

Scheherazade's Tavern: A Prototype For Deeper NPC Interactions

Rehaf Aljammaz*
Elisabeth Oliver*
raljamma@ucsc.com
elaolive@ucsc.edu

University of California, Santa Cruz

Jim Whitehead
University of California, Santa Cruz
ejw@soe.ucsc.edu

Michael Mateas
michaelm@soe.ucsc.edu
University of California, Santa Cruz



ABSTRACT

In many games, NPC-player interactions play a vital role in gameplay. Previous literature has successfully shown how NPC interaction focused on social simulation is an effective means for creating dynamic characters such as in the games *Prom Week* and *Versu*. We believe that social simulation is a key element in the creation of complex characters, which is further aided by natural language interaction and knowledge modeling. In this work, we propose an architecture for player-NPC interactions built on top of the Ensemble engine that additionally incorporates chatbots and knowledge modeling technology, with the objective of making craftable and interesting NPCs more easily authorable.

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution International 4.0 License.

FDG '20, September 15–18, 2020, Bugibba, Malta
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8807-8/20/09.
<https://doi.org/10.1145/3402942.3402984>

CCS CONCEPTS

• **General and reference** → **Design**; • **Human-centered computing** → **Text input**; • **Computing methodologies** → **Modeling and simulation**; **Agent**.

KEYWORDS

NPC systems, NPC-Player interactions, social physics, game design

ACM Reference Format:

Rehaf Aljammaz, Elisabeth Oliver, Jim Whitehead, and Michael Mateas. 2020. Scheherazade's Tavern: A Prototype For Deeper NPC Interactions. In *International Conference on the Foundations of Digital Games (FDG '20)*, September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3402942.3402984>

1 INTRODUCTION

In many video games, nonplayer characters (NPCs) are an essential component within a game's design. They provide a wealth of information, aid in the immersion of players, and contribute to the story and lore. Done badly, NPCs can bore players, mislead them, and break the players' magic circle [4]. Simply, NPCs can hinder or enhance a game.

Current research has shown great strides in NPC development through computational systems, story structures, and social simulation. Social simulation enables unique experiences relying on the player's use of social mechanics (ex: talk, gossip) in games. Players in a social simulation-based game interact socially akin to how a player solves problems in a puzzle game [9]. Classic work such as *Façade* [7], and more recently *Prom Week* [17], have demonstrated the use of natural language, drama management, and social simulations to create a unique player experience.

In this paper, we present Scheherazade's Tavern, a game prototype that aims to expand upon existing techniques for player-NPC interaction to allow for more in-depth, varied interactions. In particular, the system combines a chatbot-like interface with social simulation enacted via the Ensemble Engine (a preexisting AI framework for social simulations)[16] and knowledge modeling to create a game centered around human interaction. We believe that by expanding on these interactions, we can gain insight into what engages players and, in turn, what players gain from these interactions. In particular, we would like to support the exploration of the NPC's character via these interactions, hoping to build out the sense of the NPC's depth.

Other research has been done into character believability, where believability for autonomous characters entails certain requirements relating to that of animation such as personality, emotion, self-motivation, reaction, social relationships, and expression [6]. These have included an additional hard set of requirements for autonomous characters to portray, such as requiring the characters to have the ability to pursue the completion of their goals, implement simultaneous actions, react to situations, and exhibit broad capabilities of sensing, thinking and acting [6].

Our interest in character depth stems from this notion of believability but with our focus on whether the player perceives the characters to have an internal life, a coherent personality, and a consistent backstory.

Our prototype attempts to support and expand on the notion of character depth by including the following:

- Allowing the player to interact with the NPC through a natural language interface. We believe natural language invites the player to express themselves and supports the player's sense of exploration, as evidenced by work such as *Façade*, parser-based adventure games, and virtual assistants. Furthermore, Sali's work [15] has shown that natural language interfaces, while difficult to use, can reward players with high levels of engagement and enjoyment. In this prototype, we use a chatbot interface in lieu of a menu as the primary communication method.
- Allowing the NPC agency in the conversation. We believe by allowing the NPC to change the topic or direction of the conversation, the player would perceive them as entities with internal lives. In this prototype, this is prompted through the Ensemble Engine's social modeling.
- Allowing the NPC to know facts about the world they exist in and comment on them. This prototype models knowledge asymmetrically, with each character (including the modeled player character) possessing a unique subset of the facts in the world. This is the greatest departure from the CIF

system and its descendant the Ensemble Engine, as neither supported such asymmetric knowledge modeling.

- Allowing the author to include authored content alongside partially generated content created from what the NPC knows. In doing so, authorial control over the character and narrative is not lost to the machinations of the system, but rather, two forms of content, both hand authored and system generated, work together to support and expand each other, making up for some of the shortcomings of the other.

Furthermore, in order to validate our system's architecture and inquiries, we conducted a pilot study to examine the player-NPC interactions made possible by our architecture. We try to answer the following question: does our system's player-NPC interactions add to the NPC's sense of depth? Additionally, does it enhance player engagement and interest in the character?

2 RELATED WORK

There are many frameworks and systems established with the intention of providing a complex social simulation or a unique generative narrative system, most focusing on NPC interactions. This section will discuss academic and industry projects as they relate to different parts of our proposed model.

