최종 결과 보고서

(2021학년도 1학기)

과제명	오프라인 보드게임 “달무티”의 콘솔 게임화		
과목 /담당교수	프로그래밍설계 / 정복래교수님		
제출일자	2021. 06. 12		
연구참여자 (담당분야)	팀장: 황현석(담당분야: 팀관리, 카드 배분 함수 구현, 수행내용 보고서 작성) 팀원: 이진석(턴관리 함수 작성), 김민석 (카드 수 최신화 함수 구현), 이승헌(카드 제출 검사로직 구현)		
팀명	프로그래밍설계	**팀 번호**	7조

목 차

1. 설계과제의 제목 :

C프로그래밍을 활용한 오프라인 보드게임(달무티) 재현(PC 온라인 게임화)

2. 연구목적

1) 오프라인으로 할 수 있는 인기보드게임을 PC으로 플레이 할 수 있다.
2) 게임의 모든 구조가 랜덤으로 정해지기 때문에 공정하게 게임을 진행할 수 있다.

3. 설계과제의 필요성

최근 오프라인 보드게임 pc화 진행이 많아지고 있는 추세이다. 오프라인 보드게임을 즐길수 있는 보드게임카페 같은 사업도 늘어났지만 집에서 간단하게 지인들과 온라인으로 즐길수 있는 보드게임 플렛폼 또한 늘어난 추세이다. 이런 트랜드를 따라 기존 인기있던 오프라인 보드게임인 '달무티'를 pc콘솔화 하기로 하였다.

4. 설계과제의 목표

1. 게임 디버깅 및 테스팅을 통해 안정적이고 완성도 높은 게임 개발.
2. 알고리즘 이해 및 그래픽에 대한 이해도 증진.

현실적 제한 요소들	내 용 (Content)
기술적 문제	1. 플랫폼 호환 문제: 다양한 운영체제와 브라우저에서 게임을 실행할 수 있어야 함으로 플랫폼 호환성 문제를 고려해야 한다. 2. 그래픽 처리에 대한 알고리즘 문제 - 그래픽 처리는 많은 수의 연산을 함으로 게임이 끊기거나 지연되는 문제가 발생할 수 있다. 그렇기에 그래픽 처리를 위한 자료구조와 알고리즘을 최적화 해야 한다. 3. C언어는 다른 언어에 비해 보안이 취약함으로, 보안 문제가 있다. -(치팅방지, 이를 방지하기 위해선 서버-클라이언트 서버 간의 데이터 교환 방식을 확실하게 정의해야 한다. 서버측에선 클라이언트의 데이터를 검증하고 처리하는 방식을 사용해야 한다.)
시간적 문제	코드 개발 및 대한 시간적 문제

5. 설계과정

5.1 설계 기초이론

1. c언어 기본 프로그램에 대한 이해
2. c언어 심화 프로그램에 대한 이해
   -모듈, 함수, 전역변수, 사용자지정 헤더파일, <window.h>
   <mmsystem.h>헤더파일 등 이해
3. front-end와 back-end의 역할 분할 이해

5.2 소프트웨어 기능블록도 (개념설계)

