

WESC-16-09-개발완료보고서

# Memory2Pie

참가부문 : 스마트 챌린지 LG전자

팀 명 : 힘을내요, SSUper파월

팀구성원

| No. | 구분 | 성명  | 소속명           | No. | 구분 | 성명 | 소속명 |
|-----|----|-----|---------------|-----|----|----|-----|
| 1   | 팀장 | 김남엽 | 송실대 정보통신전자공학부 | 6   | 팀원 |    |     |
| 2   | 팀원 | 김동우 | 송실대 정보통신전자공학부 | 7   | 팀원 |    |     |
| 3   | 팀원 | 윤홍선 | 송실대 정보통신전자공학부 | 8   | 팀원 |    |     |
| 4   | 팀원 | 연지영 | 송실대 정보통신전자공학부 | 9   | 팀원 |    |     |
| 5   | 팀원 |     |               | 10  | 팀원 |    |     |

2016. 10. 03.

# 목 차

## 1. 목 차

## 2. 개 요

- 2.1 작품명
- 2.2 작품개요
- 2.3 목적

## 3. 작품 설명

- 3.1 Software 구성
- 3.2 Software 흐름도 및 클래스 다이어그램
- 3.3 Software 기능 및 프로그램 사용법
- 3.4 개발환경

## 4. 프로그램 설명

- 4.1 파일 구성
- 4.2 함수별 기능
- 4.3 주요 함수의 흐름도
- 4.4 기술적 차별성

## 5. 응용 분야

## 6. 기타

- 6.1. 문제점 및 해결
- 6.2. 개발 방법

## 7. 제작자 정보

## 8. 개발 단계별 기간 및 투입 인원

## 2. 개요

### 2.0 작품명 : Memory2Pie

360도 카메라로 찍은 사진(기억;Memory)을 파이(Pie)도형에 분류해준다는 것을 의미하여 Memory to Pie로 해석할 수 있다. 그리고 360도를 의미하는  $2\pi(2\text{Pie})$ 라디안 또한 작품명에 내포되어 있다.

### 2.1 작품 개요

일반인들이 만약 360도 카메라를 사용한다면 어떤 목적으로 사용하게 될까? 우리는 상업적 목적의 영상 저장 매체가 아닌 추억을 담기 위한 용도로 360도 카메라를 사용할 것이다. 그로 인해, 우리는 360도 카메라를 통해 소중한 기억의 순간을 하나도 빠짐없이 남기고 싶은 욕구를 충족시킬 수 있을 것이다. 하지만 과유불급(過猶不及)이라는 사자성어와 같이 360도 카메라로 찍은 사진은 너무 많은 이미지 정보를 저장하고 있다. 그로 인해, 우리는 이 정보를 제대로 소화하지 못할 것이다. 예를 들면, 광고를 보면 사용자는 360도 카메라로 추억을 담기 위해 멋진 사진을 찍는 장면만이 나온다. 우리는 그 순간이 아닌 더 먼 미래를 생각해 보았다. 오랜 시간이 지난 후, 사용자는 그 사진을 다시 볼 것이다. 하지만 과연 사용자가 그 사진의 위치가 어느 장소이고, 그곳에서 자신이 느낀 감동, 생각을 기억할 수 있을까? 우리는 사람의 순간에 대한 기억이 오래 지속될 것이라 생각하지 않는다. 인간은 그러한 정보를 오래 기억하기 위해, 오랫동안 기록이란 행위를 해왔다. 그래서 우리는 이 소중한 추억의 순간을 기억이 퇴색될 정도의 오랜 시간이 지난 후에도 사진에 남아 있는 정보를 통해 그 시절의 감동을 되새김 할 수 있도록 돕고 싶었다. 그 결과 우리는 360도 카메라로 찍어서 나온 사진을 분류하여, 사용자 임의로 기록을 할 수 있도록 돕는 어플리케이션을 제작하기로 하였다.

### 2.2. 목적

360도 카메라는 영상 매체에서는 다양한 기능이 응용될 수 있는 매력적인 도구이다. 일반인들에게, 360도 카메라는 주로 개인의 추억을 소장하기 위한 카메라의 역할로 사용될 것이다. 추억을 소장하기 위해 360도 카메라로 찍은 사진에는 많은 정보가 있다. 그러나 인간은 망각의 동물이다. 따라서 대다수의 사람들은 오랜 시간이 흐른 뒤, 사진을 찍었을 당시의 상황에 대한 정확한 기억을 잊을 것이다. 현재 일반적으로 상용화 되어있는 일반 카메라는 우리가 찍었던 한 장의 사진 안에 비교적 정확한 추억을 내포하게 한다. 왜냐하면, 우리가 모든 공간을 한 장의 사진 안에 담지 못하므로, 중요한 정보를 갖고 있는 장면을 사진기로 찍기 때문이다. 우리가 360도 카메라의 기술을 일상에서 사용하게 하는 큰 이유는 추억의 장면 중 일부분이 아닌, 그 당시의 모든 상황을 저장하고 싶어서이다.

이 때, 우리는 의문점이 들었다. 만약 우리에게 360도 사진이 기억을 저장하는 장치로 사용된다면, 오랜 시간이 흐른 뒤 우리는 360도(즉, 모든 공간에서)의 많은 정보가 들어간 이미지를 보고 정확하게 내가 무엇을 위해 이 사진을 찍었는지, 이 사진을 찍었을 때의 내 감정과 생각을 알 수 있을까? 우리는 결코 모든 것을 기억 하지 못할 것이라 생각한다. 시간이 흐른 뒤, 다시 사진을 보았을 때, 사람은 필연적으로 평면의 화면으로 접해야만하기 때문에 360도 카메라를 이용해 전체 공간을 촬영 했더라도, 한 번에 볼 수 없다. 따라서 또 다시 360 사진을 감상할 때, 놓치는 부분이 발생 할 수 있다. 그때 만약 사진을 찍었을 때의 감정과 생각을 기억

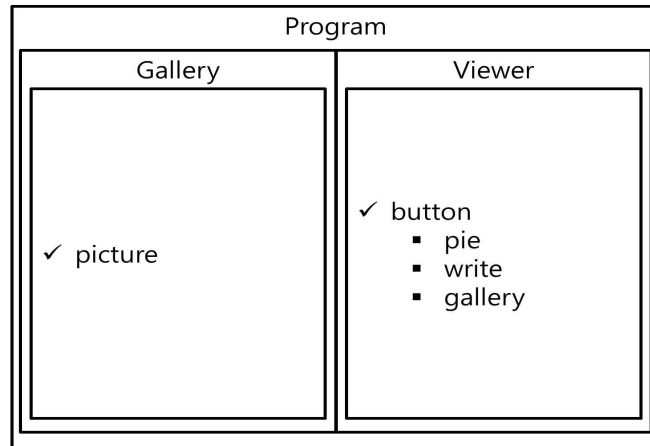
하지 못한다면, 그 사진은 추억을 기억하는 매체로 옳지 않다고 생각한다. 그래서 우리는 이 문제점을 해결하기 위해, 사진의 방향마다 추가적인 문자정보를 저장하는 서비스를 제공해 줄 것이다. 이를 통해, 우리는 순간의 모든 정보를 기억하고 싶은, 그리고 시간이 지나고 사진을 봤을 때, 그 때의 감성을 되새기고 싶은 욕구를 충족할 수 있을 것이다.

