12-2022

Trust-Based Social Behaviour in Role-Playing Games

Morgan Cartwright
Trinity University, mcartwri@trinity.edu

Trust-Based Social Behaviour in Role-Playing Games

Morgan Cartwright

Abstract

Although previous artificial intelligence research has done much to advance video games, not as much has been done to integrate that research into commercially viable titles. In this study, the game Barrel Smasher was developed, demonstrating a socially aware model of Non-Player Characters based on internal trust scores of other characters. The game allows the player to interact with these characters through dialogue, and it provides elements of Role-Playing Games like quests, items, and combat. The result is a game that combines social interactions and other forms of gameplay into a single, connected system. In doing so, the game creates interactive quest progression with more variation than is found in traditional quest systems.

Acknowledgments

I would like to thank my thesis advisor, Dr. Britton Horn, for all his assistance and feedback while researching, developing, and writing this thesis. I also would like to thank Starbucks, for helping me survive.

Trust-Based Social Behaviour in Role-Playing Games

Morgan Cartwright

A departmental senior thesis submitted to the
Department of Computer Science at Trinity University
in partial fulfillment of the requirements for graduation
with departmental honors.

November 18, 2022

Dr. Britton Horn, Thesis Advisor

Dr. Yu Zhang, Department Chair

Jennifer Henderson, AVPAA

Trust-Based Social Behaviour in Role-Playing Games

Morgan Cartwright

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Artificial intelligence (AI) has been the focus of many research studies for the past few decades, with studies spanning several different disciplines, including video game design. There are several ways in which AI can be used to expand video game design, one of which is the development of non-player character (NPC) behaviour in the role-playing game (RPG) genre. Exploring intelligent NPC behaviour in these games could potentially create unique game systems and simulate socially aware entities in a virtual environment.

But what makes an NPC socially aware? Extensive philosophical discussions could cover this subject, but for the purposes of this paper a socially aware NPC is defined as one which:

1. Interacts with other characters, both NPCs and players.
2. Has an internal model which represents its affinity for those characters (this is called "trust" in this paper and will be explored more in Section 3.2).
3. Reacts to other characters based on its internal model.
4. Is part of a social network, i.e. interactions between two characters have the potential

to indirectly affect or be affected by characters that are not directly involved.

One game that creates such socially aware NPCs is the game *Prom Week*. In this game, whenever characters interact, their internal models are modified based on a number of social rules. Their interaction may trigger further changes among other associated characters [9]. Another game, *Façade*, also creates two socially aware characters, which model relationships and play out interactions with each other and the player. This internal game system maintains the internal states of the characters and determine their actions based on a sequence of story "beats" [7]. However, NPC development is limited in both of these games, because the theoretical frameworks were created for research purposes and do not promote non-research oriented gameplay.

On the other hand, many commercially developed games create entertaining stories and combat, but NPC behaviour is simplistic. As a result, these games have a tendency to have static progression, making interactions predictable, and replaying social situations tedious. One example is Bethesda's open world fantasy game *Skyrim*[1], in which NPC interactions come primarily in the form of quests. In this game most quests come in one of two forms: linear paths for major stories, or minor side quests. The major quests follow a story, but have little potential for variation. The side quests usually involve completing a small task for a gold reward. In both cases, the quests will play out almost identically every time, aside from minor variations in dialogue and combat. There are some exceptions, for example there is a "Civil War" quest line that has two different paths, one for joining each side of two competing factions (the Imperials and the Rebel Stormcloaks). However, even in this case, each path is linear once it has been started. Outside of quests, dialogue is used for chatting, shopping, directing followers, and gaining information. This works well enough

[1] `https://elderscrolls.bethesda.net/en/skyrim`

for *Skyrim*'s combat-oriented gameplay, but it does not explore social dynamics.

Some success has been found integrating social systems as mods for existing titles; for example, one mod managed to implement *Prom Week*'s architecture, Comme il Faut, into *Skyrim* [3]. Such attempts may be restricted by the inherent limitations of modding a game that was not originally intended to support such behaviour.

The following study attempts to integrate socially aware NPCs into a traditional RPG setting in an entirely independently developed game. Using a system of trust based behaviour, the game achieves a non-linear quest framework, dynamic character interactions, and more play styles in comparison to traditional quest progression systems.

Chapter 2

Background

2.1 Video Game AI Overviews

A number of experiments in artificial intelligence have been designed using player experience modeling, content generation, data mining, and other forms of research. Many of these have shown promising results for increasing capabilities of video games [17]. However, research in NPC behaviour has been limited. In general, creating believable NPCs has not been as successful as other forms of video game AI research, due to NPCs not fulfilling player expectations of believability [15].

2.2 *Prom Week*

Prom Week is a game which uses the Comme il Faut architecture to simulate social interactions. In the game a number of high school students talk to one another shortly before an upcoming prom dance. Over the course of the game a number of social exchanges occur between two to three characters: an initiator, a responder, and an optional third character not participating in the conversation. In each of these exchanges, the initiator selects an intention, such as asking someone out on a date. Depending on the current social state of the characters in combination with a number of social rules, the responder chooses an outcome. This outcome can either be an acceptance or rejection of the initiators intended action, resulting in a change in the internal state for each of the characters involved. After the exchange, additional effects may occur depending on the traits of the characters as well as particular "trigger rules" that can be activated by the change in game state. [9] [10]

Using this system, *Prom Week* generates a social environment with many potential outcomes. However the player's control over the game is limited to initiating actions and watching the NPCs play out their relationship drama. Thus, the game is primarily a social simulator. Although simulation games have found commercial success in some cases (for example, the *Sims* series), it does not provide the level of player control needed for other genres, such as RPGs.

