Sean Cartman 3157705
Susie Tay 5717090

Introduction

Our plan is to use brute force in java to find every possible orient of cubes.

For clarity big cube is the 2 by 2 by 2 cube and sub-cube is one of the coloured cubes that make up a big cube.

Our answer was 23.

Definitions and Representations

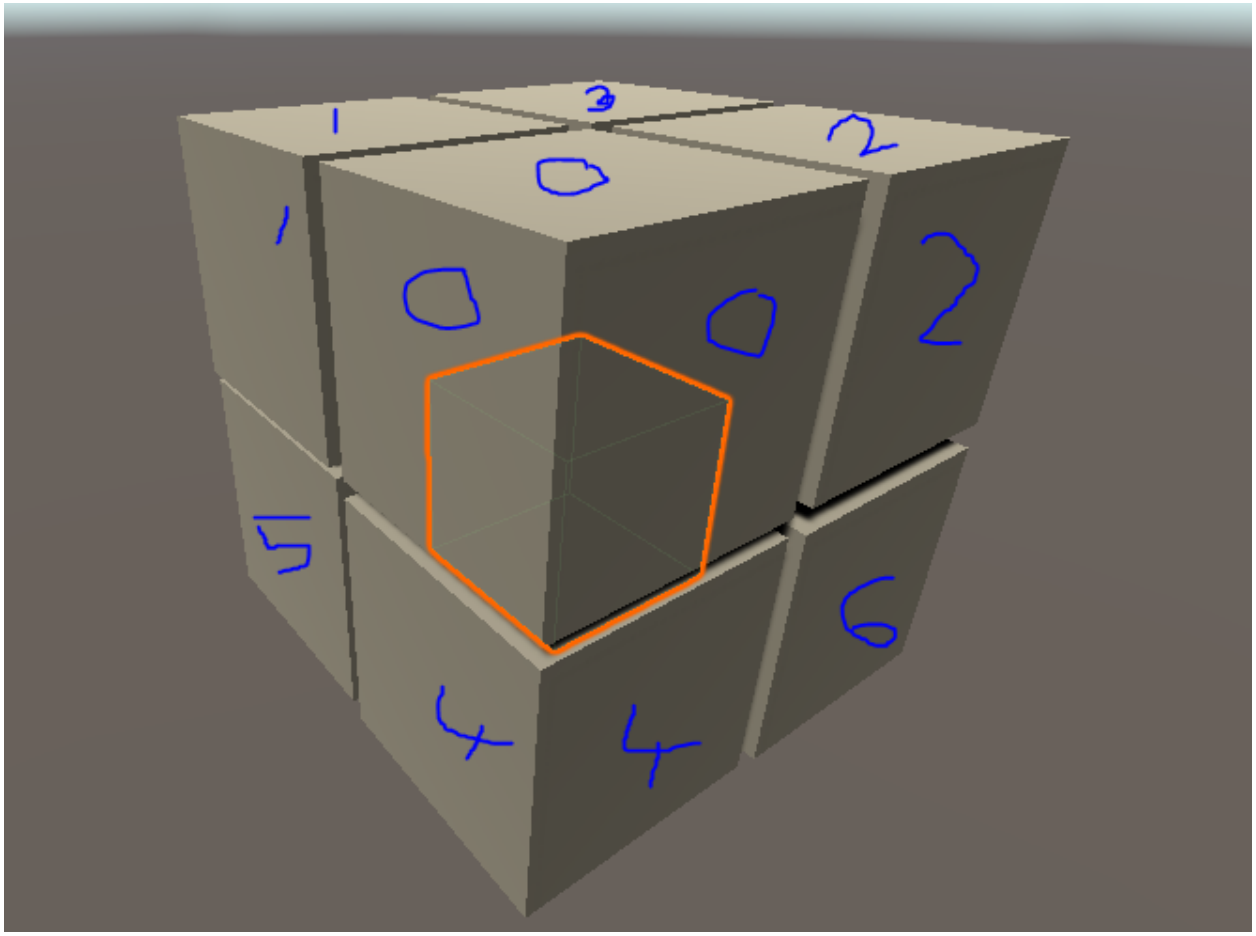
Here we will define a few Items for use in solving the problem

Big cube

We will represent big cubes as strings where each index is a sub-cube and a char represents the colour.

