

程式設計（112-1）

期末專案

題目設計：孔令傑
國立臺灣大學資訊管理學系

在這個期末專案中，你和你的組員在給定的方向上，將針對一個自選的主題自行設計並實作一個資訊系統（或軟體，或程式，看你喜歡怎麼稱呼它）。你的系統可能含有一或多個程式。除了只能使用 C++ 以外，基本上沒有任何限制¹。我們期待看到的是 (1) 你們能設定一個有趣又不失挑戰性的題目，(2) 你們能分析問題並且制定出好的系統架構和分工，(3) 你們能編寫正確、有效且容易維護的 C++ 程式，以及 (4) 你們能清楚生動地展示你們的想法和成果。希望大家都能享受做專案的樂趣。祝大家好運！

1 分組

請大家自行分組，一組應該有三至四位修課的學生（旁聽生不計），組員不能跨班。一個同學不能參加兩組。因為專案的目的是讓大家練習團隊合作（或至少感覺到它的困難），因此有最低人數限制。組裡的人數不會影響評分，因此人少「可能」不如人多。如果你想徵求組員，可以上 NTU COOL 在一篇討論（屆時會看到）底下回文。如果你不想找組員，可以自行錯過分組期限，助教會幫你隨機分組。

2 任務

在期末專案中，請大家利用這學期所學到的知識和技能開發一個遊戲。遊戲類型不拘，可以是角色扮演、冒險、解謎、益智等任何形式²。主要的要求是：

1. 我們將提供大家一些與遊戲角色和遊戲物品相關的類別（class），大家必須使用這些類別（但可以做適當的修改）。
2. 需要運用到本項所列的技能至少三種：Operator Overloading、File I/O、Inheritance、Polymorphism、Template、Exception Handling。大家必須在書面報告中

¹以往有些同學會「混搭」，例如用 HTML 和 Javascript 做網頁，只有後端用 C++ 寫，這原則上是不可以的，因為這門課就是教 C++ 嘛，這也是沒辦法的事。如果還是發生了，教學團隊會根據使用 C++ 的比例將分數做折扣。總之如果想避免這種情況，就全部只使用 C++ 就好了。

²如果你願意，文字冒險遊戲應該很適合。以下是一些文字冒險遊戲的參考案例：Colossal Cave Adventure、萬王之王、命運石之門、Fate/stay night、文字遊戲。

說明在哪裡用到哪些技能，我們也會檢視大家的程式碼去確認的確有用到。

3. 一定要撰寫自己要使用的 header files，並且把 main function 和類別定義放在不同的 CPP 檔一起編譯。
4. 不用實作圖形化介面 (graphical user interface, GUI)、過場動畫和各類圖片。當然大家若願意那也很好，但請理解這只會讓分數有些微的增加。更具體地說，如果大家想要自學 <windows.h>，我們覺得很好，但如果大家想要自學 SFML 等圖形化介面函式庫，我們就覺得有點太多了。

以下是我們提供的類別，請注意如果你覺得合適，在維持類別的基本意涵及類別間的基本架構的前提下，你可以修改以下這些類別的任意內容：

1. 「角色」(character) 是所有遊戲中的角色 (包含敵我) 的基礎類別，裡面定義了所有角色都需要有的屬性和行為，但是是一個純粹抽象類別 (pure abstract class)：

```
class Character {
protected:
    std::string name;
    int health;

public:
    Character(const std::string& n, int hp)
        : name(n), health(hp) {}

    std::string getName() const {
        return name;
    }

    int getHealth() const {
        return health;
    }

    virtual void attack(Character& target) = 0;
};
```

2. 「玩家」(player) 是玩遊戲的人可以操作的我方角色，繼承「角色」這個類別，定義所有玩家的共同屬性：

```

class Player : public Character {
private:
    int level;
    int experience;

public:
    Player(const std::string& n)
        : Character(n, 100), level(0), experience(0) {}

    int getExperience() const {
        return experience;
    }

    void attack(Character& target) {
        // Perform player's attack logic
    }
};

```

3. 「敵人」(enemy) 是所有敵方角色，繼承「角色」這個類別，定義所有敵人的共同屬性：

```

class Enemy : public Character {
public:
    Enemy(const std::string& n, int hp)
        : Character(n, hp) {}

    void attack(Character& target) {
        // Perform enemy's attack logic
    }
};

```

4. 「物品」(item) 包含所有可以被獲得、儲存、使用的物品，定義所有物品的共同屬性：

```

class Item {
private:

