WRITING FOR EACH OTHER: DYNAMIC QUEST GENERATION USING IN SESSION PLAYER BEHAVIORS IN MMORPG

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Sean Mendonca

June 2020

COMMITTEE MEMBERSHIP

TITLE:	Writing For Each Other: Dynamic Quest Generation Using In Session Player Behaviors in MMORPG
AUTHOR:	Sean Mendonca
DATE SUBMITTED:	June 2020
COMMITTEE CHAIR:	Foaad Khosmood, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Michael Haungs, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Franz Kurfess, Ph.D. Professor of Computer Science

ABSTRACT

Writing For Each Other: Dynamic Quest Generation Using In Session Player Behaviors in MMORPG

Sean Mendonca

Role-playing games (RPGs) rely on interesting and varied experiences to maintain player attention. These experiences are often provided through quests, which give players tasks that are used to advance stories or events unfolding in the game. Traditional quests in video games require very specific conditions to be met, and for participating members to advance them by carrying out pre-defined actions. These types of quests are generated with perfect knowledge of the game world and are able to force desired behaviors out of the relevant non-player characters (NPCs). This becomes a major issue in massive multiplayer online (MMO) when other players can often disrupt the conditions needed for quests to unfold in a believable and immersive way, leading to the absence of a genuine multiplayer RPG experience. Our proposed solution is to dynamically create quests from real-time information on the unscripted actions of other NPCs and players in a game. This thesis shows that it is possible to create logical quests without global information knowledge, pre-defined story-trees, or prescribed player and NPC behavior. This allows players to become involved in storylines without having to perform any specific actions.

Results are shown through a game scenario created from the Panoptyk Engine, a game engine in early development designed to test AI reasoning with information and the removal of the distinction between NPC and human players. We focus on quests issued by the NPC faction leaders of several in-game groups known as factions. Our generated quests are created logically from the pre-defined personality of each NPC leader, their memory of previous events, and information given to them by in-game sources. Long-spanning conflicts are seen to emerge from factions issuing quests against each other; these conflicts can be represented in a coherent narrative. A user study shows that players felt quests were logical, that players were able to recognize quests were based on events happening in the game, and that players experienced follow-up consequences from their actions in quests.

ACKNOWLEDGMENTS

Thanks to:

- My family for their consistent support and encouragement that made all of this possible.

- Dr. Khosmood for his advice and assistance that helped this project overcome many difficult obstacles.

- My friends, especially Chris, Alex, and Hayden, for listening to my venting and helping me out when they could.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Modern video games have become increasingly more complex over the past decade, with every advancement increasing consumer expectations for future releases. MMO games are no stranger to this trend, which is an issue as they often require a colossal amount of development time and require a critical user participation level to succeed. User participation is an essential component to MMO games, gameplay in these types of games is largely defined by how players interact with each other and how their actions affect the overall game-world. These factors make it quite difficult to test new technologies on the genre without being directly involved in the development of a commercial MMO game. The creation of the Panoptyk MMO research platform with other researchers was a key component to the central topic of quest generation.

Panoptyk is an open-source [47, 46] experimental MMORPG where NPCs are designed to be indistinguishable from human players. NPCs are given the same information and capabilities as human players, and the game is heavily based around the creation, possession, and exchange of information around the game world. Players form or join factions that have conflicting overall goals, and must engage in intrigue and deception to fulfill their faction's objectives. There is also a rank system within factions that allows leaders to give out missions and set priorities. Panoptyk allows for power struggles within factions, as well as double agents infiltrating opposing factions. Panoptyk depends on quests and interactions between various players to create a coherent and engaging story. Some high level concepts of Panoptyk have been published as a poster in FDG 2019 [39].

Story is the backbone of many games. Writers interested in story immersion must spend a large amount of time putting a story together in such a way as to make the player feel like their decisions are impacting the game. This type of work traditionally requires a handcrafted decision tree planned by the developer, with consequences being specific and hard-coded. Manually creating all possible responses to a player's actions is an extremely time consuming and expensive task for games that try to tell stories. There have been many research and commercial attempts to use procedural content generation for story and drama related content but they usually are not able to match the quality of human-created story content [58]. This type of procedural content is often relegated to side quests or minor content in major commercial games such as Skyrim, Fallout 4, and Assassin's Creed Odyssey.

Many games that rely on extensive procedural content generation have been criticized as being "shallow." The same issue can seen in procedurally generated quests [20]. To avoid these shortcomings, we propose to dynamically generate quests from a combination of current events, previous quests, character personalities, and relationships.

This thesis covers work done on the Panoptyk framework to turn it into a viable research system, as well as the experiments conducted on story generation and analysis of future works.

Chapter 2

BACKGROUND

This section provides an overview of research related to AI, video games, and narrative. It gives an overview on Procedural Content Generation (PCG), narrative breakdown/generation, and AI methods used for this project. It also discusses the motivation for Panoptyk, the framework this project was made and tested on.

2.1 Procedural content generation

PCG is the algorithmic creation of content with limited user input [62]. The main purpose of PCG is to reduce the human development effort needed to create new content. This process has been applied to many aspects of game design. Most often we can see this in the generation of vegetation or other "natural" formations in games that can be expected to hold a similar pattern. PCG shows itself to be extremely useful in generating repetitive content with minor differences. Many mainstream games take advantage of PCG at some level. The Borderlands series and likely other RPG series make of use of PCG to generate in-game items. Borderlands uses PCG to create unique combinations of stats, effects, and color layouts. It advertises a "near endless variety in weapons and item drops." Minecraft and No Man's Sky use PCG on an even larger scale, their entire playable world is generated using PCG. The large size of the worlds in those games is made possible by PCG, and is a major selling point in their marketing. The application of PCG to quests or story has been relatively limited, perhaps due to simple and repetitive applications currently associated with it. Mainstream games at the time of this paper have yet to make widespread use of procedural content generation in narrative, but have used it in related elements such as quest generation.

2.1.1 Constructive generation methods

Constructive generation methods cover any form of PCG that is completed after generating content in a single pass [69]. Any quality control on the generated content has to occur while the content is being generated. These methods provide very limited control over their output, but are extremely fast, allowing them to be used during runtime of a game [62]. This type of PCG is often used for terrain [38] and level generation [62].

2.1.2 Search-based approaches

The most widely seen form of PCG in current research is the search-based approach [62][70]. While other techniques exists, search-based methods can be used in almost every major form of PCG, including level generation, landscape generation, texture generation, and quest generation [62]. The search-based approach focuses on representing content in a way that allows the content space to become searchable and then creating an evaluation function that allows for a effective search. In contrast to constructive generation methods, search-based methods evaluate the content they generate and use that information for the next round of content generation.

2.1.3 Formal grammars

Formal grammars list a set of production rules that specify how to turn one string into another. Each symbol on the left-hand side of a formal grammar is defined as the possible combinations of symbol(s) on the right-hand side of the grammar. Formal grammars are well suited to define patterns that can be used for PCG. They have seen extensive use in level and vegetation generation due to the inherent patterns in those types of content [62].

2.1.4 ASP solvers

Answer set programming (ASP) is logic-based approach where logical relations are declared in a Prolog-like language [62][66]. This technique is well suited for forms of content generation that can be expressed as a constraint satisfaction problem. This form of PCG has been used by researchers in puzzle creation [65] and for placement of enemies and items in dungeons.

2.1.5 Mixed-initiative content creation

Mixed-initiative content creation uses human input to assist procedural content generation [62]. This type of content creation takes two forms: computer-aided design, and interactive evolution. In computer-aided design, computers assist humans in their creative process. This form of content creation functions more as a tool to assist humans than a true content generator. In interactive evolution, computers generate content and rely on human feedback to guide them to produce content that humans prefer. Interactive evolution can be incorporated in any form of PCG as a way to improve content generation. The game Galactic Arms Race uses a searched-based approach to weapon generation that incorporates interactive evolution as a way of evaluating generated content [19].

2.1.6 Machine learning

Procedural content via machine learning (PCGML) is an emerging form of PCG as of 2019. PCGML is the generation of content using machine learning models trained on pre-existing content from previous sessions [68]. Various methods of machine learning have been used, including neural networks, Markov models, and clustering. This thesis does not make use of any of these methods, but it is possible that future related research could take advantage of them.

2.2 Procedural quest generation

Quests in video games are tasks that a player-controlled character or party may complete to obtain a reward [45]. The primary function of quests is to provide players in-game goals to work towards [11]. However, quests are also often used as a tool to tell stories in video games; quest objectives and their resulting consequences are well suited to act as plot devices. The means the ideal quest is able to tell a narrative as part of its assigned tasks. Like all forms of procedural content generation, procedural quest generation emerged to reduce the human burden of content creation. Procedural quest generation dynamically assigns players different tasks at different in-game locations but usually does not feature a complex or detailed narrative. The Elder Scrolls V: Skyrim [4] and Fallout 4 [20] make extensive use of procedural quest generation which they call "Radiant Story." [5] The use of "Radiant Story" allows for potentially unlimited quests to be generated, with the downside of being extremely repetitive. Most Radiant Story quests involved unnamed generic NPCs, a simple goal (like kill all enemies at x location), and a limited set of dialogue lines. Many players have criticized certain radiant quests that were repeatedly triggered in Fallout 4, and at least one common radiant quest became a running joke on the internet [20]. Because of the power of quests to influence a game's narrative, the topic of quest generation can overlap significantly with narrative generation.

2.3 Narrative breakdown/generation

The idea of breaking down narrative structure and analyzing the underlying pattern in stories predates computers, which is not unexpected considering writing has existed for thousands of years. Propp's 1928 *Morphology of the tale* is a often cited book in literary theory for the way he is able to break down and find patterns in Russian fairy tales [51]. The narrative patterns described by Propp, such as a hero receiving a mission, fulfilling it through various tasks, and getting rewarded in the end, are also popular in computer games [61].

There has been a variety of approaches to the field of narrative generation in academia. Content generation through grammars allow many potential options to be used at a given point and has been used for plot generation [8]. However, they can also be limiting. Most implementations are unable to handle core variations during runtime, meaning they will not be able to act on history or provide unique options to different player types. The related field of quest design focuses on creating specific tasks for the player to complete in the game. Various planning models have been used in the field of quest and narrative design [62].

The most common method of representing a branching narrative in the modern game industry is decision trees [73]. The developer will usually pre-define the exact structure of all possible options and results that can happen in the story. Using decision trees always requires a developer to manually write every potential story event, many of which are never seen by the majority of players. The cost benefit ratio of this method forces developers to significantly limit either the depth or amount of storylines that can appear in their narrative.

2.4 Believable NPCs/AI

Many games have attempted to create AI agents that behave in a believable way. This problem often draws parallels to the classic Turing test, but is more focused on behavior in the context of a game's environment rather than fully human behavior. The believability of AI agents is a major component of Player Experience Modeling (PEM), a study dedicated to creating the desired player experience in a game [74]. The ability to positively affect player experience makes believable AI agents a major component and selling point of many modern games.

The believability of AI agents in games is highly dependent on the role the AI plays. In many situations, an AI agent will take the role of a NPC that is expected to display behavior that fulfills their character's role in the overall experience. In these cases, the believability of the AI agent depends on its ability to fulfill its expected plot role rather than mimic what a human player would do in that situation. NPCs that are "too smart" are often not fun to play against and hamper overall player experience [76]. AI agents that take the place of a human player are expected to be far more intelligent, and play under the same rules as a human. AI players usually serve as a substitute for humans in Player vs Player (PvP) content, but often display behavior which makes them easily distinguishable from a real human. These behaviors include predictable strategies, inhuman precision, and a failure to adapt to unexpected situations. These flaws make it so that the AI players are often unable to replace human players, and has been a major topic of research in AI [76].

2.5 Massively multiplayer online games

Massively multiplayer online (MMO) games are a major genre in video games. Over the past decades, the genre rose in both size and popularity on a global scale.

While games in the MMO genre are diverse, they share some key features: large persistent game worlds, thousands of online players, quests, and lore. These games often attract and retain their players by emphasizing social interaction and worlds full of numerous tasks to complete [9]. MMOs face several unique challenges which can make it difficult and expensive to create a successful MMO game.

2.5.1 Need for content

The creation of content to keep players invested in an MMO game can be a very costly affair. Content can vary greatly between different types of MMO games and includes but is not limited to: story expansions, cosmetic additions, new characters to play, and re-balancing of gameplay. Unlike games based on a single-player experience, MMOs are usually designed to have a stable player-base for an extended period of time. This means content creation must continue after release in order to maintain player interest. Panoptyk attempts to alleviate this burden by supporting both automatic content generation and third party contributions.

2.5.2 Boring NPC behavior

A common problem for NPCs characters in multiple game genres is their limited capacity for showing interesting behavior, resulting in them often being seen as "boring." This problem is particularly hard to address for MMO games as NPCs are often designed to be as predictable as possible to guarantee a consistent experience for human players. For example, NPCs in MMORPGs usually display scripted behavior that they rarely deviate from. This largely has to do with maintaining expectations given a game's setting and plot. While some games outside of the MMO genre, such as Skyrim, try to allow more variation in NPC behavior, they still struggle to balance consistency and expression of complex behaviors. Panoptyk addresses this by abandoning the use of scripted narrative and instead relying on emergent behavior from characters to create compelling narrative.

2.5.3 Bootstrap problem

The bootstrap problem for MMOs can be defined as the following: MMO games cannot offer their designed experience without a minimum number of players playing the game. This is because many gameplay systems are designed with multiplayer interactions in mind. Without a minimum player-base vital gameplay systems such as in-game economies, cooperative content, competitive PvP, and guilds cannot exist. Games that fail to "bootstrap" an audience are often doomed to fail as they are unable to offer their intended experience. Likewise, MMO games that cannot maintain a player-base are often considered "dead." Panoptyk approaches this issue by supporting AI bots that act like human players and make up the initial population of any server.

2.5.4 Hardware limitations/instancing

High numbers of players can pose unique problems to MMOs that allow players to freely wander the game world. Large player density in a certain area can cause significant performance and gameplay issues for players in that area. From a performance perspective, players' hardware will eventually struggle to render other characters and actions. From a gameplay perspective, high population in an area will deplete any resources (such as killable mobs that serve as quest objectives) and leave people waiting for things to respawn. Most MMOs will address this problem through the use of instancing and alternative servers. While this does solve the hardware and gameplay issues caused by high population, it also makes each area exist in a different state depending on its instance and server. This means local actions have no direct consequences or impact to players in other instances, servers, or the game world as a whole.

Instancing can usually be performed fairly seamlessly, but can becoming jarring when players transition in-and-out of zones or try to meet their friends at a certain location. The use of instancing also enforces strong limitations in determining overall world state. For example, if the leader of an important faction dies in one instance and lives in the other, which reality is carried back to the overall world? Games that make use of instancing must usually prevent players from doing actions that have a noticeable impact on the overall world to avoid this problem. In summary, instancing of a location solves hardware related performance limitations caused by crowds at the cost of lowering player impact and immersion. Panoptyk solves issues related to high population locations without resorting to instancing.

2.6 Tools

2.6.1 Typescript

The majority of the code for both Panoptyk and all bots designed for it are written in Typescript. Typescript is a open-source programming language by Microsoft that builds upon JavaScript, a programming language commonly used for web applications [37]. It builds upon JavaScript by allowing static typing in the language. Typescript was chosen to streamline the development and usage of complex models needed for both the server and connecting bot agents. Strict typing helped drastically reduce the amount of run-time errors and incorrect usage of server models.

2.6.2 Node.js

Panoptyk makes use of Node.js to handle communication between server and client. Node.js is an asynchronous even-driven JavaScript runtime designed to be used for network applications [44]. The developers claim that the asynchronous nature of Node.js makes it scale very well to large number of connections. This also means that the Panoptyk server does not have to worry about syncing multiple network threads. Node.js theoretically allows hundreds or thousands of users to connect to a Panoptyk server.

2.6.3 Phaser

The Panoptyk client makes use of Phaser to render a game world for human users. Phaser is a free open-source framework for Canvas and WebGL powered browser games [48]. While the Panoptyk server has no direct control over the visual aspects of the human client, we found it was still essential to create an aesthetically pleasing world for human users to interact with. Phaser was an ideal choice for the visual client of Panoptyk due to it being designed to make browser based games. Phaser fully supported Typescript and was easily able to make use of Panoptyk's client module. The Phaser client also servers as an example for other users to make Panoptyk-based games.

Chapter 3

RELATED WORKS

The work in this thesis builds on previous research in procedural content generation, believable artificial intelligence, and massively multiplayer online game research. Some of the following related works show some degree of overlap between those fields and are discussed under the field that seems most fitting.

3.1 Procedural content generation

3.1.1 Emergent narrative

The topic of procedurally generating compelling stories has been a difficult research problem for decades. While initially this field only involved the generation of readable text stories for humans, it has expanded to include the generation of virtual stories that can be experienced in video games. Work under this field has been called procedural quest generation, procedural story generation, and narrative generation. The term "emergent narrative" can be used to encompass all of these intersecting fields. Emergent narrative is the creation of stories through the events of a computer simulation. These simulations are often interactive experiences that let users influence the simulated world with their actions. The "narrative" part of the simulation is created through the events that occur as a natural part of characters performing actions to achieve their goals. The following works represent major contributions to the field and works that influenced this thesis.

3.1.1.1 SAGA II

James Ryan, an AI researcher who wrote his dissertation on procedural narrative generation, gives SAGA II credit for being the first computer system for story generation [57]. Unfortunately, there is very little accessible information about SAGA II, only the original operating notes and a 1960s CBS television episode can be readily found online. According to SAGA II's operating manual, scenes have a number of possible successor scenes that they probabilistically transition into depending on the world state [42].

3.1.1.2 Grammar-based story generators

Inspired by Propp's *Morphology of the Folktale*, Joseph Grimes designed a system that would use the narrative structures defined by Propp to randomly generate stories [57]. The system would randomly select several of Propp's 31 functions (structural elements that typically occurred within his analyzed tales), correctly order them, generate appropriate characters to fulfill story roles, and then use natural language generation to express the story in a readable way for humans. Representing Propp's 31 functions as rules for story generation makes Grimes's system the first grammar-based story generator. Grimes abandoned the project before publishing any work on it claiming "The thing I never put my finger on was that my computer's stories had Propp's elements and sequences, but they were all boring" [57].

Eventually, after Grimes's work, other papers detailing grammar based story generators were published [27]. Unfortunately, while grammar-based story generators were able to create readable and grammatically correct generations that resembled stories, they still fell quite far from human-authored stories and were soon attacked as a flawed approach by other researchers in the field. Black and Wilensky accused story grammars of being unable to express all types of stories, able to produce "nonstories," and of not being a useful tool to model stories with complicated plots [6]. Story grammars were defended by some researchers [15, 55], but generally fell out of favor with the emergence of planning-based story generators.

3.1.1.3 ***TAIL-SPIN* and story generation via planning**

One of the most cited early attempts to generate stories with computers is the *TAIL-SPIN* program by Meehan [36]. It was able to create simple stories based on resolving a problem/goal of main character. The program was able to use location, personality, relationships, and world state to plan out the actions each character takes. The resulting "stories" were a description of the modeled worlds and logical actions characters took to fulfill their goals. However, even though characters acted in a reasonable way to achieve their goals, there was no way for the system to differentiate a compelling story from a mundane account of facts and actions that characters took to accomplish simple goals. Nonetheless, *TALE-SPIN* is considered the first planning-based approach to story generation and is frequently referenced by other papers in the field [62]. Other planning based narrative generation programs that followed *TAIL-SPIN* include but are not limited to *AUTHOR* [10], *UNIVERSE* [29], and *Minstrel* [72].

3.1.1.4 Drama management

Drama management describes the process of guiding a narrative based on a character's personal choices and preferences. In the researcher paper *Drama Management and Player Modeling for Interactive Fiction Games*, Sharma et al. asserted that creating an effective player model that influences the occurring drama is key to success in interactive narratives [63]. Others researchers have agreed or attempted projects based on that principle. One of the earlier approaches to this was declarative optimization-based drama management by Nelson et al. [43]. Their system would use a drama manager that decides plot points based on an author-specified evaluation function. They did not go into the specifics of implantation, and mostly focused on their design principles and problems dealing with complexity. A 2008 project by Riedl et al. goes into far more detail than the previous example, outlying the design, implementation, results, and shortcomings of their Interactive Narrative Tacit Adaptive Leader Experience system [52]. They found their system to be extremely repetitive, often re-presenting an option the player refused in order to maintain certain storylines. This would lead to players being frustrated and feeling like their choices did not matter. A more recent project in 2014 by Roberts et al. tries to make it easier for drama managers to influence player choices [54]. Their system presents players with information related to their current choice, sometimes providing a heavy bias to pick a certain option. They found that players were more likely to experience the results the author intended, and that players did not feel a significant difference in control compared to narratives that lacked the system.

The term "Drama Management" is not always used when describing a system that controls the story in an interactive narrative. Other projects have focused on the same aspects as Drama Management with either a different name or smaller scale. In a 2010 project, Li et al. created a system that would alter details of the plot based on a play model to provide a narrative that fit the player best [32]. Their model required an pre-defined plot structure, but was designed to have interchangeable events at each point. Unfortunately, their system was prone to story incoherence or no valid events existing for certain playthroughs. A separate journal article on interactive narratives makes the related claim that the greatest hurdle to overcome was to create evolving relationships between players and aspects of the narrative [61]. These type of relationships put very specific constraints on procedurally generated narratives.

3.1.1.5 Dwarf Fortress

Dwarf Fortress is often seen as one of the most successful and influential practical applications of emergent narrative and procedural content generation [58]. The game procedurally generates the entire world from scratch, accounting for temperature, rainfall, drainage, vegetation, and salinity [2]. It then simulates water erosion of terrain before the generation suitable life for the newly created world. Lastly, 200 years worth of actions, including the rise and fall of civilizations, is generated through simulation. It is only at this point that the player is introduced to the game world. Rather than trying to explicitly generate stories, the game relies on the depth of the simulation to naturally create interesting events and situations that can be retold as a story. The comprehensive nature of the simulation allows for many complex narratives to be constructed from the different interactions with procedurally generated characters, factions, terrain, and settlements.

3.1.2 Quest generation

As mentioned in Chapter 2, quests serve as both a gameplay and a narrative tool. There is significant overlap between quest generation and narrative generation if one attempts to make use of the narrative potential of quests. Some authors, such as Kybartas and Verbrugge, use “quest” and “narrative” interchangeably in the context of video games [28]. Many research works use the narrative generated by quests as one of the main ways to evaluate their research, but some focus on other gameplay aspects.

3.1.2.1 Dwarf Fortress adventure quests

As mentioned in Section 3.1.1.5, Dwarf Fortress is prominent example of nearly every aspect of PCG. Procedurally generated quests appear in Adventure mode, a special mode where the player controls a single character rather than an entire fortress. In this mode, quests can be assigned by various people of authority in the game. These quests are "kill" quests that task the player with killing a certain number of members of an opposing faction or assassinating an important target [13]. Completing quests helps the player build a reputation as a loyal soldier of that faction.

3.1.2.2 Quest generation using Petri Nets

Lee and Cho propose a quest generation method using Petri Nets to represent game events [30]. Petri Nets are directed bipartite graph where circular nodes represent places and bar nodes represent transitions. The directed arcs in the graph describe what places are preconditions or post-conditions for transitions. Overall, Petri Nets are suited to be used as a planning model. Lee and Cho are able to represent the events that make up Never Winter Night quests using Petri Nets. Their system attempts to predict what type of quests a player likes by using a Bayesian Network on their play history. Each quest type has a number of related events associated with it. The system then starts a quest with an event associated with the chosen quest type. The ending result of an event determines the possible successor events. The authors tested their system on seven graduate students and claimed generally positive results. In a followup paper, Lee and Cho reiterate their work and propose using genetic algorithms and a game with more control over gameplay modules. [31]

3.1.2.3 Quest generation in MMORPGs

There have been several attempts to use procedural quest generation in MMORPGs. The “TRUE STORY” system from Pita et al. attempts to generate compelling quests for MMORPGs by using the past quests and/or history of players [49]. Their system is able to create “kill,” “steal,” “discover,” and “retrieve” quests based on positive or negative history with other players or items. The system also takes into accounts other factors such as the player’s skills when assigning appropriate quest targets. Unfortunately, the authors did not discuss any form of user testing or other validation metrics. They admitted that the system did not have a way to determine the importance of different pieces of history and therefore struggled to generate a meaningful chain of quests.

Tomai et al. propose an adaptive quest generation system capable of mitigating the conflict between the authored narrative of an MMORPG and the changes to the world caused by player actions [71]. The authors focus on the concept of “kill” quests, quests where players are tasked to kill mobs that regularly respawn. In their prototype game, the authors altered spawning so that players could reduce the population of mobs and eventually cause them to stop spawning. To compensate for the loss of of “kill” quest targets, the authors altered “kill” quests to dynamically pick new targets in different zones once mob population had been lowered enough. In an initial user test, the authors found that the dynamic quests resulted in less competition than normal quests, even when compared to normal quests that had normal spawning; they were able to increase player impact without causing negative competition between players.

Doran and Parberry describe a prototype quest generator based on their analysis of over 750 quests from Eve Online, World of Warcraft, Everquest, and Vanguard: Saga of Heroes [11]. Doran and Parberry categorize quests from said games into nine different categories representing the motivation behind the quests. The motivations were then associated with tasks that they call "strategies" that could be fulfilled with a sequence of actions. Their final prototype is able to randomly generate quests from their database of motivations and strategies as well as list out actions that players would need to do to complete them. In a followup paper, Doran and Parberry describe the implementation of their quest generator into the MMORPG Everquest [12]. They are able to run their generated quests on an emulated Everquest sever. However, their integration still requires a human designer to supply characters and dialog to be used by the quests.

3.1.2.4 Quest generation patents

There have been several attempts to patent quest generation. Guthridge and Nahari have a patent on "Dynamic quests in game" which describes the specific quest generation technique they used in Pioneer Trail [18]. Their quest generation system used the Pioneer Trail engine to created quests based on one or more template tasks and one or more template rewards. Moore has a patent on "System for generating or using quests" which defines a specific implementation of a web app that can generate or use quests [41]. There are other patents pertaining to quests such as Farone et al.'s patent on "User created content and quests" [14] and possibly other tangentially related patents.

3.1.3 Story generators for authors

Wide Ruled is an authoring tool that allows users to plan out and generate a story hierarchy [64]. The user of the tool is able to define plot points, characters, environment, and goals/plot fragments. The tool is then able to generate a story using each of the defined elements. This tool does not explicitly support the generation of procedural stories as the user must pre-define the full goal/plot fragment hierarchy. However, it does provide a straightforward way of breaking a story down into many different fragments. Skorupski and Mateas found that students of both technical and non-technical backgrounds were able to use the tool with some level of success [64].

3.2 Believable NPCs/AI

The Turing test is an extensively researched topic that is also a primary motivation behind the creation of Panoptyk. However, since this thesis primarily focuses on quest related applications, only works directly related to Panoptyk and questing are covered here.

3.2.1 MKULTRA

MKULTRA (possibly named after but unrelated to the CIA mind control project of the same name) is an experimental game by Ian Horswill designed to explore several AI-based game mechanics [23]. Game mechanics are designed around interacting with NPCs that have generative reasoning and natural language capabilities. The player controls their character by specifying goals and actions through typed natural language. The game is a "mystery-and-detection" game where the player needs to solve secrets. Players are able to influence the behavior of NPCs by injecting false beliefs directly into the NPC's knowledge base. Characters in the game only complete requests that do not conflict with any of their beliefs. The game was intended to serve as a open-source platform for AI and interactive narrative researchers.

In a follow-up paper, Horswill addresses the successes and failures of MKULTRA [24]. The Unity Prolog interpreter built for the project ended up being utilized for several projects, including the commercial game *Project Highrise*. The performance of a definite clause grammar (DCG) parser turned out to be sufficient according to Horswill. Furthermore, NPCs characters using reactive planning were able to excel at accomplishing simple tasks typed out by players. Horswill gives the example that typing “can I have an apple?” would result in the NPC going to the kitchen, opening the refrigerator, taking an apple, walking back to the player, and finally, giving it to the player. Unfortunately, many players struggled to correctly interact with the system. Players would often type incomplete commands and/or explore the environment without interacting with NPCs. They also struggled to determine what commands were valid and wasted a lot of time attempting different sentences. The limited scope of NPC knowledge bases was also a major problem; NPCs were unable to appropriately respond to prompts or behaviors outside the scope of its limited understanding of the world. Horswill concluded that attempting to portray NPCs as lifelike characters directly conflicted with the limited ways in which the game could actually be solved. This ultimately led to player confusion and inability to solve the puzzle.

3.3 Massively multiplayer online games

Panoptyk draws inspiration from various researched-based and commercial MMOs in its aim to serve as a test-bed for addressing issues that MMO developers struggle with.

3.3.1 Neural MMO

One of Panoptyk's core goals is to serve as a platform for researching AI Agents. Similarly, Open AI's Neural MMO is a game environment tailored for reinforcement learning agents [67]. Agents are able to join servers containing automatically generated tile-based maps filled with various obstacles hindering traversal. They spawn in a random location and must obtain food and water while avoiding attacks from other agents in order to survive. Agents must compete over limited food supplies that take time to regenerate. AI Agents were trained to maximize survival time by using reinforcement learning algorithms. The researchers were able to come to various conclusions regarding agent learning and successful strategies, but made no mention of testing agent behavior with human players. Ultimately, the core gameplay of Panoptyk is quite different from Neural MMO and is tailed to test complex social interactions rather than survival mechanics.

3.3.2 SpatialOS

SpatialOS is a platform designed to manage and run massive online games in the cloud [26]. Unlike many MMOs that host different copies of a game world on many different servers, SpatialOS is designed to have a single continuous game world hosted by multiple servers. This is a dramatically different approach that directly addresses issues caused by instancing. The engine supports a persistent database that all servers can access and change. Severs can be written to handle a specific part of the overall game world; this allows servers to be responsible for specific in-game regions. The design of Panoptyk draws upon the use of a continuous game world made up of regions hosted by separate servers.

Chapter 4

DESIGN

4.1 Panoptyk Engine

The original Panoptyk Engine required a re-design to function as an effective research platform [47]. Our design is inspired in part by Jeremy Bentham's Panopticon[1] in which anyone may or may not be observed. This notion spurred Panoptyk's information driven world where we establish how all inhabitants naturally generate new information which is in turn stored and catalogued by the engine and made to be observed and shared by other game agents.

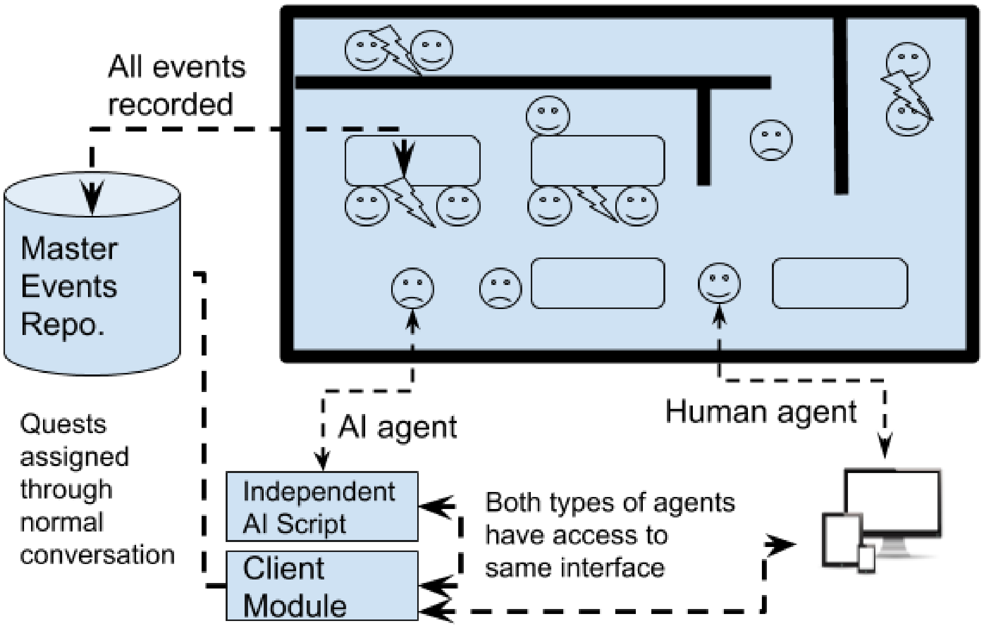

All events recorded
Master Events Repo.
Quests assigned through normal conversation
AI agent
Human agent
Independent AI Script
Client Module
Both types of agents have access to same interface

Figure 4.1: Planned design of Panoptyk

[1] see: https://en.wikipedia.org/wiki/Panopticon

4.1.1 World representation

Panoptyk represents its world as a single reality without instancing or channels used in other MMOs. Most interaction spaces in a typical game built with the engine will have maximum capacity for occupancy, providing a limitation to address networking issues grounded in the virtual world itself. While adhering to a location's capacity it is feasible for any number of players to congregate, and for those meetings to impact agents around them. Our solution draws from the real world, where human traffic and occupancy have a measurable effect for situations such as waiting to enter a crowded restaurant, or not being able to secure transportation for a popular vacation destination. Taking this a step further, different servers are synonymous with different cities rather than creating a clone of the world with its own disjoint inhabitants. Traveling between locations means migrating the character and associated computation to a new server. Following similar occupancy rules, cities will have points of entry and those points can block off entering agents if a maximum occupancy is reached in that server/city. Our design decisions strive to create a more immersive virtual world that is made up of interconnected locations. When expansion is required, adding a new server by creating a new city allows for organic growth as more players fill the world.