최종적으로 360도 카메라의 "공간을 담다"라는 장점을 최대한 살리고 "꼭 필요 한 장면을 놓칠 수 있다"는 단점을 보완 할 수 있는 어플리케이션을 만들 것이다. 이 어플리케이션의 중요한 기능은 촬영자가 특정 장면의 코멘트를 삽입했을 때, 위도, 경도, 코멘트를 포함한 데이터를 저장하여 나중에 360 뷰어를 사용할 때 파이차트 모양의 링크를 통하여 특정 장면으로 쉽게 이동하고, 감상이 가능하도록 만드는 것이다. 또한 특정 장면 중 대표 사진을 Thumbnail로 사용하여 미리보기 중에도 360도 사진이 어떤 사진인지 쉽게 알아 볼 수 있도록 하는 기능을 담당한다.

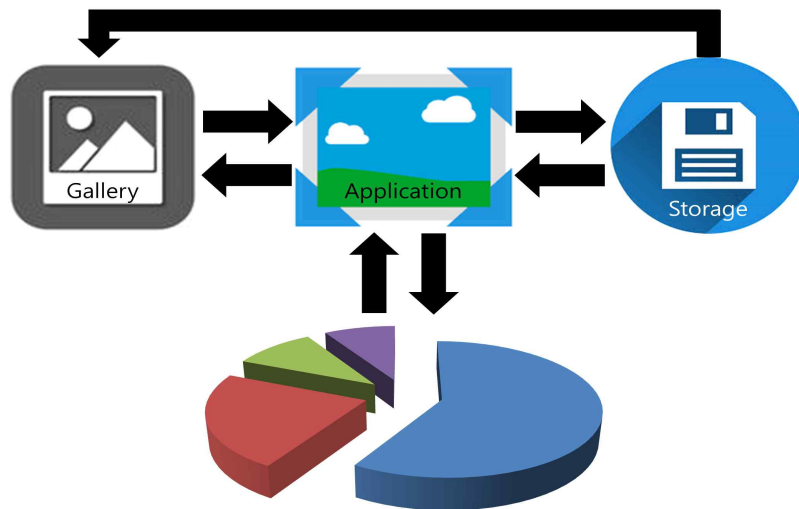
### 3. 작품설명

#### 3.1. Software 구성

software의 구성은 다음과 같다. 본 어플리케이션은 크게 Gallery와 Viewer로 되어있으며, Gallery는 사진들을 보여준다. 또한, Viewer는 사진을 구의 형태로 보여주며, Viewer에는 3개의 버튼 (pie, write, gallery)이 있다.



#### 3.2. Software 흐름도 및 클래스 다이어그램



어플리케이션의 흐름은 Gallery에서 사진을 load하여 어플리케이션에서 메모를 하면 파이 그래프를 생성한다. 저장된 파일을 클릭하면 메모를 한 위치와 메모를 보여준다. 매 프레임(Update 함수)마다 메모리의 근처 화면 각도를 순차검색을 통해 확인하여 저장했던 메모리를 열람할 수 있는 버튼을 보여준다. 화면 회전에 따른 정지와 이동은 Flag를 통해 제어하였다. 또한 메모를 한 사진은 핸드폰 내부 저장소에 저장된 뒤 Gallery로 들어가 다시 그 사진을 확인하고 싶다면, 해당 사진을 선택하여 어플리케이션으로 불러와 편집을 할 수 있다. 사진을 저장할 때는 Memory 클래스를 List에 등록하고 파일명을 hashing하여 저장한다. Loading시에는 안드로이드 플러그인에서 보내주는 파일명을 hashing하여 heap에 저장되어있는 데이터 중 같은 값이 있는지 검색하고 찾기에 성공하면 맞는 메모리클래스의 저장 값을 불러와 준다.

### 3.3. Software 기능 및 사용법

가장 처음 프로그램을 사용하기 위해서는 안드로이드 사진첩과 연동되어있는 사진을 가져오면서 시작한다. 그 곳에 있는 사진들은 전부다 가져 올 수 있으며 사용자는 사진첩에서 마음에 드는 사진 한 장을 선택 하면 사진을 어플리케이션으로 가져 올 수 있다. 그 이후에 기억하고 싶은 한 부분을 선택해서 그 곳에 메모를 작성한다.



첫 번째 그림은 테스트용 360도 사진을 담아서 촬영 한 사진의 일부분이다. 화면 오른쪽 상단을 보면 3개의 다른 색의 네모 칸이 보인다. 이 중 메모 작성을 위해서는 초록색 네모 칸(연필모양)을 선택해서 동작 시키며, 현재 생성된 파이의 목록(현재 원하는 부분을 저장한 자료)을 보고 싶으면 빨간색 버튼을 누르면 그 곳으로 자동으로 이동하며, 마지막 맨 위(파란색 네모 칸)쪽 버튼은 안드로이드(사용하고 있는 단말기)의 Gallery와 연동되어 사진을 바꿀 때 사용한다.

