BOOTSTRAPPING MASSIVELY MULTIPLAYER ONLINE ROLE PLAYING GAMES

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

Mitchell Miller

June 2020

COMMITTEE MEMBERSHIP

TITLE:	Bootstrapping Massively Multiplayer Online Role Playing Games
AUTHOR:	Mitchell Miller
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

Bootstrapping Massively Multiplayer Online Role Playing Games

Mitchell Miller

Massively Multiplayer Online Role Playing Games (MMORPGs) are a prominent genre in today's video game industry with the most popular MMORPGs generating billions of dollars in revenue and attracting millions of players. As they have grown, they have become a major target for both technological research and sociological research. In such research, it is nearly impossible to reach the same player scale from any self-made technology or sociological experiments. This greatly limits the amount of control and topics that can be explored. In an effort to make up a lacking or non-existent player-base for custom-made MMORPG research scenarios A.I. agents, impersonating human players, can be used to "bootstrap" the research scenario to reach the necessary massive number of players that define the game genre.

This thesis presents a system that makes its human players and A.I. players indistinguishable while preserving the basic characteristics of a typical MMORPG. To better achieve identical perception of human and A.I. players, our system centers around the collection, sharing, and exchange of information while limiting the means of expression and actions of players. A gameplay scenario built on the Panoptyk engine was constructed to imitate gameplay experienced in major MMORPGs. We conducted a user-study where subjects play through the scenario with a varying number of A.I. players unknown to them. Three versions of the scenario were created to assess how indistinguishable human and A.I. players were and vice versa.

We found, across 24 participants, there were 32% correct identifications, 30% incorrect identifications, and 38% answers of "I don't know". This was broken down into 20% correct identifications, 42% incorrect identifications, and 38% answers of “I don’t know” for bot characters and 46% correct identifications, 16% incorrect identifications, and 38% answers of “I don’t know” for human characters.

ACKNOWLEDGMENTS

Thanks to:

- Jennifer Cheung, Matthew Miller, Becca Miller, for their love and support
- Megan Washburn, for her encouragement and care
- IATTP: Institute For Advanced Technology and Public Policy, for introducing us to the broader implications of this project
- Dr. Foaad Khosmood, for his immense support and guidance as thesis advisor
- Andrew Guenther, for uploading this template
- Huey, Louie, Dewey, and Chewy, for being the best chickens, your lives were too short

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Massively Multiplayer Online Role Playing Games (MMORPGs) are a prominent genre in today's video game industry with the most popular MMORPGs generating billions of dollars in revenue and attracting millions of players [2]. Although games in the genre are diverse, games 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 accomplish [3]. As they have grown, they have become a major target for academic research.

Research is conducted on the underlying technologies used to build MMORPGs. Its massive number of concurrent players, ranging from a thousand to millions of players, creates large scale challenges for its underlying technology. Many systems are proposed and built upon in an effort to build the most efficient architecture to house the MMORPG's virtual world and multitude of players, ensuring a seamless experience. Moving away from the technical aspects in running an MMORPG, research also focuses upon the gameplay and retention of players. A player's captivation of the world they are in and the social interactions that come with it are often the highlight of the game genre. Also included in this, is players' interaction with Non-Playable Characters (NPCs). More than just improving the game genre itself, MMORPGs also offer massive social dynamics to analyze that can provide insights into human social behavior in the real world. This is because an MMORPG world can be a good approximation of the real world. This has led researchers to look at, not only the social dynamics, but the economics in these virtual worlds.

Apart from research in technological systems, most research is done using data gathered from large well established MMORPGs. This is because for an MMORPG to become majorly popular, a large investment of time and money is required. Additionally, the game must be maintained throughout the years. The game's servers, promotion, and a new stream of content all add to the continuous cost of an MMORPG. In order to profit from an MMORPG, many players must be attracted and retained as recurrent players. This can make it all but impossible for the small independent creation of MMORPGs, and greatly limits the amount of control and topics that can be researched. Custom made scenarios exploring new game mechanics or specific socioeconomic phenomena will have a very difficult time emulating the scale of players in MMORPGs.

In an effort to make up a lacking or non-existent player-base for custom-made MMORPG research scenarios we propose a system where A.I. agents, impersonating human players, can be used to "bootstrap" the scenario with the necessary massive number of players that defines this game genre. In this paper, we explore the efficacy of using a mass amount of robot players to give the feeling of a thriving MMORPG environment. In order to achieve this level of believably human A.I. several limitations are placed in the design of player interactions to give our system a feasible chance. The contributions of this paper consists of work on an MMORPG engine, titled the Panoptyk Engine, a game scenario that emulates the basic mechanics found in an MMORPG, A.I. bots to pose as human players, and a user-study made to assess how distinguishable human and bot players are within the system.

Moving forward, this papers will discuss essential background knowledge on MMORPGs, Typescript, Node.js, and other libraries used in the implementation. From there, other similar work in this field is discussed. The system is then discussed, separated into a chapter on the Panoptyk Engine, and a chapter on the specific design and implementation of the game scenario and bots. Afterwards, the design of our experiment and its results are written. Finally, conclusions on the result data collected and future work is suggested.

Chapter 2

BACKGROUND

2.1 MMORPGs: Massive Multiplayer Online Role Playing Games

A subset of Massive Multiplayer Online games (MMOGs), MMORPGs define a genre of games that support a large concurrent player base usually in the thousands or more, with a persistent world, many activities, and social interaction. Their history dates back to multi-user dungeons (MUDs) which were text-based adventure games modified so several players could play at one time. Because of this, MUDs are often thought of as the starting point of MMORPGs. In the modern gaming industry, MMORPGs such as World of Warcraft and EVE online have risen to great popularity, garnering massive player bases and financial success [10].

2.1.1 NPCs: Non-Player Character

Often used in MMORPGs, NPCs are all the characters within the game not operated by a human player. They are normally thought of as simple characters acting out a single role. For example a merchant NPC stands idly by, opening a shop for players when prompted. In this work, NPCs are characters within a game that are never played by humans in contrast to characters that may be played by humans or are a coded bot.

2.2 TypeScript

TypeScript is a super-set of the language, JavaScript, that adds typing. It assists in maintaining larger scale code bases, allowing type inference and enforcement. TypeScript compiles to pure JavaScript for easy deployment to the web. Many Node.js packages are also created in TypeScript for easy compatibility [12]. TypeScript was chosen for this project to enable faster development, and the easy sharing of mutual code between different sections of the system.

2.3 Node.js

Node.js is a runtime environment for JavaScript for any operating system. It allows non-graphical or web-based JavaScript to be developed and run. It provides an asynchronous event-driven platform made to support scalable network applications. In addition to baseline utilities for server-client communication, Node.js includes a package manager that can import many useful packages including the ones described below [14]. We used Node.js to run our backend server and all additional A.I. bots in our system.

2.4 Socket.io

Socket.io is a node package that allows real-time, bi-directional communication between server and client. The communication also follows an asynchronous event driven pattern [23]. This real-time connection was necessary to allow smooth communication and gameplay in our system.

2.5 Express

Express was another node package that provided a minimalist web framework built on top of Node.js. It was also a dependency used by Socket.io [4]. Express made it much easier to specify the necessary files to host, providing the website files to participants' internet browsers.

2.6 Phaser

Phaser is an open source html5 game framework. It builds off of Canvas and OpenGL, providing many features for game developers found in other game engines. Phaser makes it easy to load art assets, render animations, provide a main game loop to put logic into, calculate physics, and much more [16]. Without Phaser, the web-based client portion of our system would have taken much longer to develop.

2.7 Vue.js

Vue.js is a progressive framework for building user interfaces on the web. It provides templates to easily create dynamic dropdown menus, lists, buttons, and other html elements seen on a webpage. Vue.js manages these elements and can update them based on changes triggered by JavaScript code [25]. We used this package to provide the necessary U.I. elements to allow participants to interface with our game scenario.

Chapter 3

RELATED WORK

This chapter discusses the related research within the field.

3.1 NPC Believability

Non-player characters (NPCs) have always been said to lack a very believable intelligence. While this work looks at how believable bots written to pose as human players are, it is a very small gap between them and believable NPCs.

One paper, on creating more intelligent NPCs, a text-based style adventure game, titled MKULTRA, equipped NPCs with natural language processing to communicate with the player. However, the seemingly large knowledge of NPCs backfired and the player would try to engage in many actions/conversations that the NPC could not handle [6, 7]. This draws many similarities with our proposed system, although in place of natural language processing, an information system is created as a mechanic for bots and humans to use to communicate and interact with one another. This allowed our system to control and eliminate any unwanted actions/conversation topics that bots would not interpret.

Even an NPC's movement behavior can be enhanced in the hopes of increasing the player's experiences [11]. Here NPCs were built to display very complex movement behavior in the hopes to induce a better gameplay experience. However no validation was ever made, as the system was not incorporated into any actual playable game scenario.

3.2 Massively Multiplayer Online Game (MMOG) Systems

Many proposed MMOG systems in research are primarily focused on overcome technical challenges. Some systems proposed, aim to combat network bandwidth and sacalability issues in servers [5]. Others propose to scale A.I. capabilities in an effort to support more NPCs [26].

3.2.1 Similarly Proposed Systems

ModelMMORPG and "The Mana World" proposes a system similar to the one here, as its effort is to provide a foundation to create MMORPGs that bots can interface [19]. However the proposed solution tries to work around an MMORPG system that is not altered to assist in bot design. Bots interfacing with their system require in-detail knowledge of the specific scenario's quests [20]. In contrast, our system centers its game mechanic on something that can also assist in the creation of bots for it.

3.2.2 Commercial Systems

Several commercial systems have been created that assist developers in creating an MMORPG, such as SpacialOS, which provides a scalable system to enhance servers [9]. Another proposed system is NeuralMMO by OpenAI that would support only AI agents in an MMOG setting to study multi-agent systems [15]. While inspired by MMOGs, the system does not support human players as it is more tangential to the research in MMORPGs.

Chapter 4

PANOPTYK ENGINE

The Panoptyk Engine(aka Panoptyk) is an on-going initiative within the Institute for Advanced Technology and Public Policy[1]. Originally created by Nathan Philliber, the engine was majorly overhauled to satisfy new requirements for this work and fellow master's student and contributor Sean Mendonca's thesis. Changes included a refactor to Typescript and packaged the engine for easy import into each thesis' custom scenario. In addition, both Kaito Trias and John Potz have made contributions to Panoptyk. Panoptyk's design and implementation is covered in this chapter as it served as the foundation for the gameplay scenario in this work.

Panoptyk is an MMORPG engine that facilitates gameplay focused around creating, sharing, and exchanging information. This information is constantly generated by players' actions ensuring an endless stream of new information. This engine is a work in progress, intended to serve as a platform for simulating human/robot interaction and automatic generation of game assets, quests, and real-estate by using its information-based mechanic [13]. Panoptyk is titled an engine because it outlines game mechanics but is not a game itself. It is intended to be imported and used to create a specific game as this work does.

4.1 System Overview

As seen in Figure 4.1, Panoptyk provides a foundation for MMORPGs by laying out a template to represent the game world and its contents, client-server communication,

[1] `https://iatpp.calpoly.edu/research-areas`

and an API for connected clients to interact with the game. The engine is written in Typescript and powered by Node.js for development and package management. This choice was made to support easy asynchronous communication and the high accessibility of a web app. It is comprised of multiple components. The model component houses all classes that make up the game state data and unifies the storage and transmission of the data. A rudimentary in-memory database is used to store all game world artifacts at run-time providing a way to save and load the models via json files. On top of this, a controller component provides a list of functions to modify models in response to actions made by connected players. The controller keeps track of all modified models and a list of recipients to send them to. Lastly, a communication component uses socket.io to ingest actions connected clients request to make. These requests get validated then enacted using the controller component. Success or failure responses and model updates are also sent to clients. A client API is also provided for users of Panoptyk creating an abstracted interface to the corresponding server. The following subsections further detail the division of server and client components in Panoptyk and server-client communication.

4.1.1 Server

The server portion of the Panoptyk Engine maintains the most up-to-date version of the game state and receives communication from all game clients. In contrast to most multiplayer games, the server does not regularly send updates of the game state to all clients. Requests from game clients are handled asynchronously, processing actions made by players and updating only the necessary portions of the client's game state as-needed. Most transformations of game state are started by a connected client's communicated action to the server. In this way, driving the game world is a distributed task, as every NPC, bot player, or human player is connected from a

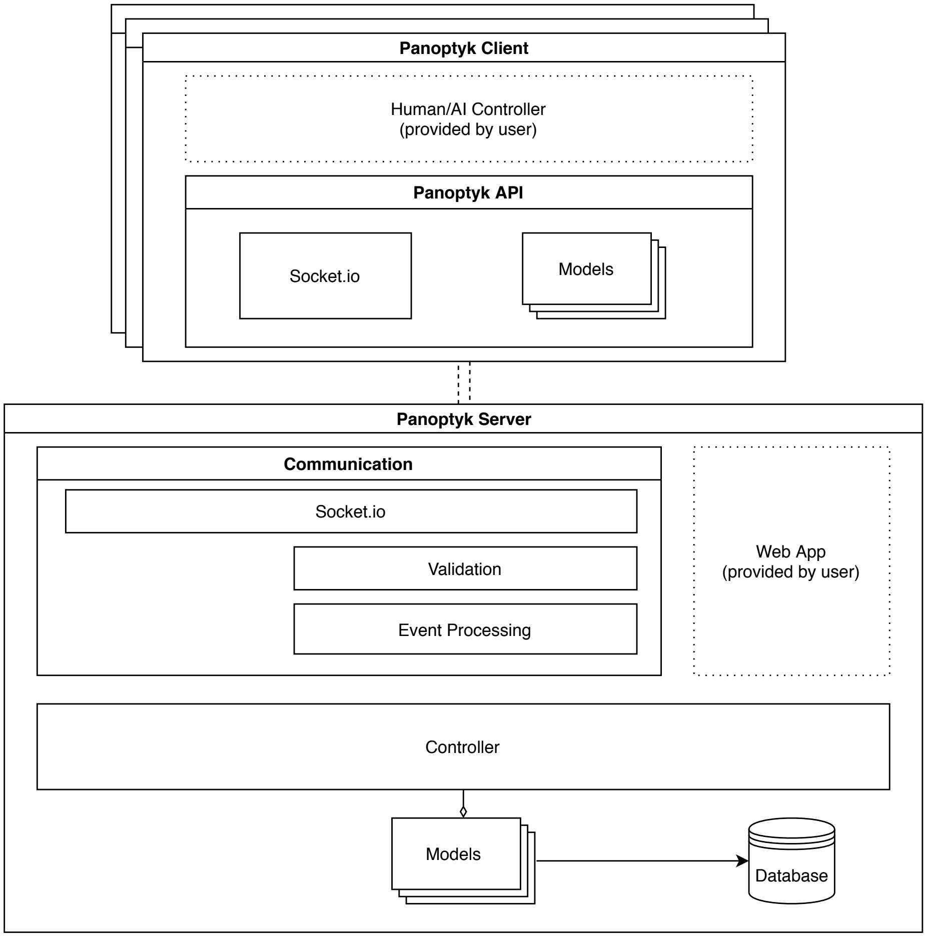


Figure 4.1: Panoptyk overall architecture

different program. Figure 4.1 illustrates the different paths of communication clients can have with the server. After the socket.io connection is established, a client must log-in as a specific character before any other actions can be taken. On a successful log-in, the server will send a message with all relevant game state data. Once a client is logged-in, they can begin to perform other actions, see Table 4.1 for examples. The server can either send a confirmation the action was performed, model data has changed, or the action was denied. A denied action message occurs if the client's character does not meet all conditions to perform the action or if an inadequate amount of time has passed since the client's previous action. If the action is validated and performed, a confirmation will be sent to the client. The server then sends clients affected by the action a personalized game state update. Alongside this

