|  |  |  |
| --- | --- | --- |
| Team No: 27 | Team Name: 第二十七組 | |
| Project Title: Flappy Bird Go | | |
| Name: 陳凱揚 | | ID: 108032053 |
| Name: 謝承恩 | | ID: 109062174 |

Project Description:

1. **概念圖、功能描述與使用到的I/O Devices或額外的機構設計**
2. 概念簡述

我們這組目前暫定的主題為flappy bird，首先可以選擇遊戲模式（單人、雙人）。

* 單人模式：以鍵盤上下鍵控制鳥的飛行高度（計畫將來改成隨著時間鳥會不斷下墜），讓鳥可以順利通過水管，若碰觸到水管則扣1命（暫定共2命），並以鍵盤左右鍵增減速度（暫定速度1~3），每通過1個水管就會增加分數（速度1加1分，速度2加2分，以此類推），共20個水管（暫定），全部通過即可通關（暫定為1關）。
* 雙人模式：與單人模式大致相同，2位玩家分別以鍵盤控制角色移動，同樣可以增減速度來驚嚇對方，當通過所有水管或是有玩家死亡則遊戲結束，遊戲結束後會回到主畫面重新開始。

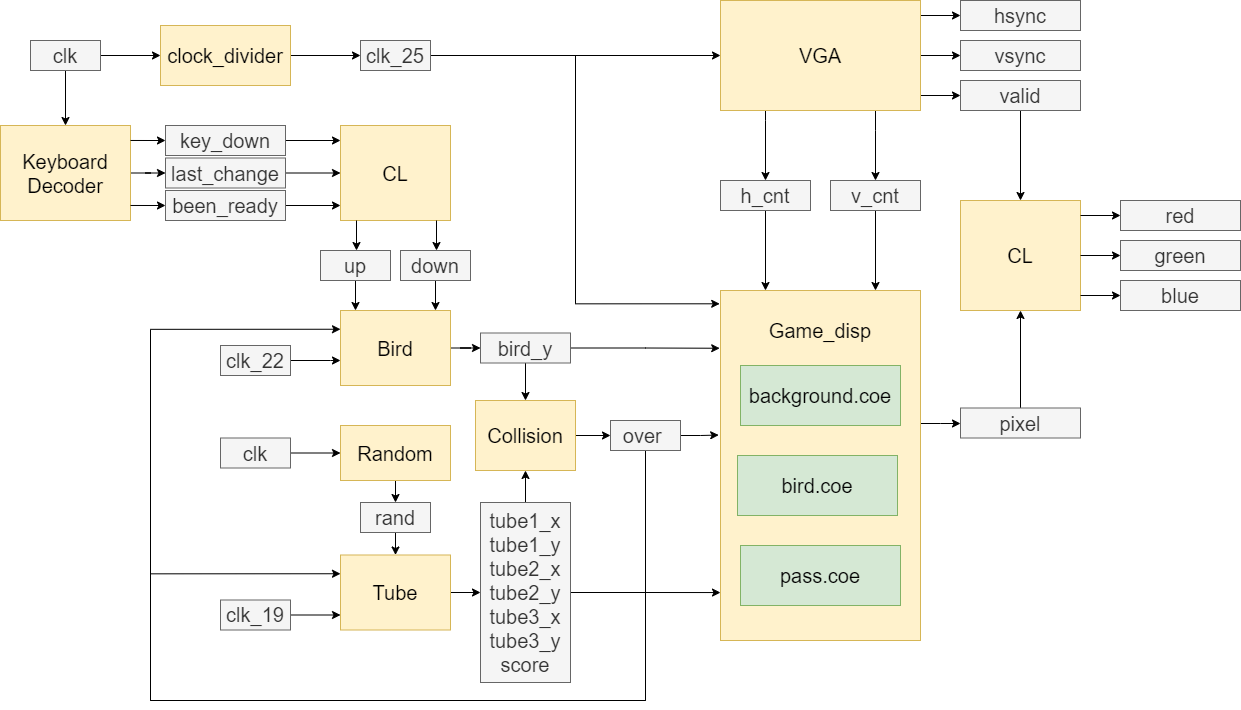


1. I/O Device

* Keyboard：玩家能以鍵盤控制鳥的飛行高度（上下、WS）和前進速度（左右、AD）。
* Switch：用來暫停或繼續。
* 7-segment：用來顯示玩家目前的分數。
* LED：代表玩家的生命數。
* Push Buttom：用來選擇並確認要哪種模式。
* VGA：顯示遊戲畫面。
* Audio：碰撞音效、通關音效、背景音樂。
* 馬達、其他IO：規劃中。

