



介紹



DogDodge！閃避蕭告！

隨著貓貓的勢力崛起，貓奴數量逐漸攀升，越來越多人...甚至動物都開啟了吸貓的新世界！在這款遊戲中，玩家將扮演可愛的橘貓或藍貓，以滑鼠控制移動方向，幫助貓貓閃避所有飛撲而來的狂熱者們！



目錄

◆ 製作動機	3
◆ 素材來源	4
◆ 閃避物們介紹	5
◆ 遊戲構思過程	6
◆ 專案建置	9
◆ 遇到困難與挑戰	11
◆ 未來可新增功能	12



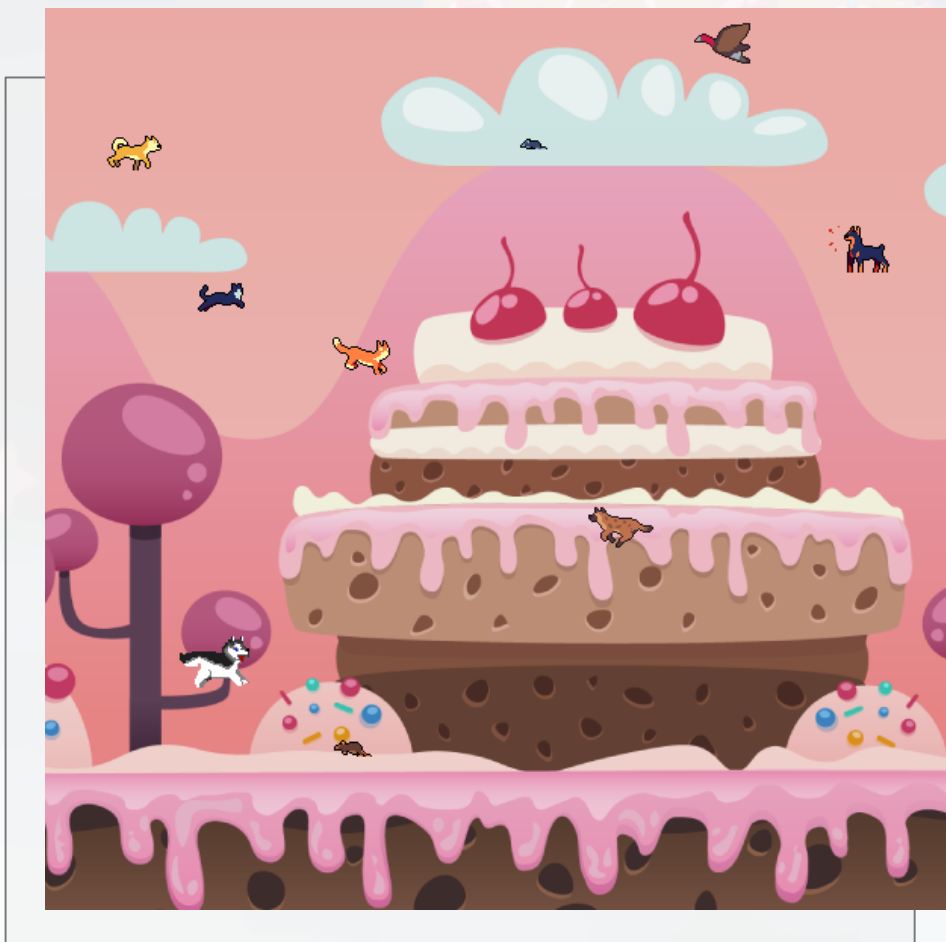
介紹

玩法介紹

每隔一段時間會隨機在畫面兩邊生成一隻動物，玩家在限定時間內避開所有動物即可獲勝，否則遊戲失敗。

製作動機

我平常很喜歡玩遊戲，覺得放鬆又舒壓，由於這次是一人作業，因此設計出這款程式上較為簡單，同時保留樂趣的小遊戲。





素材來源



免費遊戲素材網站：

- ◆ CRAFTPIX.NET
- ◆ itch.io





閃避物們介紹



閃避物1號棕色鼠鼠
簡單的向左移動



閃避物2號藍色鼠鼠
簡單的向左移動，速度更快



閃避物3號黃色狗狗
隨機決定起始位置



閃避物4號棕色鳥鳥
向左移動，並會上下飛動



閃避物5號棕色鬣狗
向左移，途中隨機向上下移



閃避物6號黑白二哈
向左移動，途中隨機折返



閃避物7號藍色狗狗
向左移動，途中會停下換動作



遊戲構思過程

Dog Dodge 企劃及程式文件

說明：

Dog Dodge 是一款輕鬆休閒的小遊戲，玩家將化身為街貓，使用滑鼠來閃避從各處撲來的動物們。因使用大量隨機數值，因此每個關卡都是獨一無二的，每次要生成的物件、各物件的起始位置、切換動作的時機，以及過關所需時間都可能不同。

