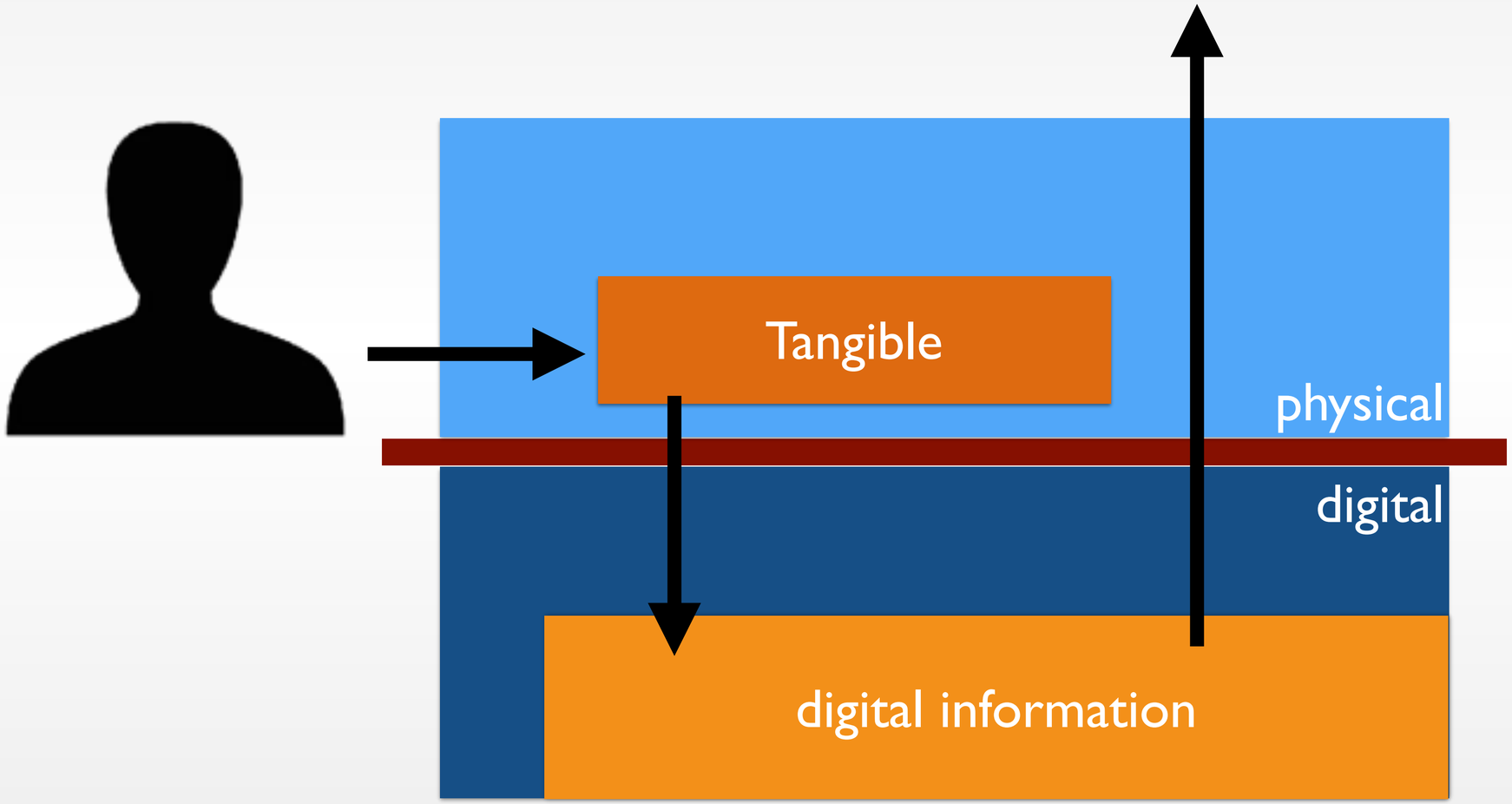
molT: Multi-Modal Tangibles to Enhance Interaction with Capacitive Touchscreens

