

CSE 241 HW5 Report

≡ Name Surname	Ayşegül Aydoğan
≡ Number	190104004009

Table Of Contents

[Table Of Contents](#)

[General Explonation](#)

[BoardGame2d Explonation](#)

[Pegsolitaire Explonation](#)

[Detecting Finish Situation](#)

[EightPuzzle Explonation](#)

[Detecting Finish Situation](#)

[Klotski Explonation](#)

[Detecting Finish Situation](#)

[Tests](#)

[playVector.txt](#)

[PegSolitaire_test - EightPuzzle_test - Klotski_test](#)

[Makefile](#)

General Explonation

First of all, I do some research about inheritance and re watch some lectures. I had some prior knowledge but I have to refreash my knowledge.

After some research and trying some stuff on my own. I decide to declare nearly all possible functions as virtual in BoardGame2d and implement it as much as they can be implementable.

BoardGame2d Explonation

In this class, I created every expected function as `pure virtual` functions. And provide some implementation to them since they can usable as same as in subclasses too.

Also, I create some additional class to expected classes, and I also declared those as `pure virtual` functions. I also showed which functions are expected and which functions

are additional that added by me in class file too.

I used `protected` for keeping data since every data will be the same in games. For example `gameBoard` , `command` . I notice that, those are will be the same for every game.

Also I create a `static` function apart from expected `vectorGame` function. I created a static function that named `createRandom` . This function is usable in other child methods. Also it could be protected function too.

Since I created a good fundamental for every derived class, All class nearly should override those functions and they are done.

- `isValidMove()`
- `isValidCommand()`
- `printGameManual()`
- `move()`
- `printFinishMessage()`

Pegsolitaire Explonation

I had prior implementation for this game since older homeworks contains this. But after passing some time, It seems my older code a bit massy. So I changed some parts to more being more clean code and meaningful implementation.

I also add `pegCount` and `emptyCound` private data members to keep those informations apart from super classes `protected` members.

Also I changed commend style form `A5-U` to `A5 U` since in PDF it mentioned like this.

Detecting Finish Situation

```
 | abcdefg
1  ...
2  ..P
3  P.....
4  .....
5  ..P....
6  ...
7  PPP
Congratulations! You won the game!
You have completed the game in 26 moves.
Your score (Remaining Peg):6
```

A pegsolitaire game finished when there is no playable move. Score is equal to remaining peg count.

EightPuzzle Explonation

This game is very funny for me because I always want to implement this game.

Since my `BoardGame2d` has a good fundamental. I only implement some functions and game is nearly done.

I also create a functionality that generating a random board for every play. But there was a bit issue for this. Because, mathematically, some puzzles not solvable. So I created another function to determine that this game is solvable or not. If this puzzle is not solvable mathematically, It will be re generate random board until a solvable board generated.

I used this source: <https://www.geeksforgeeks.org/check-instance-8-puzzle-solvable/>

Also, this game has only 4 commands `U` (UP), `D` (DOWN), `L` (LEFT), `R` (RIGHT). Because this game playable with those 4 commands.

When a command executed, for example `U`. Blank space will go down and bottom cell will go up.

This game finished when has a this shape.

Finally, I created two variable called `blankSpaceX` and `blankSpaceY` to keep blank space' location. Game will respect to the blank space. When random board generated, those variables also will update due to blank space. I choose `0` as blank space.

Detecting Finish Situation

```
+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 |   |
+---+---+---+

Congratulations! You won the game!
You made 274781 moves.
Your score is equal to your move count.
```

Game finishes when all numbers are sorted and blank space is at the bottom right corner.



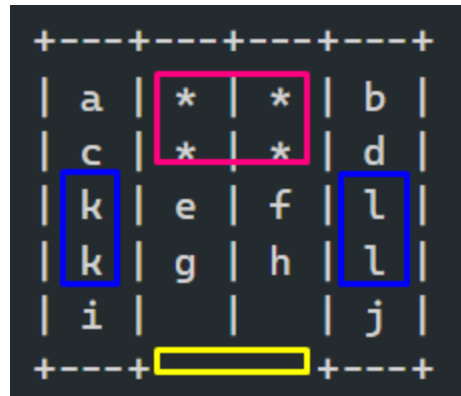
Score (move count) is very high because I don't implement a smarter random move for computer.

