pd2-Angrybird Report

H34016144 許欣穎

【How to play】

* 每次遊戲開始都有三隻豬（圖１），總共有四種鳥各出現一次，點住鳥往後拉放開候鳥就會往前飛，石頭或鳥打到豬都會得一分，豬就會消失（圖２）。
* 按Ｓ鍵即可換下一隻鳥，飛行中按下Ａ鍵，就會出現那隻鳥的功能

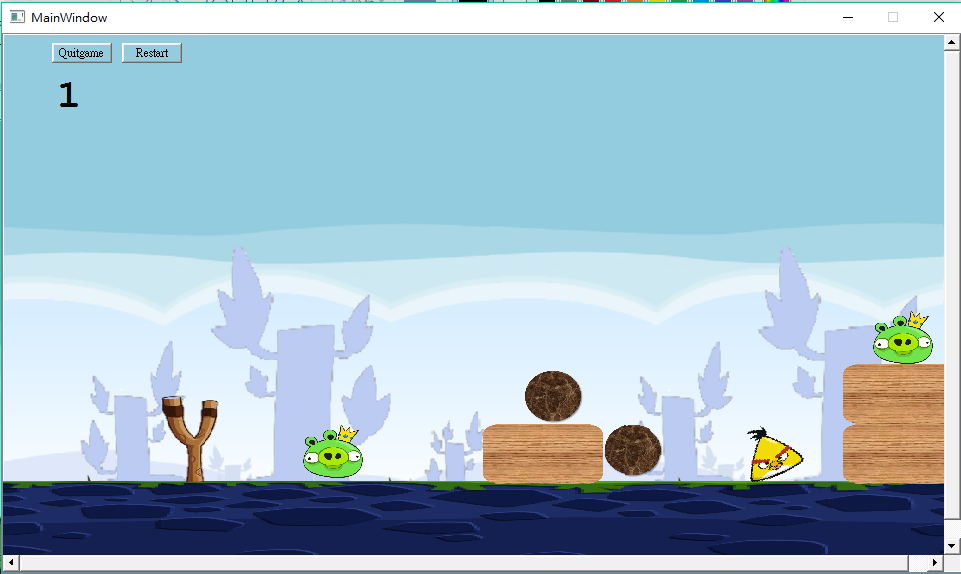
黃鳥：垂直降落（圖３）／紅鳥：往右下飛（圖４）

