

## 졸업프로젝트 결과 보고서

작품 제목	사용자 동작 시각화 앱, <Feuille>
작품 기능	사용자의 움직임을 특정한 "물체"(이하 "물체")의 움직임을 통해 표현하는 프로그램.
작품 계획 기간	2015년 01월 01일 ~ 2015년 05월 31일
작품 제작 기간	2015년 06월 01일 ~ 2015년 09월 30일
개발 환경	OS: MacOS X 언어: C#, Java, Php Tools: Unity 5.1.1f, Android Studio

이름	구분	학교	학과	학년	전화번호
김성훈	PL	한양대학교	컴퓨터공학과	4	010-9049-7185

한양대학교 공과대학 컴퓨터공학부

## 졸업프로젝트 자가 평가서

항 목	평가 점수
<b>1. 성실성</b> (개발목표 대비 완료와 정리: 20점 만점)	15점
개발하고자 하는 핵심 기능은 완성됨. 고군분투하며 여러 가지 기능을 구현하려 애씀.	
<b>2. 기술성</b> (우수한 개발 능력과 기술 구현: 25점 만점)	20점
웹에 파일 업로드, 인터넷을 통한 동기화 기능 구현. 그러나 기술적으로 높은 수준의 기술을 요구했던 프로젝트는 아닌 것 같다.	
<b>3. 창의성</b> (구현의 창의성과 차별성: 25점 만점)	25점
사람들에게 재미를 줄 수 있고 색다른 프로젝트라고 평가받기도 했음.	
<b>4. 응용성</b> (개발 결과물에 대한 현업 적용: 20점 만점)	15점
아직 발달하지 않은 분야라서 높은 점수를 주긴 어려움.	
<b>5. 사업성</b> (향후 활용성 및 사업화: 10점 만점)	7점
사업적인 것보다는 공익적이거나 교육적인 목적에 더 어울리지 않나 싶음.	

# 목 차

1. 과제 개요
  - 1.1 개발 목적 및 동기
2. 과제 내용
3. 기술 내용 (동작 방식 다이어그램)
4. 개발구조
5. Use-Case 스크린샷
6. 개발환경
7. 사용환경
8. 결론 및 아쉬운 점

## 1. 과제 개요

### 1.1 개발 목적 및 동기

- (1) 개요 : 사용자의 움직임을 특정한 “물체”(이하 “물체”)의 움직임을 통해 표현하는 프로그램.
- (2) 목적 : 시각화 된 “물체”의 움직임을 보고 자신의 움직임이 어떤 양상으로 나타나는지 살펴보고, 다른 사람들의 “물체”와 상호작용을 하며 흥미와 재미를 유발하는 것이 목적.
- (3) 활용 가능성 : 댄스클럽 등에서 춤을 출 때, 시각화 된 자신의 움직임을 프로젝터로 투사된 큰 화면을 통해 봄으로써 재미있는 경험을 주게끔 할 수 있다.

## 2. 과제 내용

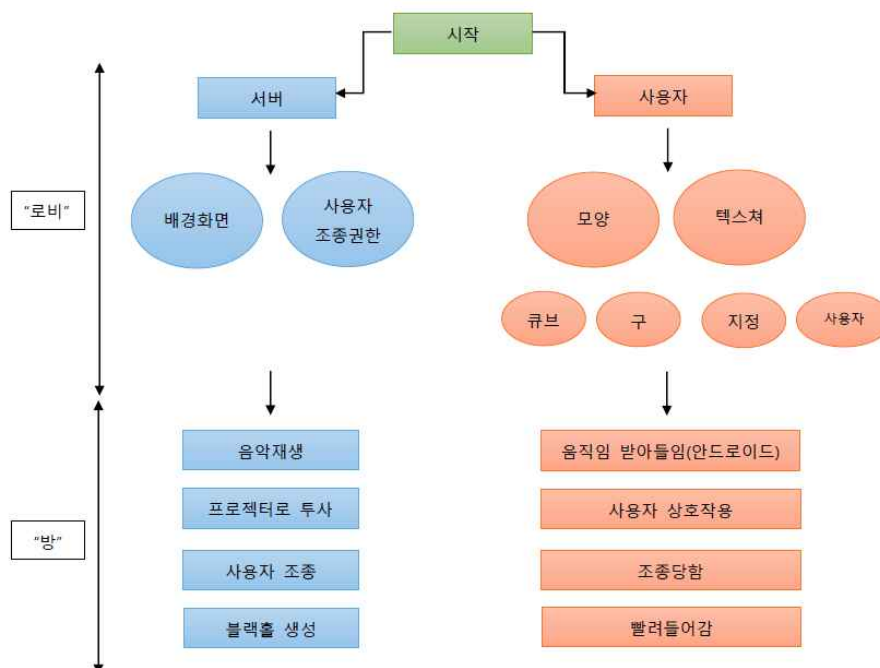
### (1) 사용자

- 한 사용자의 움직임은 화면 상의 “물체”의 움직임으로 표현된다. (이동, 크기 변화 등)
- 각 사용자는 한 서버에 접속하여 중앙의 큰 화면을 통해 서로의 움직임을 확인하기도 하고, 다른 사용자의 “물체”와 상호작용 할 수 있다. (핑퐁 등)

### (2) 서버

- 각 사용자의 “물체”를 직접 조종하여 이동시킬 수 있다.
- “블랙홀”을 생성해 물체들을 한 곳으로 빨아들일 수 있다. 마우스를 떼면 블랙홀이 사라지며 모든 사용자의 휴대전화에서 진동이 울린다.
- “물체”를 선택하여 크기를 크게 할 수 있다.
- “물체”를 선택하면 물체가 마구 움직이거나, 부르르 떠는 효과를 낼 수 있다. 이때, 선택된 사용자의 휴대전화에서 진동이 울린다.

