

WESC-14-09-개발완료보고서

작품 제목

Design your home.

참가부문 : 스마트 챌린지 웰컴코리아

팀명 : CodeWeaver

팀구성원

No.	구분	성명	소속명	No.	구분	성명	소속명
1	팀장	문성필	한양대학교	6	팀원		
2	팀원	정락준	한양대학교	7	팀원		
3	팀원	박진석	포항공과대학교	8	팀원		
4	팀원			9	팀원		
5	팀원			10	팀원		

2014. 9.

제12회 임베디드SW경진대회 개발완료보고서

결과보고서 작성 목록

1. 목차

-개요

*작품명

*작품 개요

*목적

-작품 설명

*Software 구성

*Software 흐름도 및 클래스 다이어그램

*Software 기능

*프로그램 사용법

*개발 환경

-프로그램 설명

*파일 구성

*함수별 기능

*주요 함수의 흐름도

*기술적 차별성

-응용 분야

-제작자 정보

-개발 단계별 기간 및 투입 인원

2. 개요

2.1. 작품명

Design your home.

2.2. 작품 개요

본 프로젝트는 증강현실 스마트 디바이스 어플리케이션으로 2차원으로 존재하는 건축 도면을 3차원으로 변환하는 동작을 수행한다. 3차원 건축 모델을 통해 사용자들이 주택을 구입하기 이전 가구의 배치, 채광, 동선의 시뮬레이션을 지원한다. 또한 동선 시뮬레이션에서 1인칭 가상현실을 제공하여 사용자의 도면에 대한 이해를 돕는다.

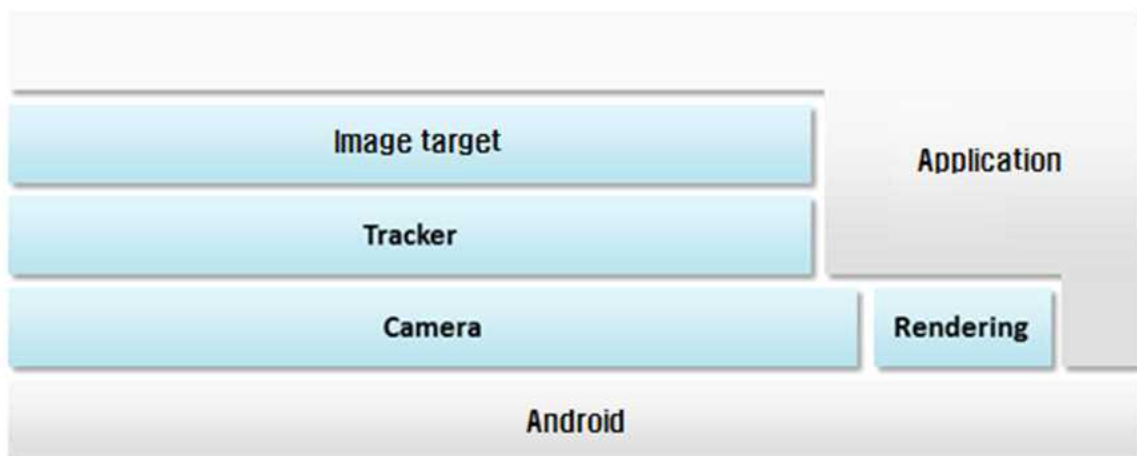
2.3. 목적

설계도면에 대한 지식이 부족한 일반인의 경우 도면만으로 그 구조물에 대한 정확한 이해를 하기는 어렵다. 따라서 팀 Code weaver는 본 프로젝트를 통하여 일반인들에게 구조물에 대한 이해를 돕는다. 또한 건축가가 소비자의 이해를 돕기 위해 물리적으로 존재하는 모델을 만들어야 하는 과정을 없앤다.

일반인 사용자는 스마트폰의 카메라를 통하여 도면을 증강하며, 각 마커를 이용하여 가구 배치를 시뮬레이션한다. 사용자는 큰 노동력을 들이지 않고 가구 배치에 따른 동선 시뮬레이션과 채광 시뮬레이션 등을 수행할 수 있다. 건축설계가는 일반인을 위해서 건축 모델을 제작해야하는 수고를 경감하며, Unity 3D, 3Ds Max, Maya 등과 같은 툴로 제작한 모델을 일반인 사용자에게 공급할 수 있게 된다.