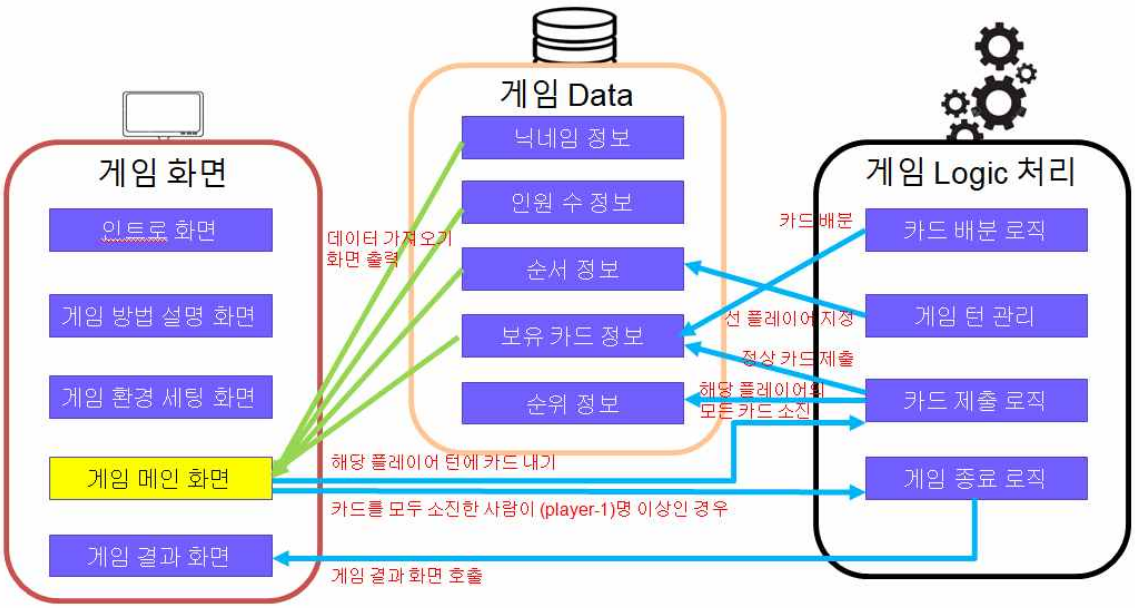


그림 5.2.1 '달무티 게임 프로그램'의 전체 기능 블록도

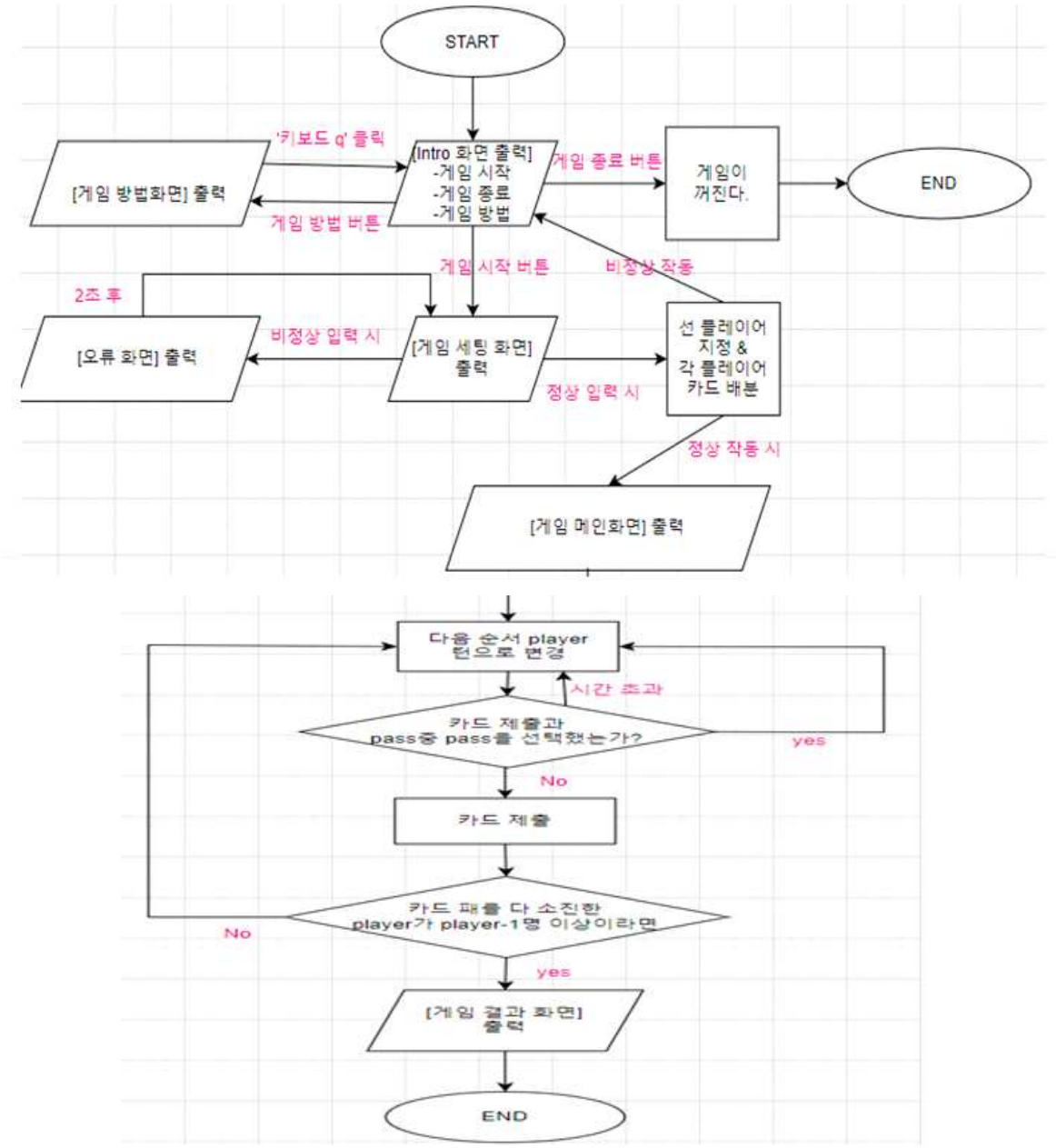


그림 5.2.2 '달무티 게임 프로그램'의 전체 상세 순서도

5.3 기능적 요구사항 명세서

ID	화면명	요구사항명	요구사항 내용	날짜	작성자	진행사항	우선순위	F/B
INTRO-0001	인트로 화면	인트로 화면 구성 게임 방법, 게임 시작, 게임 종료 버튼	달무티 콘솔 게임 인트로 화면을 구성한다. [게임 방법] 버튼을 클릭할 경우 '달무티 게임 방법을 설명해주는 텍스트 UI을 출력한다. '키보드 q' 키를 누르면 [인트로 화면]으로 돌아간다. [게임 시작] 버튼을 클릭하면 [게임 세팅 화면]으로 넘어간다. [게임 종료] 버튼을 클릭하면, 프로그램을 닫는다.	4-30	황현석	미반영	1	Front
GAME-0001	게임 환경 세팅 화면	게임 환경 세팅 화면 구성 참여 인원 수/플레이어 닉네임 입력 처리	참여 인원 수와 각 플레이어의 닉네임 입력 받는 화면을 구성한다. 현재 버전에서 참여 인원 수는 4명만 가능하도록 하고, 닉네임은 영어 닉네임 입력만 가능하도록 한다. 정상적으로 모든 정보를 입력 시 [게임 메인화면]으로 넘어간다. 비정상 입력 시 다시 입력을 받는다.	4-30	황현석	미반영	3	Front
GAME-0002	게임 메인 화면	게임 메인 화면 구성 이번 턴에 낼 카드 입력 처리	[최초 1회] 랜덤으로 정해진 플레이어 턴 순서를 보여준다. [게임 중] 각 플레이어의 이름과 남은 카드 수, 최근에 낸 카드와 카드 개수 이미지, 자신의 패을 화면에 출력하고 이번 턴에 낼 카드를 입력 받는다. 낼 카드를 입력하거나, 'pass'을 입력하여 카드를 내지 않을 수 있다.	4-30	황현석	미반영	4	Front
RES-0001	결과 화면	결과 화면 UI 구성	플레이어 이름과 등수를 표시해준다. 키보드 q 키를 누르면 [intro 화면]으로 전환된다.	4-30	황현석	미반영	7	Front