Armin Mokhtarian

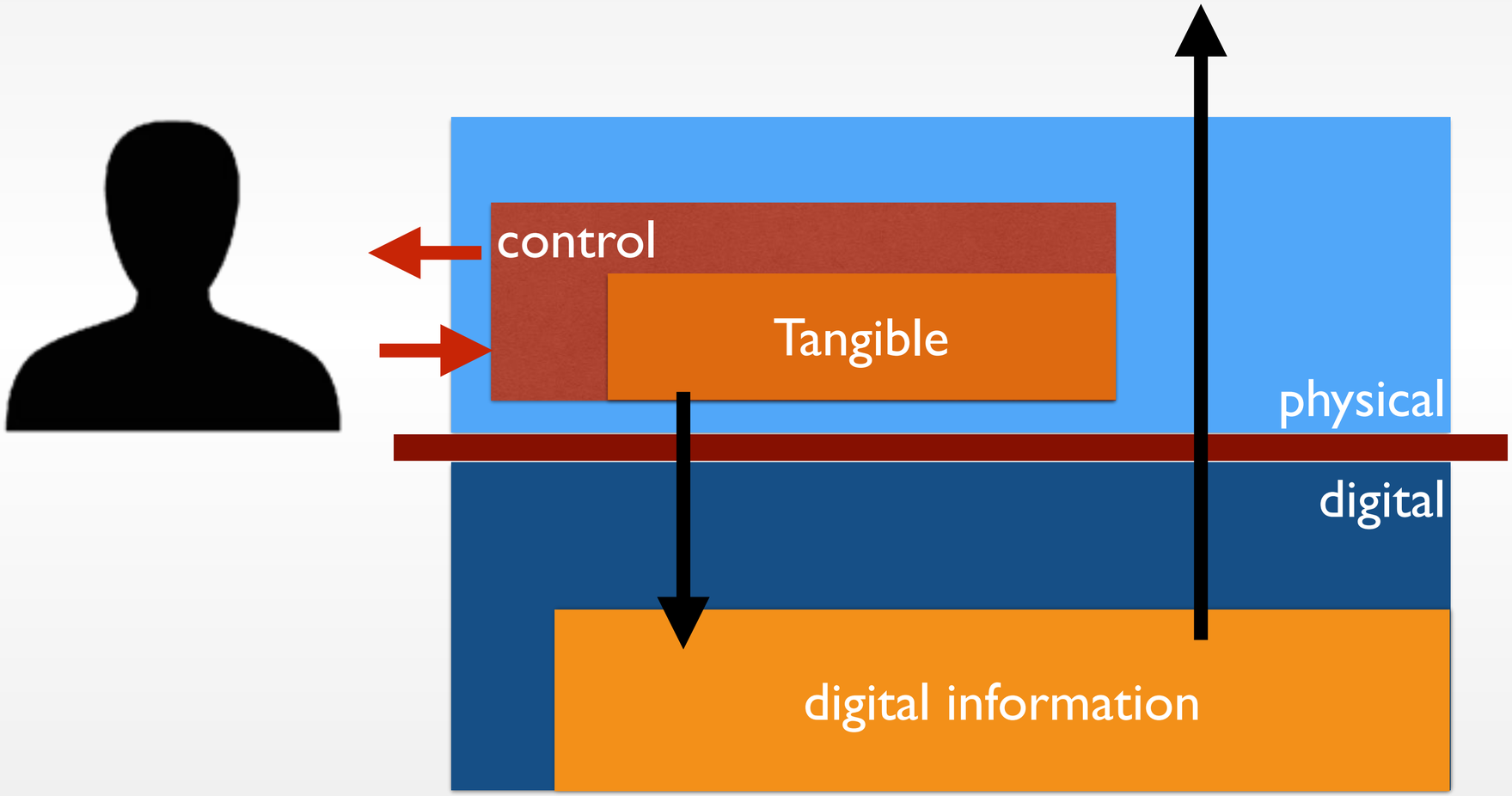
RWTH AACHEN UNIVERSITY

Tangibles

Tangible User Interface



Tangible User Interface



Multi Modal Tangible

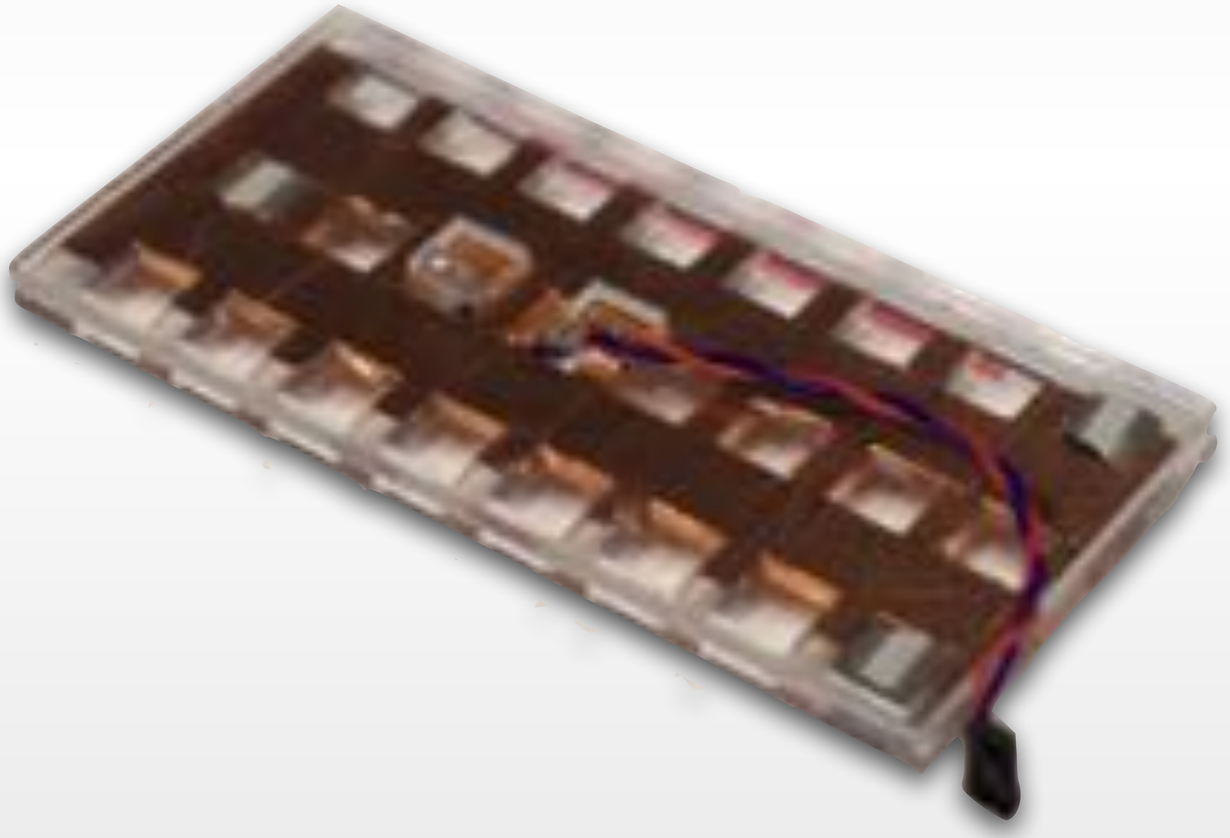
				
