

C++ Programming Project - NTU Monopoly

朱振瑀、黃聖翳、張鈺源

1. 前言

大富翁是一款家喻戶曉的經典遊戲。四名玩家的目標是保護自身的財務，同時通過購買和開發土地迫使對手破產。大富翁的遊戲規則簡介如下：

板子的每一邊分為 10 個小矩形，代表特定的財產、鐵路、公用事業、監獄以及其他各種地方和事件。遊戲開始時，每位玩家都會獲得固定數量的遊戲幣；爾後玩家們輪流根據擲出的一對骰子在棋盤上移動。若降落在無主的土地時，玩家可購買它。但如果降落在另一個玩家擁有的土地上，則必須向該玩家支付租金。此外，板子亦有機會與命運 spot。當玩家降落在此方塊時，需抽一張卡片，卡片可能為有利的（例如 earn money），也可能為不利的（例如 pay money）。

在思考要做什麼主題時，因為我們三個皆對大富翁頗有興趣，因此我們想把以臺大校園為主題的大富翁放到電腦上。

2. 主題簡介

遊戲大綱與大富翁類似，以下列出幾點臺大富翁特別之處。

- 1) 板子設計以魏宏宇教授的卡通圖代替大富翁的 mascot — Rich Uncle Pennybags。
- 2) 土地皆為臺大系館或周遭環境。
- 3) 不買 property，而是使用插「館長旗」以及「Super Big 旗」。
- 4) 機會、命運卡的設計為臺大學生的日常瑣事，如成為助教、腳踏車被偷等。

3. 物件

1) 人物



超笨的大笨鳥



比 Ya 的管中閔



嗯，我是熊仔



Random NTU Dude

2) 板子



3) 旗子

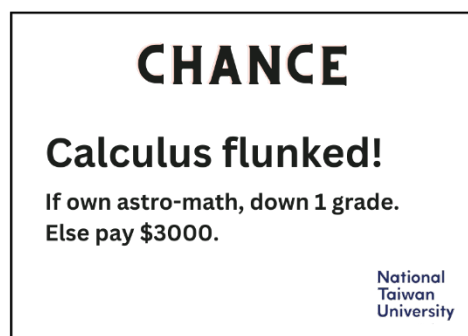


館長旗

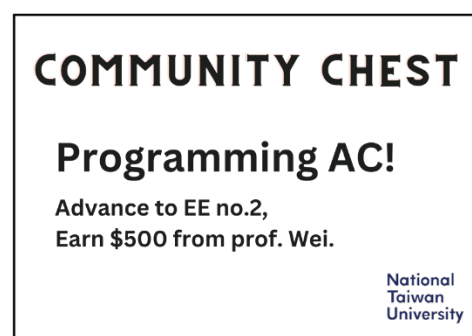


Super Big 旗

4) 機會、命運卡



微積分被當了！（機會卡）



計程 AC！（命運卡）

4. 程式執行說明

1) 遊戲初始化

一開始每位玩家皆擁有 25000 元，並且以公館捷運站為起點。



一開始顯示如下：

Turn: P1

Balance: 25000\$

Please press 'D' to roll dice.....

Pos: MRT Gongguan Station

2) 初始與一般行動

- 1) 我們沒有做出圖形觸控，因此都使採用鍵盤按鈕來玩遊戲。我們遊戲會引導大家按下正確的按鍵，像是，擲骰子，就會顯示 Please press 'D' to roll dice.....意思是按 D 來擲骰子。擲出的骰子點數會顯示在螢幕上，例如：You rolled 1，代表擲出 1。
- 2) 之後會顯示我們現在的位置，例如：Pos:Dept of Geoscience。如果這個地方沒人持有，而且有錢購買，接著顯示這個地方的價格，例如 Price:3200 以及詢問是否要購買，例如：Buy Dept of Geoscience?(Y/N)。此時就可以按照指示按下 Y/N 來告訴遊戲說是否要買這塊土地。
- 3) 當土地被玩家買下時，就會顯示 Purchased.。

