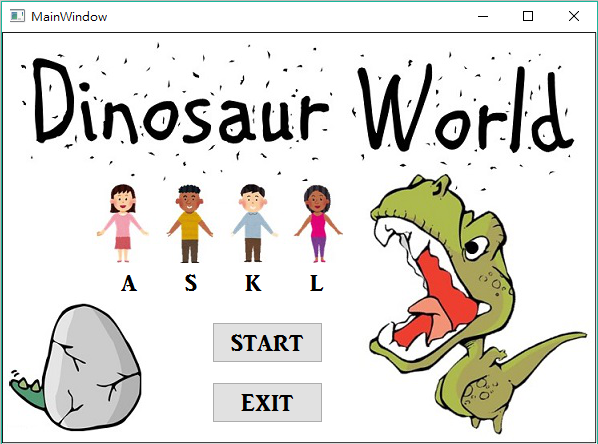
pd2-Taiko Report

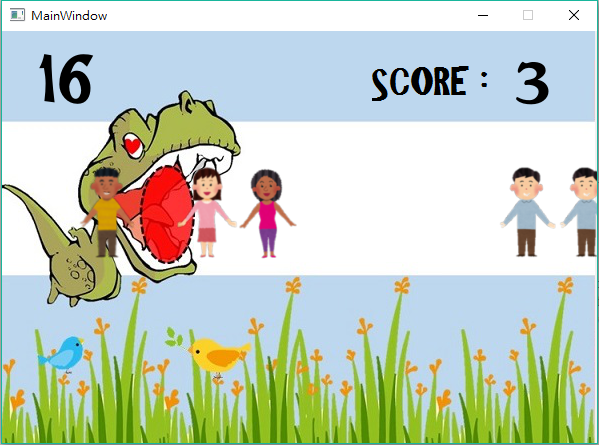
H34016144 許欣穎

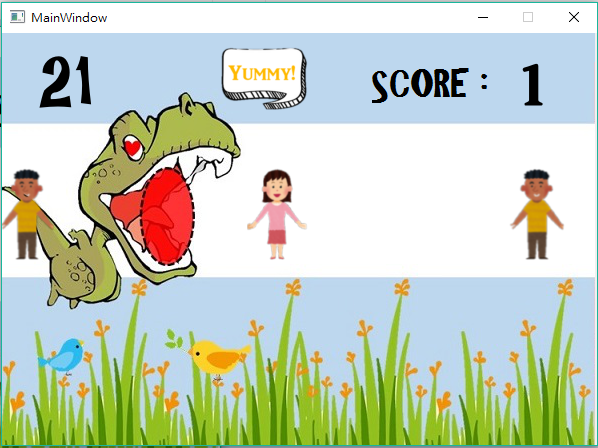
【How to play】

1. 執行後就會出現開始介面(圖1)
2. 介面上有介紹四種人物代表的按鍵，**點選Start就會開始遊戲，點選EXIT就會結束程式**。
3. 開始遊戲後，會**倒數計時３０秒**，時間到就結束遊戲(圖2)。
4. 遊戲中**四種人物會隨機出現**，當人物走到恐龍嘴巴中的圈圈時就要**按下對應的按鍵來吃掉人物**，若有**正確的吃掉就會在SCORE加一分，並且出現YUMMY!字樣**(圖3)；若**按錯按鍵或時機不對就會出現OOPS!字樣**(圖4)。**一旦分數到８分以上，會出現恐龍吃人的新聞** (圖5)，若**時間剩下１０秒時，分數未達八分，則會出現肚子餓的小恐龍**(圖6)。
5. 時間到，結束遊戲時，會**跳出一個Dialog顯示你得了幾分**，若**分數在８分以上，會出現”Good Dinosaur!”的字樣**(圖7)**，若在未達八分，則會出現”Try Harder”字樣**(圖8)。Dialog上還會有兩個選項，選擇**Play Again的話，會回到開始介面**，可以重新開始，選擇**Exit的話，則會結束程式**。

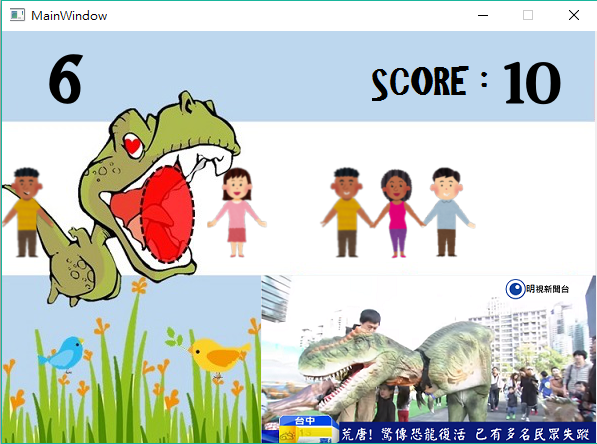
【Screen shot】

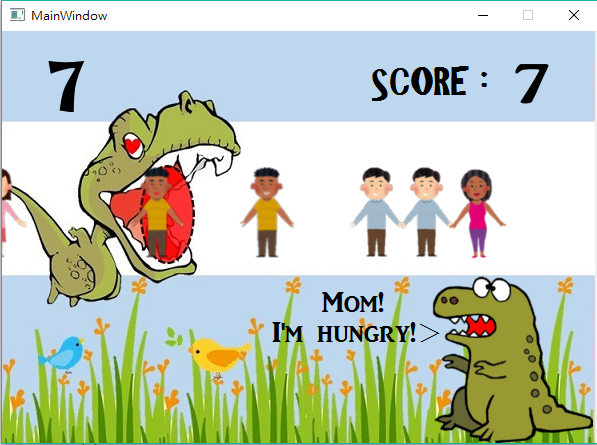
(圖1) 

