



Plants V.S. Zombies

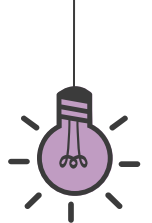
---Survival mode

113511167 廖顯佑

113511161 楊士葦

113511247 陳冠頤

助教



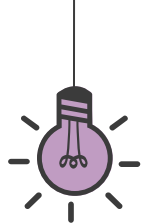
專題介紹

本專題有參考網路上的開源專案(<https://github.com/wszqkzqk/pypvz>), 參考了其:

- **檔案架構:** 借用其模組化與組織化的結構, 提升程式碼可維護性
- **場景設置和角色機制:** 借鑑場景渲染與角色移動的實現方式
- **角色動畫呈現方式:** 感謝助教友情客串

我們進行了以下改進:

- **關卡機制:** 設計了生存模式, 新增計分系統以增強遊戲性
- **介面製作與串接:** 開發並整合各使用者介面, 確保流暢的導航體驗
- **角色和背景美術改編:** 重新設計角色和背景視覺效果, 增添獨特風格



專題特色

- **生存模式實作：**

開發具挑戰性的生存模式，讓玩家面對無盡的殭屍浪潮，難度會隨生存時間增加

- **排行榜系統：**

實現排行榜功能，記錄並展示玩家的分數

- **完整用戶流程（State machine）：**

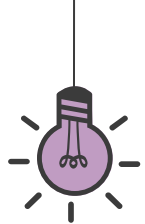
設計流暢的用戶體驗，涵蓋選單、遊戲與排行榜間的轉換

- **角色美術：**

打造幽默且創意的角色設計，提升視覺吸引力

- **線上遊玩功能：**

我們有把遊戲放到Replit上，有空的人可以去玩



OOP核心概念