3. 작품 설명 (최대한 자세하게 기술)

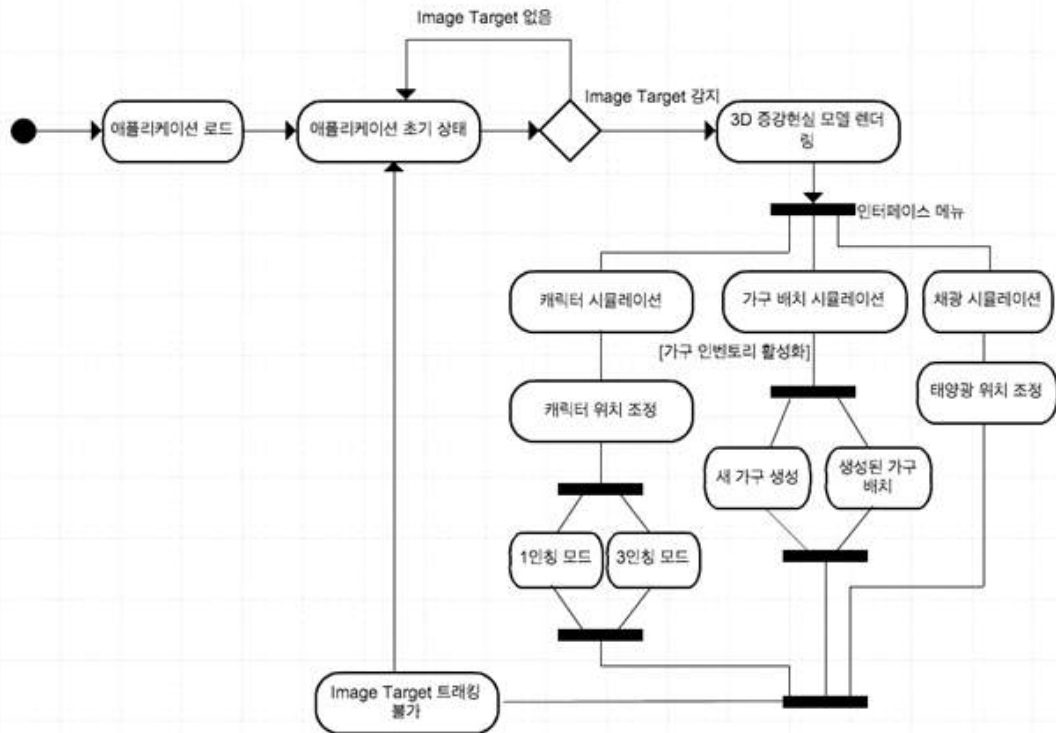
3.1. Software 구성



Vuforia SDK & Applicate 구성도

Vuforia SDK를 Unity에 Import하여 안드로이드 어플리케이션을 개발하였다. Tracker와 Image target을 SDK를 이용해 구현하였으며, Application은 Unity를 이용하여 구현하였다.

3.2. Software 흐름도 및 클래스 다이어그램 (개발언어에 따라 선택)



Application flow chart

어플리케이션 실행 시 내장된 마커 및 구조물의 3d 모델을 로드하고 각 클래스 연산에 필요한 초기화를 수행한다. 어플리케이션 로드를 완료하여 어플리케이션 초기 상태로 진입하면 카메라를 구동시켜 이미지 스트림을 불러온다.

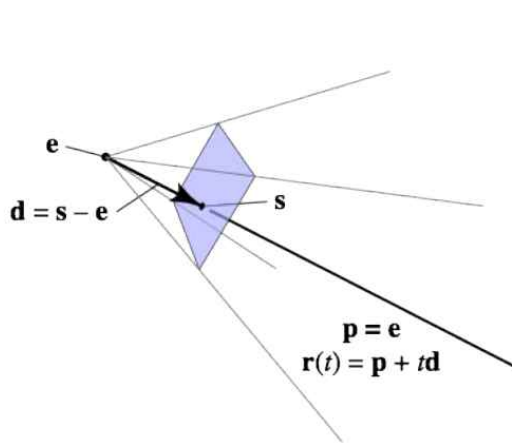
불러온 이미지 스트림에서 마커가 되는 Image target을 인식하고 마커가 없을 경우 초기 상태에서 대기한다. 마커가 인식되었을 경우 마커를 추적하고 3d 모델 렌더링을 수행한다. 이후 사용자의 모드 선택에 따라 각 기능을 구동한다. 모드는 캐릭터 시뮬레이션, 가구 배치 시뮬레이션, 채광 시뮬레이션으로 총 3가지로 구성되어있다.