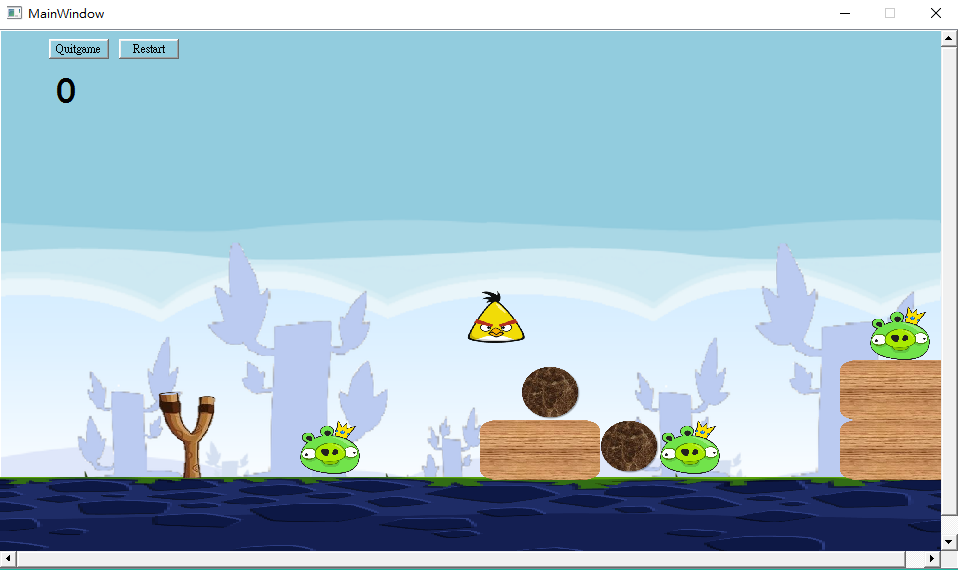
藍鳥：出現另外兩隻藍鳥（圖５）／綠鳥：往回飛（圖６）

* 按下Restart鍵整個遊戲重新開始，關掉視窗或按下Quitgame就會結束

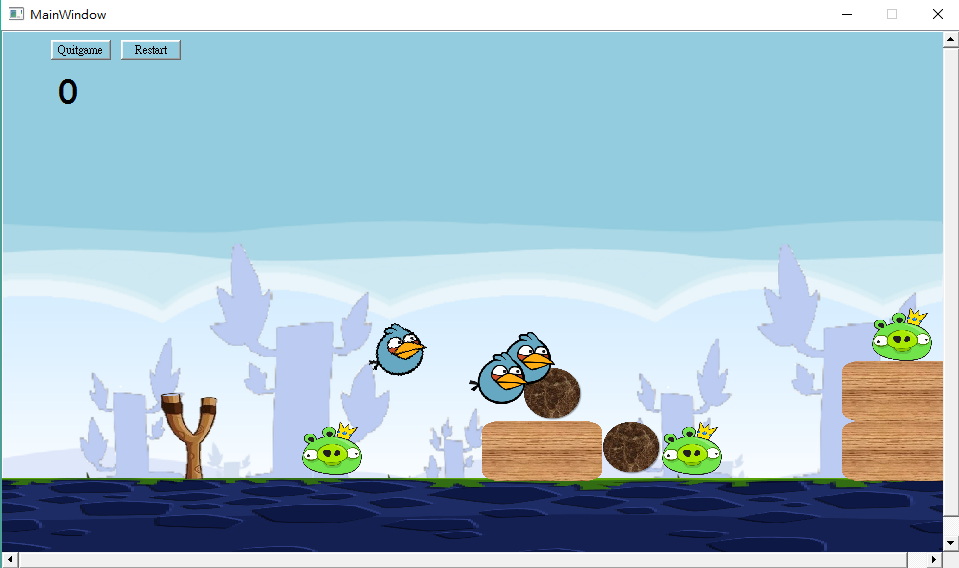
【Screen shot】

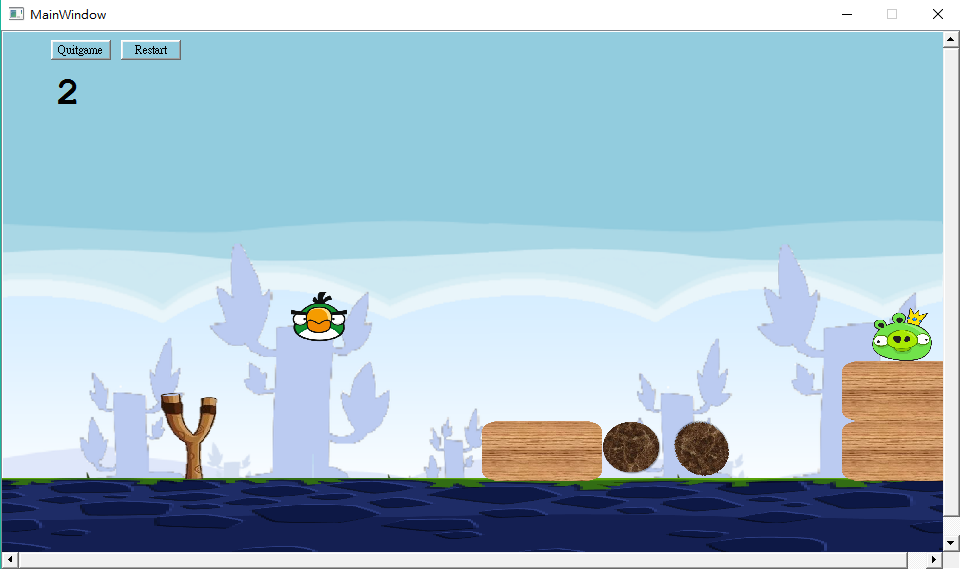
(圖1) 

(圖2) 

(圖3) 

(圖4)

