Team 30: Honey Collector CS 174A Final Project Report

Orrin Zhong, Rita Chen, Nikita Park

Inspiration:

Our inspiration for Honey Collector was largely inspired by the Chrome dinosaur game and a lot of the elements are really similar. The theme of our game is Baby Bear (who the player controls) is late to dinner at Mama Bear's house, but he was supposed to bring the honey! In the game, the player controls Baby Bear and tries to get as far as he can and collect as much honey as he can while avoiding all the obstacles in his path.

Controls:

There are three main game controls: v to jump, c to duck, and x to restart once the game ends. There is also a control (h) to toggle hitboxes, outlining the hitboxes on all objects, which is used for object collision detection.

Gameplay:

There are three different obstacles present in the game: ponds, logs, and birds. Ponds and logs are land obstacles, meaning the player must jump over them to avoid them. Birds are flying obstacles, meaning the player must duck under them to avoid them. There are also honey droplets that the player can pick up to gain 10 pts to their score.

The game pace ramps up over time, up to 3x the starting pace. While this means the score also increases faster, it is much harder to stay alive!

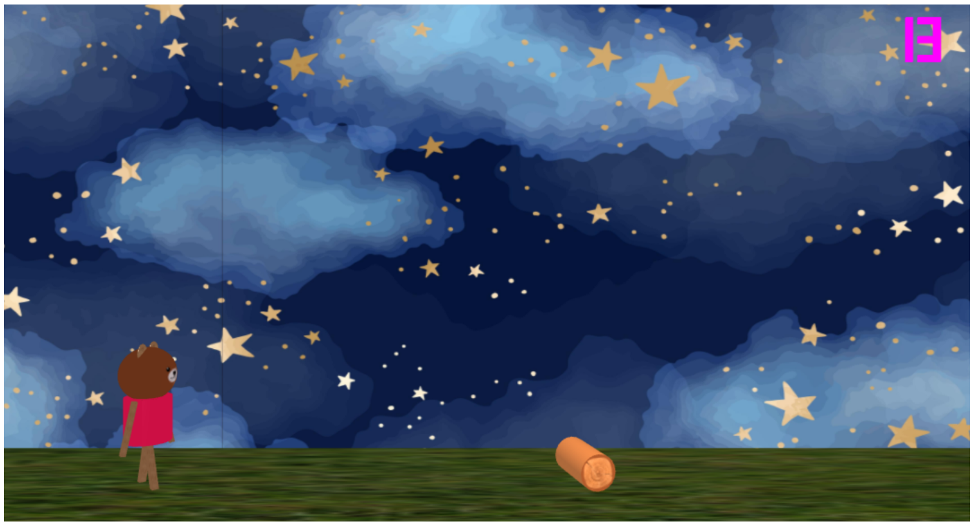

Figure 1. Game Layout

Implementation:

Player "movement" in this case is achieved through movement of everything to the left to form the illusion that the player is moving to the right. This is done through left movement of nonplayer objects as well as texture scrolling in the negative x-direction of the background on a flat square.

Object movement is implemented through setting object velocity for each object. The bear is also affected by gravitational acceleration on its way down from a jump, which will set the bear's downwards velocity. Bear jump is implemented as an impulse to the bear's upwards velocity.

Our advanced feature is collision detection and it was implemented by surrounding each object figure with an appropriately sized invisible axis-aligned bounding box. Collision detection was detected by checking if edges overshoot on all three axes. The three places we used collision detection were in: Player / Floor interactions to make sure the player does not fall through the floor, Player / Obstacle interactions to make sure the game ends when a player hits an obstacle, and Player / HoneyDrop interactions to make sure the score increases by 10 pts whenever a player picks up a honey drop. In addition, we used bump mapping to make the log have a "rough texture".

