

一、系統功能/架構/摘要/開發平台

(一) 遊戲名稱：見縫插針 (Core Ball)

(二) 如何遊玩該遊戲？

1. 將小球放置到大球上面
2. 在放置過程中不可以接觸到已經在上面的小球
3. 通過所有關卡為獲勝
4. 可以使用愛心獲得重新挑戰關卡

(三) 功能

1. 放置小球方式有兩種：點擊畫布、按下空白鍵
2. 可以點擊旁邊關卡難度決定要挑戰的關卡
 - (1) 簡單：共 5 關，僅有一個方向變動
 - (2) 中等：共 10 關，會雙向旋轉，速度中等
 - (3) 困難：共 5 關，會雙向旋轉，速度極快
3. 可以使用音效/BGM 的關閉和開啟按鈕
4. 可以點擊 Reset game 按鈕重置當前關卡
5. 如果未通關可以使用 Use Heart 重啟失敗關卡
6. 使用 Restart 按鈕回到首頁，重新挑選難度

(四) 開發

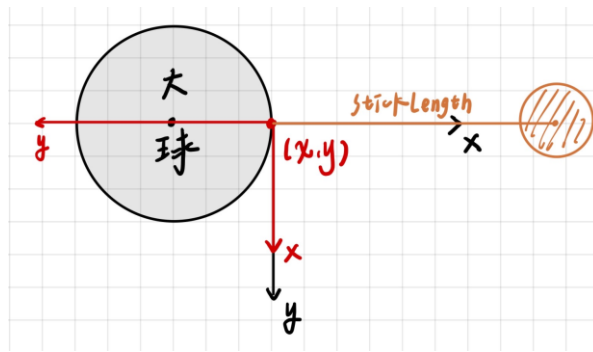
1. 開發工具：vscode
2. 執行環境：瀏覽器 Chrome
3. 作業系統：Windows
4. 系統架構：
 - (1) index.html
 - ✓ 包含基本的 HTML 結構。
 - ✓ 引用 style.css 和 script.js 文件。
 - ✓ 定義遊戲的畫布或主要容器。
 - (2) style.css
 - ✓ 用於設定遊戲的樣式和佈局。
 - ✓ 包含畫布或主要容器的 CSS 設定。
 - (3) script.js
 - ✓ 包含遊戲的主要邏輯。
 - ✓ 使用 JavaScript 控制遊戲的行為和互動。
 - (4) material 資料夾：儲存遊戲所需的資源，如圖片、音效、動畫等。

二、程式說明

(一) 物件：

1. Lollipop

- (1) 繪製 Lollipop 根據我給予的初始化數字
- (2) 關於繪製：將棒棒糖在原始座標軸上的位置當成新起點(0,0)並旋轉 rotate 度為了使棒棒糖是垂直插在大球而非按照原始座標軸走向。



