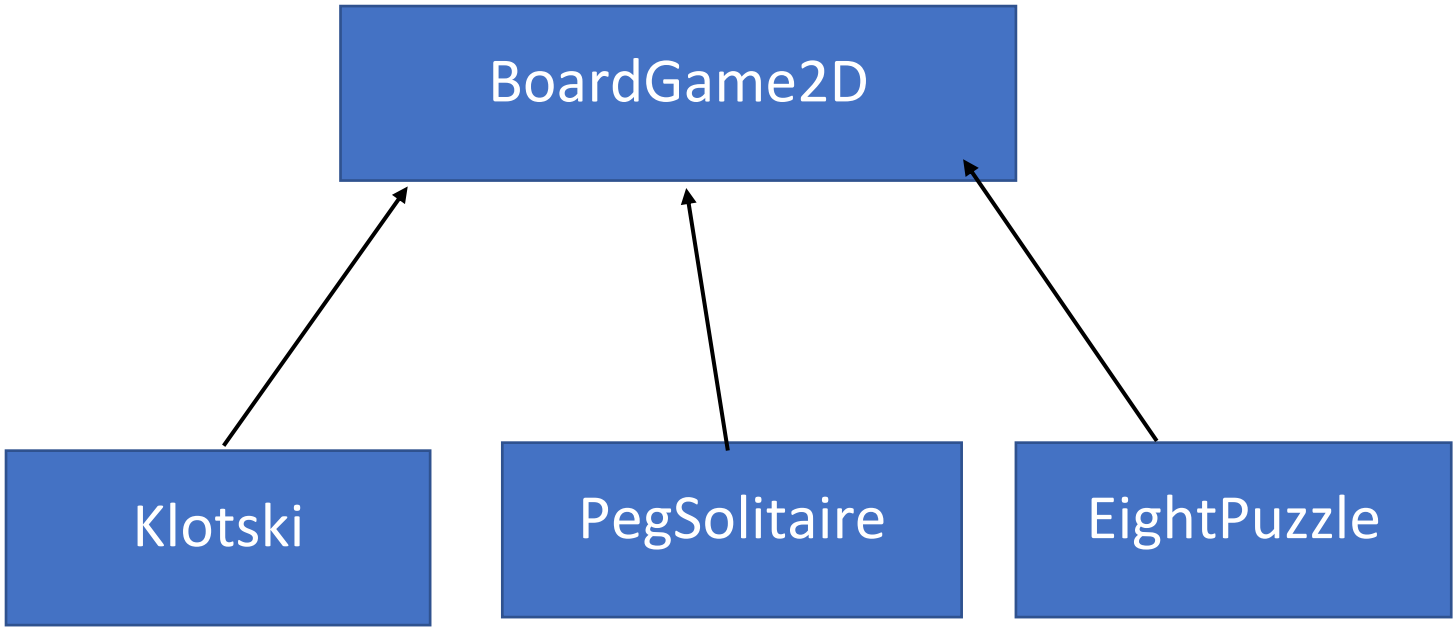
BURAK KOCAUSTA
1901042605

CSE 241 HOMEWORK 5

Class Interface Part

- Structure consists of 4 classes.
- BoardClass2D is an abstact base class, and other classes are derived from it.
- Derived classes overrides BoardGame2D class's pure virtual functions.

BoardGame2D hierarchy UML class diagram.



Polymorphic interface for the BOARDGAME2D hierarchy classes.

	BoardGame2D	PegSolitaire	EightPuzzle	Klotski
virtual void playUser(const std::string& command)	= 0	Accepts inputs like “A2 UP”	Accepts inputs like “2 UP”	Accepts inputs like “A UP”
virtual void playUser ( )	FINAL	-----	-----	-----
virtual void playAuto ( )	= 0	Makes one random move on type 2 peg solitaire board.	Makes one random move on the 3x3 board.	Makes one random move on the 4x5 Klotski board.
virtual void playAutoAll()	FINAL	-----	-----	-----
virtual void print ( )	= 0	Prints the Peg Solitaire board.	Prints the Eight Puzzle board.	Prints 4x5 Klotski board.
virtual bool endGame ( )	= 0	Returns true if Peg game is finished.	Returns true if Eight Puzzle game is finished.	Returns true if Klotski Game is finished.
virtual int boardScore ( )	= 0	Returns the number of the peg left.	Returns the number of inversion.	Returns the distance of 2x2 part of the board to the end state.
virtual void initialize ( )	= 0	Initializes the Peg board its untouched version.	Generates new random board.	Initializes the Klotski board its untouched version.

BoardGame2D Class

- It has pure virtual functions as indicated on the table.
- Apart from pure virtual functions;

friend std::ostream& operator <<(std::ostream& outs,const BoardGame2D& boardGame)

- This function overloads the << operator publicly.
- It calls print's another overloaded version, which takes ostream arg.
- With this function boards can be printed on screen, file etc.. with << operator.

virtual std::ostream& print (std::ostream& outs) const = 0

- Member Function is created for using << operator for derived classes.
- Function is protected so derived classes can use it.

static void playVector (std::vector<BoardGame2D *> boardGame)

That static function calls playAutoAll() function for each of vector's element. Prints board and finishes all the games.

virtual void playUser ( ) final;

Function calls playuser(const std::string& command) till the game ends.(It has an option to press "1" and let computer to finish the game).

virtual void playAutoAll() final;

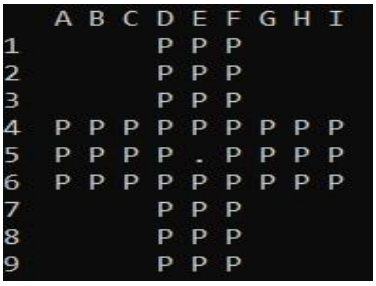
- Function plays the game till it is ended.
- It prints the board, and game's score.
- It calls the playAuto() member function.

