

DBD



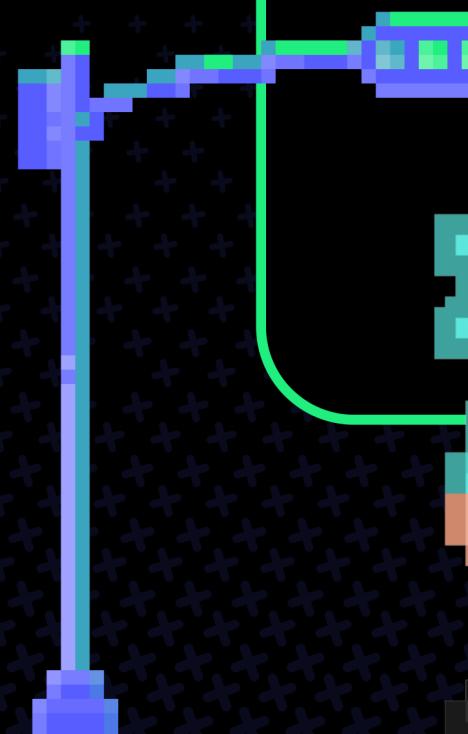
PD FINAL PROJECT



GROUP 4

DEAD

BY DEADLINE

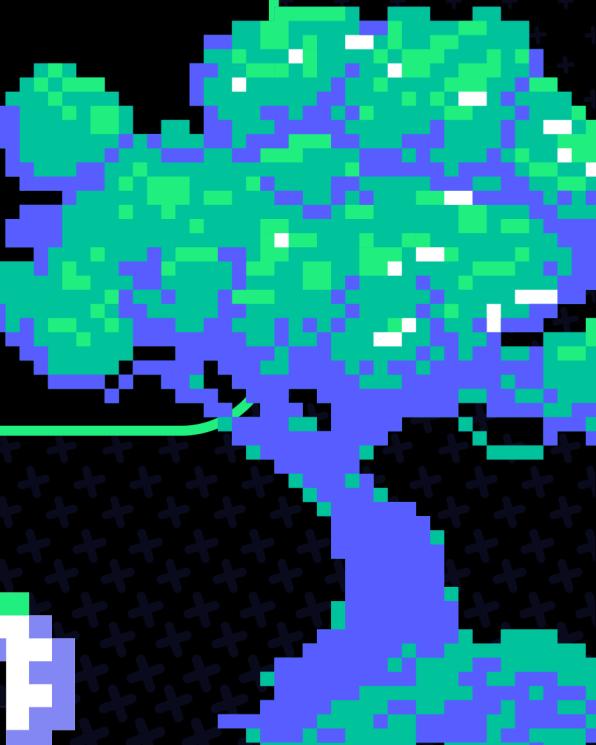


▼ RRR!



START

◆ PROGRAMMING DESIGN



MENU

01

07

12

ask for  
help



# ORIGIN OF OUR IDEA



「我是資管系的學生，死線太多我要大爆設ㄌ」

我是一個資管系的新生，上程式設計之前我是真的以為大家都跟我一樣沒學過C++，進來之後才發現只有我什麼都不懂。每個星期都像是經歷一場生死時速，要趕在星期二之前設出來(程式)，隨着作業複雜度的逐漸提升，我設(計)的速度也要隨之加快，要不然就來不及啦!!!

Title

PD 112-1 Homework 10

PD 112-1 Homework 9

PD 112-1 Homework 8

MI·TU

BACK TO  
MIDTERM WEEK

SIGN IN



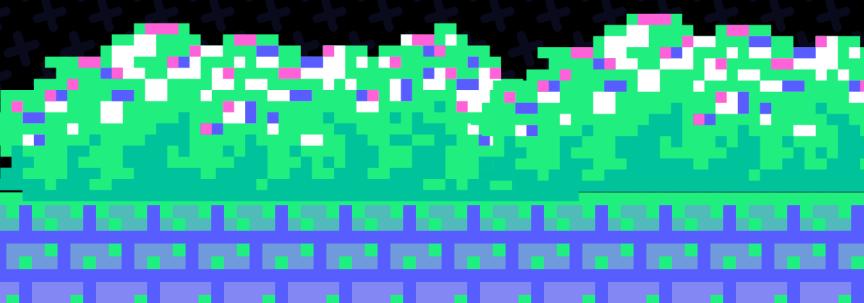
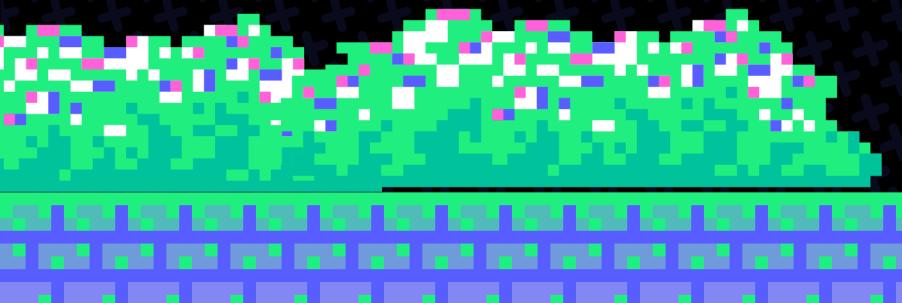
BACK TO MIDTERM WEEK

