

Remote Present

충남대학교 공과대학 컴퓨터공학과
진재연 (daramkun@daram.pe.kr)
2012. 08. 31

목차

개요

- 프로젝트 시작 동기
- 프로젝트 소개
- 프로젝트에 사용한 기술

시장 분석

애플리케이션 특징

사용자 인터페이스

- 스마트폰 앱 사용자 인터페이스
- 리시버 프로그램 사용자 인터페이스

지원 플랫폼

설계

- 사용 사례 다이어그램
- 클래스 다이어그램

결과

개요

프로젝트 시작 동기

스마트폰용으로 출시되는 많은 리모콘 애플리케이션들이 버튼 터치식으로 만들어져 있기 때문에 프레젠테이션과 같이 청중과 계속해서 아이컨택(Eye-contact)을 하면서 말을 해야 하는 상황에서 이러한 리모콘 애플리케이션을 사용하게 되면 계속해서 스마트폰 화면을 보고 버튼을 눌러야 하기 때문에 청중의 발표 집중도가 떨어질 수 있다는 단점과, 발표자의 진행이 계속 끊긴다는 단점이 존재합니다. 이는 직접 이러한 스마트폰 애플리케이션을 만들어서 사용해보거나, 공개된 애플리케이션들을 사용해보고 직접 체험해봤기 때문에 저 스스로가 새로운 방식의 애플리케이션의 필요성을 느끼게 되었습니다.

2011년에 같은 학과의 졸업반 선배가 만드셨던 키넥트(Kinect)를 이용한 발표 제어 프로그램에서 힌트를 얻어 본격적으로 아이디어를 정리하기 시작했습니다. 주요 키워드는 “제스처(Gesture)”였으며, 당시 힌트를 얻었던 프로그램은 모션 제스처를 이용했기 때문에 발표에는 적합하지 않았다고 생각하여 아주 작은 움직임으로 인식할 수 있는 터치 제스처를 이용하기로 했습니다.

프로젝트 소개

이 애플리케이션은 PC(Windows)와 같이 사용해야 100% 기능을 이용할 수 있습니다. PC와는 인터넷을 이용하여 연결되며, 이를 위해 PC에 “리시버(Receiver)”라고 불리는 프로그램을 설치해야 합니다.

PC와 연결된 애플리케이션은 오로지 터치 제스처로만 동작하며, 이 터치 제스처는 발표와 관련된 각각의 동작으로 연결됩니다. 각각의 제스처에 관한 동작은 아래 표로 정리되어 있습니다.

제스처	동작
왼쪽으로 밀기	이전 슬라이드로 이동(리얼 스와이프 기능 시 다음 슬라이드)
오른쪽으로 밀기	다음 슬라이드로 이동(리얼 스와이프 기능 시 이전 슬라이드)
탭	스크린 레이저 켜/끔(켄 상태에서는 다른 제스처 실행 안 됨)
더블 탭	통신 종료
롱 탭	프레젠테이션 시작

리시버와 애플리케이션의 연결은 자동 연결과 수동 연결로 나뉘어집니다.

자동 연결은 완전히 자동으로 연결되는 것은 아니며, “브로드캐스팅”이라는 행위를 시작함으로써 리시버가 이를 인식하여 1회용 패스워드를 입력하면 바로 연결되는 기능을 말합니다. 따라서 터치 한 번과 컴퓨터 키보드 키 네 개만 입력하면 바로 연결이 가능합니다. 수동 연결은 IP 주소로 연결되는 기능입니다. 한 번 연결했던 컴퓨터라면 기록이 남기 때문에 빠르게 연결할 수 있으나, 매번 다른 컴퓨터, 다른 IP에서 사용하는 경우에는 매번 IP를 고쳐 작성해야 하기 때문에 불편할 수 있습니다.

프로젝트에 사용한 기술

기술	자세한 내용
터치 제스처	iOS에서 기본적으로 제공하는 Gesture Recognizer API를 사용합니다.
BSD 네트워크 소켓	네트워크 통신은 모든 플랫폼에서 제대로 호환할 수 있도록 Objective-C 방식의 소켓이 아닌 UNIX 기본 내장 API인 BSD 소켓 API를 사용합니다.
WPF	리시버에서 사용한 기술으로, 디자인 코드와 실행 코드를 분리하여 설계할 수 있습니다.
C# 네트워크 소켓	C#의 소켓을 통해 리시버에서 통신을 수행했습니다. BSD 소켓과 큰 차이가 없기 때문에 사용에 큰 무리가 없었습니다.

