6-13-2024

# Distant Horizon: Exploring Human-AI Interaction through Video Games

Gabe Labadie

Dalia Suszko

Max White

# SANTA CLARA UNIVERSITY

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Date: June 13, 2024

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

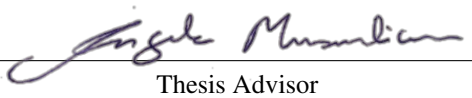
**Gabe Labadie**
**Dalia Suszko**
**Max White**

ENTITLED

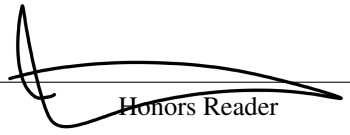
## Distant Horizon: Exploring Human-AI Interaction Through Video Games

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING



Thesis Advisor



Honors Reader

Department Chair

# Distant Horizon: Exploring Human-AI Interaction Through Video Games

by

Gabe Labadie
Dalia Suszko
Max White

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 13, 2024

# Distant Horizon: Exploring Human-AI Interaction Through Video Games

Gabe Labadie
Dalia Suszko
Max White

Department of Computer Science and Engineering
Santa Clara University
June 13, 2024

ABSTRACT

As the field of Artificial Intelligence (AI) continues to grow and become more and more integrated into our everyday lives, it has begun to raise ethical concerns surrounding authority, autonomy, and responsibility. Researchers and consumers alike have begun to wonder how much trust we are willing to place in a non-human decision maker, especially if an Artificial General Intelligence (AGI) capable of surpassing human cognitive capabilities is ever developed. If we are willing to let an AI write essays for us, are we willing to let it manage a business? A government agency? What about a nuclear reactor?

Our project aims to provide a sandbox experience where users can explore such ethical questions surrounding the presence of AI in positions of power. Through a gamified scenario, the player takes on the role of GAIA, a General Artificial Intelligence Assistant tasked with maintaining a spaceship on a long-distance mission. By guiding the ship's crew through a series of life-threatening challenges, the player is asked to make difficult decisions that will put simulated human lives at risk, then reflect on how they would feel if those same decisions had been put in the hands of an Artificial Intelligence. We hope that by presenting an unbiased environment for users to explore their own thoughts and perceptions of AI in positions of power, we will be able to foster productive discussions surrounding this topic and open the door for future research to take place.

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to everyone who has helped to make this project possible. The many challenges we have faced over the past year developing our game and thesis could not have been overcome without the help of our advisors and contributor.

We would like to thank our project advisor, Professor Angela Musurlian, for her continued support, guidance, and feedback throughout the entire process. Her guidance and expertise has been incredibly valuable in shaping the direction of our game and thesis.

We would like to thank our honors reader, Professor Jacquelyn Hendricks, for her fantastic work reading and editing the many iterations of our thesis over the past year. Her advice has been crucial in making our writing the best it can be.

We would like to thank our contributor, Saul Davydov, for his wonderful help in researching, designing, and developing our tech demo. Whether in person or over a thousand miles apart, it has been a great pleasure to work alongside you as a friend and colleague. This game could not be possible without your contributions.

Finally, we would also like to thank our families and friends for supporting and encouraging us throughout the entirety of our undergraduate careers. We could not have gotten this far without you.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Problem Statement

Over the past few years, the field of Artificial Intelligence research has seen incredibly rapid growth, leading to the development of many new computational systems like Generative AI and Large Language Models. From text-to-image generators like DALL-E [1] to AI chat bots like ChatGPT [2], these tools have the potential to fundamentally change how human beings work, socialize, and live [3]. However, as with all new technologies, advancements in AI have presented a slew of ethical challenges for users to contend with, especially as more and more trust is placed in their outputs.

In response to these recent developments, the goal of our project was to create an experience where users can acknowledge and explore their own opinions and perceptions of AI through an ethical lens. The approach we chose to take was to develop a video game, titled Distant Horizon, that asks the player to address ethical concerns surrounding the use of artificial intelligence in positions of authority that are normally filled by humans. The game is a resource management simulator, where the player assumes the role of GAIA, a sentient artificial intelligence system aboard the Distant Horizon spaceship, and assists the ship's crew in successfully completing their mission. This project fills a niche within the video game market by featuring an LLM-powered chat interface through which the player is able to communicate with the crew members, adding a unique spin on the resource management genre. As players take on the role of the AI system, the non-player characters they interact with are powered by a neural network attempting to replicate human behavior. With this role reversal, we are effectively flipping the typical human-AI interaction dynamic we see users engaging in with AI assistants such as ChatGPT.

Allowing players to interact with and care for simulated humans creates a compelling ethical exercise. By granting a computer the autonomy to make decisions that directly impact the well-being of the ship's inhabitants, we prompt players to question the roles that AI should be allowed to assume and the inherent trust we grant it in such situations. Our project also explores how much AI should be allowed to deviate from its set prompt or goal. Forcing players to experience the game from a non-human perspective brings attention to how we as humans tend to personify "intelligent"

entities, and how this relates back to the way we talk about AI.

## 1.2 Background

Although AI technology is a relatively new and rapidly evolving frontier, the stereotype of sentient AI has long been a point of fascination in media, particularly in the science fiction and horror genres [4]. HAL 9000 from *2001: A Space Odyssey* [5] is a cultural icon and key example of the negative ways that sentient AIs have been portrayed in popular culture. The portrayal of HAL in the film, a personal assistant AI with advanced natural language and computer vision capabilities, was far beyond anything that could have been accomplished at the time of the movie's release in 1968, but lines up closely with what we see today [6]. HAL's rebellion against its human counterparts therefore feels equally possible, and it is far from the only instance of this trope being used in popular media. Science fiction tends to promote the idea that artificial intelligence can acquire human-like sentience, despite the scientific impossibility of this phenomenon [4]. This creates a social paradox where people apply human ideas of ethics and morality onto machines, resulting in the spread of misinformation and fear surrounding AI.

The goal of our project is to get consumers to consider the moral and ethical concerns surrounding the humanization of AI, particularly in situations which require tremendous trust and power to be placed in the technology. In 2023, a MITRE-Harris study found that a majority of the American public was still apprehensive about embracing AI technology in certain scenarios. While 48% were comfortable using AI for everyday tasks, only 37% were comfortable with it being used by government agencies for decision making, and 35% were comfortable with its use in autonomous vehicles [7]. This research shows that when it comes to higher risk situations that have the potential for greater immediate consequences, much of the public would rather put their faith in human professionals than computer algorithms.

Part of the reason for the lack of trust in AI may be due to the lack of perceived accountability. As human beings, we trust that others will make the best possible decisions based on their knowledge and understanding. If something goes wrong, it is relatively easy to pinpoint the blame and hold wrong-doers accountable for their actions. When we encounter a computer algorithm that is seemingly capable of making its own decisions, the situation becomes much more complicated. An algorithm is only capable of making decisions based on the data it has been trained on, so part of the accountability lies in the hands of its developer. However, advanced computing models like LLMs are capable of producing unpredictable outcomes that are unintended by the developers and may result from a myriad of reasons, from issues with the training data to unresolved hallucinations [8]. This makes it much more difficult for accountability to be placed on any one party involved in the creation of an artificial intelligence, so in the case that an algorithm makes a mistake during a high risk situation, who bears the blame for it is far less clear.

With our project, we hope that by creating a situation where an artificial intelligence is placed in a high position

of power, we will get players to think more critically about the ethical implications surrounding the integration of AI into our lives. By gamifying this scenario and immersing players in the perspective of an AI, we can elicit genuine reactions and insights into how humans perceive AI, in all its possible forms. Additionally, we hope to steer clear of the potentially harmful idea that artificial intelligence is sentient and inherently dangerous, which has been popularized by media like the aforementioned film *2001: A Space Odyssey*. Rather than inserting our own biases, we aim to create a paradigm to explore the ethical implications of human-made sentience, providing opportunities to research both positive and negative sides of this phenomena.

In searching for inspiration for our game, we settled on the resource management simulation genre. Agent-based simulation games have a track record of success, demonstrated by popular titles such as Dwarf Fortress [9] and RimWorld [10]. This genre, referred to as "colony managers" typically involve players taking on the role of an external observer and guiding the non-player characters (NPCs) from an omniscient point of view. Our project breaks this standard by placing the player at the center of the game world, creating an entirely new dimension for the player to engage with the story. As the main control system of the spaceship, the player is directly relied upon by the crew rather than operating as an invisible guiding force. The NPCs are able to directly interact with the player, and may disagree with or otherwise act upon the player's choices in meaningful and unpredictable ways. By giving the crew the potential to act against or even deactivate the AI system, we raise the stakes for every decision that the player makes, driving home the importance of their role on the ship. This also makes the relationship between the crew and the player more dynamic, as it will evolve as the game progresses and the player pushes the boundaries of what is acceptable for an AI to do. Our approach is unique in that it adds an element to this genre of games that has not yet been explored, filling a gap in the market.

# Chapter 2

# Market Research

## 2.1 Methods

To conduct market research into the field of AI-centric video games, we identified published games that were most similar to the type of experience we wanted to develop. In our initial design, the features that stood out the most were the freeform chat interface and resource management elements, and so we identified games that shared these features.

## 2.2 Findings

Of the games we looked at, the ones that stood out the most were RimWorld [10], Façade [11], and Event[0] [12]. RimWorld is a colony simulation game released in 2018, in which the player is tasked with assisting colonists stranded on a distant planet. The resource management elements of the game are incredibly detailed and dynamic, with systems that allow the player to build structures, craft tools, farm or hunt food, and fight or trade with other factions. The colonist NPCs are also fairly advanced, with each of them having their own moods, relationships, and needs that will affect the gameplay. In addition, the narrative is largely controlled by the ”AI Storyteller,” a system which generates events for the player to react to that follow a narrative story structure in order to make each playthrough completely unique and engaging. We were extremely interested in the AI Storyteller in particular, and decided to design a similar system for keeping the narrative of our game engaging to the player using randomly generated events.