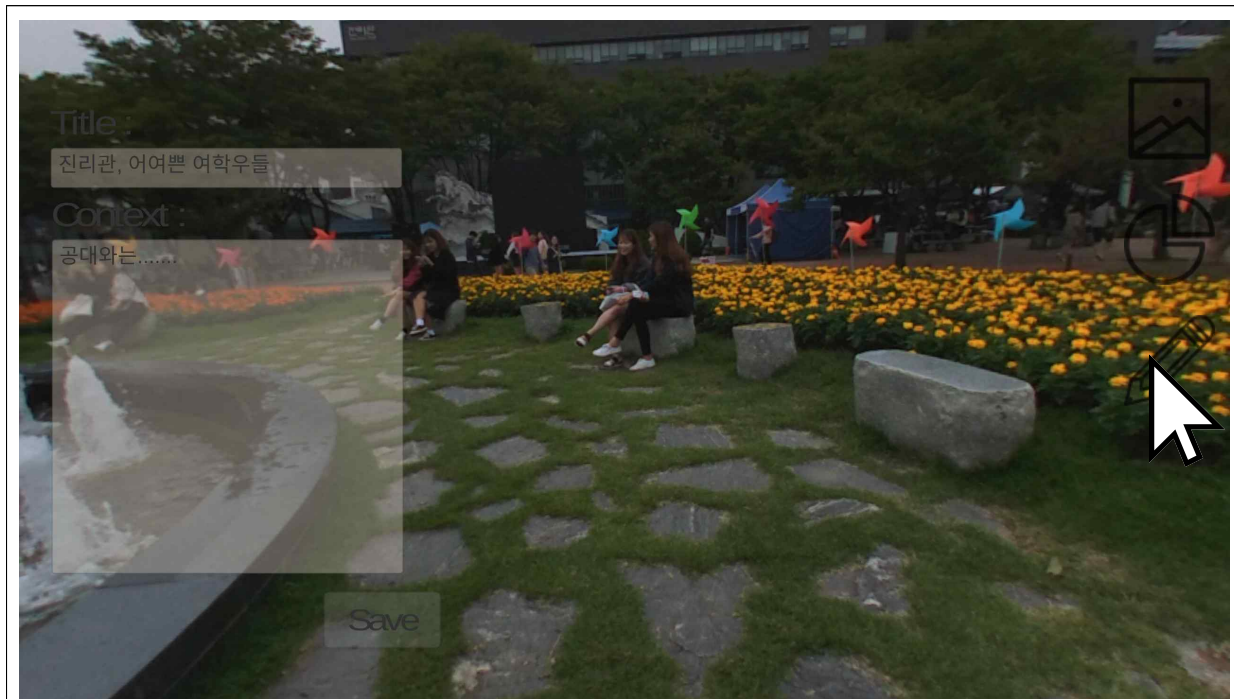


사진 3-2. 메모 작성 버튼 클릭 시 화면

다음 사진은 연필 버튼(녹색 네모 칸)을 누른 상황이다. 현재 왼쪽화면에 보이는 창은 텍스트를 집어넣을 수 있게 하는 부분으로, 이곳에 제목과 내용을 작성하고 저장을 원할 시에는 Save버튼을 누르게 되면 맨 아래 부분(파이가 생성되는 부분)에 제목이 나타나면서 동적으로 파이가 생성되는 모습을 볼 수 있다.

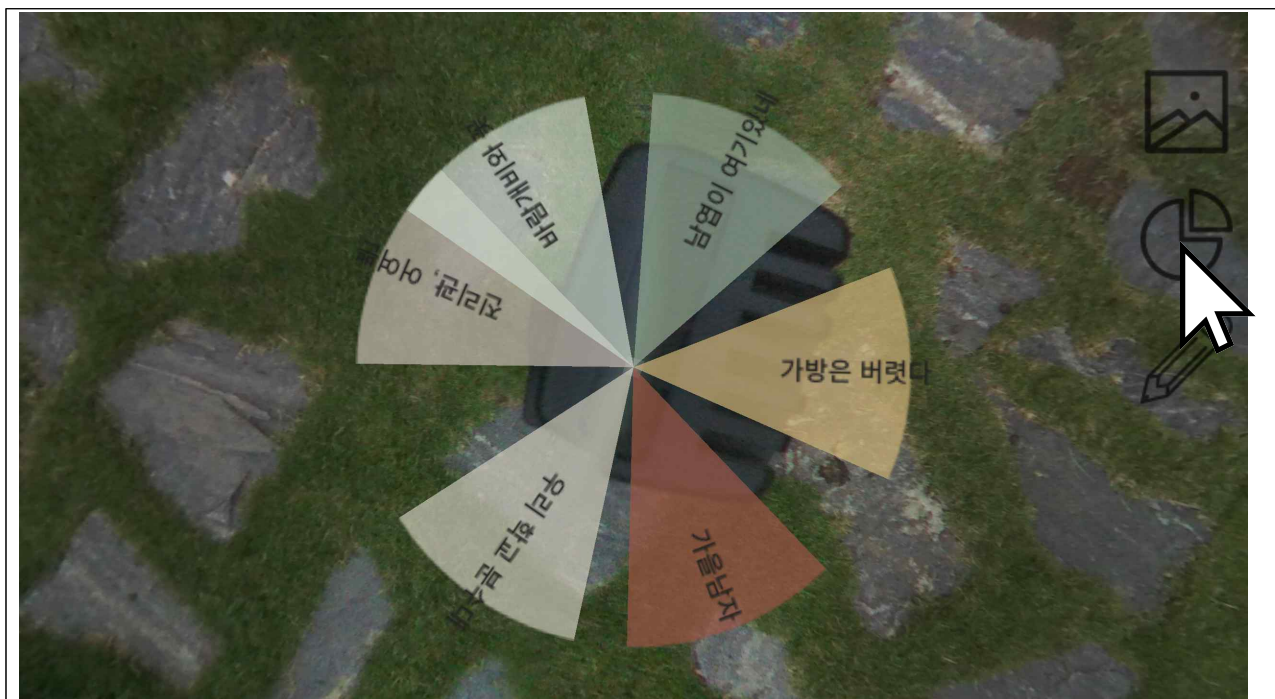


사진 3-3. Pie 버튼 클릭 시 화면



세 번째 그림은 현재 저장되어 있는 정보들을 담고 있는 파이의 모습이며, 빨간색 네모 칸의 파이 버튼을 눌렀을 때 이동한 사진을 보여준다. 현재 파이의 모습을 보면, 6가지의 화면을 각각 저장했기 때문에 총 6개의 조각이 생성된 것을 볼 수 있다. 각 부분을 파이 조각을 클릭하면 저장해 놓았던 장면으로 이동하게 되며, 그 이후에는 옆 부분에 저장했던 내용까지 화면에 띄워 주게 된다. 파이는 저장하려 했던 화면의 가운데 부분을 인식해서 바로 그 밑에 파이를 생성하게 해준다. 그 결과 겹치는 모습도 볼 수 있다.



사진 3-4. nearPie 근처에서 버튼 클릭시 화면

마지막으로 입력되었던 메모리들의 인근 좌표에서 나타나는 색상이 있는 파이를 클릭하였을 때의 화면이다. 이 때, 기존에 편집을 위한 패널이 그대로 출력되며 이곳에서 메모리를 편집, 삭제를 할 수 있다. 따라서 사용자는 메모리 저장, 편집, 삭제등을 간단한 형식에서 자유롭게 수정할 수 있다.



### 3.4. 개발환경

초기 개발을 할 때 모바일 어플리케이션 개발환경에 맞추어 안드로이드 스튜디오에서 개발을 진행하려고 하였다. 하지만 개발 초기에 개발능력의 한계로 인해 SDK 활용에 어려움을 겪었다. 따라서 좀 더 능동적인 개발 환경을 찾아 Unity3D 게임 엔진을 이용한 개발로 환경을 변경 하였다. 이로써 우리 팀은 Unity 개발 환경을 기반으로 스크립트 작성을 위해 C#언어를 이용하였고 학습하여 사용하였다.

