Worcester Polytechnic Institute

Displaced

A Major Qualifying Project

Submitted to the Faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree in Bachelor of Science
in
Interactive Media and Game Development and Computer Science

By Stephen Long and Ceren Savasan

Advised By

Dean O'Donnell, David Finkel and Robert Lindeman

Abstract

Displaced is an MQP in which we designed and implemented a story-driven game with stealth and survival elements, using the Unreal 4 Engine, through the perspective of a refugee child. The technical focus of this project was to implement an advanced AI and core mechanics such as inventory and checkpoint systems to create a refined and enjoyable experience, while informing the player of the refugee crisis. The art style we chose was photorealistic with classical music as an ambient background.

Table of Contents

Table of Figures

Introduction

1. Overview

Our Major Qualifying Project's main focus was to create a story driven game where gameplay would be limited to aiding the player to uncover the story. The initial proposal of the game was to come up with a game idea that both fit this description, and was a compelling story that was appropriately scoped to the time frame we had to develop it and that it was fun to play. We can divide the major features that we worked on into three categories; story and character development, technical development, and art and game design.

The story and the protagonists' role in our game was decided in the beginning of our development process. We decided to throw the player into the hostile environment with very little instruction as to what to do, to let them uncover the story of this child and be just as confused as any child be when left alone in a town or house by themselves, amidst a civil war. The child is of ages 10-11. The main gameplay elements were to hide whenever adults were encountered and scavenge for resources to stay alive. The goal of the game is to get to safety, to the nearby town where the child knows there is a safe house. We wrote the script for the mother character as a guiding role, and the enemy and child do not speak. The story progresses in three acts; the house where the mother and child are hiding, the forest that the child escapes to after the mother is taken away, and the town that the safe house is located in. The story ends when the child reaches the safe house and finds a tunnel inside that leads her out of the town.

The technical aspect of our game was to facilitate the story being told. We added several ways to interact with the game world; the inventory system to collect and use food or water items, the item pickup ability, and the ability to sprint or crouch to hide or escape from enemies. The statistics that we

want the player to utilize are the girl's health and stamina bars, as well as a hunger and thirst bar that directly affect the health or stamina regeneration. If the girl's hunger bar is empty, the health bar will steadily fall until the player dies. Stamina regenerates slower if the girl is thirsty. These statistics can affect the player's ability to effectively avoid enemies. A major technical component of the game was the AI character, to which we will also refer to as the enemy character. The enemy runs on a behavior tree that we will explain in detail in the technical implementation section.

We chose to have a photorealistic art style to increase immersion, and focused our research on environmental assets and props in order to match the feel of the game to that of a country in the Middle East, similar to Syria. Our characters, which consists of the enemy model and the player character, were designed to be believable, but not photorealistic. The art pipeline for character assets was to create the 3D model and reduce the polygon count in Zbrush, UV map in Autodesk Maya, texture the model in Zbrush and finally rig and animate the model in Maya. We used Adobe Photoshop to create or alter any textures. Our art pipeline for environmental assets differed for the two categories we had; original content created by us and external assets acquired online. Any environmental asset that we created was modeled and textured in Autodesk Maya. External assets were made engine ready in Autodesk Maya as well, this process usually consisted of UV mapping 3D models, texturing models and reducing the size of assets that had too high of a polygon count to be implemented into the game. We have provided a list of all external assets used and credited their original creators in Appendix 1. The design of each level and the overall feel of the game resulted from our background research section. We designed levels to increase in difficulty and enemy numbers, as well as size. We wanted enough variation in the appearance of the levels to be recognizable. We implemented a checkpoint system to allow the player to save their progress into each level, to reduce frustration if the player is being caught by enemies repeatedly. Lastly, for background ambient sound, we chose classical music tracks played at a low volume, at times overlaid with any sound effects that fit the environment.

2. Artists' Statement

Displaced was born out of a years' worth of work for our team, but what it really means to us is something deeper than the hard work and many hours that we put into it. The idea to make a game about the refugee crisis was born out of our experience following the news about the Syrian refugee crisis. Throughout the last year, this issue has gotten widely covered by many media outlets, and has been discussed by governments that had to deal with the repercussions of the war in Syria, as well as governments that refused to get involved. During the early stages of our design and writing process, we were hearing about the refugees from Syria constantly. Some claimed all refugees were evil, coming to destroy Europe, some opposed letting refugees into their countries because of their religion. Some claimed that all the refugees were men, and that they should stay and fight for their country. Some supported the Syrian government, some supported the religious extremist movement. Some had to deal with refugees in their country causing distress, and some came out to help when a boat washed ashore carrying many wounded or traumatized children that barely escaped the war. Some dropped their daily lives to help those women, men and children to acclimate to a new country where they would not be at risk of getting caught in the crossfire of a civil war or face the wrath of a religious group. The only truth that our team believes is that among the conflict and the chaos that led to the forced relocation of these people, those who are most negatively affected are children. Children are unable to fight, they are unable to make any decision regarding this conflict and they are unable to protect themselves from the outcomes of a civil war. Fathers and mothers are unable to protect their children from the effects of violence, fight for survival and displacement. Many families have fallen apart, lives were lost protecting their homes or loved ones, or lost at sea with only hope for survival is to get to some piece of land away from the conflict. This is our truth, and we do not aim to make any political claim as to who is right or who is wrong in this conflict. We simply wanted to put the player in the shoes of a child who has been forced away from their home and lost all family members. We just want to emphasize the tragedy of

what these children are going through as long as the war continues. It was deliberately not made clear which country this game is set in, even though our main influence was the Syrian crisis. We chose to omit this information to put the focus on the experience on any of the children faced with a similar situation, and not with any certain political crisis. We are aware that children have been affected similarly by the Iraq war[3], and countless other wars. If *Displaced* simply makes you think about the issue of displacement due to political conflict, we have reached our goal as game developers.

Background

1. The Displaced Children of the Syrian Conflict

Our first spark of inspiration for *Displaced* came from the recent events taking place in Syria's north eastern region in 2015. The environment of hate and fear created by the terrorist group commonly referred to as ISIS, which we will refer to as its nationally recognized representative name Daesh from this point on, disrupted the lives of ordinary families and forced them to relocate to other countries. We will make this distinction in what we call this terrorist group, as ISIS stands for "Islamic State in Iraq and Syria", and as the UK Prime Minister David Cameron eloquently put it "Frankly, this evil death cult is neither a true representation of Islam, nor is it a state." The terrorist influence over Syria has caused many families to fear for their lives and flee the areas that were considered dangerous, even if that meant leaving where they have lived their entire lives and most of their belongings.

The conflict has affected the children in these regions both physically and mentally. Their communities have been destroyed, homes demolished and they have seen their family members killed by artillery, bombings or sniper fire. Some carry scars on their bodies that remind them of the conflict constantly. In the lands that they have fled to, they may not speak the language or experience culture shock. They have trouble adjusting to these new countries, and the language barrier often halts their education. Refugee families also have trouble integrating into the country that has accepted them in, because they have not been able to bring money with them and they are often offered no work at all, and in most cases, work that they are overqualified for. With their parents struggling to support them, children are not cared for adequately and often struggle with starvation and homelessness. Many refugees in Jordan, Turkey and Iraq have resorted to begging and living on the streets to survive.

FIGURE 1: REFUGEE GIRL, WORLD VISION ©2016, JON WARREN

According the data provided by the UN Refugee Agency (UNHCR), there are currently 4.8 million registered refugees fleeing the Syrian conflict, with 51.6% of them being children between the ages 0-17. Only 10% of these refugees are living in camps provided by foreign countries they reside in, while the rest of them have been living in urban, peri-urban and rural areas with little to no assistance by the government of that country (UNHCR Statistics, 2016).

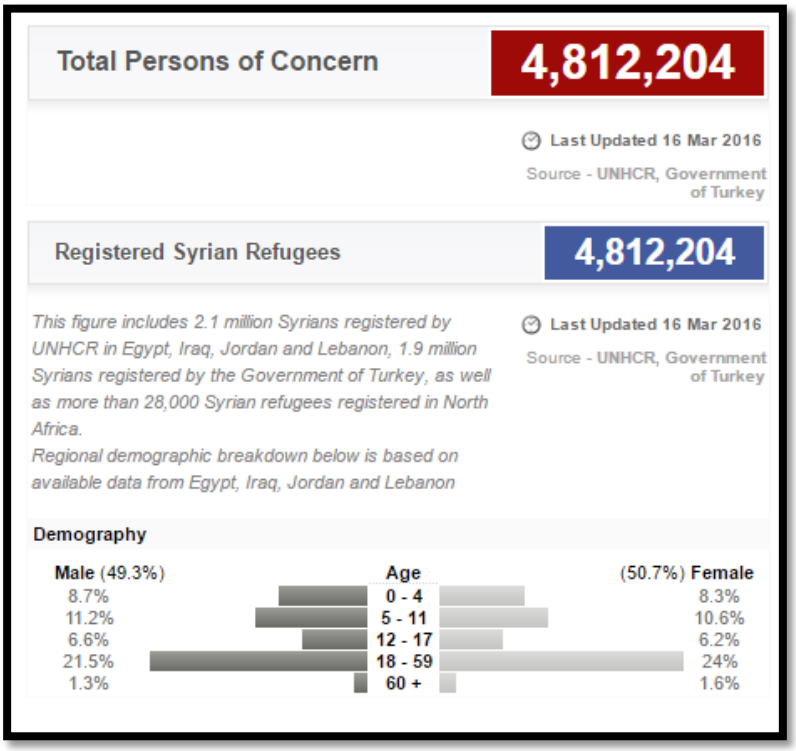

FIGURE 2: UNHCR STATISTICS, 2016

Living with the scars of the conflict, many children develop symptoms of stress and trauma. In the Jordanian Za'atari refugee camp, 71% of the children have developed "withdrawal from everyday life" as a coping mechanism. Other symptoms are bedwetting, constant crying, triggered fear by loud noises and stutters. Some children have become hyperactive, aggressive or completely closed off from human communication. Anxiety and feelings of not being safe are common amongst the children that were studied by the UNHCR. These symptoms are seen among the 2.48 million registered refugee children. The true collateral damage that rose from the conflict of Syria is the wellbeing of these children. *Displaced* was born out of this realization and the hope to encourage its players to experience only a small fraction of the helplessness and fear that refugee children experience.

2. Production Design: The Architecture and Daily Elements of Syria

To capture the look and feel of a traditional northern Syrian town, we did extensive research into actual locations that were similar in size and near the Turkey-Syria border. This northeastern region is where we based our representation of a conflict-impacted town. The cities we studied were the Syrian cities Aleppo and Manbij and the Turkish city of Suruc near the border of the two countries.[7]

The most important factor in achieving a unified and geographically accurate look was to understand how the houses and streets in Syria were built, what materials and techniques were used in construction and how the houses were decorated. Different building typologies were encountered in different areas of Syria. There was also a clear distinction between rural architecture and urban architecture. We also studied roofing methods of traditional buildings, and furniture dealers in major cities in Syria to have a better reference point when building these art assets for *Displaced*. We organized our research of architecture into the following three categories: vertical elements (walls), horizontal elements (roofs) and interior elements (furniture and daily use objects).

Walls in Syrian cities and villages mostly use stone, as it is a widely available raw material. There are several techniques commonly used in building a stone wall. The most common technique in older cities such as Aleppo and Damascus is using dressed quarry stones cut into rectangles, stacked with a rubble and soil filling between them. Specifically near Aleppo, limestone and black basalt are used the most. Second most common stone wall technique is laying uncut rough stones with a mud mortar filling to join them. One can also find mudbrick or rammed earth walls in smaller and rural towns. The outskirts of Aleppo have many examples of this kind of mud wall.