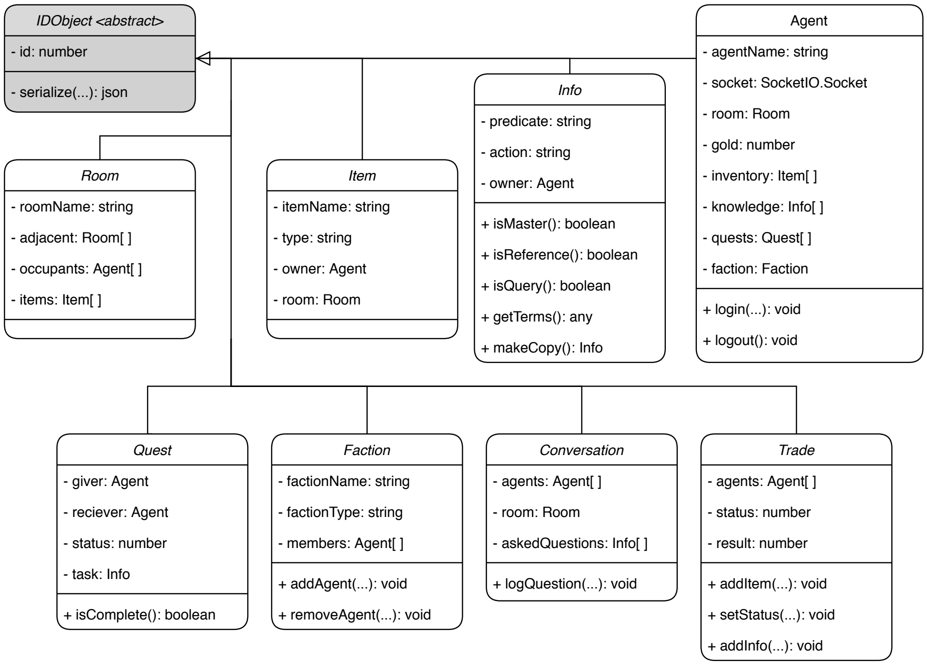


Figure 4.2: Class diagram of Panoptyk models

communication, a user can pass a customized Express app. For example this allows the hosted web client to run from the Panoptyk server to share an IP address and port.

4.1.2 Client API

The Client API abstracts the direct socket.io communication to the server wrapping it in easier to use functions. These functions are all JavaScript promises to easily allow asynchronous programming. The Client API exists as a singleton to be called upon anywhere within the specific client application being developed. Once initialized, the socket.io connection is established and a listener is added to process game state updates received from the server. If used, the client API promises that the in-memory models of the game state are up-to-date and synced with the server's version. This is intended to eliminate worry that game state data is outdated. Flags are also put in place to notify if a new server update is currently being processed. When packaging

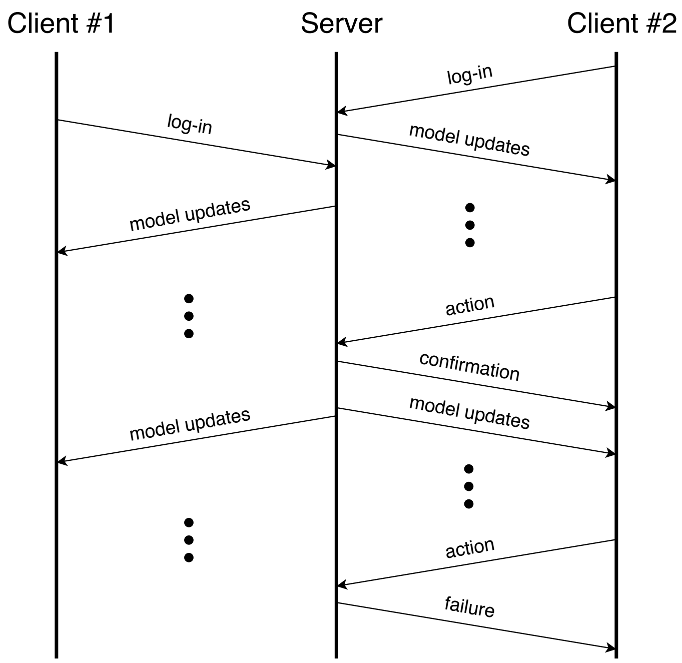


Figure 4.3: Panoptyk server communication

Panoptyk's Client API, all model class files will be included as they are necessary to interpret the game state. No web interface is included in the Client API so that it can be used for both a human web interface and a bot interface. The specific human web interface and bot interface design are discussed in the next chapter.

4.2 Model Representation

A set of model classes, all inheriting from the same base class, are used to define all the artifacts within the world of a Panoptyk game. The list of models created to facilitate basic MMORPG actions are shown in Figure 4.1. This includes models for items, information, characters themselves(agents), locations within the world(rooms), quests, conversations and trades between characters, and guilds(faction). Each model is given a unique ID number which is used in place of the model object itself when being referenced. Models can then be serialized into json with relative ease. For example, the inventory of a character will be an array of primitive numbers instead

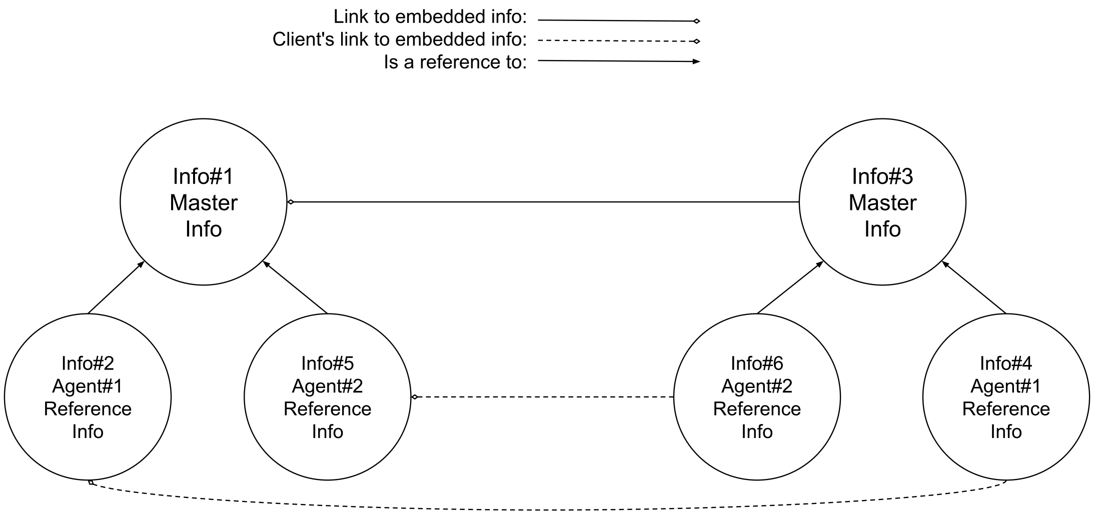
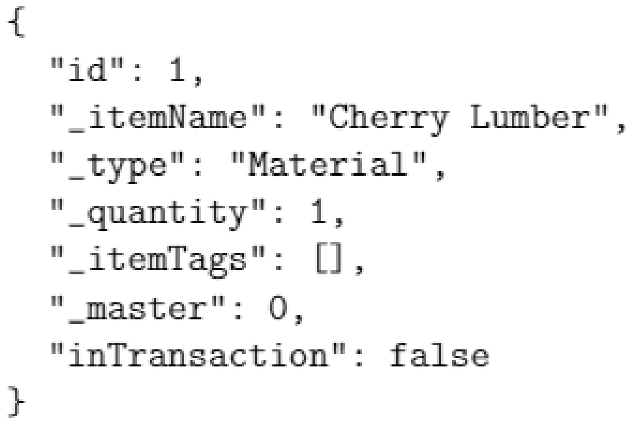


Figure 4.4: Information reference graph

of an array of item model objects, see Figure 4.2. Each model implements a serialize function that produces a json safe object and its counter-part reload from a json safe object. The models' publicly return other model objects they reference, but internally only the unique ID number is saved. The id numbers are created in a way that it is an O(1) operation to look up the in-memory object instance of a model. By maintaining this representation, models can easily be saved to a file and reloaded, but more importantly they can be distributed to connected clients.



```
{
  "id": 1,
  "_itemName": "Cherry Lumber",
  "_type": "Material",
  "_quantity": 1,
  "_itemTags": [],
  "_master": 0,
  "inTransaction": false
}
```

Figure 4.5: Example of json created from a Panoptyk item model

4.3 Information in Panoptyk

All interactions in Panoptyk result in information being exchanged. Whenever a character moves locations or picks up an item, for example, a unique piece of information is created, recording the action. This record is given to the actor themselves and other characters able to observe the action. Information is intended to be used as a commodity alongside tangible items and an in-game currency. It should be a central system that influences player's actions. As stated previously, the game created with this engine will be played by humans and programmed bots alike. This information system serves to assist bots' decision making by giving them a detailed description of the events occurring around them. Information even takes the form of predicates from temporal first order logic to strengthen the its connection to logical inference, see Table 4.1.

Table 4.1: List of actions that generate information

Action	*Predicate Variables*
Move	(Time, Agent, Location, Location)
Pickup	(Time, Agent, Item, Location, Quantity)
Drop	(Time, Agent, Item, Location, Quantity)
Converse	(Time, Agent, Agent, Location)
Give	(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)

4.3.1 Representation

Information is a model like the other artifacts in a Panoptyk game. It contains a means to extract the terms of the information depending on the specific action. These terms outline the subject of an action, anything else that pertains to the action, and the time the action occurred. For example, when a character moves location, the character, their previous location, target destination and the time are recorded in the piece of information generated. Along with this, the owner of a piece of information is recorded in its model. Because many characters will potentially witness the same event, information is separated into master and reference copies. A master copy of information object contains data for the action and all its terms, where as a reference copy of the object only points to the master object. This way, multiple reference copies with unique IDs can be created to keep track of ownership amongst multiple observers of the same event while minimizing duplicate data. Figure 4.1 illustrates how reference copies link to a specific master copy of information. The figure also shows how information which is itself an action term is pointed to. This could occur when one character tells another of an event. The record of such an event would have to include the other piece of information that was told. As seen, the new information references the master copy of the information told. But, when distributed to client of the corresponding owner, the reference is rearranged for the client so that the reference copy of the "tell" information points to the reference copy of the information told. The unique id number of information is visible to players, so this rearrangement assists in better cohesion when viewed in their list of information.

Chapter 5

SYSTEM DESIGN

This chapter outlines the design and implementation of the MMORPG gameplay scenario, human web interface, and A.I. bots created to illustrate bootstrapping. The game scenario is built using the Panoptyk Engine as its foundation.

5.1 Scenario

In order to test bootstrapping, a gameplay scenario was created. The scenario included a virtual world, unique characters for players, two guilds, a means of communication and trade between players, and a quest system linked to leveling up. All of these game features were meant to emulate the basic characteristics of an MMORPG. The features were chosen to provide players with a meaningful environment, a captivating goal, and many opportunities for players to interact with each other.

5.1.1 Playable Characters

Eight unique characters could be played in this scenario. The names of the characters were Alison, Eldric, Florence, Holden, Knox, Paige, Tuesday, and Wilfred. Along with dissimilar names, unique sprites were given to each character. All of this was done to ensure the characters were not mistaken for one another. This was done to ensure when a player was asked to distinguish whether a character was played by a bot or human, the character in question wouldn't be mistaken for another.

5.1.2 Game World

The game was set in high-fantasy, a common setting for most MMORPGs. Set inside a medieval town, players could move between different large areas and enter certain buildings including two guild halls, a cafe, a multi-room tavern, an inn, and a random house. Figure 5.1.2 displays all potential locations and how they are connected with one another. Players originally spawn in one of two locations according to their character. Afterwards, they could freely traverse between any location, encountering other players and obtainable items. Within a location, players are not given free movement control. The location their character and other characters occupied in the human client was set randomly and could be different on all human web clients. This restriction was placed to increase the chance bots have to pass as human players as it eliminates the need for pathing at a micro scale. Only simple graph traversal on a very small graph is needed to determine the path to the target location for a bot.

Items were also named in line with the high-fantasy setting. Throughout the game, items were replenished periodically in each location. Max number of items in a location and its rate of replenishment were set to create different levels of scarcity, effectively creating rarer to obtain items. Table 5.1.2 enumerates all items, their location, maximum number that can be picked up, and how many seconds before another is potentially spawned. Several royalty-free asset packs were used to visually create the town and specific characters for the human web client. "PIPOYA FREE RPG TILESET 32x32 Pixel" and "PIPOYA FREE RPG Character Sprites 32x32" were used for the game locations and game characters [18, 17]. "Shikashi's Fantasy Icons Pack" was used for the game icons [22]. These steps were taken to help immerse human players in the virtual world to increase their concentration when playing the game scenario. The final created assets of each location can be found in the appendix.