2.1 Social simulation systems

Comme il Faut (CIF) is a social simulation architecture that focuses on social agents [9]. Through CIF, authors can create rich social interactions between agents through the authoring of reusable rules for volition and actions. The authors have demonstrated the strengths of the CIF system by creating the game *Prom Week*, featuring an array of complex NPC to player interactions [17]. *Prom Week* features a puzzle-like play space for players, with the goal of helping one or more characters achieve some social state. Players can achieve this by directing any number of characters from a small cast to perform an action that they have some desire (as determined by the underlying CIF system) to do.

Other authors have built upon the CIF system. Sullivan et al.'s work in CIF-RPG and *Mismanor* [20, 21] advanced CIF by adding item and knowledge representation and additional character complexity (in multi-agent interactions). Unlike *Prom Week*, which featured a god-mode control of the social state without a player character, *Mismanor* focused on player-controlled avatars interacting with NPCs as another entity within the game's world.

Treanor et al. extended the Ensemble Engine, a CIF successor, by creating "social practices" [16]. Social practices are defined by the authors as "a normative pattern of social interaction that captures the nuances that result from the individual agent's situation in the social state" [22]. Social practices present a deeper expansion on Ensemble's social simulation, making games more interactive and playable at precise levels of interaction (e.g., at the dialogue level).

Furthermore, researchers have extended social simulations on other platforms. Shapiro et al. have implemented a real-time system using the Kinect to respond to social (player) cues, enabling moment-to-moment embodied gestural interaction with characters [18].

Other researchers have implemented social modeling architectures based on CIF within existing game engines. For example,

[10, 11] re-implement and extend CIF within the modding tools of the AAA game Conan Exiles. In [11] they created a CIF-based version of an NPC merchant and demonstrate that people find it more enjoyable and believable than the traditional merchant character, while in [10] they extend CIF to include a model of emotion and a belief system, though the beliefs focused on social state rather than facts that can be brought up in conversation as we're doing in this paper.

Façade [7] employs believable agents implemented within a reactive planner to enable conversational interaction within a dramatic scenario. But Façade does not include an explicit social simulation. Versu [3] employs explicit models of social practices that are similar to CIF and Ensemble social models, though with a focus on character centered appraisals of social state. Further, Versu makes use of menu-based rather than natural language interaction.

2.2 Natural language systems for games and characters

In the natural language processing aspect of our work, our focus is not on demonstrating a novel method within NLP research, but in demonstrating the use of existing methods that are readily accessible to game developers to add natural language interaction to a social-simulation-based game experience.

Within academic research, natural language has been used in experimental playable experiences with great results. In Façade, players can chat and interact with NPCs using a keyboard and mouse. A rule-based bottom-up island parser is used to recognize discourse acts from natural language [8]. More recently, the game SHRDLU (named after Terry Winograd's pioneering work in natural language processing and knowledge modeling) uses a natural language interface to allow the player to interact with robot NPCs in an adventure game set on a space station. This game makes use of an explicit logical model of the state of the world and agent beliefs, parsing player utterances into logical statements and queries, and generating responses from formal representations. Other uses of natural language exist in serious games, where researchers [13] observed that NLP has been mainly used for classification and speech-related tasks, noting possible future areas for natural language exploration.

Natural language interfaces have also been used in commercial games. For example, the adventure game Bots Colony [19] supports a speech-based interface allowing players to converse with robotic NPCs mid gameplay, while the game Event[0] [2] extends chatbot techniques to enable text-based interaction.

Within games, chatbots are still one of the primary approaches for including natural language interaction. Chatbots tend to statelessly map directly from patterns recognized in natural language input to natural language responses, though numerous extensions exist that add small amounts of state to condition the mappings from input to output. Two of the earliest chatbots, ELIZA [24] and PARRY [1], demonstrated how users actively collude in maintaining conversational coherence with these simple systems, an effect later termed the ELIZA effect [23]. The enduring popularity of chatbot approaches is due both to their relative implementation simplicity and to their effectiveness via the ELIZA effect. In this paper we demonstrate how chatbot approaches can be integrated

with a social simulation system to provide sophisticated state-based responses and proactive character turn taking.

2.3 Knowledge modeling

As mentioned above, the game SHRDLU makes use of predicate-logic-based representation of knowledge, allowing characters to reason about the world. However, our approach to knowledge modeling builds most directly on the approach in Talk of the Town [14]. Here, NPCs have a knowledge-base of facts about the world, organized to support fact update and retrieval (like a database) rather than formal logical reasoning. Each NPC knows certain things about who the other NPCs are, where they live and work, and what they look like. Extensive work was done modeling how this knowledge is transmitted from NPC to NPC, how this knowledge can be permuted or corrupted through transmission, and how much confidence a given NPC has in a given fragment of knowledge. We draw inspiration for our own knowledge retrieval from this system, but turn our attention away from physical traits and instead towards personality traits and remembered actions.

3 SYSTEM ARCHITECTURE

In our prototype, the player plays as a tavern keeper of a small town. We chose this setting as taverns and inns are one of the most familiar scenes a player faces in games, especially RPGs. To add interest, we shifted the roles traditionally associated with the player and NPC. In this instance, the player plays the tavern keeper while the NPC serves as the patron. We believe these settings can build on player familiarity with roles and expectations.

