Making Color Matter

A Reflective Practice

by Hien Quy Tran

Date: 12/12/2014
Student Number: 543800

Lecture: Game Design
3rd Semester
Project: Development of a Modular Game in Unreal Engine 4

Implementing a Color Detector

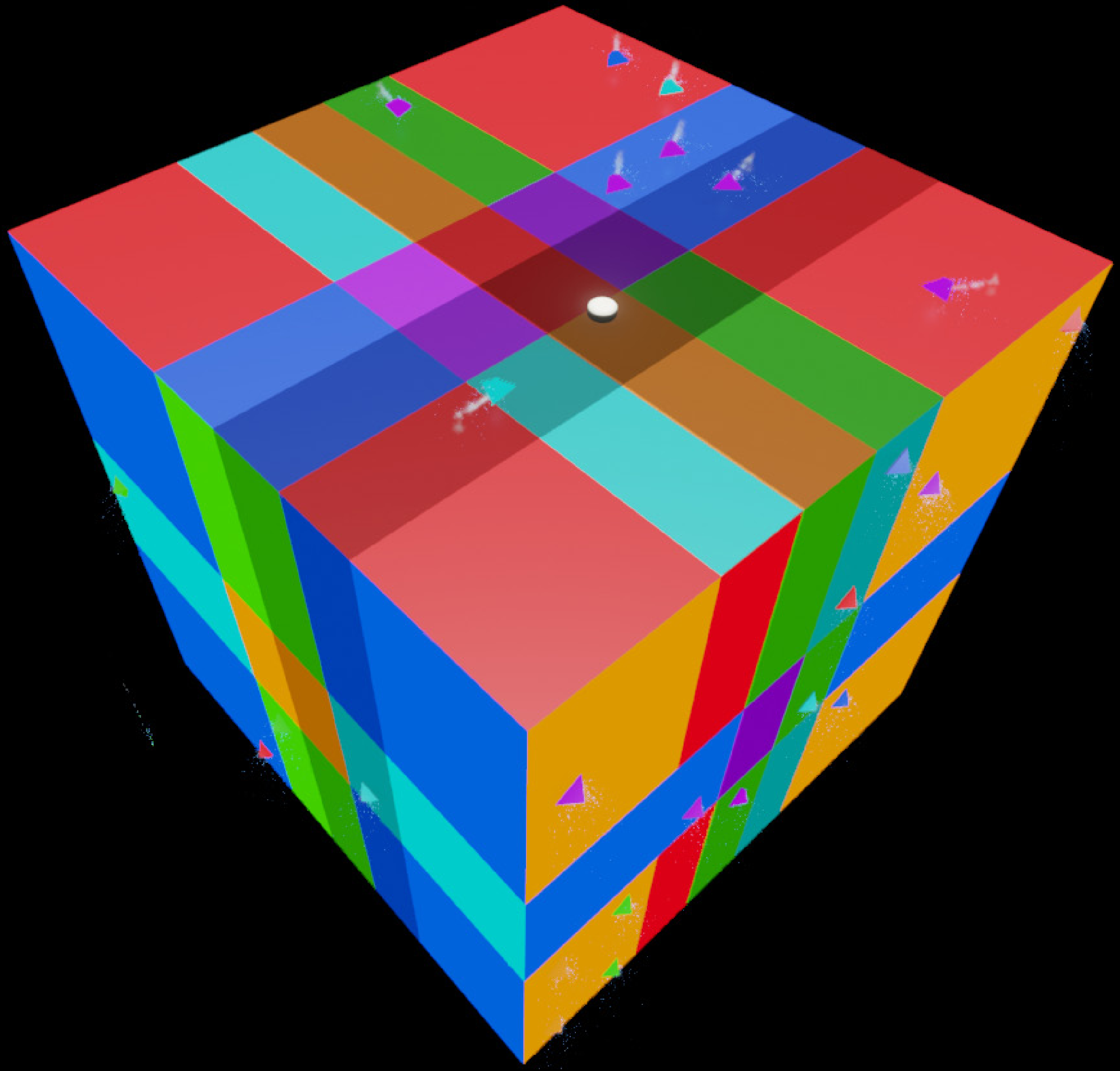
Approximate Working Hours: 12

Having all those different colored fields, the next step would be to implement different functionalities to those. In order to do so, the application needs to be able to read any cubes surface and distinguish it from others. For now, the color detector is being attached to the playable character and reads the color of the surface the playable character is standing on.

