

Dreamkeeper: 3D Game Using Unreal Engine 4

Casey Albert-Hall, Marii Boyken, Martin Coleman

Aaron Keen, Faculty Advisor
California Polytechnic State University
San Luis Obispo, CA USA

June 7, 2017

Contents

Project Credits	3
Document Revision History	3
1 Abstract	4
2 Introduction	5
3 Background	5
4 Technology	5
4.1 Unreal Engine 4	5
4.2 Version Control	5
4.2.1 Google Drive	5
4.2.2 Git via SourceTree	5
5 Design	6
5.1 Story	6
5.2 Art	6
5.2.1 Sara	7
5.2.2 Harry	8
5.2.3 Nightmare	9
5.2.4 Sid	10
5.3 Level Design	11
5.3.1 Tutorial Level	11
5.3.2 Tommy's Dream Level	12
5.4 Sound	12
5.5 Player Stats	12
5.6 Combat	13
5.7 Controls	13
6 Implementation	13
6.1 Unreal Blueprints	13
7 Analysis	18
7.1 Play-Testing	18
7.1.1 First Round	18
7.1.2 Second Round	18
8 Future Work	18
8.1 More Puzzles	18
8.2 More Dream Levels	18
8.3 Cinematics/Cut-scenes	19
9 Conclusion	19

Project Credits

Name	Date	Role	Version
Casey Albert-Hall	January 8, 2017	Story author and programmer, level designer	1.0
Marii Boyken	January 8, 2017	Environment/character artist, visual effects programmer	1.0
Martin Coleman	January 8, 2017	Physics and combat programmer	1.0
Laryssa Chan	January 8, 2017	Musical composition	1.0
Aaron Keen	January 8, 2017	Faculty advisor	1.0

Document Revision History

Name	Date	Reason for Changes	Version
Marii Boyken	May 18, 2017	Initial Background and Technologies	1.0
Marii Boyken	May 21, 2017	Initial Design, Implementation, Analysis, and Future Work	1.0
Marii Boyken	May 23, 2017	Updated Design and Implementation	1.0
Marii Boyken	June 5, 2017	Updated Design and Analysis	1.0
Marii Boyken	June 7, 2017	Updated Implementation	1.0
Casey Albert-Hall	June 7, 2017	Final edits	1.0
Marii Boyken	June 7, 2017	Final edits	1.0
Martin Coleman	June 7, 2017	Final edits	1.0

1 Abstract

The goal of this senior project was to combine our diverse skills to make a 3D game. The plot involves a young girl who has been given the powers to enter dreams and defeat nightmares. The 3D assets that we built and animated were made in Autodesk Maya. We used Unreal Engine 4 to piece together our assets and create the logic for the game. This document explains the technologies that we used, design choices we made, feedback from our player-testing, and work that we want to complete for the project in the future. Our final game demo features two levels, a combat systems that allows played to level up skills, and puzzles that allow the player to proceed to the final battle.

2 Introduction

Dreamkeeper is a story-based 3D role-playing game that follows the story of a young girl named Sara. Players explore dream worlds through a third-person perspective, solving puzzles and defeating nightmare creatures.

3 Background

We wanted to combine many of the skills we've learned in our undergraduate careers at Cal Poly to make an interdisciplinary project. To showcase our diverse talents, we wanted to design and implement a plot-driven, graphically stimulating video game. All three team members have backgrounds in computer science, but we also had story-telling experience from one member and an art background from another, so we were able to bring all of our skills together to cover a few of the typical roles in the game development industry.

4 Technology

4.1 Unreal Engine 4

To achieve the level of graphics we wanted with the amount of time we had, we needed to work in a game engine so that we could use the frameworks already in place and focus more of the logic of the gameplay. We had chosen Unreal Engine 4 after weighing the pros and cons between Unreal and Unity3D. One group member had previous experience in Unity3D, so we were leaning more toward it in the beginning, but we decided to go with Unreal instead, as two group members would be using Unreal in other classes beginning Winter 2017. We also found that Unreal was more commonly listed as a skill in job descriptions for game development, an industry into which one of our members are interested in pursuing in the future.

