the future use cases and requirements of our framework and therefore allowed us to build and refine our the Piniverse framework in an adaptable and scalable way to accommodate various digital content operations.

2.4.2.2 Game Engines

Game engines are another universal solution for digital content operations. They are game development kits that facilitate the creation and development of video games [39]. Game engines provide a comprehensive set of tools and functionalities that enable content creators to design and develop with visual feedback, real-time rendering, and native supports for distributing to various platforms and devices.

Game engines offers an intuitive interface for designers to convert their ideas to visualized output [40]. There had been tremendous amount of studies over game engine, like [21], [39]–[42], and these works have provided invaluable insights for the development of our project. In both of our products, we learned and utilized rendering techniques and engine tricks to enable efficient and seamless integration of elements and objects.

Additionally, Unreal had been extensively used outside the video game industry for video making[43], [44], filmmaking[45], [46], or even physics simulations [47], [48], and more. The versatile usage of the existing game engines showed that a framework that is compatible with game engine architectures should theoretically support the majority of content types, empowering users to create and interact with different types of virtual experience by the end.

2.4.2.3 Content Management Systems

A content management system (CMS) is a professional solution to manage and interact with digital assets, typically in enterprise scenarios [49]. CMS allows content creators to develop reusable content modules that can be combined and repurposed in various contexts, maximizing efficiency and consistency across digital experiences. By offering a one-stop solution to help users create and manage digital content intuitively and effectively, CMS streamlines the process of organizing and templating digital assets.

With the modular templates and digital asset organization, content creators can streamline the development process and effortlessly adapt their content systems to accommodate new requirements and technologies. Moreover, these systems provide built-in support for various content types, such as text, images, video, and interactive elements, enabling the creation of diverse and engaging experiences for users [49]–[51].

In our project, we explored the potential of integrating CMSs and low-code platforms, such as Appsmith [52], to provide more intuitive and accessible design interfaces. CMS allowed us to better manage and interact with digital assets and APIs, empowering us to build a dynamic survey in one day. The CMS architectures also helped us manage the real-time generated assets with ease without worrying about or learning AWS S3 consoles, Google Firestore, and a local database.

By combining CMSs with our Piniverse framework, we aimed to further streamline the content creation and management process, making it more accessible and user-friendly for individuals with diverse backgrounds and skill levels. This approach demonstrates the value of adopting CMSs and low-code solutions in promoting more inclusive and collaborative content creation experiences across various domains.

2.4.3 Universal Human-Computer Interfaces

To develop a comprehensive and accessible framework for intelligent digital content systems, it is essential to consider the human-computer interfaces that bridge users and the system's underlying technologies. Universal human-computer interfaces facilitate versatility of the framework from system level at the presentation layer.

2.4.3.1 Computer to Human Output: Multimedia

Multimedia solutions empower systems to present digital content in various forms, such as text, images, audio, video, and even video games, without worrying about the source file formats. Studies [53]–[57] showed the need and potential to use semantic technologies to offer multimedia with human-understandable markup languages. Research [58], [59] demonstrates the potential of repurposing multimedia content for different platforms and user experiences, further enhancing accessibility and availability of virtual experiences.

Additionally, multimedia solutions enable machines and humans to collaborate seamlessly, as they facilitate the exchange of information in a format that is easily understood by both parties. In our proof-of-concept demos, we used existing multimedia libraries to create a versatile and immersive user experience. However, we were constrained by the libraries due to their limited support and the incapability to present content in the form of text, as seen in [60]–[62]. As a result, we could not use AIGC technologies to automatically build up the interface. Future developments addresses these limitations and enhance multimedia integration in the Piniverse framework.

2.4.3.2 Human to Computer Input: Multimodal

Multimodal interfaces enable users to interact with digital content systems using various input methods, such as text, voice, touch, and gestures. These interfaces allow machines to receive semantic context and make context-aware decisions, thereby improving the immersive and versatile user experience [63]–[66].

By offering multiple input modalities, these interfaces accommodate users with different preferences, abilities, and contexts, enhancing the accessibility and usability of the system. However, currently, the proof-of-concept modules did not fully integrate multimodal interfaces. To achieve a complete framework, future developments will focus on incorporating multimodal input options, further enhancing the system's versatility and user experience.

2.4.4 Machine Abstraction

Machine abstraction plays a crucial role in the development of versatile digital content systems, as it provides a foundation for compatibility and interoperability across different hardware and software platforms. In this section, we explore two key low-level machine abstraction techniques that empowered the project: virtualization for universal hardware abstraction and Anything as a Service (XAAS) for universal software abstraction.

2.4.4.1 Universal Hardware Abstraction: Virtualization

Virtualization technologies enable the creation of virtual versions of physical resources, such as servers, storage, and networks. This process allows for efficient utilization of hardware resources, flexibility, and scalability in content creation and management [67]–[69]. Containerization and orchestration tools further enhance hardware abstraction by packaging software applications and their dependencies into isolated containers and automating the deployment, scaling, and management of these containers [70]–[73].

In our project, we had extensively used virtualization technologies to overcome hardware limitations and compatibility issues with the use of [74]–[77] to ensure a smooth and efficient content creation process. This was critical due to the drastic need for real-time intensive computations, which is required by the state-of-the-art AIGC technologies. We are going to further automate the deployment and usage of virtualization technologies with [78]–[80] to allow intelligent agents to self-deploy the needed resources and additional workers.

2.4.4.2 Universal Software Abstraction: Anything As a Service (XAAS)

The concept of Anything as a Service (XAAS) represents the universal software abstraction, where various software applications, tools, and services are offered on-demand via the internet. This approach eliminates the need for installing and maintaining software locally, simplifying the overall content creation process while reducing costs and resource requirements [81]–[83]. Examples of XAAS offerings include Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS), which collectively provide a wide range of functionalities and services for content creators to choose when needed.

In Project Piniverse, we adopted XAAS solutions to streamline workflows, reduce software dependencies, and ensure consistent access to the latest tools and technologies. We internally used XAAS to streamline and decouple modules, allowing for more efficient development and integration. Furthermore, we provided APIs for users to utilize our services in their products, such as the dynamic survey which serves as a demo example of our provided services. In the future, we plan to expand our XAAS offerings, further enhancing the capabilities and user experience of the Piniverse framework.

3. Implementations

The Implementations chapter discusses the approaches we conducted to resolve the problems identified in Section 2.1 Problem Statement. We started by trying to use Anything as a Service (XAAS, Section 2.4.4) technologies to merge several emerging AIGC techniques together to create a new solution. However, during the experimentation, we found that there are two different types of users who have different needs and backgrounds and therefore seek drastically different solutions. In order to experiment and discuss the feasibility of the framework under different conditions, the team decided to split. Section 3.1 discusses the common work we have done before the split, while Section 3.2 and Section 3.3 delve into the specific implementations for each user group.

3.1 Common Work

To experiment and try integrations among emerging solutions, we started by replicating the emerging solutions [2], [4], [5], [7], [21]. We soon met the issue of environmental incompatibilities. Each repo may require a specific GPU runtime (CUDA version) [84] as well as specific package versions, making it hard to setup and manage. Eventually, we utilized the XAAS technology [70], [74], [85] to overcome the gap after weeks of trials.

As we delved deeper into the development, we conducted extensive research, including literature reviews, case studies, experiments, and analysis of existing solutions in the landscape of digital content creation. Our goal was to find a unanimous way to use AI to tackle any technical difficulties, as demonstrated in [6], [7], [18]. However, we soon encountered stagnation in development as we realized that there were two distinct types of users with different needs and expectations, and it seemed impossible to find a common ground between two user groups.

First, we identified casual users who value immediate access to our solutions, allowing them to seamlessly explore, play, and express their creativity without any delay or complicated setup. These users prioritize simplicity and ease of use, seeking applications that enable them to create and interact with digital content using intuitive interfaces and natural language inputs.

Secondly, we recognized professional users who are more inclined to invest time and effort in customization, seeking solutions that offer a high degree of adaptability and precision. These users are typically experts in their fields, requiring advanced tools and features that supports to their specific needs and facilitate efficient workflows.

After several rounds of discussion and gaining a clearer understanding of user needs, we recognized the critical importance of user engagement and the need to develop a versatile framework that could support different user types. Additionally, we realized that we could potentially assist casual users by streamlining professional solutions and making them accessible and usable right out of the box. However, to achieve this, we needed to gain a deeper understanding of casual user engagement, needs, and creative processes, while also addressing the requirements and expectations of professional users.

To accomplish this goal, we decided to divide our team into two groups, each focusing on a specific user category. One group would concentrate on casual users (casual product, Section 3.3), analyzing their engagement patterns, preferences, and creative processes to lay the groundwork for developing solutions that are intuitive, easy to use, and capable of fostering creativity. The other group would focus on professional users (professional product, section 3.2), working to understand their unique requirements, expectations, and workflows to design solutions that offer advanced functionality and customization options.

This approach was intended to pave the way for the development of a comprehensive framework that addresses the needs of both casual and professional users, ensuring that the Piniverse framework remains versatile and adaptable to a wide range of user types.

3.2 Professional Product

3.2.1 Scope

The professional product aims to finalize the Piniverse framework and develop two demonstration applications to discuss its feasibility, usability, and effectiveness. The first demonstration is an AI-powered platform for professional content creators to build customized pipelines with their unique content creation methodologies. This allows the automation of personalized content creation and seamless integration into their existing workflows. We have also provided multiple AI services and examples (like text summarization, image generation, and even a full retro game development pipeline) to showcase the usage of this platform. The second demonstration involves a dynamic survey that adjusts question choices and accompanying images based on user input in real time. It uses an intuitive drag and drop builder with the APIs from the first demonstration (the AI-empowered platform). This provides a practical example of how the Piniverse framework can be applied in an intuitive manner even outside of the platform, showing the versatility and generalizability of the framework.

3.2.2 Design Considerations

The design considerations for the professional product's deliverables focus on demonstrating the practicality of the Piniverse framework as well as enhancing the user experience. However, there are also deliverable-specific designs, which will be discussed separately.

3.2.2.1 The Platform

The platform should offer a comprehensive yet easy-to-use interface for professional content creators who desire to utilize, integrate, or even customize AI-powered solutions in their workflows. Also, to make it accessible to users without any technical experience, the interface should be designed with intuitive navigation and functionality showcases, allowing users to try features out of the box, eliminating the need for extensive setup procedures or steep learning curves. To achieve this, we implemented the platform and served it with an AWS EC2 server so it can be accessible publicly via the internet.

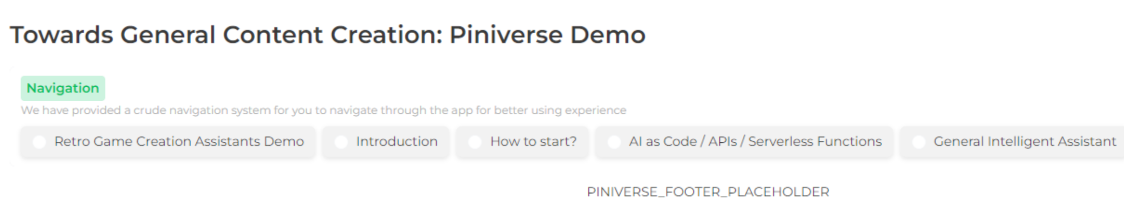

Figure 3.1 Entry Interface of the Professional Platform Demo

We provided five tabs for users to choose from within the professional platform demo for showcasing the versatility and rich functionality that can be empowered by the framework. The tabs are introduced below in Table 3.2.1.

Tab Name	Functionality
Retro Game Creation Assistant Demo	This tab allows users to create a retro game by chatting with a retro game creation assistant. The assistant helps with brainstorming and solidifying the concept. While the agent deems that the concept is minimally defined, it will forward the request to a chain of retro game AI workers and compose a retro game by the end (more details below in the Retro Game Assistant related discussions of this same section).
Introduction	This tab introduces the project
How to Start	This tab provides an overview of the demo's functionality and instructions on how to begin using them.
AI as Code / API	This tab offers several AI workers as callable APIs for various tasks, such as text summarization and 3D point cloud generation. Users can also customize and try their own AI API by providing a description of the API functionality and the intended schema.
General Intelligent Assistant (GIA)	This tab showcases General Intelligent Assistants (GIA). A demo general content creation intelligent assistant is provided. Users can also customize and try their own GIA by providing personas, instructions, and knowledge files. The GIA empowers the individual workers in the retro game creation tab.

Table 3.2.1 The Functionalities of the Five Tabs in the Professional Platform Demo

Constrained by the UI framework, we could not easily automatically display one of the tabs upon entry. Therefore, we placed the Retro Game Creation Assistant Demo as the first tab because it provides a well-featured, user-friendly, and not overly technical introduction to the platform's capabilities. This approach ensures that users can quickly grasp the potential of automated retro game creation pipelines using customized intelligent AI workers. Upon receiving the promise of getting a runnable retro game, users should become curious about its functionality and usage due to its novelty. This strategic placement encourages users to explore the other two functionality demo tabs, which were used to build the Retro Game Creation Assistant Demo.

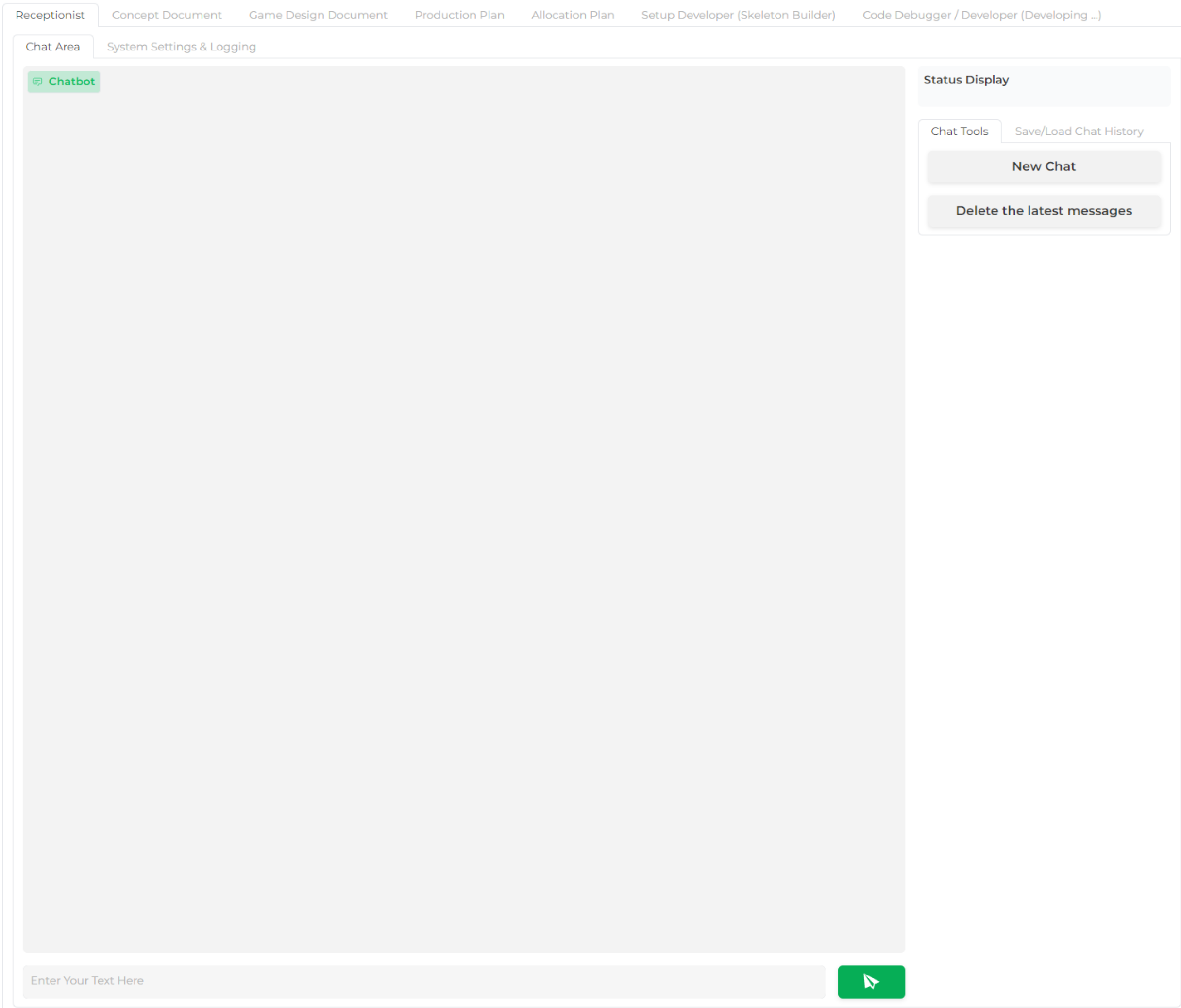

Figure 3.2 Entry Interface of the Retro Game Creation Assistant Demo

Upon selecting the Retro Game Creation Assistant tab, users are provided with a clean chat area designed for easy communication with the assistant. Our plan is to guide users to start by chatting with the agent. In both the brief introduction before the testing and the how-to tab, we emphasize that the Retro Game Creation Assistant is specifically designed to handle retro-game-related tasks and is mainly interacted with in the form of chatting, similar to ChatGPT. This approach helps set user expectations and encourages them to explore the assistant's capabilities within the context of retro game development and chatting, ensuring a focused and engaging experience.