Using a Line Trace in order to find out on which cube the Player is standing on is fairly easy. The main problem was, that each cube has six faces with ultimately each of the faces having its own material resulting in six possible different functions for each cube depending on which side of the cube the playable character is standing on. Since the Hit Results are not providing any information about the surfaces visual material, I had to find a way to work around it.

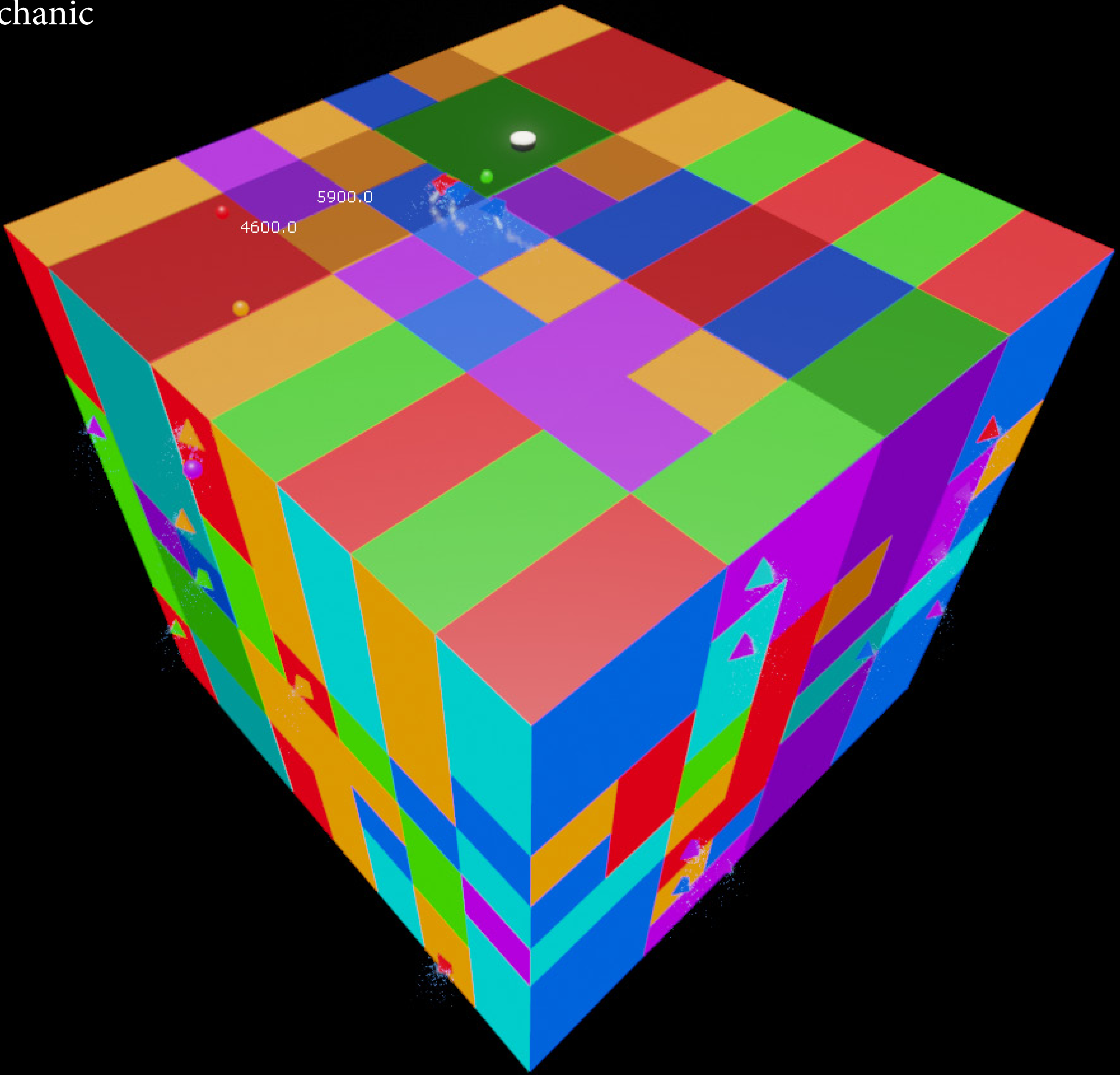
The solution was to link colors to the rotation of a cube. eg. if red is needed, no matter on which cube, it would be always applied onto the bottom face of a cube and then being rotated, so that the red surface faces out.

By reading the cubes rotation, the color detector will figure out, on which face of the cube it is standing on. Which means in our example, it would know, that the bottom face of the cube is facing up and since the bottom face is always red, also knows that it is red.