시장 분석

□



주요 세 모바일 운영체제 마켓들에 대해 비슷한 앱들을 찾아보면 매우 많은 결과가 나오는 것을 확인할 수 있습니다. 특히 Apple AppStore나 Google Play에 매우 많은데, 공급 및 수요는 많으나 실제로 발표 자리에서 사용하는 경우는 거의 없습니다.

그 이유 중 하나가 바로 사용하기에 불편하고 UX가 발표에 맞지 않다는 점입니다. 버튼식이기 때문에 발표를 하면서 버튼을 계속 확인하고 눌러주어야 합니다. 또한 대부분의 앱이 IP 입력으로 연결하는 방식이기 때문에 컴퓨터 네트워크 동작 방법(IP, LAN 등)에 대한 공부를 하지 않으면 사용하기 힘듭니다.

애플리케이션 특징

1. 처음 리시버 프로그램과 연결할 때 외에는 모든 작업을 터치 제스처로 수행합니다.
2. 같은 네트워크 안에 있는 리시버와 앱을 빠르게 연결할 수 있습니다.
3. 다른 네트워크에 있는 PC와는 수동 연결을 통해 연결할 수 있습니다(포트 열림 상태에 따라 다름).
4. 연결 시 OTP를 사용하여 안정성 및 보안성을 향상시켰습니다.
5. 확장을 고려한 패킷 설계를 하여 기능의 추가 또는 변경에 대한 업데이트 시 하위 호환성을 제공합니다.

사용자 인터페이스

스마트폰 앱 사용자 인터페이스



맨 처음 앱을 켜면 스플래시 이미지가 잠깐 표시된 후, 위의 맨 왼쪽 화면과 같은 뷰가 나타납니다. 이 뷰는 좌우로 페이지를 넘길 수 있습니다. 페이지는 두 페이지로 구성되며, 브로드캐스트 페이지와 수동 연결 페이지로 구성됩니다. 하단 오른쪽의 i 버튼을 누르면 설정 화면으로 넘어갑니다.

브로드캐스트 또는 수동 연결에 성공하면 맨 오른쪽 화면과 같이 제스처 뷰가 나타나게 됩니다. 어두운 발표회에서 밝은 색 또는 눈에 띄는 색은 청중의 관심을 그쪽으로 돌릴 수 있기 때문에 검은색 화면으로 했으며, 최소한의 정보만 노출합니다.

리시버 프로그램 사용자 인터페이스



리시버 프로그램은 매우 간단하게 구성되어 있습니다. 기기들이 브로드캐스트를 하면 리스트에 해당 기기들이 표시되고, 원하는 기기를 선택하고 더블 클릭을 하거나 연결을 누르면 패스워드 창이 나타나며, 여기에 올바른 패스워드를 작성하여 연결에 성공하면 연결이 끊길 때까지 모든 컨트롤이 비활성화됩니다.