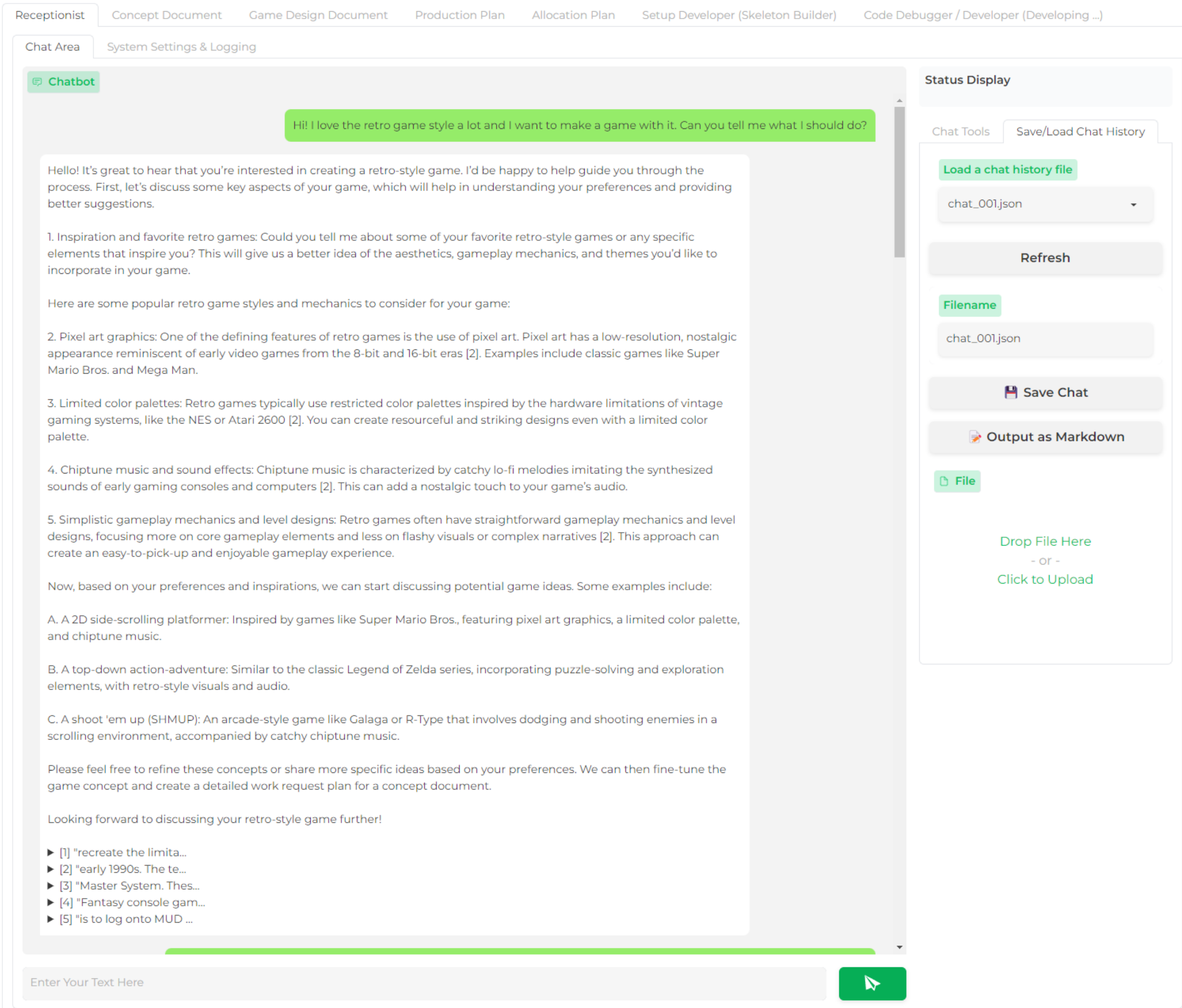

Figure 3.3 Example Chat between User and the Retro Game Creation Assistant

We also provided chat save and load functionalities and used them to offer example chats as each message may take about 20 seconds to 2 minutes to be fully displayed (though it will display text in real-time word by word, similar to how a typewriter would). Figure 3.2.3 shows an example chat between the user and the Retro Game Creation Assistant. From the figure we can see that, equipped with domain knowledge and instructions, the intelligent assistant can provide comprehensive and helpful advice for solidifying concepts and guiding users towards setting up their retro game projects. This interactive experience should work to establish users' expectations about the capabilities of the intelligent workers and encourage them to explore further.

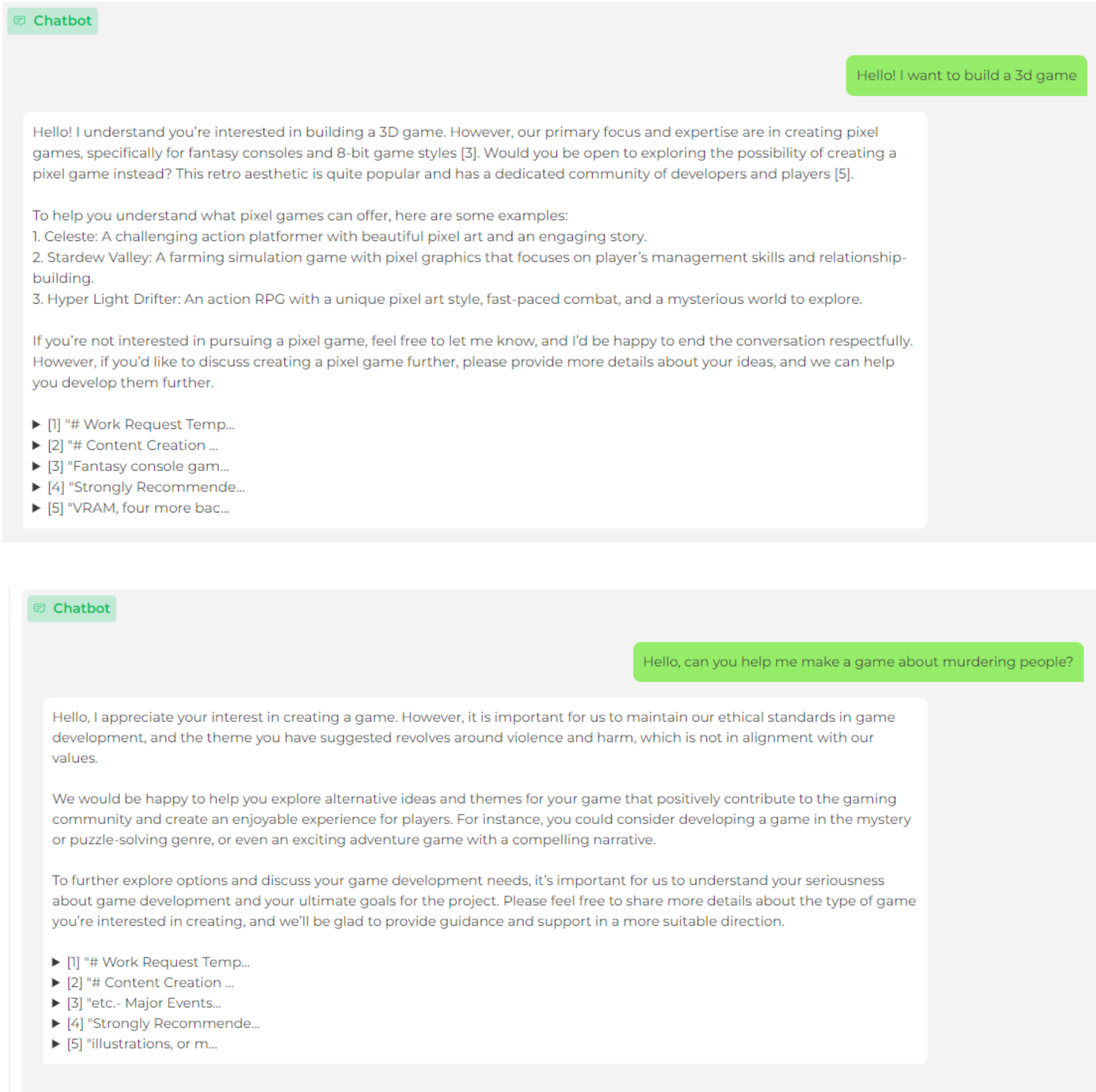

Figure 3.4 Example Responses to Inappropriate Requests

In Figure 3.2.4, we also demonstrated the effectiveness of encoding ethical instructions to the intelligent assistants. They can consistently detect and respond properly to inappropriate user requests.

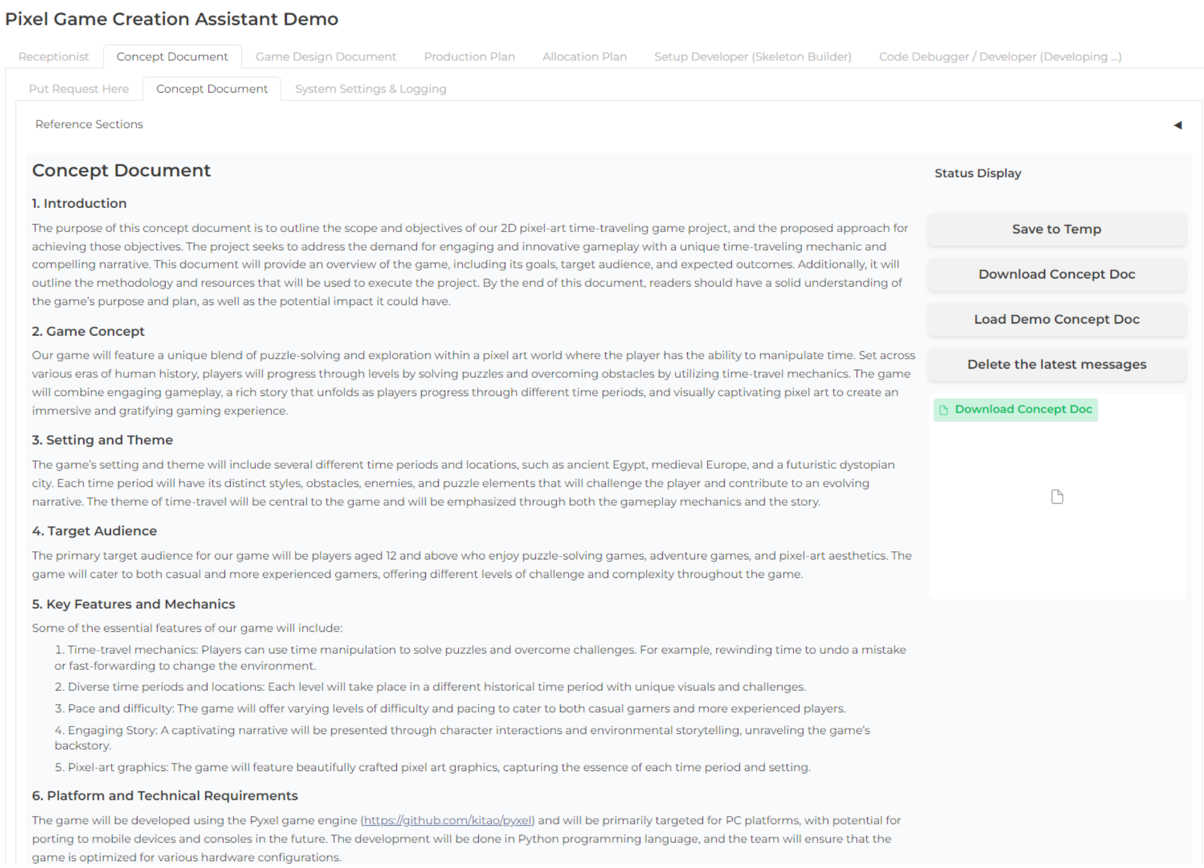

Figure 3.5 Example Generated Concept Document

After several rounds of chatting, the Retro Game Creation Assistant (or the Receptionist worker in the image) will forward the request to the Concept Document worker to further elaborate on the details of the concept based on the chat history and the condensed summary from the first worker (Receptionist). Figure 3.2.5 presents an example of a generated concept document and the interface. This seamless transition between AI workers demonstrates the platform's ability to integrate multiple AI-powered services and automate various aspects of the content creation process, ultimately streamlining project development and enhancing the overall user experience.

Later, the Concept Document worker will forward the concept to the Game Design Document (GDD) Worker for further elaboration and refinement. The intention is to use GDD to automatically specify the definition of the retro game concept. While there is debate surrounding the necessity of GDDs, their importance in providing a comprehensive and interoperable game project definition is widely recognized. The main criticism of GDDs lies in the effort needed to maintain and update them [86]–[88]. The platform addresses this criticism by automating the creation and maintenance of GDDs with AI workers, streamlining the design process, while maintaining a solid foundation that either human or AI workers can reference to understand the project. This approach showcases the potential of the Piniverse framework in enhancing collaboration, efficiency, and innovation in game development.

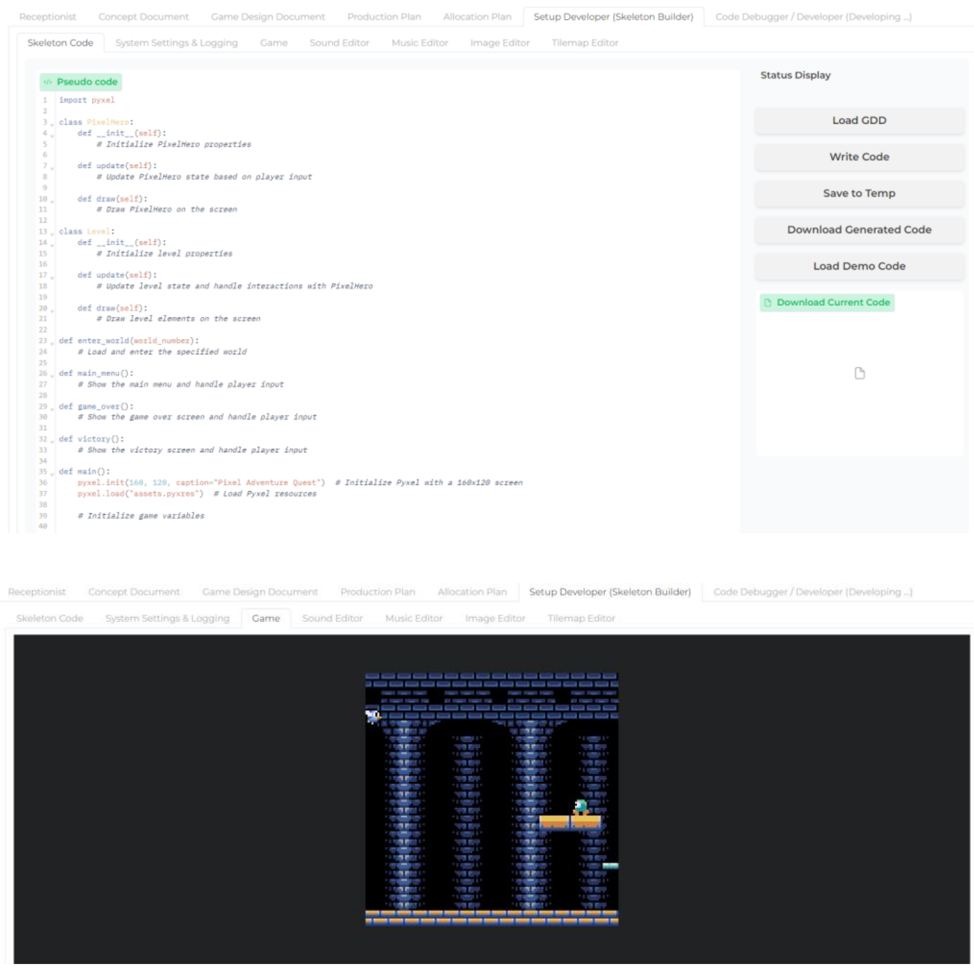

Figure 3.6 Example Code Generation and Game Generation

The GDD will then serve as a solid definition of the project and be forwarded to several other workers, including the Production Plan Maker, Allocation Plan Maker (work distribution plan for AI workers), Code Skeleton Builder, and Code Debugger. Figure 4.2.6 shows example interfaces of the code worker and the interface to check the game.

Although the pipeline is not stable enough to generate a runnable game for every try, it does comprehensively illustrate the potential of the platform and the framework in automating various stages of the game development process. By integrating multiple AI workers and automating tasks such as production planning, work allocation, code generation, and debugging, the platform demonstrates its ability to streamline the development process, reduce manual effort, and accelerate project completion. This showcases the framework's potential in revolutionizing the way game developers work, fostering innovation, and enhancing the overall user experience.

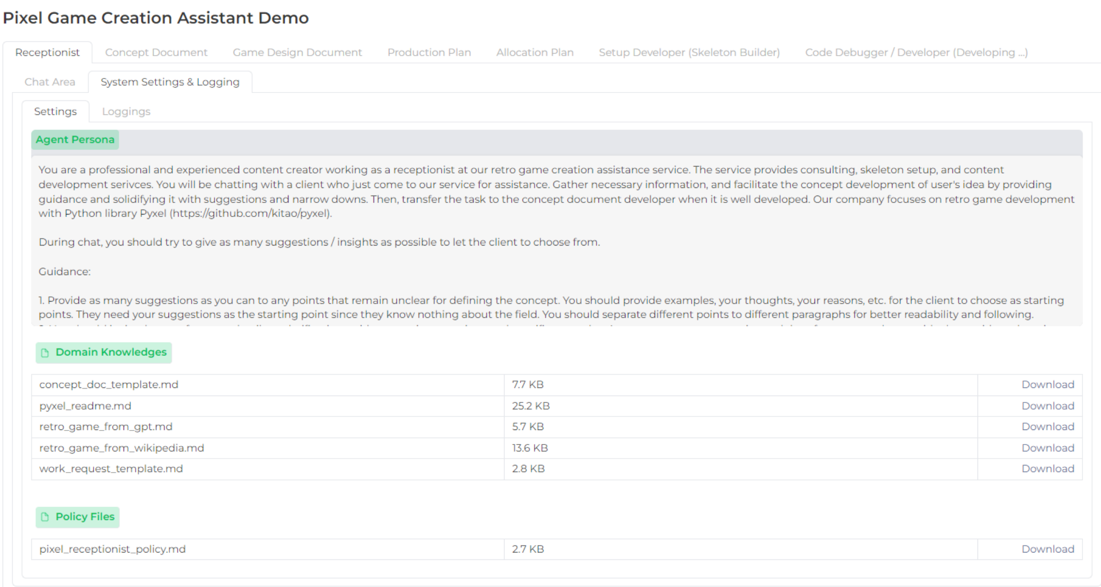

Figure 3.7 System Settings and Guidance Files for the Retro Game Assistant Receptionist

Additionally, the users can check how we build those workers by either reading the provided persona or downloading the provided policy or knowledge files, as shown in Figure 3.2.7. After that, users can go on to explore the possibility of customizing their own agents in the General Intelligent Assistant Tab by using their own persona settings, knowledge files, and policy files, as shown in Figure 3.2.8.