마커 추적에 실패하였을 경우 수행하던 모드 상태를 저장하고 어플리케이션 초기 대기 상태로 복귀한다. 추적에 실패하였던 마커가 다시 인식될 경우 저장하였던 모드의 상태를 구동한다.

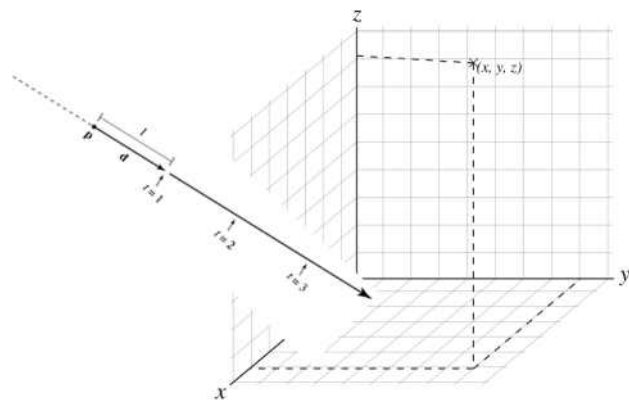
3.3. Software 기능 (알고리즘 설명 포함)

어플리케이션에서 모든 오브젝트를 배치하고 기능을 동작시키기 위해서 스크린에서 터치를 필요로 한다. 하지만 스크린 좌표계와 오브젝트가 존재하는 3D 세계의 좌표계가 일치하지 않는다. 그 때문에 스크린 위의 터치 포인트를 3D 오브젝트 좌표계로 변환시키는 작업이 필요하다. 그 후 아래 그림에서 카메라가 존재하는 e지점에서 스크린 위의 터치 좌표인 s로 벡터 뺄셈 연산하는 것으로 ray의 방향을 연산할 수 있다. 방향 벡터를 구한 뒤, 시작 지점을 더하여 3차원 벡터 선형 방정식을 구한다. $(r(t) = p + td)$

선형 방정식으로부터 z좌표가 0이 되는 t를 구하여 3D 평면 위 좌표를 구한다. 해당 좌표에서 가구 배치, 태양 배치, 캐릭터 이동 등의 기능을 수행할 수 있다. 또한 선형 방정식으로부터 어떠한 오브젝트가 선택되었는지 확인하여 오브젝트의 이동, 회전 기능을 수행한다.

