

DS Assignment Final Project Report -Rummikub

系級:電機 4B 學號:108501537 姓名:蔡雨蓁

系級:電機 4B 學號:108501025 姓名:沈冠璋

1. Build Guide

- gcc -o aa main.c card.c

2. Execute Guide

- ./aa

3. System Architecture

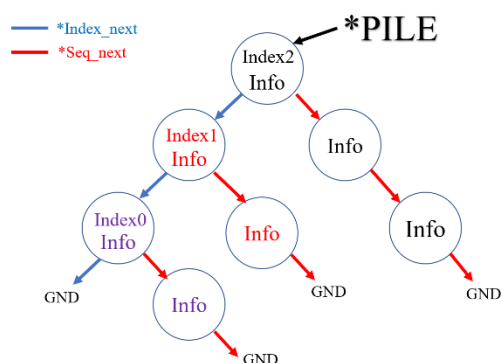
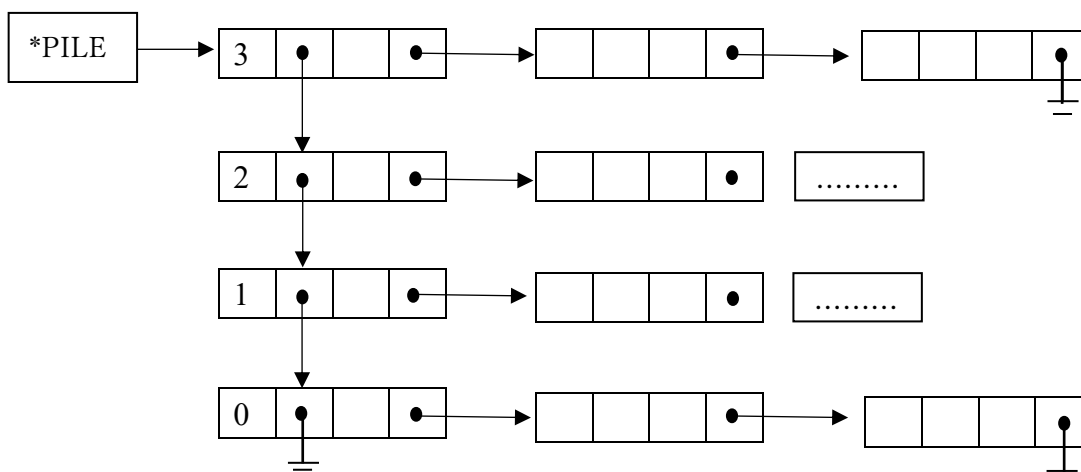
- Data structure of the field card node

```
struct CARDPILE
```

