Branches: A Game of Choice and Agency

N. Escobar

December 14th, 2021

GitHub:

https://github.com/nickesc/BranchesGame

Download the final build here:

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

1. Introduction:

For a long time, I've been very interested in the intersection of art and computer science. Going into this project, I knew that I wanted to do something that heavily took from both fields, something that blended the two together. The obvious choice for me was to create a video game. Video games, as an art-form, are so incredibly unique, because of the 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, do it without it really meaning anything. 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 in the same place.

So I decided I wanted to make something that made that choice feel meaningful. I decided that I want to make something that would facilitate a feeling of agency. To that end, I chose to make a branching-narrative based, 3D 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 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 they would feel like they have consequences. Consequences — as an aside — just means that there is 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 had 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 that works together in conjunction with the other branching paths to create something that is more than just the sum of its parts.

2. Background:

A certain amount of background on video games is necessary for understanding this project to the full extent. A narrative based game is one in which story is at the core of the experience, and the player goes through certain beats to experience that story, so a branching-narrative based game is one in which 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 them unique among other media. Most media is a straight path from beginning to end, with no deviation between different consumptions. 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 becomes 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 and agency are used differently in different games, but 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 a narrative-based game. Meaningful choice and agency 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. Literature Review

2.1 Choice and Agency

There are a lot of opinions on choice in video games, but papers from Tyler Paulley and Josua and Karen Tanenbaum give us some good ideas about how to use choice — and create agency — effectively.

2.1.1 Agency

Agency is the idea that the Tanenbaums are mostly concerned with. They propose this definition of agency: “the process by which participants in an interaction commit to meaning” (Tanenbaum and Tanenbaum 5). Commitment, which is at the heart of their definition, means that every time the player takes an action, they are committing to that action and the consequences it will bring. Asking players to commit means asking them not to interact with games passively and letting them happen to them, but rather being an active participant of creating the story and making the choices they’re given based on the story. When the author honors that choice the player made by making the story react with the expected consequences and tailoring it to the choices of the player — rather than hoping the player’s choices will align with the story — they give that commitment meaning. It’s the combination of commitment from the player and meaning given by the author 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 authors work in opposition. The author is not trying to limit the experience of the player, and the player is not an agent of chaos, consciously working against the intent of the author to break the game. The author isn’t telling a story that the player needs to receive, and the player isn’t trying to push the boundaries of the

imagined world. Instead, they build a world and a story together that both parties are able to commit to meaningfully.

2.1.2 How Choice is Used in Games

This begs the question of how choice should actually be presented to the players in game. Paulley looks at the ways that choice is used in a number of games and genres and comes up with a few things for the author to keep in mind while working with choice in games.

On RPGs like *Skyrim, Mass Effect,* and *Dragon Age,* Paulley argues that they use choice to make the player feel more invested in the narrative, making them feel a part of the narrative by giving them control over things like faction allegiances or which sidequests 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 sidequests in *Skyrim* has little impact on the narrative, and whether you’ve completed one or one-hundred, the story will always end 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 to the game aren’t upheld by the author. 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 Questline, there’s no acknowledgement outside that questline 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 questline, the world still treats the player for the most part as though they aren't in the guild. Skyrim fails to make those choices meaningful by ignoring the commitment of the player 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 it like this: "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. 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.

Telltale games, like *The Walking Dead*, and games like *Life is Strange* 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 plays out. 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 similar 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. It's a "one-size-fits-all" approach most of the time, where the game is a shell, moving from predetermined beat to predetermined beat every playthrough and filling in certain details based on the choices you made. The choices can't be meaningful, especially in a game that touts consequence of choice, because they don't really have an impact on how the story ends. It will keep flowing down the river to the same exact spot, no matter what choices you make.

Detroit: Become Human is another example of a game that is marketed around choice. It takes the idea of branching narrative to another level, but still suffers somewhat 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 pushes you in a direction, it still wants to follow a specific narrative, predetermined by the author. The choices you make don't really feel like you're building the story even if the choices themselves result in big consequences. And the number of choices also undercuts the feeling of importance you get from what meaningful choices there are, because the game is so saturated with meaningless choices that don't really have a big impact. The branching story feels more like they're just trying to fit in as much content as they can, and not focusing on making every bit of that content meaningful.