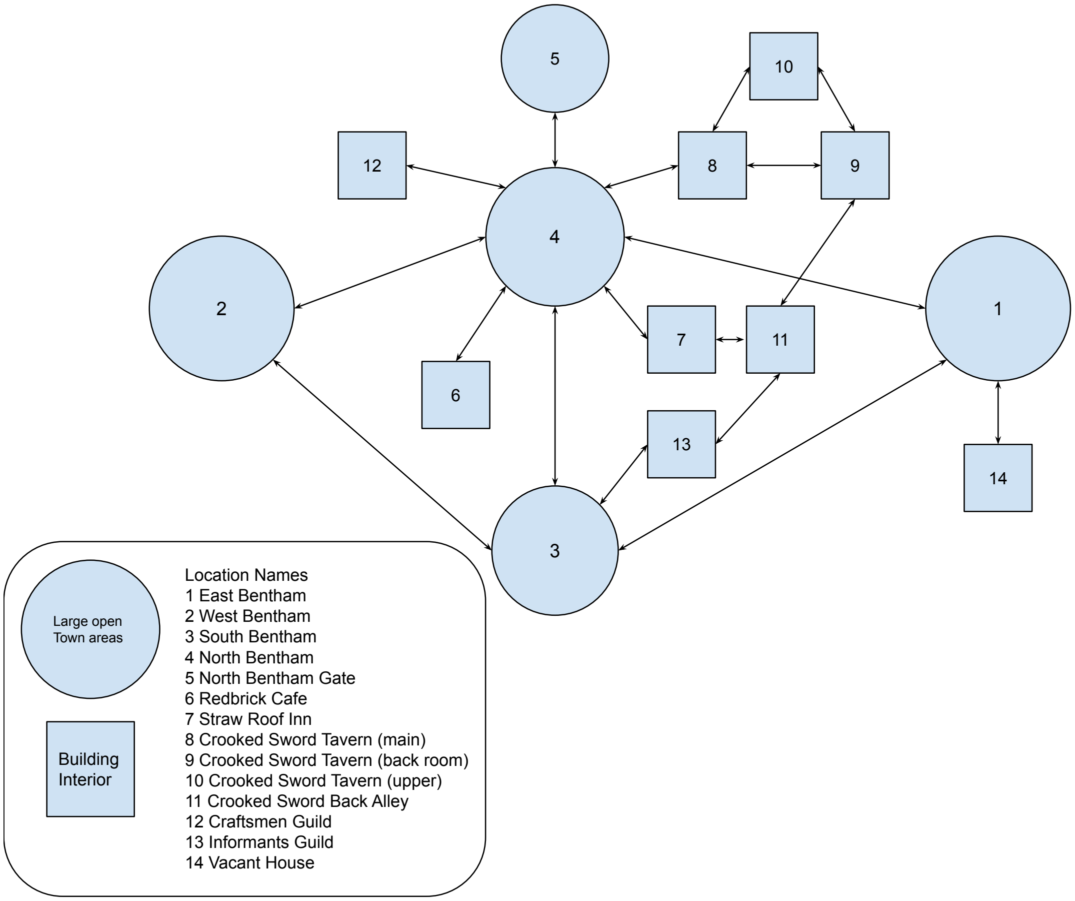


Figure 5.1: Graph representing all locations and connections between locations

5.1.3 Quests and Guilds

The eight playable characters are split into two guilds and tasked with completing quests for their guild. One guild is titled "Informants" and the other "Craftsmen". The guild a character belongs to determines the type of quests they receive. Informants are given quests that require information to complete, while Craftsmen are given quests that require items to complete. These quests are in the simple form of fetch-quests where the player must acquire information or an item and turn it in to their corresponding guild leader NPC. Similar to many MMORPGs, the guild leaders are located in specific guild hall buildings, giving, receiving, and completing.

Table 5.1: Shows all items, their location, replenish rate, and maximum

Item Name	*Location*	*Max #*	*Replenish Rate (one every)*
Birch Lumber	North Bentham Gate	2	22s
Small Stone	Crooked Sword Back Alley	2	17s
Iron Shield	Crooked Sword Tavern (upper)	2	22s
Broken Barrel	Crooked Sword Tavern (back room)	2	17s
Cherry Lumber	North Bentham Gate	1	67s
Distilled Lubricant	Vacant House	1	67s
Mythril Chef's Knife	Redbrick Cafe	1	67s
Lump of Silver	Straw Roof Inn	1	67s

When a player completes a quest, they receive experience points. This creates the basic gameplay cycle for the scenario, receive a quest, fetch the item or information in question, get rewarded experience. As will be discussed in the experiment design chapter, human participants will be tasked with trying to level their character up by completing quests. The two guilds and differing quests aim to introduce more interaction between players. Any player can acquire both information and items, but, by making only one type required for each player, a supply and demand is established. This was designed as an incentive for players to deeply interact by conversing and trading.

Two different sets of quests were created to draw from depending on the player character's current level. The first set was easier while the second set was made more difficult to incite trading between players to complete. The list of types of quests are shown below. The specific agent, item, or quantity are randomly picked from set pools to create the quest.

- Craftsmen easy quests:

  Fetch [1|2] of [*easyitems*].

- Craftsmen hard quests:

  Fetch [1|2] of [*harditems*].

- Informant easy quests:

  Give [1|2] pieces of info about [*anycharacter*].

- Informant hard quests:

  Give [1|2] pieces of info about “What item [*anycharacter*] gave to whom?”

  Give [1|2] pieces of info about “Who told [*anycharacter*] something?”

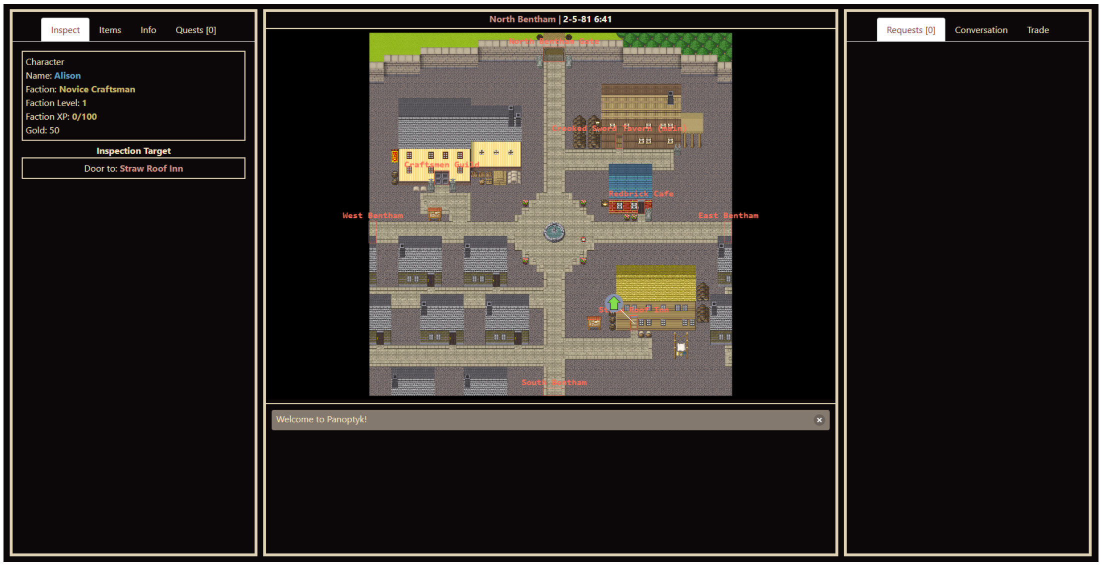
$$[anycharacter] = [Alison|Eldric|Florence|Holden|Knox|Paige|Tuesday|Wilfred]$$

$$[easyitems] = [BirchLumber|SmallStone|IronShield|BrokenBarrel]$$

$$[harditems] = [CherryLumber|DistilledLubricant|MythrilChef'sKnife|LumpofSilver]$$

5.1.4 Player Communication

In this scenario, and Panoptyk, free text communication is not possible. In place of this communication, tied heavily into the information system of Panoptyk, are facets to converse with another player and/or trade with another player. Currently conversations are limited to, telling the other player information or asking questions which can be seen as very minimal amount of communication between players. The amount of actions represented in information drives the depth of questions, defining what can be asked about. To enhance this communication and also facilitate the transfer of items and info between players, trading can be entered from an on-going conversation amongst players. Within a trade, players can offer and request gold, items, and answers to questions. If both players indicate the trade is agreeable the gold, items, and info are transferred to the correct player. The primary reason behind having limited communication, goes back to the bots made to pose as human players. Without a structured form of communication, any human player entering text chat with a bot would immediately discover its identity. In this way, the structured communication based off Panoptyk's information system equips bots to better navigate communication and ensures there is still a deeper interaction between players besides being in each other's presence. To facilitate the communication in the human client, a heavy amount of U.I. was created.



Character
Name: Alison
Faction: Novice Craftsman
Faction Level: 1
Faction XP: 0/100
Gold: 50

Inspection Target
Door to: Straw Roof Inn

Welcome to Panoptyk!

Figure 5.2: Screenshot of the full human web client

5.1.5 Human User Interface

While human players had a main window graphically displaying the game world and allowing the player to interact with what was displayed, multiple additional U.I. elements surrounded the window to give the player details, and facilitate the conversations and trades between players. As there were few corollaries to draw from about representing the information system used here, the U.I. design of it proved difficult. In total there are three components surrounding the main game window, a left set of tabs, a right set of tabs, and a log of events below. The text log below was built to additionally inform players of events occurring around them. This could be other characters moving to and from their location, notifications of requests to

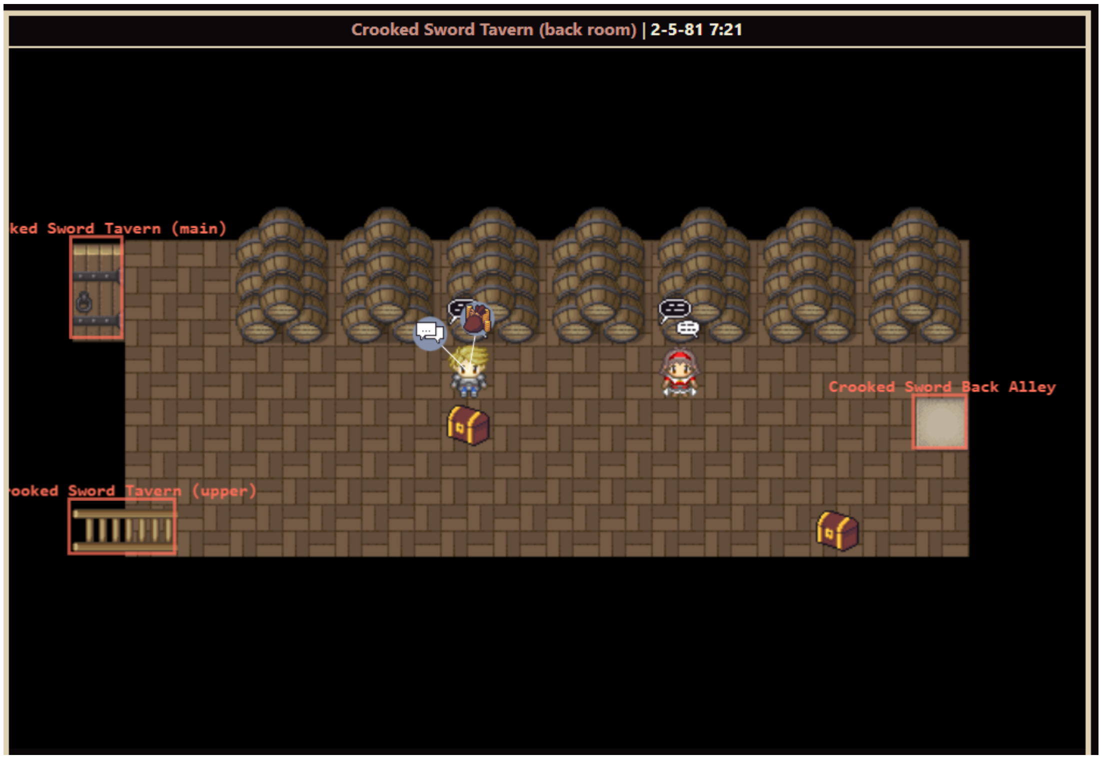

Crooked Sword Tavern (back room) | 2-5-81 7:21
ked Sword Tavern (main)
Crooked Sword Back Alley
ooked Sword Tavern (upper)

Figure 5.3: Two characters in conversation in an interior location

converse or trade, and denials of requests to converse or trade. Screenshots of all U.I. elements can be found in the appendix.

On the left side, a set of tabs provided information about the player's character and things they may interact with in the main window. The first tab sharing details of anything clicked in the main window, and details about the player themselves such as their faction, gold, and level. The next two tabs represent the player's inventory. This was a tab of all items collected by the player and a tab of all information acquired by the player. The amount of information passively observed and acquired by a player could become very large quickly, so filters were added to help assist the player when they searched through it. As information is represented just by its action and meaningful terms regarding that action, a converter was made to generate meaningful sentences from the information model. The last tab on the left side enumerated the quests received by the player. This was where a player looked to find what item or information they were being asked to collect for their respective guild.

On the right side, a set of tabs facilitated all communication between players. The first tab informed the player about any requests to converse or trade, allowing them to accept or decline. The second tab was dedicated to conversing with another player. Here through a set of drop downs, players could create questions to ask, tell the other player a select piece of info, and also turn in quests to their guild leader. Sentences were also generated here for the questions created and to preview information to be told. The last tab on the right side, was dedicated to trading. This allowed the player to select gold, items, and information to be requested or offered. When a player was not in a conversation or trade, the corresponding tab would also be empty and inform them of the fact.

5.2 Implementation

To construct this scenario four separate pieces needed to be implemented. These were the game server, human web client, guild leader NPC bots, and questing bots attempting to pass as players. All pieces were implemented using TypeScript in a Node.js environment. This allowed a large amount of code to be shared. For example, the Panoptyk Engine was imported as a node package and the relevant components used for each piece of the scenario. As stated previously, Socket.io was used for all communication between server and clients.

5.2.1 Server

For the server, most of the implementation remains the same as described in the Panoptyk Engine chapter. Additionally, minor changes were made about the representation quests and items in Panoptyk to allow the types of quests required from the scenario. Most importantly, json files of room models, item models, agent models, and faction model (guilds) were manually defined to match the game world defined by the scenario. These are automatically loaded on the start of the server. Finally, a few recurring asynchronous functions were written. One handled item replenishment while the server ran. The other watched the login of characters to identify when all eight unique characters logged-in. Only then, did item replenishment start and questing bots begin to act. This measure was put in place so that it would seem as though all characters started at them same time, maintaining the illusion that they all were human when participants entered a game that included bot players.