```

```

        std::string name;

public:
    Item(const std::string& n) : name(n) {}

    std::string getName() const {
        return name;
    }
};

```

5. 「遊戲」(game) 本身也需要一個類別，隨著遊戲進行，會陸續新增敵人、物品，玩家也會跟這些敵人和物品互動：

```

class Game {
private:
    Player player;
    std::vector<Enemy> enemies;
    std::vector<Item> items;

public:
    Game(const std::string& playerName)
        : player(playerName) {}

    void addEnemy(const Enemy& enemy) {
        enemies.push_back(enemy);
    }

    void addItem(const Item& item) {
        items.push_back(item);
    }

    // Game logic methods
};

```

你可以根據上述的類別進行修改和拓展，自行設計遊戲世界、角色、劇情和遊戲機制，加入更多元素，豐富整個遊戲。例如：建立不同類型的物品，各有一個自己的類別，

並且繼承 Item；建立一個通用的函數模板，用於在遊戲中查找特定的物品；也可以加入例外處理機制，處理可能的錯誤情況；設計玩家與敵人的互動、攻擊、防禦等機制。

因為這是專案而非作業，細節就留給大家自行決定了。在選題目、設計和開發的過程中，歡迎隨時上 NTU COOL 跟大家討論，或者找助教或老師討論。你應該會需要許多課堂上沒教過的知識和技能。遇到這種情況，請自行用任何方法教會你自己吧；相信我，你可以的！

3 團隊任務

每一組必須做四件事：交專案計劃書、製作展示影片、寫書面報告，以及展示完成的系統給一位組員的雙親。此外，也要請每位同學對其他組的影片和自己的組員給予回饋。

1. 計劃書 (proposal)：在計劃書中，你們將簡單地描述自己打算做什麼。不需要寫很多，對某些題目來說五十或一百字說不定就夠了。在計劃書中，最重要的是寫下你的組員們的姓名和學號。如果你的組不滿三個人，而且願意接受被隨機分過來的人，也請在計劃書中註明。在截止時沒報名的人會被隨機分組。如果截止前沒有自己分組，被隨機分組後又不接受隨機分組的結果，就必須失去計畫書的分數。請在 **11 月 21 日 (週二) 23:59** 前繳交 PDF 檔到 NTU COOL。
2. 展示影片 (demonstration video)：
 - (a) 請錄製一個展示影片，並將影片傳到 YouTube。你可以設定你的影片為有連結的人才看得到（在 YouTube 這叫做 unlisted）。繳交的影片將給全班同學欣賞、互相觀摩。除了影片連結，各組還必須附上最多 100 字的專案摘要、一張系統執行螢幕截圖，以及製作影片時使用的投影片 PDF 檔，供班上同學快速瞭解影片內容。請在 **12 月 19 日 (週二) 23:59** 前把摘要、截圖、投影片包在一個 ZIP 檔中，並把該 ZIP 檔和影片連結分別繳交到 NTU COOL。
 - (b) 每組的影片長度以十分鐘為原則，稍微超過沒關係，超過太多者將被酌予扣分。請先口頭說明你們要做什麼、有什麼功能，接著執行你們的系統給大家看、盡量地展示它的功能。然後請說明你們的系統設計，包括類別、類別間的關係、使用到的指定 C++ 技能等等。最後，請說說你們的心得感想與未來展望。
 - (c) 各組可以自行決定由幾位同學製作、口頭報告，也可以自行決定報告者的臉是否要出現。但不管怎樣，請努力讓這個簡報顯得有趣且專業吧！

3. 書面報告 (written report)：

- (a) 在書面報告中，請介紹你的主題、系統設計、演算法以及分工方式。不要把原始碼複製貼上到書面報告中，那沒有意義，但對於特別重要的一段程式，你可以在書面報告中用 pseudocode 做介紹。在書面報告的最後，請每個組員都各自寫下一段簡單的心得感想，去描述自己在專案中學到的東西、體驗到的困難，以及其他任何想說的話。
- (b) 報告最多**八面 A4 紙**。在報告的第一頁，請寫上所有組員的學號、姓名，以及在繳交計劃書後被指定的組別編號。
- (c) 在 **12 月 22 日 (週五) 23:59** 前，將你的書面報告的 PDF 檔，以及你所有的程式碼都包成一個 ZIP 檔上傳到 NTU COOL。專案中用到的簡報檔、圖檔、影音檔、文字檔等等也歡迎上傳（如果檔案實在太多太大，可以另外聯絡助教或老師）。

