Advanced Graphics

Lab 7 – Creating a version of Voxel Painter

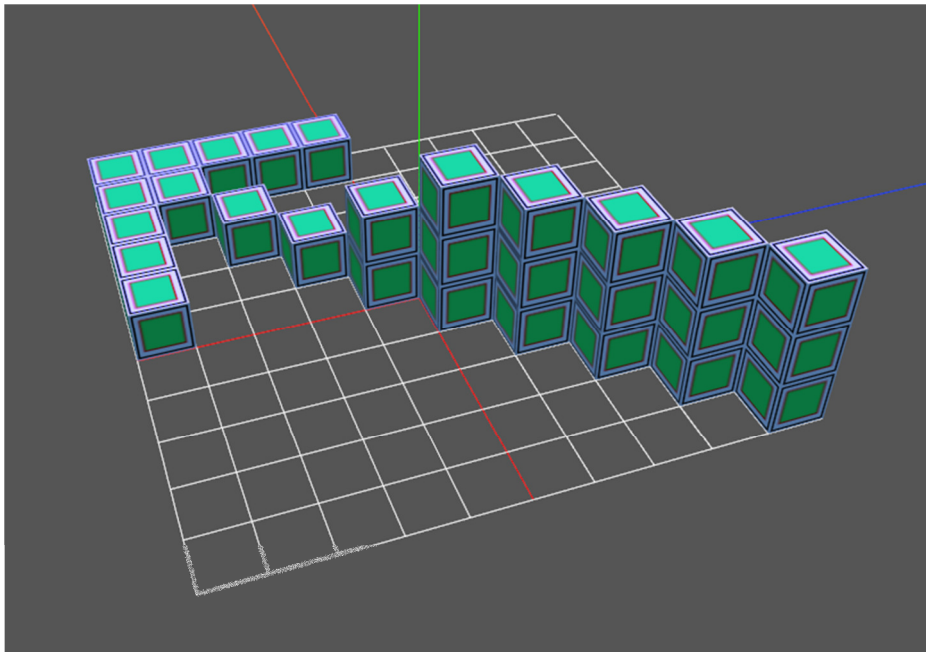
Maximum points: 10

Due: At the end of the lab to e-centennial (javascript file only). Your filename must be your first name. (except the two Winstons -> `winstonH.js` and `winstonK.js`)

NEW:
Please upload your zipped javascript file only. Failure to do so will result in a grade 0.

Objective for this lab:

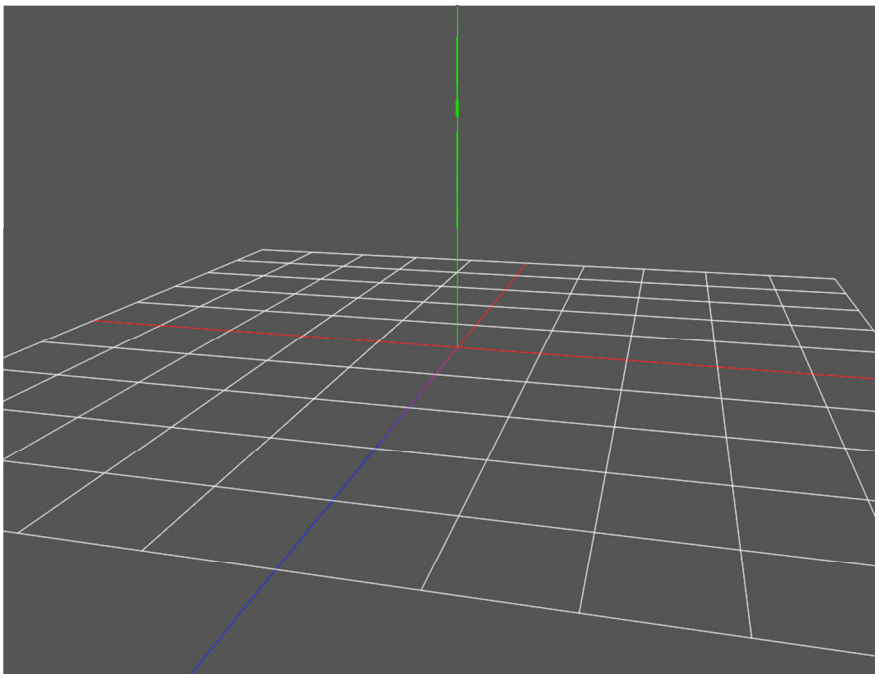
- To be hone you spatial reasoning skills.
- To recreate a voxel painter.
  https://threejs.org/examples/#webgl_interactive_voxelpainter.
  The above url is to guide you. Examine the code only if you are stuck.
- Do all the assigned problems on your own.



The completed application

Tasks:

If you are not comfortable with your Lab 5 then, there is a new base to start your application with. It contains an axis-helper, a plane, a cube, a directional and a hemisphere light. There is even a controller setup to toggle rotation of scene



1 Mark — Start with the above screen shot. In addition to the lights and the axis helper there is also a **`GridHelper`**.

Two marks for coding styling

2 Marks — **Active-Planes**: Fill the above grids with invisible mini-planes. The planes should be placed in an array. This array will be used to check for intersection with your raycaster.
You must not use magic-numbers in your code. All values should be calculated.

1 Mark — **Mouse down handler**: Create and wire-up a mouse down handler to do the following:

1 Mark — Detect which of the mini-planes was clicked.

2 Marks — Create a cube and place it at the same position as the plane. For the cube texture map, you must use an image file with path name `"../assets/textures/square.png"`

1 Mark — Move the plane to the top of the previously add cube.

If you are able to combine the last two steps in a single function of the mini-plane object, this will contribute heavily towards your code style marks.