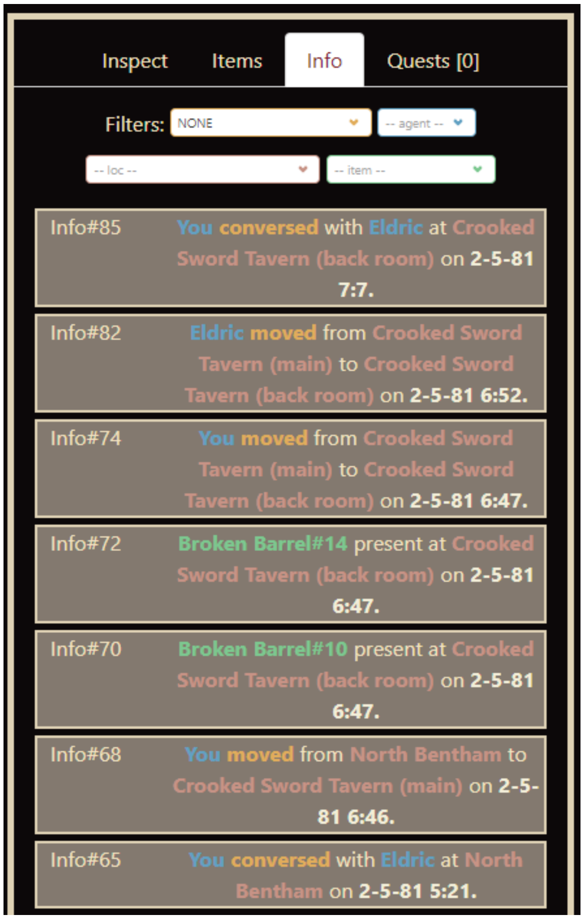
A simple Express app was also written and injected into the server. This outlined the necessary files needed to be hosted so that a participant could load the web based client including the index.html file, bundled js files, and all visual assets for the game.

5.2.2 Web Client

The web client was created, starting with Panoptyk's Client API class. As this API only covered communication with the server, the entire visual interface needed to be created. The html game engine, Phaser, was used to render the game locations, characters, and items in the main window of the web client. All peripheral U.I. were built with Vue.js, a progressive U.I. framework.

Starting with the main window, here graphics needed to be rendered based on the player's location and other characters present. All game locations were built as tilemaps in Tiled [24]. Phaser can load the tilemaps and display the resulting image. Within Phasers scenes are defined, each contains, boot function and a periodic update loop. Several boilerplate scenes were used to load all assets, then provide a login-screen for the game. Afterwards, the primary game scene was entered. This set up many image groups to define what images rendered above others. Many functions were created to perform tasks such as loading a new location's map, animating all characters leaving or entering the location, displaying action options where the player clicked, and drawing the boxes that allowed the player to move to adjacent locations. The game scene's update loop would look for changes in the game state and call upon any of these necessary actions. Additionally this loop was where hooks were called that bridged the main window's Phaser game code with Vue.js components running the peripheral U.I. Many game state changes would be put in an event queue that processed all graphical animations to ensure animations completed correctly. For example, if a character enters the player's location then leaves quickly. The event queue would ensure the player saw the character completely enter from the corresponding entrance moving to and idle position before seeing the character exit the location in the same fashion. This did mean human players may lag behind in knowing their current game state due to the time taken to visually convey the information. The lag time could only be several seconds, and it was decided the visual animations provided more to players' experiences over the immediacy of game state information.

Vue.js components were created for every peripheral U.I. window as described previously. Visually these were defined using preset and styled html components provided by Buefy which is included with Vue.js. Mostly wiring was done to hook up the elements to react to changes in game state, changes induced by the player interacting with the U.I., and changes triggered from the Phaser game main window. This allowed elements such as a constantly changing in-game clock time. All the U.I. layouts can be found in the appendix. One important task done in these components was the translation of the Panoptyk information model to human readable sentences. The sentences were not just generated, but action, subject, and other terms were systematically color coded to assist players in recognizing similar information. Figure 5.2.2 shows some of these generated sentences. As can be seen, actions are colored yellow, locations red, agents blue, items green, and time white. The specific colors used were picked to match the overall color pallet used for the web client, to ensure easy readability.



Info#85 You conversed with Eldric at Crooked Sword Tavern (back room) on 2-5-81 7:7.

Info#82 Eldric moved from Crooked Sword Tavern (main) to Crooked Sword Tavern (back room) on 2-5-81 6:52.

Info#74 You moved from Crooked Sword Tavern (main) to Crooked Sword Tavern (back room) on 2-5-81 6:47.

Info#72 Broken Barrel#14 present at Crooked Sword Tavern (back room) on 2-5-81 6:47.

Info#70 Broken Barrel#10 present at Crooked Sword Tavern (back room) on 2-5-81 6:47.

Info#68 You moved from North Bentham to Crooked Sword Tavern (main) on 2-5-81 6:46.

Info#65 You conversed with Eldric at North Bentham on 2-5-81 5:21.

Figure 5.4: Screenshot of the tab that showed all information player knew

5.2.3 Guild Leader NPCs

The guild leader NPCs were built off the same design that will be mentioned below for the questing bots, however their main action loop was replaced with a very simple set of steps seen in algorithm 1. They were designed simply to facilitate quest giving and quest completion to the server. Each guild leader is run as a separate process which connects to the server as a client. They log in as their specific character and stand idly waiting to converse with other characters to give out quests. These bots are no different then any other human or bot player to the server and could have been made to do more complicated tasks if necessary. But, generating quests was the only thing needed for these bots.

```
if given quest can be completed then
    send quest complete to server;
if conversation requested and not in conversation then
    enter the conversation;
    assign quests to the maximum allowed;
```

Algorithm 1: Guild leader NPC main action loop

To generate quests, the bot simply picked pre-constructed ones from several lists. There were two lists for each guild, craftsmen or informant, representing the easy or hard quests. Not until the player was level three did the bot start to assign quests from the more difficult list. Quests were picked randomly from each list, being removed after. Only when a list was empty would it be refilled with all versions its quests again. This ensured that all versions of the quests were assigned before duplicates in order to help diversify the items and information required to be collected.

5.2.4 Questing Bots

Even with the limitations put up by the scenario design, for the bots to be able to emulate human players, they would need to be able to handle uncertainty and be fault tolerant. To do this, bots were driven by a Hierarchical Finite State Machines (FSMs) [27]. The higher level of states are referred to as the behavior states, and the lower level referred to as action states. Behavior states encapsulated longer term activities the bot would be making, while action states primarily acted as simple wrappers for server communication. This way within a behavior state, the proper state machine of server actions could be constructed which would also handle failure if any action failed to be performed or was denied by the server. An FSM comprised of behavior states was referred to as a strategy. Bots were also programmed to never crash, suppressing all run-time errors, so that they would not disconnect from the server even if remaining idle.

Bots were designed to act as though it was also their first time playing the game. This meant that, representations of the game world map, and locations of the guild leaders were not pre-programmed in. They would be discovered via exploring just like a human player would do. Taking all this into account, the quester bot was defined with two strategies, one for completing quests and one for conversing/trading with a character. They would be swapped between depending on the immediate circumstances of the bot. Primarily if a bot came across any character they would like to speak to or were requested to converse, they may enter the discussion strategy. Several other processes are concurrently handled as well. If an item is in the location, the bot may pick it up. Additionally, every execution of the bot's main loop, a knowledgebase is updated. This will build a graph of all locations, how they are connected as well as remember item locations, and the last location of characters passed. This knowledgebase is accessed throughout the bot's code to help make logical inferences. Algorithm 2 outlines the main loop of the quest bot. Figure 5.2.4 and Figure 5.2.4 show the two strategy FSMs.

```
update knowledgebase;
if item present in location and want to pick up item then
    pick up item;
if in discussion strategy then
    use discussion strategy FSM;
else if character of interest present or conversation requested then
    initialize new discussion strategy FSM;
    use discussion strategy FSM;
else
    use quest strategy FSM;
```

Algorithm 2: Quester bot's main action loop

In the discussion strategy, Figure 5.2.4, the bot switches between conversation actions and trade actions. In a conversation the bot seeks to ask questions and potentially answer any asked of it. In a trade, the bot tries to request what is required by its quest. It tries to ensure the trade is relatively fair by counting the number of offers on each side. It will try to offer anything requested as long as it doesn't conflict with what is needed for its own quest. Any failure or simply a timeout will set the strategy as complete. This discussion strategy FSM is recreated each time a new conversation is entered, unlike the quest strategy FSM.

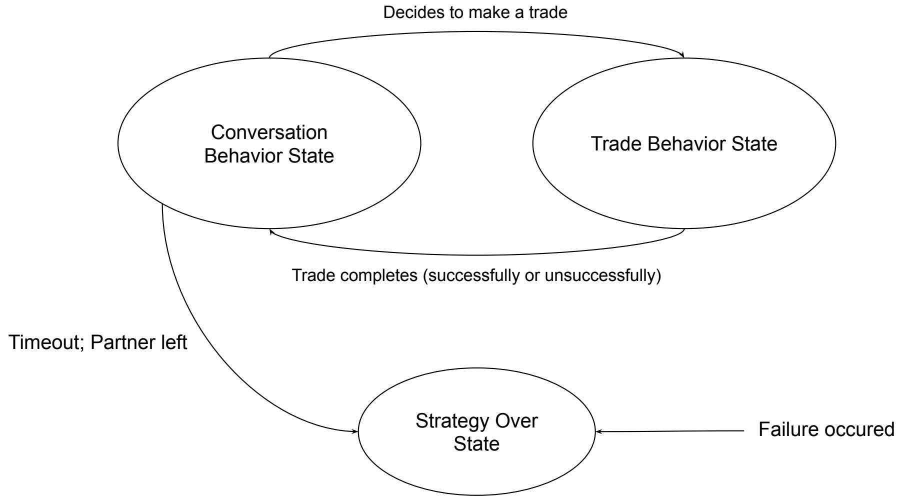


Figure 5.5: Discussion Strategy: Behavior state FSM for conversing and trading

In the quest strategy, figure 5.2.4, the bot continuously moves through locations, occasionally returning to its guild leader to complete quests and receive new ones. Its target location can be influenced by if it has completed quests or if it has a character or item to try and find. Apart from that, the bot moves locations randomly. If it has not visited all locations of the entire game world, it also seeks to do so. This strategy mainly drives the movement of the bot, relying on other checks to inform it to interact with other characters or pick up an item.

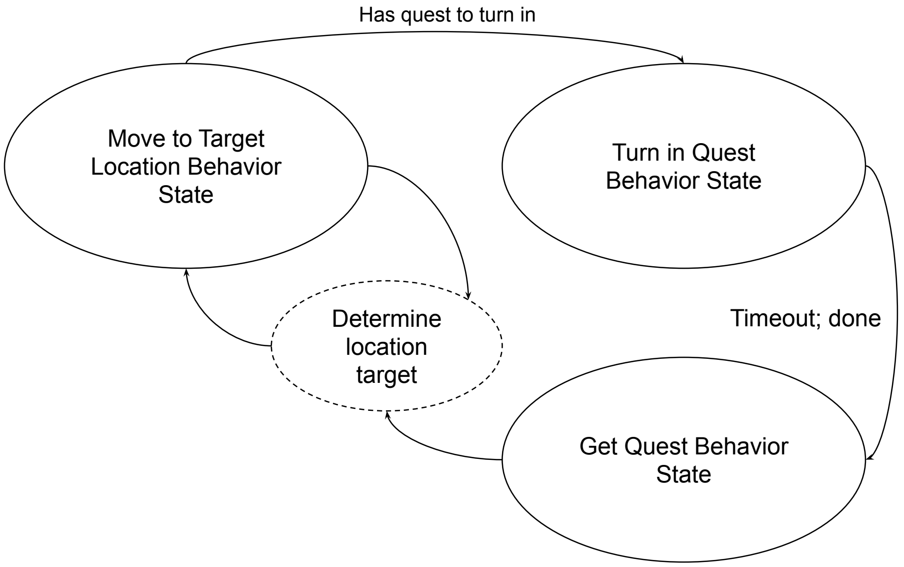

Has quest to turn in
Move to Target Location Behavior State
Turn in Quest Behavior State
Determine location target
Timeout; done
Get Quest Behavior State

Figure 5.6: Quest Strategy: Behavior state FSM for conversing and trading

Two important tables were also developed to make the bot act more human. First, there was a decision table that defined the probability to make many of the arbitrary decisions the bot faces, such as deciding to request a conversation or pick up an item. These probabilities were adjusted to make the bot act unpredictably. Second, a delay table was created outlining how long should the bot idle before taking an action. Without this, the bot would act at inhuman speeds. Actions such as moving locations, answering a request, or even indicating to complete a trade all were assigned an average wait time and variance. This way each time the action was chosen to be taken, and arbitrary wait time hovering around the average was determined. Both of these tables were aimed to capture the randomness and slowness of a human player.

Chapter 6

EXPERIMENT DESIGN

This chapter outlines the user-study performed to assess how indistinguishable human and bot players were from one another. Using the gameplay scenario described previously, a study was conducted where participants were placed in the game world with a varying number of A.I. bot players. Afterwards, participants filled out a survey that asked questions about what characters they believe were bots and additional free response questions to further classify participants' experiences.

6.1 Procedure

The user-study was crafted with a small level of deception. Participants were not aware of the fact that bots were posing as some of the eight unique characters during their play session. However, participants were informed that the two previously mentioned guild leader characters were NPCs. Not informing participants about the potential bot players aimed to see the effects on player immersion and game experience. Ultimately, bootstrapping an MMORPG should enhance human players' experience. As such, this study aims to evaluate how well the game system can pose bots as human players while not detracting from the gameplay experience.

Table 6.1: Ratio of bot and human players in each scenario version

Version	*Human players*	*Bot players*
A	1	7
B	4	4
C	8	0

Using the constructed gameplay scenario, the eight playable characters were split into three different distributions of human and bot players. Table 6.1 shows the ratios in each version A, B, and C. In version A, the game would be played by seven bot players and one human player. This looks to identify how the participant will fare when only interacting with bots. Can they easily identify the bots with only their own play behavior for comparison? Version C, covers the exact opposite scenario, where all eight playable characters are played by humans. Covering the question, can human players be identified without any bot behavior for comparison? Lastly, version B maintains an even ratio of bot and human players. With these three versions, we hope to cover the multiple situations that could be produced when bootstrapping an MMORPG and understand how that effects distinguishing human and bot players.

Table 6.2: Displays what scenario version each human played character is in for each session

Session	*Alison*	*Eldric*	*Florence*	*Holden*	*Knox*	*Paige*	*Tuesday*	*Wilfred*
1	A	A	B	B	B	B	A	A
2	B	B	A	A	A	A	B	B
3	C	C	C	C	C	C	C	C

With the three versions of the gameplay scenario, specific play sessions were outlined to ensure survey results were collected evenly from the versions. A single user-study session would have exactly eight participants grouped into versions of the scenarios. Table 6.2 the breakdown of the three sessions created. Two sessions had four version A scenario servers running and one version B scenario running. Four surveys could be collected for both scenario versions A and B. The two sessions differed only in which character roles were assigned to version A versus C. The third session option was to have a single version C scenario server run and collect eight surveys for version C. Each time, all three sessions were run, eight survey results would be collected for each scenario version and would require 24 participants. Additionally, after running each session type once, survey results would be collected from every character being played in each scenario version. Although unused, a fourth session was outlined to handle a potential 16 person session.