Index	Sub-cube location
0	top left front
1	top left back
2	top right front
3	top right back
4	bottom left front
5	bottom left back
6	bottom right front
7	bottom right back

Sean Cartman 3157705
Susie Tay 5717090



7 is the highlighted cube at the back

Sean Cartman 3157705
Susie Tay 5717090

Sets

We will define 3 sets of strings to store cubes in. We used sets so repeated cubes are not counted twice.

Name	Contains	Populated yet
orientationsCubes	all possible orientations for a cube	no
allCubes	every possible combination for a cube, where orientations are not considered repeats.	Yes as it is trivial to populate
uniqueCubes	every possible combination for a cube, where orientations are considered repeats	no

Rotations

We will need a rotation method for every axis the cube can rotate on. We will define these methods as xRotation(string cube), yRotation(string cube) and zRotation(string cube), see the appendix for details.

Solution

Base Idea

We will populate orientationsCubes for every String s in allCubes. If uniqueCubes and orientationsCubes for this s share no similar cubes we will add s to uniqueCubes. With this done the length of unique will be the number of unique cubes.

Max number of orientations for a cube

Before we start to populate uniqueCubes it will be useful for testing to find the number of orientations of a cube there are. On a cube, we have 6 faces. By rotating we on the x and y-axis we can bring any of these 6 faces to the top. On the top face of a cube, 4 possible orientations can be reached by rotating the z-axis. If we have 6 faces and 4 possible orientations for each face our max number of orientations will be 24.

Populating orientationsCubes

Sean Cartman 3157705
Susie Tay 5717090

We can use the method doAllRotations(s, depth)(see appendix for details) to populate orientationsCubes. It explores every possible combination of rotations, creating every unique orientation for the original cube.

How do we know every combination has been explored? If orientationsCubes length is 24 after generating the most complicated cubes we know every orientation was generated. After a little ad hoc experimentation, we found a maximum of 5 rotations would generate orientationsCubes to the desired length

Sean Cartman 3157705
Susie Tay 5717090

Appendix

```
doAllRotations(s, depth)
        If depth == {value that makes our cubeRotations size == 24 for s =  “yyybybyb”} = 4
                Return
        A = x(s)
        B = y(x)
        C = z(s)
        add A,B and C to cubeRotations
        doAllRotations(A, depth+1)
        doAllRotations(B, depth+1)
        doAllRotations(C, depth+1)

zRotation(string cube)
       String newCube =””
       newCube[0] = cube[1]
       newCube[1] = cube[5]
       newCube[2] = cube[3]
       newCube[3] = cube[7]
       newCube[4] = cube[0]
       newCube[5] = cube[4]
       newCube[6] = cube[2]
       newCube[7] = cube[6]
       return newCube;
yRotation(string cube)
        newCube =””
        newCube[0] = cube[2]
        newCube[1] = cube[3]
        newCube[2] = cube[6]
        newCube[3] = cube[7]
        newCube[4] = cube[0]
        newCube[5] = cube[1]
        newCube[6] = cube[4]
        newCube[7] = cube[5]
        return out
xRotation(string cube)
        newCube=””
        newCube[0] = cube[2]
        newCube[1] = cube[0]
        newCube[2] = cube[3]
        newCube[3] = cube[1]
```

Sean Cartman 3157705
Susie Tay 5717090

```
newCube[4] = cube[6]
newCube[5] = cube[4]
newCube[6] = cube[7]
newCube[7] = cube[5]
return out;
```

Sean Cartman 3157705
Susie Tay 5717090