CSE 241 - HOMEWORK 5 - REPORT

EMİRCAN DEMİREL - 1901042674

`class BoardGame2D`

`virtual void playUser(string move) = 0;`

One of the pure virtual functions of BoardGame2D class. It takes string as a parameter then play the game according to the string.

`virtual void playUser() final;`

virtual final function of BoardGame2D class.

- Board initialized according to the derived class
- Takes move string, then passes to stringValidity(move). If input format is valid, move string passes to playUser(string move) function.
- Call print() function from derived class and print the board to the top left corner of console.
- If endGame() function returns false, repeat this steps.

`virtual void playAuto() = 0;`

One of the pure virtual functions of BoardGame2D class. Randomly create move parameters -by computer- and play game for one step.

`virtual void playAutoAll() final;`

virtual final function of BoardGame2D class.

- Call initialize() function from derived class. After that, Board initialized according to the derived class.
- Call playAuto() function from derived class and play one step.
- Call print() function from derived class and print the board to the top left corner of console.
- If endGame() function return false, repeat this steps.

`virtual bool endGame() const = 0;`

pure virtual function of BoardGame2D class.

`virtual bool stringValidity(string move) const = 0;`

pure virtual function of BoardGame2D class.

`virtual int boardScore() const = 0;`

pure virtual function of BoardGame2D class.

`virtual void initialize() = 0;`

pure virtual function of BoardGame2D class.

`virtual void print() const = 0;`

pure virtual function of BoardGame2D class.

`class Klotski : public BoardGame2D`

Inherited functions

`void playUser(string move);`

this function take move string as a parameter and play game according to this parameter for one step.

- Separate move string to different variables and pass these variables to isMoveValid function.
- If it returns true; call swapFunc function with the help of if-else statement and update the board.
- Else it returns false; passed without making anything.

`void playAuto();`

this function create random move parameters with the help of rand() function.

- Get random positions and passes to isMoveValid function.
- If isMoveValid return true call playUser(string move) function and play game on KlotskiBoard for one step.
- Else call playAuto () function again.

`void print() const;`

this function prints the elements of KlotskiBoard top left corner of the terminal. But it prints ‘ ’ instead of ‘0’.

`bool endGame() const;`

If huge square which is symbolized with char ‘B’ located at mid-bottom of KlotskiBoard it returns true; else, it returns false.

`bool stringValidity(string move) const;`

This function checks the validity of string if string is not in the correct form (example: G RIGHT) it returns false. Else it returns true.

`void initialize();`

this function initializes 2D vector of <char> as a KlotskiBoard.

Other Member Functions

`bool isMoveValid(char choice, string direction) const;`

this function takes parameters and check the possibility of entered moves. If move is possible it returns true; else, it returns false.

`void swapFunc(int momentY, int momentX, int newY, int newX);`

this function swaps the elements of KlotskiBoard.

class EightPuzzle : public BoardGame2D

    EightPuzzle class inherited by BoardGame2D.

private:

    vector < vector<int> > PuzzleBoard;
        this 2D vector keep game board.
    string LastMove;
        this string keeps last move of computer for the game.

public:

void setPuzzleBoard(vector < vector<int> > puzzleBoard);
vector < vector<int> > getPuzzleBoard() const;
void setLastMove(string direction);
string getLastMove() const;

    setter and getter functions of private variables of EightPuzzle class.

string RandomDirection();

    this function returns random direction in every separate call. (left/right/up/down)

bool isMoveValid(string direction) const;

    this function checks the possibility of move according to the direction string for integer "0". If it is possible return true; else return false.

Inherited functions

void playUser(string move);

    this function takes move string as a parameter and play game according to this parameter for one step.

void playAuto();

    this function create random with the help of RandomDirection() function.

- If random move is possible setLastMove and send random string to playUser(string move) function and it plays the game for one step on the PuzzleBoard.
- else call playAuto() function again.

void print() const;

    this function print the elements of PuzzleBoard top left corner of the terminal. But it prints "_" character instead of integer "0".

bool endGame() const;

```
      if PuzzleBoard ordered as 1 2 3 ,this function returns true.
                                4 5 6
                                7 8 0
```

bool stringValidity(string move) const;

    if move is left/right/up/down (there is no case sensitivity) this function returns true. Else return false.

void initialize();

```
      this function initialize the PuzzleBoard as -> 1 2 3
      then shuffles the board with random moves      4 5 6
                                                     7 8 0
```

int boardScore() const;

    This function returns the number of wrong positioned elements according to the game over situation.

`class PegSolitaire : public BoardGame2D`

Inherited functions

`void playUser(string move);`

this function take move string as a parameter and play game according to this parameter for one step.

`void playAuto();`

this function create random move parameters with the help of rand() function.

- Take these parameters and pass to isMoveValid function. If this function returns true pass these parameters to overload version of playUser function and play the game for one step.
- Else repeat all of these steps.

`void print() const;`

this function print the elements of PegBoard top left corner of the terminal. It prints “p” for “peg::pin”; “.” For “peg::empty”; “ ” for “peg::tab”.

`bool endGame() const;`

If there is no more move on PegBoard or BoardScore function returns any number smaller than 2, it returns true; else it returns false.

`bool stringValidity(string move) const;`

This function check the validity of string if string is not in the correct form (example: 3E RIGHT) it returns false. Else it returns true.

`void initialize();`

this function initialize 2D vector of peg objects to 2D vector of Cell objects. After that set 2D vector of Cell as PegBoard.

`int boardScore() const;`

this function returns the number of peg::pins of PegBoard.

Other Member Functions

`bool isMoveValid(int positionY, int positionX, string direction) const;`

this function takes parameters and check the possibility of entered moves. If move is possible it returns true; else it returns false.

`void playUser(int positionY, int positionX, string direction);`

this function overload version of playUser(string move). Only difference of these functions is parameter types.