# 「在資管系絕地求生的經歷」



◆ 遊戲的主要目標是「生存」，程式設計「作」業會慢慢出現並逐漸向期限 (deadline) 靠近，玩家必須透過在作業到達期限之前把程式作業設 (計) 到，以消除程式作業，如果作業沒有被設到，並到達了期限，玩家這個學期的GPA就會減少 0.3 (滿分是 4.3)。

當學期 (遊戲時間) 結束或GPA過低時，遊戲就會Gameover，代表玩家Pass或fail了。在過程中，玩家會遭遇事件「Midterm Exam」和「Final Exam」，在這期間，程設作業爆率將會大大提高，而到達死線的速度也會加快。

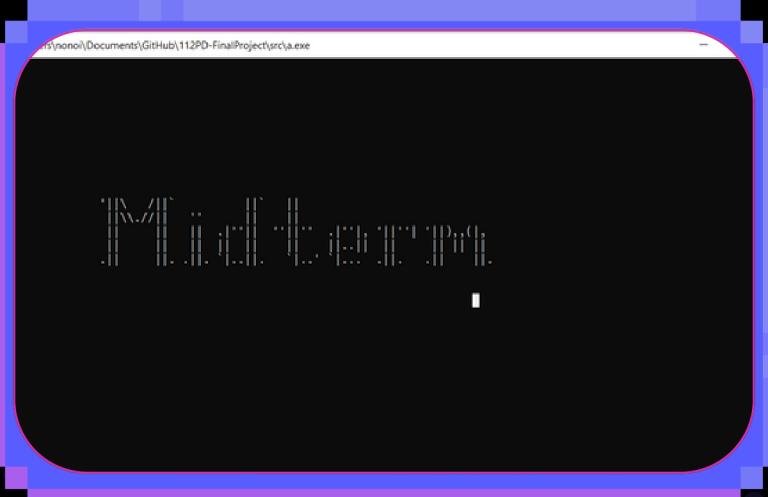
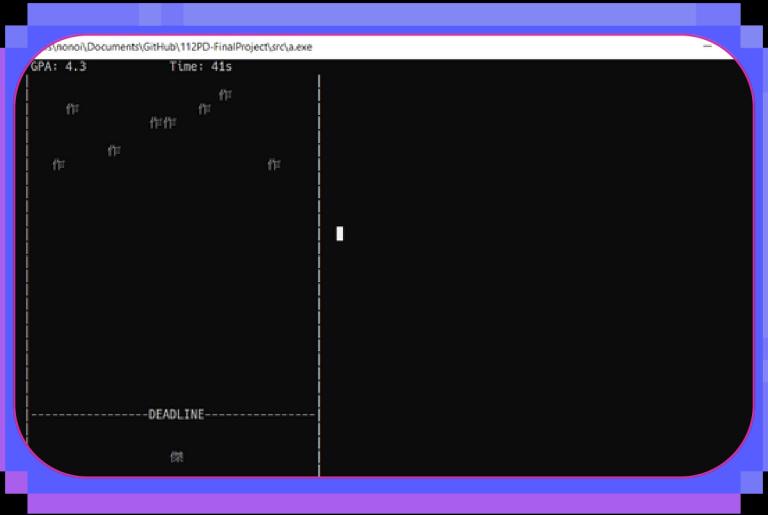
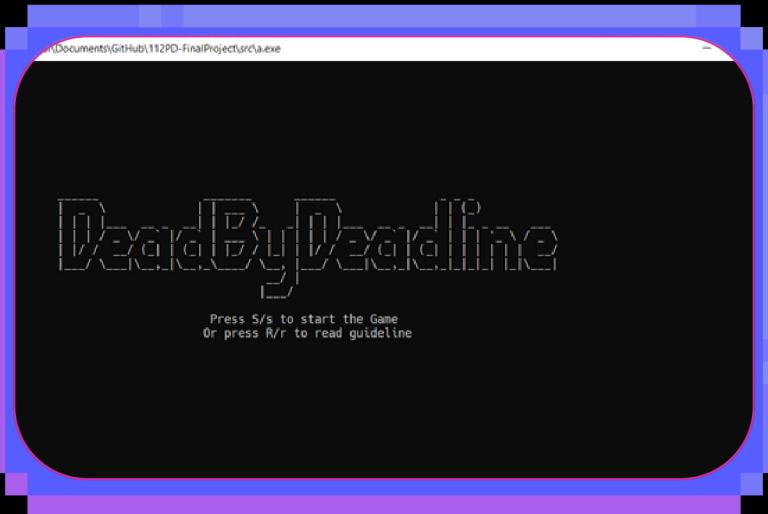