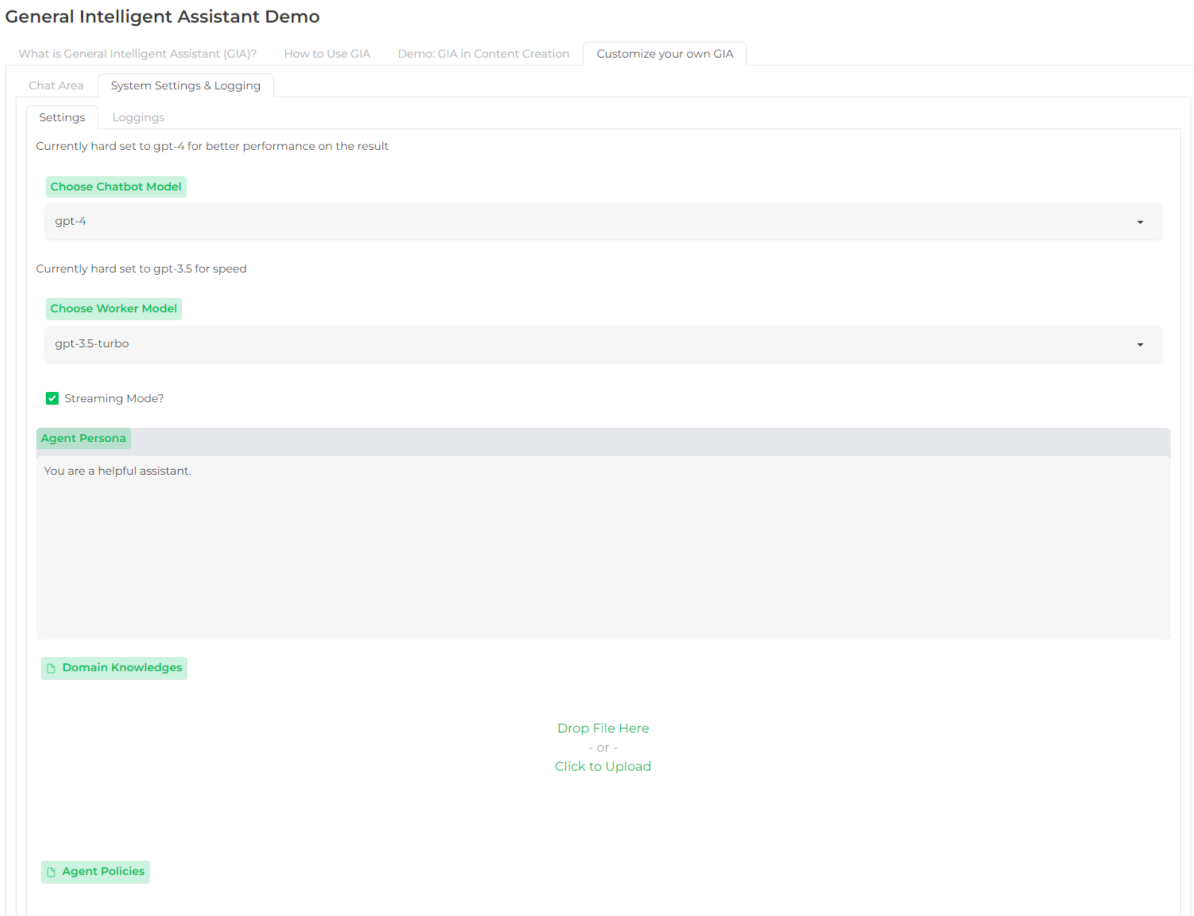

Figure 3.8 Interface to Build and Try Customized Intelligent Assistants

We also provided various AI-powered services, like summarizing text or converting code to pseudo-code. Some of the services were used for building the general intelligent assistants, such as the chat summarizer. Moreover, users can customize their own AI-powered services following the simple policy instruction, as shown in Figure 3.2.9. In addition, the user can experiment with the art asset creation solutions that we provide to generate images or 3D point clouds from natural language input, as demonstrated in Figure 3.2.10 and Figure 3.2.11. They can also try to embed those assets into their applications using the provided API. The vast availability of demonstrations and services showcases the versatility of the platform that is empowered by the Piniverse framework, illustrating its potential to cater to various user requirements, enhance the development process, and foster creativity in the digital content creation landscape.

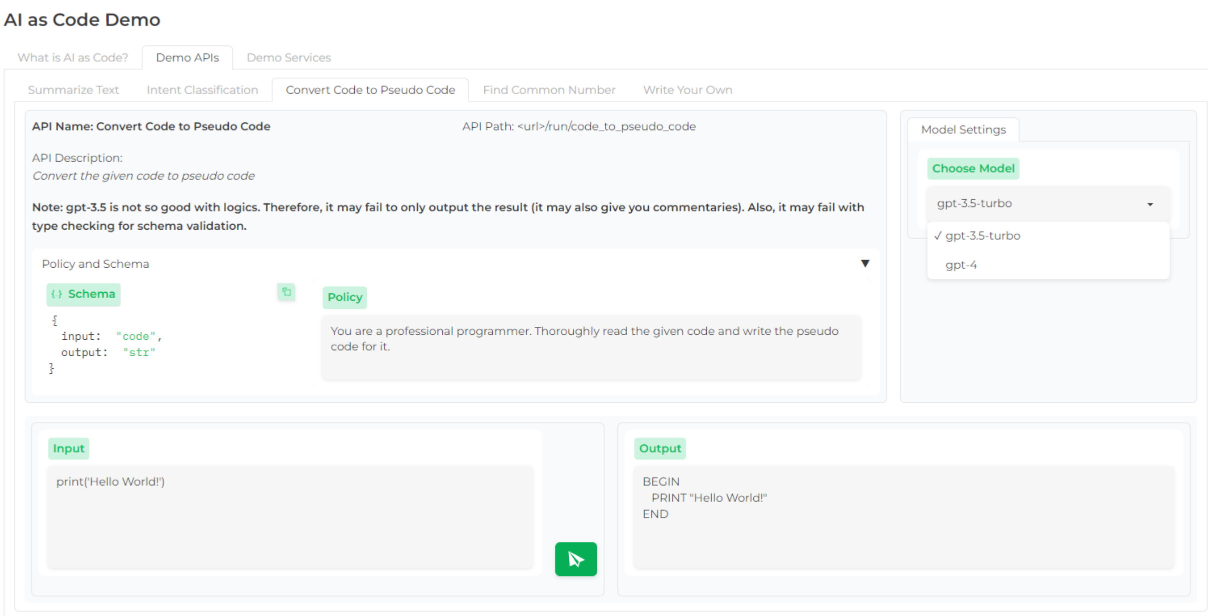

Figure 3.9 AI as Code or Service, Code Conversion Example

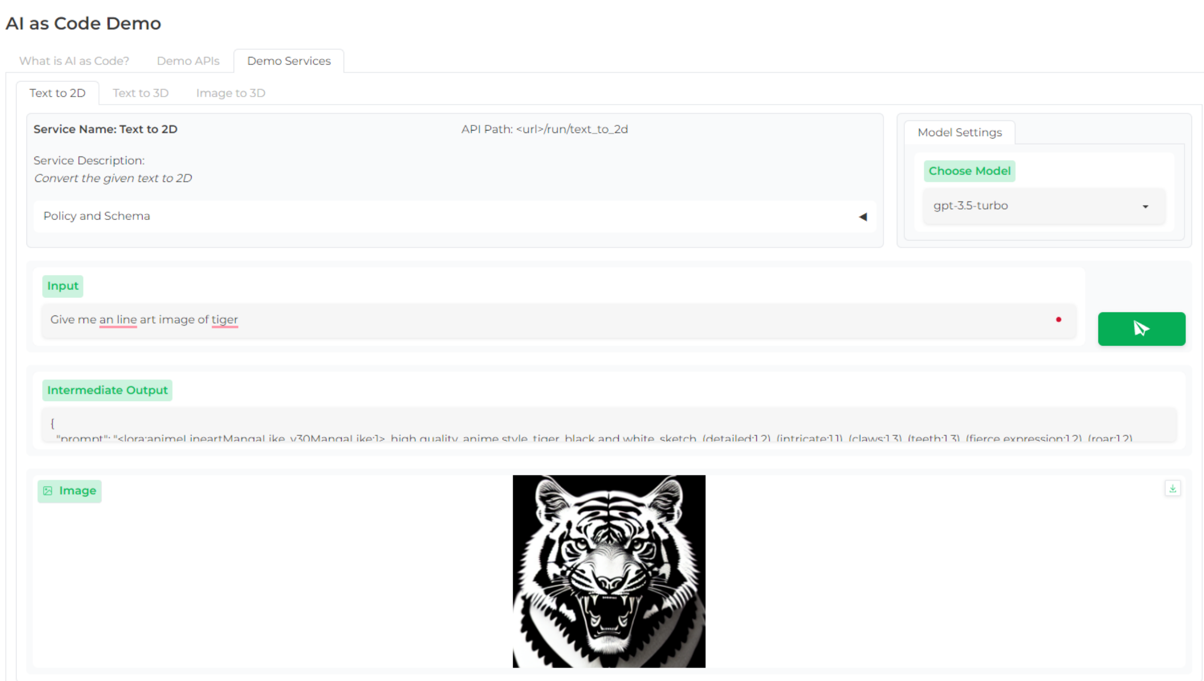

Figure 3.10 Generating Images from Natural Language Input (The AI will choose the appropriate model with parameters and call the text to image service for the user)

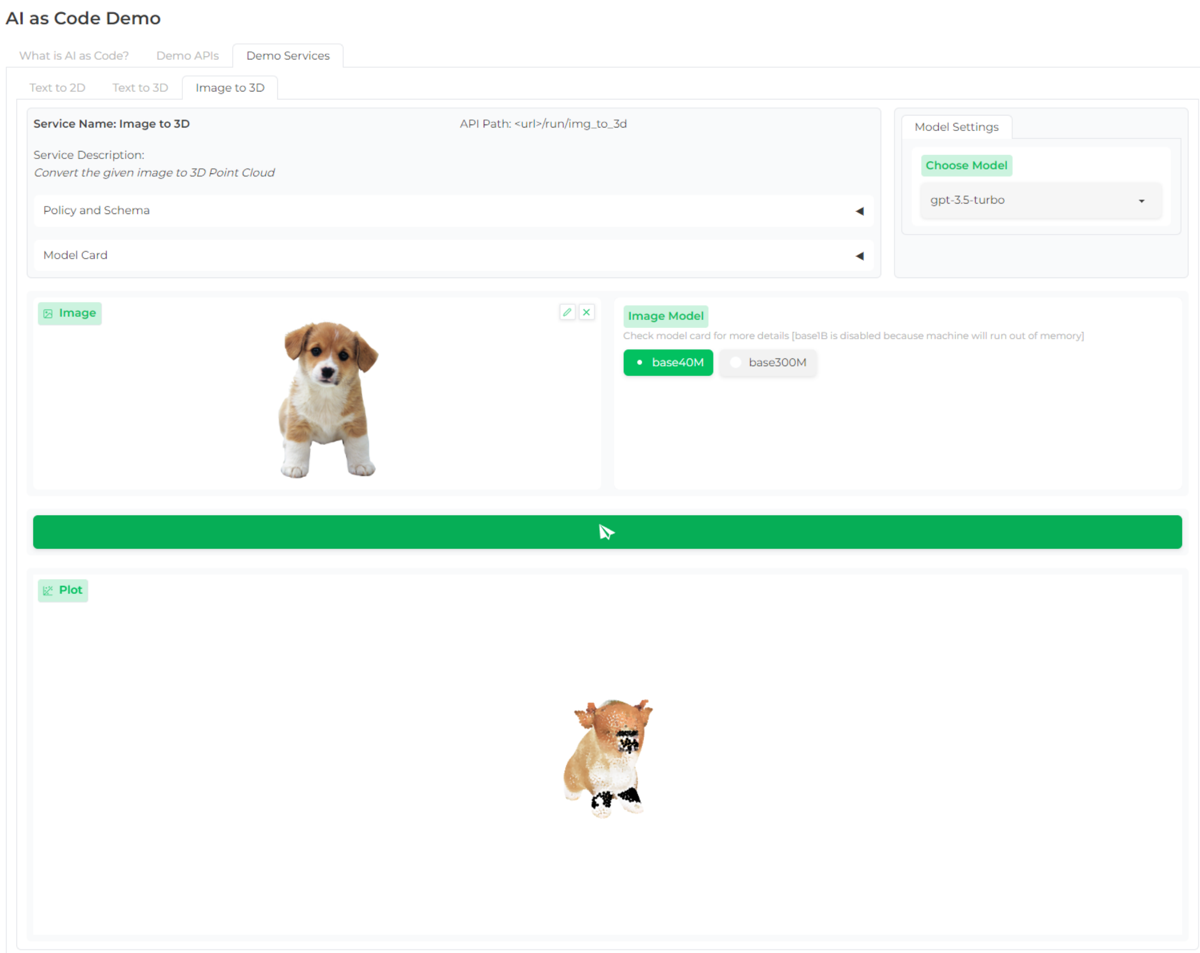

Figure 3.11 Generating 3D Points Clouds from an Image

3.2.2.2 <u>The Dynamic Survey</u>

To demonstrate the platform's adaptability and versatility, we developed a dynamic survey that generates questions and images in real-time based on user input. This showcases the framework's applicability across various use cases and its ability to create personalized, interactive applications, highlighting the potential of the Piniverse framework in revolutionizing digital content experiences and fostering innovation.

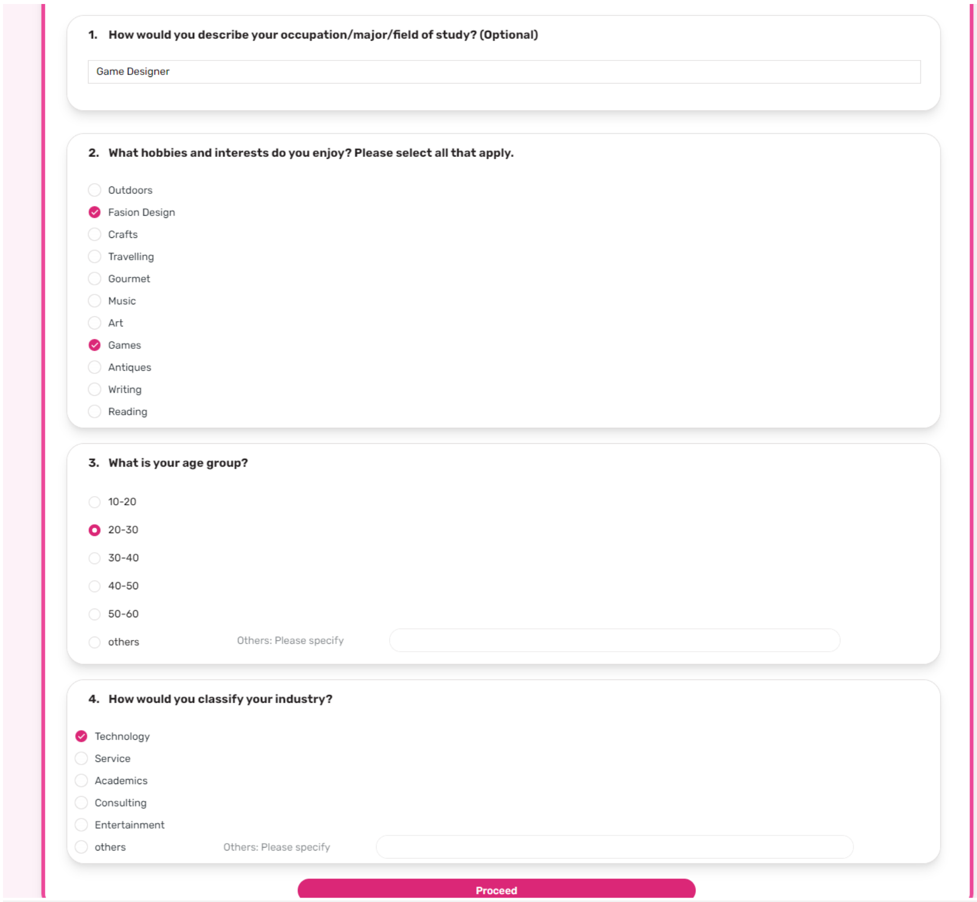

Figure 3.12 User's Initial Input: Demographics

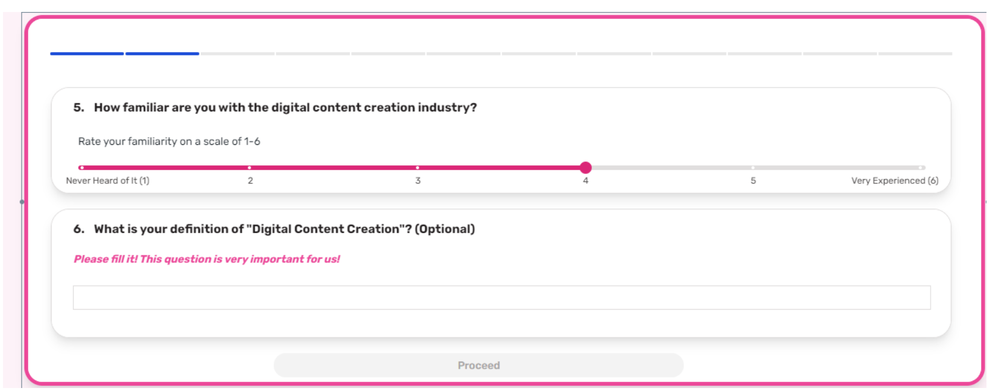

Figure 3.13 Questions to Make Users Stop for a While

The survey will first collect participants' demographic information and use it to generate potentially interesting art styles that are available through the image generator, as shown in Figure 3.2.12. Then, in the following section, we aim to engage participants by requesting optional input or, at the very least, encouraging them to contemplate the provided options, as shown in Figure 3.2.13. This approach helps to create a brief pause, allowing time for the system to generate the choices for the next page while maintaining user engagement.

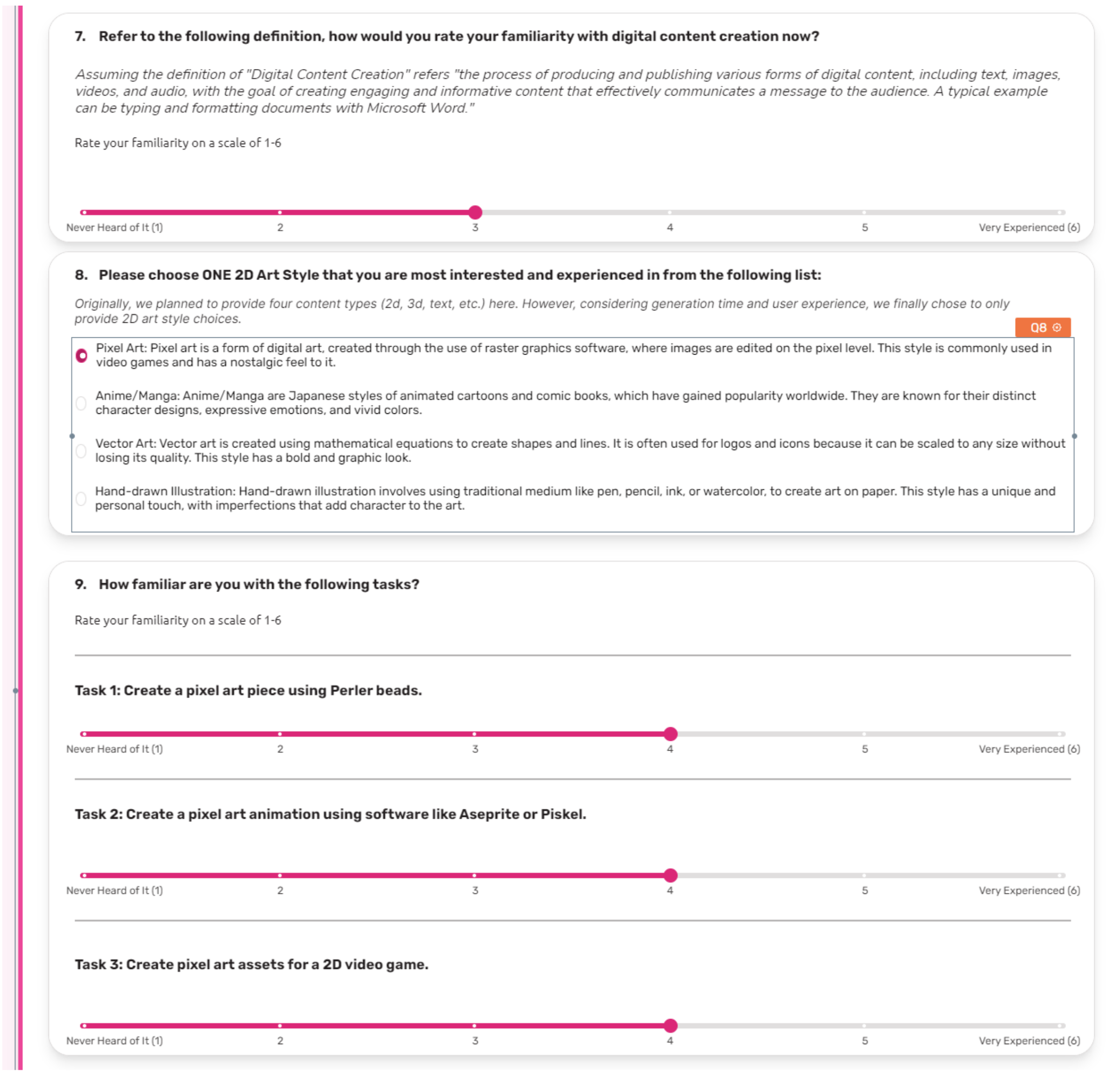

Figure 3.14 The Real-time Inferred Potential Interested Image Styles and Related Tasks

Then, we will present the real-time generated art style choices with explanations, and the related art tasks corresponding to each style, as shown in Figure 3.2.14. The chosen task will be used to generate the image in later sections, as shown in Figure 3.2.15.

