UML class diagram

|  |
| --- |
| gameWindow |
| +keyPressEvent(QKeyEvent \*event=0) :void  +movedrum1():void  +movedrum2():void  +movedrum3():void  +movedrum4():void  +movedrum5():void  +movedrum6():void  +showTime():void  +showscore():void  +on\_pushButton\_clicked():void  +on\_pushButton\_2\_clicked():void  +on\_pushButton\_3\_clicked():void |
| - count\_time:int  - x1:int  -x2:int  -x3:int  -x4:int  -x5:int  -x6:int  - a:int  - iv,y:int  -score\_num:int  -gameWindow \*game:int |

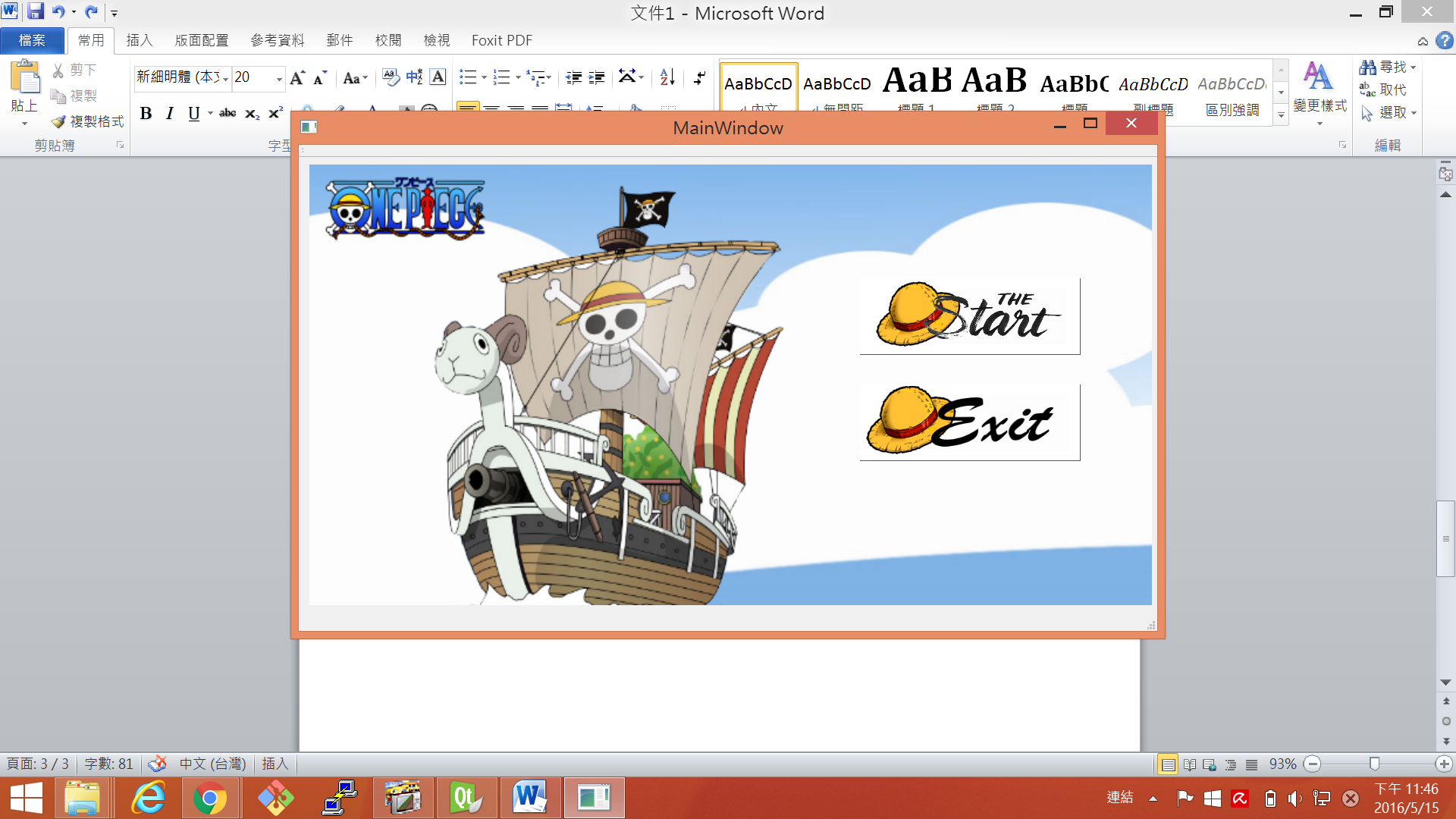
|  |
| --- |
| hardWindow |
| *+keyPressEvent(QKeyEvent \*event=0):void*  *+showpicture():void*  *+showhardtime():void*  *+showpicture1():void*  *+movedrum1():void*  *+movedrum2():void*  *+movedrum3():void*  *+movedrum4():void*  *+movedrum5():void*  *+movedrum6():void*  *+moveflower():void*  *+hshowscore():void*  *+on\_pushButton\_clicked():void*  *+on\_pushButton\_3\_clicked():void*  *+on\_pushButton\_2\_clicked():void* |
| *-pix:int*  *-cintime:int*  *-y:int*  *-fx:int*  *-fy:int*  *-h1:int*  *-h2:int*  *-h3:int*  *-h4:int*  *-h5:int*  *-h6:int*  *-v1:int*  *-v2:int*  *-v3:int*  *-v4:int*  *-v5:int*  *-v6:int*  *-* hscore\_num*:int* |