Input				
Output				

Multi Modal Tangible

				
Input				
Output				

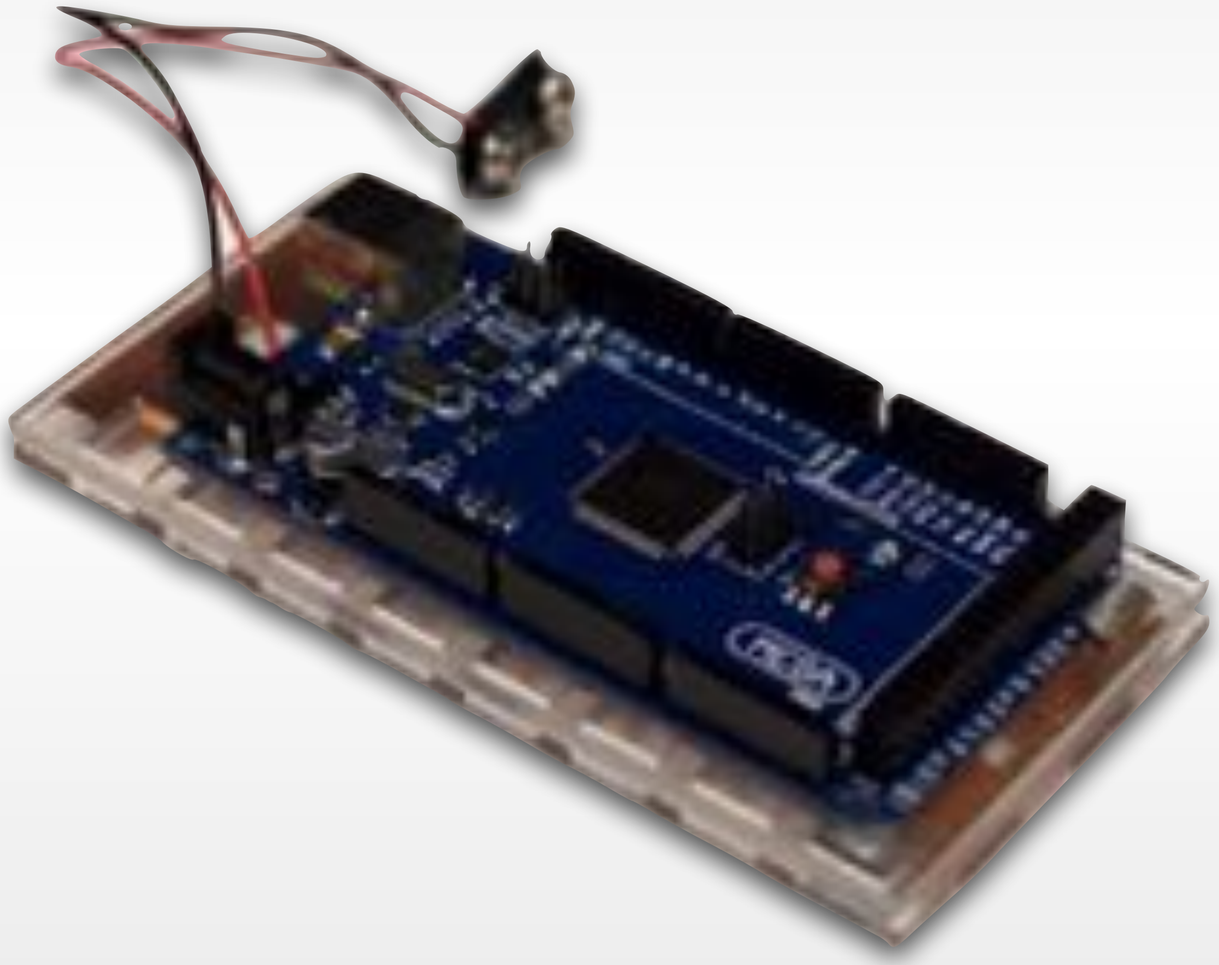
Development

- Classic PERCs ground
- Light & field sensor



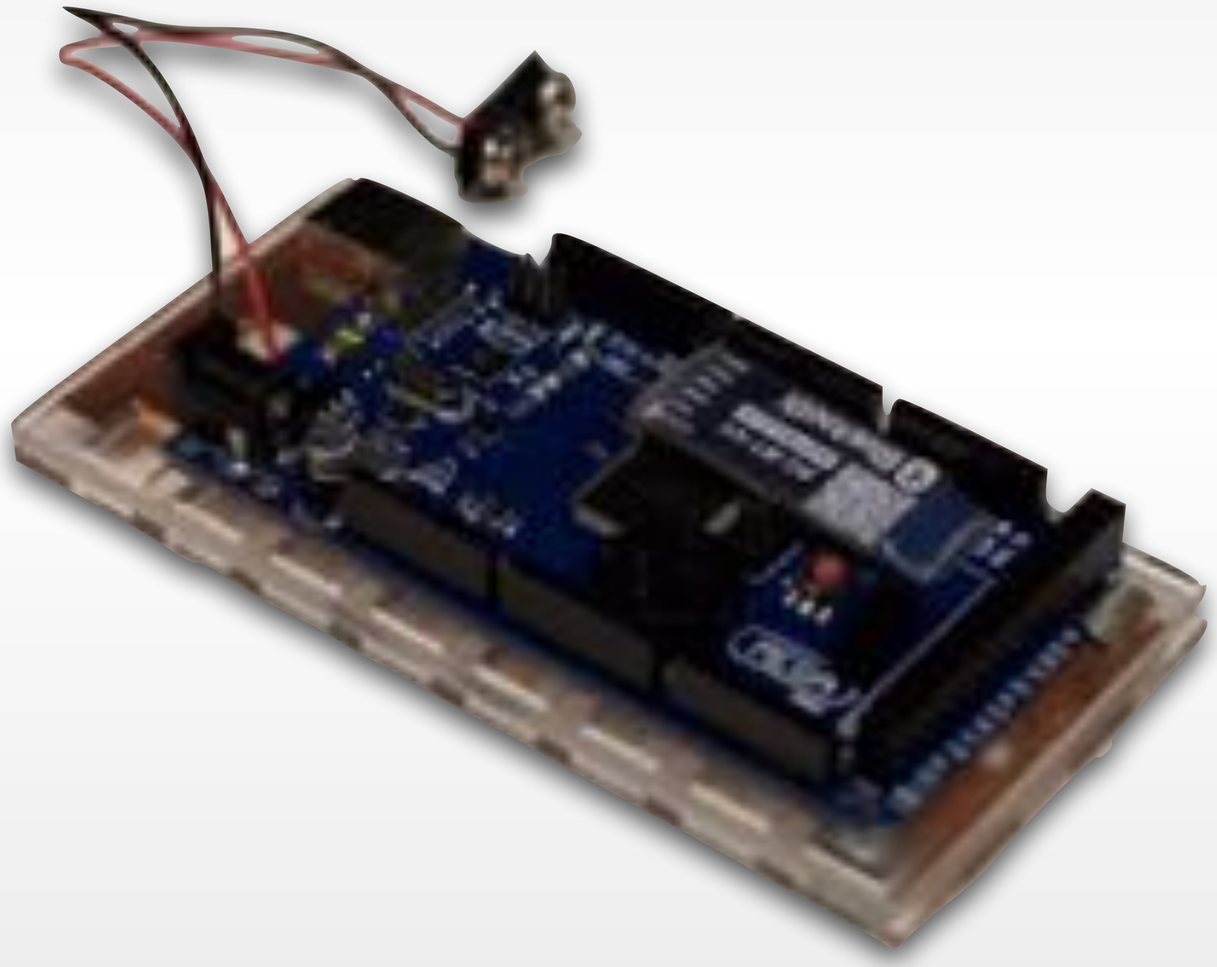