MENU



# GAME PREVIEW

BACK TO MIDTERM WEEK



## ◆ 遊戲起始畫面

進入遊戲之後會進入到遊戲大廳，裡面可以選擇兩個選項：按 R/r 來閱讀劇情，或者按 S/s 開始遊戲

## ◆ 遊戲開始後

畫面上方分別會顯示「GPA」和剩餘時間，而「作」代表的是玩家需要用射擊來消除的敵人(作業)。玩家點按左右鍵來移動，並利用空白鍵或enter鍵進行射擊，以將「作」消滅

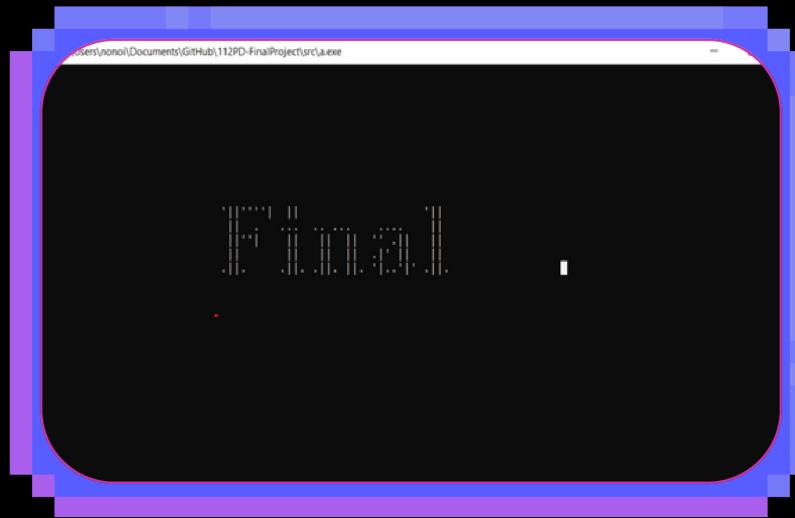
## ◆ 事件觸發：期中考

時間剩下40秒後會觸發「期中考」事件，跳到 Midterm 頁面後，再讓玩家繼續遊戲。轉回畫面之後，「作」掉落的數量和速度會顯著增加。

MENU



# GAME PREVIEW

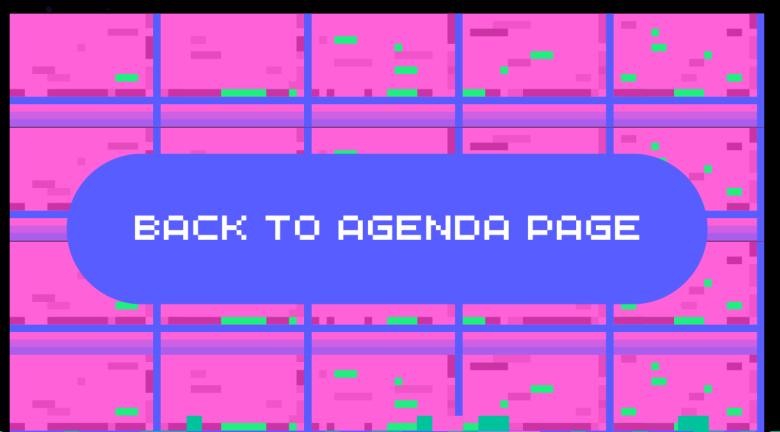


## ➔ 事件觸發：期末考

時間剩下20秒時將觸發「期末考」事件，會跳到 Final這個頁面後，再讓玩家繼續遊戲。轉回畫面之後，「作」的數量和掉落速度會宇宙大爆炸地增加。

BACK TO MIDTERM WEEK

PLAYER 1



```
## ##### ##
```

Your GPA is: 3.3  
New Record!!!  
Press E/e to exit the game  
Or press S/s to start a new ga