In this prototype, players are able to talk to one of their patrons, an NPC named Marco, through a natural language text box. The NPC then responds or takes action via a speech bubble. The accompanying text box is disabled whenever the NPC initiates an action (i.e. presents new dialog). Furthermore, the demo includes a log that represents the player's current knowledge base, including facts about other characters. The prototype also includes a simple animation to convey Marco's current emotional state which is associated with an underlying numeric value.

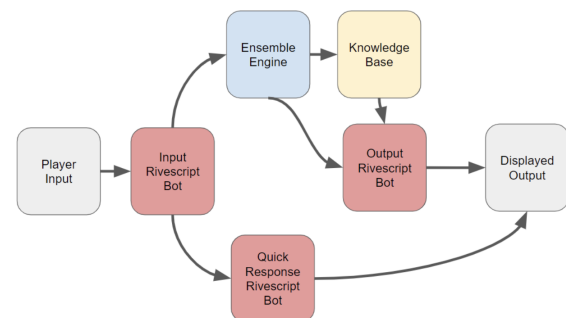


Figure 1: Overview of System.

There are three major components of our system:

- Rivescript Chatbots – for natural language input and output

- Ensemble Engine – for social simulation modeling
- Knowledge Base – contains facts a character knows about the world

An overview of the flow of control can be seen in Figure 1. Details of these components and their interactions with the player are discussed in the following sections.

3.1 Natural language interface

Our natural language interface relies heavily on an existing open-source chatbot scripting language called Rivescript [12]. Typical of chatbots, it is designed to support stateless mapping of natural language input to output. Finding the appropriate response is done through pattern matching and allows for wildcards, variables, and substitutions. It easily handles things like contractions and naturally handles irregular capitalization. Additionally, it supports “topics”, subdomains within the chatbot. These topics are intended to allow the chatbot to focus on a particular discussion topic and narrow down what the appropriate response should be. In using Rivescript, we demonstrate a new design approach that appropriates chatbot technology to map from natural language to an internal simulation model, and to then map from the output of the simulation model back to natural language. Thus the stateless chatbot technology is turned into a natural language parser and generator for a procedural simulation system.

In our prototype, we created three Rivescript bots. In the first, we take full advantage of its parsing capabilities to parse user input. But instead of the bot directly selecting an output to present to the player, it outputs actions (recognized intents) to the social simulation system (the Ensemble Engine). That is, we’re using Rivescript for intent recognition.

The second Rivescript bot fills the other end of the system’s internal pipeline. Its inputs are not natural language. Rather, they are structured text representing the action that the NPC wishes to take next. Here, the output of the bot is natural language, taking the abstract action and associated data selected by the Ensemble Engine and Knowledge Base and expanding it out into a natural language utterance which is then displayed to the players.

The final Rivescript bot is a traditional chatbot that maps directly from natural language inputs to outputs. It is filled with what we call “Quick Responses”. This bot is called upon when the player’s input does not match one of the actions we support in our social model, but which makes sense within the tavern domain. With this bot, hand authored content (in our case, a total of 755 responses) is able to fill in the holes our social model does not handle. Further, this demonstrates how heterogeneous authoring approaches can be integrated, in this case combining procedural social simulation and knowledge reasoning with traditionally authored chatbot content.

3.2 NPC agency

In our system, the NPC is able to periodically take autonomous action. How he picks which action to take is determined through the Ensemble social simulation model [16]. The Ensemble Engine was designed to allow NPCs to engage in complex social actions, taking into account their personal desires and goals to change their own state.

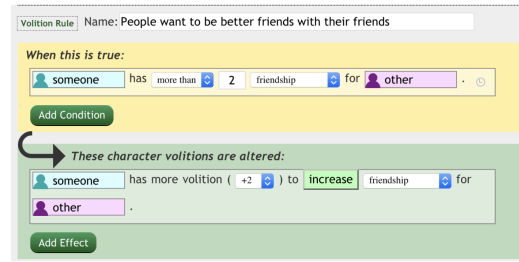


Figure 2: Example Ensemble volition, sample screenshot taken from the Ensemble interface. Volitions in Ensemble represent the motivations of a character to take particular actions.

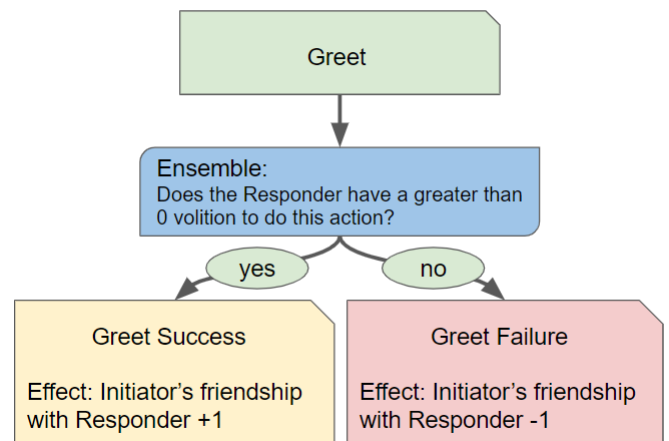


Figure 3: Example Ensemble action. Note that this “greet” action has both a success and a fail state effect. Which one is enacted depends on the friendship relationship between the two Ensemble characters.

