Branches: A Game of Choice and Agency

by N. Escobar

December 14th, 2021

Edited May 24, 2023

Occidental College – Los Angeles, CA

Computer Science Department

Comprehensive Project Final Report

An N. Escobar Media Production

Project Source:

https://github.com/nickesc/BranchesGame

Download the final build:

https://github.com/nickesc/BranchesGame/releases/tag/v1.0.0

Branches: A Game of Choice and Agency

Table of Contents:

1. Introduction 3

2. Background and Literature Review 4

2.1. Agency

2.2. Choice in Other Games

2.2.1. Superior Choices

2.2.2. Telltale Games

2.2.3. *Detroit: Become Human (2018)*

2.2.4. The Stanley Parable (2016)

2.3. Storytelling

2.4. Background and Literature Review Conclusion

3. Methods 12

3.1. Technologies

3.2. Game Overview

3.3. Story Design

3.4. Gameplay Design

4. Architecture Overview 21

4.1. Unity Overview

4.2. MngrScript and Finite State Machines

4.3. States in Branches

4.4. Game Systems

5. Ethical Considerations 25

6. Evaluation 25

6.1. Metrics

6.2. Results

7. Conclusion 27

8. References 28

1. Introduction

For a long time, I've been captivated by the intersection of art and computer science. Going into this project, I knew that I wanted to do something that borrowed heavily from both fields, to make something that blended the two together. Given my long experience with and love of video games, the obvious choice for me was to create one. Video games, as an art-form, are so incredibly unique, because of their level of interactivity. They allow for choice in their consumption, giving the player a different experience based on the different choices they make. But it's also always frustrated me that most games, even if they give the player choice, present it without any real meaning. They let the player make a choice, but they don't react the way you would want them to. Your choices feel less meaningful than they could, because a lot of the time it seems like no matter what choices you make, you end up funneled to the same place.

This frustration led me to create something that actually made the player's choice feel meaningful. I decided that I wanted to make something that would facilitate a feeling of agency. To that end, I chose to make a branching-narrative based game in which the player moves through three-dimensional space from a first person perspective, and is presented with a series of choices that let them decide the path of the story as much as the game designer. I really wanted to emphasize the role that choice plays in the game, so the central theme would really be the player's actions and choices, and making them feel like they have consequences. Consequences, in this context, are just some kind of reaction from the game to your choice. If you make a choice and nothing in the world happens differently because of it, that choice has no consequences. As the player makes choices, the story will be fundamentally altered depending on which choices they've made, making different playthroughs completely different experiences based on what you choose, and creating a branching narrative that really feels like it leads to different places. Each playthrough should feel like one, cohesive story, but one that works together in conjunction with the other branching paths to create something that is more than just the sum of its parts.

This paper will go through the theoretical framework behind a game that creates a feeling of agency and document my process to create a game like that. It will explore how choice has been used in games in the past, both effectively and not, which will create an understanding of what it

means to create agency in a game. It will go through the steps that I took to make the game, from ideation to prototyping to the final build, and it will go over the game's basic, technical architecture. Finally, the success of the project will be evaluated with user testing and feedback. In the end, I believe this paper sets out best-practices for using choice to create the feeling of player agency in video games, and acts as a record of my attempt to build a game that captures that feeling.

2. Background and Literature Review

A certain amount of background on video games is necessary to understand this project to the fullest extent. A narrative based game is one in which the story is at the core of the experience, and the player goes through certain beats to experience that story. A *branching*-narrative based game is one that is centered around the story, but that story changes based on the player's choices, creating a series of branching paths that the story can follow. This ability to create a branching narrative, rather than a single, fixed one, is one of the things that makes games unique among other media. Most media is a straight path from beginning to end, with no deviation between different consumptions other than when you stop it and when you start it. Video games, on the other hand, are almost necessarily different every time they're consumed, because you have control and input on the game in some form or another. This input the player has on the game allows for the idea of choice, and allows the player to put themselves into direct conversation with the game and achieve a form of agency that doesn't exist in other media. Choice is used differently in different games, but the background of this paper will be looking specifically at the way in which meaningful choice is given to the player as a way of facilitating agency in narrative-based games. Of course, it's important to define what "agency" and "meaningful choice" really are; they mean more than just choosing which direction your avatar will move or which line of dialogue you're going to say next — it's about making choices that have consequences the player can feel and committing to them.

2.1 Agency

We derive our understanding of agency from Karen and Theresa Tanenbaum's paper, *Commitment to Meaning: A Reframing of Agency in Games*. They propose to define agency as: "the process by which participants in an interaction commit to meaning" (Tanenbaum and Tanenbaum 5). Commitment is at the heart of their definition, and it means that every time the player takes an action, they are committing to that

action and the consequences it will bring. Committing to a choice is making a conscious decision to do something, and accepting that there will be consequences, which they can predict based on their choice, as a result. Asking players to commit means asking them not to interact with games passively by letting them happen to them, but rather to be an active participant in creating the story by making deliberate choices to affect the game's world – choices with intent. When the game designer honors the choice the player makes by making the story react with the expected consequences and tailoring the experience to the choices of the player — rather than hoping the player's choices will align with their story — they give that commitment meaning. It's that combination of commitment from the player and meaning through real consequence from the game designer that work together to create the feeling of agency. As they summarize:

> This understanding of agency proposes that game designers should strive to create game and narrative experiences in which the player can demonstrate commitment to the experience, and, crucially, where that meaningful commitment is reinforced by the game's behavior. Agency is not about selecting between options in this case, but is instead about expressing intent, and receiving a satisfying response to that intent. (Tanenbaum and Tanenbaum 8)

With this, they refute a commonly held belief that players and game designers work in opposition. The game designer does not necessarily limit the experience of the player, and the player is not necessarily an agent of chaos trying to push the boundaries of the imagined world, consciously working against the intent of the designer. Instead of telling a rigid story that is only received when the player experiences specific beats, designers can craft a narrative that works in tandem with the desires and choices of the player, and they can build a world and a story together that both parties are able to commit to meaningfully.

2.2 Choice in Other Games

This begs the question of how choice should actually be presented to players in game. Tyler Paulley looks at the ways that choice is used in a number of games and genres in *The Official Rulebook for Choice in Video Games: An Examination of Choice in Modern Narrative Games*, and suggests a few things for the game designers to keep in mind while working with choice in narrative-based games.