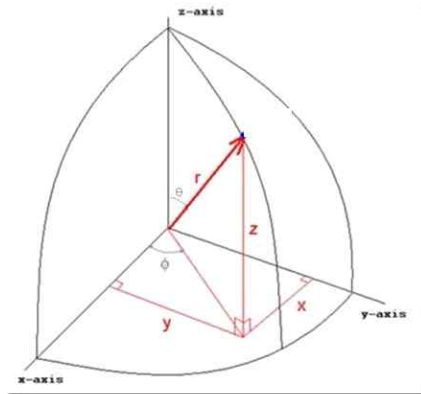


Generating eye rays - Perspective



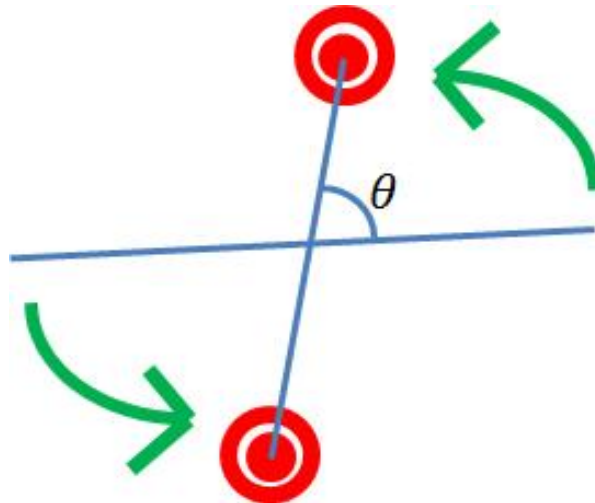
평면 위 좌표 계산

채광 시뮬레이션 수행 시 카메라의 선형방정식으로부터 터치 지점을 계산하여 태양을 배치한다. 태양을 배치한 뒤, 구조물의 중심점과의 유클리디안 거리를 구하여 태양 회전의 반지름으로 설정한다. 매 프레임마다 미리 지정된 각속도에 따른 변위를 계산하여 적용한 뒤 태양을 회전시킨다. 아래 그림에서 α 는 각속도를 의미하며 \varnothing 은 처음 태양을 생성한 x, y축과의 각도를 의미한다. 태양 오브젝트에 종속되어있는 Directional Light의 방향은 (구조물의 중심 좌표 - 현재 프레임에서의 태양 좌표)를 방향 벡터로 사용한다.



$$\begin{aligned} z &= \text{Radius} \times \sin(\theta + \alpha) \\ x &= \text{Radius} \times \cos(\theta + \alpha) \times \cos(\varnothing) \\ y &= \text{Radius} \times \cos(\theta + \alpha) \times \sin(\varnothing) \end{aligned}$$

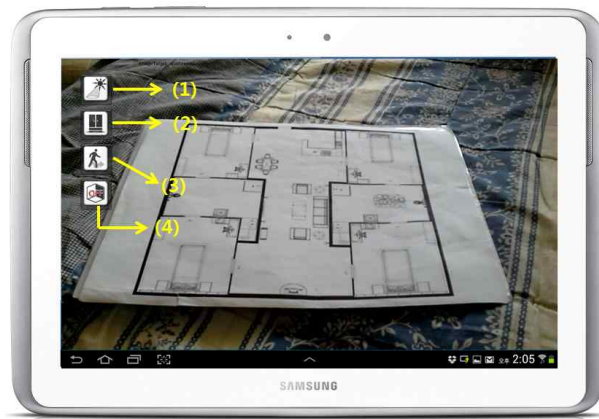
태양 좌표 계산



가구 회전 각도 계산

가구 배치 시뮬레이션 시 가구 회전을 위해서 두 손가락의 터치 회전 각도를 연산해야 한다. 가구 회전을 위한 각도는 두 손가락의 드래그 이전과 드래그 이후의 θ 를 위 그림과 같이 연산하여 사용한다. 가구 이동에서의 터치 패드는 위에서 구한 선형 방정식의 드래그 이전 포인트와 이후 포인트의 변위를 연산하여 방향 벡터로 사용한다.

3.4. 프로그램 사용법 (Interface)



Application 기본 UI

버튼 (1)은 채광 시뮬레이션 모드로 건축물 내부에 태양광이 얼마나 들어오는 지에 대한 시뮬레이션을 수행할 수 있다. 버튼 (2)은 가구 배치 시뮬레이션 모드를 제공하며 버튼을 클릭할 경우 하단에 가구 인벤토리가 나온다. 버튼 (3)은 캐릭터 시뮬레이션 모드이며, 클릭할 경우 구조물 중앙에 캐릭터를 생성한다. 캐릭터는 터치 패드를 이용하여 조종할 수 있으며, 캐릭터가 생성된 시점에서 한번 더 클릭할 경우 1인칭 시점으로 전환한다. 버튼 (4)은 어플리케이션의 모든 연산을 정지하는 기능을 수행한다.



채광 버튼 클릭

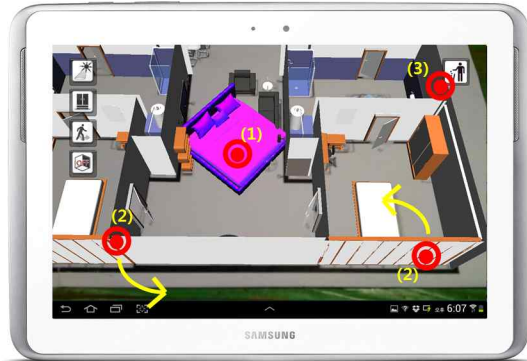


채광 시뮬레이션 동작

채광 시뮬레이션 버튼(1)을 터치하고 태양을 생성하고 싶은 위치를 터치(2)하여 채광 시뮬레이션을 동작시킨다. 스크린과 대응되는 오브젝트 평면 위에 태양이 생성되며 건축물을 중심으로 회전하여 빛의 변화를 체험할 수 있다.