Our Ensemble model includes two characters: the player and the NPC. Characters have two relationship values: “friendship” and “respect.” Friendship is intended to be the classic closeness value seen across relationship-related video games. Respect, on the other hand, is meant to represent how much the NPC values another’s opinion or advice.

The model also includes “volitions” (an Ensemble concept representing how much a given character might want to change some aspect of the social state, similar to utility) which encodes the motivations for taking action in this domain. These included ideas like “People want to be better friends with their friends” (which encourages people with moderate levels of friendship to increase their friendship) and “People with problems complain” (which encourages characters who have been flagged in Ensemble to have a problem to voice that problem to others). Many of these rules may apply to a single action summing together to contribute to that action’s final volition score. Figure 2 showcases an example ensemble volition.

These volitions are enacted by the characters through Ensemble actions. These include actions such as “Greet” and “Ask about family”. These actions have a variety of effects, from flagging characters as having complaints to increasing or decreasing their relationships. Ensemble chooses the action with the highest volition score for the NPC to take. Unlike in a traditional chatbot, this enables NPCs to proactively take conversation action rather than only passively respond.

Player action is also parsed into an Ensemble action if possible. As the player is also modeled as an Ensemble character there is a reciprocity of action representation between the player and the NPC.

This has two effects. First, it means the player’s input can modify the social state, potentially wrecking their relationship with the NPC or bringing it to new heights. Second, it allows us to use the fail or success effect built into Ensemble actions to give the NPC more freedom to respond. For example, the “Greet” action has a success effect, if the initiator has a high enough friendship, that will increase that friendship value. If the player greets the NPC in this way, the NPC returns a friendly response. However, this action can “fail.” If the initiator’s friendship is too low, the failure effect triggers instead, slightly damaging that already precarious relationship. The player would then receive a rude response to their greeting instead. It should be noted that not every action has both a success or failure effect, but all actions do affect the social state. An example Ensemble action can be seen in Figure 3.

3.3 Knowledge base

The knowledge base is a collection of facts about the world. Each fact has a subject and an opinion. Additionally, they contain either a state or an action-object pair. The subject is who or what the fact is about. The opinion is what the owner of the knowledge base thinks about that fact (a numerical value ranging from -15 to 15). States are descriptions of the subject, such as “is busy” or “is messy”. These facts represent some property of their subject. Meanwhile, facts that contain actions and objects record an event that took place in the world. In this case, the action is “what happened” and the object is “to what.”

Some Ensemble actions can be passed to the output Rivescript bot directly from Ensemble, completely skipping this module of the system. Actions like “GreetSuccess,” “Insult,” or “AskAboutFamily” do not need to contain any references to the knowledge base, and so can be passed directly to the Rivescript bot to be displayed to the player.

However, some actions, such as “TellAboutGoodThing” or “Ask-AboutFamilyResponse”, do require additional information to be properly displayed to the player. As Ensemble does not support knowledge modeling, Ensemble actions which require this kind of information get passed to the Knowledge Base before being translated into natural language to fill in the required knowledge.

Based on the action, a query is performed to look for facts with specific attributes or patterns of attributes. For example, if voicing a complaint, the fact should be one the NPC has a negative opinion about, or if responding to a question about their family, it should have a family member as either the subject or the object of the

fact. In general, queries involve testing whether a fact has a particular entity as either the “subject” or “object” attribute, or testing whether facts with numeric opinion scores are above or below a given threshold. A fact retrieved in this way will not be used a second time. This is done in an attempt to reduce the repetitiveness of the NPC.

Since the knowledge base is outside of Ensemble, Ensemble decision making does not make use of NPC knowledge in forming volitions. These knowledge base facts are used only to fill out the Ensemble Actions for display to players. Future work imagines using knowledge base queries as part of the volition formation process.

The player character has a knowledge base associated with them as well. The contents of this knowledge base are displayed within the “Notes” tab on the side of the screen. The idea was that as the NPCs brought up new facts, these facts would be added to this log. This was meant to act as a reference for players to help find things to try talking to the NPC about.

3.4 System flow

To fully illustrate how each of these components fit together, we will walk through some steps taken by a hypothetical player in different scenarios. Play revolves around the player inputting text to speak to the patron at their bar. In this prototype, that is a man named Marco.

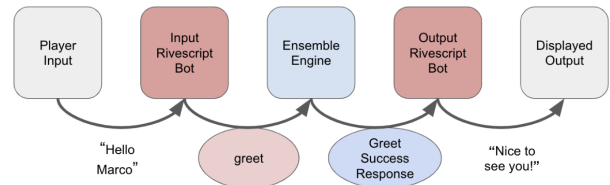


Figure 4: Greet example.

Imagine the player begins by typing “Hello Marco”. This string would be handed to the first Rivescript bot, which would recognize it as the pattern “hello[*]” which maps to the Ensemble action “Greet”.

This would be fed into the Ensemble engine, where it would be discovered the action succeeded. Ensemble would internally perform the GreetSuccess action and then pass that information on to the system at large.