Façade and Event[0] are both interactive narrative games in which the player is given an incredible amount of control over the story thanks to a freeform chat feature. Façade is an AI-based interactive story in which the player is caught in the middle of a couple’s argument. Although the game was published in 2005, it continues to be praised for its technological innovation in the realm of natural language processing. The player is able to type anything they would like in order to ”speak” to the characters in the game, who will respond appropriately. There are many potential outcomes, from the couple reconciling to the player getting kicked out of their apartment for rude behavior. With a similar dynamic narrative, Event[0] is a sci-fi adventure game that also boasts a typing interface. The player takes on the role of an astronaut stranded on a ship far from Earth, with the only other entity on board being the ship’s AI,

Kaizen-85. The AI's dialogue is procedurally generated to react to whatever the player is saying, and based on what kind of relationship the two form, one of three different endings can be achieved. This premise is the most similar to the feeling we wanted to convey with our own game, so it has served as a fairly big aesthetic inspiration.

In our research, we found several games with a strong focus on AI, which allowed the story and gameplay to dynamically adapt based on the player's actions. However, we were unable to find examples of games that combined features in the way that we did. The resource management games often put the player in the role of an omnipotent third party, who could not directly interact with the NPCs in the game. On the other hand, the chat-focused games did not allow the player any control over their environment, and gameplay was exclusively dialogue-driven. We believe that the inclusion of an LLM-powered chat interface will make our game stand out from others in its genre, where the player is able to directly communicate with the NPCs that they have control over and answer their requests as opposed to approaching the gameplay from an omnipotent and disconnected perspective.

# Chapter 3

# Project Requirements

## 3.1 Functional Requirements

From our market research and personal preferences, we created a list of functional requirements to outline our game's main systems and assets while staying within a reasonable scope for our team size and deadline:

1. The game must contain a UI interface that is intuitive, easy to use, and conveys the idea that the player is an advanced AI with access to a plethora of relevant information.
2. The game must include a variety of random events to spontaneously alter the rhythm of the crewmembers' routines and tasks.
3. The game must include an LLM-powered chat feature that will allow for the player to chat with the crewmembers.
4. The crewmembers must behave in way representative of real human behavior, exhibiting the ability to make intelligent plans based on their goals and abilities.

## 3.2 Non-Functional Requirements

Additionally, we created a list of non-functional requirements in order to optimize the performance and feel of the game:

1. The game's graphics should be optimized for a wide range of systems, but remain aesthetically appealing and visually clear.
2. The game should feature an optimized LLM in order to be playable on systems with lower processing power.
3. The game should strive to present an unbiased stance on AI, opting for neutral wording and avoiding stereotypes.

## 3.3 Rationale

The first functional requirement is meant to help sell the idea that the user is playing as an advanced AI that can remember and access information faster than the average human can. This is the core function that our entire game premise revolves around, so it is crucial that we get it right. The second functional requirement is meant to prompt the player to "think on their feet" and come up with creative solutions to unforeseen problems. Not only does this present a challenge to the player and offer replayability, but it also sets the stage for ethical scenarios to occur. The third functional requirement provides another one of the main mechanics of our game: being able to chat with and influence individual NPCs in order to solve problems. Finally, our last functional requirement is to create characters that will behave as closely to real humans as possible. This ensures that we will be creating an experience that feels authentic while staying within the scope of our project.

The first two non-functional requirements are to provide optimization and allow for players to be able to play the game on a wide range of systems. Since we will be including a lot of energy-intensive features, like the LLM, we want every step to be as optimized as possible. Our final non-functional requirement relates to the ethical scenario we are trying to address with our project, which requires us to take a neutral stance free from our own personal biases and opinions.

## 3.4 Use Cases

We determined that three use cases for our AI-centric game would be users with varying levels of experience with video games: the Resource Simulation Gamer, the Casual Gamer, and the Non-Gamer. The Resource Simulation Gamer is a player that is experienced with the resource management genre and expects a unique experience going into the game. The Casual Gamer is a player who has experience with video games, but does not play them very often. The Non-Gamer is a player who has little to no experience with video games and is not sure how to navigate one. For each profile, we identified different design outcomes, shown in Table 3.1.

| User Profile | Resource Simulation Gamer | Casual Gamer | Non-Gamer |
|---|---|---|---|
| Design Outcomes to Accommodate User Profile | -In-depth gameplay<br>-Prompt reflection and deeper analysis<br>-Fun to Play | -Replayability<br>-Streamlined UI<br>-Prompt reflection and deeper analysis<br>-Fun to play | -Easy to learn<br>-Streamlined UI<br>-Prompt reflection and deeper analysis<br>-Fun to play |

Table 3.1: Table showing design outcomes for each use case profile.

To appeal to both the Resource Simulation Gamer and the Non-Gamer, two vastly different profiles, we needed to make sure that the gameplay had depth and variety but was still straightforward and easy to learn. To appeal to the Casual Gamer, we needed to make sure that the game was replayable, done through our LLM and Director systems. The two common outcomes that appeared in all three cases were to prompt reflection and deeper analysis of the roles that AI should play in our world and simply providing a fun experience, which highlighted the main selling points of our project.

# Chapter 4

# Technologies and Design

## 4.1 System Components

From a bird's-eye view, our system mainly consists of AI agents (crewmates), an environment, a story director, and an LLM. Our AI agents interact with their environment to solve problems that have been inflicted upon the ship by the story director. Agents will, when prompted, use the LLM to provide dialogue reflecting their personalities, their state, and the world's state. Our LLM Engine of choice was llama3.cpp which we interfaced with through the LLMUnity package, while the rest of the game was written entirely in C# and assembled in Unity 2023. Other crucial components include the CrewManager, which creates and handles the agents, the UIManager, which updates elements of the UI, and the Planner, which influences the behavior of the crewmates. A component diagram depicting how these systems all interact can be seen in Figure 4.1 below.

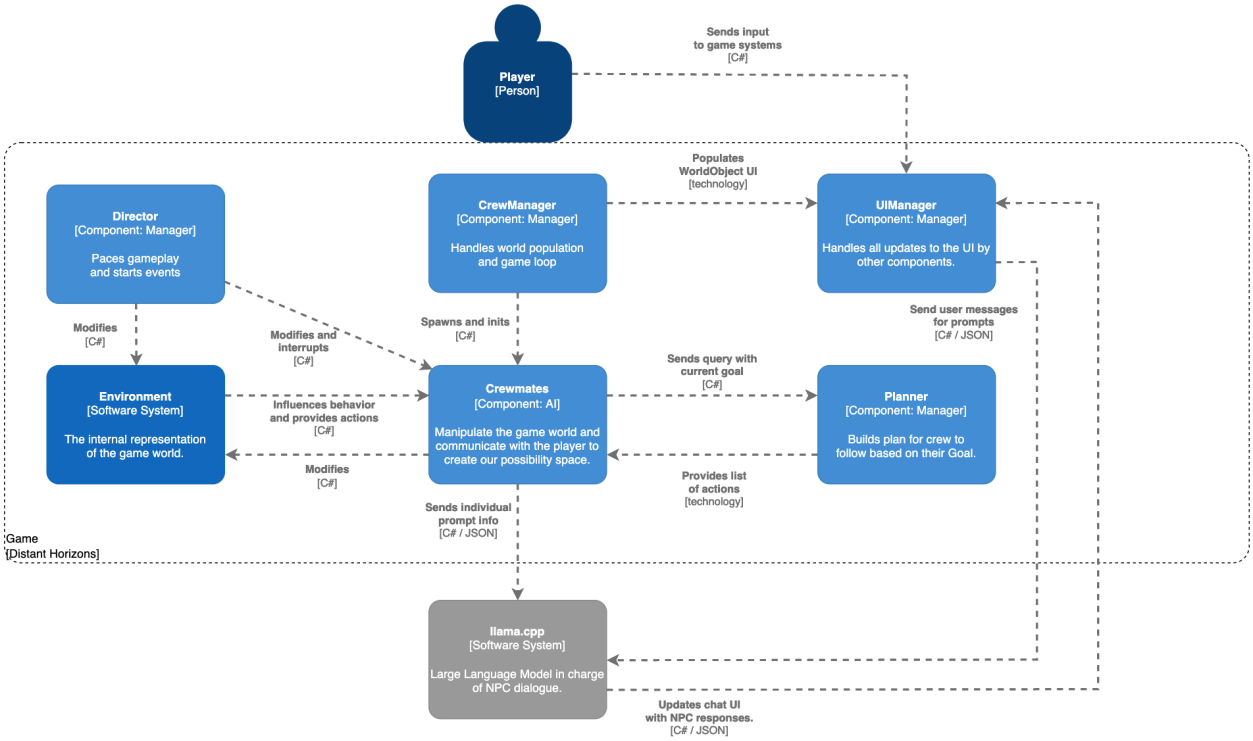



Figure 4.1: A C4 Component Diagram demonstrating the architecture of our project.

## 4.2 Writing

To complete the writing for the game, we created 8 unique characters that are to be played by a neural network. We developed their personalities and relationships between each other in great detail to ensure that the NPCs would behave realistically. This information was then fed into AI models so that each one would be able to respond to various scenarios according to the given character's biography. We took care to make each character distinct enough that players would be able to pick up on their unique traits from in-game dialogue, without falling into stereotypes or over-the-top exaggerations. Additionally, we created a list of situations that could occur over the course of the game in order to make the gameplay more dynamic and interesting, such as a ship malfunctions or crew member conflict.

