UML class diagram :

Ball

+ x : Integer

+ y : Integer

+ type : Integer

- pic : QPixmap

- vx : Integer

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

+ UpdatePosition()

+ PaintBall( painter : QPainter& )

<<destructor>> + ~Ball()

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\* )

+ RefreshPosition()

+ GenBall()

# paintEvent( event : QPaintEvent\* )

- keyPressEvent( event : QKeyEvent\* )

<<destructor>> + ~Game()

QWidget

MainWindow

- ui : MainWindow\*

+ MainWindow( parent : QWidget\* )

<<destructor>> + ~ MainWindow()

QMainWindow

QMainWindow

WelcomeWindow

- ui : WelcomeWindow\*

- mainwindow : MainWindow\*

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

- on\_start\_clicked()

- on\_exit\_clicked()

<<destructor>> - ~WelcomeWindow()

EndWindow

- ui : EndWindow\*

- welcomewindow : WelcomeWindow\*

- mainwindow : MainWindow\*

- game : Game\*

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

+ closeEvent( : QCloseEvent \* )

- on\_quit\_clicked()

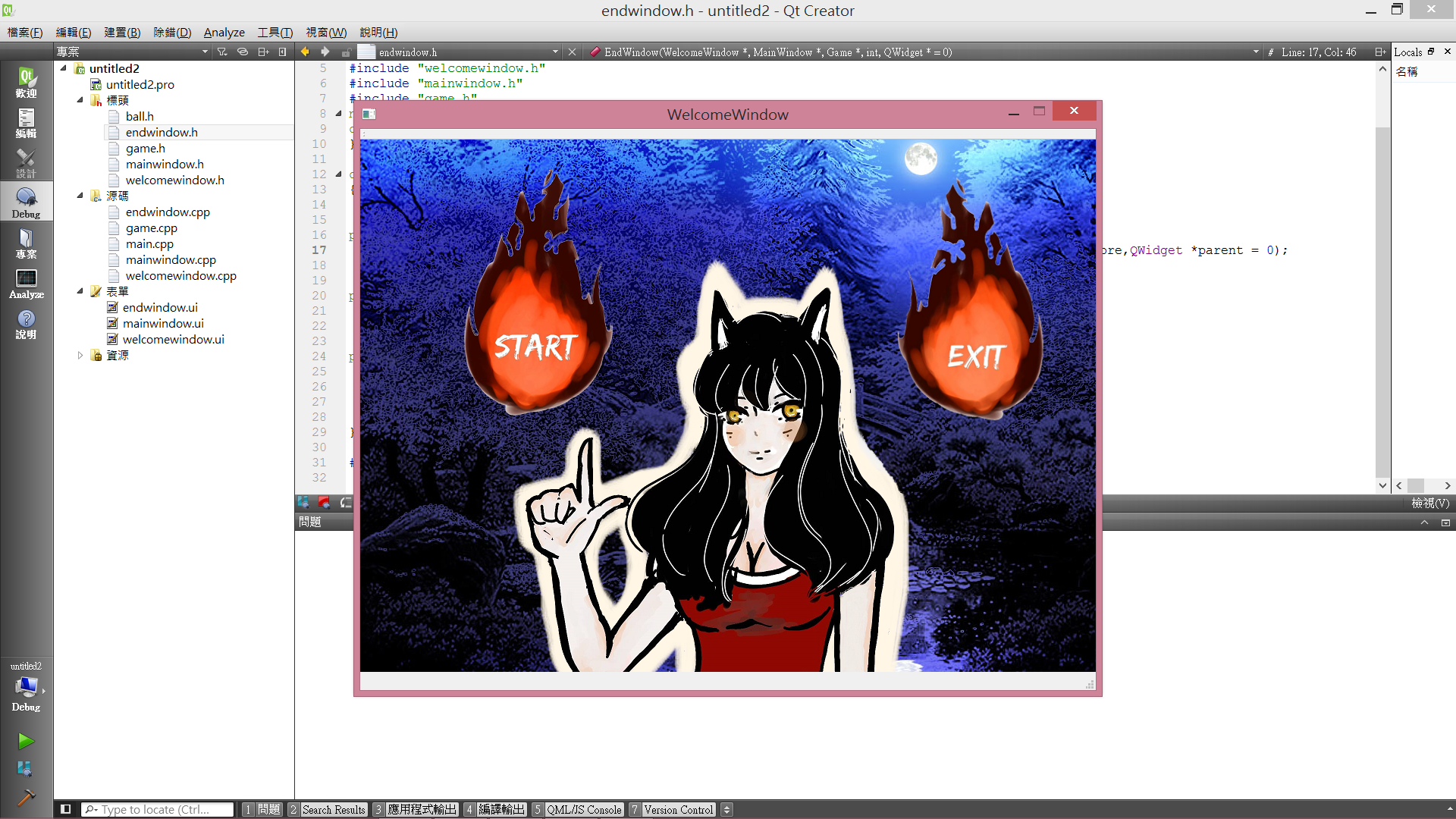
- on\_restart\_clicked()