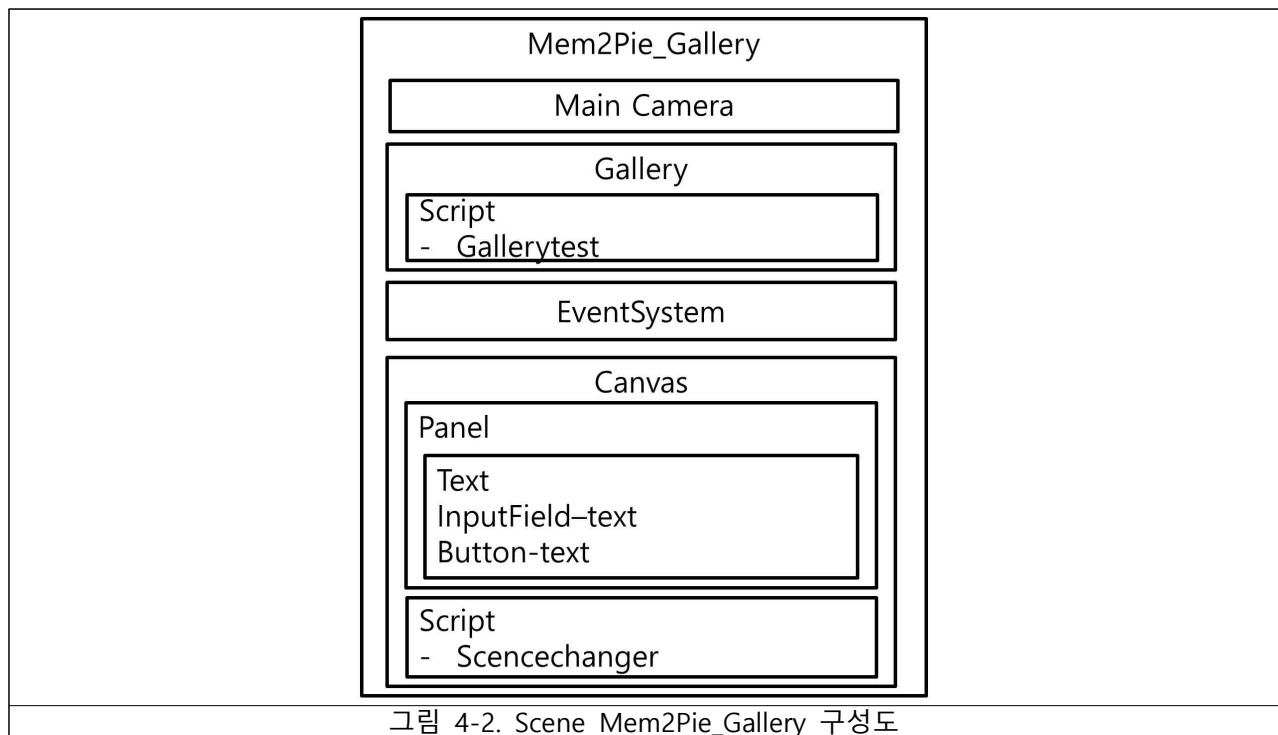
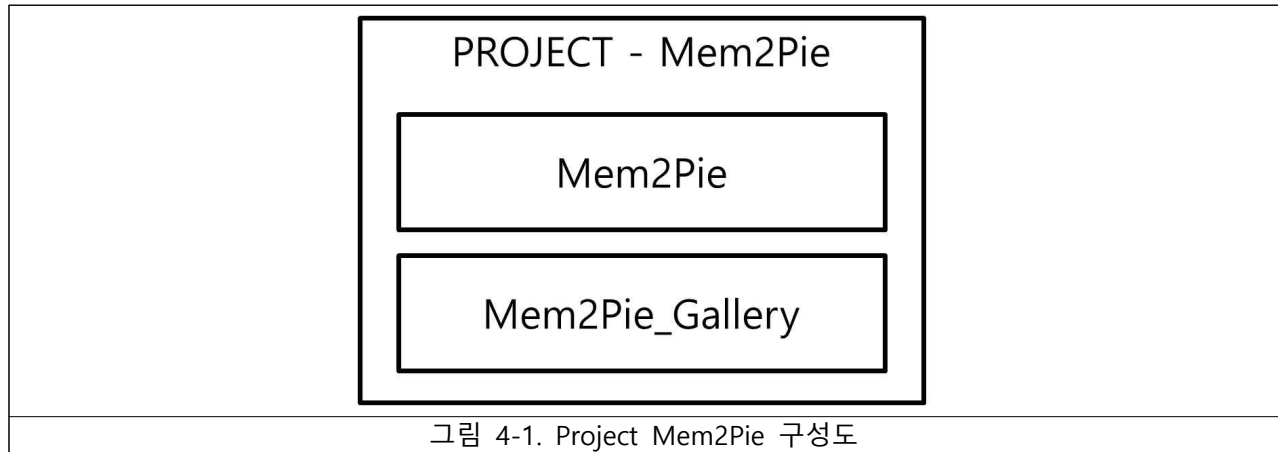
#### - Unity3D 설명

Unity는 3D 비디오 게임이나 건축 시각화, 실시간 3D 애니메이션 같은 기타 Interactive 콘텐츠를 제작하기 위한 통합 제작 도구이다. 개발자는 윈도우와 맥 OS X 상에서 실행되어 윈도우나 맥, Wii, 아이패드, iPhone 플랫폼 그리고 안드로이드로 게임을 만들 수 있다. GUI가 아주 직관적이다. 그리고 Shader, 물리 엔진, 네트워크, 지형 조작, 오디오 & 비디오, 애니메이션 등 게임을 만드는 데 필요한 웬만한 기능을 자체적으로 지원한다. 따라서 초보 개발자도 쉽게 접근할 수 있다. 예를 들면, 버튼 하나만 누르면 수많은 플랫폼으로 빌드가 가능하다. 즉, 게임 하나 만들어 놓으면 수많은 플랫폼에서 사용할 수 있다는 것이다. 물론, 어느 정도 조정은 필요하다.

## 4. 프로그램 설명

### 4.1. 파일 구성

Project파일인 Mem2pie안에는 다음과 같은 개의 scene가 있다. 각각 scene를 구성하고 있는 object는 다음과 같다.



