Our inspiration

- Craig’s Roommate
- At a hackathon...
- The idea sprouted

An automated turret!

Kevin Casey
What we made
Let’s back track

- Goal
- Requirements
- Components
- Models
- Demo
- Next steps
Creating an automated NERF gun turret that can detect, aim, and fire at a human face.
Requirements

- A turret platform with a mounted NERF gun
  - Supports an incline angle between 0 to 30 degrees
  - Supports a horizontal field of view of 45 degrees
- The turret is able to perform facial detection
- The turret can accurately hit a target within 3 meters
- Communication errors are handled gracefully without damaging the turret.
Components

● Turret
  ○ NERF gun
  ○ Turret Stand

● Controller
  ○ MBED Microcontroller
  ○ Interface between microcontroller and the turret

● Camera targeting system

● Wireless communication
High Level Model

- Camera System
- Central Server
- Turret Stand
- NERF Gun
- MBED Microcontroller

Data Flow:
- Put Data from Camera System to Central Server
- Get Data from Central Server to MBED Microcontroller
- Control from MBED Microcontroller to Turret Stand and NERF Gun
Camera Targeting System

- **Issues**
  - Accuracy, speed, and depth

- **Intel RealSense 3D Camera**
  - Low cost (free)
  - Small form factor
  - Has depth information built in
Turret NERF Stand

- **Issues**
  - Robust
  - Achieves design requirements

1. **Stepper Motor**
   - Allows more fine tuned horizontal control

2. **DC Motor**
   - High torque to rotate gun

3. **Hardware Design**
   - Rotating stand and gun mount to achieve desired degree of freedom
NERF Gun

- **Issues**
  - Small

- **RapidStrike CS-18**
  - Relatively small
  - Cheap
  - Electronic
Controllers and Communication
Controller

● Issues
  ○ Speed, power, memory
  ○ Goal is to control turret motors and firing logic

●MBED
  ○ Good balance between speed, power, memory and cost

●Arduino
  ○ Memory quite low, extra cost and practical difficulties for no real gain
Wireless Communication

- **Issues**
  - Latency...not TOO important

- **WiFi**
  - A static target
  - Average latency ~60-100 ms
  - Implementation is simple for us

Not necessarily the optimal choice, but quickest to satisfy our requirements
Turret-Controller Interface

- 8 relays
  - H-Bridge DC motor control
    (4 relays)
  - Stepper Motor Full-drive
    (2 relays)
  - Turret triggers
    (2 relays)
  - Adafruit CC 3000 WiFi chip
Simple Process Network

- **Camera System**
  - Non-blocking Write (String - 34 Bytes) 
  - \(x \ y \ z \) timestamp

- **Central Server**
  - Non-Blocking Read (String - 34 Bytes) 
  - \(x \ y \ z \) timestamp

- **Turret**
  - Non-blocking Read (Integers) 
  - Voltage
  - Blocking Write (8 integers) 
  - Various relays

- **MBED Microcontroller**
  - Non-blocking Write (String - 13 Bytes) 
  - \(x \ y \ z \)
  - Blocking Read (String - 13 Bytes) 
  - \(x \ y \ z \)
Simple Process Network

Camera System

Non-blocking Write
(String - 34 Bytes)
x y z timestamp

Central Server

Non-Blocking Read
(String - 34 Bytes)
x y z timestamp

~100ms

Turret

Non-blocking Read
(Integers)
Voltage

 negligible

Ensure accuracy with delta, hysteresis, proper timing

Non-blocking Read
(8 integers)
Various relays

MBED Microcontroller

Non-blocking Write
(String - 13 Bytes)
x y z

~100ms

Blocking Write
(8 integers)

Blocking Read
(String - 13 Bytes)
x y z
Demo
Next Steps

- Revisit requirements
- Optimize form of communication
- Update hardware design
  - Meet updated requirements
  - Ideally be more flexible
- Downsize chip into a smaller form factor
Thanks!
Sources

- OpenCV (BSD license)
- Mbed WiFi
- Intel RealSense Documentation
High Level Model

1. Accuracy
2. Speed
3. Depth
4. Small

- Camera System
- Central Server
- MBED Microcontroller
- NERF Gun
- Turret Stand

Put Data
Get Data
High Level Model

Camera System

Central Server

MBED Microcontroller

Control

Turret Stand

NERF Gun

Put Data

Get Data

1. Latency
High Level Model

1. Robust
2. Small
3. Design Requirements
High Level Model

1. Low Cost
2. Simple (for us)