2.3 *Façade*

Façade was an attempt at sequencing and coordinating character actions in such a way that they appear to react to player input in a manner that approximates real social interactions. It relies on several underlying systems, such as a behaviour language (ABL), a drama manager, and natural language processing (NLP).

A Behaviour Language represents character actions as goals (for example, make a drink), and lays out a number of behaviours for completing those goals. These behaviours may occur sequentially (such as walking and then picking up an object) or in parallel (such as talking and making hand gestures at the same time). Several goals string together to form small moments that are significant to the story, called "beats." Beats are selected over the course of the game using a weighted priority algorithm, which chooses the most appropriate beat for the current social situation. This system creates a narrative that evolves over time to build a story.

The player is given free control to move around, interact with objects, and talk to the other characters by typing out text. Physical actions are interpreted for meaning based on how they situate the player in relation to the surrounding environment. Similarly, natural language processing interprets player input text for meaningful words and phrases. Both types of actions are converted into a "discourse act," which may change the outcome of the current story beat. As a result, the player is free to chose their own actions, and may affect the overall story by doing so. [6] [7] [8]

Although *Façade* provides player freedom and reactive story planning, it has some limitations. Content is sequenced by the system, but not automatically generated. As a result, the author must write a lot of code in order to realistically accommodate all possible player actions. Additionally, providing the player with freedom of choice may produce unexpected problems, because of how hard it is to predict human input. Furthermore, the complexity of the system makes it difficult to combine social behaviour with other types of gameplay, such as exploration and combat.

2.4 Other Systems

Various methodologies were researched for quest generation [1] [5] [11], personality-based behaviour [2] [4] [14], case-based reasoning [13] [16], and drama management [12]. These provided inspirations for different opportunities in creating behaviour and simulating stories. They were not implemented in the current study, but could potentially be used to expand further research.

Chapter 3

Methods

3.1 Game Design

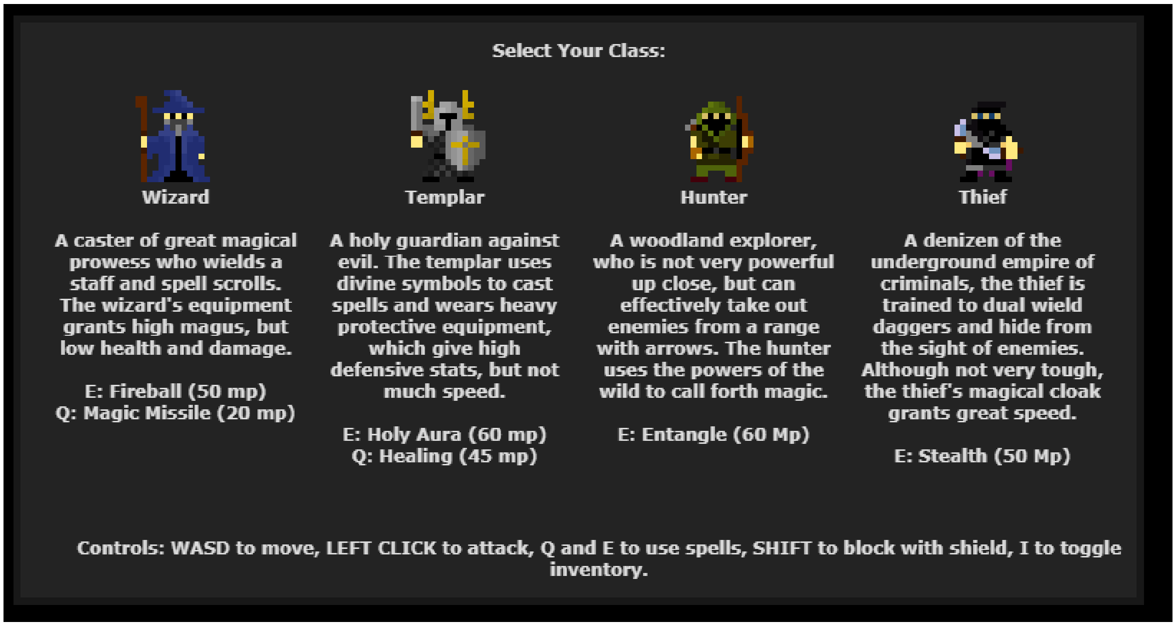
3.1.1 Game Development

The game itself was developed in HTML and JavaScript to create a system with a lot of traditional RPG elements, like player classes, monsters, quests, and equipment, etc. It is named Barrel Smasher and can be accessed online[1]. The player gets to control one of four fantasy themed heroes. They start in a town outside a castle, where they may talk to and help out three local citizens: a guard, priest, and a farmer. If the player wishes to journey into the wild, they can smash barrels and fight monsters inside for gold and loot.

3.1.2 Classes

The game starts by presenting the player with four different character class choices. These give the players various spells and equipment to use, providing advantages and disadvantages for combat. However, the classes do not affect social aspects of the game.

[1] `https://cs.trinity.edu/~mcartwri/WebApps/Barrel_Smasher_2.6.A/start.html`



Select Your Class:

Wizard

A caster of great magical prowess who wields a staff and spell scrolls. The wizard's equipment grants high magus, but low health and damage.