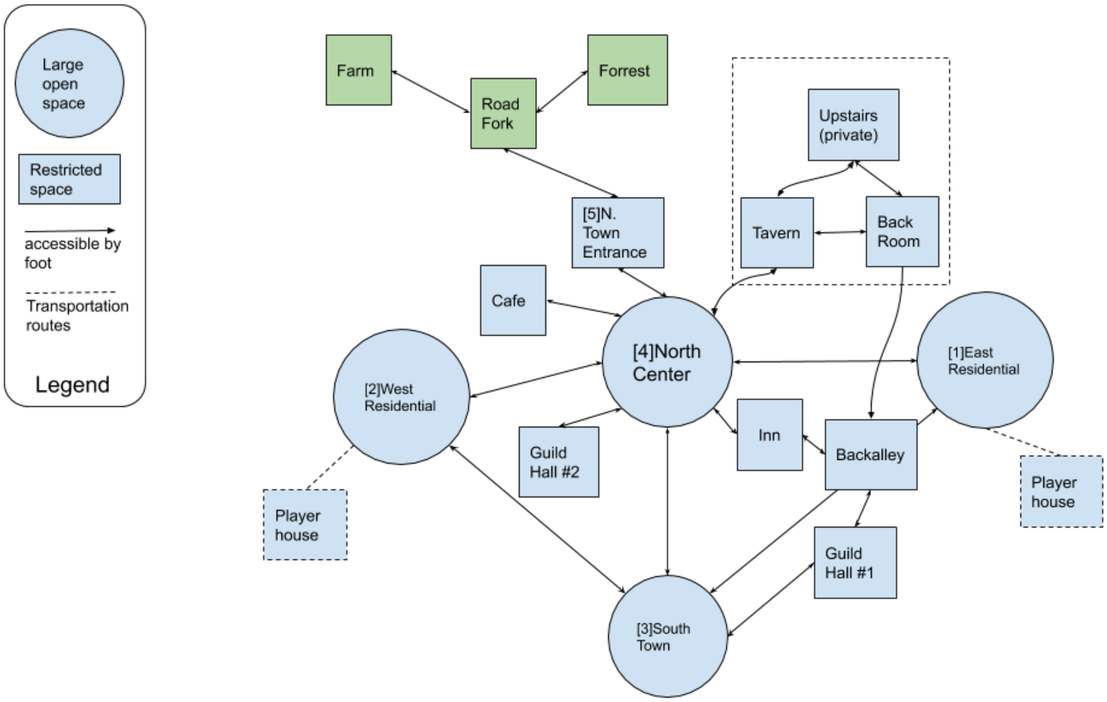


Figure 4.2: Example world node representation

Figure 4.2 shows an example world representation where large circles represent hubs with many square rooms interconnected. Only the rooms offer a potential for every agent to observe every action of other agents. The rooms will have a maximum occupancy to address hardware limitations. A similar network can be constituted on a new server with transportation links connecting the two localities.

4.1.2 Panoptyk agents

Panoptyk presents a novel way of approaching the bootstrapping issue seen in MMO games. By treating NPCs and players the same in the world, the initial launch of an MMO using the Panoptyk Engine can seed its world with an initial population of believable NPCs, a portion of whom are to be supplemented by eventual human players. In order to treat human players and NPCs as equivalent agents, the engine makes no differentiation between human and AI clients that connect to it. The engine also does not support actions that would be obvious indicators that could separate human behavior from NPC behavior, such as in-game chat and free movement for agents. The solution we have outlined also assists in creating a platform for believable AI agents in our engine.

4.1.2.1 World perception

All agents have access to the exact same information in the world; NPCs do not receive any special map or gameplay information. Much like human players, NPCs in Panoptyk will be remote clients, connecting to the server and executing all code client-side. The server informs agents of their surrounding environment and any exits leading to other locations, as well as all of the other agents in the current room and any conversations occurring. It also conditionally informs agents of actions/events caused by other agents in the room if the server determines those events are publicly observable. For example, all agents in a room can observe when 2 or more agents start talking to each other, but only those in that conversation or agents with the ability to eavesdrop can see what they are saying in the conversation.

4.1.2.2 World interaction

In order to prevent obvious human-like behavior or timing-based movement profiling, the server does not support tracking in-room movements of agents. The Panoptyk server only tracks actions that directly affect other agents or game objects, such as leaving/entering a room and interactions between other agents or items. The communication between agents is also limited so that agents can only communicate through an enumerated list of requests and by asking questions or telling about in-game actions in the same format they receive them; unrestricted chat is strictly prohibited.

4.1.3 Information driven gameplay

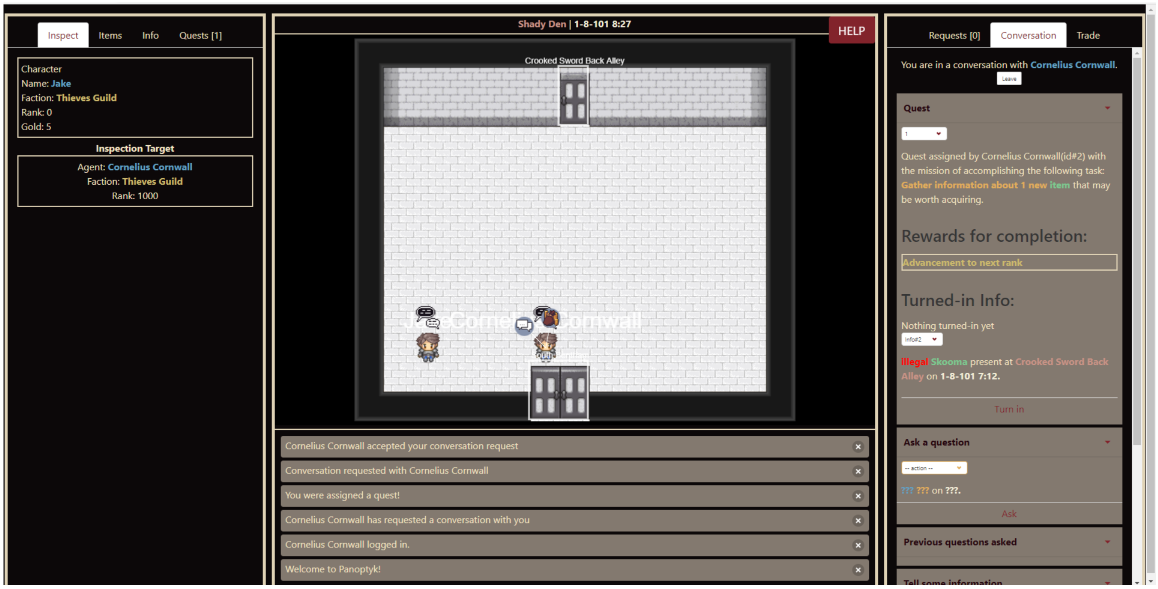
All interactions in Panoptyk result in information being exchanged. This creates an environment where an agent must carefully decide on what actions to take. An agent asking another agent a question or requesting something from them will result in new information being gained by the opposing agent. The second agent may then inform other agents about what the first agent communicated with it. Even more indirectly, agents with enough perception ability, and within observable range of any action or conversation that occurs, could gain information about such events which they could use to achieve their own goals. Information is also used as an in-game commodity; agents can use information as an item during in-game trades. The use of information as a commodity gives it a tangible financial value along with its inherent strategic value. This platform of constant information generation and its use as a commodity and game mechanic has numerous implications. First, with information at the core of our engine we can think of it as a knowledge base for AI agents to greatly enhance their awareness of the world and ability to act in more complex and robust ways. Secondly, this brings our platform closer to a model of the real world information economy which we hope allows our engine to be a good platform for exploring real world phenomena such as logical deduction, critical thinking and so-called "fake news." Lastly, by intentionally making agent actions the catalyst of new information production in our world, we create a chaotic source of content for story lines, quests and puzzles. This should reduce the recurring maintenance of MMO content creation, ultimately opening the genre to indies and researchers lacking resources.

4.1.4 Quest support

Quests are not internally generated by the Panoptyk engine, instead they are represented as information items that are created in standard agent interaction. It is intended that any agent can assign quests, but that other agents will only take these tasks seriously if they are assigned from a guild/faction leader or contain a substantial reward. We envision that future versions of Panoptyk will have human players that step into managerial roles of their faction and assign quests for newer members.

4.1.5 Human client

The Human Client for Panoptyk was designed to provide a clean user interface and visual representation of actions happening on the Panoptyk server. All visuals in the Human Client are independent from the server; the Panoptyk server does not have a visual representation of any action or object in the game. Human players are capable of seeing other agents and items within their room, but the actual placement of everything is randomized and different for every client. The goal of this early client was to provide humans players with enough tools to play the game without taking away development time from other aspects of the Panoptyk server or quest generation.



Character
Name: Jake
Faction: Thieves Guild
Rank: 0
Gold: 5

Inspection Target

Agent: Cornelius Cornwall
Faction: Thieves Guild
Rank: 1000

You are in a conversation with Cornelius Cornwall.

Quest assigned by Cornelius Cornwall(id#2) with the mission of accomplishing the following task: Gather information about 1 new item that may be worth acquiring.

Rewards for completion:

Advancement to next rank

Turned-in Info:

Nothing turned-in yet

illegal Skooma present at Crooked Sword Back Alley on 1-8-101 7:12.

??? ??? on ???.

Cornelius Cornwall accepted your conversation request
Conversation requested with Cornelius Cornwall
You were assigned a quest!
Cornelius Cornwall has requested a conversation with you
Cornelius Cornwall logged in.
Welcome to Panoptyk!

Figure 4.3: Panoptyk human client

The complexity of the game required 7 different tabs in the UI. The "Inspect" tab provides players information about their faction, rank, and gold, as well as information about any in-game object they click. The "Items" tab lets players see what items they have, details about those items, and the ability to drop those items. The "Info" tab shows every action, question, or command info pieces that the player has witnessed or been told; it also allows players to filter those info items by type, action, agent, item, or location. The "Quest" tab shows the player all active and completed quests related to them as well as extra information about those quests such as rewards and reasons for why the quest was assigned. All trade and conversation requests are displayed in the "Requests" tab. The "Conversation" tab is only active when the player is in a conversation, it lets players turn-in quests, ask questions, and tell info. Finally, the "Trade" tab allows players to trade by offering or requesting gold, answers to questions, or items. A "Help" window added after early testers had difficulty navigating the UI. The "Help" window contains instructions on how to do various basic tasks within the game, as well as tips for the user's current quest. See Appendix A for full examples of the Panoptyk Human Client.

4.2 Quest scenario

A complete scenario was created for the purpose of testing quests on the Panoptyk engine. The scenario takes place in the medieval-esque town of Bentham. The town is filled with characters that can be part of the Bentham Town Guard, the Thieves Guild, or no faction. A combination of legal and illegal items are placed throughout the town to act as a catalyst for conflict and trade.

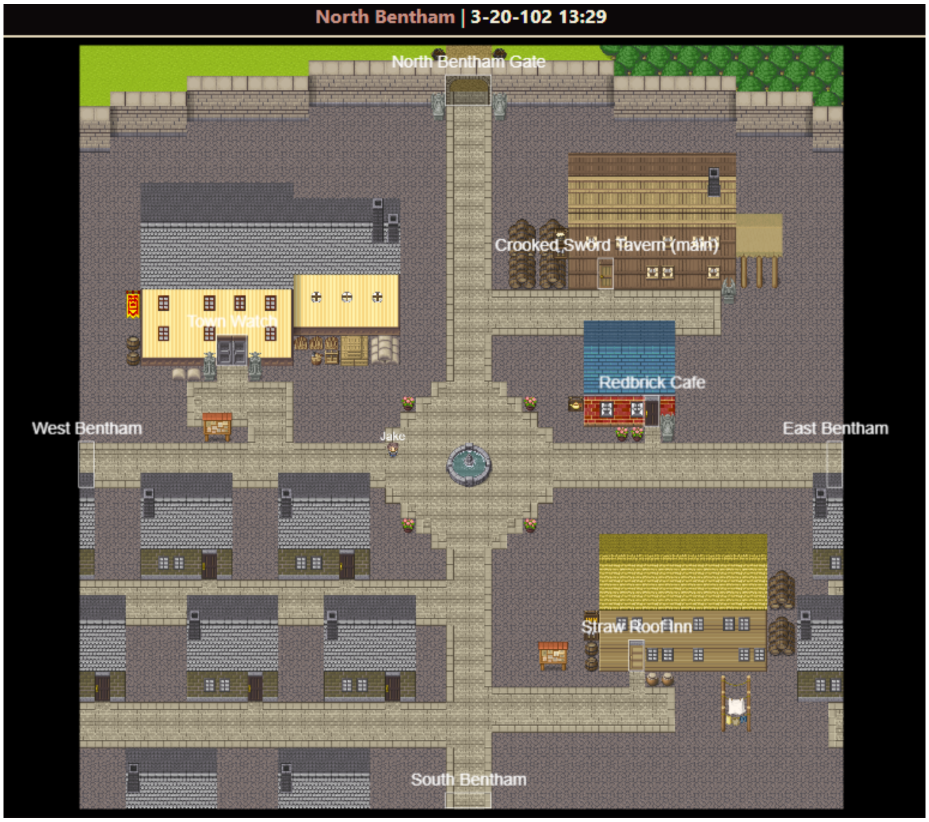


Figure 4.4: Downtown Bentham

4.2.1 Quest generation

In this early version of Panoptyk, pre-programmed NPC faction leader agents assign appropriate quests to their faction members in return for rank advancement and gold. These tasks align with their faction's goals and are reactive towards current events in the game. Since quests are designed around the agent-generated information, quests can be dynamically generated as long as new information is being created by players. It is intended that using players' actions to determine quests will increase their personal connection to quests and reduce the repetitiveness often noticed in other MMO quest systems.

4.2.2 Town Guard faction

The town guard faction leader constantly assigns quests to combat illegal activity around the city. First it assigns quests to arrest agents that have committed crimes. After assigning quests to apprehend all active criminals the town guard leader will assign quests to collect known contraband and deliver it for safekeeping. Finally, if no crimes or illegal items are detected, it will assign a quest for town guard agents to report any illegal actions or items.

4.2.3 Thieves Guild faction

The thieves guild leader is primarily concerned with the collection of illegal goods for itself. The first priority of thieves guild leader is to assign quests to "take revenge" against members of the town guard who have recently arrested members of the thieves guild. The "revenge" quest currently just involves attacking the responsible guard, but would ideally involve more clandestine actions once additional gameplay has been added to Panoptyk. Once all possible "revenge" quests have been assigned the thieves guild leader focuses on assigning quests to loot the most valuable items that it knows about. Occasionally, it may decide to assign a quest to reward an agent that have been helpful to the faction. Finally, if it cannot create any other valid quest from its current knowledge it assigns a quest to search for new treasures to acquire.

4.2.4 Player input to determine followup quests

When possible, both faction leaders prioritize assigning quests that are based on information directly relevant to the agent it is talking to. For example, revenge quests for the thieves factions are always assigned to the agent that has been arrested by the targeted member of the town guard. The same logic is used when assigning item retrieval quests; the questing agent is informed it is being asked to retrieve the item because it is the one that told about the whereabouts of that item. This rule is not absolute though, factions leaders will assign these followup quests to other faction members if it has no other available quests and the preferred agent is busy with another quest or otherwise unavailable. Ideally this system is intended to give players the impression that quests are based on their personal experiences.

4.2.5 NPC agent design

Since AI agents in Panoptyk must create a cohesive narrative while reasoning independently, it was considered important they act appropriately given their preassigned personality, their goals, and any events occurring around them. This involved agents determining appropriate behavior for themselves, or in the case of a faction leader creating appropriate quests for every member of the faction. Originally, we had AI agents make their decisions based on their previous experiences and information available to them. Unfortunately, it was quite difficult to apply most of the stored information into meaningful gameplay. Agents had no way of communicating their hostility or friendship to other agents. Agents who did not like each other could only refuse requests from each other but had no way of differentiating a hostile refusal from an automatic refusal given when an agent was busy. This mostly resulted in the uninteresting behavior of hostile agents completely ignoring each other. Eventually it was determined that, for the sake of testing quests, AI agents should only act on information that was directly applicable to their current task/quest. The AI agents retain behavior that fits their faction's personality and it is hoped that future versions of Panoptyk will allow agents to convey emotion through their actions.

Chapter 5

IMPLEMENTATION

5.1 Pantoptyk Engine

Significant work was done on the Pantoptyk MMO Engine to make it capable of running as a platform for AI and games research. The first version of the Panoptyk Engine was created by Nathan Philliber as a senior project [39], but that version was abandoned in favour of the current more scalable implementation in typescript. The current implementation of the Panoptyk Engine was developed by Sean Mendonca and Mitchell Miller to meet the requirements outlined in the design section. Additionally, Kaito Trias and John Potz provided some contributions towards the example client.

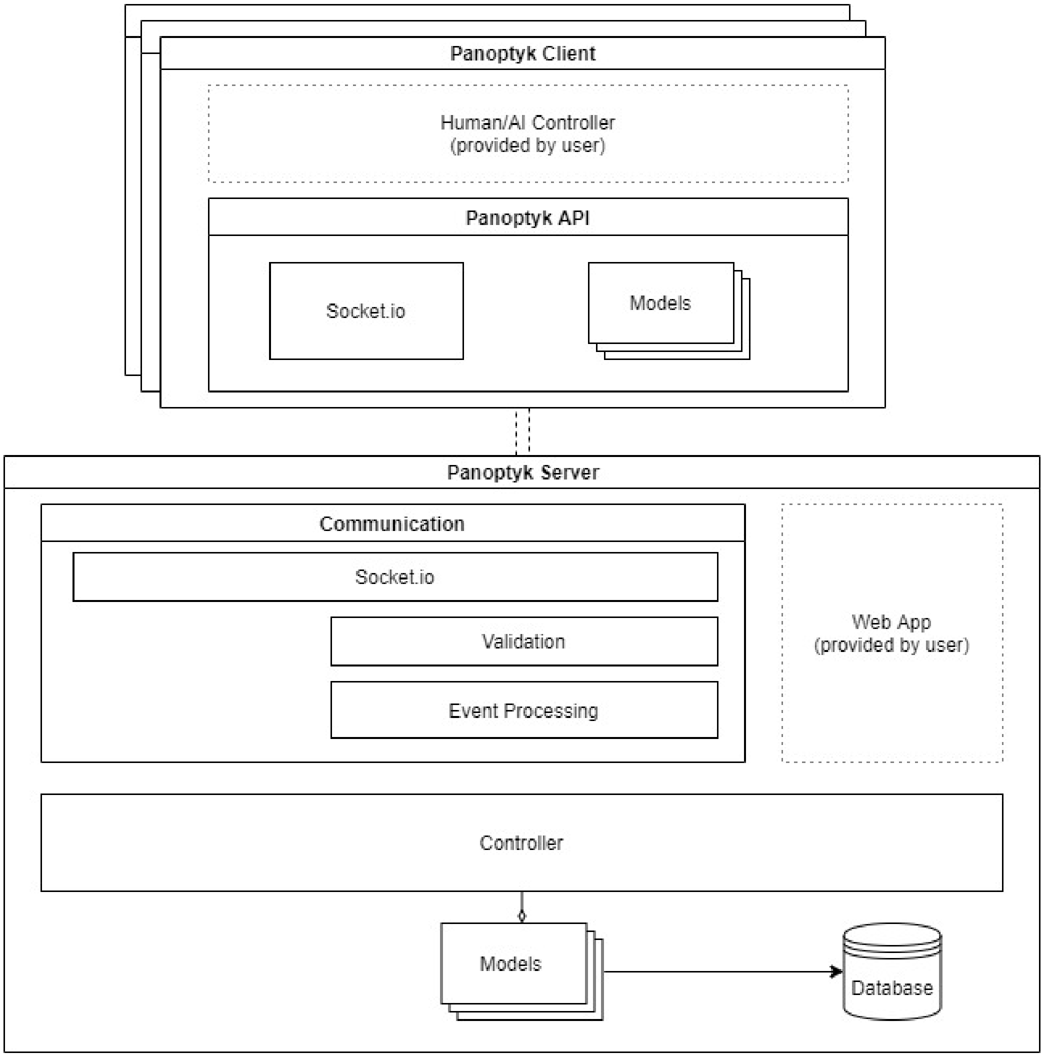


Figure 5.1: Architecture of Panoptyk

5.1.1 Server

A Pantoptyk game server is intended to be light-weight and mostly focused on enforcing rules and storing information. We decided to implement the server using Node.js, as it appears suited to support the planned features of Panoptyk. Node.js is designed to offer strong scalability along with an asynchronous event driven system. Socket.IO is used to provide bi-directional communication between the server and clients. On the server side of our engine, there is not really any constant update routines to run because agents drive the interactions with the world and other agents.

Because NPCs are run in a decentralized fashion, the server needs to only service requests from agents and ensure both the validity of the action and the validity of the data it holds about the world. This includes all information produced along with ownership and access of the information. Other world data is also kept coherent, for example agent locations, agent states, and interactions between agents. All of the checks on validity are set up to be triggered on a message from any client. The server then decides if it should follow through with the client's request. If so, Event Processing calls the Controller, which adjusts the server's models and disseminates updates to the affected clients. Node.js deals with large loads of asynchronous requests well in our initial testing. Constant updates of the gamestate for every agent can be ignored because of our limitation on player location and movement. The engine's only external task is informing the involved agents when an action occurs. Each client runs its own update loop in which it asks the server for world updates. The server sends current state values for the small portion of the world that the agent can perceive.

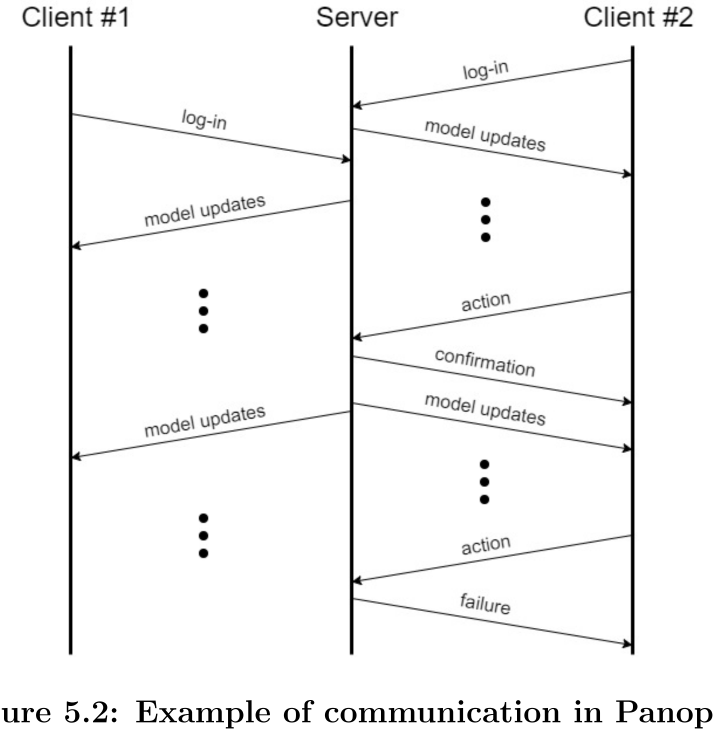


Figure 5.2: Example of communication in Panoptyk

5.1.2 Gameplay models

The models essential for the gameplay of Panoptyk are outlined in Figure 5.3. The model representation is the same on both the server and client modules of Panoptyk. All models extend the IDObject class which provides inbuilt functionality for tracking model instances and serializing model data for storage or transmission. Models refer to instances of each other by id, but this abstraction is hidden from the user in private fields; models automatically handle the retrieval of any other models they reference. Additional functionality for models is present on the server to handle the modification and storage of data.

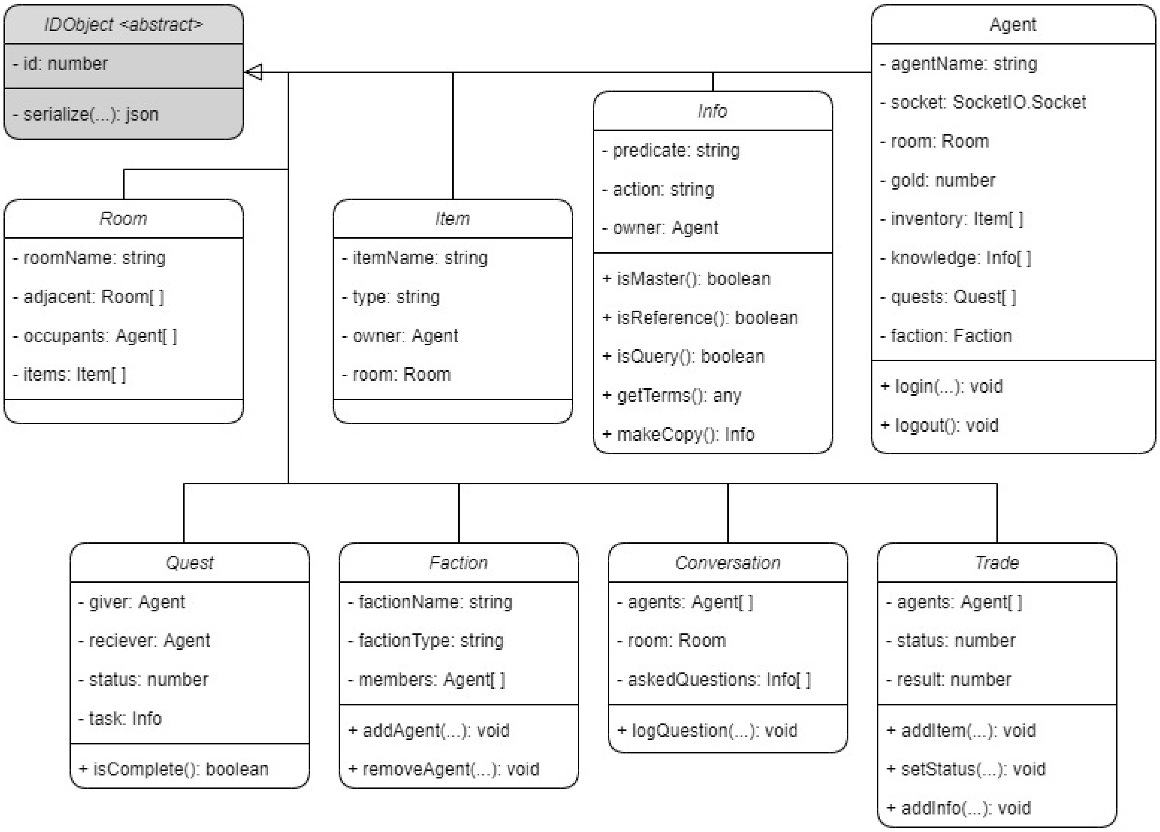


Figure 5.3: Important gameplay models in Panoptyk

5.1.3 Information representation

In general, information representation in Panoptyk is based on first-order logic predicates. This representation is designed to make it as easy as possible for NPC agents to reason about the data they receive. All events that occur in the world generate an information object. These objects link the action performed with the relevant variables (agent(s), location, time, item(s), etc.). Every possible action (shown in table 5.1) is codified and new actions specific to other games can be supported easily.

Table 5.1: Current list of actions

Action	*Predicate Variables*
Move	(Time, Agent, Location, Location)
Pickup	(Time, Agent, Item, Location, Quantity)
Drop	(Time, Agent, Item, Location, Quantity)
Steal	(Time, Agent, Agent, Item, Location, Quantity)
Pay	(Time, Agent, Agent, Location, Quantity)
Arrest	(Time, Agent, Agent, Location, Information)
Assault	(Time, Agent, Agent, Location, Information)
Converse	(Time, Agent, Agent, Location)
Gave	(Time, Agent, Agent, Item, Location, Quantity)
Ask	(Time, Agent, Agent, Location, Information)
Tell	(Time, Agent, Agent, Location, Information)
Assign Quest	(Time, Agent, Agent, Location, Information)
Fail Quest	(Time, Agent, Agent, Location, Information)
Complete Quest	(Time, Agent, Agent, Location, Information)
Show Possession	(Time, Agent, Item, Location, Quantity)

The server keeps a master copy of every piece of generated information, all information distributed to clients are a reference to the master server copy. This is done to both save memory and to allow the server to keep track of information that is based on the same event. Because of this system, clients are never given duplicate copies of the same piece of information. This is an essential feature for information types that reference another piece of information, as it would otherwise be computationally expensive to figure out what information pieces are referring to the same thing. When client attempts to access an information piece the API automatically pulls the relevant fields from the master copy.

Panoptyk supports the the usage of partial information, information where one or more fields are masked to its owner. Masked information still references the original server master copy, but the masked fields are not sent to the client; this prevents clients from somehow extracting information they should not have from the master copy. The information system gives AI bot clients sufficient details to function without the server's intervention, thus maintaining our decentralized system that allows external agents to shape the game world.

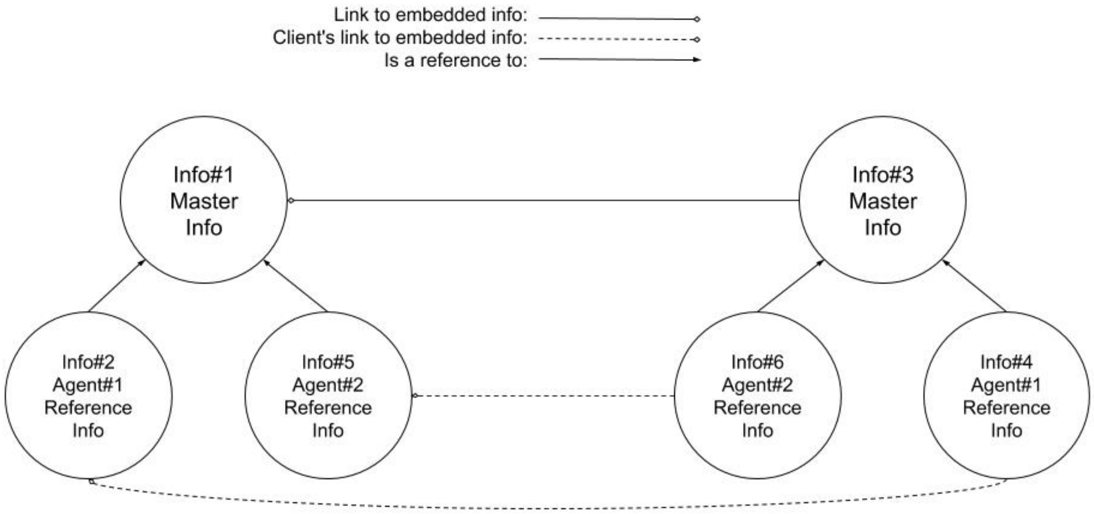

Link to embedded info:
Client's link to embedded info:
Is a reference to:
Info#1 Master Info
Info#3 Master Info
Info#2 Agent#1 Reference Info
Info#5 Agent#2 Reference Info
Info#6 Agent#2 Reference Info
Info#4 Agent#1 Reference Info

Figure 5.4: Information references example

5.1.4 Client API

The client API provides the means for agents to communicate with the server. One of the primary goals of Panoptyk is to allow bots to easily interface with the game without having an information or ability advantage over human players. Both human clients and bot clients communicate with the server using the same client module API. The client module uses an asynchronous callback system to inform connected clients about changes in the world as well as results of attempted actions. Like the server, the Client API uses Socket.IO to handle all communication with the server. It also stores object information based on the server's representation of agents, items, rooms, and information. The server has a validation system to ensure that a doctored client cannot cheat.

5.2 Quest scenario

The quests for this scenario of Panoptyk are dynamically generated from information predicates using previous history, current knowledge of the game world, faction personality, and the quest giver's disposition towards other characters. To accomplish this, the scenario required a strong implementation of AI agents that were able to react appropriately to any possible situations.

5.2.1 Structure of AI agents

The Panoptyk AI agents used for this scenario were designed as finite-state machines (FSM). Initially, all agents were coded as a single-layer FSM, but this resulted in overly complex FSMs that were hard to follow and contained states that could not be easily be generalized and reused in other agents. Once additional developers started working on AI agents, it was decided that a multi-layer FSM would reduce development time for new agents by generalizing complex behaviors that could be reused. For example, something simple like telling a target agent a piece of information is a multi-step chore even if we assume they are in the same room as the bot; the bot has to submit a request to leave a current conversation (if they are in one and it does not contain the target agent), wait for that action to succeed (repeating it or giving up on a failure), submit a request to the target agent if they are in the room (again repeating or giving up on failure), wait for the target to accept their conversation(possibly giving up after a period of time), and finally, submit a request to tell info to the target (again repeating or giving up on failure). These transitions had to be present in any state which required the bot to tell a specific agent something. It was infeasible to generalize behavior like the previous example because multi-step actions had to be able to transition in reaction to events outside of the context of their immediate goal. In other words, single-layer FSMs struggled to accomplish multi-step asynchronous tasks while remembering "the big picture."

