

RemotionControl

Hogy lesz maroktelefonból maroklőfegyver?

- mobil alkalmazás
- mozgásérzékelős távirányító
- lézer-pointer
- first person shooter kontroller

Motiváció

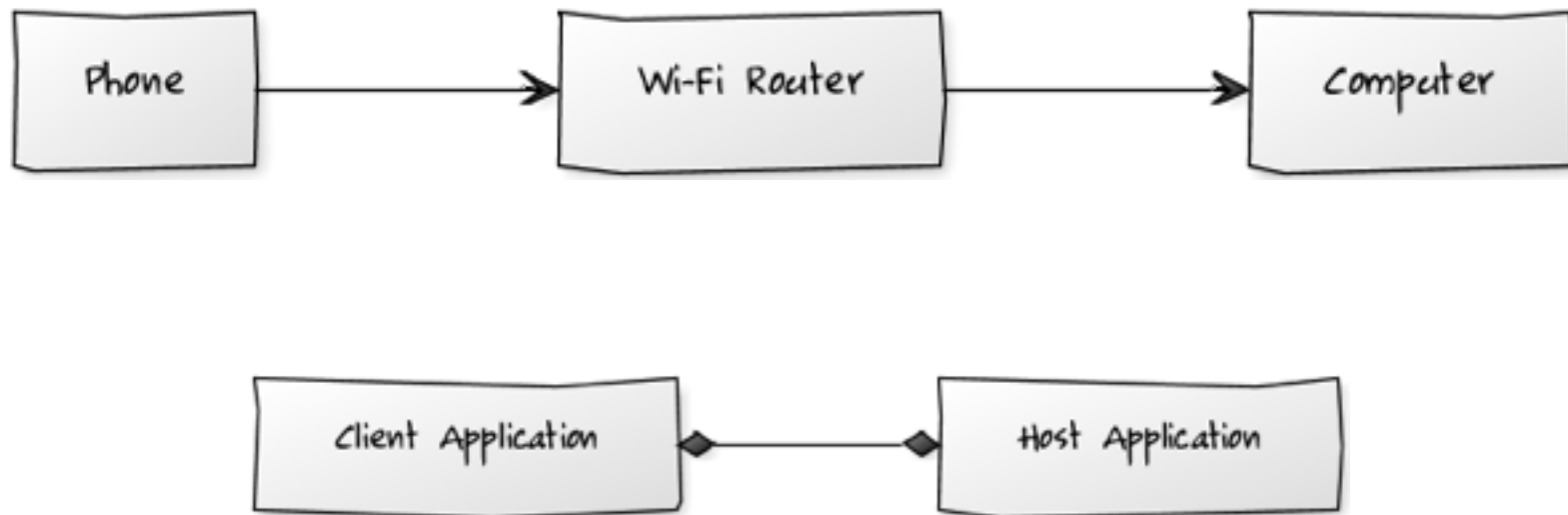
- technikailag lehetségesnek tűnt, mégsem csinálta meg senki
- kíváncsi voltam mennyire lenne használható
- kíváncsi voltam wifin keresztül mennyire real-time az UDP
- kíváncsi voltam mennyire működik a Bonjour

Nehézségek

- abszolút pozíció hiánya, fél évre leállt tőle a project
- horizontálisan most sem tartható középen a középpont
- manuális utánállítás kell

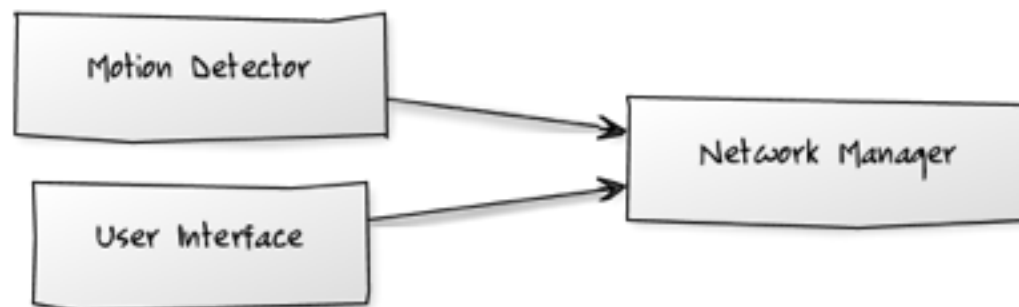
Működés

- detektáljuk a telefon lineáris és a szöggyorsulását
- wi-fin keresztül átküldjük a nyers adatokat a desktop gépre
- a desktop gépen egér eseményekké alakítjuk az adatokat

