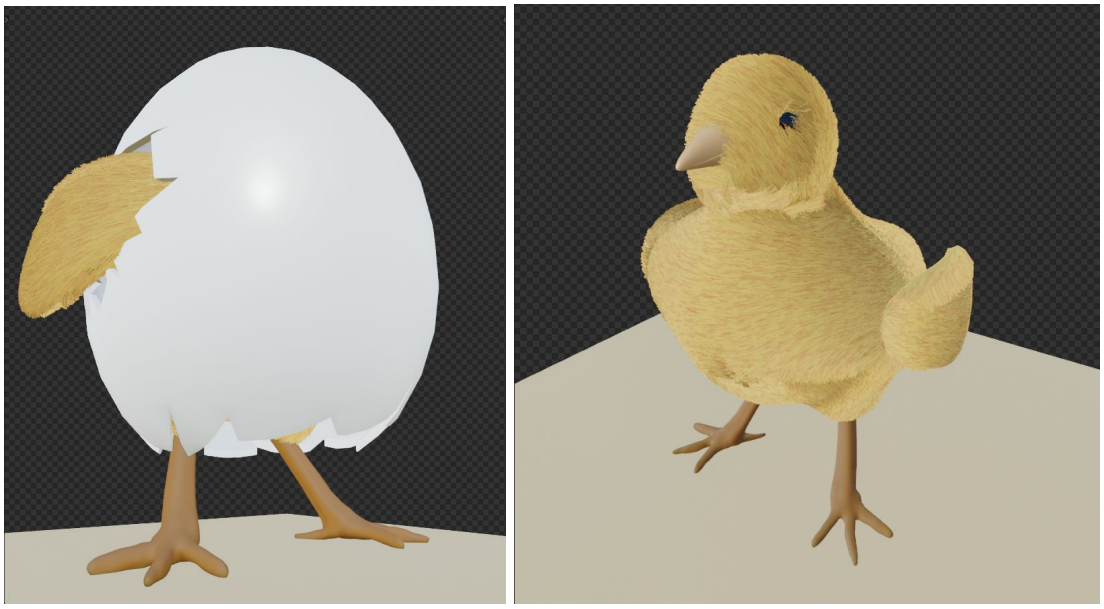
Cullen Carns
Keavon Chambers

Character - Eggbert



Our character is Eggbert, a chick that is very eager to hatch, but is having some trouble breaking out of his thick shell. Eggbert begins the game rolling out of his chicken coop and into the world. Once control of the game is yielded to the player, the player must help Eggbert break free of his shell as quickly as possible. Eggbert was able to at least break out his legs and wings on his own. This allows him to run and jump around in a traditional platforming style. However, since Eggbert is still stuck in his shell, he can also roll! This is very convenient in the rolling hill terrain he lives in. Using a combination of rolling, jumping, and platforming, Eggbert can pick up quite a bit of speed. Now he just needs to find something hard enough to help him crack open that tough shell.

Eggbert runs with a wobbly gait as he is a bit top-heavy. When Eggbert stands still he is antsy to keep moving. He desperately wants to break out and see the wide world of the farm he will grow up on.

Meeting Objectives

We used pure HTML5 and JavaScript for this project. We modeled the scene and Eggbert himself using Blender, then rendered out the scene and character model into 2D image frames. The animations were also done using Blender by animating the model and rendering out frames for each animation. We use some text on screen to indicate to the player initial instructions as well as indicating how much “shell integrity” is left to break Eggbert free. We used some sounds we found for Eggbert running around and breaking his shell, as well as some fitting background music for the farm setting and feeling of exploration. The story we attempted to convey to the player begins with

Eggbert rolling out of his coop, still stuck in his egg. The story is simple, he is trying to break out by getting speed and smacking against rocks.

Probably the most complex aspect of the game, the physics of rolling, was related to the collision aspects of the requirements. Our terrain is composed of a collection of line segments which we can actually read in and parse at the start of the game from an obj file to build the terrain. Entities in the world (Eggbert, rocks) have a basic hitbox for collisions with each other, and they also have a circle collision mask for terrain collisions. You can press X to see the hitbox info. We used circle line collisions as well as a lot of vector math to create a terrain which allows platforming and rolling along these sloped line segments. Using line segments we can create slopes, rolling hills and curves, ramps, walls, pits, everything we need to create interesting terrain.

Division of Work

Keavon was responsible for using Blender to model out the world art, level design, and character model. For engine development, he also created the render loop and sprite loading system that used some of the more advanced features of modern JavaScript to load all the assets prior to gameplay starting. Cullen primarily worked within the physics engine and gameplay code. The controls, animations, terrain collisions/rolling physics, animation and sound triggers, canvas resizing, and game mechanics.

Feedback

Overall, positive feedback for the game centered around the controls of the game, the fast and smooth gameplay, the sliding collisions, and the “speedrun” idea. Players also really liked the placeholder animations and the sounds we had in our beta version.

The biggest negative feedback we received from the QA was that our “egg game” didn’t have an egg in it. The model and animations for the character had yet to be done. Of course that had to change. One other thing a few players noted was that the game felt a little slow, especially when not rolling. We upped the movement speed of the player’s walking by about 50% and slightly increased the jump height to make the transition between walking and rolling feel a little more natural. The sample level was also very small and there’s “not much to do,” and so we created a larger level with more variety. Finally, players felt the goal was not clear. Surely part of the issue of the goal being unclear was the the player was not the “egg” character they should be. However to try to make it more clear we also created a sort of “progress bar” above the character, representing how close they are to breaking the egg. Seeing the progress bar go up after smashing into a rock should give a more clear idea of what the player is meant to do.

Note

Be sure to run the game from a localhost server, the file:/// protocol will not allow the game to run.