2. Ball

- (1) 儲存繪製小球需要得位置
- (2) 按照儲存的座標位置畫出小球

3. Level

- (1) 建立根據選擇的關卡難度建立遊戲的關卡初始值陣列

4. Game

- (1) 主要為一個關卡，一次遊戲根據很多個 Game 所組成，當一個結束且順利通關將重新宣告 Game 套用下一個關卡的變數。
- (2) 主要變數介紹

```
this.time = time; // 每間隔多久更改一次轉的方向
this.degree = degree; //這次遊戲的難度
this.animationFrameId = null; //刷屏動畫
this.rotationIntervalId = null; //變換方向的時間控制器

this.startGameLoop(); // 遊戲開始的方程

//音效相關
this.isSound = isSound; //是否播放音效
this.collisionSound = new Audio('./material/fail.mp3'); //遊戲未通關音效
this.pass = new Audio('./material/pass.mp3'); //遊戲通關音效
this.arrow = new Audio('./material/arrow.mp3') //遊戲射出子彈音效
```

```

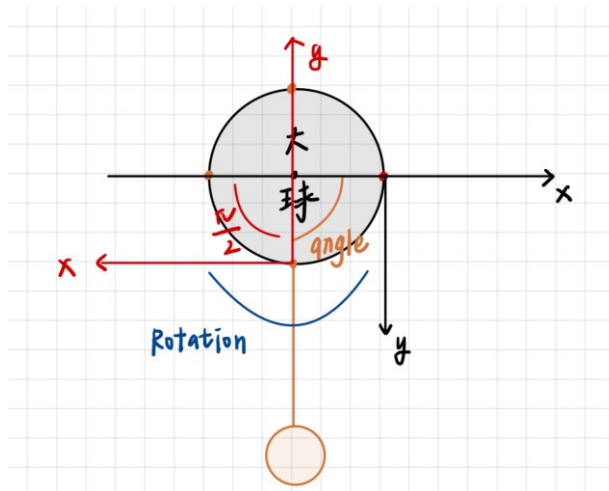
this.isStart = false; //遊戲是否仍在開始
this.CurrentLevel = CurrentLevel; //現在是第幾關
this.TotalLevel = TotalLevel; //這個難度的關卡總共有幾關
this.heart = heart; //目前剩下愛心數量
this.canvas = canvas; //畫布
this.ctx = canvas.getContext('2d'); //畫布
this.x = canvas.width / 2; //大球的位置
this.y = canvas.height / 3; //大球的位置
this.radius = 20; //大球的大小
this.OriSpeed = speed; //原始旋轉速度
this.speed = speed; //每一次的旋轉速度
this.numLollipops = numLollipops; //初始棒棒糖
this.numBalls = numBalls; // 初始球(有幾發子彈)
this.lollipopCounter = 1; //已經新增棒棒糖(已經射出幾發子彈)
this.isAdd = false; //剛剛有沒有新增棒棒糖，有的話要檢查他是否發生碰撞
this.lollipops = []; //現在在大球上的棒棒糖
this.balls = []; //所有子彈的位置

//射出子彈事件(點擊畫布、按下空白鍵)
this.canvas.addEventListener('click', this.addLollipopBelow.bind(this));
window.addEventListener('keydown', function (event) {
    if (event.code === 'Space') {
        this.addLollipopBelow();
    }
}).bind(this);

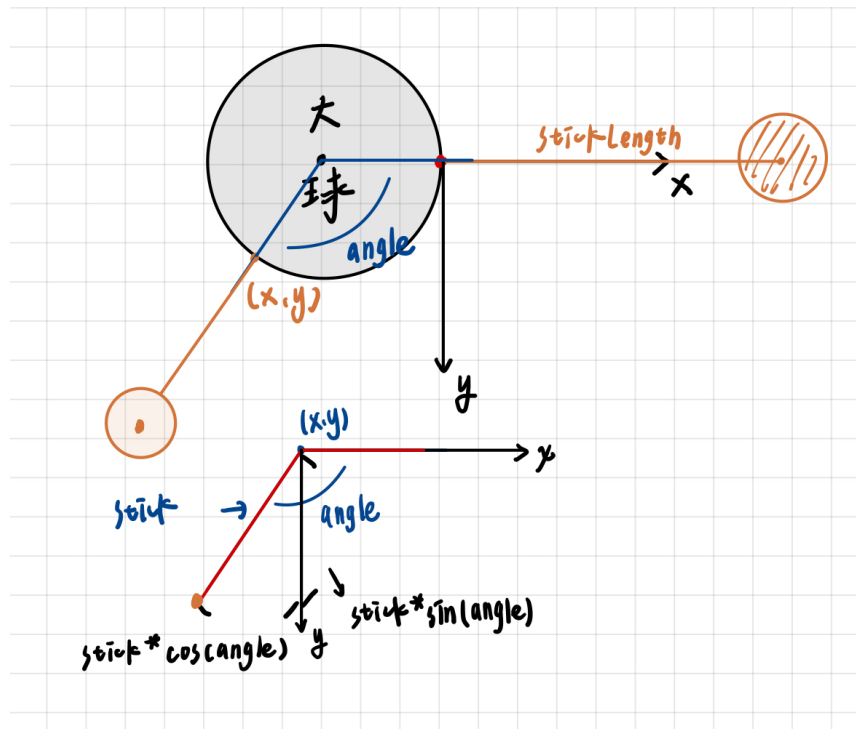
```