4.2 Version Control

4.2.1 Google Drive

In the design phase, we were solely using Google Drive to keep track of our planning documents and assets. Google Drive has a rudimentary versioning system, so we were able to update our work and keep track of older work easily. We chose Drive because it's easily accessible and editing documents is simple and live.

4.2.2 Git via SourceTree

Once implementation started, we went with Git for version control in the implementation of the project. However, we continued to maintain the documentation on Google Drive. To easily work with Git, we chose to use SourceTree. This would allow us to see a visual representation of our commits on the desktop, where we make our commits, instead of in a less visual way in the actual repository on GitLab.

5 Design

5.1 Story

We wanted our game to have a background story, so the team member that has experience in story-telling wrote our story. The game follows a young girl named Sara, who finds herself in a dream world. She comes upon an older man name Harry, who is known as Homeless Harry in her waking world. He guides her in learning about the nightmares of the dream world and her powers, such as fire casting and summoning her teddy bear, Sid.

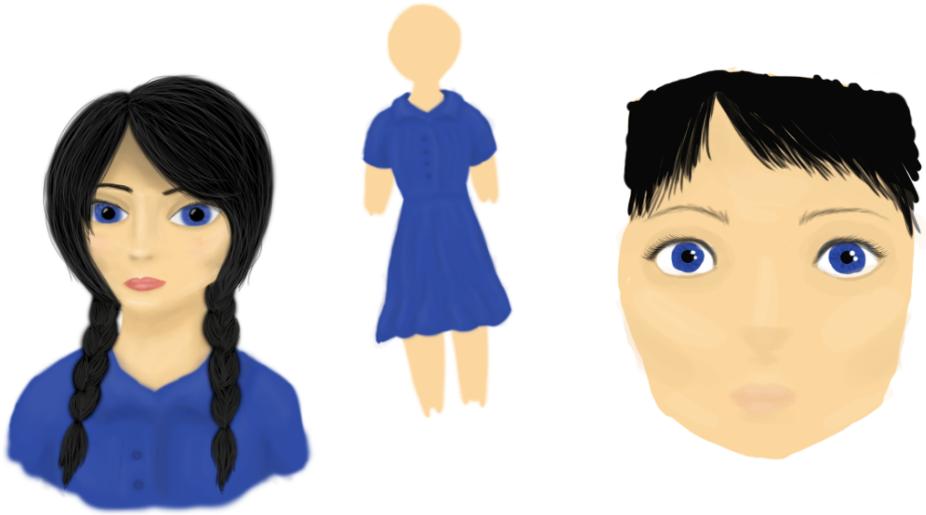
The original idea for the story included cut-scenes, but time would not allow us to produce these. Sara would wake up in her bedroom and take her old teddy bear, Sid, with her to school. On her way to school, she would see Homeless Harry sleeping on a bench and a pink cloud would sneak into her backpack, where Sid would be. Once she got to class, she would fall asleep and be transported to a dream world where Homeless Harry would introduce himself as a keeper of dreams, who controlled dreams and helped people escape nightmares. Nightmares would come to attack them and Harry would become overwhelmed and give Sara dream-keeping abilities. Harry would die in the dream and once Sara woke up and walked home, she would see an ambulance taking Homeless Harry away on a stretcher. Once she got home, she would see her younger brother, Tommy, asleep and writhing in pain. She would then fall asleep beside him to enter his dream and defeat his nightmares.

Because we were unable to produce cut-scenes, especially since we had several minutes of story that would need to be portrayed, we decided to simplify the story for the final demo. Sara begins in a dream world, where Harry is speaking to her, explaining his purpose as the keeper of dreams. He teaches her how to move, attack, and summon Sid, her teddy bear, as companion for battle.

5.2 Art

Since we decided to make a 3D game, we needed to build our characters. One of our members has a background in art, so the story-writer came up with the descriptions for the characters and enemies and the artist created concept art for each model. Once designs were approved by all members, the artist made 3D models of each, using Autodesk Maya. Bump maps, which create patterns of raised and lowered texture without actually changing the geometry, and UV maps, which assign colors to face on the geometry, were painted using Adobe Photoshop.