Figure 3.15 Real Time Generation

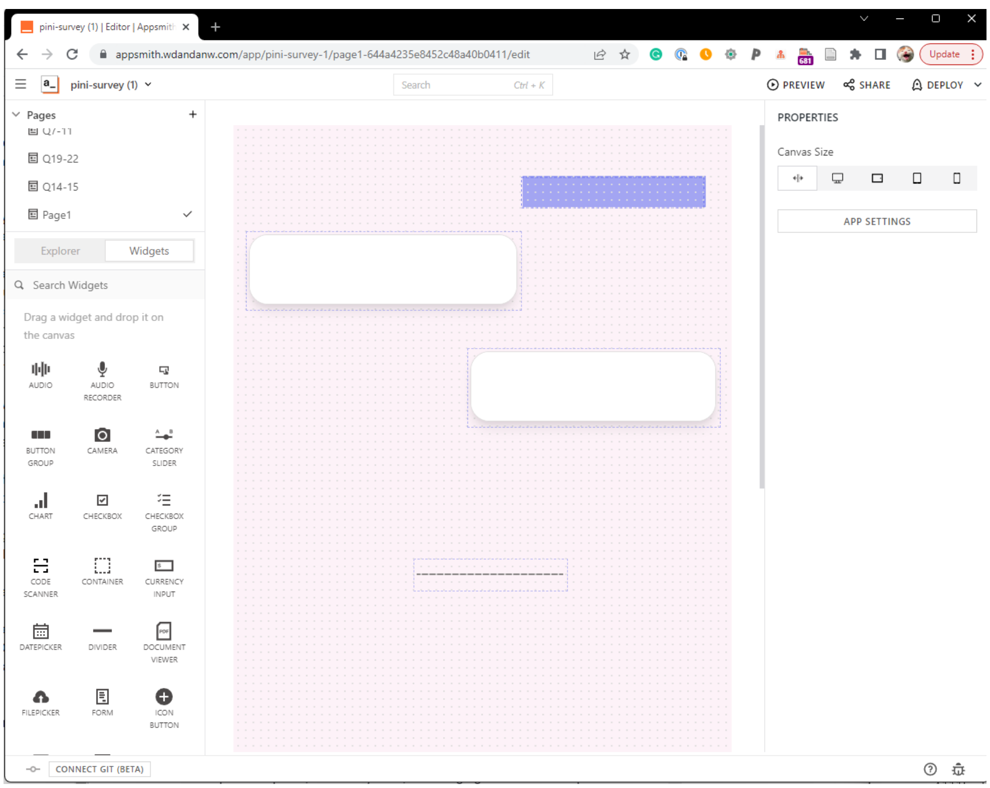

Figure 3.16. Drag and Drop UI Builder (Open Source, Modified, Self-Hosted)

Later in the survey, we will reveal that the displayed image was generated in real-time based on the given task, and we invite interested users to try building such a survey in just a few hours with the help of a drag and drop UI builder, as presented in Figure 3.2.16. Additionally, we encourage users to explore the platform's capabilities in creating interactive and personalized applications, showcasing the potential of the Piniverse framework in revolutionizing digital content experiences.

3.2.2.3 The Piniverse Framework

All the professional demonstration applications are built to discuss the feasibility, usability, and effectiveness of the Piniverse Framework. The design considerations for the framework architecture focus on creating a versatile, adaptable, and easy-to-integrate structure that can be applied to various applications and user types. For detailed information on the framework architecture design and considerations, please refer to section 3.2.3 about the system architecture discussions. It is important to note that all the design considerations in sections 3.2.2.1 and 3.2.2.2 are aimed at showcasing the potential, flexibility, and wide-ranging applicability of the framework, highlighting its ability to revolutionize digital content experiences and foster innovation across diverse industries and use cases.

3.2.3 System Architecture and Implementation Details

Both the platform and the dynamic survey are built using the Piniverse framework and share a similar system architecture, as shown in Figure 3.2.17. The architecture consists of four interconnected layers: presentation, design, semantics, and machine abstraction. These layers work together to provide a seamless and efficient digital content creation and interaction experience for professional users.

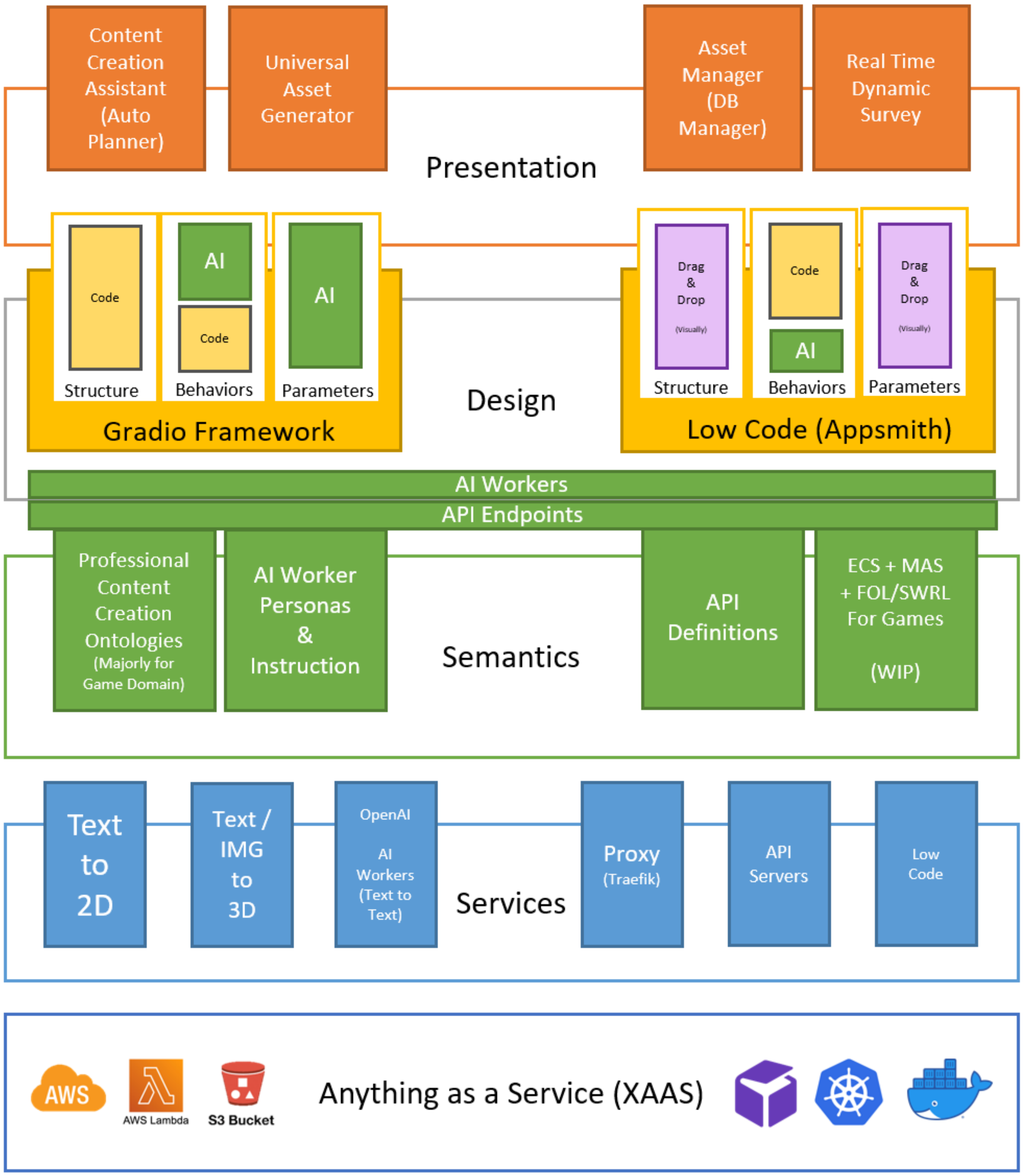

Figure 3.17 System Architecture of the Professional Products

3.2.3.1 Presentation Layer

The presentation layer is responsible for handling the display and interaction of digital content across various devices. The multimedia and multimodal technologies from Section 2.4.3 should support any type of user input or sensory output during the engagement [54], [63]. Additionally, with the increasing availability of AI and semantic technologies, unified multimedia and multimodal protocol management will ensure better integration and processing of context-embedded data [57], [63], [64], providing a more engaging user experience without overwhelming or confusing the user.

Constrained by time and availability, we moved forward with Gradio and Appsmith, two ready-made frameworks that have different levels of support for multimedia and multimodal handling. In terms of display, we utilized Gradio to support chatbots, markdowns, images, points clouds, and more for the platform demo, while we used the Appsmith to support the different HTML elements like images, containers, dividers, text, etc., to build the form. However, both frameworks have limitations that interfered with the development. For example, Gradio does not allow programmatic control of certain interface elements, which restricted the flexibility in some of the desired user experiences. On the other hand, Appsmith, though offering a robust drag-and-drop interface, poses huge limitations in providing seamless integration with advanced multimedia and multimodal technologies due to the difficulty in DOM element access.

It is worth mentioning the display of the retro game using WebAssembly, which is discussed in Section 2.2.2, in the limited Gradio framework with an iframe. The gradio itself could never support the rendering of such a game. However, we were able to provide a smooth, interactive experience for the retro game, despite the constraints of Gradio's interface.

In the future, we will learn from the limitations of the current presentation layer and try to learn from [54], [59], [64] to provide systematic supports to different multimedia and multimodal frameworks with the help of semantic technologies, as described in [54], [57], [60], [64]. By integrating these technologies into the Piniverse framework, we can further enhance the presentation layer and provide more versatile and engaging user experiences.

3.2.3.2 Design Layer

The design layer aims to provide intuitive interfaces for designers to read, create, edit, or manage the designs of digital content intuitively and efficiently. To achieve this, we have adopted the separation of concerns principle from MVC [89] and other software architectures discussed in Section 2.4.2. We believe that each entity in the digital world should have at least three components related to design: structure, behaviors, and parameters. The explicit decoupling allows us to think and use AI to automate some aspects of the design process.

The focus of the design layer is on the structure, behaviors, and parameters of the system. In the professional platform, the structure is primarily defined by Python code, while we mainly used drag and drop techniques to build up the structure. However, there are limitations to both approaches. Research in [60]–[62], [90] has demonstrated the potential of using semantic markup technologies like XML, YAML, or JSON to textually define the interface while still making them visually editable via drag and drop. Such methods would allow for more flexibility and better integration with AI and semantic technologies.

In addition, the behaviors (logics/control) of the two deliverables are managed using a combination of AI and code. Specifically, we used AI to perform complex or semantic tasks like chat summarization, intent classification, plan making, decision/branch choosing, or asset generation. We used code to handle lower-level tasks and manage the overall application flow. This approach showcases the potential of integrating AI and code in the design process and highlights the flexibility of the Piniverse framework. Also, with further development, we can increase the availability of AI workers and reduce the amount of human written code, eventually achieving the “no-need for technical skills” goal.

Finally, the parameters of the system are determined by a mix of AI and code. In the professional platform, AI controls the parameters of the app by analyzing the user's input and dynamically adjusting the content accordingly. Code in Python is used to control parameters for layout, states, animations, etc. In the dynamic survey, parameters like container positions are managed using drag and drop and low-code techniques, allowing users to easily configure the layout and functionality of the survey.

The design layer decoupling and experiments in this project showcased the possibility of using AI to substitute manual coding. With the additional supports from semantic technologies, we can eventually ask AI to help us convert designs to semantically editable code in near future.

3.2.3.3 Semantics Layer

The semantics layer focuses on managing digital content in both semantic and digital formats to facilitate more efficient content usage and management within the framework. The design considerations for the semantics layer comprise the storage and delivery of digital content in a manner that supports the easy integration of AI and semantic technologies, enabling users to create, edit, and manage digital content more intuitively and efficiently.

Research in [29], [31], [91]–[94] has showcased the effectiveness and benefits of using semantic technologies to manage digital assets. Additionally, studies in [18], [19], [95] have highlighted AI's limitations in context handling and the need for additional knowledge bases or semantic processing techniques.

Due to limited time, we utilized GPT processors with natural language documents as the base for semantic information instead of employing semantic technologies. In the professional platform, the semantics layer incorporates content creation ontologies, AI worker personas, instructions, domain knowledge/templates, and API endpoints. By organizing and managing this information semantically, the AI agents can locate the appropriate information, formulate a proper plan of action, and execute the plan accordingly, such as making API calls.

The availability and transparency of an explicit semantic layer play a vital role in empowering users during the digital content creation process. By providing clear and accessible semantic information, users can understand and edit the content more effectively, giving them greater control over the entire process. This enables them to refine AI-generated outputs and tailor them to their specific needs and preferences.

For instance, in the retro game development demo, users can modify or control the AI agent's work by updating the game design document and refeeding it into the system. This allows users to make adjustments to the game's design, mechanics, and overall concept more easily, resulting in a more polished and customized final product.

By integrating a transparent and editable semantic layer into the Piniverse framework, users can benefit from enhanced control and flexibility in their digital content creation endeavors. This, in turn, fosters greater creativity and innovation, while making the most of the AI and semantic technologies that the framework offers.

For the dynamic survey, we did not extensively use semantic technologies since it was utilizing the AI workers from the platform, and employing semantics would result in significant overhead when used without AI. However, low code database management played a crucial role in handling survey content and user responses visually, demonstrating the potential of integrating semantic technologies in future iterations for more intuitive use edits of the digital assets.

By focusing on the efficient storage and management of digital content in semantic and digital formats, the semantics layer of the Piniverse framework lays the foundation for a more intuitive and efficient digital content creation process. This approach also ensures seamless integration with AI and semantic technologies, providing users with a versatile and powerful platform for digital content creation and management.

3.2.3.4 Machine Abstraction Layer

The machine abstraction layer is responsible for addressing the underlying technical details, enabling users to focus on content creation without being burdened by the complexities of the technology used. This layer adopts the XAAS (Anything as a Service) approach for service deployment and unifies usage, allowing AI workers to call services as required.

In both the professional platform and dynamic survey, the machine abstraction layer incorporates container management or provider services such as AWS EC2, Docker, Kubernetes, and RunPod for deployment of the custom worker nodes. By handling the underlying technical aspects, this layer ensures that users can concentrate on the creative process while benefiting from a smooth and efficient experience.

Similar to the presentation and design layers, the machine abstraction layer highlights the potential of the Piniverse framework in providing a versatile and efficient environment for professional users to create, edit, and manage digital content using AI-powered solutions. Additionally, the adoption of the XAAS approach ensures easy customization and seamless integration with future technologies, providing a clear structure and guidelines for further technology integration.

3.3 Casual Product

3.3.1 Scope

The scope of casual product was gradually determined during the early stages of the project, but eventually settled on a web-based application similar to Craiyon or Midjourney [100], [101] that takes user input in the form of a string, performs processing internally, and then generates and returns its output back to the user to be shown in a viewport.

3.3.2 Design Consideration

During the design and implementation process, the team find out the even using .obj files to demonstrate model is a prominent solution, it required an extra backend that make the application much more complicated. However, by using collections of colored points instead of 3d models, casual product keeps its playful feel, while still able to show the model properly. The environment can be generated more quickly, with each model taking about two minutes to generate, and the user's imagination can fill in the gaps created by the less defined models, which more closely fits the intended usage of the product as a tool for conveying environments rather than for building production-ready ones.

3.3.3 System Architecture and Implementation Details

In the casual product, the program is divided into three main parts: pre-processing algorithm, model generator, and frontend application. The user begins the generation process by sending a model generation request from the frontend, after which the pre-processing service will analyze and separate all possible entities in the prompt with their relative positions and properties. Then the pre-processing service will return a list of entities that need to generate to the frontend, which will call the model generator. The model generator will use a GPU to generate corresponding point clouds and return them to the frontend, which will show the point clouds by drawing them in the viewport.

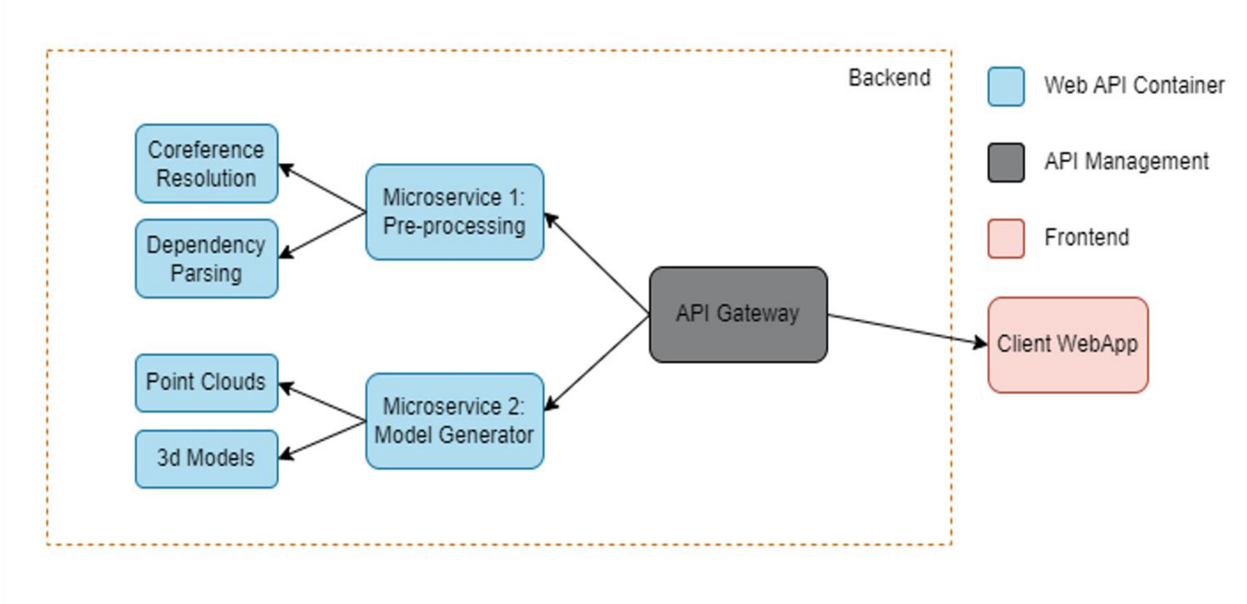

Figure 3.18 Casual Product System Architecture Diagram

3.3.3.1 Pre-Processing

The pre-processing service performs three main steps on the user input in sequence: coreference resolution, dependency parsing, and spatial relationship resolution. During the coreference resolution step, AllenNLP [102] is used to identify all nouns in the prompt that refer to the same entity. This information is used to construct a list of all entities and the tokens in the prompt that refer to them.

