

Assignment 4 Report

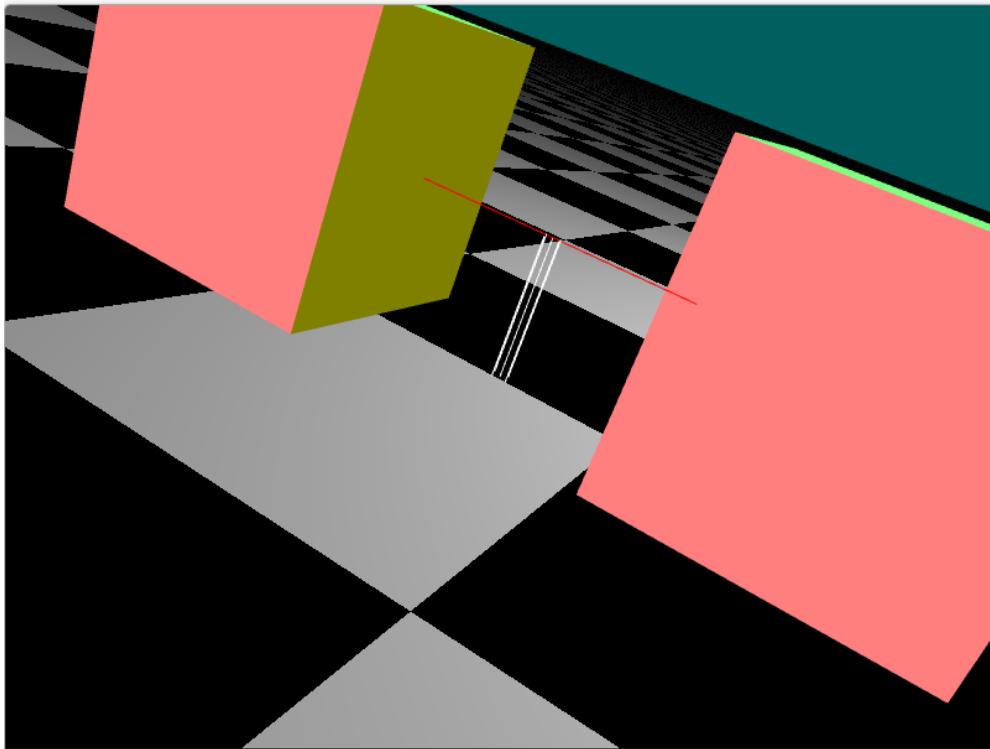
Ethan Lao and Daniel Qu

Bone Picking

Using unproject, we can intersect a ray into the geometry.

Then, we will check if the ray intersects the bone represented by its cylinder. First, we find the closest point between the cylinder line and the ray; if that distance is less than `kCylinderRadius` and the cylinder is in front of the ray, we can continue. Then we also check to see if the intersection point is between the ends of the cylinder. The intersection code will be handled in `Scene.ts` with a function called `intersect`, where `position` and `direction` represent the ray. We use this to then save and highlight the bone in the `drag` function in `Gui.ts`.

We create a cylinder for every bone translated off-screen. When a bone gets highlighted, we simply remove the translation for that bone and apply transformations in the orthogonal directions to the bone direction for each of the 4 lines representing the cylinder. In addition, we change the color of the original bone to match the color of the cylinder.



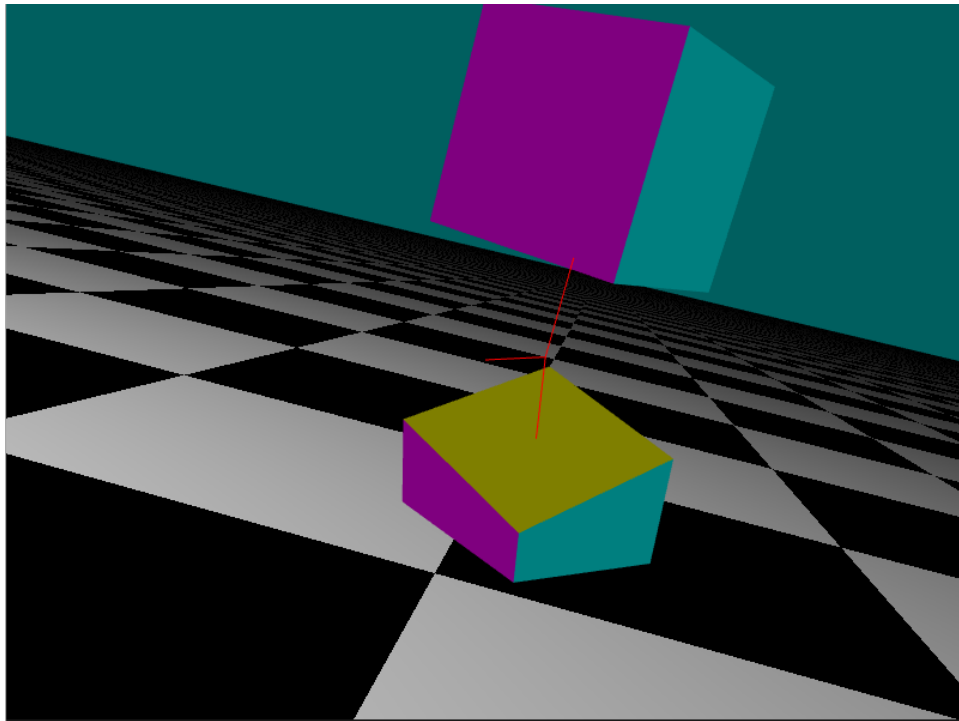
Skeleton Hierarchy

The bones were placed in a list, and the parent-child hierarchy was implemented by having bones store the indexes of their parent or children. In addition to the provided class variables, we also added local rotation and relative position attributes. These are helpers for updating the actual position and rotation of the bone. The update process is described in Bone Manipulation below.

Bone Manipulation

Updating the skeleton will be handled in the `drag()` function in `Gui.ts`. If a bone is highlighted or picked (as described in the bone picking process), we will rotate the bone instead of moving the camera. To get the direction/axis to rotate the bone on, we will take the cross product of the direction the mouse gets dragged (based on `dx` and `dy`) with the look direction. Since it's possible for the mouse to move off the bone while dragging, we will wait to update whether a bone is highlighted until the user stops holding down the left mouse button.

To update on a drag, we first update the local rotation of the highlighted bone. Then, we update the rotations and positions of all other bones. Rotations get multiplied down through the children, while positions get multiplied down along with the bone's relative position.



Implementation Challenges

Cylinder intersection works almost perfectly but can be inaccurate when the bone's parent gets rotated. We are not sure what is causing the problem as we apply the rotation to the bone position and endpoint before intersecting it with the ray.

Drawing a cylinder over a highlighted bone was also challenging; our solution added 4 times as many bones to any bone structure. For the larger bone structures, extra lines sometimes get added and we have not identified the root cause (but may be because of using too many bones).

Texture mapping was not completed.