6.1.1 Running through a Session

After gathering eight people for a session, a strict script was followed to keep sessions as identical as possible and to ensure the required deception. The directions were written for running sessions virtually. Originally intended to be in-person sessions, the sessions moved to a virtual format using Discord due to the 2020 COVID-19 pandemic.

1. Await all eight participants to assemble in voice communication channel.
2. The gameplay scenario was introduced and study described as a "playtest of new MMO system". Any mention of bots or whether all eight participants would be in the same virtual world was omitted.
3. Participants were then asked to begin filling in the pre-survey questions including, consent form, demographics and a manual on how to play.
4. After pre-survey had been completed, participants were split into their own individual voice and text communications channels.
5. Each participant was told which character to play privately and a general time limit was set for all participants. Their goal was to reach Level 4 on their character by completing quests. They were encouraged to interact with the other characters to help them complete quests.
6. During the play session, the facilitator was able to privately answer any questions from participants, ensuring not to give away the presence of bots or the true study evaluation.

7. After a time limit of 15 minutes, participants were asked to complete the survey and the game servers were shut down.

8. While the survey included a debriefing of the true goal of the study, participants were also given the chance to ask any further questions from the facilitator.

9. Session was concluded.

To make sure participants shared their first-time experience, they could not be asked to play multiple scenario versions and past participants were asked to not reveal anything to future participants.

6.2 Survey

Within this survey, questions were asked to gather, demographic information, participants' gameplay experience, and the accuracy of identifying bot and human players. All questions were administered via a Google form. A full copy of the survey can be found in the appendix.

6.2.1 Demographic Questions

Basic demographic questions were included in the survey which included age, major, and gender. Additionally, the number of weekly hours playing video games and game genres played were collected. Lastly, participants were asked how much they agreed with the statement, "I have experience with massively multiplayer online games (World of Warcraft, Runescape, FF XIV, Black Desert, etc.)" on a Likert scale. The additional questions about video games were added to help bring insight into how accustomed participants may be to video games as well as MMORPGs specifically.

These answers were collected before participants played the game scenario. All other questions were asked afterwards.

6.2.2 Questions on Player Experience

To gauge each participant's general game playing experience, excerpts from the Game Experience Questionnaire (GEQ) were asked. Participants were asked how much they agreed with nine statements. The statements were excerpts taken from the GEQ to assess Flow, Immersion, and Competence [8]. Three statements corresponded to each category. As it was the first time participants had played the game scenario, their level of competency with the systems were measured. Flow and Immersion were measured to see how well the game scenario was keeping participants captivated.

Before directed questions were asked about which characters participants perceived as bot or human, several free response questions were asked. These questions aimed to further evaluate the player experience of the scenario. Expectations of the following questions were determined before any results were collected. The first question asked was, “Please, describe how you primarily interacted with other characters.”, and looked to assess the depth of interactions the participant had. Mention of conversations and trades made with other characters or mention of a novel unexpected interaction with other characters were classifications expected to show if the participant experienced complex character interactions. The participant putting into question the identity behind a character being a bot was another classification expected to identify bot players being too easy to distinguish. Lastly, mention of the lack of interaction with other characters was a potential classification. Answers were also expected to potentially satisfy none of the classifications as well and the option was left open to evaluators. The second question asked was, “How would you describe what you did during the play session, in one or two sentences?”, and aimed to look into the type of experience the participant had. A description of solo gameplay was a classification expected to imply a primarily single player experience. Description of attempts and successes of conversing and trading in game was a classification expected to imply a more multiplayer experience by the participant. Other classifications of the responses included, the participant voicing an alternative goal over the given one and the participant describing disinterest in the scenario. As before, responses could be classified as satisfying none of the above. Additional free response questions were asked so the above questions weren't too constraining.

Following these questions, a series of questions were asked about each of the unique playable characters. The participant was asked to identify which character they had played and then asked two questions for each of the other seven characters. They were asked if they had interacted with the character and whether or not they thought the character was human. The responses available were "yes", "no", and "I don't know". This series of questions was used to directly assess how indistinguishable bot and human players were to the participant.

In the last section of the survey, the scenario version is revealed to the participant informing them how many characters were played by bots. Participants are then explicitly asked, "Did you have an easy time recognizing whether a character was human or not?". With this last question, we wanted to see if participants would voice any thoughts if they knew players could be bots. We looked for if players recognized the system was intentionally built to obscure the human or bot identity, made a bid that further proficiency did make bots more distinguishable, or voiced that bots would not detriment their experience.

Chapter 7

RESULTS

This chapter covers the results of the experiment outlined. A preliminary user-study of eight participants was done early on and helped identify some bugs within the game scenario. The main user-study included 24 participants. The results of the 24 surveyed are reported here.

7.1 Preliminary Study

A preliminary study was run with just eight participants. Four played scenario A and four played scenario B. No data was collected on scenario C. The aim of this preliminary study was to practice the facilitation and deception required by the experiment and surface any problems with the system. Data from this study is entirely segregated from the primary study below. There was also no overlap in participants between the two studies. When the study was being presented several questions by participants potentially exposed the deception leading to a stricter script and more isolation used in the primary study. A few major U.I. and feature bugs were also discovered at this time. One participant logged in as the incorrect character and disrupted the scenario and all players. This was addressed by pre-filling character names into the log-in page and increased server monitoring during the initial start of gameplay.

7.2 Primary User-Study

24 participants provided survey data across three sessions, 8 participants each, within the span of one week. No changes were made to the game scenario or bot code. Minimal fluctuations in the facilitation of the study were noted, but the necessary deception of the participants was not broken in any session.

7.2.1 Demographics

The group of participants were of ages 19 - 29, see Figure 7.2.1. 11(46%) identified as female and 13(54%) identified as male. Participants were of many different majors; all majors were declared by no more that two participants except computer science with six participants declaring it as their major, see Figure 7.2.1.

In Figure 7.2.1, a majority of participants responded that they had experience with MMORPGs. Figure 7.2.1 showed an equal distribution of participants in each category of video game hours played per week. No participants responded that they played no video games.

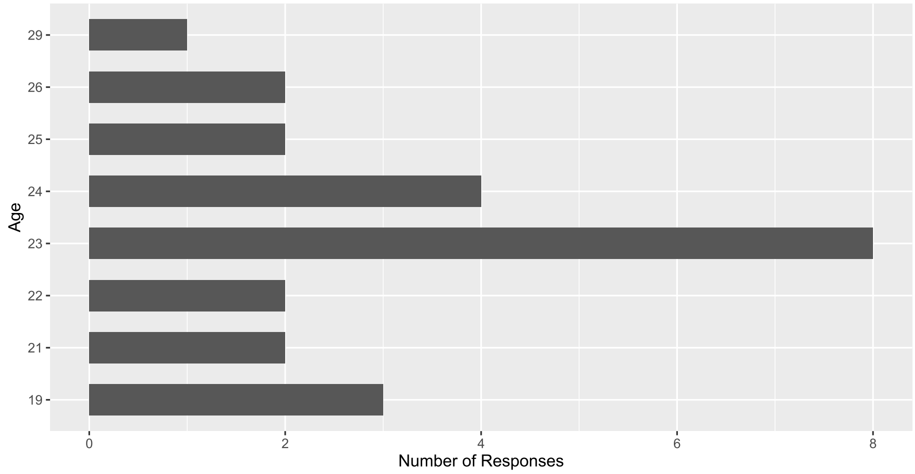


Figure 7.1: Ages of participants

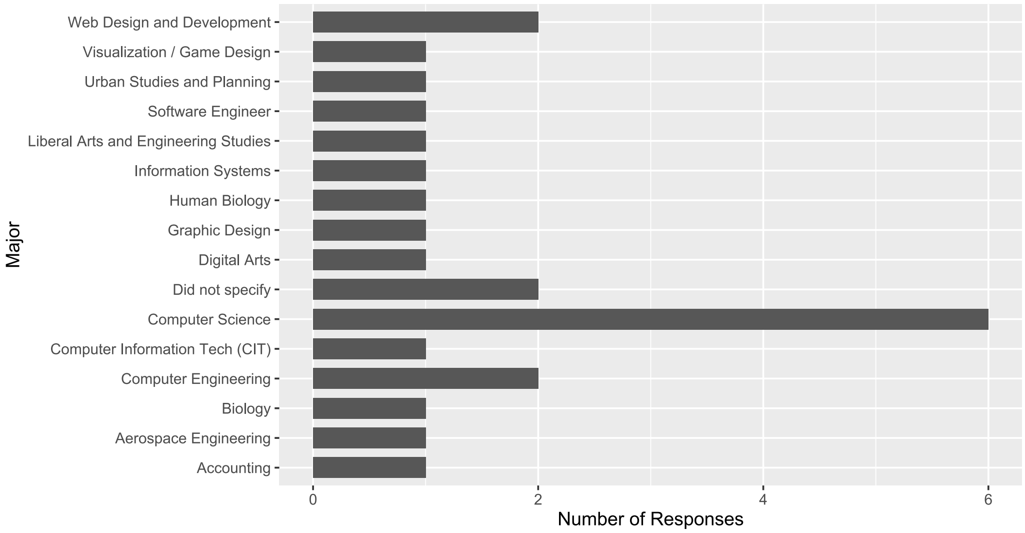


Figure 7.2: Majors of participants

7.2.2 Human and Bot Player Identification

For each of the seven characters aside from themselves, participants were asked if they interacted with the character and whether the character was played by a bot. This created 168 total data points of identification, 90 concerned bot played characters and 80 concerned human played characters. Table 7.1 and Table 7.2 show the breakdown of correct identifications, incorrect identifications, and "I don't know" responses. These counts are split by whether the character was actually a bot or human, and a second count is included that only counts identifications made when a participant also said they had interacted with the character. The total number of identifications made drops to 82 when only characters participants said they interacted with are counted.

Looking at the bot identifications in Table 7.1 first, it is shown that both incorrect identifications and uncertain responses take up the majority of answers. The number of uncertain responses drops significantly when only characters participants interacted with are counted.

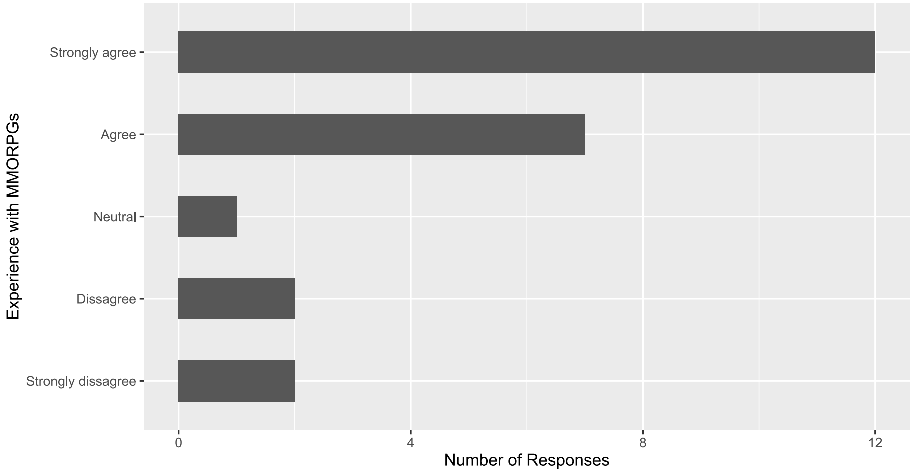


Figure 7.3: MMORPG experience of participants

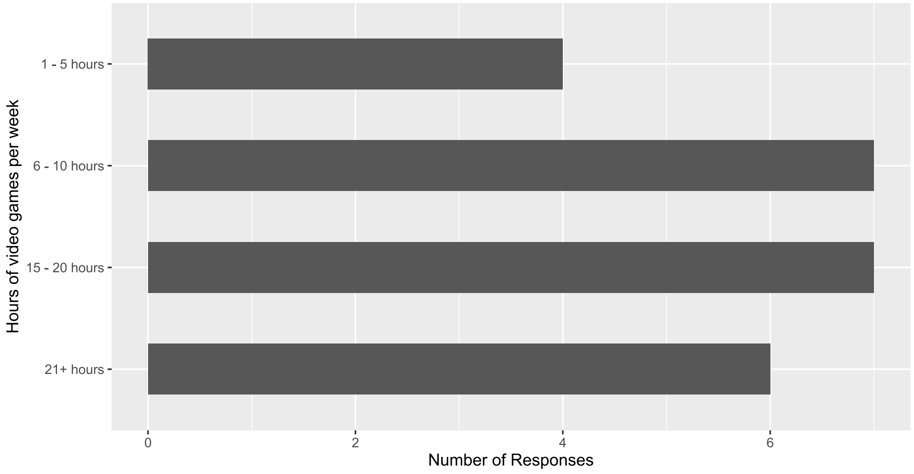


Figure 7.4: Hours of video games played per week

In the human identifications in Table 7.2 the portion of uncertain responses is shared with the bot identifications as well as its reduction when counting only the affirmatively interacted with characters. However, the number of correct and incorrect identifications leans heavily towards correct identifications in contrast to the identification of bots.

Both the previously mentioned tables' percentages do not represent accuracy of the identifications. The accuracy of human and bot identifications can be found by removing uncertain answers. That is to say, when a participant did guess yes or no

Table 7.1: Identification results of bot played characters

	correct	*incorrect*	*uncertain*
All	18(20%)	38(42.22%)	34(37.78%)
Interacted	18(28.57%)	27(55.1%)	8(16.33%)

Table 7.2: Identification results of human played characters

	correct	*incorrect*	*uncertain*
All	37(46.25%)	13(16.25%)	30(37.5%)
Interacted	20(60.61%)	7(21.21%)	6(18.18%)

to the question "Do you believe this character was a human player?" how accurate were they? Overall, with all certain identifications, the accuracy of identifications was about 50%. This is maintained whether all certain identifications are counted or just those of characters affirmatively interacted with. When split between bot and human identification, the accuracies diverge. Bot identification accuracy drops to about 33% and human identification accuracy increases to about 74%. This illustrates a difference in the identification of bot and human players.

To further delve into the human and bot accuracies, accuracies were calculated per participant and then averaged. When taking this average, the standard error was also found. This can be seen in Table 7.3. In this case, some participants did not guess at all, or due to the scenario version they were in, had no human or bot guesses. The number of participant accuracies calculated is included in the table as well. A difference between the accuracy of all certain human identifications versus affirmatively interacted with human character identifications can be seen. Although, the two bot accuracies did not differ by a significant enough margin. This is illustrated in the box plots shown in Figure 7.2.2, all data points are also plotted along each boxplot. Splitting the accuracies by categories such as played character's guild and scenario version did not show any significant difference between separated data. Also, the average number of yes and no responses given by participants, for bot or human character identification, was around 2 meaning the calculated accuracy per participant could fluctuate greatly.