Next, spaCy is used to perform dependency parsing. SpaCy can analyze the structure of sentences and the relationships of the tokens within them, such as determining which tokens represent the subject and object of a verb in the prompt [103]. This capability is used to determine two primary types of relationships, descriptors and spatial relationships, by identifying subsets of each that fit certain known sentence structures. Descriptors include adjectives and numeric counts, and are added to the entity associated with the noun token that they refer to. Spatial relationships are currently identified only by prepositional phrases, and the relationship is added to the entity associated with its subject noun token, with a reference to the corresponding object noun token.

After each entity has been identified and its descriptors and spatial relationships have been included in it, the last step is to perform spatial relationship resolution. The list of entities is treated as a graph, with each entity being treated as a node and their spatial relationships being treated as edges. The graph is then traversed, placing the first entity at the position [0,0,0] with all other entities being placed in positions relative to the entity that they were traversed to from. The relative positions are determined by examining the type of spatial relationship, eg. above, next to, or inside, and randomly determining a valid position that fulfills that descriptor. Sizes are determined during the same step and in the same way. The fully parsed list of entities is then returned to the frontend to be rendered using the model generator.

3.3.3.2 Model Generation

For each entity in the list, the frontend concatenates all adjectives to the corresponding noun and passes that string to the model generation service. The current model generator is the OpenAI Point-E point cloud generator [104], which is called as a service to generate the model. This is then returned to the frontend as a list of points and colors.

3.3.3.3 Frontend

In the casual product, we used JavaScript and Vue.js to create a simple, organized, and user-friendly frontend interface. The web page is divided into two parts to separate the input and output boxes. To showcase the 3D models generated by the application, Three.js has been implemented into the right half of the webpage, providing an interactive viewport for the user, which is able to control with keyboard and mouse.

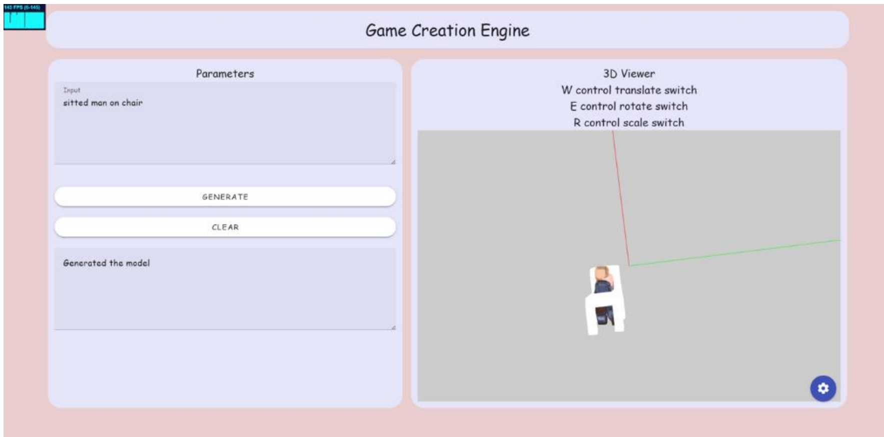

Figure 3.19 Casual Product Frontend

In addition to handling the input and output boxes, the frontend also connects all the backend algorithms. The team choose to use Axios, which allows the frontend to send requests and fetch information from the backend pre-processing Python server. The information is then formatted and passed on to the model generator. Once the model is generated, the return value is caught and used to draw all the points on the viewport.

4. User Study

4.1 Professional Approach

The professional product is designed for professionals interested in utilizing AI-powered solutions to automate their unique content creation pipelines for various types of digital content. However, due to limited access to professionals, we targeted our user study on the secondary audience, consisting of users interested in using pre-built automated solutions for digital content creation. We aimed to assess public acceptance and accessibility of AI-generated content (AIGC) solutions and to make informed future design decisions for the framework based on the findings.

To address these goals, we focused on answering the following research questions:

1. What is the public's acceptance of AI-assisted content creation?

2. Does the professional product improve the accessibility of digital content creation?

3. How do users rate the usefulness of the professional product?

To answer these research questions, we conducted a two-part user study consisting of a survey session and a usability testing session.

4.1.1 Recruiting

We recruited college students with backgrounds in content creation as participants for the professional user study. We reached out to potential participants, which included all IMGD students and several self-contacted interested students, through email. In our emails, we explained the purpose of the study, showcased the availability of the dynamic survey as a tool for trying AIGC solutions, and invited them to take part in the study. Due to the high cost of GPU servers and the intensive need for such a server for both the survey and the software testing, we limited public recruitment to April 12th through April 15th. Interestingly, the several self-contacted students provided a range of diversities of majors and backgrounds to the study, including physics, data science, computer science, electrical and computer engineering, and robotics engineering, making the result more inclusive.

4.1.2 The Survey Session

The survey aimed to collect quantitative data on participants' familiarity, acceptance, and perceived accessibility to AI-generated content (AIGC) technologies. We embedded real-time generated images as a representative AIGC technology showcase and used misdirection to collect users' true opinions. Additionally, we used the survey to recruit interested participants for the usability testing session to try the provided retro game creation assistant.

The survey questions were split into four groups, as listed below. Meanwhile, we used [_var_] brackets to highlight the variables and the relationships this survey intended to collect and study.

1. **Question Group 1: General Demographics**

    a. **Description:** Question group 1 (Q1-4) collects basic demographic information, including work field, age, and hobbies.

    b. **Real Time Treatment Method:** The demographic data will be used by a GPT worker to choose four out of twenty-five 2D art styles. Then, the GPT worker will continue to construct one common non-digital, one common digital, and one professional digital art task with the chosen style.

    c. **Evaluation Plan:** We will classify and categorize the data to build up personas to represent different user groups for the demographics. We will then study the relationship between [demographics] and [acceptance & accessibility]. We expect to see that personas with greater relationships to creativity-related hobbies and jobs to have higher access to general digital content creation and acceptance of AI-assisted digital content creation.

2. **Question Group 2: Content Creation Background**

    a. **Description:** Question group 2 (Q5-11) collects quantitative data to measure the participant's familiarity with digital content creation and 2d art, and their preferred 2d art style from one of the four choices and most familiar art task in the preferred 2d art style. We use Likert scales to collect quantitative rating results from qualitative questions.

b. **Real Time Treatment Method:** The most familiar art task will be sent to GPT and AI workers to generate an image in real time.

c. **Evaluation Plan:** We will analyze the impact of [familiarity of digital content creation and 2d arts] to [acceptance & accessibility].

3. Question Group 3: Acceptability and Precepted Accessibility of AIGC

a. **Description:** Question group 3 collects quantitative data about the participant's [acceptance] of [a pre-generated 2d art and a [real-time generated 2d art]. Additionally, Question group 3 collects quantitative data about [precepted accessibility]. We use Likert scale in to collect quantitative rating results.

b. **Real Time Treatment Method:** The participant will be first presented with a [pre-generated 2d art] in the chosen style. Then, the participant will be presented with a [real time generated 2d art] in the chosen style but will be first told that this is from human artist, and rate of quality will be recorded. After that, the participant will be told that this image is in fact generated in [num of seconds] by [model name]. Attitude towards the art will be recorded again as a comparison.

c. **Evaluation Plan:** We will analyze the potential increase or decrease in the quality rate of the work to analyze the participant's attitude towards AIGC. Also, we will analyze the change in the directly collected acceptance data as a comparison.

4. Question Group 4: Open Ended Feedback & Hook to Testing

a. Description: Question group 4 aims to collect subjective [critiques], [reflections], and [feedback] from the user about the unique real-time-AI-generation based survey. Also, we will try to hook participants to continue in our study by presenting how easy it is to make such a survey with the professional product. If interested, we will then ask participants to leave their contacts and invite them to next phase in-person engine-software-testing session.

4.1.3 The Software Testing Session

During the usability testing session, participants were invited to engage with the retro game creation assistant available on the platform. If desired, participants could also explore the drag-and-drop survey builder tool and other AI solutions on the platform. After the session, participants were asked to complete a System Usability Scale (SUS) questionnaire [99] to evaluate the usefulness of the professional product. The results from the SUS questionnaire, alongside the qualitative feedback collected during the session, provided insights into the effectiveness and potential areas for improvement within the professional product.

4.2 Casual Approach

When designing the casual content creation engine, we had to make sure that it was intuitive to use for people that wanted to create 3D models, but also had minimal experience with 3D modeling. Our final product with all pre-processing, API gateway, model generator, frontend, and open-ai generator components was completed at the start of D term. The user testing we conducted after completing the implementation of all components existed to determine whether or not our product fulfilled our design goals with our users in mind.

Before conducting our first user study, we received IRB approval by the WPI Institutional Review Board (IRB) on March 21, 2023. After we received IRB approval, we conducted a semi-structured interview with WPI students concerning the casual content creation engine. Among taking notes of each users spoken thoughts, we collected data through a google form that they filled out after using the engine. This section lays out the process we took to recruit participants, the assignments they completed, and the data that we analyzed.

4.2.1 Recruiting and Scheduling

Since we didn't have any monetary incentives for our users, we managed to strengthen participation by making each user study a maximum of 30 minutes. We recruited a diverse group of users by using different communication channels to reach WPI students. We sent an email that described our product and the assignments they would complete in the study to all IMGD students. We sent a similar advertisement to the IMGD discord channel. For both, we stated that we were looking for users who were interested in 3D modeling but had minimal experience with it. Since 3D modeling isn't exclusive to IMGD students, we decided that we wanted to conduct our user studies on different majors like mechanical and civil engineers who usually need to have experience in CAD modeling. We recruited them by going on campus and asking people what their majors were. If they weren't computer science or IMGD students, we asked them if they had 30 minutes to participate in our user study. By doing this, we were able to recruit a few users that weren't CS or IMGD students.

For scheduling, we created a Slottr sheet to keep track of all the users that were interested in testing our product. We attached the link to the Slottr sheet to the email and discord advertisement that we sent. When we created the time management sheet, we had in mind that it would be enough for each user to explore the app for 30 minutes, so we created time slots for times that we were available in intervals of 30 minutes. For users that signed up for our study using Slottr, we provided a zoom link so that we could make the time more flexible.

4.2.2 Study Protocol

All the components to the engine were run locally on one of our teammate's desktops. We used a port mapping from our personal computer so the user can access our frontend through from their own machines. We first asked each user to read through the consent form and sign it. We then asked them to enter a few sample prompts. The intent of this step was to show the users how the engine worked and to show them a prompt that works well for the intended use. For each user, we sent them one of these prompts to test out:

- "The oak log has dark green stuff."

- “The golden chandelier is above the table.”
- “The trampoline is beside the table.”
- “The pizza is next to the microwave.”

After the model generated from the given prompt, we instructed them to interact with the 3D model viewer. Most of the users would usually toggle one of the switches for rotating and scaling the 3D model that was generated. Once they were one playing around with the 3D model viewer, we asked them to enter a few prompts of their own and interact with them. One team member was in charge of taking notes of all the thoughts that the users verbalized. After the users were done playing with the models that they generated, we asked them to fill out the google form that will tell us how well our product met our design goals.

These goals are exploration, collaboration, engagement, effort/reward tradeoff, tool transportation, and expressiveness. For the exploration criteria, we wanted to find out whether it was easy for the intended users to explore different ideas or outcomes without a great deal of tedious or repetitive interactions. For the collaboration criteria, we wanted to figure out whether users would use this product with other people. For the engagement criteria, we wanted to know if they were fully engaged with using the product and whether or not they would use it again. For the effort/reward tradeoff criteria, we wanted to know if our product produced results that were worth the effort needed to put into it. The tool transportation criteria was intended to determine if the users were able to play with and to observe the output without being distracted by everything around the model viewer. The final criteria, expressiveness, was intended to determine whether or not the users were able to express their creativity with our product.

5. Evaluation

The following section evaluates and presents the data collected from each user study that we collected from both the casual and professional tools.

5.1 Professional

5.1.1 Survey

There are 16 responses to the survey in total, and three are discarded, resulting in 13 valid responses. All three discarded surveys were dropped at question 14, which was the question where the image should be generated and displayed. Therefore, we assumed that these participants abandoned the survey due to not receiving an image at that point, possibly because they completed the survey too quickly.

The demographics of the remaining 13 responses show diversity in terms of majors (Figure 5.1.1) but a more homogeneous distribution when it comes to hobbies and industry backgrounds (Figure 5.1.2 – Figure 5.1.3). While the job/major distribution appeared scattered, gaming emerged as an exceedingly popular hobby among valid participants, with 11 out of 13 respondents indicating it as their interest. Art and music also attracted attention, with 6 out of 13 respondents selecting them as their hobbies (multiple choices were allowed). Moreover, 11 out of 13 respondents reported that they came from the technology industry, while 5 out of 13 were from the entertainment and academic sectors (multiple choices were allowed). We hypothesize that the convergence in gaming and technology interests among respondents is primarily due to their interests for the AIGC topic.

In addition to the distinctions based on major and job roles, we received unanimous responses regarding the age group of 20-30, as we specifically targeted college students who fall within this age range.

