UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Frederik Raud

A Ludic Dialogue System for Interactive Fiction

Bachelor's Thesis (9 ECTS)

Supervisor: Raimond-Hendrik Tunnel, MSc

Tartu 2021

A Ludic Dialogue System for Interactive Fiction

Abstract:

Video games typically represent social interaction through branching dialogue systems, which have the specific goal of entertaining the player but are significantly less interactive than other systems found in video games. In this thesis, a ludic dialogue system is proposed with the purpose of increasing interactivity through the use of procedures. The ludic dialogue system consists of the movement of dialogue agents through a topic-oriented dialogue space, using pre-written dialogue choices similar to the ones in branching dialogue. The system was used in the prototype use case "Last Call" to prove its viability.

Keywords:

Dialogue system, game AI, branching dialogue, game design, interactive fiction

CERCS:

P170: Computer science, numerical analysis, systems, control

P176: Artificial intelligence

Mänguline dialoogisüsteem interaktiivsele fiktsioonile

Lühikokkuvõte:

Videomängud esindavad suhtlust tüüpiliselt hargnevate dialoogisüsteemide kaudu, mille peamine eesmärk on mängijat lõbustada aga mis on tunduvalt vähem interaktiivsed kui teised videomängudes leiduvad süsteemid. Selles töö pakutakse välja mänguline dialoogisüsteem eesmärgiga suurendada interaktiivsust kasutades protseduure. Mänguline dialoogisüsteem koosneb dialoogiagentide liikumisest läbi dialoogiruumi kasutades eelkirjutatud dialoogivalikuid, mis on sarnased hargnevas dialoogis leitavatele dialoogivalikutele. Süsteemi kasutati rakendatavuse tõestamiseks prototüübis „Last Call".

Võtmesõnad:

Dialoogisüsteem, mängu TI, hargnev dialoog, mängudisain, interaktiivne fiktsioon

CERCS:

P170: Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

P176: Tehisintellekt

Table of Contents

1 Introduction 4

2 Player versus Dialogue 9

2.1 The Standard Approach to Dialogue 10

2.2 Alternative Approaches 14

2.3 Common Patches and Interesting Innovations 15

3 Designing a Ludic Dialogue System 18

3.1 Dialogue Agents 18

3.2 Dialogue Space 20

3.3 Dialogue Mechanics 24

3.4 Architectural Comparison 25

3.5 The Complications 27

3.6 Definition 29

3.7 The Ludic Dialogue System as Software 30

4 Use Case: *Last Call* 33

4.1 Implementing the Stack Structure 34

4.2 Implementing the Attention Module 34

4.3 Turn-Taking Strategy 35

4.4 Authoring Content 37

4.5 Evaluation 38

5 Conclusion 42

References 44

Appendix 47

I. Accompanying Files 47

II. Glossary 47

III. License 50

1 Introduction

Interactive fiction is a type of fiction where the reader participates in the narrative. This thesis is concerned with the interactive fiction of video games, especially role-playing video games, originating from their analogue counterpart, tabletop role-playing game[1]. They featured combat and character-progression but also social interactions between characters. One famous example of a role-playing game is *Dungeons & Dragons*. The emergence of the genre coincided with the emergence of computing and computer games, and the experience of being an adventurer would be translated to the medium many times over, from games likes *Rogue* (1980) to the games of *Gold Box* (1988), a collection of games based on the *Dungeons & Dragons* rulesets. The rules for combat and character-progression were easy to computerize. However, as there were few rules for social interaction, the role-playing video game, and other interactive fiction likewise, required dialogue systems to be built from the ground up.

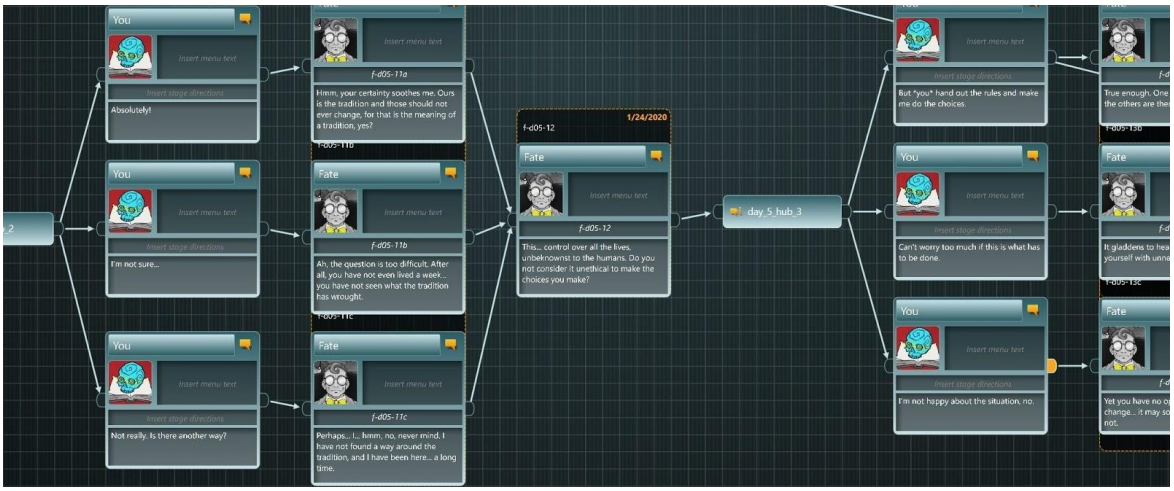


Figure 1. Typical branching dialogue structure in the game *Death and Taxes* designed with the illusion of choice by Leene Künnap [23] and visualized in Articy:Draft [24].

The result was a compromise, a branching dialogue system that was more interactive than reading a novel but less interactive than other game systems. In branching dialogue, the player character (PC) chooses what they say from a list of options, kind of like an actor following a script, and the non-player character (NPC) reacts with a response appropriate to that option, also following a script (see Figure 1). Branching dialogue has become

[1] Both the virtual and the physical variants are called role-playings games or RPGs colloquially. When it is not obvious from the context, the physical variant is called a TTRPG, short for tabletop role-playing game.

ubiquitous with dialogue in video games. The resources dedicated to creating text for them, however, can be tremendous, as exemplified by *Planescape: Torment* (1999)[2], a classic dialogue-heavy role-playing video game. The script for its dialogue is over 800,000 words long [1]. A more recent example would be *Disco Elysium* (2019), with over a million words [2]. To put it into perspective, a novel should be between 80,000 and 89,999 words long, according the magazine *Writer's Digest*[3] [3].

One source for this length discrepancy it that role-playing games take longer to complete than novels. It took me about 30 hours to complete *Planescape: Torment* and 20 hours to complete *Disco Elysium*, about two thirds of my playtime were spent in dialogue. In contrast, at the average reading rate of 260 words per minute [4], it would take less than 6 hours to read through a 90,000 word novel. In addition to raw reading-time, the branching of the dialogue leads to the player being unable to read through it in the course of a single play-through. All in all, it takes longer to create an hour of textual branching content for a role-playing video game than for a novel.

Interactivity through branching is expensive. Dialogue trees vary, but they usually take the form of a directed rooted pseudo-tree[4]. For there to be n unique decision points (nodes) with m unique decisions (outgoing edges), m^n exchanges between the characters would need to be written. In the worst-case scenario, the cost of writing dialogue would explode exponentially. A large part of dialogue design comes down to reducing the number of nodes while maintaining interactivity.

The video game designer Alexander Freed[5] [5] writes in part one of his blog series *Branching Conversation Systems and the Working Writer* on *Gamasutra*[6]:

> *In many cases, this (e. non-interactive dialogue) is sufficient – but interactivity is*

[2] *Planescape: Torment* shares the canon of and has its systems based in the rulesets of *Dungeons & Dragons*.

[3] An American magazine for both amateur and professional writers that has been running since 1920.

[4] Most dialogue trees are not real trees since they make use of cycles. Despite that, they behave like trees for the most part. Referring to their depth is practical. They use cycles only when it would make sense to return to some point in the conversation, often removing the cycles once explored. Players themselves will not repeat the same exchanges ad nauseam either.

[5] Alexander Marsh Freed has written for BioWare franchises such as *Star Wars: The Old Republic*, *Mass Effect*, and *Dragon Age*.

[6] A website featuring news and community-generated posts about game-development.

> *one of the strengths of the medium (e. video games). Thus, the dream of a truly reactive conversation, where a Player can engage in a compelling back-and-forth with non-player-characters, approaching them in different ways according to the Player's whims while still producing witty and dramatic lines appropriate to the situation. We've been trying to do this almost from the beginning as well. We've come up with new tricks over time, but for the most part, games like Mass Effect and The Walking Dead use systems awfully similar to ones pioneered in the 1980s.*

This dream does not pan out in reality. The impression players have is that reading dialogue is as engaging as reading a novel. While the comparison could be seen as a positive comment regarding the work's literary quality, what these players mean is that reading dialogue is as dull as reading a novel. Steam[7] user evilwillhunting [6] in the most helpful negative review of *Disco Elysium* writes:

> *You see, I love CRPGs and adventure games. What I do NOT love is slogging through pages upon page upon pages of text. Absorbing volumes of context and history with nothing going on-screen is not my idea of fun. It's boring. Yes, story is important to a good game, but I tuned out after an hour in and reading endless dialogue trees. It was more work than fun.*
>
> *If I want to read a novel, I'll read a novel.*

This is a far cry from the sense of freedom the role-playing tabletop game and the interactive video game are supposed to give to the player.

There is also the sense among video game designers and players alike that gameplay and story are two different beasts. Dialogue systems are in the story's corner since it is easiest to tell a story through words. Unlike the story, however, the gameplay is integral to the game. As John Carmack[8] [7] puts it, "Story in a game is like a story in a porn movie. It is expected to be there, but it's not that important." An interactive story is even more expensive to create than interactive dialogue. It would require whole new sets and characters to be

[7] Steam offers digital distribution of video games and the system to create communities around them. This includes the users being able to give games positive (*Recommended*) or negative reviews (*Not Recommended*), which other users can vote up (*This review was helpful*) or down (*This review was not helpful*).

[8] John Carmack is the founder of id Software and the lead programmer for games like *Wolfenstein 3D*, *Doom*, and *Quake*.

created. Usually therefore, dialogue systems are linear because the stories they tell are linear.

Nonetheless, the medium of video games cannot abandon "the dream of a truly reactive conversation" just because conversations have been considered unengaging and prohibitively expensive so far. To dismiss this domain of the human experience is to miss an opportunity. It is instead necessary to find new avenues for reducing costs and increasing engagement.

This thesis hypothesizes that one way to do this is to reintegrate gameplay with dialogue. We must utilize ludic[9] elements, meaning elements that allow for the act of play. Game-like rules can generalize social interaction, allowing for the reuse of elements in dialogue. Such rules could also allow for more layers of engagement than the call and response of branching dialogue.

While there have been such attempts to gamify social interaction as seen in Figure 2, they usually do so by doing away with words. Symbols, meaning, and conflict become procedurally generated. Dialogue loses the dimension of written literature and becomes more of a representation of the act. This thesis maintains that dialogue can still be proceduralized without this reduction in resolution.

Figure 2. *Griftlands* (2020) uses card-game mechanics to simulate social conflicts, abstracting speech in the process.

[9] Meaning related to play (*ludum*).

Another approach to procedurally creating dialogue is to use autonomous AI for text-generation, which this thesis also rejects. As Chris Crawford[10] [8] writes in a farewell letter to the development of interactive storytelling:

You will NEVER see anything like literature coming out of deep learning AI.