GM-0001	게임 로직 관리	카드 내기 로직 처리	case 카드 제출 (정상): [GAME-0002]에서 제출한 카드가 이전에 제출한 카드 등급보다 높은 등급이고 개수가 일치한 카드인 경우 A) [Player Data]에 정상적으로 카드가 제출한 것이 적용되고 카드를 낸 플레이어가 카드를 모두 소진했는지 검사한다. B) 카드를 모두 소진했다면 해당 플레이어의 등수를 처리해준다. C) 다음 플레이어의 턴으로 전환된다. case 카드 제출(비정상) : 정상적으로 제출되지 않은 카드는 카드 내기 로직을 처리하지 않는다. case pass : A) 다음 플레이어의 턴으로 전환된다.	4-30	황현석	미반영	5	Back
DATA-0001	플레이어 데이터 관리	인게임 플레이어들의 정보들을 관리	현재 인게임 플레이어의 순서, 현재 가지고 있는 카드, 순위, 닉네임 정보를 저장한다.	4-30	황현석	미반영	2	Back
GM-0002	게임 로직 관리	게임 턴 관리	[게임 시작 최초 1회] 랜덤으로 플레이어의 턴을 정한다. [게임 중] 1. 카드를 마지막으로 제출한 player 기준으로 나머지 player들이 모두 pass을 했을 경우 카드를 마지막으로 제출한 player가 '선 플레이어'가 된다. 2. 카드를 모두 소진한 플레이어인 경우 해당 플레이어 턴에서 제외시킨다.	4-30	황현석	미반영	5	Back
GM-0003	게임 로직 관리	카드 배분 로직	80장의 카드를 카드 덱 배열을 이용하여 각 플레이어가 80/player 개의 카드를 나눠 가지도록 한다. (비복원 추출 방식)	4-30	황현석	미반영	5	Back
GM-0004	게임 흐름 관리	게임 종료	해당 조건에 해당하는 경우 게임을 종료하고 [결과 화면] 으로 이동한다. 1. [게임 중] (전체 player - 1) 명이 카드를 소진하였을 경우	4-30	황현석	미반영	6	Back

5.4 전체/부분 순서도 (상세설계)