本專題應用到的OOP原則如下：

➤ **繼承 (Inheritance) :**

建立類別層次結構，例如基礎Plant與Zombie類別延伸至特定角色類型

➤ **抽象 (Abstraction) :**

使用抽象類別定義和管理關卡狀態，增加實作靈活性

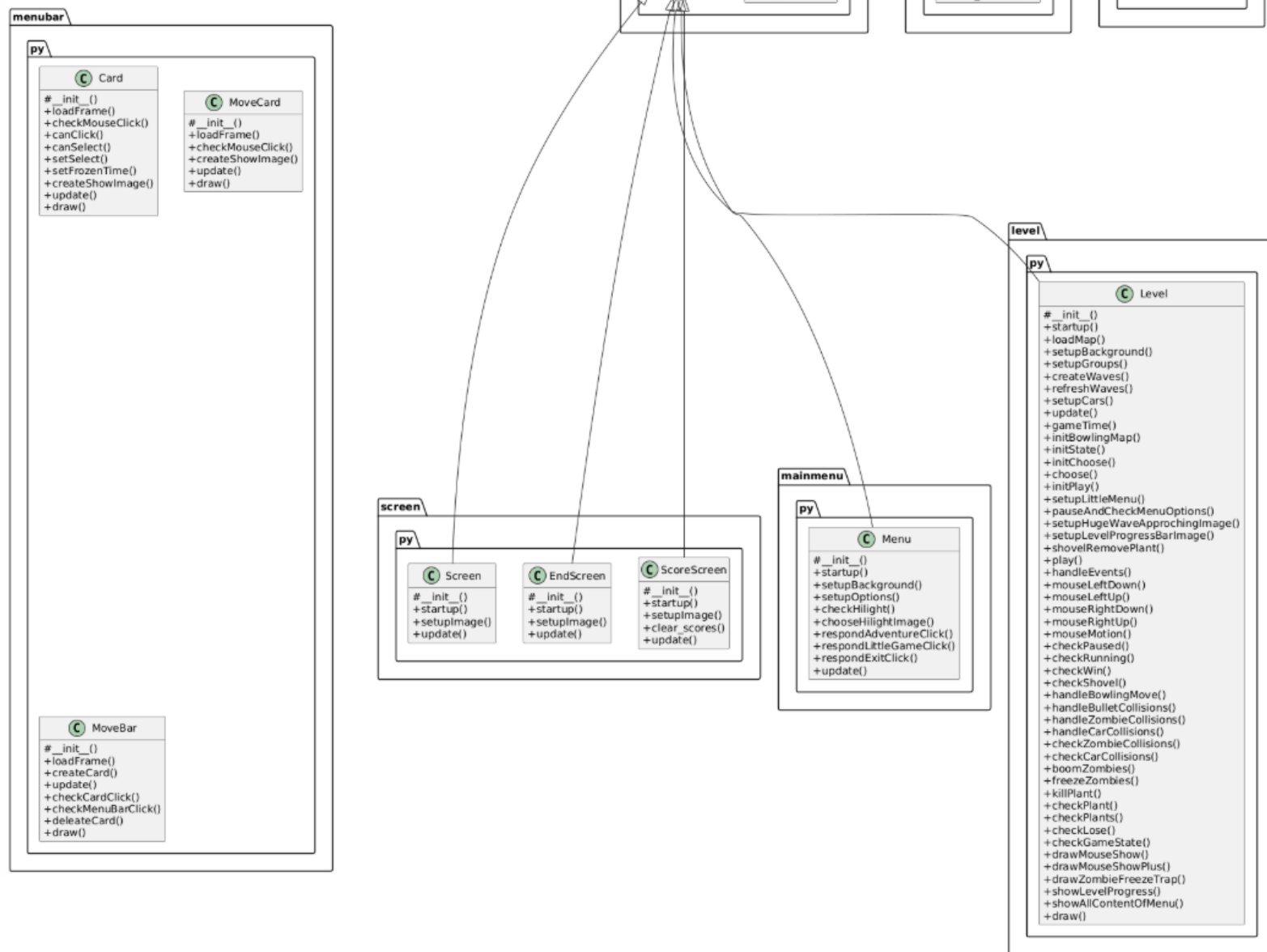
➤ **封裝 (Encapsulation) :**

將遊戲資料（如血量、位置）封裝於類別中，僅暴露必要介面

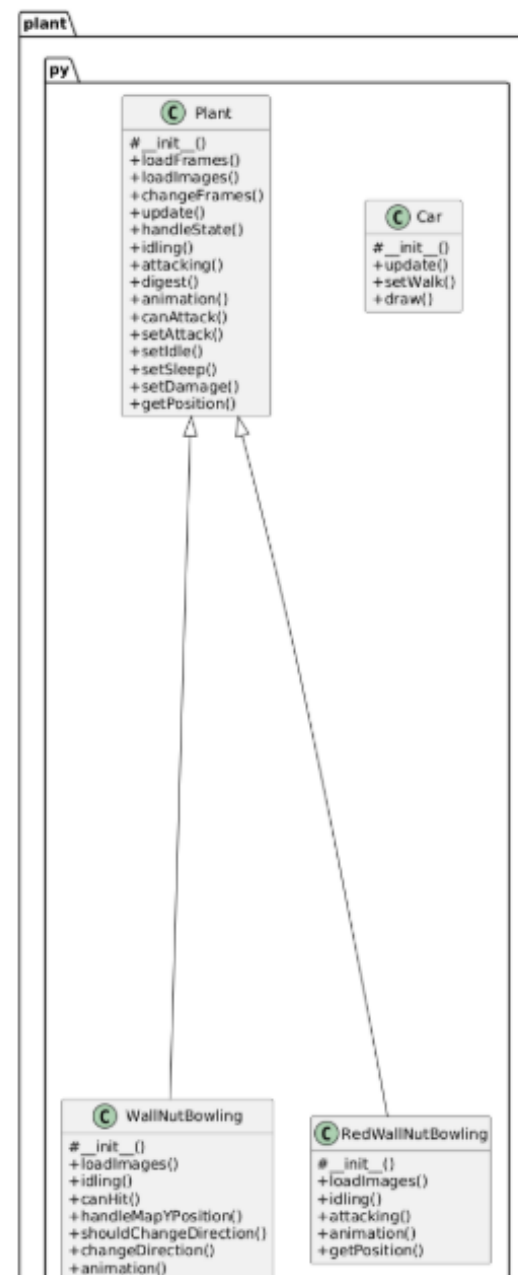
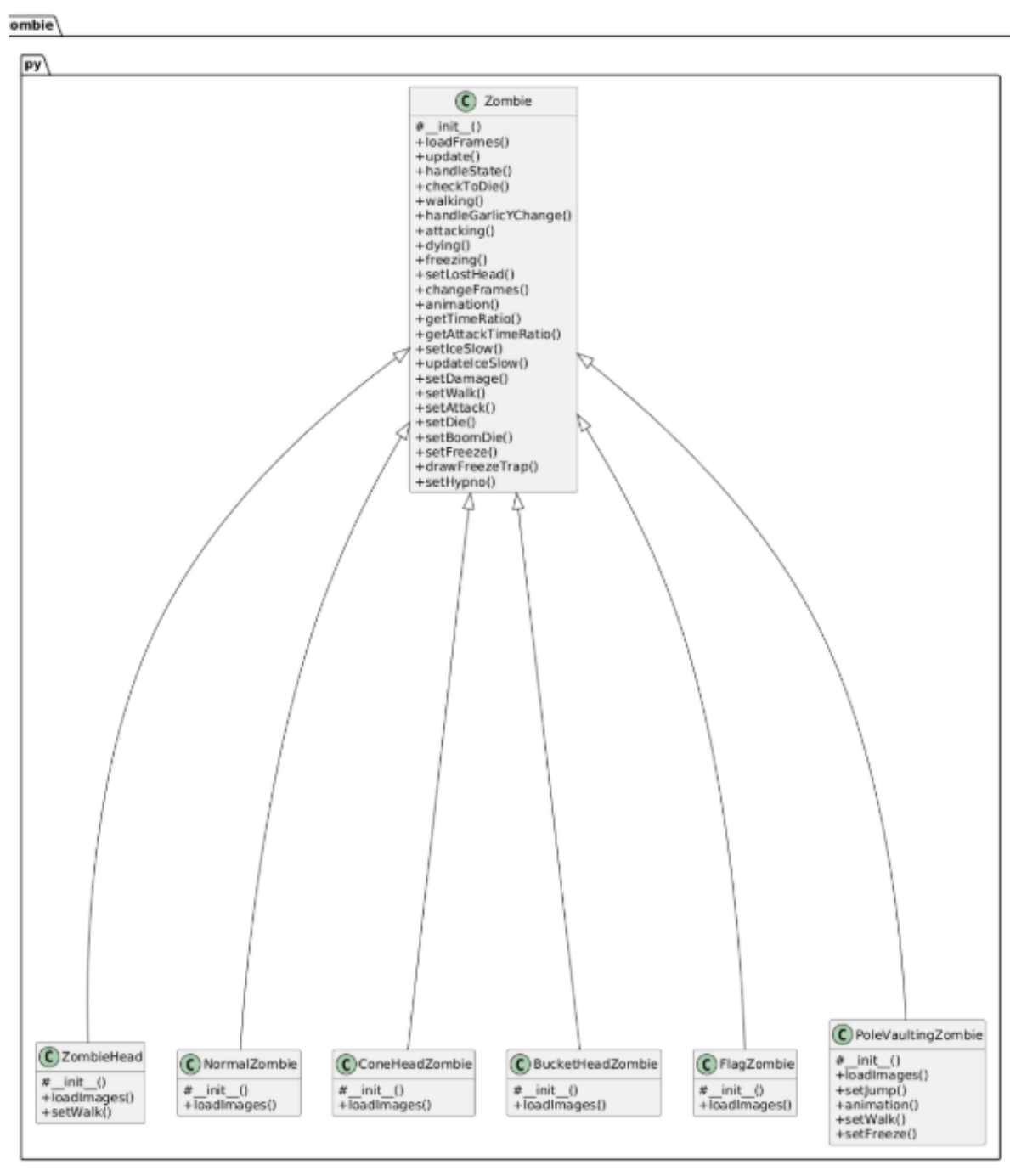
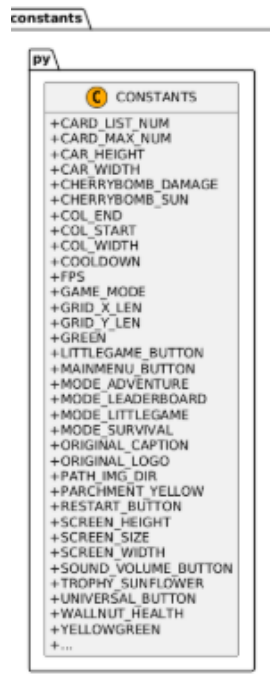
➤ **多型 (Polymorphism) :**