The aim is not to build a new system from the ground up but to utilize the groundwork that previous dialogue systems have already built for interactive dialogue. The second chapter will introduce the reader to the standard branching dialogue system. I will highlight the related design decisions and analyse their results. In the third chapter, I will use the lessons learnt in the previous chapter to design a new dialogue system with the properties of play. In the fourth chapter, I will apply this system to a game development use case and produce a prototype[11], which I will use to see how players respond to the system.

[10] Chris Crawford is the founder of the Game Developers Conference and a long-time champion of interactive storytelling.

[11] The link to the project page can be found on https://raud0.github.io/A-Ludic-Dialogue-System/.

2 Player versus Dialogue

In the introductory chapter, I set out two areas where dialogue systems required improvement: cost efficiency and engagement. I claimed that utilizing ludic elements could help in both of these areas. As the first step in that direction, let us analyse how both dialogue and non-dialogue systems benefit from having ludic elements.

Basing his claim in *Dark Souls* (2011), Daniel Stamatov [9] states that violence in video games is "a potent mode of engaging with fiction". In *Dark Souls*, every material action on the part of the player has an immediate reaction within the reality of the video game. One common challenge in the game is to outmanoeuvre enemies, where avoiding the enemy's attack is rewarded with a chance to counterattack. Should the player manoeuvre themselves behind the enemies back, they get to smash a button to perform a backstab, which can be crucial to defeating the enemy. The player gets intense and immediate feedback of the brutality of this act of violence, conveyed through animation and sound.

As a reward, defeated enemies leave behind a currency called *souls*. These the player can lose as a result of repeatedly failing to avoid the enemy's attacks. This depletes the player's health bar[12] and once it is fully depleted, the player dies, losing all of their *souls*. Should they keep them for long enough, the player will get to opportunity to use their *souls* to strengthen their character. It serves as a direct representation of growth in power and rise in tension through conflict. The ludic elements complement the fantasy of being a duellist.

According to Marc LeBlanc [10], the aesthetics that make games dramatic (and thus fun) are **uncertainty** and **inevitability**. In the previous example of *Dark Souls*, we see both. It is uncertain whether the player will win or lose any given combat encounter. Simultaneously, it is inevitable that any given combat encounter will end as both combatants are spending their limited resources to stay in the fight. As the player continues to win and grow in power, previously tougher enemies will inevitably stop posing as a threat to them. For these aesthetics to have any meaning however, the player needs to have agency within the game.

Bettina Bodi and Jan-Noël Thon [11], in an attempt to explain how games like

[12] Health is a common resource found in video games that is visually represented in the form of a bar getting smaller as the player approaches the loss state of losing all their health.

Disco Elysium and *Astroneer* give players agency without the use of combat systems, distinguish between four different types of player agency in games:

- ***Spatial-explorative agency***. The player's ability to traverse a space with their avatar. Most games have this in the form of traversing a physical space.
- ***Temporal-ergodic agency***. The player's ability to explore the game as a temporal system through their avatar. Present in most games where the order of actions matters.
- ***Configurative-constructive agency***. The player's ability to configure their avatar and construct the game world. This one is a mainstay of role-playing games.
- ***Narrative-dramatic agency***. The player's ability to participate in the game's story through their avatar.

Even if the player spends most of their time with *Disco Elysium* in dialogue, it is not seen as an interactive story but as a game. It won 4 awards in The Game Awards 2019, one of them being the award for Best Role-Playing Game. Bettina Body and Jan-Noël Thon [11] credit *configurative-constructive agency* and *narrative-dramatic agency* specifically for *Disco Elysium*'s success as a game. Despite innovations on those fronts, its approach to dialogue is still standard.

2.1 The Standard Approach to Dialogue

Freed describes the standard approach to dialogue in part two of his blog series on *Gamasutra* [12]. Freed starts by defining the two foundational structures: *hub and spoke*, and *waterfall* (see Figure 3). Most games use a blend of both side-by-side, but both have their trappings that prevent the dialogue from being engaging.

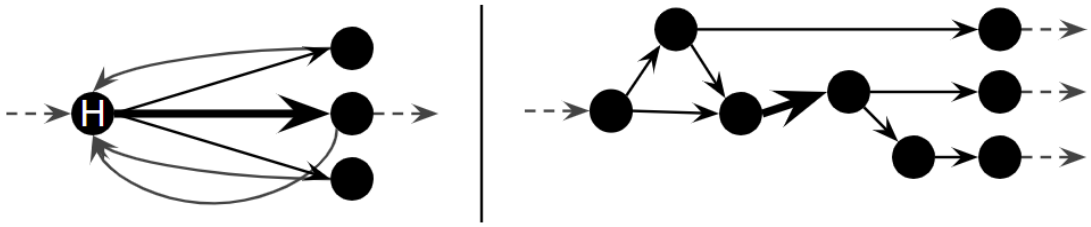


Figure 3. Example of the hub and spoke (left) and the waterfall (right) structure. The dialogue exchanges are the player's options, with the bold arrows representing the critical path. The circle with the 'H' is the hub. The grey dashed lines show the connections to the rest of the dialogue.

Hub and spoke gives the player several dialogue options, or spokes. The player chooses one, gets a reply, and then returns to the hub, which presents the same dialogue options again. As Freed [12] describes it, "At their worst, hubs turn NPCs[13] into information vending machines where the player feels obligated to punch every button and get each available tidbit of data."

Waterfall is similar to *hub and spoke* in that it offers choices, but it does not feature hubs. In essence, when the player chooses an option, they can never return to that point. There is no backtracking. Here Freed brings up the danger of "key information" getting lost in unchosen dialogue branches.

Due to the ability to backtrack, *hub and spoke* is less dramatic than *waterfall*. At worst, backtracking blocks both uncertainty and inevitability. Should the player repeatedly come to the same choice with no state change, the act of choosing becomes irrelevant. The act of looping through a conversation without consequence gives the impression that it never needs to end, ruining the aesthetic of inevitability. Compare this to the combat of *Dark Souls*, where every action has a cost. Some games do reduce backtracking by limiting the amount of times the spokes can be chosen.

While *waterfall* is more dramatic than *hub and spoke*, it comes with its own problems. One of these is the implicit mandate for linearity. Because one line of dialogue must also follow from another, *waterfall* also favours encapsulation of topics. One topic must end before another can begin, and dialogue must pass single-mindedly, allowing for little side-tracking. The inability to reuse lines also means that the writer has to do more work for the same amount of content seen by the player.

There are also more subtle design decisions to dialogue. Freed [12] brings up some himself, but others are so ubiquitously solved that they do not seem like a decision at all. The three of interest from the ones he does mention are timers, interrupts, and mechanical aspects. They are all ludic elements, requiring some degree of interactivity, and should increase the player's agency over the dialogue.

Both timers and interrupts force the player to react to the NPC, much like the player in

[13] NPC - Non-playable character. These are the characters that inhabit the game world but which the player cannot control. PCs (playable character), on the other hand, the player can control.

Dark Souls would have to react to an attack by side-stepping, rolling, blocking, or parrying. This is *temporal-ergodic agency*. Ideally, timers create a sense of urgency and add drama through the aesthetic of inevitability. If the player fails to answer the NPC in time or interrupts the NPC's line, the game can react to this by either going down a special branch that was not listed in the options or pick one of the branches automatically.

Timers and interrupts are both undermined by the design decisions inherent to both *waterfall* and *hub and spokes*. Having to hand-craft a new branch for them at every node, even if it does not make sense, is too much work. Alternatively, they can be used selectively, an example of which can be seen in Figure 4. Freed [12] mentions how in practice interripts varied from being amusing gimmicks to immersive conversation elements. As a constant dialogue element, they are resource-expensive, but as an occasional element, they are gimmicks.

Figure 4. Interrupts in *Mass Effect 2* (2011). The player has to notice the interrupt symbol on the screen and react in time by pressing a button.

Mechanical aspects are cheaper, as they build on already existing game systems, typically character-building or resource management. This is *configurative-constructive agency*, the most common way for role-playing games to patch player engagement into dialogue. Still, they are not free additions, as the game systems themselves were not free.

Freed [12] also discusses the importance of deciding on the number of dialogue options available at any given time and how to present them. For games with a focus on dialogue, a set of 3-4 options is the standard. More options allow for more nuance but also increase the cost of dialogue. Games going for a more cinematic experience want to keep the scene moving, so spending a long time reading will not be optimal. In general, the incentive is to keep the number of options low. On the other hand, Freed notes that the player should never have to pick a dialogue option that "contradicts their image of the character", undercutting *configurative-constructive agency*. For ease of use, the dialogue options more relevant to the player's image of the character should appear first.

Freed then expands on the idea of *forced player lines*. The branching dialogue system requires the designer to ask how much control the player has over their character. Should the player character only speak when the player makes them speak, or should they respond to some lines automatically to allow the conversation to flow better? In one case, the player has no choice, in the other, the writer contrives a meaningless choice into existence for the sake of consistency.

Here we pass into the land of hidden design decisions that make up the conventional branching conversation in video games. There are also exceptions which I will summarize in the last part of this chapter.

The first of these is as common as it is unnatural. Video game dialogue tends to have a neat back and forth rhythm. The player selects an option, the NPC responds, and the player gets new options. A strict AB pattern emerges where the player is in complete control. The player chooses the entire exchange, both the PC line and the given NPC response. In other words, the NPC has no agency and is purely reactive. It is natural for dialogue to be structured like this since decision points always occur on the player's end, as seen in Figure 5. However, dramatically it is unnatural for one character to be in complete control of the story. In combat systems, the characters act in parallel as separate agents, both able to exert influence over the other.

Figure 5. An example of *hub-and-spoke* dialogue by Alexander Freed [25]. The PC lines are marked by blue and the NPC lines by red. The presumption of AB structure is evident in the fact that the software requires empty PC lines to be added in-between NPC lines, should the NPC say multiple things in a row.

A related design decision is whether entering a conversation switches the game mode. In most games, the game world pauses during dialogue. The player has to leave the conversation before being able to move or attack again. The action in combat systems happens in real-time, and the player acts spontaneously by default. Real-time dialogue

systems seem to be on the rise with newer role-playing games like *Fallout 4* (2015) and *Cyberpunk 2077* (2020). These games restrict player freedom in dialogue to a lesser degree, leaving them with control over the movement and the camera. Using *waterfall* in one's dialogue system complicates real-time dialogue and problems compound with the presumption of a *critical path*, the central conversation the player cannot miss [12]. If the dialogue is interrupted, the dialogue system needs to return the player to this path, but *waterfall* was never meant to be traversed more than once.

There is also a design decision regarding what sort of a purpose the dialogue serves in the game. Linguistics has two main views of the function of language. The *transactional view* places importance on the informational component of dialogue. Games that separate story from gameplay take this approach to dialogue as dialogue is solely responsible for informing the player of what is happening. The player may begin to dread dialogue as sections that they have no control over. The other view is the *interactional view*, which emphasizes forming social relationships through conversation [13].

While "laying the plumbing" is essential, it does not make for great drama. Interesting dialogues should have conflict, which is why a game that wants to use dialogue as a core mechanic must look towards the interactional view to be engaging. The performance of roles, solidarity and mood becomes just as important as conveying new information.