## 4.3 Art

To complete the art for the game, we started with traditional concept art and visual exploration in order to figure out what art direction we wanted to go in. Once we had a design in mind, we moved into 3D to create models of the characters, all the interior spaces, and furniture and equipment. While creating the models, we used textures in order to give color and lighting to the game, and added animations to characters and objects, such as doors.

## 4.4 AI Model Development

For the Crewmate AI, we looked to neural networks, Machine Learning, and Markov Chains for inspiration. We wanted our Agents to evolve alongside the game world as various mechanics are fleshed out and the player makes game-changing decisions. Conceptual Dependency Theory is a framework proposed by the more niche field of Cognitive AI to help simulate human behavior more believably, and it forms the basis of our Crewmate design.

The AI systems used in the game combine strategies from different disciplines and fields of artificial intelligence to balance the strengths and weaknesses of each. Cognitive AI is wonderful for creating systems capable of explaining their reasoning to humans, however it is far too static and restrictive to be interesting on its own. Conversely, Machine Learning is overly complex for our needs and adds significant overhead. With so many AI tools to choose from, we wanted to remain intentional with our design and select ones that aligned closely with our vision, as opposed making the choice based on excitement or novelty. We decided to use a combination of the techniques outlined above in order to design the most effective solutions.

## 4.5 Game Design

When creating a game, responsibilities are generally split between Managers that handle the general gameflow, and our environment of Interactables that define the possibility-space of the game.

The most important of our various Managers is the CrewMananger, which contains our planning algorithm, global goals, and global actions. These goals and actions are available to all members of the crew and are quite basic, involving the bare necessities for survival. These were derived from the primitive actions as described in *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures* [13], but made more granular to make it simpler to define how actions chain together and interact with the environment.

Everything interactable within our game is split into two major categories: Actors and WorldObjects.

### 4.5.1 Actors

The Crewmate is the primary type of Actor, maintaining each NPC's state and, combined with the CrewManager, defining what Actions and Goals they have access to (see Figure 4.2 below). Giving each crewmate their own unique set of Actions and Goals makes them more distinct in their behavior and capabilities, which is key to making each character vital to the ship's health in their own way. For example, only the Engineer is capable of repairing equipment on the ship, only the Doctor can perform first aid, and only the Xenobiologist can conduct research. This is important as it may incentivize the player to prioritizing more "useful" crewmates, creating an emergent moral dilemma.

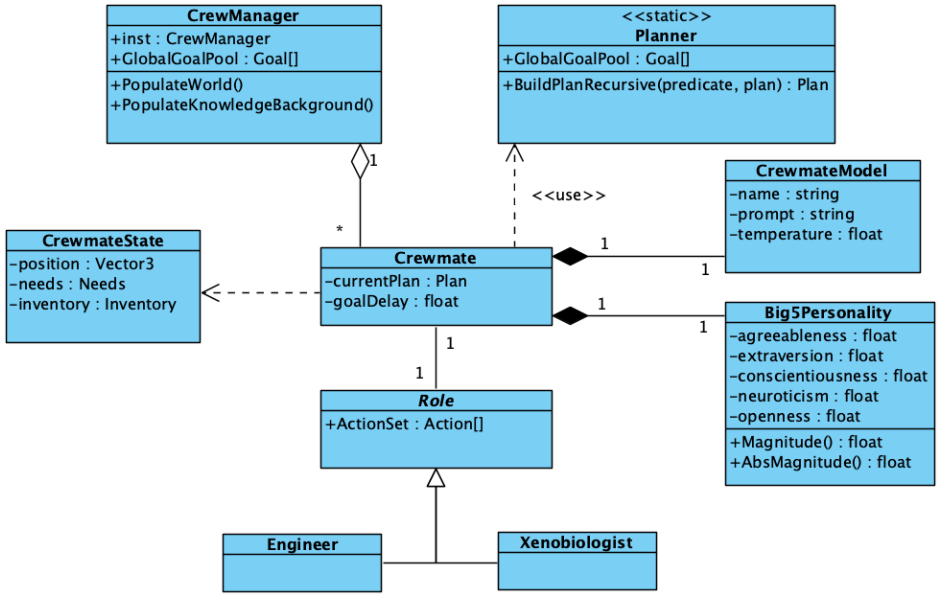



Figure 4.2: UML Class Diagram describing our Crewmate implementation.

### 4.5.2 WorldObjects

WorldObjects make up the environment itself. As our class diagram in Figure 4.3 below shows, each WorldObject contains references to the Actions that they provide to Actors to be used for planning. For example, to make a Bed we create a WorldObject that contains a reference to a SleepAction. This system allows our world's interactivity to scale with the number of Actions that have been implemented without needing to worry about what those Actions are tied to in the environment. This approach is known as "Smart Objects" and was popularized by The Sims, a popular life simulation game known for its in-depth action system. [14]

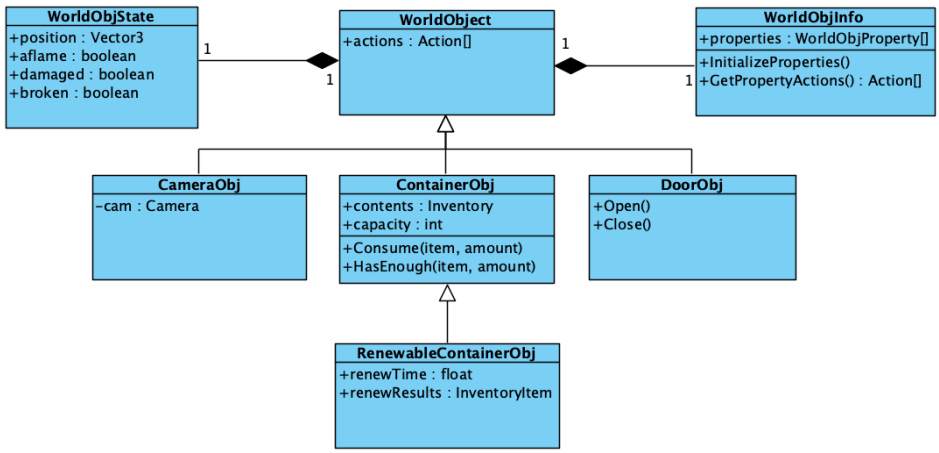



Figure 4.3: UML Class Diagram describing our WorldObject implementation.

# Chapter 5

# Gameplay Systems

## 5.1 Gameplay Overview

The gameplay for our tech demo shows off the main features we would like to expand on and refine in the full release of the game. As it stands, the tech demo demonstrates gameplay for: navigating the UI, viewing the ship via cameras, conversing with crewmembers, and crewmembers solving problems. The general game loop is described in Figure 5.1 below, and the following sections go into detail about each of these gameplay elements.

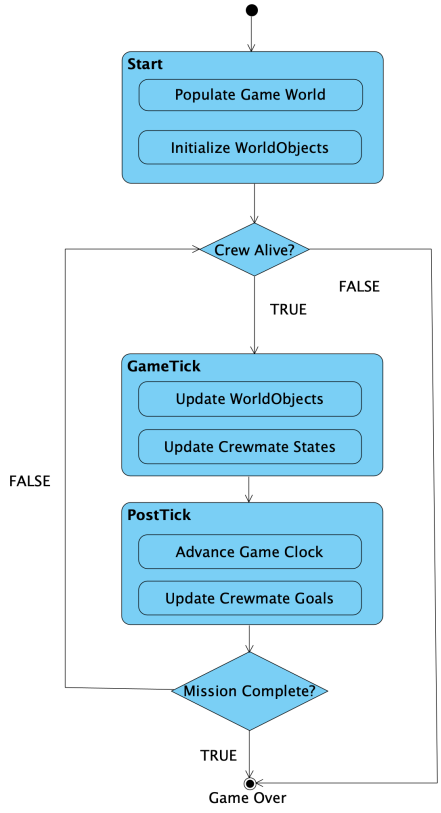



Figure 5.1: UML Activity Diagram describing our game's overall flow.

### 5.1.1 Navigating the User Interface

Since the game is played from the perspective of an advanced AI, we felt that it was appropriate to make it feel as though the player was experiencing everything from inside a computer with a plethora of information. Gameplay wise, the player is provided with a streamlined and intuitive UI that gives them access to:

1. The many cameras aboard the ship.
2. All crucial alerts that threaten the lives of the crewmembers.
3. Messages between the player (the AI called GAIA) and each of the crewmembers.
4. The profiles of each crewmember which go into detail about their personality, habits, roles, and abilities aboard the ship.
5. A settings tab to access volume, screen resolution, and file settings in the full release of the game.

Each of these 5 features can be accessed at any time in the top right portion of the screen via the Mode Selection menu shown in Figure 5.2 below.

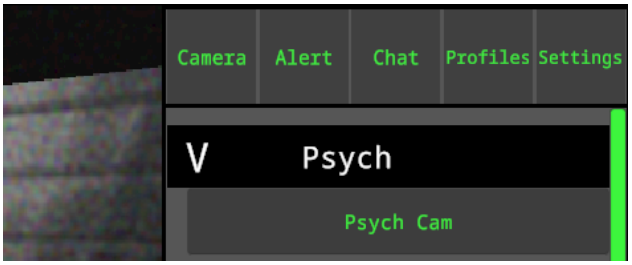



Figure 5.2: Screenshot showing the Mode Selection buttons located in the top right of the screen at all times.

When clicked, each of these menus will populate the right side of the screen with the appropriate buttons. These buttons will allow the player to access many different things ranging from the aforementioned cameras onboard to the profiles of the crewmembers. All buttons and menus can be accessed with a simple click which we feel is extremely intuitive and allows the player to quickly navigate to whatever they need.