가구 배치1. 모드 선택 & 가구 선택



가구 배치2. 이동, 회전, 삭제

가구 배치1. 가구 배치 시뮬레이션 버튼을 터치(1)하면 하단에 가구 인벤토리가 생성된다. 드래그(2)를 통해서 넘겨볼 수 있다. 현재 9개의 가구들을 제공하고 있으며 추후 다양한 가구들을 제공할 예정이다. 가구를 배치하기 위하여 선택한 가구를 드래그(3)하여 건축물의 원하는 위치에 놓을 수 있다. 배치된 가구는 건축물의 스케일에 맞추어 자동으로 크기가 변경된다. 가구들은 충돌 처리가 되어 있어서 가구들끼리 겹치게 배치할 수 없도록 구현되어 있다.

가구 배치2. 가구 이동, 회전, 삭제를 위해서 배치된 가구를 먼저 터치(1)하여 선택한다. 선택된 가구는 보라색으로 마킹된다. 가구를 회전시키기 위해서 두 손가락을 이용하여 원하는 방향으로 드래그(2)하여 회전시킨다. 드래그(2)한 각도에 따라 가구의 회전 각도가 계산된다. 가구를 이동시키기 위해서 한 손가락을 이용하여 스크린을 터치하면 가상의 이동 패드가 생성된다. 패드를 이용하여 드래그하면 평면 위를 좌표 축과 수직으로 이동시킨다. 배치된 가구를 삭제하기 위하여 우측 상단의 삭제 버튼을 터치(3)하면 가구 오브젝트가 삭제된다.



캐릭터 시뮬레이션 모드1. - 3인칭



캐릭터 시뮬레이션 모드2. - 1인칭

캐릭터 시뮬레이션 모드1. 캐릭터 시뮬레이션 모드 버튼을 터치(1)하여 구조물에 캐릭터를 생성한다. 그 후 스크린을 터치(2)하면 가상 패드가 생성되며, 이를 이용하여 캐릭터를 구조물 내부에서 이동시킬 수 있다.

캐릭터 시뮬레이션 모드2. 캐릭터가 생성된 상태에서 캐릭터 시뮬레이션 모드 버튼을 한번 더 터치(1)하면 구조물에 생성되어있던 캐릭터의 시점으로 전환한다. 1인칭 모드에서 사용자는 직접 구조물 내부에 있는 것과 같은 가상현실을 체험할 수 있다. 모드 1과 같이 스크린을 터치(2)하면 가상 패드가 생성되어 캐릭터를 조종할 수 있다. 디바이스의 자이로센서를 이용하여 시점을 위아래로 움직일 수 있다.

3.5. 개발환경 (언어, Tool, 사용시스템 등)

- 개발 언어 : C#, C/C++
- Library : Vuforia SDK
- 개발 Tool : Unity 4.3.0 pro, Mono developer, Visual Studio 2013
- OS : Android 4.1.2
- 개발 디바이스 : Galaxy note 10.1

4. 프로그램 설명 (최대한 자세하게 기술)

4.1. 파일 구성

- ButtonController Class : 스크린 위에 버튼 UI 구동 클래스.
- ContentManager Class : 추적 중인 마커와 모드 정보를 싱글톤으로 저장하고 있는 클래스
- SetSunPos Class : 터치한 스크린 좌표를 기준으로 태양을 생성하고 건축물을 중심으로 회전시켜서 채광 모드를 제공하는 클래스.
- InventoryController Class : 배치할 가구를 선택할 수 있는 메뉴바를 제공하는 클래스.
- FurnitureMaker & FurnitureController Class : 가구를 선택하고 회전 및 이동을 지원하는 클래스. 터치한 스크린 좌표를 기준으로 Ray tracing을 사용하여 가구를 선택.
- FurnitureCollider Class : 가구와 다른 오브젝트가 충돌할 경우 겹치지 않도록 위치를 보정하는 클래스.
- PadController Class : 동선 모드 시 캐릭터의 움직임을 터치한 가상 패드를 이용하여 제어하는 클래스.
- SetCamPos Class : 동선 모드에서 1인칭 시점을 제공하기 위해 카메라 오브젝트를 제어하는 클래스.