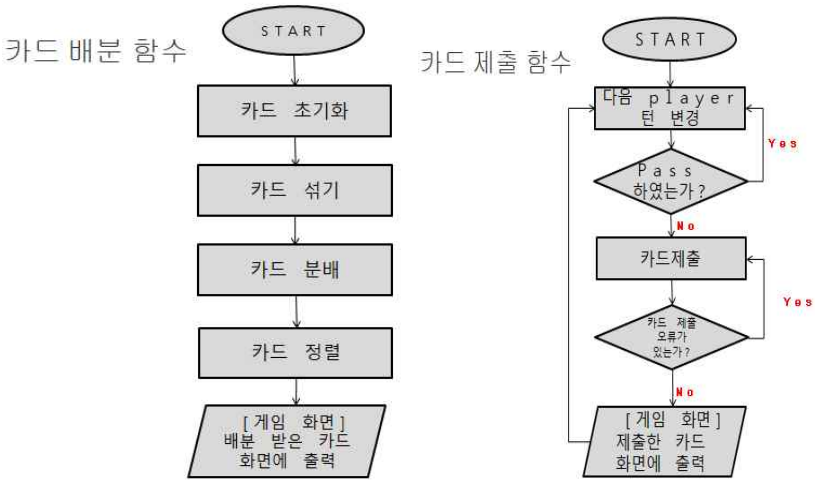


그림 5.2.2 '카드 배분 함수'의 상세 순서도
그림 5.2.3 '카드 제출 함수'의 상세 순서도

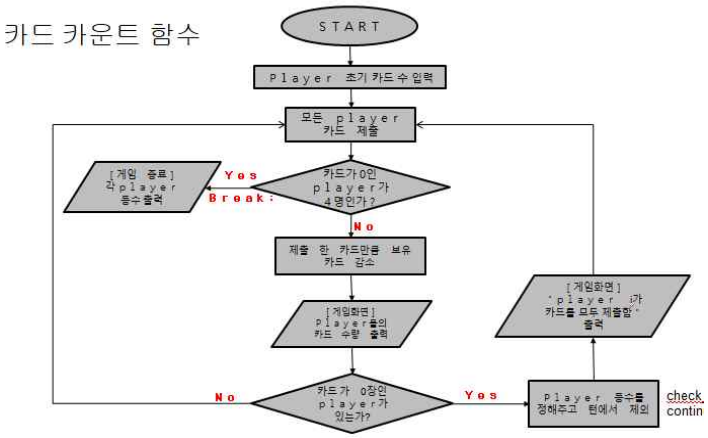


그림 5.2.4 '카드 카운트 함수'의 상세 순서도

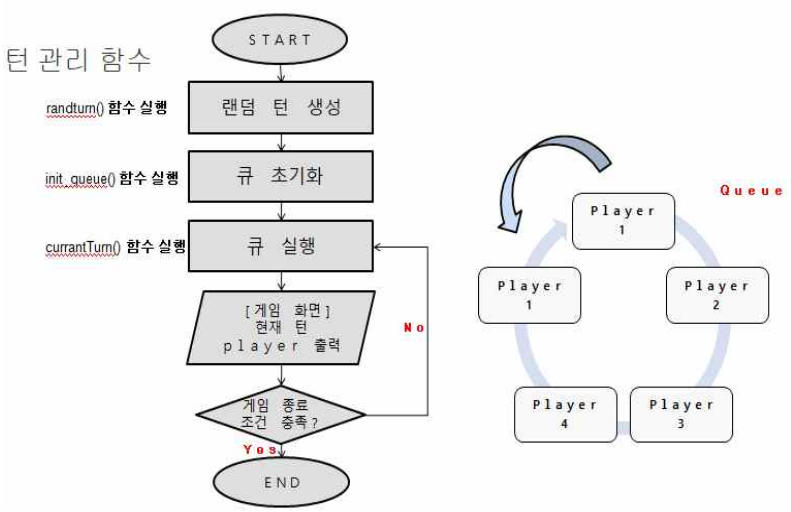


그림 5.2.5 '턴 관리 함수'의 상세 순서도 before

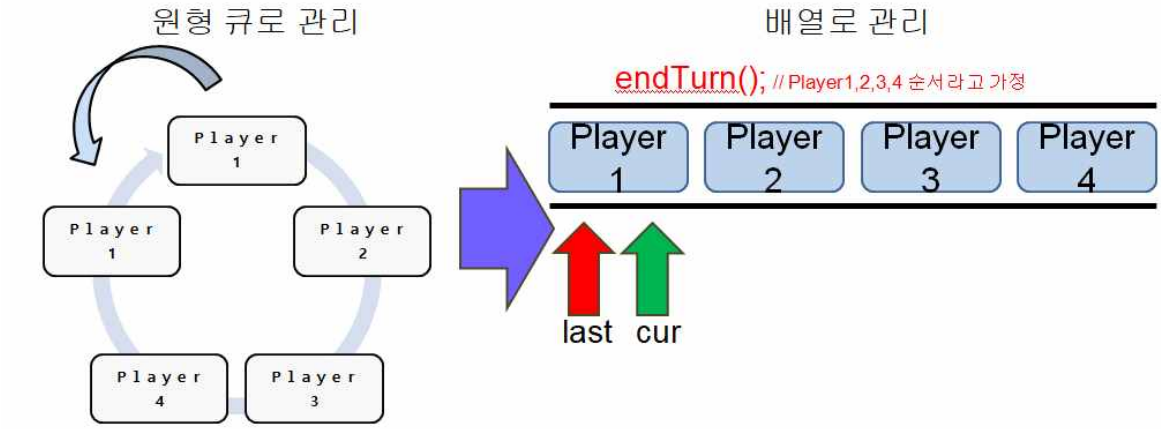


그림 5.2.6 '턴 관리 함수'의 상세 순서도 after

6. 제작(Implementation)

6.1 제작과정

-1 주차(설계 제안 주차)-

1. 기존 C언어를 이용한 '윈도우 구현 프로젝트'로 결정하였으나,
현실적 제약 조건 (ex.printf와 스크린 버퍼 지우기와 같이 현실적 제약 조건을 나열)로 인해 주제를 다시 생각해볼 필요가 있었다. 그래서 나름 사람들에게 익숙하면서 기존에 서비스를 하지 않고 있는 것에 대해 논의하던 중 보드게임인 "달무티"를 콘솔(PC)게임으로 만들기로 결정하였다. 이후 달무티 게임에 관련한 규칙에 관해 구현이 가능한 제약 조건 하에서 논의해보고 달무티 게임 규칙을 일부 수정하였다.