GreetSuccess does not need a Knowledge Base fact to be verbalized, so we pass “GreetSuccessResponse” to the final Rivescript bot (“response” marks it as Marco is responding to a player-initiated GreetSuccess action, rather than him initiating a greet action himself, as these are different). The Rivescript bot would respond with something along the lines of “Nice to see you!” which would get displayed to the player in Marco’s speech bubble as shown in Figure 4.

Now imagine that the player then asked Marco, “How is your family?” Again, the first Rivescript bot would parse that to an Ensemble action, in this case, “AskAboutFamily.” This action would be executed in Ensemble, just as before. In this case, the action

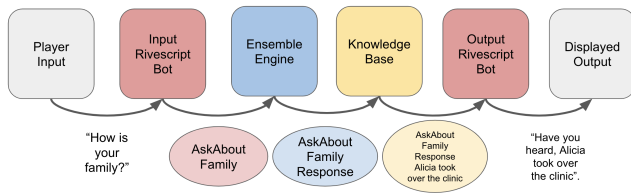


Figure 5: Ask about family example.

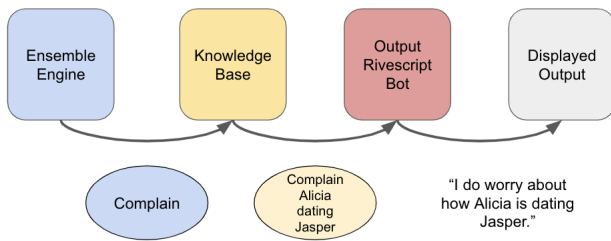


Figure 6: Marco initiating a phase of the conversation.

has only a success effect so “AskAboutFamilyResponse” would get passed along to the next step. In this case, a Knowledge Base fact is needed for this response, so the knowledge base would find a fact about Marco’s family. It might grab a fact about his daughter Alicia, or perhaps about how she recently took over the town’s clinic. If this were the case, “AskAboutFamilyResponse Alicia took over the clinic” would get passed to the final Rivescript bot. It would expand that string to something along the lines of, “Have you heard, Alicia took over the clinic.” This example can be seen in Figure 5.

Marco can also direct the conversation by taking an Ensemble action. Due to his internal state, one of the things he most wants to do is complain to the player. He might at this point decide to initiate a “complain” action. In this case, Ensemble would execute the “complain” action, which again has no success or failure state. This complain action would be passed to the Knowledge Base, as a fact is needed to fill in what he complains about. The knowledge base would pick a fact he has a negative opinion about, as he is only interested in complaining about things he does not like. It might pick “Alicia is dating Jasper”. The complain action and this fact would be passed to the final Rivescript bot. It might expand this combination out to “I do worry about how Alicia is dating Jasper”, as seen in Figure 6.

The player then might ask, “What did Jasper do?” This is currently not recognized as an Ensemble action, so the first Rivescript bot will pass the player’s input to our Quick Responses Rivescript bot instead. This bot will recognize the name “Jasper” and throw the player into its internal topic about Jasper. Marco then replies with “Jasper, Jasper, Jasper... Sorry, can you repeat your question again?” If the player does repeat the same question, the Quick Responses bot will match on the pattern within the Jasper topic that starts with “What” and includes the words “Jasper” and “do”, and output a random response associated with this pattern. In this case, the bot

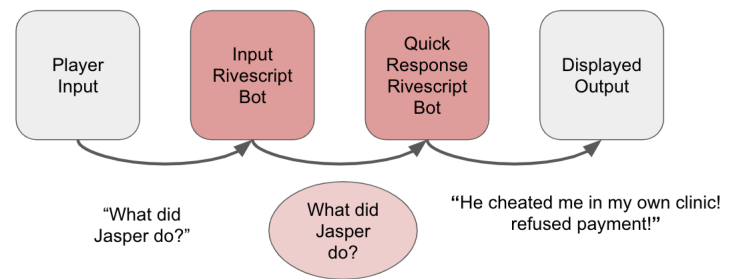


Figure 7: Example of the quick response Rivescript bot in action.

replies with “He cheated me in my own clinic! Refused payment!” as shown in Figure 7. The Quick Responses bot included 755 strings, with Jasper’s topic area containing 100 responses.

4 METHODOLOGY

In order to validate our system’s design goals and test if our system architecture enhances character depth, we conducted a small exploratory pilot study. The participants included nine players: eight students, and one narrative designer. All nine participants were familiar with video games.

The procedure included a set of preliminary and post-interview questions, tailored towards engagement, interest (particularly in backstories), NPC agency, and interaction methods. The preliminary questions inquired upon the players’ past experiences in video game genres and general interest in NPC interactions; these questions were meant to gauge the players’ prior experience and account for biases prior to gameplay.

After the preliminary questions, we introduced players to the scenario presented in the prototype as a browser game. We provided our players with control instructions, brief context to account for the fantasy theme, title, the NPC character’s name (Marco), GUI elements (such as noting the speech bubble, NoteTab, and chatbox), and the role the player portrays. Figure 8 shows a screenshot of the prototype.

During the study, the players interacted with the NPC, Marco, in a freeform style without a specific task, merely to chat with him. The players can do so by responding to Marco’s statements, typing anything, or relying on their notes tab. If we noted a bug and the player was stuck in a loop, we would suggest a statement to move or exit the conversation. Throughout gameplay, players were advised to apply the think-aloud method[5] standard in gameplay research. Through this method, we captured emergent responses and gained additional insight into the system and player interactions. At around the ten-minute mark, we interrupted players and started the post-interview process.

