

# 수행내용 보고서

## ( 2 차 보고)

과제명	오프라인 보드게임 “달무리” 의 콘솔 게임화		
발표자	김민석, 이승현	발표일자	2023.05.15
연구참여자 (담당분야)	팀장: 황현석(담당분야: 팀관리, 카드 배분 함수 구현, 수행내용 보고서 작성 ) 팀원: 이진석( 턴관리 함수 작성 ), 김민석 ( 카드 수 최신화 함 수 구현), 이승현(카드 제출 검사로직 구현)		
수업명/조번호	프로그래밍설계/7조		

## 1. 연구일정 점검

1) 단기일정 : 2023년 5월 7일 ~ 2023년 5월 14일 (이전, 이후 단계는 절사)

수행 내용	일정 (1일 단위)								달성 실적
	1	2	3	4	5	6	7	8	
- 목표 설정 - 게임 rule 재편 -요구사항 명세서 작성 -기능별 모듈화									완료
- 프로그램 로직 부분 코딩									70% 달성
- UI 구현									0% 달성

위의 달성 실적란에는 발표일 기준으로 한 학기 기간을 고려하여 전체 목표 대비 발표일 시점의 실적 비율을 표시하며 완료한 수행내용은 완료로 표시함.

## 2. 연구수행 내용

: 지난 발표 후 연구수행 내용을 구체적으로 적을 것

: 제출일 기준 발표자료 내용과 설명을 포함할 것

-카드 배분 로직-

게임 Logic 처리 부분에서 ‘카드 배분 로직’, ‘게임 턴 관리 로직’, ‘플레이어의 현재 카드 수 표시’, ‘카드 제출 로직’을 구현하고자 하였습니다.

카드 배분 로직은 플레이어 4명 고정 기준 80장의 카드를 4명이 비복원 추출 방식으로 랜덤하게 나누어 가지게 됩니다. (한 플레이어 당 20장의 카드를 가지게 됩니다.)

카드를 배열로 관리하는데, 1번 카드는 0인덱스, 2번 카드는 1,2 인덱스, 3번 카드는 3,4,5인덱스, 4번 카드는 6,7,8,9인덱스 ... 이며 마지막 조커 카드 2장은 78,79인덱스로 저장합니다.

이때 랜덤으로 카드를 뽑는다는 것은 0부터 79까지 중의 숫자를 하나 추출하여 해당 인덱스에 해당하는 카드 번호 값을 카드로 저장하게 되도록 구현 하였습니다.

#### -카드 제출 로직-

카드 제출 로직은 각 플레이어의 턴이 될 때마다 카드를 제출할지, pass하고 다음 순서 플레이어에게 넘길지 정하는 로직입니다. 우선 플레이어의 턴이 되면 pass를 할지 안할지 여부를 물어봅니다. pass를 한다면 1을 입력, pass하지 않고 카드를 제출하고 싶다면 0을 입력 받습니다. 다음은 if문으로 가서 pass 변수가 1이면 pass를 한 후 다음 플레이어의 순서로 넘어갑니다. pass 변수가 0이라면 카드 제출 로직으로 넘어갑니다. 카드 제출 로직에서는 플레이어가 제출하고 싶은 카드의 번호와 장수를 입력받습니다. 이후 sub\_fail\_check()라는 사용자정의 함수에서 입력받은 카드의 정보가 정상인지 오류인지 확인해 줍니다. sub\_fail\_check()에서는 정상 제출이면 0을 return, 카드의 범위가 1~13이 아니라면 1을 return, 제출할 카드 장수만큼 플레이어가 가지고 있지 않다면 2를 return 등 여러 가지 오류와 오류 코드들을 return해 줍니다. 로직에 마지막에는 sub\_fail\_check()의 return값이 0이면 '카드 정상 제출'을 출력하고 다음 플레이어 턴으로 넘어가게 됩니다. 만약 오류가 있는 카드 제출이었다면, 각 오류 코드에 맞는 경고문을 띄워준 뒤 다시 카드제출 로직으로 돌아가 카드를 다시 제출하게 해줍니다.

