Document for Level Design Project

In this project, I designed a new level based on the start code of the Unread Learning Kit. This is a complete game with enemies, pickups, health kits, etc. My design is based on several steps, and each step works for a different part of the game. In this document, I will introduce each of the steps and the way to design.

First of all, I create the environment of the whole game. I put several islands inside the map and made them follow from beginning to end. To show the player the way to the end, I put a shining pink cloud on top of the last island. Then after putting all the primary islands, I arranged their location of them and added some connections like stairs to connect all the islands. Then add some buildings and grass to make the game looks better and more attractive. After all these, the basic structure and layout are finished.

Then I start to build the enemies. There are three different kinds of enemies. The first one is the pursuer. This kind of enemy will be looping patrol path that it follows; after it detects the player, it will chase the player and hit the player; this will Knock the player back and remove player control for a short duration, also decreasing the health of the player. I implement this by first making the enemy patrol on a specific route, then using "detect player" as a trigger to trigger the event that chases the player. If it can't detect the player, it will return to its patrol route. Then the second enemy is the mortar. This is an unmoving enemy placed on the ground and will shoot projectiles randomly at all the places around it. Also, for this enemy, I call a particular function that, once it detects the player, will shoot a different projectile at the player and face the play as long as it can see the player. This makes the game more interesting for the third enemy. I make it able to teleport, I set several points, and this enemy can teleport to any of those points randomly. Also, it will shoot projectiles at the player once the player is detected. All the enemies will be destroyed when the players collide with their "heads." The player can do this by jumping to the head of the player. Also, the player can shoot a projectile. On the other hand, if the player collides with other parts of the enemy rather than the head, the player will be knocked back, and his health will get reduced.

After finishing all these three kinds of enemies, I create a health system and a collection system for the player character. The health system will show the player's current health, and the collection system will show how many gems and coins the player has. Also, connect this system with the event when the player gets hurt, or player gets some pickups. Then I add the projectile for the player, and it shoots the projectile that can kill the enemy when collied. After all, these have been done, all the essential elements are finished.

Then the next step is to place all the enemies and pickups in suitable places. I put many different enemies on the way the player must go across to block the player's path. Also, put health kits after the enemy so the player can increase their health once they get hurt. Finally, put a total of five gems and five coins on the map and put the last gem and coin at the end island to make the player win once they reach the endpoint of the map. He will win if the player can survive and get those gems and coins.

The last step is creating the different widgets, one for the health bar, one for the collection, one for restarting the game, and one for completing the game. Connect each widget to the corresponding events. This includes when a player's health decrease to 0 or falls down to a specific height, and then the restart widget will show up; When play gets both five gems and five coins, the complete widget will show up. Also, the health bar and the collection will show on the top left and bottom left corners to show the current health and collection.

After finishing all the steps above, I complete the game's design, which is challenging and exciting.