- 4) 當沒錢購買這塊土地時，會顯示 Not enough money to purchase!
- 5) 如果走到的這塊土地被人持有，就會顯示
- 6) 走到事件格時，以玩家 4 為例，會顯示 P4 lost 1200\$
- 7) 走到水源阿伯時，會延遲一回合並罰錢。
- 8) 最後是顯示 Press' A' to continue，代表按 A 繼續，然後結束這一回合。

3) 事件格

- 1) 走到事件格時，以玩家 4 為例，會顯示 P4 lost 1200\$
- 2) 走到水源阿伯時，會延遲一回合並罰錢。而下一回合會顯示 P2 skips this turn



4) 過路費

- 1) 如果經過的土地有人持有，要依土地的價格以及土地的旗子數來決定赴多少錢給對方，版面顯示 You paid 410\$ to P1 之類的
- 2) 過路費價格為：

若無旗子，則為土地價格的 1/10 倍

若有一面館長旗，則為土地價格的 1/2 倍

若有兩面館長旗，則為土地價格的 $3/2$ 倍
 若有三面館長旗，則為土地價格的 3 倍
 若有四面館長旗，則為土地價格的 4 倍
 若有一面 super big，則為土地價格的 $11/2$ 倍

5) 命運卡

- 1) 命運卡著重在位置移動。
- 2) 當玩家走到命運格時，系統會自動抽出。
- 3) 抽出的卡不會再出現，直到整份卡牌全部抽完為止。



6) 機會卡

- 1) 和命運卡有點類似，只是機會卡出現位置改變的機率相對較低。
- 2) 當玩家走到機會格時，系統會自動抽出。
- 3) 抽出的卡不會再出現，直到整份卡牌全部抽完為止。



7) 買旗子

- 1) 當重複經過一塊買過的土地且有足夠錢時，就可以插旗子，以提高過路費的價格。當插滿 4 面館長旗時，可以升級成 super big 旗。
- 2) 插旗子時會顯示 Purchase a flag? (Y/N)，按下 Y/N 就可以決定是否買房。

8) 抵押

- 1) 當錢不夠時，可以抵押地，抵押可拿回的金額為原本價值的一半。
- 2) 系統會問你要不要抵押財產，要按就按 Y，不要就按 N。
- 3) 若不把財產抵押，系統會一直跳出財產，直到錢為正數為止。

9) 破產

- 1) 當所有財產都抵押玩且前為負數時，就是破產，也就是該玩家輸了這場遊戲。
- 2) 系統會顯示 P4 has no properties left! P4 is bankrupt!

10) 遊戲結束

- 1) 當只有一個玩家沒有破產時，遊戲結束。
- 2) 系統會顯示 Game Over! P3 has won!

5. 程式碼說明

1) Player.h

此 header file 中定義 player（玩家）這個 class，player 底下的 member variables 如 int money（儲存玩家現有的金額，初設為 \$25000），int pos（儲存玩家的位置，0~39，代表玩家當前在板子上的第幾格。例如，若位於起點，則 pos = 0），bool bankrupt（玩家是否破產，預設為 false），static int bankrupt_count（儲存共幾名玩家破產，預設為 0）等。而玩家的 member function 如 *int posx()* 以及 *int posy()*，此兩函數會根據玩家的 pos（也就是玩家在哪一格）回傳玩家所在的二維座標，方便要 render 玩家的 texture 時使用。

```
int player::posx()
{
    if (pos < 10)
        return (65 + 64*pos); // d=60
    else if (pos < 20)
        return 725;
    else if (pos < 30)
        return (65 + 64*(10 - (pos-20)));
    else
        return 15;
}
```

X 座標

```
int player::posy()
{
    if (pos <= 10)
        return 725;
    else if (pos < 20)
        return (625 - 64*(pos-11));
    else if (pos <= 30)
        return 15;
    else
        return (625 - 64*(39-pos));
    return 0;
}
```