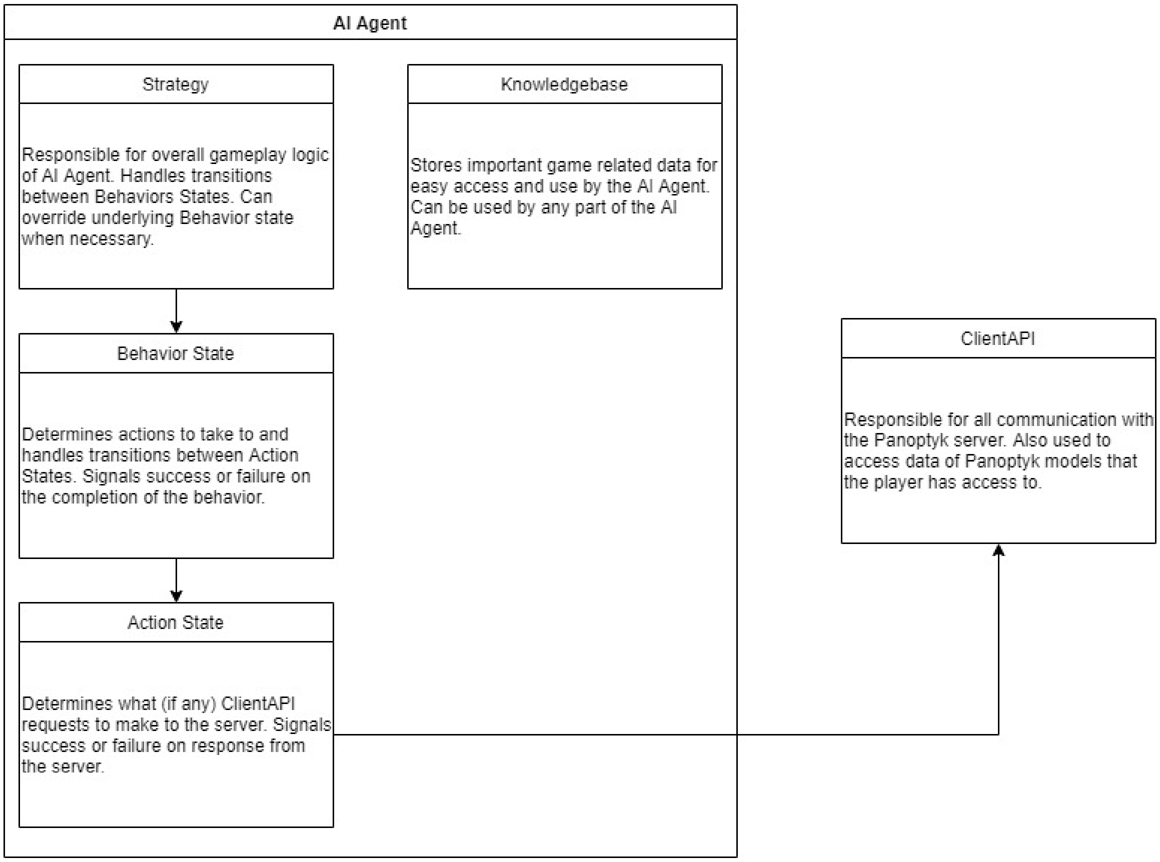

AI Agent
Strategy
Responsible for overall gameplay logic of AI Agent. Handles transitions between Behaviors States. Can override underlying Behavior state when necessary.
Knowledgebase
Stores important game related data for easy access and use by the AI Agent. Can be used by any part of the AI Agent.
Behavior State
Determines actions to take to and handles transitions between Action States. Signals success or failure on the completion of the behavior.
ClientAPI
Responsible for all communication with the Panoptyk server. Also used to access data of Panoptyk models that the player has access to.
Action State
Determines what (if any) ClientAPI requests to make to the server. Signals success or failure on response from the server.

Figure 5.5: Example bot

The multi-layer FSM fixed the issues with the single-layer FSM by abstracting "big picture" strategy into its own layer. An Action State represents a single action request to the server while a Behavior State represents an FSM of Action States. By extension, a strategy represents an FSM of Behavior States. This allows us to successfully separate high-level goals of the AI agents from the monotonous low-level actions needed to accomplish them. See Figure 5.5 to see how AI Agents use this to interact with the game.

5.2.2 Knowledgebase of AI agents

The knowledgebase of AI Agent agents is used to store data that is not contained on any Panoptyk models or only relevant to a specific Behavior State. These knowledgebases include room connections, information processing for quest or personality related actions, and stored opinions of other agents. As part of each agent's "personality" the knowledgebase is responsible for processing all incoming agent actions and deciding how those actions affect their opinion of the agents doing the action.

Agent opinions of actions are constant integers that are coded into each AI agent. If there is no secondary agent involved in an action, then the change in opinion is based entirely on the action itself.

$$\Delta opinion(agent) = opinion(action) \tag{5.1}$$

If there is a secondary agent involved in the action then it means that agent has been targeted by the action. The change in opinion then becomes based on whether a positive or negative action is being performed on a liked or disliked agent. For example, an agent performing a negative action on a disliked agent would result in a positive change in opinion.

$$\Delta opinion(agent) = opinion(action) * opinion(otherAgent) \tag{5.2}$$

Default opinions towards other agents are integer values that start negative for agents of opposing factions, zero for unaligned agents, and positive for agents in the same faction. The knowledgebase also saves references to actions that cause a significant change in opinion. This allows AI agents to refer to actions that they especially liked or hated. Currently the system is only used to select targets for “gift quests,” but is in place to create more personalized quests and interactions between agents once Panoptyk supports more ways to communicate and express emotion.

The knowledgebase is also responsible for assisting with navigation through its memorization of visited locations. The knowledgebase performs an A* search of room connections to determine the shortest path to a destination. A* search finds an optimal path by expanding connections that minimize the distance traveled from the starting room and remaining distance to the destination.

$$f(n) = g(n) + h(n) \tag{5.3}$$

Once the destination is found, the algorithm is able to return the shortest possible path to the destination. When there is not enough room data, AI agents resort to random movement to reach their destination.

5.2.3 Quest generation

Quests are represented in special information pieces that signify a goal to complete. Quests are generated based on the current information knowledge of the quest giver and their unique pre-defined goals. Quest generation for the two playable factions in the scenario differ due to the different goals of the factions. Quest assignment logic for factions is shown in Algorithm 1 and Algorithm 2.

```
Data: target agent to give quest
Result: valid quest to give to target agent
if an opposing agent has done an action to harm target agent then
    return a revenge quest targeting the opposing agent that refers to the
      harmful action done to target agent;
else if some other agent's action(s) have pleased the faction leader then
    return a gift delivery quest targeting the other agent that refers to how
      they pleased the faction leader;
else if there is a valuable item that we don't own then
    assign item retrieval quest that refers to last known information about
      that item;
else
    assign generic quest to discover new items in the world;
end
```

Algorithm 1: Thieves Guild quest assignment logic

```
Data: target agent to give quest
Result: valid quest to give to target agent
if there is a crime that has gone unpunished then
    return an arrest quest targeting the criminal and referring to the crime
      committed;
else if there is an illegal item that is not currently impounded by the Town
  Guard then
    return an item retrieval quest that refers to last known information about
      that item;
else
    return a quest to patrol the town and report any illegal actions;
end
```

Algorithm 2: Bentham Guard quest assignment logic

Chapter 6

VALIDATION

In this section the validation of results in the thesis is discussed. The first stage of validation was a internal review of results generated by the project. After that, a user-study was conducted to evaluate the public's perception of quests.

6.1 Internal review

Before user testing began, we ran multiple internal tests to make sure that the questing system was working as intended and that the AI faction members were acting in a manner befitting their current situation and pre-programmed faction personality. Given the unique information system of Panoptyk, the majority of our early testing was focused on finding ways to have new information flow to faction leaders. This was essential because unlike most games, quests in Panoptyk have to be created entirely from an agent's personal knowledge. Giving faction leaders global knowledge would have made the discovery and trading of information (the core premise of Panoptyk) entirely pointless.

6.2 User study/survey

Three user studies were carried out to validate the quality of generated quests. The questions and format of the study changed slightly between each iteration to ameliorate issues as they appeared. In all studies, each subject was placed on either the Bentham City Guard or Thieves Guild factions. Subjects were expected to play through a few quests and then answer a survey on their experience. All Versions of the survey consisted of five different sections to evaluate multiple parts of Panoptyk and the scenario. Free-response questions are grouped into categories of expected answers.

The first section of the survey asked for basic demographic information on the participant's chosen major and experience with video games and video game features related to quests. The questions in this section were designed to correlate any possible source of bias that could appear in answers to subsequent questions. It particular, this section focuses on the participant's experience and opinion of quests and procedurally generated quests.

The second section of the survey focuses on general feedback towards their experience with the Panoptyk game itself. Participants are asked about their overall enjoyment and to list what the game's strengths and weaknesses were. They are given the option to input additional feedback to explain their choices. This section is used to determine if any other parts of the game hampered the participant's experience with quests.

The third section covers the essential topic of quests. This section saw moderate revision between the three studies, but all have the same base questions. Participants are first asked how many quests they were assigned and how many they completed to determine the eligibility of the rest of their answers. They are then asked about how difficult they thought quests were to see if that had an effect on the rest of their answers. Immediately after, the primary questions of this section ask users to rate their agreement on statements measuring quests based repetitiveness, clarity, consequences, meaningfulness, and relatability to in-game events.

Table 6.1: Example responses for each classification of the estimated generation method free-response question

Class	*Example Response*
Based on world events	The quests were based on events happening in the world.
Based on personal actions	The quests were based on things I did.
Random	The quests were randomly generated from a template of valid quests.
Other	The quests were generated based on the items in the world

Table 6.2: Example responses for each classification of interesting quest encounters free-response

Class	*Example Response*
Arrest quest	I arrested a criminal after finding evidence of a crime they did.
Revenge quest	I took revenge on a Town Guard member for arresting me.
Other valid quest	I had to do a complicated trade to obtain an item for my quest.
Unintended/Bug	I was able to get infinite money by repeatedly stealing and selling items!

The fourth section asked participants questions on their experience with the faction they were in. This was used to separate user data by faction and to see if quest experiences were different across the two factions. Participants were also asked if they thought the quests they received made sense given their faction archetype; quests that felt out of place could have potentially harmed player immersion and resulted in a more negative perception of quests.

Table 6.3: Example responses for each classification of the item knowledge free-response question

Class	*Example Response*
The player's actions	The faction leader knew about the item because I told him about it.
Another player's actions	Someone probably told him about it.
Other	It was programmed into his memory.

The final section queries participants about the overall impact they felt their actions had and if other agents had a significant impact on their experience. Given the multiplayer dynamics of the game, it was possible for players to greatly affect the experience of other players. While this section has some overlap with quests, it allows participants to describe interesting scenarios that emerged independently from quests. This section also asks if participants had enough in-game options to make gameplay engaging, something that was judged to be a potential shortcoming of the current gameplay.

Table 6.4: Example responses for each classification of the advantages of player input influencing quests free-response

Class	*Example Response*
Personalization	I felt like my actions were relevant to the quests.
Replayability	I never ran out of interesting quests to do.
Other	There were no advantages.

Table 6.5: Example responses for each classification of the disadvantages of player input influencing quests free-response

Class	*Example Response*
Repetitiveness	All the quests felt the same.
Lack of depth	The quests were fairly shallow.
Vagueness	The objectives were vague and hard to complete
Other	I hated having to always search for new information to get a new quest.

Table 6.6: Example responses for each classification of actions that caused players to treat each other differently free-response

Class	*Example Response*
Quest interference	I picked up an item and Chris wanted to trade me for it.
Quest consequence	I was arrested for picking up skooma.
Other	Unknown

Table 6.7: Example responses for each classification of the gameplay actions free-response question

Class	*Example Response*
Sufficient actions	The game had enough actions.
Insufficient actions	I did not have enough actions to do anything useful. blah blah blah would have made it 1000% better.
Other	I could not figure out how to play.

6.2.1 First study

The first study was offered to a college class of Interactive Entertainment Engineering students. The testing environment was chaotic due to multiple groups of students trying to get their games tested. Subjects were given basic instructions to follow the quests issued by their assigned faction and attempt to complete them to the best of their ability. Unfortunately there were several technical and design issues that hampered the testing session. This was the first attempt to have more than three humans testers connected to the scenario at the same time. These issues meant almost all users struggled to play the scenario.

6.2.2 Results of first study

The participants of the first student were all members of a class on Interactive Entertainment Engineering, and were either computer science or software engineering majors. All participants agreed they had played games where they have had to complete quests. The majority of participant claimed they enjoy games where quests are given, with only one participant disagreeing. 5 out of 7 participants agreed that most games they played had a good quest system while the remaining two were neutral. On the other hand, only 2 participants disagreed that they had been disappointed by quests in a game. Regarding experience with procedural quests, two participants did not have experience with them, one participant was unsure, and the remaining four did have experience with procedural quests. Of the participants that had played procedural quests, two found them enjoyable and two did not.

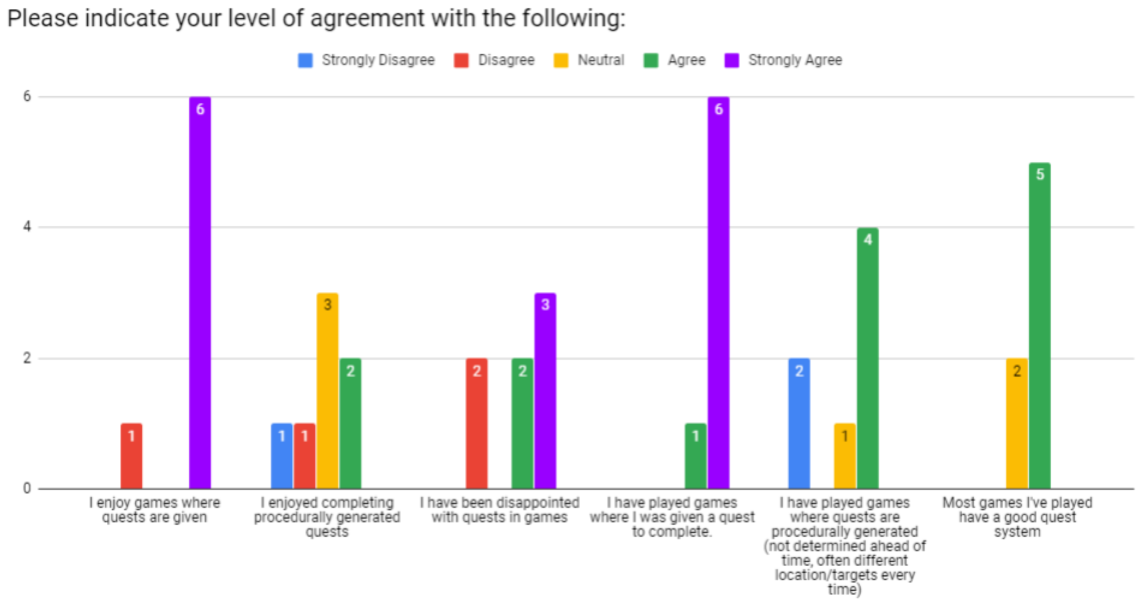


Figure 6.1: First study screening

Issues with the UI meant that most users were not able to provide much feedback on quests. Only 2 of the 7 testers claimed that playing was an enjoyable experience. 6 out of 7 testers claimed that the UI was a weakness of the game. In a more positive note, 5 out of 7 testers claimed that the information system was a strength of the game, which is important as the information system is core to the Panoptyk engine. In terms of quests, only 3 of the 7 testers were able to complete any quests; only one of them was able to complete more than one quest. Unsurprisingly, 6 out of 7 testers to describe quests as “a little hard” or “way too hard” and resulted in mostly neutral or negative responses for most questions related to quests. Fortunately, the player who was able to complete the most quests had extremely positive feedback for the quests. Feedback on the faction section showed that only 1 of the 7 testers thought that assigned quests did not make sense for their faction identity. 5 of the 7 testers felt that their actions had an impact on other players in the game, multiple players noted that other players could pick up items required for their quest. Three players felt that there were not enough actions in the game, while another three felt that the actions were too complicated or hard to use.

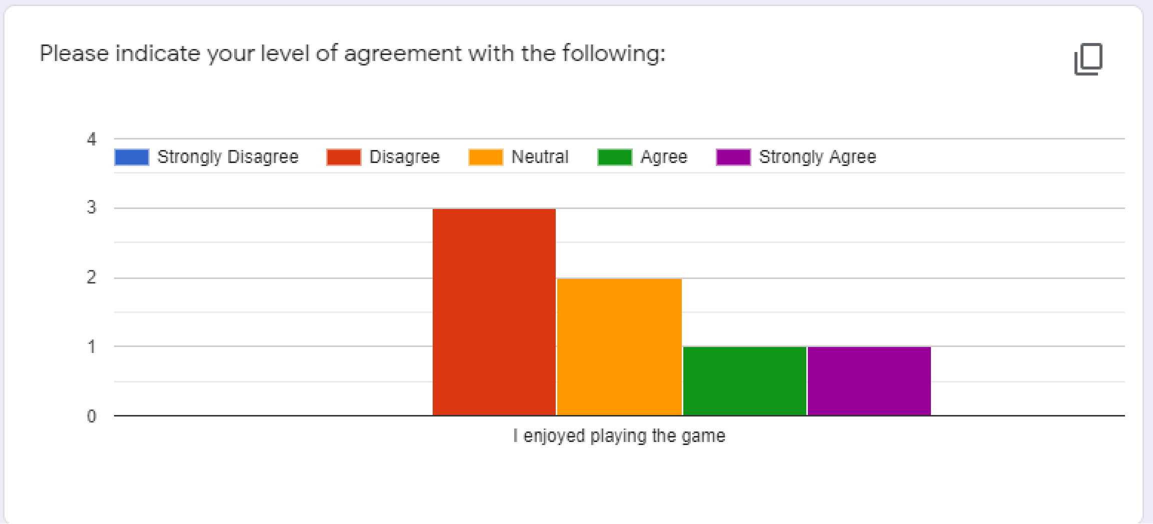


Figure 6.2: First study overall enjoyment

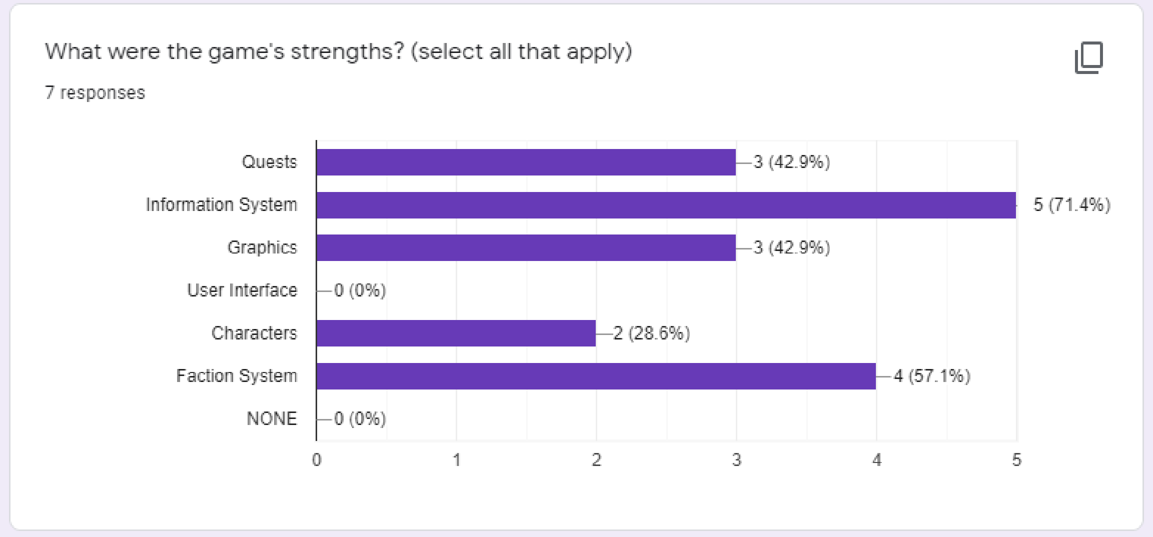


Figure 6.3: First study strengths

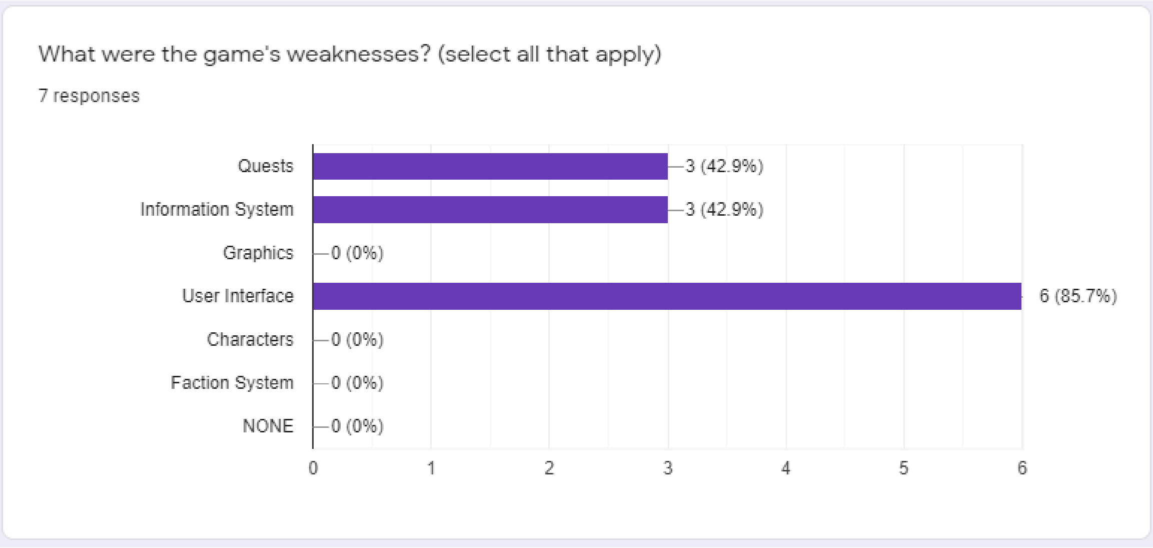


Figure 6.4: First study weaknesses

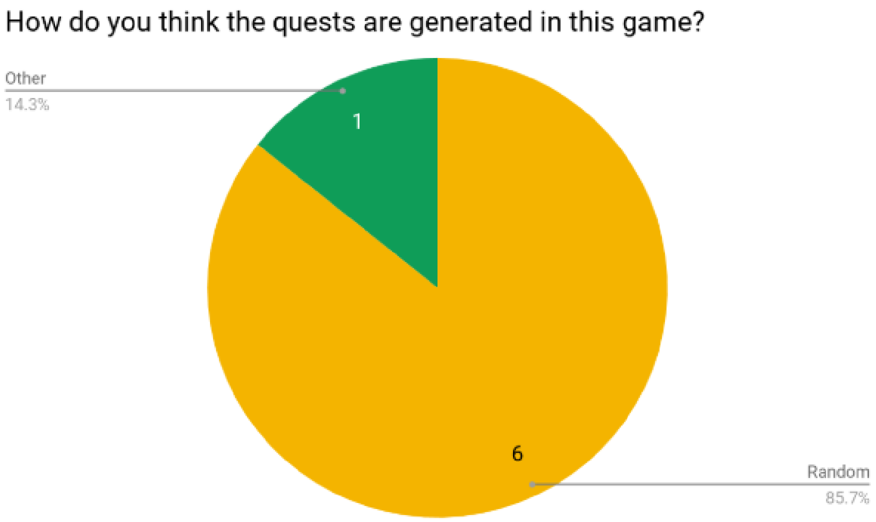


Figure 6.5: First study free-response on quest generation method

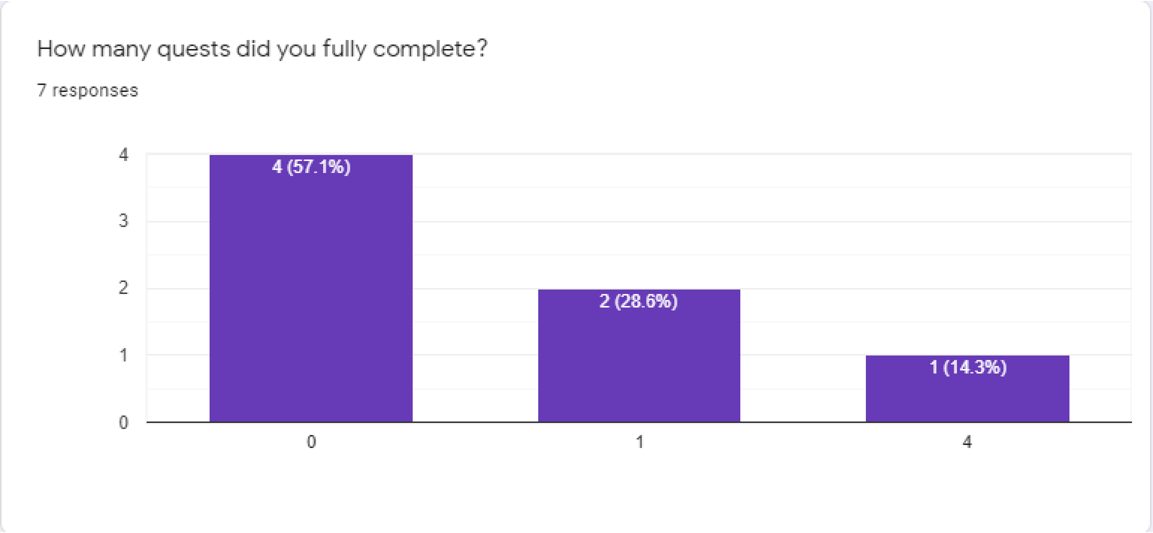


Figure 6.6: First study quests completed

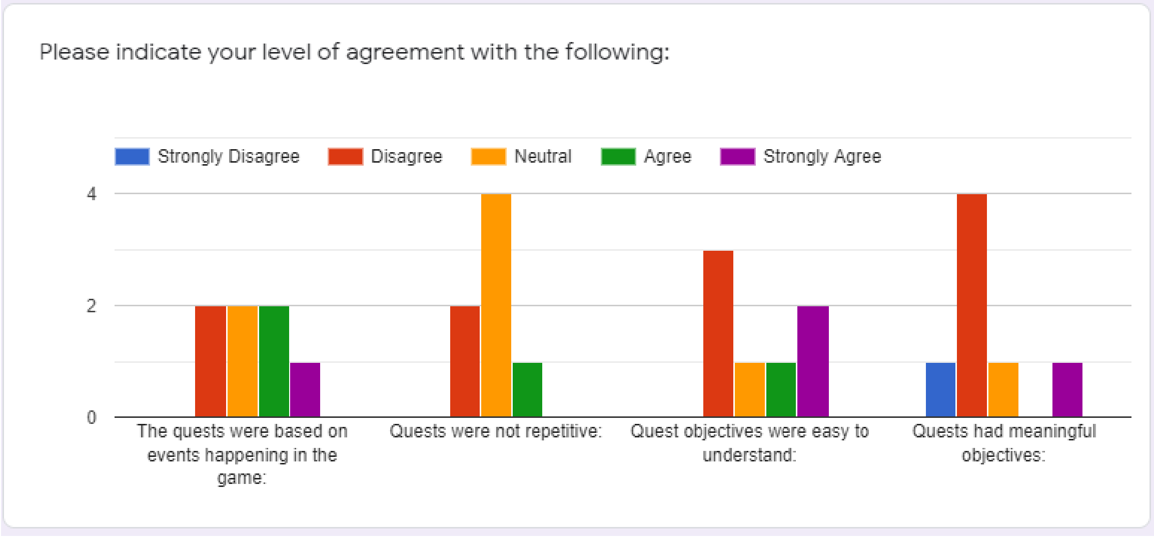
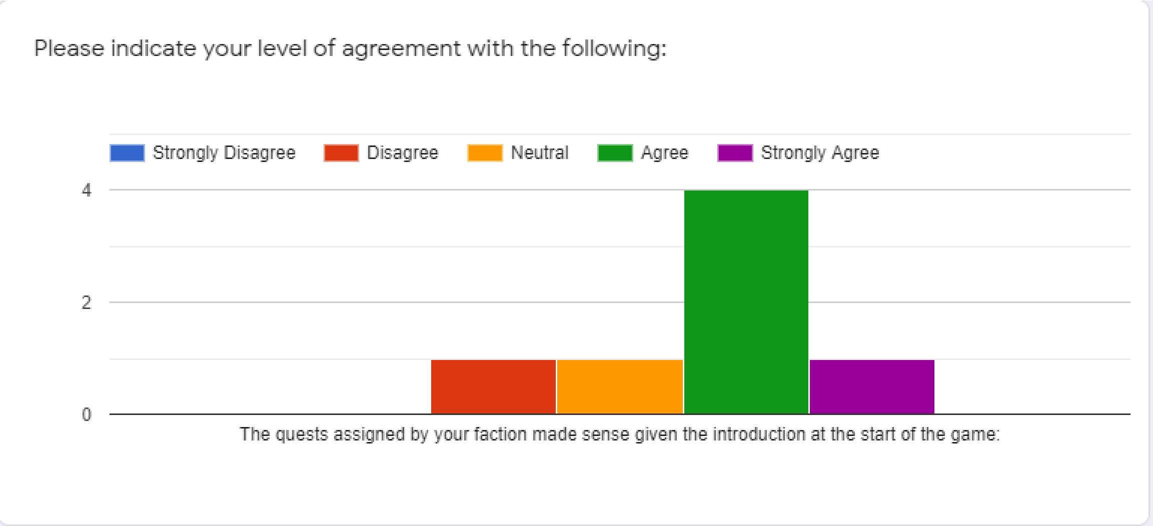
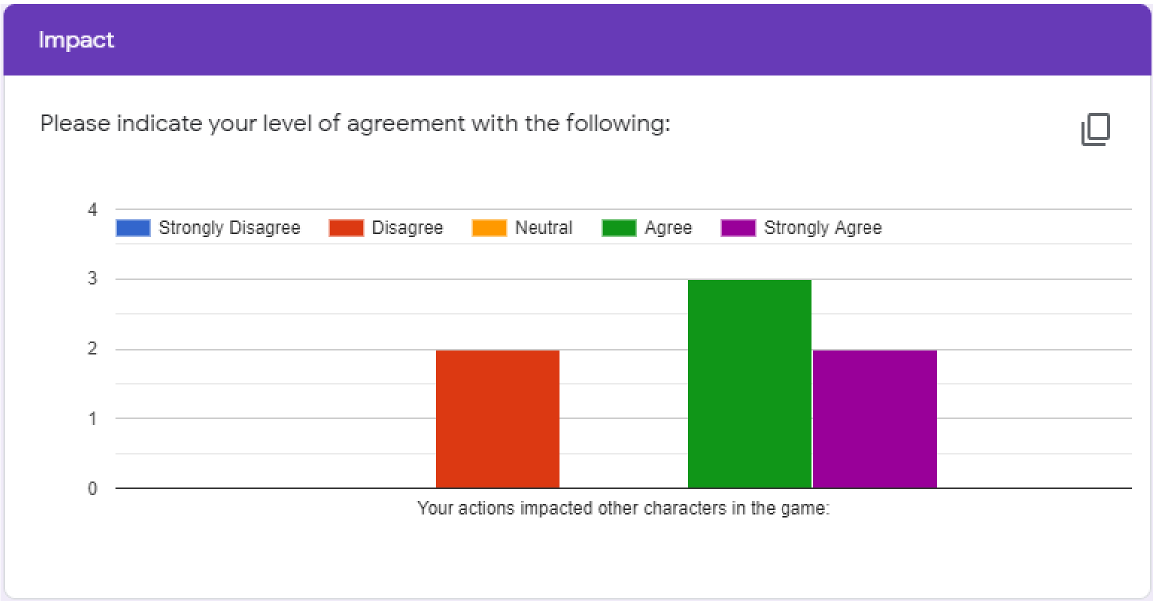


Figure 6.7: First study quest questions



Please indicate your level of agreement with the following:

Figure 6.8: First study faction immersion



Please indicate your level of agreement with the following:

Figure 6.9: First study multiplayer impact

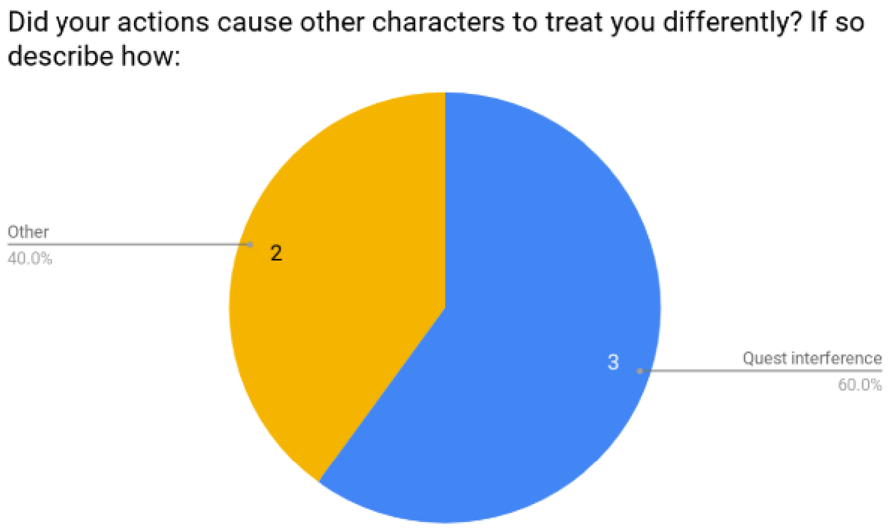


Figure 6.10: First study free-response on actions affecting players

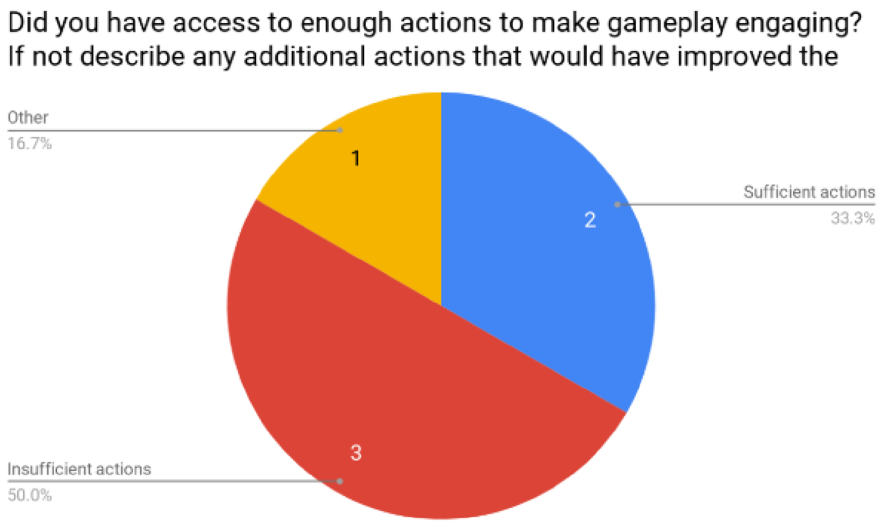


Figure 6.11: First study free-response on actions making gameplay engaging

6.2.3 Second study

The second study was completed over voice chat with five members of the Cal Poly Game Development Club. The scenario and human client were significantly refined to be more streamlined for this version. These changes included more detailed quest reasoning, many bug fixes, and a new “Help” window designed to teach the UI and give tips to confused players. Despite many hours of bug fixing and informal testing with other people, this session still suffered from some less severe issues. There were two servers crashes caused by one or more testers not correctly following given instructions. Fortunately, once the problem was found the rest of the test was able to proceed. The results of the second study are discussed with the results of the third study.

