

(1) the team members' names and school IDs
B07902108 翁祖毅

(2) how you divide the responsibilities of the team members
100 % myself

(3) the relations between the classes that you design

我設計一個線上多人俄羅斯方塊遊戲，分為前後端，後端處理方塊的移動、轉動、計分等規則，前端負責連線、GUI 等事項。

class Main: game entry point, 在此定義各種形狀的 tetrominoes, 以及各種移動 tetromino 的規則。呼叫 class TetrisServer 建立一個多人 tetris game server。

class TetrisServer: 建立 server socket, 處理 telnet 進來的 client, 並為後者建立一個 thread, 即是 class ClientThread, 並與之交流 client 輸入的資訊, 以及 server 輸出的資訊。一旦 client 輸入名稱後, 建立一個 class TetrisGame 的 instance, 後者即是後端, server 會對其輸入 client 對 tetromino 的操作指令, 並且把後端的運算結果返回給 client。

class ClientThread: 處理與 client 之間的IO。

class TetrisGame: 遊戲的後端, 擁有許多 class Player 的 instance, 代表每個玩家的資訊。此外定義了整個遊戲的流程, 並且負責修改或取得以下所有物件的狀態。

class Player: 每個 instance 擁有一個 class Board 的 instance, 代表該玩目前的遊戲版面狀態。

abstract class Tetromino: 定義一個 tetromino 由許多 class TetrominoCell 的 instance 組成, 以及如何在 class Board 上放置/移除。

class *Tetromino: extend abstract class Tetromino, 定義各種不同形狀與參考點。為 OCP, 可以加入自己想要的 tetromino !

abstract class ActionTetromino: 定義 tetromino 在 class Board instance 上的操作。

class Action*: extend abstract class ActionTetromino 定義各種不同的 tetromino 操作, 有些允許 Player 使用, 如移動、轉動等。

class PlayerTimeoutEvent: 由 class TetrisGame 產生其 instance, 目的在於定期觸發 tetromino 的下落。

class GUI, class ScoreEntry: 前端的輔助物件。

(4) the advantages of your design

後端分工明確, 甚至可以加入自己想要的 tetromino 的形狀。前後端分離使得可以設計不同的 GUI 介面。

(5) the disadvantages of your design

還有很多bug尚未修復, 由於時間問題, 很抱歉期末太趕了。

前端的部分處理 race condition 採用大量 synchronized function, 應該會有更好的寫法, 目前看起來就像一碗義大利麵, 一些 class 互相呼叫, 分工不太明確且複雜, 易有 bug。

(6) other packages that you have used : no

(7) how to play your 2D game

server: make run [port]

client: telnet [server-ip] [port]

規則即是我們一般熟知的俄羅斯方塊基本規則(沒有T轉, 那太難了=)

keyboard z, x: 逆時針、順時針旋轉

keyboard Left, Right, Down: 往左、往右、直接落下