#### -턴 관리 로직-

randTurn()로 정수배열 0 1 2 3을 섞은 배열을 만듭니다. 전역변수로 선언된 입력받은 플레이어 이름 문자열 2차원 배열players\_name[4][20] = {One, Two, Three, Four}을 아까 만든 랜덤 정수배열의 순서대로 재배치합니다. 이제 재배치된 이름들을 Queue에 순서대로 넣습니다. 이제 initTurn을 하게 되면 새로 선언한 Queue를 initializeQueue 하여 생성되고 안에 랜덤 순서대로 배치된 players\_name를 입력합니다. 위 과정이 끝나면 currentTurn()을 하게되면 큐의 맨 앞에 있는(현재 순서인)플레이어 이름을 보여줍니다.

endTurn()하면 큐의 맨 앞에 있는 플레이어를 dequeue로 삭제하고 다시 enqueue를 해서 맨 뒤에 배치되도록 합니다.

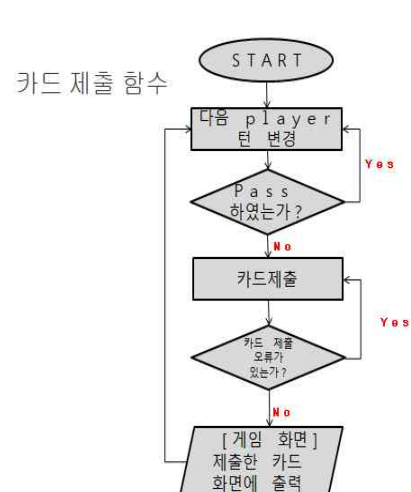
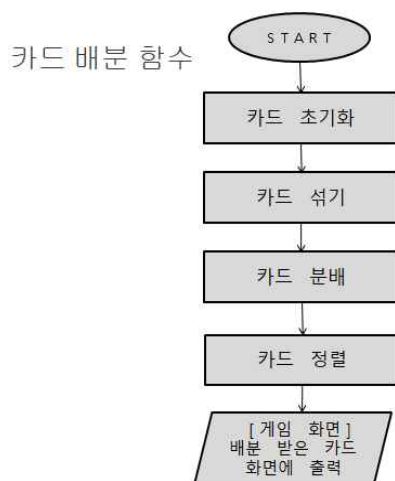
### -카드 카운트 로직-

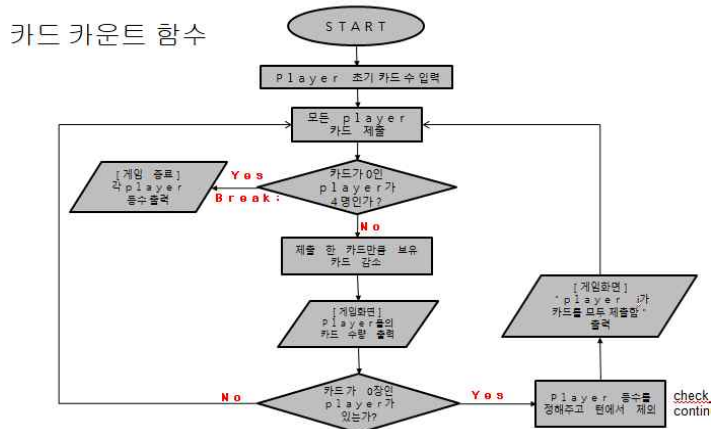
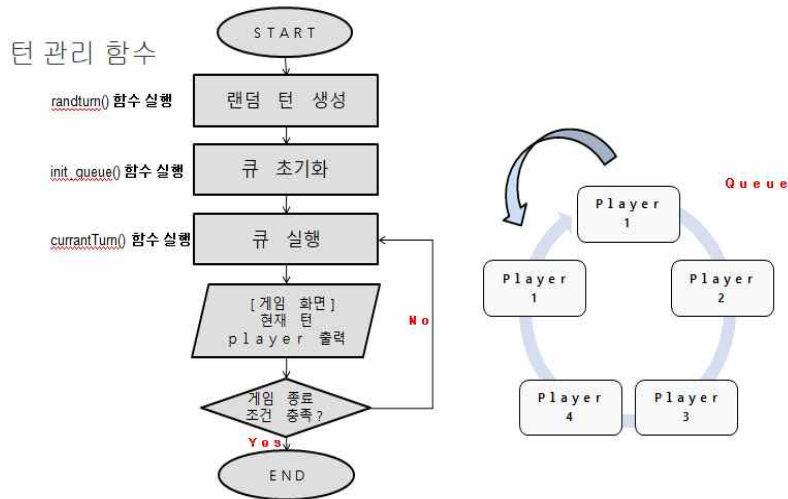
카드 카운트 로직은 매 턴마다 낸 카드 개수를 입력하고 초기 카드 수에서 제출한 카드 수를 빼어 모든 플레이어들의 카드가 0장이 되면 게임 종료를 출력하고 각 플레이어들의 등수를 출력 해주는 로직입니다. 우선 플레이어들의 초기 카드 수를 입력해 줍니다. 그 후 while문으로 크게 감싸고 매 턴마다 각 플레이어들의 입력 받은 카드 수를 출력해 줄 수 있게 하고 if(zero\_count==4)로 매 턴마다 카드가 0인 플레이어가 4명인지 확인하고 맞다면 게임이 종료되게끔 해줍니다. 그리고 각 플레이어의 차레시 카드가 0인 경우에 입력하지 못하도록continue로 반복문을 빠져나가 줍니다. 이제 0이 아니라면 각각의 플레이어들이 for문을 통해 카드를 제출 해 줍니다.