6.2.4 Third study

The final session was designed to address the majority of the issues from the previous sessions. The invitation to participate was sent out in the Cal Poly Computer Science newsletter as well as posted to the department’s Facebook page. Six volunteers conveyed their interest in participating. Testers were assigned to two separate online sessions in groups of three in order to minimize the chaos that was present in other tests. It was also decided that the help interface was enough to quickly educate players on how to operate the UI, so a guided tutorial over screen-share was made part of the testing process. Instructions were emailed to subjects multiple days before their assigned session. Frustratingly, only 1 out of 6 volunteers actually showed up to their assigned session. While the participation rate was disappointingly low, the lone tester did not run into any UI or technical glitches during his testing session.

6.2.5 Results of second and third studies

The results of the second and third studies are combined since there were few participants and only minor changes between the two sessions. 5 of the 6 participants were members of the Cal Poly Game Development Club. All participants agreed that they had played and enjoyed games with quests. However, only about half of the participants agreed that most games they played had a good quest system. All but one participant claimed to have played games with procedurally generated quests, but only three of them agreed to have enjoyed procedural quests. One tester indicated that they had participated in a testing session of a different scenario of Panoptyk that was being used as as Turing test.

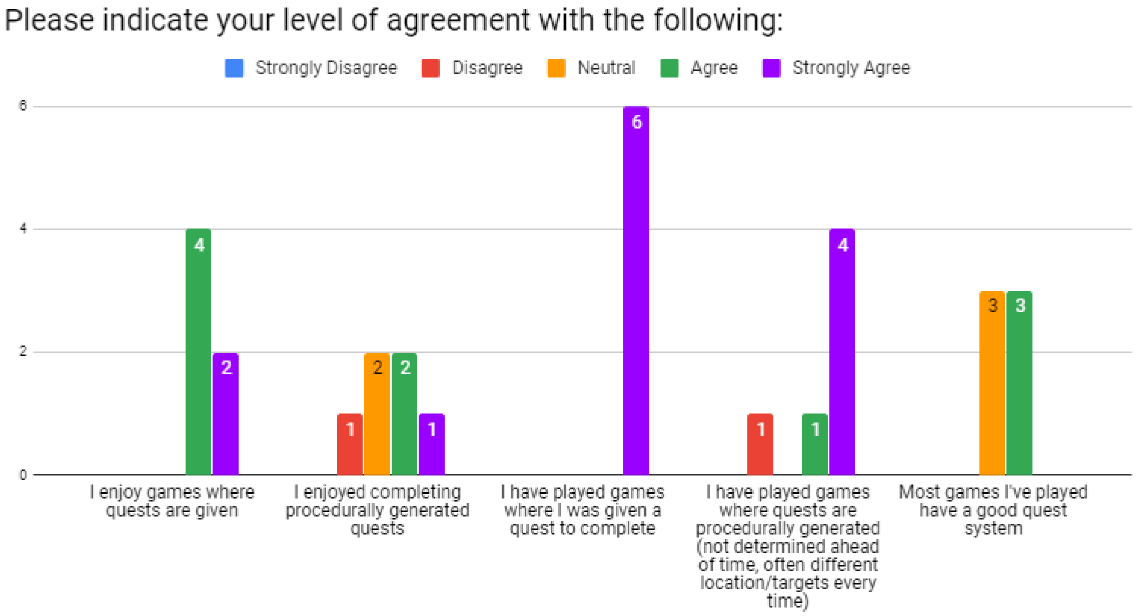


Figure 6.12: Second study screening

Enjoyment of the game scenario was far better than in the previous study, 4 out of 6 participants claimed to have enjoyed playing the game. There was a significant change in opinion of the game's strength and weaknesses, no aspect of the game was universally considered a strength or weakness. The increased perception of weakness is likely due to the fact that more players were able to experience intended aspects of gameplay. A higher percentage of participants considered quests, User Interface, and characters a strength of the game. In the free response one player remarked that "It was fun trying to find items." There was still significant complaints about the UI, especially parts of the UI related to information pieces. One player complained it was hard to find their targeted agent for the quest he was given.

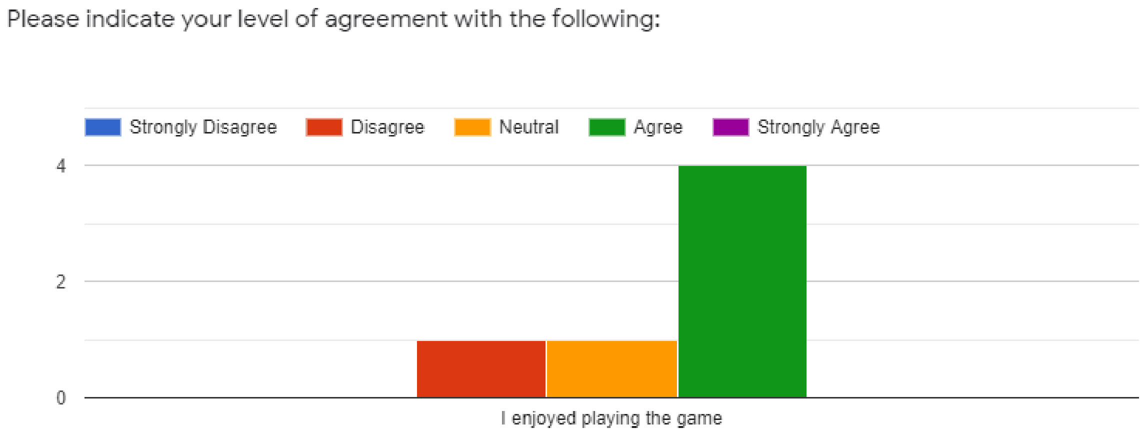


Figure 6.13: Second study overall enjoyment

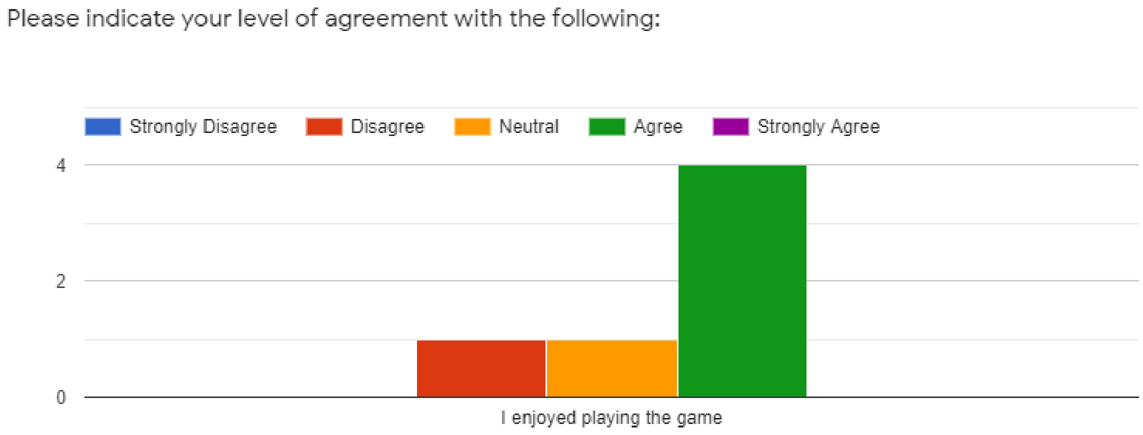


Figure 6.14: Second study strengths

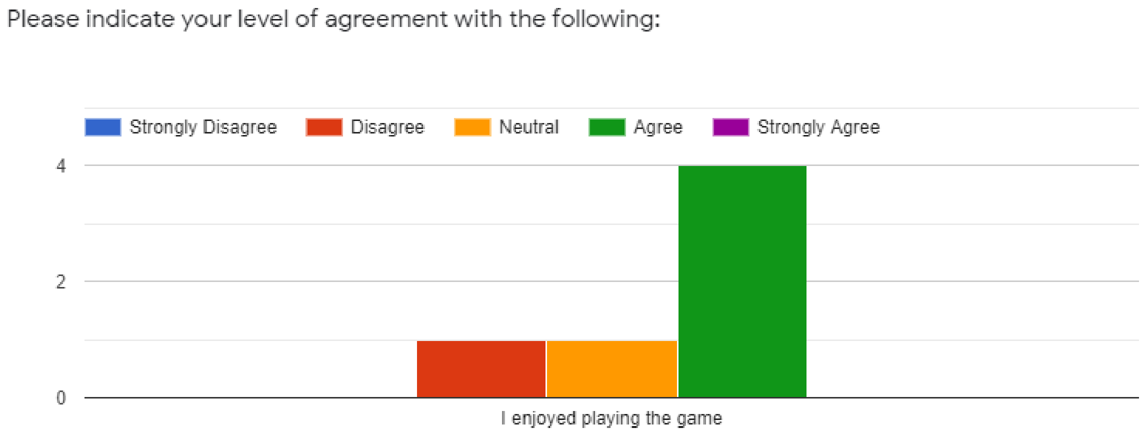


Figure 6.15: Second study overall weaknesses

Results regarding quests were far more meaningful than the previous session. 2 of the 6 players were able to deduce that the quests were based on events happening on the game. Every participant was able to complete quests, with 4 out of 6 of them completing 5 or more quests. One player turned in 55 quests due to an exploit that allowed him to turn in the same quest repeatedly. Of the six participants, three of them found quests "a little hard," one found quests "way too hard" and one found them "a little easy." The participant who found quests "a little easy" was the participant who played the game without any other testers in spite of the fact that the scenario was balanced around having 3-6 testers. Every participant agreed that quests were based on events happening in the game. Unfortunately, due to the limited possible action space for quests, all but one user found quests repetitive. Participants were divided whether quests had meaningful objectives or not, 2 of them thought they were not meaningful, 3 were neutral, and 1 thought they were meaningful. Participants were extremely polarized on whether they received any positive or negative consequences from quests, 2 agreed or strongly agreed they did and 2 strongly disagreed they did. In the free response section asking for participants to describe interesting encounters they had as a part or result of a quest, one participant mentioned he got stuck getting assigned the same quest for the whole game, another participant noted they did a lot of "thanking" quests, and the remaining participants talked about an arrest event. In the feedback section, there was four complaints related to bugs or UI issues. The remaining participant wanted more detail in the empty areas of the city.

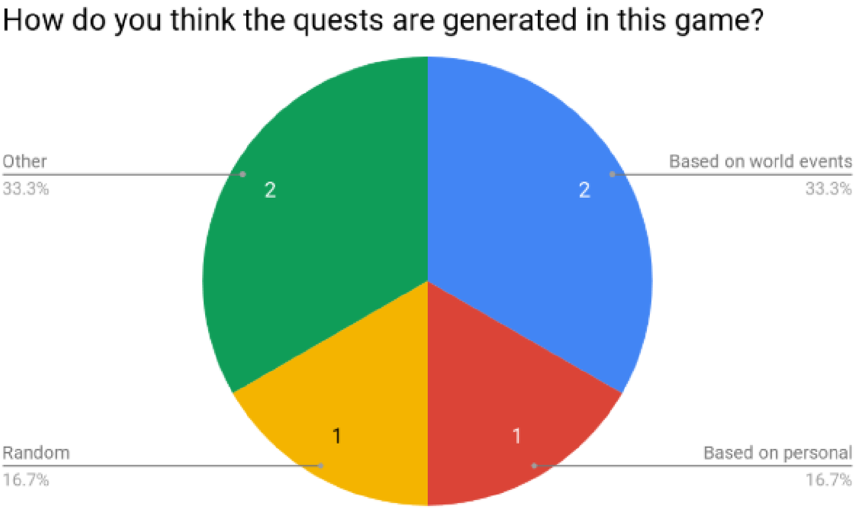


Figure 6.16: Second study free-response on quest generation method

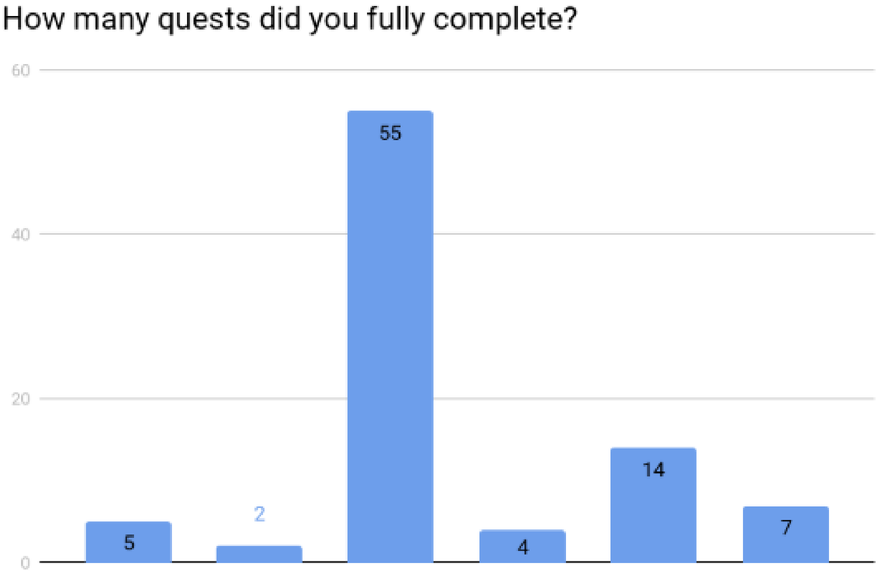


Figure 6.17: Second study quests completed

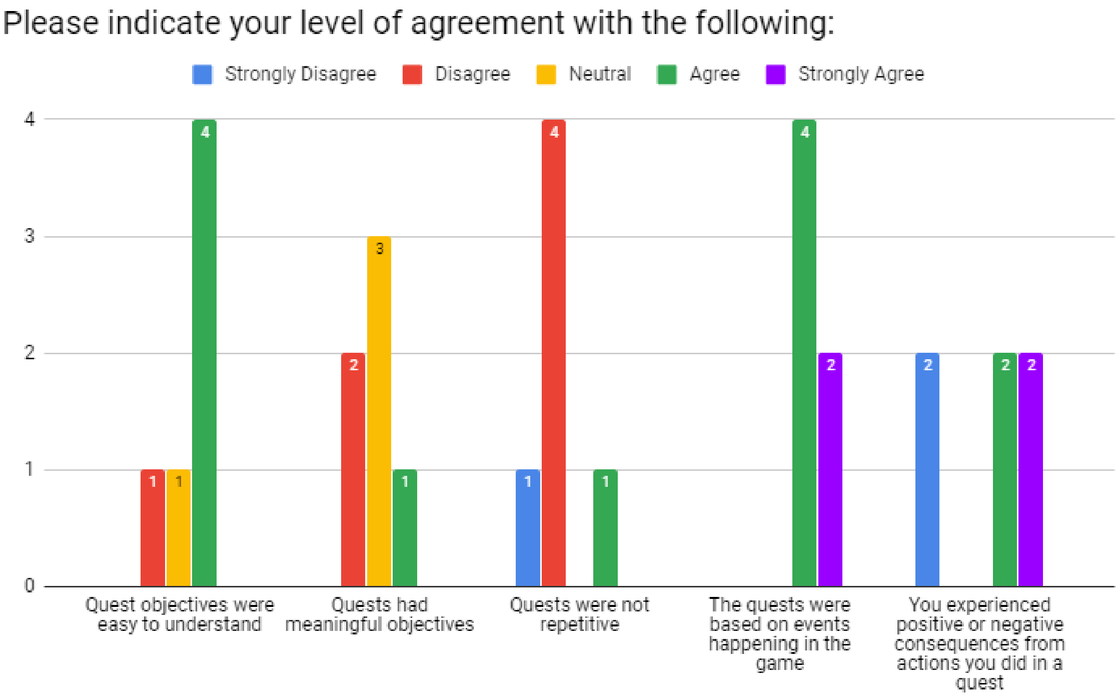


Figure 6.18: Second study quest questions

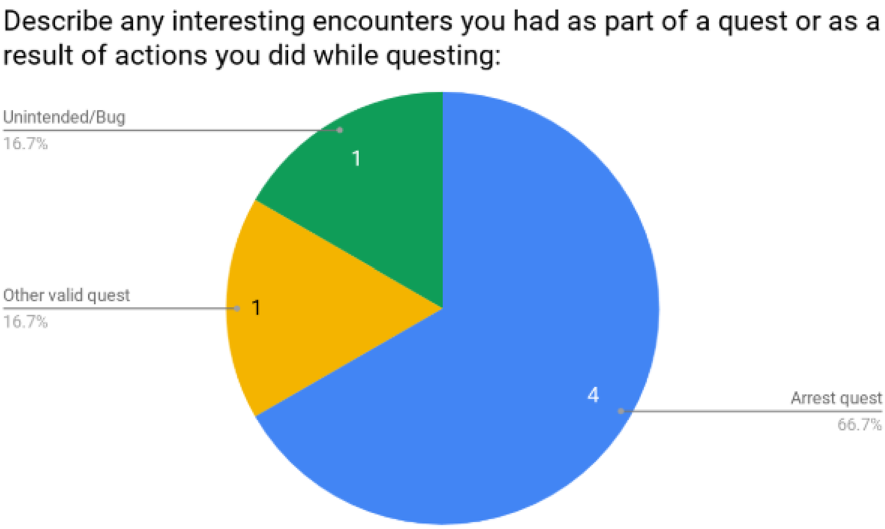


Figure 6.19: Second study free-response on memorable quest encounters

Only one participant was able to complete the extra questions designed for the third study. The participant agreed having quests based on player input had a positive impact on his experience. When asked about the advantages of such quests he wrote “It helps make the story unique (see previous comment about Chris and being arrested). It can also help the player get more involved in the story if they are having a direct impact on the direction that it is going, instead of a quest-on-rails storyline.” which was the exact intention behind the system. When asked about the disadvantages of the system he responded “I went and looked at a lot of different items before coming back to the faction leader. Since I had done that, I had a lot of options to choose from for turning in the first quest. The issue was that the information wasn’t removed from the dropdown menu, so there were times when I would accidentally try to submit information that I submitted before,” this was a definitely a shortcoming of the UI.

The feedback towards factions was similar to the first study. No participants thought that the quests assigned were not appropriate for their faction. For additional feedback, one participant complained that there was not enough backstory given. Another participant never ended up interacting with someone from a different faction, this player was also the player assigned an abnormal number of “thanking” quests. The feedback on impact towards other players correlated heavily with the question that asked about quest impact. The only participants who disagreed that their actions impacted other characters in the game were the players who were not assigned quests targeting other players. When asking for additional feedback on gameplay actions, two participants said there were enough actions to make gameplay engaging. Three participants claimed that they had issues with either the information system or UI.

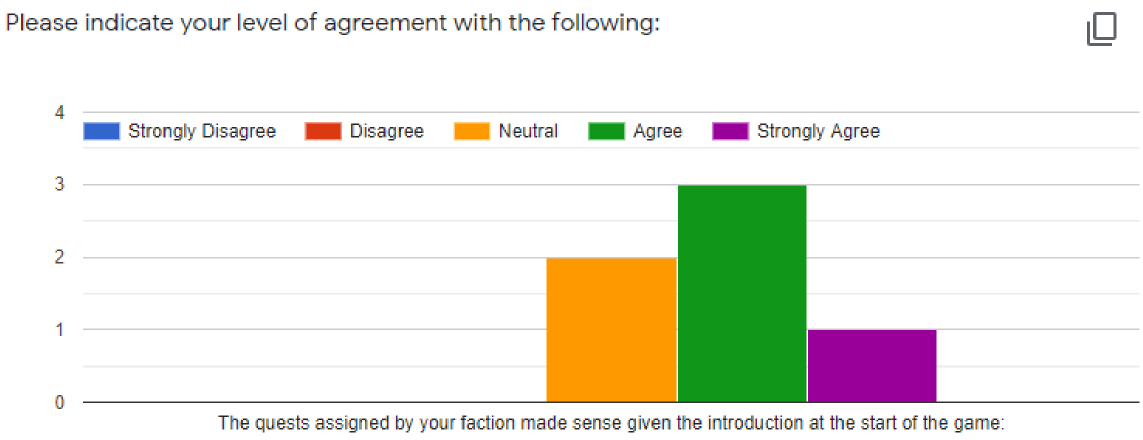

Please indicate your level of agreement with the following:

Figure 6.20: Second study faction immersion

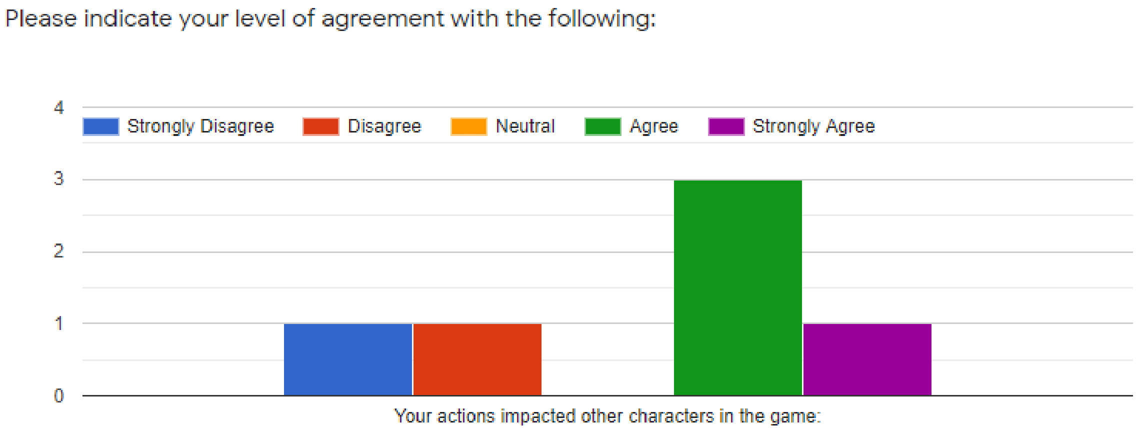

Please indicate your level of agreement with the following:

Figure 6.21: Second study multiplayer impact

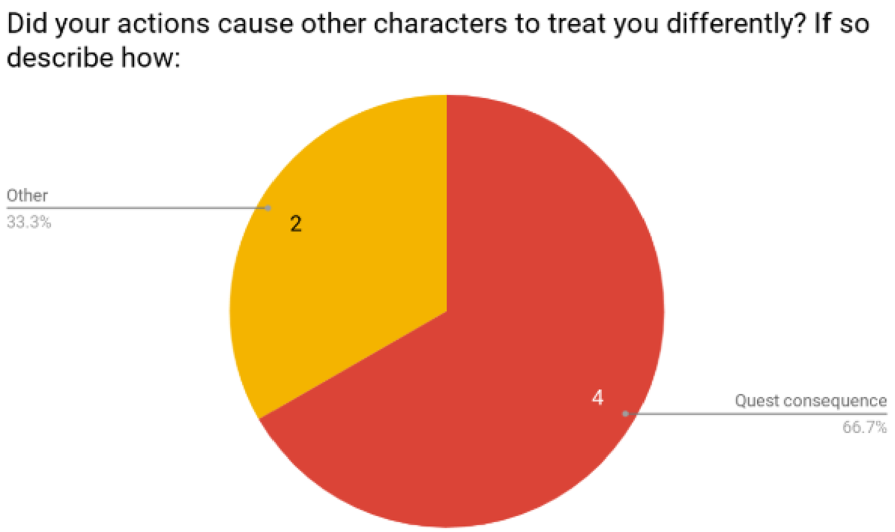


Figure 6.22: Second study free-response on actions affecting players

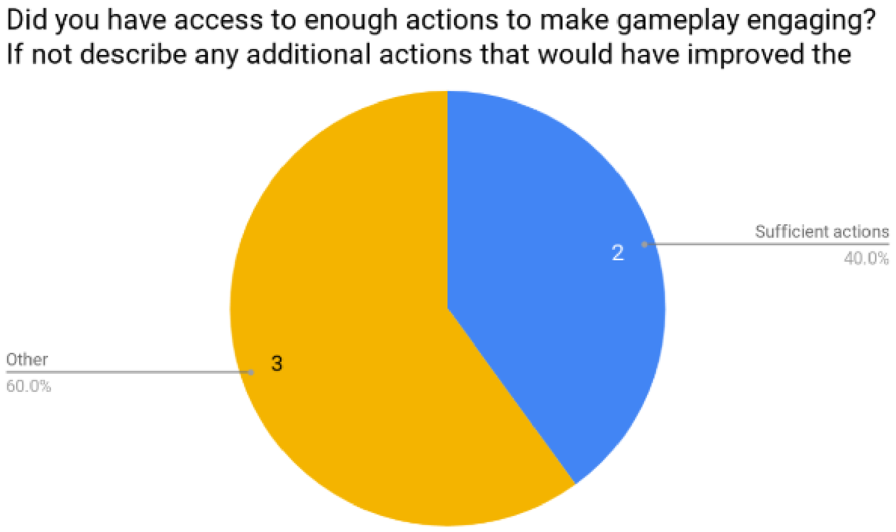


Figure 6.23: Second study free-response on actions making gameplay engaging

6.2.6 Analysis

Overall, the results from user testing show positive trends with room for improvement. The majority of users who did not experience crippling bugs were able to perceive that the generated quests were relevant to the events they were experiencing, and that their actions could have consequences that affected them later. The action/consequence implications of quests shows that our system contains the foundations needed to generate immersive quests. All players who experienced arrest-related quests talked about arrest-related encounters when asked about interesting encounters related to quests. This shows that players associated events around the central plot point of being arrested.

The mixed feedback regarding the meaningfulness of quest objectives was likely due to a number of factors. A few players ran into critical bugs that prevented them from completing quests. Players who were able to experience quests were not guaranteed to experience all possible types of quests due to the dynamic nature of their generation. In the second study, a few of the players were able to gather all the items around the world and deprive the slower players of any chance to complete a quest. This issue was corrected by restrictions to carry capacity in the third study, but was not able to be thoroughly tested due to the lack of participants. Another factor that may have hampered the meaningfulness of quests objectives was the lack of backstory and context surrounding the scenario. Unfortunately, the creation of an interesting and immersive game world is complex task that goes beyond the context of quests.

Complaints about the repetitiveness of quests are unfortunate, but perhaps unavoidable given the limited gameplay possible in this early build of Panoptyk. The main gameplay loop of finding items did not really have any exciting moments associated with it. The fact that no player commented on the different rewards they received from completing quests probably means they were uninteresting enough to go unnoticed. The way quests objectives were displayed could have been another reason for why quests were perceived as repetitive; quest objectives always displayed the end goal rather than steps that a player needed to take to complete it. For example, the item quests always required a player to given an item to their faction leader, but did not display any intermediary objectives leading up to the final objective. Assigning intermediary objectives may have helped players notice that the actions they were taking to complete quests were unique and dependent on the overall state of the world. Further discussion on ways to improve upon the system are discussed in Chapter 7.1.

6.3 Additional validation through internal testing

Due to the low number of participants, additional validation was performed on the scenario using 12 agents consisting of 10 AI agents and 2 humans familiar with the system. The purpose of this test run was to empirically show that quests could be generated with limited information, that relevant quests are generated, and that quests have consequences. The human testers involved in this internal test completed quests so that data on generated quests could be collected. Other than the use of internal testers, the overall scenario remained the same as previously described. A total of 30 quests were completed.

6.3.1 Validation on quest generation with limited information

To show that quest generation is possible with limited information, factions leaders must always be able to assign a quest to a faction member without one. To prove this, we start a fresh server instance in which no agent has any information given by default. The faction leader is able to, without knowledge of anything other than the room it is in, request a conversation with any faction members in its room and assign a generic quest to gather more information. This generic quest is possible by creating a type of information predicate without specifying its action. For example, a quest command of TILQ could be completed with any information that has a time, item, location, and quantity. AI agents and human clients are able to interpret this as a quest to find any information that has the given properties. Therefore quests are able to be generated with only basic knowledge of the structure of information. This was confirmed in the internal validation test where quest givers of the two different factions were able to immediately assign quests. Figure 6.3.2 shows the total amount of information known by quest givers for their first 15 assigned quests during the test.

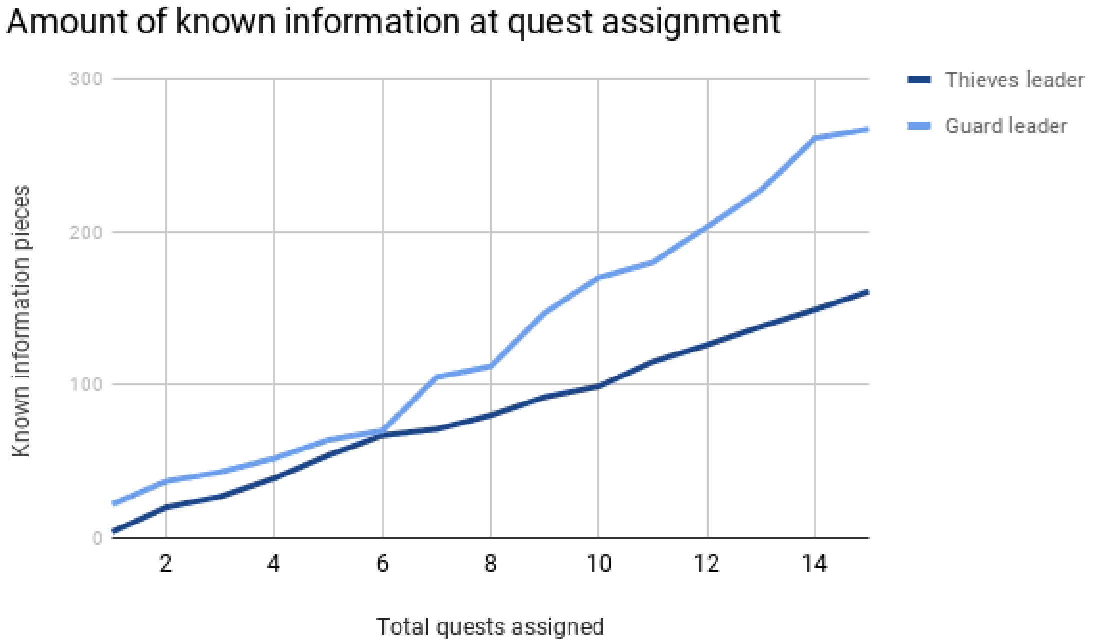


Figure 6.24: Information known during quest generation using 10 AI agents and 2 humans

6.3.2 Validation that relevant quests are generated

“Relevant” can be a highly subjective term, so we will define relevant as “based on events happening in the game and related to the overall goals of a quest giver.” In order for a quest to be based on events happening in the game, the quest must have been created from an action that occurred during gameplay. From the generic information gathering quest, quest givers are able to gain information about events occurring in the world. Quest givers use the logic defined in either Algorithm 2 or Algorithm 1 to create a valid quest that aligns with their goals. The generic information gathering quests themselves do not meet our definition of “relevant” but allow for quests that do to be generated. Internal testing found that the full range quests defined in said algorithms were able to be assigned and completed. We can conclude that, given our definition of “relevant,” the system is able to produce relevant quests. This was confirmed in the internal validation test, data represented in Figure 6.3.2 shows that the majority of the assigned quests referenced an in-game event.

