Adam Taylor

CPTR355

Dr. Foster

March 3, 2022

Hedge Project description

I have created a hedge maze in Babylon.js that allows the user to walk through it in first person and navigate it to reach the end. There are a few short dead ends, a dead end loop, and a pretty long dead end that forces the user to actually try it out and has the possibility of them not getting it the first try. Because of this and if the user is feeling overwhelmed, at the start there is an elevated platform so they can plan their route before they begin if they would like. This maze also has parkour in it that require the user to jump, using the space bar, from one moving platform to the next. The platforms do not move the player; this adds a difficulty to the play of tracking the platform with their movement as well as planning their next jump. If they fall on this jumping puzzle, they are reset to the beginning of the jumping puzzle, requiring the user to get it right all at once. Then at the end there is a particle effect and a congratulations message to show the user they have successfully made it to the end.

* Shapes – there are multiple boxes of different shapes and sizes, as well as paneled boxes and normal ones. There are cylinders with tessellations enabled and a zero bottom radius to make a pyramid. There is a paneled ground to create a nice texture for the ground. There is a plain to make a ramp up and down from the jumping puzzles. There is also a rock shape but that uses blender as well.
* Position, rotation, sailing – The ramps are rotated and translated so they fit their given areas. The hedge maze is created by translating and copying one original mesh over and over based on a matrix to create that. The rocks are created by translating and rotating the rock to create a wall of rocks that doesn’t look entirely uniform and they are scaled from their original size to make them look like an actual barrier.
* Material – I used a significant amount of textures form my different object. The ground and ramps have one texture, the moving platforms have a texture, the starting and ending parkour platforms have a texture, the hedge has a texture, and the elevated platform at the beginning has a texture. There is also the skybox if you count that. Also each mesh has had their specular color suppressed, ambient color increased so everything isn’t dark, and other mesh to mesh based changes.
* Cameras – I am just using a single free camera with edited controls to ‘a’ ‘s’ ‘w’ ‘d’, minimum render range, and speed, as well as a constant application of gravity to it so you continue falling even if you aren’t moving the camera.
* Lights - I am also only using one light source. It is a white point light that is positioned so it looks like the sun on the sky box is the light source. It has specular light removed and intensity dialed up.
* Animation – There are two sources of animation in my program: the player jump, and the moving platforms. The player jump uses the animation method to move the player up and then allows gravity to take them back down to the ground, with a brief pause to make sure the player has actually reached the ground. Then the platforms use a render loop to constantly move them back and forth, changing Boolean values to make sure they reach the points where the start going the other direction. There are vertical, horizontal, and diagonal translations in these animations.
* Collisions – The player’s camera can collide with pretty much every mesh visible. This allows for the parkour and for the maze to actually be forced to be navigated, as well as preventing the camera from falling through the world. But I also use collisions to trigger the user to be teleported to the beginning of the parkour if they fall. They hit an invisible mesh and are teleported instantly. I also use another invisible mesh to activate the end message and particle effect at reaching the end threshold.
* Sprites – I use a sprite for the end messages as well as a forest that is outside the boundary of the hedge maze and the rock boundary. I have a single sprite for the end message and a randomly create set of sprites for the forest. Both of these sprites have been created by me.
* Particles – I have two particle fountains at the end that are formed when the end plate is activated. Just a little fancy celebration at the end.
* Environment – I have a skybox but I used one from the Babylon resources area. It fits very well though.
* Height map – I created my own height map for the elevated platform at the beginning of the maze so I could create a slightly more realistic hill than just some plains or a square.
* Shadows – Basically everything has shadows. I ran into some performance issues so they aren’t very high quality shadows but they look nice. The player and the complex rocks are the only thing that doesn’t create their own shadow but everything that is in the area of a shadow receives a shadow. I also created an ambient color for everything so the shadows aren’t pitch black.
* Events – There is a keyboard observer that tracks when the space bar has been pressed to enable jumping. There is another one that listens for ‘m’ or ‘M’ to mute the music in the program. There is also an event listener that makes sure that the pointer is locked on the screen so the mouse moves the camera without clicking but I did not create this code, I just found it and edited it very slightly to serve my purposes.
* Sounds and music – I added ambient music, which I created myself through a program called music scores, that starts on the start of the program if the browser allows it and repeats forever. This music can be stopped or restarted at any point with the ‘m’ key.
* Physics – gravity has been enabled and you fall when you jump or when you fall off a platform. This is not using the complex physics engine though. Just the gravity feature.
* Gimp – I created my tree sprite, my end message, and my height map in gimp.
* Blender – I created and sculpted my rocks in blender
* Creativity – that is up for you to decide.
* Complementary features – I think that a parkour puzzle and a maze fit together quite nicely.
* Overall design – I think I created a fairly beautiful world that is fairly easy to use and still presents a small challenge to the user.