(3) 主要方程介紹

- a. startGameLoop()為開始這一個關卡，並在有 reset 關卡的時候進行重新設定基礎參數，將已經射出的小球數量更改為 0、並初始化遊戲開始頁面使用 initLollipops()和 initBalls()。stopGameLoop()將關卡進行結束，使用 clearAnimationFrame()和 clearRotationTimer()將時間和動畫刷屏器刪除。
- b. stopGameLoop()將關卡進行結束，使用 clearAnimationFrame()和 clearRotationTimer()將時間和動畫刷屏器刪除。
- c. clearRotationTimer()清除變換方向的時間控制器
- d. clearAnimationFrame()清除畫布持續刷屏繪製
- e. initLollipops()將初始設定的棒棒糖找到他們畫置的位置以及方向。如果初始在大球上的棒棒糖有 4 個，那將會平均分配棒棒糖在大球上的位置，並記錄要旋轉的角度(針對原始座標軸的角度再加上 $\pi/2$)



- f. `initBalls()` 找到球的初始位置訂在大球的正下方，按照坐標軸依序往下要畫的所有子彈的位置，在每一個子彈射出之後都會將所有未射出的子彈更改位置往前進行挪動。
- g. `draw()` 將所有該在畫布中出現的所有東西畫出來，例如棒棒糖 (`drawLollipops`)、小球 (`drawBalls`)、大球 (`drawCircle`)，如果有發生新增棒棒糖行為檢測是否碰撞原始有的 (`collisionDetection`)，或者已經將所有棒棒糖射出以通過該關卡 (`checkWin`)
- h. `drawCircle` 依照初始設定位置將大球繪出
- i. `drawLollipops` 因為要進行旋轉所以先改變棒棒糖的位置在進行繪製，先找到原來在大球上的角度再新增轉動的角度 (`speed`) 重新找尋位置儲存並使用 `class Lollipop` 的 `draw` 進行繪製。
- j. `drawBalls` 更改 `balls` 陣列中的小球位置和設定，並透過 `class Ball` 的 `draw` 進行繪製。
- k. `addLollipopBelow` 在大球的正下方增添一根棒棒糖並將小球已射出數量進行+1，如果有打開音效播放射出的音效，並觸發檢查是否發生碰撞方程確認是否結束遊戲
- l. `collisionDetection()` 因為新增棒棒糖繪放在陣列中最後一個，將其拿出並和所有棒棒糖做比對，找到棒棒糖上球的中心點在原始座標軸的位置。並與所有在大球上的棒棒糖小球中心點進行比對，如果兩個中心點的直線距離為小於小球直徑稱為未碰撞，大於稱為碰撞，觸發 `showGameOverPopup()` 方程。下圖是中心點計算方式。



- m. `checkWin()` 新增棒棒糖在通過碰撞檢查後，如果已經與總可以射出的子彈數相同及通過該關卡，將會結束這關卡 `stopGameLoop()`。同時跳出前往下一關或者恭喜你已完成這一個遊戲的介面。如果有打開音效，將會播放通關音效。
- n. `showGameOverPopup()` 在碰撞檢查過後發現遊戲失敗，觸發結束這關卡 `stopGameLoop()`。同時跳出 game over 的視窗，並根據已剩愛心數量決定是否可以使用愛心重啟遊戲，如果可以會跳出使用愛心按鈕，不可以將會回到開始頁面。
- o. `toggleRotationDirection()` 控制球旋轉的方向和每隔幾分鐘要旋轉一次。並根據給定的等級有不同的變化程度，同時希望每一次變動的時間和速度不一樣使用 `random` 將速度和更改方向時間進行調整。這樣可以避免一直在同一個區塊進行移動。如下：
 - ✓ 等級 1：速度落在原始速度正負 0.075，更改時間落在原始時間正負 700
 - ✓ 等級 2：速度落在原始速度正負 0.01，更改時間落在原始時間正負 500
 - ✓ 等級 3：速度落在原始速度正負 0.015 更改時間落在原始時間正負 300

