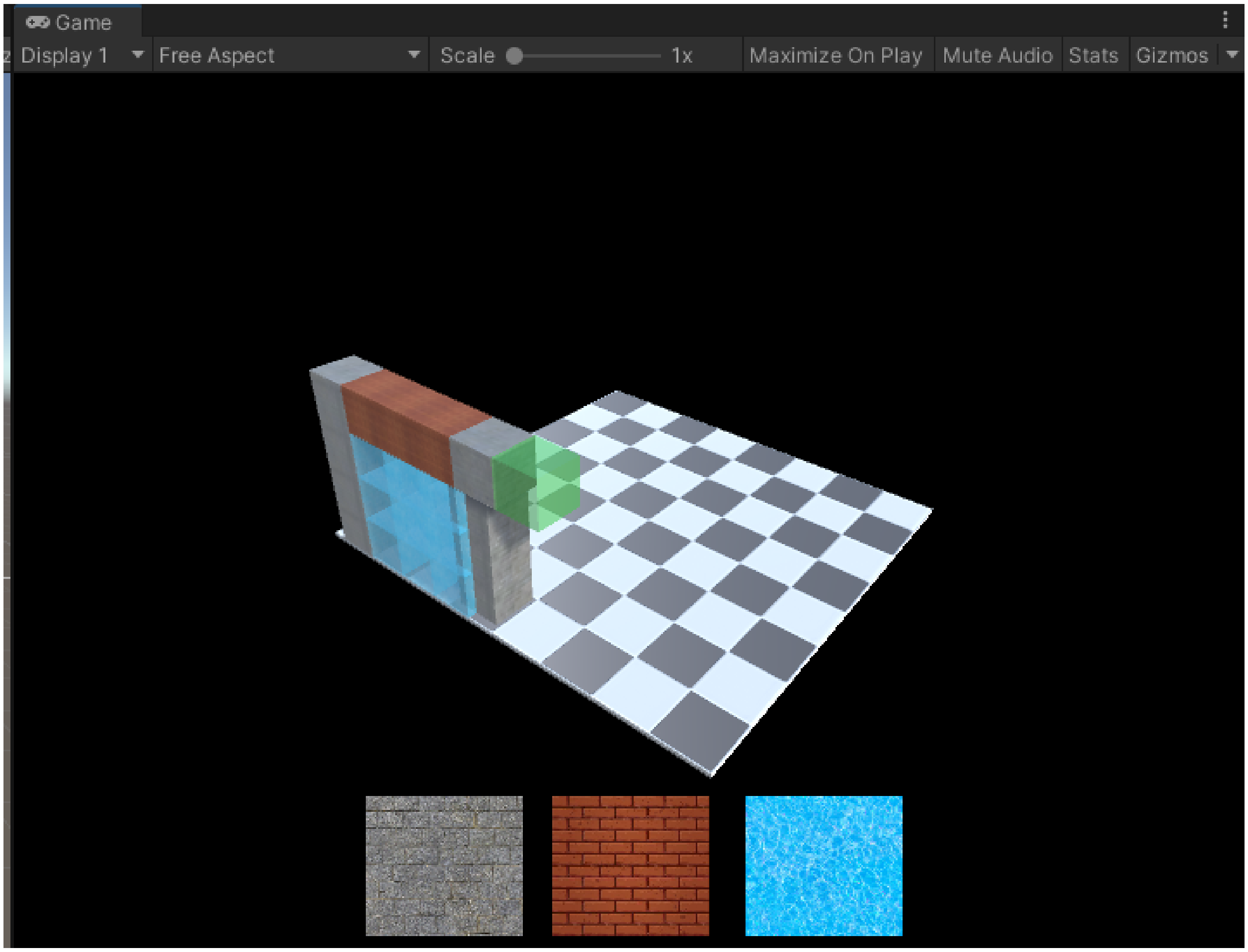
Homework 2.0 - Building Blocks:

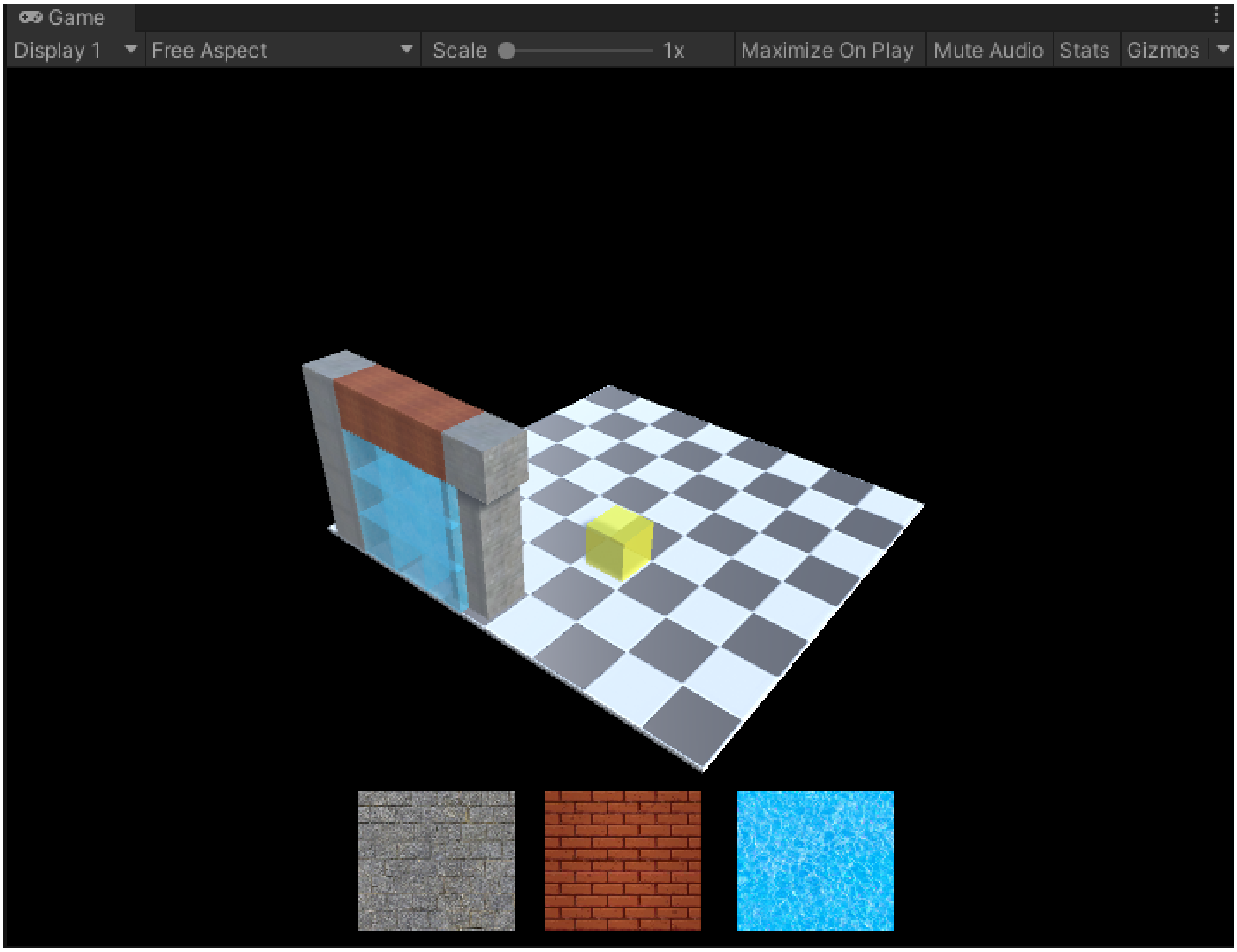
Tristin Greenstein
9/18/2021

Problem:
Creating a chess board like plane where the user is to place 1 unit cubes. A “indicator cube” shall appear where the mouse is to show where the cube will be placed when mouse click.The indicator cube should be able to snap to objects already on the board as well as turn green when snapped to that object.

Objective:
To increase your understanding of GameObjects, World Space, Transforms, Camera Controls, Mouse and Keyboard Inputs.

This homework assignment was quite interesting. I had randomly gotten the inspiration to start this on 9/17 and finished it that same day.
Using the baseline code from the in class example, I was easily able to get to the point of figuring out the indicator cube. I knew I had two ways to approach this. One by creating a cube in the code that follows the mouse around, and Two creating the cube in unity and having the script attached to it.
I first tried creating the cube in the code to some interesting effects.It started making many cubes every frame. So I went back and tried a safer solution of using a existing cube. Since the baseline script could be attach to any object, I attached it to the new cube I created. Here I tried to copy the raycast code to make the cube follow the mouse. Instead it decided to keep flying the cube at the camera but hey at least it created a cube on that path when I clicked my mouse. Next I tried the hitInfo equals base code for making the cube follow the mouse. It “worked” but the accuracy was way out of reach of where the indicator was for it to be submittable. There also was a problem of depth, it would not place blocks past a certain depth. I tried then copying all the if statements from below but removing the “cube.” part so it would transform the cube it was attached to/
During my experimenting, aka throw at the wall and see what sticks, I was able to find out how to reduce the thousand if statements I have in one line “ cube.transform.position = hitInfo.transform.position + hitInfo.normal;” This takes the baseline hitInfo value and the indicator of being hit and adds those coordinates together to find the position. Using this as well as integrating the tag equals base and tag equals mycube code, I was finally able to get the tracking done correctly. Since I detected if it hits a cube, it was easy for the snapping to work. I then added something we did in class but was not required on the homework, material switching. Using public Material and a void setMaterial function, it took little time to setup this material switching function.


Game
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
Gizmos


Game
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
Gizmos