UML class diagram :

Ball

+ x : Integer

+ y : Integer

+ type : Integer

- pic : QPixmap

- vx : Integer

<<constructor>> + Ball( \_x : Integer, \_y : Integer )

<<destructor>> + ~Ball()

+ UpdatePosition()

+ PaintBall( painter : QPainter& )

QObject

Game

+ score : Integer

- sec : Integer

- timespace : Integer

- welcomewindow : WelcomeWindow\*

- mainwindow : MainWindow\*

- movetmr : QTimer\*

- gentmr : QTimer\*

- balls : QList<Ball\*>

<<constructor>> + Game( welcome\_window : Welcome

Window\*,main\_window : MainWindow\*, parent : QWidget\* )

<<destructor>> + ~Game()

+ RefreshPosition()

+ GenBall()

# paintEvent( event : QPaintEvent\* )

- keyPressEvent( event : QKeyEvent\* )

<<destructor>> + ~Game()

QWidget

MainWindow

- ui : MainWindow\*

<<constructor>> + MainWindow( parent : QWidget\* )

<<destructor>> + ~ MainWindow()

QMainWindow

QMainWindow

WelcomeWindow

- ui : WelcomeWindow\*

- mainwindow : MainWindow\*

<<constructor>> + WelcomWindow( parent : QWidget\* )

<<destructor>> - ~WelcomeWindow()

- on\_start\_clicked()

- on\_exit\_clicked()

EndWindow

- ui : EndWindow\*

- welcomewindow : WelcomeWindow\*

- mainwindow : MainWindow\*

- game : Game\*

<<constructor>> + EndWindow( welcome\_window : WelcomeWindow\*, main\_window : MainWindow\*, oldgame : Game\*, score : Integer, parent : QWidget\* )

<<destructor>> + ~EndWindow()

+ closeEvent( : QCloseEvent \* )

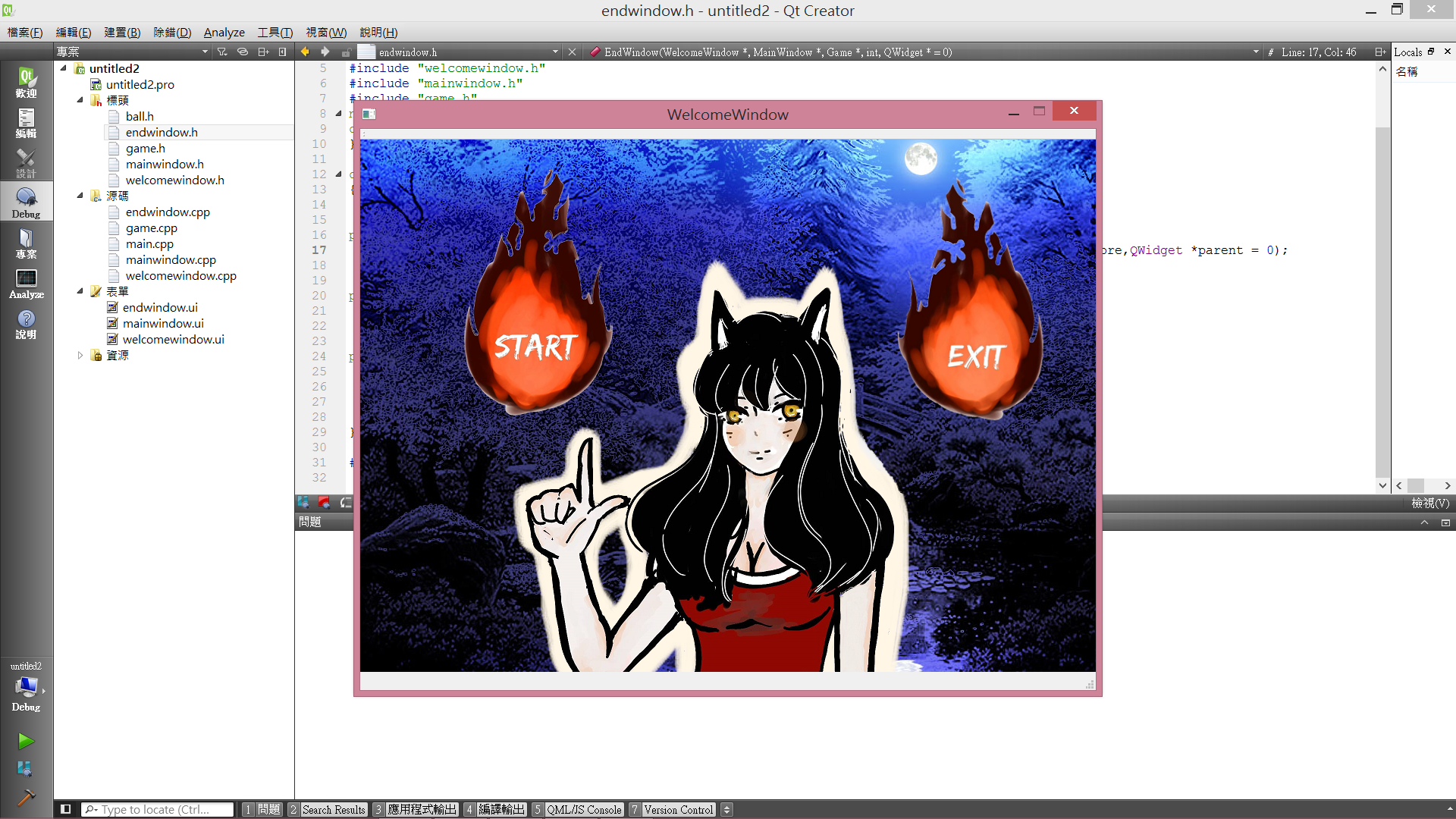
- on\_quit\_clicked()