2.2 Alternative Approaches

Freed [12] also brings up alternatives to branching dialogue systems. The first, *non-interactive dialogue* allows for no interaction. Since there is no interaction, there is no way to add ludic elements to make it more engaging. The *simple choice* system for dialogue builds on that, giving the player one choice, which summarizes the PC's general approach to dialogue but does not give the player direct control over it. This keeps branching to a minimum and keeps the writer in control of the flow of the dialogue. Lastly, the *simple hub* dialogue system is similar to *hub and spoke*, but always returns the player to the same hub.

The experimental interactive storytelling game *Façade* (2005) uses an expanded implementation of the idea of a *simple hub*. A set amount of global *discourse acts* are available to the player from the start, but the characters' reactions depend on the current story *beat*. *Beats* are a way to give dramatic structure to the story without strictly branching

paths. The dramatic engine chooses these *beats* based on their ability to fit into the game's "Aristotelian" tension arc. The beats themselves are scripted sequences of *discourse acts*, and the player has access to around 30-40 of these at any given moment. They are picked unwittingly through textual input, as the use of a NLP[14] layer obfuscates the matching process [14].

Lastly, there is an interesting approach to dialogue that is interactive without branching. The player only has one input ("talk!") which the game interprets as participation, compliance, interest or love. *Dark Souls* is one example of this. In *Dark Souls*, the player does not have to interact with the NPCs and can stop whenever. The player has control over the dialogue through *narrative-dramatic agency*. Therefore, the fundamental player decision in any dialogue system is "talk!" or "do no talk!". In contrast, many branching dialogue systems lock the player into interacting, by blocking progress in game until a terminal node in the dialogue tree is reached.

2.3 Common Patches and Interesting Innovations

The addition of emotive audio-visual effects is the most common way to add drama to dialogue. Voiceover is a selling point for indie games like *Disco Elysium*, but it is a feature the consumer has come to expect from role-playing games in the last two decades. The original first two Black Isle Studios *Fallout* games (1997 and 1998) are only partially voiced, while the newer games, *Fallout 3* (2008), *Fallout: New Vegas* (2010), *Fallout 4* (2015), are fully voiced. Figure 6 displays the dialogue UI of the first two games, where voiceover was limited to the important characters, who were also given a "talking head".

Figure 6. "The talking heads" of *Fallout*.

[14] NLP is an abbreviation for *natural language processing*.

The same trend is apparent in other franchises like *Baldur's Gate* and *Divinity*. The trend is interesting because it displays an attempt to improve engagement with dialogue without innovating the dialogue system itself.

Voiceover increases the resource cost of a single line of dialogue [12], which is another incentive to keep branching as low as possible. There are several methods for reducing branching, which Freed [15,16] describes in parts four and five of his blog series. The primary method is to simply write dialogue with fewer decision points. Focusing efforts on improving the critical path and the few choices on it is cost-effective [12,16]. Alternatively, the player could have more dialogue choices with less immediate impact.

While both techniques reduce the m^n problem of the exponentially expanding dialogue trees, they also endanger engagement. Freed [15] recommends that there should be fewer choices with more impact. By impact, he means control over the story, or *narrative-dramatic agency*. Games like *Fallout* (1997), *Fallout: New Vegas* (2010), and *Tyranny* (2016) enable this by offering the player a chance to talk down the final boss instead of fighting them. *Tyranny* and *Fallout: New Vegas* also creates actual branches in the game's narrative, which the player selects through choices in dialogue. Lastly, there is *configurative-constructive agency*, which emerges from the combination of character creation and dialogue choices.

Figure 7. Dialogue UI in *Disco Elysium*.

Disco Elysium utilizes all of these methods. It offers dramatic dialogue choices that display the character's political beliefs and personality, and change the narrative outcomes of events

[11]. *Disco Elysium* complements this by making context and the exploration of both the dialogue tree and the game world relevant to the advancement through the dialogue. Finding some item, doing something at the right time, or saying something can grant penalties or bonuses to dialogue checks, which are shown to the player, as seen in Figure 7. These checks determine how the PC expresses himself and what route the story takes.

However, with all of the patches mentioned above, a designer needs to ask themselves whether it is worth adding external engagement to an otherwise unengaging system. In *Dark Souls*, the combat system functions without *souls*. *Disco Elysium*'s narrative branching and skill economy are so core to the dialogue system that their impairment would also impair the dialogue.

3 Designing a Ludic Dialogue System

Dominic Arsenault [17] compares the interactive fiction of gamebooks[15] to the interactive fiction of video games:

> *Barred a few rare exceptions, video games, on the other hand, do not function on the basis of choice but according to a "repertoire of action". According to Janet Murray, digital environments are procedural: "the computer is not fundamentally a wire or a pathway but an engine. It was designed not to carry static information [that is, a tree structure of choices with pre-defined consequences] but to embody complex, contingent behaviors."*

Uplifting dialogue systems from a simple choice to a repertoire of action allows dialogue to take centre-stage as the gameplay itself. In other words, they become ludic systems.

3.1 Dialogue Agents

To enable ludic behaviour within dialogue through a repertoire of action, we must first get rid of the strict AB structure and script-like nature of dialogue described in the previous chapter. Instead, the participants in the dialogue should be able to speak out of order. This gives us two requirements. First, the system must proceed in real time, and second, the participants must act as separate agents, both having their own decision-making. Both elements can be found in most action games, including the aforementioned *Dark Souls*.

Placing games in real-time is common and this is accomplished using a software design pattern called a game loop. In *Game Programming Patterns*, Bob Nystrom[16] [18] writes about the pattern:

> *Game loops are the quintessential example of a "game programming pattern". Almost every game has one, no two are exactly alike, and relatively few programs outside of games use them.*

[15] Books where the readers influence the story by making choices, typically by following page references. Two well-known gamebook series are *Choose Your Own Adventure* and *Fighting Fantasy*. The former is also used as a general term for gamebook-like fiction.

[16] Bob Nystrom worked as a game developer at Electronic Arts, an American video game company, for 8 years.

Actions taking time is the first element we are borrowing from the domain of video games. The game does not wait for player input, which is the opposite of how standard branching dialogue systems work. Building on this, we can begin to use time as a resource and a basis for new game mechanics.

Given that speech acts will take time, it will be possible for the player to react to them. We can use this as a basis for a new core dialogue concept: interruption of speech acts. This behaves just as avoiding or countering attacks in *Dark Souls*. In a conversation, interruption of a speech act can signal both cooperation and non-cooperation. You do not always want the other person to finish what they are saying. Unlike in *Mass Effect*, interrupts are the default.

Removing the ordered AB structure also allows for more varieties of turn-taking, which can also express either cooperation or non-cooperation. The linguists Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson [19] describe how to take turns in a cooperative environment:

1. The current speaker selects the next speaker.
2. If they refuse or the speaker selects no one, everyone gets the opportunity to self-select.
3. If no one self-selects, the current speaker continues or the conversation ends.

Following this structure gives conversations a more natural rhythm. There can be lulls where no one is speaking, and there can be heated moments where everyone is having trouble getting a chance to speak their mind. It also removes the constraint that every line of dialogue should follow logically from the one before. This will be a negative if the writer wants direct control over the dynamics of the conversation.

For turn-taking to work, both the NPC and the PC need to be able to take turns. Since a human controls the PC, they do not have to be told when to speak, they have the social intelligence to do so. The NPC, on the other hand, requires social etiquette and social strategies to be expressed by rules. A possible solution would be to assign each line of dialogue a value that tells the NPC whether the current speaker should continue, wait for a response, or neither. In the same article, Sacks, et al. [19] also make the point that the speaker may even expect interruptions at times. These types of interruption can be used to signal agreement, disagreement, confusion, and other emotions.

By now, we have created several requirements for the NPC.

- It needs to act according to social etiquette like cooperation, detecting whether others are observing etiquette and observe etiquette itself.
- It needs to decide when to speak.
- It needs to decide what to speak.

Considering these constraints, it would make sense to also give the NPC something analogous to emotional calculation for responding to social *faux pas*. A tit-for-tat logic could accomplish this easily: past cooperation begets cooperation, past non-cooperation begets non-cooperation. The NPC could have other emotions as well, like affinity for the player. The more the NPC likes the PC, the more accepting they are of the PC's transgressions.

The NPC's emotional state directly influences the course of the conversation. Managing an NPC's emotional state should be one of the ludic elements this dialogue system features.

3.2 Dialogue Space

At this point, the ludic dialogue system has been rendered structureless by removing the AB structure. There is no longer a graph along which the dialogue should move. To begin solving this problem, let us consider dialogue as taking place within an undefined dialogue space. The actors move through this space by speaking. In our structureless dialogue world, every possible thing to say is available from any point in that space.

From here, if we limited each dialogue option to only be selectable from a single point, and revealed unique dialogue options at that point, we would be back to the graph structure of standard branching dialogue. It does not make use of the spatiality of a conversation. Moreover, we do not even know what this space should look like; we only know that we should be able to move through it over time.

To solve the problem, we can look to real conversations, a process which begins with Wallace Chafe[17] [20], who in *Discourse, Consciousness, and Time* tries to describe the flow

[17] Wallace Chafe was a linguist who worked with Indian languages and valued observing language in the wild. He saw himself as one forced to work on the periphery of linguistics due to the success of Chomsky's generative revolution, which he deemed to fail in describing semantics. Chafe also believed that language and the mind have a strong relation to one another and that it is impossible to study one without the other. He

of language and consciousness over time, specifically in conversation and storytelling. He makes several preliminary statements about consciousness. For our dialogue system, the following are relevant:

- *Consciousness Has a Focus*
- *The Focus Is Embedded in a Surrounding Area of Peripheral Consciousness*
- *Consciousness Is Dynamic*

In other contexts and for the purposes of this thesis, the phenomenon of a "surrounding area of peripheral consciousness" is called attention. As the conversers' attention and focus shift, their object of discussion changes.

When we first considered conversation as a space, it was hard to describe. After all, what does movement through such a space mean? The answer is that movement through a conversation has an element of focus-switching. While focus provides us with a unit of distance, we still need axes along which to move.

Wallace Chafe describes attention as being centred on topics. Much like a single speech act, a topic is a division of conversation, a frame of reference [20]. We can imagine each topic as a single axis, which allows the conversation to be at a single point in conversation space at any given moment in time (see Figure 8). This location vector can be normalized, meaning that the sum of these values adds up to one. This would reflect attention being a limited resource, there is only so much of it to give. Because movement through this space represent change in focus, let us call this space *focus space*.

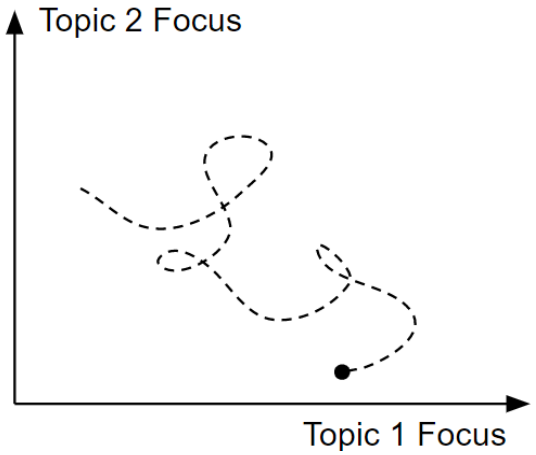


Figure 8. **Focus space**. The dashed line represents the movement of a hypothetical conversation through a two-topic space.

We can break this space down further and assign each dialogue option an area that it appears

hoped that seeing language as a *temporal phenomenon* would become fundamental to linguistics [27].

in. Dialogue options once again appear in a structured way, only the structure is not the order of appearance but the structure of association. In this thesis, we will consider each dialogue option to only belong to a single topic.