그 후 낸 카드 수를 입력받고 처음 입력한 카드 수에서 빼줍니다. 그 후 플레이어[i]배열이 0보다 작으면 아래 check\_rank함수에서 플레이어 순위를 출력시켜 줍니다. 그 후 while문을 통해 또 현재 카드 수를 출력 해 주고 반복해 줍니다. 그 과정에서 계속 카드를 낸 후 4명 다 패가 없다면 break문을 통해 while문을 빠져 나가 주고

for 문을 통해 check\_rank함수에서 저장된 rank배열을 for문을 통해 각각 1등부터 4등까지 출력하게 해 줍니다.

### -각 함수에 따른 상세 순서도-





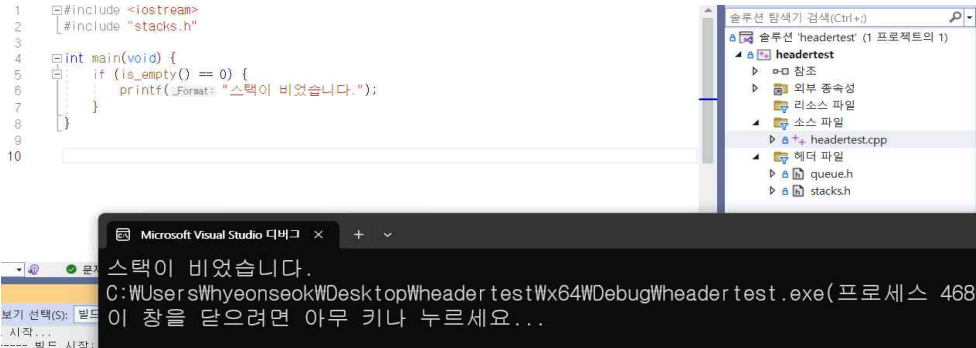
### 3. 연구계획 (향후 연구수행 예정인 내용 요약)

구분		수행내용 요약
팀전체		달부터 게임의 카드 로직을 함수별로 구분 한 후 개별 구현하여 헤더 파일로 합치기.
개인별	황현석	카드를 플레이어 수에 맞게 배분 후 정렬해주는 함수 구현
	이진석	턴의 종료와 턴 순서를 정해주는 함수 구현
	김민석	플레이어의 현재 카드 수를 표시해주는 함수 구현
	이승헌	카드 제출 시 규칙에 맞는지 검사하는 함수 구현

#### 4. 문제점 및 애로사항

: 과제 진행중에 발생한 문제점들과 이를 해결한 내용들

: 애로 사항들을 구체적으로 기술한다.