Development

- Arduino Mega 2560
- 51 pins - 256 kb Memory



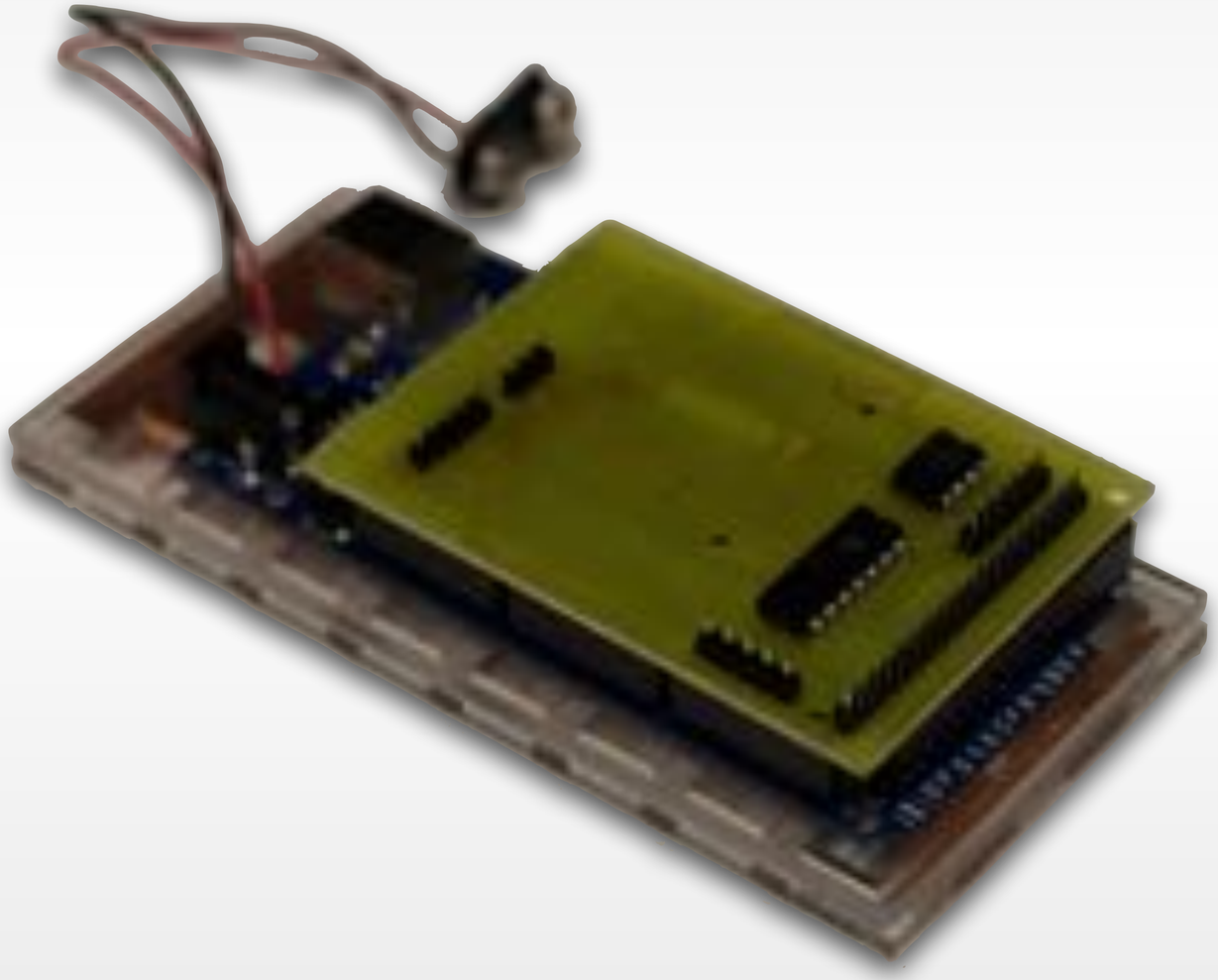
Development

- **Bluetooth LE**
- **Speaker**
- **Tiltsensor**



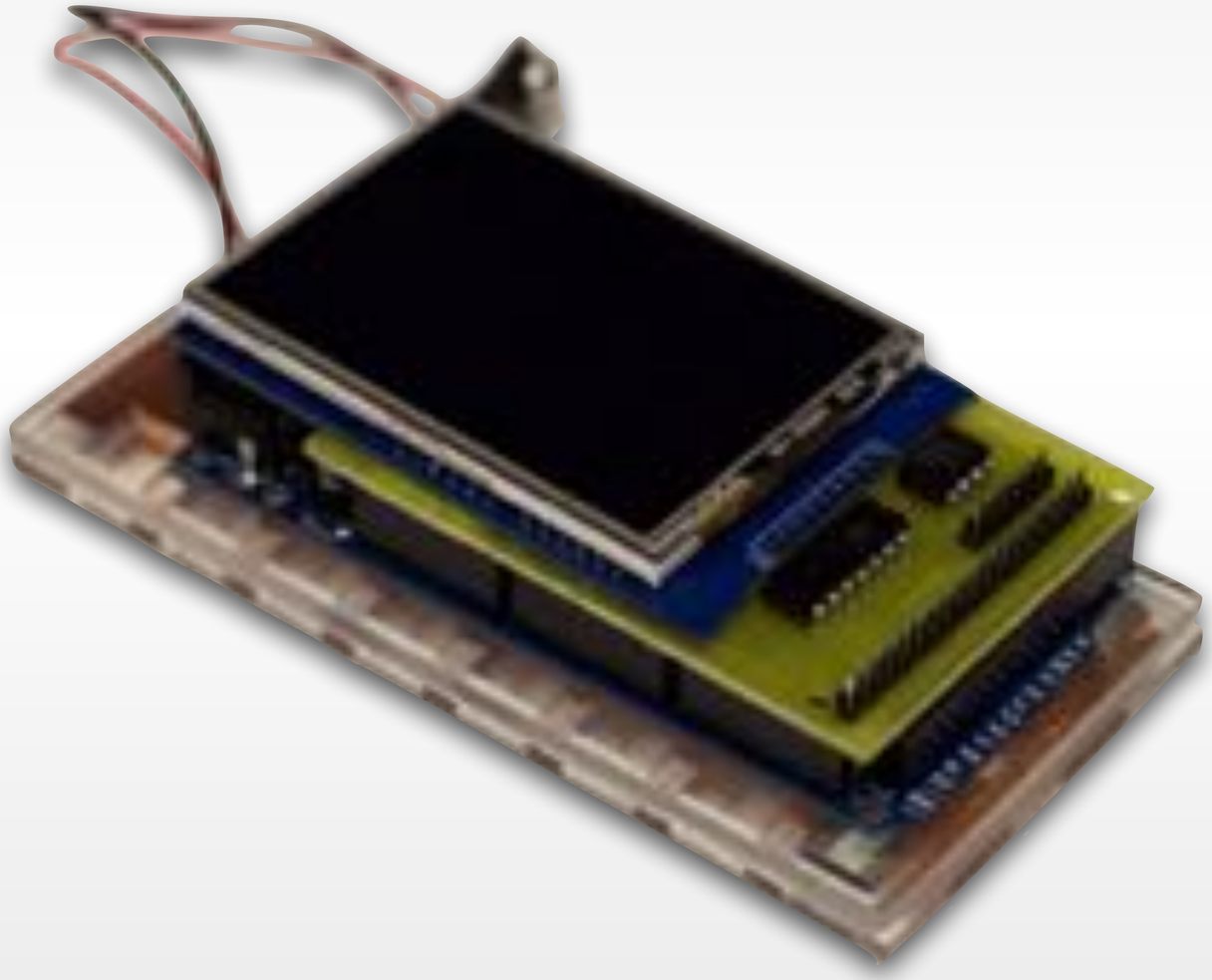