<<destructor>> + ~EndWindow()

QMainWindow

Screen shot :

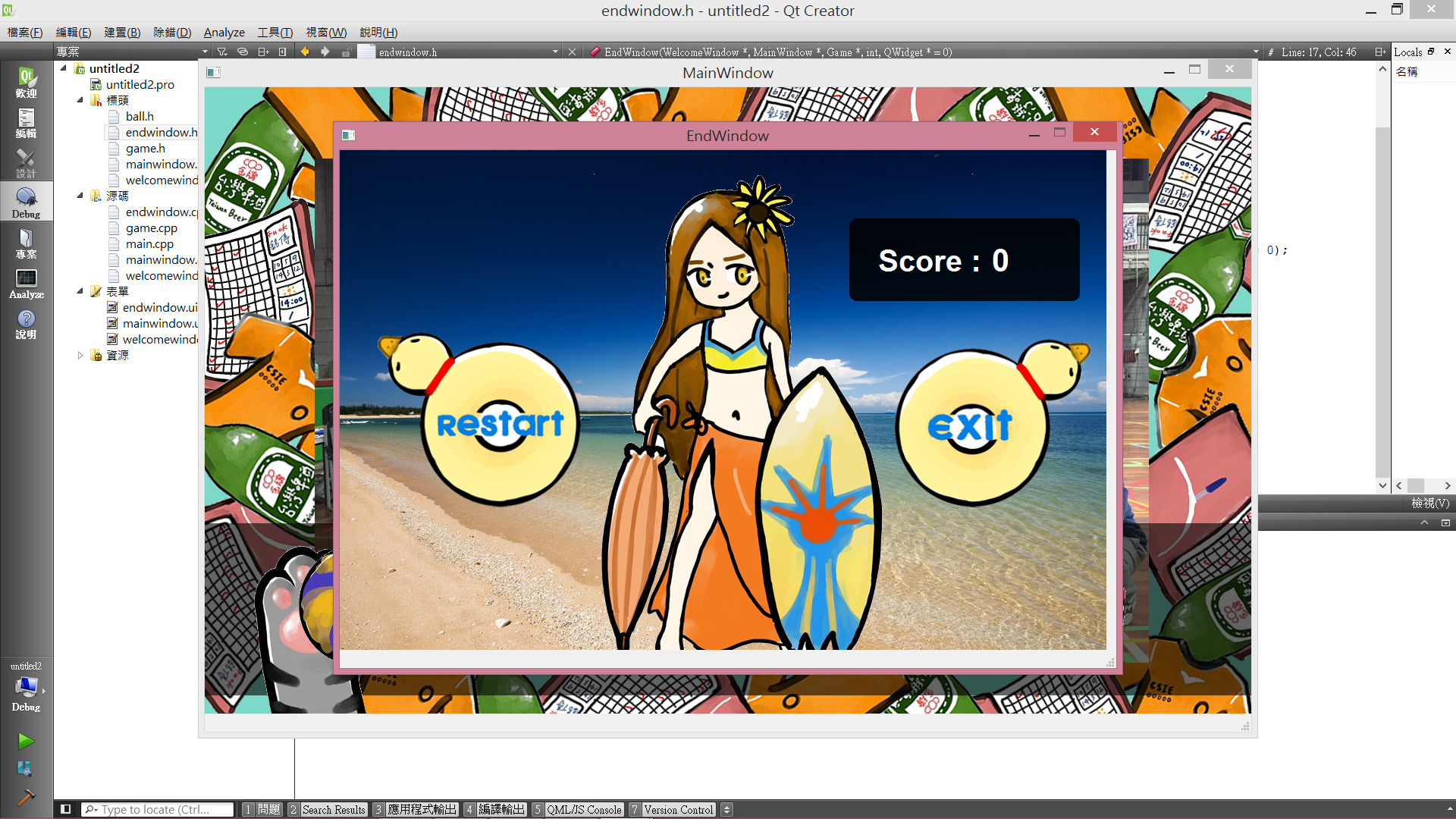
開始畫面



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 :

把視窗分成三個部分：歡迎畫面WelcomeWindow、主畫面MainWindow、結束畫面EndWindow。歡迎畫面再選擇開始後先隱藏，並產生data member主畫面，遊戲開始時把新Game Class 畫在主畫面上，並定時產生排球、定時移動排球、定時減少秒數。秒數歸零時new出一個結束畫面，隱藏主畫面，顯示分數，如果選擇restart就delete剛剛的遊戲，回到歡迎畫面重複循環。