

Project 2 - Chess

1092 OOP projects

Duration - 5/03 ~ 5/17(—)

Outline

Introduction

Classes

Game mechanism

Example

Grading

Q&A

Introduction

- Chess 是一種棋盤遊戲，棋盤是由黑白相間的8x8網格構成，雙方各有16個旗子，一國王、一皇后、兩城堡、兩騎士、兩主教和八士兵，各具不同功能與走法，目標是將對方的國王處在不可避免的威脅之下以將死對方，也可以通過對方自知無望、主動認輸而獲勝，另有相當多的情況可導致和局。--- Wikipedia
- 本次要實作一款可以遊玩的 Chess，至少有**操作、輸贏判斷(將死、投降)、畫面顯示**。

Classes

遊戲由下列物件構成:

1. GameManager - 控制遊戲盤面、遊玩狀態
2. Player - 玩家
 - a. Human player, AI player
3. Viewer - 將棋盤顯示到畫面上

檢查要看code, 只看Class **declaration**, 看得出來有使用到以下規格即可算分。

Class GameManager Specs

管理遊戲流程

```
class GameManager{  
    Player* players[2];  
    int current_player;  
    Board board;  
    Viewer viewer;  
};
```

Class GameManager Example

根據遊戲生成 player

```
player[0] = new HumanPlayer(...);
```

```
player[1] = new AIPlayer(...);
```

Class GameManager Example

```
if(state == MOVE_PIECE){  
    player[current_player]->OnMove(...);  
    // do something ...  
    if(promote) { // 移動後升階  
        player[current_player]->OnPromote(...);  
        // do something ...  
    }  
    // 1.Display, 2.Check gameover, 3.Switch player  
}
```

Class Player Specs

使用 abstract class Player, 產生不同種類的 Player來 overrider 這些函式

```
class Player {  
    public:  
        virtual void OnMove(/*ex. Board& const board, Position& outFromPos,  
Position& outToPos*/) = 0; // 選擇要從 fromPos 走到 toPos  
        virtual void OnPromote(/*ex. Board& const board, Position& const  
pawnPos, PieceType& outType*/) = 0; // 選擇升階類型  
};
```


Class Player Specs

```
class HumanPlayer : public Player{  
    public:  
        virtual void OnMove(/* parameters */) override;  
        void OnPromote(/* parameters */) override;  
        // your code...  
};  
  
class AIPLayer : public Player {  
    // same ...  
};
```