Post gameplay, we asked the players a set of interview questions. The interview questions were a mixture of short and long questions that inquired upon the player’s sense of engagement, a recollection of memorable moments, use of natural language, and NPC agency as it relates to our architecture and character depth definition. We

also included one scalar question, related to the overall player's experience and satisfaction. The following subsections discuss our initial results in more detail.



Figure 8: A screenshot of the prototype. Left to right: Macro (NPC), chatbox, speech bubble featuring Marco's response, note icon, and notes tab.

5 PLAYTEST

Our playtest results are organized into three categories: engagement and interest, interface and natural language, and NPC agency. Please note, quotes in the following sections have been slightly altered to improve grammar.

5.1 Engagement and interest

When prompted on the general satisfaction with the system, the player's feedback post gameplay indicated that the system overall was somewhat satisfactory (measured on a scale from 1 to 5, 5 being "extremely satisfactory"), four players rated it with a 3/5, three players rated the system with a 2.5, 3.5, and 4, respectively. The remaining two participants were somewhat unsatisfied with the experience, rating it a 2/5, as illustrated by figure 9.

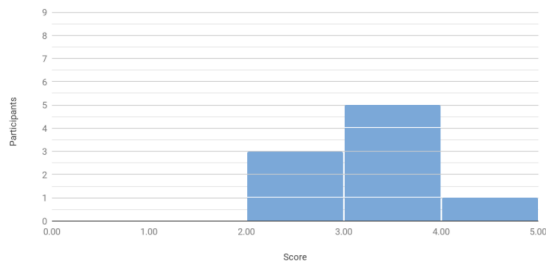


Figure 9: The chart represents the user's overall satisfaction with the system, when prompted post game play.

When prompted, eight out of nine testers recalled memorable moments within the demo. Interestingly most of the users (seven out of nine) recalled facts about other characters, as conveyed by the main NPC, Marco, rather than Marco's own facts or stories. For example, player 6 explained, "The stuff about Midna (a character

not in scene) being in the dark arts guild ... why Alicia does not like dogs, they were interesting statements that I wanted to have a deeper conversation [about]." Two players recalled facts about Marco's opinion of other characters as one user said, quoting Marco "I do not like them dating, it has drama!" One user, in addition to recalling other character facts, recalled a fact about Marco, in this case, his favorite drink.

Users were interested in obtaining additional information from Marco in the form of character gossip, small talk, and NPC facts. Seven users expressed interest in talking with Marco. However, two users noted problems with the conversation due to nonsensical responses or communication issues resulting from lack of authoring, as further elaborated in the limitations section. Additionally, eight users were interested in attaining extra background information about the character(s), including character motives, and interpersonal relationships, as represented by Figure 10.

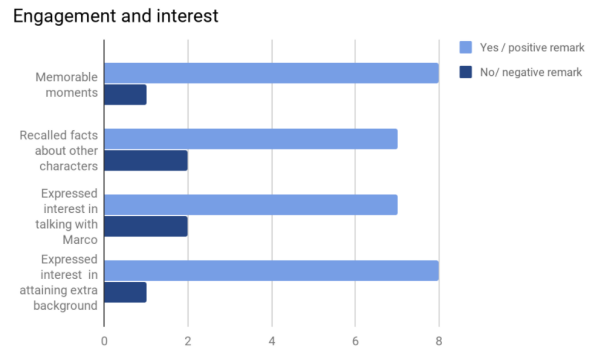


Figure 10: User engagement and interest.

5.2 Interface and natural language

The users' interaction via natural language was mostly positive, where 7/9 users felt that the interaction mechanic was intuitive or that it enhanced the experience, as illustrated in Figure 11. Although two users reported natural language hindering their experience, the majority preferred natural language with the incorporation of additional authoring. Players generally preferred this kind of open-ended dialogue with our NPC.

Interestingly, one user developed a personality for the player character through play, which we had not written for and had not expected in a system that put no constraints on the player's input. However, the use of natural language created a back and forth that allowed the player to develop a distinct personality for their player character. As player 4 said: "I liked that I came up with a character for the bartender. Like he thinks Midna is cool and he called Marco out [for spreading unflattering rumors about her]. I could play that and that was nice."

Our system's interface provided natural language as input through the text box; additionally, the player's log acted as a mixed-method, assisting the input process with topic options if the player so chooses. When prompted, the majority of users preferred open-ended dialogs as presented in this system (5/9), while the remaining

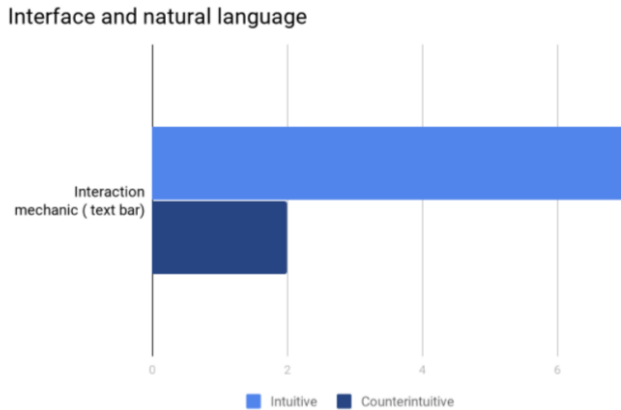


Figure 11: Interface and natural language.

four players each preferred mixed methods or discrete options (two each). The log (as a mixed-method of sorts) had mixed results. One user, for example, expressed how the use of the log improved their gameplay by freeing them from 'paralysis' or not knowing what to type, while another user thought of the log as too constraining.