新紀錄!

```
## ##### ##
```

Your GPA is: 1.8  
Your best GPA is: 3.3  
Press E/e to exit the game  
press S/s to start a new ga

紀錄先前比較高

```
## ##### ##
```

Your GPA is: 0.8  
You didn't pass  
Press E/e to exit the game  
press S/s to start a new g

在學期內(一分鐘內)GPA低於1.0

觸發結束的條件會有兩個：GPA低於1.0，不論剩下時間有多少直接結束。這種情況，第二行會印出“**You didn't pass**”。

第二種狀況會是玩家撐過了一個學期(一分鐘)，才進入結算畫面。

而進入結算畫面會有兩種情況：我們會判斷這一次的紀錄有沒有比之前玩過的高，如果有的話，顯示“**New Record!**”，如果沒有的話就顯示你的最佳GPA。

## 遊戲結束後結算畫面

MENU

→ 01

◆ 07

★ 12



# FILE I/O

讀取文字檔案，像是劇情文字。劇情文字會被視為”new window”，並且置中在遊戲畫面中，而讀取檔案時，我們也使用了 Exception Handling 來確保程式碼不會錯誤。

```
Game::NewWindows(string File)
{
    Open File
    if (File is not opened) throw("File not found")

    while(read each lines){
        align the text center
    }
}
```

MENU

01

07

12



# 標頭檔 GAME.H

這個標頭檔中主要存放對於物件的宣告，以及遊戲參數的設定。由於篇幅的關係，在此我們列舉幾個重要的遊戲參數作說明：

這邊主要宣告的是在不同遊戲階段（遊戲一開始、期中考、期末考三個階段）所設定的敵人下落速度，以及隨機生成數量的參數。我們的程式是每20毫秒刷新一次（一幀）看使用者有沒有做任何事情，而速度的定義代表敵人每一幀會在畫面上移動多少格。

```
const int SleepPerLoop = 20; // 每20毫秒就刷新一次  
const double PlayerSpeed = 4; // 速度的定義是每幀移動幾格  
const double BulletSpeed = 0.8;  
  
// 敵人的生成還有移動速度  
const double NormalEnemySpeed = 0.05;  
const double NormalEnemySpawnRate = 0.05;  
const double MidtermEnemySpeed = 0.2;  
const double MidtermEnemySpawnRate = 0.05;  
const double FinalEnemySpeed = 0.3;  
const double FinalEnemySpawnRate = 0.25;
```

# EXCEPTION HANDLING

我們總共有三個Exception handling，分別用於處理：

- 沒有成功開啟檔案，會跳出提醒錯誤訊息“File not found”
- 螢幕視窗開太小，會跳出提醒錯誤訊息“Window width is too small”
- 偵測遊戲狀態的參數(用來決定敵人生成數量及移動速度)有沒有異常，如果有則顯示“Status Error”

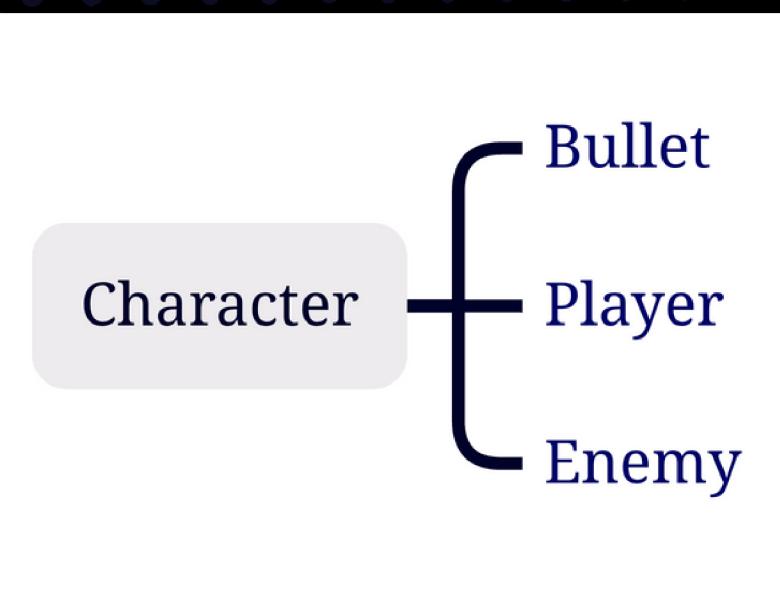
```
if (!file.is_open())
    throw(File + " File not found");

if (WindowWidth - len < 0)
    throw("Window Width is too small");
```