Table 7.3: Accuracy identifications of human and bot players

Grouping	*avg. accuracy*	*standard error*
All (N=24)	52.53%	6.47%
All Interacted (N=23)	59.9%	7.79%
Bot (N=14)	36.73%	9.01%
Bot Interacted (N=13)	41.73%	10.24%
Human (N=16)	67.49%	8.49%
Human Interacted (N=15)	78.44%	9.32%

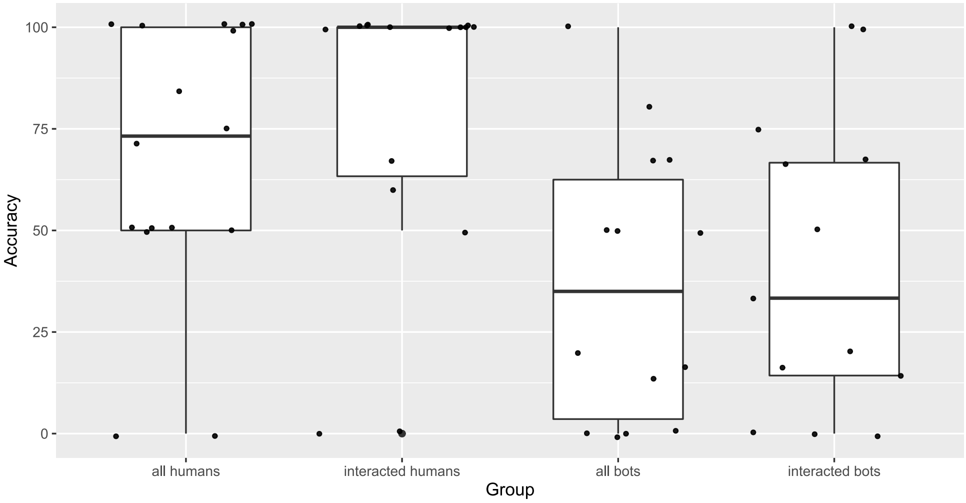


Figure 7.5: Boxplot of human and bot identification accuracies

7.2.3 Identifications by Scenario

The identification results were also divided by scenario to see what effect each had on participants. Overall counts of identification answers are shown in Figure 7.2.3. We see that scenario C has the highest correct answers, in-line with the higher correct identifications of human players. Scenario A has a much lower number of correct answers, also in-line with the lower correct identifications of bot players. Scenario B had human and bot players, and we see this reflected in its correct and incorrect answer counts. Additionally, no matter which scenario participants were in, the portion of uncertain answers remained the same.

Table 7.4 compares the accuracies of bot, human, and total identifications by scenario. In scenarios which both had bot or human players, identification accuracy did not change. This shows that the scenarios did not affect participants ability to identify human or bot players. Looking at total accuracy, we see the accuracy increase in-line with the decrease in bot players within the scenario.

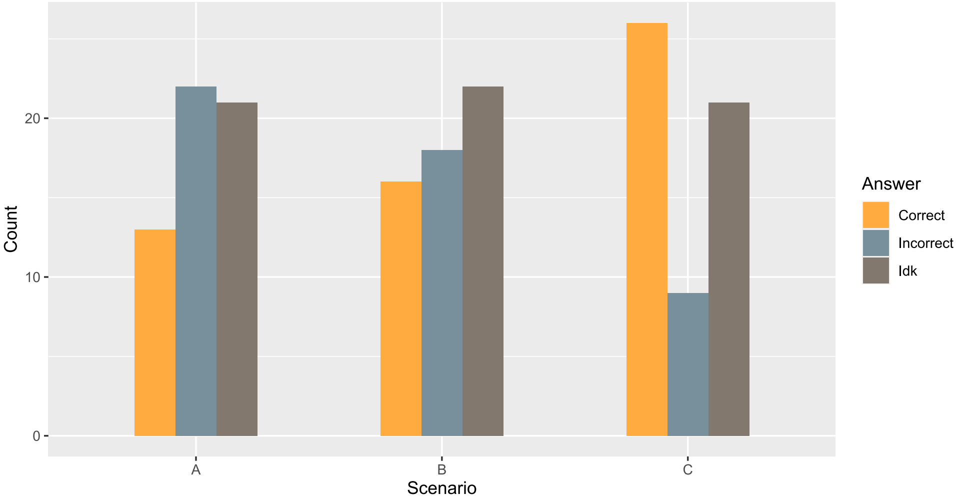


Figure 7.6: Identification results by scenario

Table 7.4: Accuracy results by scenario

Scenario	*Human*	*Bot*	*Total*
A	NA	35.37%	35.37%
B	68.75%	38.10%	53.84%
C	66.22%	NA	66.22%

7.2.4 Game Experience Questions

The GEQ exerts asked to the participant were weighted from 0 to 4 and averaged together to gather their overall flow, immersion, and competence while playing the game scenario. The flow score was 3.1 with a standard error of 0.19. Showing that the participants were overall captivated by the game and focused on playing. The immersion score was 2.3 with a standard error of 0.15. This indicated that the overall game's visuals and world were not very outstanding. The competence score was 1.96 with a standard error of 0.21. This low score was expected as it measured participants' competence with the game's systems. With only five minutes or so of instruction and 15 minutes of play, a relatively low competence score was expected. This score, however, was not as low as it could have reached.

7.2.5 Free Response Questions

Amongst all free response questions two indications were looked for from the responses: An indication of confusion and an indication of technical difficulties. Of the 24 participants, 5 stated confusion while playing and only 3 stated technical difficulties with the game itself. Several examples of confusion included, "I was unable to figure out how to get answers to the questions I was asking or engage in a trade. You probably explained it in the video, but I would have needed a refresher if I wasn't trying to figure things out quickly." and "I didn't know how to do trades, only how to do conversations....".

The categorizations of the three specific questions asked are shown in, Table 7.5, Table 7.6, and Table 7.7. Classifications were made from two different people, independently, and recorded only if the two chose the same classification.

Table 7.5: Categorization of question: "Please, describe how you primarily interacted with other characters."

Category	*count*
Talks about conversing and trading	11
Talks about emergent interaction	0
Questions identity behind character(s)	0
Mentions lack of interaction with others	5
None of the above	8

In the responses to the first question, 11 answers talked about conversing or trading while 5 mentioned a lack of interaction. Statements, such as, “I primarily asked other characters questions.”, “I asked them questions, and tried to trade with them.”, and “Tried talking to them first since that was always the first available option. And tried to trade information/items with characters to no avail.” were examples of talking about conversing and trading. Statements like “I tended to keep to myself and turn in the items i found” and “I only interacted with the guild leader. I was a bit confused with the question system of the game.” were examples of mentions of lack of interaction with others.

Table 7.6: Categorization of question: “How would you describe what you did during the play session, in one or two sentences?”

Category	*count*
Describes a solo experience	10
Describes attempts/successes at communication	7
Describes making an alternative goal	0
Describes getting bored	0
None of the above	7

The second question had statements fall into two of the expected four categories. Statements like “I walked around to find items, then dropped them off at my Guild.”, “I explored the various locations and discovered items to complete the quests assigned to me”, and “I did the information that the quests were asking for so I could turn them in instead of asking other people for answers” were examples of responses describing a solo experience. Statements like “I explored the town as well as figured out where i could pick up certain items. Then with those items, I attempted to be able to get people to give information for the items.” and “Tried to find the answers to information by talking with others” were examples of responses describing communication with others, a more multiplayer experience.

In the last question, many responses did not fit any expected categories. One response, “No, I just assumed everyone was human. But there were a limited set of behaviors,

Table 7.7: Categorization of question: "Did you have an easy time recognizing whether a character was human or not?"

Category	*count*
Recognized system intentionally built to obscure identity	2
Further proficiency would increase ability to distinguish identity	1
Voiced bots did not detriment their experience	1
None of the above	20

which could be easily emulated by an A.I.", was a response that recognized the intentional design of our system to obscure human/bot identity.

Chapter 8

CONCLUSION

This chapter describes the conclusions we draw from our results and its limitations. We also go over potential threats to the validity of our results and unforeseen influences that may have occurred.

Looking at the results of our user-study, we were able to get a large set of participants' identifications to be uncertain about the identity of the characters they played with. About 37% of answers were unsure. When looking at the accuracy of identifying a bot character it was at 36%. In this way, we can see that our system provides an environment where bots can be difficult to distinguish from human players. Even when the participant directly interacts with a bot and indicates so, the accuracy of identification does not increase significantly. With participants indicating they were captivated by the game and their goal, and no mention of getting bored we can be sure enough attention was given to the game to judge a character's identity. While there are other instances of games having bots pose as humans successfully, our system was able to maintain a fundamental MMORPG experience, which we hope opens the door to further complex scenarios in the genre being supported.

Although we have succeeded, there are a few limitations that were put in place. First, our MMORPG experience had a limited set of interactions as well as removing several ways of human expression, free form chat and movement. We also did not run our user-study play sessions very long. Since it concerns the MMORPGs, much longer amounts of playtime would better approximate the genre. This means we are unaware of how well our system fares with repeated log ins and interactions in hiding bot identities.

8.1 Threats to Validity

After concluding our user-study several outcomes may pose threats the validity of the results. For one, the accuracy in identifying human played characters is very high. This would imply that human players could stand out in comparison to bot players. However, this also could be strongly due to the deception used in this user-study in implying all characters were played by humans. This deception may have been orchestrated too well, over-assisting our system in its attempt to obscure bots. This could have contributed to our low bot accuracy. There were also statements of confusion and a relatively low score in competency from the GEQ excerpt. The lack of ability to play the game, or the distraction of trying to remove confusion could have softened the scrutiny of other characters' identities.

Looking at our survey questions for identifying characters, the wording of the question or fact that a third answer "I don't know" was allowed could have had unknown consequences on the study results. In some cases, it also seemed characters may have been misidentified.

Chapter 9

FUTURE WORK

This chapter lists suggestions for what could be explored, building on this work.

9.1 Enhanced Scenario and Testing

After running the user-study on the game scenario created, many notes were made about additional mechanics, better graphics, and more intuitive U.I. Further work could be done to improve the gameplay scenario, perhaps moving it into a more permanent setting like MMORPGs usually are. Alongside this, the survey on identifying human and bot players could be improved in wording and expectation. Additional user-studies without deception, telling participant upfront there are bots, could be devised to further test how discernible bot players are. This could further expand the context in which the system is confirmed to support indistinguishable bot players.

9.2 NPC Enhancement

Moving tangentially, work on NPCs that have a much wider range of actions within an MMORPG could be created starting from the bots designed in this work. An even more interesting game experience could be crafted taking into account these more complex NPCs.

BIBLIOGRAPHY

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

[2] Top 6 most popular MMORPGs sorted by population (2020), 2020. `https://altarofgaming.com/all-mmos-sorted-by-population-2018/`.

[3] 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.

[4] Express. `https://expressjs.com/`.

[5] N. Gupta, A. J. Demers, and J. E. Gehrke. SEMMO: A scalable engine for massively multiplayer online games. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD 08, page 12351238, New York, NY, USA, 2008. Association for Computing Machinery.

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

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

[8] W. IJsselsteijn, Y. De Kort, and K. Poels. The game experience questionnaire. *Eindhoven: Technische Universiteit Eindhoven*, 2013.

[9] What is spatialos? https://documentation.improbable.io/spatialos-overview/docs/what-is-spatialos.

[10] J. N. Kelly. Play time: An overview of the MMORPG genre. *Internetquelle: http://www. anthemion. org/file/playtime_over. pdf (08.03. 16)*, 2004.

[11] D. Maggiorini, L. Ripamonti, and S. Panzeri. Follow the leader: a scalable approach for realistic group behavior of roaming NPCs in MMO games. In *Artificial Life Conference Proceedings 13*, pages 706–712. MIT Press, 2013.

[12] Typescript - javascript that scales. http://www.typescriptlang.org/.

[13] 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.

[14] Node.js. Node.js. https://nodejs.org/en/about/.

[15] Neural MMO: A massively multiagent game environment. https://openai.com/blog/neural-mmo/.

[16] Phaser - a fast, fun and free open source html5 game framework. https://phaser.io/.

[17] PIPOYA FREE RPG Character Sprites 32x32. https://pipoya.itch.io/pipoya-free-rpg-character-sprites-32x32.

[18] PIPOYA FREE RPG TILESET 32x32 Pixel. https://pipoya.itch.io/pipoya-rpg-tileset-32x32.

[19] M. Schatten and B. O. urić. Social networks in the mana world-an analysis of social ties in an open source MMORPG. *International Journal of Multimedia and Ubiquitous Engineering*, 11(3):257, 2016.

[20] M. Schatten, B. O. urić, I. Tomičič, and N. Ivkovič. Agents as bots an initial attempt towards model-driven MMORPG gameplay. pages 246–258, 06 2017.

[21] M. Schatten, B. O. urić, I. Tomičič, and N. Ivkovič. Automated MMORPG testing–an agent-based approach. In *International conference on practical applications of agents and multi-agent systems*, pages 359–363. Springer, 2017.

[22] Shikashi's Fantasy Icons Pack. `https://shikashiassets.itch.io/shikashis-fantasy-icons-pack`.

[23] Socket.io. `https://socket.io/`.

[24] Tiled map editor. `https://thorbjorn.itch.io/tiled`.

[25] Vue.js. `https://vuejs.org/`.

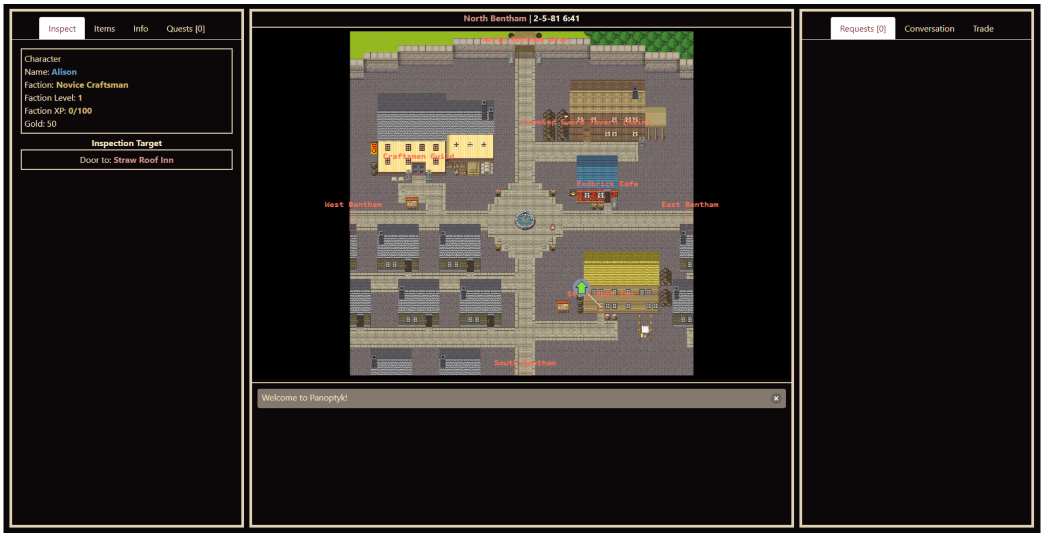
[26] W. White, A. Demers, C. Koch, J. Gehrke, and R. Rajagopalan. Scaling games to epic proportion. pages 31–42, 01 2007.