2.1.3 Superior Choice

One of the things Paulley seems to laud, not related to a 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, more 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 in reality all this does is undermine the initial options instead of making the choice more meaningful. It sounds 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, 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 to make a specific choice, again undermining the agency the player has by making them feel like they have to make a specific choice rather than committing to their decision on their own.

2.1.4 The Stanley Parable (2016)

Another very intriguing example of choice in games is *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. There is no guiding force, propelling the player to a predetermined destination, regardless of what the player does on the way. Almost every time the choice you make in the game legitimately branches the narrative, taking you on a completely different adventure than if you'd chosen another option. 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 completely different ways. To steal the phrase from Haley Williams, *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's an actual example of the player and the author, 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 to the game, to create a feeling of real agency. However, *The Stanley Parable* 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 the intention of the author themself" (Paulley 25) 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 this 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 he 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 doesn't care about what the author wants, and when they only care about having the experience they want, 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). At the same time, the experience of playing the game entirely obediently is fairly dissatisfying for the player. The ending is ironically 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 themselves 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 authors are just going to force the stories they want to tell on players. It suggests there needs to be a balance of player intent and authorial intent, the player and author working together to build a story they're both committed to — sound familiar? *The Stanley Parable* is a great example of a game that does this 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 game *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 much meaningful choice, it does masterfully tell an intimate, compelling, deep story through the use of things like environmental storytelling and voiceover, and those are both things that I drew from heavily. When the player is placed in the world at first, they are just presented with an empty house. As they play and move through that house, more of the story is unveiled to them, and they get a better sense of the people in the house, and you are 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), as well as objects the player can interact with and hear voiceover about, which can also give exposition and character, but are also often used to move the story forward. Environmental storytelling and voiceover feel like the most natural, potentially most immersive, ways to direct the player, and give instruction, and while they can be supplemented, finding ways to tell the story without taking the user out of the experience and reminding them it's a game feels like the most appealing way to go about it.

2.2 Conclusion

Meaningful choice and agency in video games are things that it's hard to get right, but with the information here, it definitely feels possible to create an experience, similar to *The Stanley Parable*, in which the player and author 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 author and player, and not how they can work together to make really satisfying experiences. By using choice meaningfully and understanding the pitfalls of choice in modern video games, we can look at how to build a game

like *The Stanley Parable,* one that gives the player agency, while at the same time emphasizing 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 they need to do next.

4. Methods:

4.1 Technologies:

Unity 2020.3.19f1, the Unity Hub, and the Unity Editor — Unity was used as this project's game engine. Unity is one of the easiest game engines to work in, and it's one of the simplest to use if you don't have a lot of experience, providing solutions for 2D, 3D and XR games. Code in Unity is written in C#, which makes learning how to code for Unity easier than some other game engines. It's free, and there are a near infinite number of resources on Unity development. Other considerations were various Python based game engines like Panda3D and Ren'Py, which were both underdocumetned and not built for the kind of project I wanted to make, and Unreal Engine 5. Unreal was the other main contender for the engine. While it is heavily documented, like Unity, Unreal uses C++. While that isn't terribly difficult to learn, I made the decision to make it slightly easier of myself and learn 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. Unity became the clear choice after taking all options into consideration, but I believe the project could work just as well in Unreal 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 I used in this project was 2020.3.19f1.

JetBrains Rider — Rider is the IDE I used to write scripts for Unity. It was one of two main options, Rider or Visual Studio, and I chose Rider because of my familiarity with JetBrains products. Either, or another of your choice, will work fine, this decision was all about personal preference.

Sweet Home 3D — Sweet Home 3D is a floor planning program that was used to create the buildings 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 an actual 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 my own, hopefully higher quality, creations. However, look was less important to me than the way the game played, and this was not a priority, so the buildings from Sweet Home 3D serve their purpose fine.

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 would have used a more intuitive 3D modeling program like Maya or 3D Max. But for a project like this, those are prohibitively expensive, and Blender works fine.

