ver2

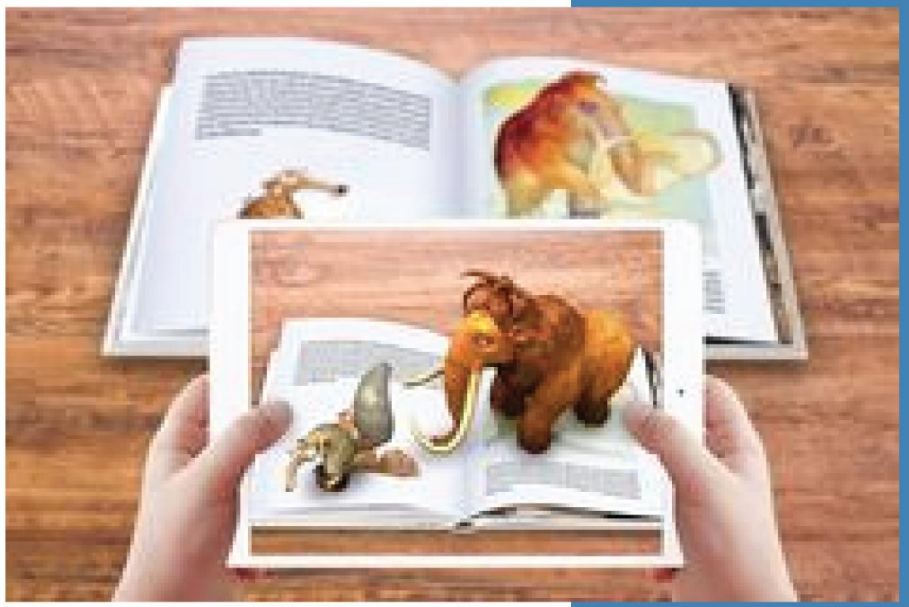
2020_1
졸업프로젝트1

AR CARD BATTLE

T3

컴퓨터공학과 _ 201611300 _ 조승현
기계공학과 - 201310805 - 장혁준
산업디자인학과 _ 201512755 _ 현인수

CONTENT

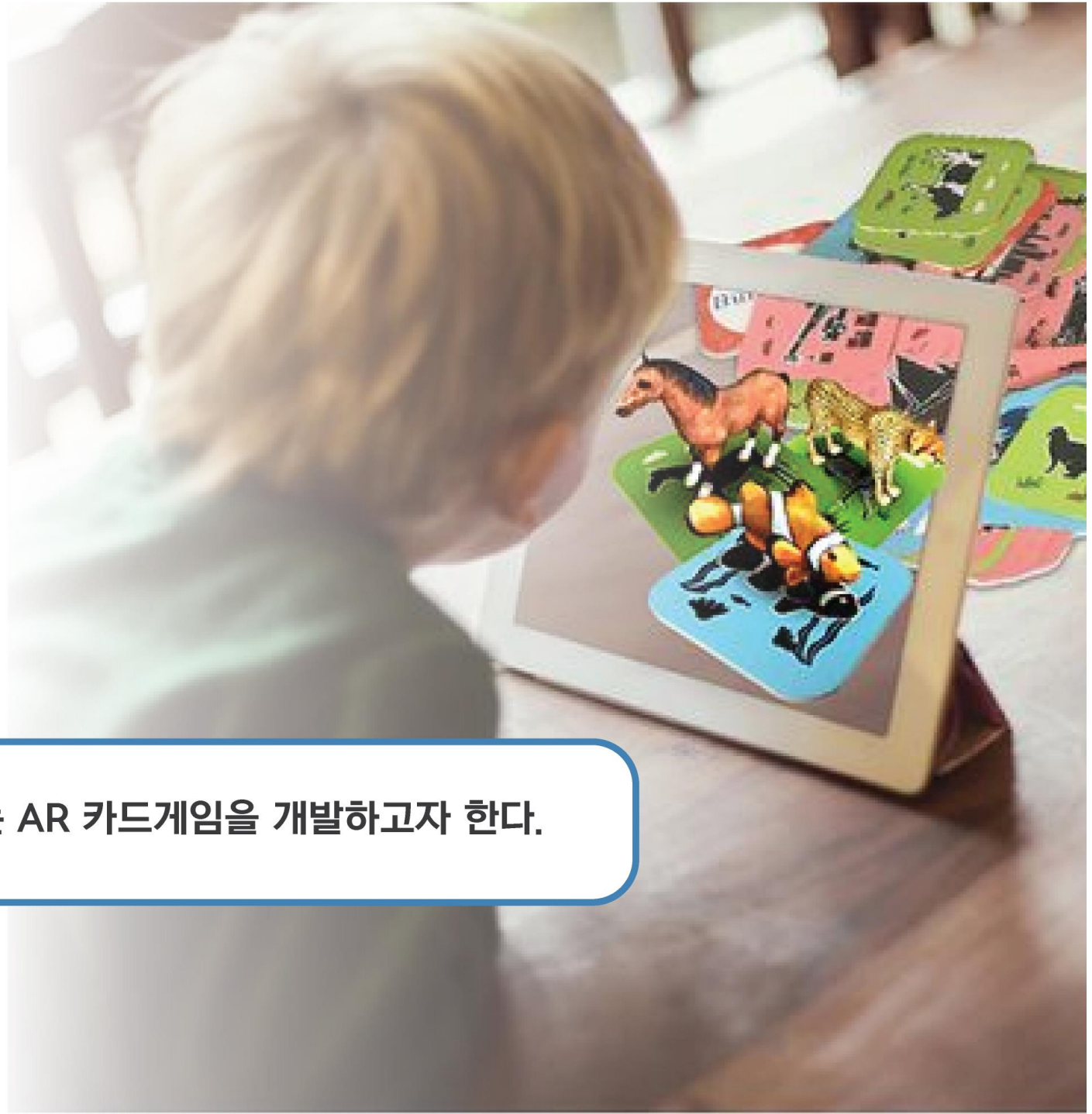


Purpose

WHY?