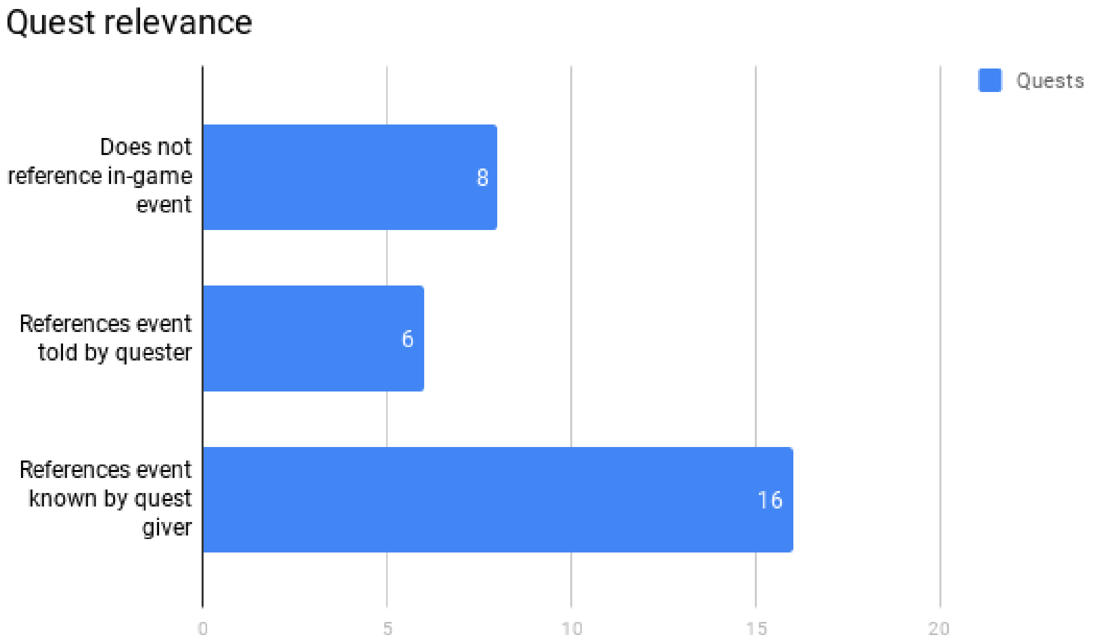


Figure 6.25: Quest relevance during scenario using 10 AI agents and 2 humans

6.3.3 Validation that quests have consequences

For the purpose of internal validation, we consider quest related “consequences” as in-game quests assigned because of any actions done as part of a previous quest. We can show that quests cause consequences that can affect both the questing agent and other agents in the game. It is straightforward to see how agents experience personal consequences from completing quests, the generic information gathering quest often has the consequence of a followup quest referencing the information turned in as part of the information gathering quest. In specific cases, such as when an agent does an illegal action to complete a quest, another agent can experience the consequence of having to hunt down and arrest the first questing agent. This in turn can cause additional consequences for the first quest agent, where they will be directed to take revenge on the agent that arrested them (another illegal action). There is also the positive consequence of a gift quest that can be assigned to reward an agent that has done things to please a quest giver. Internal testing shows that all of the described quest consequences regularly occur, proving that quests in the system have what we have defined as "consequences." To show that quests cause follow-up consequences, we tracked consequences on the 30 quests generated in the internal test. Figure 6.3.3 shows the resulting follow-up quests from the internal test; a follow-up quest was any quest that referenced an event turned in to complete a previous quest. The "Quests without immediate follow-up" could have resulted in follow-up quests under the right circumstances (such as someone witnessing a theft), but were either undetected by the opposing faction or not considered a priority by the quest givers. The net impact on the original questing agent is shown in Figure 6.3.3, the majority of follow-up quest consequences were for agents to hand over items involved in previous quests.

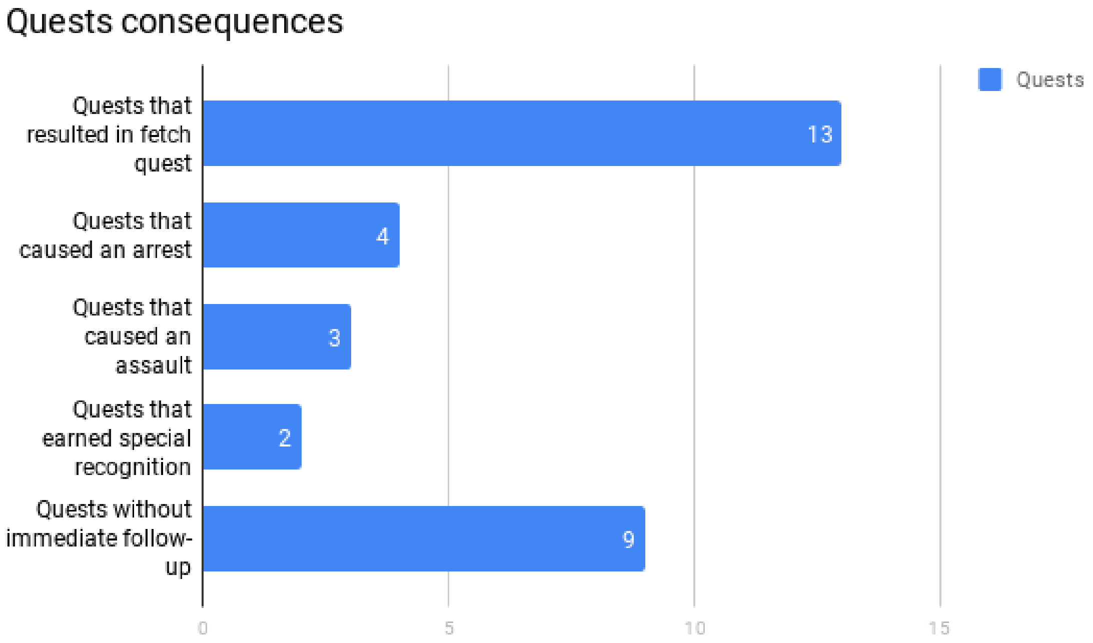


Figure 6.26: Follow-up consequences to quests generated using 10 AI agents and 2 humans

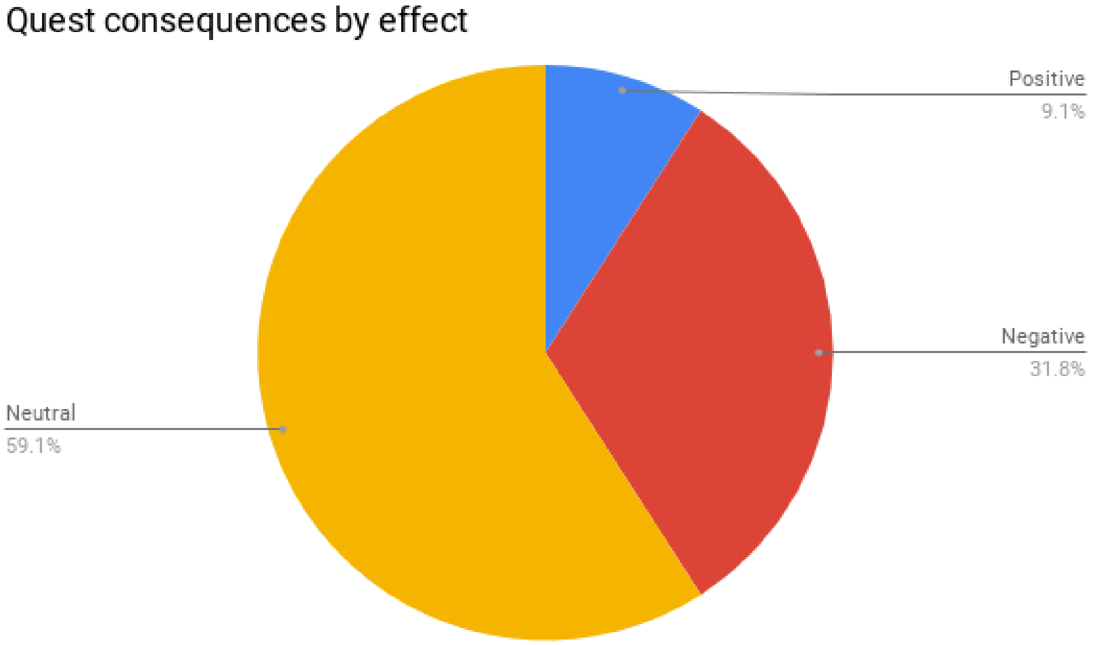


Figure 6.27: Impact of follow-up quest consequences to quests generated using 10 AI agents and 2 humans

Chapter 7

CONCLUSION

7.1 Future work

7.1.1 Improved communication between Panoptyk agents

The UI and mechanics that govern communication between agents may have fundamental shortcomings that make it difficult to properly communicate with other agents. The current system did not allow players to easily express their intentions and was made worse by the fact that they could not say that they did not know or have something unless they were in a trade. Another fundamental issue is that agents totally lack avenues to express emotion or personality; there was no ways for trusted friends or hated enemies to communicate differently to each other. Methods of representing emotion or personality in actions and conversations should be investigated as it would allow for far more varied and unique interactions between agents.

7.1.2 Additional gameplay actions in Panoptyk

Our quest generator would benefit greatly from additional gameplay additions to the core Panoptyk engine. The relatively basic gameplay offered by the current iteration of engine made it difficult to offer varied quests that humans player consider interesting. Gameplay mechanics based on traditional RPGs mechanics, such as unique abilities that can be leveled up, basic combat between adversaries, and stealth actions would likely make the game more interesting to the average gamer. Additional mechanics would also allow for different quest types with specialized requirements based on each agent's strengths and weaknesses. The current quest system could dynamically pick new goals based on information, but only had 4 basic actions (2 of which are very situational and rarely assigned) it could assign to players. Having quests that encourage players to use a variety of different gameplay mechanics would make gameplay feel far less repetitive. In all, we believe a more exciting gameplay loop would increase player enjoyment and make it easier for them to find value in quests they are assigned.

7.1.3 Valuing uncertain information

Despite being a core premise of Panoptyk, the vast majority of generated information has no practical value to players. This is largely due to the fact that only the most recent information pieces offer information that could be helpful. For example, the information piece "Bob moved from the backyard to the kitchen at 12:30" only holds value if you are trying to find Bob. However, it does not hold any value if Bob is continuously moving and has gone to the opposite part of the town by 12:35. The same conundrum applies to the location and ownership of items; all information can very quickly become outdated and the player has no way of knowing if they are receiving something useful when trading for it. This issue was manageable with AI agents that had relatively static behavior, but became a serious impediment once human players with erratic behaviors started influencing the world. Given that all quests in Panoptyk are based on in-game information this quickly had a catastrophic effect on the entire quest system. It became impossible to assign most types of actions as quests because those actions could end up being impossible from unexpected changes. For example, it was originally intended that the thieves guild would tell its members to trade for a target item if they were friendly with its owner or steal it if they were not, but either quest became impossible if the owner somehow lost possession of the targeted item. This inevitably resulted in all item retrieval quests turning into "Give Target Item to Faction Leader," a very generic goal lacking any faction personalization. Future work on the Panoptyk engine should investigate ways of giving some value to the large amounts of mostly useless information. The engine's trade system should also be overhauled so that players are able to get an indication if information offered in a trade is more recent then anything they already know on the subject.

7.1.4 Better sentence converter for human client

Panoptyk events are not always converted to easily readable English sentences. For example, a player trying to ask "Who possesses the pizza" would have to ask the question "??? POSSESS pizza on ???." This formatting would likely confuse any users who have not had experience with formal logic as they might not understand that "POSSESS" is a predicate that accepts up to 3 parameters (agent, item, and time). Enhancements to the sentence converter would ideally make questions more readable and help users digest information. More robust improvements to the sentence converter could include a sentence generator capable of naturally including appropriate adjectives when referring to different items or agents. Some amount of text variation for quest descriptions could also reduce the feeling of repetitiveness associated with quests.

7.1.5 Additional user testing

The relatively small number of user-testers made it difficult to come to strong conclusions. The early tests of Panoptyk were severely hampered by bugs and confusion. Fortunately, the latest iteration of user-tests seemed to have no major bugs and far less user confusion than previous tests. Unfortunately, participation in the final round of user-testing was abysmally low due to the COVID-19 Pandemic despite being advertised through multiple school channels. Finding alternate ways to gain user-testers would be essential for further analysis of quests.

7.2 Summary of contributions

Our first contribution is the described implementation of Panoptyk as a research framework to solve problems in procedural content generation, believable AI, and MMO games. Unlike other MMO research platforms, Panoptyk focuses on social interactions between humans and AI agents. As part of our work on Panoptyk, we created a playable scenario in which human and AI agents were able to complete quests together with no handicaps. This contribution also includes the creation of example AI agents that are able to use the same information as human players to competently evaluate the world and perform complex actions for their factions.

Our second contribution is a decentralized dynamic quest generation system based on the observations and inputs of agents in MMORPGs. This system was able to dynamically generate personalized quests based on the experiences of the agents interacting with it. We show, through a user study, that humans recognized that our dynamic quests were based off of in-game events rather than procedurally generated facts. We also show that players noticed positive or negative consequences from actions they did as part of a quest. The consequences of quest actions show the building blocks of generated drama and story. The drama of "arrest" quests led to many players describing their encounters in the context of that central plot point. It is hoped that additional gameplay in Panoptyk will open up more opportunities to create dramatic plot points. As part of this contribution, we describe the experimental design used to evaluate quests so that others may repeat our experiments.

BIBLIOGRAPHY

[1] T. Adams. Dwarf fortress. *Game [Windows, Mac, Linux], Bay*, 12, 2006.

[2] T. Adams. Simulation principles from dwarf fortress. *Game AI Pro*, 2:519–521, 2015.

[3] F. Amato and F. Moscato. Formal procedural content generation in games driven by social analyses. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 674–679. IEEE, 2017.

[4] Bethesda. Elder scrolls skyrim, 2011.

[5] Bethesda. Radiant ai, 2011.

[6] J. B. Black and R. Wilensky. An evaluation of story grammars. *Cognitive science*, 3(3):213–229, 1979.

[7] S. Boluk and P. LeMieux. Dwarven epitaphs: Procedural histories in dwarf fortress. *Comparative Textual Media: Transforming the Humanities in the Postprint Era*, pages 125–154, 2013.

[8] V. Bui, H. Abbbass, and A. Bender. Evolving stories: Grammar evolution for automatic plot generation. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[9] T. Debeauvais, B. Nardi, D. J. Schiano, N. Ducheneaut, and N. Yee. If you build it they might stay: retention mechanisms in world of warcraft. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 180–187. ACM, 2011.

[10] N. Dehn. Story generation after TALE-SPIN. In *IJCAI*, volume 81, pages 16–18, 1981.

[11] J. Doran and I. Parberry. A prototype quest generator based on a structural analysis of quests from four mmorpgs. In *Proceedings of the 2nd international workshop on procedural content generation in games*, pages 1–8, 2011.

[12] J. Doran and I. Parberry. A server-side framework for the execution of procedurally generated quests in an mmorpg. In *GAMEON'15-Proceedings of the 16th Annual European Conference on Simulation and AI in Computer Games*, pages 103–110, 2015.

[13] Df2014:quest.

[14] M. Farone, T. Hicks, J. Benjamin, and C. Cao. User created content and quests, June 26 2012. US Patent 8,206,224.

[15] A. M. Frisch and D. Perlis. A re-evaluation of story grammars. *Cognitive Science*, 5(1):79–86, 1981.

[16] Cal Poly Github. `http://www.github.com/CalPoly`.

[17] P. Golle and N. Ducheneaut. Preventing bots from playing online games. *Computers in Entertainment (CIE)*, 3(3):3–3, 2005.

[18] I. Guthridge and E. Nahari. Dynamic quests in game, Nov. 3 2015. US Patent 9,174,128.

[19] E. J. Hastings, R. K. Guha, and K. O. Stanley. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4):245–263, 2009.

[20] P. Hernandez. The internet loves making fun of Fallout 4's Preston Garvey. *Kotaku Australia*, Jan 2016.

[21] P. Hingston. A turing test for computer game bots. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(3):169–186, 2009.

[22] P. Hingston. A new design for a turing test for bots. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 345–350. IEEE, 2010.

[23] I. D. Horswill. Mkultra. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[24] I. D. Horswill. Postmortem: Mkultra, an experimental ai-based game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[25] I. D. Horswill and L. Foged. Fast procedural level population with playability constraints. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012.

[26] What is spatialos?

[27] S. Klein, J. D. Oakley, D. I. Suurballe, and R. A. Ziesemer. A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1971.

[28] B. Kybartas and C. Verbrugge. Analysis of regen as a graph-rewriting system for quest generation. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2):228–242, 2013.

[29] M. Lebowitz. Story-telling as planning and learning. *Poetics*, 14(6):483–502, 1985.

[30] Y.-S. Lee and S.-B. Cho. Context-aware petri net for dynamic procedural content generation in role-playing game. *IEEE Computational Intelligence Magazine*, 6(2):16–25, 2011.

[31] Y.-S. Lee and S.-B. Cho. Dynamic quest plot generation using petri net planning. In *Proceedings of the Workshop at SIGGRAPH Asia*, pages 47–52, 2012.

[32] B. Li and M. O. Riedl. An offline planning approach to game plotline adaptation. In *AIIDE*, 2010.

[33] D. Livingstone. Turing's test and believable ai in games. *Computers in Entertainment (CIE)*, 4(1):6, 2006.

[34] M. Mateas. Expressive ai: Games and artificial intelligence. In *DiGRA Conference*, 2003.

[35] J. McCoy, M. Treanor, B. Samuel, B. Tearse, M. Mateas, and N. Wardrip-Fruin. Authoring game-based interactive narrative using social games and comme il faut. Citeseer.

[36] J. R. Meehan. Tale-spin, an interactive program that writes stories. In *Ijcai*, volume 77, pages 91–98, 1977.

[37] Javascript for any scale.

[38] G. S. Miller. The definition and rendering of terrain maps. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 39–48. ACM, 1986.

[39] M. Miller, S. Mendonca, N. Philliber, and F. Khosmood. Panoptyk: information driven mmo engine. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–4, 2019.

[40] I. Millington and J. Funge. *Artificial intelligence for games.* CRC Press, 2009.

[41] R. J. Moore. System for generating or using quests, May 23 2013. US Patent App. 13/304,024.

[42] H. R. Morse. Preliminary operating notes for saga ii. Technical report, MIT Technical Memorandum 8436-M-29, 1960.

[43] M. Nelson, M. Mateas, D. Roberts, and C. Isbell. Declarative Optimization-based Drama Management in Interactive Fiction. *IEEE Computer Graphics and Applications*, 26(3):32–41, 2006.

[44] Node.js. Node.js.

[45] G. Oh and J. Kim. Effective quest design in mmorpg environment. In *Game Developers Conference*, volume 2005, 2005.

[46] Panoptyk. panoptyk/example-game, May 2020.

[47] Panoptyk. panoptyk/panoptyk-engine, Mar 2020.

[48] Phaser - a fast, fun and free open source html5 game framework.

[49] J. Pita, B. Magerko, and S. Brodie. True story: dynamically generated, contextually linked quests in persistent systems. In *Proceedings of the 2007 conference on Future Play*, pages 145–151, 2007.

[50] A. Popescu, J. Broekens, and M. Van Someren. Gamygdala: An emotion engine for games. *IEEE Transactions on Affective Computing*, 5(1):32–44, 2013.

[51] V. Propp. *Morphology of the Folktale*, volume 9. University of Texas Press, 2010.

[52] M. O. Riedl, A. Stern, D. Dini, and J. Alderman. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):23–42, 2008.

[53] M. O. Riedl and R. M. Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

[54] D. L. Roberts and C. L. Isbell. Lessons on Using Computationally Generated Influence for Shaping Narrative Experiences. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2):188–202, 2014.

[55] D. E. Rumelhart. On evaluating story grammars. *Cognitive Science*, 4(3):313–316, 1980.

[56] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[57] J. Ryan. Grimes' fairy tales: a 1960s story generator. In *International Conference on Interactive Digital Storytelling*, pages 89–103. Springer, 2017.

[58] J. Ryan. *Curating simulated storyworlds*. PhD thesis, UC Santa Cruz, 2018.

[59] J. Ryan and M. Mateas. Simulating character knowledge phenomena in talk of the town. *Game AI Pro*, 3:433–448.

[60] J. O. Ryan, A. Summerville, M. Mateas, and N. Wardrip-Fruin. Toward characters who observe, tell, misremember, and lie. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[61] M.-L. Ryan. From Narrative Games to Playable Stories: Toward a Poetics of Interactive Narrative. *StoryWorlds: A Journal of Narrative Studies*, 1(1):43–59, 2009.

[62] N. Shaker, J. Togelius, and M. J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[63] M. Sharma, S. Ontañón, M. Mehta, and A. Ram. Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence*, 26(2):183–211, 2010.

[64] J. Skorupski and M. Mateas. Interactive story generation for writers: Lessons learned from the wide ruled authoring tool, Jan 2010.

[65] A. M. Smith, E. Butler, and Z. Popovic. Quantifying over play: Constraining undesirable solutions in puzzle design. In *FDG*, pages 221–228, 2013.

[66] A. M. Smith and M. Mateas. Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):187–200, 2011.

[67] J. Suarez, Y. Du, P. Isola, and I. Mordatch. Neural mmo: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784*, 2019.

[68] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

[69] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. Search-based procedural content generation. In *European Conference on the Applications of Evolutionary Computation*, pages 141–150. Springer, 2010.

[70] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.

[71] E. Tomai, R. Salazar, and D. Salinas. Adaptive quests for dynamic world change in mmorpgs. In *Proceedings of the International Conference on the Foundations of Digital Games*, pages 286–287, 2012.

[72] S. R. Turner. Minstrel: a computer model of creativity and storytelling. 1993.

[73] N. Wardrip-Fruin. *Expressive Processing: Digital fictions, computer games, and software studies*. MIT press, 2009.

[74] G. N. Yannakakis. Game ai revisited. In *Proceedings of the 9th conference on Computing Frontiers*, pages 285–292, 2012.

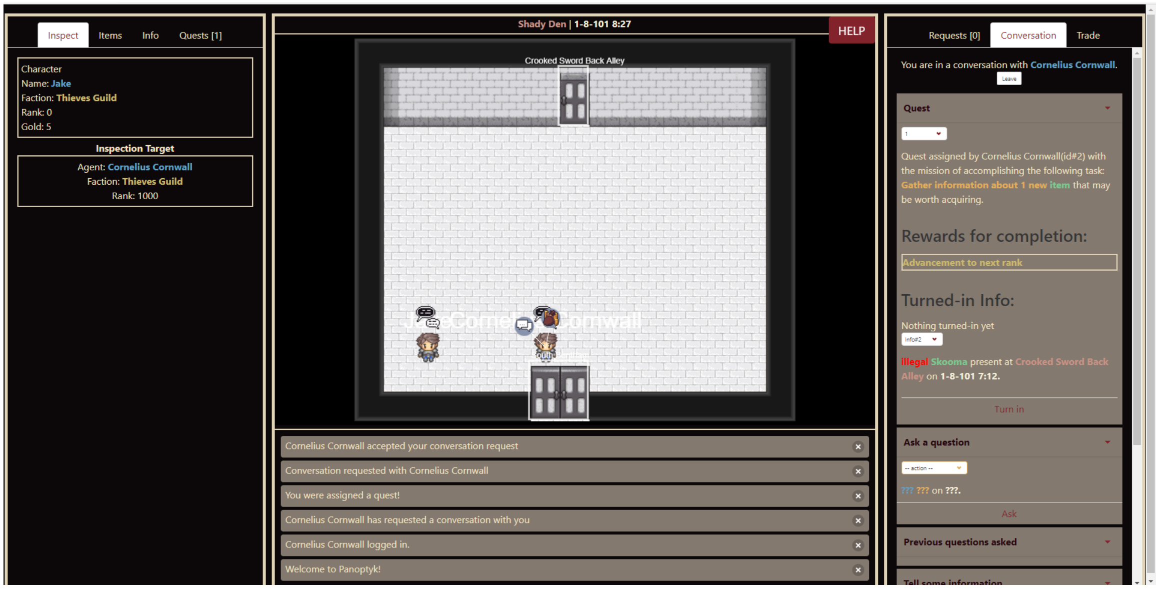
[75] G. N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.

[76] G. N. Yannakakis and J. Togelius. *Artificial intelligence and games*, volume 2. Springer, 2018.

APPENDICES

Appendix A

PANOPTYK HUMAN CLIENT



Character
Name: Jake
Faction: Thieves Guild
Rank: 0
Gold: 5

Inspection Target

Agent: Cornelius Cornwall
Faction: Thieves Guild
Rank: 1000

Cornelius Cornwall accepted your conversation request

Conversation requested with Cornelius Cornwall

You were assigned a quest!

Cornelius Cornwall has requested a conversation with you

Cornelius Cornwall logged in.

Welcome to Panoptyk!

You are in a conversation with Cornelius Cornwall.

Quest

1

Quest assigned by Cornelius Cornwall(id#2) with the mission of accomplishing the following task: **Gather information about 1 new item** that may be worth acquiring.

Rewards for completion:

Advancement to next rank

Turned-in Info:

Nothing turned-in yet

Info#2

illegal Skooma present at **Crooked Sword Back Alley** on **1-8-101 7:12.**

Ask a question

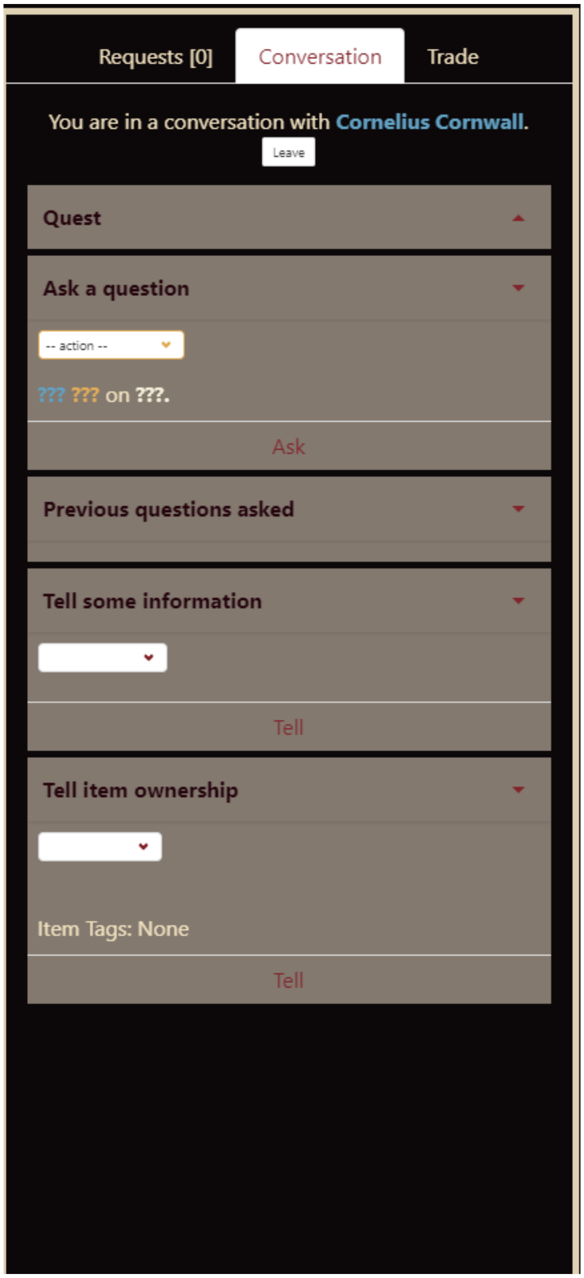
-- action --

??? ??? on **???.**

Previous questions asked

Tell some information

Figure A.1: Panoptyk Human Client Full Game Window

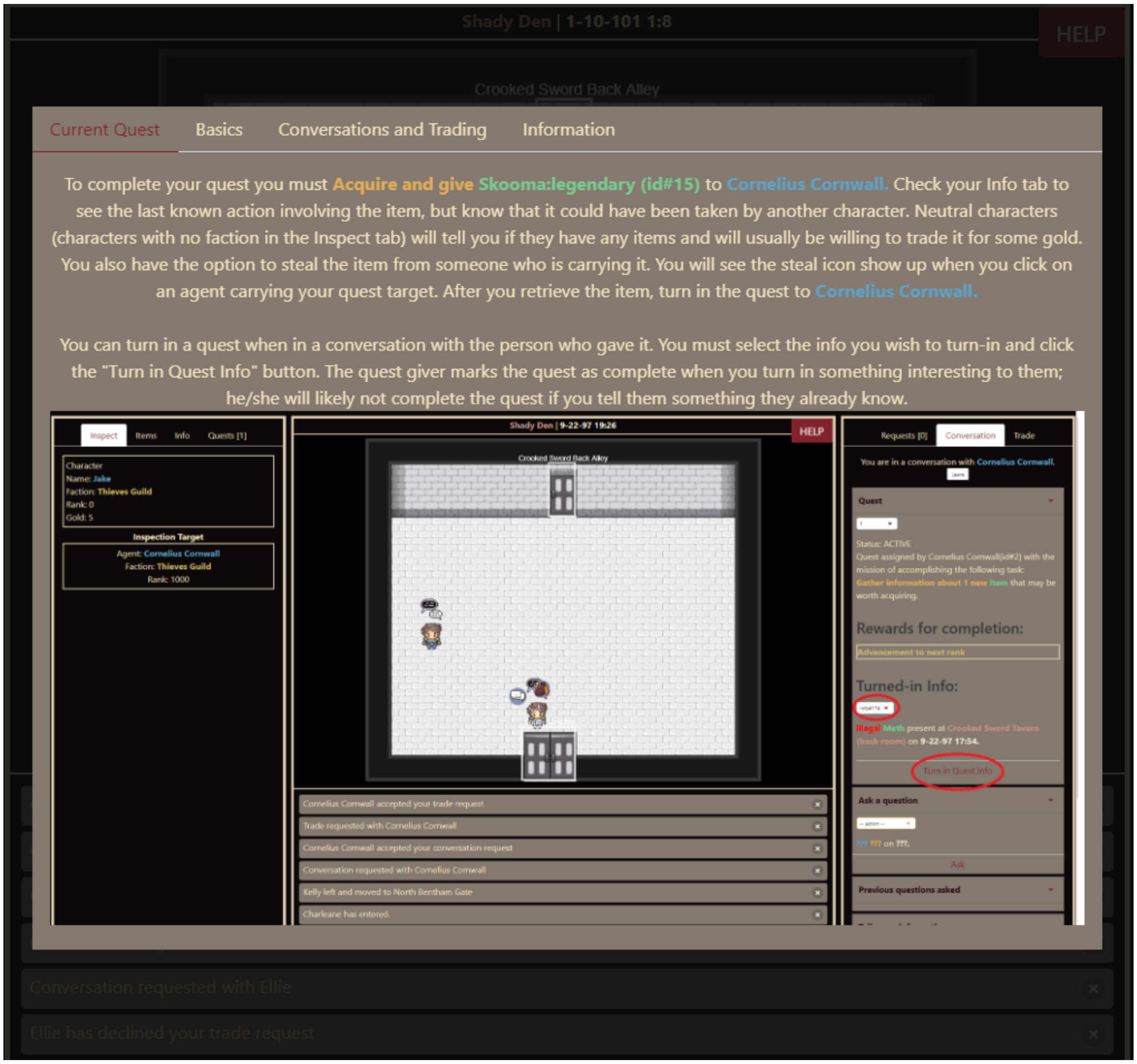


You are in a conversation with Cornelius Cornwall.

??? ??? on ???.