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 used to edit things like alpha channels and create the splash screen sequence using the basic timeline features in Photoshop.

NormalMap-Online and IMGonline Make Seamless Texture Online — these two websites were used to create seamless textures for things 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 it being obvious where edges are or that it repeats — from an image that isn't tileable.

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 feet movement, as well as editing their provided gamepad control scripts to work correctly with my project. This package was used mostly because I already owned it, and it worked well enough. If I was making this production ready, 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.

4.2 Game Overview

Branches follows the story of a lighthouse keeper on an island, returning to his lighthouse after helping his late wife's parents move. At the beginning of the game, you arrive on the island and are told to get back to your lighthouse. Upon arriving, you find a note from Lou, the shopkeeper on the island who was looking after the lighthouse while you were gone, which tells you that after you take a nap, you should either come see him to get more gas for the light before it runs out, or go light the light before it gets too late. Either way, the player has to make a decision about which they want to do first, and based on that first choice, the rest of the game unfolds.

Note for future development: at the moment, only the part of the story where you light the light first is implemented. In future versions, I would like to finish the story that I set out to make

When the keeper arrives at the top of the lighthouse, he realizes that there is a ship coming for the shore, and they can't see anything because the beacon is broken. 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 the light. 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 either goes down to the boat they came in on to use the loud foghorn, or they 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 turns around — the "at what cost" ending.

Development note: while I give the endings names here, one isn't necessarily a good or bad ending, they're different. This is more a way to categorize the feeling I try to create with each ending.

4.3 Story Design:

In order to create 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. Initially, the options were very broad, and 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 lighthouse and setting itself were inspired by the Outer Banks in North Carolina, though the game's location ended up being quite different.

From there, it was figuring out a way for this story to branch. I came up with the idea of the shipwreck fairly quickly, but from there, figuring out how to handle that, it became a lot harder. The first question I had to answer was how the world was going to work. Would the timeline of the story remain consistent regardless of the choices the player made? Would all choices exist within the same world or would new, distinct and separate worlds be created at each choice.

Consistent World	*Changing World*
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* The keeper goes into town, and while there a ship crashes into the beach on the other side of the island.	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* The keeper goes into town and no ship approaches the island

Ex 4.3.1 — a look at an example of differences in a story with a consistent world and a changing world. In a changing world, the ship doesn't exist in every timeline, only in the one where he notices it.

Once the rules of the world and the setting were set, I started to work through the various branches, tweaking their stories here and there as I went. The full branches weren't actually realized and finalized until well into building the game. Each story path was designed so that it would feel distinct and have its own identity.

Development note: This is one place where I wish I had spent more time. Doing the project again, I would have started by deciding exactly what the branches would be ahead of time. 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 from more time and focus in the ideation stage.

Also in this stage, I started thinking about exposition and characters for the game. I made character bios for the Keeper, his late wife, and Lou, as well as a short description of the island and its history, 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 and protect it, becoming it's ever vigilant protector. 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 to save a ship, it should be a difficult decision for him, given his past and his promise to protect the space.

While almost none of this world building appeared in the final product, it was incredibly important to figuring out the tone and story, and informed the way that I went about building the other parts of the game, like designing the lighthouse model and the layout of the island.

Development note: this was where a good deal of the content was cut. I wanted to include a lot of environmental storytelling, background, and exposition, using passive ways to show you character and background in the game, but due to time constraints most of it needed to be cut. A small amount, however, can be seen in the game through things like the picture above the fireplace or the picture frames on tables. With more time, I would include objects around the world that you could interact with 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. I would also spend a good deal more time thinking about potential pictures or text snippets you could find around the lighthouse which would give you the same thing.

4.4 Gameplay Design

After I had the setting and world done, I moved onto demoing the game and starting to build the systems used. This step had 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, which went on to be edited and tweaked a number of other times.

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 wown and a number of other factors, and bagan 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 more natural way. The terrain needed to separate the parts of the game, so I made long paths between the different locations on the island. I wanted to feel like going to a space was distinct, which I believe I achieved. Next, the lighthouse and other assets were finally added to the terrain, and after a very long build up, I was ready to start scripting in the game.

