Gesture Connect
Trevor Pering
Yaw Anokwa
Roy Want
Lots of people out there
Lots of people have phones
NFC/RFID is emerging tech for tangible interaction – mostly for payment
Starting to crop up in all sorts of places
NFC extends RFID to provide a mechanism for two-way data transfer.
However – the interaction is still very limited, and dominated by interaction through the small display of the phone.
So – consider the scenario where trusty Yaw is returning home, and he has been watching a video on his phone
He sits down, but it's kinda painful to watch the video on his small device
So, he turns on the TV to see what is playing
He really likes it, so he makes a simple gesture to capture the program so he can view it later.
Now he can easily check the details of the program, saved for later viewing.
So, now he can happily finish watching his program...
Or to capture the location and/or menu for later return, if he doesn't find anything better
Also can be used to easily push his room temperature settings to the thermostat.
Alternately, he can use the on-screen interface to do a Bluetooth scan or something similar to discover the TV and push his content – not very convenient.
However – the interaction is still very limited, and dominated by interaction through the small display of the phone.
So, what we wanted to do was extend the capability of the phone.
Using a standard SD-card interface
To form a prototype system allowing people to explore emerging interaction techniques with such devices.
Whizzy schematics to impress people
Very very quickly
Combined with custom hardware to add NFC + Accel capability
The same basic technique can be applied to different phones, providing they provide the standard SD-card interface.
```c
#include "stdio.h"

#define _GNU_SOURCE

#define __LD__

/* This is the signature for the signal function that
   takes a signal number and a function pointer as
   arguments. */

int signal(int sig, void (*handler)(int)) {
    int old_handler;

    old_handler = signal->signal_table[sig];
    if (old_handler == 0) {
        signal->signal_table[sig] = handler;
        return 0;
    } else {
        return old_handler;
    }
}

#define PREG_SIZE 16

struct reg {
    int reg[PREG_SIZE];
    short reg_base;
    short reg_size;
    short reg_num;
};

#define SIG_ERR 1
#define SIG_DFL 2
#define SIG_IGN 3

#define SIGTRAP 0
#define SIGSTOP 1
#define SIGKILL 2
#define SIGCONT 3
#define SIGSTOP 4
#define SIGTRAP 5
#define SIGTRAP 6
#define SIGTRAP 7
#define SIGTRAP 8
#define SIGTRAP 9
#define SIGTRAP 10
#define SIGTRAP 11
#define SIGTRAP 12
#define SIGTRAP 13
#define SIGTRAP 14
#define SIGTRAP 15
#define SIGTRAP 16
#define SIGTRAP 17
#define SIGTRAP 18
#define SIGTRAP 19
#define SIGTRAP 20
#define SIGTRAP 21
#define SIGTRAP 22
#define SIGTRAP 23
#define SIGTRAP 24
#define SIGTRAP 25
#define SIGTRAP 26
#define SIGTRAP 27
#define SIGTRAP 28
#define SIGTRAP 29
#define SIGTRAP 30
#define SIGTRAP 31
#define SIGTRAP 32
#define SIGTRAP 33
#define SIGTRAP 34
#define SIGTRAP 35
#define SIGTRAP 36
#define SIGTRAP 37
#define SIGTRAP 38
#define SIGTRAP 39
#define SIGTRAP 40
#define SIGTRAP 41
#define SIGTRAP 42
#define SIGTRAP 43
#define SIGTRAP 44
#define SIGTRAP 45
#define SIGTRAP 46
#define SIGTRAP 47
#define SIGTRAP 48
#define SIGTRAP 49
#define SIGTRAP 50
#define SIGTRAP 51
#define SIGTRAP 52
#define SIGTRAP 53
#define SIGTRAP 54
#define SIGTRAP 55
#define SIGTRAP 56
#define SIGTRAP 57
#define SIGTRAP 58
#define SIGTRAP 59
#define SIGTRAP 60
#define SIGTRAP 61
#define SIGTRAP 62
#define SIGTRAP 63
#define SIGTRAP 64
#define SIGTRAP 65
#define SIGTRAP 66
#define SIGTRAP 67
#define SIGTRAP 68
#define SIGTRAP 69
#define SIGTRAP 70
#define SIGTRAP 71
#define SIGTRAP 72
#define SIGTRAP 73
#define SIGTRAP 74
#define SIGTRAP 75
#define SIGTRAP 76
#define SIGTRAP 77
#define SIGTRAP 78
#define SIGTRAP 79
#define SIGTRAP 80
#define SIGTRAP 81
#define SIGTRAP 82
#define SIGTRAP 83
#define SIGTRAP 84
#define SIGTRAP 85
#define SIGTRAP 86
#define SIGTRAP 87
#define SIGTRAP 88
#define SIGTRAP 89
#define SIGTRAP 90
#define SIGTRAP 91
#define SIGTRAP 92
#define SIGTRAP 93
#define SIGTRAP 94
#define SIGTRAP 95
#define SIGTRAP 96
#define SIGTRAP 97
#define SIGTRAP 98
#define SIGTRAP 99
#define SIGTRAP 100
#define SIGTRAP 101
#define SIGTRAP 102
#define SIGTRAP 103
#define SIGTRAP 104
#define SIGTRAP 105
#define SIGTRAP 106
#define SIGTRAP 107
#define SIGTRAP 108
#define SIGTRAP 109
#define SIGTRAP 110
#define SIGTRAP 111
#define SIGTRAP 112
#define SIGTRAP 113
#define SIGTRAP 114
#define SIGTRAP 115
#define SIGTRAP 116
#define SIGTRAP 117
#define SIGTRAP 118
#define SIGTRAP 119
#define SIGTRAP 120
#define SIGTRAP 121
#define SIGTRAP 122
#define SIGTRAP 123
#define SIGTRAP 124
#define SIGTRAP 125
#define SIGTRAP 126
#define SIGTRAP 127
```

Which is fairly common (but not ubiquitous) for multimedia capable devices.
So, he turns on the TV to see what is playing.
He really likes it, so he makes a simple gesture to capture the program so he can view it later.
Combined with custom hardware to add NFC + Accel capability
Ideally, this design could be shrunk down to the size of a standard card (actual Wifi SD card shown for comparison), and eventually integrated into the phone.
NFC extends RFID to provide a mechanism for two-way data transfer
Point of the project is to explore phone-centric novel tangible interaction techniques that can eventually be easily pushed to the mass market. The iPhone – BTW – has an accelerometer in it... but no NFC...
To these people
However – the interaction is still very limited, and dominated by interaction through the small display of the phone.
Which is fairly common (but not ubiquitous) for multimedia capable devices.
Conclusions/Summary

Extend phones with NFC + Gestures

Use simple gestures for simple commands

Towards wide-spread use of streamlined tangible interaction

Which is fairly common (but not ubiquitous) for multimedia capable devices.