Item Tags: None

Figure A.2: Panoptyk Human Client Conversation Tab



To complete your quest you must Acquire and give Skooma:legendary (id#15) to Cornelius Cornwall. Check your Info tab to see the last known action involving the item, but know that it could have been taken by another character. Neutral characters (characters with no faction in the Inspect tab) will tell you if they have any items and will usually be willing to trade it for some gold. You also have the option to steal the item from someone who is carrying it. You will see the steal icon show up when you click on an agent carrying your quest target. After you retrieve the item, turn in the quest to Cornelius Cornwall.

You can turn in a quest when in a conversation with the person who gave it. You must select the info you wish to turn-in and click the "Turn in Quest Info" button. The quest giver marks the quest as complete when you turn in something interesting to them; he/she will likely not complete the quest if you tell them something they already know.

Character
Name: Jake
Faction: Thieves Guild
Rank: 0
Gold: 5

Inspection Target
Agent: Cornelius Cornwall
Faction: Thieves Guild
Rank: 1000

Cornelius Cornwall accepted your trade request
Trade requested with Cornelius Cornwall
Cornelius Cornwall accepted your conversation request
Conversation requested with Cornelius Cornwall
Kelly left and moved to North Bentham Gate
Charleane has entered.

You are in a conversation with Cornelius Cornwall.

Status: ACTIVE
Quest assigned by Cornelius Cornwall(id#2) with the mission of accomplishing the following task: Gather information about 1 new item that may be worth acquiring.

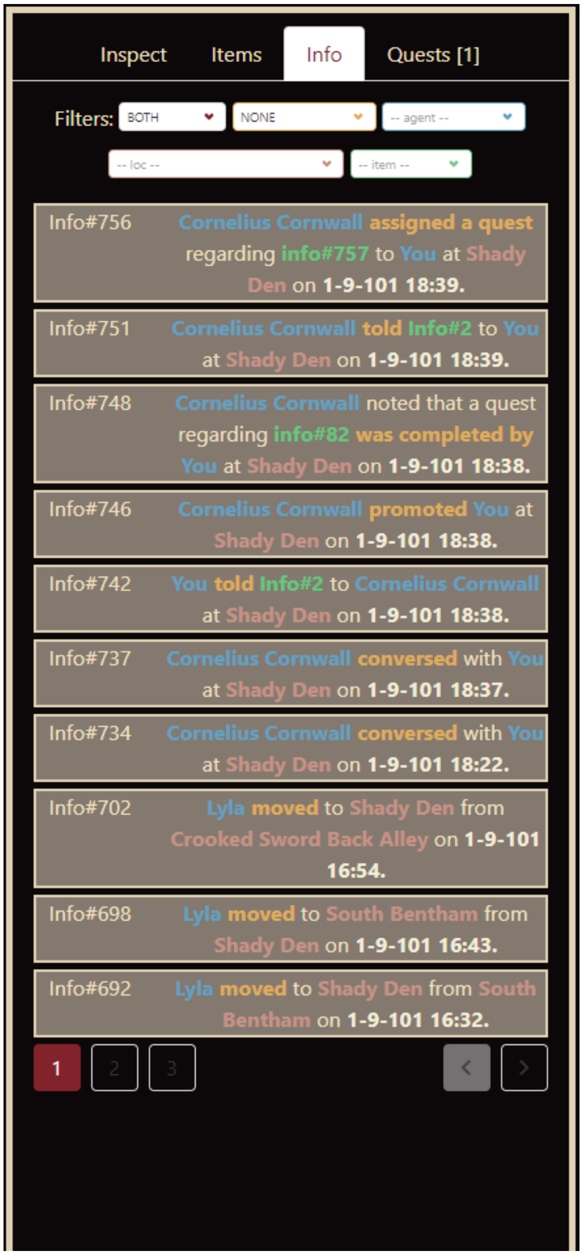
Rewards for completion:
Advancement to next rank

Turned-in Info:
illegal Meth present at Crooked Sword Tavern (back room) on 9-22-97 17:54.

??? ??? on ???.

Conversation requested with Ellie
Ellie has declined your trade request

Figure A.3: Panoptyk Human Client Help Window



Filters: BOTH NONE -- agent --

-- loc -- -- item --

Info#756 Cornelius Cornwall assigned a quest regarding info#757 to You at Shady Den on 1-9-101 18:39.

Info#751 Cornelius Cornwall told Info#2 to You at Shady Den on 1-9-101 18:39.

Info#748 Cornelius Cornwall noted that a quest regarding info#82 was completed by You at Shady Den on 1-9-101 18:38.

Info#746 Cornelius Cornwall promoted You at Shady Den on 1-9-101 18:38.

Info#742 You told Info#2 to Cornelius Cornwall at Shady Den on 1-9-101 18:38.

Info#737 Cornelius Cornwall conversed with You at Shady Den on 1-9-101 18:37.

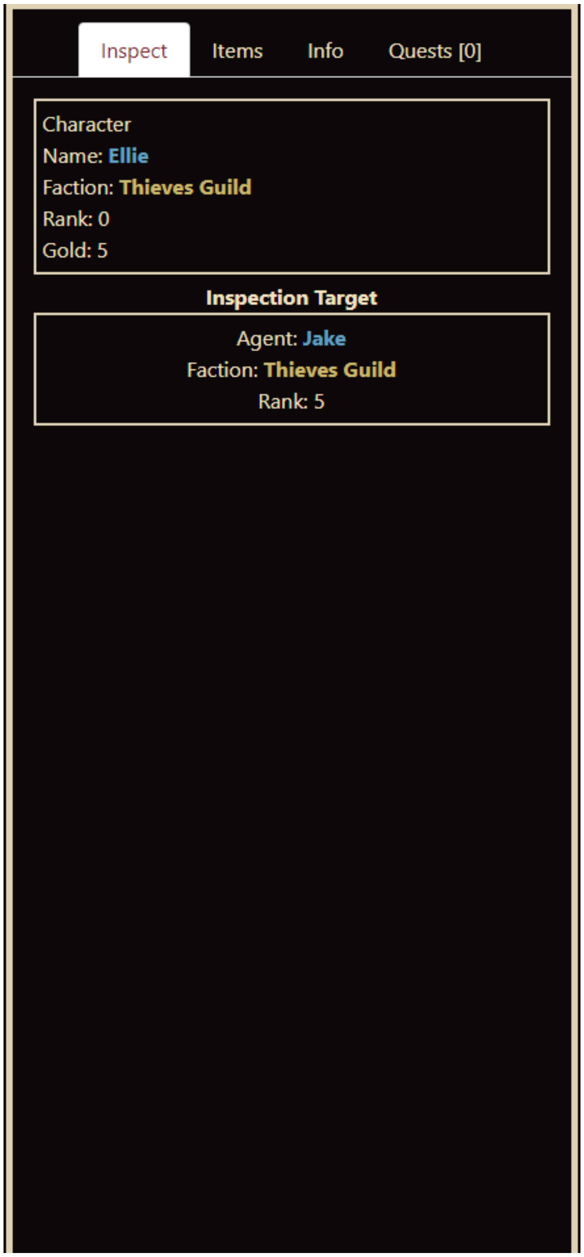
Info#734 Cornelius Cornwall conversed with You at Shady Den on 1-9-101 18:22.

Info#702 Lyla moved to Shady Den from Crooked Sword Back Alley on 1-9-101 16:54.

Info#698 Lyla moved to South Bentham from Shady Den on 1-9-101 16:43.

Info#692 Lyla moved to Shady Den from South Bentham on 1-9-101 16:32.

Figure A.4: Panoptyk Human Client Information Tab

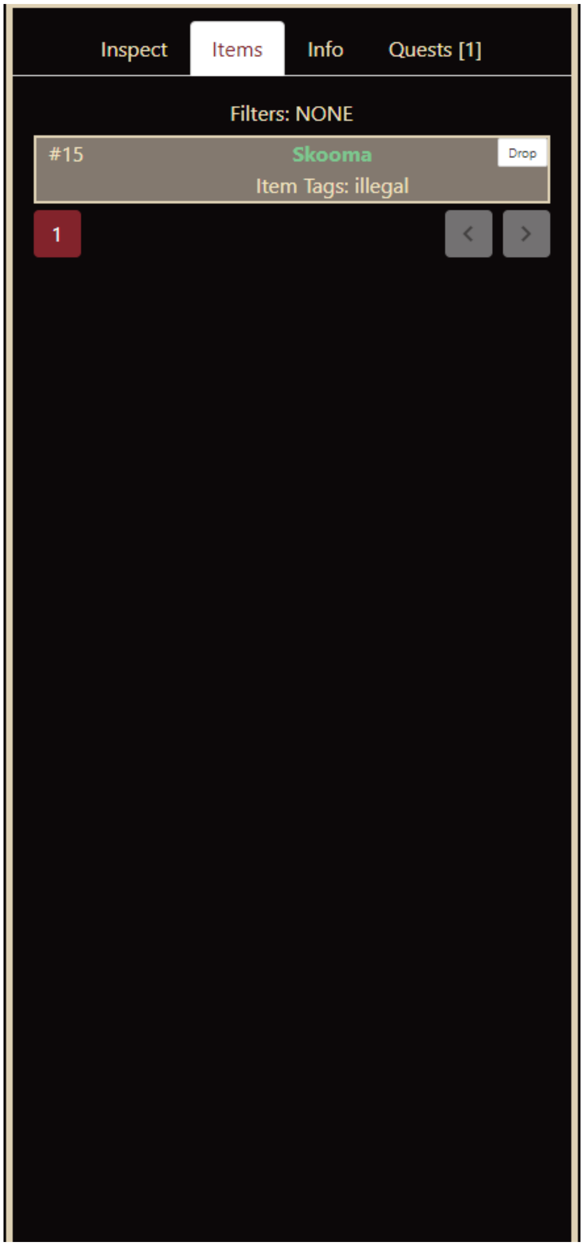


Character
Name: Ellie
Faction: Thieves Guild
Rank: 0
Gold: 5

Inspection Target

Agent: Jake
Faction: Thieves Guild
Rank: 5

Figure A.5: Panoptyk Human Client Inspect Tab

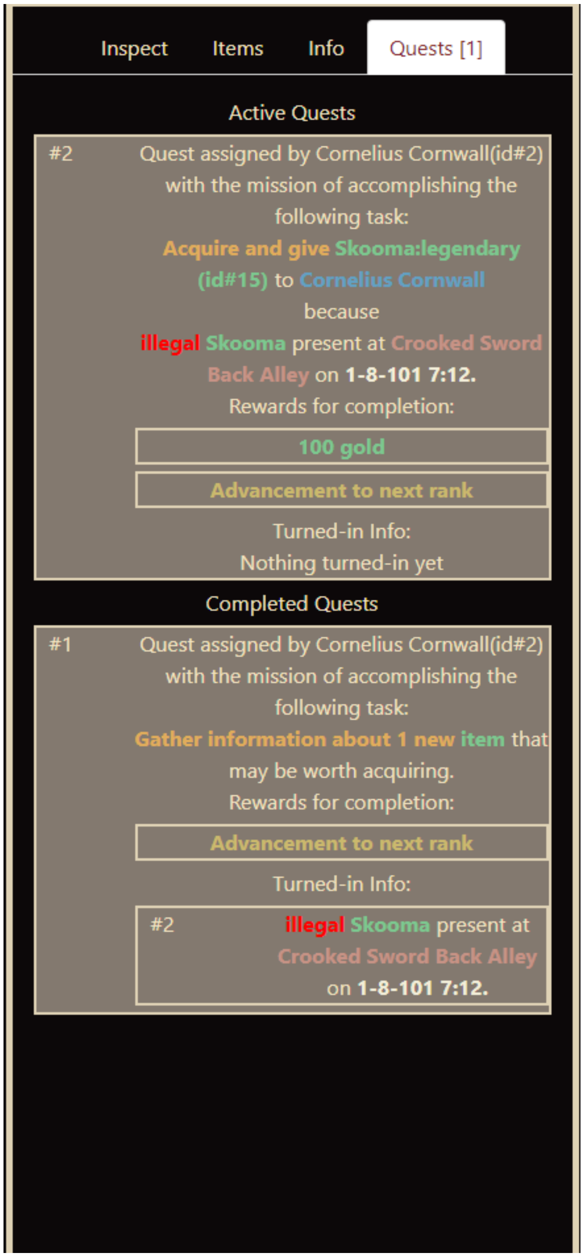


Filters: NONE

#15 Skooma

Item Tags: illegal

Figure A.6: Panoptyk Human Client Items Tab



Active Quests

#2 Quest assigned by Cornelius Cornwall(id#2) with the mission of accomplishing the following task:
Acquire and give Skooma:legendary (id#15) to Cornelius Cornwall
because
illegal Skooma present at Crooked Sword Back Alley on **1-8-101 7:12.**
Rewards for completion:

- 100 gold
- Advancement to next rank

Turned-in Info:
Nothing turned-in yet

Completed Quests

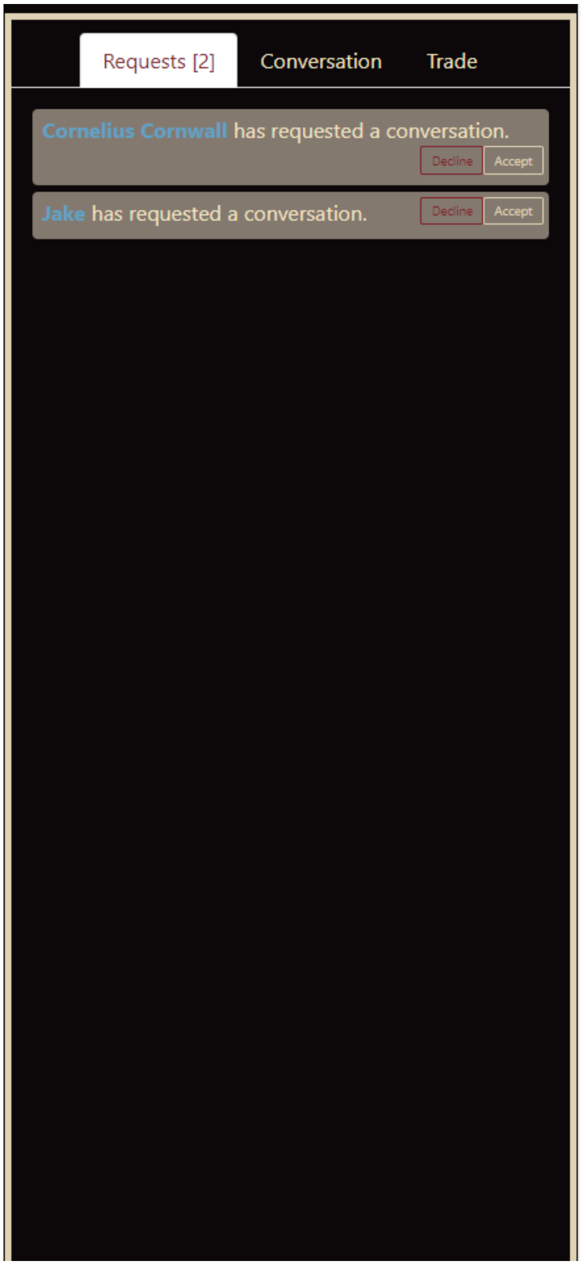
#1 Quest assigned by Cornelius Cornwall(id#2) with the mission of accomplishing the following task:
Gather information about 1 new item that may be worth acquiring.
Rewards for completion:

- Advancement to next rank

Turned-in Info:

#2 illegal Skooma present at Crooked Sword Back Alley on **1-8-101 7:12.**

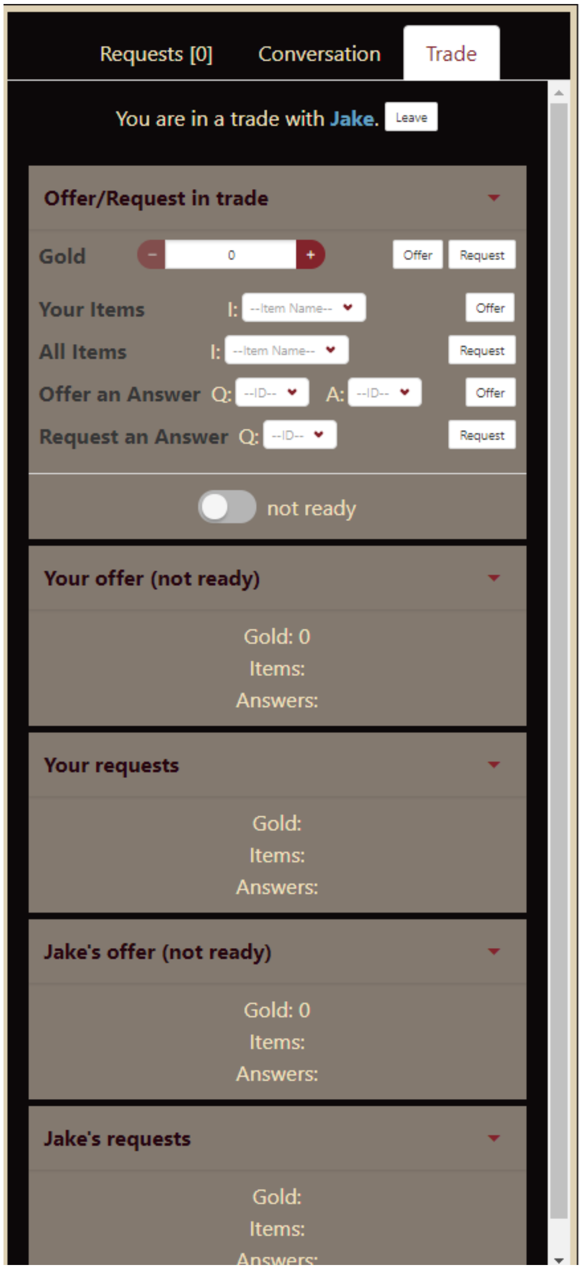
Figure A.7: Panoptyk Human Client Quest Tab



Cornelius Cornwall has requested a conversation.

Jake has requested a conversation.

Figure A.8: Panoptyk Human Client Requests Tab



You are in a trade with Jake.

Offer/Request in trade

Gold 0

Your Items I:

All Items I:

Offer an Answer Q: A:

Request an Answer Q:

not ready

Your offer (not ready)

Gold: 0
Items:
Answers:

Your requests

Gold:
Items:
Answers:

Jake's offer (not ready)

Gold: 0
Items:
Answers:

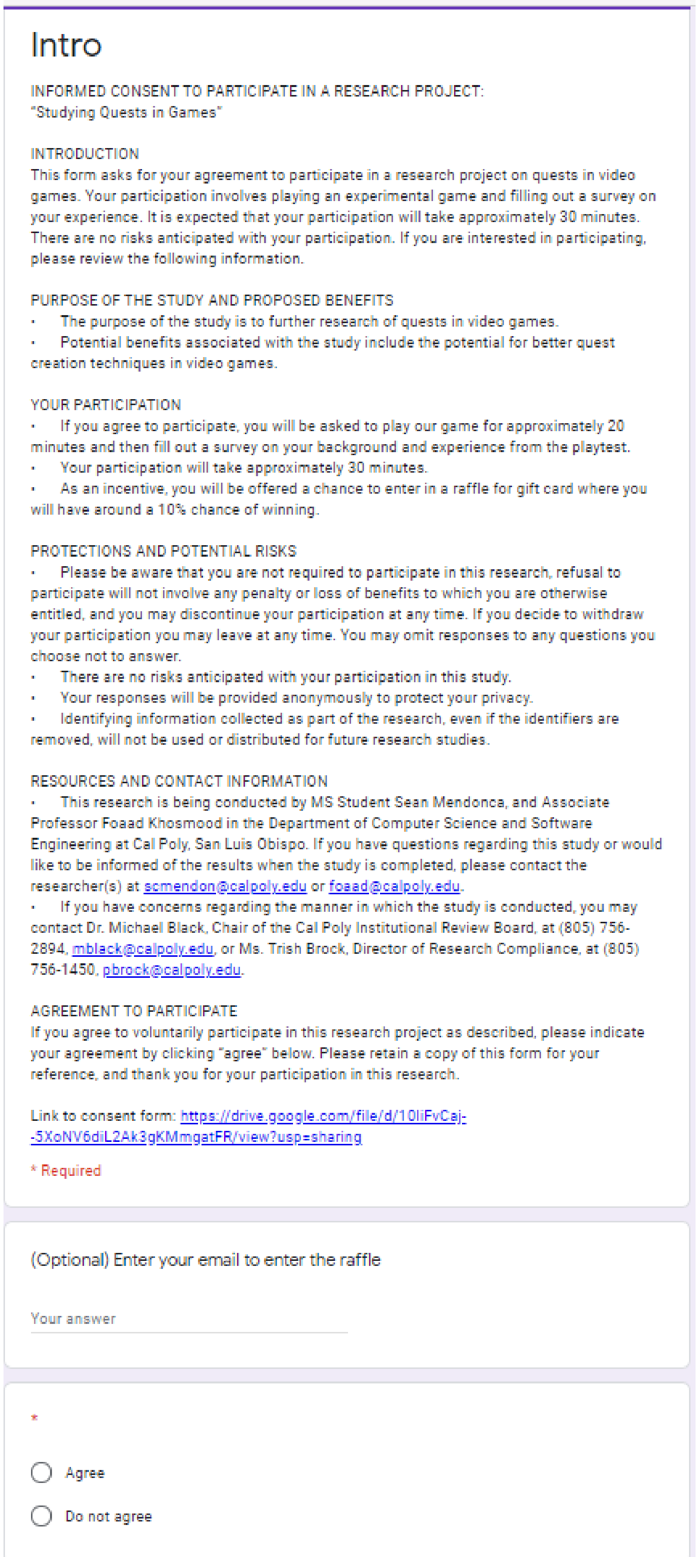
Jake's requests

Gold:
Items:
Answers:

Figure A.9: Panoptyk Human Client Trade Tab

Appendix B

INITIAL STUDY



Intro

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT:
"Studying Quests in Games"

INTRODUCTION
This form asks for your agreement to participate in a research project on quests in video games. Your participation involves playing an experimental game and filling out a survey on your experience. It is expected that your participation will take approximately 30 minutes. There are no risks anticipated with your participation. If you are interested in participating, please review the following information.

PURPOSE OF THE STUDY AND PROPOSED BENEFITS

- The purpose of the study is to further research of quests in video games.
- Potential benefits associated with the study include the potential for better quest creation techniques in video games.

YOUR PARTICIPATION

- If you agree to participate, you will be asked to play our game for approximately 20 minutes and then fill out a survey on your background and experience from the playtest.
- Your participation will take approximately 30 minutes.
- As an incentive, you will be offered a chance to enter in a raffle for gift card where you will have around a 10% chance of winning.

PROTECTIONS AND POTENTIAL RISKS

- Please be aware that you are not required to participate in this research, refusal to participate will not involve any penalty or loss of benefits to which you are otherwise entitled, and you may discontinue your participation at any time. If you decide to withdraw your participation you may leave at any time. You may omit responses to any questions you choose not to answer.
- There are no risks anticipated with your participation in this study.
- Your responses will be provided anonymously to protect your privacy.
- Identifying information collected as part of the research, even if the identifiers are removed, will not be used or distributed for future research studies.

RESOURCES AND CONTACT INFORMATION

- This research is being conducted by MS Student Sean Mendonca, and Associate Professor Foaad Khosmood in the Department of Computer Science and Software Engineering at Cal Poly, San Luis Obispo. If you have questions regarding this study or would like to be informed of the results when the study is completed, please contact the researcher(s) at scmendon@calpoly.edu or foaad@calpoly.edu.
- If you have concerns regarding the manner in which the study is conducted, you may contact Dr. Michael Black, Chair of the Cal Poly Institutional Review Board, at (805) 756-2894, mblack@calpoly.edu, or Ms. Trish Brock, Director of Research Compliance, at (805) 756-1450, pbrock@calpoly.edu.

AGREEMENT TO PARTICIPATE
If you agree to voluntarily participate in this research project as described, please indicate your agreement by clicking "agree" below. Please retain a copy of this form for your reference, and thank you for your participation in this research.

Link to consent form: https://drive.google.com/file/d/10liFvCaj--5XoNV6diL2Ak3gKMmgatFR/view?usp=sharing

* Required

(Optional) Enter your email to enter the raffle

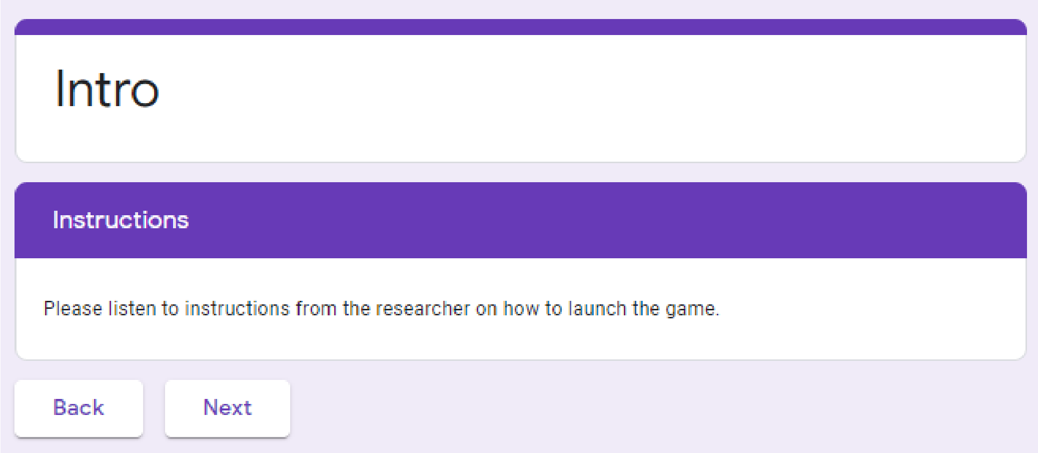
Your answer

*

○ Agree

○ Do not agree

Figure B.1: Consent Form of Survey

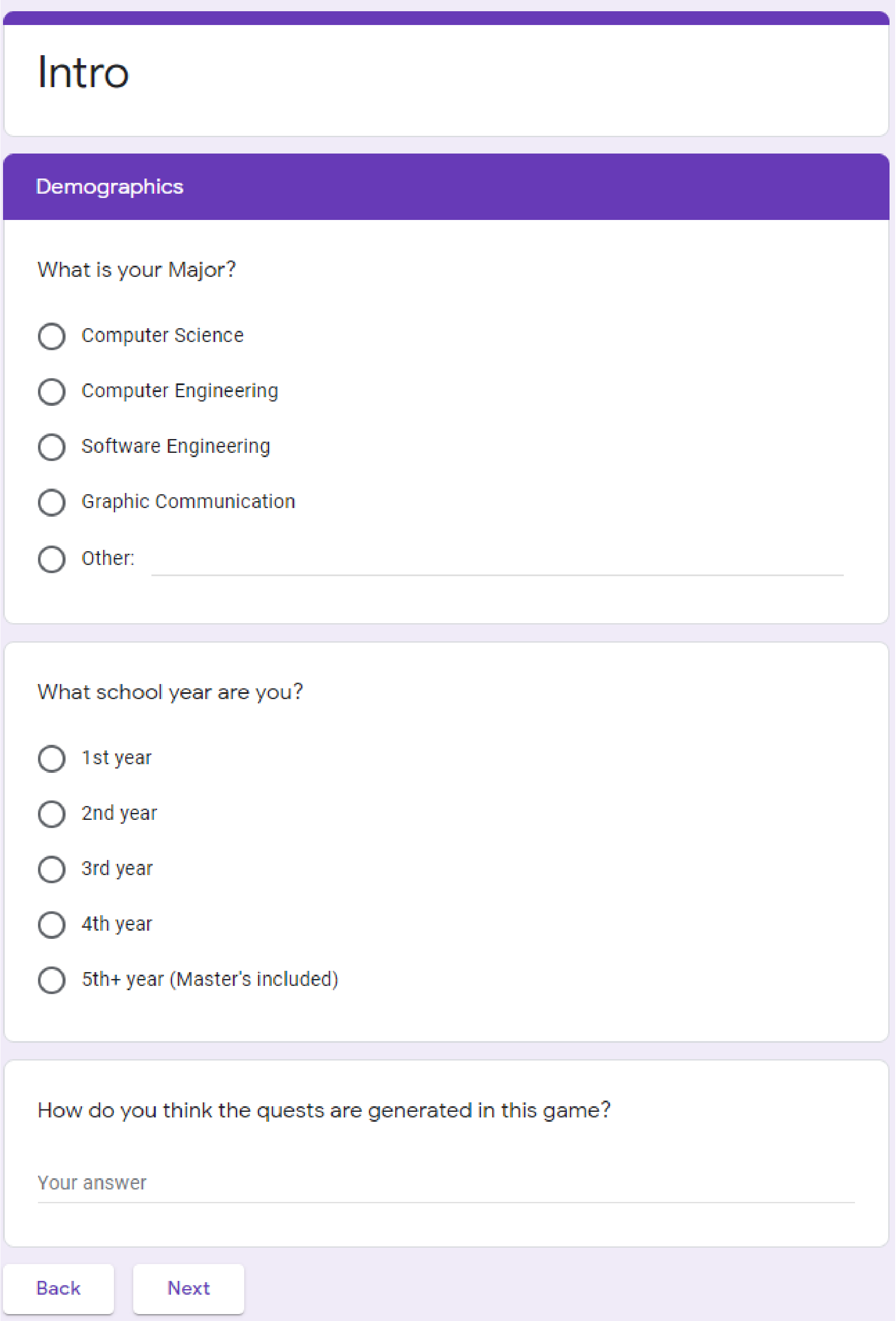


Intro

Instructions

Please listen to instructions from the researcher on how to launch the game.

Figure B.2: First survey instructions



Intro

Demographics

What is your Major?

○ Computer Science

○ Computer Engineering

○ Software Engineering

○ Graphic Communication

○ Other: ____________

What school year are you?

○ 1st year

○ 2nd year

○ 3rd year

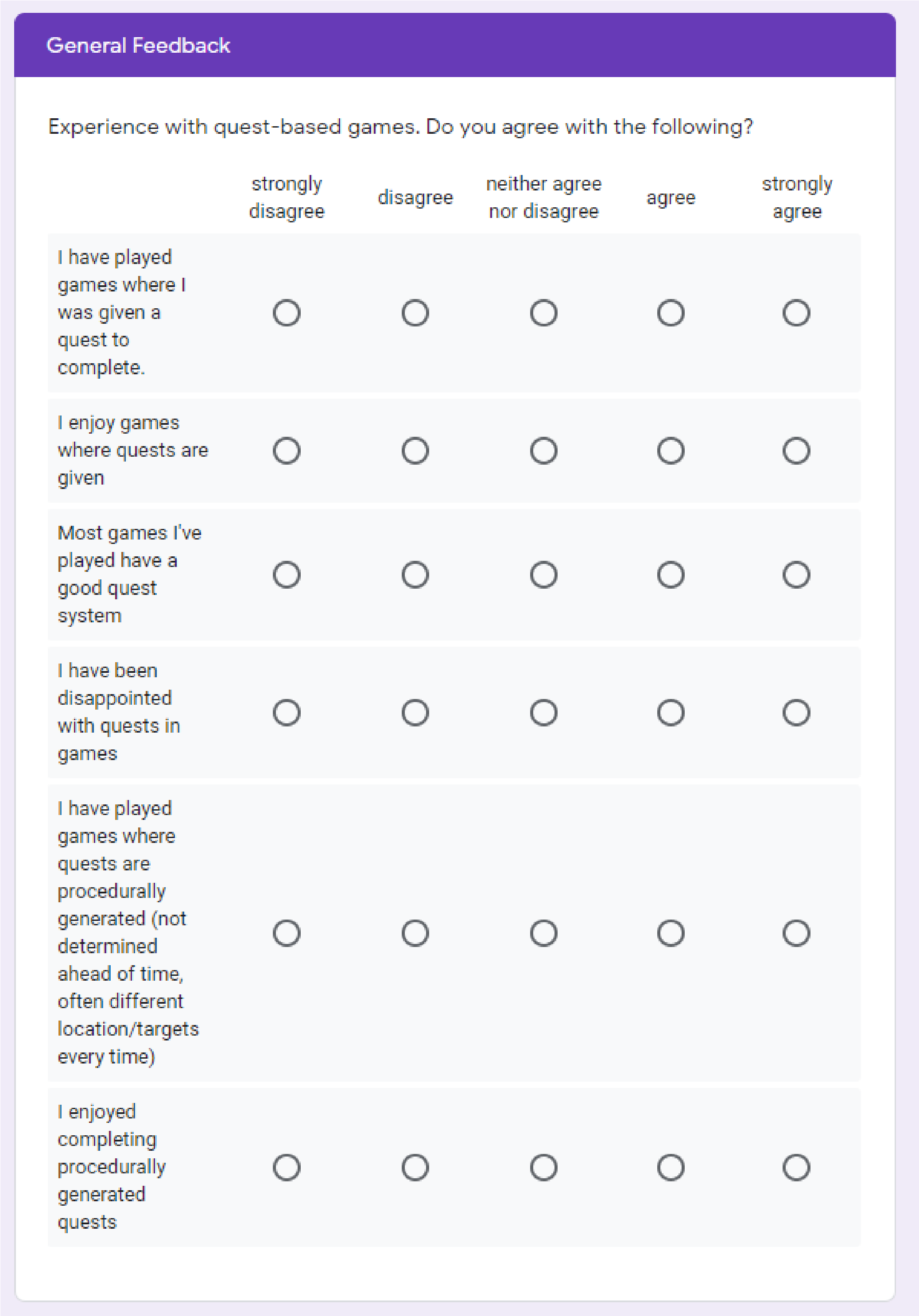
○ 4th year

○ 5th+ year (Master's included)

How do you think the quests are generated in this game?