I needed to figure out how to interact with the objects and trigger events, which I did through the use of invisible walls and highlighting objects. 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 around it made it incredibly clear that it was interactable. Even so, I wanted to make sure that the player knew what was happening when they saw it, so I added a prompt the first time you see an outline to tell you what it is.

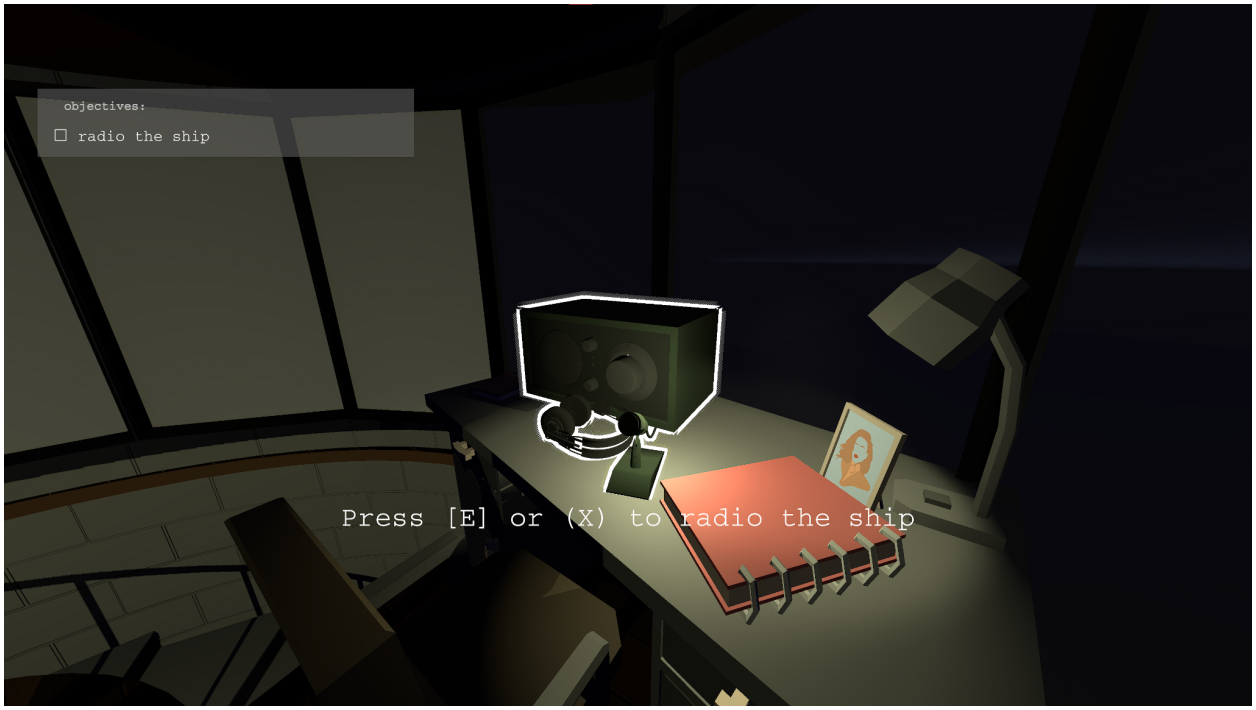

Fig 4.4.1 — see the white outline around objects you can interact with.

Another decision that made an enormous impact was to include voice over and dialogue. Again, initially I'd wanted to omit it and rely on environmental storytelling if I could, but the game felt empty and bare without any sound. I worked with my dad to record lines for the game, which was a challenge when living across the country. I set up a webpage (https://nickesc.github.io/Branches-Script) that I could push lines to, and he used that to record them. I wanted the lines to sound good, so I brought him a nice microphone and gave him lots of context and notes for how each line should sound. I think this was the most important part of making my game feel like an actual enjoyable experience, rather than just random images on the screen. This was done in conjunction with the subtitle system, which provided a visual way to get the audio across to the player.