Development

- Shield
  - Own comparator



Development

- TFT 2.8” Touchscreen
  - with microSD card



Development

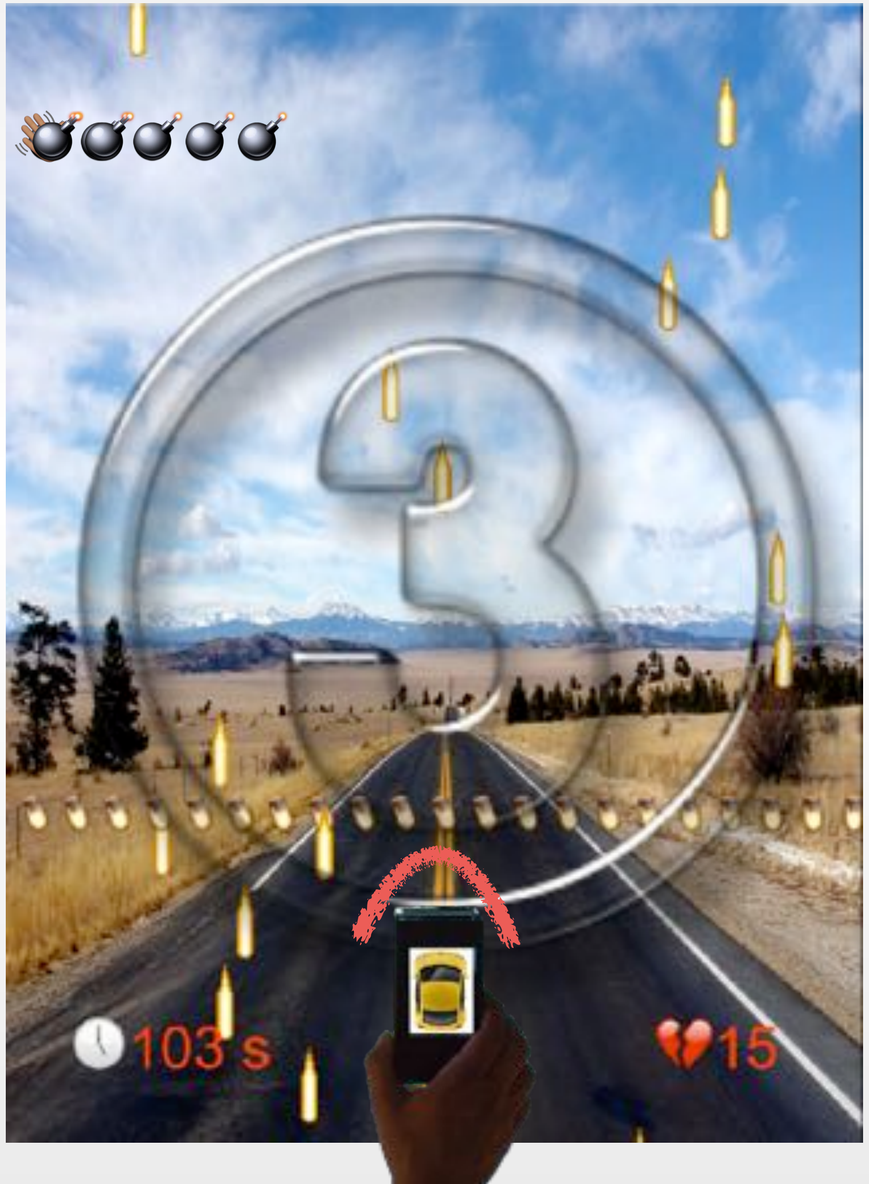
- Vibration & Button
  - connected to Case