4.2. 함수별 기능

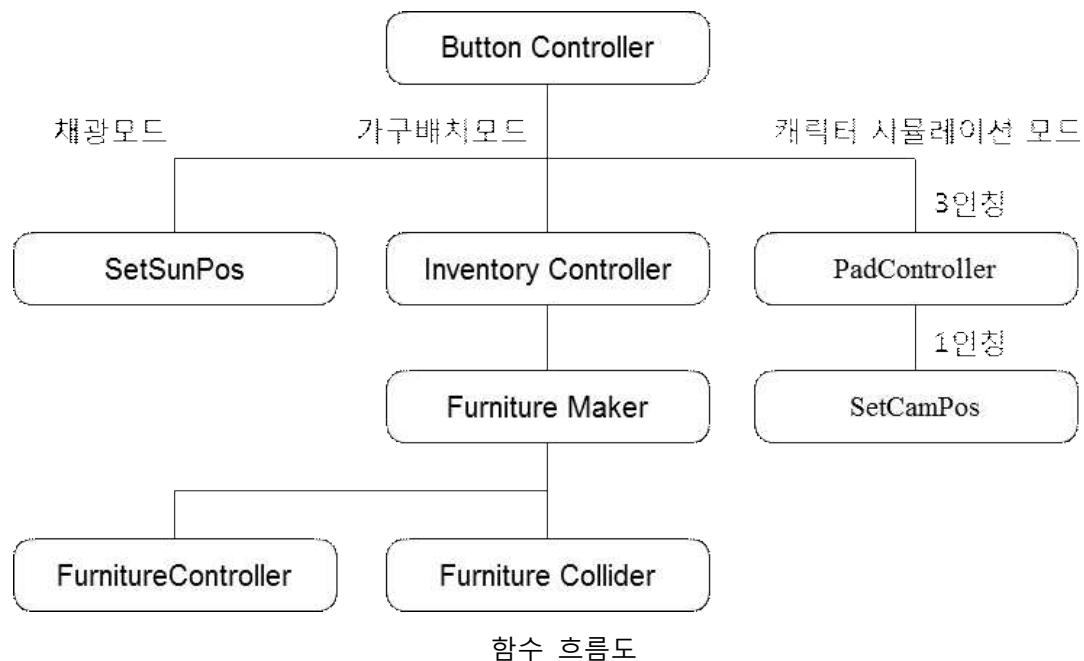
- ButtonController Class
 - Start : ButtonController가 시작될 때 이미지를 불러오고 초기화를 시킴.
 - OnGUI : 각각 버튼을 화면 위에 찍으면서 클릭 시 ContentManager에 모드를 변경시켜 모드가 변했다는 것을 알려주고, Initiating을 위한 세팅을 한다.
 - Mode3Initialize : 특별하게 캐릭터 시뮬레이션 버튼은 두 번 눌렀을 경우 1인칭 모드로 들어가기 때문에 따로 처리함.
- SetSunPos Class
 - Update : 채광 모드일 때 태양을 놓기 위한 터치를 인식하고 태양을 회전시키는 함수를 실행함..
 - CalculateSunPos : 터치 지점에 태양을 생기게 만드는 함수.
 - CalculatePlanePos : 건축물이 있는 Plane과 카메라의 거리를 측정하는 함수.
 - RotateSunLight : 태양을 회전시킴.
- InventoryController Class
 - initUISize : UI사이즈를 화면크기에 따라 다르게 세팅하는 함수.

