

# Behoshee - Case Study

## **Introduction**

The choice to change the game from a stealth adventure to a survival one was taken based on the knowledge-supported design judgement that a survival game would be more enjoyable and challenging for players. This led to the decision to switch the focus of the game from stealth to survival adventure. In the early phases of development, the concept of a stealth adventure seemed intriguing; but, as development advanced, it became obvious that the gameplay elements of a survival game would create a more engaging experience for the player. As a result, the game was redesigned to focus on survival rather than stealth. The core narrative is still part of the game where a player needs to gather memories in order to recall who they are.

## **Game Features**

In-game, you'll experience everything from a third-person perspective. In addition to walking, and running, the player can also jump. Stones thrown at the enemy may serve as a diversion, but the ability to manipulate time is crucial. The time-bending effect will slow down the movement of enemies and some hazardous objects, giving the player more time to either outrun or avoid them.

Memory orbs are the main feature of the game, when the player collects the memory, a part of the memory is played which will help the player to understand the protagonist of the game. As the player progresses the game will be much clearer.

## **Game Development**

### **Time-Bending**

As soon as I had the player controller figured out, I got to work on the most important aspect of the game, which is called "Time-Bending." Instead of slowing down the time in unity, I would slow down the time of animation of the object I want to get slowed down. The reason for going forward with logic was that when I was using the unity time

(Time.deltaTime), it was slowing down the whole game, including the player, so I came to the conclusion that I will slow down the animation. After a lot of testing and implementing new methods to slow down the time, I came up with a simple logic. Therefore, in order to give the impression that the enemies are moving at a slow pace, I slowed down both the animation and the chase speed that they use. The object's slowing down uses the same conceptual framework.

## **Memory Orbs**

Another item that was high on the list of priorities was the memory orbs. Although I am aware that I may initiate the cut scene by triggering the object, the most important aspect was determining when the scene would end given that the duration of each of my memories will differ. To compose and edit the clips, I first produced some storyboard tiles for the video, and then I imported those tiles into After Effects. This gave me the ability to do so. The process of creating orbs with Unity's particle effect was quite enjoyable. Because I wanted my memories to have the appearance of memory, I decided to go with a design that consisted of an orb with a circle in the centre and other particles surrounding that circle. Adding a collider to the memory using the trigger box check as the method of attachment. My memory was able to play the clip when the player was colliding with the orb.

## **Huddles and New Implementation**

The main huddle was when I was designing the enemies. I have done a lot of research to design AI behaviour on YouTube, Unity Forms etc. But I cannot find the right way to execute it. When I tried to implement an enemy with a Raycast system, when the player hits the Raycast the enemy would chase and attack the player. But when I executed this, the player speed was much faster and it surpassed the Raycast very fast thus, the enemy was not getting enough time to register the position values of the player, so they were not able to catch the player.

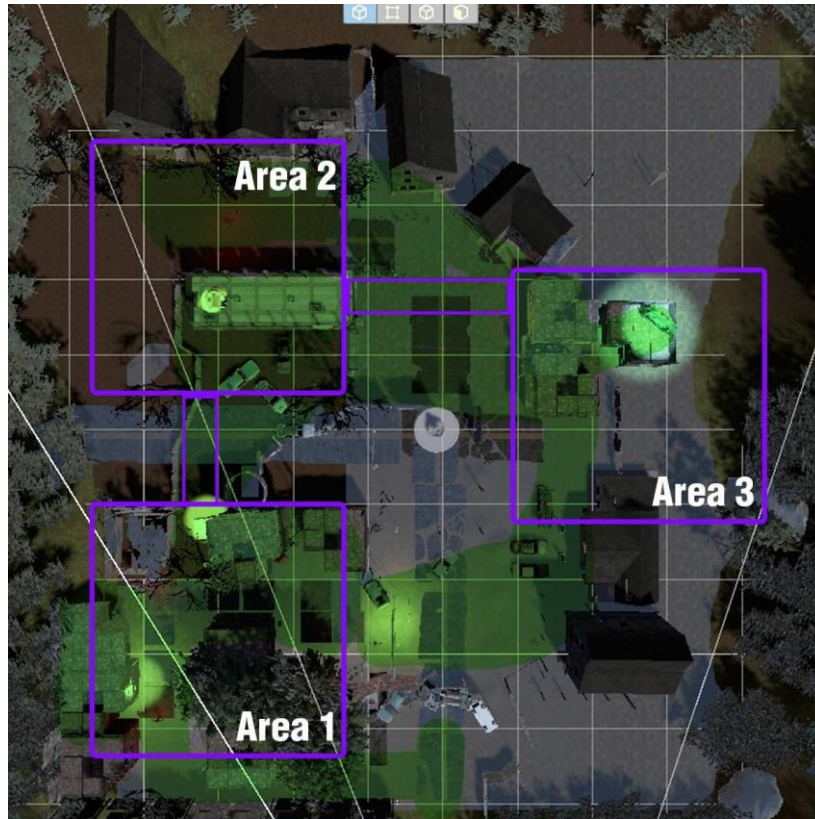
Consequently, I merely wiped the whole thing away and began developing brand new strategies that were straightforward for the enemy. Which will patrol the waypoints as well I have introduced randomization between the waypoints, so that the patrols are varied and enemies are less predictable. This will ensure that the waypoints will be