The next, incredibly important, aspect was the objective blurb box. Before implementing this, the feedback I received was that players didn't know what to do. They didn't know what the current

options for them were, and knowing those was extremely important. So I added a way to keep track of those objectives after getting the information from the audio. Originally, it didn't have the "or" between the two options, and I found people didn't realize they could do either, and that they weren't ordered. After adding this, I saw a much larger variation in which paths people took, which I took to mean it was successful, as people realized they had more choice.

Lighting was another incredibly important factor. Initially, the lighthouse was evenly lit, and it made it confusing which rooms you were supposed to go in and which you weren't. One user mentioned that I should look at Portal 2, which does an amazing job with directing the player with lighting. Looking at some examples from that game, I made much harsher lighting that directed the experience, and helped make it much much clearer where the player was supposed to actually go.

A number of other decisions were incredibly important for getting the ideas in the game across to the player — 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 prep them for the rest of 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 to show controls and alleviate confusion. Also, the doors to the lighthouse became confusing, so I made sure that doors were only open when they should be.

5. Architecture Overview:

5.1 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, to build games without much technical know-how. The more technical knowhow 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, 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 and Filters — which tell the GameObject what it looks like and how to display it — and Colliders — which tells the GameObject whether or not something is touching it. These Components can also be scripts, 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 designer do things beyond what basic GameObjects and Components allow.

5.2 MngrScript and Finite State Machines

The heart of *Branches* is a finite-state-machine (FSM) StateMachine and script called MngrScript. MngrScript 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 for the game, as well as helps manages the current state of the game at all times in conjunction with StateMachine. At the top of the MngrScript file, it defines the interface for an IState, the abstract class implemented by each state. It requires each state to have i. a getName() function, ii. an Enter() function, iii. an Execute() function, and iv. an Exit() function. getName() returns the name of the current state (i.e. "ApproachedLighthouse"). Enter() is called once when StateMachine enters the state and sets up whatever things are needed for that state, Exit() is called once

when StateMachine exits the state and prepares it to move onto the next one, and Execute() is called once every frame that the state is active. MngrScript has a ChangeState() function that calls the first state's Exit() function, sets the current state to the new state, and calls the new state's Enter() function. 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 system — attached to GameObjects across the game use this getCurrentState() function and a number of public boolean properties in MngrScript to tell how they are supposed to behave in that moment, and as those properties toggle on or off they change their behavior accordingly. States in StateMachine check the properties of MngrScript to see if they should initiate a state change, which is the last command called in their Execute() function.

Note for future development: at the moment, MngrScript and related story scripts are fairly messy. They work, and work consistently, but it's hard to keep track of state changes and transitions, as they can be called from anywhere. I'd like to make all state transitions occur in the state's Execute() function, as having these spread out over story scripts throughout the project makes it incredibly hard to debug or change anything. In general, the code needs to be cleaned up and made consistent, especially the things I was writing at the beginning of the project when I was still learning how Unity and C# worked.

5.3 States in *Branches*

Branches has 20 total states (see Fig 5.3.1). Five of those states are for the introduction, then they branch based on the player's decisions.