1. 실물 카드게임에서 다소 부족한
시각적 재미

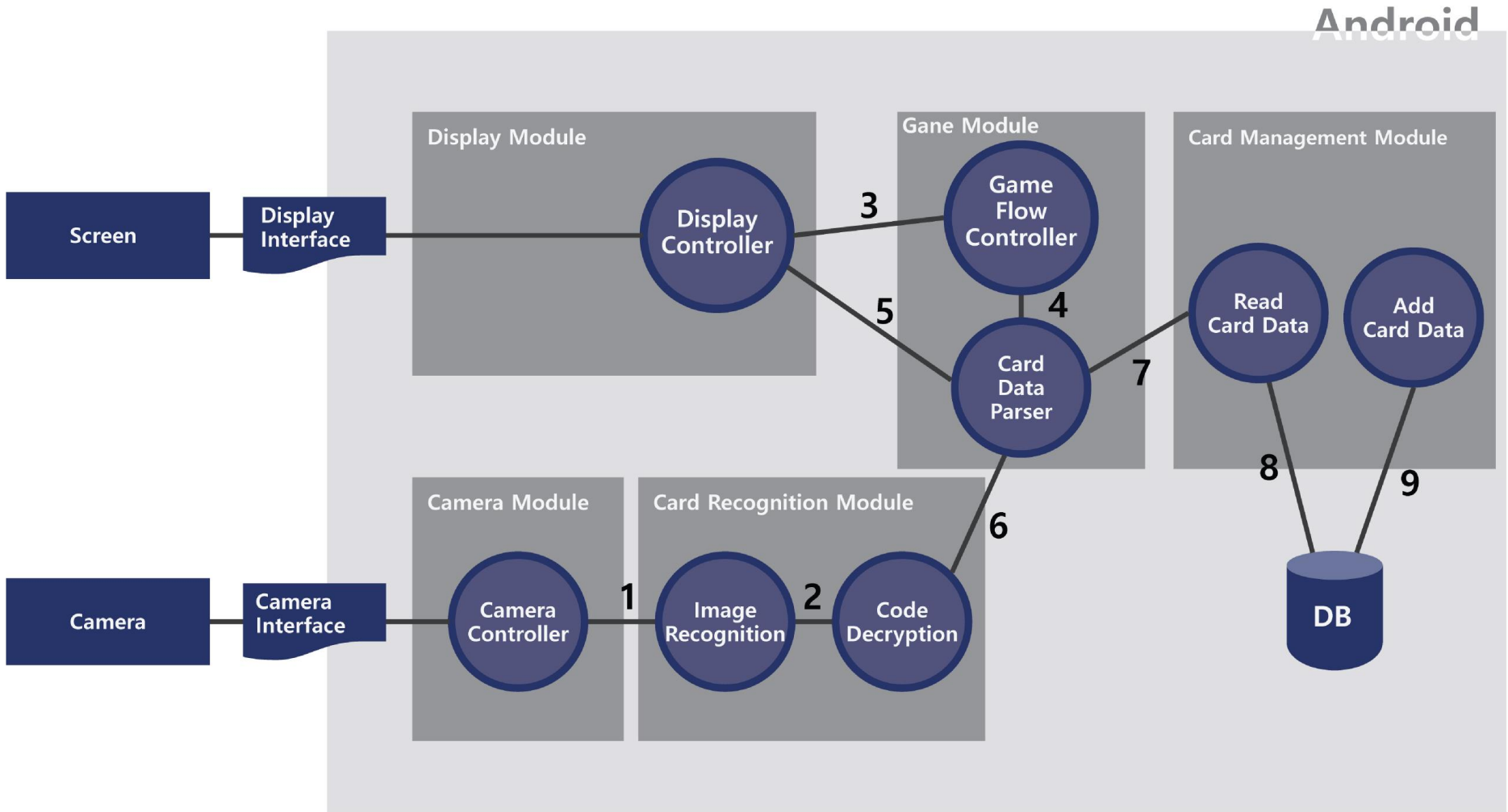
2. 온라인 카드게임에서 다소 부족한
수집적 재미



두가지 재미를 모두 챙기는 AR 카드게임을 개발하고자 한다.