Development

- **molT**

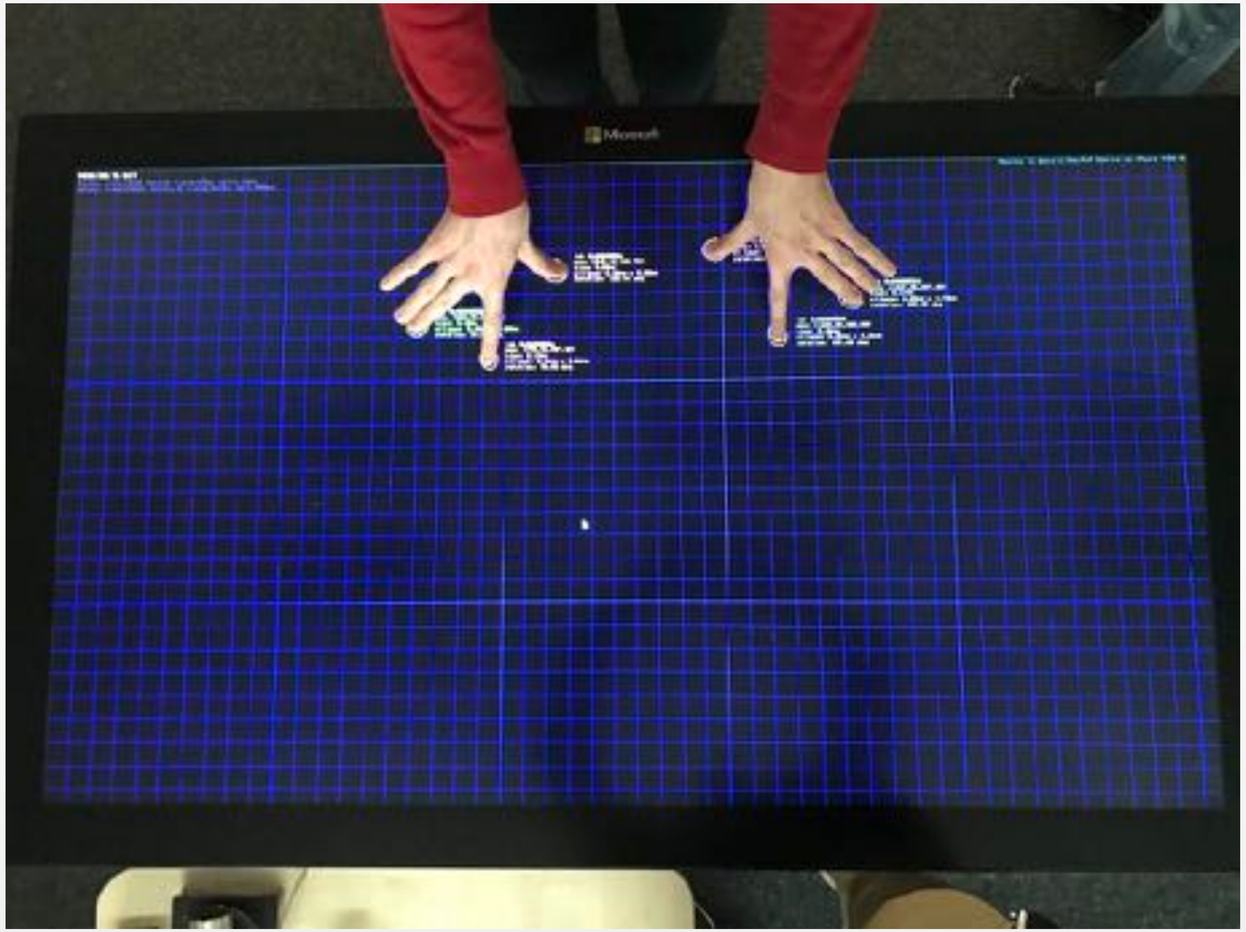
Demo Scene

- Well known game **Space Invaders**
  - Changing environment
  - Unavoidable bullets
- Requires controller with input/output



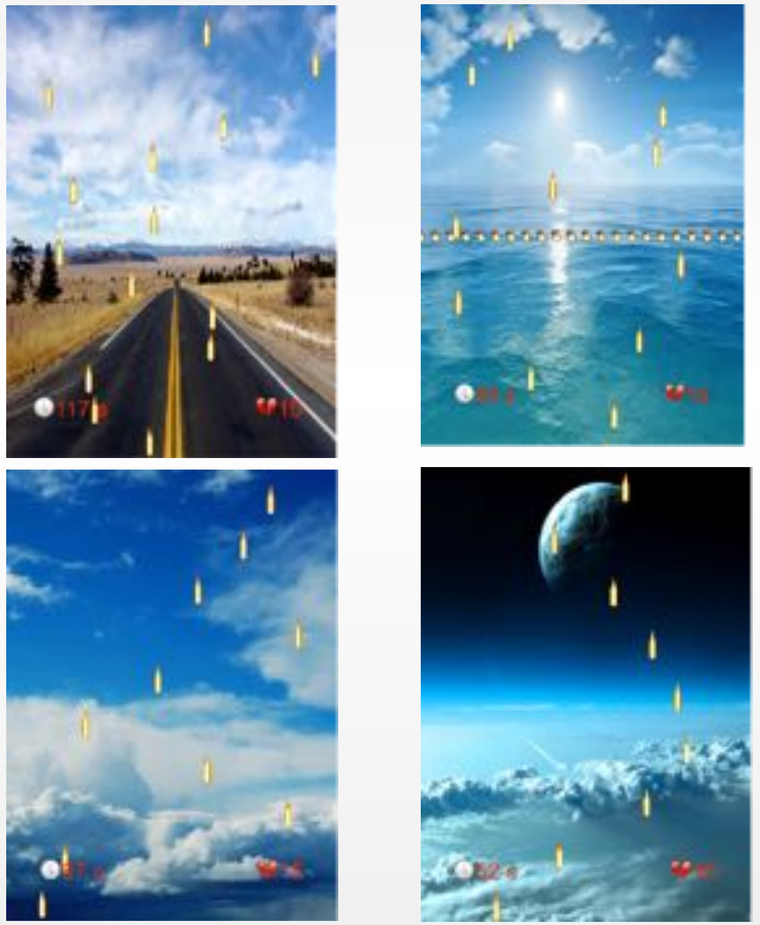
OS X - Objective C

- Touches get detected by virtual Windows machine
- MultiTouchKit makes Touchpoints usable for OS X deployment
  - developed by Rene Linden 2015
  - using Objective C



SpriteKit - RepeatActionForever

- creates parallelism
- used for
  - spawning Bullets
  - Countdown
  - Background



RepeatActionForever

- Environment changing in intervals:

```
SKAction *delayBackGround = [SKAction waitForDuration:15];
SKAction *chBackGround = [SKAction runBlock:^(void)
    {

    [self changeBackground];
    specialBulletSpeed = specialBulletSpeed - 0.2;
    BulletSpeed = BulletSpeed - 0.3;
    [self startCounter];

    }];

SKAction *sequenceBackGround = [SKAction sequence:@[delayBackGround, chBackGround]];
SKAction *repeatBackGround = [SKAction repeatActionForever:sequenceBackGround];
[self runAction:repeatBackGround];
```

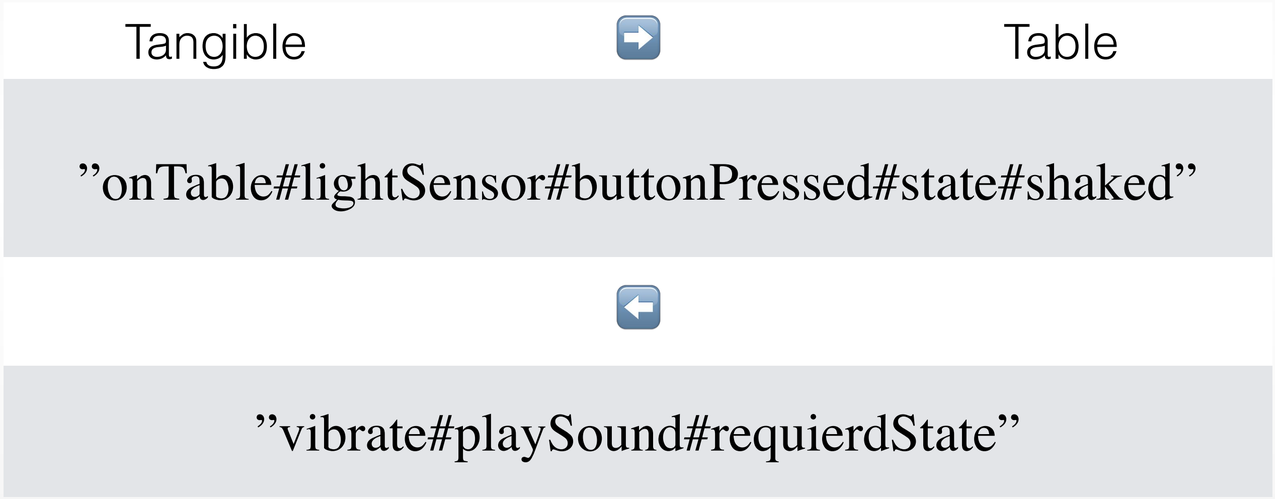
SpriteKit - Physics

- Adding physical behaviour

```
//for colision detection
_normalBullet.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:_normalBullet.frame.size];
_normalBullet.physicsBody.usesPreciseCollisionDetection = YES;
_normalBullet.physicsBody.affectedByGravity = NO;
_normalBullet.physicsBody.categoryBitMask = bulletCategory;
_normalBullet.physicsBody.mass = 2;
[self addChild:_normalBullet];
```

