

2020-1

# 졸업프로젝트 1

2020-06-04 중간발표2 : Iteration 2nd

# AR Card Battle

---

## ■ Team

Team3

## ■ Team Member

조승현  
201611300  
컴퓨터공학과

장혁준  
201310805  
기계공학과

현인수  
201512755  
산업디자인학과

## ■ INDEX

1. Summary
2. Iteration2 Demo
3. Pass Fail criteria
4. Detail
5. Traceability
6. Schedule



기존의 온라인 카드게임인 하스스톤은 카드를 수집하는 재미가 있지만 실물이 아닌 게임 상에서의 카드이며, 입체가 아니기 때문에 수집의 재미가 떨어진다.

포켓몬카드의 경우 실물카드를 모을때의 수집 재미는 있으나 다른사람과 카드게임을 하는데 있어서 재미가 떨어진다.

## 프로젝트 동기 Summary

---

기존의 실물카드 게임은 카드를 수집하는 재미가 있지만 시각적인 재미가 부족하다.

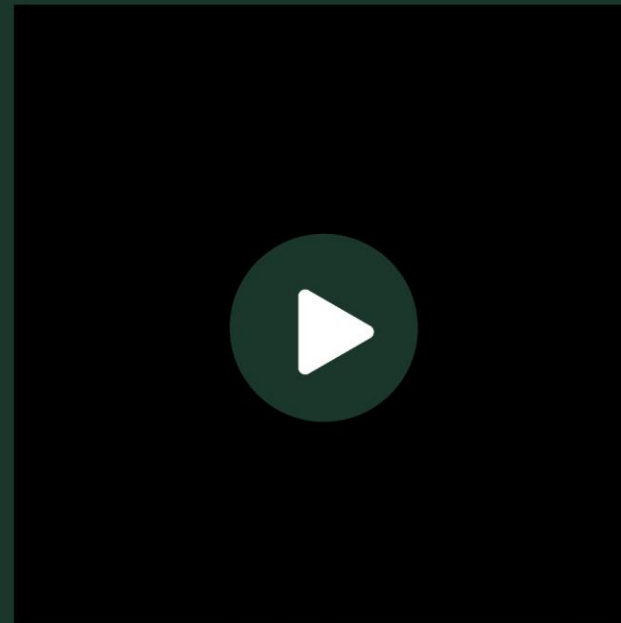
이를 컴퓨터 게임으로 만들 경우 시각적 재미는 있지만 실물이 없어 수집하는 재미가 떨어진다.

AR 기술로 수집하는 재미와 시각적인 재미를 모두 챙길 수 있는 게임을 개발하고자 한다.

# DEMO\_2

---

영상



파일 첨부

# PassFail Criteria

Identifier	Feature	Valid Value
1.1		
2.1	카메라를 통해 캡처된 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다	List<Contour> 중 사각형 영역 존재 해당 사각형이 다른종류 사각형이 아닌 태그 사각형임.
2.2	코드를 찾았을 경우 코드의 4가지 방향중 알맞은 방향으로 설정한 후 코드를 해독한다.	10bit의 Integer 값
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트해준다.	Update 된 <u>값</u> 확인
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X - 미구현	플레이어 차례를 나타내는 글로벌 변수 업데이트 확인
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	Update 된 <u>값</u> 확인
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다. X - 미구현	0~3 <u>까지의</u> Integer Value
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어온다.	카드의 정보를 담은 data Class를 얻었는지 확인
8		
9	카드를 신규 제작하였을때, 해당 카드 정보를 DB상에 추가할 수 있고, 중복된 데이터는 실패처리한다.	콘솔 로그 메시지로 db 쿼리 수행결과 확인
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당 카드를 3초 이내에 인식 완료하여야 한다.	타이머 경과시간 <= 3sec
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 <u>필요소들</u> 가시성과 가독성이 좋게 배치한다.	-

Iteration 1	Iteration 2	Iteration 3
Pass	Pass	
Fail	Pass	
Fail 50%	Fail 50%	모듈간 연결 미구현
Fail	Fail 50%	모듈간 연결 미구현
Fail	Fail 50%	모듈간 연결 미구현
Fail	Fail 50%	모듈간 연결 미구현
Fail 50%	Fail 50%	모듈간 연결 미구현
Pass	Pass	
Fail	Fail	영상처리 정확도
Pass	Pass	

## Iteration 2 구현 내용

1. 인코딩 했을 때 원하는 값을 받는다.
2. 디코딩 했을 때 알맞는 값을 받는다.
3. 코드의 방향에 관계없이 상하좌우를 읽어 값을 받는다.
4. 게임로직 완성
5. DB 완성
6. 게임 에셋 수집, 제작 완성.