Adding A Color Field Based Mechanic

Approximate Working Hours: 4



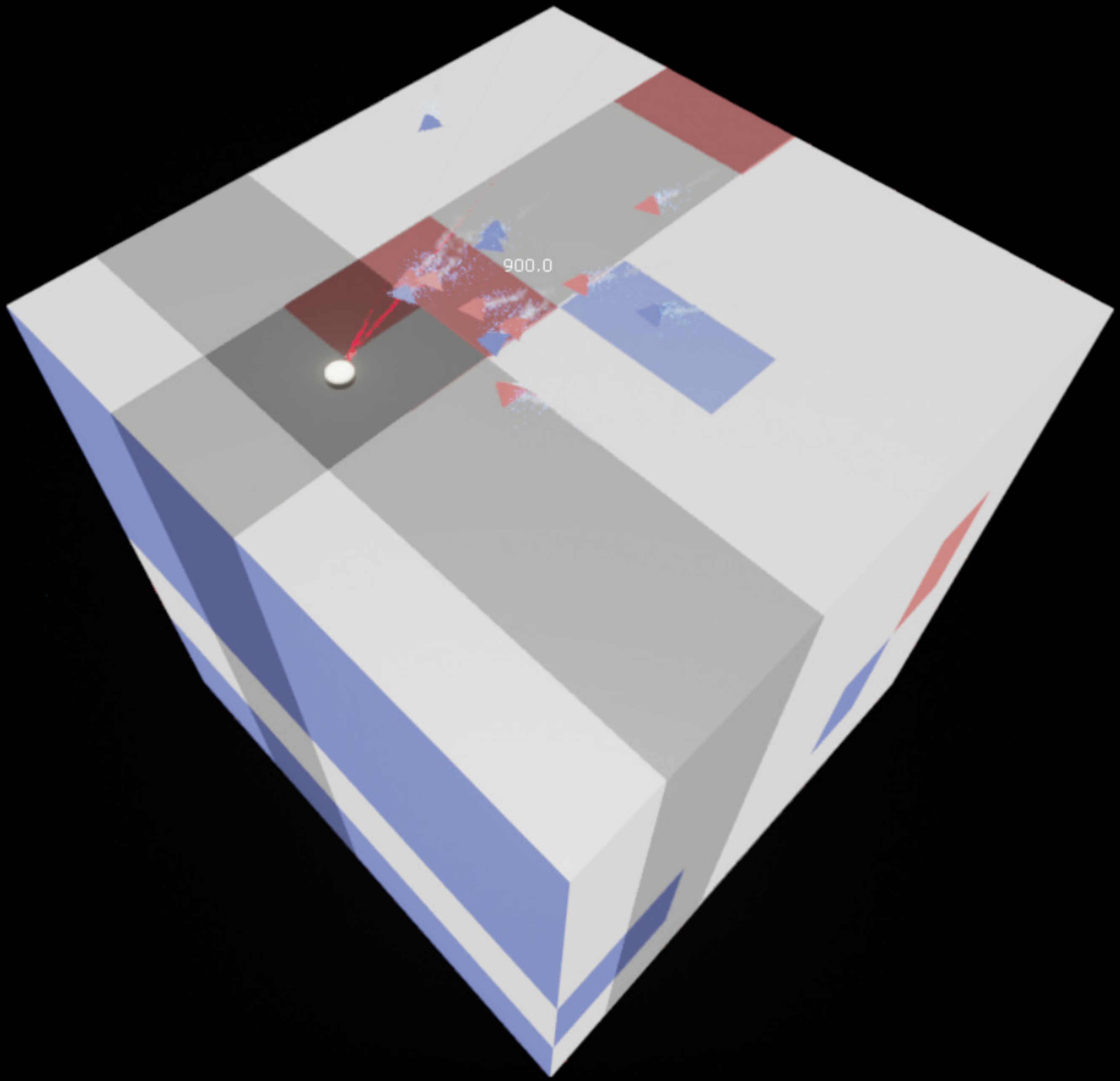
In order to test the color detection in combination with a simple game mechanic, I added a Twin-Stick-Shooter mechanic to the playable character. The player can shoot colored projectiles into the direction the right stick is pulled to while he is moving his character by using the left stick. The twist is, that the player is shooting projectiles in the same color as the field he is standing on and can only eliminate those enemies, which are same colored as the projectile.