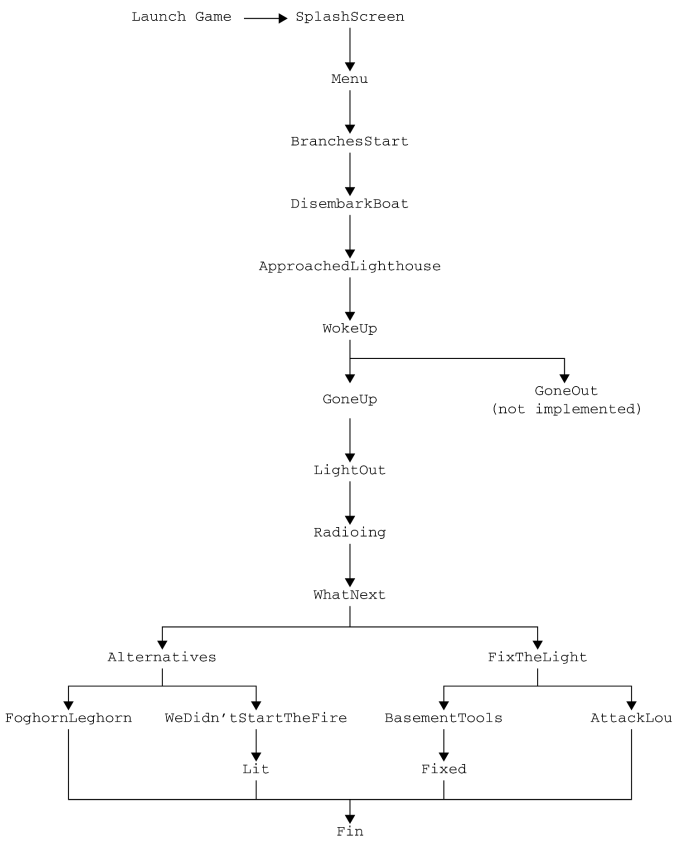

Fig 5.3.1 — all available states and transitions in Branches at this time. All games initiate into SplashScreen at launch, and end with Fin, which initiates the ending sequence and pushes the end credits to the screen.

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

5.4 Game Systems

In order to achieve consistent functionality, I had to set up a number of systems for *Branche.*

Branches Systems:

i. general UI system
    - allows the designer 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 designer to freeze the player's feet and or mouse, 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 freezer system is used to pause the game with the Escape key, and lets the user escape from all popups.

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

Fig 5.4.1 — In this screenshot, you can see the prompt UI, the image UI, and the objective box in use in the game, working together.

iv. subtitle and a voice acting audio system

- allows the designer 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 as they are received. Also lets the designer choose if the subtitle should appear as a sound effect or as a line of dialogue, letting them push sound effects with subtitles to the queue as well and formatting them correctly. Also allows the designer to check if there is audio playing at any time, which is used for state transitions and story sequences.

Fig 5.4.2 — In this screenshot you can see the subtitle system in use; also in that image, the game is waiting for the voice acting to stop playing to move on to the next objective.

v. wait system

- allows the designer 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.

6. Ethical Concerns:

Though the ethical concerns with this project are limited, it was important to keep a few things in mind while I was making the game: accessibility, attribution, and representation. For accessibility, it occurred to me that confining users to mouse and keyboard inputs was limiting 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 with the audio was integral to that.