- on\_restart\_clicked()

QMainWindow

Screen shot :

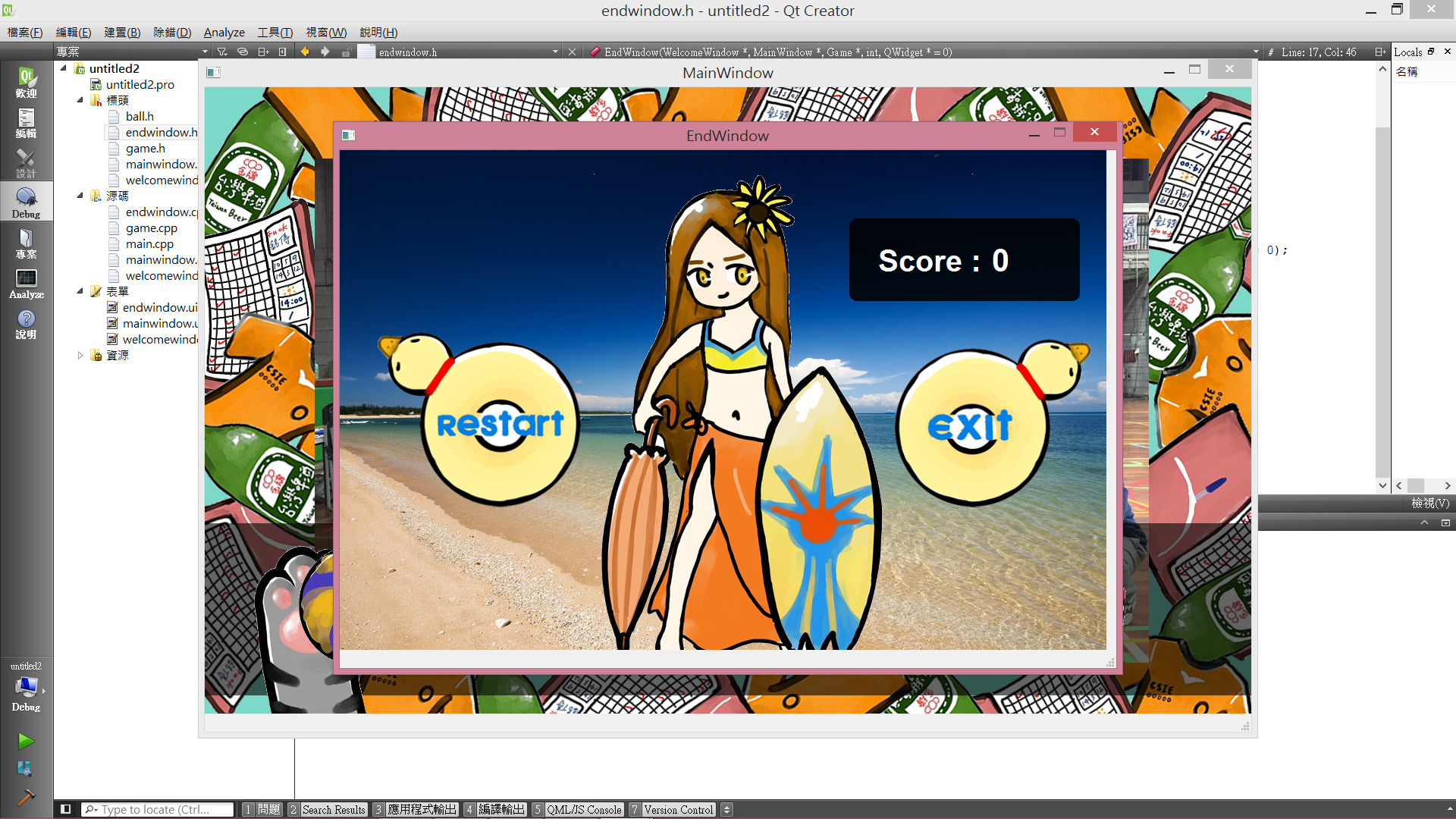
開始畫面



遊戲畫面



結束畫面



How to play :

按C鍵消去Conti 按M鍵消去Mikasa

在排球到達貓掌的時候按鍵消去

Program architecture :

把視窗分為三大類：歡迎視窗 WelcomeWindow、主視窗 MainWindow、結束視窗 EndWindow。

主視窗為歡迎視窗的 Data Member，當遊戲開始時，歡迎視窗會先隱藏自己並顯示主視窗，同時 new 出一個 Game object，負責繪製遊戲畫面於主視窗上。遊戲過程中將定時產生新的排球、移動舊有排球以及減少秒數。

當秒數歸零時會隱藏主視窗並 new 出一個結束視窗。在結束視窗中，若使用者選擇 Restart，則會 delete 剛剛的 Game object 並回到歡迎視窗重複上述步驟，反之將結束程式。