In order to do so, I spawned projectiles and instantly applied the same material to them as of the surface the playable character is standing on. Furthermore the projectile contains a variable of the type integer. This variable is set to a certain number from 0 - 5 with each of them representing a color.

e.g.: A projectile turned to red also gets the integer, which stands for red. By reading the integer the enemies can identify the color of the projectile. Only when the integer of the projectile matches the enemies color, both actors get destroyed On Event Hit.

Making Color Placeable

Approximate Working Hours: 4



In favor of Level Design; The next step was to make it easy to manually apply colors to the fields, without having to change the material and rotate the cube by hand everytime.

For that reason, I created a Blueprint, which shoots a Line Trace towards the Rubik's Cube and takes the first cube it is hitting.

It will then color the right face into the according color and rotate this face out, while adding other information to the cubes variables, like e.g. "Left Ammo".

To use this Blueprint it is enough to drag and drop it onto the surface of the Rubik's Cube and the cube on that spot will be rotated, while the color will be set accordingly.

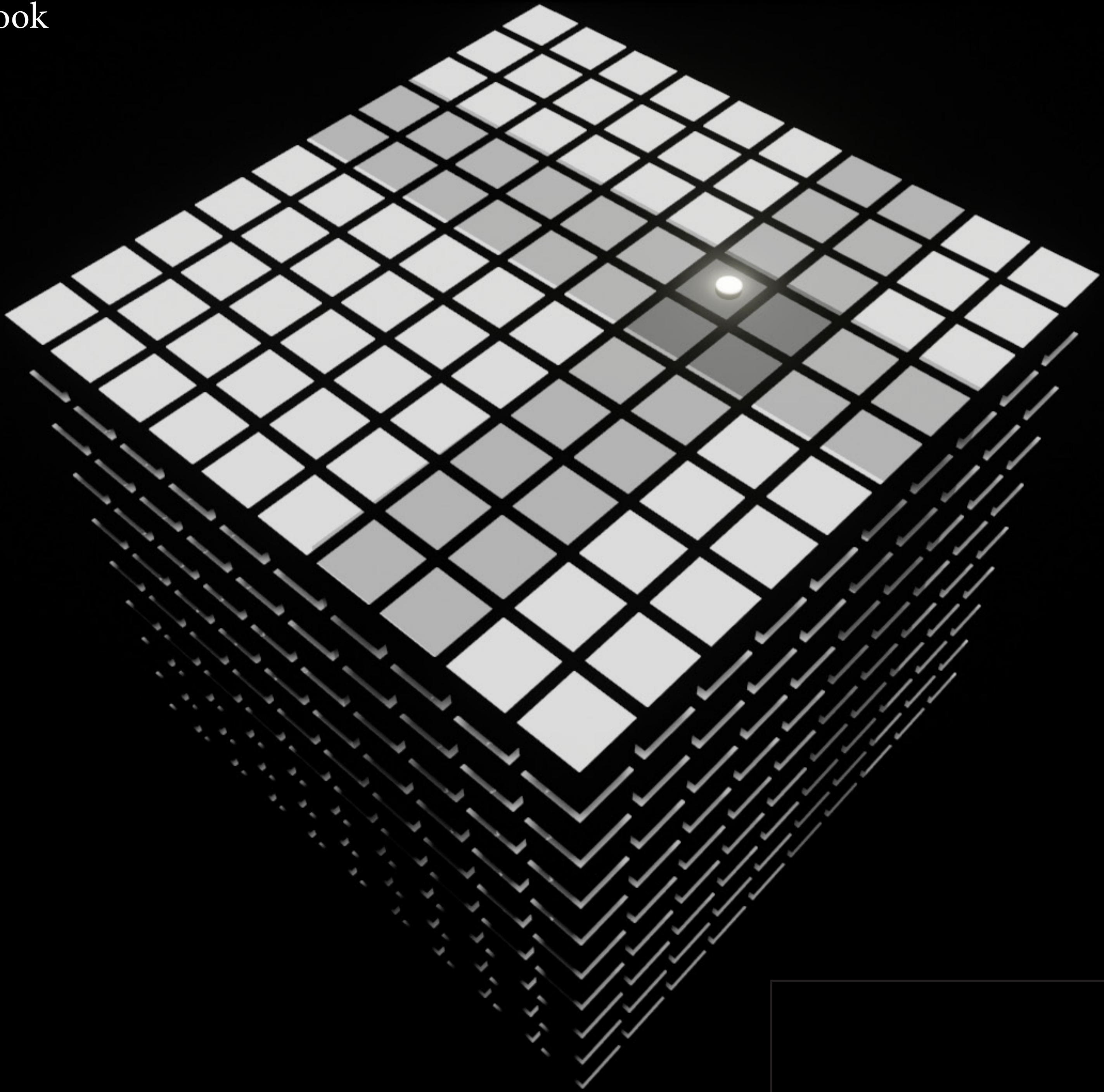
I have created three of them, each representing another color: blue, red and yellow.

