

Introduction of Simulation on shiny

ui.R

網站的畫面呈現，主要分為兩個部分(兩個 navbarpage)，雙人與多人模擬。
其中紀錄了股市模擬所有需要的輸入選項與輸出圖表名稱，由 server 端以 id 接收。

server.R

網站的模擬計算處理，記錄按鍵被點擊後的動作，以"input + \$ + id"取得輸入內容，
呼叫遊戲模擬、傳入參數，並再次回傳資料到畫面上。

global.R

讀取套件，需要的程式碼等，用於雙人遊戲。

global_2.R

讀取套件，需要的程式碼等，用於多人遊戲。

game.R

模擬的核心程式，game 本身為一物件。讀入網站畫面上輸入的資料(initialize)，
讀取前必須先創立變數，以"self + \$ + 變數"來做處理，只要需要在其他物件做取
用就必須這樣做設定，若只是特定區塊需要計算使用，則可用區域變數(不用加
self)。

主要有初始化(initialize)、遊戲進行(playing)、模擬(simulate)。

遊戲進行會創建兩個玩家物件、一個市場物件，呼叫玩家執行決策並呼叫市場計
算，最後呼叫玩家結算資訊，重複回合數(100)次，將資料儲存。

模擬會重複執行遊戲(playing)輸入的次數，將所有資料儲存再回傳至 server。

game_2.R

供多人玩家使用，主要一樣有初始化(initialize)、遊戲進行(playing)、取得所有玩
家資訊(getPlayerdata)、模擬(simulate)。

遊戲進行會創建多個玩家物件、一個市場物件，呼叫玩家執行決策並呼叫市場計
算，最後呼叫玩家結算資訊，重複回合數(100)次，將資料儲存。

取得所有玩家資訊目前使用在取得所有玩家該回合的決策，以供市場做波動計算。

模擬會重複執行遊戲(playing)輸入的次數，將所有資料儲存在一起回傳至
server。

player.R

為一玩家物件，主要有初始化、決策(decide)、結算(ending)。

決策會先判斷是否為 noise，再判斷是否有 anchoring，再判斷是否為 probability 模式，根據玩家類型做不同的機率運算；最後則為 internal goal，機制與過去討論的架構相似，不在此檔案贅述。

player_2.R

與上述 player.R 相近，僅在 internal goal 中不會因對手決策不買或不賣，而調整自身買賣比率。

market.R

為一市場物件，用於雙人遊戲。主要有初始化、市場狀況(condition)、每回合計算(game)。

市場狀況會判斷現在為 balance, bubble 或 burst。

每回合計算會取得兩位玩家決策，進行市場當回合波動計算。

market_2.R

為一市場物件，用於雙人遊戲，與 market.R 相近。主要有初始化、市場狀況(condition)、每回合計算(game)。

每回合計算與雙人遊戲為市場波動的計算方式不同，會加總所有玩家的買賣張數，根據淨張數與總張數比率做相對應的波動調整。

rsconnect

為上傳至 shiny 所生成的必須檔案資料夾。