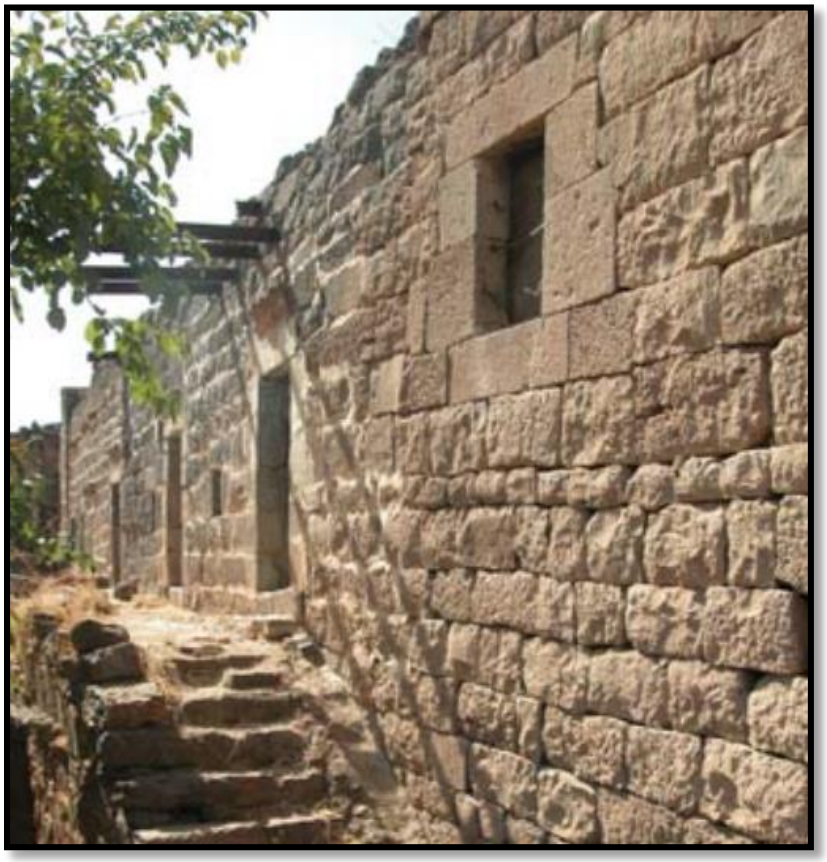

FIGURE 3: QUARRY STONE WALLS

FIGURE 4: SYRIAN ALLEY

Roofing comes in two main categories in Syrian northern regions. Cupolas are dome shaped roofs that are made of mud or limestone. They are found mostly in the countryside of northern Syria. Flat roofing is more common in larger cities with basalt rock being used as material. Some roofing is thatched with mud covering it. Lime plaster rendering is used to cover walls and roofs as a protective element as well as to beautify the look. Brick coverings are used in urban areas on top of these two techniques.

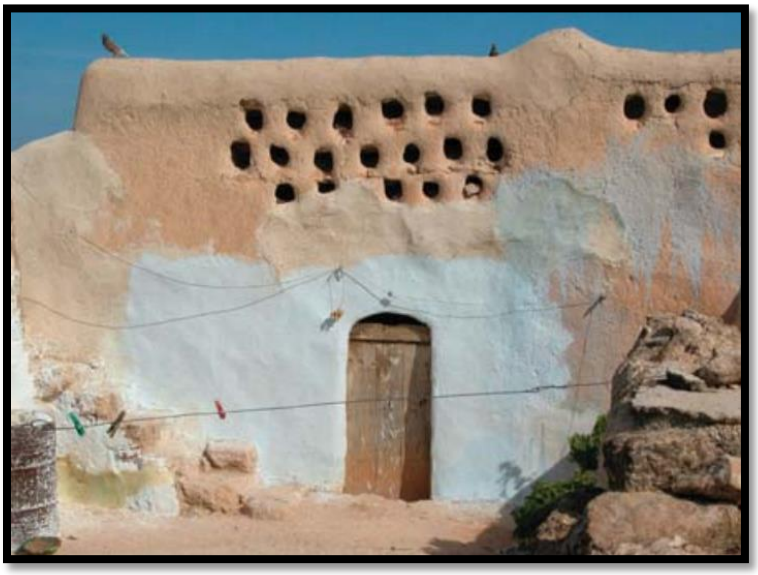

FIGURE 5: MUD WALLS WITH PLASTER RENDERING

For furniture and objects found around the house, we had less guidance. We examined an extensive library of pictures from houses in war affected areas, as well as furniture vendors' websites. The traditional furniture types used wood, plastic and traditional Arabic style patterned fabrics, such as Arabesque or Islamic geometric patterns. Furniture is mostly hard edges, and circular features are only found in domes or arches.

FIGURE 6: WAR AFFECTED DAMASCUS HOME

FIGURE 7: SYRIAN ARCH

Design

Initially, the project was going to be designed as a story driven walking simulator; it was originally designed to only have one level, and no gameplay mechanics besides movement, and instead focus exclusively on the decorations, atmosphere, and story. The project ultimately evolved to become a hybrid stealth/survival game spanning multiple levels, with gameplay mechanics for hiding, movement, interaction with the environment, and item pickup/usage.

1. Gameplay

When the game was first being theorized, the group's first design of the game was a heavily story-driven game with almost no gameplay. The player originally would get through the game by walking through a straightforward level, experiencing the story by listening to NPCs and analyzing the environment. What the project ended up becoming was a game in which the player would need to navigate levels, while avoiding enemies and using items to stay healthy. In general, these decisions were all made to make the game more engaging, and to make the player feel more empathy for the player character and more immersed in the game.

The gameplay flow is simple to learn, as it consists of only two core mechanics. The player is able to collect and use items while monitoring their health, stamina, hunger and thirst gauges. The player must also avoid enemies as mentioned, which entails their perception skills to be aware of where enemies are at all times, as well as where items are located. The player must be able to strategize when to grab items, which includes timing their traversal through different parts of the level not to be located by enemies, or deciding on the advantages or disadvantages of gathering items that may result in them getting caught by enemies. For example, in the second level, the player has the option to approach a campsite and sprint away to gather water and bread. They can choose to move further in the level

instead of attempting this, but they are risking running out of health if they don't gather any resources at all. The player also has options to take several routes when traversing the level. In the second two, the player can choose between jumping over rocks in a river that may result in them falling in the river, or going over a bridge that is occupied by enemies. The player can also choose to sneak along the railing of the bridge while crouching. These examples show the different paths a player can go about to complete the three levels.

2. Technical Design

a) Artificial Intelligence

We did not plan to utilize any sort of AI when the game was originally being designed. Any NPCs in the game would not interact with the player in any way besides saying scripted lines and performing scripted actions for the player to see. Because we wanted the game to be more than just viewing scripted scenes, we decided to implement enemy characters that the player needs to avoid in order to survive. These enemy characters are controlled by an AI behavior tree that allows them to perform the necessary actions such as patrolling an area or following the player character.

Each enemy in the game is driven by a simple state machine, written using Unreal Engine 4's blueprints system. The machine has four basic states; Patrol, Search, Slow Chase, and Fast Chase, as shown in Figure 8. For each of these states, we have corresponding animations to indicate to the player which state an enemy is in. While in the Patrol state, the enemy characters patrol along a set course of points set across the level. Once the enemy character reaches the end of the list of points, it goes through the same list in the opposite direction until it reaches the beginning. The machine leaves the Patrol state when it detects the player character. The state it enters upon seeing the player is based on the distance from the enemy character to the player character.

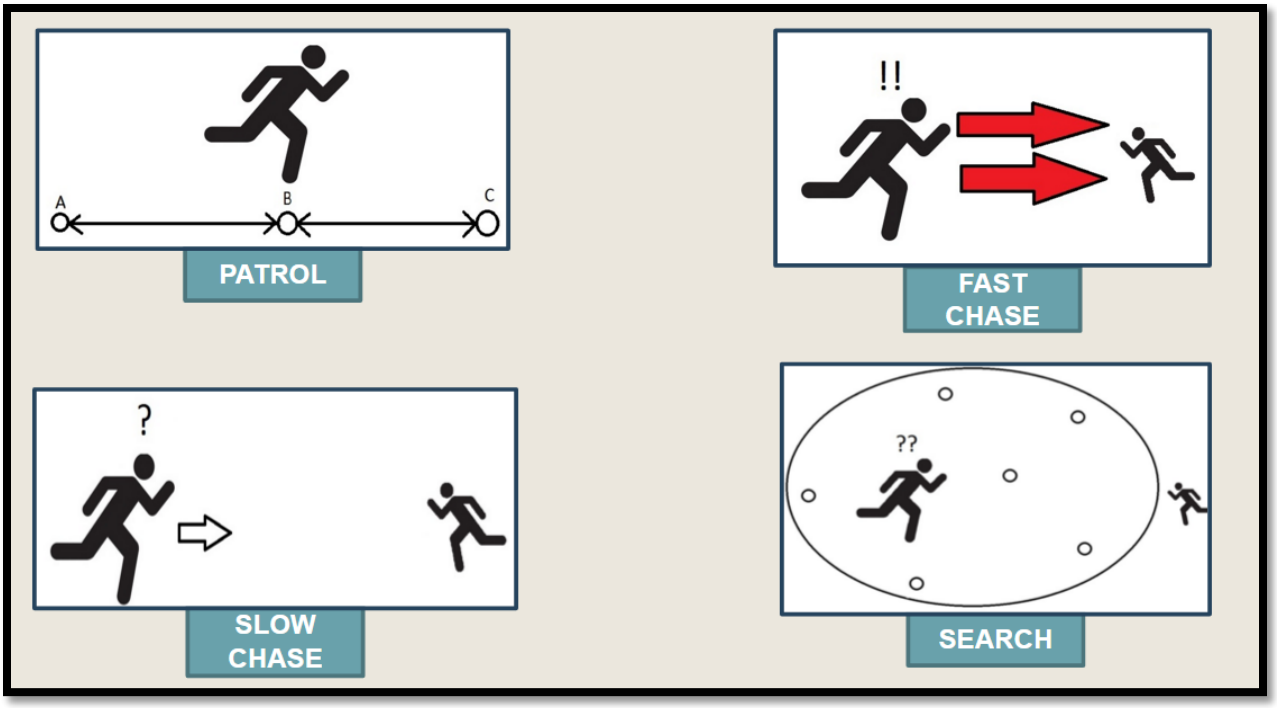

FIGURE 8: ENEMY STATE MACHINE

If the enemy character is far away from the player character upon detecting them, it enters the Slow Chase state. In this state, the enemy believes it saw something, but is not completely sure what it was and thus doesn't try to chase down the character immediately. Instead, it slowly moves towards the spot in which it detected the player character. The character could potentially move away from the AI without being detected, so the enemy shouldn't move towards the player, and instead move towards the player's location when it was detected. If, while in this state, the enemy detects the player again, it moves to the Fast Chase state only if the player is close enough to the enemy. Otherwise, it updates the spot the player was most recently seen, and continues to slowly advance to that position. When the enemy character can't detect the player anymore, it goes to the spot the player was most recently seen, and enters the Search state.

When the enemy character detects the player character and is sufficiently close to the player, it enters the Fast Chase state. This state functions almost the same as the Slow Chase state, but the enemy moves at a considerably greater speed. When the enemy character can no longer detect the player, it

enters the Search state. In this state, the enemy character first moves to the position where the player was last seen, and then patrols the area randomly, selecting random points in a fixed area and moving to them, then repeating this action a set number of times. If the player is detected again while the enemy is in the patrol state, it returns to either the Fast Chase or Slow Chase state, based on distance to player. After the enemy has patrolled the immediate area for a sufficient time, it returns to the Patrol state.

b) Stealth

When the enemy character was originally being designed, the plan for how the AI should detect the player and switch states was based on a "5 - point detection" model. In this theoretical model, the player character would have 5 points on its body; at its feet, its head, its center, and one at each arm. In the above state machine, the state the machine would go to on detection was based on how many of the five points could be seen by the enemy at a given time. This idea was ultimately discarded and replaced with using Unreal Engine 4's built in detection system to let the machine know when the player was detected, and then choosing how to act based on distance.

The reason for our decision to change to the built-in system was the lack of significant difference between the behaviors exhibited using this system as opposed to the 5-point detection. The only significant difference would be the extra time it would take to implement the new detection system. We decided to focus on different areas of development to make better use of our time.

c) Item Discovery and Usage and Player Stats

The way we implemented item discovery and usage ended up being almost identical to the way we had designed it at the game's inception. The purpose of discovering, storing, and using items is for

the player to regulate three main stats; Hunger, Health, and Thirst. Health and Thirst constantly decrease as the game progresses, and if either of them get low enough, Health also begins to deteriorate. If the player's Health ever hits 0, the player dies. When the player finds and uses items, they can raise their Hunger and Thirst stats in order to stave off losing Health.

d) Assets

I. CHARACTERS

Each of the characters in this game is designed loosely to match the inhabitants of the modern Middle East. The color and design of the clothing, and the skin tone and hair color of each characters were chosen to match their real-life equivalents as closely as possible.

ENEMY

Each enemy in the game wears a dark full-faced mask, a long-sleeved green shirt, camouflaged pants, and dark brown boots. This garb was chosen based on the attire of real-life insurgents as seen below in Figure 9, as well as the color palette of the rest of the game as a whole. Insurgents in reality tend to wear extremely dark and muted colors such as gray or brown and the color palette of the game tended to be more vibrant and colorful. As such, the enemies were colored to be the average of the two, and as a result wear darker and less saturated shades of green and brown, and thus are dark and muted by comparison to the rest of the world, but still have a degree of vibrancy and color to them. The final model implemented into the game can be seen in Figure 10, in the idle pose and the walking animation still image.

FIGURE 9: SYRIAN REBELS

FIGURE 10: FINISHED ENEMY MODEL AND ANIMATIONS

PLAYER CHARACTER

The only part of the player character that can be seen is her hands. The hands were designed to as closely incorporate the traits that differentiate adult hands and children hands as possible. For instance, the hands' skin tone is notably lighter than that of the other characters in this game, because she is far younger. In addition, her fingers are much more slender and thin.