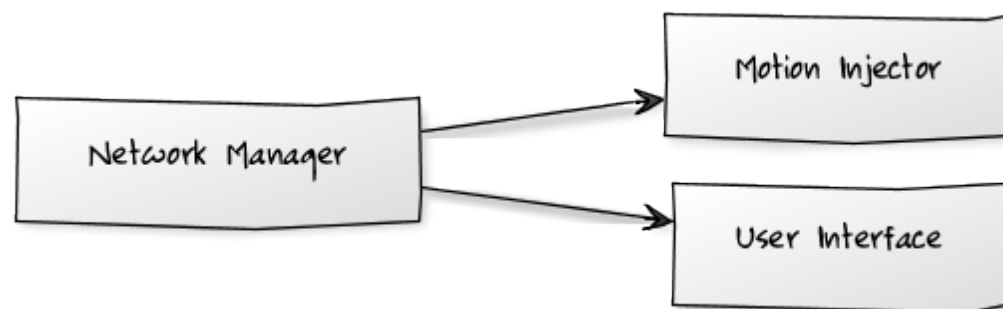


Alkalmazás séma

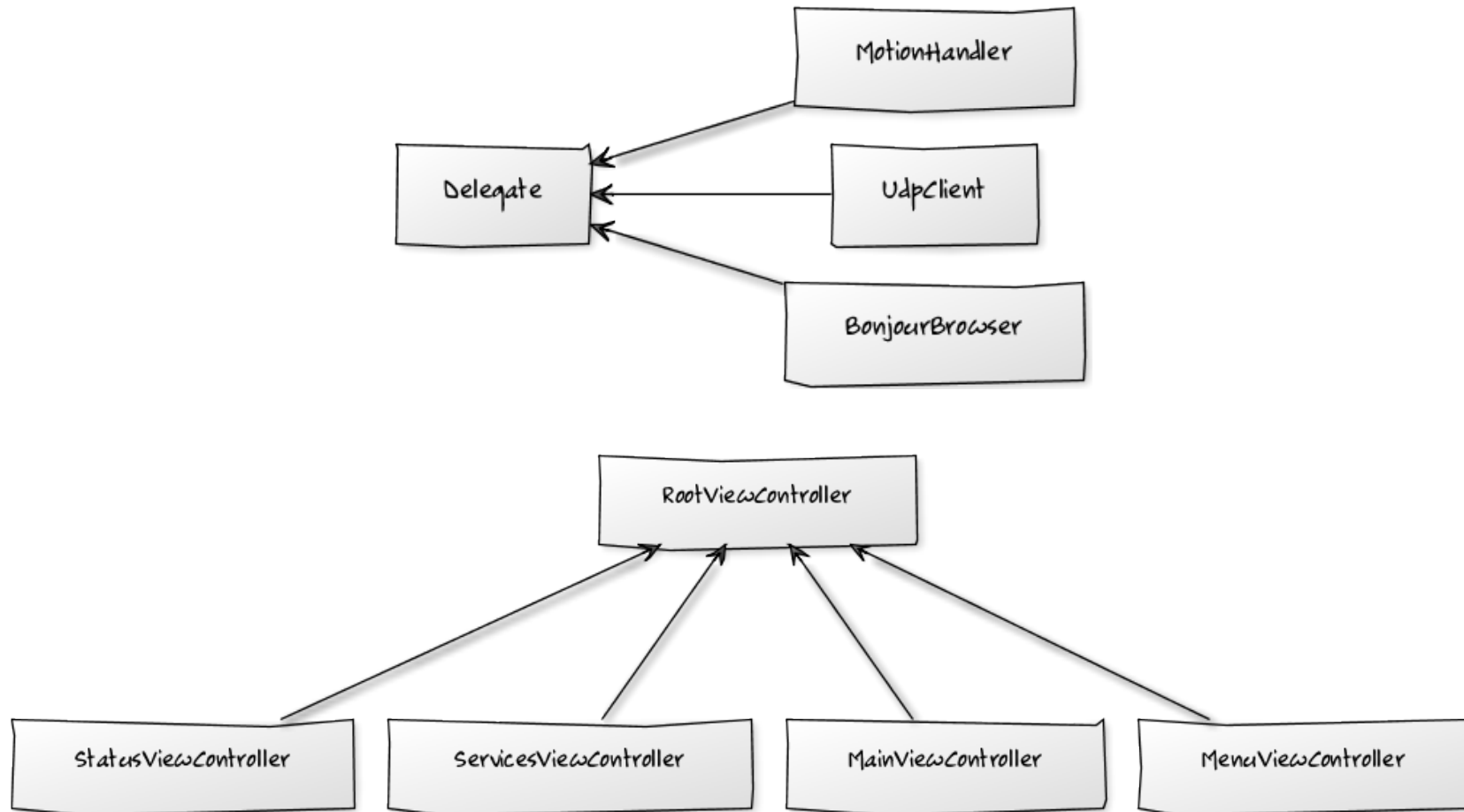
kliens



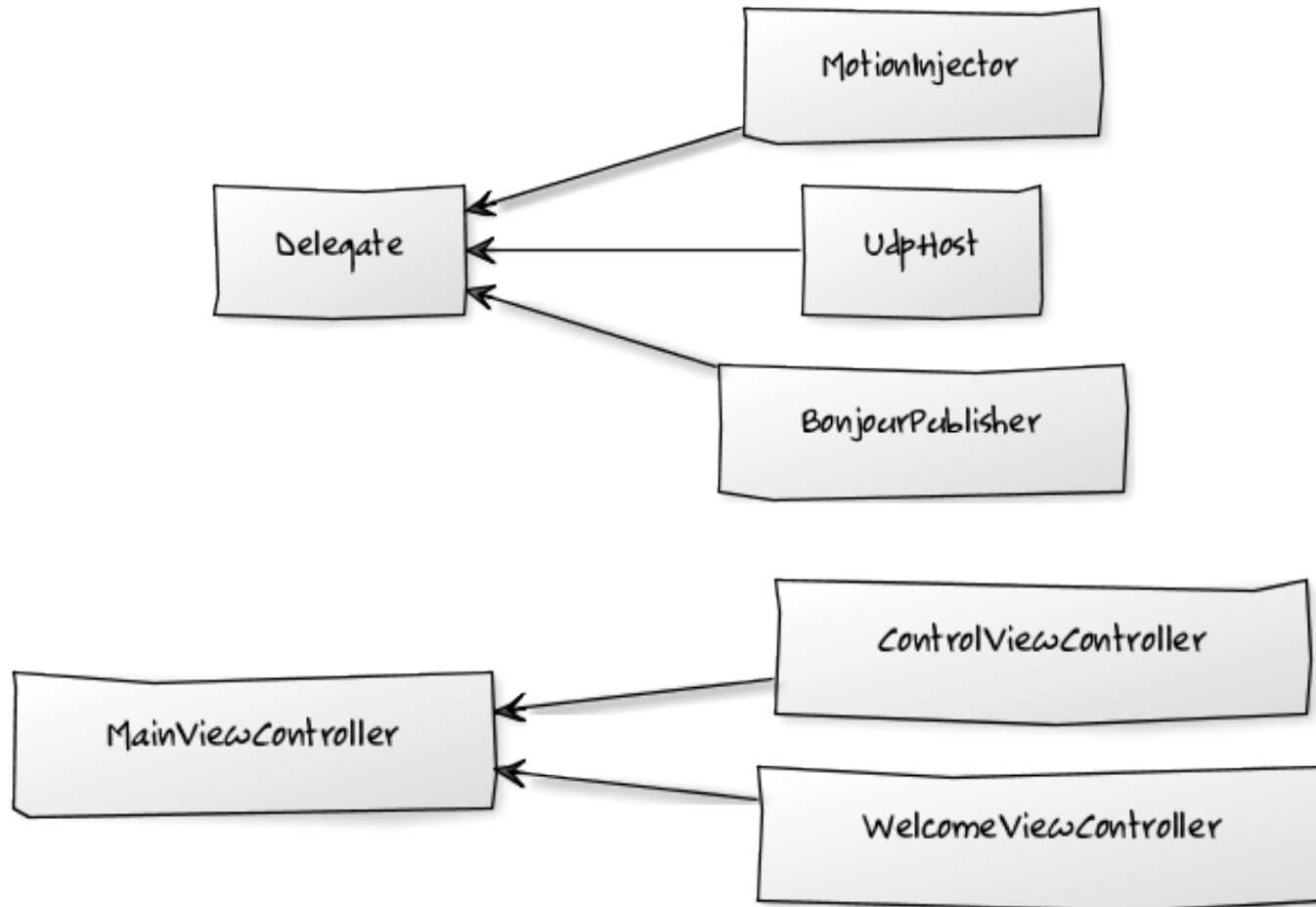
host



Részletes séma - kliens



Részletes séma - host



Protokoll

- 25 byteos csomagokat használunk, típus + 3 * 8 double

```
// packet format 1 ( type ) + ( 24 ) empty
// or          1 ( type ) + ( 3 * 8 ) rotation / motion
// or          1 ( tyoe ) + ( 1 ) button id + ( 1 ) button state + ( 22 ) empty

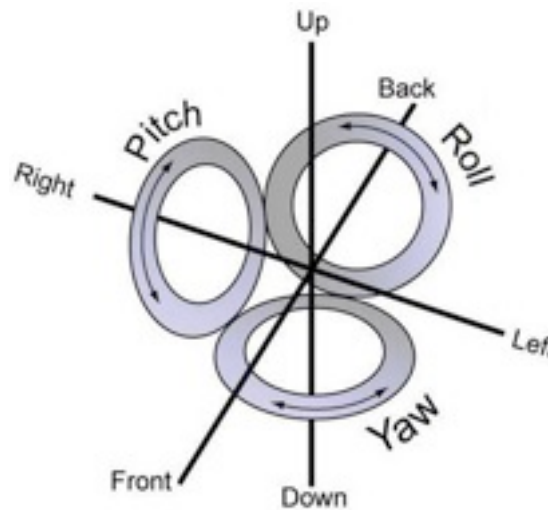
#define kProtocolPacketSize          25
#define kProtocolPingDelay          .5
#define kProtocolTimeOut            2

#define kProtocolTypePing           0x00
#define kProtocolTypePong           0x01
#define kProtocolTypeButton         0x02
#define kProtocolTypeRotation       0x03
#define kProtocolTypeDisconnect     0x04

#define kProtocolButtonA            0x00
#define kProtocolButtonB            0x01
#define kProtocolButtonC            0x02
