

# CAP 4720 Final Project

## 3D Slice Puzzle - Summary

This project demonstrates the use of the stencil buffer and clipping planes to create a simple interactive 3D puzzle.

After the intro animation, the object will be sliced into several pieces and shuffled randomly. The user can *click and drag* on the individual pieces to rearrange their vertical order to restore the shape.

Once the shape is reconstructed successfully, it will spin around and shuffle again.

## Project Details

The way this project achieves its effect is by creating several duplicates of a loaded-in model (using OBJLoader). Each one of these duplicates is offset by a different amount, and then clipped by two clipping planes, one above and one below, to create the illusion that the object has been sliced.

However, since the 3D models are hollow, the top surfaces of the clipped models must be "filled in" to create the illusion of solid geometry. Achieving this effect was difficult to figure out, but a solution was found using the stencil buffer. A duplicate of each slice's geometry is first written to the stencil buffer, and then a plane coplanar with the top clipping plane is rendered immediately after. The plane is only drawn at the intersection, thus, creating the missing surface. The buffer is then cleared and repeated for each slice.

The interactivity is added by raycasting from the camera's viewing frustum toward the mouse pointer to find the slice the user clicks on. However, raycasts that are intersecting with the invisible, clipped portion of each slice's geometry must be identified and discarded.

An invisible plane is aligned with the puzzle's geometry where the raycast hits to aid with translating the objects by dragging. However, because of the stylistic choice of an orthographic camera, a secondary raycast is needed from the center of the screen at

the pointer's camera y position to prevent extra vertical movements as the pointer moves left or right.

The sliced objects, plane meshes and clipping planes are stored in a series of arrays. Their positions are swapped as the user rearranges them. The animation loop smoothly interpolates each object to its position defined in the ordering arrays.

Once the puzzle has been solved, the full geometries of each clipped slice will be at the same world position. Therefore, the puzzle can identify when it has been solved. It then rotates the model around again, and then reshuffles the puzzle.