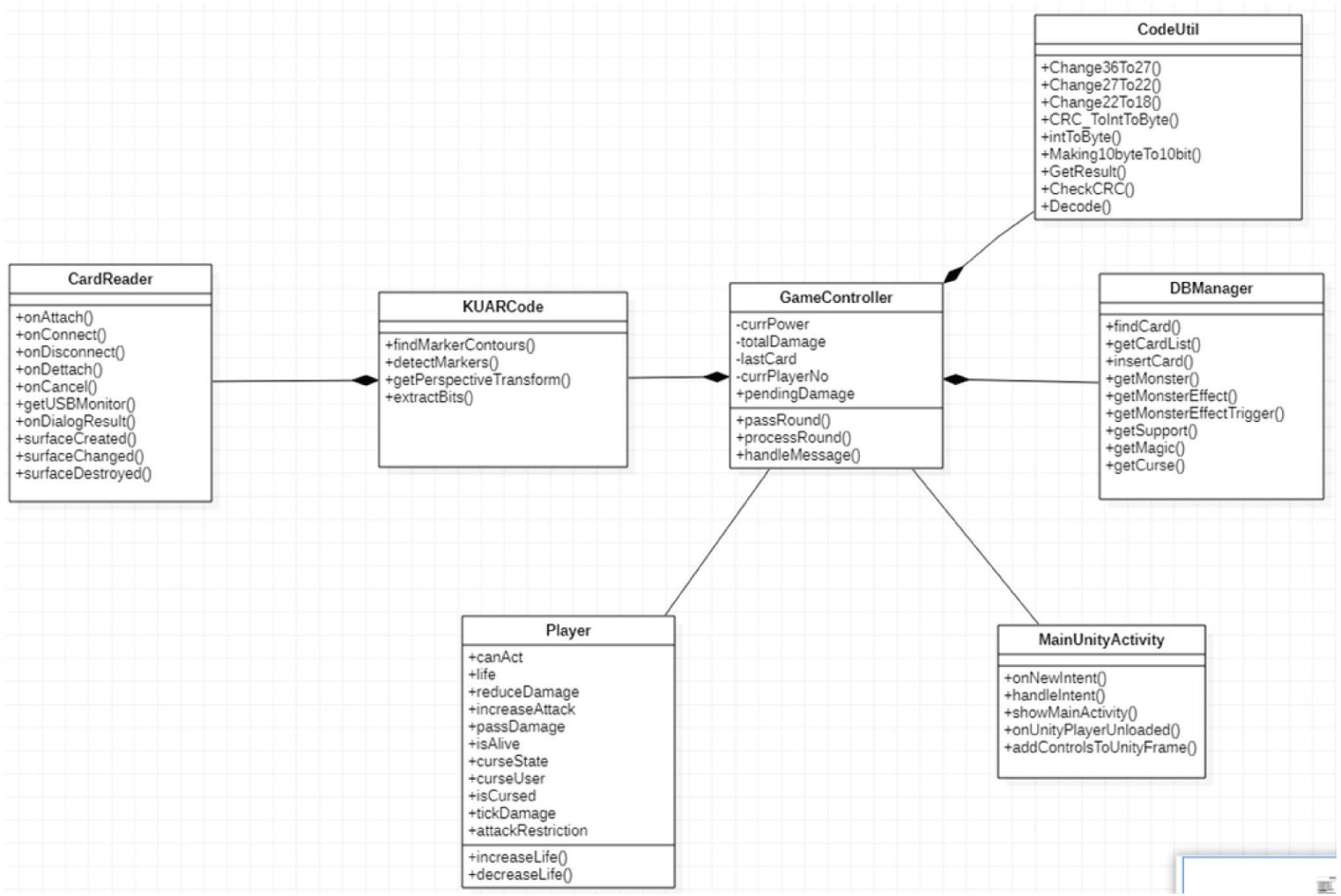
DEMO

Context Diagram



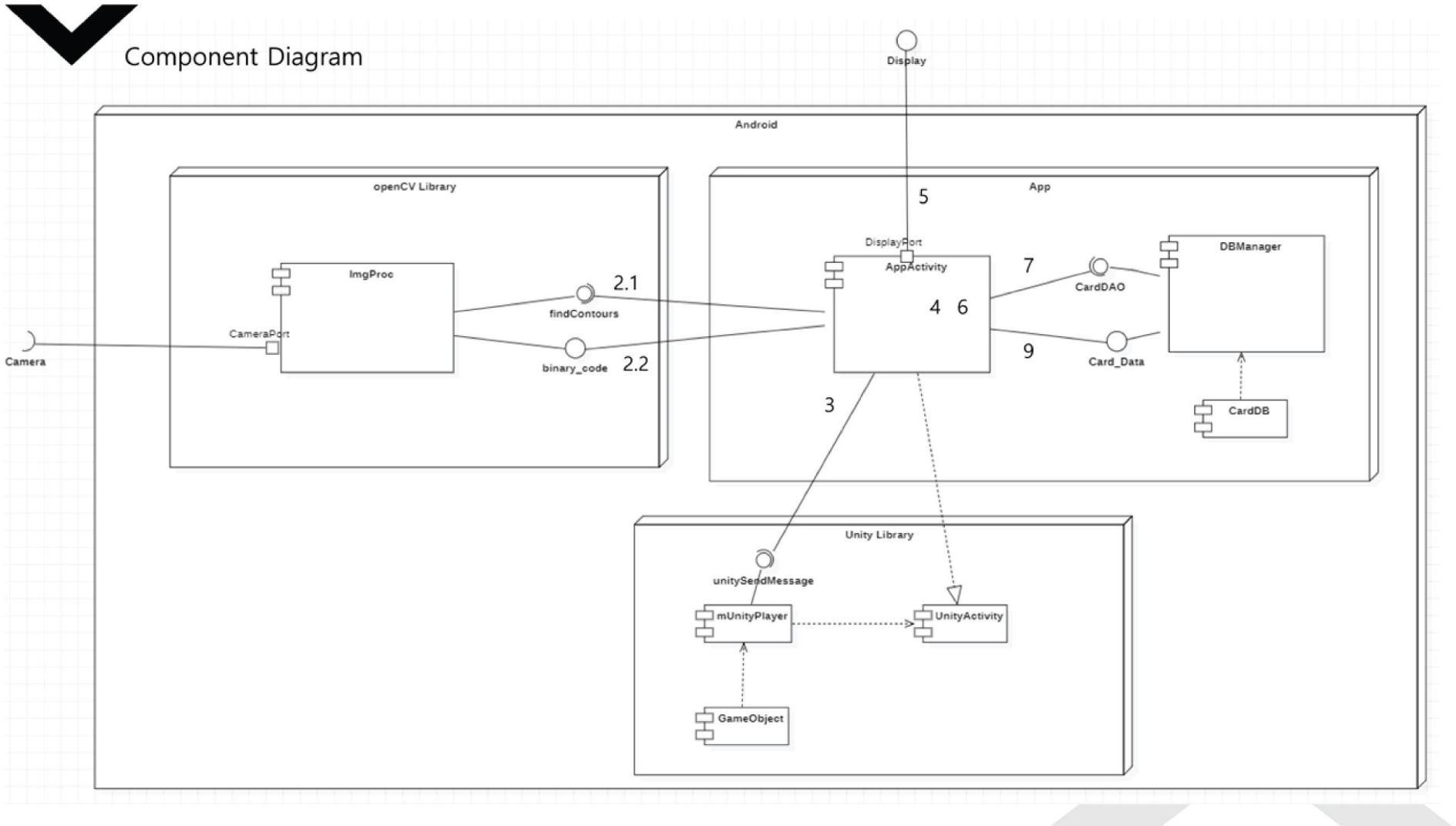
Camera Interface를 시작으로 영상을 받아와 영상처리를 진행한다. 이 후, 코드 발견시 복호화작업을 거쳐 GameLogic에 값을 전달한다.
진행 중에 코드가 들어올 경우 DB에 접근하여 알맞는 정보를 받아 게임을 진행한다. DB에서 받은 값을 통해 Interface의 Unity Library 3D 모델을 화면에 띄울 수 있다.