Wallace Chafe breaks topics down further, describing both progression and hierarchy. To establish hierarchy, he defines three types of topic: *subtopics*, *basic-level topics*, and *supertopics*. Conversations consist of *basic-level topics*, with *subtopics* adding granularity to them. *Supertopics* are more like conversational poles around which *basic-level topics* clump up. The equivalent of the critical path mentioned by Freed would be something Chafe calls a *central topic*.

While conversations have an element of focus-switching, they also have an element of progression. Chafe's progression starts with the topic entering into the speaker's attention, then gaining and losing focus, and then ending with the topic exiting the speaker's attention. The following list describes the general progression of Chafe's topics:

1. **Orientation**. The speakers start the discussion of the topic, the speakers set the stage by defining a state of normalcy and establishing the conflict. The topic enters attention.
2. **Complication**. The speakers remove themselves from normalcy and move from orientation to climax. Issues related to the topic are brought up and resolved. The discussion can further break down into subtopics. The discussion removes from normalcy.
3. **Climax**. The speakers resolve the conflict of the topic, arriving at a revelation or a similar meaningful change. The discussion is at its furthest from normalcy.
4. **Denouement**. The speakers return to normalcy. If the conflict has been resolved, there is nothing more to say. The topic begins to leave the attention.
5. **Coda**. The speakers step back from the discussion of the topic and reflect on it from the point of normalcy.

The main takeaway here is that different **stages** of a topic's lifespan have different functions. The distance from normalcy we will call **depth**. Both give the topic something similar to a dramatic structure, but they also reflect the idea that a conversation cannot endlessly go around in circles but has to reach some logical conclusion eventually, at which point there is no reason to complicate things further. Therefore, a topic has a beginning (orientation), middle (complication), and a naturally occurring end (climax, denouement).

How to interpret the progression through topic space is up to the given designers of a game. Depth is the unit of movement here, and the stages are the states of progression that make up the discussion of a topic. Since going deeper into a topic involves focusing on the topic, it follows that depth and focus are similar concepts and could be merged. However, using these same stages in combination with this exact focus arc might not be necessary. Let us call this space *depth space* (see Figure 9).

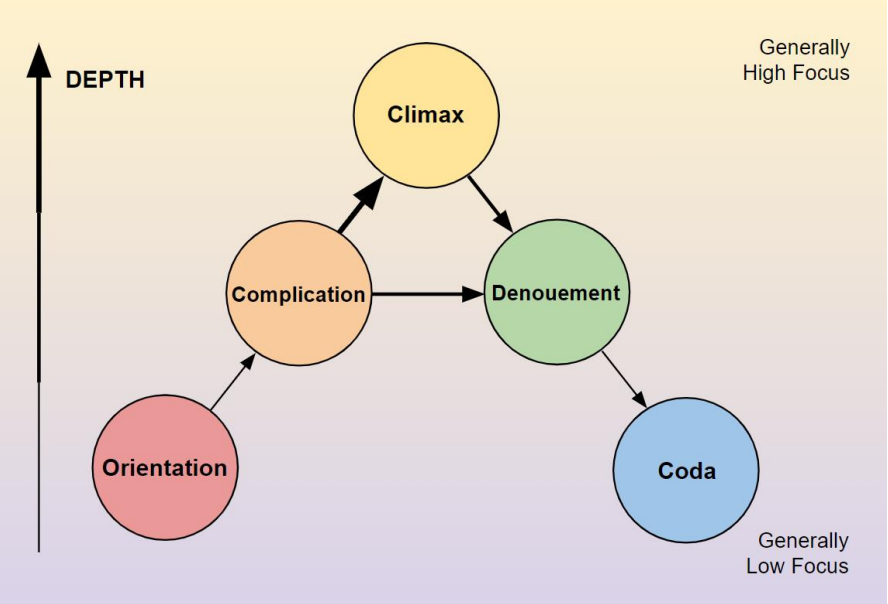


Figure 9. **Depth space**. A higher depth usually corresponds to a higher focus, and higher level stage, require higher depths to access. Not all the stages need to be traversed and a topic can disappear at any stage if it falls out of focus.

Finally, we also need to move through these spaces. We will consider the traversal component of each dialogue option as some function, which considers the current position in the conversation (both in *depth space* and *focus space*) and outputs a new position. Dialogue options should also have an availability component, which considers whether the option is available from the current position in the conversation. As we are talking about dialogue, let us think of these options as thoughts, which have the potential of being actualized as speech.

Our final elements of our dialogue are the following:

- **Agents** (and game-related context): the agents between whom the conversation takes place and the context of the game world they exist in.

- **Focus space**: n-dimensional space with each axis representing the focus given to a topic.
- **Depth space**: a directed progression graph laid over a 1-dimensional space representing the depth of discussion regarding a topic.
- **Stages**: areas within a topic space that behave differently.
- **Thoughts**: formerly dialogue options. Display text, move the position in *dialogue space* and are only available from a select area in *dialogue space*.

Defining the conversation through a *dialogue space* allows movement through this space which is not discretely defined as movement along the edges of a graph, where the conversation has to start at some point and end at some point. Very few conversations start or end formally. Such constraints are still possible but no longer mandatory.

The fundamentality of interruptions to social interactions also becomes apparent. One dialogue agent moves the conversation in a direction that the other does not want it to go in, so they interrupt their movement and move the conversation in another direction instead.

Lastly, the *dialogue space* is not a physical space that the agents share. Instead, it exists entirely in the "consciousness" of the dialogue agent. While all agents could have the same perception of this space, considering them as separate realities can be the basis of unique gameplay mechanics and nuanced interaction. It also makes more sense when considering the PC as the player's avatar in the conversation. Regardless of the actual position in *dialogue space*, the player has their own illusory perception of it.

The interplay of the complexities allow for the natural complexities of human interaction to emerge as gameplay challenges. For example, the player thought they were moving the conversation in one direction, however the non-playable character was moving in a completely different direction. This is one of many dialogue mechanics that this system allows for.

3.3 Dialogue Mechanics

With the addition of space and agents that move within that space over some time, the game maker's ability to implement procedures for conversation has significantly increased. There is more surface here than within a graph-based system. Considering this, let us return to our original intention of creating a ludic dialogue system. We should start with establishing the gameplay mechanics that create the desired aesthetics of uncertainty and inevitability.

The primary way this system would express inevitability would be to make each thought one use only. It can still appear in several areas, but the agent can only pick it once. As the agents move through the conversation, the thoughts are used up one-by-one, and some areas of the *dialogue space* become inaccessible due to the depletion of movements. Unless the direction of movement is known, it is also unknowable where the conversation may end up.

To add to the sense of inevitability, the player can be put under time pressure by automatically decaying the depth of topics over time, especially when they focus on another topic. Each action has a cost and it is uncertain whether the player will ever return to the same topic again to finish it.

We could also give extra mechanics to stages. Perhaps the climax or coda stage should be unexitable until both characters say something climactic or conclusive.

Dialogue options from one topic can give focus and depth to other topics, making them available, but there also needs to be a way to provide the player with some starting topics. An obvious choice is to have the player start with the *central topic*, but a more naturalistic approach would be to draw topics from the environment, uniting the conversation with the game world and giving the player *spatial-explorative agency*. Click on a cup of coffee to talk about coffee, click on the NPC's face to talk about their emotions, shoot a gun to talk about violence, move away to talk about leaving.

3.4 Architectural Comparison

So far, we have only talked about how the player chooses dialogue options. For the system to be complete, we also need to consider how a computer-controlled agent would choose dialogue options. Let us start by drawing on architecture for dialogue systems outside of games. These fall into two categories:

- **Task-oriented systems**. The emphasis is on fulfilling a task, like answering a user's question. Corresponds to the *transactional view* of conversations being about sharing new information.
- **Non-task oriented systems**. The emphasis is on interacting with the user, like a chatbot, instead of answering their question. Corresponds to the *interactional view* of conversation being about forming social relationships.

These systems utilize *generative methods* and *retrieval-based methods* for finding a response to the user's input [21]. The *generative* agent has to generate its answers, so using them for task-oriented systems requires giving them a knowledge base for answering questions. The *retrieval-based* agent knows all possible answers, but it needs to match them to the user input. It cannot handle unpredictable input like the *generative* agent could.

For *retrieval-based* conversational agents a response is found by calculating a score for all possible responses and finding the one with the highest score. The function for finding this score is known as the matching function. The standard approach can only ever match one response to any input [21].

Our system will still be non-task oriented, as the focus is on the interaction between agents, not telling a story. It will still mainly be *retrieval-based*, as the NPC reads from a script. We need to match the current position in the conversation space to the NPC's affinity to a possible response. Unlike a standard branching dialogue, our system will also feature *generative* elements. We do not know what order the dialogue will come in, so we have to generate an order. We do not know the current focus of the dialogue, so we have to generate a focus. The NPC might "like" a thought so much that they are inclined to express it despite social etiquette. Their emotional state (position in game space) might be so extreme that they might say something they otherwise would not. In sum, we generate the context and retrieve an answer.

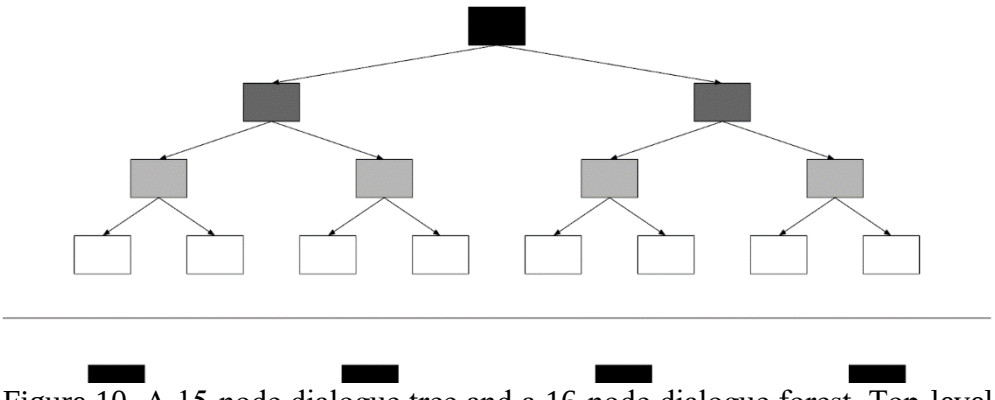


Figure 10. A 15-node dialogue tree and a 16-node dialogue forest. Top-level topics are black, lower level topics are shades of grey, stubs are white, arrows are dialogue options. The arrows signify the decisions revealed when reaching a node. For branching dialogue, once the player choose a dialogue option, the player moves to a new node and gets a new set of dialogue options. For ludic dialogue, the player is never at any single node.

Let us also consider the cost of creating content for this system. If the branching dialogue is similar to a tree, the topic-oriented dialogue is similar to a forest. Given k basic-level topics, n levels of topics, and m things to say about each topic, the number of nodes this forest would have would approach km^n. In this system, costs would be cut by having a breadth of topics instead of a depth of topics, assuming a fixed number of edges (m). For a visual comparison, see Figure 10.

3.5 The Complications

The system creates new problems and removes old ones. It is clearly more elaborate and breaks some of the functionality of standard branching dialogue. Being a simulation of real conversations, it more naturalistic than theatrical in its structure when compared to branching dialogue. As such it should not be considered an absolute improvement but a possible alternative with a focus on player engagement.

The AB structure might also be more natural to a script writer. Even if AB structure was removed, it will still occur naturally, so if unwittingly used, the system becomes branching dialogue with extra steps.