- Start : initUISize() 실행과 초기화를 함.
- Update : 터치 이벤트를 인식하여 인벤토리를 좌우로 움직일지, 아이템을 놓을지 설정함.
- CalculatePlanePos : 건축물이 있는 Plane과 카메라의 거리를 측정하는 함수.
- OnGUI : 가구를 배치할 수 있는 메뉴바를 화면에 출력하는 함수.
- ScaleTexture : 메뉴바에 들어간 이미지 Scale을 변경함
- plainColor2DTexture : 텍스처를 plainColor로 바꿈.
- DestroyInventoryItemInstance : 카메라가 타겟을 인식하지 않았을 때 Item을 놓게 되면 제거함.
- FurnitureMaker
 - Update : 가구배치 모드에서 배치되어있는 가구를 클릭했을 때 가구 컨트롤로 가도록 유도함.
 - setFurnitureLayer : 선택된 가구가 파란색으로 변하도록 컨트롤함.
- FurnitureController Class
 - Update : 터치된 손가락 개수를 카운트해서 한 개면 패드가 나와서 무빙하도록 하고, 두 개일 때는 좌 우로 회전 할 수 있도록 세팅함.
 - setAngle : 두 손가락으로 터치했을 때 angle을 계산함.
 - calculateButtonPos : 버튼이 눌렸을 때 그 버튼이 클릭된 위치에서 plane까지 어느 정도 떨어진 위치를 받아옴.
 - getModelDirection : 이미지 타겟이 얼마나 회전되어있는지 체크함.
 - getScreentoFloorConst : Floor constant로부터 카메라까지 떨어진 거리의 비율을 구함.
- FurnitureCollider Class
 - Update : 물체가 유니티의 Physics 엔진을 이용하여 Rigidbody를 움직일 때 지금 움직이는 가구가 아닌 부딪치는 가구가 움직이거나 회전되지 않도록 고정함.
 - moveFurniture : 물체가 원하는 방향으로 움직이도록 함.
 - rotateFurniture : 물체가 원하는 방향으로 회전하도록 함.
- PadController Class
 - Update : 3인칭 캐릭터 시뮬레이션 모드 일 때 화면에 터치가 일어나면 패드를 생성하여 사람 모델을 움직일 수 있도록 세팅함.
 - CalculateButtonPos : 버튼이 눌렸을 때 그 버튼이 클릭된 위치에서 plane까지 어느 정도 떨어진 위치를 받아옴.
 - GetModelDirection : 이미지 타겟이 얼마나 회전되어있는지 체크함.
 - GetScreentoFloorConst : Floor constant로부터 카메라까지 떨어진 거리의 비율을 구함.
 - GetFloor_pos : 가구가 GetScreentoFloor에서 받아온 비율을 이용해서 카메라로부터 물체까지의 거리를 측정함.
 - InRectCheck : ButtonController에서 만든 버튼을 눌렀을 때는 패드가 생성되지 않도록 설정함.
- SetCamPos Class
 - Update : 1인칭 캐릭터 시뮬레이션 모드일 때, 자이로센서를 이용하여 위아래로 움

직이면 시점을 위아래로 움직일 수 있도록 하였고, 화면에 터치가 일어나면 패드를 생성하여 모델을 움직여 건물의 다양한 부분을 볼 수 있도록 함.

- Cam_posSet : 위아래로 시점을 움직이지 못하도록 고정함.

4.3. 주요 함수의 흐름도



4.4. 기술적 차별성

기존 증강현실을 이용한 어플리케이션은 주로 게임과 같은 엔터테인먼트 분야에 분포되어있다. 증강현실을 활용하여 사업이나 실용적인 분야에 어플리케이션은 소수이며, 사용자에게 꼭 필요한 정보가 아닌 부수적인 정보를 제공하고 있다. 본 어플리케이션은 실제로 산업에서 사용되는 건축 도면을 마커로 활용하여 증강현실에 대한 사용자의 거부감과 거리감을 줄일 수 있다. 또한, 사용자가 이해하기 어려운 도면의 정보를 가상현실의 정보와 혼합하여 직관적으로 이해할 수 있도록 도와준다.