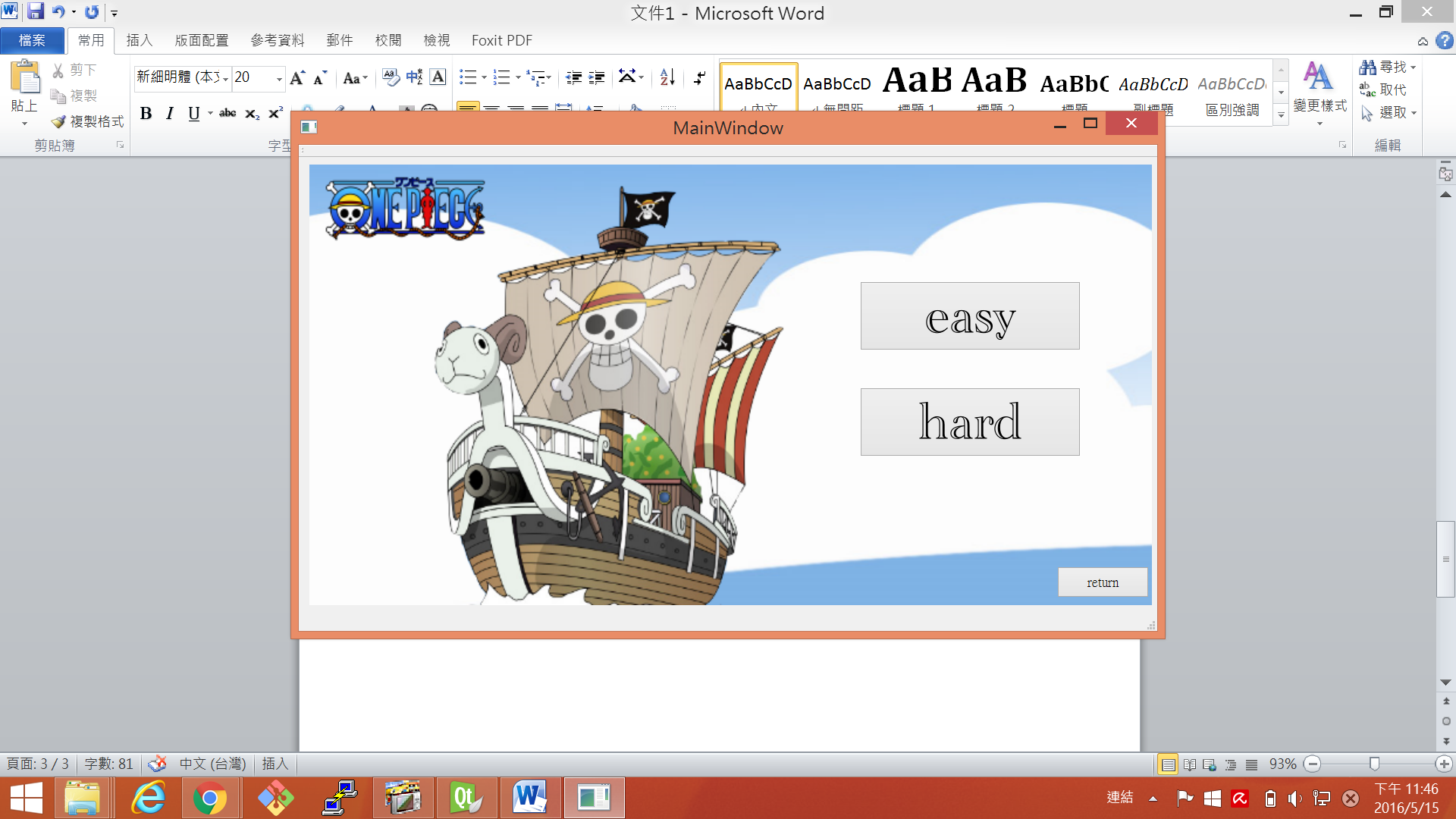
*-*

|  |
| --- |
| *mainWindow* |
| *+ on\_startButton\_clicked():void*  *+on\_pushButton\_2\_clicked():void*  *+on\_easy\_clicked():void*  *+on\_hard\_clicked():void*  *+on\_pushButton\_clicked():void*  *+on\_re\_clicked():void* |
|  |