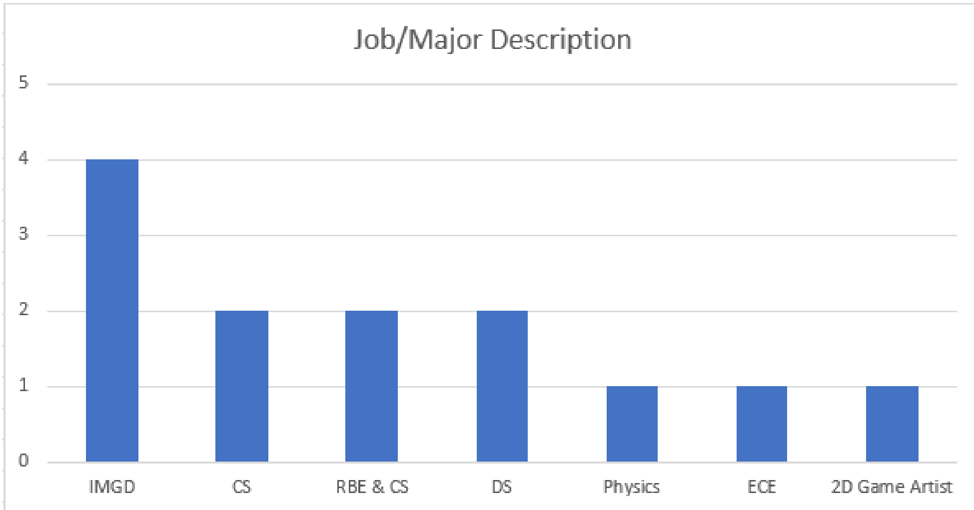

Figure 5.1 Q1, Job/Major Distribution of the AIGC Survey

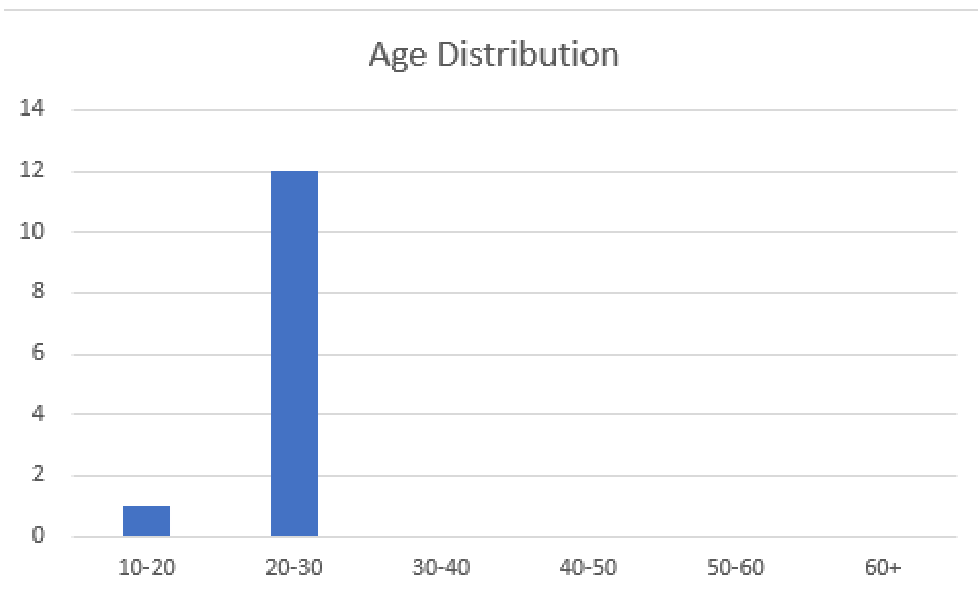

Figure 5.2 Q2, Age Distribution of the AIGC Survey

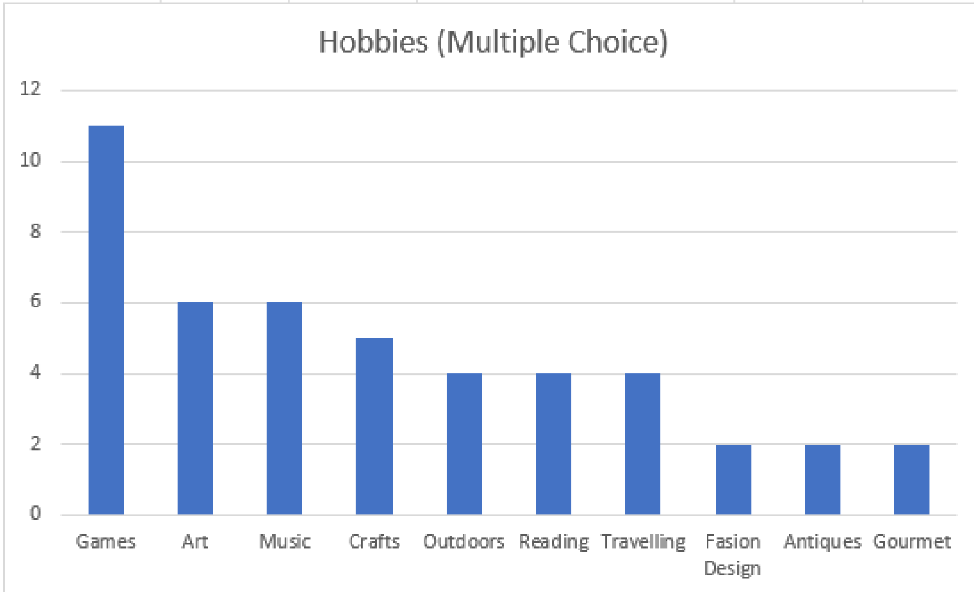

Figure 5.3 Q3, Hobbies Distribution of the AIGC Survey

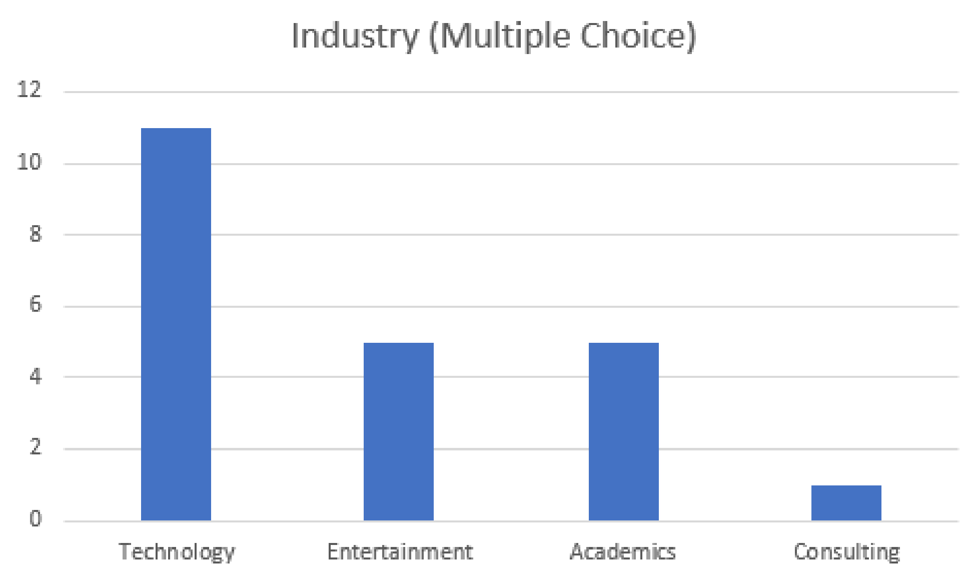

Figure 5.4 Q4, Industry Distribution of the AIGC Survey

At Q5-Q7, the first question of the second group where we collected participant's familiarity data about AIGC, we began by establishing a unified definition of digital content creation to set a common ground for discussion, avoid confusions, and reduce deviations errors originated from subjective interpretations. To achieve this, we first asked participants to rate their familiarity with the digital content creation industry and provide their definitions of "digital content creation." Subsequently, we provided a definition of digital content creation as "the process of producing or publishing various forms of digital content, including text, images, videos, and audio" and emphasized that even activities like "typing and formatting documents with Microsoft Word" can be considered forms of digital content creation. This approach was intentionally designed to boost participants' confidence in the subject matter, hoping them to respond with greater accuracy and conviction. As shown in Figure 5.1.4, participants' familiarity with digital content creation significantly increased from 3.39 to 4.39 after being provided with the standardized definition.

It is worthy to mention here that the end result of the study seemed to show that the confidence tricks worked as a trend can be drawn from the responses, even though our participants may come from different industries, answered different set of personalized questions, and received different images.

The inferred potentially interested 2D art styles were provided immediately after responding the second familiarity question. The choices seemed to be converged on several choices, as shown in Figure 5.1.6, even though 21 art styles occurred during the response.

Then, participants got 3 different art tasks, and they were asked to rate the familiarity with each of them. The system then picked the one with highest familiarity to ensure that the participant can be more connected to the generated art (as the generated art uses this task response for generation).

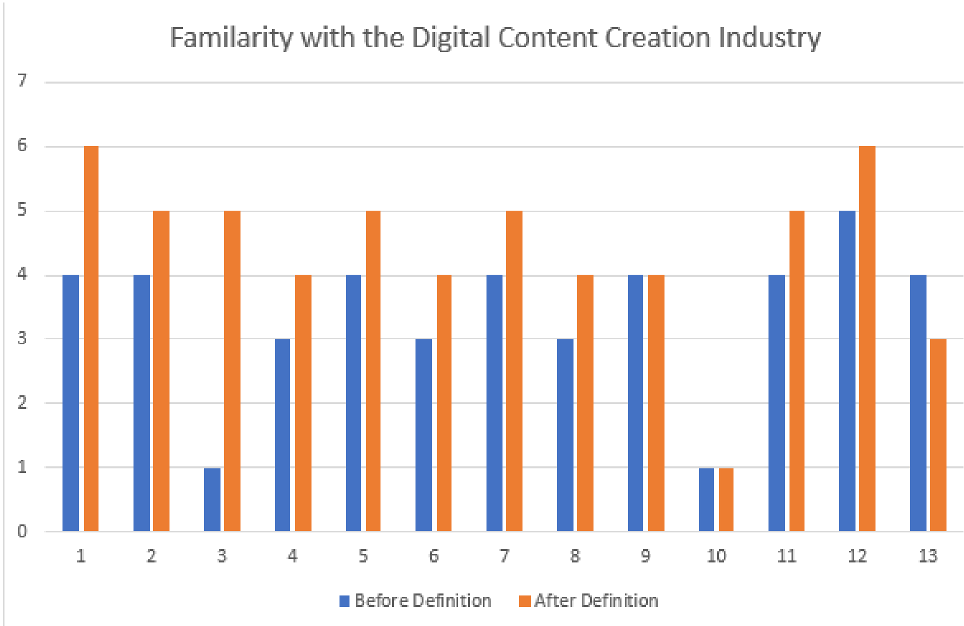

Figure 5.5 Familiarity with the Digital Content Creation Industry

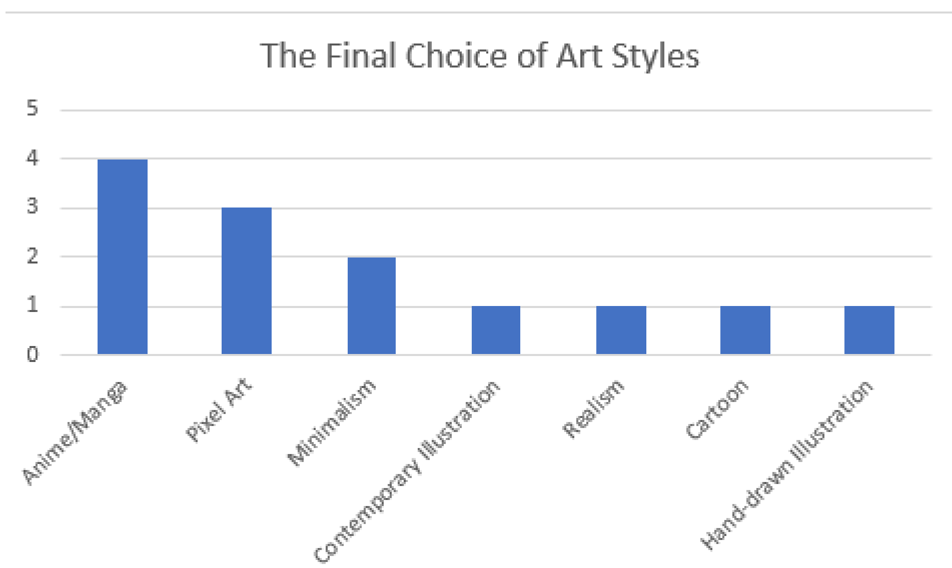

Figure 5.6 The Final Choice of Art Styles

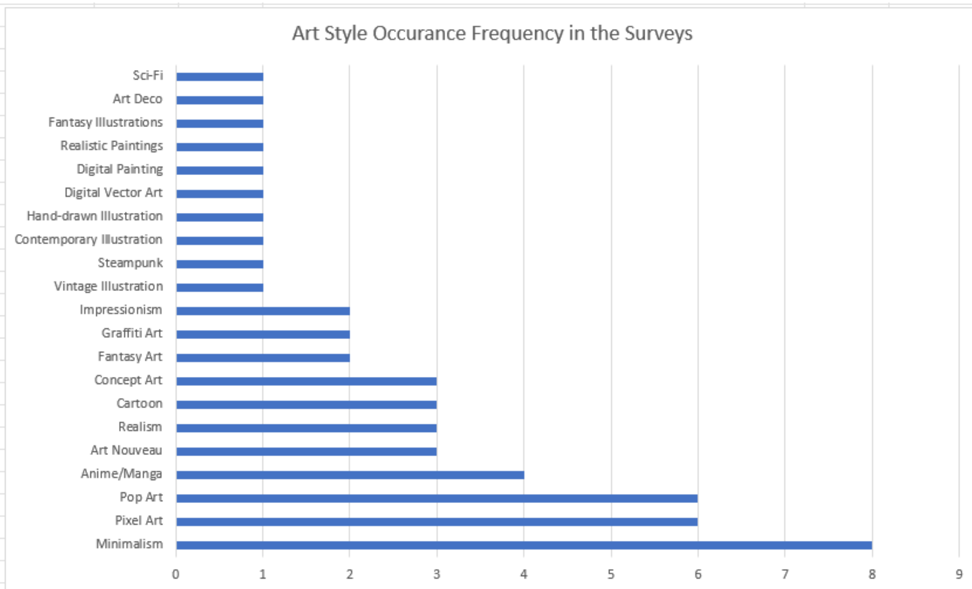

Figure 5.7 Art Style Occurrence Frequency in the Surveys

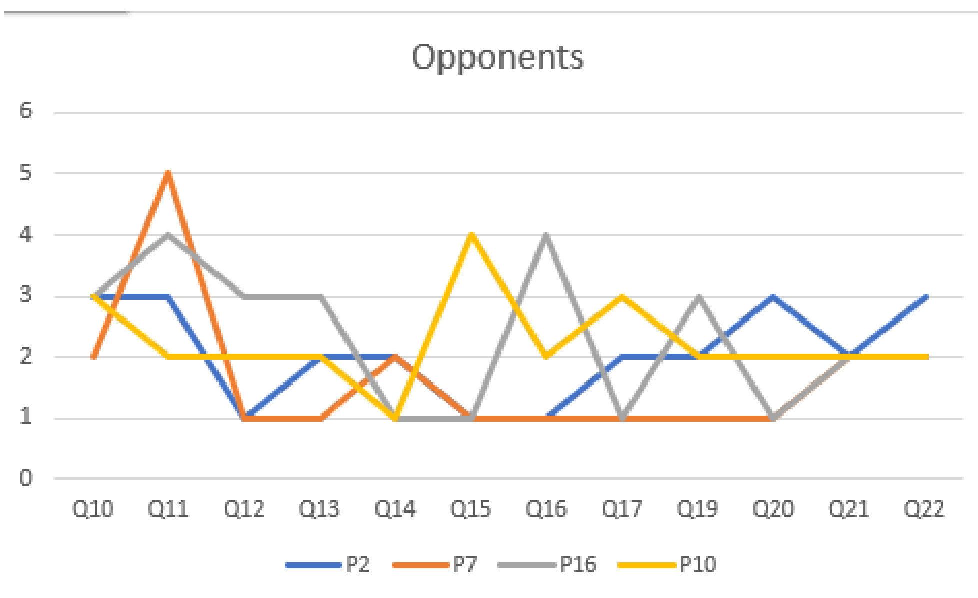

Figure 5.8 The AIGC Survey, Opponent Group

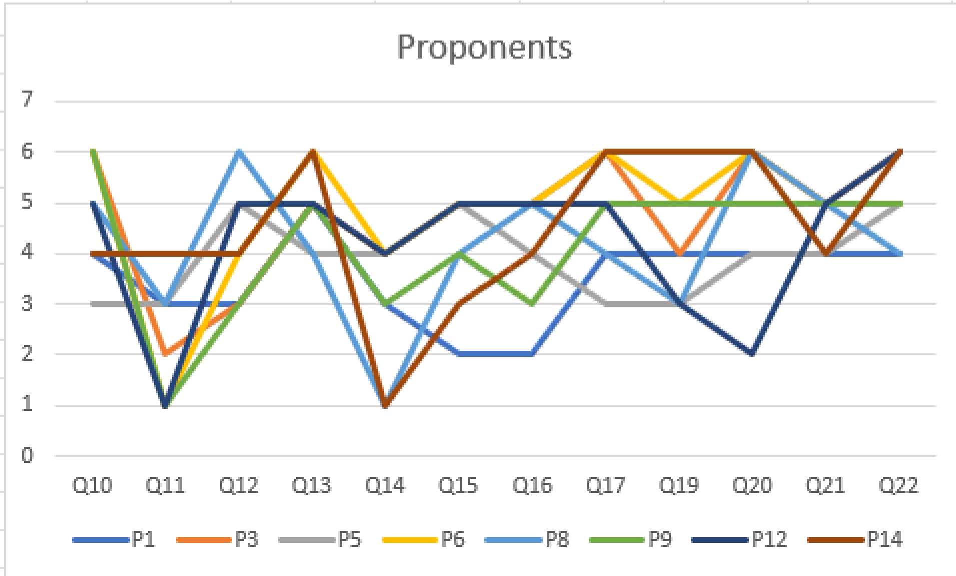

Figure 5.9 The AIGC Survey, Proponent Group

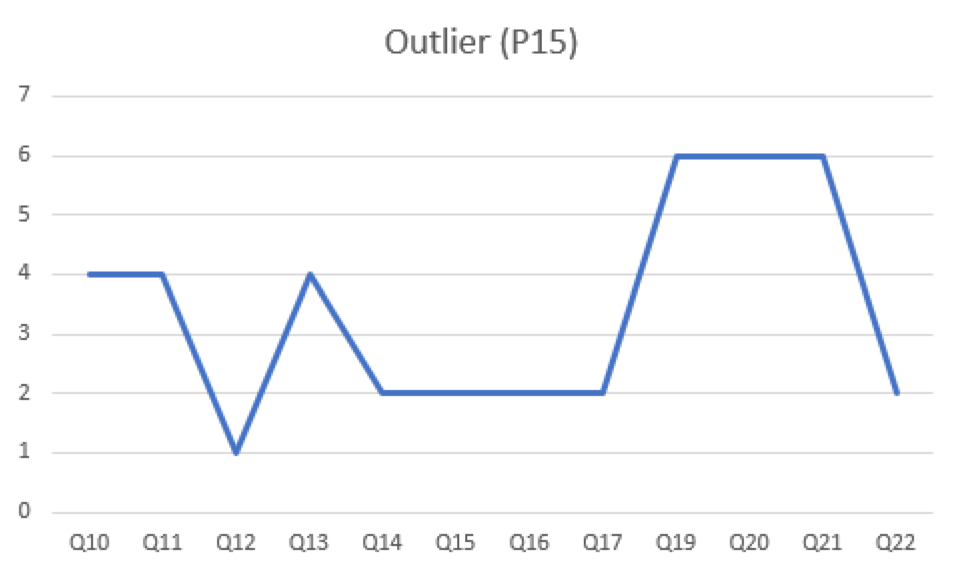

Figure 5.10 The AIGC Survey, Outlier Group

In the following 10 tasks (Q10-Q22), we used Likert scale to record participant's subjective rating of the complexity of the art work from Q9, expected work time, quality of several AI-generated art works, etc. In conclusion, we found:

1. The misguidance of saying that the generated art is found online does not guarantee a positive impact the rate of the quality of the work (Q15-Q16).

2. The proponent group generally thinks that the Q9 task will take a substantial amount of time for completion, while the opponent group generally thinks that the Q9 task would not require a significant of time (Q10 and Q11. For Q10, 3 indicates 10+ hours, 4 indicates 40 hours, 5 indicates a week, while 6 indicates a month. For Q11, 1 indicates impossible, while 6 indicates too easy)

3. The proponent group constantly rated the quality of a pre-generated art higher than the opponent group (Q12) and is willing to see more of such generated content in the future (Q13)

4. Some proponents think that the generated art only takes 1 hour for a human to complete, yet some proponents think that the generated art takes longer than 2 days. However, opponents constantly thought that the generated art can be done within hours by a human (Q14)

5. The proponent group constantly shared positive feedback (over 3) on accepting AIGC (Q 17, Q19, and Q20). Also, they all highly rated their experience with AIGC (Q21) and the belief that the AIGC can improve accessibility of digital content creation (Q22). On contrast, the opponent group shared negative feedback on accepting AIGC and does not believe that AIGC can improve accessibility. Interestingly, they are also rating them as not so experienced with AIGC.

6. When connecting the groups back to demographics, we found something interesting: the opponents are all interested in art, though some of them are IMGD majors, while some of them are the other majors. On the other hand, the technical people (non-IMGD majors) composed the majority of the proponent group, though there are 2 IMGD people inside the proponent group as well.

We also interviewed P10, one of the opponents, and P15, who was the outlier in our survey. P10 stated that he would certainly rate negatively about AIGC because he was seeing AI doing something he could only do after 10 years of learning in 10 hours within 10 minutes, which is frustrating. P15, on the other hand, seemed to be too experienced with AIGC and deemed that AIGC will still require high technical skills to use, which can be a future topic to research on.

5.1.2 Software Testing

We also performed software testing of the platform by inviting the participants who left their contacts in the survey. Unfortunately, due to the limited time frame and the sample size, as well as the time needed for a testing session (30 minutes+), we only got 3 SUS score responses. However, we interviewed all of them, collecting generally positive feedback.