(二) 主程式

1. 重要變數：

```
var currentLevelIndex = 0; //現在是第幾關
var game; // 關卡的宣告
var isHard = false; // 可以選擇關卡難度嗎? 只有還未開始遊戲時可以
var harddegree = 1; //遊戲初始難度可以調整
var heart = 5; //愛心初始數量
var levels; //關卡中間變數儲存陣列
var isBGM = true; //是否播放BGM
var isSound = true; //是否播放音效
const breakheart = new Audio('./material/heart.wav'); //心碎音效
const BGM = new Audio('./material/BGM.mp3'); //BGM
const levelHeading = document.getElementById('HowLevel'); //寫第幾關的container
BGM.volume = 0.3; //BGM音量
BGM.play(); //BGM一開始就播放除非有關掉
```

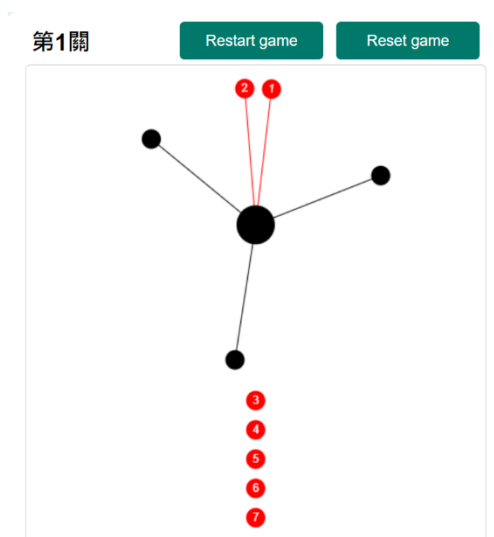
2. 主要頁面

遊戲結束和遊戲通過的頁面會覆蓋在遊戲頁面 overlay 上面

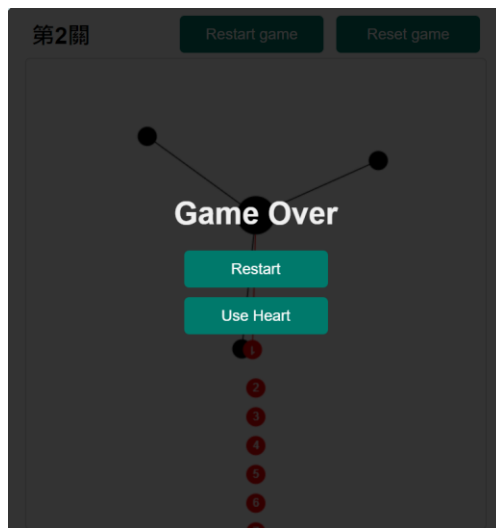
(1) start-game-container



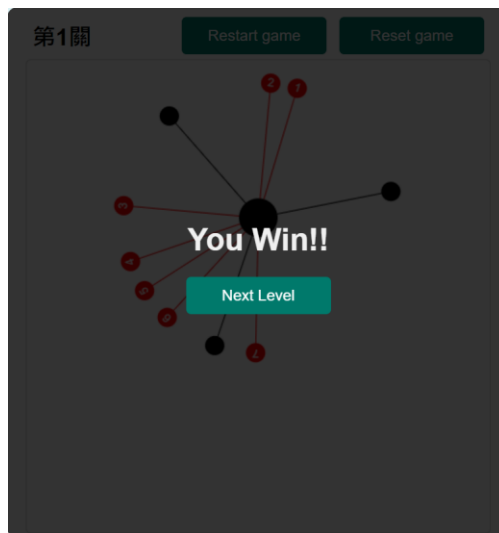
(2) overlay



(3) game-over-container



(4) next-level-container



3. 主要方程介紹

- decisiondegree()根據選擇的難度，進行關卡的宣告（未來希望可以自動產生關卡）
- startLevel()找到現在是第幾關並且宣告 class Game 根據 CurrentlevelIndex，同時將每一個頁面上調整過的 container 裡面的文字回到初始值。並進行 class Game 的 startGameLoop()正式進行關卡
- nextLevel()將關卡數+1，並會重新確認是否已經到達最後一關，以免造成誤觸發 startLevel()
- updateHearts()根據現在剩餘愛心數量進行愛心形狀的變更，如果剩餘 3 個愛心，將會有 3 個粉色愛心，2 個空心的愛心。
- toggleColor()根據旁邊所選定的按鈕改變按鈕的顏色跟遊戲困難程度變數 harddegreem，選定將會是綠色，未選定是黃色。
- 點擊 start-game-button，將隱藏 start-game-container 開啟 overlay 跟

