

一、程式原理

1. Boxbrick.h:

BoxBrick(int x, int y): x 和 y 分別代表生成位置，之後再指定 BoxbricPixmap 的圖片、以及其大小

boundingRect(): 回傳該物件的邊界

getY():回傳圖片的 Y 座標

getX():回傳圖片的 X 座標

2. Brokenbrick.h:

BrokenBrick(int x, int y): x 和 y 分別代表生成位置，之後再指定 BrobricPixmap 的圖片以及其大小

3. Bullet.h:

Bullet(qreal x, qreal y, qreal mouseX, qreal mouseY): mouseX(Y)為滑鼠點擊時的 X(Y)座標，用來設定子彈的 x 方向(vx)以及 y 方向(vy)飛行速度。

x 和 y 分別代表生成位置，之後再指定 bulletPixmap 的圖片以及其大小。

Timer 連接函數 move，每 20ms 會呼叫 1 次

move(): 隨時更新子彈的位置，以及偵測和 toxic mushroom、flower、所有的磚塊以及水管之間的碰撞，碰到之後就刪除子彈

4. Coin.h:

Coin(int x, int y): x 和 y 分別代表生成位置，之後再指定 coinPixmap 的圖片以及其大小，Timer 連接函數 touchmario，每 10ms 會呼叫 1 次

touchmario(): 偵測和 mario 之間的碰撞，碰到之後執行 getCoin()

5. Fire_flower.h:

Fire_flower(int x, int y): x 和 y 分別代表生成位置，之後再指定 fireflowerPixmap 的圖片以及其大小

6. Flag.h:

Flag(int x, int y): x 和 y 分別代表生成位置，之後再指定 flagPixmap 的圖片以及其大小，Timer 連接函數 fall，然後再印出他的 x 座標

fall(): 當呼叫此函式時，啟動計時器每 20ms 執行一次，讓 flag 以同樣的速度下降(vy)，直到 Y 座標大於 460

7. Flag_pole.h:

Flag_pole(int x, int y): x 和 y 分別代表生成位置，之後再指定 flagpolePixmap 的圖片以及其大小

8. Floor.h:沒有特別的功能

9. Flower.h:

Flower(int x, int y): x 和 y 分別代表生成位置，之後再指定 FlowerPixmap 的圖片以及其大小。Timer 連接函數 move，每 20ms 會呼叫 1 次。timerID 用來記錄 timerEvent 執行的頻率

timerEvent(): 當經過了 timerID 設定的週期且 timerID 不為 0 時就會將 time1 遞增，直到 time1 為 40 時歸零

move(): 偵測和 mario 的碰撞，當碰到且此時 flower 還在畫面上時會呼叫 touchtoxic()。在動畫的設定中每 8 秒為 1 週期，在這 8 秒鐘 flower 都會持續開關，其中

0 ~ 1.2 秒:穩定從水管出現

1.2 ~ 3.6 秒:在最高點

3.6 ~ 4.8 秒:穩定下降

4.8 ~ 8 秒:隱藏

10. Gamebase.h:

- 呼叫 constructor 來設置初始畫面，設定 timer 和基本參數
- gameStart、switchToUndergroundScene:將場景地圖生成，並把對應的 timer 連結到 updateGame 中。
- updateGame:判斷瑪莉歐入洞、呼叫 moveMario 並檢測其回傳的狀態。有存活、死亡、進入水管、無敵等，並撥放對應的音樂
- 滑鼠左鍵點擊時判斷是否可以發射子彈，並尋找對應的位置、速度發射。

11. Main: 呼叫 Mainwindow

12. Mainwindow.h:

呼叫 gamebase.cpp

13. Mario.h:

time:碰到 toxic mushroom 的間隔時間

time2:碰到 super star 的無敵時間

star: 是否為無敵狀態

timerEvent(): 當經過了 timerID 設定的週期且 timerID 不為 0 時就會將 time1 遞減直到 0，time2 遞增

touchtoxic、touchsuper():

- 根據指令加減 hp、變更瑪莉歐大小
- 在吃到對應物件時直接設定新的移動速度

moveMario():

- 移動同時改變 posX、posY 與呼叫 moveby()函數，讓瑪莉歐的刷新率變高，更流暢
- 上下碰撞的判斷為碰到時碰撞箱的上下位置，左右碰撞則檢查物件中點與瑪莉歐的碰撞箱頂、底部相對位置，若頭尾分別在物件中點的上及下，則判斷為左右阻擋，此時停下瑪莉歐往那個方向的移動。
- 上下移動每次都先更新垂直速度、垂直位置，再判斷若碰撞腳下則將 posY 的位置微調，並重新設定狀態。
- 碰到問號磚時，根據磚塊的編號生成對應的道具，這是為了吻合遊戲整體剛好 40 金幣。
- 碰到道具也要有相應的互動。直接呼叫瑪莉歐中對應的函數。

changeImage():

