**Happy Cube Solver**

Program is implemented using Java 1.8 and Eclipse IDE. Project can be run using IDE by running HappyCubeSolver.java class or using Maven clean install and then executing the following in the command line:

java –jar HappyCube-0.0.1-SNAPSHOT.jar

One of the solutions in unfolded form of the cube is printed on the console after program runs.

**Algorithm**

Happy Cube consists of six faces namely: Main, Right, Upper, Left, Bottom, Other face.

Each face consists of four sides. Each side consists of five blocks. Blocks can either be filled (“o”) or empty (“ “).

There are total of eight orientations of a given face namely: Base, Right, Upsidedown, Left, Flipped, Flipped Right, Flipped Upsidedown and Flipped Left.

Program starts with selecting one of the faces, the most symmetric one, as main face as in Figure1. It marks the faces used as it proceeds. Then it searches for each orientation of not used faces as the right face. It continues with finding suitable upper, left and bottom faces respectively. Lastly it checks the remaining face namely “other face” for consistency with other neighbor faces. For checking consistency of faces, the algorithm checks for neighbor sides of the faces (at most one of the two neighbor blocks can be “filled”) and corners (at most one of the three neighbor faces joining in one corner must be “filled”).

UPPER FACE

LEFT FACE MAIN FACE RIGHT FACE

BOTTOM FACE

OTHER FACE

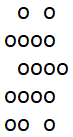
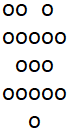
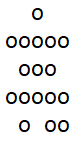
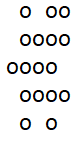
**Design and Implementation**

**HappyCubeSolver.java:** Program starts execution with main method. Firstly the most symmetric face is selected as main face and findFace method of the following faces are called consecutively. If a happy cube can be constructed with given faces, the solution is printed on the console in unfolded format.

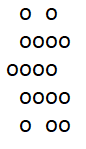
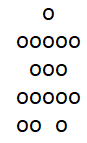
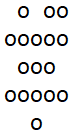
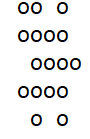
**HappyCube.java:** Class representing happy cube. It has an array containing six faces and each face index. print() method prints the happy cube in unfolded format as in Figure 1.

**EnumOrientation.java**: Contains eight different orientations of the face.

Base: Right: Left: Upsidedown:

Flipped: Flipped Right: Flipped Left: Flipped Upsidedown:

**Face.java:** Class representing one face of the cube. It keeps four sides of the face in a Side object array. isUsed variable is needed since each face can only be used once. orientation variable keeps the current orientation of the face, since each eight orientation of the face can be used while constructing the cube. Face constructor takes the string representation of the face as an argument, and construct the four Side objects as an array of integers, 1 representing a filled block, 0 representing an empty block.

Face construction from string representation:

" o o " 0 1 0 1 0 Side 0 : {0,1,0,1,0}

"oooo " 1 0 Side 1 : {0,0,1,0,0}

" oooo" 🡪 0 1 🡪 Side 2 : {0,1,0,1,1}

"oooo " 1 0 Side 3 : {1,1,0,1,0}

"oo o " 1 1 0 1 0

**Inherited Faces:** Each face is inherited from the base Face class since they have different side and corner checks. findFace() method in Face class checks for side and corner consistencies according to each different face rules and calls the findFace method of the next face in the following order: Main, right, upper, left, bottom, other.

**CubeReader.java:** Input is read from file “cube.txt” under resources folder. Each face of the cube must be given one under the other in the input file.