Class Diagram



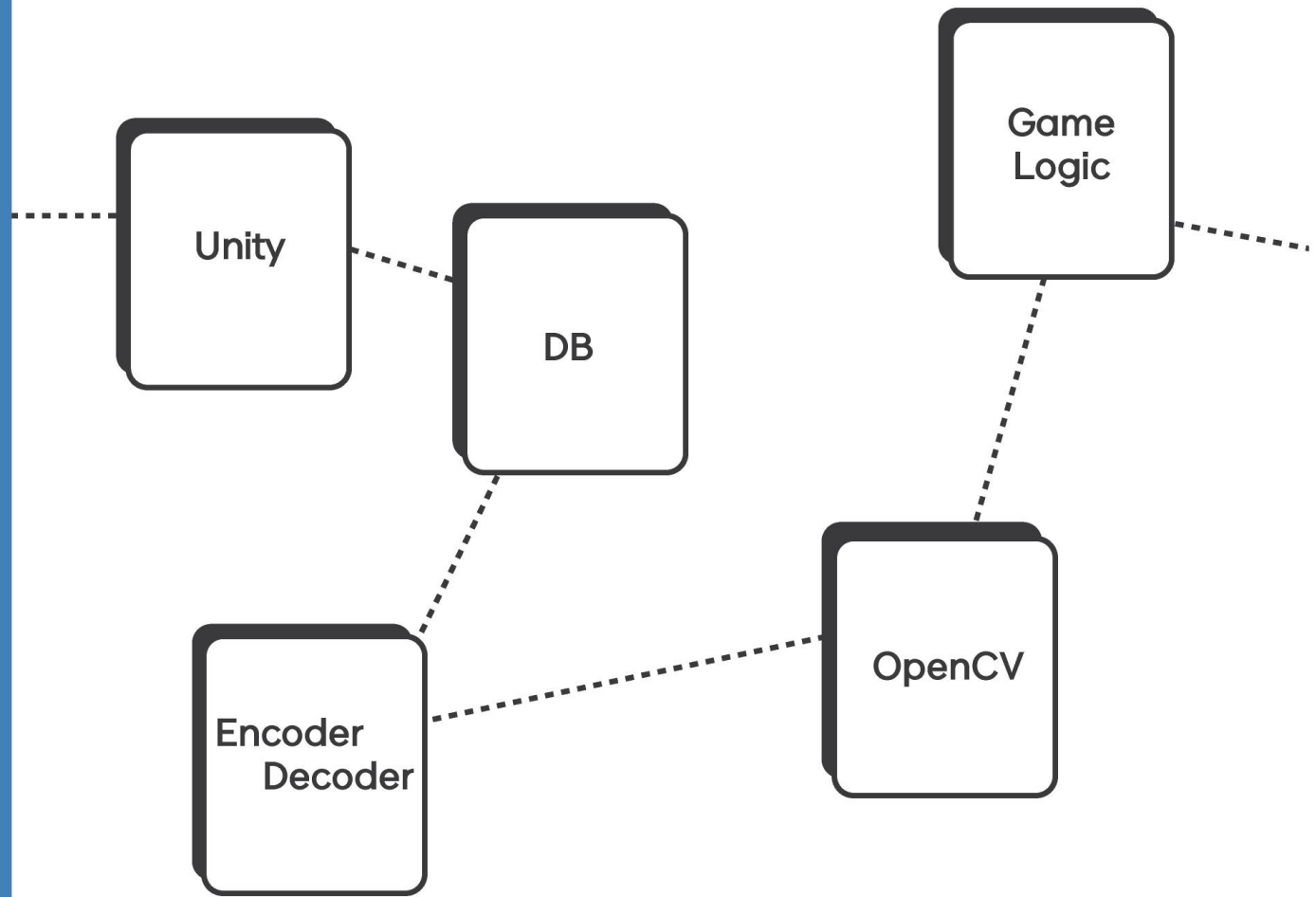
각각의 클래스들은 Context Diagram과 비슷하게 Game Controller로 값을 전달해 처리할 수 있게 설계 했다. DB도 DBManager를 통해 가져올 수 있도록 설계했다.

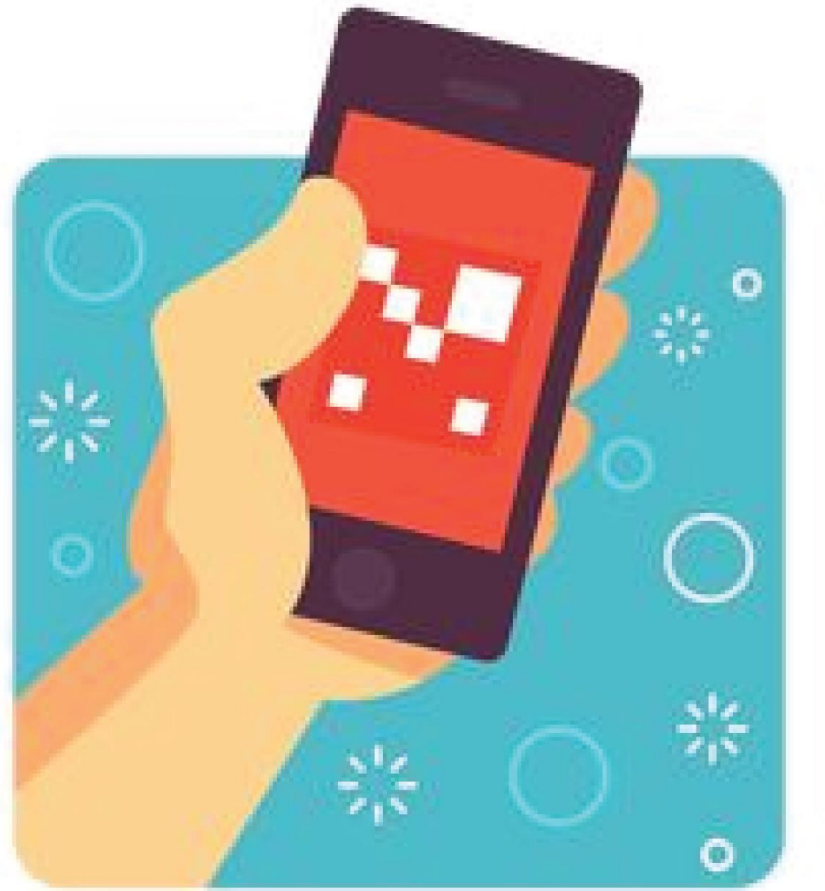
Component Diagram



각 클래스들이 어떻게 상호작용하는지 Component Diagram을 통해 확인한다.