The technical overhead is also something to consider. There are more components here that need to move in unison for the system to work. This system will be harder to test and visualize as a whole since the positions it could be in will be harder to predict, more so when using discrete values that are dependent on time. If the player has goals in the dialogue, the goals might become unreachable from early on, *softlocking*[18] the player.

Overall, the conversation itself will become less predictable as well as less directable. There is no guarantee that the player remains in a state which makes narrative and dramatic sense. To create the illusion of a conversation, we rely on the player's ability to decode a series of symbols in procedurally generated order. A computerized drama director like the one in *Façade* could be added to increase directability, but this has problems of its own.

At the start of the development of *Façade*, the player was intended to be the protagonist of the game's story, not simply an observer. But by the end of development, the player was

[18] A *softlock* is a game state where the player may act but may not progress by doing so, often under the impression that they could.

precisely that, an observer. The NPCs could be relied upon to produce drama, but the player could not [14,22]. For a controlling writer, a programmable actor that does whatever the writer wants them to, is much more appealing than a player who can deflate any story by not playing their role (see Figure 11). Branching dialogue systems likewise use their structure to force their players into acting dramatically by not continuing until they do so. This is in effect a variation of the *critical path*.

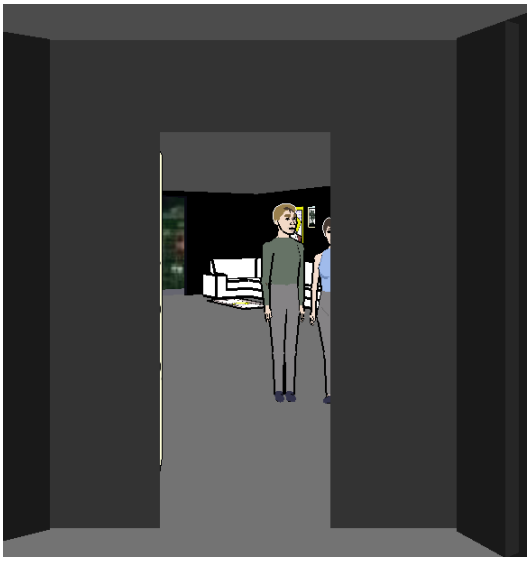


Figure 11. An example of a broken state, narratively speaking, in *Façade*, which can be achieved by doing nothing. The player is expected to knock on the door themselves, but if they do not, one of the NPCs will open the door for them, saying that they "thought they heard somebody moving". The NPCs proceed stare at the PC with a blank expression with a few lines of dialogue in-between. Eventually the AI will think that the player is trying to leave, even though they have never entered the apartment.

Ludic (dialogue) systems require that the writers and the designers make the player central to the unfolding of the conflict. Generally, this is nothing new. Combat and survival systems have long been able to force players into conflict and react accordingly otherwise. A ludic dialogue system would deal with the issue of a *critical path* the same way they would. Games already do so by defining the conflict loosely or funnelling the player down a path with a cutscene. At the end of the day, most people play games because they want to participate in them, and any game can be broken by simply turning it on and then staring at the screen in blank silence.

3.6 Definition

The ludic dialogue system is a tuple $(\boldsymbol{C}, \boldsymbol{A})$, where a conversation $\boldsymbol{C}$ takes place within some context $\boldsymbol{A}$.

The context $\boldsymbol{A}$ should at least contain the set of all dialogue agents $\{\boldsymbol{a_i}\}$. These agents act in real-time, which may either be shared $(\boldsymbol{t})$ or individual $(\boldsymbol{t_i})$. Context can also include elements from the general context of the game which the conversation affects.

A conversation can be seen as a tuple $(\boldsymbol{F}, \boldsymbol{D}, \boldsymbol{T}, \boldsymbol{O})$, where every topic from topic set $\boldsymbol{T}$ has a focus axis in *focus space* $\boldsymbol{F}$ and *depth space* $\boldsymbol{D}$. $\boldsymbol{O}$ is a set of all dialogue options. Each dialogue option is available only from a limited area of space. The conversation space is either hypothetical and individual to each agent, or real and shared by all agents.

- If the conversation space is hypothetical, then every agent $\boldsymbol{a}$ has an individual set of modules for perceiving the *focus space*, the *depth space*, and the set of dialogue options available from the combination of these spaces.
- Avery dialogue agent $\boldsymbol{a}$ also has an individual set of modules for decision-making and tracking state.
- Agents use *thoughts* (dialogue options) to create speech. They have an audio-visual component, and a functional one, which changes the agents' positions in *dialogue space*.

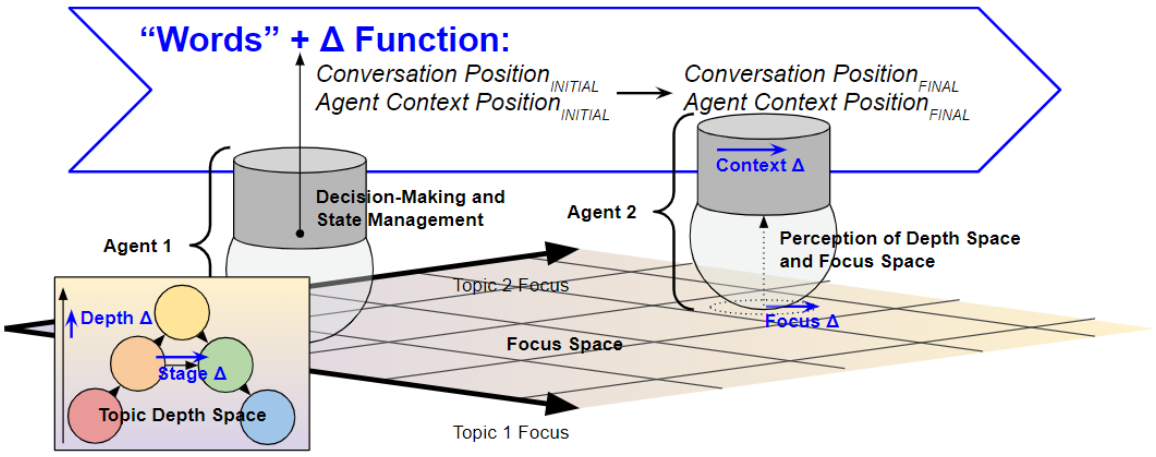


Figure 12. A two-topic **dialogue space** with two **dialogue agents** located on it.

The resulting model is analogous to a physical space, where agents are located at some point and can apply change to their own and everyone else's positions. A general model of this is visualized on figure Figure 12.

3.7 The Ludic Dialogue System as Software

In practice, the ludic dialogue system can be realized as a multi-agent system where agents act asynchronously through some shared medium, using an object for each speech action. In software, it does not make sense to represent everything in this system as instructions. The procedures of topics and thoughts can be generalized and used through shorthand. This allows them to be data-driven, and written by a non-programmer. The primary focus of this subchapter will be to find how to translate the theoretical model to software.

In this thesis, we have only considered the basic situation where a user-controlled agent (PC) is conversing with a computer-controlled one (NPC). As before, the agent's responsibilities can be divided between perception (input) and acting (output). The conversation medium is the implicit or concrete space between them. The conversation emerges from the feedback loop between their inputs and outputs (see Figure 13).

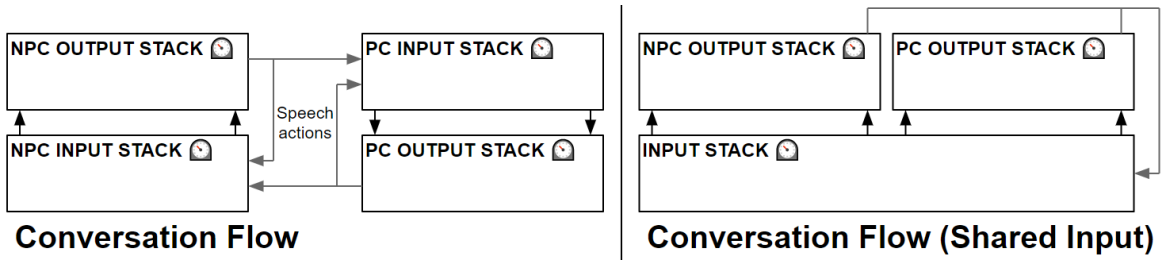


Figure 13. Possible conversation flows between two dialogue agents.

As before, the perception of the conversation space could be individual and both agents have a unique input stack, or it could be shared and both agents share a single input stack. However, the agents can never share decision-making or state-tracking.

Agents express thoughts through speech actions. These are objects that allow the transmission of thoughts from one agent to another within some length of time. Speech actions can be received using the *observer programming pattern*. A *mediator pattern* acting as the conversation medium can be used to decouple the agents further from one another[19].

The input stack (see Figure 14) has a database for all possible thoughts, which are authored

[19] Both patterns are outlined in *Design Patterns: Elements of Reusable Object-Oriented Software*. The *observer pattern* describes how a subject automatically notifies its observers of changes. The *mediator pattern* describes how an in-between object breaks up communication between other objects.

individually by the writer. Thoughts are queried and filtered by an attention module, which frames the current conversation and decides which thoughts might be relevant and forwards them to the output stack. The `Observer` module perceives actions and activates a process of change across the agent. This thesis considers each thought to be related to a single topic.

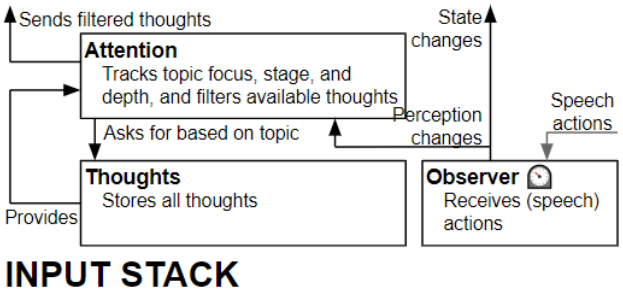


Figure 14. Division of labour in the input stack.

Thoughts consist of three main parts: display text, their topic, and change variables. Instead of defining a change function for each thought individually, topics use a general function for moving the agents through *dialogue space* and influencing their state, although thoughts may have variables to specify some part of that change to that thought. This makes thoughts easier to write. Thoughts can have other parts besides these three as well, for example, who can express them.

Topics have a keyword for reference. They also have relations with one another (*subtopics* and *supertopics*).

The output stack (see Figure 15) is inherently different for a PC and an NPC agent because the PC needs to wait for player input while the NPC needs to decide on a course of action on its own. The responsibility of outputting speech is similar.

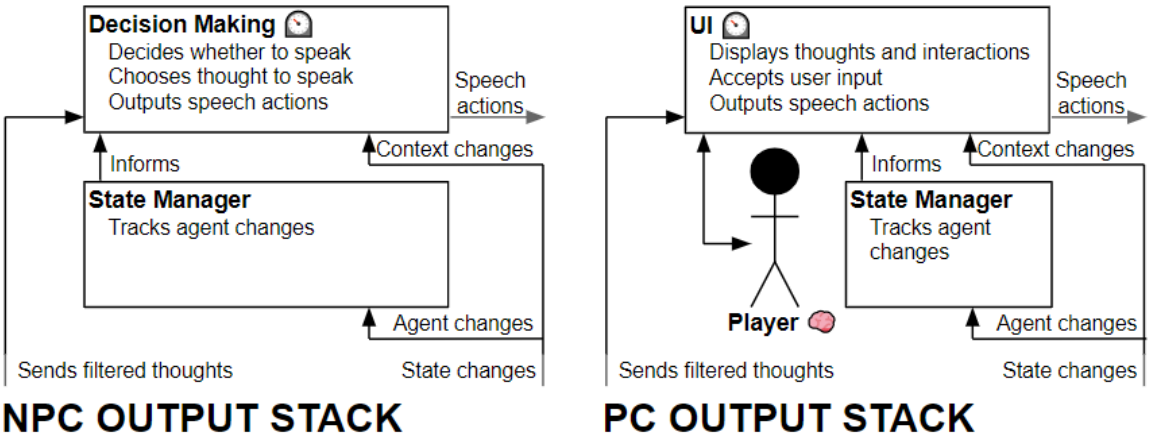


Figure 15. Division of labour in the NPC and PC output stacks.