-2 주차-
첫 주차에 설계이후 팀원 4인이 각자 담당할 함수를 분배한 후 제작하기로 결정. 각각 카드 분배 함수, 카드 제출 함수, 카드 카운트 함수, 턴 관리 함수 4분야로 나눠서 제작. 카드 분배 함수에서는 총 80장에 카드를 4명의 플레이어한테 분배한다. 카드 관리는 subCardArr[80] 배열로 관리한다. 카드 제출 함수는 자신의 턴에 pass와 제출 여부를 묻고, 제출한다면 원하는 카드 제출후 sub_fail_check()함수에서 오류가 없는 제출인지 확인후 화면에 출력해 준다. 카드 카운트 함수는 각 플레이어가 카드 제출 후 남은 카드 장수를 카운트 해서 화면에 출력해준다. 턴 관리 함수는 게임 진행에 필요한 턴 순서 로직을 관히해주는 함수이다.

-3주차-
변수/함수명을 통일화 하여 4명의 프로그래머들이 하나의 프로그램의 코드를 혼동없이 이해할 수 있게 변동 하였다. 기존 코드에 있던 goto문을 while로 바꿔 스파게티식 코드를 변동하였다. 전역변수를 사용하여 모듈화된 여러 가지 함수에서 하나의 독립된 변수를 불러올수 있도록 변동하였다. 턴 진행방식을 기존 설계했던 queue 데이터구조 방식이 코드를 쓸데없이 복잡하게 만든다고 판단하여 플레이어 턴을 담당하는 배열을 사용하게 변동하였다. 카드 제출 방식을 기존 11/3(11 3장 제출)방식에서 11 11 11 방식으로 변동하여 가시성을 확대하였다. 상세룰 추가를 통해 기존 설계했던 게임 룰에 위반되지 않게 모든 코드를 안정화 하였고 게임 종료 로직 또한 포함하여 온전한 게임 구실을 할수 있게 변경하였다.

6.2 제작시 문제점 및 개선사항

-둘째 주차 결의사항-
이전에는 제출할 카드를 '4 4 4' 이런 형식으로 제출할 카드를 모두 입력을 받는 것으로 설계를 하였다. 하지만 이러한 방식은 카드를 몇 개 입력받을지 모르므로 scanf 안의 형식지정자 개수를 몇 개를 써줘야 할지에 대한 문제점이 있었다.
Ex. scanf("%d %d", &submit_card[0], &submit_card[1]); // '4 4'처럼 2개를 제출할 경우에만 유효
또한, pass을 입력받으려면 %s 형식 지정자가 필요한데, pass을 할지 카드를 제출할지 먼저 물어보는 것이 우선이라고 판단했다.

위 문제는 오버로딩으로 해결이 가능하지만, 2학년 학부생 수준에 맞추어 솔루션을 의논한 결과는 다음과 같다.
1) 먼저 pass을 할 것인지 물어보고, pass을 한다고 입력하면 pass을 진행한다.
2) pass을 진행하지 않는다면 낼 카드와 개수를 입력 받는다.
예를 들어, 이전에는 '4 4 4' 이런 식으로 제출할 카드를 입력했다면,
'4 3' (카드 4를 3장 낸다.) 로 수정한 것이다.

-셋째 주차 결의사항-
1) 카드 분배 로직 수정
카드 정렬, 배분 받은 카드를 화면에 출력 부분은 front-end 부분으로 옮기고 카드 분배 로직은 카드 분배 기능만을 수행하도록 한다.

2) 게임 턴 관리 로직 구현 방식 설정
기존 '원형 큐 방식'을 사용할 경우 정보통신공학과 2학년 학부생 수준에서 막 큐를 배우고 있는 단계라 원형 큐를 적용하기 어려움이 있다고 판단하여 이를 배려해 '배열을 통한 인덱스 접근' 방식을 사용한다.
성능도 크게 차이가 없고, 더 코드 해석이 쉬워 이 방식을 채택하게 되었다.

3) goto문에서 while문으로 전환
goto문은 스파게티코드를 유발한다.
while문으로 수정하여 구현한다.

4) 룰 통일 작업
같은 게임이더라도 다른 룰을 생각하고 구현하면 안되므로, 확실하게 통일하고 구현에 넘어가야 한다.

