

CECS 528 Homework Assignment 3

Download the Unity game projects, ShootingGallery and CB_SurvivalIsland_2012 from the Blackboard. Test play these two games.

1. (20 points) Describe your game experience of playing CB_SurvivalIsland_2012 and explain how to complete the game.

The game takes place on a large tropical island with palm trees, rolling hills, a volcano, and large grass plains. The player can walk all around the island and into the blue water, but the player can eventually fall perpetually if the player wanders too far away from the island and out of the water. In the version of the game distributed on Blackboard, the player cannot jump and so spawns on one of four cube-shaped rocks. On top of this rock is a battery that the player can immediately go forward and “pick up” by touching the collider attached to the battery, making the large battery disappear from the game and into the user’s inventory of batteries. The player has a battery icon with four slots to indicate the required amount of batteries to obtain.

The player then is to talk toward a campfire, which clearly announces via UI text to the player that “I’ll need some matches to light this camp fire...” This is a clear prompt giving the player directions on what to do. The player then searches the nearby house, suspecting that the matches are inside the house. The player cannot enter the house, however, as it prompts the player with UI text reading “The door needs more power...” This is a clear prompt that the player needs to gather more energy from picking up more batteries. The second battery is right to the left of the front of the house; the third battery is to the right of the house.

The last battery that needs to be gathered is earned by walking on to the platform with three targets. When the player walks on to the platform, the player is prompted to “Knock down all 3 at once to win a battery!” The player then realizes that this is the fourth and final battery required to unlock the door and obtain the matches so that the player can light the bonfire and complete and win the game. The player then has to quickly aim and knock down all three targets with unlimited coconut ammunition. If the player is not fast enough in knocking down the targets, however, the targets will rise back up after a certain amount of time and the user must attempt knocking them down again, but quicker. Once the player is able to successfully knock down all three targets in rapid succession fast enough, a fourth battery spawns in front of the targets so that the player can go and grab the fourth and final battery.

Once the player collects the fourth and final battery, the player has enough power to open the door to the house. The player can then walk inside the house and pick up the matches by walking over them inside the house. The player then wins and completes the game by going to the fire and immediately lighting up the bonfire.

This was a fun game to play and the ambient music and beautiful scenery made the game even more fun to play. The FPS shooting of the targets in rapid succession was actually challenging for me and I felt accomplished and had fun while trying to knock them all down. The game also had a bit of a puzzle aspect to it, for it requires the user to think about how they can light the bonfire and how they can gather power to open the door to the house. Overall, this was a very fun first person shooter/puzzle/survival game, for it combined many different fun aspects of different genres of video games into one game cohesively and smoothly. It is definitely a game with much potential that could continue to be fleshed out with more items/objectives/puzzles and could be made into a much longer and more challenging game if desired.

2. (20 points) Describe your game experience of playing ShootingGallery (scar-L rifle shooting and ball throwing) and explain the difference in ball throwing gameplay if applying or not applying the gravity on the throwing ball

This shooting had a scar-L rifle model, a series of moving targets that can be shot and disappeared with particle effects, and some dynamic cans that can be shot down as well from their original pyramid position. I began playing by clicking on positions on the screen that would immediately throw a ball with a smiley face on it and direct it full speed in that direction. If I aimed it just to the right of the targets moving to the right, the ball would often hit the targets and the target would disappear into particles using particle effects. Every time a target is hit the "Hit: 0/0" increments the first number. Every time a ball is thrown, the second number is incremented regardless of what the ball hits. For instance, if I click and successfully hit a target, the UI on the left side of the screen will increment to "Hit: 1/1," but if I then miss a target on my next throw and the ball just lands aimlessly in the black box, the text will update to "Hit: 1/2."

The balls always disappear after they are thrown even if they are thrown into the abyss to the side of the box. Balls thrown into the abyss can forever be seen to continue going forward until their Timer runs out and the balls are destroyed. The player also has a scar-L rifle in the center of the screen. The angles of the gun can adjust its angle of shooting along the local z-axis by using the up and down keys. The local y-axis angle of the gun can also be changed using the left and right keys. The player can then shoot the gun by pressing space. This fires a red capsule-shaped ball that is spawned right at the gun's muzzle and is thrown with a force that is a ray pointing directly from the angle of the gun forward. The bullets shot from the gun have the same scoring behavior as the balls discussed in the previous paragraph, incrementing points if the targets are hit with the fired bullets successfully and always incrementing the total number of projectiles thrown regardless of if the bullets successfully strike the targets or not. The cans can also fall off the table if they are hit at the right angle and position to knock them off the counter, but they don't increase the number of targets hit when they are knocked down.

I found it much easier to hit more targets with the thrown balls via clicking, for I found it way easier to be more accurate this way. This is because I did not have to estimate the angles when I was throwing the balls; the balls would always be thrown pretty much at a straight angle from where I clicked. When firing the gun, however, I had to estimate the gun's muzzle angle relative

to the target. As a result, I was way more accurate in hitting the desired targets and cans when throwing the balls and not firing the rifle.

When gravity is not applied to the balls thrown balls, the gameplay of throwing the balls via clicks is changed. The player is forced to aim a little higher than usual to account for the changes of gravity on the ball. Gravity also makes the ball feel “heavier” and makes throwing more challenging. When gravity is *not* applied to the ball, the ball is easier to throw because it always goes directly to the spot clicked by the player. It’s thus easier, more accurate, and feels faster and lighter to throw the balls without applying gravity. It does feel a bit unrealistic when the ball does not have gravity applied because the ball can, for instance, float towards the top of the scene after it misses a target and bounces upward after hitting the black wall behind the moving targets. In summary, gravity makes the ball feel more realistic and heavier while lack of gravity makes the ball feel lighter and easier to aim and throw.

3. (10 points) In the CB_SurvivalIsland_2012 game, we could attach CoconutThrow.cs to the Main Camera game object rather than to the Launch game object (see page 6 of the Aiming and Firing 1 notes). Explain what the effect we want to accomplish by using the Launch game object with proper transform settings (Position and Rotation). Also explain the transform settings.

When CoconutThrow.cs is added to the Launcher game object that is attached to the camera, the script uses the local rotation and transform variables that are attached to that game object. The exact desired starting position (with an x value shifted right 1) and starting rotation of throwing the ball from (in this case, a Y rotation of 352) makes the ball achieve the effect of making it appear like the coconuts are being thrown from from the player’s right arm rather than just spawning a ball magically from the eyes of the player. This makes the gameplay feel more realistic because the rotation of the ball makes it appear like the player is actually throwing the coconuts from their right arm and hand rather than from their eyes.

If the CoconutThrow script were just added directly to the Main Camera, the coconuts being thrown would have the same rotation as the camera and would always shoot directly from the center of the screen, making it appear like the coconut was thrown from the player’s head and not from their arm, which has the effect of feeling unrealistic. By adjusting the Position and Rotation to always be the same, the player can look anywhere with the Main Camera and the coconuts will always be thrown from the same initial position and rotation relative to the player and not just to the main camera’s world angles and position.

The transform setting of the Launcher object has an x position value of 1 so that the coconuts are spawned from the right side of the screen to make it appear like the player is throwing the coconuts with their right arm. The transform y and z positions are set to zero so that it appears that the player is throwing the ball at a realistic height. The angle of 352 of the Y rotation is created so that it appears the player always throws the coconut at an angle relative to the player’s arm. The X and Z rotations are 0 so that the thrown coconut always appears to be realistically spinning like a throwing motion of an object would rotate in real life.