E: Fireball (50 mp)
Q: Magic Missile (20 mp)

Templar

A holy guardian against evil. The templar uses divine symbols to cast spells and wears heavy protective equipment, which give high defensive stats, but not much speed.

E: Holy Aura (60 mp)
Q: Healing (45 mp)

Hunter

A woodland explorer, who is not very powerful up close, but can effectively take out enemies from a range with arrows. The hunter uses the powers of the wild to call forth magic.

E: Entangle (60 Mp)

Thief

A denizen of the underground empire of criminals, the thief is trained to dual wield daggers and hide from the sight of enemies. Although not very tough, the thief's magical cloak grants great speed.

E: Stealth (50 Mp)

Controls: WASD to move, LEFT CLICK to attack, Q and E to use spells, SHIFT to block with shield, I to toggle inventory.

Figure 3.1: The player class selection screen. Each class has different combat advantages, including different equipment and spells. However, they all function the same in terms of social interaction.

3.1.3 Moving and Fighting

After selecting a class, the player can control their character by moving around with the WASD keys, attacking by clicking in a direction with the mouse, and blocking or using spells with other keys. For the combat part of the game, the player must destroy barrels, and then defeat a variety of monsters inside. Monsters will move around and attack similar to the player. The player can avoid attacks by moving or mitigate damage by blocking with a shield. They use weapons or spells to damage and kill the monsters. Once monsters are killed, they will increase the player's gold, and may drop items. If the player is killed, the game is over.

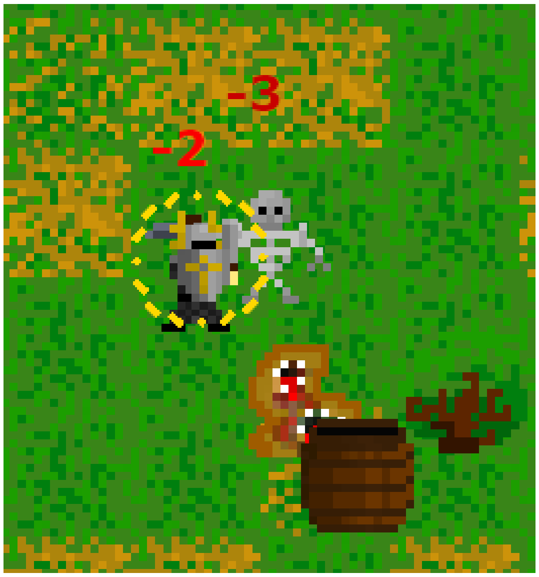


Figure 3.2: An example of a templar player, who is fighting against a slime and a skeleton.

3.1.4 Items and Gold

The player can buy and sell items in exchange for gold. These items provide different combat bonuses, such as increasing health, doing more damage, or making spells more effective. Additionally, certain quests may require obtaining certain items to complete.

3.1.5 Talking to NPCs

For the social part of the game, the player can initiate conversations with the F key and then may select dialogue choices using the mouse or number keys. The selections available may vary depending on the NPC. This will be explained further in the next section.

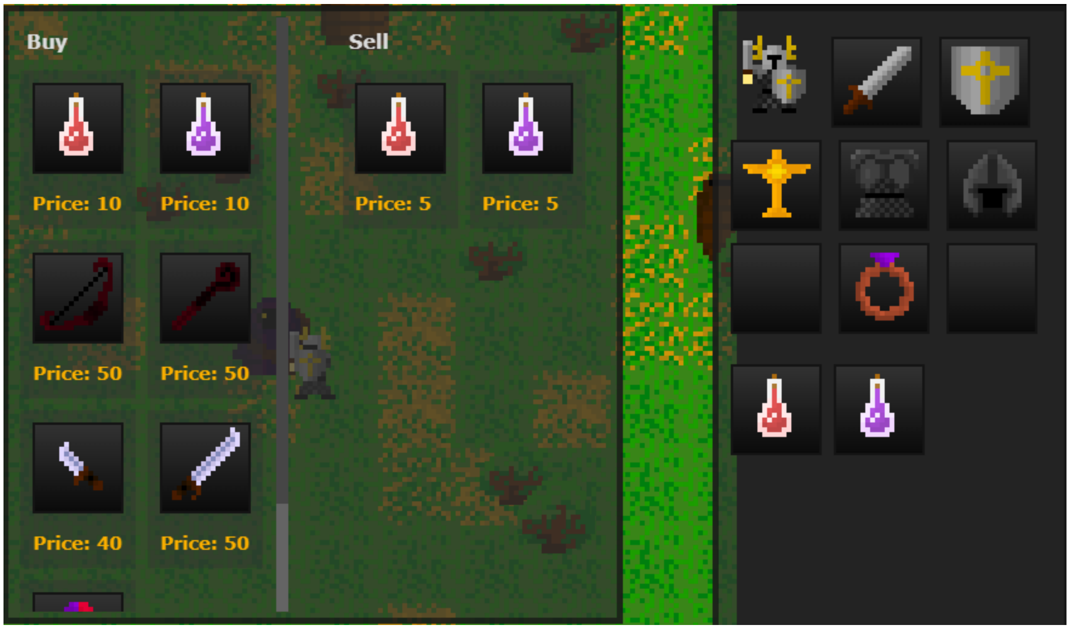


Figure 3.3: The player buying/selling items. Items provide the player with stat benefits, and may also be used in certain quests. Gold can also impact quests.