On RPGs like *Skyrim, Mass Effect,* and *Dragon Age,* Paulley argues that choice is used to make

the player feel more invested in the narrative, making them feel like a part of the story by giving them control over things like faction allegiances or which side quests to complete, while still always ending up at the same point: "[in these games] it is likely that a player is more interested in the game's narrative since they decide the cast that accompanies them and the adventures that are taken. While the destination in these games is roughly the same, the player decides how they get there and when" (Paulley 30). With games like this, especially games like Skyrim where the choices are expansive — there are hundreds of side quests that the player can choose whether or not to do — it's important to remember that all the player's choices can't have a meaningful effect on the narrative, because there are just too many. The choices in *Skyrim* are wide but shallow; there are a lot of them, but very few offer opportunities for meaningful choice. Completing side quests in *Skyrim* has little impact on the narrative, and whether you've completed one or one-hundred, the story will always progress the same. This creates a great deal of freedom for the player, to allow them to play however they want, choosing which quests to complete, but that choice and commitment the player makes aren't honored and upheld by the game. The game doesn't react to your choices in a meaningful way — for instance, if the player chooses to go down *Skyrim's* Thieves' Guild quest-line (a string of quests that lead from one to another), there's no acknowledgement outside that quest-line that your character has chosen a life of crime. Guards aren't more suspicious, they don't get additional stealth options for completing quests in the main quest-line – the world still treats the player as though they aren't in the guild. Skyrim fails to make those choices meaningful by ignoring the commitment the player makes to their chosen action, and the consequences those choices should have. Paulley notes, however, that because *Skyrim* is so expansive, it would be impossible for all the player's choices to all impact the narrative. The genre struggles to facilitate meaningful choice because these games have a specific destination, decided by the author, that the player needs to end up at. Paulley describes the phenomenon like a river while talking about Bioware's critically acclaimed 2010 RPG, *Mass Effect 2*:

> Regardless of the dialogue choices made by the player, the player is led down a clear narrative progression that the game designers intended… 'Mass Effect 2 is like a river' where the current will eventually carry the player to a predetermined destination. However, the player is allowed to determine the speed of the current (Paulley 16).

As they flow down the river to the final destination, they're given different choices, and they can drift from bank to bank, but they're always flowing in the same direction, seeing generally the same things. They may give the player a final choice to decide how the game ends, but they are always flowing towards that climactic decision; they fail to honor the commitment of the player by chugging along towards their intended destination, regardless of the choices the player is committing to.

2.2.1 Superior Choices

One of the things Paulley seems to laud, not tied to any particular game genre or type, is the existence of hidden, "superior" choices. These are choices that aren't available unless other criteria have been met, and they're usually a middle-of-the-road, reasonable path between two more extreme options that weren't hidden. They usually result in the positive consequences from both choices being realized, while they mitigate the negative ones. Paulley uses this example:

> In [*Dragon Age Origins*] the player is faced with the options of killing a demon-possessed boy or to undergo a lengthy quest to free him of possession by entering a spirit world. Even if the longer more moral choice is taken, the conflict still ends with the death of the boy's mother. However, a hidden resolution to this conflict exists. If the player has sided with the mage faction previously, then they can use their magic to free the boy. However, if the player has not yet completed that storyline or sided with a different faction then this option is left unknown. When previous choices come to affect future choices such as this it "makes the whole world seem very [inter]connected." (Paulley 10-11)

In this, we see that the outcome from the superior choice results in the good ending, where everyone walks away happy and no damage has been done. Paulley argues that this makes the world feel more connected, but he fails to see that this could very well *undermine* the initial options instead of making the choice more meaningful. It might sound silly to say about a game, but if you know that there's a hidden option, it gamifies unlocking that option. It encourages the player to go out of their way, to suspend their commitment to their character and to the other choices they've made, to get the objectively best outcome. It allows the player to have their cake and eat it too, not having to choose an option that would result in any negative consequences. It doesn't ask the player to build the game based on their choices, it asks them to artificially manufacture those choices in the game. Rather than give two, different choices

that both have meaningful consequences, it asks the player to forget about those and seek out the superior choice. Along the same lines, Paulley does criticize games for providing obviously black and white options, like in the example above. The game gives the player the option to save a boy or kill a boy, and 9/10 times people will probably be inclined to save the boy because it feels like the objectively right choice to make. One of those options is better than the other. This kind of choice is an issue because it coerces the player into making the "better" choice, again undermining the agency the player has by making them feel like they have to make a certain choice rather than committing to their decision on their own.

2.2.2 Telltale Games

Telltale Games is a studio known for producing games like *The Walking Dead* and *The Wolf Among Us,* which are designed and marketed around the choices you make in the story, and use choice to emphasize and elicit emotion. They play more like an interactive movie, with little emphasis on gameplay and more on how the narrative progresses. The player's choices determine how certain parts of the story play out, and certain actions have consequences that will affect your options later in the game. However, they too suffer from a familiar problem. They're still beholden to the flow of the river, to the author's intended narrative path. The story is impacted based on the choices the player makes, not built by them. They claim to present choice, but really it's an illusion:

> Telltale is possibly the best example of the trickery of choice-based games, with more and more gamers feeling that their games present the illusion of choice while ultimately herding you down the same path each and every time. (Williams)

The story bends around the choices, it can adapt to whatever variables you put into it. You can make a choice in the moment, and the rest of the scene you're in at the time might play out like you made that choice, but at the end of the scene, no matter which choice you made, it'll almost always move on to the same story beat next. Games like this and Feral Interactive's *Life is Strange* are a "one-size-fits-all" approach most of the time, moving from predetermined beat to predetermined beat every time and filling in certain details based on the choices you made. The choices can't be meaningful, especially in a game that touts the consequence of choice, because they don't really have an impact on how the story ends, or most of the time even what happens next. It will keep flowing down the river to the same exact spot, no matter what choices you make.