5.2.1 Sara



The image above on the left is the initial concept art for Sara. The original description this concept was based on listed "12-year-old, 7th grade girl, about 5'1" with small, black pigtails, blue eyes, blue dress".

The image above on the right is the painted texture of Sara's face from the UV texture map, which is the texture that appears on the final 3D model in the game. Her features were softened from her concept art to make her look younger. Even though the model is rendered in 3D, contours were painted onto Sara's skin to give more depth to her face while keeping the model at a relatively low polygon count.



The image centered above is the final model and texture of Sara, rendered in Unreal Engine 4. The lighting is vastly different from the in-game lighting so that her features can be more visible. After rigging Sara and painting the weights of each joint on the model, different animations were made, including a run cycle, an idle cycle, an attack, and a jump. Rigging, painting weights, and animating were all done in Autodesk Maya.

5.2.2 Harry



The image above to the left is the initial concept art for Harry. Harry's only description was "homeless", so his design was more open-ended for the artist to interpret. The image above to the right is the final painted texture of Harry's face from the UV map, which appears on the final model. He was given a blind eye to add more mystery and perceived wisdom to his character. Contours and wrinkles were painted onto the texture so that the model could maintain a relatively low polygon count.



The image above is the final model and texture of Harry, rendered in Unreal Engine 4. Lighting has been changed from the in-game lighting to make his features more visible. After rigging and painting the weights of each joint, only an idle cycle animation was made for him, as his model was not going to move in the new scope of the game. Another feature to try to make the character look unique and experienced is his missing limb, which is implied to have been lost when he was drafted into a war.

5.2.3 Nightmare



The image above is the initial concept art of the Nightmares that act as the enemies in the game. Since the second level that we decided to complete in our new scope was Tommy's dream, we wanted to have possessed dinosaurs as enemies. The design was modeled after a raptor and made black with red eyes to give an evil appearance. When we had first come up with our ideas for cut-scenes, which we weren't able to do in the new scope, we wanted the dinosaurs to look normal until black balls of smoke possessed them to turn the dream into a nightmare. Though, we had to drop the cut-scenes, we kept the idea of the black smoke and it appears as particle effect in-game.



The image above is the final model and texture of the Nightmare creature. The texture of the skin consists of matte black over a bump map that produces the depth of the scales. The eye is simply a specular red and the claws and teeth are matte white.

5.2.4 Sid



The image above to the left is the initial concept art for Sid, the companion and summon for Sara. In our original scope that included cut-scenes, we wanted Sara to have an old teddy bear in her room. When you get into the tutorial level, Harry asks about the teddy bear and tells Sara that she is able to summon him to help her. Since Sid was supposed to start out as an old teddy bear, he was going to be missing one of his button eyes. Once Sid becomes full size, he has a scar where his missing button eye would be.

The image above to the right is the final model and texture of Sid. One of his eyes is missing, but he does not have a scar on his final texture.

5.3 Level Design

An important aspect of a challenging and interesting game is the level design. We wanted to make the space interesting so that players could have a sense of where to go but still have obstacles to get there. Game mechanics such as nightmares attacking Sid and the exchange of damage were play-tested physically to ensure a more fluid transition into the digital implementation.



Above is an example of the process of developing the design of our levels. We used figurines and a Dungeons & Dragons dungeon master grid. We drew sample obstacles on the grid and moved the figurines to find out what worked well in the space.

5.3.1 Tutorial Level

The first level of the game is meant to act as a tutorial and short exposition. We originally wanted the tutorial to occur after a cut-scene that provided exposition and introduced Harry as character in the real world before he shows up in the dream world. Instead, we now have text boxes appearing on the screen to give an exposition and explain Harry's character. Harry also teaches the player the controls of the game, such as movement, combat, and health/mana bars. In our original scope, the nightmares were going to be different figures in each level. Time wouldn't allow for modeling characters for each level, so we decided that the nightmares would take the form of dinosaurs in all levels, since Tommy's dream was the most important.