CoreBluetooth

- Connection set on demand
- Start up central Manager
- Search for advertising peripherals
  - peripherals broadcast some of the data they have
- Used CoreBluetooth & IOBluetooth
  - subscribing to a characteristic value



BLE Class

- Use CoreBluetooth & IOBluetooth

```
-(void) enableReadNotification:(CBPeripheral *)p
{
    CBUUID *uuid_service = [CBUUID UUIDWithString:@"FFE0"];
    CBUUID *uuid_char = [CBUUID UUIDWithString:@"FFE1"];

    [self notification:uuid_service characteristicUUID:uuid_char p:p on:YES];
}
```

```
-(void) write:(NSData *)d
{
    CBUUID *uuid_service = [CBUUID UUIDWithString:@"FFE0"];
    CBUUID *uuid_char = [CBUUID UUIDWithString:@"FFE1"];

    [self writeValue:uuid_service characteristicUUID:uuid_char p:activePeripheral data:d];
}
```

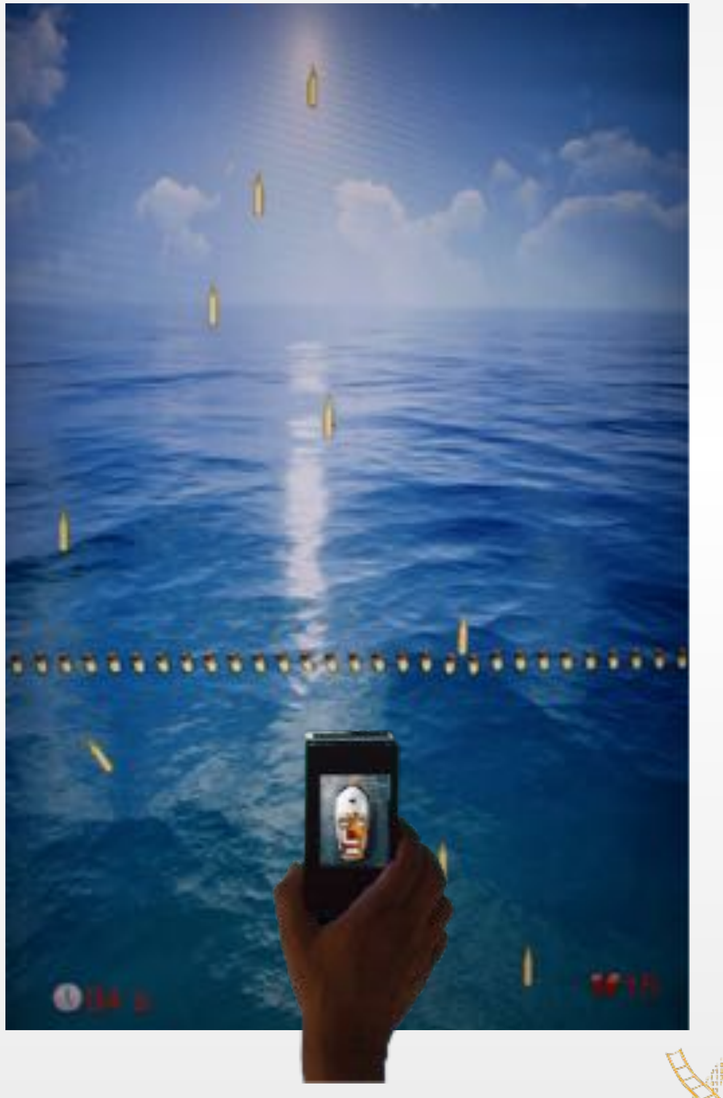
Bluetooth

- Using the predefined functions:

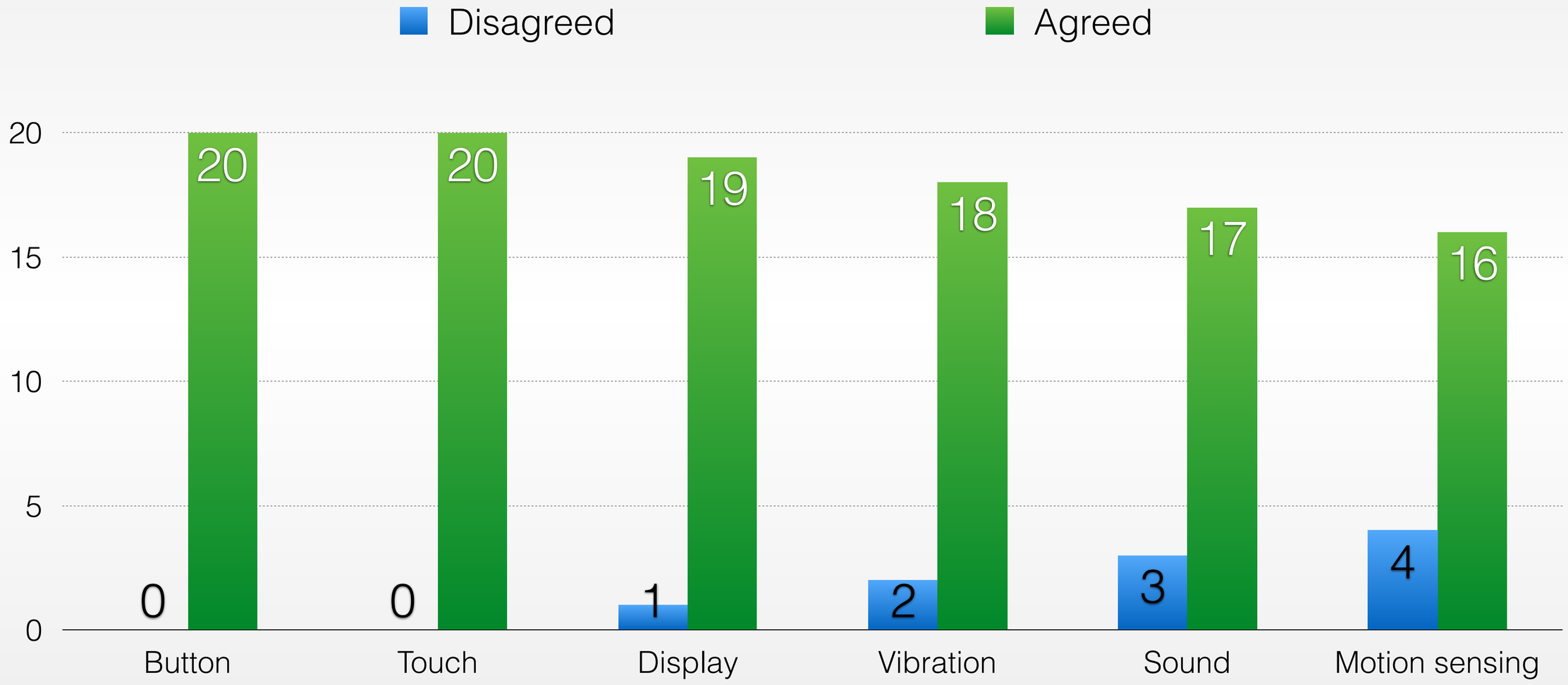
```
-(void)sendTextOut{
    NSLog(@"SEND TEXT OUT");
    UInt8 buf[20];
    NSString *str = [NSString stringWithFormat:@"%i%i%i", requierdState,vibrate, playSound];
    [str getCString:buf maxLength:20 encoding:NSUTF8StringEncoding];

    NSData *data = [[NSData alloc] initWithBytes:buf length:3];
    [_ble write:data];
}

-(void) bleDidReceiveData:(unsigned char *)data length:(int)length{
    NSLog(@"Message received with length: %d", length);
    (...)
```

