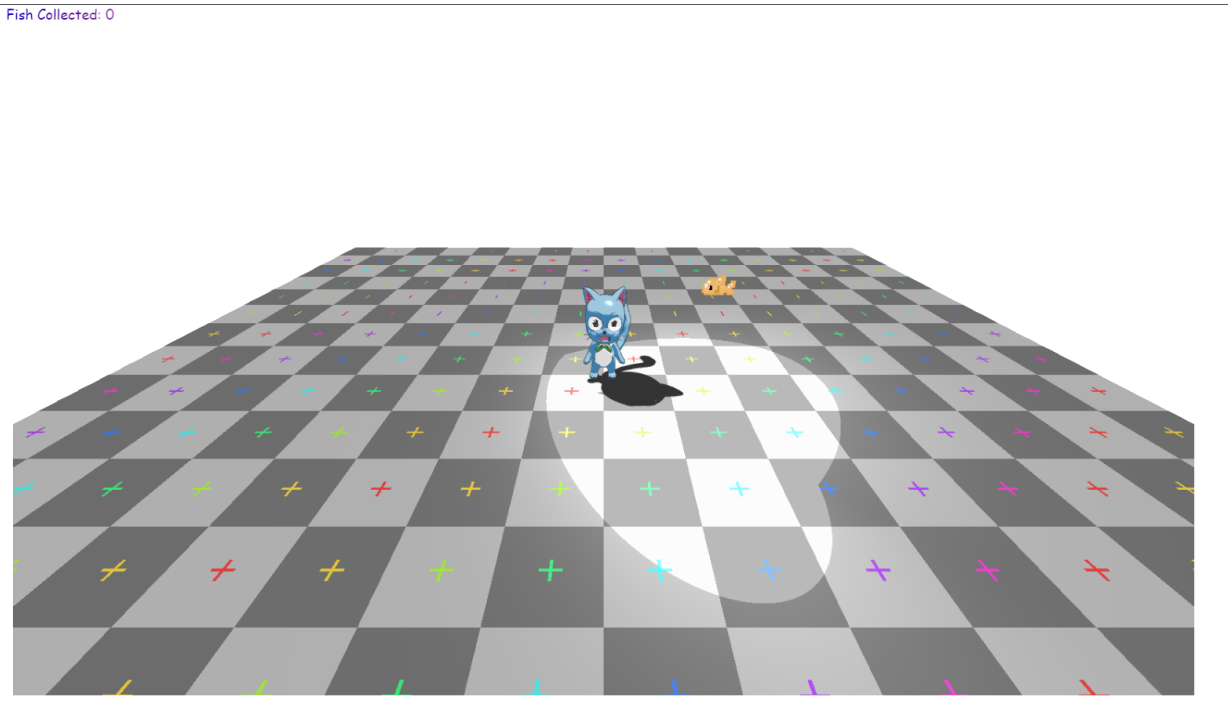
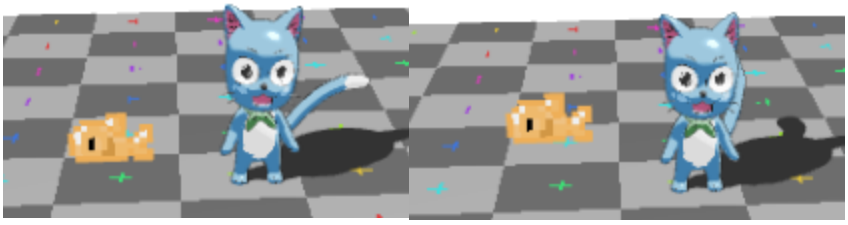
Jared Cascino
Randy Krauss
Nick Christen-Adams

CS 537 Final Report

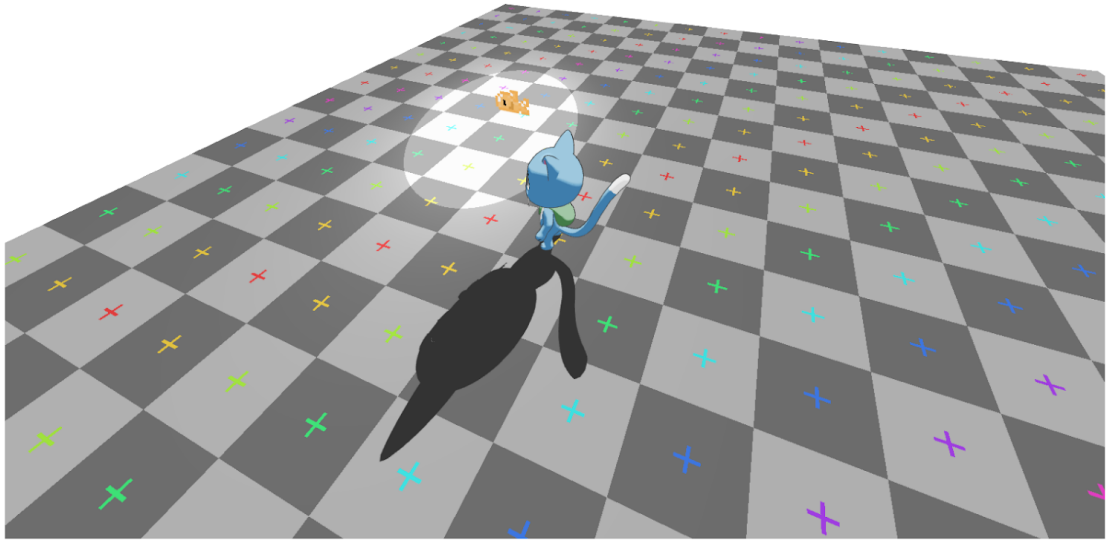
For our final project, we decided to make a mini game of Happy the cat from Fairy Tail running around and collecting fish. One of our group members, Jared, has a hobby of creating 3d models and animations, so he created the model / texture for Happy as well as his idle animation and running animation. Nick created the 8-bit fish that he runs around collecting. As far as the interactive components of the project goes, it's pretty simple in the fact that you can use WASD to control Happy and guide him towards the fish. You can see his starting position in the screenshot below, the fish will always start off in a different location and will respawn in a random location within the play area when it's collected, and the counter in the top left corner will increment along with it. Jared implemented the movement controls for Happy, and Nick implemented the logic for the fish positioning and collection.



This one is a bit hard to capture in screenshots, but the fish also has an idle hover animation. As you can see, the fish is higher in the screenshot on the right.



Here's a screenshot of an example game state after collecting a few fish:



The "Fish Collected" counter increments when happy gets close enough to a fish for it to be considered collected.
As you may have noticed, the camera isn't static, it follows Happy as he runs around the screen. Jared was the one who implemented this camera as well as built the mesh array for Happy's model. Randy built the texture array and the mesh information for the fish and floor. Randy was also responsible for the lighting information and shading of the objects.
The shader is based off of the phong shading model, but the values are clipped with a smoothstep function, and then blended with soft light blending.
Another technique implemented that is shown in various screenshots is shadow projection. We are projecting a copy of the "outline" mesh for Happy onto the floor plane.

Above are examples of some frames from the two animations. The first shows the run animation for Happy, and the other two show different frames of the idle animation. As we mentioned in the midway report, since skinned mesh rendering was out of scope for this project, we opted for baking the vertex position and normal data into 2d arrays for each animation. We are keeping track of the frame time and using constant frame rates to ensure the frames are being displayed at the correct time. When a new frame of animation needs to be rendered, the render function for Happy simply passes a row of the position and normal data in their respective 2d arrays.