5.3.2 Tommy's Dream Level

Tommy's dream is the climax and finale of the game, where Sara saves her comatose younger brother from his nightmares. Since Tommy is young, the nightmares took control of the otherwise peaceful dinosaurs that he was dreaming about. Tommy's dream begins with Sara having to hop across moving platforms, and proceeds to various puzzles that let you find and eliminate all of the dinosaurs on the level.

One puzzle involves a button that only Sid is heavy enough to press, and once he steps on it a large boulder moves to the side and reveals a cave entrance. Another puzzle involves a pillar that Sara must charge into or shoot to knock over, thus allowing her to jump on it and use it as a bridge. Some dinosaurs are hidden in the forest, and others are hidden in a ceiling of a distant castle. Once all nightmares are eliminated, the barrier to the great temple disappears, and Sara faces the final boss.



5.4 Sound

Due to the nature of our skillsets, sound took a backseat to the art and programming. The sound effects we have in place are simple and free to use under the creative commons license, but significantly improved the atmosphere to our game. Fortunately, we were able to bring in our friend Laryssa Chan to compose the musical score of the game. Laryssa has a bachelor's degree in music and just finished an internship at PlayStation working with music for video games. She composed different songs for the title screen and levels, including battle music.

5.5 Player Stats

Because we wanted to make a role-playing game, we needed a way for the player to level up their skills. We wanted to make a simple stats system where the player gains experience from defeating nightmares, levels up from certain amounts of experience points, and can allocate increases to health points, attack points and mana (magic) points. Sid's health and attack stats can also be increased instead of Sara's, allowing for even further diversity of play-styles.

5.6 Combat

The two methods of combat were a ranged attack that results in a burst of fire and summoning your companion Sid. The ranged attack can be upgraded through the player stats. This attack has a range limit of 2000, within the AI range of following. This was designed in order to prevent the player from attacking the nightmares from a far enough distance such that the nightmare would not attack the player and remain stationary. The attack's point of contact is based on a ray trace using the cursor as reference.