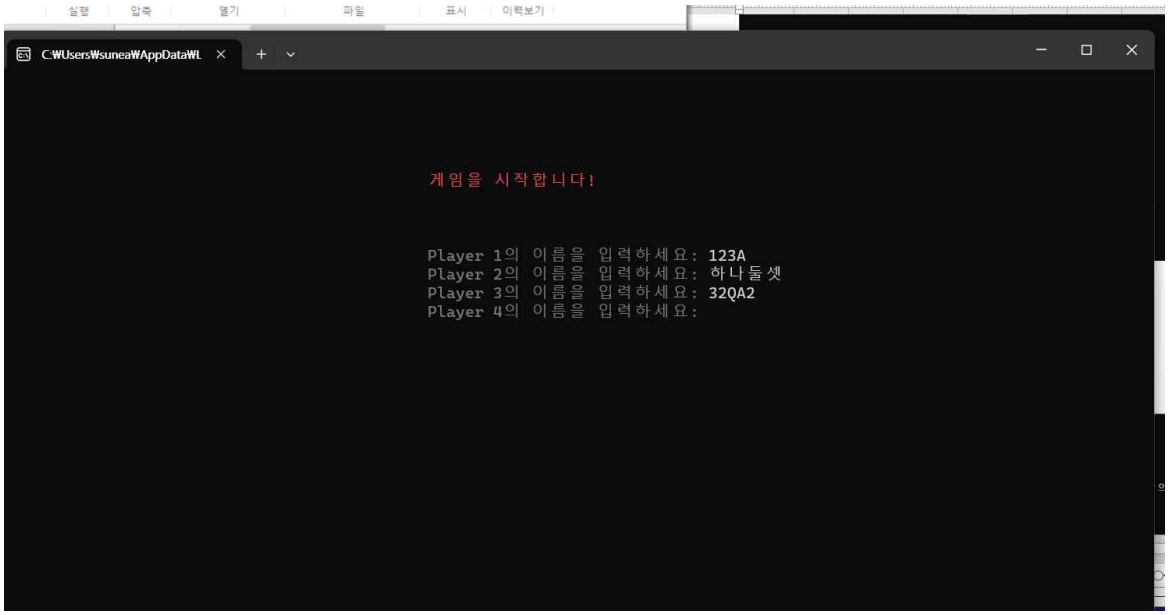
6.3 지도교수 지도내용 및 지적사항 조치결과

상담 주차	상담 안건	상담 내용	조치 사항
1주차	1. 요구사항명세서 작성 요령 2. 기능블록도에서 예외처리를 추가하여 세세하게 해야 할 지에 대한 의문점 3. 시장 조사 방식 4. 발표 전략	요구사항 명세서는 기능별로 간추린다. 기능블록도에서 예외처리와 같은 상세한 것은 발표 내용이 산으로 갈 수 있으니 생략하고 나무보다는 숲을 바라보고 발표를 해야 한다. 발표는 알아듣기 쉽게 하는 것이 가장 중요하다. 시장 조사 시 콘솔(cmd) 게임이지만 PC 게임으로 수요도 조사를 해도 되냐는 질문에 해도 된다고 말씀하셨으므로, PC 게임으로 수요도 조사를 진행한다. 이번 발표에서 다음에 따로 룰 설명이 필요 없도록 모두가 이해할 수 있는 룰 설명을 끝내놓는 것이 좋다고 말씀하셨다.	요구사항 명세서를 조금 더 추상화하였다. 예를 들어, 인트로 화면 구성에서 처음에는 [게임 종료 버튼], [게임 시작 버튼] 등의 각 버튼과 UI마다 모두 명세서를 작성했다면 이 것을 하나로 합쳐서 조금 더 가시적으로 바꾸었다. 기능적인 설계도 front-end와 back-end를 나누어 조금 더 역할 분담에 대해 구체화를 하였다. PC게임으로 수요도 조사를 진행하였다. (cmd, 콘솔 이라는 단어에 익숙하지 않은 투표자들을 배려하는 차원)
2주차	-	-	-