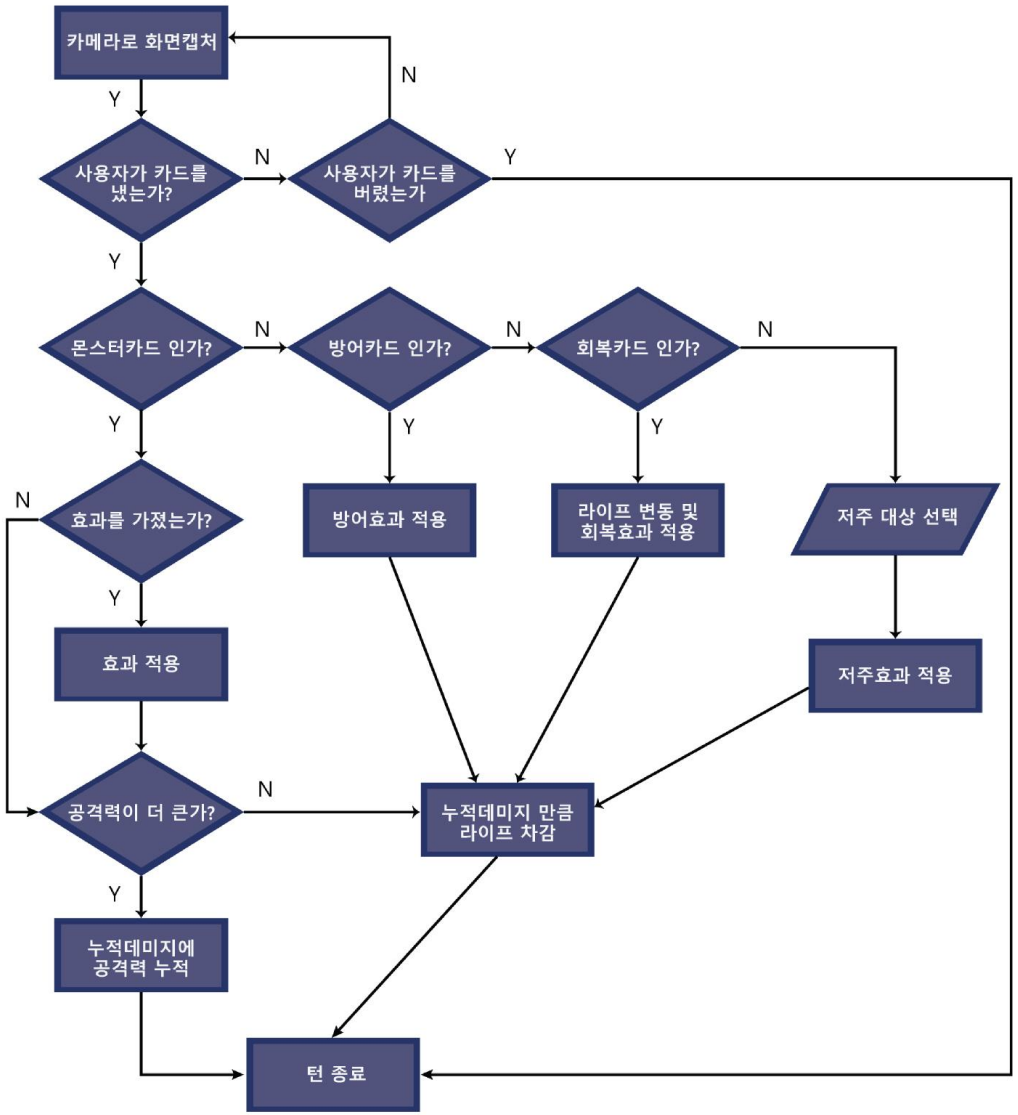
개발과정





AR 카드배틀을 만들기 위해 Unity, OpenCV, UVCcamera API를 이용하여 GameLogic, Decoder, DB, 영상처리등을 구현했다.

Game Logic



작성했던 Flow Chart에 맞게 규칙을 정의하고 개발했다.

요약

- 게임 로직에 따라 게임이 진행되는지 확인한다.
- 플레이어 턴 변경 확인
- 누적공격력을 받을 경우 생명력이 깎이는지 확인
- 공격카드를 냈을 때 누적 공격력이 카드의 공격력만큼 증가하는지 확인
- 몬스터카드 / 보조카드 / 마법카드 / 저주카드가 속성에 맞게 작동하는지 확인

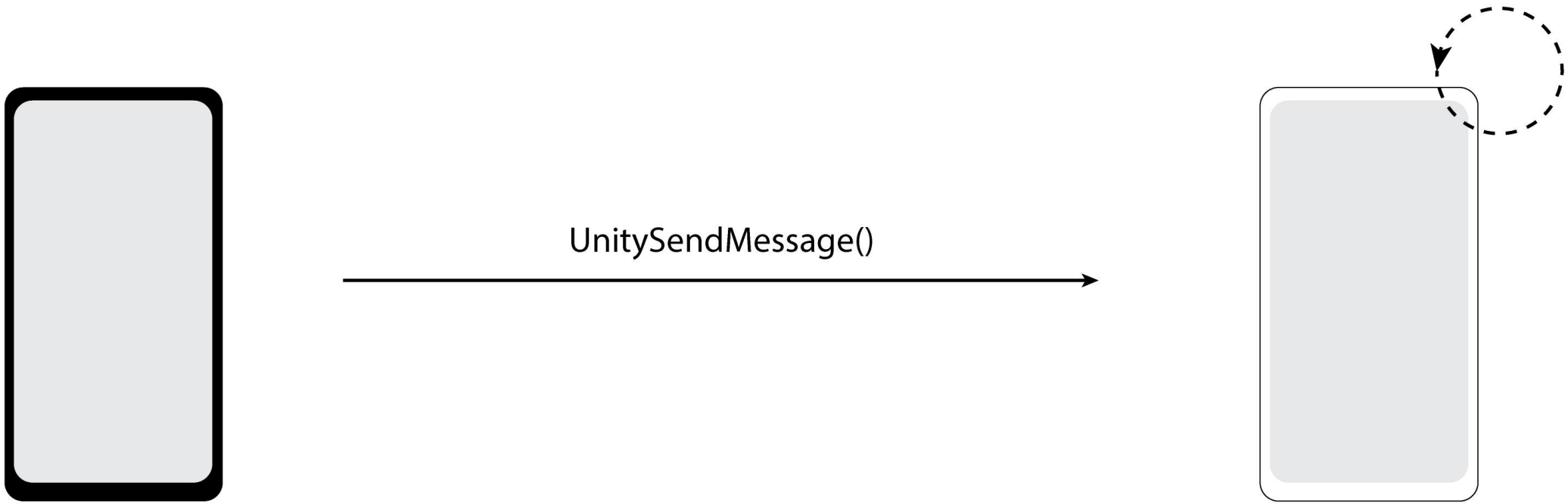
현재 카드는 총 50장을 뽑아 사용중이며, 필요에 따라 2^10 만큼 카드를 만들어 사용할 수 있다.