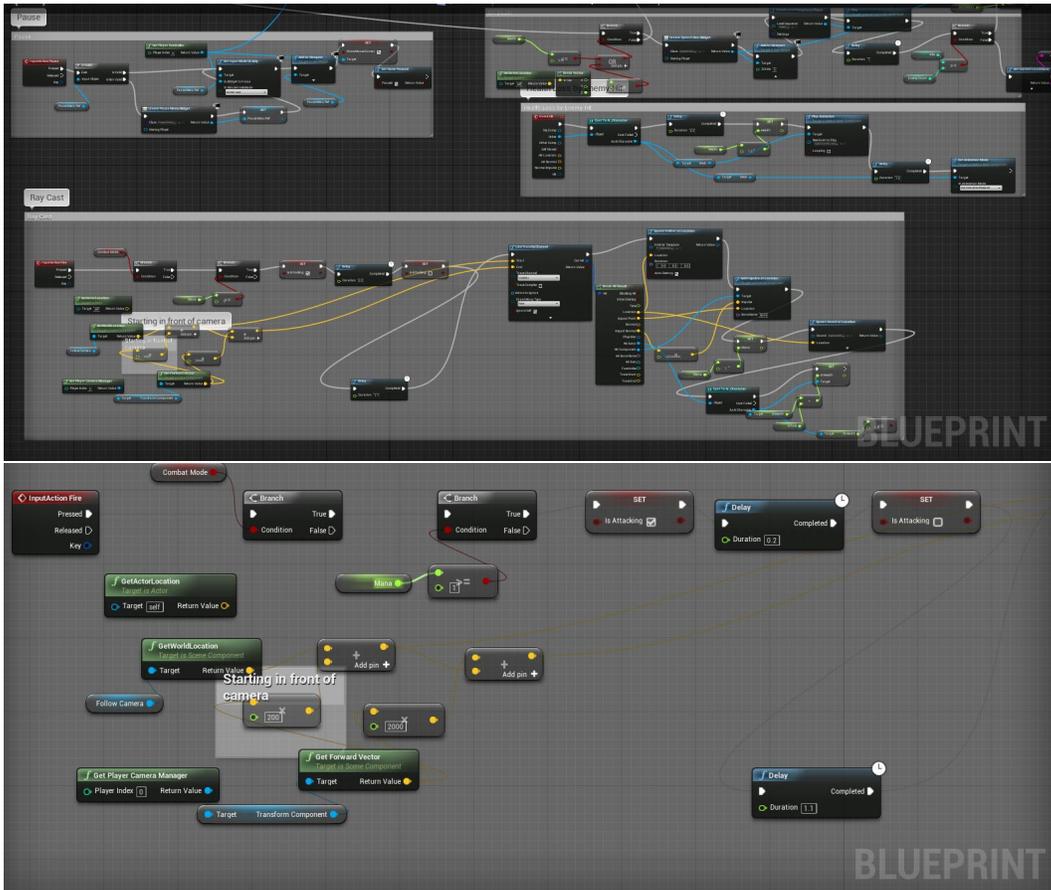
5.7 Controls

We has decided in the beginning that we wanted a 3D game because the artist had experience in Autodesk Maya and wanted to apply that knowledge. We also felt that a 3D environment fit our story more appropriately than 2D would. We were deciding between first-person camera and third-person camera. In the end, we went with third-person perspective so that the player would see Sara while playing and get more of the sense of being a young girl; we wanted the player to identify with Sara more. Other games that have first-person perspective that still allows player character development typically have voice acting that brings the character to life. With our already large scope, we didn't want to add to that scope by including voice-over.

