**我如何設計結構：**

|  |  |
| --- | --- |
| **玩家：**  typedef struct \_player {  card cards;  int number\_of\_cards;  int buildings[12];  int numbers\_of\_buildings;  int point;  int product\_limit;  int card\_limit;  c buildingInfo[12];  int numbers\_of\_products;  special\_effective specials;  int card\_under;  } \_player; | 1. 卡牌部分：用linling list儲存卡牌資訊 並記錄當前卡牌數，以及數量上限。 (刪除及增加卡牌比較方便) 2. 建築部分：用array存取卡牌編號及其他資訊(如：是否有貨物)，並記錄擁有的建築數量，以及當前貨物的數量。 (建築上限固定，用array較方便增加及記錄建築資訊，遇到起重機時也方便替換，紀錄貨物數量及卡牌資訊有利於在遇到生產者及商人時可直接使用) 3. 特殊卡牌功能：用一個union儲存有哪些特殊建築，其功能可以開始使用。 (建造建築時可直接藉由bit shift 及卡牌編號將該卡牌的功能開啟，也方便判斷使否已經建有此建築) 4. 紀錄有多少牌在禮拜堂下： 因遇到起重機時，在禮拜堂下的卡牌仍保留，且不會有其他行為需要紀錄卡牌是被放在哪張牌之下，因此直接記錄在玩家的struct中。 |
| **卡牌：** typedef struct {  int id;  char en\_name[20];  char ch\_name[20];  int pay;  int point;  char effective[600];  } cardSet;  cardSet cardType[29] | 用一個結構陣列紀錄卡牌資訊，這樣建造或是讓玩家選擇保留哪些牌時，只要記錄卡牌的編號，就能藉由此陣列取得其他卡牌資訊並使用或輸出。 |

**我如何設計我的程式：**

玩家在做完前面level的選擇後，進入遊戲主體。  
之後進到一個無窮迴圈(計算總督是誰)，再進到另一層for迴圈(每個玩家選角色)，之後再呼叫對應的角色的函式，使各個玩家執行該角色的特權跟行動。

較特殊的卡牌，如禮拜堂則是每回合檢查一次是否已建造。同業會館或宮殿等則是等到遊戲結束計算分數才檢查是否已建造。其他特殊建築，則是跟隨相應的行動檢查並發揮效用。

**流程圖：**