2.2.3 Detroit: Become Human (2018)

Detroit: Become Human is an example from Quantic Dream of a game that is marketed around choice, but in the end fails to live up to that. It attempts to take the idea of branching narrative to another level, but still suffers from the flow of the river. The game is huge, with lots of branching paths that let conversations spiral in different directions, and the branching *at some points* makes it feel like you're experiencing things in completely different ways based on your choices. Hailey Williams, in a Kotaku Australia article, even goes so far as to claim that "*Detroit: Become Human* is possibly the first game that has come through on the promise of a truly branching, reactive storyline" (Williams). However, despite its expansive branching narrative, it still has the same fundamental problem other games do when it comes to choice: it pushes you in a direction, it still wants to follow a specific narrative, predetermined by the author. In the end, it wants to tell a story with a specific flow. The choices you make don't really feel like you're building the story even if the choices themselves result in big consequences, because you still end up going to the same place, even if you see different things on the way there. And the number of choices also undercuts the feeling of importance you get from what meaningful choices there are. At every turn, it asks you to make a choice. Most of these choices mean nothing – as in they have no impact on the game, and they are just there for the sake of making a choice. Because the game is so saturated with these meaningless choices, it makes it feel like nothing you do really has any impact. The branching story feels more like they're just trying to fit in as much content as they can, and like they're not focusing on making every bit of that content meaningful. *Detroit: Become Human* is a good lesson for what not to do if you want the player to feel like all of their choices matter. It fails to commit to the choices the player makes, and because of that it fails to create a sense of agency.

2.2.4 The Stanley Parable (2016)

Another very intriguing example of choice in games is Galactic Cafe's *The Stanley Parable*. *The Stanley Parable* tells the story of a man who finds himself alone in an office building. The player, playing as Stanley, is given instructions from an omnipresent narrator, and you choose whether or not to follow those instructions. What makes *The Stanley Parable* so unique, however, is that unlike all the narrative based games mentioned up until now, *The Stanley Parable* has no river. Despite the specific instruction, there is no guiding force propelling the player to a predetermined destination, because the choice to disobey is just

as meaningful as the choice to obey. Almost every time you make a choice, the game legitimately branches the narrative, taking you on a completely different adventure than if you'd chosen another option, regardless of what the narrator wants or says will happen. The choices you make actually determine where you will end up, and each branching path is independent of the others. There are nineteen different endings – nineteen different, self-contained stories that work by themselves and in conjunction with the others, and each one is told in a completely different way. To steal the phrase from Haley Williams, I believe that *The Stanley Parable* "is possibly the first game that has come through on the promise of a truly branching, reactive storyline" (Williams), not *Detroit: Become Human*. It is an actual example of the player and the designer, working together in conversation with each other, building the game and the story in collaboration through meaningful choice, honoring the commitments that the player makes in the game, to create a feeling of real agency.

However, *The Stanley Parable* specifically comments on the tension between authorial intent and player desire. The choices themselves are always choosing to listen to what the narrator tells you to do or to refuse. To obey or disobey the author's instructions. Paulley suggests that the author is using a narrator to symbolize themself and Stanley to symbolize the player, and that "the narrative and design of the choices in The Stanley Parable are used to explore the idea of player agency interfering with the author's intent" (Paulley 25). He brings up a particular interaction between the narrator and Stanley. If the player defies the narrator enough on a certain run, they'll hear a certain line of dialogue: "there's someone you've been neglecting Stanley, someone you've forgotten about. Please stop trying to make every decision about yourself." He goes on to tell Stanley that he had a plan, but that because Stanley wanted to disobey and make their own decisions, they ruined it. And when the player is obedient enough, following all the narrator's instructions, the narrator will encourage Stanley, telling him how satisfied he is because the story played out the way he'd intended. It's a way of saying that when the player makes playing the game about having the experience they want and doesn't care about what the author intended, and they can get so off track from the narrative the author wanted to tell that they make it "impossible to reconcile the story" (Paulley 25), and the game loses whatever meaning and significance it was trying to have. At the same time, the experience of playing the game entirely obediently is fairly dissatisfying

for the player. It's funny the ending associated with this is named the "Freedom Ending," because even when Stanley escapes the office building in this ending and gets his own freedom, the player hasn't been able to make a choice for themself once, only obeying the instructions of the author. It asks whether choice really has any meaning in games if players are just going to play the game how they intend and designers are just going to force the stories they want to tell on players. It suggests there needs to be a balance of player desire and authorial intent, the player and designer working together to build a story they're both committed to. But at its core, *The Stanley Parable* is a great example of a game that facilitates agency really well, even while making this meta, self aware commentary that is so unique to the experience.

2.3 Storytelling

The major stylistic, thematic and storytelling inspiration for the project is the critically acclaimed Fullbright game, designed by Steve Gaynor, *Gone Home*. *Gone Home* puts the player in the place of someone visiting their childhood home, exploring a story through that space. While it doesn't really give the player meaningful choice, instead opting for a fairly linear experience, it does masterfully tell an intimate, compelling, deep story through the use of things like environmental storytelling and voice-over, which are both things that I drew from heavily. The game's website frames the story like this:

> You arrive home after a year abroad. You expect your family to greet you, but the house is empty. Something's not right. Where is everyone? And what's happened here? (Fullbright)

When the player is placed in the world, at first they are just presented with an empty house. As you play and move through that house, more of the story is unveiled to you, and you get a better sense of the people that live in the house – you're able to piece together the story through static objects in the world that exist to give background and exposition on the characters and build their identity in the game. Think about the kinds of things you would learn about someone from walking through their room and seeing what was on their walls. There are also objects the player can interact with and hear voice-over about, which can also give exposition and character, but are also often used to move the narrative at hand forward. Environmental storytelling and voice-over feel like the most natural, most immersive, ways to direct the player and tell the story. They may, however, need to be supplemented using other means of direction and storytelling.

2.4 Background and Literature Review Conclusion

Meaningful choice and agency in video games are things that it's hard to get right, but it definitely feels possible to create an experience that does actually create that feeling of agency, like *The Stanley Parable* does. It's important that the player and designer can both commit to the game and the choices they make. While *The Stanley Parable* is a great example of a game that creates agency through choice, it does that by highlighting the tension between game designer and player, and not how they can work together to make meaningfully satisfying experiences. By using choice meaningfully and understanding the pitfalls of choice in modern video games, we can understand how to build a game that gives the player a feeling of agency – something like *The Stanley Parable,* but emphasizes how they can work together to form cohesive, meaningful narratives that the player feels like they built as much as the developer through player agency and choice. We can also tell stories through environments and immersive means, that don't take the player out of the experience to tell them what's going on.