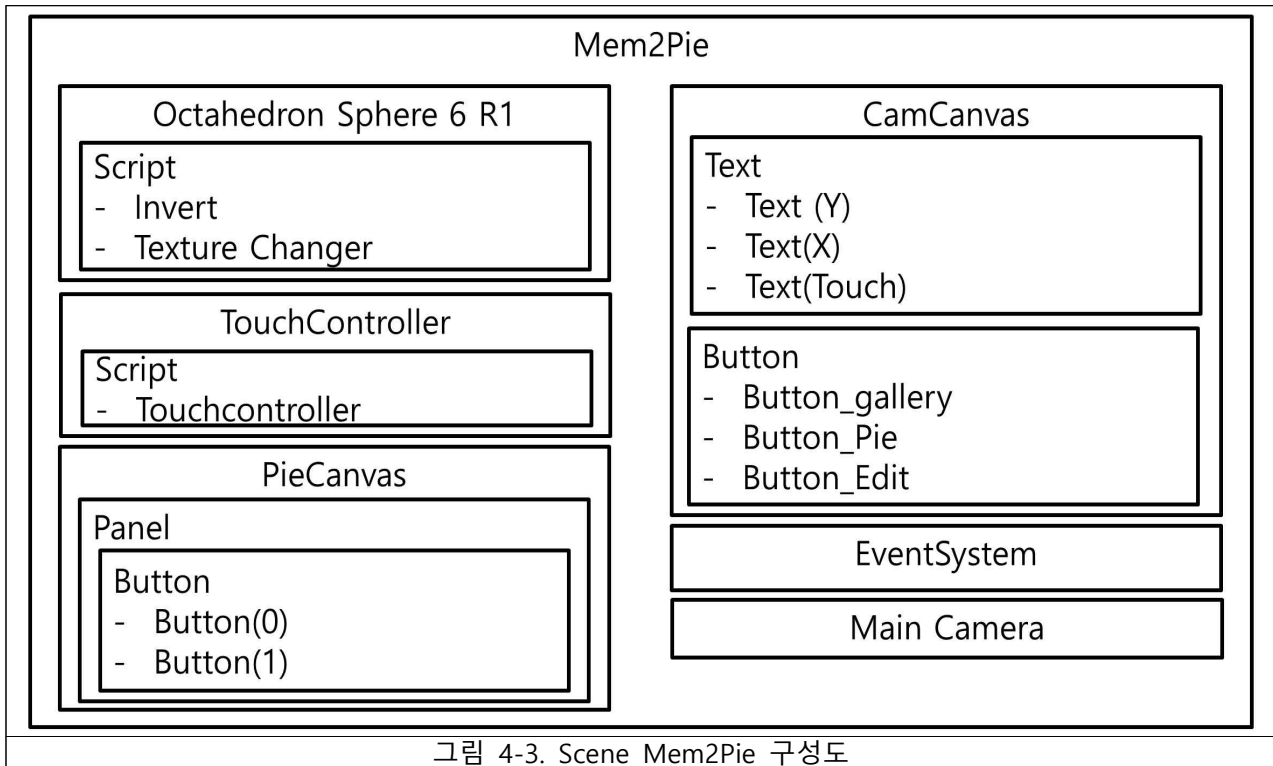


그림 4-3. Scene Mem2Pie 구성도

## 4.2. 함수별 기능

| 클래스 별 기능           |   |
|--------------------|---|
| Gallerytest        | 스마트폰 Gallery에서 사진을 가져 온다.                                 |
| GeneralController  | 스마트폰 화면에 보이는 사진의 크기, 방향을 조정한다.                            |
| PieGraph           | 각도를 계산하여 Pie 모형을 그린다.                                     |
| InputMemController | 입력한 메모의 위치, 색깔, 내용을 저장한다.                                 |
| SaveAndLoad        | 스마트폰 내부 저장소에 편집 내용을 저장하고, 로드한다.                           |
| Scenechanger       | 연결된 버튼 클릭 시 Build에 0번으로 설정된 Scene으로 화면을 전환한다.             |
| TextureChanger     | Gallery에서 선택한 사진을 Viewer로 보여줄 수 있게, 구 형태의 mesh에 사진을 입혀준다. |

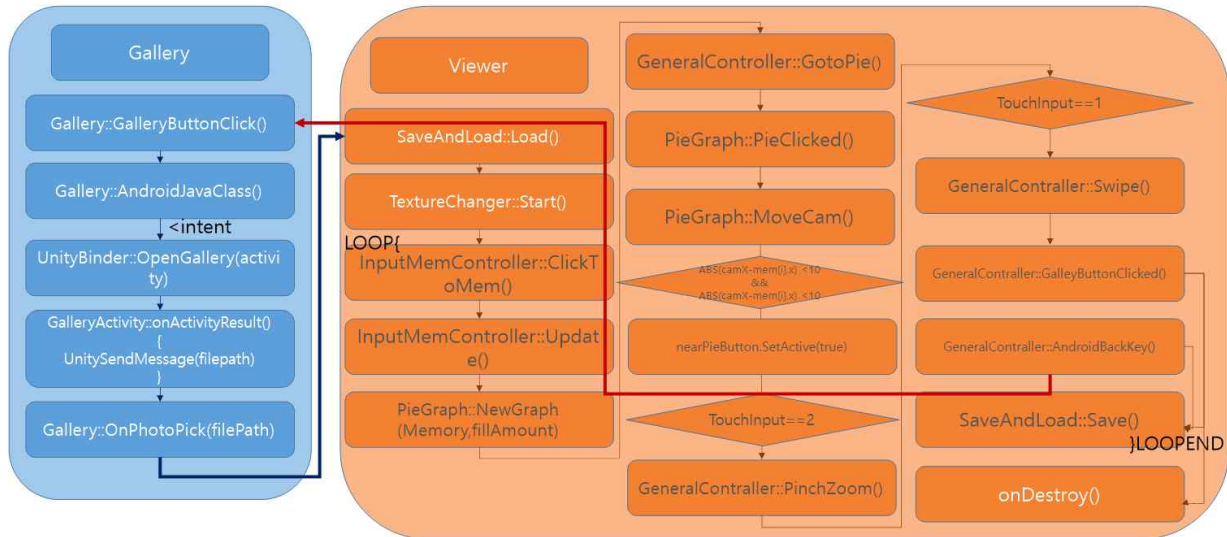
| GeneralController |  |
|-------------------|--|
| FixedUpdate       | 하나의 터치 입력 시 화면 이동, 두 개의 터치 입력시 화면을 확대한다. |
| ClickGoPie        | pie모형의 위치로 카메라를 회전시킨다.                   |

| PieGraph |                       |
|----------|-----------------------|
| NewGraph | pie모형을 그린 후 메모를 저장한다. |
| MoveCam  | 카메라를 회전시킨다.           |

| InputMemController |  |
|--------------------|--|
| ClicktoEnabled     | 메모창의 색을 저장한다.                                  |
| ClicktoSave        | 메모를 저장한다.                                      |
| Scenechanger       |  |
| Clickto0           | 저장 후, build에 0번으로 설정된 화면으로 화면을 전환한다.           |
| Gallerytest        |  |
| OnPhotoPick        | Gallery에서 사진을 가져온다                             |
| SaveAndLoad        |  |
| Save               | 편집된 사진의 정보를 PlayerPrefs에 저장한다.                 |
| Load               | 저장된 사진의 정보를 다시 리스트로 캐스팅해준다.                    |
| TextureChange      |  |
| Start              | 구 형태의 mesh에 선택된 사진을 입혀줌으로서 화면에서 사진을 볼 수 있게 한다. |

### 4.3. 주요 함수의 흐름도

Gallery와 Viewer 두 개의 part안에는 각각을 구성하고 있는 함수들이 있다. 주요 함수의 흐름도는 위와 같다.