본 게임의 카드는 4가지 종류의 카드로 나누어 정의한다.

+ α

DB

4개의 카드에 대하여 DB를 작성했다. 4개의 카드 모두 Card_ID라는 고유의 FK를 가지고 있고, 몬스터카드의 경우 특수한 효과를 가진 경우 추가로 관계를 연결해 새로운 테이블에서 사용했다.



Asset > Resources > Models

Model 1. 2, 3 , 4, ...

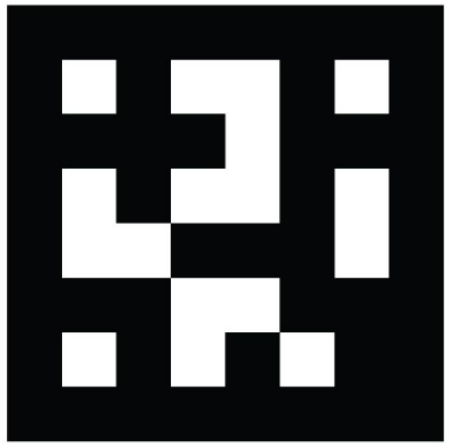
OpenCV 3.4.1 -> OpenCVwithUVCcamera Library

ImageProc, Core . . . 패키지 라이브러리. -> 영상보정처리

회색조 전환	보다 더 정밀한 보정을 하기 위해서는 우선적으로 회색조 전환이 필요하다.
Gaussian blur	주변의 다양한 빛과 물체처리를 보다 더 용이하기 위해 Blur 효과를 주어 Noise를 제거한다.
Delate	하얀 부분을 더욱 강조한다. 이를 흔히 퇴적과정이라고 한다.
Erode	Delate로 인해 부푼 부분의 외곽을 집어 넣어주는 역할을 한다. 이를 흔히 침식과정이라고 한다.
Brighness	물체 인식에 있어 가장 중요한 것이 환경인데 어떤 환경이냐에 따라 설정해주기 좋은 보정이다.
Contrast	Brightness와 마찬가지로 주변환경에 따라 설정해주기 용이한 보정이다.
Threshold	보정된 값을 0과 1로 양분하는 과정이다. 이를 통해 시스템내에서 사용할 수 있는 자료가 만들어진다.

이렇게 만들어진 값을 통해 물체 내에서 특정한 무언가를 뽑아낼 수 있다. 이를 컨투어 검출이라고 하며 본 프로젝트의 과정에서
본 팀에 맞는 컨투어 후보를 얻기 위한 작업을 진행했다.

OpenCV를 통해 받아진 2차원코드의 Decoding 과 Encoding.



AR카드배틀에 쓰이는 2차원코드

1. 2D 코드 개발. (8x8)

Code 영역 검출을 위해 가장자리의 27Bit을 할당한다.

상하좌우 식별을 위한 인식점 3bit와 그 bit를 감싸는 검정 bit를 합쳐 총 9bit를 할당한다.

총 10비트의 데이터정보를 위해 할당하여 약 2^10 즉, 1024장의 카드 데이터를 입력할 수 있다.

데이터 암호화를 위해 CRC 코드 8bit , Hamming Code 4 bit와 f(x) 5bit 를 추가한다.

=> 이로써 8x8 총 64bit 2차원 코드가 생성된다.

2. 2D 코드 해독. (8x8)

코드의 해독은 Encoding과 정 반대의 순서로 이루어진다.

코드가 해독되는 과정에서 전송문제와 옳지 않은 bit가 발견되지 않았다면 이를 최종 2차원비트로 확정짓고

이를 GameController에 보내면 Game Controller는 DB에 해독된 값을 보내 알맞은 데이터를 받아 처리하게 된다.

TEST CASE

	Feature	Valid Value
1.1	: Iteration 1중 동일한 FR로 삭제.	
2.1	카메라를 통해 캡쳐된 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다.	List <Contour> 중 사각형 영역 존재, 해당 사각형이 다른종류의 사각형이 아닌 태그 사각형이어야 한다.
2.2	코드를 찾았을 경우 코드의 4ㅏ지 방향 중 알맞은 방향으로 설정한 후 코드를 해독한다.	10bit의 Integer 값
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트 한다.	Update 된 UI를 확인한다.
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X	플레이어 차례를 나타내는 글로벌 변수 확인.
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	Update 된 UI를 확인
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다.	0~3 까지의 Integer value
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어온다.	카드의 정보를 담은 data class를 얻었는지 확인
8	Iteration 1 중 동일한 FR로 삭제.	
9	카드를 신규 제작시, 해당 카드 정보를 DB상에 추가할 수 있고, 중볼 데이터는 실패처리한다.	콘솔 로그 메시지로 db 쿼리 수행결과 확인
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당카드를 3초 이내에 인식 완료하여야 한다.	타이머 3초 이내에 인식완료.
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 UI 요소를 가시성과 가독성이 좋게 배치한다.	-

TEST CASE

		지난 Iteration 1, 2	
1.1	: Iteration 1중 동일한 FR로 삭제.		
2.1	카메라를 통해 캡쳐된 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다.	P	P
2.2	코드를 찾았을 경우 코드의 4ㅏ지 방향 중 알맞은 방향으로 설정한 후 코드를 해독한다.	F	P
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트 한다.	F -50%	F -50%
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X	F	F -50%
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	F	F -50%
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다.	F	F -50%
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어온다.	F -50%	F -50%
8	Iteration 1 중 동일한 FR로 삭제.		
9	카드를 신규 제작시, 해당 카드 정보를 DB상에 추가할 수 있고, 중볼 데이터는 실패처리한다.	P	P
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당카드를 3초 이내에 인식 완료하여야 한다.	F	F
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 UI 요소를 가시성과 가독성이 좋게 배치한다.	P	P