3. Methods

3.1 Technologies

i. *Unity 2020.3.19f1, the Unity Hub, and the Unity Editor:* Unity was used as this project's game engine. Unity is designed to be a fairly simple game engine to work in, providing solutions for 2D, 3D and XR games. It's free, and there are a plethora of resources on Unity development. Code in Unity is written in C#. Other considerations were Unreal Engine and Python based game engines, Panda3D and Ren'Py – which were both under-documented and not built for the kind of project I wanted to make. While Unreal is heavily documented, like Unity, Unreal uses C++, which has a reputation for being slightly harder than C#. Unreal is also mainly advantageous over Unity if you are looking for graphical fidelity, which I wasn't really concerned with in this project. I chose to use Unity after taking all options into consideration, but I believe the project could work just as well in Unreal or another game engine if the programmer is more comfortable in that engine. The Unity Editor is the program that unity games are made in, and the Unity Hub is the program used to install and launch versions of the Unity Editor. The specific version of Unity used in this project was 2020.3.19f1.

ii. *JetBrains Rider:* Rider is the IDE I used to write code and scripts for Unity. It provides code completion and built-in integration with the Unity Editor. There were two main options, Rider or Visual Studio, and I chose Rider because of my familiarity with JetBrains products. Either, or another of the programmer's choice, will work fine, this decision was all about my personal preference.

iii. *Sweet Home 3D:* Sweet Home 3D is a floor planning program that was used to create the lighthouse and shop in the game. Sweet Home 3D lets the designer create a floorplan on a 2D grid, and then translates that into a 3D model of a building. The program has its limitations, but was serviceable for what I needed. The models struggle with scale and lighting in Unity, making them unusable in a production quality game. If I'd had more time, I would have wanted to look more into modeling buildings, and I would have scrapped the models from Sweet Home 3D in favor of custom models. However, look was less important to me than the way the game played, and this was not a priority, which meant the buildings from Sweet Home 3D served their purpose fine.

iv. *Blender:* Blender is an open source 3D modeling software. Blender was used to both create original assets and edit existing assets. It works well for this purpose, but the UI and hotkeys can be cumbersome. If I had an infinite budget, I might have used a professional 3D modeling program like Maya or 3D Max. But for a project like this, those are prohibitively expensive, and Blender can produce great quality models if you know how to work with it.

v. *Adobe Photoshop and Illustrator:* Adobe's Creative Cloud apps were integral in creation of assets. All custom 2D assets were created using a combination of these two programs. Illustrator was used mainly to create graphics from scratch, while Photoshop was used to edit things like alpha channels and create the splash screen sequence using the basic timeline features in Photoshop.

vi. *NormalMap-Online and IMGonline Make Seamless Texture Online:* these two websites were used to create seamless textures for surfaces like the walls and floors in the game. NormalMap-Online creates normal maps — texture height data — that can be used in Unity to create the illusion of a 3D surface. Make Seamless Texture Online is used to create seamless, tileable textures — textures that can be placed next to each other over and

over without visible edges — from an image that isn't tileable.

vii. *Character Movement Fundamentals:* this is a package on the Unity Asset Store that provided the base for character movement in the game. It provides the designer with prefabs (precreated assets in Unity) for first- and third-person character movement, with scripts that are easy enough to understand and edit. For my project, I edited these scripts to create a way to freeze the player's camera and movement, as well as editing their provided gamepad control scripts to work correctly with the project. This package was used mostly because it was something I already had access to, and it worked well. If I was making this a fully fledged release, I would want to create my own character control scripts — this package uses the legacy input manager in Unity, rather than the newer Input System package. This made individual keypresses — as opposed to holding the key — more difficult to detect, and meant I needed a lot of workarounds in my code.

3.2 Game Overview

Branches follows the story of a lighthouse keeper on an island, returning to the lighthouse after having been gone for some time. When the player is given control, you have arrived on the island and your first objective is to get back to the lighthouse. After you make the trek up there, you find a note from Lou. Lou is the shopkeeper on the island who was looking after the lighthouse while you were gone. The note tells you that after you take a nap, you should either come see him to get more gas for the beacon before it runs out or go light the beacon before it gets too late and see him after. Either way, the idea is that player has to make a decision about which they want to do first, and based on that first choice, the rest of the game will unfold.

If the player chooses to go up to light the light first, when the keeper arrives at the top of the lighthouse he realizes that there is a ship coming for the shore, but that they can't see anything because the beacon is broken and they cannot tell a dark island is in front of them. From here, the player is given the choice of either trying to fix the light, or of trying to find an alternate way of signaling the ship. If they choose to fix the light, they can either go down to the basement and bring up tools, or they can go into town to try to get Lou to fix it. If they get tools, they are successful and the light is repaired — the "safe" ending. If they run down to Lou, it takes too long and the ship crashes — the "I messed up" ending. If they choose

to signal them another way, the player can either go down to the boat they came in on to signal with the loud fog horn, or they can pour gas through the lighthouse and use it as a massive torch. If they use the foghorn, the ship also crashes — the "there was nothing I could do" ending. And if you choose to light the tower ablaze, the ship is able to see the lighthouse go up and can turn around — the "at what cost" ending. While I give the endings these names here, one isn't necessarily a good or bad ending, they're different. This is more a way to categorize the feeling I tried to create with each ending.

Originally, the plan had been to implement both initial choices, going into town to get the gas and lighting the light first. The largest cut during development was the decision not to implement going into town to get the gas. In the end, it was simply too much to do in the time that I had, and I needed to cut the number of branches I was designing and developing in half in order to get it done. This was the hardest decision to make, and I am still sad I was unable to do every branch I had hoped to. Many of the elements of the cut content are still visible in the final build, with the main one being the design of the island. The cut branches had a lot more island exploration, with an entire side of the island sculpted and designed as though it would be used. There is a path that leads to a cave and a beach on the far end of the island, which is where the boat would have crashed. If the player had chosen to get gas first, the boat would have crashed into the island, and the player would have been tasked with dealing with the fallout of the crash. With more time, I absolutely would have finished implementing the story I designed.

3.3 Story Design

In order to actually begin the process of creating this project, the first step was ideation. This was a long process. I went through a number of revisions of what I actually wanted to do before I decided on this story, and finding a way to tell a meaningful story became quite difficult. I needed to find a thrust for the player that could lead them in a lot of different directions — initially I'd wanted to have a huge amount of endings, somewhere between ten and fifteen — so finding a way to start a story like that took a good deal of time. I ended up deciding between the lighthouse story and a space exploration story. Both would have tried to create that feeling of agency and give the player meaningful choice, but in the end I felt like it would be easier to ground the experience, as well as more meaningful, if it took place in a more real and familiar location. From the outset, the

setting and lighthouse itself were inspired by the Outer Banks in North Carolina, a place I've vacationed with family many times, but the game's actual location ended up being quite different.