Y 座標

2) block.h

此 header file 定義了 block 此 base class。所謂的 block，即是大富翁板子中的格子，共 40 格。Block 底下的 data members 有各種格子皆有的資訊，如 int pos（儲存該格是第幾格，0~39，起點是第 0 格），以及 string name（儲存該格的名字）。另外，我們也於 block 此 base class 中定義許多 virtual functions，如 virtual void block_function (player&)，即當玩家位於某一格時，會將該名玩家以 pass by reference 的方式傳入該格的 block_function，以對玩家執行該格的功能。

```
class block{
friend void chance1(player&);
public:
    block(int, string);
    ~block(){}
    int getpos () const {return pos;}
    string getname () const {return name;}
    static void InstantiateVector();
    virtual bool mortaged () {return 0;}
    virtual int getrent () {return 0;}
    virtual int getmoney() {return 0;}
    virtual void run_event () {return;}
    virtual void block_function(player&){return;}
protected:
    const int pos; // 0 ~ 39
    const string name;
};
```

Block 的 data members 與 virtual functions

繼承自 block 的 derived class 包含 property, event, corner, chance, community_chest，以下皆會針對這些 derived class 進行介紹。

3) Property.h

此 header file 中定義了 property 這個繼承自 block 的 derived class，其底下額外定義的變數有儲存各種價錢的 `const int price`（買地的價格）、`const int flag_price`（買一個旗子的價格）、`const int super_big_price`（將四個棋子換成一個 super big 的價格），`int lv`（儲存該地當前的等級，若無任何旗子則 `lv = 1`，每增加一根旗子則 `lv += 1`），`player* owner`（即指向地主 player 的 pointer，若該地未被任何玩家持有，則 `owner = NULL`）。Property 指的是大富翁板子中玩家可以進行購買或是蓋房子旅館（我們的版本中改成插旗）的土地（我們的版本中是設為台大校園中的各系館及公共建物）。Property 底下定義的 member function 有 `int getrent() const`，即根據該地當前的等級回傳過路費的值。至於 property 的 block function，運作方式大致如下：

當某玩家走到此格時：

I. 若此地無持有者：確認玩家是否有錢購買

- 若有，則詢問玩家是否購買
- 若是，則將擁有者設為該名玩家，並對玩家扣錢

```
if (yes || flag1)
{
    //update property texture
    game2_message = "Purchased.";
    if (!blk_run)
    {
        P.money -= price;
        owner = &P;
        yes = false;
        blk_run = true;
        flag1 = true;
    }
    game3_message = " ";
}
```

玩
家
購
買
土
地

II. 若該名玩家持有此地：確認玩家是否有錢升級

- 若有，則詢問玩家是否購買旗子
- 若是，則該地升級，多插一根旗子，並對玩家扣錢

```
if (yes || flag1)
{
    if (!blk_run)
    {
        lv++;
        P.money -= flag_price;
        yes = false;
        blk_run = true;
        flag1 = true;
    }
    //update property texture
    game2_message = "Upgraded to " + to_string(lv-1) + " flag(s).";
    game3_message = " ";
}
```

玩
家
購
買
旗
子

III. 若其他玩家持有此地：玩家付過路費給地主

```
else
{
    game2_message = "You paid " + to_string(getrent()) + "$ to " + owner->getname() + ".";
    if (!blk_printed)
    {
        P.money -= getrent();
        owner->money += getrent();
        blk_printed = true;
        blk_run = true;
    }
}
```


4) event.h、corner.h

此二 header file 分別定義 event, corner 兩個繼承自 block 的 derived class，其底下定義的 member variables 如 const int money、const int skip、const int tp（分別代表玩家走到此格時，對玩家銀行金額的增減、玩家須暫停的回合數、玩家須移動到哪一格（0~39））。Corner 是位於板子四個角落的事件格，event 則是位於板子其他地方的非機會與命運事件格。event 與 corner 的 block function 運作方式相似，大致如下：