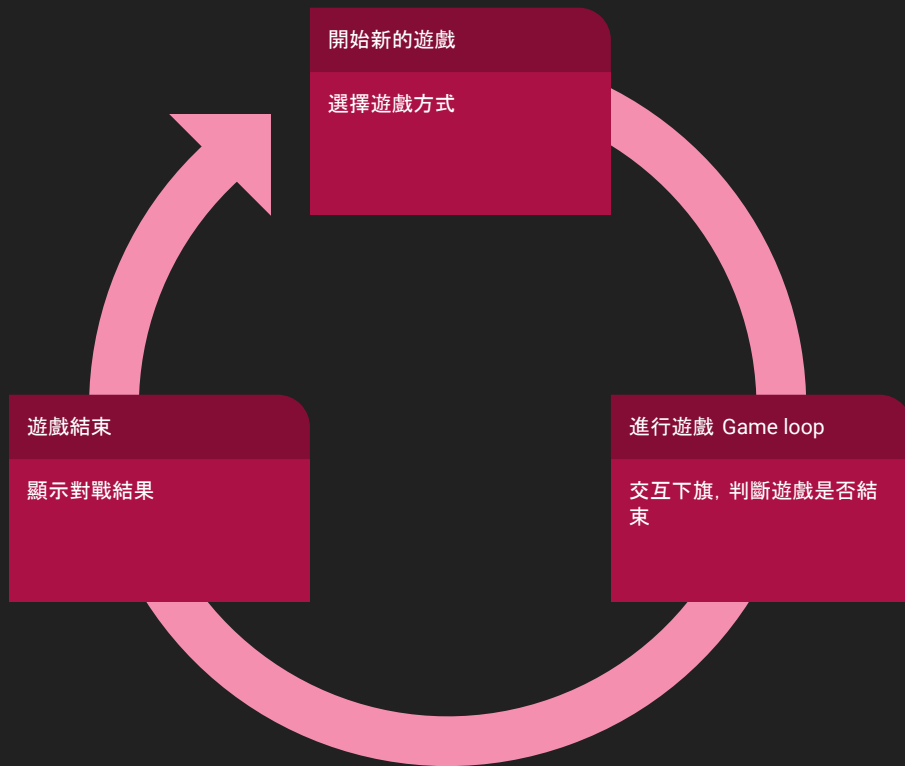
Class Viewer Specs

不限, 本 class 旨在作為專門呼叫顯示用api (cout, printf, GL, Qt, FLTK...etc.)。

讓 GameManager 透過 viewer 來顯示而不是直接呼叫顯示用 api 即可。

Game mechanism

遊戲流程



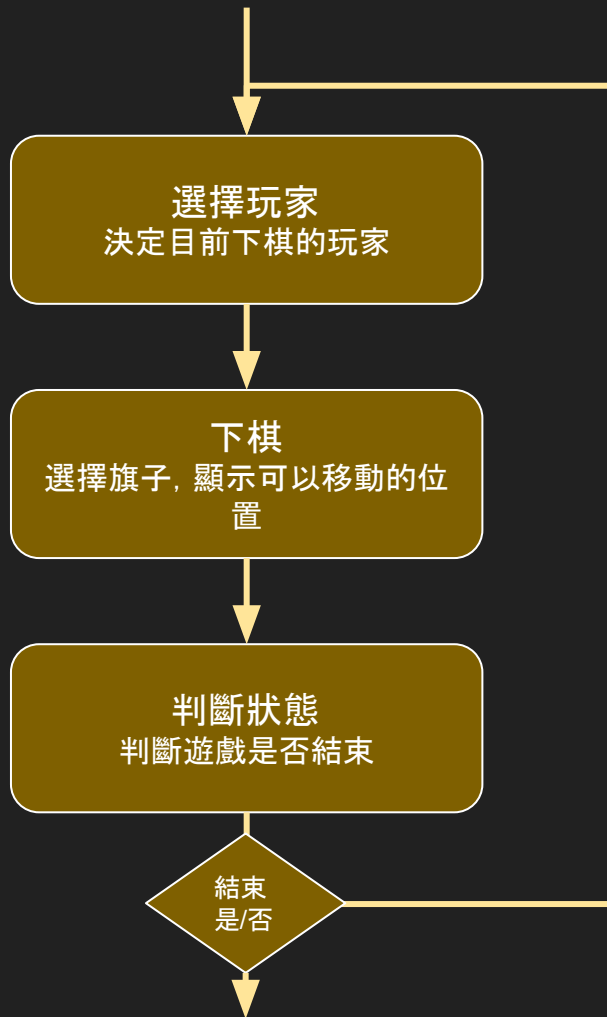
Game mechanism

開始新的遊戲

- 選擇遊戲方式 ex.
 - Local match
 - Online match - 先手、後手
 - Against AI

Game mechanism

Game loop



Game mechanism

遊戲結束

- 顯示結束資訊
- 再來一局
 - 計分板

Game mechanism

Rules

棋子移動規則參照

[國際象棋/規則 - 維基教科書, 自由的教學讀本 \(wikibooks.org\)](https://zh.wikipedia.org/wiki/國際象棋/規則)

遊戲結束規則, 使用:

1. 將死
2. 認輸
3. 時間到
4. 和局 - 非將死狀況下無子可動

Grading

Project 2 Grading Sheet

<https://docs.google.com/document/d/1CL7Kb4jdW-jbAYc8ttuRG8x4NSjw4U4Fg88Fa6Xt4nY/edit?usp=sharing>

Q&A

Deadline - 05/17 (一)

Project 2 Question Sheet

<https://forms.gle/H111snW2SZeF5jtt8>

Responses

https://docs.google.com/spreadsheets/d/1hQfxMsZrK3Fbp06_y2fQkkZqZJmj_-DxIP3EjbwWiNU/edit?usp=sharing