From there, I needed to figure out the narrative mechanics for how the story would branch. Would all choices exist within the same world or would new, distinct and separate worlds be created at each choice? Would the timeline of the story remain consistent regardless of the choices the player made? It was a complicated question, and both options had different implications.

Consistent Timeline	*Changing Timeline*
The keeper climbs the tower to light the light and looks out to see a ship on the water, about to crash into the beach.	The keeper climbs the tower to light the light and looks out to see a ship on the water, about to crash into the beach.
or	*or*
The keeper goes into town, and while there, a ship crashes into the beach on the other side of the island.	The keeper goes into town and no ship approaches the island

Ex 3.3.1 — a look at an example of differences in a story with a consistent timeline and a changing timeline. In a changing timeline, the ship doesn't exist in every branch, only in the one where he notices it.

The decision was eventually made to keep the world consistent. This is different from *The Stanley Parable*, which creates strange, different worlds at every turn. It was a question, I think, of most effectively upholding the commitment the player makes with their choices. *The Stanley Parable* feels wacky and unpredictable because you never know what your choices could lead to – and the consequences of your actions become, in a way, unpredictable. Playing through the game more than once, knowing the world could throw anything at you without any consistency, is a small way of undermining the commitment that the player makes. I wanted to avoid doing that, which meant consistency in the timeline of events was important.

The problem with making the timeline consistent was that it made the game more beholden to the flow of the river. The way to combat this was to make the game centered less around those events that remain consistent, and more around the actions that the player takes and the decisions that they make. I hope that this was the right decision, but I am not completely sure it was. I do worry that the game I've designed is still centered too heavily around the ship crashing into the shore. I feel I've been able to mitigate that with the strategy above, and also by making the endings very separate things. With more time, I'd have made the branches stretch out further and have

significantly more decisions for the player to make, to be able to get the player further away from the central events of the timeline.

The next stage was coming up with a lot of the background and context for the story. The game starts with the Keeper coming back to the island, and in the story, he's just returned from helping his late wife's parents move into a new home. I made character bios for the Keeper, his wife, and Lou, as well as a short description of the island and its history, all of which informed the tone for the project. I started to think about time and time period, and how things may have changed over time for the Keeper. I thought about the ways that the keeper's choices and actions would have affected and impacted his life, and then tried to make branches that would feel significant to that character. For example, in my exposition and character background the Keeper lost his wife in a house fire. While he was off at sea, a fire started in their home and took the life of his wife and daughter. After this, the Keeper swore he would watch over the island, becoming its ever vigilant guardian. He sees it as his responsibility to protect the people of the island and the boats coming in at all costs. So when he's faced with starting another fire in his lighthouse to save a ship, it should be a difficult decision for him, given his past and his promise to protect the island.

Due mostly to time constraints, very little of the world building made it overtly into the final game. There are some places where environmental storytelling is used to get across some of those ideas, like the picture above the fireplace or the picture frames on tables, but much of what this expository process did was inform how the game should look and feel, and how the story's branches should play out. This did, unfortunately, mean that a lot of content had to be cut. I wanted to include a lot more environmental storytelling, with passive ways to tell the player more background and exposition without shoving it directly under their nose. And with more time, I would also include objects around the world that you could interact with, but that don't move the story forward – things you can pick up and look at to get more insight into the characters and history in the story. But even without those, the amount of environmental storytelling the game *does* have starts to build a coherent and interesting picture.

Once the rules of the world and the background were set, I started to work through the various branches, tweaking their stories here and there as I went. Each story path was designed so that it

would feel distinct and have its own identity, and the endings were each written so that they would elicit a certain feeling or emotion from the player. However, the different branches and endings weren't actually fully finalized and realized until well into building the game, which slowed things down in a lot of places.

Doing the project again, I would have made sure that I had given the story a full treatment and had everything planned out, and that I understood all of the endings and story beats I wanted before I started coding the game at all. When I got into actual story scripting and the story wasn't done, it became much harder to move forward, and I found myself held back by not knowing exactly what was going to happen next. This project would have benefited greatly from more time and focus in the ideation stage.

3.4 Gameplay Design

At the same time as I was writing the story, I moved onto building demos of some of the necessary game systems and making the initial test builds of the project. This step resulted in one of the biggest mistakes that I made in the entire project: I assumed that my demo scripts would work in a larger game, which they categorically did not. This did, however, provide the first prototype of the lighthouse model, which went on to be edited and tweaked a number of other times in order to better fit the game and the events that took place inside it.

With a demo done, I decided to move onto the terrain for the island. This took an incredibly long time, and I thought a lot about the design of the mountain, the cliffs, the location of the town and a number of other factors, and began to build my world in a way that I would be able to guide the player using the natural landscape. I wanted the player to feel like they knew where they were supposed to be moving to next in the world, which I did through the creation of natural paths and barriers, with treelines and fences put up to cut off player access in a natural way. The terrain needed to separate the parts of the game, and long paths and distinct land features made clear distinctions between them. I wanted the player to feel like when they entered a new space, they were somewhere new and unique, which I believe I achieved. Next, the lighthouse and other prop assets were finally added to the terrain – things like trees, rocks, fences, lamp-posts, etc. – and after a very long build up, I was finally going to begin scripting the story and the rest of the game systems.

I needed to figure out how to interact with objects and trigger events, and eventually I landed on the idea of highlighting them with a white outline. Initially, I'd planned to not have the objects highlighted when looked at, but it became clear that it was really confusing what could be interacted with and what couldn't after user testing. The prompt would come up, but there was no indication it had become interactable. So I started looking into ways to highlight the object and found that putting a white outline along with the text prompt made it the most clear that it was interactable.

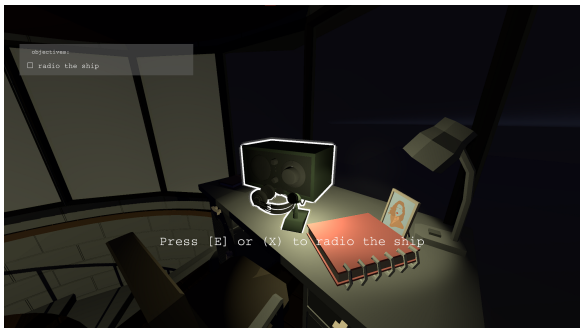

Fig 3.4.1 — see the white outline around objects you can interact with.