5. 응용 분야

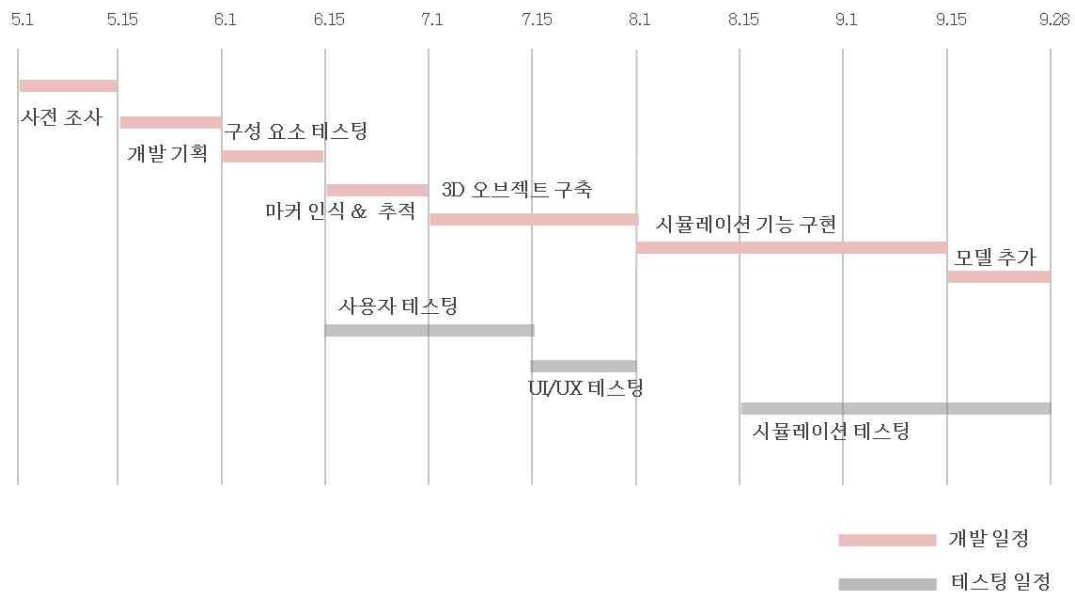
건축디자이너가 자신 디자인 건축물을 일반인 사용자에게 설명하고자할 경우 어디에나 존재하는 스마트 디바이스를 활용하여, 도면을 증강시켜 건축물 모델을 보여줄 수 있다. 이를 통하여, 물리적 종속성을 가지는 건축 모형 제작에 대한 부담을 줄일 수 있다. 또한, 다양한 시뮬레이션을 활용하여 일반인에게 이해를 도울 수 있다.

주택을 마련하고자 하는 사용자에게 해당 주택에 가구 배치와 같은 시뮬레이션을 미리 제공하여 개인에게 적합한 주택 선택을 돕는다. 그리고 자신의 주택에 인테리어를 바꾸고자 하는 경우, 직접 가구를 움직여보지 않고 시뮬레이션을 활용하는 것으로 미리 체험해 볼 수 있다.

6. 제작자 정보

No.	구분	성명	소속(학교)	부서(학과)	입학년도	담당업무
1	팀장	문성필	한양대학교	컴퓨터공학부	2010년	일정 관리. 컴퓨터 그래픽스.
2	팀원	정락준	한양대학교	기계공학부	2010년	가구 배치 구현. 3D 모델 제작.
3	팀원	박진석	포항공과대학교	기계공학과	2011년	UI / UX 제작. 시뮬레이션 알고리즘 설계.

7. 개발 단계별 기간 및 투입 인원



개발 단계별 기간

- 사전 조사 : 증강현실 어플리케이션 시장 조사 및 증강현실 응용분야 조사.
- 개발 기획 : 사전 조사를 바탕으로 한 개발 아이템 브레인 스토밍 및 개발 제안서 작성.
- 구성 요소 테스트 : Vuforia SDK 성능 테스트 및 개발 디바이스 성능 테스트.
- 마커 인식 & 추적 : Unity3D와 Vuforia SDK를 활용한 간단 테스트 어플리케이션 구현.
- 3D 오브젝트 구축 : 건축물 3D 오브젝트 제작, 가구 모델 제작, 캐릭터 모델 제작 및 애니메이션 구현.
- 시뮬레이션 기능 구현 : 채광, 가구 배치, 캐릭터 시뮬레이션 구현.
- 모델 추가 : 다양한 건축 모델과 가구 모델을 제공하기 위한 추가 구현 작업중.
- 사용자 테스트 : 구현된 어플리케이션 프로토타입으로 간단한 시연을 진행하고, 사용자로부터 기능에 대한 피드백 의견 수렴.
- UI / UX 테스트 : 버튼, 어플리케이션 아이콘 디자인 초안을 바탕으로 사용자에게 기능을 직관적으로 이해할 수 있는지에 대한 테스트 진행.
- 시뮬레이션 테스트 : 목표로 하는 FPS를 각 기능에서 만족하는지 확인 및 각 시뮬레이션이 요

구조건을 만족하는지 테스트. 최적화 작업 진행중.

Jun 29, 2014 – Sep 21, 2014

Contributions: **Commits** ▼

Contributions to master, excluding merge commits



Github repository commit graph