patrolled. In addition to that, I have included a radius for the enemy to check to see whether the player is inside the radius before deciding whether or not to pursue and attack the player. After the player displays that they are quicker than the enemy, the game will take them to the nearest waypoint. The present method makes use of the navMesh component that is available in Unity. This component assists the adversary in determining which regions are permissible to walk through and which parts are off-limits. I was able to successfully integrate all of this, but doing so will alter the gameplay significantly; in fact, this is the primary reason why the focus of the game will now be on survival rather than stealth.

There were a lot of little obstacles in the way of the process, such as when I changed the clips in one memory orb it would get replaced in all of the memory orbs, so to get around this I just made a prefab of the memory object, which helped me solve the problem. There were also a lot of little huddles like this in the way of the process.

## **Level Desing**

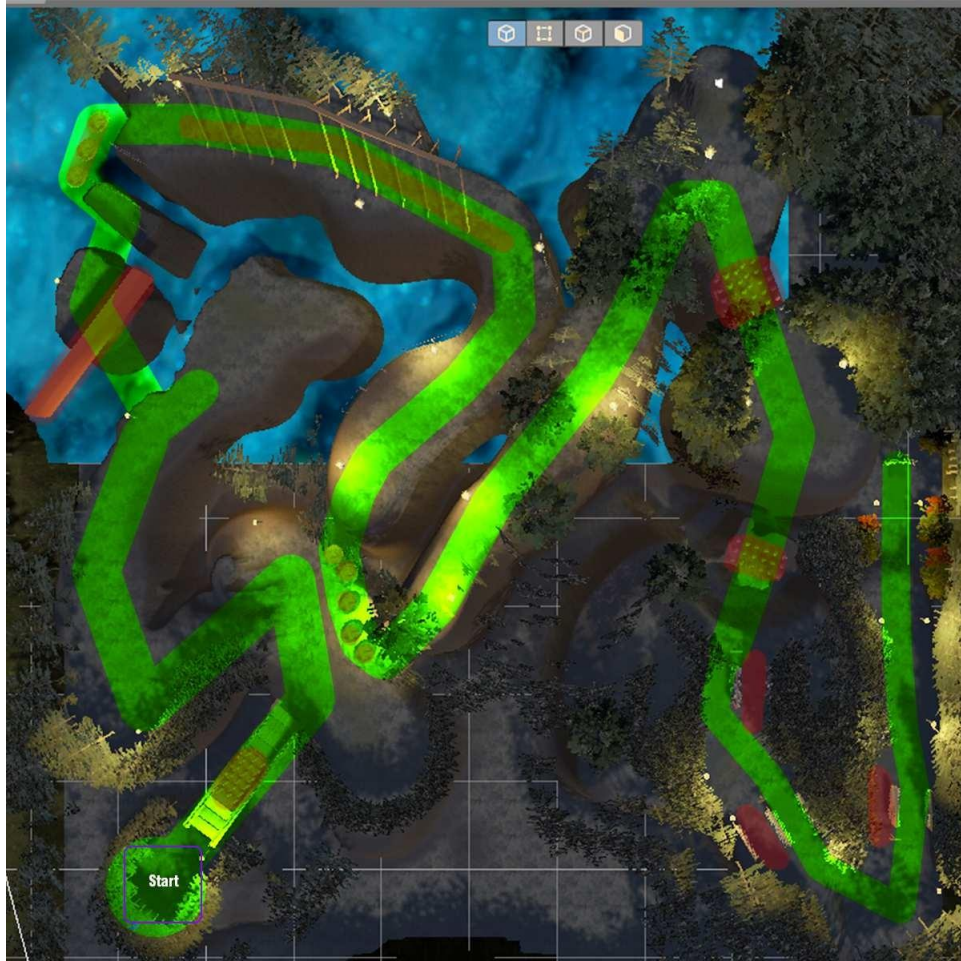
When the behaviour of the enemy change level was about to change, where the in the pitch I was going implement a first level inside a house, it got changed to an outside level. At first, I designed an open space where the player needs to roam in the empty town where they have to find the memories which will be scattered all over the place, but in one of the feedback I was suggested that it would be great if I could design the level a little linear so that player knew which area to start from. Thus I altered level one in such a manner that the player will guide to the area which has the first memory orb and thereafter the second area. But in this area, there are multiple ways to collect memory.