Another decision that made an enormous positive impact on the overall game was to include voice-acting. Again, initially I'd wanted to omit dialogue and voice-acting and rely entirely on environmental storytelling, but the game felt incredibly empty and bare, especially because I couldn't get to so much of that environmental storytelling. I worked with my dad to record lines for the Keeper in the game, which was a challenge when living across the country. I set up a webpage that would mostly function as the game's script, at https://nickesc.github.io/Branches-Script. On it, the page would list the lines for each character from a database, with context and notes for where the line appeared in the game and how it should sound. At the bottom of the page, I added a form where I could push further lines to the script database, and new lines would appear at the top of the list on the webpage. I wanted the voice-over to sound high-quality, and guided him through how to record using a nice microphone I'd left him and in a controlled environment. As lines of dialogue are heard in the game, the subtitle system also provides a visual way to get the words across to the player, and serves not only as an important accessibility feature, but it makes it easier to follow along with the dialogue. I think voice-acting was the most important part of making my game feel like an actual enjoyable experience, rather than just random images on the screen. Without voice-acting, the game felt empty and directionless, and having that voice alongside you really added to the connection felt to the character, and the level of investment and immersion in the game.

The next important aspect of the game was the objective box. Before implementing this, the feedback I received was that players didn't know what they were supposed to do. They didn't know what the current options for them to choose from were, even if the audio listed them, and knowing those choices was extremely important to creating the feeling of agency. This meant the player needed a way to keep track of those objectives after getting the information from the dialogue. The objective box just lists the current objective(s) in the top coroner to the player, actively updating based on where they are in the game. If there is only one thing for the player to do, it just shows that one thing in the list. If there are multiple options for what to do next, they will appear listed (each on a new line) as "x or y." Originally, it didn't have the word "or" between the two options, and I found people didn't realize they could choose either and that they weren't meant to be done one after another. After adding "or," I saw a much larger variation in which paths people took, which meant that people now saw that they had more choices. With more time, I would have preferred to use other means of directing the player and showing them what their options were, but with the time I had I was content to use a UI element.

Lighting was another important factor to the game's tone and feeling of direction. Initially, the lighthouse was evenly, flatly lit. The lighting made everything bright and fully viewable, but it made it confusing which rooms you were supposed to go in and where you were supposed to go next. One play-tester mentioned that I should look at *Portal 2*, which does an amazing job with directing the player with lighting. Looking at examples from that game, I made much harsher lighting that directed the experience, and helped make it much clearer where the player was being guided and what the player should focus on.

A number of other decisions were made that were important to getting the ideas in the game across to the player — for instance, the inclusion of an intro paragraph at the beginning of the game with a little exposition helped to show the player why they were on the island and to prep them for the rest of the story. Some of the doors in the lighthouse were also confusing to some testers, so I made sure that doors were only open when they were supposed to be in the story. I also had people tell me that the controls were confusing, especially because it shows both inputs every time, so I made a graphic shown at the beginning of the game to show controls and alleviate confusion.

Overall, the biggest issue during the actual development was that the story design was not as thorough or thought out as I had wanted it to be. Its design process happened as the game was being made in a lot of cases, which resulted in a lot of challenges and slow downs, but also in a way made the production fluid and easy to change based on feedback. The entire process likely would have greatly benefited from more time spent on design before I started coding, especially in terms of how the player would interact with the world and where the story was going.

4. Architecture Overview

4.1 Unity Overview

For this project, as laid out in the methods section, I used the Unity game engine. Unity allows for a lot of different workflows, and its interface is designed in a way that lets both new users and veterans, programmers and artists, build games without much technical knowledge. The more technical knowledge you have, however, the better your game will probably look and run.

Unity projects are built mainly using Scenes, GameObjects, Components and Scripts. Scenes are the 3D or 2D spaces that the player is placed in in the game. Inside each scene, there is a hierarchy of GameObjects, the actual things in the game in that space — this can be anything from a wall, which the player can see and walk into, to an audio source, which is invisible to the player, but plays audio in the scene when active. Each of these objects has Components that make them up, which tell the GameObject what it is and what it does. Every GameObject has a 'transform' component, which tells it where the object exists in space, and they can have things like Mesh Renderers — which tell the GameObject what it looks like and how to display it — and Colliders — which tells the GameObject whether or not another GameObject is touching it. These Components can also be scripts written by the developer, which are custom programs and definitions of behavior written in C# for the game and attached to a GameObject. These scripts are at the core of Unity development, and they let the developer do things beyond what basic GameObjects and Components allow.

4.2 MngrScript and Finite State Machines

The heart of *Branches* is the finite-state-machine class, `StateMachine`, and the game's central-control script, `MngrScript.cs`. `MngrScript.cs` defines a number of classes, interfaces and functions, including its titular class, `MngrScript`, which is a singleton — a persistent GameObject that must exist once and only once in every scene of the

project — created at runtime that houses the major systems of the game and helps manage the current state of the game at all times in conjunction with `StateMachine`. Also defined at the top of `MngrScript.cs` is the `IState` interface, an abstract class implemented by each state:

```
public interface IState {
  public string getName();
  public void Enter();
  public void Execute();
  public void Exit();
}
```

It requires each state to have four functions:

i. `getName()`*:* returns a string with the name of the current state
ii. `Enter()`*:* called once when StateMachine enters the state and sets up whatever is needed at the start of the state
iii. `Execute()`*:* called once every frame that the state is active
iv. `Exit()`*:* called once when StateMachine exits the state and prepares it to move onto the next one

A number of states that implement the `IState` interface for each section and branch of the game are defined in `MngrScript.cs` as well.

Below the interface definition is the `StateMachine` class:

```
public class StateMachine {
  public IState currentState;
  public void ChangeState(IState newState) {
    if (currentState != null) {
      currentState.Exit();
    }
    currentState = newState;
    currentState.Enter();
  }
  public void Update() {
    if (currentState != null) {
      currentState.Execute();
    }
  }
}
```

`StateMachine` holds the active state in the `currentState` variable, and defines a `ChangeState()` function that calls the active state's `Exit()` function, sets the active state to a new state, and calls the new state's `Enter()` function. `StateMachine` also defines an `Update()` function, which calls the active state's `Execute()` function every frame. `MngrScript` also has a public function, `getCurrentState()`, which returns the value from the current state's `getName()` function. Story scripts — scripts that progress the story forward rather than implement a game system — attached to GameObjects across the game use this `getCurrentState()` function and a number of public boolean properties in `MngrScript` to determine whether or not certain game conditions have been met. States in `StateMachine` check the properties in `MngrScript` to see if they should call

`ChangeState()` and initiate a state change. Much of the code is, unfortunately, somewhat messy. Everything works correctly, but there are unnecessary variables and methods all over. With more time, I would have gone back and cleaned up a lot of `MngrScript.cs`, but as the end drew closer, I became less concerned with maintainable code and more concerned with just getting it done.