[27] M. Yannakakis. Hierarchical state machines. In *IFIP International Conference on Theoretical Computer Science*, pages 315–330. Springer, 2000.

APPENDICES

Appendix A

HUMAN WEB CLIENT USER INTERFACE SCREENSHOTS



North Bentham | 2-5-81 6:41

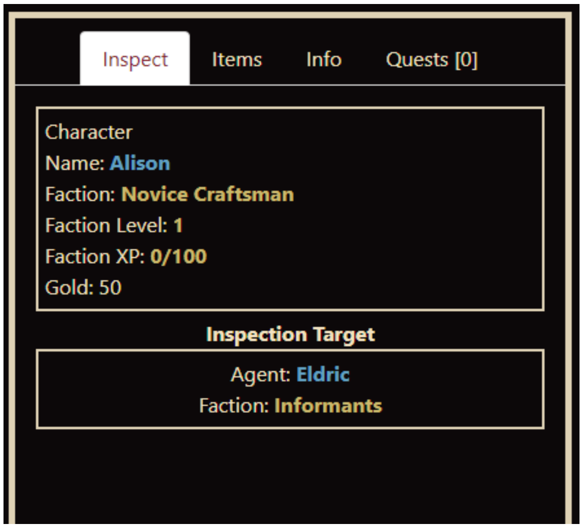
Character
Name: Alison
Faction: Novice Craftsman
Faction Level: 1
Faction XP: 0/100
Gold: 50

Inspection Target

Door to: Straw Roof Inn

Welcome to Panoptyk!

Figure A.1: Human web client full screenshot

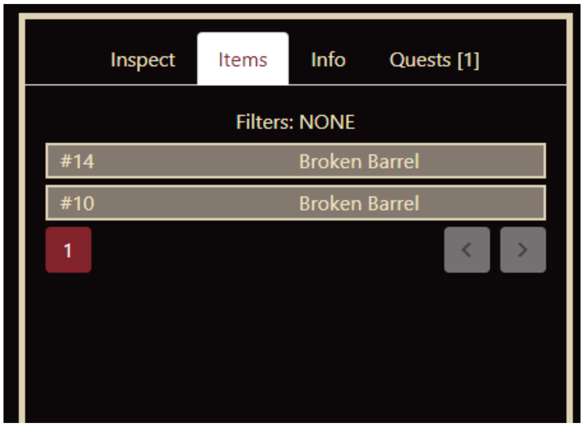


Character
Name: Alison
Faction: Novice Craftsman
Faction Level: 1
Faction XP: 0/100
Gold: 50

Inspection Target

Agent: Eldric
Faction: Informants

Figure A.2: Inspect tab first tab on left side

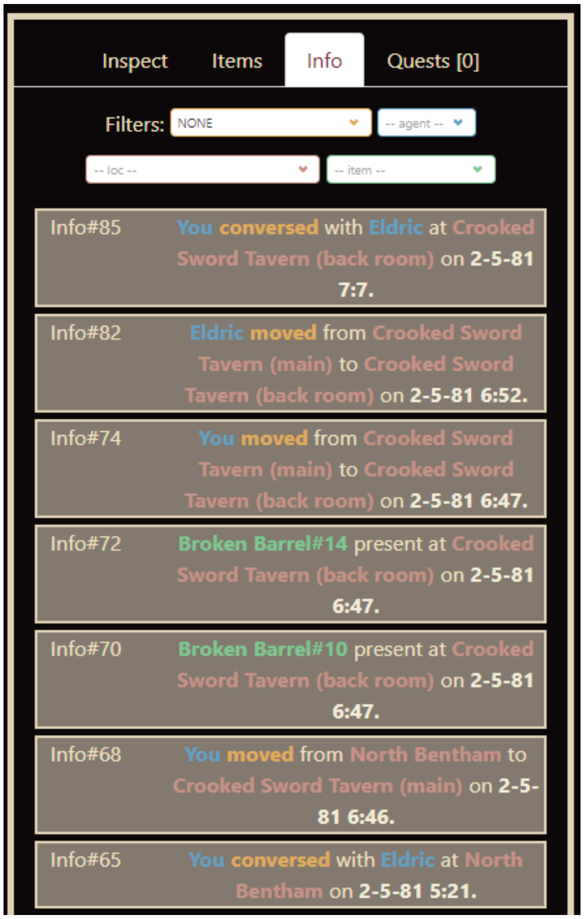


Filters: NONE

#14 Broken Barrel

#10 Broken Barrel

Figure A.3: Item tab second tab on left side, displays inventory



Filters: NONE

Info#85 You conversed with Eldric at Crooked Sword Tavern (back room) on 2-5-81 7:7.

Info#82 Eldric moved from Crooked Sword Tavern (main) to Crooked Sword Tavern (back room) on 2-5-81 6:52.

Info#74 You moved from Crooked Sword Tavern (main) to Crooked Sword Tavern (back room) on 2-5-81 6:47.

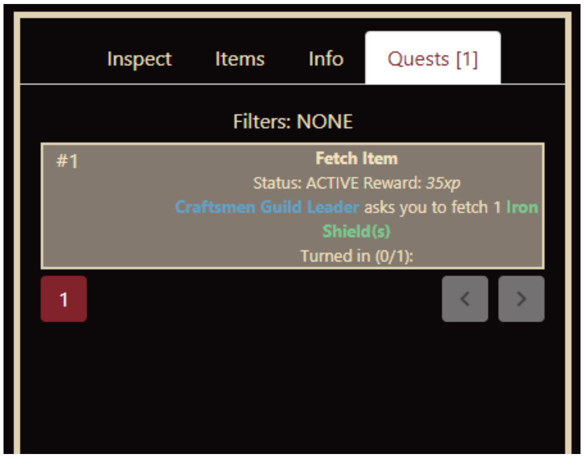
Info#72 Broken Barrel#14 present at Crooked Sword Tavern (back room) on 2-5-81 6:47.

Info#70 Broken Barrel#10 present at Crooked Sword Tavern (back room) on 2-5-81 6:47.

Info#68 You moved from North Bentham to Crooked Sword Tavern (main) on 2-5-81 6:46.

Info#65 You conversed with Eldric at North Bentham on 2-5-81 5:21.

Figure A.4: Information tab third tab on left side, displays known information



Filters: NONE

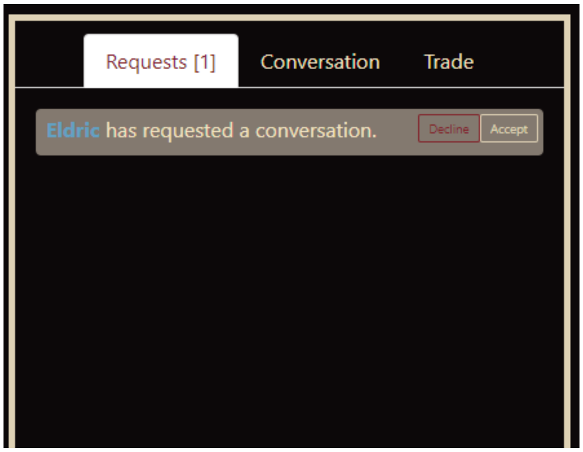
#1 **Fetch Item**

Status: ACTIVE Reward: *35xp*

Craftsmen Guild Leader asks you to fetch 1 Iron Shield(s)

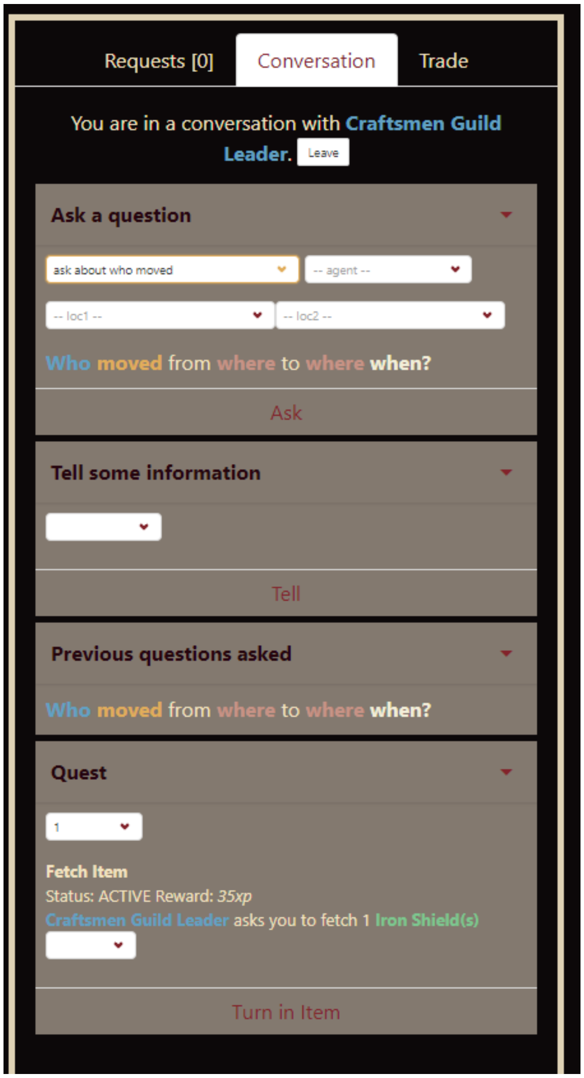
Turned in (0/1):

Figure A.5: Quests tab fourth tab on left side, displays active quests



Eldric has requested a conversation.

Figure A.6: Requests tab first tab on right side, displays requests by other characters



You are in a conversation with Craftsmen Guild Leader.

Ask a question

ask about who moved

-- agent --

-- loc1 --

-- loc2 --

Who moved from where to where when?

Tell some information

Previous questions asked

Who moved from where to where when?

Quest

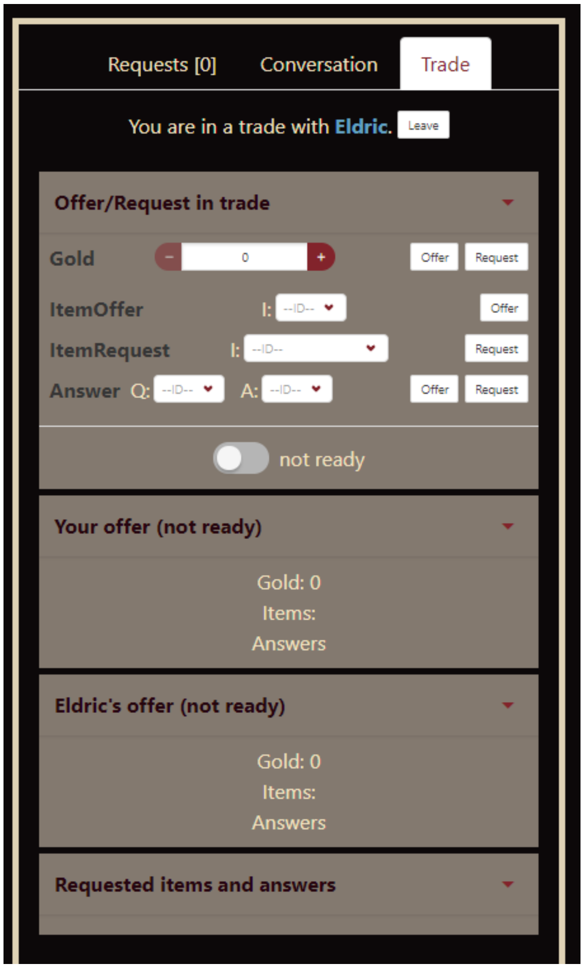
1

Fetch Item

Status: ACTIVE Reward: *35xp*

Craftsmen Guild Leader asks you to fetch 1 Iron Shield(s)

Figure A.7: Conversation tab second tab on right side, assists in conversing with another player



You are in a trade with Eldric.

Offer/Request in trade

Gold

ItemOffer I:

ItemRequest I:

Answer Q: A:

not ready

Your offer (not ready)

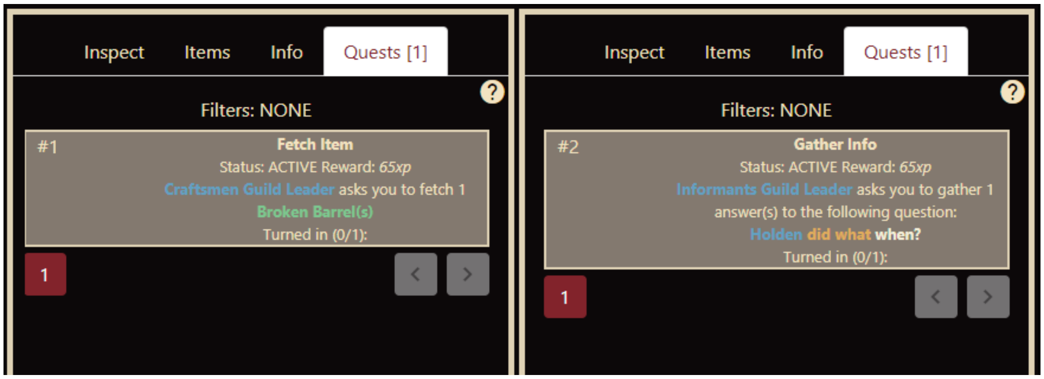
Gold: 0
Items:
Answers

Eldric's offer (not ready)

Gold: 0
Items:
Answers

Requested items and answers

Figure A.8: Trade tab third tab on right side, assists in trading with another player



Filters: NONE

#1 Fetch Item
Status: ACTIVE Reward: *65xp*
Craftsmen Guild Leader asks you to fetch 1 Broken Barrel(s)
Turned in (0/1):

Filters: NONE

#2 Gather Info
Status: ACTIVE Reward: *65xp*
Informants Guild Leader asks you to gather 1 answer(s) to the following question:
Holden did what when?
Turned in (0/1):

Figure A.9: Displays example quests from each guild, craftsman(left), informant(right)

Appendix B

USER-STUDY SURVEY

Bootstrapping Massively Multiplayer Online Games

Thank you for taking the time to participate in my master's thesis research! I ask all those participating to please follow this survey. Your email is asked for both receipt of your responses, if requested, and used as entry into the raffle.