MOTHER

Due to design constraints, the mother is only able to be interacted with from behind a closed door. However, her dialogue was still written to emulate reality as closely as possible. We strived to make sure her dialogue felt as realistic as possible, while providing a guiding role to the main character's story line.

II. ENVIRONMENTAL MODELS

Around the environment of this game, there are a wide variety of objects, such as foliage, street signs, fences, etc. We designed each of these objects to mimic the kinds of objects one would find in the Middle East.

Along the course of designing the three levels, the sort of foliage to use was consistently an area of debate. Initially, we felt that it would be fitting to implement foliage that looked sparse and thin, to emulate foliage that could be found in arid environments. However, after we researched the environment in areas such as Syria, we discovered that the foliage in areas such as the northern regions of Syria was actually far more lively lush and vibrant. It was more reminiscent of areas such as New Hampshire than the desert. Based on this information, we chose to create or use foliage in the form of full trees, patches of grass and large ferns.

Our inspiration and reference for household and city items such as street lamps and faces came from images of actual cities in areas such as Syria and Turkey. The general architecture of buildings, the concentration of objects such as lights and satellite dishes, and materials that buildings and roads were made out of in the game were all based on reality. In addition, any use of English (or any other languages) on objects such as labels, graffiti, and signs

was heavily avoided. Instead, we used Arabic to make the city feel as natural as possible.

There were many objects such as garbage, crates, and rubble that are not too different from the counterparts in different countries and cities. For items such as these, we found models and reference from various pictures and cities, as there was not a strong need to only focus on cities in the Middle East.

e) Level Design

i. Level 1: The House

The first level of the game takes place inside of a house that the player character and her mother are taking refuge in. It serves as the tutorial of the game. The level begins with the player character talking with their mother. Initially, the mother prompts the player character to find some food and water. This serves as an item tutor usage tutorial. In order to advance the story, the player must be able to move around the house, find and pick up food, and learn how to use food with the inventory.

FIGURE 11: INSIDE THE HOUSE

Once the player has learned how to use resource items, the mother prompts the player to find somewhere in the house to hide, claiming to have heard noises outside the house. This

prompts the user to begin thinking about stealth and finding good places to hide. As the player hides, they hear enemies storm into the house and kidnap the mother.

After the enemies leave, the player must exit the house and make their way past enemies that stayed behind to patrol the area, and escape the house through a gate in the house's backyard. If they are caught by an enemy before they make it to the gate, they will be brought back to the moment where they leave the house. We have implemented a checkpoint system to prevent the player from having to go through parts of the levels repeatedly.

The layout of the house was designed to be as simple as possible; there are no branching paths, and was designed to be as easy as possible for the player character to navigate. Its purpose is to teach the player the basics of the game, not to challenge them too greatly. Enemies and their patrol paths are placed in such a way that they are a valid threat, but are also easily avoidable.

ii. Level 2: The Forest

Once the player has escaped the house level, they find themselves in a large forest and in order to complete this level, they must navigate through the foggy wilderness and find a campsite that has vital information within it. Along the path to the campsite, the player will need to allocate their very limited resources, struggle to not get lost, and avoid enemies in order to survive. This level will challenge the player to not only search the environment for edible items, but also to not waste the few items they have.

The layout of this level is designed to be as straightforward as possible while at the same time giving the player ample opportunities to explore the forest. This design was implemented by first creating a path through the forest which contains the main challenges of

the level such as the bridge, from the beginning of the level to the campsite. To either side of the path extends a length of wilderness and foliage. By building the level this way, the player will be able to explore the forest while never being too at risk of getting entirely lost. We were aiming for a linear story-line that is aided by our level design choices. The aforementioned path is implemented in the level by using the terrain. For the most part, the terrain of the level is hilly, and predominantly dirt and grass. Along the path the player needs to follow, the terrain is smoother and flatter, and is made up of gravel. When the player explores the area, they only need to search for a patch of gravel to get back onto the path.

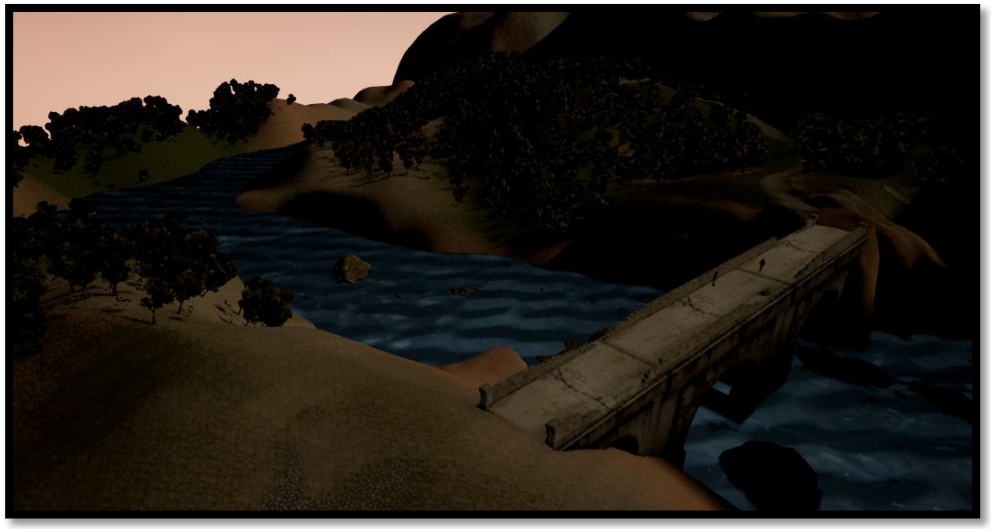

FIGURE 12: THE BRIDGE

Enemies were placed strategic points throughout the level; some will be placed at choke points such as one patrolling the bridge that spans a river, and some will simply wander around the level, looking for the player. Enemies are placed at choke points, because the player is guaranteed to pass through these points in order to beat the level, and placing enemies near these assures that the player will be challenged. The purpose of the wandering enemies is to add an element of uncertainty to the level, thus adding an element of excitement and risk.

i. Level 3: The Town

Once the player finds their way through the forest, they come across a note which gives directions to a nearby town in which they could possibly find refuge. The town level has the player navigate through a more urban area until they can find a specific building. There are many doors and many houses in the city, but the rest of the buildings are either ruined and empty, or locked. In some of the locked houses, the inhabitants of the houses will yell at the player and tell them to go away. Level 3 contains the most enemies of all the levels in the game, as well as ample resources. The player will have to make a conscious decision to avoid enemies or to attempt to get these resources while taking on the risk of being caught. Enemies are constantly patrolling the streets, looking for the player. As it is dusk time, it will be harder for the player to detect where the enemies are, and they will have to use their acquired stealth skills in order to survive and avoid more enemies than ever before.

FIGURE 13: THE MAIN ROAD IN THE TOWN

The layout of the level is designed to make the player feel tense and claustrophobic, while also requiring the player to explore a large area. The way this goal was met is by building the city level mainly of a complex and tight maze of buildings, streets, and alleyways. By designing the level like this, the player will feel tense, because they won't know where the enemies in the large level are until they are nearby. It also makes it easy for the player to get lost and run out of their food and water. In addition to providing difficulty and conflict for the player to overcome, combining the abundance of enemies with people constantly denying the player sanctuary will make the players feel desperation. It will make them empathize with the victims of conflict such as the player character.

Implementation

1. Inventory Management

a) Item Logic

Along the course of the game, the player must find and use resources in order to survive. Each resource object inherits from a basic parent class, BP_PickupActor. The only differences between each food object are the meshes that each have, and the specific stats that will be increased upon consumption. As such, all of the functionality for items is written within the parent class.

The following blueprint is the logic for picking up an item. Upon being picked up by the player, the item is first converted to a struct and added to the player's inventory using the player's "AddItem" function, which will be explained in depth later. A struct is a data type that allows us to create custom variable types to organize our object's data. The object that was converted to a struct is then set to be unable to be interacted with to avoid being used twice.

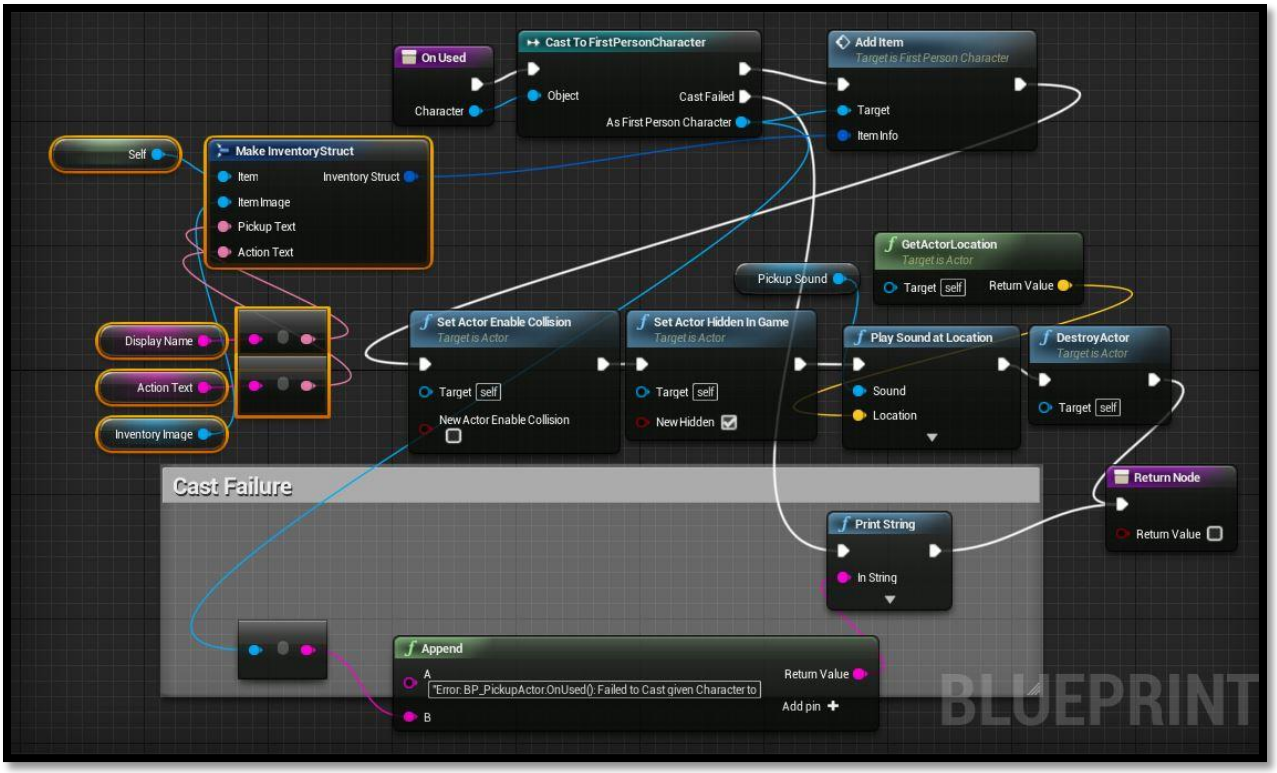

FIGURE 14: PICKUP OBJECT EVENT GRAPH

Whenever the player consumes an item from within their inventory, the following function is called for the consumed item. Each item stores a number of floats, each that represent the amount of health/hunger/thirst/energy the item will give to the player on consumption. For example, a piece of bread gives the player 40 hunger, 0 health, 0 thirst, and 0 energy when it is used. Upon consumption, each of these stats for the player is updated. After each of the stats is updated, the struct inside the player's inventory which represents the item is set to be destroyed by using the "Auto-destroy when finished" node.