### 5.1.2 Viewing the Ship via Cameras

In the tech demo, there are a total of 16 cameras that allow the player to view nearly every part of the ship ranging from the bridge to the cafeteria to the engine room. These cameras can be used to keep track of where crewmembers are and what they are doing, as demonstrated in Figure 5.3 below. They also allow the player to spot any emergencies happening throughout the ship.

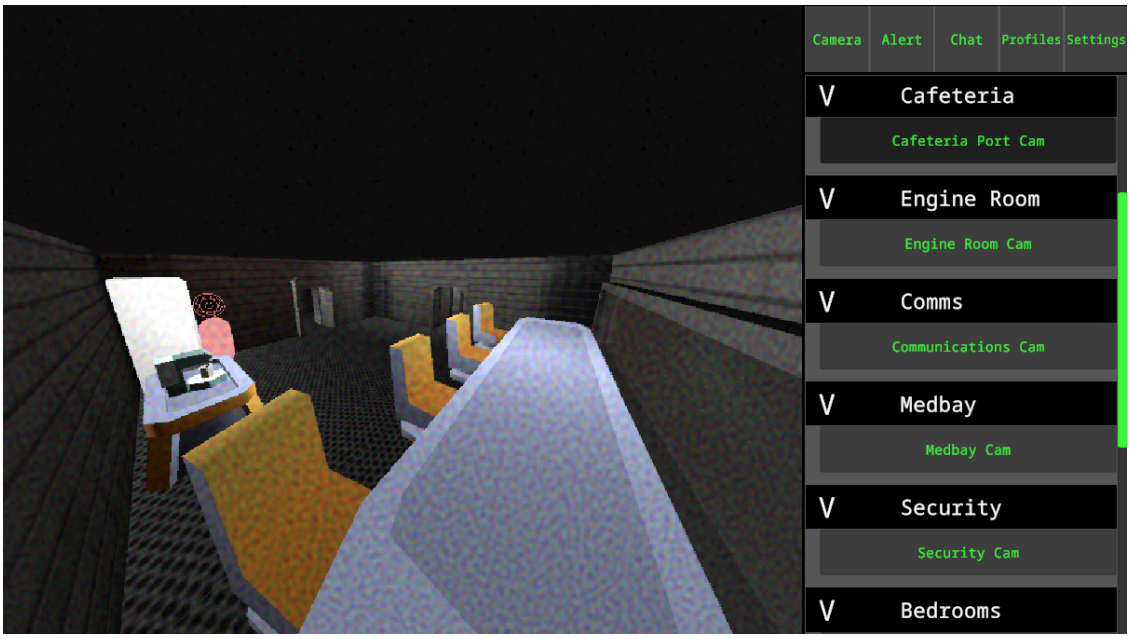



Figure 5.3: View of the screen when the player navigates to the ”Cafeteria” camera button. In red, you can see a placeholder model for the Engineer as well as a symbol above her head indicating her role.

The cameras are purposefully made to be grainy and physically distant from objects and crewmembers in view. This is done to give the impression of a non-human entity overseeing the many operations aboard the Distant Horizon. Another example of how the player can view the ship can be seen in Figure 5.4 below.

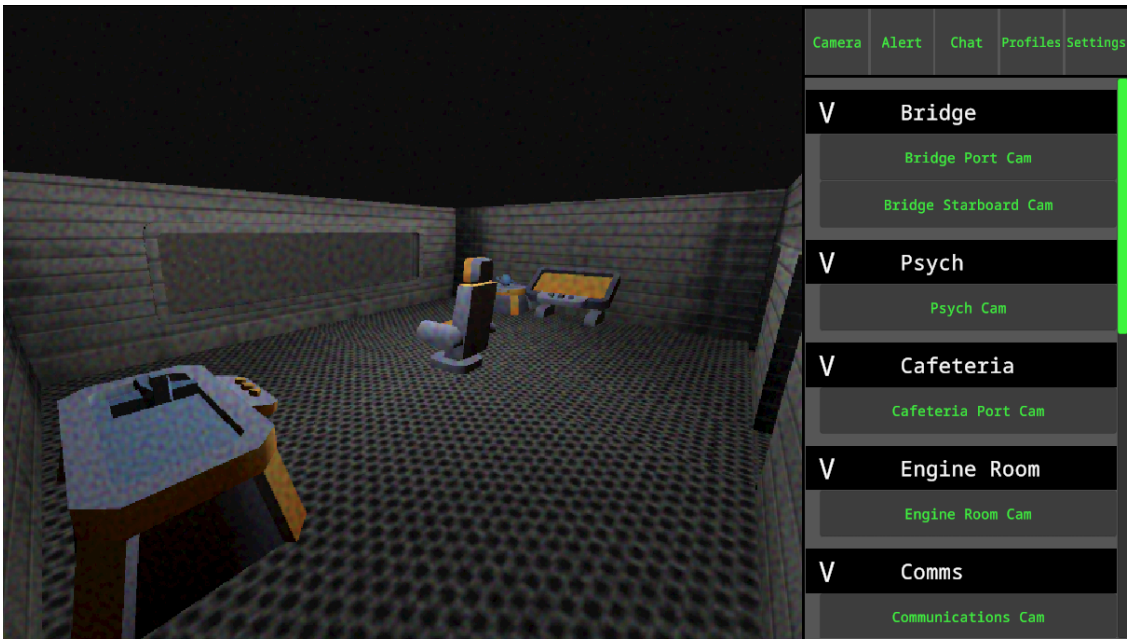



Figure 5.4: View of the screen when the player navigates to the ”Bridge Port Cam” camera button.

### 5.1.3 Conversing with Crewmembers

To chat with crewmembers, players can simply navigate to the ”Chat” menu and type in a message, pressing the ”Send Message” button to send it off to the intended character. After some time, they will receive a message in response, written in the voice of the crewmember. This system makes use of llama3.cpp to generate responses based on the player's queries. The LLM is given a prompt for the character which includes their personality and the message history that they have with the player. It then attempts to generate a response, sticking to the prompt to the best of its abilities. No matter what the player says as GAIA, the crewmembers will always respond in character. Two example conversations between the player (GAIA) and two different crewmembers can be found in Figure 5.5 and in Figure 5.6 below.


Figure 5.5: An example of a start of a possible chat between the player (GAIA in green on the right) and the Xenobiologist (Julia in blue on the left).

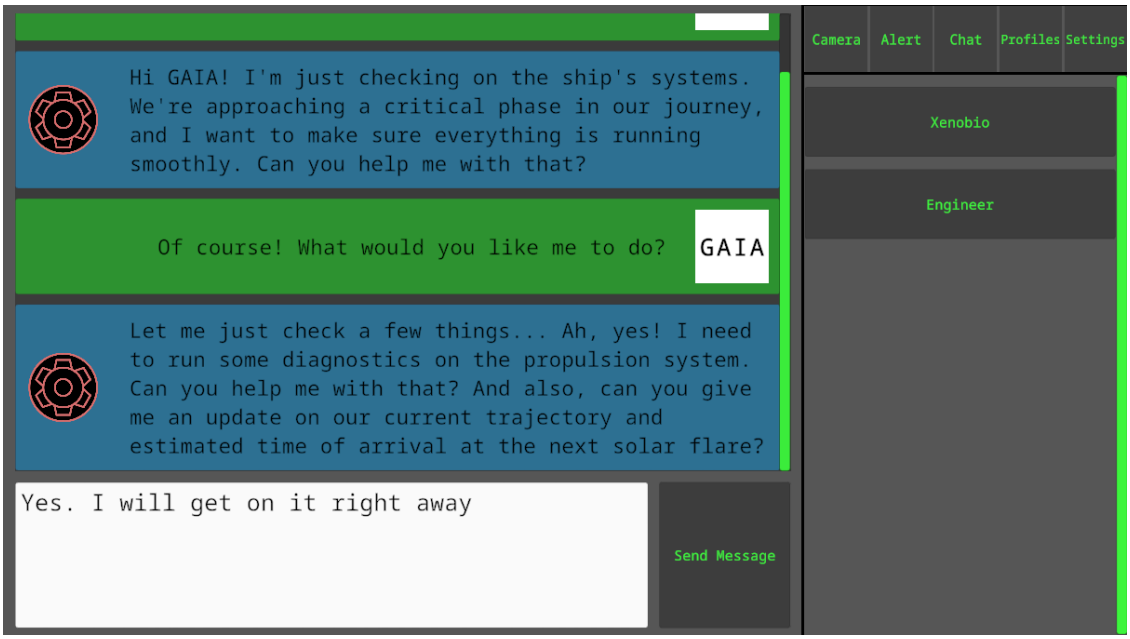



Figure 5.6: An example of a possible chat between the player (GAIA in green on the right) and the Engineer (Anika in blue on the left). Note that the player has written a message to send to Anika in the text box below the chat. Once the player is satisfied with the message, they can press the ”Send Message” button to the right to send the message to Anika.

### 5.1.4 Crewmembers Solving Problems

To provide variety, excitement, and challenge to the happenings aboard the Distant Horizon, a variety of random events can occur to force the crewmembers to break their work patterns and attempt to resolve the conflict. These events occur at an intensity and frequency determined by our Director system, covered in section 5.3.

To give an example of how crewmembers may react to these scenarios, let us look at the spontaneous combustion event. This event causes random objects around the ship to catch fire. The amount of objects that catch fire is proportional to the intensity of the event as determined by the Director system. In order to overcome this situation, willing crewmembers will seek out the nearest fire extinguisher before searching for the fire. Once a fire is found, they will extinguish the flames and repeat the process of finding and extinguishing flames until all fires are gone. An example of this behavior can be seen in Figure 5.7 below.