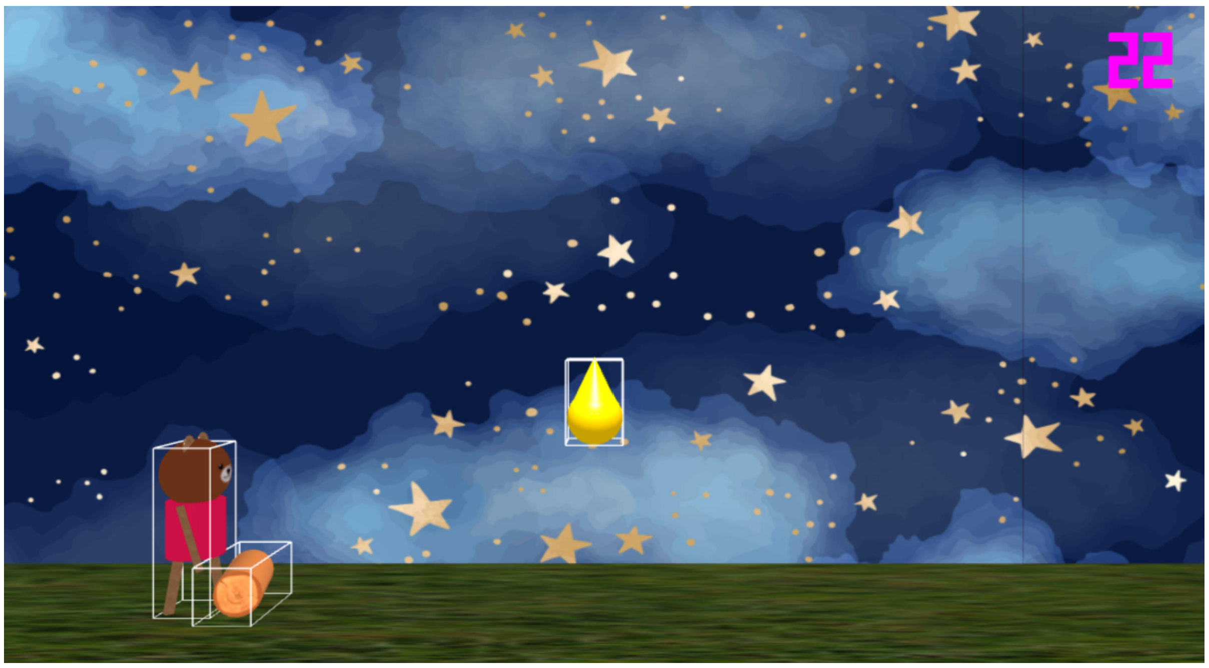

Figure 2. Collision Detection Implementation: Axis-aligned bounding boxes shown here with white edges. They serve as the basis that the game's collision detection system works off of.

The bear model was created using various polygons, such as a sphere for the head, capped cylinders for the body and limbs, and spheres scaled along one axis to be flat for the ears. For the fur, we wanted a seamless texture that could wrap around both flat and curved surfaces so using photoshop and a technique in which the edges are lined up and blended we were able to make the fur look seamless. In addition, for the head, we added a texture of the face.

The pond obstacle is two flattened spheres that overlap slightly to give it a unique shape, with a water image used for its texture. The log, in order to give its chopped ends a different look to the sides, is actually made up of two different cylinders in the same location to have the inside and outside look different but appear as one, and has bump mapping on the outside as well. The bird is a small model made up of multiple polygons to shape the head, body, and wings. The honey drop is made up of a sphere with a cone on top.

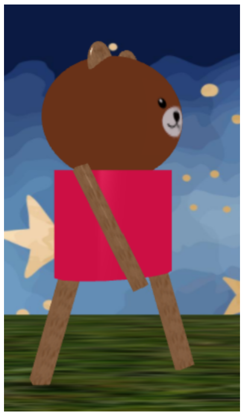
Figure 3.1 Standing Bear Model

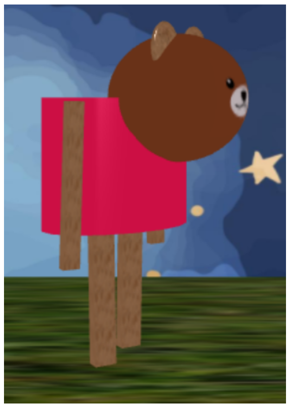
Figure 3.2 Ducking Bear Model

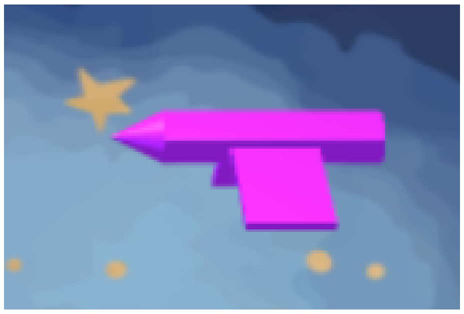
Figure 3.3 Bird Model

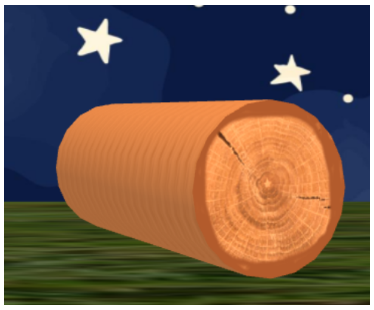
Figure 3.4 Log Model

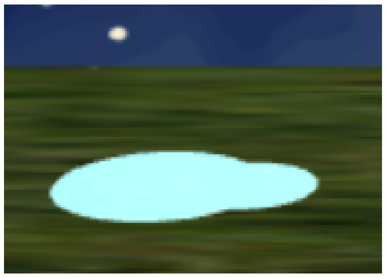
Figure 3.5 Pond Model

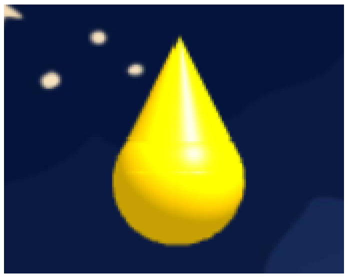
Figure 3.6 Honey Drop Model

Figure 3.7: Grass Texture

Figure 3. Models / Textures: Here are the models and textures used in Honey Collector