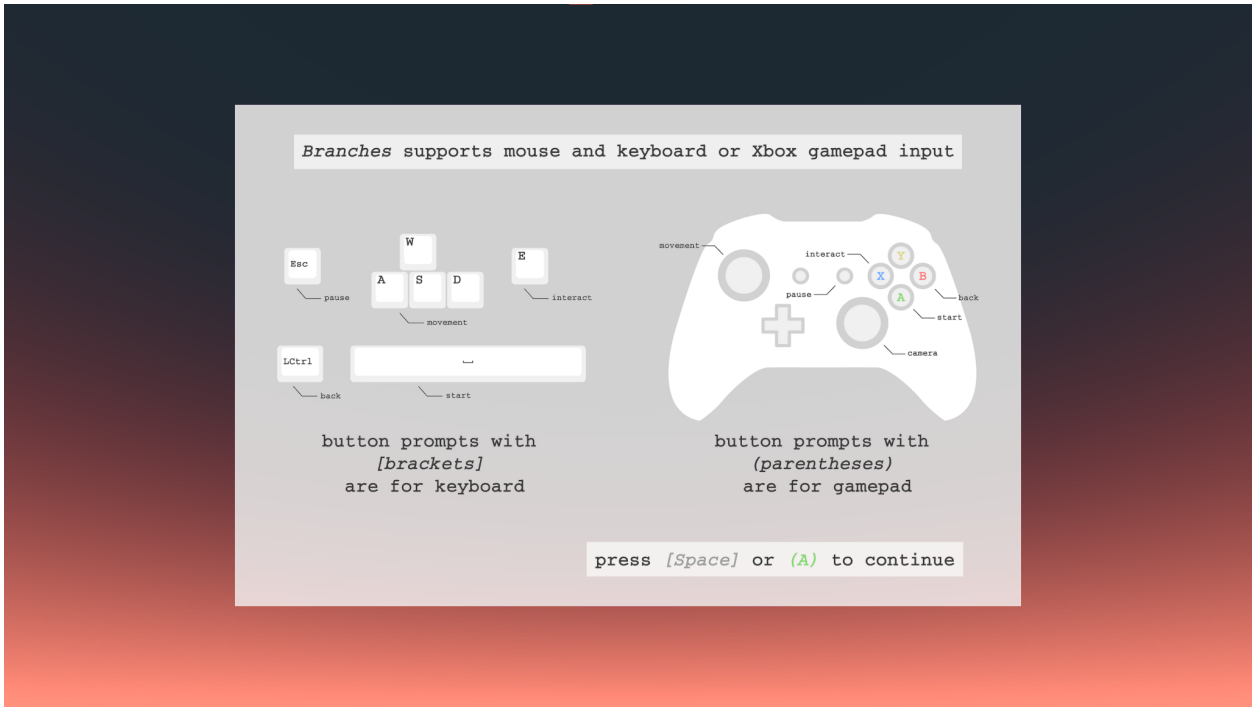

Fig 6.1 — the controls screen displayed at the beginning of the game, showing two different control schemes.

Also important was attribution. I found a lot of the assets I was using online on platforms like SketchFab or CGTrader, so keeping track of all of them was important so I could 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, so I ended up with a long list of sources at the end. This is visible on the GitHub. Finally, representation — as I was building the game of course it was important to keep in mind cultural sensitivities and factors, avoiding representing an identity poorly. I also wanted the Keeper to be easy to project yourself onto, so not giving him a specific nationality or race, or even appearance, became important too, so that as many people as possible could envision themselves in him.

7. Evaluation:

7.1 Evaluation 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 team 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.

7.2 Evaluation Results

To evaluate my project and to determine whether or not I achieved my goals, I did interviews with players after playtests. Players 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 are 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 interviews and playtests showed this: Most importantly, everyone I asked these questions agreed that, at least on some level, their choices mattered significantly. Second, everyone picks a different path. There seems to be no most frequent path, which is promising for the idea that users are choosing these freely. 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 world played out. Users who playtested multiple times also commented on how much the additions of things like lighting, the objectives blurb box, and the 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. 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.

Overall, I think that my project was incredibly successful. Reassessing my initial scope, I was far too ambitious. To get to this point has been an incredible effort and amount of work, and to expect twice this much to be done in the same time was unreasonable. Despite cutting half the content, which I assumed would ruin the game, players were still able to feel that sense of agency, and the choices provided to them didn't feel restrictive or forced. For 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 set out to, the final product was still a system that was able to create the feeling of agency, and with that I am satisfied.


for more information visit
https://github.com/nickesc/BranchesGame

Works Cited:

Bethesda Game Studios. *The Elder Scrolls V: Skyrim Special Edition*. 1.5.16, Bethesda Softworks, 2016, https://store.steampowered.com/app/489830/The_Elder_Scrolls_V_Skyrim_Special_Edition/.

Galactic Cafe. *The Stanley Parable*. July 2014, Galactic Cafe, 2013, https://store.steampowered.com/app/221910/The_Stanley_Parable/.

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

Gorisse, Geoffrey, 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, Frontiers Media SA, July 2017, doi:10.3389/frobt.2017.00033.

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

People Make Games. (2020). The System Behind Hades’ Astounding Dialogue. Youtube. Retrieved from https://www.youtube.com/watch?v=bwdYL0KFA_U

Quantic Dream. *Detroit: Become Human*. 1.0.8, Sony Interactive Entertainment, 2018, https://www.playstation.com/en-us/games/detroit-become-human/.

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*, edited by Stefano Triberti, vol. 13, no. 7, Public Library of Science, July 2018, doi:10.1371/journal.pone.0200724.

Tanenbaum, Karen, and Joshua Tanenbaum. *Commitment to Meaning: A Reframing of Agency in Games*.

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.532.2370&rep=rep1&type=pdf. Accessed 28 Mar. 2021.

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/.