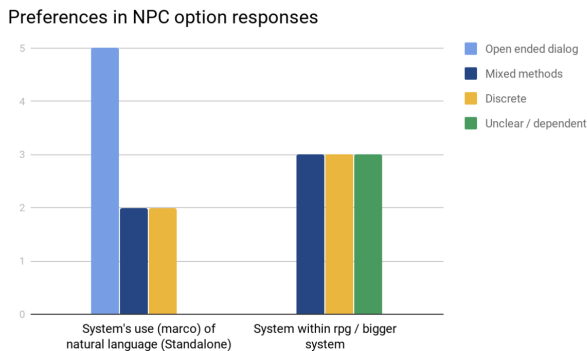


Figure 12: The users' preference in NPC option response between open ended dialogue systems such as our open text box with natural language understanding, mixed methods where the user is encouraged to draw from system prompts but is still given a freedom of input and discrete options such as those popular in choice based narrative games, as a standalone system vs. an embedded system. Some users were not sure which they preferred or felt it was dependent on other specifics of the game in question.

Lastly, we asked users about their opinion of having this or a similar natural language-based system implemented within an RPG scenario, as an element of a larger game. Players responded with a mixed result, where three indicated discrete options, three indicated a mixture of both, and the rest were unclear, but stating it depended on the specifics of the game this kind of interaction would appear in, as shown in Figure 12.

5.3 NPC agency

NPC agency in this system is presented by the NPC's (Marco's) ability to take action via the Ensemble Engine. To the user, this is mainly reflected through the NPC's conversation change or the player's disabled input bar. When we asked the player's opinion on Marco directing the conversation, we were met with mixed results.

Four users recognized Marco's attempts at conversation change as successful; one user pointed out a need for a transition at the graphical level beyond Marco's speech bubble and disabled text bar. Given that the system is a prototype, we anticipated this observation and discuss it further in the limitation section. Three users found Marco's actions had a negative impact, leading users out of a conversation or topic area they still wished to explore. To some, it was a jarring experience.

6 DISCUSSION AND LIMITATIONS

From our results, our system did encourage players to explore the characters' backstories, including those not present in the scene, supporting our notion of depth through natural language. Through this prototype, players were able to learn about the game's world from the main NPC through gossip, with players often asking about other characters. This was generally prompted by either information gained from the log or from mentions in conversation as a result of Marco's Ensemble actions.

Moreover, the prototype did leave an impact on players in the form of memorable moments; when prompted post gameplay, players could recall bits within the game that were interesting, sometimes even quoting Marco. Just as many of these were authored quick responses as they were generated from facts from Marco's knowledge base, suggesting both methods have the potential to capture the interest and attention of players.

Furthermore, players expressed interest in NPC's gossip ability. We believe these types of conversations are a promising form of storytelling. Players can assemble information from characters to form a story or role-play a character as we saw happen with one of our playtest participants.

Contrary to expectation, authoring limitations and random responses on Marco's part occasionally improved the player's perception of both the player character and NPC personalities. For example, when faced with unexpected input the NPC usually responds with inquiries or repetitions. Some players viewed the response as the NPC's own personality (ex: senile or fearful), playing further into the scenario and creating a positive Eliza-like effect and furthering the player's notion of depth as an NPC having a coherent personality.

Another unexpected result from our playtesting happened in two of the playthroughs when the NPC inquired about the player. Both users expressed excitement at the NPC showing interest in the player. This presents a promising area for future exploration.

However, the system is not without weaknesses in its current form. Open dialog systems tend to introduce an authoring problem, where the openness of the input requires a large amount of authoring. Players in our study did experience limitations due to limited authoring. While we found it difficult to author and predict a player's input, open dialog systems have their advantages. We found that natural language input enhanced the player's ability

to take on roles and contribute to a sense of freedom. Thus open dialog systems are a double-edged sword.

Limited authoring also affected some player's interest level at the end of playtime. Although, after the allocated time, some players did express interest in additional playtime, some had already exhausted the knowledge base and quick responses, leading to repetitions.

Similarly, the knowledge base is limited in content, affecting the range of topics an NPC can draw from as a result of taking action. The limited knowledge base, in turn, affects the NPC's capability in steering the conversation, causing repetitions in some cases or a generic phrase when the knowledge base is out of unspoken facts to share. Furthermore, the system in its current state needs better reasoning about when to change a conversation (initiate an Ensemble action) as the NPC agency in changing topics had mixed results, causing both satisfying transitions and jarring conversational jumps in equal measure. This makes it difficult to say at this stage if the NPC's agency via ensemble actions contributed to depth, as the results were mixed.

Since our system is a prototype, it lacks sufficient feedback elements, such as proper animations and sound effects. Animations can help in sensing the character's mood and signaling changes in the topic. Furthermore, the system was tested with one character. We have yet to explore the effects of talking with the other characters.