Final Success Criteria

1.1	: Iteration 1중 동일한 FR로 삭제.	
2.1	카메라를 통해 캡쳐된 영상을 분석하여 해당 이미지에 코드 태그가 존재하는지 분석한다.	P
2.2	코드를 찾았을 경우 코드의 4ㅏ지 방향 중 알맞은 방향으로 설정한 후 코드를 해독한다.	P
3	플레이어 라이프나 상태에 변동이 생기는 경우, 변경된 정보로 업데이트 한다.	P
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다.	P
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	P
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다.	F 70%
7	Integer 데이터를 이용해 데이터베이스에서 카드 정보를 읽어온다.	P
8	Iteration 1 중 동일한 FR로 삭제.	
9	카드를 신규 제작시, 해당 카드 정보를 DB상에 추가할 수 있고, 중볻 데이터는 실패처리한다.	P
10	사용자가 카드를 테이블에 내려놓은 순간부터 시작해서 카메라는 해당카드를 3초 이내에 인식 완료하여야 한다.	P
11	이 게임을 한번도 해보지 않은 플레이어도 화면을 보고 1초 이내에 수치를 읽을 수 있도록 라이프와 데미지의 UI 요소를 가시성과 가독성이 좋게 배치한다.	P

6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다.	F 70%

- OpenCV 영상처리 값의 조절과 다양한 환경에서의 테스트 부재로 인해 카드의 정확한 Bit 값을 읽어오지 못했다.

- 알맞은 카드의 정보가 읽혀올 경우 알맞은 처리를 진행하지만 한 칸에 해당하는 픽셀정도의 차이로 잘못 읽는 경우가 있었다.
  그런 경우 알맞는 처리가 이루어질 수 없다.

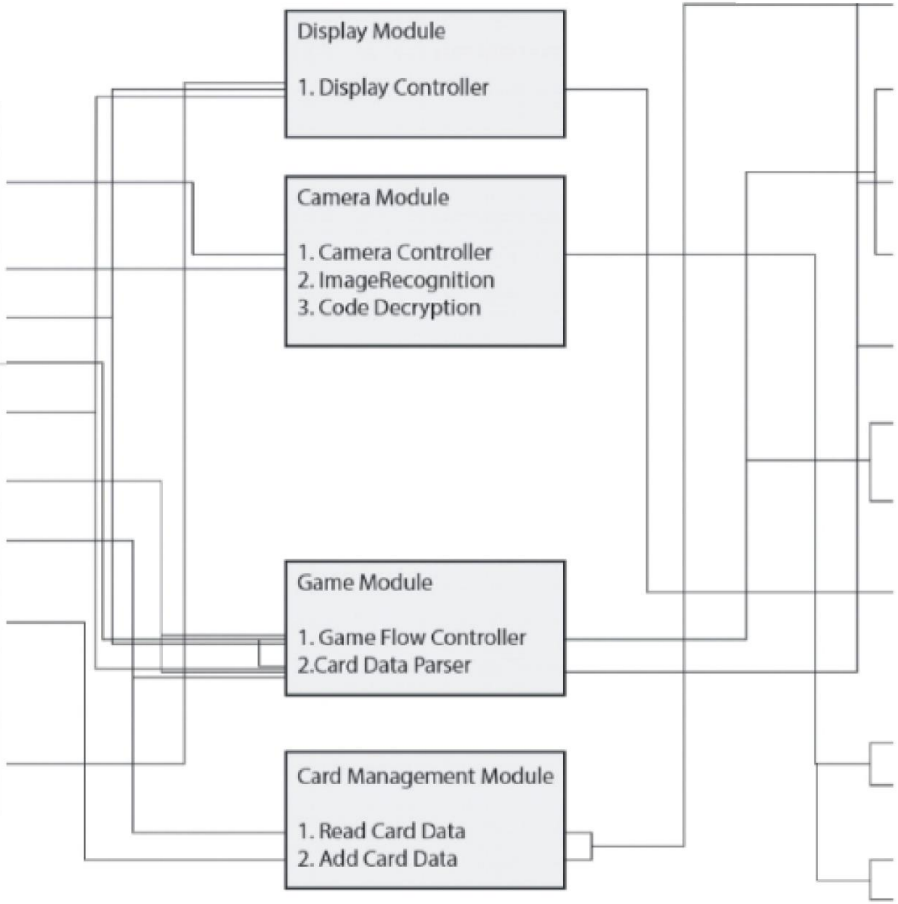
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	P

- 처음 시도한 방법은 런타임 중 몬스터를 소환하는 것이었으나 비효율적이라 코드를 미리 소환해놓고 보였다가 안보였다가 하는 방법으로 진행하였다. 그러나 Final 발표까지 화면에 그래픽이 보이지 않았으나 중간에 코드를 검사해본 결과 유니티 C#코드에 문제가 발견되었고, 이를 해결함으로써 Pass 하게 되었다.

Test Case

0518 1차 데모

Identifier	Feature	Valid Value
1.1		
2.1	카메라를 통해 캡쳐된 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 UI를 확인
4	읽어온 카드 데이터를 가지고 알맞은 행동을 취하며 게임을 진행시킨다. X - 미구현	플레이어 자례를 나타내는 글로벌 변수 업데이트 확인
5	플레이어가 실물 카드를 냈을 경우, 해당 카드의 그래픽을 화면에 띄워준다.	Update 된 UI를 확인
6	카드 정보를 읽어 어떤 종류의 카드인지 분석한다. X - 미구현	0~3 까지의 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 UI요소들을 가시성과 가독성이 좋게 배치한다.	-



DbQuery

<GameFlowController>
RequestGraphicRender
CalculateAccumulatedDamage
Select_target_player(Player)
applyCardEffect(CardEffectList)
ProcessTurn
Endturn