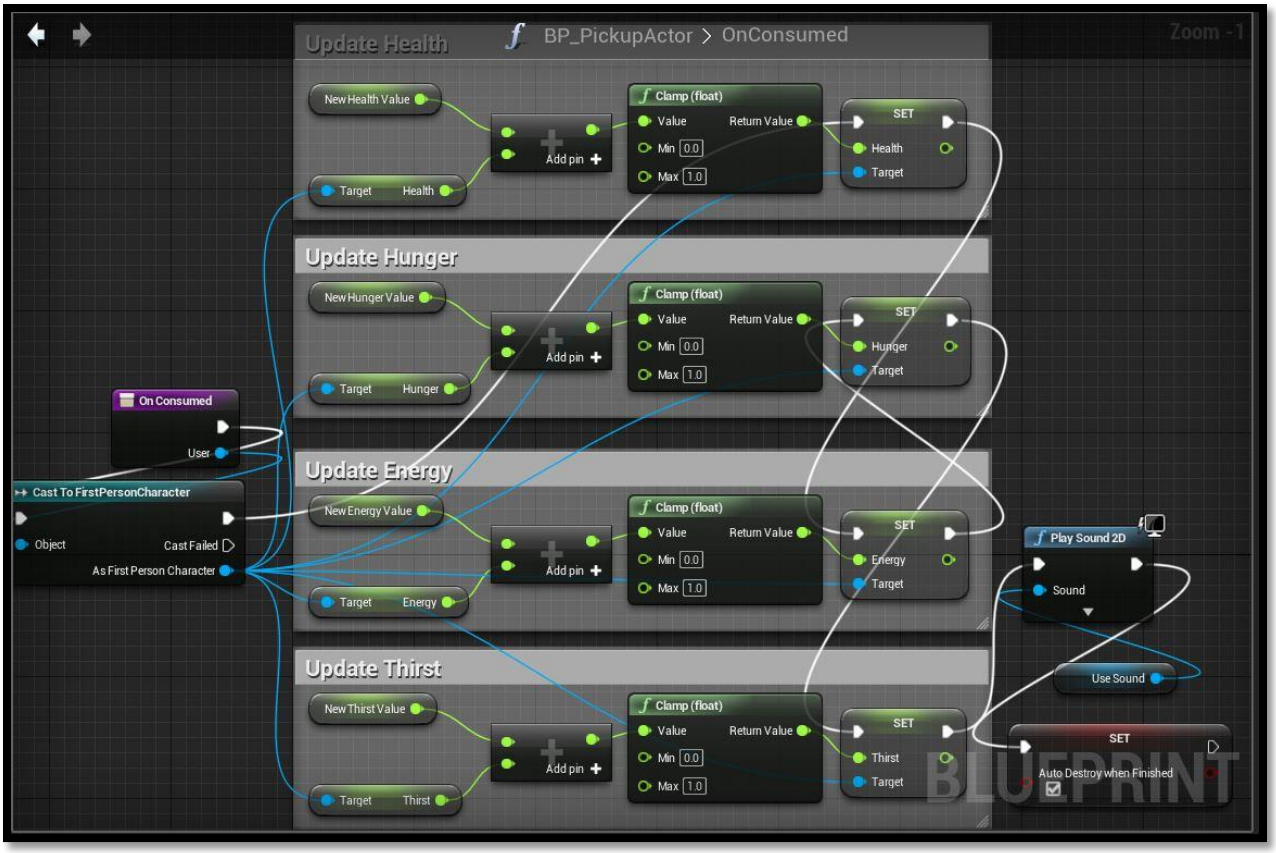

FIGURE 15: ON CONSUMED FUNCTION WITHIN PICKUP OBJECT ITEM LOGIC

b) Player Inventory

The player has its own functionality for handling an item that is being added to its inventory. The player, upon picking up the item, creates a struct from the picked up item, and adds the resulting struct to its own inventory. In this case, the inventory is an array of “BP_PickupActors”, the aforementioned parent class. The following blueprint is our implementation of handling added items.

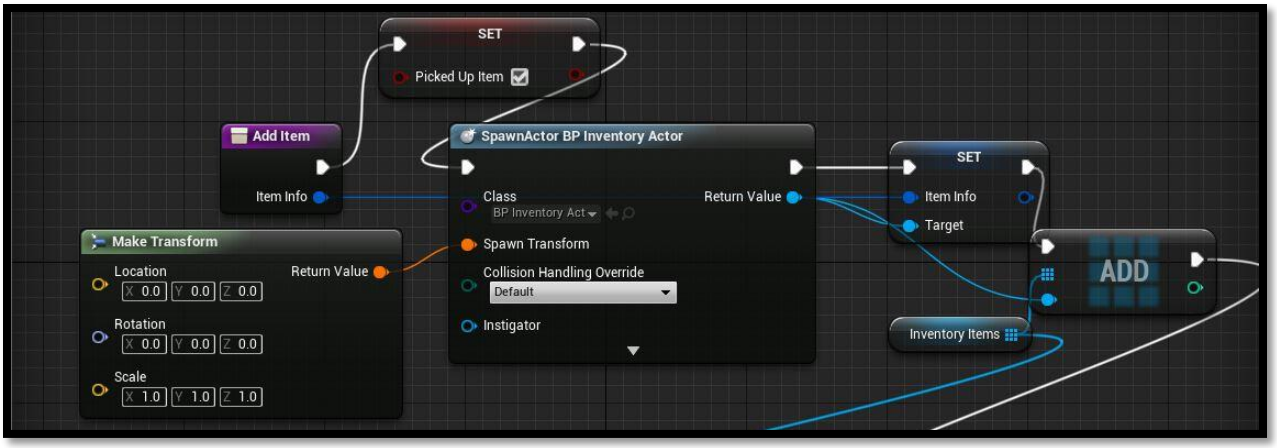

FIGURE 16: ADD ITEM TO PLAYER INVENTORY

When the player goes through its inventory, they are manually selecting which object in the player's array of owned objects to work on. When clicked, the HUD shows a drop-down menu with the options to drop or use the item and to cancel selecting that item. The "Use Item" function is called when the player chooses to use an item, and is represented by the following blueprint in Figure 14. First, the blueprint checks to see if the element of the array that the player chose is valid and contains an item. If so, it casts the struct in the array back into a BP_Pickup actor (using the information in the struct), and then calls the "Use Item" function on the new item.

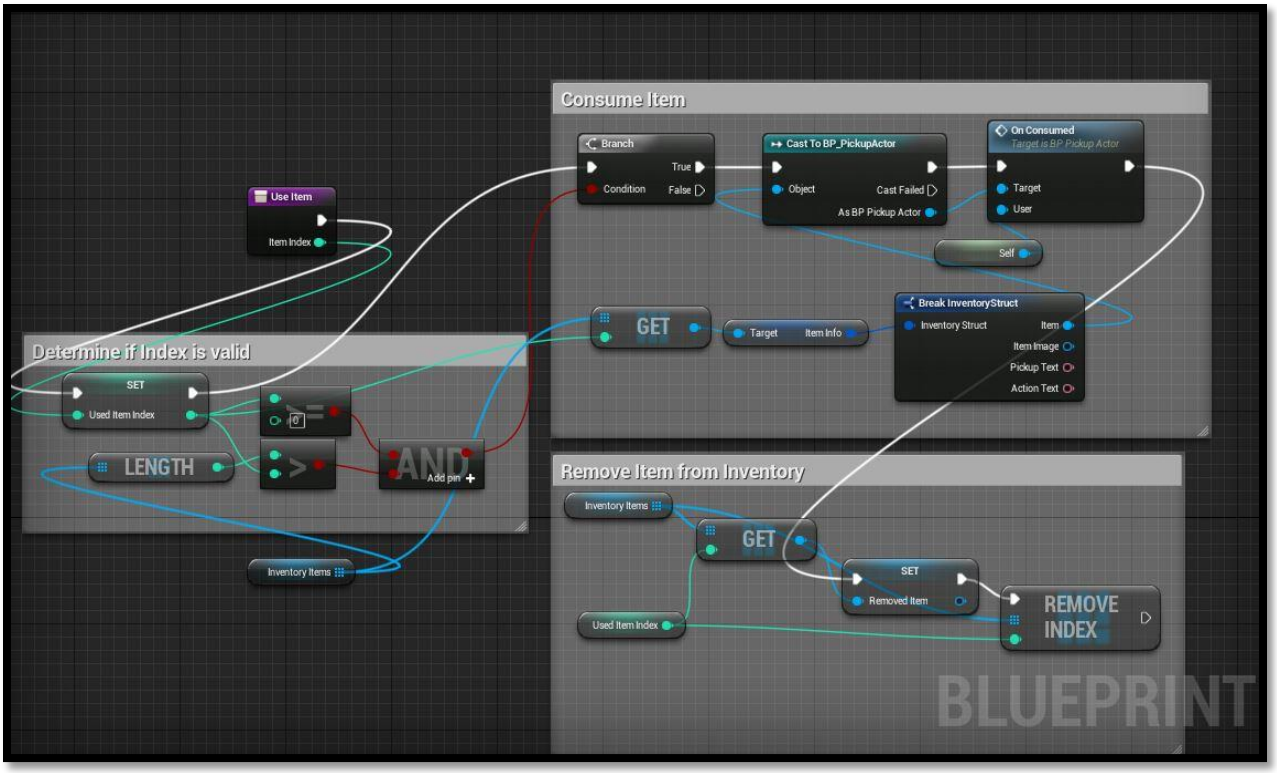

FIGURE 17: CHECK AND CONSUME ITEM

Figures 15 and 16 show the blueprint representation of the logic that occurs when an item is dropped from the player's inventory. First, the code checks if the selected array is valid, using the same logic as consumed items. Then, it creates a new item object from the data inside the struct, and sets the item to be located directly in front of the player. The item will immediately fall to the ground, but it will be visible to the user. Then, it removes the item from the player's inventory.

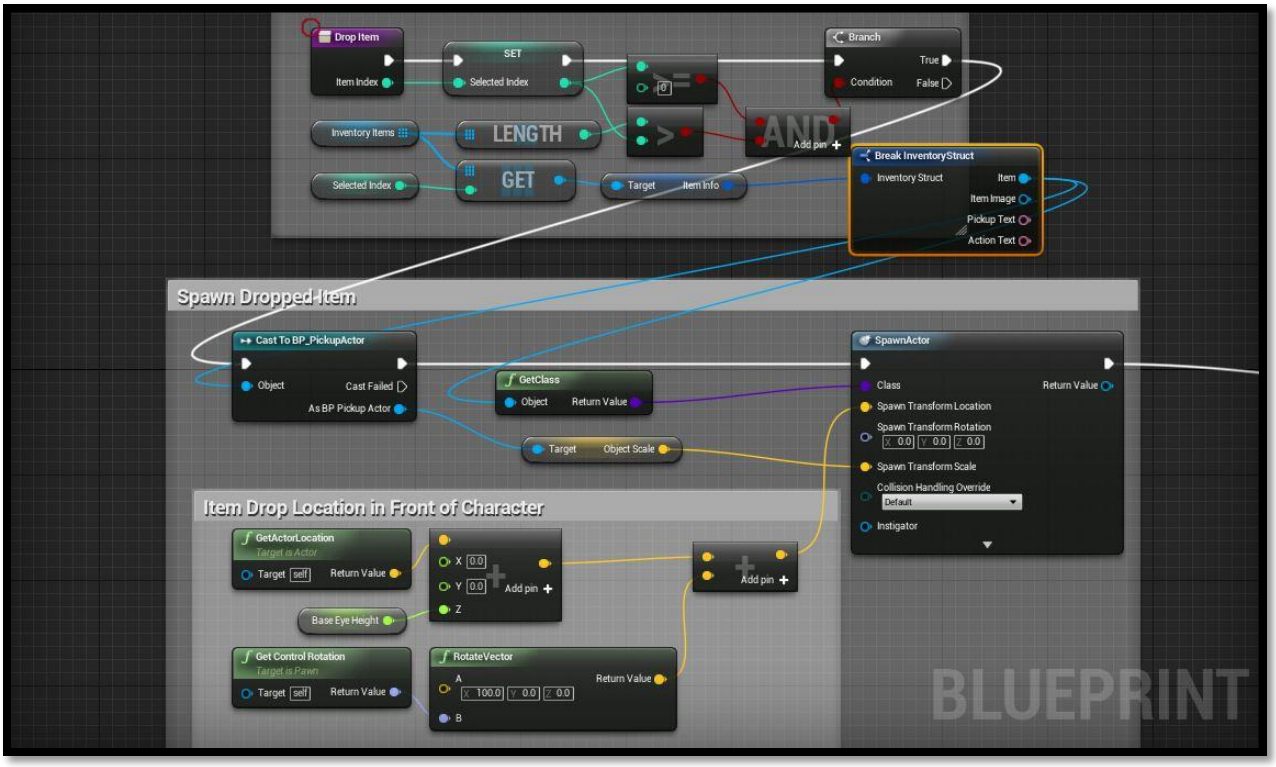

FIGURE 18: DROPPING AN ITEM

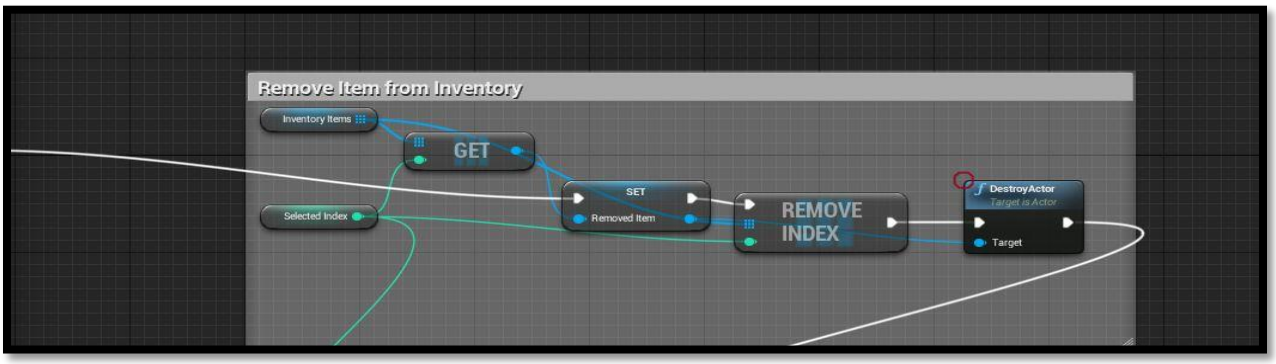

FIGURE 19: REMOVE DROPPED ITEM FROM INVENTORY

2. Key Commands

The player is able to perform a list of actions that make up the core of the gameplay mechanics. The list consists of opening its inventory to look at, use or drop items, pausing the game to exit, or

change options, sprinting to move faster and crouching to avoid being detected by enemies. The player can also interact with specific items and pick them up by pressing the “E” key.

