UML:

|  |
| --- |
| **MainWindow** |
| -scene: QGraphicsScene \*  -world: b2World \*  -itemList : QList<GameItem \*>  -timer : QTimer  -exi : QPushButton \*  -res : QushButton \*  -birdie : Bird \*  -birdie\_b : Bird2 \*  -birdie\_w : Bird3 \*  -birdie\_g : Bird4 \*  -pig : Pig \*  -building : Building \*  -building2 : Building \*  -building3 : Building \*  -building4 : Building \*  +press = 1:int  +fx : int  +fy : int  +ax : int  +ay : int  +score : bool  +sc : QLabel \*  +grade = 0 : int |
| +<<explicit>> MainWindow(parent:QWidget \*=0) : void  +~MainWindow()  +showEvent(QShowEvent\*) : void  +eventFilter(QObject \*, event:QEvent) : bool  +closeEvent(QCloseEvent\*) : void  +calculate\_score():void  +<<signals>> quitGame() :void  -<<slots>> tick() : void  -<<slots>> QUITSLOT() : void  -<<slots>>END() : void  -<<slots>>RESTART():void |

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

|  |
| --- |
| **Bird** |
|  |
| +Bird(x:float,y:float,radius:float,timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*)  +setLinearVelocity(velocity:b2Vec2) : void  +skill() : virtual void |

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

|  |
| --- |
| **Pig** |
|  |
| +Pig(x:float,y:float,radius:float,timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*) |

|  |
| --- |
| **Building** |
|  |
| +Building(x:float,y:float,w:float,h:float:timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*) |

|  |
| --- |
| **Bkg** |
|  |
| +Bkg(x:float,y:float,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*) |

|  |
| --- |
| **Bird2** |
|  |
| +Bird2(x:float,y:float,radius:float,timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*)  +skill():virtual void |

|  |
| --- |
| **Bird3** |
|  |
| +Bird3(x:float,y:float,radius:float,timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*)  +skill():virtual void |

|  |
| --- |
| **Bird4** |
|  |
| +Bird4(x:float,y:float,radius:float,timer:QTimer \*,pixmap:QPixmap,world:b2World \*,scene:QGraphicsScene \*)  +skill():virtual void |

Screen shot:





How to play:

1. 滑鼠按住往任一方向拉，即可射出
2. 紅色為普通鳥
3. 藍色會往前方射出
4. 白色會向下墜落
5. 綠色會瞬間變大
6. 按下綠色鍵為重新開始
7. 按下藍色鍵為結束遊戲

Program architecture:

先以助教的source code為雛型進行修改

一開始先做障礙物的部分，跟鳥最大的不同是需要用到setAsBox還有world與scene之間的轉換(scaled)

再來開始做鳥的部分，我是用globalX(), globalY()來取得滑鼠”按下與放開”的位置，再將兩次的位置坐相加減，來得到發射的效果

其他鳥的效果主要是利用多型的方法在Bird.h裡宣告一virtual void skill()，然後

在Bird2, Bird3裡override這funtion即可，其他的變數便不用再次宣告

Bird4的效果比較不一樣，當呼叫skill()時，改動原本的scaled與QSIZE就可讓鳥變大，所影響的範圍也會隨之變大

換鳥&發射鳥&啟動鳥技能的判定是利用press的次數，當我release一次press就會+2次，然後利用if判斷每兩次過後，所要使用的功能

最後是判斷計分，如果原本不動的敵人，有了angular velocity或linear velocity，就判定分數上升