1. Block Diagram

以下為簡略版的block diagram示意圖。

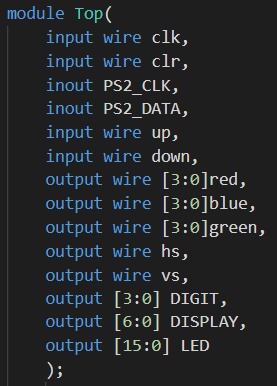
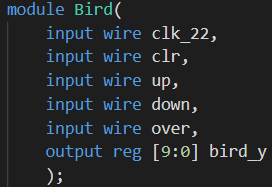
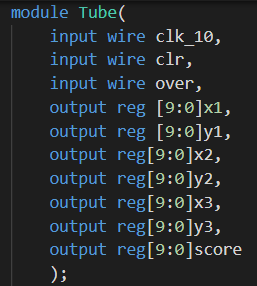


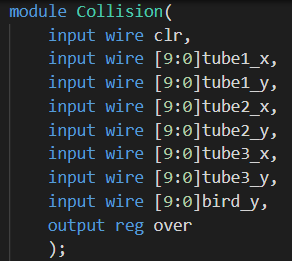
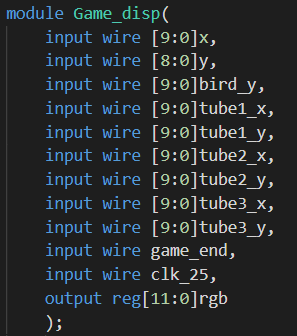
1. 功能描述

目前功能：

* Module Top：目前Top module會使用到的I/O包括keyboard、push bottom、VGA display、7-segemnt、LED。
* Module Bird：用於更新鳥的位置，其中up、down為鍵盤的上下訊號，而鳥的x座標為固定的，因此只會輸出鳥的y座標。
* Module Tube：用於更新水管的位置，並以module Random產出亂數決定新的水管座標，目前預計一個畫面會出現三根水管，因此會輸出三組座標位置。此外，此module也會計算分數，輸出score回module Top裡，並以module SevenSegment顯示出來。
* Module Collision：用於判斷鳥與水管是否發生碰撞，因此會傳入鳥與三根水管的座標，並輸出over訊號，當over為1時代表發生碰撞，遊戲就會結束。
* Module Game\_disp：用於控制整個螢幕輸出的內容，包括鳥、水管、背景圖片、通關畫面，其中會輸入鳥與三根水管的座標，在此module內部會判斷目前位置應該輸出哪種物件，最後送出12bits的pixel回module Top裡，並以module VGA顯示出來。