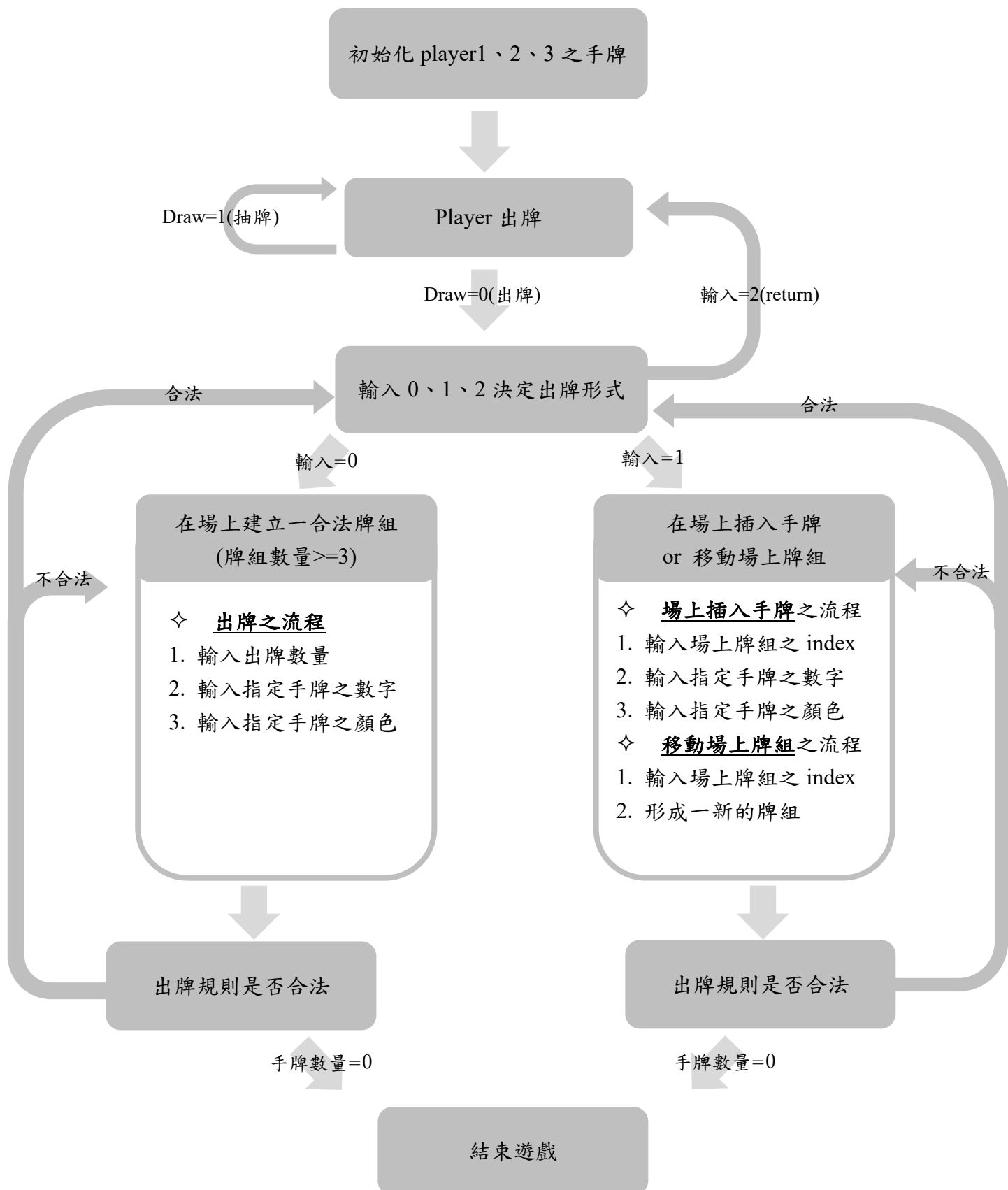
index	Index_next	info	Seq_next

- Data structure of the field card Architecture

- ✧ 用 linked-list 將場上的牌串聯起來
- ✧ 每個 node 皆有兩個 pointer(*Index_next、*Seq_next)，因此此架構又類似於 Binary Tree，因此在搜尋與複製時，皆可用此 BST 之方式



4. Flowchart



5. Function Explanation

PART1. 和手牌處理相關之 function	
shuffle()	<p>輸入參數: 牌堆</p> <p>將抽卡牌堆初始化並隨機打亂，打亂方式為將每張牌與隨機一張牌做交換</p>
deal()	<p>輸入參數: 手牌、抽卡牌堆、洗亂牌堆</p> <p>將洗亂的牌發給玩家，並把剩下的放入抽卡牌堆</p>
drawcard()	<p>輸入參數: 手牌、抽卡牌堆</p> <p>POP 抽卡牌堆一張牌，PUSH 進玩家手牌</p>
showcard()	<p>輸入參數: 手牌</p> <p>顯示目前所有玩家的手牌</p>
show_one_card()	<p>輸入參數: 手牌</p> <p>顯示特定一名玩家的手牌(方便進行遊戲)</p>
sorting_num()	<p>輸入參數: 手牌</p> <p>以 Bubble Sort 排序玩家手牌</p>
after_playcard()	<p>輸入參數: 手牌、打出的牌、出牌數</p> <p>根據出牌數呼叫 delete_hand 函式</p>
delete_hand()	<p>輸入參數: 手牌、打出的牌</p> <p>將打出的牌從手牌中移除</p>
clone_hand()	<p>輸入參數: 手牌</p> <p>回傳: 手牌</p> <p>複製該玩家的手牌作為出錯時的回復</p>

PART2. 和遊戲進行相關之 function	
check()	<p>輸入參數:打出的牌、出牌數、玩家、模式 回傳: 1(成功) 或 0(失敗)</p> <p>呼叫 check_sequence 並判斷玩家是否正確的打出手牌</p>
check_index()	<p>輸入參數: PILE、index 回傳: 1(成功) 或 0(失敗)</p> <p>判斷玩家輸入的 index 是否正確</p>
check_insert()	<p>輸入參數: 打出的牌、指向某 index 的指位器、判斷是插頭還是插尾的 flag 回傳: 1(成功) 或 0(失敗)</p> <p>判斷玩家在插入單張牌時是否有符合規則</p>
check_sequence()	<p>輸入參數: 打出的牌、出牌數 回傳: 1(成功) 或 0(失敗)</p> <p>判斷玩家在建立新牌堆時是否符合規則</p>
check_field()	<p>輸入參數: PILE 回傳: 1(成功) 或 0(失敗)</p> <p>判斷玩家在分割牌組及移動場上的牌後，場上的所有牌堆是否符合規則(大於 3 張)</p>
game_finished()	<p>輸入參數: 玩家手牌 回傳: 獲勝玩家或 0(遊戲繼續進行)</p> <p>若有玩家手牌數為 0，回傳該玩家，沒有則回傳 0 表示繼續遊戲</p>
getNewNode()	<p>輸入參數: 一張牌的資訊 回傳: 一個新的節點</p> <p>創造一個新節點並複製原本節點的資訊</p>
cloneBinaryTree()	<p>輸入參數: PILE 回傳: PILE</p> <p>以遞迴方式複製整個 BST</p>

move_to_other_cardpile()	<p>輸入參數: index、PILE</p> <p>回傳: PILE</p> <p>詢問玩家移牌的相關資訊並呼叫相關函式並回傳移完後的 PILE</p>
chosecard ()	<p>輸入參數: 玩家、PILE、回合、模式、複製的玩家手牌、複製的場上牌</p> <p>回傳: PILE</p> <p>呼叫其他函式、詢問遊戲進行中所需資訊</p>
PART3. 和場上牌處理進行相關之 function	
showfield()	<p>輸入參數: PILE</p> <p>顯示目前所有場上的牌</p>
PUT_ON_PILE ()	<p>輸入參數:</p> <p>回傳: PILE</p> <p>將玩家新建立牌堆放入 PILE 中</p>
insert_cardpile()	<p>輸入參數: index、PILE、手牌</p> <p>回傳: PILE</p> <p>將玩家的手牌插入到 PILE 中</p>
cut_cardpile()	<p>輸入參數: index、PILE</p> <p>回傳: PILE</p> <p>將目前場上的某 index 牌堆切成兩個牌堆</p>
move_insert()	<p>輸入參數: 目的 index、移動的牌、PILE、判斷頭尾的 flag</p> <p>回傳: PILE</p> <p>將某張牌由其中一個 index 序列移到另一個 index 序列</p>
move_delete()	<p>輸入參數: 來源 index、移動的牌、PILE、判斷頭尾的 flag</p> <p>回傳: PILE</p> <p>刪除在原始 index 序列中的移動牌</p>