The following blueprint shows how we enable the user to bring up and close the inventory by pressing “I”. When the inventory is brought up, the code first checks to see if a widget for the inventory has already been created. If not, it creates it. Next, regardless of if a new one needed to be made or not, the code sets the users input to only affect the UI. The player cannot move or look around while the inventory is up. It also pauses the game to prevent enemies from taking the player by surprise when they are looking at their items. When the button is pressed to remove the inventory, the code sets the input back to normal, removes the widget, and resumes the game.

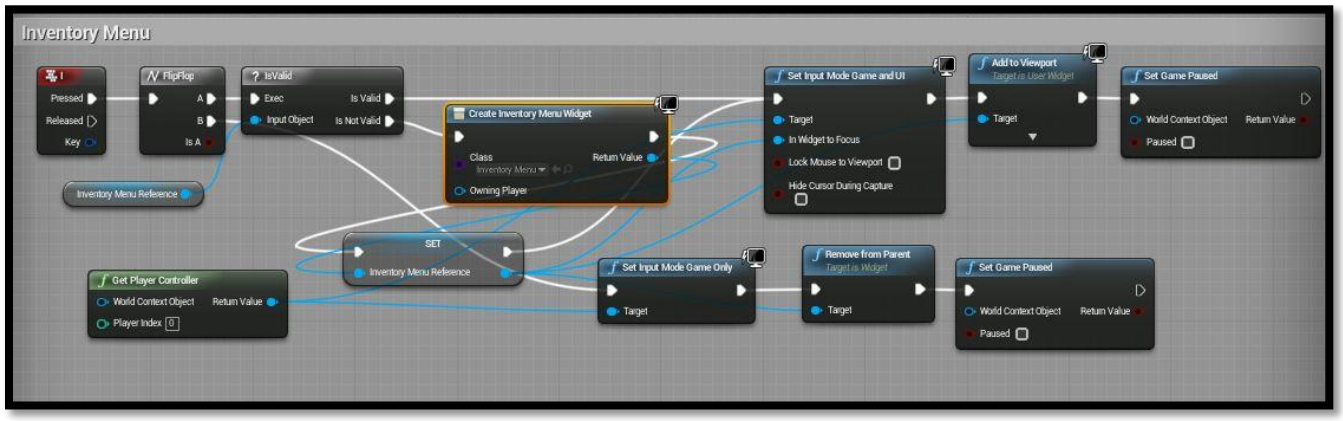

FIGURE 20: OPENING INVENTORY

The pause menu blueprint shown in Figure 18 operates similarly to the inventory menu. Upon hitting either “P” or “Escape”, the player control is disabled and a widget is created for the pause menu. Opening this menu pauses the game just as the inventory does.

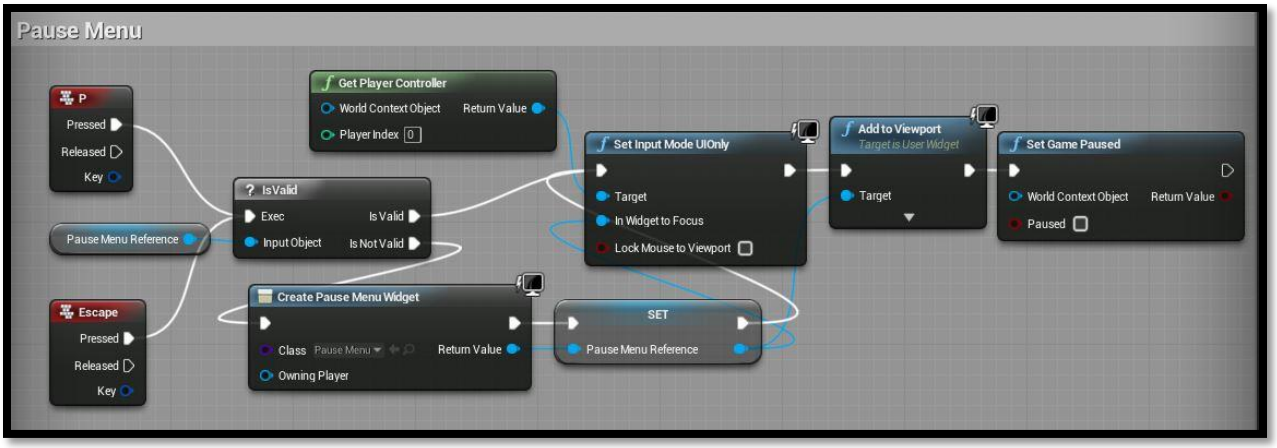

FIGURE 21: PAUSE MENU

Figure 19 shows the sprint blueprint, where it simply toggles the player's maximum walk speed between two values when the "Shift" key is pressed. The crouch blueprint, shown in Figure 20, functions similarly to sprint, but instead of toggling to the crouch state one has to hold down the "C" key in order to remain crouched.

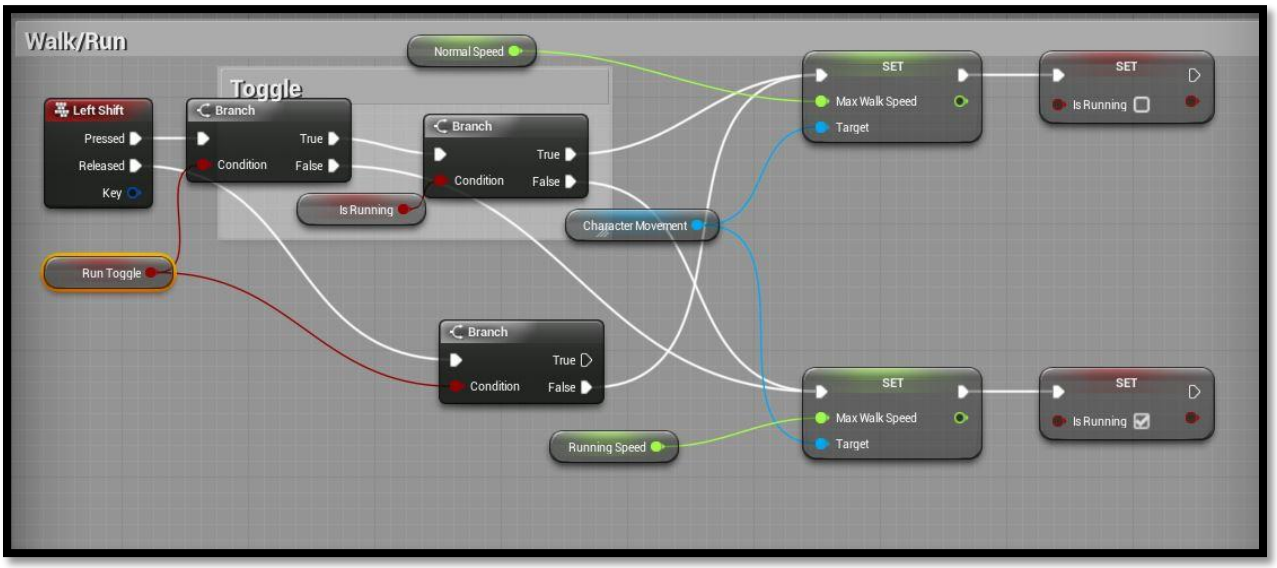

FIGURE 22: SPRINT

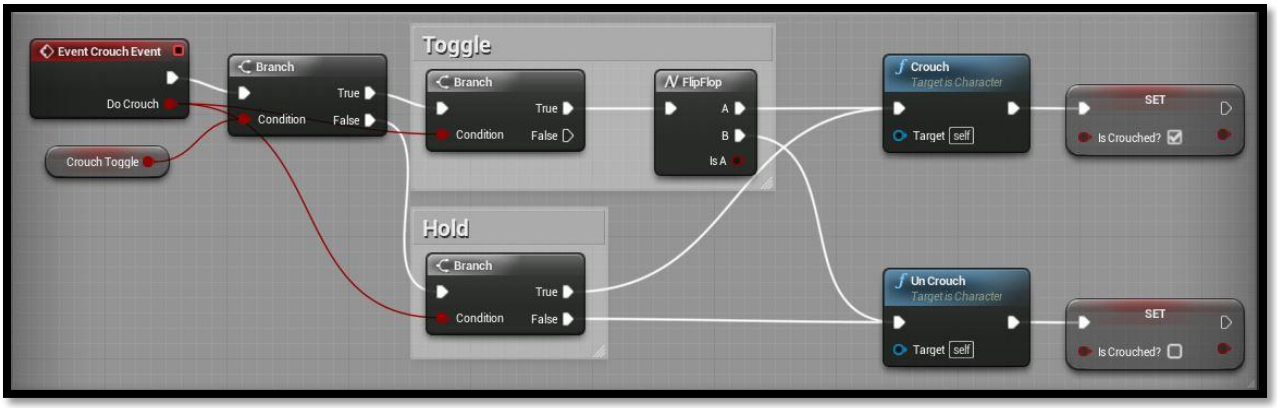

FIGURE 23: CROUCH

3. Enemy Artificial Intelligence

The following block of blueprint code initializes the enemies, as well as dictates what action the enemy should take every tick based on Boolean flags and a vector representing the player's last seen location.

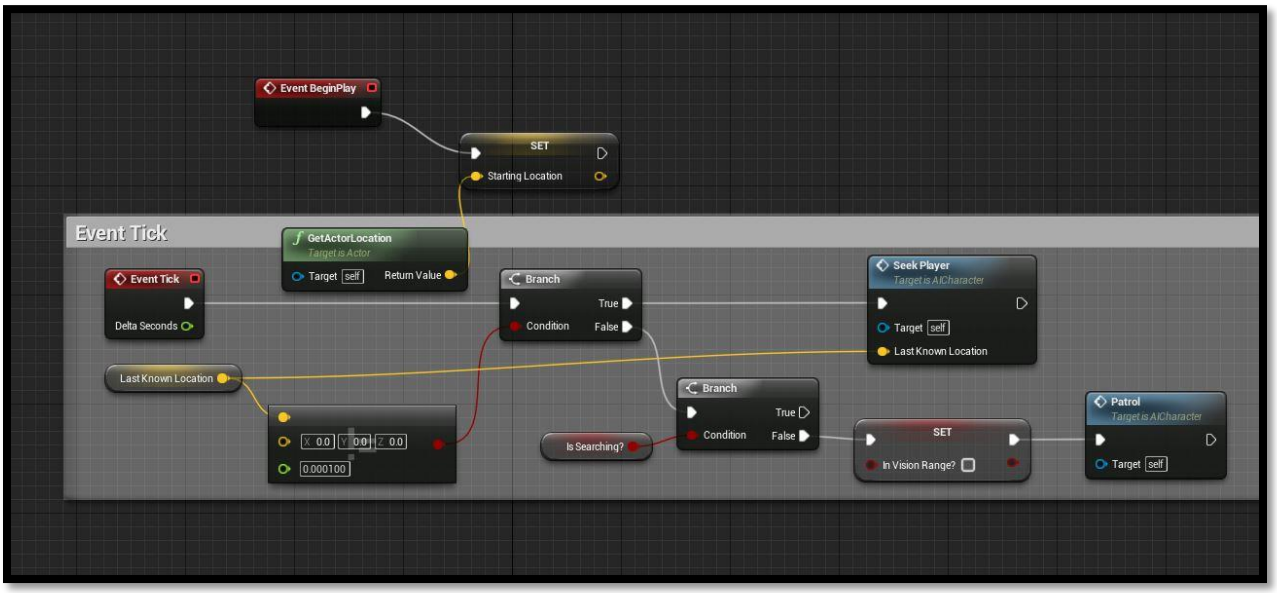

FIGURE 24: ENEMY TICK FUNCTION

It begins by setting the "Starting Location" vector as each enemy's own starting position in the game world. Every tick, it checks if the players last known location for each AI is zero. If it is zero, this

indicates that the AI does not know where the player is. As such, it then checks to see if the AI is in the middle of searching for the player. If the enemy was previously chasing the player, and loses sight of them, its last known location is still zero but it needs to act differently than if the enemy wasn't actively looking for the player. If not actively searching for the player, the enemy will return to its initial patrol by calling the patrol function, to be explained later.

If the players last known location is non-zero, this indicates the enemy can see the player, and thus should be following the player. It does this by seeking to the last known location with the Seek Player function, also to be explained later.