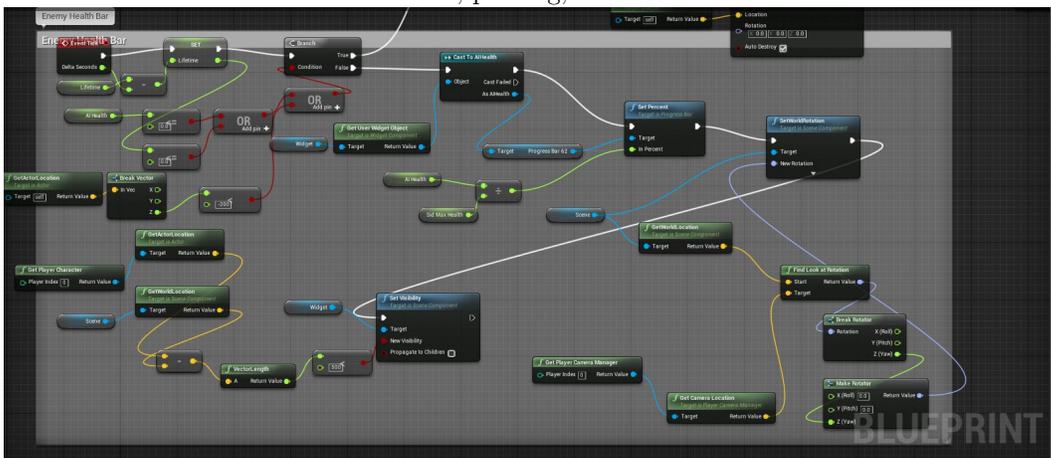
6 Implementation

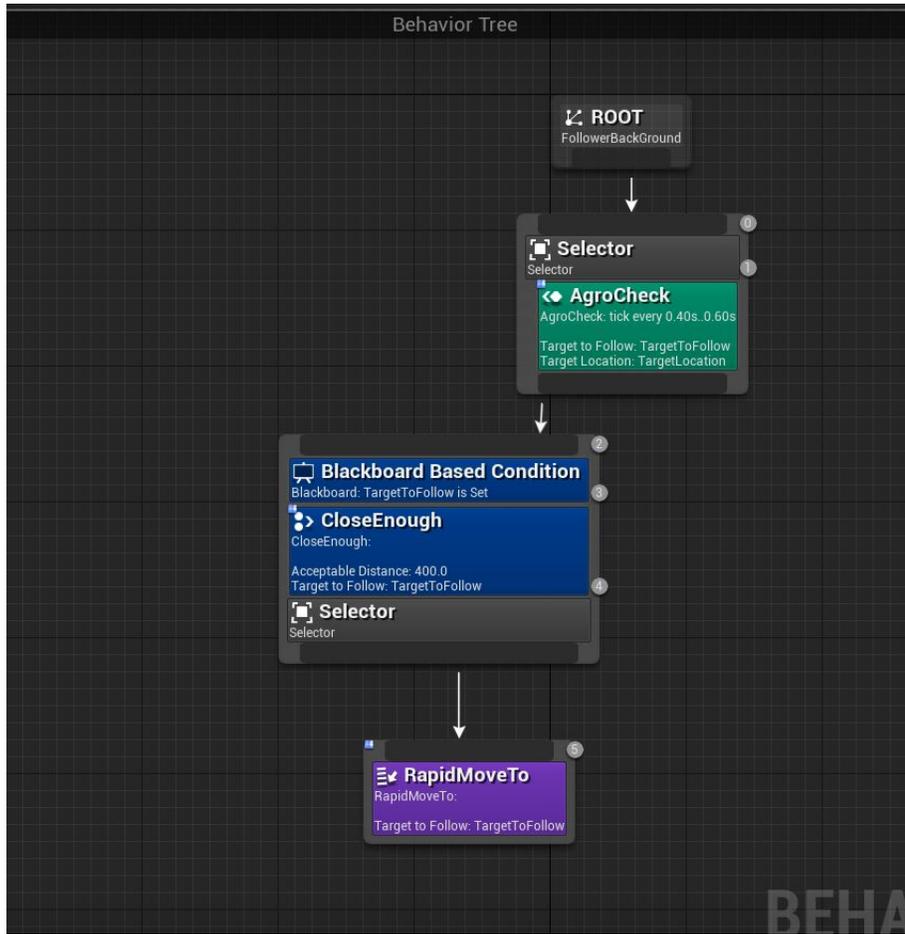
6.1 Unreal Blueprints

All game logic was created through Unreal Engine 4's Blueprints Visual Scripting System. This allowed us to easily access objects in the game and tools within the engine by connecting each function through nodes in sequence.