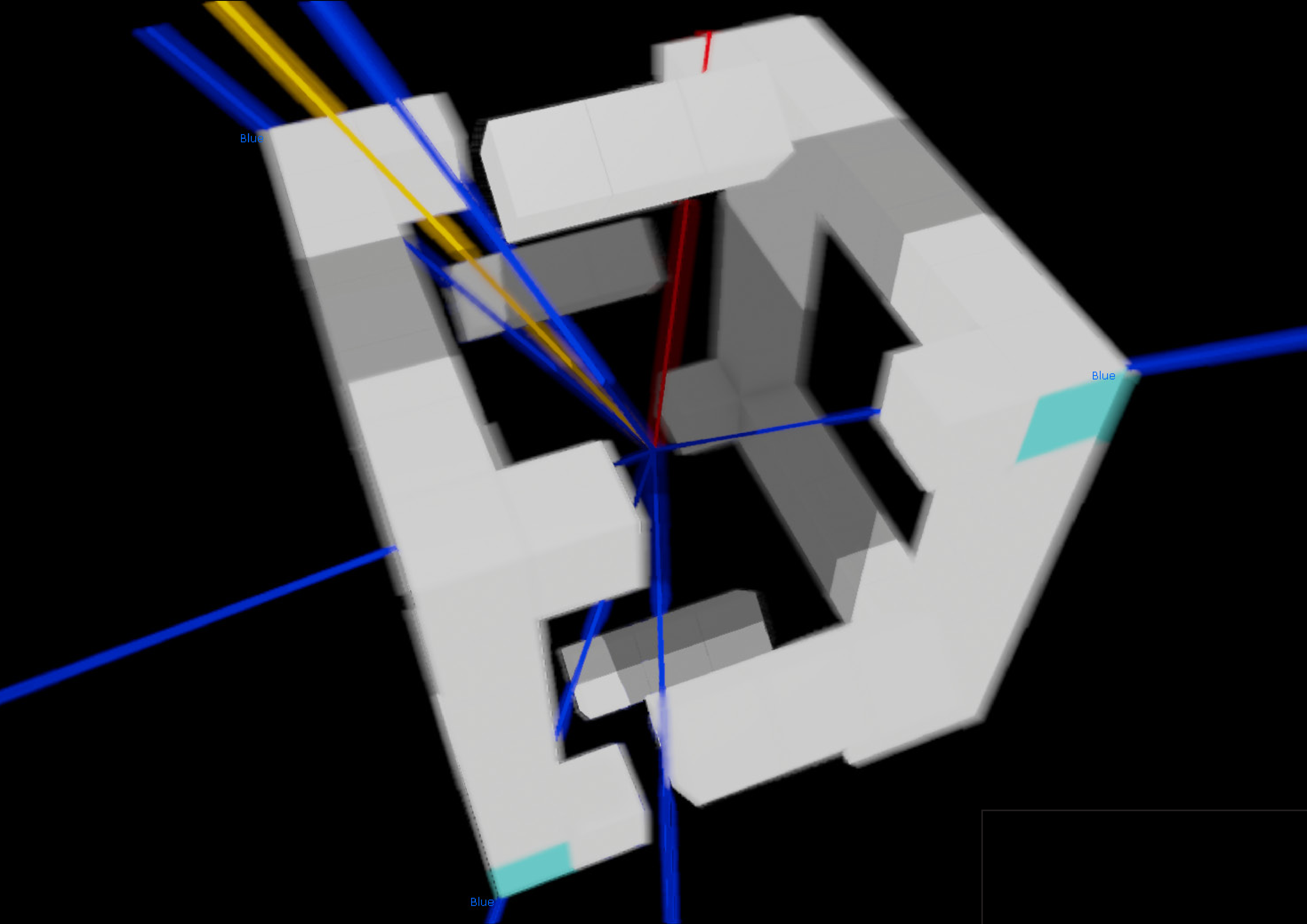
Experimenting with the Cubes Look

Approximate Working Hours: 2

Meanwhile, I have experimented with the cubes materials and setups.

Here are two examples.




Blue
Blue
Blue