Your answer ____________

Figure B.3: First survey demographics

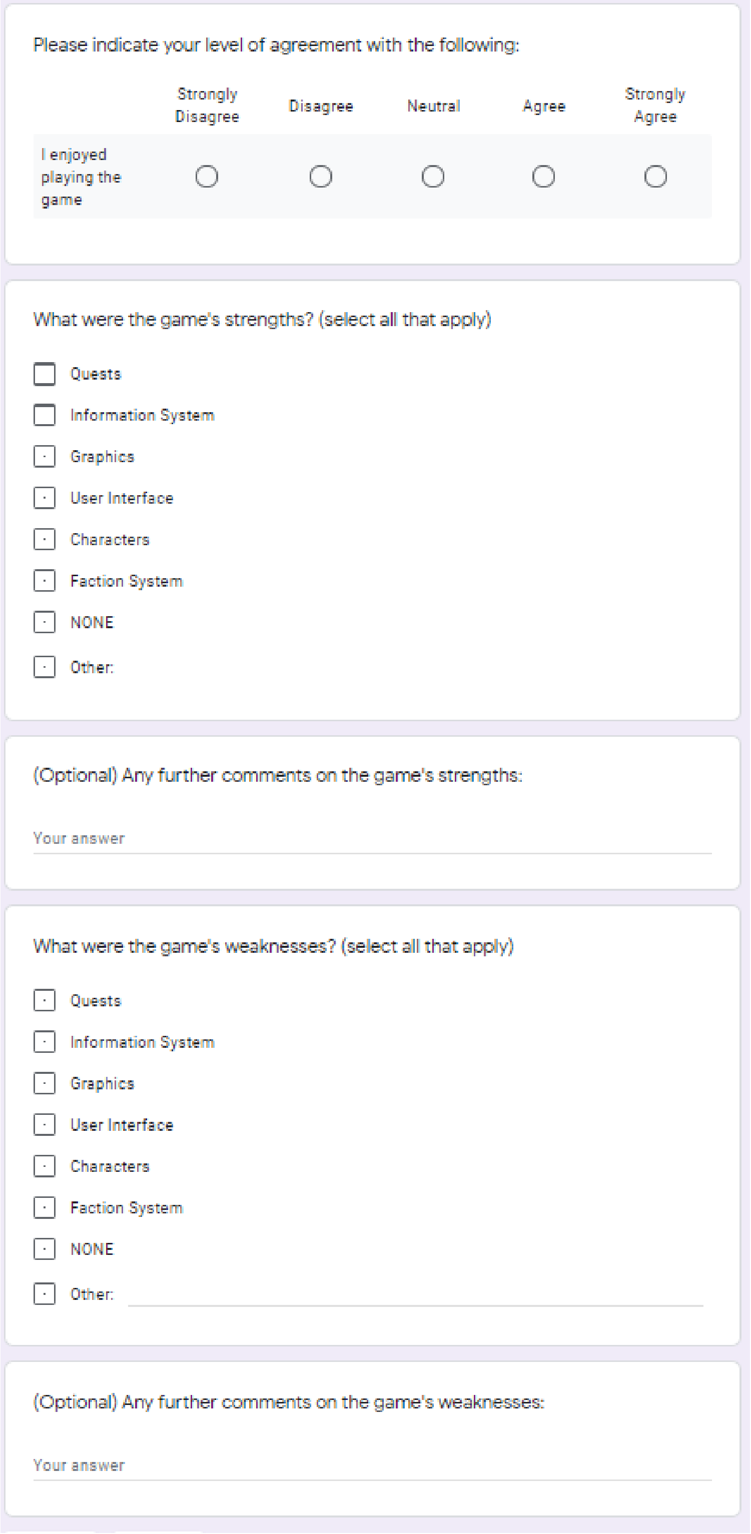


General Feedback

Experience with quest-based games. Do you agree with the following?

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I have played games where I was given a quest to complete.	○	○	○	○	○
I enjoy games where quests are given	○	○	○	○	○
Most games I've played have a good quest system	○	○	○	○	○
I have been disappointed with quests in games	○	○	○	○	○
I have played games where quests are procedurally generated (not determined ahead of time, often different location/targets every time)	○	○	○	○	○
I enjoyed completing procedurally generated quests	○	○	○	○	○

Figure B.4: First survey general feedback



Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I enjoyed playing the game	○	○	○	○	○

What were the game's strengths? (select all that apply)

- □ Quests
- □ Information System
- □ Graphics
- □ User Interface
- □ Characters
- □ Faction System
- □ NONE
- □ Other:

(Optional) Any further comments on the game's strengths:

Your answer

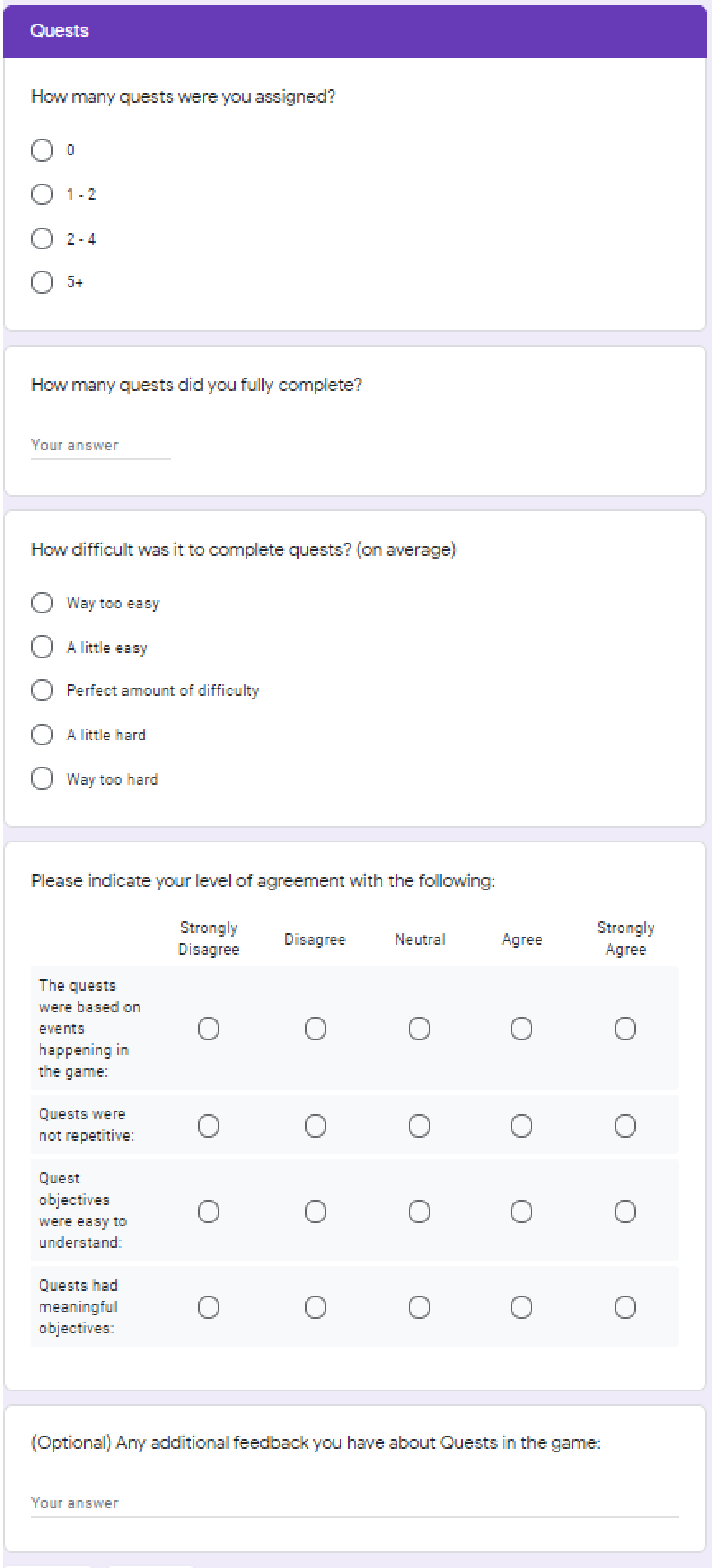
What were the game's weaknesses? (select all that apply)

- □ Quests
- □ Information System
- □ Graphics
- □ User Interface
- □ Characters
- □ Faction System
- □ NONE
- □ Other:

(Optional) Any further comments on the game's weaknesses:

Your answer

Figure B.5: First survey survey general feedback (cont.)



Quests

How many quests were you assigned?

○ 0

○ 1 - 2

○ 2 - 4

○ 5+

How many quests did you fully complete?

Your answer

How difficult was it to complete quests? (on average)

○ Way too easy

○ A little easy

○ Perfect amount of difficulty

○ A little hard

○ Way too hard

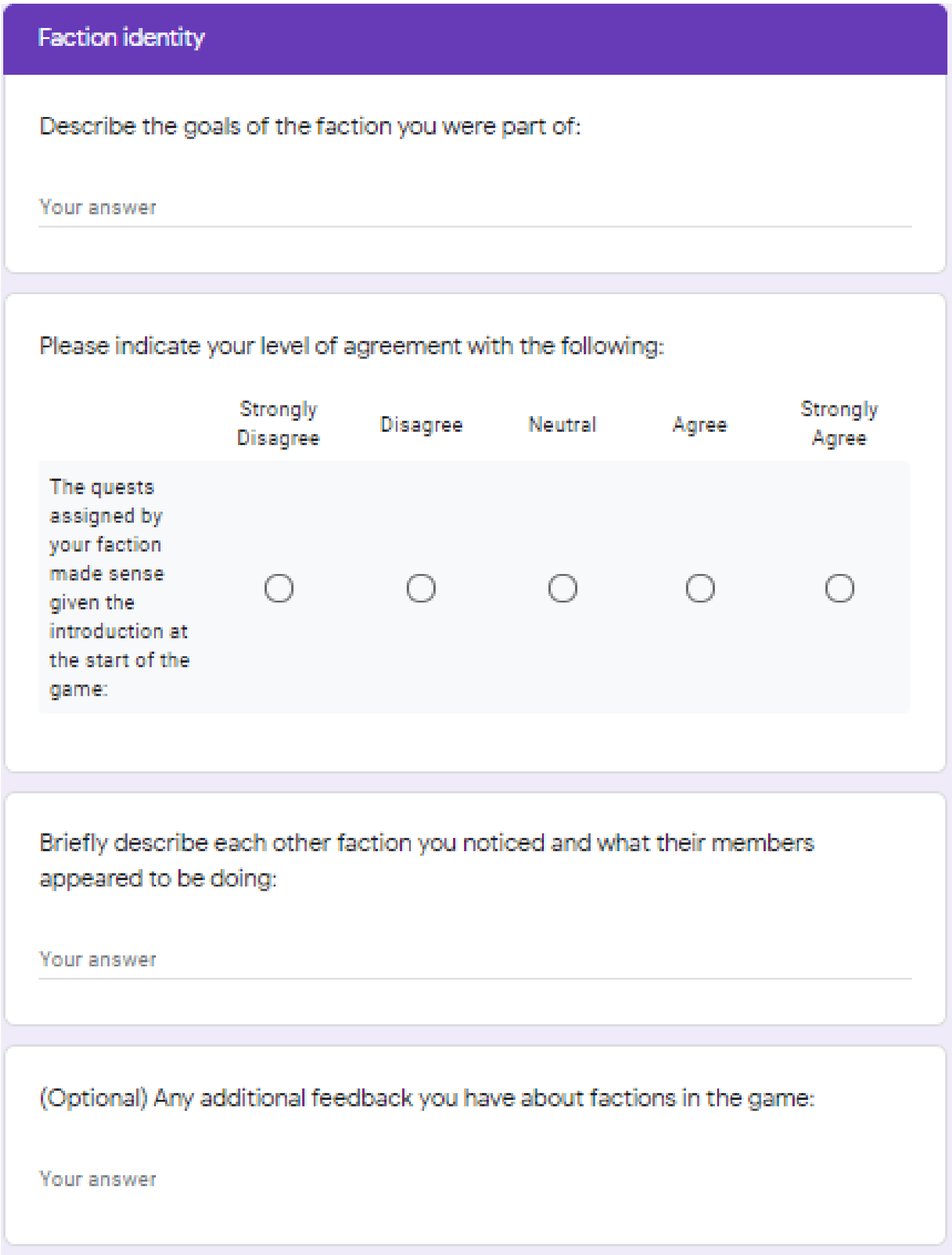
Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The quests were based on events happening in the game:	○	○	○	○	○
Quests were not repetitive:	○	○	○	○	○
Quest objectives were easy to understand:	○	○	○	○	○
Quests had meaningful objectives:	○	○	○	○	○

(Optional) Any additional feedback you have about Quests in the game:

Your answer

Figure B.6: First survey quest feedback



Faction identity

Describe the goals of the faction you were part of:

Your answer

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The quests assigned by your faction made sense given the introduction at the start of the game:	○	○	○	○	○

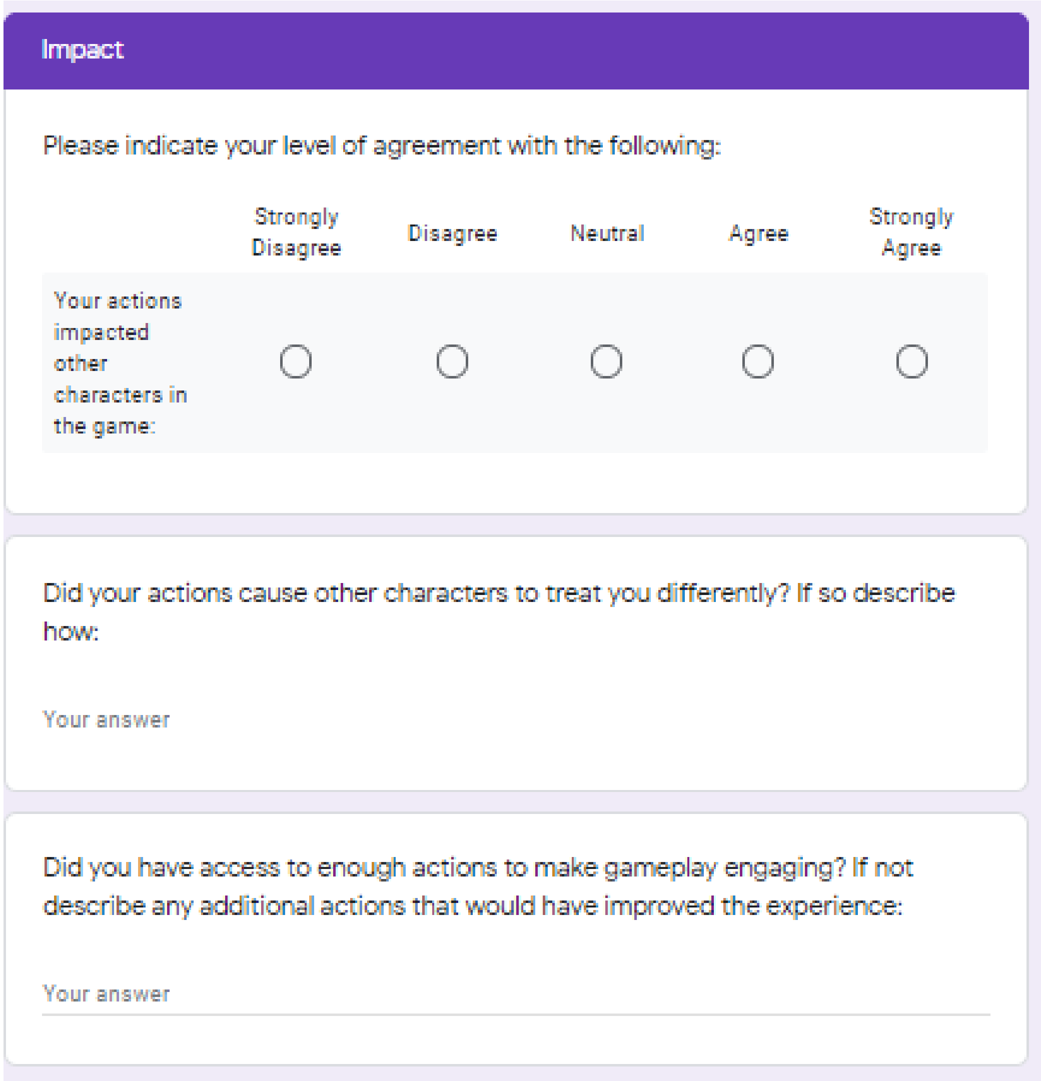
Briefly describe each other faction you noticed and what their members appeared to be doing:

Your answer

(Optional) Any additional feedback you have about factions in the game:

Your answer

Figure B.7: First survey faction feedback



Impact

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Your actions impacted other characters in the game:	○	○	○	○	○

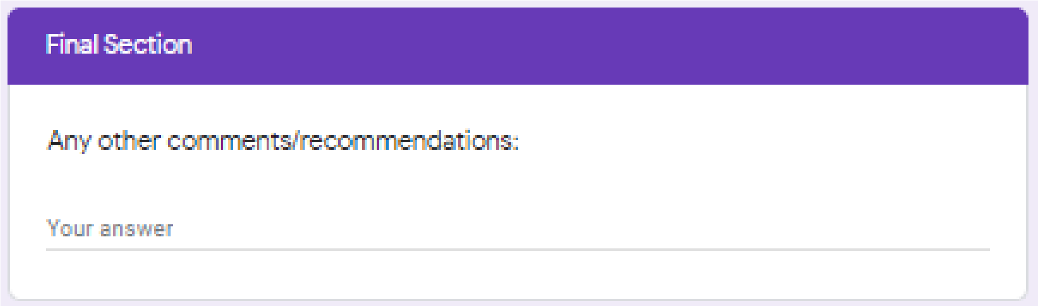
Did your actions cause other characters to treat you differently? If so describe how:

Your answer

Did you have access to enough actions to make gameplay engaging? If not describe any additional actions that would have improved the experience:

Your answer

Figure B.8: First survey impact feedback



Final Section

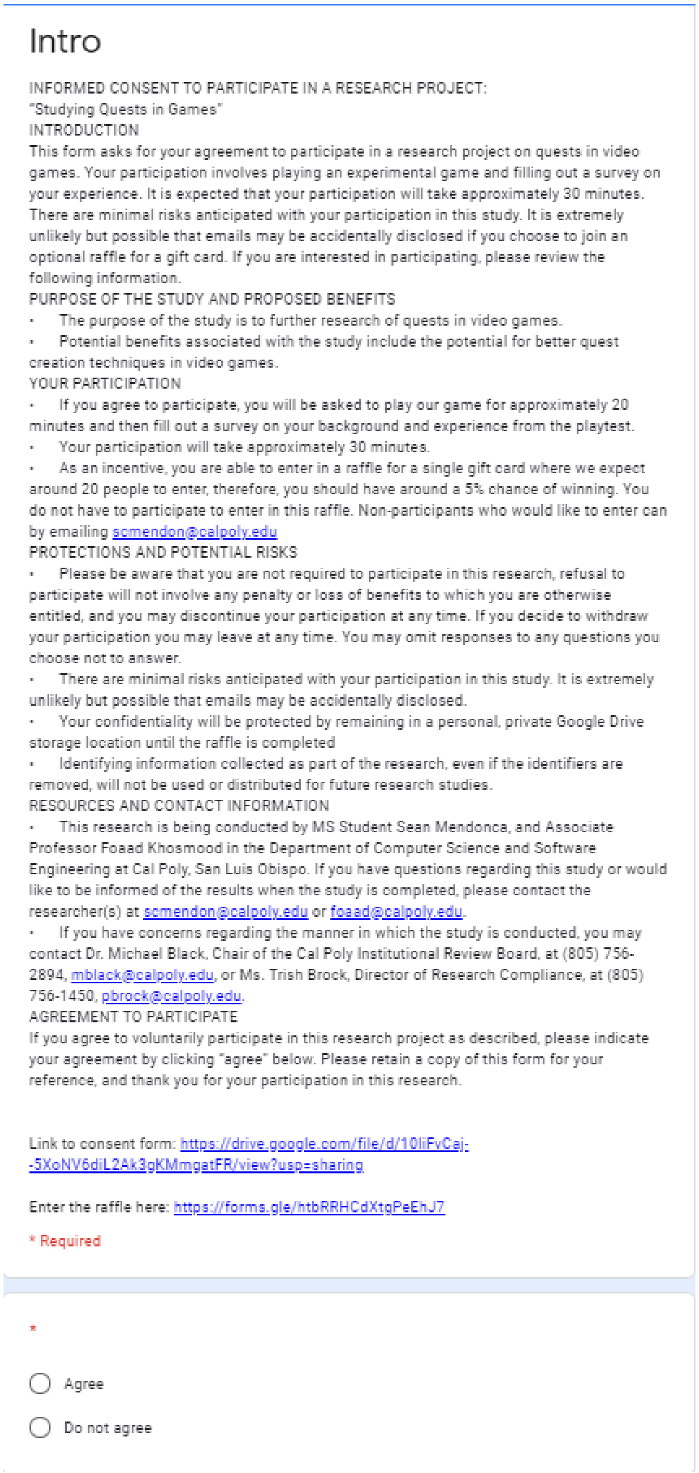
Any other comments/recommendations:

Your answer

Figure B.9: First survey end

Appendix C

REVISED STUDY



Intro

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT:
"Studying Quests in Games"
INTRODUCTION
This form asks for your agreement to participate in a research project on quests in video games. Your participation involves playing an experimental game and filling out a survey on your experience. It is expected that your participation will take approximately 30 minutes. There are minimal risks anticipated with your participation in this study. It is extremely unlikely but possible that emails may be accidentally disclosed if you choose to join an optional raffle for a gift card. If you are interested in participating, please review the following information.

PURPOSE OF THE STUDY AND PROPOSED BENEFITS

- The purpose of the study is to further research of quests in video games.
- Potential benefits associated with the study include the potential for better quest creation techniques in video games.

YOUR PARTICIPATION

- If you agree to participate, you will be asked to play our game for approximately 20 minutes and then fill out a survey on your background and experience from the playtest.
- Your participation will take approximately 30 minutes.
- As an incentive, you are able to enter in a raffle for a single gift card where we expect around 20 people to enter, therefore, you should have around a 5% chance of winning. You do not have to participate to enter in this raffle. Non-participants who would like to enter can by emailing scmendon@calpoly.edu

PROTECTIONS AND POTENTIAL RISKS

- Please be aware that you are not required to participate in this research, refusal to participate will not involve any penalty or loss of benefits to which you are otherwise entitled, and you may discontinue your participation at any time. If you decide to withdraw your participation you may leave at any time. You may omit responses to any questions you choose not to answer.
- There are minimal risks anticipated with your participation in this study. It is extremely unlikely but possible that emails may be accidentally disclosed.
- Your confidentiality will be protected by remaining in a personal, private Google Drive storage location until the raffle is completed
- Identifying information collected as part of the research, even if the identifiers are removed, will not be used or distributed for future research studies.

RESOURCES AND CONTACT INFORMATION

- This research is being conducted by MS Student Sean Mendonca, and Associate Professor Foaad Khosmood in the Department of Computer Science and Software Engineering at Cal Poly, San Luis Obispo. If you have questions regarding this study or would like to be informed of the results when the study is completed, please contact the researcher(s) at scmendon@calpoly.edu or foaad@calpoly.edu.
- If you have concerns regarding the manner in which the study is conducted, you may contact Dr. Michael Black, Chair of the Cal Poly Institutional Review Board, at (805) 756-2894, mblack@calpoly.edu, or Ms. Trish Brock, Director of Research Compliance, at (805) 756-1450, pbrock@calpoly.edu.

AGREEMENT TO PARTICIPATE
If you agree to voluntarily participate in this research project as described, please indicate your agreement by clicking "agree" below. Please retain a copy of this form for your reference, and thank you for your participation in this research.

Link to consent form: https://drive.google.com/file/d/10liFvCaj--5XoNV6diL2Ak3gKMmgatFR/view?usp=sharing

Enter the raffle here: https://forms.gle/htbRRHCdXtgPeEhJ7

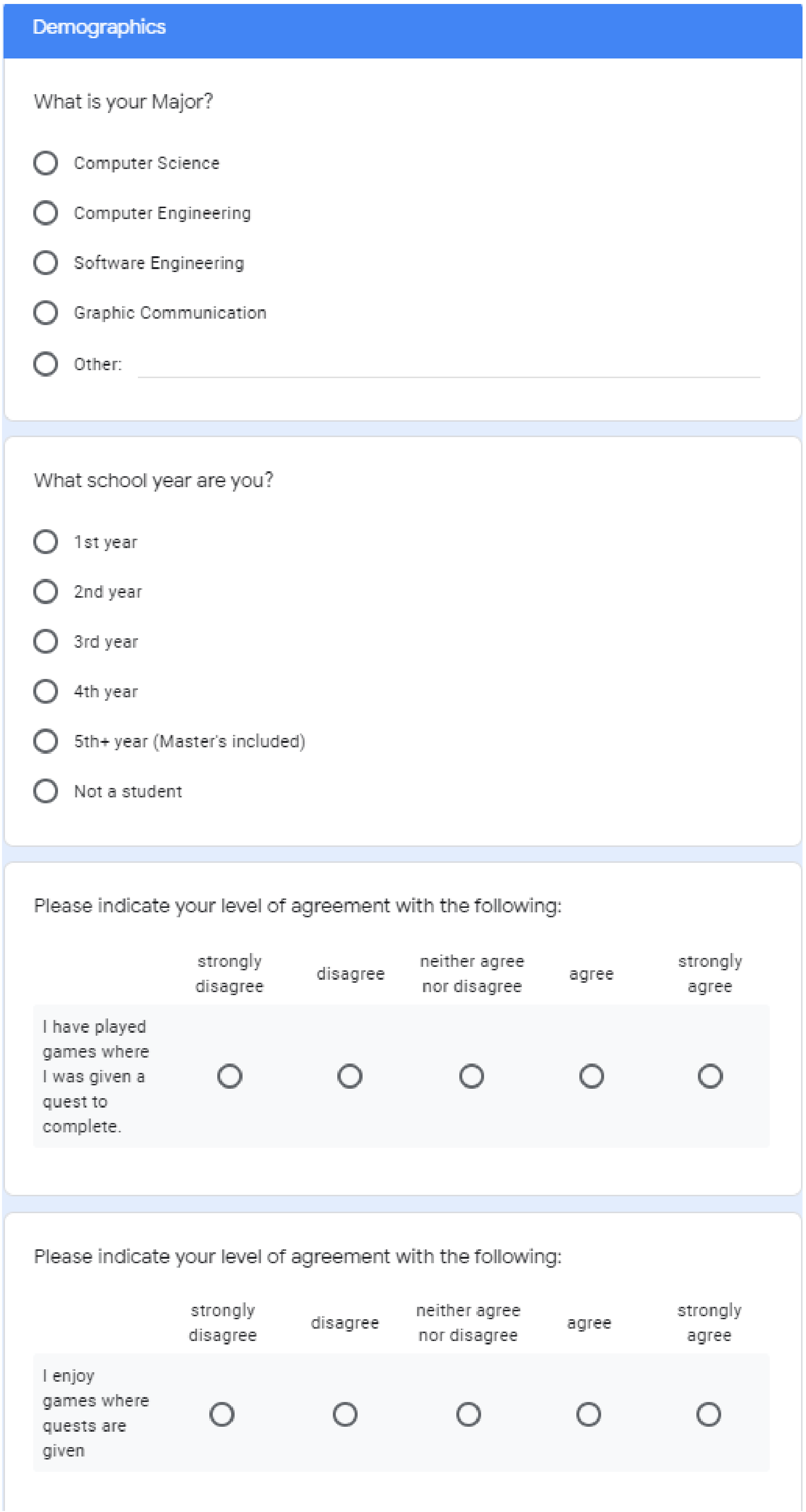
* Required

*

○ Agree

○ Do not agree

Figure C.1: Consent Form of Survey



Demographics

What is your Major?

○ Computer Science

○ Computer Engineering

○ Software Engineering

○ Graphic Communication

○ Other: ____________________

What school year are you?