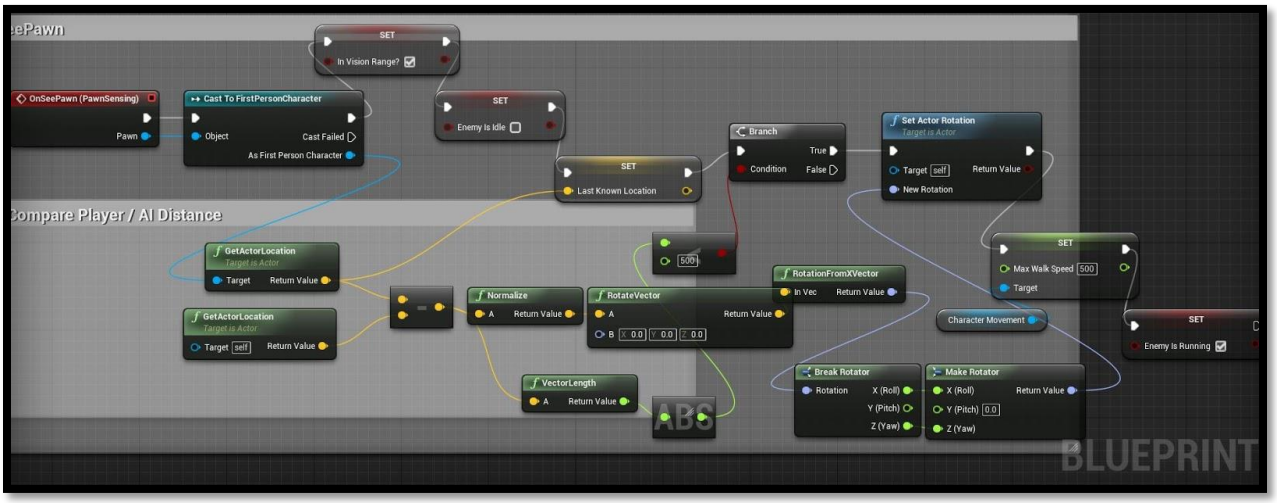

FIGURE 25: ENEMY CONTROLLER

As shown above, the enemy's controller dictates action based on a series of Boolean statements, and the vector representing the players last known location. Whenever the enemy has line of sight with the player, it updates the vector, Booleans, and its own movement speed based on player distance. Initially in this block of code, upon seeing the player the enemy sets its Boolean flags to indicate the player being in range to true, and the enemy's aggression also to true. It also updates the last known location to enable the enemy to chase the player. The next block of code is based on the player's distance to the enemy. If the player is far enough away from the enemy, it will meander over to

the players last known location and will not aggressively turn itself towards the player. If the player is within a set distance, the enemy will first rotate itself towards the player's position, in order to make the enemy more difficult to be avoided by the player. It will also then set its movement speed to be the sprinting speed, to allow it to fully chase after the player.

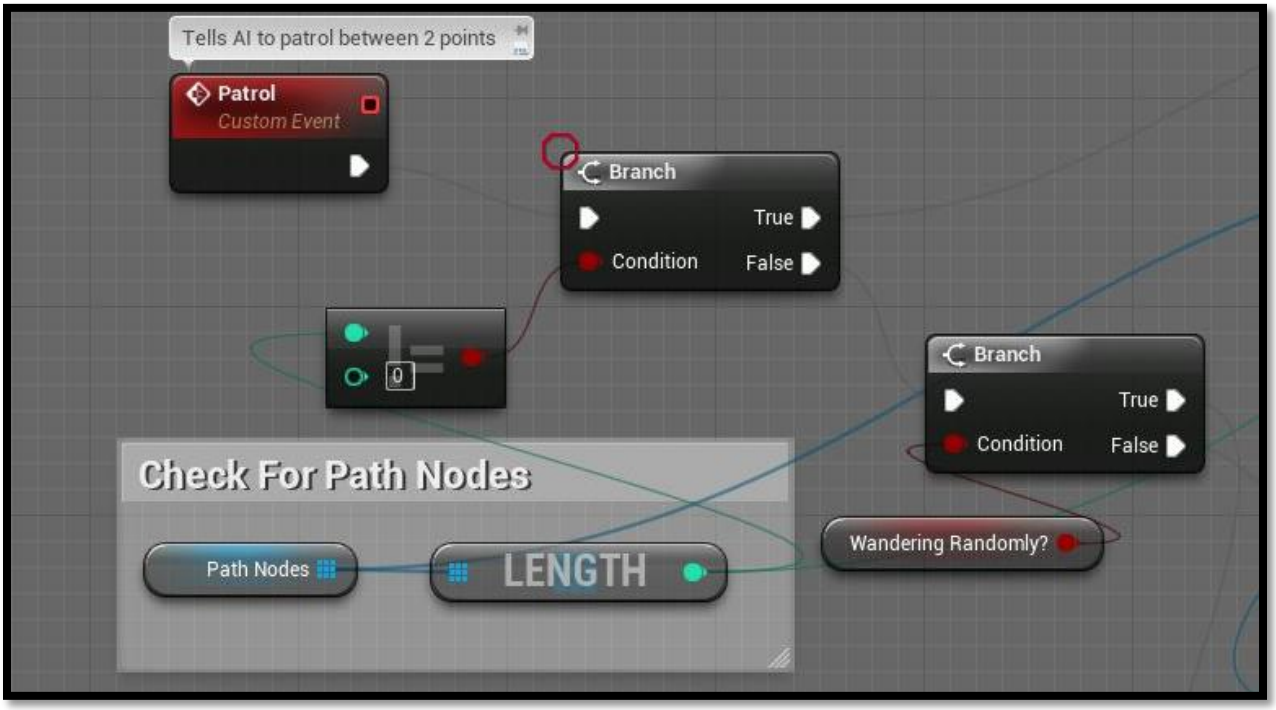

FIGURE 26: ENEMY PATROL

This block of code is the first part of the patrol functionality. It uses 2 switch statements to tell the enemy to do one of three things; patrol along a set path of nodes, wander randomly around their starting point, or to simply idle. It first checks if the enemy has been assigned any path nodes, and if so, the blueprint runs through this set of nodes. The function then goes through the list of path nodes, and continually tells the enemy to go to the next node in their list, and if it reaches the end of the list it goes back to the first node in the list.

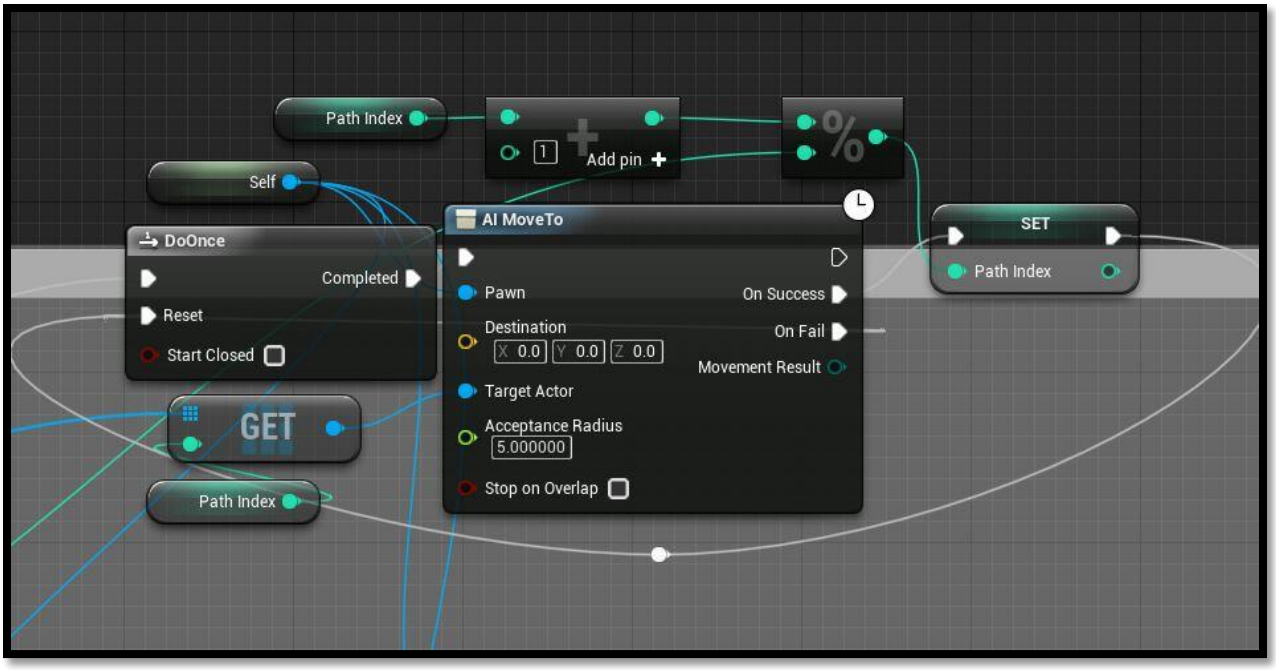

FIGURE 27: ENEMY MOVE TO FUNCTION

If so, the blueprint runs through this set of nodes. What this does is goes through the list of path nodes, and continually tells the enemy to go to the next node in their list, and if it reaches the end of the list it goes back to the first node in the list. After checking if the enemy has nodes in its path, the patrol function then checks if the enemy should be wandering randomly or not, indicated by a Boolean flag that can be individually set for each of the enemies on the map.

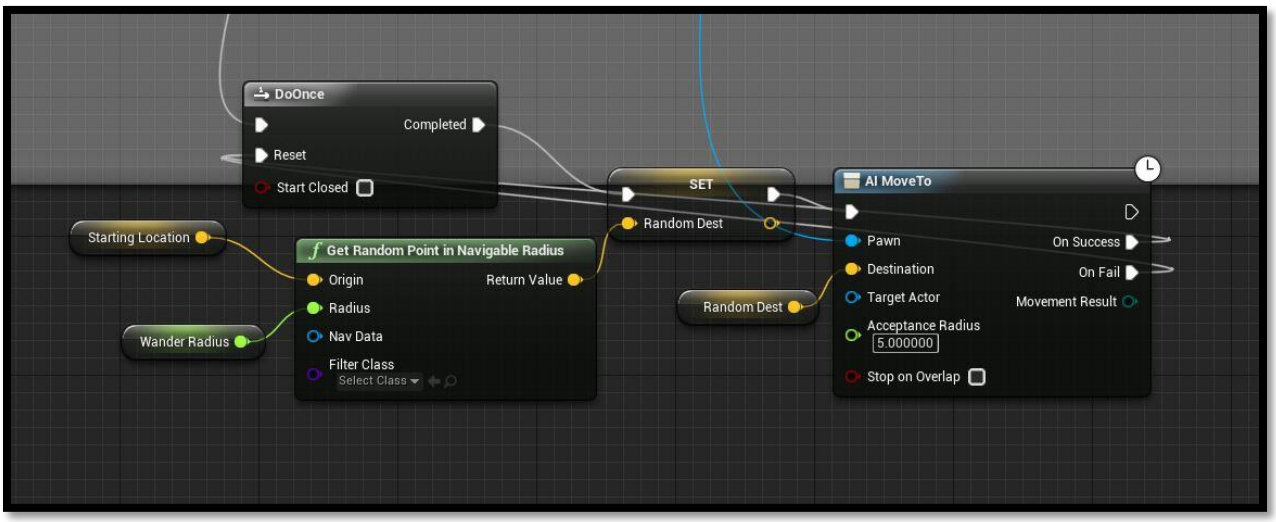

FIGURE 28: ENEMY WANDER FUNCTION

If the enemy should be wandering randomly, the blueprint runs these nodes. These nodes take the enemy's starting location in the world, and find a randomly generated location within a set radius from that point. Then, it sets the enemy to move to that random point. Every time the enemy reaches that point, it generates a new random destination.