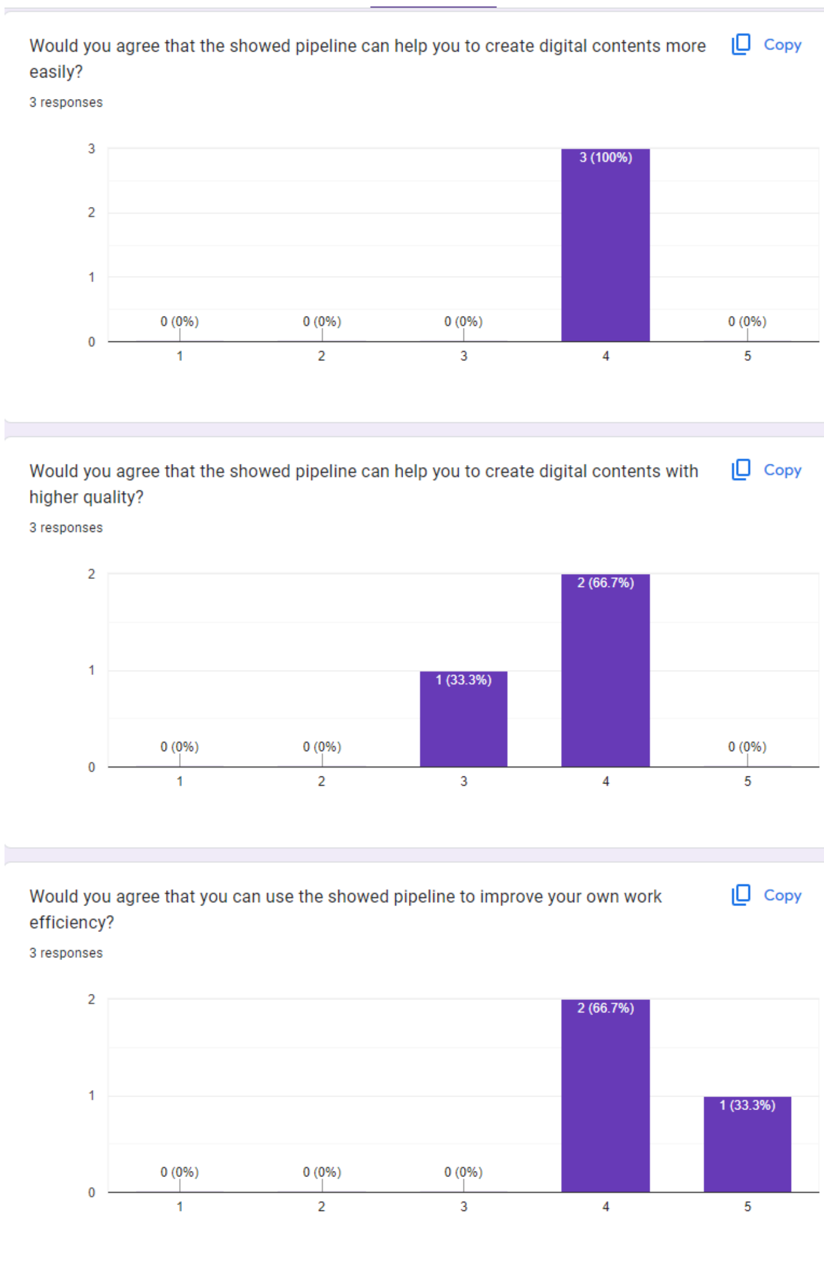

Figure 5.11 Post-Usage Survey

Participant	SUS Score
A	47.5
B	47.5
C	52.5

Table 5.1 SUS Score of the Professional Demo

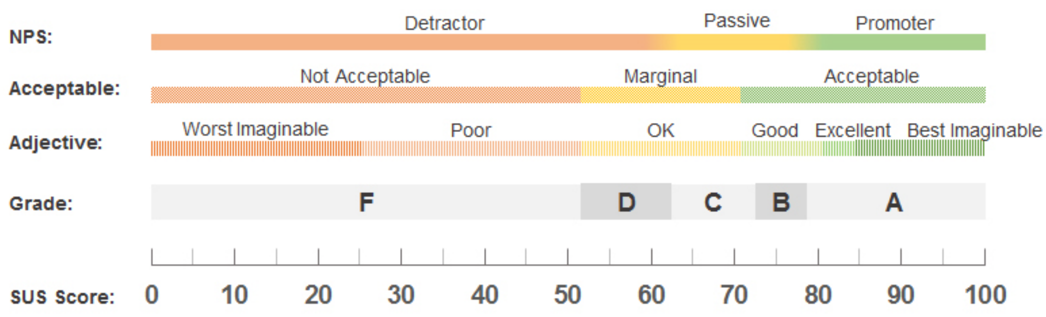

Figure 5.12 SUS Score Metrics [100]

With the SUS score, it seems that the professional platform exists at the edge of being accepted. We also conducted long discussions and interviews with the three participants, and their opinion is unanimous. They all said that they see the potential of using such platform, but it is still in its infancy state. They would like to try and use later versions if there is any. Additionally, they all believed that the project and the platform showed a cutting-edge solution towards content creation, and they agreed that the platform could help them and the public to create digital contents in easier manner.

They also talked about the complexity issue of the UI and the framework. Some of the instructions are not clear enough, and there were just too many functionalities that overwhelms users. However, this UI issues was not easily fixable due to the constraints from Gradio, and therefore we may need to switch to other frameworks in future development.

5.2 Casual

The first part of the survey includes seven questions that are designed to evaluate the degree to which the casual tool supported users in their creative work. These questions were specifically aimed at evaluating how the casual tool facilitated each of the six dimensions of the Creativity Support Index (CSI), as presented by [100], [101], namely exploration, collaboration, engagement, effort/reward tradeoff, expressiveness, and tool transportation. Users were asked to rate, on a scale of 1 to 7, how well they felt the casual tool supported their creative work in each dimension of the CSI. A rating of 1 indicated strong disagreement with the statement that the tool supported their creativity in that particular dimension, while a rating of 7 indicated strong agreement. It's important to note that the sample size for this user study was 13, which may limit the reliability and validity of the results, and caution should be exercised in interpreting the findings.

Although the small sample size may limit the generalizability of the findings beyond the specific sample or context of this study, it's worth noting that small sample sizes are not uncommon in certain research contexts, such as pilot studies, preliminary investigations, or studies involving rare populations. Despite this limitation, the findings from this study provide valuable insights into the perceptions of these 13 users regarding the casual tool's support for their creativity.

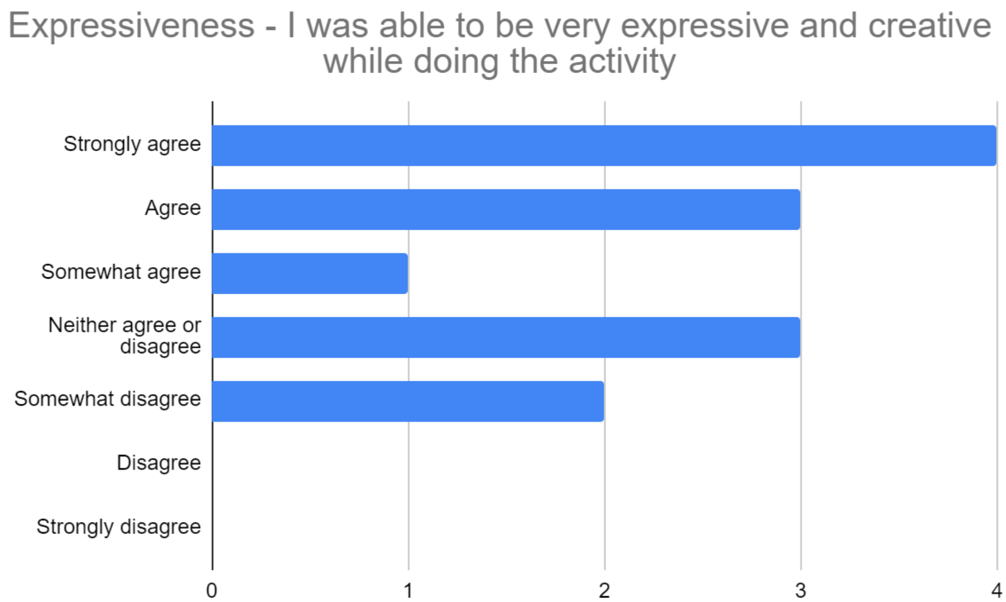

Figure 5.2.1 Users' Ability to be Expressive/Creative

The results show that 30.8% (4) of the users strongly agreed that they were able to be expressive/creative while using the casual tool, and another 23.1% (3) of the users agreed but with less strength. No users in the study disagreed strongly that the tool enabled them to be expressive.

Creativity Dimension	Average User Rating
Exploration	4.846
Collaboration	4.461
Engagement	4.692
Effort/Reward Tradeoff	4.538
Expressiveness	5.307
Tool Transportation	4.692

Table 5.2 Average User Rating for each Creativity Support Dimension

From the 13 users that tested the casual tool, the average user rating for each of the creativity support dimensions shows that users mostly felt that their creativity was slightly enabled by the tool. The highest average user rating was for the expressiveness dimension which means that users somewhat agreed that the casual tool supported their creativity by giving them the ability to express their creative work. This is especially notable because expressiveness was tied with exploration for the dimension of creativity that was rated the most important by users in the study.

The majority of the remaining questions of the survey were intended to gauge how successful the tool was at completing different goals related to the users' experience, such as being fun to use, having output consistent with its users' intent, and being an effective way to translate their thoughts into a 3d environment. The full list of questions and their associated answers can be seen in Figure 5.14.

Question	Average	Variance	Sparkline (1→7)
I enjoyed my time working with the tool	5	3.166666667	
I found the tool fun to use	5.230769231	3.025641026	
The tool's initial output was consistent with my intent	3.307692308	4.230769231	
Manual adjustments were easy to make	3.846153846	3.974358974	
The scenes created by the tool were enjoyable to look at	3.692307692	3.897435897	
I found the scenes created by the tool to be interesting	5.538461538	1.602564103	
Working with the tool helped me refine my vision for the scenes I was creating	3.384615385	3.923076923	
The tool was an effective way to translate my thoughts into a scene	3.615384615	4.423076923	
I felt that the tool understood my intent	3.384615385	3.256410256	
I felt like I was working with the tool, not against it	4.538461538	4.602564103	

Figure 5.13 Data from users' review of the tool meeting its goals or not. Average is based on a scale of 1-7, with higher being more agreement. The sparkline shows the number of responses from strong disagreement to strong agreement, left to right.

The user results for these questions were much more varied than with the creativity section. A summary can be seen in figure 6.3. In general, users reported that they had a fun time working with the tool, found the tool fun to use, and found its output interesting. In contrast, they reported that the tool did not meet its goals of being consistent with their intent, being easy to work with, or being especially effective at its job. This indicates that although the tool fulfills its purpose of being fun, it could be greatly improved in the functionality department.

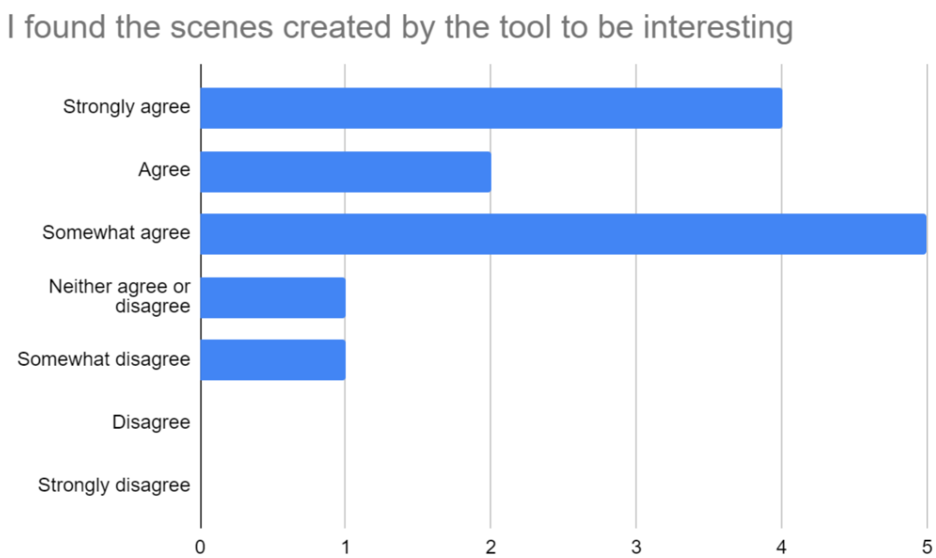

Figure 5.14 Bar Graph of Users' Interest in Scene Output

The goal of the scenes being interesting to look at had the highest user rating of being met. As shown in the figure above, most of the users somewhat agreed that the scenes created by the AI were interesting. There were 4 users that strongly agreed, and 2 others agreed that the AI generated scenes were interesting. This tells us that there is a chance that the users would use the casual tool again since they found it to be interesting. All together, these user study results indicate the successful creation of a compelling tool, but the room to improve it much further as well.

6. Conclusion and Future Work

Although the Piniverse project has come a long way, it's clear that there are opportunities that could be used to fix issues, improve features, and extend the capabilities of both products. This is a discussion of the most important areas that could realistically be used to improve the quality of the Piniverse products, and a conclusion about the state of the project in relation to those possibilities and the expectations created at the beginning of the project.

6.1 Professional

For the professional product, prominent examples of promising areas for future work include:

- Further integrating the AI-powered platform with open-source 3D Game Engines
- Enhancing implementation of semantic technologies in presentation and design layers
- Integrating distributed infrastructures
- Building a sharable knowledge base between private AI workers and humans

First, the professional product can significantly benefit from the ongoing integration with BabylonJS, an open-source 3D game engine. Successful integration will enable AI agents to create, update, and manage virtual entities and other AI workers within the game environment. This enhanced interaction will provide users with powerful visualization tools and automate the scheduling and coordination of tasks among AI workers. The visualization will allow users to better understand, customize, and enjoy the automation pipelines.

Second, there is a need to swap out the UI framework of the platform, as Gradio or Appsmith is too limited for AI to semantically support multimedia and multimodal content. Adopting semantic technologies to empower more comprehensive UI frameworks, like [60], will enable AI to better understand and manage a diverse range of digital content. Furthermore, integrating semantic technologies within the platform will provide better performance and availability compared to the existing GPT-based query solution.

Third, integrating distributed infrastructure will lead to improved scalability, collaboration, and service availability in the professional product. By adopting distributed technologies, protocols, and standards [78]–[80], the system will be able to seamlessly scale its services and resources, accommodate a growing number of users and AI workers, and provide a robust and reliable platform for users to interact with digital content. This improved infrastructure will also foster greater collaboration among users and AI agents, enhancing the overall user experience.

Fourth, a sharable knowledge base between private AI workers and humans will allow agents to access and utilize a vast repository of information, enabling more efficient and effective digital content creation and management. This shared knowledge base will facilitate the easier exchange of data and ideas between users and AI workers, fostering a more collaborative and innovative digital content ecosystem.

6.2 Casual

For the casual product, prominent examples of promising areas for future work include:

- Improving the preprocessing algorithm to allow for better object positioning and sizing
- Improving the 3d viewer to expand its editing capabilities
- Allowing the user to intercept and edit the scene graph
- Integrating additional 3d output types

To start with, improving the preprocessing algorithm to more reliably detect entity relationships in the input prompt would lead to more accurate output, which would help reduce a common source of the negative feedback about the output accuracy from the casual product user study. To improve the spatial relationships in the output, a wider variety of sentence structures could be parsed for relationships, most importantly those that involve verbs as the primary spatial relation, such as “a man sitting on a chair.” Parsing these verbs would be difficult, however, as the spatial relationship that they represent can vary depending on how the verb is used, including its tense and related nouns. Verbs that represent a state of being may be the same as those used to represent an action, so reliably parsing only those verbs that are being used to show the current state of the scene would require a way to distinguish between these two usages [99].

The other current problem with the preprocessing algorithm is determining entity sizes. We were not able to find a centralized database of object dimensions or even general sizes, as this is common sense to most humans. In addition, many objects vary considerably in size, although for the casual product it may be acceptable to randomly determine the size from within an acceptable range. The solution for this could be to incorporate user feedback to modify a database of known sizes, to employ the use of a third-party service that has not yet been discovered, or to train a prediction algorithm to provide size data. It should also be noted that sizes can be modified or restricted, such as a user asking for a model skyscraper or a car that is inside of an egg. Both of these situations change the size of the objects involved significantly from their default states.

Currently, the 3D viewer for the casual product can only be used to view the output, not to modify it. This limits the ability of the user to shape the output as they see fit, reducing their agency and preventing them from correcting perceived mistakes or improving upon the output of the product. Improving the interface to give the user the capability to move, scale, and rotate generated models would give the user more control over their creative works. Allowing them to regenerate models that have potential but were rendered in a way that differs from their vision would also be a useful feature. Allowing the user to render the scene using different methods, such as a longer but more polished render that uses full 3d models, or a very short render that only outputs bounding boxes and positions of objects before rendering them to the scene, would cover a wider variety of use-cases and pair well with the ability to manually move models. For example, rendering could be broken up so that bounding boxes are initially rendered when the frontend receives the scene graph, allowing the user to play around with the scene's layout as models automatically populate afterwards.

One of the more powerful features that could be added is the ability to download, upload, and edit the scene graph directly. Editing features could include adding or removing entities, editing the nouns, descriptors, and sizes associated with each entity, and adding, removing, or editing the spatial relationships between entities. This would allow the user to correct any mistakes made by the AI in parsing their input, as the scene graph is the single piece of information used to generate the 3d environment. The ability to download and upload the scene graph would also allow the user to use scene graphs from other sources to generate environments, use the scene graph generated by the casual product in other generation tools, and share the base information about the environment with others without generating models. In short, it gives the user a lot more control over the generation process with little overhead in the product.

6.3 Conclusion

All in all, Piniverse was an experiment to see how we could contribute to the ecosystem of artificial-intelligence-based tools in a way that helps lower the barrier of entry for digital content creation, expand the capabilities of existing creation tools by combining them in a modular manner, facilitate communication between users, and solve other problems that the technology is suited to help with. This resulted in two products: one to entertain casual users and allow them to bring their ideas to life with little overhead, and one to collaborate with professional users and help them define their ideas in more detail. Our prototype products for both groups saw success in their intended goals, but improvements to their reliability and other important secondary aspects remain an enticing possibility.

References

[1] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955,” *AI Magazine*, vol. 27, no. 4, Art. no. 4, Dec. 2006, doi: 10.1609/aimag.v27i4.1904.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical Text-Conditional Image Generation with CLIP Latents.” arXiv, Apr. 12, 2022. doi: 10.48550/arXiv.2204.06125.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models.” arXiv, Apr. 13, 2022. doi: 10.48550/arXiv.2112.10752.

[4] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, “Point-E: A System for Generating 3D Point Clouds from Complex Prompts.” arXiv, Dec. 16, 2022. doi: 10.48550/arXiv.2212.08751.

[5] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “DreamFusion: Text-to-3D using 2D Diffusion.” arXiv, Sep. 29, 2022. doi: 10.48550/arXiv.2209.14988.

[6] *I Made a Game with CHAT GPT!*, (Jan. 19, 2023). Accessed: Mar. 28, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=JFalS5pziPc

[7] A. Mayne, “Building games and apps entirely through natural language using OpenAI’s code-davinci model,” *@AndrewMayne*, Mar. 17, 2022. https://andrewmayneblog.wordpress.com/2022/03/17/building-games-and-apps-entirely-through-natural-language-using-openais-davinci-code-model/ (accessed Mar. 21, 2023).

[8] T. Karlsson, J. Brusk, and H. Engström, “Level Design Processes and Challenges: A Cross Section of Game Development,” *Games and Culture*, p. 15554120221139228, Nov. 2022, doi: 10.1177/15554120221139229.

[9] A. Dimitriadou, N. Djafarova, O. Turetken, M. Verkuyl, and A. Ferworn, “Challenges in Serious Game Design and Development: Educators’ Experiences,” *Simulation & Gaming*, vol. 52, no. 2, pp. 132–152, Apr. 2021, doi: 10.1177/1046878120944197.

[10] N. Giri and E. Stolterman, “An Exploratory Study on Game Developer Needs and Game Studio Challenges,” in *HCI International 2022 - Late Breaking Papers. Interaction in New Media, Learning and Games*, G. Meiselwitz, A. Moallem, P. Zaphiris, A. Ioannou, R. A. Sottilare, J. Schwarz, and X. Fang, Eds., in Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2022, pp. 581–591. doi: 10.1007/978-3-031-22131-6_43.