## 3. 기술 내용 (동작 방식 다이어그램)



## 4. 개발구조

### (1) 안드로이드

#### ① 역할

- 가속도계로 움직임을 받아들이고 유니티 단으로 내려줌.

#### ② 함수 내용

- 유니티와 연결해주는 함수가 대부분 (onCreate, OnDestroy 등은 ndk로 작성된 유니티 모듈을 불러오기 위한 것)
- setCustomTexture(), setCustomBackground()은 사용자가 지정한 사진을 불러와 이미지 서버에 업로드 하는 함수. 갤러리 인텐트를 호출하여 사용자로부터 사진을 입력받고, getPathFromUri()를 통해 절대경로를 return 받음.
- vibratePhone()은 서버에서 조종하면 진동하게끔 하는 함수.
- setCurrentObjectName()은 현재 움직일 내 “물체”가 무엇인지 지정하는 함수임. 내 것이 아닌 물체를 움직이면 사용자 단에서는 움직이는 것처럼 보일 수 있으나, 실제로 움직인 것이 아니기에 서버와 움직임이 동기화 되지 않음. 이런 문제점을 피하기 위해 사용하는 함수.
- mTask는 0.1초마다 유니티 단의 ChangeLeafPosition()을 호출하는 Timer Task임. onResume()때 호출되므로, 나갔다 들어와도 연속적으로 움직임을 받을 수 있음. onPause()에서 리스너와 mTask 해제.
- UploadFile 클래스는 POST 프로토콜을 이용하여 이미지 업로드 해주는 클래스.
- 자세한 사항은 <별첨 1>의 소스코드 참조.

### (2) 유니티

- ① 3. 작동방식에서 밝힌 흐름대로 따라가는 것이 개요.
- ② 안드로이드에서 받은 신호를 바탕으로 물체에 힘을 주는 것이 골자. 나머지는 단순 기능 구현.
- ③ 서버 측과 클라이언트 측이 나뉘어서 프로그래밍 되어있음. (예. PhotonInit, PhotonInitServer, Scene도 나뉘어서 편집되어 있음.)
- ④ 네트워크는 PhotonCloud 플러그인을, “물체”를 흔드는 것은 iTween 등의 플러그인을 별도로 사용하였음.
- ⑤ 함수내용은 너무 많아서 지면상으로 서술하기 곤란함. 또, 유니티 내부의 scene과 game object 별로 스크립트가 묶여있으므로 전반적으로 서술한다고 해도 구조적이진 않음. <별첨 1>을 통해 소스코드를 따로 공개함.

### (3) 이미지 서버(PHP)

- ① 사용자 지정(커스텀) 텍스처나 배경 화면을 사용하면, 이미지 서버에 업로드 함.
- ② 이미지 서버는 Amazon EC2 서버를 이용하여 리눅스 가상 서버를 호스팅 하였음.
- ③ httpd와 php를 이용하여 파일을 올림.
- ④ 사용자가 방에 입장하는 순간, RPC를 호출하여 이미지의 주소를 받아 각자의 텍스처나 배경화면을 변경하게 하였음.

## 5. Use-Case 스크린샷

(1) 클라이언트 Use-case (쭈쭈는 Use-case)



(2) 서버 Use-case (Drag & Drop, Zoom 기능 시연 중)



\* 이를 활용한 Use-case는 같은 압축 파일에 동영상상이 있으므로 참조 바람.

## 6. 개발환경

- ① 유니티 5.1.1 f1
  - 움직임을 시각화 하는 핵심 기능 구현, 사용자들의 “물체”와 상호작용 하도록 물리 시뮬레이션
- ② Android Studio 1.3
  - 움직임을 유니티로 전달해주는 역할, 유니티와 연결해주는 통로 역할.
- ③ MacOS X Yosemite, Amazon EC2 Linux Server, PHP
  - 업로드 서버 프로그래밍.

## 7. 사용환경

- ① 클라이언트 : 안드로이드 진저브레드 이상.  
(테스트 단말 : 삼성 갤럭시S, 갤럭시S4, 갤럭시U, 갤럭시 알파,  
LG GK, HTC 넥서스원)
- ② 서버 : MacOS X Yosemite 이상, Windows 8.1 이상.

## 8. 결론 및 아쉬운 점

- ① 한 번의 사이클을 돌고 프로젝트를 완료한 것이 일단 매우 기쁘다. 하지만 아래와 같은 점이 아쉽다.
- ② 기능 구현에 급급해 객체화를 완벽히 하지 못하고 스파게티 코드를 작성한 점. 모듈화를 확실히 하여 기능을 분리하여야 했지만, 유니티 특성 상 game object를 벗어나는 호출을 하기가 어려움. (비용이 큼) 비용과 효율성 사이에서 고민을 하다 보니 이도저도 아닌 스파게티 코드를 만들게 됨.
- ③ 충분한 테스트 셋이 없었던 점. 단말기가 부족하여 3대 정도까지 붙여봤을 때는 괜찮았었는데 5개가 넘어가니 심하게 끊어지기 시작. 어려운 조건에서 겨우 영상을 찍을 수 있었음. 단말기를 더 확보하는 것이 중요하겠구나 싶었음. (그리고 동작하지 않거나 이상하게 동작하는 단말기도 있었다.)

\* 배경음악은 저작권 문제로 제출한 소스에 포함되어 있지 않음.