# POLYMORPHISM, INHERITANCE, ETC.

## Character

我們主要以物件 Character 作為被繼承的虛擬物件進行設計，延伸出三個不同的子物件：Bullet, Player, Enemy，並且加了 `erase` 函數來消除物件，以及用 `OutOfBorder` 函數偵測這個物件有沒有超越遊戲邊界。



```
class Character
{
protected:
    double x, y;

public:
    Character(double x, double y);
    virtual bool OutOfBorder();
    virtual void Erase();
    virtual ~Character(){}
};
```

# POLYMORPHISM, INHERITANCE, ETC.

## Bullet

Bullet 物件指的是玩家透過按下空白鍵射擊出的子彈，並且用Hit函數判斷有沒有擊中 enemy物件，若有擊中則以Erase函數消除 Enemy 物件；再以TouchDeadline函數判斷 Bullet 物件是否有碰到 Deadline 線，若碰到為true 則重劃一次 Deadline 線

```
class Bullet : public Character
{
private:
    double speed;

public:
    Bullet(double x, double y);
    void Draw();
    void Move(); //子彈往上一個 speed (定義在標頭檔)
    bool Hit(Character *character); // 判斷有沒有打到「作」
    bool OutOfBorder(); //判斷他有沒有碰到最上面
    void explode(); // 碰到「作」之後爆炸
    bool TouchDeadline();
    //判斷他有沒有碰到deadline這條線，如果有的話那條deadline的線要重劃一次
};
```

# POLYMORPHISM, INHERITANCE, ETC.

## Player

class Player是用來定義玩家可以進行的操作。Player物件中定義了bullet物件的動態陣列，因此基本上沒有子彈數量限制。

```
class Player : public Character
{
private:
    std::vector<Bullet *> bullets;
public:
    Player(double x, double y);
    void Draw();
    void Move(char key);
    void Move();
    void Shoot(std::vector<Bullet *> &bullets);
```

MENU

01

07

12



# POLYMORPHISM, INHERITANCE, ETC.

## Enemy

主要控制Enemy物件的落下及生成。而判斷Enemy物件被消除的時機已在 class Bullet 中定義。

```
class Enemy : public Character
{
    friend class Game;

private:
    double speed;
    std::string pic;

public:
    Enemy(double x, double y); // 建構子
    Enemy(double x, double y, double speed);
    void Draw(); // 生成在螢幕上
    void Move();
};
```

MENU

→ 01

◆ 07

★ 12



# GAME的物件程式

在class Game 中執行了所有函數的初始化，而函數包含記錄遊戲得分、遊戲狀態(0 : normal, 1 : midterm, 2 : final)、遊戲背景區域生成以及處理玩家點擊等功能。

```
class Game

private:
    double gameScore;
    int status; // 0: normal, 1: midterm, 2: final
    double EnemySpeed;
    double EnemySpawnRate;
    Player *player;
    std::vector<Enemy *> enemies;

public:
    Game();
    void Run();
    void DrawBackground();
    void UserClick();
    void DrawDeadline();
    void UpdateInfoBar(double gameScore, int LeftTime);
    void DrawWhiteSpace(int a_x, int a_y, int b_x, int b_y);
    void EnemiesSpawn();
    void EnemiesMove();
    void ChangeStatus(int status);
    double GetEnemySpeed();
    double GetEnemySpawnRate();
    void ChangeEnemySpeed();
    void ChangeEnemySpawnRate();
    void BulletsOutOfBorderCheck();
    void ReadNextPage();
    int NewWindow(std::string file);
    void GameStart(std::chrono::system_clock::time_point &end);
    void Welcome();
    void GameOver();
    ~Game(){};
```

# 未來展望

## 1. 擴充不同的 Enemy 種類

我們Bullet 中的程式碼中有一個函數

**bool Hit(Character \*character);**

目前要判斷的種類只有一種 enemy，即「作」，但我們在函數裡面保留了 Character，作為未來能增加更多種類的 enemy，並且具有不同的屬性，像是觸發劇情等等。此外，在落下來的物件中，也可以新增一些對玩家有利的 Enemy，像是射到後可以清除周邊的其他 Enemy。

## 2. 增添各種遊戲玩法

在Player方面，期望之後可以擴充另外一個道具物件，在玩家進行遊戲前，可以選擇道具來協助玩家對付惱人的作業，例如雙槍、分身、放大子彈等等。

在遊戲機制本身，我們預計可以新增射擊子彈數量限制的玩法來增添刺激性。