○ 1st year

○ 2nd year

○ 3rd year

○ 4th year

○ 5th+ year (Master's included)

○ Not a student

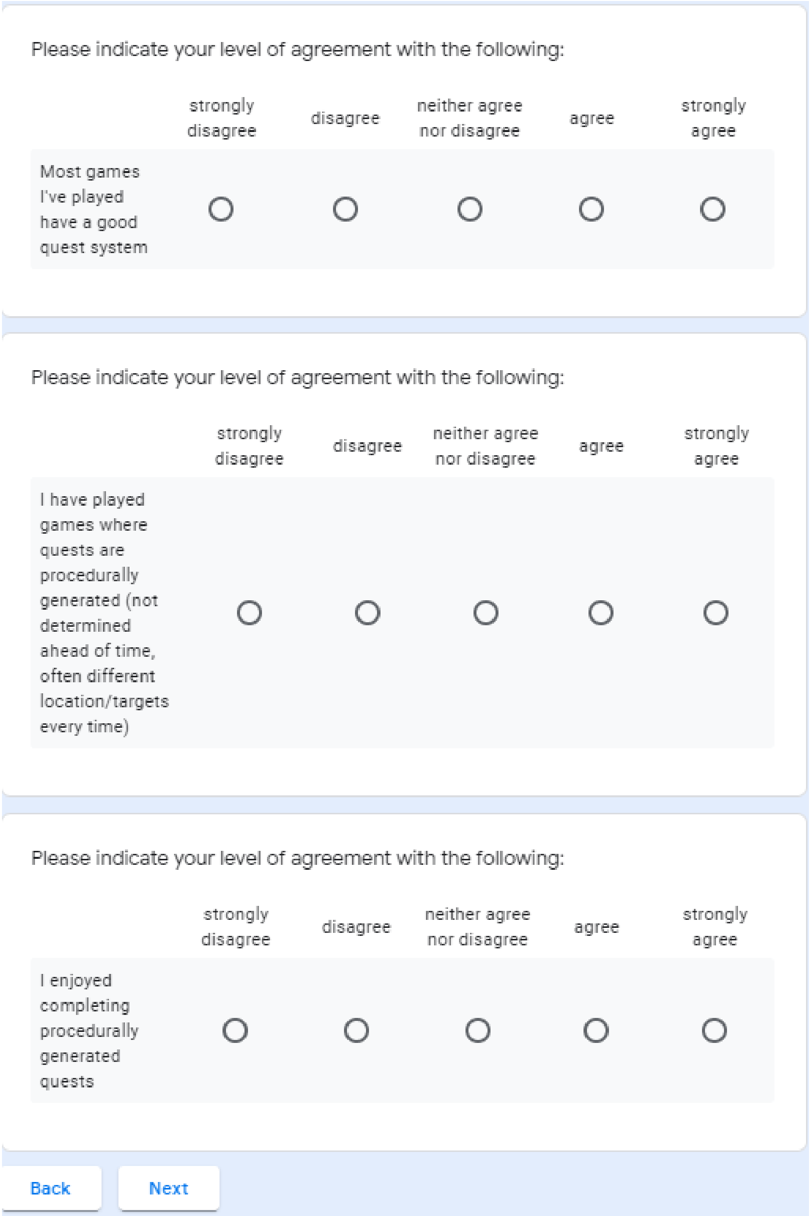
Please indicate your level of agreement with the following:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I have played games where I was given a quest to complete.	○	○	○	○	○

Please indicate your level of agreement with the following:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I enjoy games where quests are given	○	○	○	○	○

Figure C.2: Revised survey demographics 1



Please indicate your level of agreement with the following:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
Most games I've played have a good quest system	○	○	○	○	○

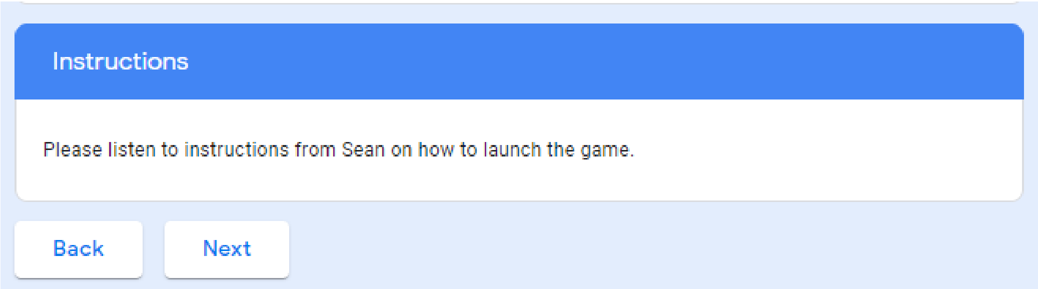
Please indicate your level of agreement with the following:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I have played games where quests are procedurally generated (not determined ahead of time, often different location/targets every time)	○	○	○	○	○

Please indicate your level of agreement with the following:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I enjoyed completing procedurally generated quests	○	○	○	○	○

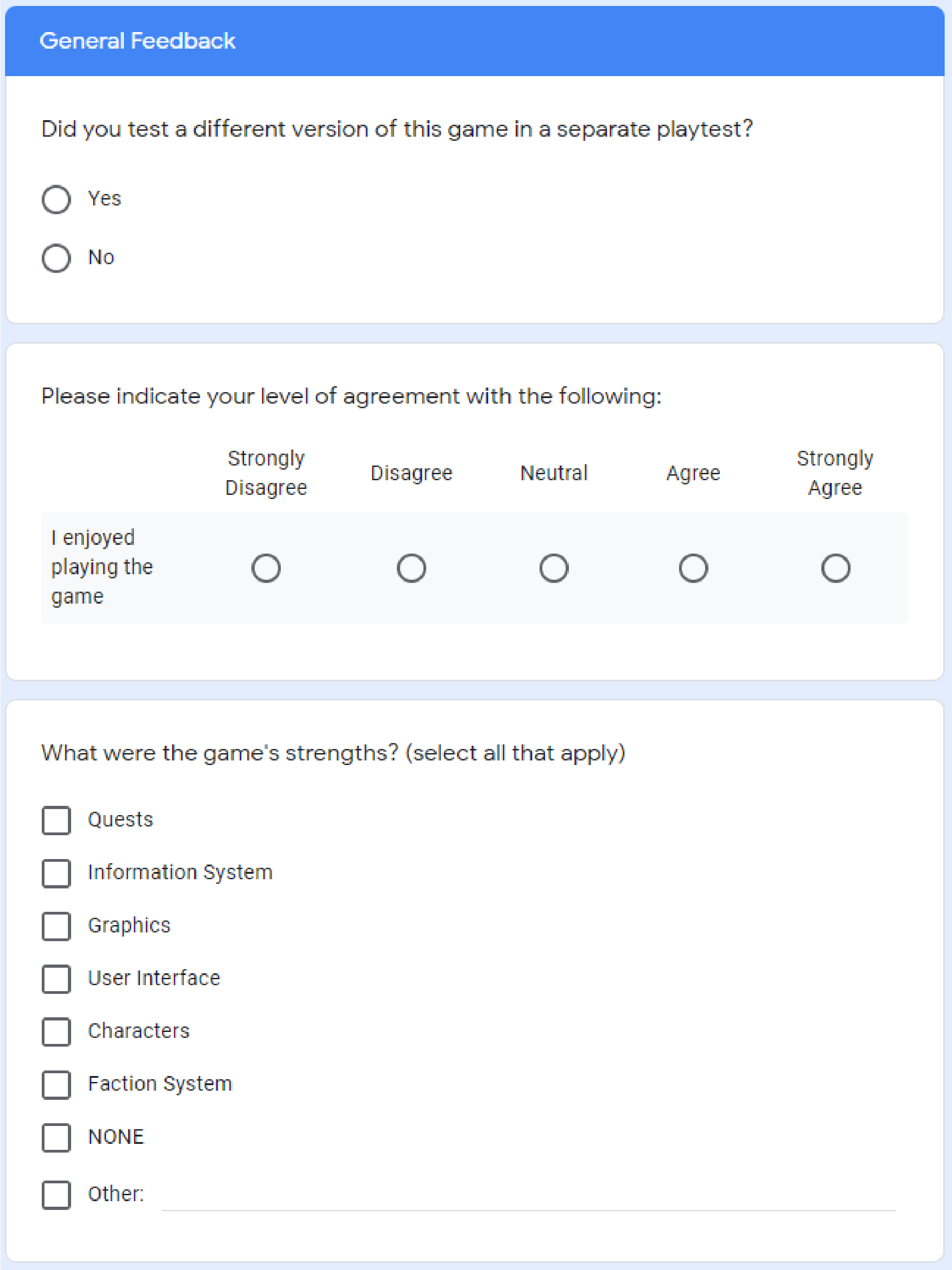
Figure C.3: Revised survey demographics 2



Instructions

Please listen to instructions from Sean on how to launch the game.

Figure C.4: Revised survey instructions



General Feedback

Did you test a different version of this game in a separate playtest?

○ Yes

○ No

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I enjoyed playing the game	○	○	○	○	○

What were the game's strengths? (select all that apply)

□ Quests

□ Information System

□ Graphics

□ User Interface

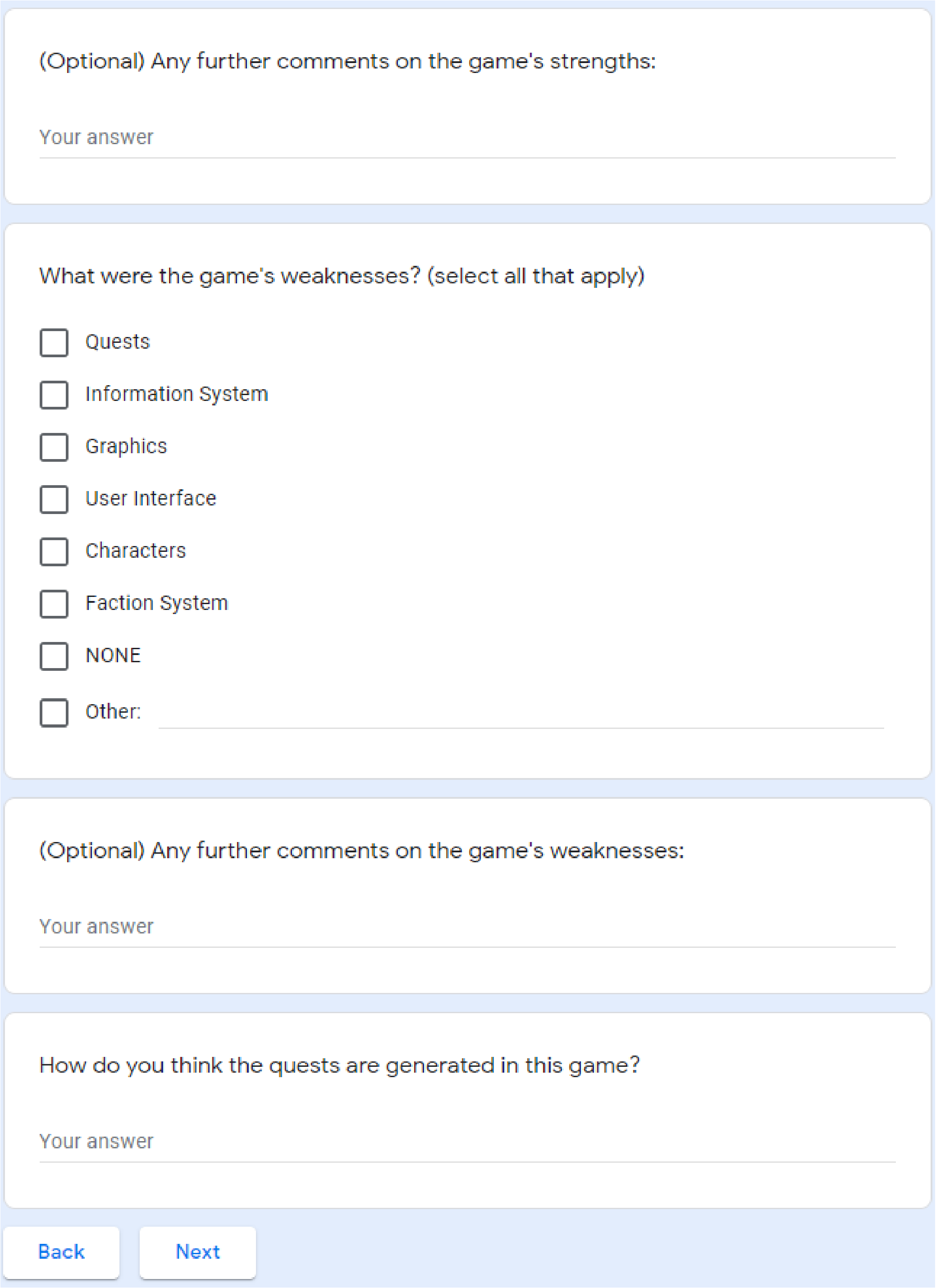
□ Characters

□ Faction System

□ NONE

□ Other: ____________

Figure C.5: Revised survey survey general feedback 1



(Optional) Any further comments on the game's strengths:

Your answer

What were the game's weaknesses? (select all that apply)

- □ Quests
- □ Information System
- □ Graphics
- □ User Interface
- □ Characters
- □ Faction System
- □ NONE
- □ Other: ________

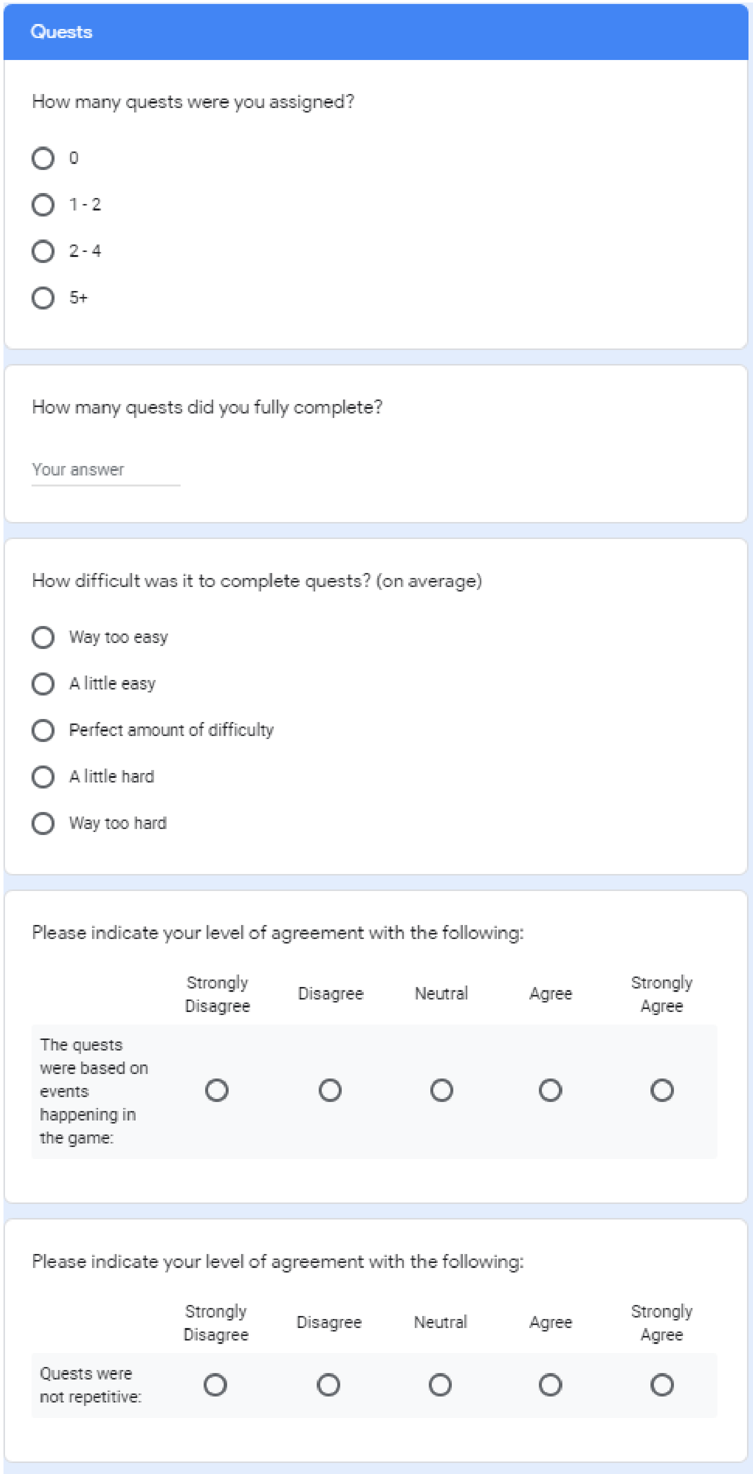
(Optional) Any further comments on the game's weaknesses:

Your answer

How do you think the quests are generated in this game?

Your answer

Figure C.6: Revised survey survey general feedback 2



Quests

How many quests were you assigned?

○ 0

○ 1 - 2

○ 2 - 4

○ 5+

How many quests did you fully complete?

Your answer

How difficult was it to complete quests? (on average)

○ Way too easy

○ A little easy

○ Perfect amount of difficulty

○ A little hard

○ Way too hard

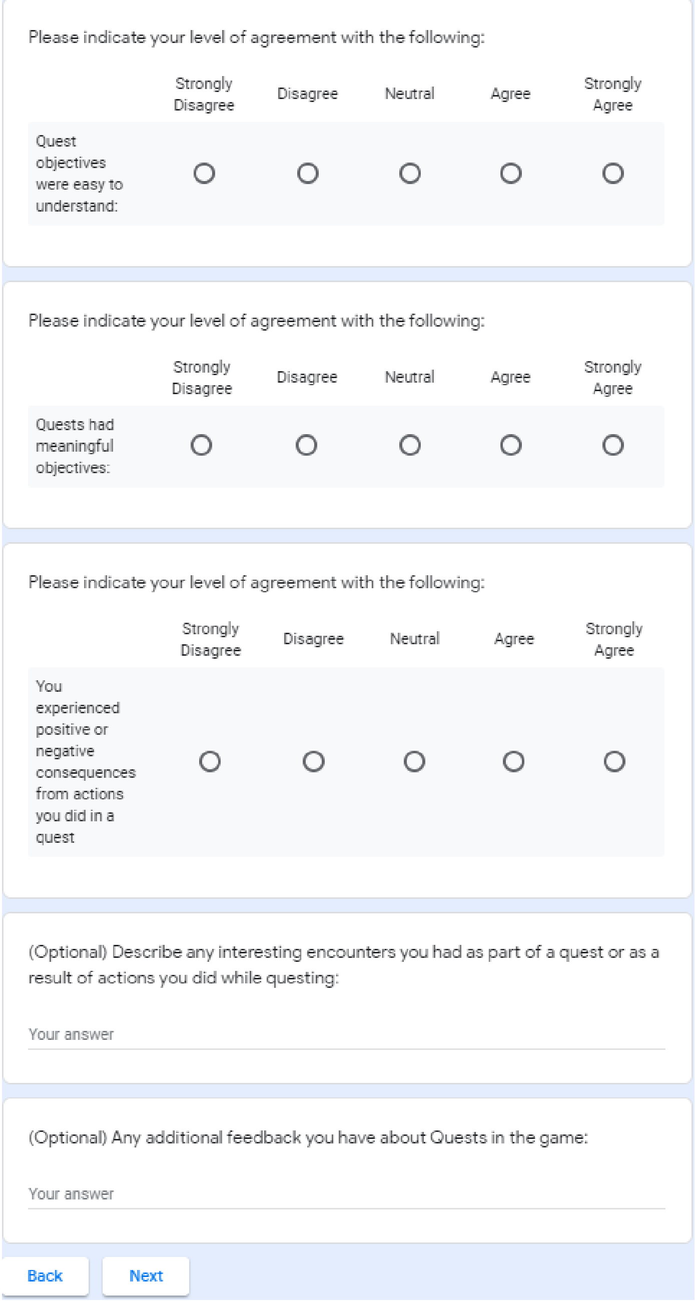
Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The quests were based on events happening in the game:	○	○	○	○	○

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Quests were not repetitive:	○	○	○	○	○

Figure C.7: Revised survey quest feedback 1



Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Quest objectives were easy to understand:	○	○	○	○	○

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Quests had meaningful objectives:	○	○	○	○	○

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
You experienced positive or negative consequences from actions you did in a quest	○	○	○	○	○

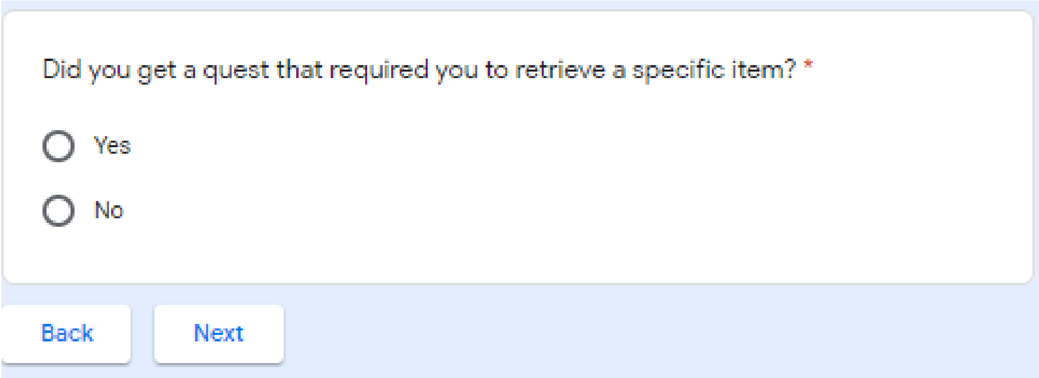
(Optional) Describe any interesting encounters you had as part of a quest or as a result of actions you did while questing:

Your answer

(Optional) Any additional feedback you have about Quests in the game:

Your answer

Figure C.8: Revised survey quest feedback 2

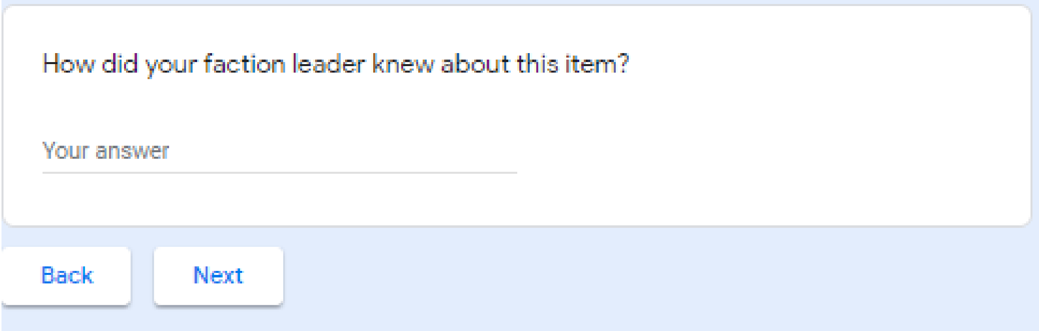


Did you get a quest that required you to retrieve a specific item? *

○ Yes

○ No

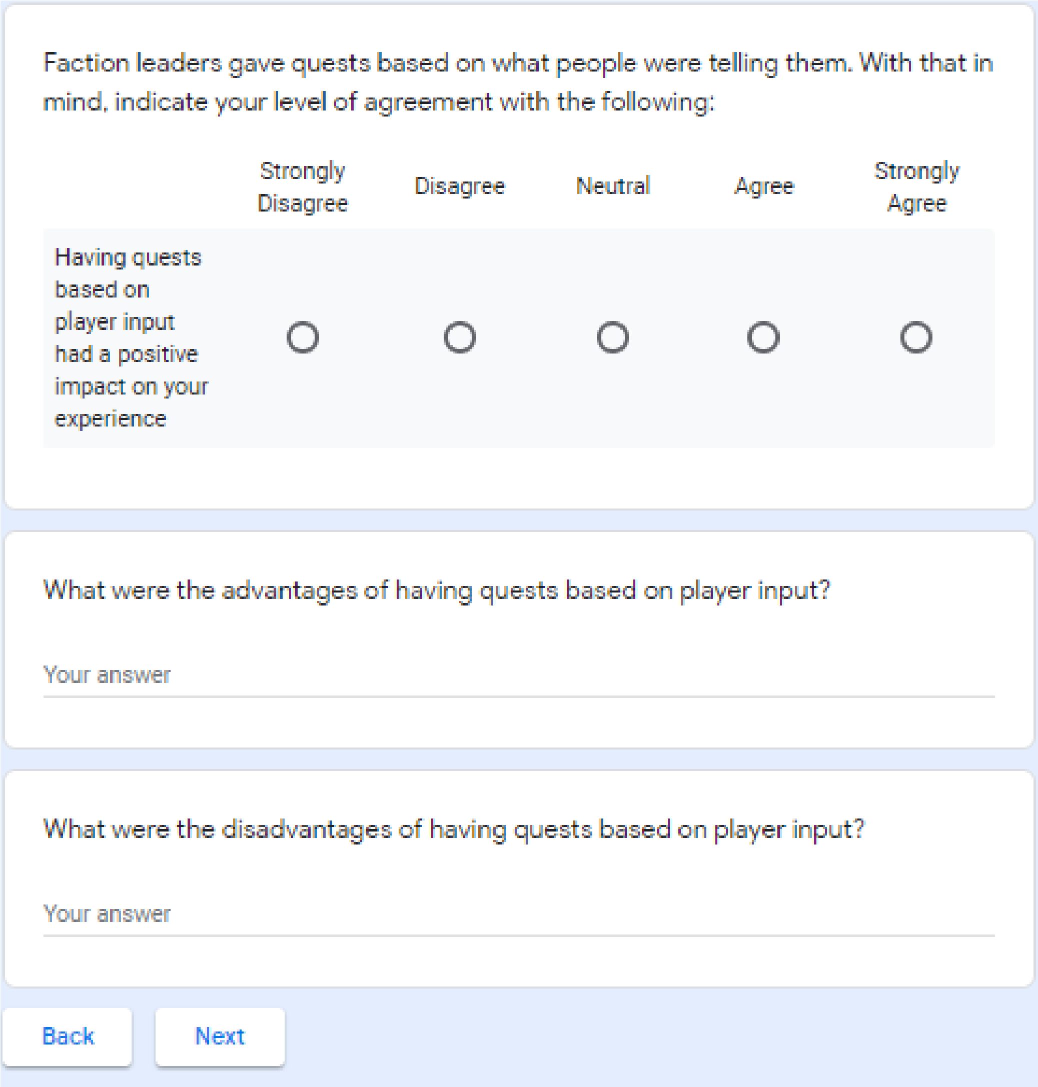
Figure C.9: Revised survey item quest 1



How did your faction leader knew about this item?

Your answer

Figure C.10: Revised survey item quest 2



Faction leaders gave quests based on what people were telling them. With that in mind, indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Having quests based on player input had a positive impact on your experience	○	○	○	○	○

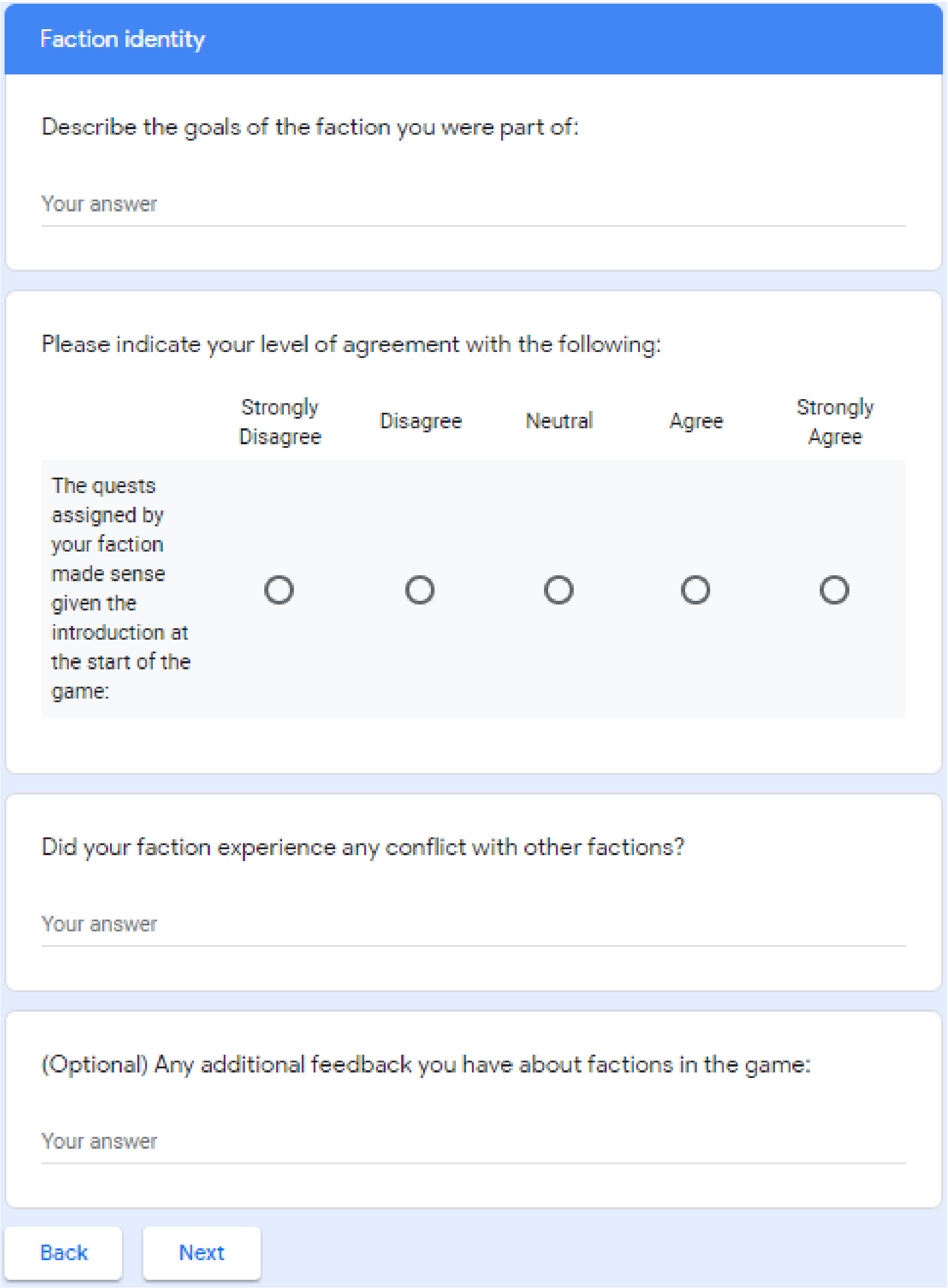
What were the advantages of having quests based on player input?

Your answer

What were the disadvantages of having quests based on player input?

Your answer

Figure C.11: Revised survey player driven quests feedback



Faction identity

Describe the goals of the faction you were part of:

Your answer

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The quests assigned by your faction made sense given the introduction at the start of the game:	○	○	○	○	○

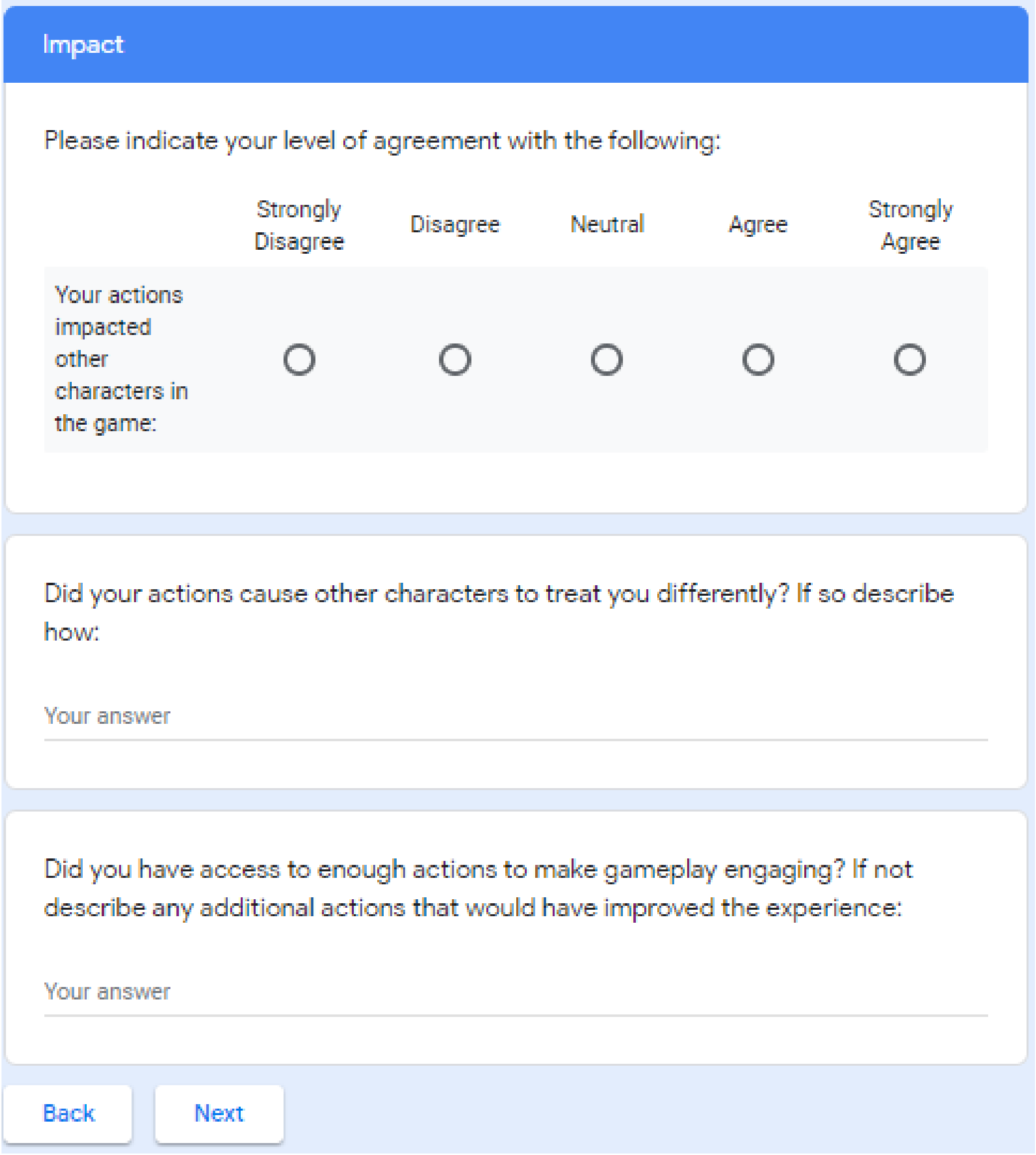
Did your faction experience any conflict with other factions?

Your answer

(Optional) Any additional feedback you have about factions in the game:

Your answer

Figure C.12: Revised survey faction feedback



Impact

Please indicate your level of agreement with the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Your actions impacted other characters in the game:	○	○	○	○	○

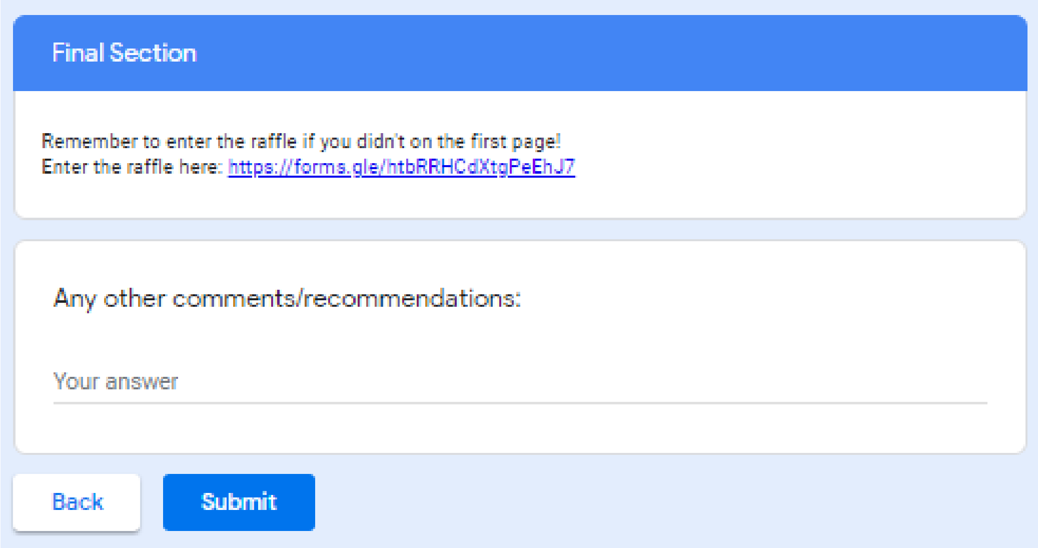
Did your actions cause other characters to treat you differently? If so describe how:

Your answer

Did you have access to enough actions to make gameplay engaging? If not describe any additional actions that would have improved the experience:

Your answer

Figure C.13: Revised survey impact feedback



Final Section

Remember to enter the raffle if you didn't on the first page!
Enter the raffle here: https://forms.gle/htbRRHCdXtgPeEhJ7

Any other comments/recommendations:

Your answer

Figure C.14: Revised survey end