회의록			
회의명	주간 회의		
일시	23. 05. 12	장소	온라인 그룹통화
의제	코드 구현 진행상황 및 개인별 어려움 공유 협력		
회의 내용	<p>1) 헤더 파일 모듈화 연습 진행</p> <p>각자 헤더 파일을 만들어 include하고 호출 실습을 진행하면서 헤더 파일 모듈화 연습을 진행하였다.</p>  <p>민석: 게임 메인화면 우측상단에 띄울 플레이어 현재 카드 수 구현 - 0장인 플레이어에게는 해당 플레이어의 랭킹을 띄움</p> <p>승헌: 카드 제출 로직 함수 구현 중, 제출시 pass 혹은 숫자 카드를 입력받아 게임을 진행해야 하는데 문자열 pass와 정수 배열을 동시에 입력받을 수 없는 오류 발생 -&gt; pass여부를 먼저 묻고 pass 하지 않는다면 카드 제출을 하는 것을 묻는 방식으로 해결. (추후 변경가능성O)</p> <p>진석: 턴 관리 로직 구현. 현재 시작시 턴 랜덤 섞기 구현완료 하였고, 턴 종료 함수 구현중임</p> <p>현석: 카드 분배 로직 구현 완료,</p> <p>승헌님이 구현에 어려움을 겪는 부분인 카드 제출 시 제출한 카드 개수를 입력받는 부분에서 1개, 2개, 3개... n개마다 다른 scanf 인자 개수로 입력받아야 하는 문제에서 오버로딩을 이용한 해결 아이디어를 제시 해주었다.</p> <p>추후, GitHub을 통한 코드 및 버전 관리 진행에 대해 의논할 예정이다.</p>		
결의 사항	<p>이전에는 제출할 카드를 '4 4 4' 이런 형식으로 제출할 카드를 모두 입력을 받는 것으로 설계를 하였다. 하지만 이러한 방식은 카드를 몇 개</p>		

	<p>입력받을지 모르므로 scanf 안의 형식지정자 개수를 몇 개를 써줘야 할지에 대한 문제점이 있었다.</p> <p>Ex. scanf("%d %d", &amp;submit_card[0], &amp;submit_card[1]); // '4 4'처럼 2개를 제출할 경우에만 유효</p> <p>또한, pass을 입력받으려면 %s 형식 지정자가 필요한데, pass을 할지 카드를 제출할지 먼저 물어보는 것이 우선이라고 판단했다.</p> <p>위 문제는 오버로딩으로 해결이 가능하지만, 2학년 학부생 수준에 맞추어 솔루션을 의논한 결과는 다음과 같다.</p> <p>1) 먼저 pass을 할 것인지 물어보고, pass을 한다고 입력하면 pass을 진행한다.</p> <p>2) pass을 진행하지 않는다면 널 카드와 개수를 입력 받는다.</p> <p>예를 들어, 이전에는 '4 4 4' 이런 식으로 제출할 카드를 입력했다면, '4 3' (카드 4를 3장 낸다.)로 수정한 것이다.</p>			
이견 사항	<p>추후 지도교수님 상담을 통해 두 솔루션(오버로딩과 결의사항에서 결정된 방식)의 시간복잡도를 비교해 더 좋은 방향으로 구현을 할 예정이다.</p>			
참석 현황	대상	성명	학번	서명(자필)
	참석자	황현석 이진석 김민석 이승현	20191033 20191008 20200892 20200931	황현석 이진석 김민석 이승현
	불참자			
불참 사유				
작성자	황현석, 이진석	작성일	23. 05. 12	

5. 지도교수 코멘트 내용 : 지도교수님한테 상담받은 내용 또는 수업시간에 받은 코멘트를 정리함

## 지도교수 상담일지

과제명			
일지 NO		학년도/학기	2021 학년 / 1 학기
소속	정보통신공학과 2학년		
지도교수 확인	2021년	월	일                      지도교수:                      (인)
상담 안건			
상담 내용			
코멘트에 대한 조치사항			
상담 참여자	<div> <div>_____ (인)</div> <div>_____ (인)</div> </div> <div> <div>_____ (인)</div> <div>_____ (인)</div> </div>		