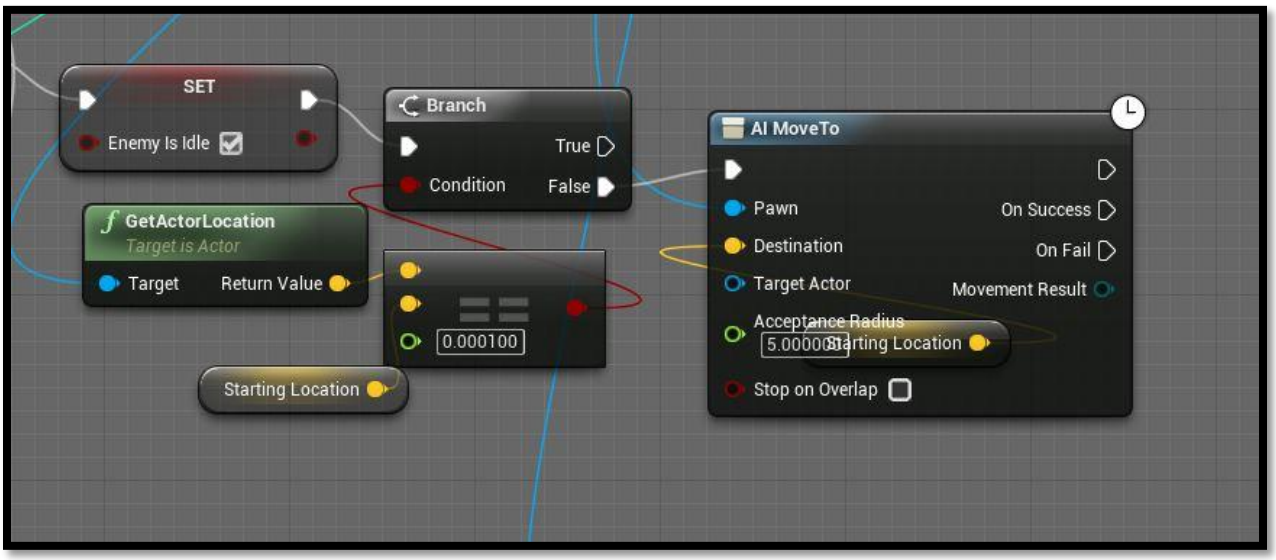

FIGURE 29: ENEMY IDLE FUNCTION

Finally, this block of blueprint dictates the action for an enemy that should be idling. This simply checks if the enemy is at its starting location, and if it isn't it tells the enemy to go back to its starting point. Thus, if an idle enemy sees the player, chases them, and loses them, it will return to its initial location.

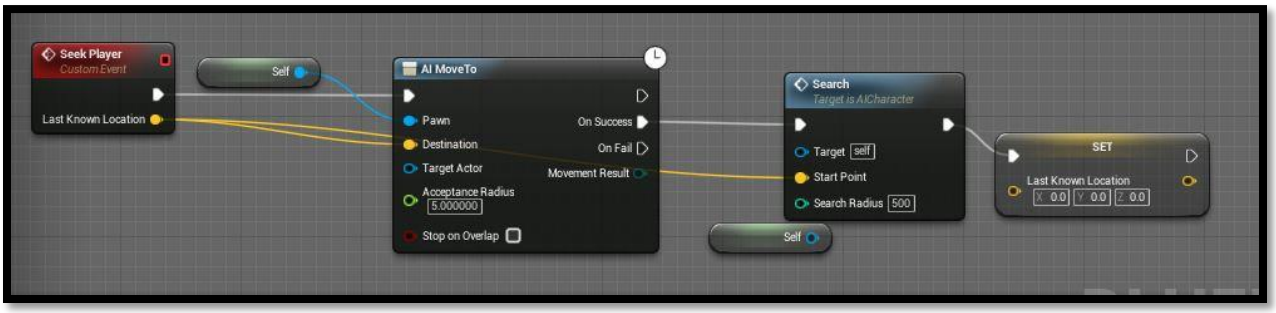

FIGURE 30: ENEMY SEEK FUNCTION

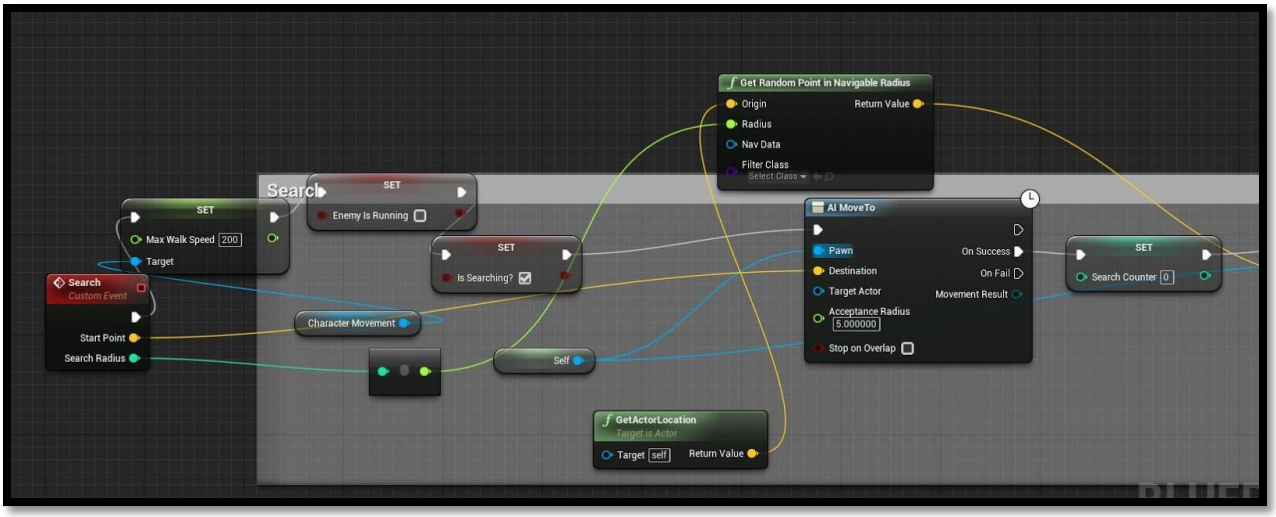

FIGURE 31: ENEMY SEARCH-FOR-PLAYER FUNCTION

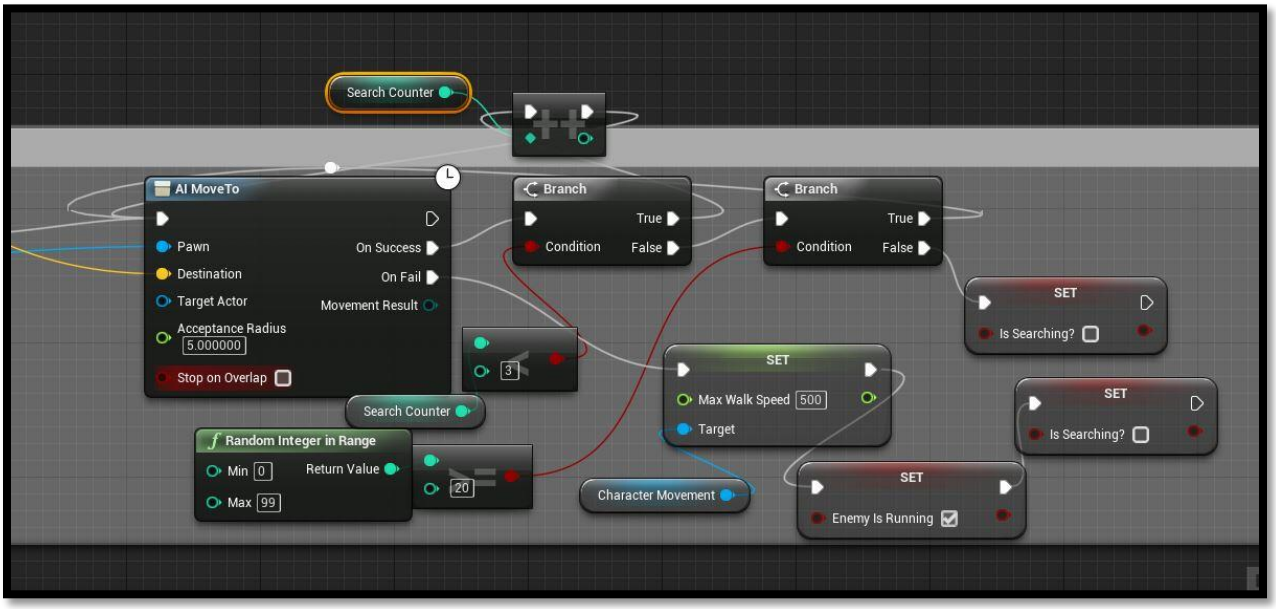

FIGURE 32: ENEMY SEARCH-FOR-PLAYER FUNCTION, PART 2

These blocks of code indicate how the enemy behaves when it loses sight of the player after chasing them. First, it enters the search state using the player's last known location. Then the player's last known location is set to zero, to indicate that the enemy doesn't know where the player is. Upon entering the search state, the enemy first sets its Boolean flags, to be used on the first shown block of code. After setting the flags, it first moves to the player's last seen location. If it reaches that point, it

indicates that the player wasn't seen on the way to that point, because if it was the enemy would move to a different point. Upon reaching that point, it gets a random position in a small navigable radius from the last seen point, and moves to that point. It repeats this action, finding new random points repeating for a minimum of 3 times. After those three movements, it continues to go to random points, but there is a constant 20% chance that after a point is reached it will return to the patrol state.

Post Mortem

1. Team Dynamics

Starting off this project, we decided to create a document titled MQP Team Contract that would explain in detail what was expected of each role within the team. The contract can be found in Appendix 2. This document also contained what we expected from each other in order to maintain a good working relationship between our team members. The four main values we addressed were communication, accountability, honesty and respect. We started to create *Displaced* with three members in our team, but ended it with two. We believe it is important to discuss the four main values we identified and how we failed to meet these standards as a three person team.

Our understanding is that communication and accountability go hand in hand. If a team member were to fail to communicate their absence for a certain meeting and they are unable to be reached by team members, they should take the responsibility of being accountable for their actions. We initially saw that a yearlong project had its very unique drawbacks, such as the feeling of having plenty of time to get the development started. This led to our team taking time for themselves that we had set aside for meetings. We had discussed that sickness and family related issues as being valid excuses to set aside our project for a day, but we had not accounted for the fact that the project may fall on a lower priority within our daily responsibilities. The document we all signed stated that every member was allowed to miss one meeting per week with a valid excuse as long as they communicated this with the rest of the team. When this rule was violated many times, we did not take immediate action. We gave our third teammate the benefit of the doubt. If we could relive this process, in order to save the unity of the team, we would definitely take quick action and intervene.

Honesty is a major component of any healthy relationship, whether it is one between friends or coworkers. Team members may not know each other prior to working together, but honesty is what keeps their working relationship in order. Without honesty, communication is very difficult, and working with individuals who are not honest can become a hindrance to any project. We experienced this is several ways such as excuses to miss meetings, low quality of work and the total hours we were each putting in individually. We believe that if we cannot trust each other, we cannot work together. During our post mortem evaluation process, we discussed ways to combat dishonesty. We believe that the best solution is to work with people who will be honest about their intentions and their actions, out of respect to the team, that there is no workaround to dealing with dishonest teammates. This was the major breaking point in our team's decision to part ways with our former third partner.

We believe that mutual respect is the driving force for the three components we addressed. One is honest if they respect another individual. One is accountable for their actions if they respect another individual and they communicate efficiently in order to keep the other party's respect as a team member in development. When we felt that our time was not being respected, we resorted to demanding this respect, only to get more pushback from our former team partner. We believe that making a clear distinction between what is acceptable to do within the team and what is clearly disrespectful would have gone a long way in solving this issue. We would suggest putting clear guidelines on what personally offends each team member on the aforementioned document.

2. Reception at PAX

For the most part, the players who tried our game at PAX East this year enjoyed the game. Roughly half of the players completed the game, and most of the players got past the first level. In general, the players were most impressed by the artistic look of each level, and the seriousness of the topic that the game covered. In addition, many players commented that the gameplay was fun, and that trying to avoid enemies made the game far more engaging than if it was focused exclusively on conveying the story.

However, while people enjoyed the game, players managed to find a large number of flaws with the game that were previously overlooked. For example, the text during the tutorial left the screen too quickly for players to read instructions given by the mother, and as such a lot of players got stuck on the first level of the game. Another critical flaw is that in the first level, the players can jump over the exterior wall and walk off of the game map.

The enemies in the game were also subject to several complaints. Some players argued that the enemies were too difficult. They were hard to avoid, placed at too many choke points, and they were relentless in their chasing. The bridge section of the second level was mainly responsible for these complaints, as it gave the players the most difficulty. However, other players made the opposite complaint, saying that the enemies were too easy to avoid. They argued that the enemies should operate on a radial detection system as opposed to a line of sight system, and that the enemies were too easy to predict once the player understood their behavior. The complete list of found bugs can be found in Appendix 3.

FIGURE 33: PLAY TESTERS AT PAX EAST

The most common complaint was a problem that was universally spread throughout all of the levels. Players took the most exception to the lack of direction in the game. In the first level, the tutorial prompts disappeared too quickly, and players rapidly lost sight of what they needed to do. In the second level, there was no initial indication to which direction players needed to travel in, and a large amount of players ended up wandering in a random direction before ultimately trying to go along the path. In addition, there was no indication that the player needed to pick up the note item in the campsite to beat the second level. A lot of players assumed that they needed to keep walking down the path, and missed the note altogether because of this. The third level was the least confusing to a lot of players, although a few didn't realize they needed to actually move into the tunnel.

If the team had more time to work on the project, we would attempt to rectify the problem by adding many more prompts, and keeping the prompts on screen for longer. We would also add a journal

that the player could pull up at any moment and remind themselves of what they need to do next. This should help users to learn what they needed to do to progress in the game a lot more readily. In addition, the team would perform much more comprehensive and extensive playtesting, to discover more exploits and overly difficult sections in the game.