* Required

1. Email address *

INFORMED CONSENT TO PARTICIPATE

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT:
“Bootstrapping Massively Multiplayer Online Games”

This form asks for your agreement to participate in a research project on game playing experiences. Your participation involves taking a survey after participating in a gameplay experience, and it is expected that it will take approximately 30 minutes total. There are no risks anticipated with your participation. Those in the game industry and game research community may benefit from your participation. If you are interested in participating, please review the following information:

The purpose of the study is to examine non-player character(NPC) behavior in Massively Multiplayer Online Games(MMOG). Potential benefits associated with the study include a greater understanding of NPCs in MMOG.
If you agree to participate, you will be asked to play a game and answer questions afterward. It is expected to take 30 minutes in total.

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. You may omit responses to any questions you choose not to answer. There are no risks anticipated with your participation in this study, as your survey responses will be maintained confidentially. Your email is used only to provide a receipt for your own response and entrance to the raffle of $10 Amazon gift card, and will never be used or publicized in any other form. Your email will be maintained as confidentially as allowed by the survey platform, Google Forms.

Data will be retained by the advisor indefinitely. If you would like your data deleted, contact the researches to request the deletion of both/either your email and data entries. Identifying information collected as part of the research, even if the identifiers are removed, will not be used or distributed for future research studies.
This research is being conducted by student Mitchell Miller, along with faculty member Dr. Foaad Khosmood in the Computer Science Department at Cal Poly. If you have questions regarding this study or would like to be informed of the results when the study is completed, please contact the researchers at (Mitchell) mmille95@calpoly.edu or (Dr. Khosmood) foaad@calpoly.edu.

The raffle will be for a $10 Amazon gift card. Approximately, a 1 in 10 chance of winning will be maintained scaling to the number of people entered into the raffle. If you wish to enter the raffle but do not wish to participate, you may contact Mitchell, mmille95@calpoly.edu and request entrance into the raffle.
If you have any concerns about the conduct of the research project or your rights as a research participant, 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 or pbrock@calpoly.edu.

If you are 18 or older and agree to voluntarily participate in this research project as described, please indicate your agreement by completing the activity and survey. Please keep a copy of this form for reference, and thank you for your participation in this research.

Link to this consent form text:
https://docs.google.com/document/d/1FT13r9DCudq5He7IHQ3ukMwJqCUgykh2RRY1dEKDosc/edit?usp=sharing

2. *

Mark only one oval.

○ Agree

○ Do not agree

Demographic Information

This information is purely optional. As stated in the consent form, your responses will be provided confidentially to protect your privacy.

3. Major

____

4. Gender

Mark only one oval.

○ Female

○ Male

○ Non-binary

○ Decline to state

○ Other: _______________________________

5. What is your age?

____

6. How many hours of video games would you say you play a week?

Mark only one oval.

○ 0 hours

○ 1 - 5 hours

○ 6 - 10 hours

○ 15 - 20 hours

○ 21+ hours

7. What types of games do you play? (Check all that apply)

Check all that apply.

□ Action, Shooters
□ Adventure
□ Simulation
□ Sports
□ Strategy/Puzzle
□ RPGs
□ Platformers
□ Multiplayer (2-8)
□ Massively Multiplayer Online
□ VR
Other: □ ____________________

8. Indicate your level of agreement with the following statement: "I have experience with massively multiplayer online games (World of Warcraft, Runescape, FF XIV, Black Desert, etc.)"

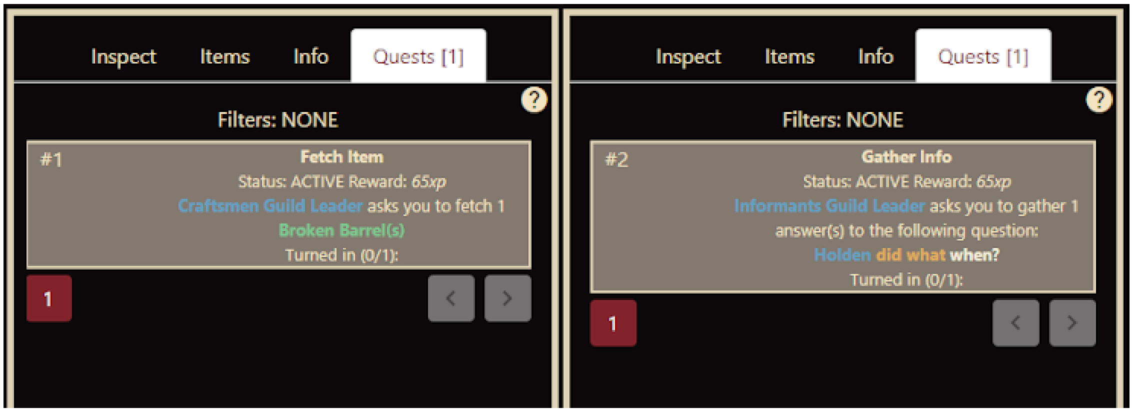
Mark only one oval.

○ Strongly agree
○ Agree
○ Neutral
○ Dissagree
○ Strongly dissagree

Before playing...

Please read through this section to learn about the controls and systems within the game you are about to play. Do not proceed to the next section until the gameplay experience has concluded, thank you.

OBJECTIVE: Please try to reach level 4 of your corresponding guild (informant or craftsman). To do so you must turn in quests. Some quests you will be able to complete by yourself others will require you to converse and trade with others. Below are what quests for a craftsman or informant will look like respectively:



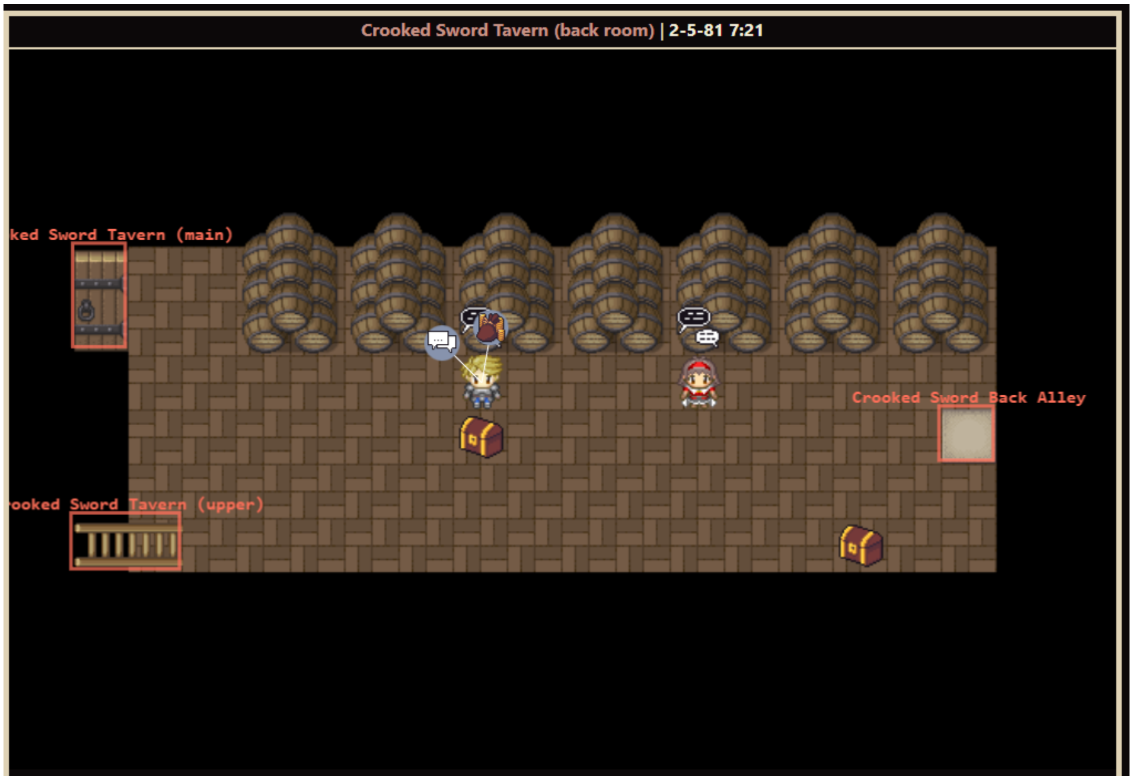
Filters: NONE

#1 Fetch Item
Status: ACTIVE Reward: *65xp*
Craftsmen Guild Leader asks you to fetch 1 Broken Barrel(s)
Turned in (0/1):

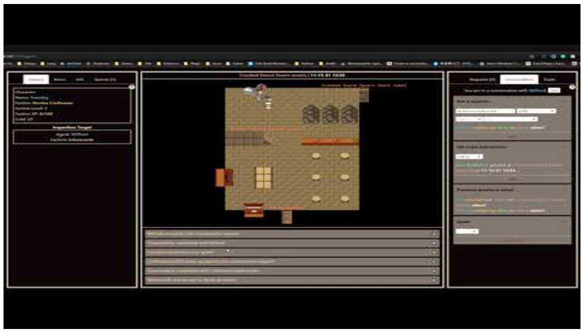
Filters: NONE

#2 Gather Info
Status: ACTIVE Reward: *65xp*
Informants Guild Leader asks you to gather 1 answer(s) to the following question:
Holden did what when?
Turned in (0/1):

CONTROLS: All interactions in this game are done using the mouse. Left-clicking on Characters, Items(chest icon), or doors(red outline) will open up options for you to interact with each: Converse & trade, pick up, and enter respectively for each target. The rest of the UI is operated like you would any other website.

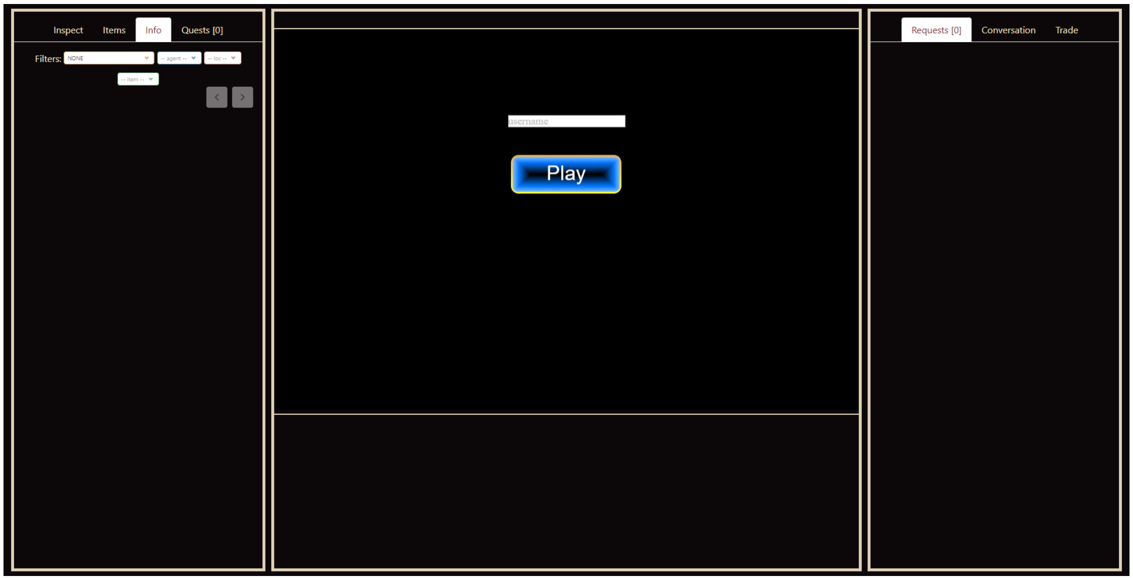

Crooked Sword Tavern (back room) | 2-5-81 7:21
ked Sword Tavern (main)
Crooked Sword Back Alley
ooked Sword Tavern (upper)

INFORMATION SYSTEM: Please watch this short video going over the information system and conversing/trading with others:



http://youtube.com/watch?v=af8GXVZdn7U

START: When you reach this screen make sure you can see everything in this photo. Your assigned character name should already show in the text input box. If it does not, please type your assigned character's name into the field (case-sensitive). Please wait to begin...



9. Have you completed the gameplay experience? *

Mark only one oval.

○ Yes

Gameplay Experience

Please respond to the following on how you felt about the game overall

10. Group (ask the host if not already filled)

Mark only one oval.

○ A

○ B

○ C

11. Please indicate how you felt while playing the game for each of the items

Check all that apply.

	not at all	slightly	moderately	fairly	extremely
It was aesthetically pleasing	□	□	□	□	□
I felt that I could explore things	□	□	□	□	□
It felt like a rich experience	□	□	□	□	□
I was deeply concentrated in the game	□	□	□	□	□
I was fully occupied with the game	□	□	□	□	□
I lost track of time	□	□	□	□	□
I felt successful	□	□	□	□	□
I was fast at reaching the game's targets	□	□	□	□	□
I felt competent	□	□	□	□	□

12. Additional comments?

Character interactions

Please answer the following questions, freely.

13. Please, describe how you primarily interacted with other characters.

14. How would you describe what you did during the play session, in one or two sentences?

15. Additional comments?

Character interactions (con.)

The following questions are about specific characters you may have interacted with during your experience. You do not need to answer the questions about the character you were playing as.

16. Which character were you?

Mark only one oval.

○ Alison
○ Eldric
○ Florence
○ Holden
○ Knox
○ Paige
○ Tuesday
○ Wilfred

17. Alison

Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

18. Eldric

Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

19. Florence

Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

20. Holden

Check all that apply.

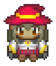
	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

21. Knox

Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

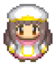
22. Paige



Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

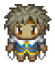
23. Tuesday



Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

24. Wilfred



Check all that apply.

	yes	no	I don't know
Did you interact with this character?	□	□	□
Do you believe the character was a human player?	□	□	□

Debriefing

Each experiment that you have participating in is aimed to assess how well an A.I. could pose as a human player within this MMOG(1) scenario. This could be done to assist in increasing the perceived player population of an under-populated MMOG as well as other potential applications.

To fully test this and to assist in hiding the truth, you were placed in one of 3 groups as follows:

A - you were alone with 7 A.I. bots

B - half (4) of the characters were A.I. bots

C - you were in a scenario where there were NO A.I. bot characters

Now that this has been revealed to you please answer the following questions to complete the survey. Thank you again for your time and participation!

(1) MMOG - Massively Multiplayer Online Game

25. Did you have an easy time recognizing whether a character was human or not?

26. Additional comments?

27. Did you partake in any outside communication with other participants during the gameplay experience or survey questionnaire?

Mark only one oval.

○ Yes

○ No