#### - Part. Gallery

프로그램의 시작 혹은 Gallery버튼을 클릭했을 때, GalleryButtonClick함수가 실행된다. 그에 이어서 Gallery 사진을 보여주는 함수들이 실행된다. 사용자가 사진을 선택했을 시, Viewer Part로 넘어간다.

#### - Part. Viewer

Gallery 에서 사진을 선택하면 Load함수가 실행된다. 그리고 사진을 360도로 보여주도록 돕는 TextureChanger Class의 Start함수가 호출된다. 그 후 메모 입력과 저장이 반복되며, 사용자가 Gallery 버튼을 누를시 Gallery Part로 넘어간다.

#### 4.4. 기술적 차별성

##### 일정 위치에 메모 저장

기본 사진 편집 어플리케이션은 사진의 모양이나, 스티커 붙이기의 기능이 대다수이다. 메시지를 남기는 어플리케이션 또한 있지만, 360도 뷰어에서는 아직까지는 이러한 구체적인 기능을 수행 할 수 있는 어플리케이션이 존재 하지 않는다. 하지만 우리가 구현한 어플리케이션은 사용자가 원하는 사진의 위치에 사용자가 원하는 메시지를 입력함으로써 보다 사용자의 추억을 기록하기 유용하게 하였다.

##### 다양한 플랫폼에 적용가능

멀티 플랫폼을 지원하는 Unity를 사용한 어플리케이션으로 안드로이드뿐만 아니라 IOS, windows등 다양한 플랫폼에 최소한의 최적화 과정을 통해 쉽게 적용이 가능하다.

##### 파이 모양에 제한하지 않으며 응용가능성 많다

파이 그래프는 하나의 시각화 형식에 불과하다. 사용자가 입력한 제목과 내용의 텍스트 데이터와 경도 위도 데이터는 하나의 구조체로 관리되어 다양한 모양으로 mapping될 수 있다. 파이 그래프뿐만 아니라, 포스트 잇, 꽃잎 모양, 브랜드 로고 등, 시각화 할 수 있는 방법이 존재한다.



## 5. 응용 분야

이 어플리케이션은 기본적으로 360도 사진에서 정보를 입력하는 서비스를 제공한다. 이를 통해 동영상에도 표시를 하여 중요한 부분을 빠르게 찾아 볼 수 있는 기능으로 응용 가능하다. 또한 360도 콘텐츠에서 정보 분류를 응용한 기능들을 생각해 볼 수 있다. 예를 들면, 360도의 가장 큰 장점은 순간의 모든 장면을 한 번에 찍을 수 있다는 장점이 있다. 일반 카메라와 같이 사용자가 원하는 장면 일부만이 아닌, 전체의 모습을 다 저장할 수 있다. 하지만 이 정보를 나중에 보게 됐을 때 우리는 어떤 장면들이 중요 했는지 잊을 수 있다. 따라서 360도 카메라는 정지된 상태로 사진을 찍는 특성을 이용하여 스마트 폰의 자이로센서를 통해 촬영자가 그 당시 보고 있던 시각 방향을 추적하는 기능에도 응용 될 수 있다. 360도 카메라의 확장성도 살리면서 촬영 작가의 의도 또한 파악 할 수 있다. 또한, 사진의 메모를 저장하는 Pie는 다양한 형태로 존재 가능하며, 이는 더욱 다양한 분야에서의 사용을 가능하게 해준다.

응용의 예를 들어보면, 한국인들은 꾸준히 셀프 인테리어에 관심을 가지며, 그로 인해 가구시장 또한 소비자 입장에 맞춘 발전과 도전은 해오고 있다. 그 중 IKEA는 최근 우리나라에 유입된 세계에서 유명한 가구회사 중 하나이다. 소비자들이 IKEA에서 가구를 구입하는 방식을 살펴보면 우선 3층의 쇼룸을 살펴보면서 마음에 드는 가구마다 붙어 있는 태그를 확인하여 제품의 창고 진열대 번호를 메모한다. 그 뒤, 2층에서 메모된 제품위치 번호로 원하는 가구의 위치를 찾는다. 마지막으로 그 가구들을 카트에 담아 결제한다.



사진 5-1. 제품번호를 작성하기 위한 종지와 연필

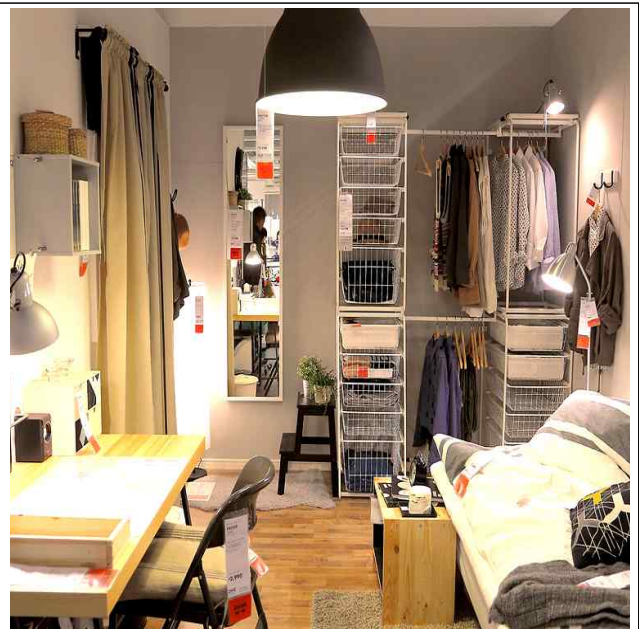


사진 5-2. 사진 수십 개의 태그가 포함된 쇼룸

그러나 축구장 8배에 달하는 25만 6000 $m^2$  공간에서 65개의 쇼룸을 둘러보며 종이에 메모를 하는 방식은 그 가구를 사고 싶었을 때의 인테리어와 여러 제품을 구매를 원할 때 수많은 제품 번호는 고객이 기억하기에는 너무 많은 정보이고 이는 2층 창고에서 구매를 망설이게 하는 요인이 될 수 있다. 또한 한국 매장은 항상 북적이는 인파로 태그에서 번호를 써가며 구경하기가 쉽지 않다. 현재 IKEA는 어플리케이션을 제공하여 제품번호를 휴대폰에 저장할 수 있는 서비스를 제공하고 있으나 이 또한 쇼룸과 연계되어 있는 것이 아닌 제품을 직접 검색하고 제품번호를 리스트에 저장하는 방법으로 직접 태그를 확인하는 것과 다른 점이 없다.