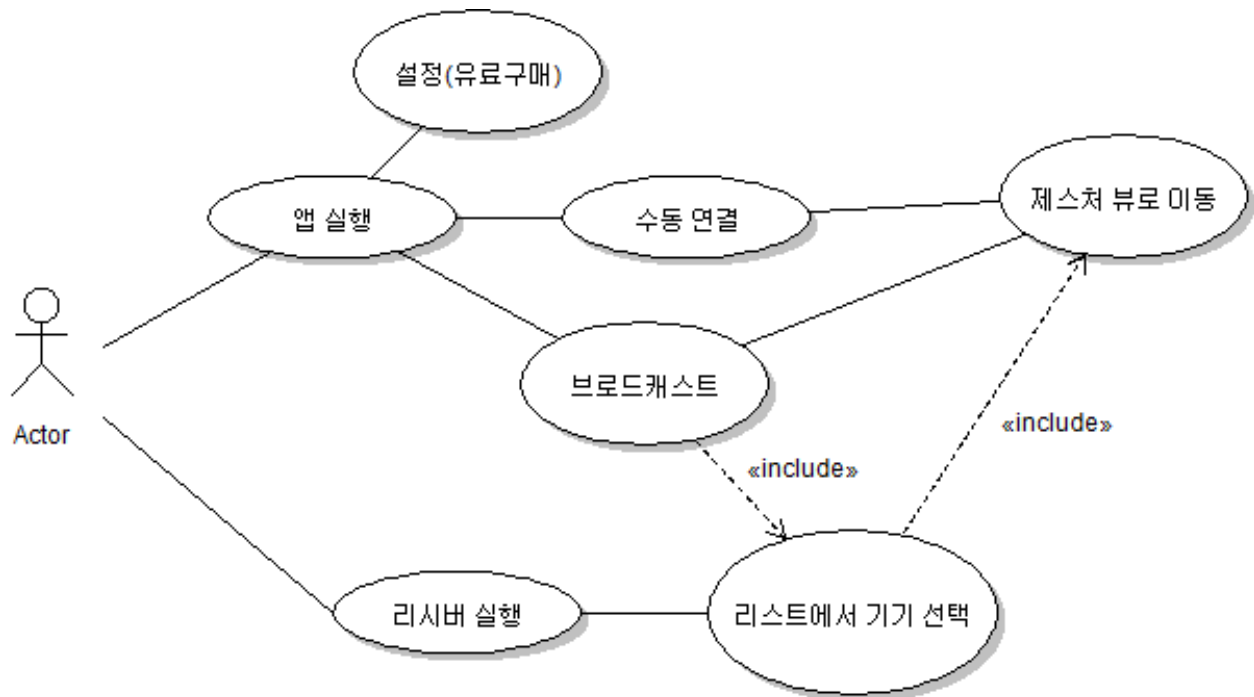
지원 플랫폼

플랫폼	지원 최소 버전
iOS	5.0
Windows Phone 7	7.1
Android	4.0.3 (예정)

Remote Present의 Windows Phone 7용 앱은 이미 7.1(Mango)용으로 마켓플레이스에서 판매되고 있습니다. iOS용은 현재 마무리 테스트 단계에 있으며, iOS 6.0에 대비한 디자인 수정 작업을 병행하고 있습니다. Android는 iOS용 앱 작업이 마무리되면 시작하게 될 예정입니다.

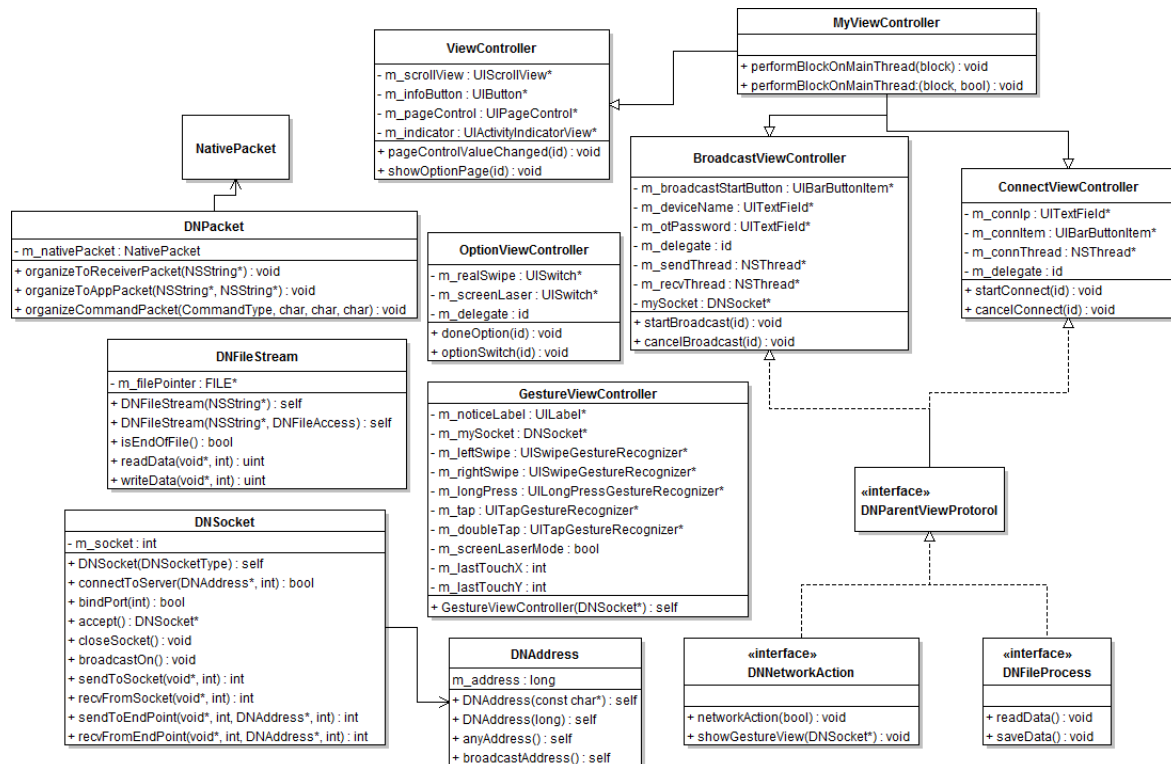
설계

사용 사례 다이어그램



사용자는 반드시 앱과 리시버를 같이 사용해야하면 제스처 뷰를 이용하여 원하는 기능을 사용할 수 있습니다.

