Programming Design (II) Project 2 【2048】

**report**

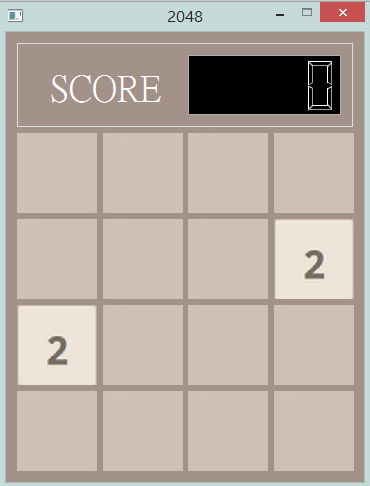
資訊四 F74004046 劉慎之

--------------------------------------------------------------------------------------------------

* 【程式執行方式】：

利用Qt creator 來執行建立出執行檔之後便可以執行遊戲

★ 【遊戲執行畫面】：

⮉ 遊戲開始畫面 ⮉ 遊戲進行畫面 ⮉ 遊戲結束畫面

★ 【遊戲功能說明】：

1/ 遊戲開始執行即會出現開始畫面

2/ 藉由上下左右移動的方式移動方塊，遇到相同數字會合併

3/ 直到出現數字2048的方塊，遊戲即停止

4/ 若遊戲失敗，會彈出視窗，決定要重新開始遊戲還是要離開

遊戲設置計分功能，可以記錄下分數

★ 【UML class diagram】：

|  |
| --- |
| Widget |
| － ui: Ui::Widget\*;  －tile[4][4]:Tile\*  －score:int  －resetb:QPushButton\*  －exit:QPushButton\*  －endgame:QWidget\*  －layout:QVBoxLayout\*  －word:QLabel\* |
| <<constructor>>Widget(QWidget \*parent = 0)  <<destructor>>~Widget()  + reset();  + quits();  + changed(score:int);  － createboard()  － startGame()  － randomtile()  － keyPressEvent(event:QKeyEvent\*)  － isFull()  － exchange(Box1:Tile \*,Box2:Tile \*)  － emptytileMoveLft()  － emptytileMoveDown()  － emptytileMoveRight()  － emptytileMoveUp()  － moveUp()  － moveRight()  － moveLeft()  － moveDown()  － movepossible()  － ndGame(ch:int) |

|  |
| --- |
| Tile |
| － number:int  －image:QPixmap\* |
| <<constructor>> Tile(QWidget \*parent = 0);  + paintEvent(QPaintEvent \*);  + getNumber()  + setNumber(num:int)  + isEmpty()  + upgrade() |

★ 【程式設計邏輯】：

1. 一開始執行時先執行createboard函數，先把ui界面中拉出來的frame指定給形態為Tile的pointer，依序為tile [0] [0] ~ tile [4] [4]

★ 相對應的坐標如下

+-----+-----+-----+-----+

| | | | |

| 0，0 | 0，1 | 0，2 | 0，3 |

| | | | |

+-----------------------+

| | | | |

| 1，0 | 1，1 | 1，2 | 1，3 |

| | | | |

+-----------------------+

| | | | |

| 2，0 | 2，1 | 2，2 | 2，3 |

| | | | |

+-----------------------+

| | | | |

| 3，0 | 3，1 | 3，2 | 3，3 |

| | | | |

+-----+-----+-----+-----+

而每一個tile中都有一個number，初始為0，代表是空的格子，顯示的圖示為0.png

1. 在程式執行時會依據number來替換圖片，

Ex： tile[1][1]的number為2；則它顯示的圖片會是2.png，若是4則是4.png etc

1. 再來則執行startgame，將隨機兩格number變成2，然後使用者開始使用上下左右移動方塊，若是按上，則會先用movepossible檢查是否能移動

判斷的規則是先判斷是否所有格子的numner大於0，若沒滿代表可移動，若是滿了則檢查相同的行列，是否有number相同且兩兩相鄰的格子，如果有代表可移動。

1. 判斷可以移動則執行moveup，會先執行emptytileMoveDown將空的格子移動到最下面，再來看往上時有沒有相同number的格子相鄰，如果有則下方的格子歸零，上方的格子執行upgard函數(number\*2)

Ex：[0][0]和[1][0]number都是2，往上移動時[1][0]變為0，[0][0]upgrade為4，再來把score加上upgrade後的數字，然後觸發changed函數，改變lcdnumber顯示的分數，

1. 之後如果upgrade之後有出現2048則endgame，移動完之後；則執行randomtile，抽出隨機兩個空格子並將他們number變為2或4，按其他方向時以此類推最後出現2048或者沒出現2048而且也無法在移動則執行endgame，會跳出一個視窗顯示2048或者gameover，下方則有按鈕reset和exit，按下reset則會觸發信號槽裡的reset函數，所有格子number歸0分數也歸0，然後重新執行startgame，按下exit則是關閉程式。