(圖5) 

(圖6) 

【UML】

|  |
| --- |
| GameItem |
| #\*g\_body:b2Body  # g\_pixmap : QGraphicsPixmapItem  # g\_size : QSizeF  #\*g\_world : b2World  # g\_worldsize, g\_windowsize : static QSizeF |
| <<constructor>> + GameItem(\*world: b2World)  + ~*GameItem*()  + setGlobalSize(worldsize:QSizeF, windowsize:QSizeF):static void  <<slot>> + paint() |

|  |
| --- |
| Bird |
| +bluebirdpress : bool  -canmove : bool  - func : bool  - flied : bool  - \* bluebird1 : Bird  - \* bluebird2 : Bird |
| <<constructor>>+Bird(id:int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  +setLinearVelocity(velocity : b2Vec2)  + *press*()  + getPositionX():doubla  + getPositionY():double  + deleteblue() |

|  |
| --- |
| Stone |
|  |
| <<constructor>> +Stone(id:int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  + *press*() |

|  |
| --- |
| Pig |
| +m\_contacting : bool |
| <<constructor>> +Stone(id: int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  + *press*()  +startContact() |

|  |
| --- |
| redbird |
|  |
| <<constructor>> + redbird (id : int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  + *press*() |

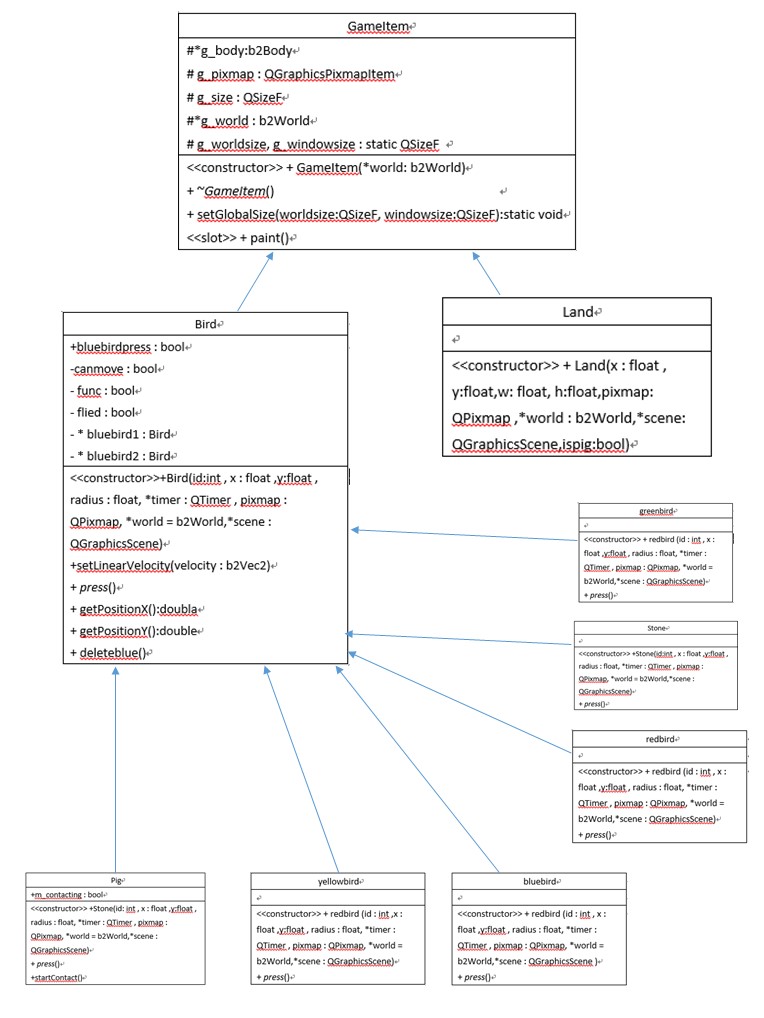
|  |
| --- |
| yellowbird |
|  |
| <<constructor>> + redbird (id : int ,x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  + *press*() |

|  |
| --- |
| greenbird |
|  |
| <<constructor>> + redbird (id : int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene)  + *press*() |