The `Decision Making` and `UI` components might be more or less rudimentary depending on the implementation. They are similar in that both have a list of thoughts for

the agent to pick from and construct speech actions based on these thoughts. The NPC needs to score the thoughts to find the one it would most like to express. Additional logic is necessary to keep the NPC from constantly interrupting themself and the player. However, making decisions depends on the agent's current state.

The `State Manager` informs the `Decision Making` about how the other agent has affected them on a permanent emotional or functional level. The state manager is not as necessary for the PC stack, as the player's emotions already determine which of the dialogue options they will choose. However, both may have variables to track in the context of the game.

So far, I have only described the design of the dialogue system in theoretical ideals. The following chapter will show how the system behaves in an environment where I want it to accomplish a specific game design goal.

4 Use Case: *Last Call*

To demonstrate the viability of the ludic dialogue system, it was used in a short demo game with the system as its only gameplay mechanic. With this system, a 15 minute experience between the player and the computer was created. If the ludic dialogue system is a viable core gameplay mechanic, the players should recognize the system as a game without the inclusion of additional gameplay mechanics[20]. The final product can be seen in Figure 16.

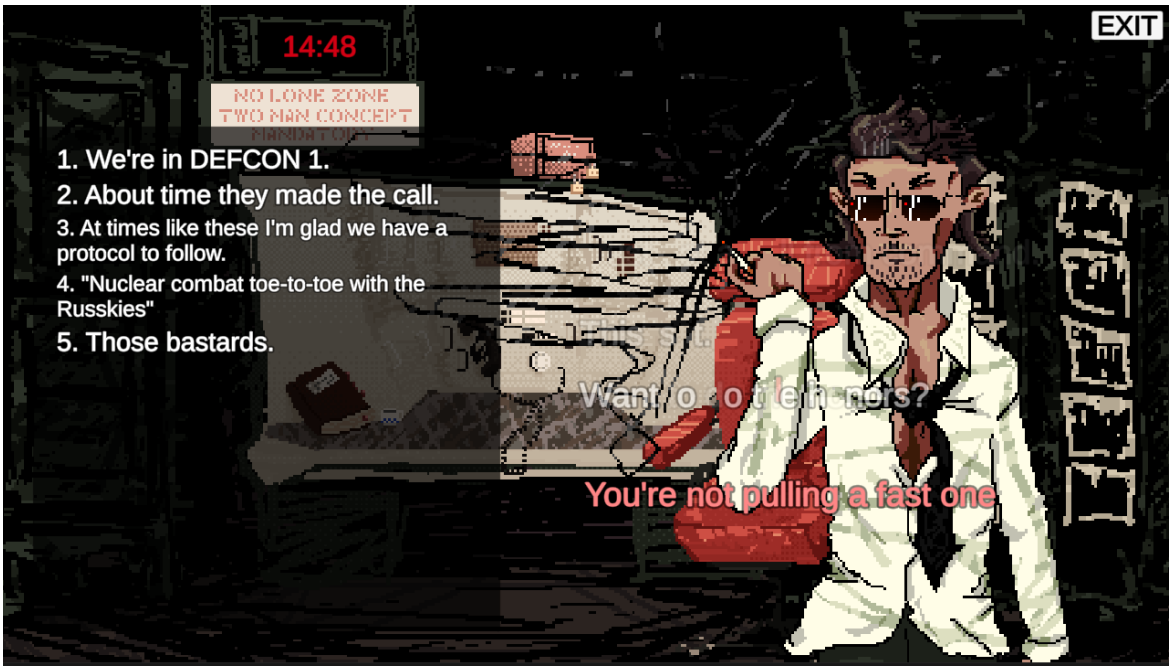


Figure 16. The UI of *Last Call* has three main components: the dialogue picker, similar to what one would see for standard branching dialogue, the dialogue displayer, which shows text and plays sound as it is being spoken, and the visual representation of the NPC the player was talking to, whose appearance changed based on emotional state.

The Unity game engine, which uses C# as its scripting language, was used. Unity had many advantages over other game engines: I was familiar with it and did not have to pay anything for it. Unity also comes with several out-of-the-box solutions for common gameplay elements like the game loop, audio, and display.

[20] Specifically the ones I did not mention in the previous chapter, like health bars, levelling, and freedom to move through a physical space.

4.1 Implementing the Stack Structure

The system was implemented using the stacks visualized in the previous chapter, creating an abstract class for each module, allowing specific implementations to vary without worrying about communication between the modules. The latter was taken care of by the classes representing the stack: `InputStack`, `OutputStack`, and `ActorStack`. The input stack and output stack knew about their respective modules, and the actor stack facilitated communication between the two. There were two main reasons for this architecture. First, the system is a proof of concept for the functionality of the architecture and should clearly follow that architecture, and second, it makes it possible to reuse code.

The communication between stacks was implemented using the *publish-subscribe pattern*[21], with the decider module being the publisher and the observer module being the subscriber. As the agents spoke, they sent out *Speech* objects, which expressed a thought, and its progress towards completion. Completion was divided into three stages with different effects occurring at each.

So far in the thesis, a dialogue has only been talked about as something happening between two dialogue agents, but with this implementation it is possible to have conversations between any number of dialogue agents. In this game, the agents were allowed to soliloquy on occasion, although progression through some topics was harder or impossible without input from the other agent.

4.2 Implementing the Attention Module

The `Attention` module is the most unusual part of this dialogue system because it is here that the flow of conversation emerges. This implementation kept track of the current stage of the topic and its current focus (focus and depth were equated). The distance between the current depth and their desired depth allowed them to be filtered. The change function was based on closing the difference between the current position of the topic and the desired position of a related thought. The entire logic is laid out in Table 1.

[21] Another pattern outlined in *Design Patterns: Elements of Reusable Object-Oriented Software*. The *publish-subscribe pattern* describes how message are turned into objects (published), which any subscriber can receive. The publishers and subscribers do not need to know each other.

Table 1. Logic governing states.

Stage	Pickable thoughts	Special rules	Exits to	Relative Depth
None	-	Only used by thoughts to show that they are not tied to a stage	-	None
Hidden	-	Initial stage for all topics	**Orientation** when a tangentially related thought is heard	Zero
Orientation	Orientation, Complication	-	**Complication** when a complication thought is heard	Low
Complication	Orientation, Complication, Climax, Denouement	-	**Climax** when a climax thought is heard; Denouement when a denouement thought is heard	Medium
Climax	Complication, Climax, Denouement	Focus must rise or exits to denouement	**Denouement** when a denouement thought is heard or focus fails to rise	High
Denouement	Complication, Denouement, Coda	Focus can only fall	**Coda** when a coda thought is heard	Medium
Coda	Coda	Picking any thought exits	**Ended** when another coda thought is heard	Low
Ended	-	Topic becomes locked	-	None

This dual approach allowed a higher depth level to unlock movement to more involved stages, but it also allowed them to be skipped. Some special rules were also added to close the topic quicker after arriving at the climax. The key here is that stages can only be traversed in one direction while depth can move up and down.

4.3 Turn-Taking Strategy

Turn-taking may seem hard to implement, but it really boils down to two aspects: how much

someone desires to speak and how much someone desires to hear someone else speak, which was interpreted as a simple product of variables that influenced the waiting time until speaking. Since thoughts were already being ranked based on the agents' ranking for expressing them.

Ranking aggregated depth, focus, and emotional affinity of a thought into a single numerical value, which were became a threshold for how time waited until speaking it. With a high enough rank and a low enough respect for the other, the agent would not care whether the other agent was speaking. The simplified basic calculation was:

1. $w_{thought,emotion}, d_{thought}, t_{base\ threshold}$ are static values inputted as data.
2. $f_{topic}, e_{emotion}, C$ are dynamic values known at the start of calculation.
3. Depth being d and focus being f: $d_{topic} = f_{topic}$
4. $d_{thought\Delta} = |d_{topic} - d_{thought}|$
5. $f_{topic\%} = \frac{f_{topic+}}{f_{\Sigma}}$ or 1, if $\frac{f_{topic\Sigma}}{f_{\Sigma}} = NaN$
    a. $f_{topic+} = f_{topic} + f_{subtopic+} \cdot 0.7$
    b. $f_{\Sigma} = \sum f_{topic+}$
6. Filter score being s: $s_{thought} = d_{thought\Delta} \cdot (1.5 - f_{topic\%})$
7. Thoughts, where $s_{thought} < 10$, were filtered out.
8. Affinity being a, emotional weight being w, and emotional state being e:
   $a_{thought} = \sum_{emotion} w_{thought,emotion} \cdot e_{emotion}$
9. Rank being r: $r_{thought} = s_{thought} \cdot 2 - a_{thought}$
10. Time being t and context being C:
    $t_{threshold\ for\ speaking\ thought} = t'_{base\ threshold} \cdot r'_{thought} \cdot$
    a. $t'_{base\ threshold} = AccountForContext_t(t_{base\ threshold}, C)$
    b. $r'_{thought} = AccountForContext_r(r_{thought}, C)$
11. $DecideBasedOnContextWhetherToSpeak(t_{threshold\ for\ speaking\ thought})$.

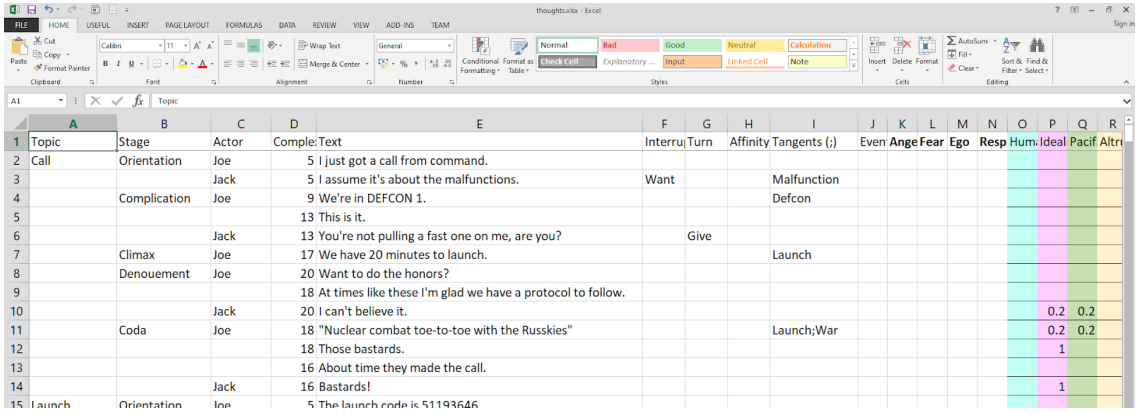
Taking context into account and a layer of final decision making gave a semblance of civility to turn-taking. A variable was given to each thought, which determined turn-selection strategy following the thought. The agent would wait for a shorter or longer time depending on emotional state, respect for the other, and what the last heard turn-selection strategies were.