進行 startLevel() 的方程，並將 ishard 變更為 true，禁止在遊戲途中調節遊戲的困難程度

- g. 點擊 reset-button，將會重新開始現在進行的關卡，進行 startLevel() 方程
- h. 點擊 next-level-button，分為兩種，如果已經是最後一個關卡將會跳出 start-game-container，恢復可以調整難度以及愛心數量，並將關卡回到第一關。如果不是最後一個關卡將會觸發 nextLevel()，並重新跳出 overlay
- i. 點擊 game-over-button，展開 start-game-container，隱藏 game-over-container，恢復可以調整難度以及愛心數量，並將關卡回到第一關。
- j. 點擊 use-heart-button，將愛心-1，並進行 updateHearts() 調整頁面上愛心的顯示。使用愛心如果有開啟音效，將會播放愛心碎裂的聲音。隱藏 game-over-container，顯示 overlay 頁面。並觸發 startLevel()。
- k. 點擊 BGM icon。如果有播放音樂，將音樂暫停，更改為音樂關閉的圖案；如果沒有播放音樂，將音樂開始，更改為音樂開啟的圖案。
- l. 點擊 sound。如果可以播放音效，將可以變成不可以，有觸發音效的方程將不會播放音效，更改為音效關閉的圖案；如果不可以播放音效，將不可以變成可以，有觸發音效的方程將會播放音效，更改為音效開啟的圖案。同時因為 class Game 有需要音效開關的按鈕，同時也要進行 game.isSound 的變動。

(三) 補充檔案 (Material)

1. 音效圖示

(1) 音效開啟中



(2) 音效關閉中



2. BGM 圖示

(1) BGM 開啟中



(2) BGM 關閉中



3. 愛心圖示

(1) 未用掉愛心



(2) 已用掉愛心



4. 音樂檔案

(1) arror.mp3：射箭音效

(2) BGM.mp3：背景音效

(3) fail.mp3：遊戲失敗音效

(4) heart.wav：心臟破裂音效

(5) pass.mp3：遊戲通過音效

三、結論及心得

進行完這個遊戲製作讓我更了解如何使用 JS 去設計網頁，而不是只用 html 和 css，它讓我可以與網頁進行互動。此外，在進行設計過程中一開始我直接使用我以前打程式的方式寫並沒有使用物件導向程式設計。後來使用物件導向的方法寫後發現可以有邏輯的展開一關和一關的遊戲，更加的便利。

在流暢度方面，因為每次旋轉全部都要重新刷新棒棒糖的位置，希望之後可以嘗試旋轉畫布的方式，看是否可以降低遊戲 lag 程度。最後，希望之後的關卡，可以想出一個演算法去跑關卡初始設定，而不是我用手先寫在遊戲當中，只根據難度做出選擇。