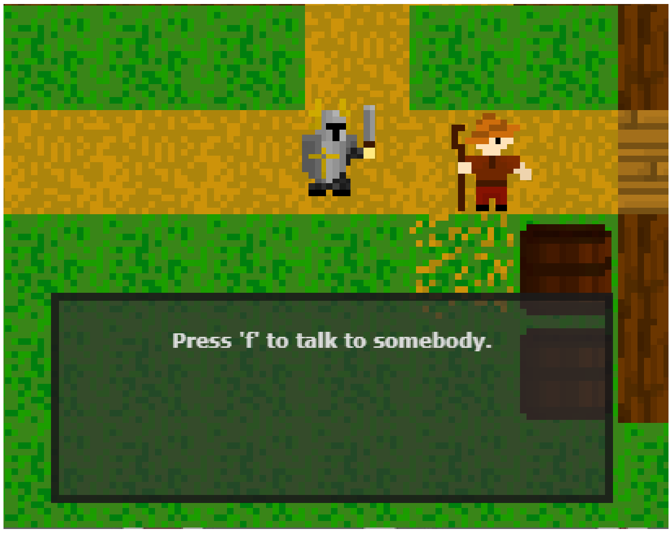

Press 'f' to talk to somebody.

Figure 3.4: The player and a farmer NPC. NPCs will move around and talk to other NPCs on their own. The player can choose to talk to NPCs by walking up to them and pressing F.

3.2 NPC Behaviour

3.2.1 Initiating Dialogue

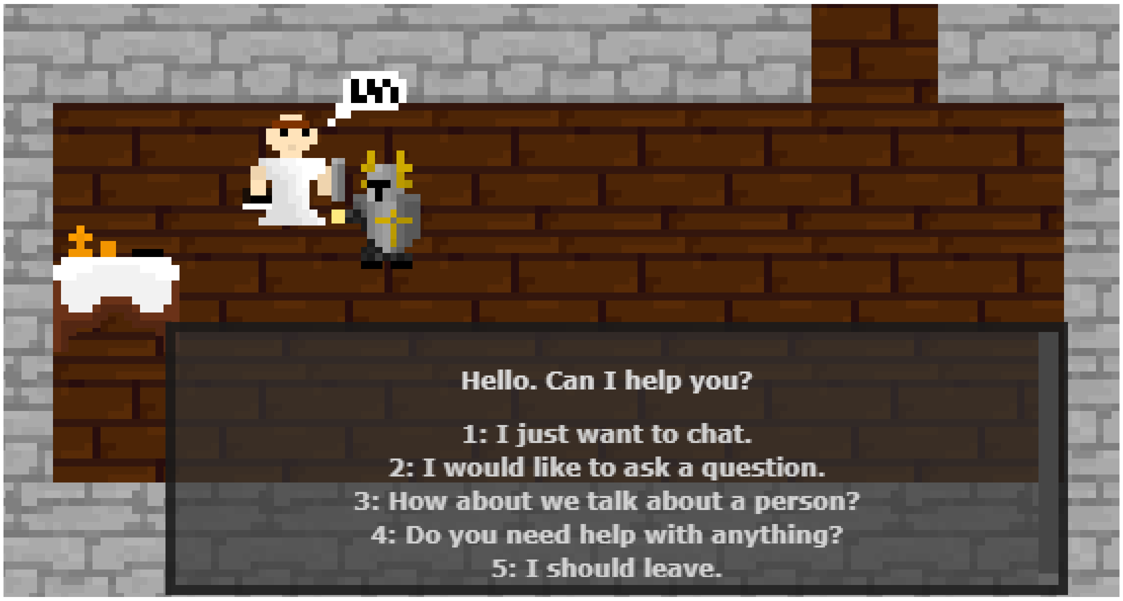
When first interacting with an NPC the player will have a number of possible conversation choices available. There are four major dialogue trees available:

- Chatting
- Questions
- Gossip
- Quests

All trees except for asking questions either affect or are affected by the NPCs internal trust scores. Trust here is defined as an integer value which is associated with another NPC or the player. There is no limitation on what the score can be, although within the scope of the first level, it tends to range from -50 to $+50$. Trust can be modified by conversations, and it affects how well the NPC responds to certain types of input from the player. Trust generally starts at $+5$ towards the player, and between -20 and $+20$ towards other NPCs.

3.2.2 Chatting

The most direct way to influence NPCs is by chatting. Different chatting choices are available for different NPCs, such as compliments and insults. Depending on the selection, the NPC may have an increase or decrease in their trust towards the player. The change amount ranges from -10 to $+10$. In order to prevent this form of conversation from being overused, each choice can only be selected once per game.



Hello. Can I help you?

1: I just want to chat.
2: I would like to ask a question.
3: How about we talk about a person?
4: Do you need help with anything?
5: I should leave.

Figure 3.5: The player talking to a priest NPC. The player has several options, including chatting, asking questions, gossiping about people, and starting quests. Chatting may increase or reduce the priest's trust of the player. Asking questions provides more information. Talking about other people influences the priest's trust towards someone else. Completing quests provides several different rewards, including increasing trust.

3.2.3 Questions

The player can also attempt to gather information by asking questions. Most questions do not involve trust, but there are exceptions. Certain questions may come in the form of asking for a favor. If the NPC trusts the player enough they may provide the requested aid.