Klotski Explonation

Klotski game was a bit hard due to others. First I played a mobile game to understand the mentality of the game.

Android Game Link: <https://play.google.com/store/apps/details?id=com.alcamasoft.juegos.klotski.android&hl=en&gl=US>

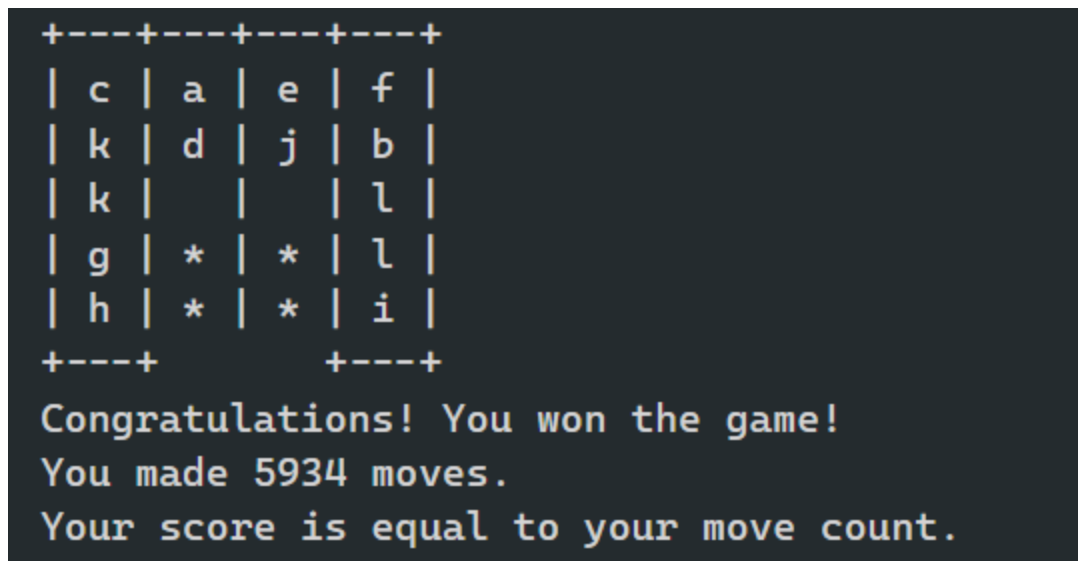
After understanding the basics of the game, I tried to implement accordingly.



I used this picture to show what I did. Blue ones (`k` and `l`) are two sized. They are moving together. Other ones are one sized. Yellow location is the purpose of the game. When Purple area `*` ones can exit the board (when access the yellow line) game will be finish.

As others, I only implement few functions and game was playable. But it is a bit hard to moving 2 sized blocks (`k` and `l`).

Detecting Finish Situation



When `*` ones arrives to the exit. Game will finish. Like [EightPuzzle](#) game. This games counts move counts as score.



Like EightPuzzle game, I don't implement a smarter random generator. Because of that, score (move count) is very high.

Tests

In `Test Results` file. There is a 4 `.txt` file.

playVector.txt

This file contains the result of `static void playVector(..)` function. It is printed as resulted.

Also when user calls this vector, This function will ask to play as human or computer. When computer selected, this will also show to see final result or play one by one.

PegSolitaire_test - EightPuzzle_test - Klotski_test

Those files contains same tests but for different games. All tests results are printed to this files.

Makefile

When user types `make` in terminal, all files will be compiled and will generate a `output` file. And than, this `output` binary file will be executed.

If anyone don't change my implementation, `playVector()` function will be called and this function will start playing as `PegSolitaire` twice, `EightPuzzle` twice and than `Klotski` twice as expected.