4.3 States in Branches

Branches has 20 total game states. A few of those states are for the introduction, then they branch based on the player's decisions:

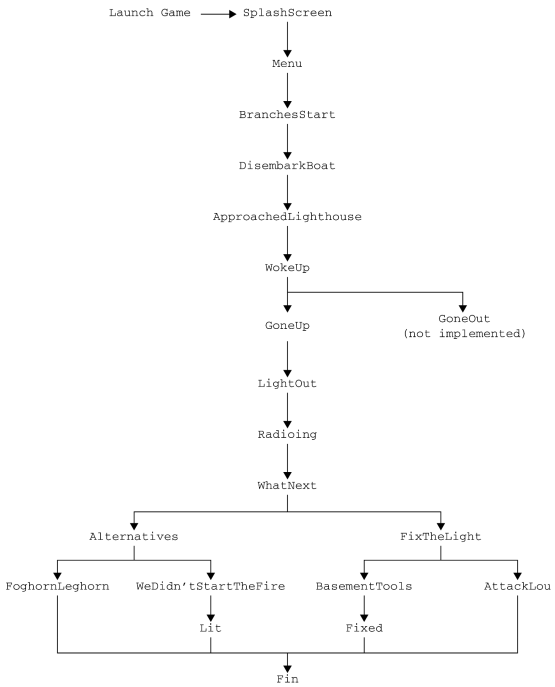

Fig 4.3.1 — all available states and transitions in Branches. All games initiate into `SplashScreen` *at launch, and end with* `Fin`*, which initiates the ending sequence and pushes the end credits to the screen.*

State transitions are mostly initiated in game through user interactions with game objects or the user crossing boundaries. For example, in the `FixTheLight` state, when the user presses 'E' to pick up the toolbox, the `grabTools` script sets the boolean property `ChoosingFix` in `MngrScript` to `false`, which indicates that the user is no longer choosing between ways to fix the lighthouse, and tells the `FixTheLight` state to transition to the `BasementTools` state.

4.4 Game Systems

In order to achieve consistent functionality through the game, I created a number of game systems for *Branches*.

Branches Game Systems:

i. *general UI systems:* allows the developer to call a number of functions to set the general text prompt on screen, push an image to the screen or blackout the screen

ii. *game freeze systems:* allows the developer to freeze the player's feet and or mouse, which can be exited either with a designated escape key or through code. Used to stop the player moving when things popup on screen and

prompt the user for a key before continuing, as well as during the ending sequence. One freeze system is used to pause the game with the 'Escape' key, and lets the user close all popups.

iii. *objective system:* allows the developer at any point to get, set, and check off the objectives in the objective box. The developer can choose to set a single objective or two objective options for the user to choose between. The developer can also clear the objectives without setting or checking off, allowing a great deal of flexibility within the system, as long as the developer keeps the logic associated with checking off objectives straight.

Fig 4.4.1 — In this screenshot, you can see many of the game's various UI systems: a text prompt at the bottom, an image pushed to the screen in the middle and the objective box in the top left corner, all in use, working together.

iv. *subtitle and voice acting audio system:* allows the developer at any point in the game to push an audio file to a queue, along with a subtitle, which will be played and displayed on screen in the order they are added. Also lets the developer choose if the subtitle should appear as a sound effect or as a line of dialogue or as a thought, letting them push multiple kinds of subtitles to the queue as well and formatting them correctly, and allows the developer to check if there is audio playing at any time, which is used for state transitions and story sequences.

v. *wait system:* allows the developer to call a wait function and indicate that the game is waiting from the state class, which cannot call coroutines. It is limited, but works with some state transitions and the subtitle system.

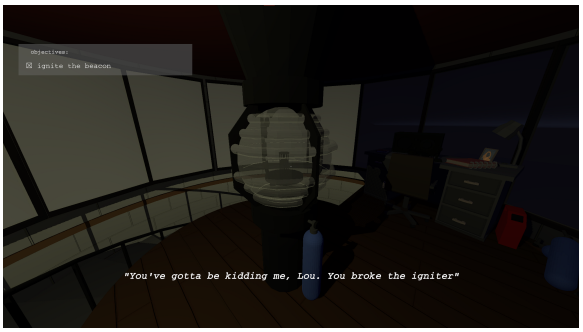

Fig 4.4.2 — In this screenshot you can see the subtitle system in use; at the time the screenshot was taken, the game was also waiting for the voice acting to stop playing in order to move on to the next objective.

5. Ethical Considerations

Though the ethical concerns with this project are limited, because strictly speaking it doesn't "do" much, it was important to keep a few things in mind while I was making the game, namely: accessibility, attribution, and representation. For accessibility, confining users to mouse and keyboard inputs limits the number of people that could play the game. For someone with a motor impairment, a mouse and keyboard's big movements and inputs can be very difficult, so I wanted to look into optionally using a gamepad to control the game. After spending some time with Unity's Input Manager and editing the movement scripts I was using, I had basic controller support. Through development, I worked to make that support better and to ensure that nothing I did in the game broke that support. I also wanted to make sure the game didn't need audio to be played, and including subtitles alongside the lines of dialogue and sound effects was integral to that.

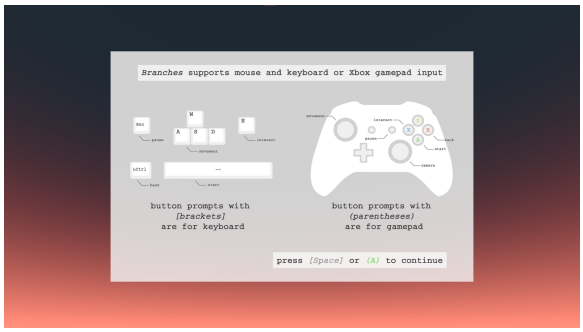

Fig 5.1 — the controls displayed at the beginning of the game, showing different control schemes.

Also important was attribution. I found a lot of the assets I used online, some on platforms like SketchFab or CGTrader, so keeping track of all of them was important in order to give them proper attribution at the end. I didn't want to use an asset without documenting it, and I didn't want to use assets I didn't have the right license for, and I ended up with a long list of sources at the end. This is accessible through the project's GitHub at https://github.com/nickesc/BranchesGame.