3.2.4 Gossip

The player may also gossip about other characters. Doing so requires that they choose another NPC as the subject of gossip. The player may choose to offer their opinion, influencing the trust held by the listener towards the subject. The resulting change is based on a logarithmic formula:

$$\Delta listener.trust.subject = \pm ln(|player.trust.subject * listener.trust.player|)$$

where $player.trust.subject$ is dependent on the player's dialogue choice, and $listener.trust.player$ is dependent on the trust held by the listener towards the player. One may notice that the formula uses an absolute value of these two factors, however this is only done to prevent undefined logarithm operations, and does not mean the change is always positive. If the trust values are both positive or both negative, the change will be positive. If one is positive and the other negative, the resulting change will be negative. In other words, if the NPC trusts the player, they will be influenced by the gossip, but if they distrust the player, the influence will be reversed. This function is logarithmic so that very high or very low trust

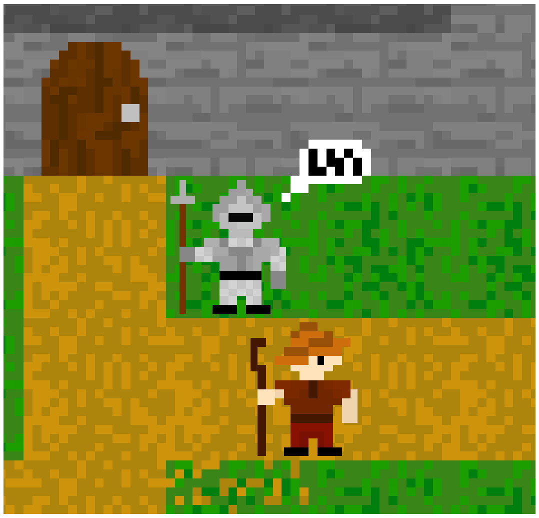


Figure 3.6: Two NPCs, a guard and a farmer, gossiping. When NPCs interact they can change the trust they have towards other people, including the player. The resulting change comes from a logarithmic function. It is based on the trust the guard and farmer have towards one another, as well as their trust towards the people being talked about.

values will not produce drastic changes. Additionally, NPCs will occasionally gossip with other NPCs. In this case a similar formula is used:

$$\Delta listener.trust.subject = \pm ln(|speaker.trust.subject * listener.trust.speaker|)$$

where *speaker.trust.subject* is determined by the trust value held by the speaker towards the subject. For a given NPC, the player may only gossip once about each subject before the listener's willingness to gossip must be reset. This reset will happen next time the given NPC gossips with another NPC.

3.2.5 Quests

Lastly, the player may ask the NPC if they have any quests to offer. After accepting a quest, the player must perform some sort of task or favor. These may include fighting monsters, providing items or gold, or talking to other NPCs. After completing the quest, the player may return to the quest giver to receive their reward, which will usually include an increase in trust (usually about +20).

Action	Effect	Change
Complete Quest	Positive	20
Compliment	Positive	3-10
Insult	Negative	3-10
Good Gossip	Positive	0-10
Bad Gossip	Negative	0-10
Liked Action	Positive	1-5
Disliked Action	Negative	1-5

Table 3.1: Trust modifications. This is a list of all actions that increase or reduce an NPC's trust score of the player. Additionally, gossip actions can affect the NPC's trust of other NPCs. Many of these actions are repeatable, but there are a limited number of compliments and insults allowed, and quests can only be completed once.

Action	Requirement
Open Door	50
Start Quest	0-10
Influence	0-100
Receive Aid	0-10

Table 3.2: Trust requirements. These are the trust scores needed in order for the player to receive certain types of help from NPCs. Opening the castle door, starting quests, and receiving aid have static requirements. Influencing NPCs through gossip is always possible but higher scores allow for higher levels of influence.

Chapter 4

Results

4.1 Gameplay

The goal of the first level is to enter the castle by convincing the guard to open the door. The player may also interact with two other NPCs to indirectly influence the guard. There are a number of ways to complete the level. Figure 4.1 provides two examples, which show how a game might be played.

4.2 Example 1

In the first example, they player starts by fighting monsters. Fighting monsters is an action that the guard likes, boosting their trust towards the player by +5 (for a total of +10, since it starts at +5). Fighting also provides the player with some gold. After that, the player chats with the guard to get another +5. Then they start the “Get Beer” quest. However in order to do complete it, they need to find the beer first. By asking questions, they find that the farmer NPC has some beer. They must first complete the “Pay Fines” quest to get it

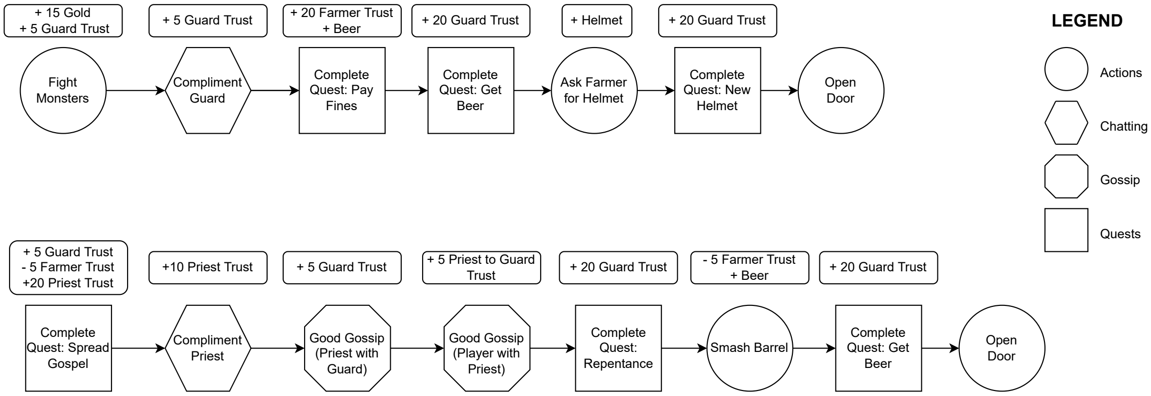


Figure 4.1: Gameplay examples. In the first example, the player performs favors for the guard until they trust the player enough to open the door. In the second example, the player uses their influence from another NPC, the priest, to affect the trust the guard has towards the player.