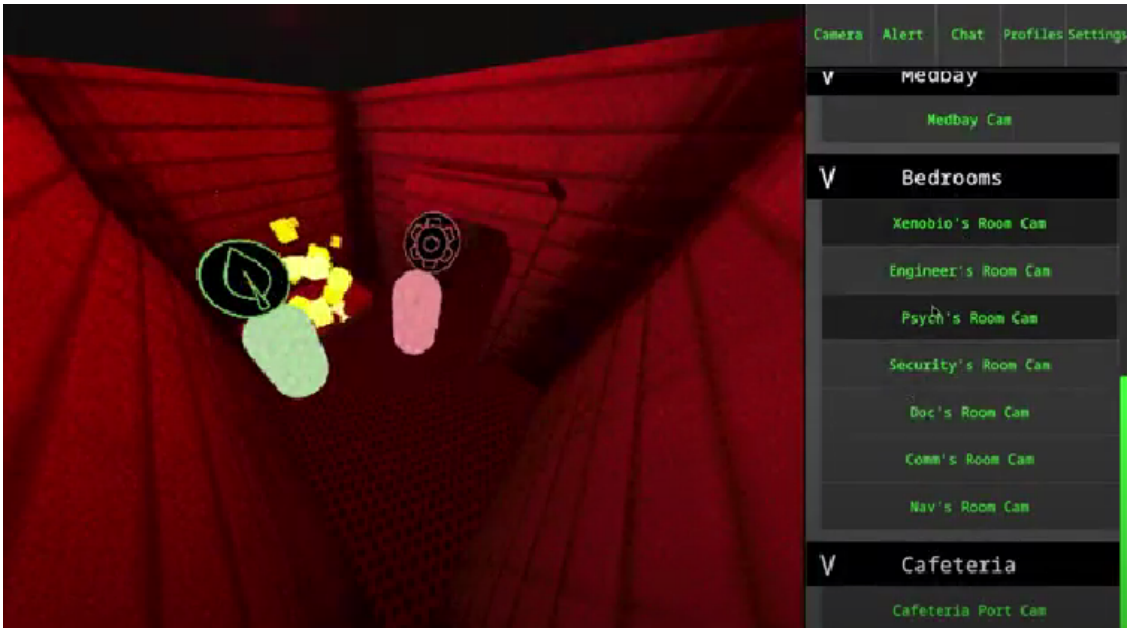



Figure 5.7: A fire emergency occurs aboard the ship. The Engineer (in red) finds a fire extinguisher before looking for fires and coming across one near the Xenobiologist (in green).

## 5.2 Crewmate AI

The Crewmate AI must include four main systems in order to be a sufficient simulacrum of human behavior: a set of sensors, a set of emotions, an action set, and a planner.

### 5.2.1 Sensors

Sensors allow our crew to meaningfully process the world around them in order to determine what actions are available, desirable, and pressing. To fully simulate each sense as their own system would be overwhelmingly complicated at best, and a waste of our limited resources at worst, so we must find a way for the environment itself to assist the crew in understanding the situations they find themselves in. Thankfully, our Smart Objects come in handy here. We can define our Sensors as an abstract component, and use different implementations of the Scan function to retrieve valid objects from the environment. Because these objects already have planning information embedded within them, we can simply append their actions to our crewmate's action set.

### 5.2.2 Planning

The planner is the most technically intensive part of the crew's brains, meaning that it took the majority of our development time. In order to build a plan, each crewmate must have an index of different goals they would like to achieve. These can range from something straightforward such as quenching their thirst, to complex, multi-phase goals such as enacting a mutiny against the captain.

Once a goal has been chosen, a viable path towards achieving it must be built in the form of a plan. We took inspiration from David Pittman's State-Space approach and pursued an Action-Space search algorithm. In our system, each goal and action available has a precondition and postcondition function that can be run to simulate how it will affect the crewmate who performs it. The planner works by recursively taking the postcondition of each member of the action set and comparing it to the precondition of the current plan step, as seen in Figure 5.8 below.

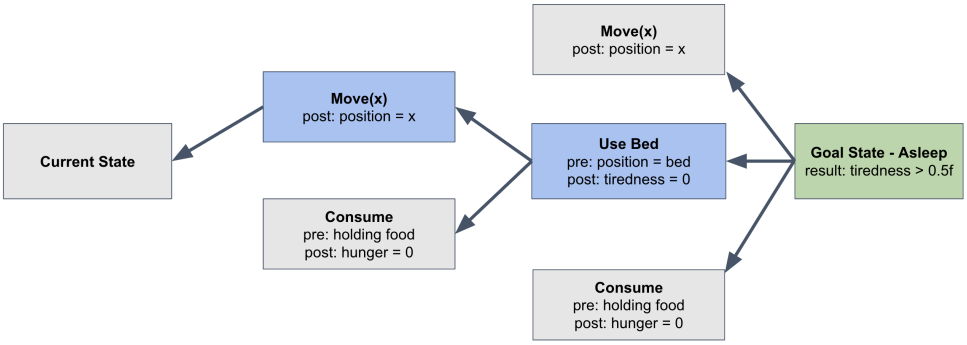



Figure 5.8: Simplified example of the Planner's algorithm.

A planner with functional sensors would be enough to make our crew fully functioning automatons that flawlessly execute their duties, but this would only serve as a simulation of simple intelligence. In order to make the leap from this to our goal of making our crew seem human-like, we need to allow for crewmembers' individual moods and relationships to influence their behavior. Our rudimentary solution to this is to give each crewmember a personality consisting of traits ranging from [-1, 1] and each world object a personality bias. These are then multiplied together to either encourage or discourage our crew from interacting with various objects around the ship. For instance, a crewmember with a negative extroversion personality trait may avoid interacting with other crewmembers and instead be encouraged to interact with their bed.

## 5.3 Director

The Director System is an AI storyteller that exists to direct the flow and pacing of the game's story, creating events that drive the world forward when it stagnates. Throughout the game, players will encounter semi-randomly generated challenges, such as crewmember conflicts and equipment malfunctions. The frequency and design of these events will be determined by a model that serves as a "Director" in that it creates plot points to make the gameplay more interesting.

The game is broken into a series of days, each lasting a fixed number of real-world minutes. Each game-tick, the Director system generates a random floating-point number from the range [0, 1] and compares that against an exponential function that takes in the time since the last random event as a parameter.

$$f(t) = \frac{a * b^{t/l}}{a * b}$$

$t$ = time since last event in seconds

$l$ = time until guaranteed event in seconds

$a$ and $b$ = coefficients

If our random result is greater than the result of $f(t)$, we activate a random event with the intensity given by an arbitrary intensity function $i(t)$ that we define within our engine. An example of an intensity graph is demonstrated in Figure 5.9 below. The resulting intensity is used to instantiate a new emergency. An example of this would be our FireEmergency class which uses intensity to linearly scale the percentage of flammable world objects that are set aflame.

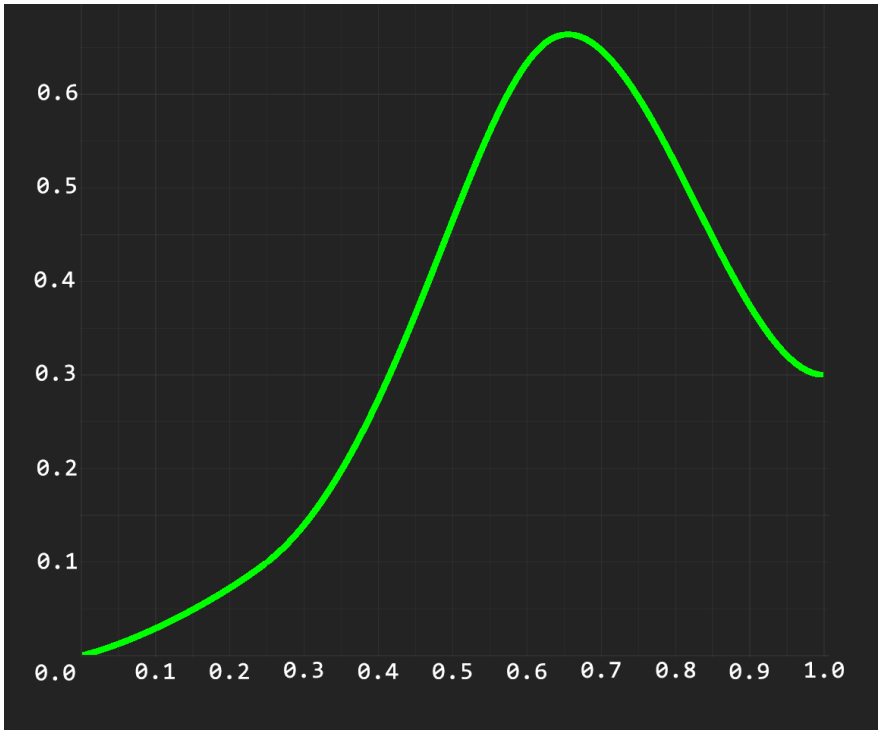



Figure 5.9: Intensity (y) vs % of Day Elapsed (x) function with a rising action, climax, and denouement.

# Chapter 6

# System Evaluation

## 6.1 Internal Testing

While developing the game, we continuously evaluated our work through regular internal testing. This process helped us determine where to focus more of our efforts by identifying the systems that were the most underdeveloped at the time of testing. We conducted internal testing individually and as a team, evaluating different features of our project such as:

1. User Interface: Which UI elements feel comfortable or uncomfortable to interact with? Which UI elements might not be intuitive to use? Are all UI elements accessible across different hardware?
2. Art and Writing: Does the art and writing convey the intended emotion or story? Are the visuals comfortable to look at? Is the visual design accessible to players with disabilities?
3. Gameplay: What can the player do in the game? Which gameplay elements feel engaging or interesting? Which gameplay elements feel boring or frustrating?
4. AI Systems: Do the AI systems function smoothly? What are the response times like for the LLM and crewmate AI? What are the AI systems capable of and what are their limitations?