[11] J. Tacchi, “Open content creation: The issues of voice and the challenges of listening,” *New Media & Society*, vol. 14, no. 4, pp. 652–668, Jun. 2012, doi: 10.1177/1461444811422431.

[12] “Main challenges in Software Development | Apiumhub,” Apr. 08, 2020. https://apiumhub.com/tech-blog-barcelona/challenges-software-development/ (accessed Mar. 22, 2023).

[13] “10 Most Common Software Development Challenges.” https://www.orientsoftware.com/blog/software-development-challenges/ (accessed Mar. 22, 2023).

[14] S. Y. El-Mashad and E.-H. S. Hamed, “Automatic creation of a 3D cartoon from natural language story,” *Ain Shams Engineering Journal*, vol. 13, no. 3, p. 101641, May 2022, doi: 10.1016/j.asej.2021.11.010.

[15] R. M. Sindhu, L. S. P. Annabel, and G. Monisha, “Development of a 2D Game using Artificial Intelligence in Unity,” in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2022, pp. 1031–1037. doi: 10.1109/ICOEI53556.2022.9776750.

[16] P. Migdał, B. Olechno, and B. Podgórski, “Level generation and style enhancement -- deep learning for game development overview.” arXiv, Jul. 15, 2021. doi: 10.48550/arXiv.2107.07397.

[17] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, “Zero-Shot Text-Guided Object Generation with Dream Fields.” arXiv, May 04, 2022. doi: 10.48550/arXiv.2112.01455.

[18] Y. Cao *et al.*, “A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT.” arXiv, Mar. 07, 2023. doi: 10.48550/arXiv.2303.04226.

[19] C. Zhang *et al.*, “A Complete Survey on Generative AI (AIGC): Is ChatGPT from GPT-4 to GPT-5 All You Need?” arXiv, Mar. 21, 2023. doi: 10.48550/arXiv.2303.11717.

[20] F. Chen and W. Ren, “On the Control of Multi-Agent Systems: A Survey,” *SYS*, vol. 6, no. 4, pp. 339–499, Jul. 2019, doi: 10.1561/2600000019.

[21] C. Marín-Lora, M. Chover, J. M. Sotoca, and L. A. García, “A game engine to make games as multi-agent systems,” *Advances in Engineering Software*, vol. 140, p. 102732, Feb. 2020, doi: 10.1016/j.advengsoft.2019.102732.

[22] “Auto-GPT: An Autonomous GPT-4 Experiment.” Significant Gravitas, Apr. 26, 2023. Accessed: Apr. 26, 2023. [Online]. Available: https://github.com/Significant-Gravitas/Auto-GPT

[23] “AgentGPT: Autonomous AI in your browser 🤖.” https://agentgpt.reworkd.ai/ (accessed Apr. 26, 2023).

[24] A. BinSubaih, S. Maddock, and D. Romano, “A Survey of ‘Game’ Portability,” Dec. 2015. [Online]. Available: http://staffwww.dcs.shef.ac.uk/people/S.Maddock/publications/BinSubaihEtal2007_CS0705.pdf

[25] U. Rafiq, C. Filippo, and X. Wang, “Understanding Low-Code or No-Code Adoption in Software Startups: Preliminary Results from a Comparative Case Study,” in *Product-Focused Software Process Improvement*, D. Taibi, M. Kuhrmann, T. Mikkonen, J. Klünder, and P. Abrahamsson, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 390–398. doi: 10.1007/978-3-031-21388-5_27.

[26] M. Fryling, “Low code app development,” *J. Comput. Sci. Coll.*, vol. 34, no. 6, p. 119, Apr. 2019.

[27] A. C. Bock and U. Frank, "In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms," in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Oct. 2021, pp. 57–66. doi: 10.1109/MODELS-C53483.2021.00016.

[28] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[29] P. Hitzler, "A review of the semantic web field," *Commun. ACM*, vol. 64, no. 2, pp. 76–83, Jan. 2021, doi: 10.1145/3397512.

[30] "W3C Standards - Semantic Web." https://www.w3.org/standards/semanticweb/ (accessed Mar. 30, 2023).

[31] "The History of the Semantic Web is the Future of Intelligent Assistants," Aug. 19, 2016. https://www.explainablestartup.com/2016/08/the-history-of-semantic-web-is-the-future-of-intelligent-assistants.html (accessed Mar. 26, 2023).

[32] P. Hitzler, K. Janowicz, and A. Hogan, "The Semantic Web: Two decades on," *Semant. web*, vol. 11, no. 1, pp. 169–185, Jan. 2020, doi: 10.3233/SW-190387.

[33] Otaku, "RDF-RPG." Aug. 14, 2021. Accessed: Mar. 29, 2023. [Online]. Available: https://github.com/mschroeder-github/rdf-rpg

[34] B. C. Bridger and C. S. Groskopf, "Fundamentals of artificial intelligence in game development," in *Proceedings of the 38th annual on Southeast regional conference*, Clemson South Carolina: ACM, Apr. 2000, pp. 51–55. doi: 10.1145/1127716.1127727.

[35] K. J. L. Nevelsteen, "Virtual World, Defined from a Technological Perspective, and Applied to Video Games, Mixed Reality and the Metaverse," *Comput Anim Virtual Worlds*, vol. 29, no. 1, p. e1752, Jan. 2018, doi: 10.1002/cav.1752.

[36] M. Vellon, K. Marple, D. Mitchell, and S. Drucker, "The Architecture of a Distributed Virtual Worlds System," presented at the 4th Conference on Object-Oriented Technologies and

Systems (COOTS 98), 1998. Accessed: Mar. 26, 2023. [Online]. Available: https://www.usenix.org/conference/coots-98/architecture-distributed-virtual-worlds-system

[37] S. Sloane, *Digital fictions: storytelling in a material world*. in New directions in computers & composition studies. Stamford, Conn: Ablex Pub, 2000.

[38] C. G. Jung, J. Lind, C. Gerber, M. Schillo, P. Funk, and A. Burt, "An architecture for co-habited virtual worlds," *SIMULATION SERIES*, vol. 31, pp. 197–202, 1999.

[39] J. Gregory, *Game engine architecture*, Third edition. Boca Raton: Taylor and Francis, CRC Press, 2018.

[40] D. H. Eberly, *3D game engine design: a practical approach to real-time computer graphics*, 2nd ed. Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2007.

[41] C. Politowski, F. Petrillo, J. E. Montandon, M. T. Valente, and Y.-G. Guéhéneuc, "Are game engines software frameworks? A three-perspective study," *Journal of Systems and Software*, vol. 171, p. 110846, Jan. 2021, doi: 10.1016/j.jss.2020.110846.

[42] D. Cvetković, *Simulation and Gaming*. BoD – Books on Demand, 2018.

[43] H. Miller, "Epic Games' Unreal Engine takes movie production to the next level," *Business North Carolina*, Feb. 04, 2020. https://bnc.wxp.io/epic-games-unreal-engine-takes-movie-production-to-the-next-level/ (accessed Apr. 26, 2023).

[44] S. S. Newsletter, "Is the Unreal Engine the future of the economy?," *Sunday Special Newsletter*, Jun. 06, 2021. https://sundayspecial.substack.com/p/is-the-unreal-engine-the-future-of (accessed Apr. 26, 2023).

[45] S. Moroz, "The prospect of applying a real-time gaming engine in filmmaking: A short investigation," Thesis, 2021. Accessed: Apr. 26, 2023. [Online]. Available: https://repin.pjwstk.edu.pl/xmlui/handle/186319/1131

[46] "Unreal Engine Powers Film & Television Production," *Unreal Engine*. https://www.unrealengine.com/en-US/solutions/film-television (accessed Apr. 26, 2023).

[47] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles,” in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds., in Springer Proceedings in Advanced Robotics. Cham: Springer International Publishing, 2018, pp. 621–635. doi: 10.1007/978-3-319-67361-5_40.

[48] “The Role of Physics-Based Simulators in Robotics | Annual Review of Control, Robotics, and Autonomous Systems.” https://www.annualreviews.org/doi/abs/10.1146/annurev-control-072220-093055 (accessed Apr. 26, 2023).

[49] T. C. Chieu, L. Zeng, and A. Mohindra, “An extensible enterprise content management system with Service Component Architecture,” in *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, Oct. 2008, pp. 1131–1137. doi: 10.1109/SOLI.2008.4686569.

[50] “What Is a Content Management System (CMS)?,” *Kinsta®*, Mar. 20, 2023. https://kinsta.com/knowledgebase/content-management-system/ (accessed Apr. 26, 2023).

[51] C. Maican and R. Lixandroiu, “A system architecture based on open source enterprise content management systems for supporting educational institutions,” *International Journal of Information Management*, vol. 36, no. 2, pp. 207–214, Apr. 2016, doi: 10.1016/j.ijinfomgt.2015.11.003.

[52] “appsmithorg/appsmith.” Appsmith, Apr. 26, 2023. Accessed: Apr. 26, 2023. [Online]. Available: https://github.com/appsmithorg/appsmith

[53] M. K. Bekele, R. Pierdicca, E. Frontoni, E. S. Malinverni, and J. Gain, “A Survey of Augmented, Virtual, and Mixed Reality for Cultural Heritage,” *J. Comput. Cult. Herit.*, vol. 11, no. 2, pp. 1–36, Jun. 2018, doi: 10.1145/3145534.

[54] O.-I. Holthe, “A Multimedia Presentation Framework for Web 3.0 Computer Game Experiences,” *IEEE MultiMedia*, vol. 17, no. 4, pp. 70–79, Oct. 2010, doi: 10.1109/MMUL.2010.71.

[55] B. Meixner, “Hypervideos and Interactive Multimedia Presentations,” *ACM Comput. Surv.*, vol. 50, no. 1, p. 9:1-9:34, Mar. 2017, doi: 10.1145/3038925.

[56] D. Duke, L. Hardman, A. Hauptmann, D. Paulus, and S. Staab, Eds., *Semantic Multimedia: Third International Conference on Semantic and Digital Media Technologies, SAMT 2008, Koblenz, Germany, December 3-5, 2008. Proceedings*, vol. 5392. in Lecture Notes in Computer Science, vol. 5392. Berlin, Heidelberg: Springer, 2008. doi: 10.1007/978-3-540-92235-3.

[57] R. Troncy, B. Huet, and S. Schenk, *Multimedia Semantics: Metadata, Analysis and Interaction*. John Wiley & Sons, 2011.

[58] M. S. Hossain and A. El Saddik, “Multimedia Content Repurposing for Heterogeneous Wireless Clients,” in *2007 IEEE International Conference on Signal Processing and Communications*, Nov. 2007, pp. 1447–1450. doi: 10.1109/ICSPC.2007.4728602.

[59] M. S. Hossain, M. A. Rahman, and A. El Saddik, “A framework for repurposing multimedia content,” in *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, May 2004, pp. 971-974 Vol.2. doi: 10.1109/CCECE.2004.1345278.

[60] “lowdefy/lowdefy.” Lowdefy, Apr. 26, 2023. Accessed: Apr. 26, 2023. [Online]. Available: https://github.com/lowdefy/lowdefy

[61] “APIJSON.” Tencent, Apr. 26, 2023. Accessed: Apr. 26, 2023. [Online]. Available: https://github.com/Tencent/APIJSON

[62] “baidu/amis.” Baidu, Apr. 27, 2023. Accessed: Apr. 26, 2023. [Online]. Available: https://github.com/baidu/amis

[63] R. Zhao, K. Wang, R. Divekar, R. Rouhani, H. Su, and Q. Ji, “An Immersive System with Multi-Modal Human-Computer Interaction,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, May 2018, pp. 517–524. doi: 10.1109/FG.2018.00083.

[64] H. Khallouki, M. Bahaj, P. Roose, and S. Laborie, “SMPMA: Semantic multimodal profile for multimedia documents adaptation,” in *2014 5th Workshop on Codes, Cryptography and Communication Systems (WCCCS)*, Nov. 2014, pp. 142–147. doi: 10.1109/WCCCS.2014.7107908.

[65] F. Spiess and H. Schuldt, “Multimodal Interactive Lifelog Retrieval with vitrivr-VR,” in *Proceedings of the 5th Annual on Lifelog Search Challenge*, in LSC ’22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 38–42. doi: 10.1145/3512729.3533008.

[66] “A Survey on Immersive Visualization,” *JCADC*, pp. 497–507, doi: 10.3724/SP.J.1089.2021.18809.

[67] S. N. T. Chiueh and S. Brook, “A survey on virtualization technologies,” *Rpe Report*, vol. 142, 2005.

[68] R. Di Pietro and F. Lombardi, “Virtualization Technologies and Cloud Security: advantages, issues, and perspectives,” *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday*, pp. 166–185, 2018.

[69] “Chapter 1. Introducing virtualization in RHEL Red Hat Enterprise Linux 8 | Red Hat Customer Portal.” https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_virtualization/introducing-virtualization-in-rhel_configuring-and-managing-virtualization (accessed Apr. 26, 2023).

[70] V. G. da Silva, M. Kirikova, and G. Alksnis, “Containers for Virtualization: An Overview,” *Applied Computer Systems*, vol. 23, no. 1, pp. 21–27, Apr. 2018, doi: 10.2478/acss-2018-0003.

[71] “Emerging Trends, Techniques and Open Issues of Containerization: A Review | IEEE Journals & Magazine | IEEE Xplore.” https://ieeexplore.ieee.org/abstract/document/8861307 (accessed Apr. 26, 2023).

[72] “Mobile Edge Computing Platform with Container-Based Virtualization Technology for IoT Applications | SpringerLink.” https://link.springer.com/article/10.1007/s11277-018-5856-5 (accessed Apr. 26, 2023).

[73] “Containerization technologies: taxonomies, applications and challenges | SpringerLink.” https://link.springer.com/article/10.1007/s11227-021-03914-1 (accessed Apr. 26, 2023).

[74] “Docker: Accelerated, Containerized Application Development,” May 10, 2022. https://www.docker.com/ (accessed Apr. 26, 2023).

[75] “Production-Grade Container Orchestration,” *Kubernetes*. https://kubernetes.io/ (accessed Apr. 26, 2023).

[76] “Amazon EC2,” *Amazon Web Services, Inc.* https://aws.amazon.com/pm/ec2/ (accessed Apr. 26, 2023).

[77] “Rent Cloud GPUs from $0.2/hour.” https://www.runpod.io/ (accessed Apr. 26, 2023).

[78] F. Beetz and S. Harrer, “GitOps: The Evolution of DevOps?,” *IEEE Software*, vol. 39, no. 4, pp. 70–75, Jul. 2022, doi: 10.1109/MS.2021.3119106.

[79] T. A. Limoncelli, “GitOps: a path to more self-service IT,” *Commun. ACM*, vol. 61, no. 9, pp. 38–42, Aug. 2018, doi: 10.1145/3233241.

[80] R. López-Viana, J. Díaz, and J. E. Pérez, “Continuous Deployment in IoT Edge Computing : A GitOps implementation,” in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, Jun. 2022, pp. 1–6. doi: 10.23919/CISTI54924.2022.9820108.

[81] S. Bhattacharya and L. Bhattacharya, *XAAS: EVERYTHING-AS-A-SERVICE: The Lean and Agile Approach to Business Growth*. World Scientific, 2022.

[82] V. N. S. S. Chimakurthi, “Strategic Growth of Everything-as-a-Service (XaaS) Business Model Transformation,” *Engineering International*, vol. 9, no. 2, pp. 129–140, 2021.

[83] M. Classen, C. Blum, P. Osterrieder, and T. Friedli, “Everything as a service? Introducing the St. Gallen IGaaS management model,” 2019.

[84] “CUDA Toolkit - Free Tools and Training,” *NVIDIA Developer*, Jul. 02, 2013. https://developer.nvidia.com/cuda-toolkit (accessed Apr. 27, 2023).

[85] “NVIDIA NGC | NVIDIA.” https://www.nvidia.com/en-us/gpu-cloud/ (accessed Apr. 27, 2023).

[86] “Game Design Document: Is It Worth the Effort?,” *Innovecs Games*. https://www.innovecsgames.com/blog/game-design-document/ (accessed Apr. 27, 2023).

[87] P. O’Connor, *The Craft and Science of Game Design: A Video Game Designer’s Manual*. CRC Press, 2020.

[88] M.-U.-S. Fahme and T. H. Khan, “It’s All in My Head: Writing a Game Design Document,” in *How to Make a Game: Go From Idea to Publication Avoiding the Common Pitfalls Along the Way*, M.-U.-S. Fahme and T. H. Khan, Eds., Berkeley, CA: Apress, 2021, pp. 41–62. doi: 10.1007/978-1-4842-6917-6_5.

[89] “MVC - MDN Web Docs Glossary: Definitions of Web-related terms | MDN,” Feb. 21, 2023. https://developer.mozilla.org/en-US/docs/Glossary/MVC (accessed Apr. 27, 2023).

[90] U. Technologies, “Unity - Manual: Structure UI with UXML.” https://docs.unity3d.com/Manual/UIE-UXML.html (accessed Apr. 27, 2023).

[91] H. Liu *et al.*, “CFS: A Distributed File System for Large Scale Container Platforms,” in *Proceedings of the 2019 International Conference on Management of Data*, in SIGMOD ’19. New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 1729–1742. doi: 10.1145/3299869.3314046.

[92] F. Christ and B. Nagel, *A reference architecture for semantic content management systems*. Gesellschaft für Informatik e.V., 2011. Accessed: Apr. 27, 2023. [Online]. Available: http://dl.gi.de/handle/20.500.12116/18506

[93] V. Damjanovic-Behrendt, T. Kurz, R. Westenthaler, W. Behrendt, A. Gruber, and S. Schaffert, “Semantic Enhancement: The Key to Massive and Heterogeneous Data Pools,” Apr. 2023.

[94] A. Gangemi, “Ontology Design Patterns for Semantic Web Content,” in *The Semantic Web – ISWC 2005*, Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 262–276. doi: 10.1007/11574620_21.

[95] A. Koubaa, “GPT-4 vs. GPT-3.5: A Concise Showdown.” Preprints, Mar. 24, 2023. doi: 10.20944/preprints202303.0422.v1.

[99] “Types of Verbs,” *Utah Valley University*. https://www.uvu.edu/writingcenter/docs/handouts/grammar-usage/typesofverbs.pdf (accessed Apr. 27, 2023).

[100] “Craiyon V3 - Free online AI image generator from text,” https://www.craiyon.com/ (accessed Apr. 27, 2023).

[101] “Midjourney,” https://midjourney.com/ (accessed Apr. 27, 2023).

[102] “About AllenNLP,” https://allenai.org/allennlp/ (accessed Apr. 27, 2023).

[103] “Industrial-Strength Natural Language Processing,” https://spacy.io/ (accessed Apr. 27, 2023).

[104] “Point-E demo: text to 3D,” https://huggingface.co/spaces/openai/point-e/ (accessed Apr. 27, 2023).