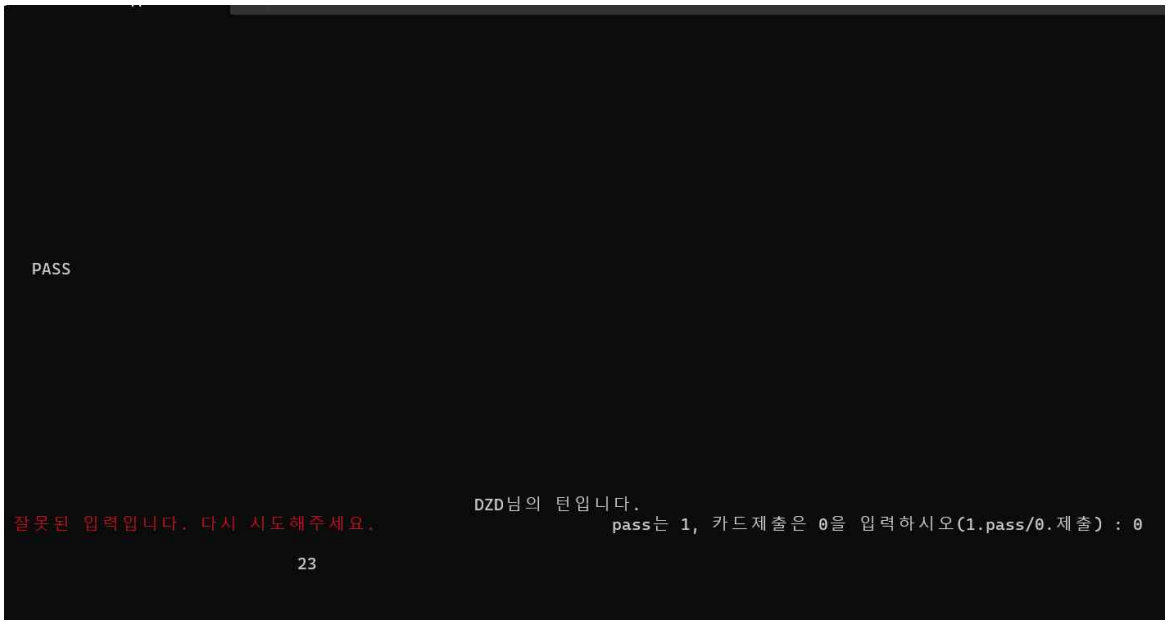
3주차	1. 5월 15일 발표 피드백과 앞으로의 일정 (룰 설명이 이해되기 쉽게 청중의 입장에서 적절한 설명이었는지 의문점 해결, 최종보고 발표 때 시연은 어떤식으로 보여주면 되는지에 대한 의문점 해결) 2. 오버로딩으로 여러 개의 코드(12개)를 작성할 경우 코드가 더러워지지 않는지에 관한 질문 3. [팀장 개인 상담 질문] 학생 연구원에 관한 상담	Q. 5월 15일, 해당 발표가 달무티 룰을 청중에게 이해시키는데 적절했는지와 앞으로도 이런 룰 설명을 지속적으로 할 필요가 있을지에 대한 의문점 A. ppt 애니메이션을 활용하여 저번보다 달무티 룰을 청중에게 이해시키는데는 도움이 많이 되었던 것 같다. 하지만 일주일 뒷면 룰을 또 까먹는 경우가 많으므로 룰 설명을 지금처럼 초반에 간단하게 설명하고 넘어가는 것이 청중의 이해와 흥미를 높여주는데 좋을 것 같다. 반영 : 해당 슬라이드를 그대로 청중들에게 3차, 4차 발표때도 설명을 하여 달무티 룰을 지속적으로 상기할 수 있도록 한다. Q. scanf 입력 받는 인자의 경우의 수가 12개여서 오버로딩을 통해 구현을 하고자 하는데 오버로딩을 통해 이렇게 여러개의 함수를 만들 경우 코드를 더러운 코드라고 판정하시지는 않을지 궁금합니다. A. 모듈화를 한 코드를 더러운 코드라고 판단할 수는 없다. 12개의 함수를 오버로딩으로 구현해도 된다. Q. 변수명, 함수명을 지을 때 카멜 표기법, 스네이크 표기법은 크게 상관이 없는지 궁금합니다. A. 크게 상관 없다. 통일성만 있으면 된다.	이번 상담을 통해 오히려 함수를 최대한 많이 사용하고 main 부분을 짧게 가져가는 것이 더 좋다고 판단하여 최대한 함수를 많이 사용하고 main 부분을 짧게 가져가고자 설계하였고, 협업을 위해 변수명 규칙을 팀 내에서 정하여 사용하였으며 .cpp 파일과 .h(헤더 파일)을 분리하여 팀 작업이 원활하게 되도록 개발 환경을 조성하였다.

7. 시험

1번 누를시 게임 시작



닉네임 이름 입력



게임 카드 제출 0,1가능

이런식으로 자기가 가진 카드 패가 오른쪽 아래에 뜬다. 룰에 맞게 제출할 수 있다.

패를 제출하고 나면 바로 화면이 바뀌니 카운트 다운을 통해 다음 사람의 패가 안 보이게 한다.
이런 식으로 게임이 구성된다

2번 누를시 방법 설명

```
                    게임 방법:


1.총 80장의 카드로 구성되어 있고 시작시 4명의 플레이어들에게 각각 20장의 카드를 분배해줍니다.

2.게임은 시계방향으로 돌아갑니다.

3.1부터 12까지의 카드가 있고 13은 조커 카드입니다.
(조커 카드는 카드 복사 기능을 합니다.)

4.앞선 사람이 낸 카드 갯수와 같은 카드갯수를 내고
그 카드의 숫자보다 더 높은 등급의 카드를 내야 합니다.

5.낼 수 있는 카드가 없거나, 전략을 노출시키기 싫다면 패를 안내고 PASS할 수 있습니다.

6.모두가 PASS를 하면 그 턴을 리셋 하고 마지막에 카드를 낸 사람이 다시 카드를 냅니다.)

7.손에 든 패가 다 소진될 경우 승리합니다.


엔터 키를 눌러 메뉴로 돌아가세요...
```