though. By talking to the guard and giving him some gold, the player is able to complete the farmer's quest, providing the beer and getting a +20 boost in the farmer's trust. Then they return to the guard to complete the beer quest, receiving +20 guard trust. The player asks about another quest "New Helmet." They find that they can get the helmet from the farmer, who already trusts the player enough to provide the required item. Once completed, the guard's trust increases by +20, for a total of +55, which is enough to open the door.

4.3 Example 2

In the second example, the player decides to start by talking to the priest instead of the guard. Completing his quest increases the priest's trust by +20, and has the side effect of increasing guard's trust by +5 and reducing the farmer's trust by −5. To further cement the player's ties to the priest, they decide to compliment him for another +10 trust. Later on, the priest initiates gossip with the guard. The guard begins with a trust towards the priest of +10 and at this point the priest has a trust of +35. So the increase is $+ln(|10*35|) = +5.86$ (rounded down to 5). The player decides to talk to the guard, and learns that a new quest "Repentance" has opened up. In order to complete it, they must talk to the priest, and increase his trust towards the guard using gossip. After doing so, the player is able to complete the quest, gaining +20 trust from the guard. Lastly, the player steals some beer from the farmer, and completes the "Get Beer" quest for another +20 guard trust (at the cost of some loss of trust from the farmer). Once again, the guard's trust has reached +55, so the player is able to complete the level.

4.4 Unanticipated Behaviours

In both scenarios the player actually has +5 more trust than is required (since the guard starts with +5 trust towards the player). This could be helpful, in case something unexpected happens. For example, the player may accidentally say something rude to the guard, or maybe another NPC decides to spread negative gossip. In general, when adding new levels, there should be more ways to gain trust than is required to complete the next goal. Doing so ensures that the player's choices are flexible, and that they do not get stuck behind a requirement threshold they cannot achieve.

4.5 Indirect Social Influence

The second examples shows that certain gameplay routes may rely on using gossip to affect trust indirectly. Although one NPC may be the primary target of a particular goal (such as the guard needing to trust the player in order to open the door), gossip may be used to indirectly increase trust through other NPCs. Additionally, NPCs may need to trust other

NPCs in order to advance some quest lines (for example, the “Repentance” quest requires that the priest trust the guard). So instead of focusing entirely on one NPC at a time, the player may wish to consider the entire social setting that they are interacting with.

4.6 NPCs and Combat

Although the examples focused more on the social side of the game, NPCs can help out with combat indirectly. For example, completing quests provides gold, which can be used to buy items. Or in the first example, instead of giving the farmer’s helmet to the guard, the player could use it on for their own purposes. NPCs cannot fight monsters directly (but this would make a good addition for future versions of the game). Conversely, combat may also be used to influence NPCs. Fighting monsters may provide trust if its an action that an NPC likes, and the gold and items dropped by monsters may be useful for some quests. Thus social and combat aspects of gameplay may both affect one another.

4.7 Trust Based Progression

The examples show that there is no strictly enforced chain of events. Each event – fighting monsters, completing a quest, or gossiping about other people – can be attempted or avoided as desired. However the system of trust maintains an overall progression. As more trust is acquired, more options are available to progress further. And as the player progresses further, they may gain more trust. So actions continually lead to other actions in a largely progressive cycle.

4.8 Losing Trust

In some cases trust may be lost instead of gained, but this is not entirely a bad thing. It both ensures that the player has obstacles to avoid or overcome, or can even be used intentionally by a clever player to assist them. For example, suppose the farmer distrusts the player and the priest distrusts the farmer. If so, the gossip system may actually increase the priest's trust of the player, since two negative trust values result in positive change.

Chapter 5

Discussion

5.1 Quest Progression

From the results shown, Barrel Smasher has more complex system of quest progression than most traditional video games. In the majority of story based RPGs, quests are hard coded into a sequence of logical events. Typically, quests follow a straightforward progression with few variations due to player choice. This may be due to authorial constraints, since as the tree splits off into more and more pathways, the number of quests creates exponentially larger story writing requirements. In less story based games, quests may be more isolated, and don't progress from one another. This allows the player to proceed in whatever order they wish, but sections off the world so it is not lifelike. Barrel Smasher combines combat, individual quests, and smaller conversations, where each aspect of the game can affect the others. In doing so, the game stands in between the previous two methodologies. Because the cause and effect of individual actions can spread across disparate parts of the game, the different parts of the game are connected. However, because quests are still somewhat independent, the player has a variety of progression choices.