## Iteration 2 진행중

1. OpenCV = GameModule 연결중
2. Decode - OpenCV - GameModule 연결중
3. Unity - GameModule 연결중

22 [0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0]

Data Code	HammingCode	CRC	f(x)	
10bit	4bit	8bit	5bit	: 27bit

```
/[0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
void Testing(){
    int result=0;
    byte[] sample = new byte[]{
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 0, 0,
        1, 1, 0, 1, 0, 1,
        0, 0, 0, 0, 1, 0,
        0, 0, 1, 0, 0, 0,
        0, 0, 1, 0, 0, 0};
    result = Decode(sample);
}
```

```
2020-06-04 16:12:19.582 1448-1448/kr.ac.konkuk.arcadbattle D/InsuLog2: [0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
2020-06-04 16:12:19.583 1448-1448/kr.ac.konkuk.arcadbattle D/InsuLog2: 22bit [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0]
2020-06-04 16:12:19.583 1448-1448/kr.ac.konkuk.arcadbattle D/InsuLog2: Success : hammingcode
2020-06-04 16:12:19.583 1448-1448/kr.ac.konkuk.arcadbattle D/InsuLog2: 18bit [0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
2020-06-04 16:12:19.583 1448-1448/kr.ac.konkuk.arcadbattle D/InsuLog2: testteststset 22
```