<CardEffectLookup>
lookupCardeffect

<Player>
SetLIfe
Seteffect
setTurn

<UnityController>
Rendergraphic

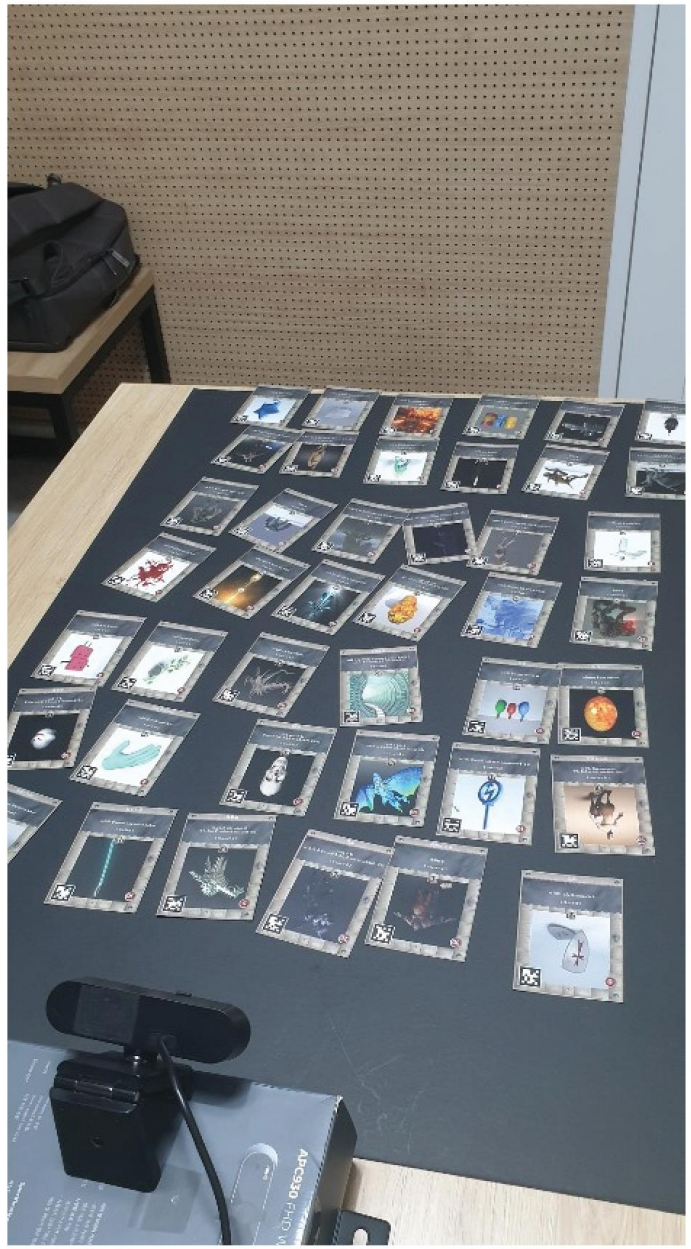
<UnityClass>

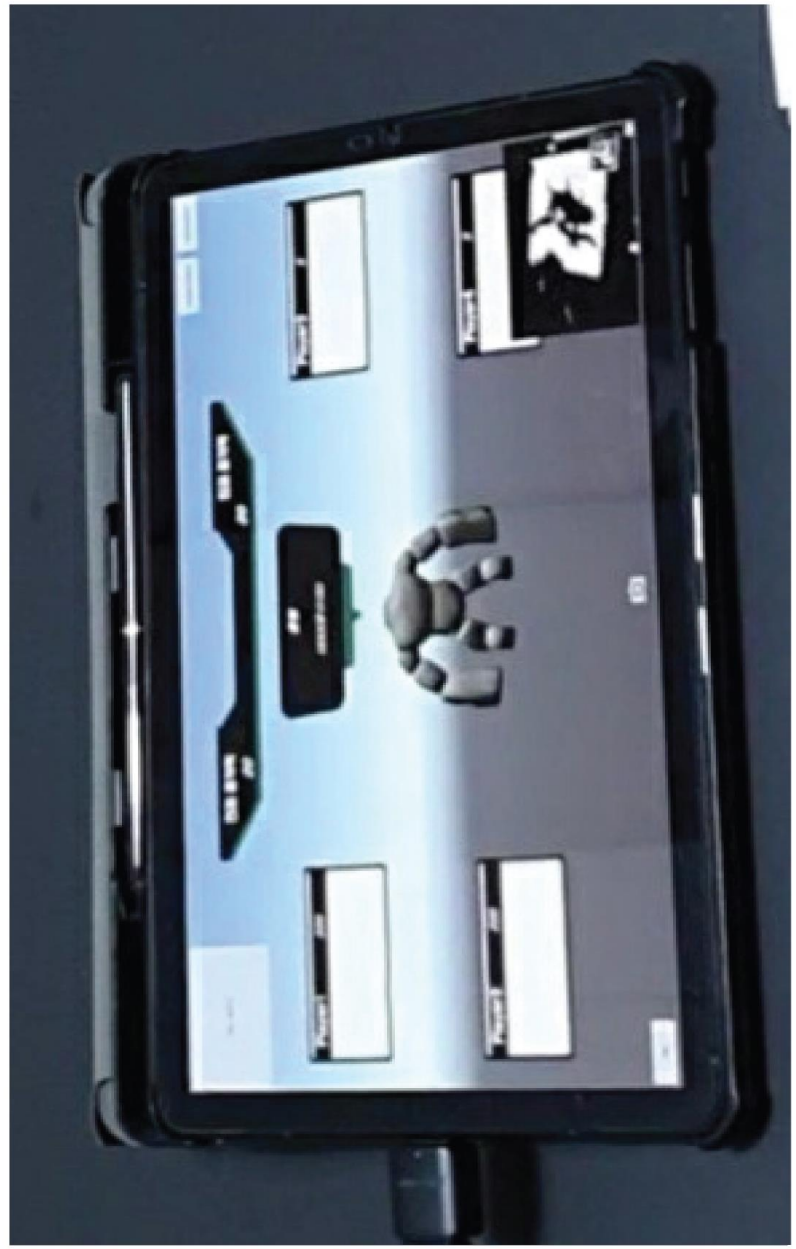
<CameraController>
EnableCamera
stopCamera

<ComputerVision>
Preprocessimage
ParseTagFromImage
DecryptTag

<CardData>


10
3. 공격력
6. 2차원코드
1
1. 카드번호
[몬스터카드]
- 4. 카드종류
다음 플레이어는 효과가 있는 몬스터 카드를
낼 수 없다.
- 5. 카드효과
고스트
2. 카드이름





감사합니다.