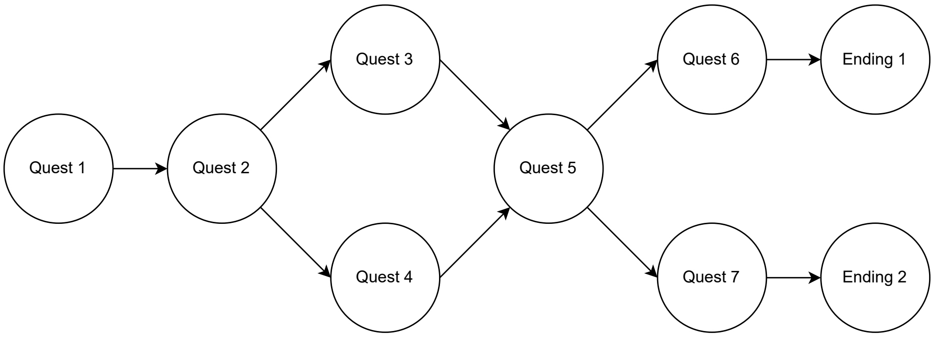


Figure 5.1: A graph of a traditional quest system. Many games may have variations on this basic format, but due to the authoring requirements, most are fairly linear.

5.2 Social Connection

The socially aware NPCs of Barrel Smasher may talk to one another and hold opinions towards one another, which is not a feature normally seen in other RPGs. Certain RPGs may have surface level dialogue between NPCs that may appear to represent some sort of social connection, but relatively few have any internal game mechanics for representing trust or something similar (thus lacking requirement 2 of our social awareness definition). Even fewer RPGs have social connections that expand beyond one-on-one interactions (requirement 4 of social awareness). Thus Barrel Smasher's socially aware gossip system provides a level of connection between individuals of in-game societies not usually seen in other RPGs.

5.3 RPG and Social Integration

Although it adds several social features for research purposes, Barrel Smasher maintains many traditional staples of the RPG genre enjoyed by video game players. Combat, spells, items, and more are all included. And since story elements are not enforced, the player can choose what part of the game they like best. They may decide to play as a hack and slash

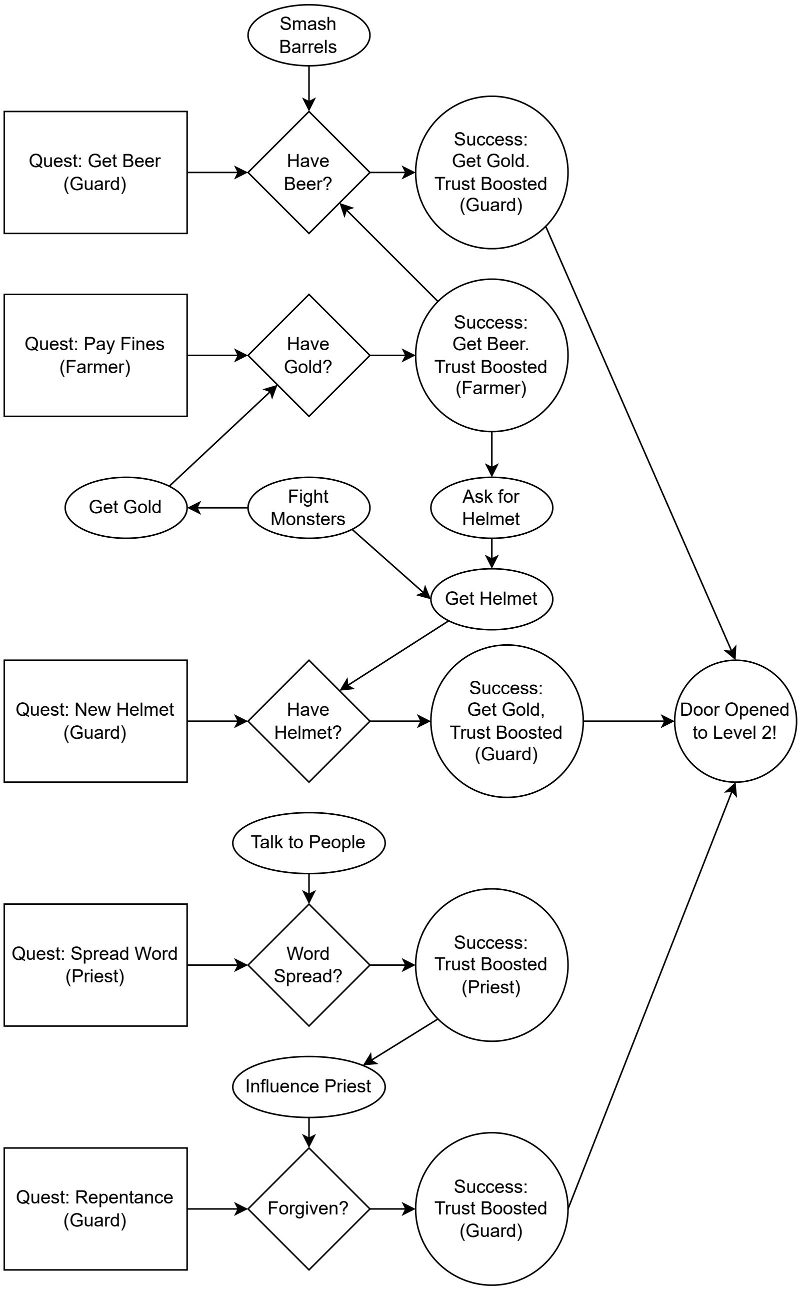

Smash Barrels
Quest: Get Beer (Guard)
Have Beer?
Success: Get Gold. Trust Boosted (Guard)
Quest: Pay Fines (Farmer)
Have Gold?
Success: Get Beer. Trust Boosted (Farmer)
Get Gold
Fight Monsters
Ask for Helmet
Get Helmet
Quest: New Helmet (Guard)
Have Helmet?
Success: Get Gold, Trust Boosted (Guard)
Door Opened to Level 2!
Talk to People
Quest: Spread Word (Priest)
Word Spread?
Success: Trust Boosted (Priest)
Influence Priest
Quest: Repentance (Guard)
Forgiven?
Success: Trust Boosted (Guard)

Figure 5.2: Story graph for Barrel Smasher. Since each quest is independent from other quests, there are more possible ways of completing the objective than typically seen in a traditional quest system. In level 1, there are 5 major quests implemented. Each quest increases the NPC's trust towards the player by 20. In order to complete the level, the guard must have a trust score of 50 towards the player. So 2-3 guard quests should be completed to finish. Quests from other NPCs can also influence the outcome of guard quests, as can other actions like fighting monsters.

fighter, a well connected socialite, or whatever else they desire.