## Iteration 2 진행중

---

### 1. OpenCV = GameModule 연결중 및 디버깅

- 화면 분할 및 알파문제

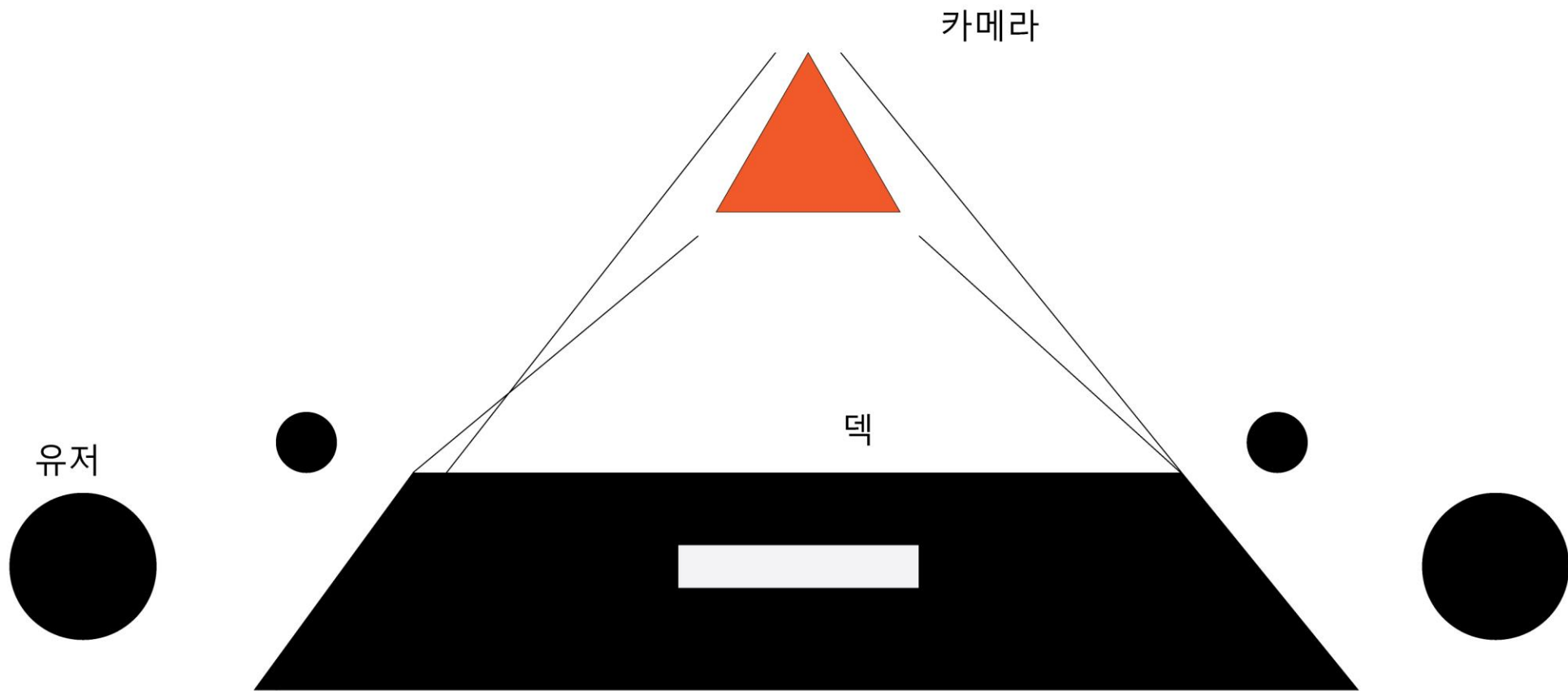
### 2. Decode - OpenCV - GameModule 연결중

- 디코더 모듈에 잘못된 코드 수정하면서 진행중

### 3. Unity - GameModule 연결중

- 유니티 모델 중복문제 해결중

연결 이후 테스트 진행하면서 디버깅.



참여한 유저는 손에 카드를 들고 각자의 턴에 맞게 가운데에 카드를 낸다.  
카드는 덱에서 뽑게 된다.  
카메라는 사용자가 낸 카드의 2차원코드를 인식하고 이를 바탕으로 게임이 진행되며  
라이프가 0이 되면 패배하고 마지막까지 남으면 승리하게된다.

Identifier	Feature	Valid Value
1.1		
2.1	카메라를 통해 <u>캡처된</u> 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다	List<Contour> 중 <u>사각형</u> 영역 존재. 해당 <u>사각형이</u> 다른종류 <u>사각형이</u> 아닌 태그 <u>사각형임</u> .
2.2	코드를 찾았을 경우 코드의 4가지 <u>방향중</u> 알맞은 방향으로 설정한 후 코드를 해독한다.	10bit의 Integer 값
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트해준다.	Update 된 <u>비트</u> 확인
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X - <u>미구현</u>	플레이어 차례를 나타내는 글로벌 변수 업데이트 확인
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	Update 된 <u>비트</u> 확인
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다. X - <u>미구현</u>	0~3 <u>까지의</u> Integer Value
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어온다.	카드의 정보를 담은 data Class를 <u>얻었는지</u> 확인
8		
9	카드를 신규 <u>제작하였을때</u> , 해당 카드 정보를 DB상에 추가할 수 있고, 중복된 데이터는 <u>실패처리</u> 한다.	콘솔 로그 메시지로 db 쿼리 수행결과 확인
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당 카드를 3초 이내에 인식 완료하여야 한다.	타이머 경과시간 <= 3sec
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 <u>비율소들을</u> 가시성과 가독성이 좋게 배치한다.	-

연결중

연결중

영상처리 정확도

:적어도 화면크기대비 1/30사이즈여도 인식이 잘 되어야 한다.

