BoardGame2D Game class Report

Base Class
BoardGame2D

```
virtual void playUser(string)
virtual void playUser() final;
virtual void playAuto() = 0;
virtual void playAutoAll() final;
virtual bool endGame( ) const = 0;
virtual bool isVAlidInput(string) const = 0;
virtual int boarScore();
virtual void initialize() = 0;
virtual void print() const = 0;
void start();
static void playVector(vector<BoardGame2D*>v);
ostream& operator<<
--------------------------
string GameType;
int score;
```

Derived PegSolitaire

Deriverd Eight Puzzle

Derived Klotski

```
EightPuzzle( );
EightPuzzle(string)
EightPuzzle(const EightPuzzle& Obj);
~EightPuzzle( );
void createBoard( );
bool isSolvable(string);
void setMove(Direction, row, column);
--------------------------
int**Board;
```

```
Klotski();
Klotski(string);
Klotski(const Klotski&)
Klotski& operator=();
~Klotski();
void setRectMove(KlotskiElements, Direction);
void setRedMove(KlotskiElements, Direction);
void setSquareMove(KlotskiElements, Direction);
bool isValidMove(KlotskiElements, Direction);
--------------------------
KlotskiElements**Board;
int auto_move = 1;
```

```
PegSolitaire();
PegSolitaire(string);
void setMove(int, char, Direction);
bool isValidMove()
bool isRandomMoveValid(int, int column, char);
bool isValidInput(string)
void findScore();
        Class Cell
                Cell();
                checkElement(GameElements)
                setElement();
--------------------------
vector<vector<Cell>>Board;
void CreateBoard();
```
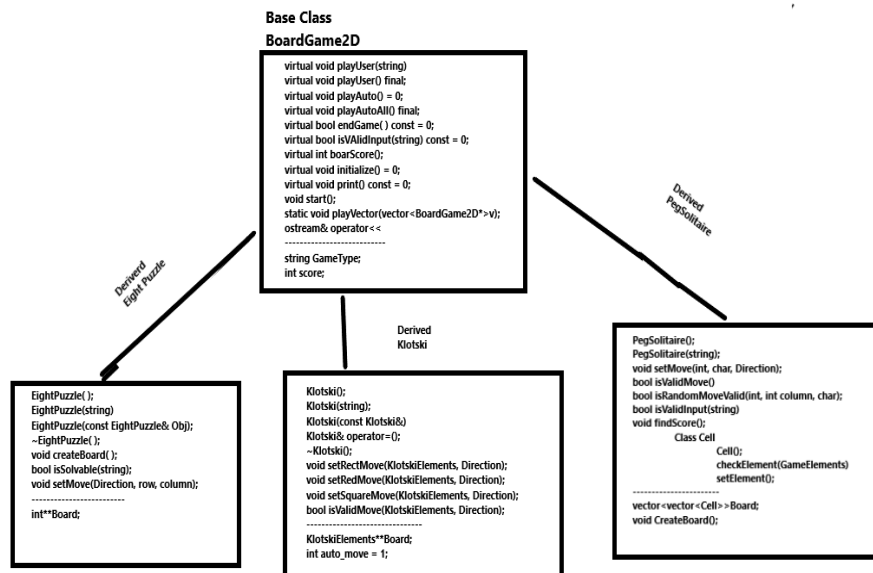
Base Class BoardGame2D:

This class includes two final functions ;

- void playUser( ) for Human Game
- void playAutoAll() final for Compter Game

Definitions of this classes are the same for every derived class, Derived classes calling these final functions , these function's definitions includes overriden pure virtual functions such as void playUser(string), void playAuto( ), bool endGame( ) etc. Each Derived class has its own definition for these pure virtual functions. Therfore, when a derived class calls a final function, inside of the definiton its calling its own functions, that saves us from overriding same definiton again and and again.

I wrote a void start function that checks game type(computer game/human game) then starts the game. To start any type of game, you have to call this function.

I called boardScore function inside of my final functions. However, I haven't found any way to calculate score for Klotski and EightPuzzle game.

For PegSolitaire game it Works as expected.

Finally static void playVector function plays every game inside of it in a row.

Class EightPuzzle

- Game Board is dynamically alloacated → int **Board
- This class includes Big three to prevent logical and sementical errors.
- EightPuzzle Constructors calling Base class Constructors to set gameType and they also creates gameBoard by calling override initialize function.
  Example
  EightPuzzle::EightPuzzle(string GameType)
         :BoardGame2D(GameType)
  { initializes();)
- Bool isSolvable function checks if random game is solvable, if it is not void createBoard function creates another board until isSolvable function returns true. I wrote this function because not every created board is solvable fort his game.
- For playAuto function it takes approximately five minutes to solve this game randomly

Class PegSolitaire

    - It includes an inner class called Cell, Peg Solitaire board is a double dimensional vector function in Cell class type.

    - void boardScore() function Works only for this function, Because score is computable only for PegSolitaire game among others

    - Basically, it works like other derived class, first it checks input by calling isValidInput function, then it check legality of move by calling isValidMove function. If both of them returns true, it sets the move and checks end game condition. This algorithm works same for every game !

- It doesn't include big three, because it utilizes double dimensional vector for board.

Class Klotski;

- Since game board is created dynamically, It includes big three.
- Game board is of Enum type.
- Game board consist of three type of elements, Squares, Rectangles and RED object. The purpose is sliding RED object to bottom of game board.
- Priting game board is my only flaw fort his function, it could've been better but without using any graphical library it is almost impossible to print game board properly.
- Unlike other derived classes, it includes three type of set functions, setSquareMove, setRedMove, setRectangleMove. I haven't found a common algorithm to implement every game element, therefore i handled them seperately
- Also Auto function works different for this class, this board has only one solution so that I solved it manually. This board is different than the one on the wikipedia page. It was not my choice to choose this board in the first place . I didn't see the board on wikipedia page, when I searched for the game , it was first board I saw. This is my second flaw . Other than that everything works as expected.

Mehmet Mete Şamlıoğlu 200104004093