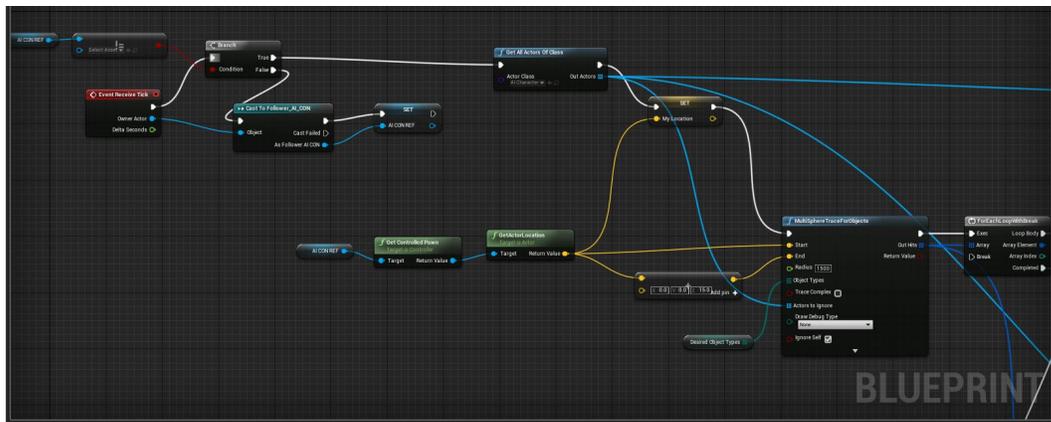


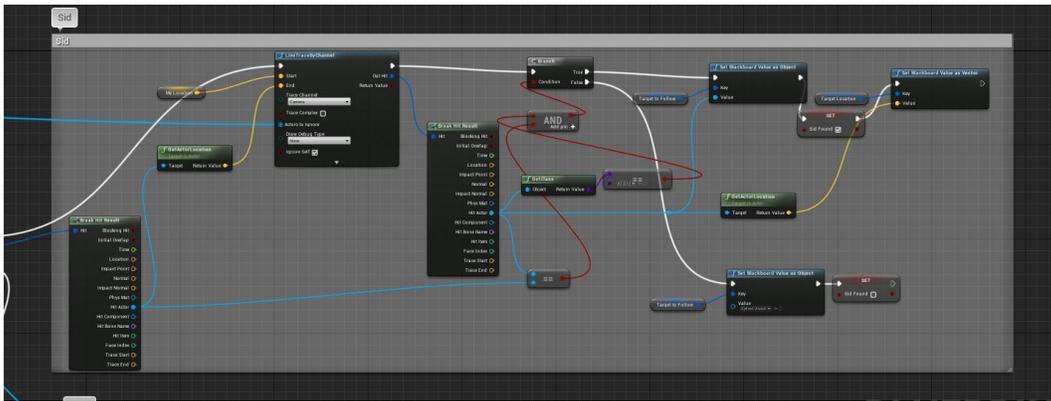
Above are screenshots of the blueprint for the the main character, which includes the logic for the attack, pausing, and health.