이 구매방식에서 우리의 어플리케이션을 응용할 수 있다. IKEA가 쇼룸을 구성할 때



360 카메라를 통해 이미지를 저장하고 각 쇼룸에서 태그를 붙일 각 가구에 이 어플리케이션을 통해 제품정보를 입력해 준다면 태그와 같은 역할을 할 수 있다. 고객은 쇼룸을 보며 구매를 원하는 제품을 보고 이 어플리케이션에서 같은 제품을 확인 하고 그대로 태그를 클릭하여 어플리케이션에서 리스트를 만들 수 있다. 결과적으로 고객은 번거롭게 연필과 종이를 들고 다닐 필요성이 없고, 자신이 구매를 원하던 제품의 전체적인 인테리어를 잊어버려서 셀프서브 공간에서 구매를 망설이는 일을 막을 수 있다.

따라서 꼭 파이를 만들 필요 없이 Memory클래스의 정보를 파이 대신 카메라 앵글을 통한 좌표를 받아 그 지점에 태그라는 Object를 선언하여 만들어 주면 쉽게 응용 할 수 있다. 파이모양에 국한 되어 있지 않고 다른 형태의 인터페이스를 쉽게 변형해서 적용할 수 있도록 개발하였다.

## 6. 기타

### 6.1. 문제점 및 해결

개발 과정 중 여러 문제점을 도출하고 해결하거나 다른 방향을 제시하며 진행되었다. 파이모양을 바닥 부분에 그려줄 때 입력한 메모리(Memory 클래스로써 제목, 내용이 포함됨)와 앵글 값을 받아 동적으로 파이를 카메라의 방위각(azimuth)을 기준으로 그려주어야 하는 문제였다. 이때 정확한 렌더링을 위해 파이의 값을 고정한 상태로 그려주고 채워줄 값(변수 이름 : fileAmount)의 1/2값만큼 반대 방향으로 돌려서 파이를 그려주어 정확한 메모리 방위각을 중심으로 그려주는 파이 이미지를 생성할 수 있었다.

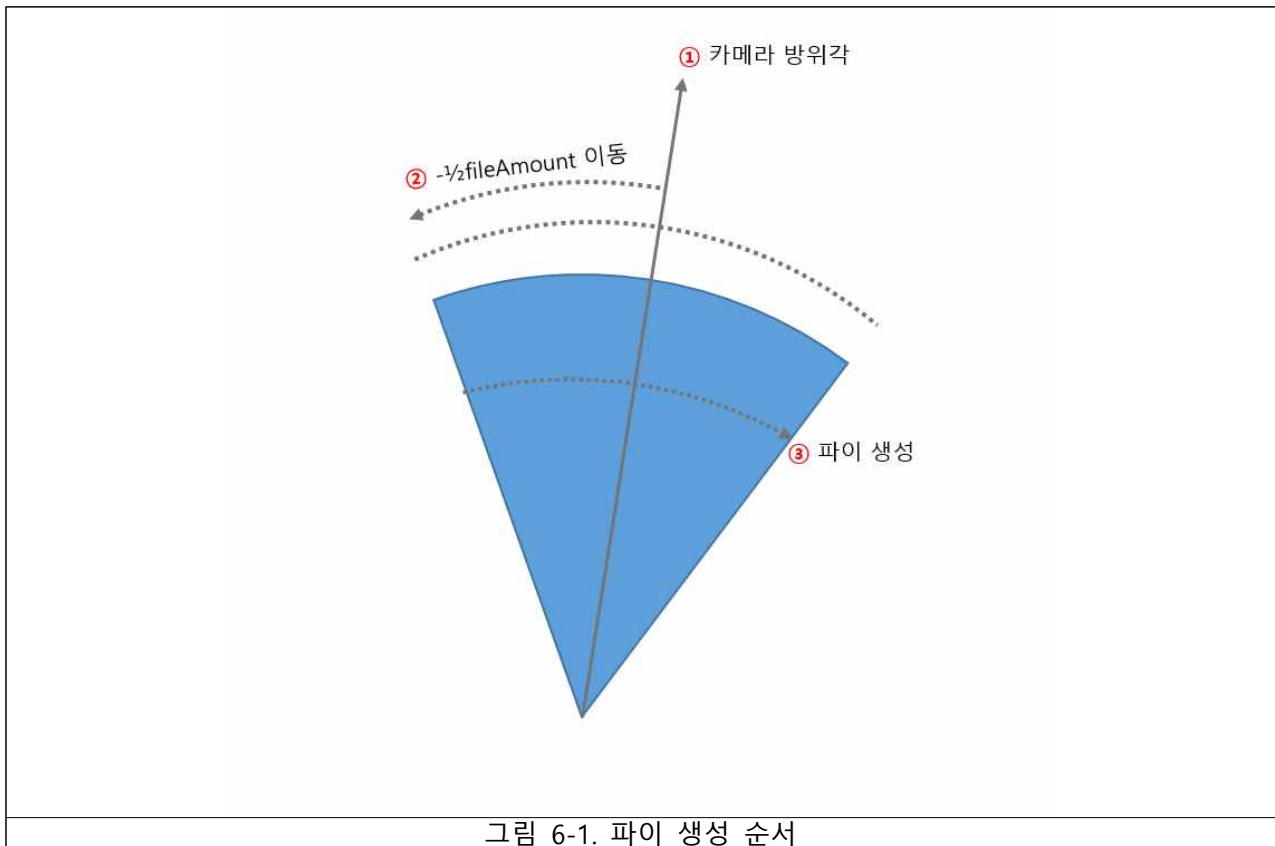


그림 6-1. 파이 생성 순서

카메라 각도와 터치 Swiping (터치로 화면을 밀어주는 동작)의 index 차이 또한 해석하는데 어려움이 있었다. 표면적인 클래스 변수 이름  $x$ ,  $y$ ,  $z$ 에 의존하여 프로그램을 설계했을 때 터치의 부자연스러움을 확인할 수 있었다. 이를 해결 위해 우리는 카메라 Canvas(카메라 화면에 그려주기 위한 UI)에 2개의 Text UI를 정의하여 카메라 앵글  $x$ ,  $y$ (Altitude, Azimuth)를 상속하여 유닛 testing의 방법인 Driver를 사용하였다. 실행을 통해  $x$ ,  $y$  값이 터치 이후 수시로 값 전환이 되는 것을 확인하여 이 문제가 카메라  $x$ ,  $y$ 와 터치의  $x$ ,  $y$  값의 변환 차이임을 알게 되었다. 이 때, 알아낸 사실은 touch의  $y$ 값(Vertical)은 카메라앵글의  $x$ 값(Altitude; 고도)과 연계가 되어야 한다는 것이다. 이에 문제를 해결 할 수 있었다.