[Level 1 Diagram]

The player can move freely inside the green region, which denotes the playable territory. Additionally, you are able to see that all of the regions in the level are linked together via linear space. Which, in turn, will lead the player to the next area. Also, the enemies are placed statically in every area.

The setting of level 2 has been altered to take on the appearance of a forest, and this level also features the introduction of a few hazardous fast-moving objects. There are a variety of hazards at this level, including spikes, falling rocks, pendulums, and floating plates. The player has the ability to control the time and the speed of these hazards, allowing them to pass through the level and continue on to the next one. This area of the map has three memory orbs, and once the player has collected all of the memory on this level, they will be transported to the next and final level through a green teleporter. All the hazardous objects in this level are made with the pro builder plugin of Unity.



[Level 2 Diagram]

In the last and final level, the rules are simple the player needs to run and free our protagonist “Akira” which will help her to wake up in the real world. The level is very straightforward. In this level, the hazardous element is the poisonous fog which will be flowing the player throughout the level, if the fog catches up to the player then the player will die and has to restart the level. In this level, there are rocks falling on the player, and they need to dodge them and outrun them, the rock will also damage the player's health. The environment of this level is just a simple terrain with mountains where starting part is forest, the middle part is desert and the last part is ice. In the end the player has to go through the teleporter and there will be an ending scene after that the player goes back to the main menu.





[Level 3: Diagram]

## User Interface/User Experience

In this, I have used the radial dial to fit the aesthetics of the game, the bar is commonly used in most games thus creating it lower profile. In early planning, I was going to put a player avatar in between the health dial. But then I dropped the avatar idea and replaced that with a time-bending area. When the player is getting hurt there is a hurt indication by showing the blood splatter on the screen, this way the player will get to know that the character is getting hurt. When the player is using the time-bending ability, there is a visual indication a shock wave is formed around the player. The memory orbs have a 3D sound in them so that they will get to know the location of the memory when they are nearer to them. The enemy also has a scary sound so that the player can notice that sound. In the UI/UX segment, the main huddle was the sound. Somehow the sound was not rendering properly, and I started getting an audio loss. This bug is not

yet solved, but this forced me to use visual indications more. A simple main menu and pause menu are also present in the game.

Apart from Jacobs's law, I have also implemented the law of feedback is applicable to this situation. It is necessary, while attempting to create an immersive and engaging experience for the player, to make use of both visual and audible feedback in order to communicate to the player the current state of their health, the activation of special powers, and the vicinity of critical things. According to this law, giving the user with timely and useful feedback is one of the most important things that can be done to create a pleasant experience for the user.

The game concept is continuously refined and improved as a result of feedback obtained through playtesting, which occurs throughout the development process. I relied on my previous expertise with playtesting and user research to make sure that the game was well-balanced, exciting and offered a satisfying experience for the player.

In conclusion, I would want to say that this project has been one of the most major endeavours for me, and I am really pleased with how it turned out. This experience has taught me a great deal, from the fundamentals of game creation to specific technical abilities, and it has also provided me with a wonderful opportunity to demonstrate what I have learned. I have a lot of delight in what I have accomplished, and I firmly feel that working on this project has contributed to my development as a person and in my portfolio. I would want to express my gratitude for the chance to develop this project, and I am looking forward to continuing to use the information I have gained in future endeavours.