Overall, we believe our system's unique approach to combining chatbot technology, social simulation and knowledge bases managed to engage players with the characters. Notably, the system supported the exploration of an NPC character through a natural language conversation led in part by the NPC's autonomous choice in topic. This approach also helped create a more natural feel in the NPC interaction, increased the player's sense of exploration and NPC depth, and increased player interest in learning more information about different characters. While we had encouraging initial results, difficulties were also identified due to limited authoring in our prototype and lack of more sophisticated dialogue management.

7 CONCLUSION AND FUTURE WORK

In summary, this paper presents a novel architecture that combines different components of social simulation, natural language, and knowledge modeling to further enhance and study player-NPC interactions. Additionally, we conducted an exploratory pilot study to test the effects of the architecture in terms of character depth and interactions. Further research and system development are needed to fully explore the possible outcomes of the architecture, including a dialogue management system, additional authoring, and greater reasoning capabilities.

Future research should consider the potential effects of a more in-depth mixed-method approach to natural languages, such as providing suggestive text to the user instead of merely displaying assistive information such as the log. Furthermore, the study showed great promise in asymmetric character knowledge; we

think these conversations can be used as a great source for storytelling and structure as it captured player interest beyond the NPC's personal narrative.

REFERENCES

- [1] 2020. PARRY: The AI chatbot from 1972. <https://phrasee.co/parry-the-a-i-chatterbot-from-1972/>
- [2] K. Cross. 2016. Making a chatbot that drives a narrative in sci-fi exploration game Event[0]. *Famasutra* (2016). https://www.gamasutra.com/view/news/281842/Making_a_chatbot_that_drives_a_narrative_in_scifi_exploration_game_Event0.php
- [3] Richard Evans and Emily Short. 2013. Versu—a simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 2 (2013), 113–130.
- [4] Johan Huizinga. 2009. *Homo ludens: a study of the play-element in culture*. Routledge.
- [5] Patri Lankoski, Staffan Björk, et al. 2015. *Game research methods: An overview*. Lulu. com.
- [6] Aaron B Loyall. 1997. *Believable Agents: Building Interactive Personalities*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.
- [7] Michael Mateas and Andrew Stern. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, Vol. 2. 4–8.
- [8] Michael Mateas and Andrew Stern. 2004. Natural Language Understanding in Façade: Surface Text Processing. In *Proceedings of the Conference on Technologies for Interactive Digital Storytelling and Entertainment*.
- [9] Joshua McCoy, Mike Treanor, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. 2011. Comme il faut: A system for authoring playable social models. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*.
- [10] Luís Morais, João Dias, and Pedro A Santos. 2019. From caveman to gentleman: a CiF-based social interaction model applied to conan exiles. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. 1–11.
- [11] Miguel Oliveira and Pedro A Santos. 2019. A model for socially intelligent merchants. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. 1–8.
- [12] Noah Petherbridge. [n.d.]. RiveScript. <https://www.rivescript.com/>
- [13] Davide Picca, Dominique Jaccard, and Gérald Eberlé. 2015. Natural language processing in serious games: a state of the art. *International Journal of Serious Games* 2, 3 (2015), 77–97.
- [14] James Owen Ryan, Adam Summerville, Michael Mateas, and Noah Wardrip-Fruin. 2015. Toward characters who observe, tell, misremember, and lie. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.
- [15] Serdar Sali, Noah Wardrip-Fruin, Steven Dow, Michael Mateas, Sri Kurniawan, Aaron A Reed, and Ronald Liu. 2010. Playing with words: from intuition to evaluation of game dialogue interfaces. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. 179–186.
- [16] Ben Samuel, Aaron A Reed, Paul Maddaloni, Michael Mateas, and Noah Wardrip-Fruin. 2015. The ensemble engine: Next-generation social physics. In *Proceedings of the Tenth International Conference on the Foundations of Digital Games (FDG 2015)*. 22–25.
- [17] Ben Samuel, Mike Treanor, Josh McCoy, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin. 2019. Prom Week. <https://promweek.soe.ucsc.edu/>
- [18] Daniel G Shapiro, Josh McCoy, April Grow, Ben Samuel, Andrew Stern, Reid Swanson, Mike Treanor, and Michael Mateas. 2013. Creating playable social experiences through whole-body interaction with virtual characters. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- [19] North Side. 2016. Bots Colony. [Steam].
- [20] Anne Sullivan, April Grow, Tabitha Chirrick, Max Stokols, Noah Wardrip-Fruin, and Michael Mateas. 2011. Extending CRPGs as an interactive storytelling form. In *International Conference on Interactive Digital Storytelling*. Springer, 164–169.
- [21] Anne Sullivan, April Grow, Michael Mateas, and Noah Wardrip-Fruin. 2012. The design of Mismanor: creating a playable quest-based story game. In *Proceedings of the International Conference on the Foundations of Digital Games*. 180–187.
- [22] Mike Treanor, Josh McCoy, and Anne Sullivan. 2016. A framework for playable social dialogue. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- [23] Noah Wardrip-Fruin. 2007. Three Play Effects—Eliza, Tale-Spin, and Sim City. *Digital Humanities* (2007), 1–2.
- [24] Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (1966), 36–45.