|  |
| --- |
| bluebird |
|  |
| <<constructor>> + redbird (id : int , x : float ,y:float , radius : float, \*timer : QTimer , pixmap : QPixmap, \*world = b2World,\*scene : QGraphicsScene )  + *press*() |

|  |
| --- |
| Land |
|  |
| <<constructor>> + Land(x : float , y:float,w: float, h:float,pixmap: QPixmap ,\*world : b2World,\*scene: QGraphicsScene,ispig:bool) |

|  |
| --- |
| MainWindow |
| -\*ui : Ui::MainWindow  -\*scene:QGraphicsScene  -\*world:b2World  -\*itemList : QList<GameItem\*>  -timer : QTimer  -\*thisbird : Bird  -count : int  -\*pig1 : Pig  -\*pig2 : Pig  -\*pig3 : Pig  -\*myContactListenerInstance : MyContactListener  -\* score : QGraphicsTextItem  -check1 : bool  -check2 : bool;  -check3 : bool  -\*quit : QPushButton  -\*restart : QPushButton  -\* stone1:Stone  -\* stone2:Stone  -p : QPoint  -x1,y1,x2,y2,bluex1,bluey1:double  - checkblue : bool |
| <<constructor>> + MainWindow(\*parent : QWidget);  +~*MainWindow*();  +*showEvent*(QShowEvent \*)  +*eventFilter*(QObject \*,\*event : QEvent):bool  +*keyPressEvent*(QKeyEvent\* event);  +*closeEvent*(QCloseEvent \*)  +deletepig()  <<slot>>+newbird()  <<signal>> quitGame()  <<slot>>-tick()  <<slot>>-QUITSLOT()  <<slot>>-quitgame()  <<slot>>-restartgame()  - GetFixtureA():b2Fixture\*  - GetFixtureB():b2Fixture\* |
| MyContactListener |
| +score:int  -\*pig1:Pig  -\*pig2:Pig  -\*pig3 : Pig |
| <<constructor>> + MyContactListener(\*pig1:Pig,\*pig2 : Pig\*,\*pig3:Pig)  +*BeginContact*(\*contact : b2Contact) |



【Program Architecture】

* 將class Land的各個障礙物以及地板以及彈弓new出來
* newbird()：用thisbird指向鳥，讓四種鳥輪流被new出來，new下一隻鳥出來前會先delete上一隻鳥。
* 將繼承自class Bird的石頭跟豬new出來
* new一個MyContactListener，隨時監控是否有碰撞發生，若有碰撞發生會執行BeginContact，取出兩個碰撞東西的userdata，若兩者的ispig都是true，表示是石頭或鳥或豬互撞，就繼續看碰撞的其中一個是否是某一隻豬，就讓該隻豬執行startContact()，m\_contacting會變成true，score++
* 設定分數出現的地方跟字體跟一開始設定為0
* 設定restart跟quitgame button，按下去會發出release()的signal，分別跟quitgame()跟restartgame() 連結
* 如果是還沒飛過的鳥，按下按鈕會讓canmove變成true，此時移動滑鼠的話，先取得滑鼠的QT座標，轉換成BOX2D的座標，在若限定的範圍內，鳥就會跟著滑鼠一起移動。
* 放開滑鼠會根據當下取得鳥的位置和彈弓的距離差來決定起飛的加速度，func會變為true，在開始飛行後按下A鍵的話，會執行那隻鳥的功能；若按下Ｓ鍵，執行newbird()會換成下一隻鳥。
* tick()每過1/60秒會執行一次，其中會不斷根據當下的分數來改變顯示分數；如果豬的m\_contacting變成true，並根據check來判斷有沒有被碰過，被碰到就會delete掉那隻豬的pointer；如果bluebirdpress是true表示藍鳥飛行過程按下A鍵執行功能，會new出兩隻新的藍鳥，從當下的位置以不同加速度開始飛行
* 如果按下restartgame按鈕，會先判斷豬有沒有被delete過，如果有就new新的豬，然後把pig傳參數去設定新的MyContactListener，接著把thisbird刪掉，重新newbird()，stone也會delete掉重新生成。