當某玩家走到此格時：

- I. 若 money != 0：
給玩家錢或是扣玩家錢
- II. 若 skip != 0：
將玩家須暫停的回合數設為 skip
- III. 若 tp != -1: (-1: 該格不需改變玩家位置)
將玩家的位置設為 tp

```
void event::block_function(player &P){
    if (pos == 28){// student acc. fee
        if (P.paid_student)
            game2_message = "Talk to the hand!";
            blk_run = true;
            return;
        }
        game3_message = P.getname() + " realizes student association fees are useless.";
        if (!blk_run) P.paid_student = true;
    }
    if (!blk_run) P.money += money;
    game2_message = P.getname() + " lost " + to_string(-money) + "$.";
    if (tp != -1)
        if (!blk_run) P.pos = tp;
        game3_message = "Tp:" + board[tp]->getname();
    }
    if (skip)
        if (!blk_run) skips = 1;
        game3_message = P.getname() + " will stop for one turn.";
    }
    blk_run = true;
}
```

5) chanceandchest.h

此 header file 定義了 chance, community_chest 兩個繼承自 block 的 derived class，兩者皆未定義額外的 member variables。Chance 代表大富翁板子上的機會格子，community_chest 則是命運格子。兩者的 block function 運作方式即玩家走到時抽一張機會卡或命運卡，並執行該卡的功能。

6) card.h

此 header file 定義了 chance_card、community_chest_card 兩個 class，分別代表機會卡和命運卡。每張卡片會有個 int num（儲存該卡的代號），且兩個 class 底下皆有定義名為_deck 的 static vector，分別代表機會卡以及命運卡的卡堆，在 chance_card 以及 community_chest_card 的 constructor 中會將卡片自動加入到卡堆

中。每一張機會與命運卡皆有個別定義的 card function，card function 可能會改變玩家的銀行餘額、暫定回合數、位置，以及會將該卡的 texture render 到畫面中。

7) data.h、text.h

data.h 中宣告了 40 個大富翁格子的屬性，以及 4 名玩家。text.h 則宣告 LTexture 這個 class（參考自 lazyfoo），以及宣告儲存各種 texture 的變數。

```
player P4(2500000, "P4");
corner gongguan (0, "MRT Gongguan Station", 5000);
property geo (1, "Dept of Geoscience", 3200, 2500, 2500);
```

data.h

```
class LTexture
{
public:
    //Initializes variables
    LTexture();

    //Deallocates memory
    ~LTexture();

    //Loads image at specified path
    bool loadFromFile( std::string path );

    #if defined(SDL_TTF_MAJOR_VERSION)
    //Creates image from font string
    bool loadFromRenderedText( std::string textureText, SDL_Color textColor );
    #endif

    //Deallocates texture
    void free();

    //Set color modulation
    void setColor( Uint8 red, Uint8 green, Uint8 blue );
```

text.h 之
LTexture class

8) main.cpp

主程式中除了建立視窗，載入 texture 外，最重要的即是在遊戲結束前反覆執行的 void turn (player&) 函數。Turn 函數中又包含三個子函數，分別是顯示玩家資訊的 *display()*，處理擲骰子以及玩家移動的 *roll_dice()*，以及檢查玩家是否需抵押財產以及是否破產的 *mortgage_check()*。Turn 函數的運作方式大致如下：
（display 函數會不斷更新要顯示的遊戲資訊）

- I. 若玩家已遭淘汰：結束回合。
- II. 若玩家該回合暫停：玩家須暫停的回合數-1，結束回合。
- III. 當玩家按下 D（擲骰子的按鍵）：

roll_dice（隨機產生介於 1~6 的整數，並使玩家移動到目的地。若玩家於此時經過起點，則給予玩家 5000 元）、執行玩家所在位置的 block_function。