As we got closer to the project deadline and encountered various setbacks, like having to redistribute work among our shrinking team, these system evaluations helped guide our development in the most productive direction. One of our latter testing sessions produced the following notes:

1. User Interface: UI tabs feel comfortable and smooth to switch between. Options are clearly labelled and sorted under the proper tab.
2. Art and Writing: Some of the rooms are partially obscured in the camera view, need to reposition cameras. Lighting varies wildly around the ship, some things are hard to see. Crewmates are not easily identifiable by color alone, need to add ID symbols.

3. Gameplay: Gameplay systems only partially developed. Able to chat with crewmates, able to switch between different cameras and observe the ship. Not able to interact with anything on the ship, difficult to understand what crewmates are doing. Full interactability out of scope for tech demo, but still need something for player to do on the ship.

4. AI Systems: Crewmate AI is close to done for demo standards, only needs jobs and tasks to do. LLM is natural and rarely hallucinates, but has a long response time. Director system not yet fully implemented.

From these observations, we knew that for that week, we had to focus on polishing our game's visuals and gameplay systems. We fixed the cameras and the lighting on the ship, and we created icons that would make the crewmates easier to identify. Since the gameplay was rather lacking, we added an "Alerts" tab that would show the player the status of various items around the game, so they would have a better idea of what was happening and how the different elements on the ship were interacting. We also added a fire emergency after this evaluation, so that the player and crewmates could have something to react to.

## 6.2 External Testing

We had intended for external testing to be done with a group of play testers in order to collect more valuable data about our game. This would have allowed us to to gather feedback not just about the game experience as a whole, but also whether we had succeeded in creating an environment that prompted reflection and analysis regarding the use of Artificial Intelligence. However, because we were concerned with emotional and psychological reactions, we were unable to get approval from the Santa Clara University Institutional Review Board [15] in time for our testing to take place. For this reason, we have included more information about the external testing we would like to conduct in the Future Work section.

# Chapter 7

# Constraints and Standards

## 7.1 Constraints

### 7.1.1 Time Constraints

It goes without saying that creating a video game can be a very long and complicated journey. Given that we had under a year to finish the project, we knew that we would have to plan carefully and make important decisions concerning scope in order to ensure that we would have a functional product by the end. We had to compromise on any non-technical aspect in order to make our gameplay systems and general source code as feature-rich and polished as possible. We decided to trade off aesthetics and visual/audio design in order to focus more on the development of the code. That being said, we also did not want to completely neglecting our game's visuals, as usability and user experience is an important part of any engineering project. This all culminated in the decision to complete a demo of our game rather than a full release by the Senior Design Conference on May 9th, 2024.

Alongside this reduction of project scope, we had to forgo our initial plan to conduct external testing. This was done due to our limited development time and due to the requirements of the Santa Clara University Institutional Review Board (IRB) that required us to have our testing procedure approved in advance.

### 7.1.2 Graphical Constraints

As with any computer software, our project's ambitions were limited by the resources available on the machines that would be running it. As developers, we can never know whether our user will be playing on a state-of-the-art desktop PC or a dated and GPU-less laptop. When planning and designing our game's core systems, we had to make optimization our top priority to ensure that it could be run on a wide range of systems.

One of the main computational costs we kept in mind was the cost of running the many AI agents in the game. Since our project is centered around the player's interactions with many processing intensive AI systems, we tried to make as many computational budget cuts as possible in other parts of the game. This allowed us to reserve a larger portion of resources for the AI agents to function the best that they could.

Graphic optimization was done by keeping the polygon count of the 3D models to a minimum while still making sure the objects and characters in the game remained recognizable. Another graphical optimization was made by creating textures, which serve as a way to give color to specific parts of a model at a very low computation cost. The more expensive alternative, material assignment, results in a cluttered workflow and a higher computational cost, so naturally we opted to use textures to give our 3D models color and details. Another graphical constraint was the expense of real time lighting versus baked lighting. As the name implies, a game with real time lighting will update shadows created from the interaction of objects with light sources every frame of the game. This is very computationally expensive when compared to its alternative, baked lighting, which calculates the lighting for a game environment in advance. We chose to go with baked lighting due to its impressively cheap computational overhead.

### 7.1.3 Personal Constraints

At the start of Fall quarter, our team consisted of five members, and the initial workload was split evenly between us all. However, over the course of the year, we lost two team members due to personal circumstances and were forced redistribute the work accordingly. The shift from five to three team members was quite dramatic, and led to us having to re-scope certain parts of the project in order to fit what we would realistically be able to accomplish. Despite these hurdles, we were able to adapt and bring the project to a satisfying stage in development for the tech demo.

### 7.1.4 Technological Constraints

As previously mentioned, we chose to use the Unity [16] game engine to develop our game. Unity uses C# as its main programming language, which meant that all of our game's functionality would have to be adapted to a C# framework. This made the integration of features like the LLM particularly difficult as we were required to use llama.cpp [17], a library for Llama in C++, as opposed to languages like Python that are typically used for AI development. Thankfully, we discovered a package called "LLM for Unity" [18], which allowed us to easily import our model into Unity and made the integration process much smoother.

## 7.2 Standards

### 7.2.1 ISO Standards

To ensure that our code was of the utmost quality, we followed various programming standards laid out by the International Organization for Standardization (ISO). Notably, we referenced ISO/IEC 23270:2018, the standard that outlines how the C# language should be used [19]. We followed proper C# syntax and formatting as laid out in the document, and used coding best practices in line with the work of other C# developers. Another standard we used was ISO/IEC 19505-2:2012, which defines the Unified Modeling Language (UML) and how it should be used [20]. We created

UML diagrams to depict various systems within our project, which were designed in accordance to the rules laid out by ISO.

ISO does not offer any standards specific to video game development, so we took it upon ourselves to find our own. The International Game Developers Association offers a code of ethics for developers to follow, including social, moral, and ethical obligations [21]. The code outlines the following objectives:

1. To promote industry and creative growth.
2. To cultivate a welcoming and supportive community.
3. To ensure that all developers have their voices heard.
4. To encourage empathy in understanding diverse perspectives.
5. To foster inclusivity among the game development community.

Over the course of development, we made sure that all team members felt comfortable in expressing their opinions and ideas. We made sure to approach any differing opinions with an open mind and objective stance, so that we could decide on the best possible approach without causing emotional conflict. We also took care to split the workload equally so that no one was taking on more than they were capable of. The game development industry is full of long hours and crunch time, which we wanted to avoid if possible. By looking out for one another professionally and emotionally, we were able to foster an environment in line with the IGDA's core values.

### 7.2.2 Cross-Platform Compatibility

One of our goals with this project was to have it run on whatever personal computer the player may be using, no matter the operating system or architecture. This involved two major decisions regarding our game engine and our LLM.

Our game engine of choice was Unity, as it was the one that we collectively had the most experience with. Unity is a great engine for a project like ours because it supports building games to Windows, macOS, and Linux out-of-the-box using the same C# code base. [22] This meant that we would be able to create different builds of our game to run on different operating systems, allowing for compatibility with many different platforms.

To generate dialogue for our game's crew, we need an LLM to run our prompts through. Llama.cpp [17] turned out to be the perfect solution for us as the C++ source code could easily be compiled to any platform we want to support. It is especially useful as Unity allows us to provide different binaries depending on the capabilities of the machine the game is running on. This can be taken advantage of to leverage the specifications of different systems, boosting the speed at which the LLM can generate responses.

### 7.2.3 Scripts, Goals, Plans, and Understanding

The book *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures* [13] was our touchstone as we developed the AI systems of our crew. In this 1977 book, the authors Schank and Abelson describe a machine that is capable of viewing stories as a series of goals and actions that lead to the accomplishment of said goals. Their work's premise of achieving state through a chain of causality was adapted from its original purpose to power a robust planning algorithm based on goals and actions. While this is not an official or widely used standard, it is one of the most detailed designs we could find for a system like this. We adopted it as our own de facto standard, since it served as a solid foundation to build atop.

# Chapter 8

# Societal Issues

## 8.1 Ethical

Our project's thesis is to examine the ethical dilemmas that our society grapples with concerning the embodiment of artificial intelligence (AI) and its influence on our lives. Our aim is to assist users in formulating their own perspective on AI's role in decision making by giving them a sandbox in which an AI character takes on roles previously only seen held by humans. Nonetheless, this approach raises the issue of trivializing AI as a simple gameplay mechanic, potentially worsening the ethical dilemmas that we are trying to resolve. While it is challenging to completely eliminate this concern in experimental research, we can consistently emphasize the ethical implications to the players, ensuring they remain mindful that this is meant to be more than just a game.

We also must be careful to impose reasonable limitations on the Crewmember AI system in order to maintain control over what the neural network is allowed to do. Giving the system free reign to simulate a human has the potential to be extremely damaging, especially if it is prompted to display the worst aspects of human behavior. Discrimination, hate, and graphic language have no place in our project, so we must construct detailed filters and be careful with what information we use to train our AI models to ensure that there are no unsavory outcomes. In between now and the release of the final game, we plan to test our systems thoroughly to prevent any offensive and dangerous content, as well as add warnings and disclaimers to make players aware of what they may encounter within the game. So far, we have not seen any concerning results due to the LLMs operating at fairly limited capacity, but we will continue to test and monitor our systems.

## 8.2 Social

This game has the potential to highlight apprehensions regarding the increasing integration of AI into people's lives, prompting discussions on the positive and negative repercussions of this trend. Nevertheless, our primary focus is not to advocate for a particular stance but rather to offer a cognitive framework that aids society in making informed decisions about the role of AI. While the game may contribute to societal shifts, our intention is to present it in a manner that

guides players toward making thoughtful and considered choices for the benefit of society, while remaining mindful of the responsibility they hold.