Conclusion

As a group, our overall goal was to create a successful, fun, well designed game that was also able to spread awareness about the refugee crisis. In addition to the creation of a well-rounded game, the group was also responsible for self-educating ourselves to ensure that we conveyed the message of our game in a manner that was at least close to accurate. Before beginning intensive production, we decided that researching the tribulations that actual children went through should be our first task. Research was also done on the architecture of real life towns and cities in the Middle East, to impact our own level design.

After thorough research, we actually created the game, by first creating simple gray boxes for each level and creating simple artificial intelligence for enemies. For the majority of the design process, the finer details and more advanced aspects of gameplay were created by constantly iterating and enhancing the levels and code that were already in place. By designing the game in such a way, the group was able to constantly work on the specific aspects of the game that needed the most work, as well as ensuring that most aspects of the game were on the same level of completion and quality. By the end of the design process, the product was complete, was polished, featuring 3 distinct levels, advanced AI functionality, and thematic lighting.

There were a large amount of obstacles that the group underwent along the process of creating this game. The first and foremost obstacle was the problem of scoping. At the beginning of the year, the group was tentatively made up of six people. Three were designated artists, two were designated programmers, and one was to be the sound designer. By the end of the first term of classes, the artists had left the project. By the end of the middle of the third term, one of the programmers had to be ejected from the project. As such, the remaining two members of the group became responsible for creating and implementing all of the art and technology that was required for a project originally

designed for a group of six members. One of the skills that the group had to learn was how to cut certain things from the project while keeping the overall tone and design the same. Some examples of the cut content were the model and animations for the mother character, large reductions to the size of the second and third levels, and removing voice acting from the game.

The group also learned versatility. Originally, each member of the group had a distinct role in the project. Due to the large amount of member who ultimately left the project, the remaining members needed to learn to take a large number of roles outside of their comfort levels. For example, none of our team members had extensive experience in creating and working with art assets such as models and animation, but due to the lack of artists the members needed to create their own assets despite their lack of experience in order to make the game match the initial design of the game. If the group was large enough, members could have worked on and focused on the aspects of the game they each had the most experience in, but ultimately the members needed to learn to branch out and do work they had little to no previous experience with.

The overall mentality that the members adapted was that regardless of our own skill or our own desires, the thing that mattered above all else was the quality of the final product. In an ideal world, the group would have been able to implement all of their original ideas, and would never have had to leave their proverbial comfort zone. That, however, ultimately proved to be a pipe dream. The most valuable thing the group learned from working on this product was how to react to the unforeseeable obstacles that reality put in the way of the members, as well as doing so in a way that maximizes the quality of the final product.

Bibliography

1. *Ancient City of Aleppo*. *UNESCO World Heritage Centre*. N.p., n.d. Web. 27 Apr. 2016.

2. CORPUS Levant. "Tradition Architecture in Syria." *Media Corpus*. Feb. 2004. Web. <http://www.meda-corpus.net/libros/pdf_expo/2004 expo syria.pdf>.

3. CORPUS Levant. “Traditional Syrian Architecture”. *Media Corpus.* Feb. 2004. Web. <http://www.meda-corpus.net/libros/pdf_manuel/syria_eng/ats_eng_3.pdf>

4. Irshaid, Faisal. "Isis, Isil, IS or Daesh? One Group, Many Names." *BBC News*. BBC, 2 Dec. 2015. Web. <http://www.bbc.com/news/world-middle-east-27994277>

5. “Iraq.” UN Refugee Agency, n.d. Web <http://www.unhcr.org/pages/49e486426.html>

6. Porterfield, Katherine. "Resilience of Refugee Children After War." *American Psychological Association*. Web. <http://www.apa.org/pi/families/refugees.aspx>

7. “Refugees/Migrants Emergency Response Data - Mediterranean.” UN Refugee Agency, n.d. Web <http://data.unhcr.org/mediterranean/regional.php -For Statistics>

8. Reid, Kathryn. "Syrian Refugee Child: A Story in 23 Pictures." *World Vision U.S.* 07 Mar. 2016. Web, <www.worldvision.org/wv/news/syrian-refugee-child-story-in-pictures>

9. Suruc Municipality. “Photo gallery of the Suruç Municipality”. Web. 20 Apr. 2016. Web. <http://suruc.bel.tr/Galeriler>

Figures Reference List

Figure 1: Warren, Jon. *Refugee Girl*. Digital image. World Vision. World Vision ©2016, n.d. Web.

Figure 2: *UNHCR Refugees/Migrants Emergency Response - Mediterranean*. UNHCR Refugees/Migrants Emergency Response - Mediterranean. N.p., n.d. Web. 27 Apr. 2016.

Figure 3: *Traditional Syrian Architecture.* N.p.: Meda Corpus, 2004. PDF.

Figure 4: Alizzi, Giuseppe A. *Aleppo Yesterday*. Sirialibano. N.p., n.d. Web.

Figure 5: *Traditional Syrian Architecture.* N.p.: Meda Corpus, 2004. PDF.

Figure 6: *Aleppo Ancient City Victim of War*. Digital image. ODT. N.p., n.d. Web.

Figure 7: *Traditional Architecture in Syria*: *a Mediterranean Dwelling.* N.p.: Meda Corpus. 2004. PDF.

Figure 9: *Syrian Rebels*. Digital image. N.p.: Behind The New Israel., n.d. Web.

(All other figures have been created by our team.)

Appendix 1

Model Name	Level to Use	Website	Author	Link	Texture (if externally acquired)
Bridge	2 and 3	Turbosquid	igorlmax	http://www.turbosquid.com/FullPreview/Index.cfm/ID/666679	-
Gate	1 and 3	Turbosquid	Vegarian	http://www.turbosquid.com/FullPreview/Index.cfm/ID/863708	-
Chair1	1, 2 and 3	Turbosquid	Austro	http://www.turbosquid.com/FullPreview/Index.cfm/ID/791045	-
Bread	1, 2 and 3	TF3DM	azenichi	http://tf3dm.com/3d-model/bread-43520.html	-
Tractor	2 and 3	TF3DM	3dregenerator	http://tf3dm.com/3d-model/tractor-75027.html	-
Niche	1	archive3d	-	http://www.archive3d.net/?a=download&id=1957bd0d	http://hhh316.deviantart.com/art/Seamless-marble-texture-163763954
Table with Cloth	1	TF3DM	akimka	http://tf3dm.com/3d-model/table-with-cloth-57780.html	-
Box	2 and 3	Turbosquid	CG_monsi	http://www.turbosquid.com/FullPreview/Index.cfm/ID/883775	-
Dirty Box	2 and 3	Turbosquid	CG_monsi	http://www.turbosquid.com/FullPreview/Index.cfm/ID/883775	-
Carpet1	1	Turbosquid	muresanvlad	http://www.turbosquid.com/FullPreview/Index.cfm/ID/769728	http://www.textures.com/download/persiancarpets0016/21999
RoundCarpet	1	Turbosquid	muresanvlad	http://www.turbosquid.com/FullPreview/Index.cfm/ID/769728	-
Pillow	1	??	??		
Stop Sign	1 and 3	TF3DM	tyrosmith	http://tf3dm.com/3d-model/stop-sign-58062.html	http://www.textures.com/download/signstop0006/5417?q=stop&filter=all
Plant	1, 2 and 3	TF3DM	animatedheaven	http://tf3dm.com/3d-model/plant-model-352.html	
Rock 1	1, 2 and 3	TF3DM	tyrosmith	http://tf3dm.com/3d-model/low-poly-rock-4631.html	
Tree	2 and 3	TF3DM	silviuq12	http://tf3dm.com/3d-model/tree-67970.html	
Ornate Door	3	archive3d	Yun Sun	http://archive3d.net/?a=download&id=909e510a	
Brown Door	1 and 3	Turbosquid	AlirezaSadeghi	http://www.turbosquid.com/FullPreview/Index.cfm/ID/612776	
Urban Debris	2 and 3	TF3DM	adithep001	http://tf3dm.com/3d-model/urban-debris-85706.html	
Rusty Pipe	3	TF3DM	vergiliusz	http://tf3dm.com/3d-model/rusty-pipe-60301.html	
Couch	1 and 3	TF3DM	silviuq12	http://tf3dm.com/3d-model/living-room-interior-home-decoration-furnishing-10402.html	
End Table	3	Turbosquid	Fworx	http://www.turbosquid.com/FullPreview/Index.cfm/ID/166355	
Street Light	3	Turbosquid	bartonpartners	http://www.turbosquid.com/FullPreview/Index.cfm/ID/473402	
Queen Bed	3	Turbosquid	Seekdestry	http://www.turbosquid.com/FullPreview/Index.cfm/ID/617314	
Infinity Blade Plains Pack	1,2 and 3	UE4 Store	-		

Appendix 2

MQP TEAM CONTRACT
Date:

_____

A. MEETINGS

1. Meetings will be held on the following days:
   Meeting with Advisors:
   Friday 1-2 p.m.

   Core Meetings:
   Determined each term

   Other additional meetings during a given week may be added if the opportunity arises.

2. An agenda for each meeting will be issued 1 day in advance of the meeting.
3. Meeting Minutes for each meeting will be issued 1 day after the meeting.
4. We will make team decisions by majority vote, but strive for all in favor.
5. If a team member knows that they cannot make a meeting, he or she must inform the team members 48 hours in advance in case of a scheduling conflict, and if the meeting is with the advisors, inform all parties by emailing the group alias.
6. In case of emergencies such as health problems, the team member must inform the team members as soon as they are able to, but no later than 1 hour in advance. This can obviously be excused if you are rushing to the hospital or something of equal urgency.
7. The cap for missed meetings is 2 hours out of the 6 mandatory meeting hours each week, if excused. Random catastrophes and the like will also be excused.
8. The necessary hours each team member has to put into the project each week is at least 14 hours per person, excluding meeting times.

B. ETHICS
Team members will:

1. Only commit to what we are capable of doing.
2. Focus on what is best for the team and the project.
3. Promptly notify the producer of any changes that could potentially affect the project's success. .
4. Respect the confidentiality of our project and share essential information with team members only.
5. Meet goals and deadlines on an individual level.
6. Meet goals and deadlines as a group.
7. Follow the Gantt chart to complete the project on time.
8. Anticipate problems and devise solutions before they occur.
9. Not edit the work of their peers without permission.

C. GUIDELINES
Team members will:

1. Be open and honest in dealing with other team members.
2. Listen openly to other team member's points of view.
3. Build on other team members' ideas and achievements.
4. Each member should keep in mind that the ideas they come up with might not be perfect, and should strive to be open to constructive criticism.
5. Respect the time and effort of other team members and delegate tasks when necessary.

6. If a particular task is giving a group member excessive trouble, that member should inform the rest of the group as soon as possible.
7. Pick up where another team member has failed instead of focusing on the negative, unless this is repeated behavior. Team members understand that the important outcome is that the work got done well.
8. If at any point, a team member chronically fails to deliver work or is late to meetings or misses meetings, oral warning, followed by written warning will be issued by the other members of the team. If repeated after these measures, the advisors will be brought into the conversation to help find a solution to the issue.

Signature of Team Members:

_____
_____
_____
_____

Appendix 3

DISPLACED - LIST OF KNOWN BUGS			
Code Name for Bug	**Description of the Bug**	**Found Level**	**Game Breaking?**
Water	Enemies go in the water and get stuck.	Level 2	No
Grab	Enemy grab animation plays repeatedly as player is caught, appearing as if he is running around the player.	All	No
Map Bounds	Map bounds have a hole that allows the player to jump off the terrain.	Level 1	Yes
Note	The note that lets you transition to the next level has physics. Jumping on to it makes it clip through the terrain, makin the player unable to reach it.	Level 2	Yes
Inventory	After their second death, if carrying any items, the player will be unable to use any items in their inventory.	All	No
Rocks	Between the two rocks right underneath the bridge, the player has to sprint to be able to reach the other side. Often the player runs out of stamina to be able to sprint and there is limited chances to get water or food.	Level 2	No
Hand after Matinee	After the matinee of enemies leaving campsite, the camera is moved to a lower location, creating an odd hovering hand instead of the correctly placed one. This is fixed if player moves on to the next level.	Level 2	No
Tree Texture	The bark texture for trees is missing.	All	No
Kitchen	The kitchen counters and door are scaled to be too large in the bakery building.	Level 3	No
Crouch	When crouching, the hand mesh of the player moves abnormally in relation to the camera.	All	No
Tutorial Text	The mother tutorial text fades too fast, and the hand can sometimes be located behind the text, making parts of the text hard to read.	Level 1	No
Item Persistance	Items do not carry over between levels, and are too sparsely placed in Level 2 and 3.	Level 2 and 3	No
Shadow Rendering	The benches behind the bonfire have falsely rendered shadows that look odd.	Level 3	No
Bonfire Blindness	The bonfire in the town square is too blinding, that the player is unable to see the enemies approaching.	Level 3	No