IV. mortgage_check:

- 當玩家的餘額<0：檢查玩家是否仍有財產可抵押

若是，則依序詢問玩家是否抵押某財產（按 Y/N 決定），直到玩家的餘額不再為負或玩家已無剩下財產可抵押

若否，宣告玩家破產，結束函數

- 結束回合

<pre>void turn(player &P){ if (P.dead){ turn_complete = true; return; } if (P.skip != 0){ turn_message = "Turn:" + P.getname(); money_message = "Balance:" + to_string(P.money) + "\$"; game1_message = P.getname() + " skips this turn."; over_message = "Press 'A' to continue....."; if (enter){ P.skip --; turn_complete = true; } return; } }</pre>	<div>玩家已淘汰</div> <div>玩家暫停該回合</div>
<pre>if (P.bankrupt){ turn_message = "Turn:" + P.getname(); money_message = "Balance:" + to_string(P.money) + "\$"; dice_message = "You rolled " + to_string(dice); game1_message = "Pos:" + board[P.pos]->getname(); mortgage1_message = P.getname() + " has no properties left!"; mortgage2_message = P.getname() + " is bankrupt!"; over_message = "Press 'A' to continue....."; if (enter){ turn_complete = true; player::bankrupt_count++; P.dead = true; } return; }</pre>	<div>玩家破產</div>

主程式中的 while loop 中主要即是 render 各 texture，如遊戲板子、玩家的 figure（若玩家未破產，則根據玩家的 *posx()* 以及 *posy()* 函數的回傳值進行 render）、旗子（檢查每格 property 的等級，並根據等級 render 指定數量的館長旗或 super big）、game message（即是遊戲資訊，如當前是誰的回合、玩家的餘額、玩家擲骰子擲到多少、遊戲中發生的事件……），以及執行 turn 函數。要 pass 進去 turn 函數的玩家會取決於 *who* 這個變數。*who* 最初為 0，每執行完某玩家的回合則 *who* += 1，若 *who* = 4 則將 *who* 設為 0。而也有另一個 `vector<player*> player_list`，依序儲存 P1、P2、P3、P4 四名 player 的 pointer。因此，若 *who* = 1，代表現在是 P2 的回合，`turn(*player_list[who])` 即是將 P2 傳入 turn 函數進行運算。

此外，因 turn 函數以及其中的小函數要在 while loop 中反覆執行，因此在程式中有宣告許多 flag（這裡的 flag 指的是控制程式流程所用的布林值，而非我們遊戲元素中的館長旗！），這些 flag 可以控制 turn 中的某些部分，使他們每一回合只執行一次，例如玩家的移動或是改變玩家的金額皆僅需執行一次，否則玩家將會無限次數地在

board 上繞圈圈（因為若骰子擲到 6，玩家每經過一次 while loop 都會走六格，但我們希望達到的結果是該回合玩家走 6 格），且玩家的金錢可能無限上漲或下跌。

〈註〉以上文字的程式碼較長，請助教們參閱 main.cpp 的 code

5. 未來展望

本程式使用 OOP 編寫，結合 SDL，使原始碼可做簡單的視覺化設計。

由於我們整組都是 C++ 的初學者，要把程式的邏輯運算寫好已非易事，更不用說程式的視覺化。因此，本此我們的程式中並未加入視覺特效、動畫，或是音效以及背景音樂，介面也是僅採用文字顯示遊戲資訊。希望若有機會，我們能將 SDL 或是其他相關的 library 徹底學好，以讓我們的遊戲不只是能玩，也要能帶給使用者視覺以及聽覺上的享受。此外，我們的大富翁或許對使用者來說友善度可以加強。本次遊戲的操作方式為按鍵盤上的 A、D、Y、N 鍵，若能在介面中加入按鍵供使用者以滑鼠點選，或許玩起來會更加方便。

再者，我們並未提供使用者可以隨時查看自己財產以及哪塊地的擁有者是誰的方法，或許下次製作時，可以考慮加入上述這些元素。此外，這次尚未開發的 dice animation，這也是將來想優化的部分。

總而言之，希望我們製作視覺化遊戲的實力增強後，未來至少能自行做出如同網頁遊戲般的水準。

以上是我們的臺大富翁 NTU Monopoly，謝謝。

6. 分工表

組員	工作
朱振瑀	程式設計、debug、錄 demo
黃聖翳	遊戲架構設計、製作 ppt、上台報告、上台影片剪接
張鈺源	美編設計、介面設計、report