Finally, representation — as I was building the game, of course it was important to keep in mind cultural sensitivities and factors, and to avoid misrepresenting anything or playing into stereotypes and tropes. It was important to me that the Keeper be easy to project yourself onto, and part of that was not giving the keeper a particular nationality or race, or even appearance, so that as many people as possible could envision themselves as him.

6. Evaluation

6.1 Metrics

The goal for this game was to create something that felt like it facilitated the feeling of agency. In order to do that I decided that I needed to create a system that allowed for player choice, that reacted to that choice, and that upheld that commitment from the player through the rest of the game. I set

out to build a world that reacted to your choices, one where the story was defined not solely by the creator, but as a joint exercise. It needed to not only give the user choice, but also make those choices feel important and meaningful. The choices needed to be made freely, and without the idea that one was the correct choice or that one was superior. Choices needed to be distinct, and lead to distinct endings, which the user could feel like they had helped craft.

To evaluate the project and to determine whether or not I had achieved my goals, I monitored playtests and gave post-playtest interviews with 10 individuals, some of whom had given feedback on the game's development builds, and some of whom had not. Testers were given the following prompts and asked to respond:

> Please give your thoughts on how you felt about the choices, whether things were clear, whether you want more or less direction, if you feel like you're able to make choices freely and whether those choices feel meaningful.
>
> Important questions: Did you feel like your choices were your own? And did you feel like it mattered to the story what choices you made (did the story and world respond to your choices)?
>
> What do you wish was improved?

The evaluation was limited and somewhat informal, but focuses on the important aspects of whether or not choice is used effectively to create agency, and how it could have done better. I watched some playtesters play the game in person and some described their experience to me over the phone or through text. Testers played without any additional instruction from me or anyone else. Testers were interviewed immediately after playing. Some interviews were conducted verbally and some were conducted through text. Testers were pushed to give further explanation for their answers.

6.2 Results

The interviews and playtests showed this: Most importantly, everyone interviewed agreed that, at least on some level, their choices mattered significantly. Second, people picked a variety of paths, and no single path drew everyone. There was no most frequent path, which is a very good indication that users are choosing freely, and without influence from the designer. When pushed about the ideas of choice and agency, the responses showed that they felt like the choices were at the core of the game, and that they were important to how the story played out. Users who had seen the game before also commented on how much the additions of things like lighting, the

objectives box, and the object outline helped them engage with the story — before these things were included, they felt it was vague and that they didn't have a purpose. Once they were added, they understood what they were supposed to be doing in the game. Each helped solve problems related to getting the story and the choices across, which they said worked very well in the final build. Users also said, fairly universally, that they wanted more environmental storytelling, and to see more of the context and background that the world hinted at because they were interested in what they saw and wanted to see it expanded further. Together, all this tells me that while the game isn't perfect, it does succeed in creating a feeling of agency for the player, in which they are able to commit to their choices and see them reflected in the story and in the world, and that the story was still compelling, even with what little environmental storytelling there was.

7. Conclusion

This game was incredibly ambitious. To get to this point was an incredible effort and amount of work, and to expect twice that much to be done in the same time was unreasonable. It might have been doable if I could have dedicated all of my time to *Branches*, but I had three other classes with just as much work to do at the same time. The amount of content and the level of quality I expected to be able to produce were not realistic, especially for someone who was still learning how to use Unity at the same time. Through the process, I made a number of mistakes that I only saw the error in after-the-fact, because I did not have the experience to know better. This was the first game that I was building solo, the first time I was using Unity and the first time I was working in 3D – I expected a lot from myself.

That being said, I still think that my project was incredibly successful. While the game was not everything I had hoped it would be, the purpose of the project was *not* to make a perfect, ultra-high quality, completely finished game. The purpose was to make a game that created a feeling of agency through choice, which user testing indicates was still successful, even with all of the cuts to the story. In the future, I would love to finish this game fully and add the rest of the endings, but I think they would be supplementary, not necessary additions. While I didn't achieve exactly what I planned to, the final product was still a system that was able to create the feeling of agency, and with that I am satisfied.

8. References

Bethesda Game Studios. *The Elder Scrolls V: Skyrim*. Windows PC version, Bethesda Softworks, 2011.

Feral Interactive. *Life is Strange - Episode 1*. Windows PC version, Square Enix, 2015.

Fullbright, and Steve Gaynor. *Gone Home*. Windows PC version, Fullbright, 2013.

Fullbright. "Gone Home: A Story Exploration Video Game." *Gone Home: A Story Exploration Video Game*, Fullbright, 2013, https://gonehome.com/.

Galactic Cafe. *The Stanley Parable*. Windows PC version, Galactic Cafe, 2013.

Geoffrey, Gorisse, et al. "First- and Third-Person Perspectives in Immersive Virtual Environments: Presence and Performance Analysis of Embodied Users." *Frontiers in Robotics and AI*, vol. 4, 2017. *Frontiers Media SA*, doi:10.3389/frobt.2017.00033.

Paulley, Tyler. *The Official Rulebook for Choice in Video Games: An Examination of Choice in Modern Narrative Games*. Undergraduate thesis, 19 December 2017, Bellarmine University, Louisville, KY. *ScholarWorks@Bellarmine*, https://scholarworks.bellarmine.edu/cgi/viewcontent.cgi?article=1031&context=ugrad_theses.

People Make Games. "The System Behind Hades' Astounding Dialogue." *YouTube*, 19 November 2020, https://www.youtube.com/watch?v=bwdYL0KFA_U.

Quantic Dream. *Detroit: Become Human*. PS4 version, Sony Interactive Entertainment, 2018.

Roettl, Johanna, and Ralf Terlutter. "The same video game in 2D, 3D or virtual reality – How does technology impact game evaluation and brand placements?" *PLOS ONE*, vol. 13, no. 7, 2018, pp. 1-24. *Public Library of Science*, https://doi.org/10.1371/journal.pone.0200724.

Tanenbaum, Karen, and Theresa J. Tanenbaum. "Commitment to Meaning: A Reframing of Agency in Games." *eScholarship*, UC Irvine: Digital Arts and Culture 2009, 12 December 2009, https://escholarship.org/uc/item/6f49r74n.

Valve Corporation. *Portal 2*. Windows PC version, Valve Corporation, 2011.

Williams, Hayley. "In Detroit: Become Human Your Choices Finally Matter." *Kotaku Australia*, 23 July 2018, https://www.kotaku.com.au/2018/07/detroit-become-human-is-the-first-game-where-your-choices-actually-matter/.