Evaluation

- With user: **Model Extraction**
  - user tries to explains elements after using molT
  - user evaluates the system



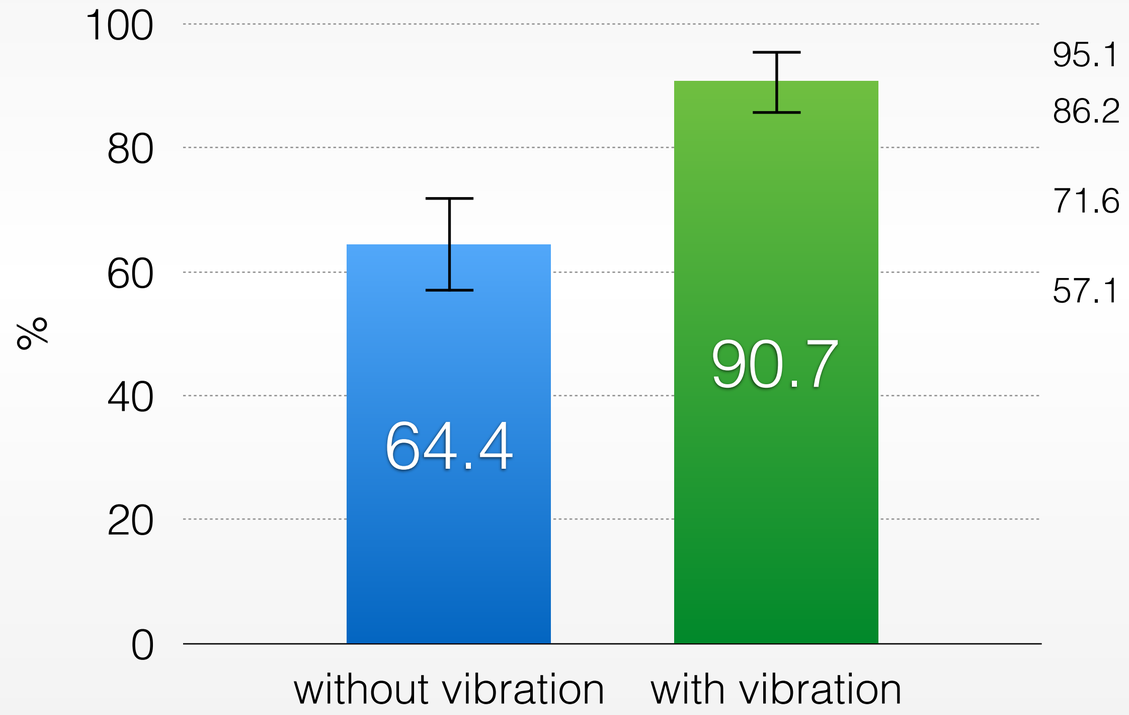
Results



RWTH AACHEN UNIVERSITY

Evaluation - Haptic Feedback

- With user: **Controlled Experiment**
  - user plays one **with** and one **without vibration**
  - plays 2 minutes
  - asked how many collisions he had



Future Work

More use cases

- other than gaming
- more suitable

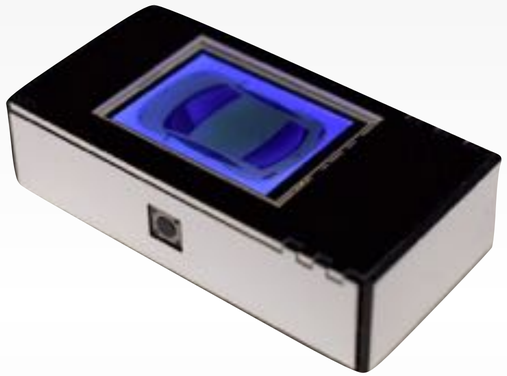
Enhanced Product

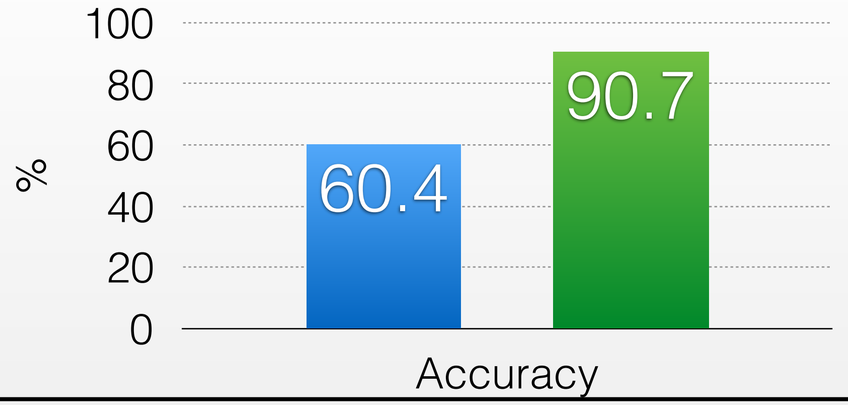
- smaller
- more powerful

More Input/Output

- Forcesensor
- mouse-wheel like input

Summary





RWTH AACHEN UNIVERSITY