

程式設計期末專案第四組

Dead By Deadline

組員

賴韻文 B12705003 徐郁翔 B12705027 梁德銘 B12705045 張佑丞 B12705055



一、遊戲摘要：

我們的專案叫Dead By Deadline, 是個深刻反映學期間作業量的射擊遊戲, 以及莘莘學子學期的心路歷程。在一分鐘(學期)內, 玩家(學生)需要射擊作業, 不要讓作業壓過死線, 才可以維持高GPA ㄟb

(一)遊戲緣起:「我是資管系的學生, 死線太多我要大爆設ㄟ」

主題:「這不只是一個遊戲專案, 這是我在這學期的心路歷程」

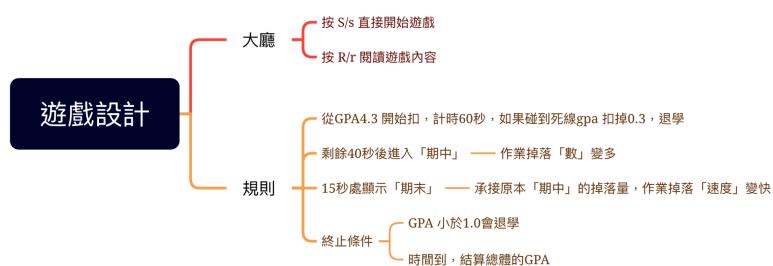
我是一個資管系的新生, 上程式設計之前我是真的以為大家都跟我一樣沒學過C++, 進來之後才發現只有我什麼都不懂。每個星期都像是經歷一場生死時速, 要趕在星期二之前設出來(程式), 隨着作業複雜度的逐漸提升, 我設(計)的速度也要隨之加快, 要不然就來不及啦!!!

(二)遊戲劇情:「在資管系絕地求生的經歷」

遊戲的主要目標是「生存」, 並盡量拿到更高的GPA, 正如我們當初進來資管系一樣, 隨着遊戲的進行, 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或已經被當了。在過程中, 玩家會遭遇事件「期中考」和「期末考」, 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。嘗試盡你所能去設計程式作業, 並努力活過這學期吧!

二、遊戲內容

(一) 遊戲架構



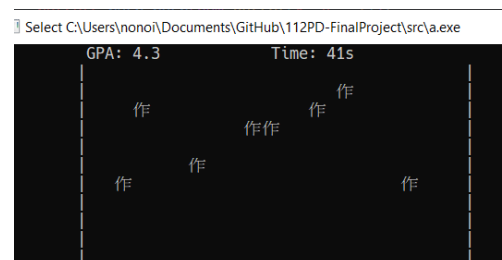
(二) 遊戲預覽



遊戲起始畫面

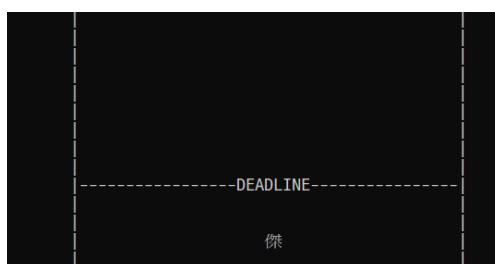
進入遊戲之後會進入到遊戲大廳，裡面可以選擇兩個選項：按 R/r 來閱讀劇情，或者按 S/s

以下以按 S 進行展示



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）。



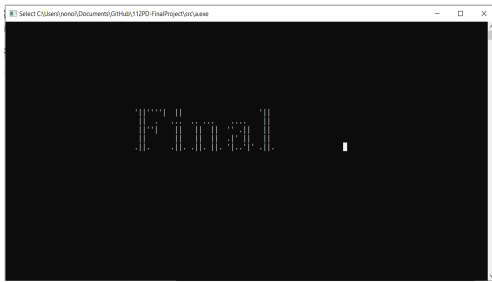
操作方式

點按左右鍵來移動，並利用空白鍵或者enter 鍵來進行射擊，將「作」消滅掉



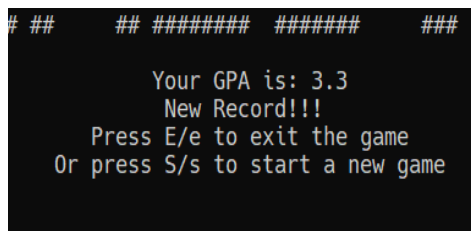
事件觸發：期中考

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

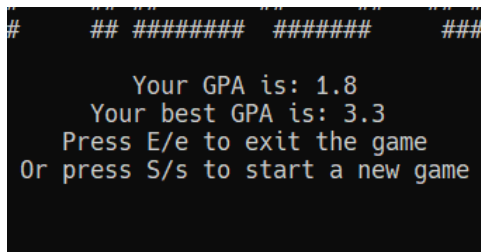


事件觸發：期末考

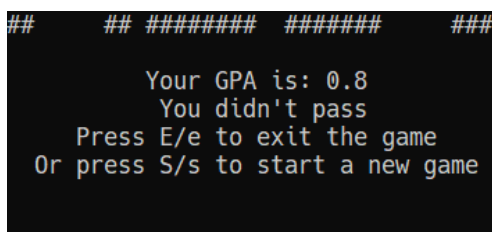
時間剩下20秒後會觸發「期末考」這個事件，會跳到Final這個頁面後，再讓玩家繼續遊戲。轉回畫面之後，「作」的數量和掉落速度會宇宙大爆炸地增加。