(尚未實作)生成後將數據儲存到資料庫中，並賦予關卡編號，玩家可透過主選單的「載入遊戲」功能，輸入指定的關卡編號即可前往該關卡。此遊戲為單人單機遊戲，但會有一個線上的排行榜可以查看。

目錄：

- 一、 [流程圖](#)
- 二、 [系統說明](#)
- 三、 [介面說明](#)
- 四、 [未來規劃](#)

撰寫企劃及程式文件

內容包含說明需求、流程架構、以及各部件功能和串接檔案等。



遊戲構思過程

關卡編號	過關秒數	閃避物種類數	閃避物總數量	閃避物名稱	圖檔資料夾	尺寸	移動速度
1001	10	1	20	棕色鼠鼠	YM	27*13	10
1002	20	2	60	藍色鼠鼠	BM	20*10	15
1003	30	3	120	黃色狗狗	YD	40*25	4
1004	40	4	200	棕色鳥鳥	YB	39*30	7
1005	50	5	300	棕色鬚狗	HY	42*32	8
1006	60	6	420	黑白二哈	HS	50*39	5
1007	70	7	560	藍色狗狗	BD	40*34	8

編號	說明	細項
1	玩家角色表內欄位： 玩家角色編號、玩家角色名稱、圖檔資料夾、尺寸、移動速度、生命值、說明 圖檔資料夾路徑為：.../img/...	未填欄位預設為0
2	閃避物表內欄位： 閃避物編號、閃避物名稱、圖檔資料夾、尺寸、移動速度、說明 圖檔資料夾路徑為：.../img/...	未填欄位預設為0
3	資料庫資料表1內欄位： 關卡編號、過關秒數、閃避物種類數、閃避物總數量 用來設定閃避物種類數及數量，隨機抽取閃避物出現。	未填欄位預設為0