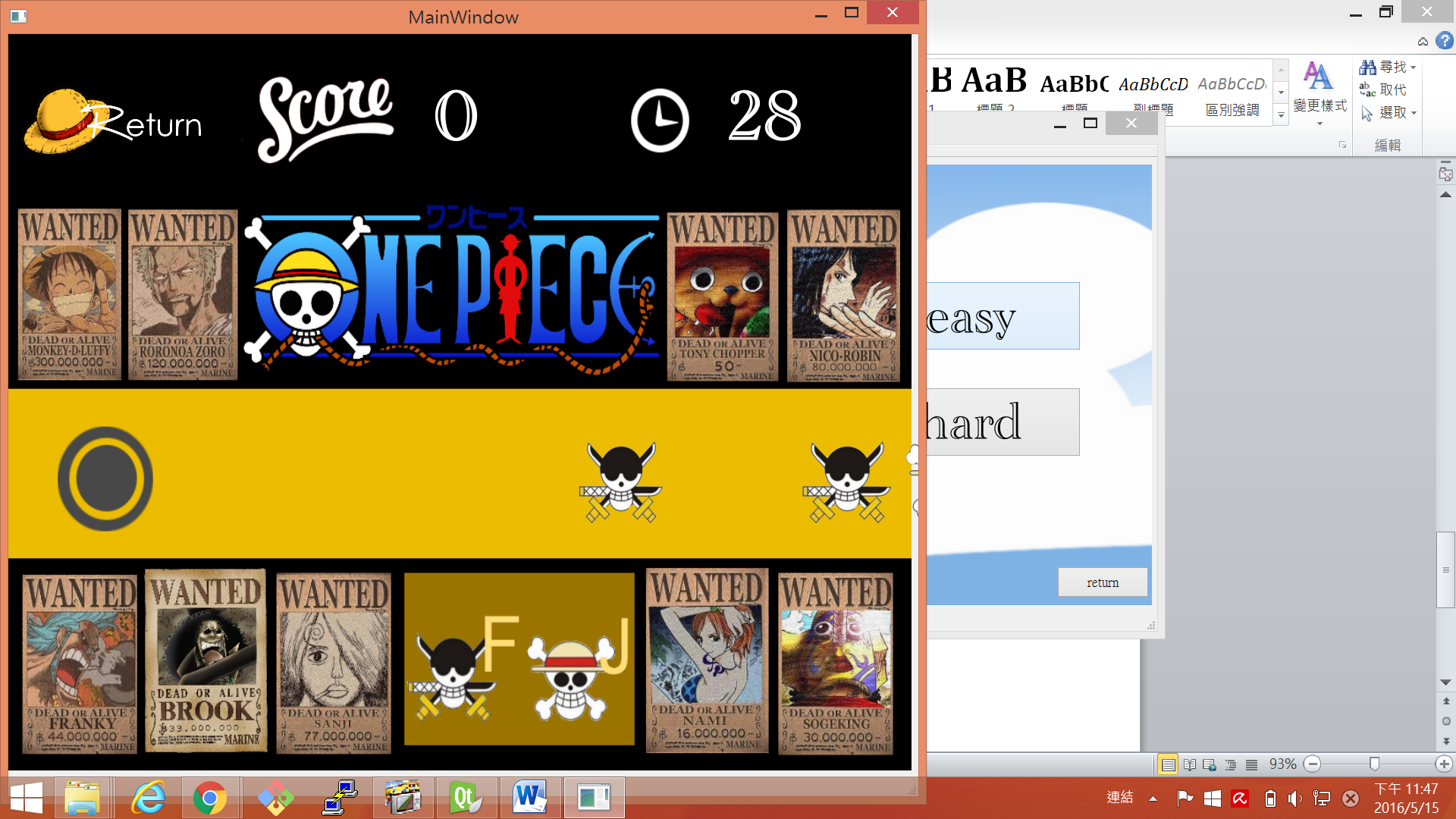
Screen shot

stratwindow

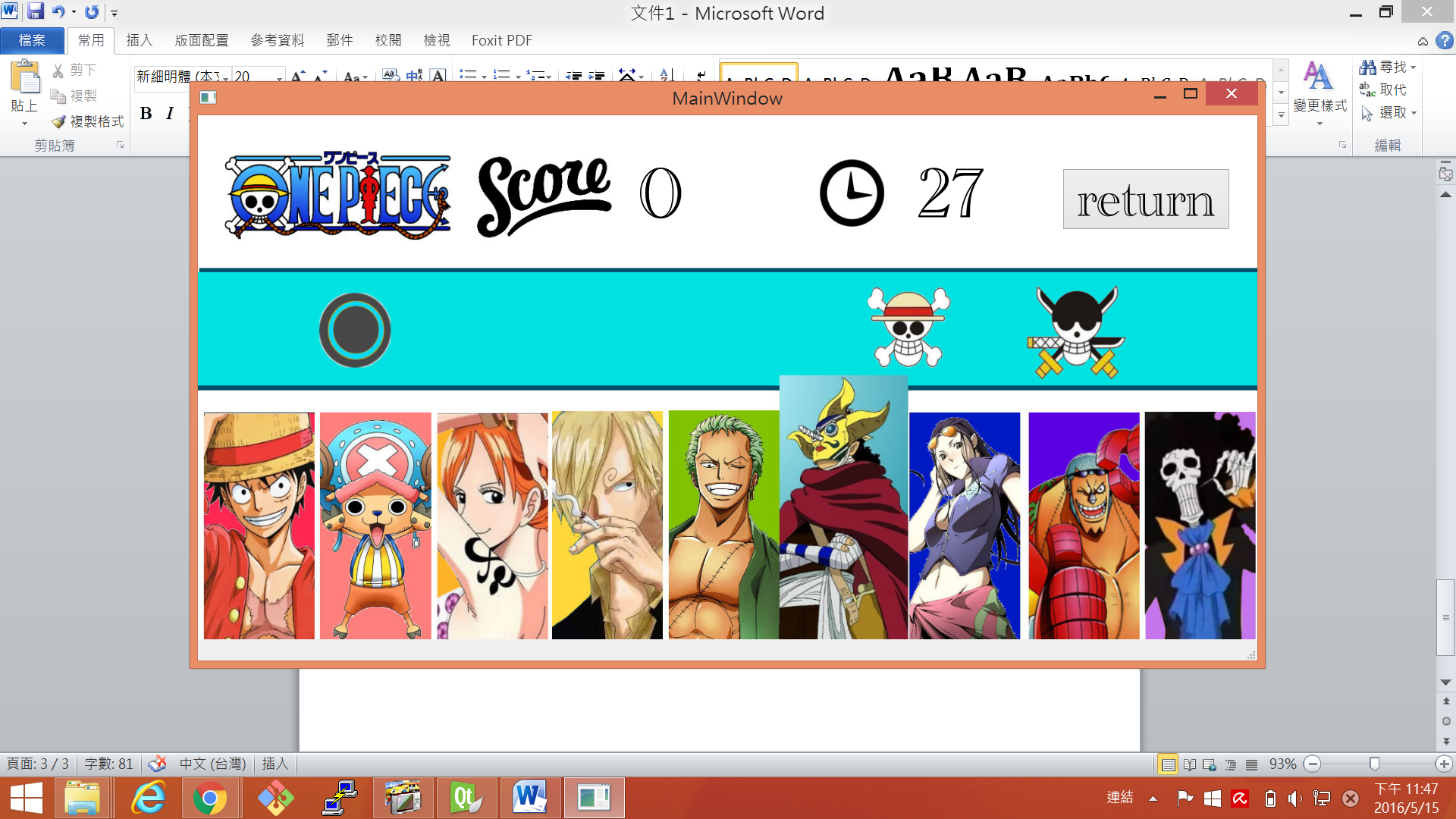




easy



hard



score



How to play

在開始介面的時候

點start會有兩種模式可以選擇

選簡單的速度是固定的

困難得速度會是不固定的

黑色的鼓按F

白色的是J

program architecture

在簡單的遊戲中

我用random跑圖片位置讓他從不同位置跑出來

造成random鼓的樣子

在困難的遊戲中

我除了random位置之外也random我的速度

讓他每跑完一次速度會不一樣

另外我利用signal&slot讓背景圖片動