- 使用 file scope enum 來判斷要切往哪個狀態:大/小的面向左/右的/站/跑/跳。再切換之前優先判斷是否死亡。
- 在狀態改變時就呼叫

marioGoDie():

- 無條件殺死瑪莉歐，換成死亡狀態的圖片。
- 配合檢測遊戲失敗、掉入洞穴等非自然死亡而設定。

14. Mediaplayer.h:

組共 11 個音效及音樂，分別是

gameOver: mario 的 hp 為 0 時觸發；ground: 按下 start 按鈕後，載入遊戲時觸發；underground:載入隱藏場景時觸發；superstar:mario 吃到無敵星星後觸發，4 秒後停止；levelcomplete:遊戲結束，mario 到達終點時觸發；jumpingsmall 和 jumpingbig: 分別是 small mario 和 big mario 的跳躍音效；powerup: mario 吃到超級蘑菇時觸發；fireball:mario 發射火焰彈時觸發；coin: mario 吃到金幣時觸發；pipe: mario 進入 pipe 通往隱藏場景時觸發。

15. Normalbrick.h:

16. Stonebrick.h:

StoneBrick(int x, int y):

17. Super_mushroom:

18. Super_mushroom(int x, int y): x 和 y 分別代表生成位置，之後再指定

mushroomPixmap 的圖片以及其大小，Timer 連接函數 move，每 20ms 會呼叫 1 次

move(): 隨時更新蘑菇的位置，以及偵測和所有的磚塊、子彈、水管以及 mario 之間的碰撞，水平碰到水管以及磚塊時，x 方向速度乘以-1，其他則會固定在該物件上面。當碰到 mario 時則會消失並執行 touchsuper()

19. Superstar.h: SuperStar(int x, int y): x 和 y 分別代表生成位置，之後再指定 superstarPixmap 的圖片以及其大小

20. Toxic_mushroom.h:

Toxic_mushroom(int x, int y): x 和 y 分別代表生成位置，之後再指定 toxicmushroomPixmap 的圖片以及其大小。Timer 連接函數 move，每 30ms 會呼叫 1 次。timerID 用來記錄 timerEvent 執行的頻率

timerEvent(): 當經過了 timerID 設定的週期且 timerID 不為 0 時就會將 time2 +1，time1 遞減直到歸零

time1: 用來計時碰到 mario 後 0.5 秒刪除

time2: 動畫的時鐘

disappear(): 將圖片換成扁的型態且 0.5 秒後刪除圖片

move(): 隨時更新蘑菇的位置，以及偵測和所有的磚塊、子彈、水管以及 mario 之間的碰撞，水平碰到水管以及磚塊時，x 方向速度乘以-1，其他則會固定在該物件上面。如果 mario 碰到且此時 mario 垂直速度向下(mario 踩到毒蘑菇的頭部)或是有拿到 Super star 時，就會執行 disappear()，如果 disappear0 為 false 時，就會執行 touchtoxic()。毒蘑菇碰到子彈時會一起消失。每 0.25 秒會往前走一步直到死亡。

21. Turtle.h:

Turtle(int x, int y): x 和 y 分別代表生成位置，之後再指定 TurtlePixmapPixmap 的圖片。Timer 連接函數 move，每 20ms 會呼叫 1 次。timerID 用來記錄 timerEvent 呼叫的頻率。

time1: 動畫的時鐘

time2: 第 1 次踩到第 2 次踩的最大間隔時間(3 秒)