建立資料表格

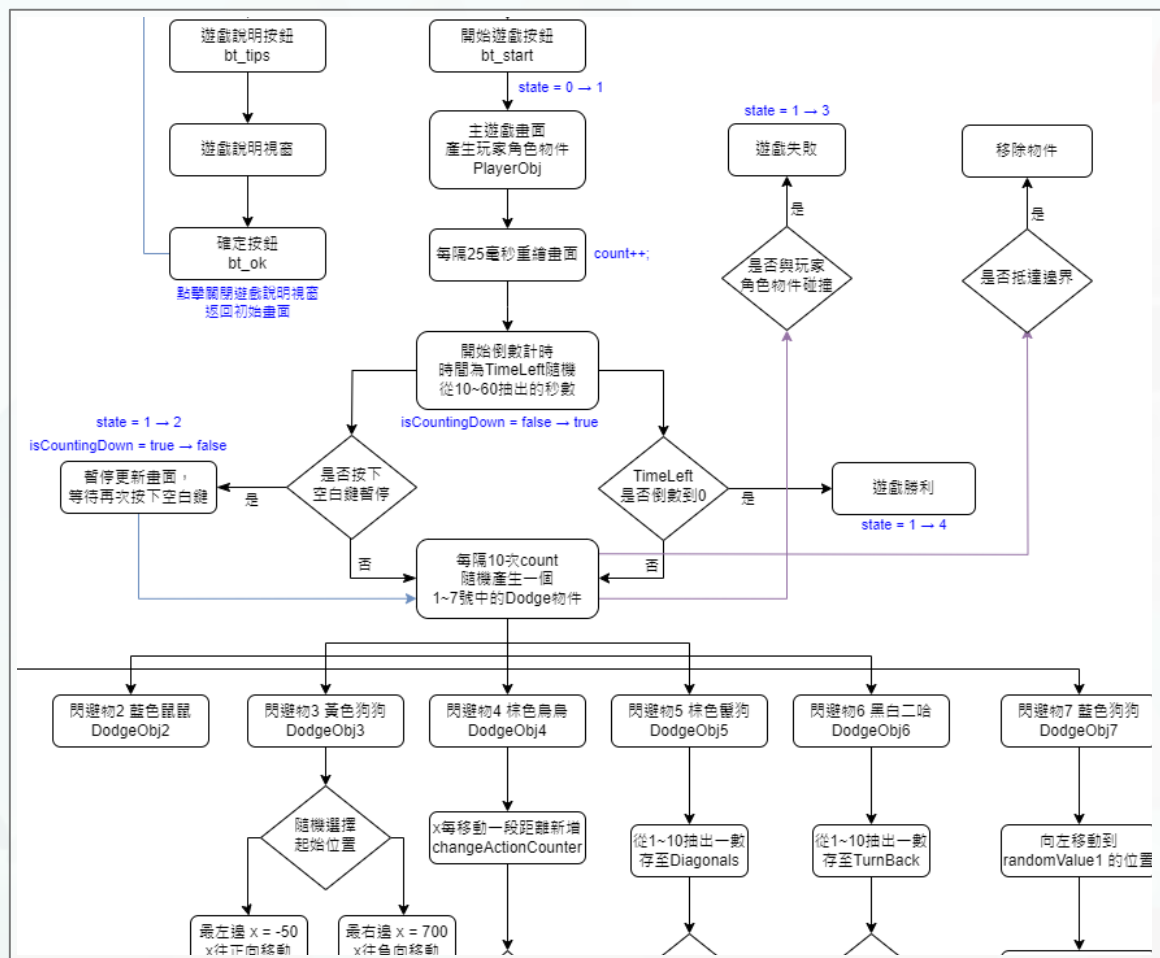
紀錄詳細的數值資訊，
以便日後維護與整理。



遊戲構思過程

建構流程圖

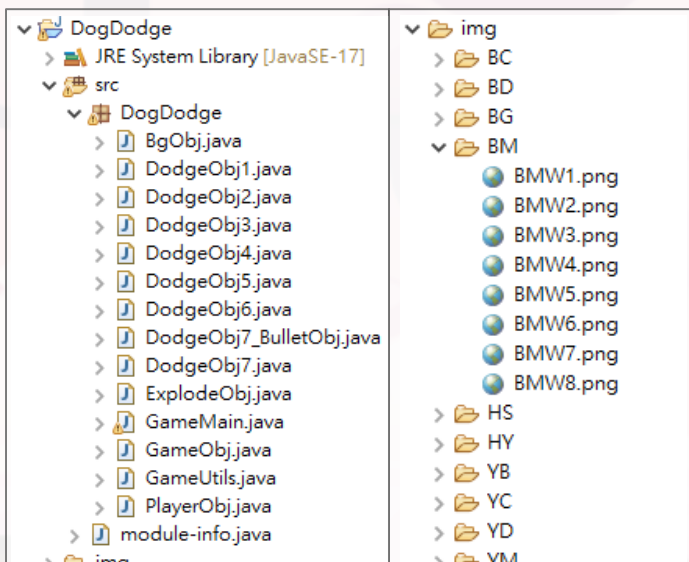
釐清並整理思路，以及
程式邏輯規劃與檢查。





專案建置

建立各物件檔案與動畫圖片檔，
每間隔一段時間隨機生成物件。



```
180 根據抽到的數字選擇要生成幾閃閃遺物
181 (count % 10 == 0) {
182     int randomDodgeIndex = (int) (Math.random() * 7) + 1; // 生成 1 到 7 的隨機數字
183
184     if (randomDodgeIndex == 1) { // 1號 棕色鼠鼠
185         GameUtils.DodgeObjList1.add(
186             new DodgeObj1(DodgeObj1.DodgeImgs1, 700, (int) (Math.random() * 13) * 47, 27, 13, 10, this));
187         GameUtils.gameObjList.add(GameUtils.DodgeObjList1.get(GameUtils.DodgeObjList1.size() - 1));
188     } else if (randomDodgeIndex == 2) { // 2號 藍色鼠鼠
189         GameUtils.DodgeObjList2.add(
190             new DodgeObj2(DodgeObj2.DodgeImgs2, 700, (int) (Math.random() * 10) * 60, 20, 10, 15, this));
191         GameUtils.gameObjList.add(GameUtils.DodgeObjList2.get(GameUtils.DodgeObjList2.size() - 1));
192     } else if (randomDodgeIndex == 3) { // 3號 黃色狗狗
193         int x = (Math.random() < 0.4) ? 700 : -50; // 從左邊出現的機率是3成
194         GameUtils.DodgeObjList3.add(
195             new DodgeObj3(DodgeObj3.DodgeImgs3, x, (int) (Math.random() * 25) * 24, 40, 25, 4, this));
196         GameUtils.gameObjList.add(GameUtils.DodgeObjList3.get(GameUtils.DodgeObjList3.size() - 1));
197     } else if (randomDodgeIndex == 4) { // 4號 棕色鳥鳥
198         GameUtils.DodgeObjList4.add(
199             new DodgeObj4(DodgeObj4.DodgeImgs4, 700, (int) (Math.random() * 30) * 20, 39, 30, 7, this));
200         GameUtils.gameObjList.add(GameUtils.DodgeObjList4.get(GameUtils.DodgeObjList4.size() - 1));
201     } else if (randomDodgeIndex == 5) { // 5號 棕色鬃狗
202         GameUtils.DodgeObjList5.add(
203             new DodgeObj5(DodgeObj5.DodgeImgs5, 700, (int) (Math.random() * 32) * 19, 42, 32, 8, this));
204         GameUtils.gameObjList.add(GameUtils.DodgeObjList5.get(GameUtils.DodgeObjList5.size() - 1));
205     } else if (randomDodgeIndex == 6) { // 6號 黑白二哈
206         GameUtils.DodgeObjList6.add(
207             new DodgeObj6(DodgeObj6.DodgeImgs6, 700, (int) (Math.random() * 39) * 15, 50, 39, 5, this));
208         GameUtils.gameObjList.add(GameUtils.DodgeObjList6.get(GameUtils.DodgeObjList6.size() - 1));
209     } else { // 7號 藍色狗狗
210         GameUtils.DodgeObjList7.add(
211             new DodgeObj7(DodgeObj7.DodgeImgs7, 700, (int) (Math.random() * 34) * 18, 40, 34, 8, this));
212         GameUtils.gameObjList.add(GameUtils.DodgeObjList7.get(GameUtils.DodgeObjList7.size() - 1));
213     }
214     DodgeCount++;
```