4.4 Authoring Content

When it came to authoring content, the `Attention` module and the *dialogue space* it generated played the most significant part in the process. Since topics were no longer directly connected through order, they were written so that they could appear more or less in any order. The stage structure was used to force some order back into the dialogue. This required spacing the desired depths of the thoughts in a topic in a way that would allow filtering based on distance limits to prevent them from appearing out of other.

The spacing promoted a pattern in my writing where going deeper into a topic would require the player to be more opinionated and ending early less opinionated. As a result, smaller topics were more linear, as the agent's only choice was to commit to an opinion or fall back from it.

Thoughts being out of order also meant that pronouns were unavailable and the same noun phrases had to be written out in full repeatedly. Connecting sentences through linking words was also more challenging, as a topic might be cut into by another topic, and every thought had to appear to link to every other thought. The linking was occasionally indirect, like someone not really responding to the other and choosing to monologue. This is a somewhat realistic feature of the dialogue system, as we often already know what we want to say before the other person speaks. The use of pronouns and linking words could be automated based on whether the thought is building on the previously spoken thought or not.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Topic	Stage	Actor	Comple	Text	Interru	Turn	Affinity	Tangents (;)	Even	Ange	Fear	Ego	Resp	Hum	Ideal	Pacif	Altri
2	Call	Orientation	Joe	5	I just got a call from command.													
3			Jack	5	I assume it's about the malfunctions.	Want			Malfunction									
4		Complication	Joe	9	We're in DEFCON 1.				Defcon									
5				13	This is it.													
6			Jack	13	You're not pulling a fast one on me, are you?		Give											
7		Climax	Joe	17	We have 20 minutes to launch.				Launch									
8		Denouement	Joe	20	Want to do the honors?													
9				18	At times like these I'm glad we have a protocol to follow.													
10			Jack	20	I can't believe it.											0.2	0.2	
11		Coda	Joe	18	"Nuclear combat toe-to-toe with the Russkies"				Launch;War							0.2	0.2	
12				18	Those bastards.											1		
13				16	About time they made the call.													
14			Jack	16	Bastards!											1		
15	Launch	Orientation	Joe	5	The launch code is 51193646													

Figure 17. A screenshot of a worksheet Microsoft Excel 2013. Note how every cell is not filled in. Topic, stage, and actor would only be written in when they changed. Complexity and text were the only columns that were required. Other values would default to zeroes, empty lists, or nulls when read in.

The lack of formal structure promoted thinking on the level of conflict rather than talking points during writing. While sometimes order was reintroduced into dialogue, the system did not require it. This organization was very natural to think about, as it was similar to creating a mind map. The workspace was Excel and an example of what it looked like can be seen in Figure 17. A specialized tool could formalize and automatize the technique.

That said, there are some things I would do differently in the future. First, I would keep focus and depth clearly separate. I equated the two to simplify the system, but this coupling had undesired effects with minimal time gain. Having everything be calculated on the fly made the dialogue flow less comprehensible.

I also noticed a pattern in how I used the stages of a topic to create specific behaviour within the dialogue, like asking and answering questions. In the future, it might be more profitable to think of stages as repeating blocks of a conversation, within which agents must act a certain way, rather than categories of progression. The change would reintroduce even more order into topic progression without resorting to creating strict branching between thoughts.

4.5 Evaluation

In total, it took me 31 hours to implement the general structure of the ludic dialogue system. It was a sizable effort with a reusable result. The creation and design of game systems such as the display and the state-based behaviour of agents took me 21.5 hours, which is a comparatively small effort but one which I would expect to increase with a more elaborate game. Writing dialogue took me 25 hours, or 1500 minutes. There were 645 lines of dialogue totalling up to 6726 words. This means I wrote at a speed of 4.4 words per minute. The writing time only includes some light editing and proofreading. A conservative estimate for the speed of writing would cut that speed in half to around 2.2 words per minute[22].

The methodology for tracking time expenditure on tasks included the occasional short breaks. After all, it was spent ruminating on the more complex problems. A stopwatch was used to keep track of time, from when the work on a new task started until when it was finished. The timekeeping erred on the side of overstating the time it took to complete a

[22] Writing speed is difficult to quantify. Some professional writers say they write 5000 words a day, others say they are happy when they can get out 500 good words (words that require little to no revisions). Creativity, quality, preparation, and genre-constraints also act as variables in the process.

task, as this thesis would like to demonstrate that the resource costs of the ludic dialogue system are as small as possible.

Some people were also asked to play the *Last Call* and answer some questions about it. Since the aim of my thesis might have been known, precautions against bias needed to be taken. For each question, possible bias is acknowledged for qualitative evaluation and flat-out removed for quantitative evaluation. For the quantitative evaluation, see Table 2.

Table 2. Table of yes or no questions. Questions 2-5 only include those who answered 'No' to the first question.

Question	**Yes**	**No**
Were you in any way aware of what I was trying to accomplish with my thesis?	5	**5**
Would you call LAST CALL an interactive experience?	4	1
Would you call LAST CALL a real game?	5	0
Would you call the conversation in LAST CALL natural?	4	1
Would you call the conversation in LAST CALL engaging?	5	0

Each yes or no question also asked the person to explain, "Why or why not?" they gave that answer. We can see from the table that everyone who took the questionnaire and did not know about the thesis goals considered the experience to be a game. Because the game had no other systems, the dialogue system could indeed be considered ludic.

In calls to action to try the demo out, language which would code the experience as a game was avoided. The channels used were computer-science-aligned communities, mainly the Computer Graphics and Virtual Reality Lab at the University of Tartu, which was contacted through a mailing list. Sharing it with a gaming-aligned community would have also coded it as a game. Despite that, the Unity game engine and the itch.io distribution site were used. Neither is used exclusively for games, but both having a gaming focus.

Some responses explained that the definition of a game is pretty broad, and if the choice was between one and the other, it was more game than not. Others compared it to genres that already existed, saying that it was like a more interactive visual novel or like an RPG with multiple endings. Others saw it as a game because it was engaging, interactive, or fun.

Interestingly, one person thought that it was a game despite it not seeing it as being

interactive. They said that it was still a game, even if the interactivity was an illusion. Overall, the basis for a ludic system, interactivity, seems to be there.

Some among those who said they knew about the goals of my thesis, “felt” like they could influence the superior officer (the other dialogue agent in the *Last Call*). While the goal of the thesis never outlines the ability to influence NPCs, it could be inferred from the goal of interactivity and the framing of the story being about convincing someone. The person that did not deem the experience as interactive, supposed that it was based on the illusion of choice, not true interactivity. However those who knew me or my thesis goals might have supposed that I would not create an experience where choice was just an illusion.

The main reason for having a ludic system, engagement, also seems to have been achieved. One person felt they were a “cocreator of the dialog”, a sentiment echoed by others. People were interested in the conflict between the two characters. The one person that felt that the experience was not engaging said that the dialogue went by too fast for them to engage at any moment.

The system seems to have failed in making the dialogue seem natural. The main reasons brought up were the franticness of the conversation, the pickability of old dialogue and the superior officer ignoring the player. Only people who knew about the goals of my thesis saw the dialogue as natural. One of them writes that while the player was able to express thoughts from a long time ago, it was up to them whether they wanted to do so.

Based on this, I believe that part of the issue with this system is that the system is not very intuitive and the user interface was poor. Several people reported not realizing at first that they had to hold down the dialogue option to express it. Several people also complained that the conversation was too frantic and that the superior officer seemed to ignore them. What only those familiar with my thesis knew was that the superior officer’s behaviour depended on following social etiquette.

While the character’s mood was represented by incremental changes in the sprite of the superior officer, an apparent immediate change or sound effect would have helped players figure out that they had just committed a faux pas. Some tutorializing on interfacing with the UI and the ways in which to influence the NPC would also have been useful as well.

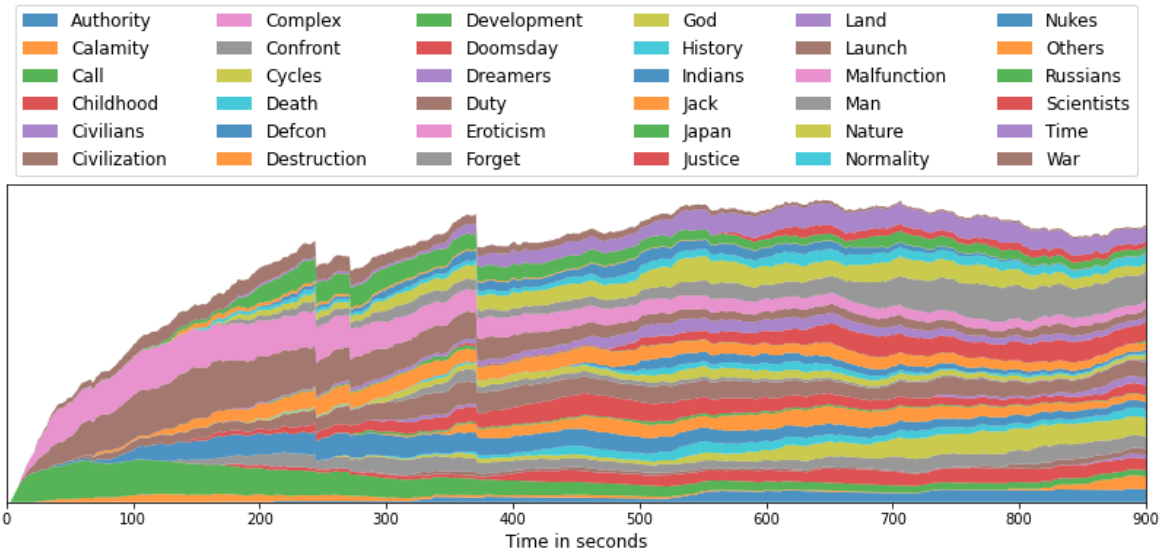


Figure 18. Cumulative depth of topics across several play-throughs over in-game time. Sudden drops in cumulative depth signify when a play-through of *Last Call* ended.

While a player could operate under the illusion of choice and call the experience interactive, I also wanted to check whether people could indeed influence the flow of the conversation. For this purpose, I gathered logs to determine how the depths of topics changed over time. There were 15 logs in total. At the start of the experience, the first available topics (Call, Malfunction, and Launch) dominate the experience. Afterwards, the experience becomes very varied (see Figure 18) Otherwise, a handful of topics would continue to dominate the graph one after the other.

In conclusion, *Last Call* shows that the ludic dialogue system could be considered a system that enables dialogue to be the core gameplay mechanic of a game. *Last Call* also demonstrates that the system is a viable medium for interactive storytelling. The success of the system was, however, dependent on adjusting variables and even then the result was unnatural. In other words, it is reliant on game design. For commercial use, steps should be taken to make the system less of a black box. Creating tools for simulating possible movements through *dialogue space* would be the main way to do this.

5 Conclusion

Video game systems offer interactive experiences with the purpose of engaging the player. Interactive fiction emphasizes storytelling and often uses dialogue between player and non-playable characters to tell its stories. The standard dialogue systems used for this purpose attempt to be engaging, but they are by their nature less interactive than other gameplay systems. The ludic dialogue system created in this thesis plays to the strengths of video games, and incorporates procedure-based interactivity into dialogue. The increase in procedural surface area, opens dialogues up to game design.