spin: 是否踩到第 2 次

shell: 烏龜是否為龜殼狀態

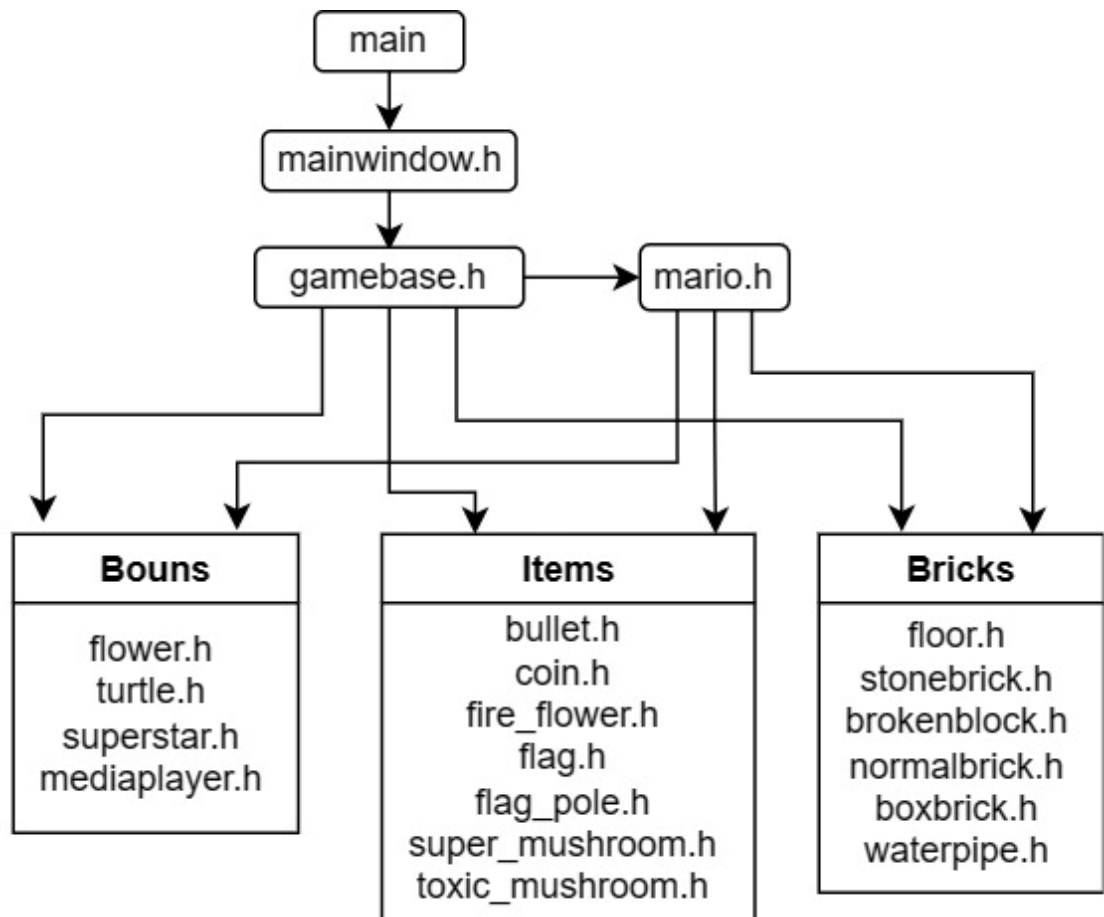
timerEvent(): 當經過了 timerID 設定的週期且 timerID 不為 0 時就會將 time1 遞增，當 mario 第 1 次踩到時 time2 倒計時 3 秒

move(): 隨時更新烏龜的位置，以及偵測和所有的磚塊、子彈、水管以及 mario 之間的碰撞，當 mario 踩到時會進入龜殼狀態，圖片會換成龜殼。在 3 秒內再次踩到時，龜殼就會開始旋轉。如果 mario 不是踩到的話就會扣血(和碰到毒蘑菇一樣的效果)，和其他物件的碰撞效果跟毒蘑菇一樣。每 0.4 秒會往前走一步直到變成龜殼。龜殼狀態下速度會增加 5 倍且每 0.2 秒自轉 1 圈。

22. Waterpipe.h:

WaterPipe(int x, int y): x 和 y 分別代表生成位置，之後再指定 pipePixmap 的图片以及其大小

二、架構



三、困難的地方

1. Mario 的上下碰撞比較好解決，左右碰撞再利用 pixmap function 時一直無法正確判斷 mario 的位置，導致左右碰撞一直無法運作。

解決方式：

我們發現是因為我們沒有使用內建的座標導致。所以 x 位置不會是真實的 x 座標。因此我們使用碰撞箱的上下來判斷，用這種方式定義當瑪莉

歐與物件重疊的部分過多時為左右碰撞來解決。

2. 製作場景音樂時，若觸發條件放在不同的 class 會導致音樂無法正確關閉並切換下一首音樂，造成聲音重疊。

解決方式:將主題音樂全部放到 gamebase 中播放，將判斷播甚麼背景音樂的邏輯放到 gamebase 中。

3. 切換至隱藏場景時，我們會把舊的場景刷掉，這樣導致金幣無法正確判斷是否被吃掉

解決方式:

1. 在金幣的檔案中新增靜態變數儲存金幣生成的數量、將金幣編號並將該編號的指標加入對應的矩陣中。
2. 如果瑪莉歐沒有吃到金幣，則把所有金幣生成；瑪莉歐吃到金幣時，將陣列中對應編號的指標設為空，這樣在第二次生成場景時，就可以跳過為空的金幣位置。
3. 跳躍射擊子彈時常常卡死

解決方法:讓瑪莉歐只在地面時發射子彈。

四、分工表

廖奕喆(E14104080): boxbrick 和其他 items 的特性以及 mario 碰到後的反應

楊子毅(AN4104016): 音效、音樂，gamebase 地圖、各種 brick 的建置

許彥喬(E24119029): mario、gamebase 等物件的程式架構、debug