## 6.2. 개발 방법

### 6.2.1 버전 관리 툴 - GitLab

공동 개발에 편의를 위해 GitLab을 사용하였다. GitHub를 사용하려고 하였으나, Private 한 저장소가 필요하므로(팀원만 Branch를 하고 코드를 읽어야 하므로) 이를 무료로 제공하는 GitLab을 라즈베리파이에 설치해서 우리 팀만의 개발 공간을 만들어 개발을 진행하였다. 그리고 프로젝트의 복잡성이 크지 않기 때문에 하나의 branch만을 이용하여 모든 팀원들이 같은 branch에 push를 해주어 일관되는 개발 진행을 하였다.

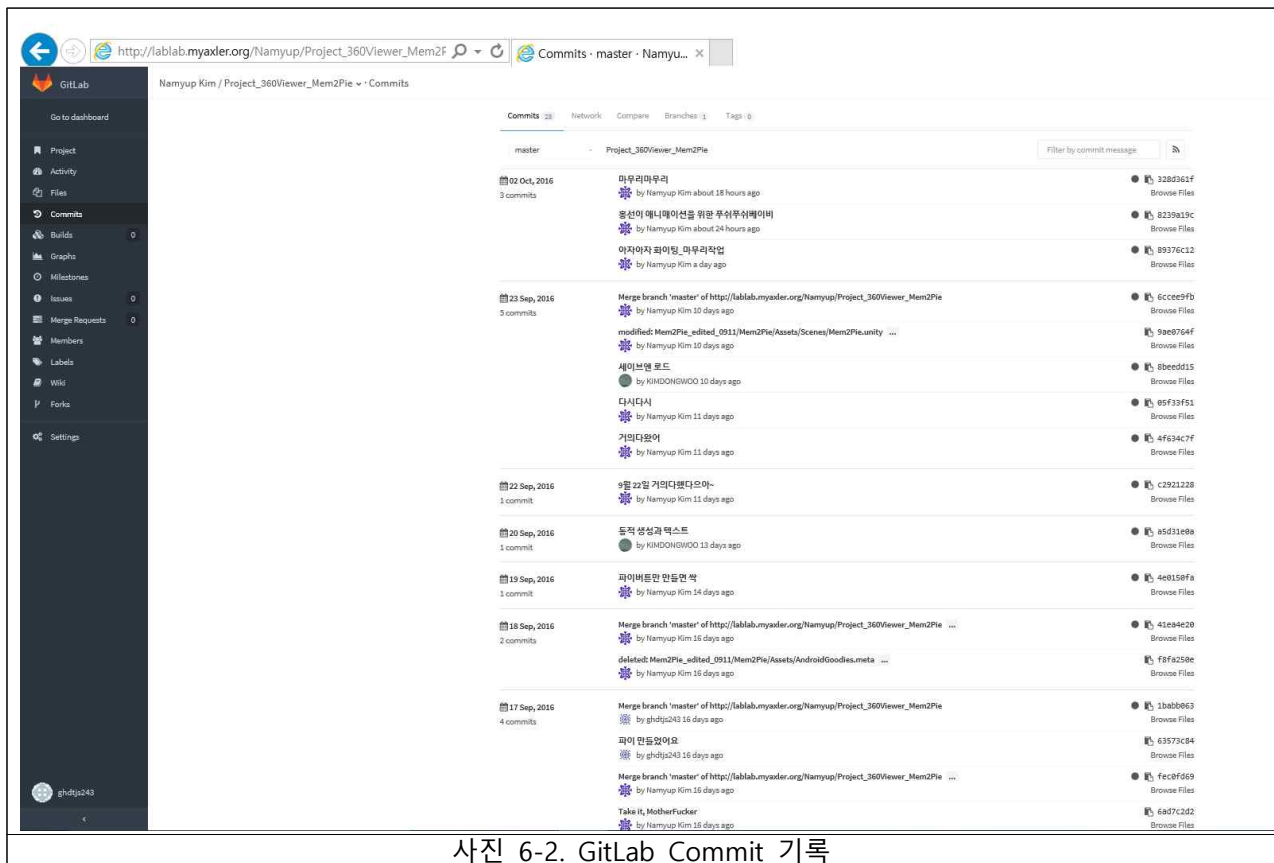


사진 6-2. GitLab Commit 기록

### 6.2.2 스크럼(Scrum) - Google Sheet

개발에 앞서 체계적인 진행을 위해 상호, 점진적 개발방법론인 스크럼을 적용하였다. Google Sheet를 통해, 개발해 나가기 위한 틀을 잡고 그 틀에서 변형이 이루어 질 수 있으나 이에 맞춰 각자 할 일을 나누어 진행하였다. Google Sheet는 Excel의 형태로 Cloud 서비스를 제공하여 스크럼 기법을 이용하기 편리하게 하였다. 따라서 이 Sheet안에 어플리케이션의 구조, 그에 맞는 임무, 각 팀원 별 진행상황 및 결과, 그리고 자유 회의록을 포함하여 매번 미팅을 하지 못하는 상황에서도 팀원 간의 소통이 자유로이 이뤄졌다. 또한, 날짜별 팀원 간의 진행상황을 공유하고 메모를 삽입시켜주는 기능을 이용하여 피드백을 주고받았다.

[illegible]

사진 6-3. GitLab Commit 기록

## 7. 제작자 정보

| No. | 구분 | 성명  | 소속(학교) | 부서(학과)    | 입학년도  | 담당업무                           |
|-----|----|-----|--------|-----------|-------|--------------------------------|
| 1   | 팀장 | 김남엽 | 송실대학교  | 정보통신전자공학부 | 2011년 | 파이 Panel 구현<br>메모리 클래스 구현      |
| 2   | 팀원 | 김동우 | 송실대학교  | 정보통신전자공학부 | 2011년 | Gallery 구축 및 Save<br>& Load 구현 |
| 3   | 팀원 | 윤홍선 | 송실대학교  | 정보통신전자공학부 | 2012년 | 이미지 Mesh 구현<br>파이 Panel 구현     |
| 4   | 팀원 | 연지영 | 송실대학교  | 정보통신전자공학부 | 2014년 | 카메라 Controller<br>구현           |

## 8. 개발 단계별 기간 및 투입 인원

[illegible]