The devised and presented ludic dialogue system builds on the commercial needs of standard branching dialogue systems, outlined by Freed, by proceduralizing turn-taking and the order of focus. The dialogue it creates, is based on the naturalistic way everyday people hold conversations, as described by the cognitive linguistics of Chafe. Because of this, it cannot rival the directed quality of branching dialogue. However, the system does not simply seek to simulate conversation. It seeks to enable video games as a medium for interactive storytelling, as described by Crawford, et al.

To enable the proceduralization of conversation, a *dialogue space* was defined along with dialogue agents who traverse it. The conversation would emerge from the agents' movement across this space. The *dialogue space* comprises of the elements of attention, represented by the *focus space*, and the elements of dramatic progression, represented by the *depth space*. The dialogue agents' social behaviour is divided into sets of procedures, or modules. The key one is the attention module, which tracks the agent's movement across *dialogue space*.

The demo project *Last Call* demonstrated the ludic dialogue system's viability for creating engaging interactive fiction, where dialogue was at the core of the gaming experience. However, turn-taking and the order of focus were still unnatural, and the user interface was unintuitive and abstruse. It also remains to be seen, whether the ludic dialogue system is usable in a professional environment and in large-scale projects.

The main areas of further research on the ludic dialogue system lie in improving the attention module and in finding appropriate ways for the user to interact with the system. In addition, automatically generating unifying elements of speech, like linking words and pronouns, could simplify writing for this system. The tonality of the speech could also be generated.

Regarding potential uses, I recommend trying to translate long-form social interactions into video games. This makes the best use out of the naturalistic nature of the ludic dialogue system. Examples for these would be the interviews and court hearings of detective fiction or the courtships of romantic fiction. Lastly, of course, the system could be used to translate the freeform social interaction of tabletop role-playing games to role-playing video games.

I hope that this thesis has enabled the avid reader to see the potential of dialogue for crafting new types of video game experiences, and that it has also inspired them to use a ludic dialogue system to craft a video game experience of your own.

I would like to thank all those who tried out *Last Call*, especially the ones who filled in the questionnaire. The data gathered was vital for determining the viability of the system from a player's perspective. My sincere thanks to my supervisor Raimond-Hendrik Tunnel whose experience with video games and knowledge of scientific and academic standards raised the quality of this thesis considerably. Lastly, Professor Raul Vicente's introductory course on computational neuroscience was irreplaceable to my thinking about the cognitive aspect of the social behaviours of dialogue agents.

References

[1] *PC Gamer UK*, no. 137, pp. 104-105, 2005, Referenced through Codex Gamicus. [Online]. https://gamicus.gamepedia.com/Planescape:_Torment (9.12.2020)

[2] @studioZAUM. (2020, Jan.) Twitter. [Online]. https://twitter.com/studioZAUM/status/1215597783509610497 (9.12.2020)

[3] Chuck Sambuchino. (2016, Sep.) Word Count for Novels and Children's Books: The Definitive Post. Article. [Online]. https://www.writersdigest.com/whats-new/word-count-for-novels-and-childrens-books-the-definitive-post (30.1.2021)

[4] Marc Brysbaert. How many words do we read per minute?: A review and meta-analysis of reading rate. [Online]. https://doi.org/10.31234/osf.io/xynwg

[5] Alexander Freed. (2014) Branching Conversation Systems and the Working Writer, Part 1: Introduction. Blog Post. [Online]. https://www.gamasutra.com/blogs/AlexanderFreed/20140902/224609/Branching_Conversation_Systems_and_the_Working_Writer_Part_1_Introduction.php (9.12.2020)

[6] evilwillhunting. (2020, Mar.) 'Not Recommended' review of Disco Elysium on Steam Reviews. [Online]. https://steamcommunity.com/profiles/76561198030961604/recommended/632470/ (21.3.2021)

[7] @ID_AA_Carmack. (2018, Aug.) Twitter. [Online]. https://twitter.com/ID_AA_Carmack/status/1030511549688016899 (30.1.2021)

[8] Chris Crawford. (2018) Why I am ending further work on interactive storytelling. [Online]. https://web.archive.org/web/20180820035048/http://www.erasmatazz.com/library/interactive-storytelling/why-i-am-ending-further.html (31.1.2021)

[9] Daniel Stamatov. (2015) Murder & Masochism: Exploring Violence as a Mode of Engaging with Fiction in From Software's Dark Souls.

[10] Marc LeBlanc, , Katie Salen and Eric Zimmerman, Eds.: The MIT Press, 2006, pp. 438-459.

[11] Jan-Noël Thon and Bettina Bodi. (2021) Playing Stories?: Narrative-dramatic agency in Disco Elysium (2019) and Astroneer (2019). [Online]. https://doi.org/10.1515/fns-2020-0012

[12] Alexander Freed. (2014) Branching Conversation Systems and the Working Writer, Part 2: Design Considerations. Blog Post. [Online]. https://www.gamasutra.com/blogs/AlexanderFreed/20140909/225281/Branching_Conversation_Systems_and_the_Working_Writer_Part_2_Design_Considerations.php (9.12.2020)

[13] Gillain Brown and George Yule, *Discourse analysis*. Bath: Cambridge University Press, 1983.

[14] Michael Mateas and Andrew Stern. (2003) Façade: An Experiment in Building a Fully-Realized Interactive Drama. Research Article.

[15] Alexander Freed. (2014) Branching Conversation Systems and the Working Writer, Part 4: Key Principles. Blog Post. [Online]. https://www.gamasutra.com/blogs/AlexanderFreed/20140924/226302/Branching_Conversation_Systems_and_the_Working_Writer_Part_4_Key_Principles.php (31.1.2021)

[16] Alexander Freed. (2014) Branching Conversation Systems and the Working Writer, Part 5: Arts and Craft. Blog Post. [Online]. https://www.gamasutra.com/blogs/AlexanderFreed/20141001/226764/Branching_Conversation_Systems_and_the_Working_Writer_Part_5_Arts_and_Craft.php (31.1.2021)

[17] Dominic Arsenault, *Narration in the Video Game: An Apologia*.: Université de Montréal, 2006, (Translated from French to English by the author 2006-2007).

[18] Bob Nystrom. (2014) Bob Nystrom. Book. [Online]. http://gameprogrammingpatterns.com/contents.html (1.3.2021)

[19] Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson, "A Simplest Systematics for the Organization of Turn-Taking for Conversation," *Language*, vol. 50, no. 4, 1974. [Online]. https://doi.org/10.2307/412243

[20] Wallace Chafe, *Discourse, Consciousness, and Time*. Chicago: The University of Chicago Press, 1994.

[21] Chen H., Liu X., Yin D., and Tang J. A. (2017) A Survey on Dialogue Systems: Recent Advances and New Frontiers. Research Article. [Online]. https://doi.org/10.1145/3166054.3166058

[22] Michael Mateas and Andrew Stern. (2002) A Behavior Language for Story-based Believable Agents. Research Article. [Online]. https://users.soe.ucsc.edu/~michaelm/publications/mateas-is-2002.pdf (2.3.2021)

[23] Leene Künnap. (2019) Surmateemalise arvutimängu disainimine virtuaalturule. Dissertation. [Online]. https://digiarhiiv.pallasart.ee/wp-content/uploads/2019/06/Leene_K%C3%BCnnap_kirjalik_ja_lisad.pdf (5.4.2021)

[24] Death And Taxes: Placeholder Gameworks. Article. [Online]. https://www.articy.com/en/showcase/death-and-taxes/ (5.4.2021)

[25] Alexander Freed. (2014) Branching Conversation Systems and the Working Writer, Part 3: Building a Conversation Tree. Blog Post. [Online]. https://www.gamasutra.com/blogs/AlexanderFreed/20140916/225763/Branching_Conversation_Systems_and_the_Working_Writer_Part_3_Building_a_Conversation_Tree.php (31.1.2021)

[26] Wallace Chafe, "Searching for Meaning in language: a Memoir," *Historiographia Linguistica*, no. 29, pp. 245-261, 2002.

Appendix

I. Accompanying Files

The thesis is accompanied by the following:

- *LAST CALL (BUILD).zip*, an archive which includes a playable version of the demo project created for this thesis, it is the same version played by the people who responded to the questionnaire.
- *Last-Call-Thesis-version-hotfix-3 (GIT REPO).zip*, an archive which includes the Unity project used to build the playable version of the project, also found in the GitHub repository https://github.com/Raud0/Last-Call.
- *Anonymized questionnaire results*, a file folder with the following files:
    - *LAST CALL exit questionnaire (Responses).xlsx*, an Excel worksheet with the individual responses of the responders. The links to their logs and the dates of the responses have been removed from the file, as they were not used for analysis and may be used to identify the responders.
    - *LC_log_*.csv*, several .csv files, all of which are records of the topic depths of a single play-through over time.

II. Glossary

Ludic	Relating to games
Player character	An in-game character controlled by the player, often acts as their avatar within the game world; abbreviated as PC.
Non-playable character	An in-game character controlled by the computer, often governed by game AI.
Dialogue	A systematized social interaction that typically takes place between a PC and an NPC.
Conversation	Social interaction as it is happening, regardless of systems.

Game AI	Artificial intelligence that controls an agent within a game focused on creating the illusion of intelligent behaviour rather than actual intelligent behaviour.
Dialogue branch	A split in the dialogue, where the line of conversation branches into several possible conversation lines; a typical branch is composed of an exchange of words between the PC and the NPC.
Uncertainty and inevitability	The dramatic aesthetics that establish that the conclusion of a conflict is unknown but its concluding is inevitable.
Spatial-explorative agency	The player's ability to traverse a space with their avatar. Most games have this.
Temporal-ergodic agency	The player's ability to explore the game as a temporal system through their avatar. Most games have this. Many older dialogue systems stopped the game for the duration of the dialogue.
Configurative-constructive agency	The player's ability to configure their avatar and construct the game world.
Narrative-dramatic agency	The player's ability to participate in the game's story through their avatar.
Hub and spoke	A typical structural element in branching dialogue, where the player returns repeatedly to a node with the same set of dialogue branches.
Waterfall	A typical structural element in branching dialogue, where the player cannot return to a node after selecting a branch.
Critical path	A line of conversation that contains key information and should thus be unavoidable.
Interactional view of linguistics	A view of linguistic analysis that places importance on the expression of social relationships between speakers.
Transactional view of linguistics	A view of conversations that places importance on the expression of new information between speakers.

AB structure	An implicit pattern in standard branching dialogue where the dialogue agents switch taking turns, responding to each other directly.
Topic	Something that the conversation is about.
Basic-level topic	A primary topic that is self-contained and does not depend on another topic.
Subtopic	A secondary topic that expands and depends on another topic.
Supertopic	A meta topic that basic-level topics are related to but which exists on a higher than consciousness level, leading to a lack of structure.
Stage	A category of progression in a topic, each being functionally distinct.
Dialogue Agent	An intelligent system that participates in dialogue.
Focus	A numerical value representing the attention given to a topic.
Depth	A numerical value representing the consciousness' removal from normalcy, as the agents drive discussion of a topic towards a climax.
Focus Space	The spatial structure traversed during dialogue as the object of focus within dialogue shifts. The unit of movement is *focus*.
Depth Space	The spatial structure traversed during dialogue as the resolution of a conflict progresses. The unit of movement is *depth*.
Dialogue Space	The spatial structure traversed during dialogue by dialogue agents. A combination of *focus space* and *depth space*.
Retrieval-based and generative conversational methods	The methods of responding to the other conversational agent by either retrieving an existent response or generating a new one

III. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Frederik Raud,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   A Ludic Dialogue System for Role-Playing Video Games,
   supervised by Raimond-Hendrik Tunnel.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Frederik Raud
27/04/2021