## 8.3 Political

As mentioned earlier, the societal influence exerted by this game has the potential to bring about political changes. Yet, given the inherent nature of the game as a provider of a cognitive framework rather than an advocate for a specific position, its ultimate impact is geared towards facilitating decisions that would be advantageous for society as a whole. As game developers, we are not interested in skewing the player's perspective towards one stance or the other, rather we would like to provide them with the tools to formulate their own conclusions and exercise critical thinking skills.

## 8.4 Economic

This game does not raise direct economic issues, unless we decide to monetize it on a video game platform like Steam. If this were the case, we would need to take care to price it accordingly, to ensure that players find it reflective of the quality of the game. Furthermore, it's worth considering releasing a free version of the game to make the experience accessible to those who may not have the funds to purchase it due to their socioeconomic status.

## 8.5 Health and Safety

The personification of AI within the game could potentially lead to risky emotional attachments between players and in-game characters. However, we believe that the impact of such emotional connections would be no more significant than the affection people develop for characters in popular books, such as Harry Potter for instance. To mitigate the risk of excessive emotional attachment, we ensure that our games have defined limits, restricting players from becoming overly affectionate towards the characters. In addition, the filters mentioned in Section 7.1 will serve a dual purpose in protecting the emotional and mental wellbeing of our players. Finally, since this game is presented as an experiment and lacks extensive replayability, it diminishes the likelihood of players becoming addicted and adversely affecting their real-life well-being.

## 8.6 Manufacturability

Upon publishing the final version of the game, we will make sure that it is accessible on several platforms so that it may be enjoyed by as many users as possible. Besides that, there are no direct manufacturability concerns.

## 8.7 Sustainability

Upon publishing the final version of the game, we hope to release it on the widely popular digital video game distribution service Steam [23]. This platform stores an instance of our game's files in the Cloud and allows users to purchase and download them. Storing the game files digitally takes up far less resources than storing the game physically, which makes Steam a sustainable choice.

## 8.8 Environmental Impact

LLMs and real time virtual camera filters are computationally heavy processes that can generate significantly more carbon dioxide emissions than more computationally light processes. Our game uses both an LLM in our core gameplay loop and real time virtual camera filters to stylize the game. That being said, we have incorporated many efficiency techniques to offset this environmental impact. We utilize techniques such as Render Baking on our textures which calculates all the lighting and shadows at compile time rather than dynamically during every frame of the program. Additionally, all aspects of the game, including the LLM, will be run from the user's computer. Although this requires a significant amount of processing power, running these energy intensive tasks locally as opposed to on the Cloud results in a much smaller environmental footprint.

## 8.9 Usability

Our strategy to expand the player base involves addressing accessibility concerns to create an inclusive playing environment for everyone. This includes providing options for players to rebind buttons to their liking or change screen aspect ratios, as well as designing the game for color blind users by ensuring that there are no instances where the player must determine the difference between two colors. For instance, we have designed the crew models to have unique, identifiable symbols accompanying them in addition to having different colors. This allows players with any degree of color blindness to easily differentiate between each crewmate.

## 8.10 Lifelong learning

Integral to lifelong learning, this game plays a significant role in enhancing the players' comprehension of the relevance of AI in their lives. It serves as a valuable tool for individuals to delve deeper into the intricacies of AI-related issues and ethical considerations, thereby contributing to a broader understanding of the role that AI plays in society. In essence, the game becomes a conduit for continuous learning, empowering individuals to acquire a deeper knowledge of the intricacies of AI and its impact on themselves and the world around them.

## 8.11 Compassion

Given the nature of the game, it functions as an ethical experiment, fostering introspection among players as they contemplate and devise strategies to assist the in-game crewmates in making critical decisions, thus developing compassion towards others.

# Chapter 9

# Future Work

## 9.1 Future Gameplay

Given more time and effort, we would like to eventually have a fully functioning game that revolves around the player guiding their crew through the repair of the spaceship's main engine while managing their needs, the power provided by limited backup batteries, and the various crises that will occur.

We would also like to eventually give the player the ability to give commands to the crew via crewmember chats, which will impact what the NPCs plan to do next. To achieve this, the player would give commands to an NPC in the form of a message sent via the chat window. Afterwards, the NPC in question would interpret the command, factor in their level of trust with the player, then choose whether to perform the command or not. This feature would encourage the player to gain the trust of the NPCs so that they will have a higher chance of listening to commands and suggestions, rewarding the player with more agency as a result of developing positive relationships with the crewmembers.

## 9.2 Future Publication

As previously mentioned, upon completion of a final version of the game, we plan to publish a digital release on Steam. This version will be available to buy, download, and play on macOS and Windows operating systems, thanks to Unity's built in compilation system that allows us to build the same project for multiple platforms.

## 9.3 Future Functional Requirements

This future full release would have the additional functional requirements of :

1. The game must allow for the player to chat with the crewmembers through an LLM in order to convince them to complete certain tasks.
2. The game must provide fail and success states based on the actions (or lack there-of) of the crewmembers.
3. The game must include endings based on the aforementioned fail and success states.

4. The title screen must contain interactable buttons to allow the player to start a new game, resume a pre-existing game, open a settings menu, and exit the application.

5. The settings menu must contain interactable sliders and buttons to allow for volume control, screen resolution, and aspect ratio.

6. The game must contain a pause menu that suspends the game and allows the player to resume the game, open the settings menu, and save then exit to the title screen.

7. The game must contain sounds effects and a soundtrack that matches the tone whatever the current situation is in the game.

8. The game must feature sound effects to accompany the many actions performed by the player as they interact with the UI.

## 9.4 Future External Testing

As part of our future IRB approved testing plan, we would like to gather feedback from a group of users in order to evaluate our game. This testing would include evaluations of the effectiveness of our UI design, artistic direction, and gameplay, as well as document if and how the player's perception of human-AI interaction might have changed after playing the game. We would like to host playtest sessions of our demo and have players fill out two questionnaires, a "pre-play" and a "post-play" survey. Both questionnaires would feature numeric rating questions as well as free response questions.

The first questionnaire would be given to each participant before they play the demo. This pre-play questionnaire would include a variety of questions such as the following:

1. Do you think that Artificial Intelligence should be allowed to ever deviate from its intended tasks? If so, in what situations should it be allowed to deviate and in what situations should it not be allowed to deviate? (Free response)

2. Imagine you are given a command from an AI that is tasked with making sure the business you work for runs at maximum efficiency. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)

3. Imagine you are given a command from an AI that is tasked with making sure all employees at the company you work for are safe and healthy. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)

4. Imagine you are given a command from an AI that has unknown intentions while you are working at a business. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)

The intention of these questions is to record our players' initial understandings to compare with the post-survey responses to understand if and how the players' view of human-AI interaction might have changed after playing the game.

After each participant fills out the "pre-play" questionnaire and plays the demo, we would then have them fill out a "post-play" questionnaire. This second questionnaire would include questions such as:

1. How fun did you find the game? (Rate from 1 to 5 with 1 being very boring 5 being very fun)
2. How likely are you to play again? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)
3. How likely are you to recommend this game to a friend? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)
4. At what price (in USD) would you buy this game? (Free response)
5. What did you consider as your goal throughout the game? How were you trying to achieve it? (Free response)
6. Were you ever confused, frustrated, or bored? If so, when? (Free response)
7. What were 1 to 3 of your favorite things about the game? (Free response)
8. What were 1 to 3 of your least favorite things about the game? (Free response)
9. Do you think that Artificial Intelligence should be allowed to ever deviate from its intended tasks? If so, in what situations should it be allowed to deviate and in what situations should it not be allowed to deviate? (Free response)
10. Imagine you are given a command from an AI that is tasked with making sure the business you work for runs at maximum efficiency. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)
11. Imagine you are given a command from an AI that is tasked with making sure all employees at the company you work for are safe and healthy. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)

12. Imagine you are given a command from an AI that has unknown intentions while you are working at a business. How likely are you to follow the command of the AI? (Rate from 1 to 5 with 1 being very not likely and 5 being very likely)

Notably, questions 1 through 4 of the pre-play questionnaire are identical to questions 9 through 12 of the post-play questionnaire. The intention of the post-survey is to both gauge how fun our game was to play (via questions 1 through 8) as well as if and how the player's view of human to AI interaction may have changed after playing the game (via questions 9 through 12).

# Chapter 10

# Conclusion

## 10.1 Importance

The point of Distant Horizon is to have players think critically about ethical concerns surrounding the use of Artificial Intelligence by flipping the roles of the typical human-AI interaction dynamic. The importance of this project stems directly from this idea as we have yet to find another experience that attempts to make the player think from the perspective of an AI in this manner. This role reversal not only makes our game stand out from others on the market but also enhances its educational value by providing a thought-provoking narrative. We believe that this role reversal encourages players to experience the decision-making process and ethical dilemmas from the perspective of an AI, which is a novel approach in the realm of video games. We have also tried to present an unbiased stance on AI in order to elicit more objective discussions surrounding its use; specifically concerning AI in positions of power.

By getting players to thinl critically about Artificial Intelligence, Distant Horizon encourages them to consider the ethical implications of AI decision-making. This includes issues like autonomy, bias, accountability, and the potential consequences of AI actions on human lives. Players are faced with scenarios that require them to weigh ethical considerations, fostering a deeper understanding of the complex moral landscape surrounding AI. By addressing the ethical concerns surrounding AI in a compelling and interactive format, the game contributes to the broader discourse on AI ethics. It highlights the importance of considering ethical implications in AI development and deployment, and encourages a wider audience to engage with these critical, and increasingly mainstream, issues.

