CSE241 – HOMEWORK 5 REPORT

There are 3 classes (*PegSolitaire, EightPuzzle, Klotski*) that derived from base class BoardGame2D.

**-----> class BoardGame2D**

* This class has ***final virtual***, ***pure virtual*** and ***static*** functions**. playAutoAll()** final function plays the game in while loop until end of the game. It calls **endGame()** and **playAuto()** functions of related object. **playUser()** final function takes input and calls **playUser(string)** function of related object.
* **operator<<** function calls print\_board function.
* **playVector()** function plays all games randomly. In for loop, calls **playAutoAll**() function for all games. Then it prints scores of all boards. ***!!! It will take a couple of minutes to finish all puzzles because it will make to many steps.***

**-----> class PegSolitaire**

* This class inherited from BoardGame2D class. It has virtual functions from base class and some neccessary exclusive functions.
* Objects of this class have ***2D integer vector*** to represent board.
* It has *row, column* and *direction* as integer.
* **initialize()** function fills vector with type 2 game of Peg Solitaire puzzle.
* **print\_board()** resets terminal screen and prints board. PegSolitaire game is printed with 300 milliseconds delay. These value is *changeable* in PegSolitaire.cpp file.
* **decompose\_input()** : Because, “the string ***could be*** “2B UP” is written in the PDF, I decided to take input in ***“2B-U”*** format for PegSolitaire game.
* When input is decomposed into row, column and direction values, **SingleStep()** functions call **check\_position()** which controls chosen position is peg or not. If it is peg, then **check\_direction()** is called. This function controls movement in the given direction is valid or not. If it is valid, **check\_direction()** makes movement.
* **endGame()** function starts looking from top left to bottom right for every peg, and calls **find\_valid\_move** function. If that function finds a valid movement game continues.
* Different from the others, score of the PegSolitaire puzzle is calculated from remaning pegs. So **boardScore()** returns number of remaining pegs.

**------> class EightPuzzle**

* This class inherited from BoardGame2D class. It has virtual functions from base class and some neccessary exclusive functions.
* Objects of this class have **3x3 sized 2D integer** vector to represent board.
* It has *row, column, emptyRow, emptyColumn, score* as integer and *input\_validty* as boolean.
* **initialize()** functions fills vector randomly from 1 to 8.
* EightPuzzle game may be in some random boards. So, in constructor**, isSolvable()** function checks it. If it is not solvable, **initialize()** is called until it creates solvable board.
* **print\_board()** resets terminals screen and prints board. ***Because, number of steps for solving the EightPuzzle bu random is too high to represent every step, it has very small amount of time delay (1 nanoseconds). By auto, it may solve puzzle in 50.000 steps or 2.000.000 steps. It is completely random. So it may take a couple of seconds or minutes to solve puzzle.***
* **decompose\_input()** : In this puzzle, direction is not important because there is no direction option for any cell. So, input format of EightPuzzle game is ***“2B”*** as row and column respectively. Also this function controls validity of input.
* **check\_direction()** searchs an empty cell around given position. If it finds registers index of empty cell into *emptyRow* and *emptyColumn* variables.
* **endGame()** functions controls from top left to bottom right if all numbers are ordered.
* **boardScore()** function returns counter of step. At each successfull step, counter increases. By auto, it may be changed from 10.000 to millions.

**------> class Klotski**

* This class inherited from BoardGame2D class. It has virtual functions from base class and some neccessary exclusive functions.
* Objects of this class have **5x4 sized 2D char** vector to represent board and vector of ***Klotski\_Cell (inner class)*** to hold positions of chosen block type.
* It has *score, direction, flag, valid\_direction* as integer and *block* variable as char.
* **initialize()** function fills vector with letters from ‘A’ to ‘J’. Each letter represent its own block. For example: EE is 1x2 block and it cannot be seperated. A,B,C,D is 1x1 block, G,H,I,F is 2x1 block, E is 1x2 block, J is 2x2 block.
* **endGame()** : Checking top left and bottom right corner is enough to decided game is ended or not. So, if board[3][1] and board[4][2] corresponds ‘J’ game is ended.
* ***print\_board()*** resets terminals screen and prints board. ***Because, number of steps for solving the Klotski puzzle bu random is too high to represent every step, it has very small amount of time delay (1 nanoseconds). By auto, it may solve puzzle in 50.000 steps or 2.000.000 steps. It is completely random. So it may take a couple of seconds or minutes to solve puzzle.***
* **decompose\_input()** : In this puzzle, input format is ***“J-U”*** , letter and direction respectively. It also controls validity of input.
* After decomposing input, in **check\_direction(),** it registers positions into *Klotski\_Cell* object which has variables for row and column. Then adds it to *vector<Klotski\_Cell> .*
* Then in the **find\_direction(),** checks given direction in input or generated by random. Here, there is different conditions for each block type for some directions. For example, condition of 2x1 and 1x2 block types are different. Flag variable is used in here to decide movement is valid or not in that direction.

**|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||**

* Because, **EightPuzzle** and **Klotski** games have too many steps, in text file only **1** game of them will be written.
* ***!! -- Games will not stuck in infinite loop, they just takes long time to solve puzzle --!***
* **boardScore()** function return 0 only for ***PegSolitaire*** puzzle if remaining peg number is 1. Returning 0 at the end of the game for ***EightPuzzle*** and ***Klotski*** puzzle is pointless because PegSolitaire score is finite but these two games might have infinite score.
* **playVector()** function tests all functions of classes except **playUser** function. To test it, remove comments in main function.
* All functions that can be **const** is const.

***To run program:***

**make**

**./program**