

San Juan

一、總覽

- 程式碼由上至下的函式分別為：
 - safe_flush：清除 scanf 的緩存
 - yourans：使用者輸入字串轉數字
 - setCardNum：亂數模擬抽牌
 - com_level2：在建築階段，當選擇 Com 是 level 2 時會呼叫此函式
 - builder：建築階段
 - producer：生產階段
 - trader：商人階段
 - prospector：掏金者階段
 - counciler：市長階段
 - printallcom：列出 3 個 Com 的建築
 - printplayer：列出使用者的建築和牌
 - finalvps：遊戲結束後計算並顯示加上特殊牌的得分、顯示冠軍
 - main：選擇 Com level，亂數抽 governor 和 Com 亂數選行動卡（例如：建築師）

二、變數

- 幾乎是全域變數
- 所有可以找出關連的變數都是用 array 連接

有關牌的 array 有 6 個，分別為名稱、價格、得分、使用的階段（e.g. builder phase）以及每種牌的個數（110 張牌分為 29 類）。

與 4 個角色（包含使用者）有關的 array 有 8 個，除了名稱、分數、得到的牌總數、建築數、貨物數，還有用二維陣列表示手上有哪些牌、有哪些建築、哪些建築有生產貨物。

三、主函式(main)

- 初始化遊戲：每人都先有一張 Indigo plant
- 將空的 array 的值從 0 改設為 30，因為 0 是 Smithy（特殊建築卡）
- 呼叫 setCardNum()讓每位玩家一開始有 4 張牌
- 選擇 Com 的 Level
- 印出遊戲畫面

在印出遊戲畫面後都要按 enter 確定使用者知道現在進行到哪一步。

- 抽選 governor，Com 亂數選而使用者輸入要進行的階段
每個階段輪四個人後換下一個 governor，順序是：Player → Com1 → Com2 → Com3 再回到 Player 的迴圈，角色選到的每個階段都會呼叫外面的函式處理。
- 判斷是否有玩家蓋 12 棟建築物，有就呼叫 finalvps() 並結束遊戲
- 印出一回合後的遊戲畫面
- 判斷每一回合（同一個 governor，四階段都完成）結束後是否要使用特殊卡（Tower, Chapel）

四、setCardNum

- 亂數抽牌
亂數除以牌的總數再從第一種牌的個數開始減，減到等於零時就表示抽到那張牌，並將該種牌的個數減一，牌的總數減一。
- 沒有牌時
將被丟棄的牌（以 discard[110] 儲存，一張張加回 piece[] 對應種類的牌，再將被丟棄的牌總數（discard_sum）加到牌的總數（sum）中。

五、com_level2

- 在建築師階段中，判斷 Com 的牌是否有可以生產或最後結束時可加分的牌，沒有就回到一般的建築方式（見 builder）。

六、builder

- tobuild 表示要蓋 player (p) 的第幾張牌
- 判斷如果 p 是 governor，可減一張牌的費用
- 如果 p 手上牌的總數為零，表示不能蓋建築，直接跳出
- 判斷是否有 Black market 的建築而且有生產貨物
如果有，使用者可以選擇是否要用貨物抵費用，Com 會用最多兩個貨物來抵費用。
- 判斷是否有 Crane 的建築
如果有，使用者可選擇是否要用 Crane 的功能、用了之後要覆蓋在哪張建築物卡以及可抵多少費用；Com 則是會隨機選一個建物覆蓋上去。
- 選擇要蓋哪張卡

使用者在選擇後才會判斷可用哪種建築的特殊功能抵費用，所以可能會出現選擇後輸出 input error 的狀況，如果每張卡都不行蓋可選 0 退出。level 1 的 Com 會從最後一張卡片開始判斷能否蓋，不行就在往前找直到小於 0；level 2 的 Com 會傳至函式 com_level2，都沒有再回到 level 1。

- 建造特殊建築且有 Carpenter 建築可加 1 手牌
- 將卡牌建造成建築

如果有使用 Crane 且成功選到能變建築的卡，就會將原本的建築丟棄，減去原本建築所獲得的分數，如果有貨物也會消失，費用會變成減掉原本建築的費用；沒有 Crane 則是建在最後，增加建築數。兩者都會加得分，將卡從手牌刪除。

- 用手牌付費

如果費用小於等於 0，不需要支付手牌，判斷是否有 Poor house 且符合條件就加手牌，然後結束。

如果費用等於手牌，將手牌全部丟棄，手牌總數設為 0。

如果以上兩者都不符合，使用者可選擇要用哪些手牌支出，Com 則是由後往前支出手牌。

- 判斷是否有 Poor house 且符合條件就加手牌

七、producer

- 計算該玩家最多可以生產多少貨物

Governor、Aqueduct、Library

- 計算現在有多少生產建築 (production)
- 計算可以生產多少建築

可生產值 = production-product_sum[p]

如果可生產值大於等於最大可生產值，會全部生產。

如果可生產值大於最大可生產值，就需要選擇生產。

- 選擇生產

使用者可選擇要在哪些尚未有貨物的生產建築生產，或選擇 0 不生產。

Com 會從費用較高的生產建築開始生產。

- wantproduce 表示實際生產數

輸出最終該玩家生產多少貨物，如果大於 1 個會在判斷是否有 Well 建築，可以增加一張手牌。

八、trader

- 計算該玩家最多可以販售多少貨物
Governor, Library, Trading post
- 販售貨物
使用者可選擇要賣幾個貨物或不賣；Com 會從建築費用較高的貨物開始販售，存取販售後能獲得的牌數（金錢）。
- 增加手牌的特殊建物卡
Market stand, Market hall
- 全部結束後再抽牌

九、prospector

- 計算能抽幾張牌，並先抽牌
Governor, Library
- 玩家有 Gold mine
會亂數抽 4 張牌，如果有任 2 張是相同費用，要全數丟棄；如果費用都不相同，使用者可以選擇要哪張牌，Com 會選擇第一張，並將剩下三張牌丟棄。

十、councilor

- 計算能抽幾張牌(draw)
Governor, Library
- 計算能選幾張牌入手牌（getcard）
Prefecture
- 從抽到的牌中選擇要入手的牌
使用者如果有 Archive 可以從原本的手牌加上抽取的牌中抽出(手牌 + getchar)，沒有的話就是從抽到的牌開始抽選；Com 無論是否有 Archive 都會從抽到的牌開始抽選。

十一、finalvps

- 計算最終得分，如果有特殊的牌會再加分
Guild hall, City hall, Triumphal arch, Palace
- 印出所有人的得分和冠軍