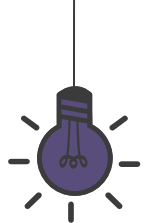
讓不同植物與殭屍類型共享共同介面，但表現出獨特行為

Class Diagram



Class Diagram





實作架構

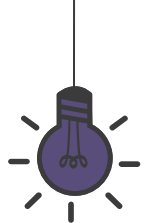
State machine結構

- **class Menu:**
遊戲啟動的起始畫面
- **class Level:**
遊戲機制
- **class Endscreen:**
顯示遊戲時間及輸入姓名紀錄
- **class ScoreBoard:**
顯示歷史前五遊戲紀錄
- **class Control:**
控制遊戲流程並檢測狀態轉換

```
game = tool.Control()

state_dict = { c.MAIN_MENU:    mainmenu.Menu(),
               c.PLAY:        level.Level(),
               c.ENDSCREEN:    screen.EndScreen(),
               c.SCOREBOARD:   screen.ScoreScreen()
               }

game.setup_states(state_dict, c.MAIN_MENU)
game.run()
```



實作架構

Survival Mode

- **NUM_FLAGS:**

設定需克服的旗幟數量

- **refreshWave():**

重置殭屍波次以迎接新挑戰

- **createWave():**

根據遊戲進度生成殭屍波次

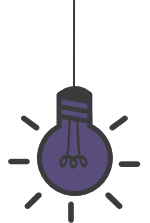
- **survival_rounds:**

記錄已克服的旗幟數，用於動態調整難度

```
class Level(tool.State):
    def createWaves(self, useable_zombies, num_flags, survival_rounds=0,
                    waves):
        self.waves = waves
        self.created_zombie_from_pool = False

    def refreshWaves(self, current_time, survival_rounds=0):
        """
        刷新波次；支援無盡生存模式：
        - 打完 NUM_FLAGS 面旗幟、場上殭屍清空 → 直接重開下一輪，
        並以 self.survival_rounds 逐輪遞增難度。
        """
```

```
{
    c.BACKGROUND_TYPE: 6,
    c.GAME_TITLE: "NNNuts Bowlings",
    c.CHOOSEBAR_TYPE: c.CHOOSEBAR_BOWLING,
    c.SHOVEL: 0,
    c.SPAWN_ZOMBIES: c.SPAWN_ZOMBIES_AUTO,
    c.INCLUDED_ZOMBIES: ( c.NORMAL_ZOMBIE, c.CONEHEAD_ZOMBIE,
                          c.POLE_VAULTING_ZOMBIE, c.BUCKETHEAD_ZOMBIE, ),
    c.NUM_FLAGS: 100,
    c.CARD_POOL: { c.WALLNUTBOWLING: 300,
                   c.REDWALLNUTBOWLING: 100, }
},
```

流程介紹

流程:

1. 初始畫面:

顯示主選單，包含開始遊戲與查看排行榜的選項。

2. 遊戲主體:

進入生存模式，玩家放置植物對抗殭屍浪潮

3. 排行榜:

遊戲結束後展示高分排行榜，讓玩家比較表現。

4. 返回初始畫面:

循環回到主選單，增加可重玩性。



Demo

運行環境: Ubuntu 22.04 + Python + Pygame

https://drive.google.com/file/d/1WWg8f6spG-bCWQu3ubRMI0tVnJkvBKrO/view?usp=drive_link



困難處

PROBLEM 01

殭屍生成問題:

問題: 生存模式中，打完第二波旗幟後無法生成殭屍

解決方案:

1. 調整constant.py中的常數 (NUM_FLAGS) ，確保殭屍持續生成
2. 新增除錯日誌追蹤波次進展，修正波次計數器的邏輯錯誤



困難處

PROBLEM 02

程式碼理解問題：

理解與修改開源程式碼在最初較為困難

解決方案：

使用**AI輔助工具**分析程式碼結構並提出改進建議，加速開發與除錯過程



廖: 角色美術修改、口頭報告

楊: 畫面串接、初始頁面和排行榜製作

陳: 生存模式製作、簡報製作、專案發想

助教: 友情客串

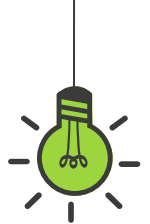


克服的挑戰：

修正了關鍵錯誤（如殭屍生成問題），並成功將複雜的開源程式碼改編為符合我們需求的版本

經驗教訓：

學會了清晰文件、模組化設計與早期測試的重要性，以避免如波次錯誤的問題



未來展望

- **功能增強：**

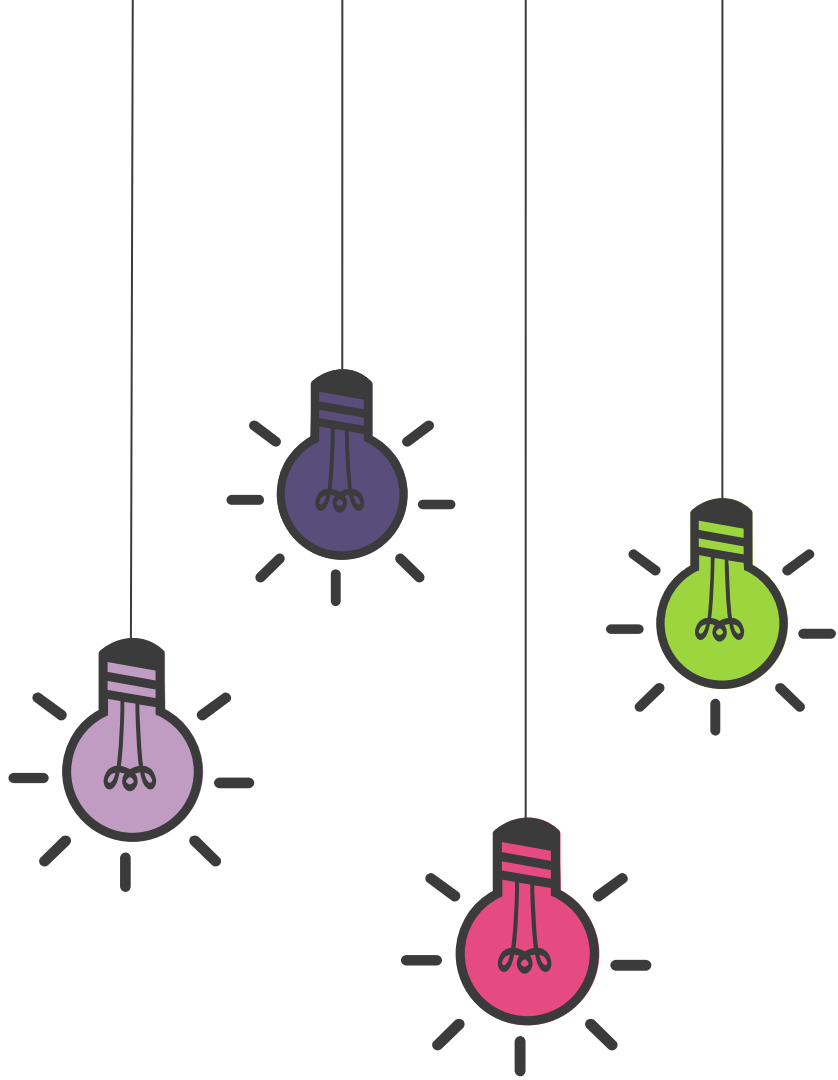
新增更多植物與殭屍種類，加入更多音效與背景音樂，提升沉浸感

- **技術改進：**

優化大量殭屍波次的性能。增加單元測試，確保程式碼可靠性

- **潛在功能：**

開發自訂關卡編輯器，支援玩家創作內容



謝謝聆聽
