Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

Отчёт

по лабораторной работе №1

По дисциплине: Проектирование программного обеспечения в интеллектуальных системах

Выполнил: Пилат М.Д.., гр. 121702

Проверил: Никифоров С. А.

Минск, 2022

Цель: описать класс реализующий игру-головоломку “Кубик Рубика”. Класс должен реализовывать следующие возможности:

1. Случайное начальное размещение цветов;
2. Загрузка начального размещения цветов из файла;
3. Поворот грани кубика;
4. Проверка правильной расстановки цветных клеток;

Описание структуры классов

Класс Cube содержит массив из шести граней (типа Face). Класс Face в свою очередь содержит вектор векторов для горизонтальных полосок и вектор векторов для вертикальных полосок, также содержит map, в котором хранятся соседние грани данной грани.

Структура класса Cube:

```
class Cube
{
private:

    int cubeSize;
    Face faces[6];

    void changeOrientToLeft(int neighbor);
    void changeOrientToRight(int neighbor);
    void rotateY(int face, int line, string currentNaighbor);
    void rotateX(int face, int line, string currentNaighbor);

public:

    Cube(vector<vector<vector<vector<int>>>>&);     ///<\brief
    void print();   ///<\brief Print the scan of the cube in th
    void rotate(int face, int direction, int line);     ///<\br
    bool check();   ///<\brief Check if cube is built
};
```

У класса 1 конструктор, который принимает многомерный вектор с плитками кубика. Деструктор по умолчанию (сгенерирован автоматически, неявно), т.к. в классе не присутствует динамическое выделение памяти, а используются STL-контейнеры.

Краткое описание методов сгенерировано с помощью Doxygen и описано в комментариях Cube.h и Face.h.

Самым сложным методом для реализации был метод, осуществляющий вращение грани кубика (по оси X или Y). На картинке представлено вращение по оси Y.

```
void Cube::rotateY(int face, int line, string currentNaighbor)
{
    Face currentFace = faces[face];
    vector<int> helper;

    helper = faces[face].vertical[line];

    for (int i = 0; i < 3; i++)
    {
        faces[currentFace.index].vertical[line] = faces[currentFace.naighbor[currentNaighbor]].vertical[line];

        for (int j = 0; j < 3; j++)
        {
            faces[currentFace.index].horizontal[j][line] = faces[currentFace.naighbor[currentNaighbor]].vertical[line][j];
        }
        currentFace = faces[currentFace.naighbor[currentNaighbor]];
    }
    faces[currentFace.index].vertical[line] = helper;

    for (int j = 0; j < 3; j++)
    {
        faces[currentFace.index].horizontal[j][line] = helper[j];
    }
}
```

Метод принимает на вход номер грани (которую хотим вращать), номер полоски и соседнюю грань для данной (сосед определяется исходя из направления в котором собираемся вращать грань). Под полоской понимается набор из трёх плиток грани, которые расположены в одну линию (по вертикали или горизонтали).