專案建置

```
// 是否切換動作
private static boolean switchImage = false;
// 第二個動作執行的次數
int secondActionCount = 0;
// 切換動作時機點，在初始化時產生兩個隨機x值
private int randomValue1;
private int randomValue2;
Random randomX = new Random();
{ // 生成第一次和第二次換圖片的時機點
    randomValue1 = randomX.nextInt(351) + 150;
    randomValue2 = randomX.nextInt(5) + 1;
}
```

```
// 檢查是否改變移動方向
if (x <= DiagonalsPlace) {
    if (Diagonals < 5) { // 40%的機率改變
        moveUpDown = true;
    }
}
```


設置動畫邏輯與切換動作判斷。

```
// 每次刷新畫面時將x往負數移動，達成背景畫面移動的效果
@Override
public void paintSelf(Graphics gImage) {
    super.paintSelf(gImage);
    x -= speed;
    if (x <= -1200) {
        x = -90;
    }
}
```

```
// x -= speed; 換圖片時停止奔跑
// 做完第二個隨機值次數的動作後，換回動作1
if (secondActionCount < randomValue2) {
    if (DodgeImgCount7_2 < 12) {
        img = DodgePics7_2[DodgeImgCount7_2];
        super.paintSelf(gImage);
        DodgeImgCount7_2++;
    } else {
        // 當一個動畫循環結束，再重置回0，實現無限loop
        DodgeImgCount7_2 = 0;
        secondActionCount++;
    }
}
```



遇到困難與挑戰

- ◆ 找不到或無法載入主要類別
- ◆ 呼叫其他檔案內的方法，大小寫不一致，導致執行時卡死並瘋狂跳錯
- ◆ 使用多張圖片製作動畫效果，嘗試調慢切換速度，但被Thread.sleep擋住
- ◆ 背景圖片移動到邊界時的銜接調整
- ◆ 每次生成不同外觀的閃避物
- ◆ 一直找不到錯誤在哪裡，最後發現原來 { 包到其他段
- ◆ 閃避物反向移動時，圖片無法成功水平翻轉
- ◆ 閃避物圖片重疊 (未解決) 
- ◆ 初始畫面的按鈕顯示異常，要滑鼠滑過去才顯示 (未解決)
- ◆ 時間不夠用！！！（無解）



未來可新增功能

- ◆ 新增更多角色，並可讓玩家自選，每隻都有不同特性
- ◆ 加入更多趣味互動以及更多動畫表演
- ◆ 開放即時連線對戰
- ◆ 遊戲中放入能讓玩家蒐集的貨幣/獎勵，並新增商店
- ◆ 加入更多樣的地圖和閃避物，並根據不同節日推出不同外觀的內容
- ◆ 設定**Bonus Time**，可加速破關
- ◆ 新增排行榜功能，可顯示玩家所有關卡加總的存活時間排行
- ◆ 新增**Boss**關卡，擊敗**Boss**可獲得各種獎勵
- ◆ 讓玩家自行選擇遊戲難度
- ◆ 將關卡資料存至資料庫，並可讓玩家加載指定關卡



感謝！

THANK YOU !!