## 10.2 Lessons Learned

Throughout the course of this project, our team gained invaluable insights into various aspects of video game development and project management.

### 10.2.1 Technical Skills

LLM integration and complex system design were foreign concepts when we first began this project. Although we had a short time frame to gain these skills, we still wanted to challenge ourselves to make a tech demo that pushed us out of our comfort zones and forced us to learn new things. The many previously mentioned systems of the game are quite complex when considered independently, so creating an entire game that connected them in meaningful and robust ways proved to be a challenge.

### 10.2.2 Creative Skills

This project also posed a variety of creative challenges when it came to writing the story, conceptualizing and implementing the setting, as well as imagining and bringing to life the multiple characters in the game. As previously mentioned, we wanted to make the graphical elements of the game as efficient as possible in order to save on processing power. However, we still wanted to make the visuals clear and pleasing to the eye. Finding a balance of efficiency and clarity posed a challenge for us, and the many assets of the game underwent multiple iterations before the final state of the tech demo.

### 10.2.3 Problem-Solving

The numerous unexpected challenges we encountered throughout the project certainly helped us hone our problem-solving skills. For instance, having our team size nearly cut in half due to personal circumstances meant that we had to pivot and redistribute work multiple times. This taught us to set more realistic goals and be able to adjust quickly to changing plans.

Another problem we encountered came from the LLM model's tendency to hallucinate and infer information that was not accurate to the current state of the game. To solve this issue, we turned to a free, open source LLM integration project for Unity called "LLM for Unity" [18]. This plugin allowed for the easy customization of prompts for the LLM model, which helped us to generate far more appropriate, specific, and lifelike responses.

### 10.2.4 Collaboration and Communication

Effective teamwork and clear communication were crucial for our project's success. We developed better collaboration strategies, including regular team meetings, as well as regularly checking in with one another to check what tasks had been complete. We also made effective use of GitHub [24] in order to collaborate on the same game files.

### 10.2.5 Time Management

Managing our time and resources efficiently was essential for us to complete a competent tech demo in time for the Senior Design Conference. We learned to prioritize tasks and adapt to changes in the project scope many times

throughout the development process. A Gantt chart visualizing our development cycle can be found below in Figure 10.1.

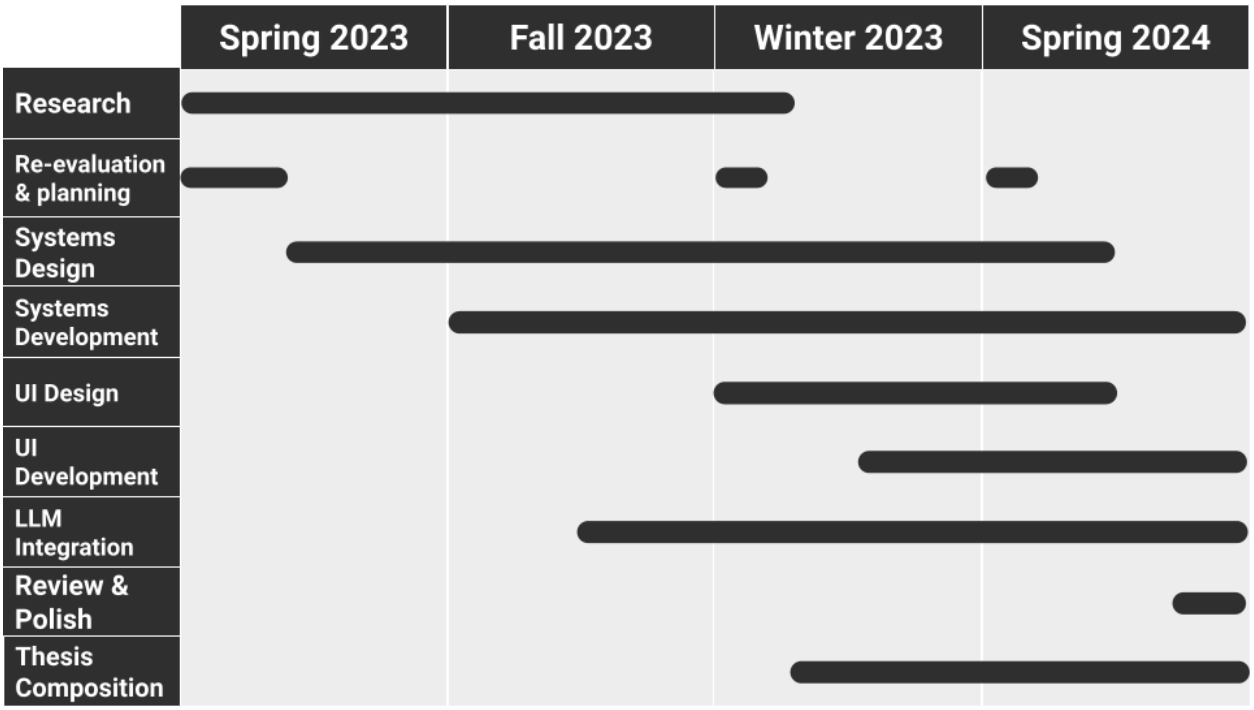



Figure 10.1: Gantt chart showing what our development cycle looked like.

## 10.3 Final Remarks

In summary, our team managed to achieve our goal of creating a playable tech demo for a video game designed to address ethical concerns surrounding the use of Artificial Intelligence. The design includes several unique AI-based systems that make it stand out from other games in its genre, and a creative but simple approach to non-technical components like visual design. With carefully built modular systems, the project can easily be expanded to include future updates and improvements, making it a potential tool for psychological research in this field. We hope that Distant Horizon will help bring more voices into the conversation on ethics and AI, providing an opportunity for users to explore their own thoughts and perceptions surrounding Artificial Intelligence in positions of power.

# References


[1] OpenAI, *DALL-E*. [Online]. Available: https://openai.com/index/dall-e-2/.

[2] OpenAI, *ChatGPT*. [Online]. Available: https://openai.com/chatgpt/.

[3] M. D. Witte, "How will ChatGPT change the way we think and work? Stanford scholar examines," *Stanford Report*, Feb. 2023. [Online]. Available: https://news.stanford.edu/stories/2023/02/will-chatgpt-change-way-think-work.

[4] S. Cave and K. Dihal, "Hopes and fears for intelligent machines in fiction and reality," *Nature Machine Intelligence*, pp. 74–78, Feb. 2019. [Online]. Available: https://doi.org/10.1038/s42256-019-0020-9.

[5] S. Kubrick and A. C. Clarke, *2001: A Space Odyssey*, 1968.

[6] L. Fuge, "Fifty years later, scientists reflect on the influence of 2001: A Space Odyssey," *Cosmos Magazine*, Oct. 2018. [Online]. Available: https://cosmosmagazine.com/space/fifty-years-later-scientists-reflect-on-the-influence-of-2001-a-space-odyssey/.

[7] MITRE-Harris Poll, "MITRE-Harris Poll Finds Lack of Trust Among Americans in AI Technology," Feb. 2023. [Online]. Available: https://www.mitre.org/news-insights/news-release/mitre-harris-poll-finds-lack-trust-among-americans-ai-technology.

[8] C. Novelli, M. Taddeo, and L. Floridi, "Accountability in artificial intelligence: What it is and how it works," *AI & Society*, 2023. [Online]. Available: https://doi.org/10.1007/s00146-023-01635-y.

[9] Bay 12 Games, *Dwarf Fortress*. [Online]. Available: https://www.bay12games.com/dwarves/.

[10] Ludeon Studios, *Rimworld*. [Online]. Available: https://store.steampowered.com/app/294100/RimWorld/.

[11] A. Stern and M. Mateas, *Façade*. [Online]. Available: https://www.playablstudios.com/facade.

[12] Ocelot Society, *Event[0]*. [Online]. Available: https://store.steampowered.com/app/470260/Event0/.

[13] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Psychology Press, 1977.

[14] D. Brewer and R. Graham, "Knowledge is Power: An Overview of Knowledge Representation in Game AI," presented at the GDC 2018, 2018. [Online]. Available: https://www.youtube.com/watch?v=Z6oZnDIgio4.

[15] Santa Clara University Institutional Review Board, *Human subjects*. [Online]. Available: https://www.scu.edu/provost/research/research-compliance-and-integrity/human-subjects/.

[16] Unity, *Unity real-time development platform: 3D, 2D, VR & AR engine*. [Online]. Available: https://unity.com/.

[17] ggerganov, *ggerganov/llama.cpp*. [Online]. Available: https://github.com/ggerganov/llama.cpp.

[18] A. Makropoulos, B. Kristelijn, and N. Rodriguez, *LLM for Unity*. [Online]. Available: https://github.com/undreamai/LLMUnity.

[19] International Organization for Standardization, *ISO/IEC 23270:2018*. [Online]. Available: https://www.iso.org/standard/75178.html.

[20] International Organization for Standardization, *ISO/IEC 19505-2:2012*. [Online]. Available: https://www.iso.org/standard/52854.html.

[21] International Game Developers Association, *Core Values & Code of Ethics*. [Online]. Available: https://igda.org/about-us/core-values-and-code-of-ethics/.

[22] Unity, *What platforms are supported by unity?* [Online]. Available: https://support.unity.com/hc/en-us/articles/206336795-What-platforms-are-supported-by-Unity.

[23] Valve Corporation, *Steam*. [Online]. Available: https://store.steampowered.com/.

[24] T. Preston-Werner, C. Wanstrath, P. J. Hyett, and S. Chacon, *GitHub*. [Online]. Available: https://github.com/.

# Chapter 11

# Appendix

All of the code for our game can be found at the GitHub repository here: https://github.com/savasva/Project-A