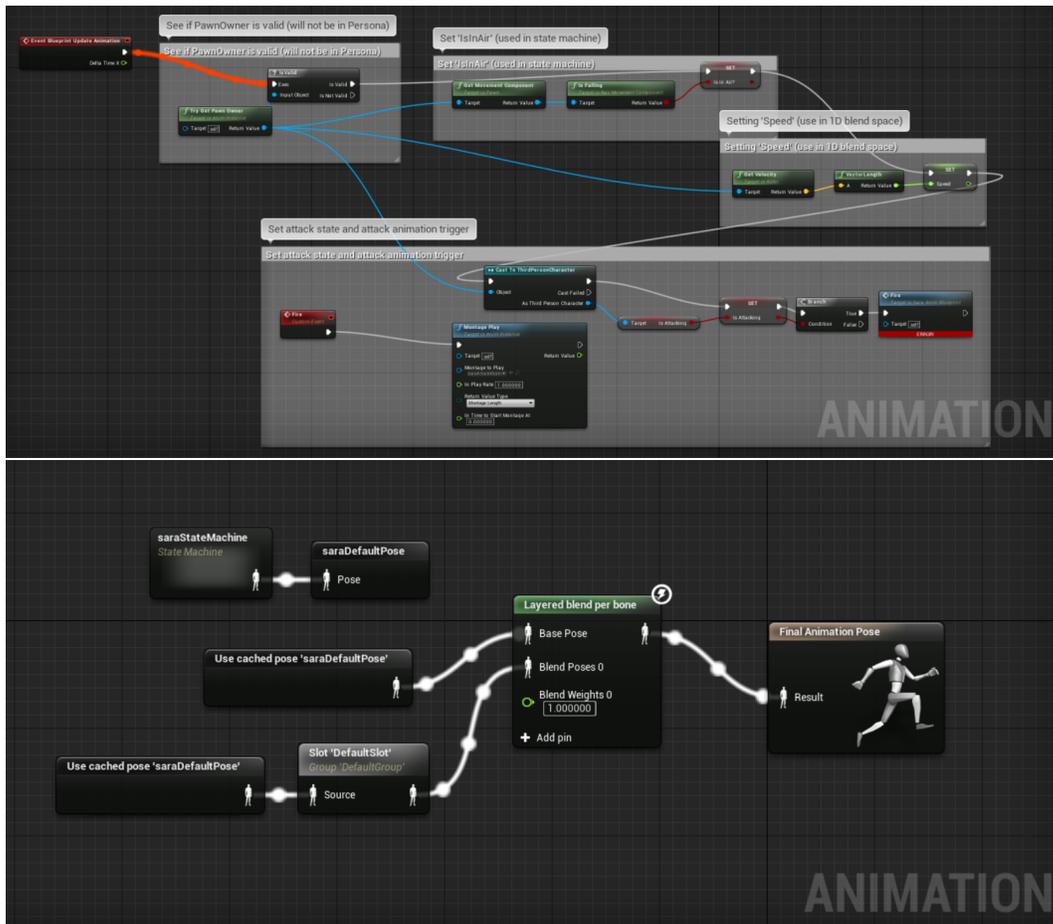


Above are screenshots of the AI health and behavior tree.





Above are screenshots of the agro-check on the AI blueprints, which make the AI characters follow Sara if she is within a certain range.



Above are screenshots of the animation blueprint for Sara, which trigger and blend animations for a cohesive look.

7 Analysis

7.1 Play-Testing

7.1.1 First Round

For our first round of play-testing, we had implemented most of the first level. Since it is a tutorial level, we had mostly play-testers who do not play computer games. This was to see if our tutorial was fit for teaching. The biggest criticism from the players was that the camera was difficult to control, as it could only pan while the user was also using the keys for movement.

7.1.2 Second Round

For our second round of play-testing, both the tutorial level and Tommy's dream were completely playable. We had minor bugs that still needed be fixed, such as the inconsistency of the puzzle that requires the player to shoot a ranged attack at a pillar in order to get across a space to defeat nightmares. One player did run into the bug while playing, so we proceeded to move the priority to a higher level. Many players found that Sid did not seem useful, so we changed his stats so that he could be of more use to the player. We also changed Sara's stat increases on skill assignment to balance her abilities with Sid's.

8 Future Work

One member of the team is required to complete an additional quarter of this project for another degree requirement, so future work will take place from September to December.

8.1 More Puzzles

We felt that a game with only basic combat could end up being boring, so we wanted to integrate puzzles into our levels to make the game more challenging and engaging. We were able to implement a few puzzles into the Tommy's dream level, though we had initially imagined future levels containing more complex and drawn out puzzles.

In the future, we would like to have one puzzle in the tutorial level so that the player could become familiar with the game's puzzles at the same time as familiarizing themselves with the movement and combat controls. We also want to add more complexity to puzzles in Tommy's dream, since it is supposed to be the most difficult level and players would be more accustomed to the challenges by that point in the game.

8.2 More Dream Levels

Since our scope in the beginning was fairly ambitious with at least four complete levels, we weren't able to meet the original proposals. The four levels that we had planned included Sara's house in the real world, the tutorial level, a dream level inside Harry's mind that took place in a war-zone, and the final boss level that took place in Sara's younger brother Tommy's dream that included dinosaurs that turned evil. We were able to implement the tutorial world and Tommy's dream world in the time that we had.

In the future, we'd like to add another level to the game, acting as an intermediate level between the tutorial and dinosaur levels. We would also like to add a real-world level as originally proposed that would assist in storytelling and give more context to the events of the game as it is now.

8.3 Cinematics/Cut-scenes

Our scope for the art was also unattainable, as we had proposed to have five or more 3D models, fully textured, rigged, and animated. To actually reach our goal of a playable demo, we lowered the requirements to only four essential models, fully textured, rigged, and animated. Since each character took 10+ hours to model, 4+ hours to texture, 8+ hours to rig and paint weights, and 10+ hours to animate the basic in-game requirements (walking, idle, and attack), there wasn't time to make fully animated cut-scenes. Instead, we went with text displayed to the screen to portray the story.

In the future, we'd like to be able to work on a fully-animated cut-scene for at the exposition of the story.

9 Conclusion

Ultimately, we believe we've made a fun and exciting game that most people could learn quickly and that has a moderate degree of challenge. There is some replayability in terms of experimenting with different stat builds and a story that is both fresh and heartwarming. Certainly there were many times we had to adjust scope as development introduced plenty of unforeseen challenges, but we learned a great deal about working in a game development team and it shows in the final product.