4. 展示完成的系統給一位組員的雙親：

- (a) 好的資訊系統應該能被任何人看出其價值，包括從未寫過程式的人，以及年齡與系統開發者差距甚大的人。為了展現出你的系統像白居易的詩一樣老嫗能解，也為了增進親子情誼，請在完成系統開發之後，自行找時間地點向至少一位組員的父親和（或）母親做十分鐘的口頭報告與現場展示。如果有必要，視訊也可以³。
- (b) 請將十分鐘的口頭報告與現場展示過程全程側錄下來，接著讓受訪人以一至三分鐘說自己的心得感想，並將這一段心得感想也全程側錄。請不要太擔心音質與影質，用任意一台智慧型手機或電腦簡單地錄即可。
- (c) 在 **12 月 29 日 (週五) 23:59** 前，將側錄的影片合成一個影片並上傳到 YouTube（如果需要，可以設定為「unlisted」，讓有連結的人才可以看），然後繳交連結到 NTU COOL。

4 個人任務

每一位修課同學都必須挑選至少 10 組其他同學的作品做欣賞（可以跨班，亦即 01 班的學生可以看 02 班的作品，反之亦然），並在看完影片之後填寫表單提供評分與回饋。給每一組的文字回饋需要至少有 50 字。每一位修課同學也都必須針對組內所有其他同學的貢獻度給予評分與回饋，給每一位組員的文字回饋需要至少有 10 字。

³如果實在所有組員在期限內都無法對父親或母親做展示，請跟授課教師聯繫。

上述評分與回饋應在 **12 月 26 日 (週二) 23:59** 之前透過指定的表單上傳。這些評分與文字回饋都會被匿名地提供給各組做參考。授課團隊保證各組和每位同學收到來自其他同學的回饋時，不會看到給該評分和撰寫該回饋的同學的個人資訊。

5 評分標準

1. 計劃書 (10 分)：只要準時交，就是滿分；計劃書只是提供組員資訊，並且讓授課團隊知道你們的計畫（以便在必要時給點建議）。最後做的主題跟計劃書不一樣也沒有關係。總之，找到組員，寫一段話！
2. 使用者介面以外的系統功能 (25 分)：根據系統的價值和完整性（以遊戲來說，就是有多有趣、多豐富、遊戲性有多平衡、多引人入勝），授課團隊會給予評分。總之，選個好題目、準備好口頭報告和書面報告，讓大家知道你的系統有多棒。
3. 使用者介面 (5 分)：一個系統的使用者介面當然是重要的。雖然我們期待大家做的是文字介面的遊戲，但介面是否容易操作、是否清晰好用還是有意義的。請大家留意，如果有一組做了超棒的圖形介面，因為圖形介面不是我們期待大家做的，在這裡會多幾分，但最多也就 5 分而已了。
4. 系統架構與設計 (30 分)：根據你們的系統架構與設計（主要藉由書面報告呈現），特別是使用到指定 C++ 技能的數量和合適性，授課團隊會給予評分。那些 C++ 技能應該要被用在合適、合理、必要的地方，而不是一些可有可無的地方。總之，好好設計你們的系統，並且寫個正式、清楚且完整的報告來介紹你們的系統。
5. 向雙親（或目標使用者）展示 (10 分)：只要影片確實符合要求、時間大致是十分鐘上下、品質沒有過份糟糕，而且連結有在期限內正確地繳交，就是滿分。
6. 組間回饋 (10 分或更多)：一位同學若能確實完成指定數量的組間回饋（有給分數、字數足夠），可以得到 5 分，反之則為 0 分。超過指定數量後每多給一組回饋再加 0.3 分，在基本的 5 分之外最多加 6 分。若回饋亂寫被授課團隊發現，會被酌予扣分。另外 5 分則為其他組給這位同學所屬之組的分數平均。
7. 組內回饋 (10 分)：一位同學若能確實給所有組員回饋（有給分數、字數足夠），可以得到 5 分，反之則為 0 分。若回饋亂寫被授課團隊發現，會被酌予扣分。另外 5 分則為其他組員給這位同學的分數平均。

所有有截止時間的項目，若遲交在 12 小時內，成績會打八折做為懲罰，超過 12 小時但不滿 24 小時者打五折，超過 24 小時的以零分計。