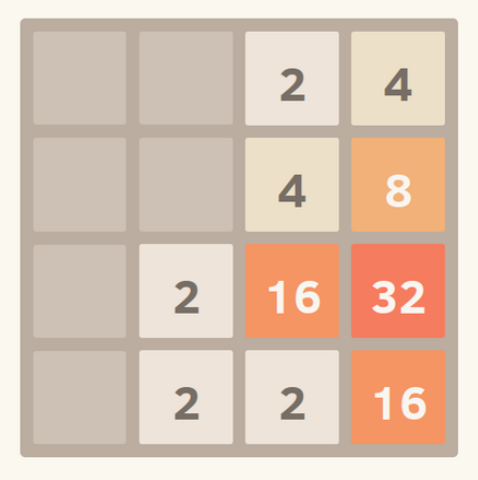
Week 14

Remind

- 抄襲一律 0 分 （包含被抄襲者）
- 繳交期限: **12/20(Sun.) 11:59 p.m.**
- 繳交的檔案格式、名稱請符合以下規定
  - 請繳交 zip檔至 Ceiba作業區，名稱為 <student_id>.zip
  - 解壓縮後須符合格式、名稱
  - e.g. b12345678.zip
- 必須完成 Demo 才可以提早離開
- 若沒有完成 Demo 就中途早退，視同缺席
- 若當天沒有完成Demo，請以螢幕錄影解釋程式碼，並於繳交期限前將影片寄至助教信箱
  ee10042020@gmail.com

Problem - 2048(alphabet ver.) (10%)

Introduction



https://en.wikipedia.org/wiki/2048_(video_game)

The alphabet version of 2048 uses A to denote 2, B to denote 4, C to denote 8, and so on.

You will be implementing two classes: the `Board` class and `Game` class. For each class, please write a header file only contains the declaration and a implementation file (.cpp) contains the implementations.

Class `Board` (5%)

1-1.

Description

Declare one private variable:

1. tiles (4×4 2D array)

1-2. Constructor

Description

Initialize the tiles to empty.

Then, randomly pick two tiles on the board and set the value to 'A'.

1-3. Display the board

`void displayBoard()`

Description

Print the board with the values of titles.
Note that both sides of the value contain a white-space.

Sample output

```
+---+---+---+---+
|   | A |   |   |
+---+---+---+---+
|   |   |   |   |
```

```
+---+---+---+---+
|   |   | A |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
```

1-4. New a tile

`void newTile()`

Description

Randomly new a tile in an empty spot on the board and the value is also randomly picked from 'A' or 'B' or 'C'.

1-5. Move a single line

`int moveSingleLine(char line[4], char new_line[4])`

Description

Slide left and merge the tiles of the array. `line[4]` is the original line and store the result in `new_line[4]`.

Meanwhile, calculate the score gained from this single line move and return the score.

The score is increased whenever two tiles combine, by the value of the new tile. The value of 'B' is 4, 'C' is 8, 'D' is 16, and so on.

Sample

`line[4]`

```
+---+---+---+---+
| A |   |   | A |
+---+---+---+---+
```

Result

`new_line[4]`

```
+---+---+---+---+
| B |   |   |   |
+---+---+---+---+
```

return `4`

```
+---+---+---+---+
```

```
+---+---+---+---+
```

```
| B |   | B | B |
+---+---+---+---+
```

```
| C | B |   |   |
+---+---+---+---+
```

return `8`

1-6. Move the board

`int moveBoard(Direction dir)`

Description

Use **1-5** to update each line in the input direction.

Use **1-4** to new a tile after update the board. If the board doesn't changed, do not new the tile.

Also, calculate the score gained from this move and return the value.

Sample

`tile[4][4]` , `Direction = LEFT`

```
+---+---+---+---+
| A | A |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | C |   |   |
+---+---+---+---+
```

Result

`tile[4][4]`

```
+---+---+---+---+
| B |   |   |   |
+---+---+---+---+
|   | A |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
| C |   |   |   |
+---+---+---+---+
```

return `4`

`tile[4][4]` , `Direction = RIGHT`

```
+---+---+---+---+
| A |   | C | B |
+---+---+---+---+
|   | A | D | D |
+---+---+---+---+
|   |   |   | A |
+---+---+---+---+
```

`tile[4][4]`

```
+---+---+---+---+
|   | A | C | B |
+---+---+---+---+
|   |   | A | E |
+---+---+---+---+
|   |   |   | A |
+---+---+---+---+
```

```
|   |   |   |   |
+---+---+---+---+
```

```
| B |   |   |   |
+---+---+---+---+
```

return `32`

1-7. Check game-over

`bool checkGameOver()`

Description

Check if the game is over and return a bool value.

Sample

```
+---+---+---+---+
| A | A |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | C |   |   |
+---+---+---+---+
```

Result

return `False`

```
+---+---+---+---+
| F | A | F | G |
+---+---+---+---+
| D | C | A | C |
+---+---+---+---+
| B | A | B | D |
+---+---+---+---+
| C | B | C | A |
+---+---+---+---+
```

return `True`

File

Game.h

Game.cpp

Class `Game` (3%)

2-1.

Description

Declare two private variables:

1. board
2. score

2-2. Constructor

Description

New a Board object and set the value of the score to zero.

2-3. Destructor

Description

Delete the board.

2-4. Read input

`Direction read_input()`

Description

Read the user input and return the direction.

Let 'e' be UP, 'd' be DOWN, 's' be LEFT, 'f' be RIGHT, others be INVALID.

2-5. Display the score

`void printScore()`

Description

Print the score with a newline at the end.

Sample output

```
Score: 100
```

2-6. Start game

`void start()`

Description

Use the functions and the class implemented above to construct the game.

Print the score first and the board whenever a move is made.

If the game is over, print "Game Over" and the final score.

File

Board.h

Board.cpp

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/53b90049-1238-43a7-ad8f-0d624e11667e/2048.cpp

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/cf8c5c2a-a054-41d7-9f91-4920c969fbf7/Direction.h

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6ced7265-72a3-4b52-93be-c76dcac0ffe4/Game.h

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/cce7158d-c76c-49d0-a100-2588450e768d/Game.cpp

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c9a67e41-60fd-4a61-93d0-8e88b0c50b3f/Board.h

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2a894a27-72dc-4092-b44f-156f6b415d43/Board.cpp

Challenge Problem (2%)

Design a strategy to get the highest score.

You can modify anything of your code while having the following functions.

Class `Game`

3-1. Get score

`int getScore()`

Return the value of the score.

3-2. Auto play

`void autoPlay()`

To start playing, invoke this function instead of `void start()`.

We will judge the performance by playing 1000 times and averaging the scores.

Note that you don't need to print the board and score during playing but we will check the correctness of your code.

```
int main(void){
    int acc_score = 0;
    for( int i = 0 ; i < 1000 ; i ++){
        Game *game = new Game();
        game->autoPlay();
        acc_score += game->getScore();
        delete game;
    }
    cout << "Average score: " << (double)acc_score/1000 <<endl;
    return 0;
 }
```

File

Game_auto.h

Game_auto.cpp

Board_auto.h

Board_auto.cpp