新紀錄！



紀錄先前比較高



在學期內(一分鐘內) 沒有

遊戲結束後結算畫面

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紀錄為何。

三、遊戲程式碼分析

(一)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 來確保檔案有被順利開啟。

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

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

(二)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：

```
const int SleepPerLoop = 20; // 每20毫秒就刷新，看使用者有沒有做什麼事情
const double PlayerSpeed = 4; // 速度的定義是看每一個20毫秒要動幾格
const double BulletSpeed = 0.8;

// 敵人的生成率以及移動速度
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;
```

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。

(三)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");

if (status == 0)
    return NormalEnemySpawnRate;
else if (status == 1)
    return MidtermEnemySpawnRate;
else if (status == 2)
    return FinalEnemySpawnRate;
else
{
    throw("status error");
}
```

(四) Inheritance, Polymorphism, etc.

1. **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(){};
};
```

2. **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
};
```

3. **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); };
```

4. **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();
};
```

(五) Game 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(){};
};

```

四、未來展望及心得

(一)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); // 判斷有沒有打到「作」
```

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。

(二)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。

(三)心得

梁德銘:透過這次專案, 我了解到原來程式設計課堂上教的東西都只是基礎, 實習中的很多東西都要靠自己透過不同的方式去自學, 例如window.h等等。看著遊戲從當初的一個小構思, 到最後真的實作出一個完整, 可以遊玩的遊戲, 看起來真的很有成就感, 當然在這裏就不得不感謝張佑丞, 製作遊戲的過程中, 由於我的程式底子比較差, 有很多地方我都要向他請教, 但他仍然不厭其煩地一一細心解答我的問題, 也很感謝我的其他組員, 感謝他們在過程中提出很多不同的意見, 讓我們在過程中透過不斷合作, 一起步步生出, 精進這個專案。

賴韻文:這次的程設專案讓我參與並學習一個遊戲(或小產品)從無到有的過程。首先感謝我的組員張佑丞很有效率地實作出了遊戲的雛形, 讓我們可以在初步架構下增加功能, 還有梁德銘構想了有趣的遊戲劇情, 讓遊戲體驗更完整, 以及最後有徐郁翔製造了精美的簡報, 盡力呈現給大家有趣的一面, 儘管它是個會扣GPA的遊戲。

在這個專案中, 我學習到最多的是團隊合作與溝通彼此的想法以達成共識。像是在討論遊戲功能和劇情的取捨時, 實作程式的工程師和創意總監會有不同考慮的地方, 要了解彼此認為重要的點才能進行整合, 執行決策。

張佑丞:其實在上這門課程以前,我比較少接觸到class的應用,也常常疑惑為何不直接寫一堆struct和function就好xd,但在這次專案中,如果真的像前面所說的直接寫function的話,程式碼不僅變得很冗長,整體可閱讀性也會下降許多。此外,我也實際運用了上課所學到的許多寫法(File I/O, Inheritance, etc...)讓我們在開發上能減少許多在define的困擾,讓大家只要寫好一個member functoin 的 prototype後 就可以大家分工下去做了!在遊戲玩法的點子上,也謝謝組員們提供許多很有趣的提案,尤其是發想出「大暴設」的Felix!

徐郁翔:我們這次所實作的遊戲,我覺得相較於其他組來說,可能相對精簡了一些,但是這個專案更能讓我了解到為什麼在第六週以後教的東西對於協作一個程式專案那麼的重要。在這次的專案中,我更了解這些模組化以及class、繼承、函數命名等較為軟性的程式基礎真的會大幅度影響別人閱讀程式碼的效率。這次的遊戲設計專案中,我也了解到我們的提案以及遊戲流程設計真的要仰賴一些圖示來讓所有組員都了解遊戲是怎麼進行的,並最後取得共識。最後,也要非常感謝張佑丞讓大家發想的點子可以順利得被實現出來,以及提供了很多的程式相關的建議!!

五、組員分工表

組員	工作內容
B12705003 賴韻文	製作書面報告、專案影片、提供程式碼意見
B12705027 徐郁翔	製作書面報告、專案簡報製作、建議程式碼
B12705045 梁德銘	製作書面報告、專案影片、提供程式意見
B12705055 張佑丞	協助書面報告、實作程式碼、完善程式架構