

-back-end-

#### 1. 변수/함수명 통일

처음 전체적인 큰 함수 4개를 4명의 팀원이 각각 1개씩 담당하여 따로 작업을 하였기에 모든 함수를 다시 하나의 프로그램으로 합치는데 생각보다 많은 시간이 들었습니다. 그 이유중 하나가 서로의 함수/변수명이 통일 되지않고 의미 파악도 쉽게 되지 않았기 때문에 이번주 작업 중 가장 먼저 해야 할 일이라고 생각했습니다. 우선 기존 card\_count, card\_name 같은 변수명을 cardCount, cardName 등 전체적으로 모든 변수명은 카멜표기법으로 변경했습니다. 또한 의미 파악이 한눈에 되지 않는 변수/함수명들을 의미 파악이 쉽게 변경했습니다. 예를들면 pass\_3 -> passCount, now\_player\_card() -> delete\_submitted\_card(), cards[20] -> subCardArr[20] 등으로 한눈에 이 변수/함수가 어떤 역할을 하는지 파악하기 쉽게 변수/함수명을 바꿨습니다.

#### 2. goto문 삭제

기존 코드에 있었던 goto문을 삭제하였습니다. 간단한 코드에서는 문제가 없을 수 있으나 최종 프로젝트까지 코드길이 길어지고 복잡해짐에 따라 goto문은 코드 진행을 꼬이게 할 수 있는 매우 위험한 요소이기 때문에 이번 회의에서 빼기로 했습니다. 원본 코드의 goto문은 잘못된 카드를 제출하였을 때(ex. 카드 번호 범위 위반, 자신의 패에 없는 카드 제출 등) 다시 카드를 재제출하기 위해 사용되었습니다. 이것을 While 반복문으로 바꿔 카드를 정상 제출하거나 pass를 하였을 조건에 조건문을 빠져나오고 그 이외에 모든 오류 상황은 카드 제출 로직을 반복하게 만들었습니다.

#### 3. 전역변수 사용

여러 가지 함수를 합치는 과정에서 다른 소스파일에서 같은 역할을 하는 변수를 불러오고 싶을 때 사용을 용의하게 하기 위한 변수들은 전부 전역변수 처리를 했습니다. 다른 소스파일에서도 extern 식별자를 통해 동일한 변수를 사용하고 가져올 수 있게 했습니다. 또한 다른 소스파일 뿐만 아니라 하나의 소스파일 안에서도 여러 가지 사용자지정함수에 반복적으로 나오는 변수들은 전부 전역변수 처리를 하여 코드 관리를 조금더 용의하게 하였습니다.

#### 4. 턴 진행방식 변경

이전 진행 상황에서는 턴 진행을 Queue자료 구조를 도입하기로 계획했으나 실제 프로젝트를 구성하는 단계에서 코드가 불필요하게 복잡해지는 것을 느껴 변경하였습니다. 플레이어의 패를 저장하는 2차원 정수배열 playerCards와 플레이어들의 닉네임을 저장하는 2차원 문자배열 player\_name의 인덱스를 활용하여 간단하게 턴을 돌아가는 로직을 새로 구현했습니다.

## 5. 카드 제출 방식 변경

기존 프로그램에서는 카드를 제출 할 때 '12/3' 이런식으로 scanf에 인수를 입력받아 12번 카드 3장 제출이라는 의미를 활용했지만 사용자가 입력할 때 식별이 어렵거나 복잡할수도 있다라고 생각했습니다. 개발 초기 단계에도 카드 제출 방식은 '12 12', '11 11 11 11'등 원하는 카드를 원하는 장수만큼 직접 타이핑하는 것을 원했습니다. 그럼에도 이전에 저 방법을 사용한 이유가 12를 2장 제출한다면 scanf에 2개의 인자를 받아야하고 11을 4장 제출한다면 인자를 4개를 받아야 하는 등 매입력마다 인자의 개수가 미지수이기 때문에 이 문제점을 해결하지 못했습니다. 하지만 이번 주 회의 결과 gets\_s함수에 원하는 카드번호와 장수를 문자열 배열로 받은 다음에 문자열 자르기 함수인 strtok()함수 활용하여 '공백'을 기준으로 문자열을 잘라 사용자가 입력한 카드 번호를 알고 전역변수 subCardArr 배열에 저장하고 잘린 개수를 이용하여 카드 장수를 return 받을 수 있게 만들었습니다.

## 6. 상세 룰 추가

이전 진행 상황에서는 상세적인 룰까지 카드 제출 로직에 도입하지는 못했습니다. 그 중 프로그램에 도입하지 못한 룰 중 하나인 이전 플레이어보다 '높은 등급'의 카드를 '같은 장수' 만큼 제출 알고리즘을 도입하지는 않았습니다. 그래서 이번주에 이것도 도입하였습니다. 이전 플레이어가 제출한 카드 번호와 카드 장수를 각각 beforeCardName과 beforeCardCount 전역변수에 저장을 한 뒤 이후 플레이어가 제출한 카드가 beforeCardName보다 낮거나(이전 플레이어보다 높은 등급의 카드 제출) beforeCardCount와 같지(이전 플레이어와 같은 장수의 카드 제출) 않다면 오류 코드가 생기게 했습니다.

또한 위에 로직을 추가함에 따라 턴 초기화 조건(마지막으로 제출한 플레이어 이외에 모든 플레이어가 pass를 할시 처음부터 원하는 카드를 원하는 장수만큼 제출할 수 있도록 초기화)도 추가했었어야 했습니다. 이를 위해 전역변수 passCount변수를 만들어 플레이어가 카드를 제출하였다면 passCount를 0으로 초기화 하고 플레이어가 pass를 하였다면 passCount를 1씩 증가시켜 4인 플레이 기준 passCount >= 3 조건을 만족하였을 때 beforeCardName과 beforeCardCount 변수를 초기화 하여 다음 플레이어는 카드 제출에 제약 없이 원하는 카드를 원하는 장수만큼 제출 할수 있도록 기능을 추가했습니다.