5.4 Content Limitations

There are several limitations with the current game however, mostly due to the limited scope of this project. There are only a handful of classes, spells, items, monsters, and NPCs. Additionally, each NPC has few quests or dialogue options, and most of the dialogue is generated using a standardized template, without much variation for individual personalities.

5.5 Integration Limitations

Another limitation is that different parts of the game – chatting, gossip, quests, combat – are all only loosely connected. For example, chatting and gossip don't affect most quests, combat doesn't do much to alter trust scores, NPC to NPC gossip is never required, and NPCs never directly participate in combat. These different parts of the game can still affect each other indirectly, but the connections may not be apparent to the player.

Chapter 6

Conclusion and Further Studies

6.1 Conclusion

This study created a social simulation game that was integrated into an RPG setting. It simulated a social environment using trust scores, and also contained combat elements typical to most Role-Playing Games, like combat, items, and quests. The social simulation aspect of the game synthesized with some of the RPG elements, and allowed for a variety of choices for the player. The resulting quest progression system was both connected and unrestrictive, potentially making for a unique experience every time the game was played.

6.2 Further Studies

Some parts of the game are limited in scope, but have a high potential for expansion. Much more content could be added to the in-game world, such as monsters, items, levels, NPCs, quests, dialogue, etc. Additionally, the game could better integrate social and combat elements. For example, chatting and gossip could be required to complete certain quests.

Or more NPC to NPC interactions could be added, like trading rumors, selling each other items, and forming social groups. Likewise, NPC to player interactions could also be added, such as assisting in combat, providing services like healing and combat buffs, or providing secret information. Lastly, more complex types of AI research could be included – Case Based Reasoning could let NPCs learn from past experiences; Facade's system of Drama Management could allow for more lifelike story progression, and Procedural Content Generation could add additional, varied content without having to write it out by hand. Overall, the results of this study showcase the potential for social behaviour in video games, and could make a good baseline for more advanced game developments that integrate RPG elements and NPC AI.

Bibliography

[1] Calvin Ashmore and Michael Nitsche. The quest in a generated world. In *DiGRA conference*. Citeseer, 2007.

[2] Andry Chowanda, Martin Flintham, Peter Blanchfield, and Michel Valstar. Playing with social and emotional game companions. In David Traum, William Swartout, Peter Khooshabeh, Stefan Kopp, Stefan Scherer, and Anton Leuski, editors, *Intelligent Virtual Agents*, pages 85–95. Springer International Publishing, 2016.

[3] Manuel Guimaraes, Pedro Santos, and Arnav Jhala. Cif-ck: An architecture for social npcs in commercial games. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 126–133, 2017.

[4] Erica Jurado, Kirsten Emma Gillam, and Joshua McCoy. Non-player character personality and social connection generation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–7. ACM, 2019.

[5] Boyang Li and Mark O Riedl. An offline planning approach to game plotline adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6, pages 45–50, 2010.

[6] M. Mateas and A. Stern. A behavior language for story-based believable agents. *IEEE Intelligent Systems*, 17(4):39–47, 2002.

[7] Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, 2003.

[8] Michael Mateas and Andrew Stern. Structuring content in the façade interactive drama architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 93–98, 2005.

[9] Josh McCoy, Mike Treanor, Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. Comme il faut: A system for authoring playable social models. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[10] Josh McCoy, Mike Treanor, Ben Samuel, Aaron A. Reed, Noah Wardrip-Fruin, and Michael Mateas. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, page 235–237, New York, NY, USA, 2012. Association for Computing Machinery.

[11] Mark O Riedl, C J Saretto, and R. Michael Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 741–748, 2003.

[12] David L Roberts and Charles L Isbell. A survey and qualitative analysis of recent advances in drama management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):61–75, 2008.

[13] Stephen Slade. Case-based reasoning: A research paradigm. Technical report, Defense Technical Information Center, 1988.

[14] Karel Van den Bosch, Arjen Brandenburgh, Tijmen Joppe Muller, and Annerieke Heuvelink. Characters with personality! In Yukiko Nakano, Michael Neff, Ana Paiva, and Marilyn Walker, editors, *Intelligent Virtual Agents*, volume 7502, pages 426–439. Springer Berlin Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[15] Henrik Warpefelt. *The non-player character exploring the believability of NPC presentation and behavior*. Department of Computer and Systems Sciences, Stockholm University, 2016. OCLC: 1001633909.

[16] Ian Watson. Case-based reasoning is a methodology not a technology. In *Research and Development in Expert Systems XV*. Springer, 1999.

[17] Geogios N. Yannakakis. Game ai revisited. In *Proceedings of the 9th Conference on Computing Frontiers*, CF '12, page 285–292, New York, NY, USA, 2012. Association for Computing Machinery.

Appendix A

Source Code

All source code for this project can be found in the following git hub project:

`https://github.com/morgancartwright2022/Barrel-Smasher`