3번 누를시 종료

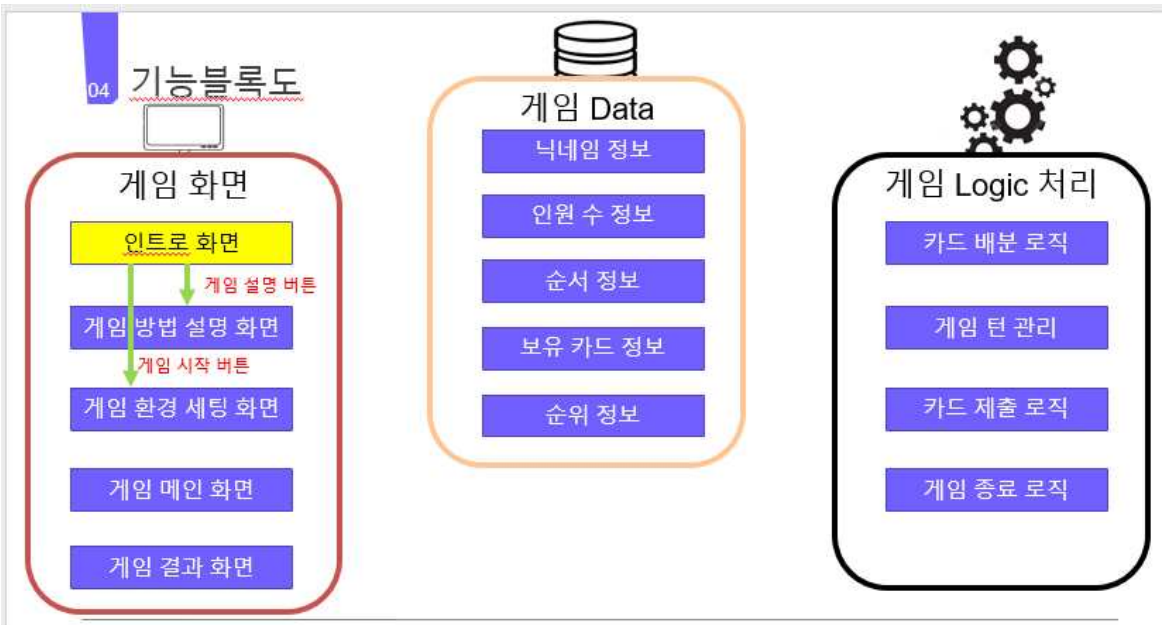
```
C:₩Users₩suneaW₩AppData₩L  ×  +  ⌄
프로그램을 종료합니다.
엔터 키를 눌러 메뉴로 돌아가세요...
```

8. 평가

8.1 정량적/정성적 목표달성도 평가

항목	목표값 (4장 항목별 목표값 참조)	달성률 (%)	비고
기술적 문제	그래픽 처리를 위한 자료구조와 알고리즘 최적화	100%	딜레이 없이 잘 구동 되며 색감 및 카운트 다운.
시간 제한 요소	코드 개발에 대한 시간적 문제	90%	운영체제 차이로 인해 실행간 차이점이 존재했음
보안적 요소	치팅 방지	100%	1대의 PC를 기준으로 치팅 방지가 카운트 다운으로 적용됨

8.3 기능적 요구사항 달성도 평가



기능블록도 내 요구사항 다 달성 했습니다.

8.2 문제점 및 해결방안

5.3- 룰 설명에 있어서 추가적인 애니메이션 효과가 필요할 것으로 교수님께서 판단-
PPT를 애니매이션 효과를 이용한 설명으로 해결.
5.15- 오버로딩으로 여러 개의 코드(12개)를 작성할 경우 코드가 더러워지지 않는지에 관한 질문-모듈화를 통해 코드를 깔끔히 유지.
-변수명, 함수명의 통일화 및 모듈화를 통해 코드 시각화 개선.

9. 추진체계

조장 (황현석): 팀 총괄
간트차트작성
기능블록도작성
요구사항명세서작성
흐름도작성
설계구조고안
PPT 작성 및 발표
외부 의사소통 (예상 질문LIST)
회의록 작성, PPT 작성
파일 분리, 깃허브사용법 공유 및 관리
개발 규칙 제작
Back-end파트구현
Front-end 코드 오류 솔루션 제시 및 수정
팀원 (이진석): Front-end 파트장
프로그램구조
콘솔UI 구성도작성
달무티소개도작성
달무티콘솔 화면 구현
전체 코드파일 취합, 수정및 구현
PPT 제작
팀원 (이승헌): Back-end 구현
게임전체 룰 정리
설계제안서 작성
자료조사
설문조사
외부 의사소통 (예상 질문LIST)
PPT 제작 보조
Back-end 구현, 카드 제출 로직 강화
Front- end 테스트 및 오류 수정
팀원 (김민석): Front-end 구현
설계 제안서작성
자료조사
외부 의사소통 (예상 질문LIST)
Front end 테스트 및 화면 로직 제작
게임 로직 제작 구현
테스트 케이스 제작
PPT 제작

10. 설계 추진 일정: 2023년 4월 ~ 2023년 06월

11. 결론

팀 협업의 중요성에 대해 느꼈습니다.

게임 개발은 종종 팀으로 진행되는 작업입니다. 팀원들과 함께 아이디어를 나누고 의견을 조율하는 과정은 매우 중요했습니다. 서로의 강점을 살리고 협력하여 게임을 개발하는 것은 팀 협업의 중요성을 깨달을 수 있었습니다.

디버깅과 테스트에 대해 중요성을 느꼈습니다.

게임 개발은 디버깅과 테스트 단계가 매우 중요합니다. 프로그램의 버그를 찾고 수정하는 과정은 시간이 많이 소요되기도 했지만, 게임의 안정성을 보장하기 위해서는 필수적인 작업이었습니다. 디버깅과 테스트를 통해 문제를 해결하고 c언어 프로그래밍 실력을 향상시킬 수 있었습니다.

c++이나 자바를 이용한 것이 아닌 순수 c언어로 결과물을 만들고 실행시켜보는 것으로 많은 경험과 성장을 이뤄나갈 수 있었습니다.