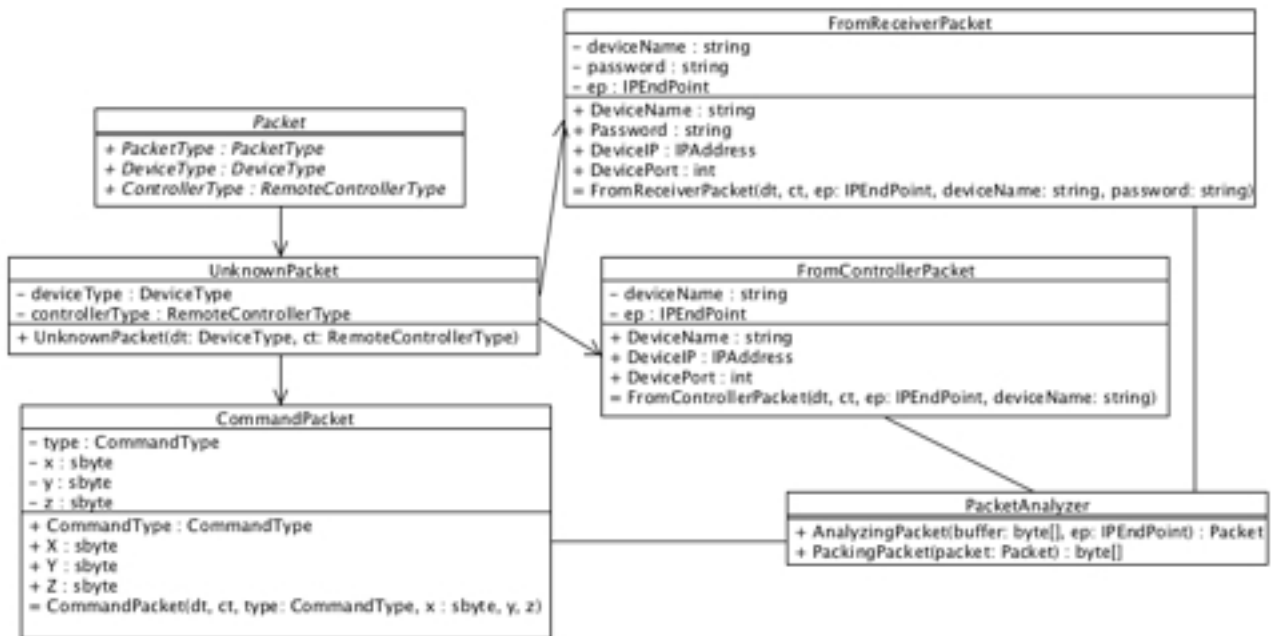
클래스 다이어그램



RemotePresent에서 사용하는 클래스의 개수는 총 14개로, 이 중 직접 구현되지 않는 Protocol은 세 개이며 C 형태의 구조체가 한 개로 구성됩니다. 대부분의 클래스는 ViewController로서 화면 구성에 사용됩니다. BSD 소

from Smartphone to PC

켓은 DNSSocket이라는 이름으로 재구성하여 사용하고 있으며 C 형식의 파일 입출력 함수들도 DNFileStream이라는 이름으로 재구성하여 사용하고 있습니다.



위 클래스 다이어그램은 리시버 프로그램의 클래스 다이어그램입니다. C#에서는 구조체를 통째로 읽어올 수 없기 때문에 먼저 바이트 배열로 읽어와 패킷으로 구성하는 방식으로 하고 있기 때문에 위와 같은 설계를 통해 제작하였습니다. 위 클래스 외에 XAML과 같이 사용되는 클래스들은 중요하지 않아 따로 첨부하지 않습니다.

결과

직접 사용할 스마트폰 애플리케이션을 만들어보면서 각종 플랫폼을 공부할 수 있어서 좋았습니다. 아직 Android용 앱 개발이 남아있지만 스스로 만족스럽습니다. 플랫폼에서 지원하는 기능인지 제대로 알지 못하여 설계가 틀어지기도 했고, 초반에 의도했던 대로 코드가 실행되지 않아 좌절도 했었지만 하나하나 차근차근 고쳐 결국은 구현하려고 했던 모든 기능을 구현할 수 있었습니다. 아직 고쳐야 할 점이 많기 때문에 지속적으로 보완해서 어떠한 오류도 없고 사용자가 더 편하게 사용할 수 있도록 만들어볼 계획입니다.