- Some 2D games solving time might be long, so while printing the moves, Each move printed with some stop but if number of the move gets bigger than 100, board will no longer printed in each step, function continues to call playAuto(), but print() is no longer called. After the game is finished result and total number of the moves is printed.

PegSolitaire Class

- PegSolitaire inherits from BoardGame2D.
- Includes numOfPeg_, board_ and EndState_ in its private section
- It overrides print(), playAuto(), playUser(const std::string& command), endGame(), boardScore(), initialize(), and it also overrides print (std::ostream& outs) in protected section
- It has its inner class named Cell and it includes each cell's information like peg state, column, and row number.
- No need to big three because there aren't any dynamic data, but virtual destructor defined.

Board is printed like that, and playUser(..) function accepts commands like "E7 UP", "F3 DOWN" etc..

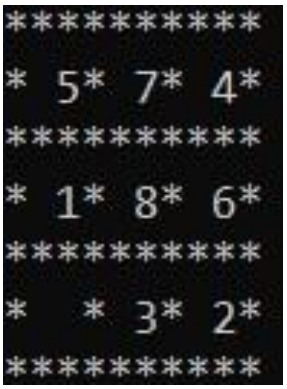


boardScore() function for this class returns the number of the peg left, and less peg means less score that means better score.

EightPuzzle Class

- EightPuzzle inherits from BoardGame2D.
- Includes boardSize_, board_ and EndState_ in its private section.
- Board is randomly generated while constructing or initialize() call occurs. It handles the unsolvable board situation.
- It overrides print(), playAuto(), playUser(const std::string& command), endGame(), boardScore(), initialize(), and it also overrides print (std::ostream& outs) in protected section.
- It keeps vector of int vector to represent the board.
- No need to big three because there aren't any dynamic data, but virtual destructor defined.

Board is printed like that, and playUser(..) function accepts commands like "7 UP", "3 DOWN" etc..



boardScore() function for this class returns the number of the inversion, if inversion equals to 0 this means game is finished. Also inversion is for the check boards validity. If it is not even, that means board is unsolvable. These situations are handled in implementation file with some helper functions in an unnamed namespace.

Klotski Class

- Klotski inherits from BoardGame2D.
- Includes board_, colBound_, rowBound_ and EndState_ in its private section.
- It has its inner class named KlotskiCell it holds each cell's label_(character), width_,length_.
- board_ is a vector of KlotskiCell vectors. Each cell is represented by KlotskiCell.
- It overrides print(), playAuto(), playUser(const std::string& command), endGame(), boardScore(), initialize(), and it also overrides print (std::ostream& outs) in its protected section.
- No need to big three because there aren't any dynamic data, but virtual destructor defined.

Board is printed like that, and playUser(..) function accepts commands like "P DOWN", "B LEFT" etc..

Same letter represents one block. For example B's are 2x2 block, and they are moving together.

boardScore() function for this class returns a value according to the distance of 2x2 block to winning place which means less score is better.

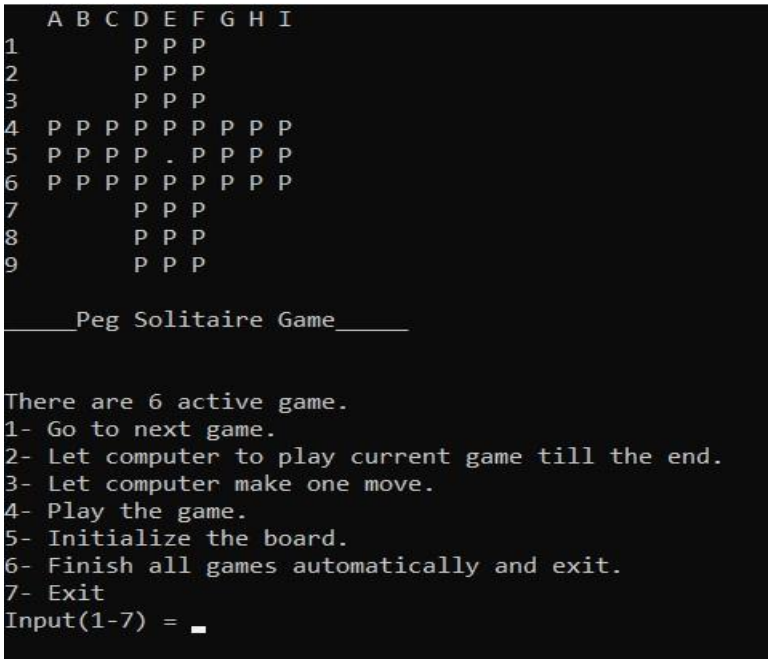
TESTING PART – DRIVER CODE

Main function consists of 5 important test functions.

test1(), test2(), test3(), test4(), writeToFile() ‘s results are inside of a separate TXT files.

Some of the test functions are commented out, If these are wanted to tested, please remove the comment lines.

- testMenu1() is a sample menu for to test functions easily. It must be used to test playUser() function. And all the other functions can be tested with this menu. playVector() can also be tested with this function.



- test1(), tests the PegSolitaire game.
- test2(), tests the EightPuzzle game.
- test3(), tests the Klotski game.

- test4(), is for testing playVector() function, it takes 6 board and finishes all of them. This test's result is submitted. If this is wanted to test, please remove comment lines.
- writeToFile( ) is for testing << operator while printing the file.

Compile the program with single make command, and run program with ./hw5 command.