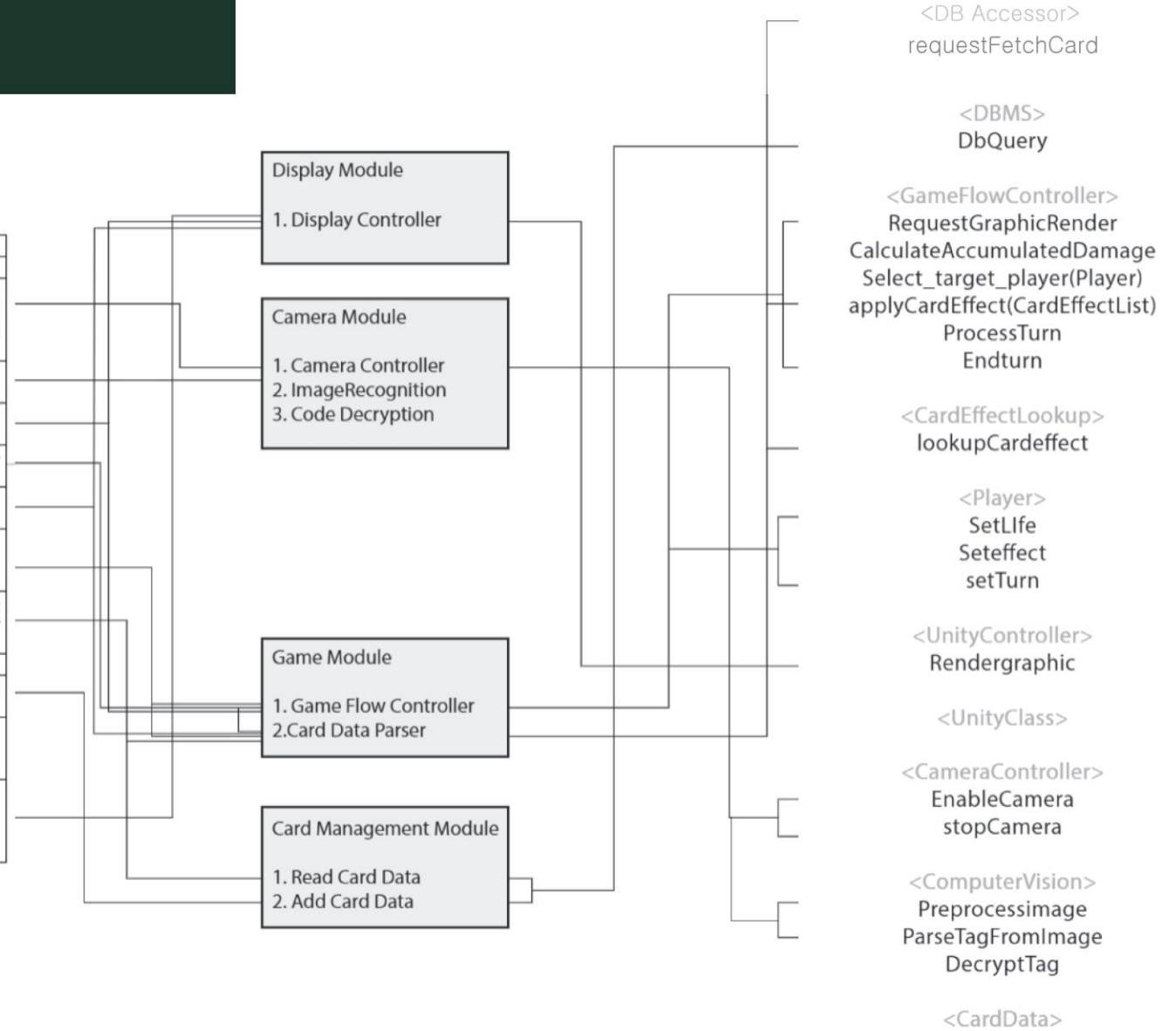


# Traceability

## Test Case

0518 1차 데모

Identifier	Feature	Valid Value
1.1		
2.1	카메라를 통해 캡처된 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다	List<Contour> 에 사각형 영역 존재, 해당 사각형이 다른종류 사각형이 아닌 태그 사각형임.
2.2	코드를 찾았을 경우 코드의 4가지 방향에 알맞은 방향으로 설정한 후 코드를 해독한다.	10bit의 Integer 값
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트해준다.	Update 된 <u>값을 확인</u>
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X - 미구현	플레이어 자제를 나타내는 글로벌 변수 업데이트 확인
5	플레이어가 실물 카드를 넣을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	Update 된 <u>값을 확인</u>
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다. X - 미구현	0~3 까지의 Integer Value
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어준다.	카드의 정보를 담은 data Class를 만였는지 확인
8		
9	카드를 신규 제작하였을때, 해당 카드 정보를 DB상에 추가할 수 있고, 중복된 데이터는 <u>실패처리한다.</u>	콘솔 로그 메시지로 db 쿼리 수행결과 확인
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당 카드를 3초 이내에 인식 완료하여야 한다.	타이머 경과시간 <= 3sec
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 <u>값요소들을</u> 가시성과 가독성이 좋게 배치한다.	-



# Schedule

---

Iteration 1

Iteration 2



Pass

Pass

Fail

Pass

Fail 50%

Fail 50%

Fail

Fail 50%

Fail

Fail 50%

Fail

Fail 50%

Fail 50%

Fail 50%

Pass

Pass

Fail

Fail

Pass

Pass

모듈간 연결 미구현  
모듈간 연결 미구현  
모듈간 연결 미구현  
모듈간 연결 미구현

영상처리 정확도

실제 게임 테스트까지 완벽하게 구현.