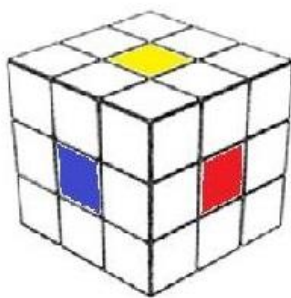


Cubul Rubik

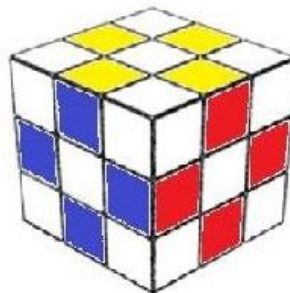
1) Obiective

- realizare interfata grafica pentru cubului rubik 3x3x3
- realizare algoritm de amestecare a cubului
- realizare algoritm de rezolvare automata a cubului
- realizare functii de rotatie in 90° a fetelor cubului

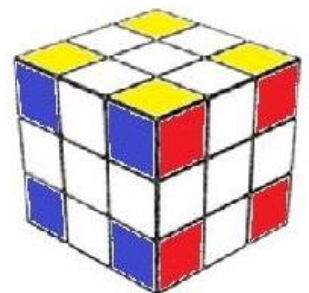
2) Functionalitati



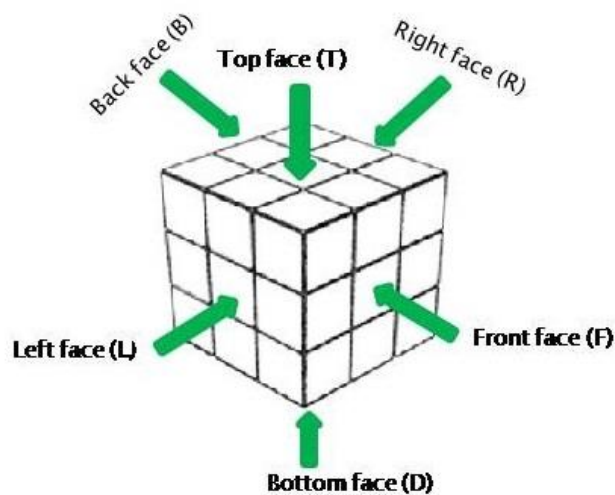
Centre pieces = 6



Edge pieces = 12



Corner pieces = 8



Vor exista 6 functii de rotatie care vor roti fetele cubului in sensul aceluia de ceasornic.

- **F** - Fata (**F**ront)
- **B** - Spate (**B**ack)
- **R** - Dreapta (**R**ight)
- **L** - Stanga (**L**eft)
- **U** - Superior (**U**p)
- **D** - Inferior (**D**own)

Algoritmii de amestecare si de rezolvare automata vor folosi aceste 6 functii.

3) Implementare

CameraMovement este clasa care realizeaza miscarea camerei. Camera se poate misca in orice directie folosind mouse-ul.

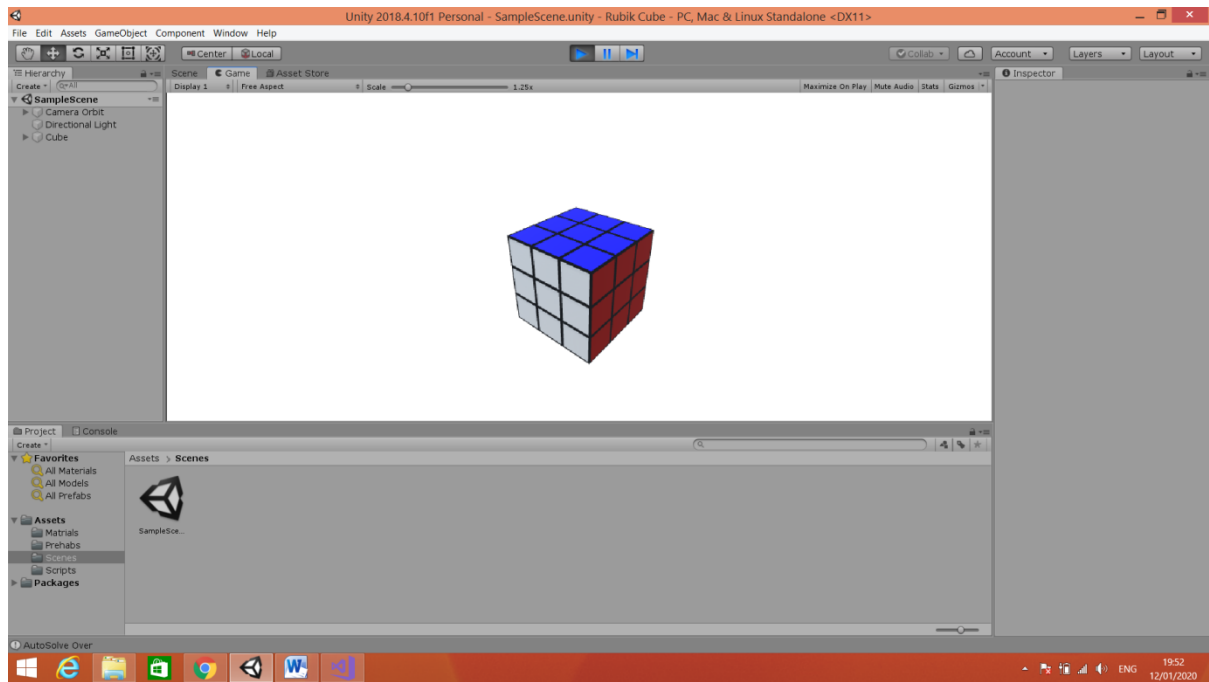
In clasa *CubePieceScr* avem metoda *SetColor* care seteaza culoarea pentru planurile fiecarui cube. Cubul Rubik este format din 27 de cuburi fiecare plan avand o culoare unica. La inceput fiecare plan are culoarea dezactivata. Metoda va activa culoarea pentru planurile vizibile ale cuburilor.

In clasa *CubeManager* avem metodele care realizeaza miscarile de rotatie in 90° ale Cubului Rubik, algoritmul de amestecare si algoritmul de rezolvare automata. Algoritmul de amestecare alege un numar random de rotatii si n random rotatii si apoi le aplica. Algoritmul de rezolvare automata va executa rotatiile in ordine inversa.

4) Tehnologii folosite

Interfata grafica a cubului va fi realizata in Unity.

Partea de code va fi realizata in Visual Studio in C++.



5) Testarea aplicatiei

<https://www.youtube.com/watch?v=TXAHRNvJuAQ&feature=youtu.be>