#define kProtocolButtonD            0x03
#define kProtocolButtonE            0x04
#define kProtocolButtonF            0x05
#define kProtocolButtonG            0x06
#define kProtocolButtonH            0x07
#define kProtocolButtonI            0x08
#define kProtocolButtonJ            0x09
#define kProtocolButtonK            0x10
#define kProtocolButtonL            0x11
#define kProtocolButtonM            0x12
#define kProtocolButtonN            0x13
#define kProtocolButtonO            0x14
#define kProtocolButtonP            0x15

#define kProtocolButtonStateDown    0x01
#define kProtocolButtonStateUp      0x00
```


Mozgásérzékelés



gyorsulásérzékelő és giroszkóp

Állapotkezelés

- kilenc működési fázisa van a kliens alkalmazásnak ugyanolyan vagy különböző UI állapotokkal

```
#define kStateMenu           0
#define kStateBrowsing      1
#define kStateSelecting     2
#define kStateResolving     3
#define kStateNoService     4
#define kStateConnecting    5
#define kStateTimeout       6
#define kStateDisconnected  7
#define kStateConnected     8
```

kilenc függvényünk is lesz állapotváltáshoz, mindegyikben meg kell vizsgálnunk az előző állapotokat.

```
- ( void ) switchToMenuState;
- ( void ) switchToBrowsingState;
- ( void ) switchToSelectingState;
- ( void ) switchToResolvingState;
- ( void ) switchToNoHostState;
- ( void ) switchToConnectingState;
- ( void ) switchToTimeoutState;
- ( void ) switchToConnectedState;
- ( void ) switchToDisconnectedState;
```

Megosztott iOS-OS X kód

```
#define BUILD_TARGET_OSX 1

#ifdef BUILD_TARGET_OSX

#import <AppKit/AppKit.h>
@interface ButtonView : NSView

#else

#import <UIKit/UIKit.h>
@interface ButtonView : UIView

#endif

#ifdef BUILD_TARGET_IOS

- ( void ) touchesBegan : ( NSSet* ) theTouches
                  withEvent : ( UIEvent* ) theEvent
{

}

#endif

#ifdef BUILD_TARGET_OSX

- ( void ) mouseDown : ( NSEvent* ) theEvent
{

}

#endif
```

Köszcsiii!!!

milgra.blogspot.com
milgra.com