以下為各個module的IO。

待新增功能：

* 鳥會不斷下墜：目前的設計是鳥的水平高度完全由玩家用上下鍵控制，我們希望能夠做出地心引力的感覺，讓鳥會不斷往下墜，同時也增加遊戲的困難度。
* 用LED顯示生命：讓玩家可以擁有多條命，生命數用LED顯示出，但考量到玩家玩遊戲時必須一直盯著螢幕，因此若鳥撞到牆壁會輸出音效，提示玩家少一條命。
* 障礙物從對面飛過來：有在規劃讓螢幕的右側每隔一段時間就會有障礙物飛出來，若鳥被障礙物砸到也算遊戲失敗。
* 改變飛行速度：可以用鍵盤控制增減速度，增加遊戲難度，難度越高，增加的分數也越快。
* 增加雙人模式：使兩名玩家可以進行比賽，先死的就輸了。
* 增加音效及背景音樂：目前計畫發生碰撞時會有音效表示失去一條命，通關或結束時也能有音效表示，另外也想嘗試加入背景音樂。
* 吃金幣：路徑上加入金幣，吃了可以增加分數。
* 遊戲的流程控制：以push bottom選擇及決定模式，在遊戲中能以switch控制遊戲的暫停與繼續，遊戲結束後可以選擇重新開始或是離開。

**2. 規劃工作項目、進度與分工**

首先，因為我們已經將block diagram上的module都大致寫出來了（基本功能），因此接下來會先測試這些功能是否順利運作，接著再一步步新增我們想加上的新功能，也確保每次添加新功能後，舊功能依然正常運作。而以下為我們暫訂的進度與分工（距離Demo還有五週）。

|  |  |  |
| --- | --- | --- |
|  | 陳凱揚 | 謝承恩 |
| 第一週 | 確認Top module的完整架構，將可能用到的I/O module都先接好線，清楚分工每個module的功能。 | 測試角色的移動、各種物品、背景等要素如何顯示，包括memory的讀取。 |
| 第二週 | 加入遊戲控制流程，設計state diagram並實作，並增加不同的模式（單人、雙人），以push bottom控制。 | 畫面和角色的移動（邊界問題、水管的隨機）、碰撞的真實性（避免還不夠接近或是太接近才發生碰撞的問題）。 |
| 第三週 | 加入生命、分數、金幣、障礙物等增加遊戲體驗，以LED、7-segment控制。 | 改進角色的控制（如鳥會不斷下墜）、以及玩家能夠以鍵盤控制前進速度。 |
| 第四週 | 控制不同音效、背景音樂等同時出聲時如何配合。 | 決定哪些時間點出現音效並實作（如碰撞、通關等）。 |
| 第五週 | 反覆測試各種情況並決定最終成果（尋找是否有未發現的bug）。 | 反覆測試各種情況並決定最終成果（尋找是否有未發現的bug）。 |

**3. 可能遭遇之困難與預期解決方法或備案**

1. 螢幕顯示：同時在螢幕上畫出多種物品，且物品會移動，對我們是一個挑戰，因為目前尚未嘗試過畫出會移動的小物品。預計採用的方式為分別計算每個物品的座標，其餘部分才輸出背景圖片，這部分可能需要實際操作才能發現更多小問題。
2. 隨機產生水管：隨機的部分目前使採用xor和left shift來產生新水管的座標，但還需要配合之後的螢幕輸出才能驗證是否足夠真實、有隨機的效果，否則必須再另外想出演算法來達成隨機。
3. 碰撞判斷：有些物品是不規則形狀，若單純使用矩形來判斷碰撞，會有很多不真實的碰撞，因此必須加強碰撞的判斷，並實際測試，不斷try and error來達成最佳的碰撞效果。
4. 各種音效和音樂的配合：由於目前對聲音的操作還不夠熟悉，只想到多種聲音同時出現時，必須互相配合，才能和諧的發聲，這部分也需要再研究。
5. Memory空間不夠：由於我們的FPGA上只能存1.9張320\*240像素的圖片，因此在開場、背景圖片、結束時沒辦法各存不同的圖片來顯示，目前想到的辦法為除了背景圖片必須存之外，其他圖片不需要佔到整個螢幕大小，只需在螢幕中央顯示對應的方塊，其餘部分顯示背景圖片，因此可以存更小的圖片來節省空間。假如還是不夠的話，再來嘗試使用兩個FPGA。
6. 各種意料之外的問題：由於很多小問題都是在實作時才會發現，因此我們在設計時，也會同步測試，不僅能夠及時發現問題，有足夠的時間debug，也對自己的設計、各個流程更為清楚。