(圖2) 

(圖3) 

(圖4) 

(圖5) 

(圖6) 

(圖7) 

(圖8) 

【UML】

|  |
| --- |
| Start |
| \*ui : Ui::Start  \*newgame : MainWindow |
| <<constructor>>+Start(QWidget\*)  +stopgame()  -on\_pushButton\_clicked()  -on\_pushButton\_2\_clicked() |

|  |
| --- |
| Dialog |
| - \*ui:Ui::Dialog  - &w:Start  - &newgame : MainWindow |
| <<constructor>> + Dialog(\*parent: QWidget ,&start: Start,&newgame:MainWindow)  - on\_pushButton\_2\_clicked()  - on\_pushButton\_clicked()  *- closeEvent*(\*event: QCloseEvent) |

|  |
| --- |
| MainWinow |
| -cnt:int  -score:int  -\*ui : Ui::MainWindow  -&w:Start  -x1:int  -x2:int  -x3:int  -x4:int  -x5:int  -x6:int  -x7:int  -x8:int  -y:int  -av1: bool  -av2: bool  -av3: bool  -av4: bool  -av5: bool  -av6: bool  -av7: bool  -av8: bool  -\* dialog:Dialog |
| <<constructor>> + MainWindow(\*parent: QWidget , &start: Start)  + *closeEvent*(\*event: QCloseEvent)  -timer\_timeout();  -movep1()  -movep2()  -movep3()  -movep4()  -movep5()  -movep6()  -movep7()  -movep8()  -newP()  -low()  -yummyDisappear()  - oopsDisappear()  -*keyPressEvent*(\* event: QKeyEvent) |

【Program Architecture】

1. main.cpp中產生一個Start的物件w，並執行Start的ctor，w.show()會讓w這個開始介面出現。
2. Start的ctor中會讓ui出現，ui上有兩個按鈕，按下start這個按鈕會執行on\_pushButton\_clicked()會生成一個type是Mainwiondow的newgame，並且讓newgame show出來；按下exit這個按鈕，會執行on\_pushButton\_2\_clicked()，這個遊戲介面就會關掉。
3. 產生newgame時，Mainwindow的ctor中：

* 首先生成timer\_count，setInterval為1000，connect讓timer\_count每過一秒，就發出一個signal，讓timer\_timeout()執行一次，timer\_timeout()每執行一次，cnt少1，因此每過一秒cnt就會少1，當cnt為0時，所有timer都會停止，cnt回歸成30，會生成一個type是Dialog的dialog出來。
* 生成timer\_new，每過(rand()%2)\*800+200)個千分之一秒會執行一次newP()，在newP()中，會隨機進入SWITCH的0~7八種情況，若進入1，av1為true的話，讓av1變成false，生成timer\_move1，每過0.08秒就會執行一次movep1()，讓p1往左移動10個像素，直到移動到x<-59，timer\_move1就停止，讓p1回到原本在畫面外的x位置，delete掉timer\_movep1，並讓av1變成true，這樣下次newp()時就能夠再次讓p1移動。這樣隨機時間執行newp()，以此類推其他七種情況。如此一來就會不斷有隨機的人物從畫面右邊移動到左邊。
* 生成timer\_20，過20秒後就會執行low()，在low()中，若score低於8，baby等三個label就會出現在畫面上。
* 生成timer\_yummy跟timer\_oops，每過0.5秒，就會執行yummyDisappear()和oopsDisappear()把yummy跟oops的label移動到畫面外一次，這裡是為了配合後面的keyPressEvent中的功能。

1. 30秒鐘若有按按鍵就會執行一次keyPressEvent(QKeyEvent \* event)：

* 若按下A的時候，代表A鍵的P1或是P5剛好在指定的位置內，會做跟X<-59時一樣的事情，並且讓yummy這個label出現在畫面中，若timer\_yummy是已經start的狀態，先stop後，才再start一次，過0.5秒後，就會執行yummyDisappear()把yummy移動到畫面外，讓score加1，如果score大於等於8，就讓news這個label出現在畫面上。
* 若按下A時，p1或p5不在指定的範圍內，若timer\_oops已經start，先stop再start一次，讓oops這個label出現在畫面上。
* 其他三個按鍵的狀況也以此類推。

1. 若按下Mainwindow右上角的叉叉鍵，執行closeEvent(QCloseEvent \*event)，其中會執行stopgame()，delete掉newgame。
2. 30秒結束後生成的dialog，其ctor中會顯示剛剛那場newgame的score，並且根據是否達到８分以上來出現兩種不同的圖案跟字。若按下play again的按鈕會執行on\_pushButton\_clicked()，生成新的一個Start calss的n，並且執行stopgame()來刪掉舊的newgame，若按下exit會執行closeEvent(QCloseEvent \*event)，直接執行newgame來刪掉newgame。