# Laser Target Hit Sensor

Senior Design Project: May 2023

Group 49
Client - Trigger Interactive
Advisor - Jaeyoun Kim

Team members:
Lincoln Khongmaly, Elijah Bryant, Akashkumar Patel, Sidney Stowe IV, Adam Runde, Neftali Medina

# Project Overview

- Trigger Interactive wanted a modified version of their live-fire targets
  - Respond to laser fire instead of live fire(bullets).
  - Ability to connect multiple targets to central module for demonstration
  - Interaction with the Trigger Interactive mobile app
  - Parts and production cost under $100

**Benefits:**

1. Laser substitute eliminates the cost of ammo
2. Provides a harmless alternative to firearms training with real bullets
3. Customers can use laser gun and targets in areas where live fire would be restricted
4. Ease of access at a gun range or within the user's home

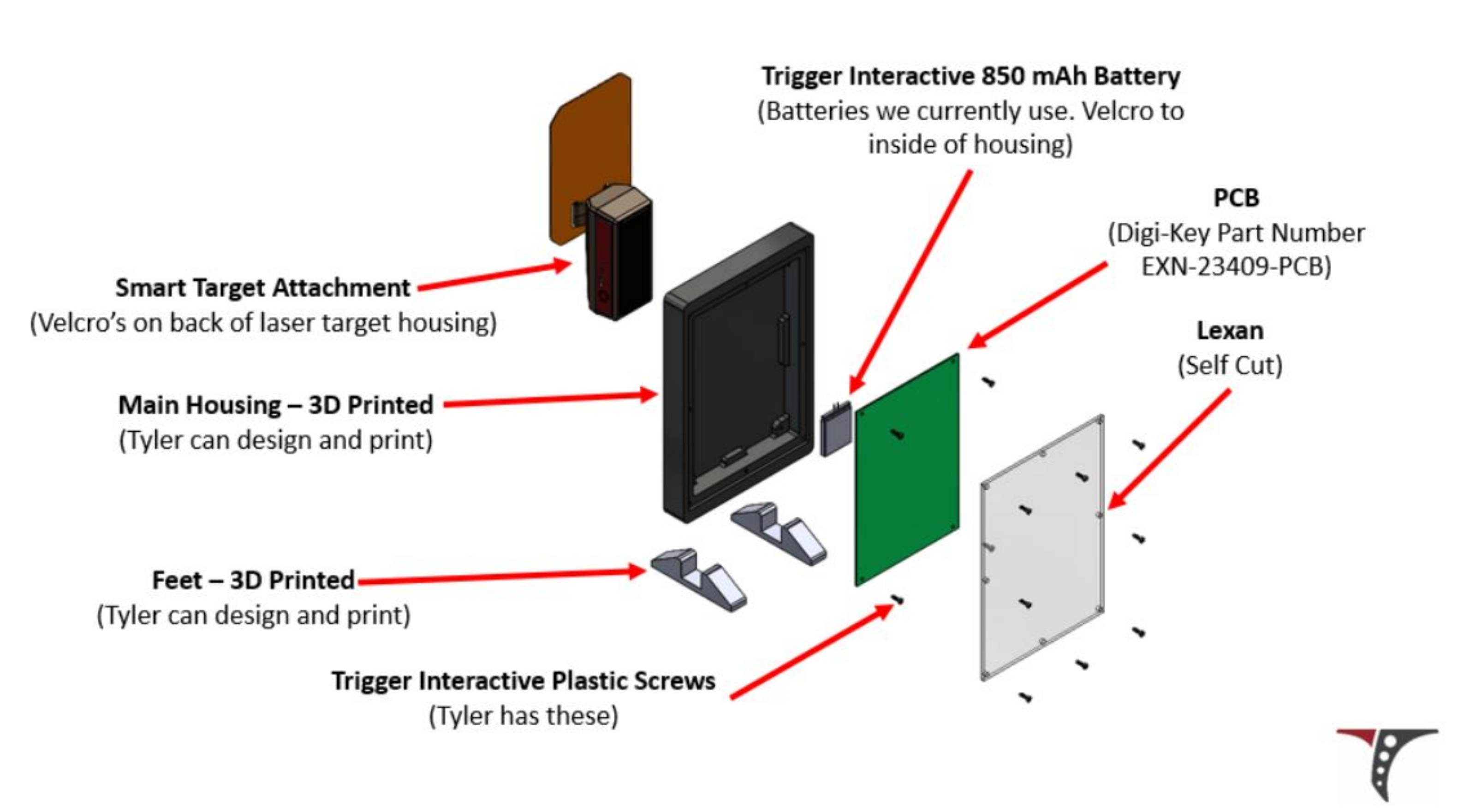
# Bill of Materials

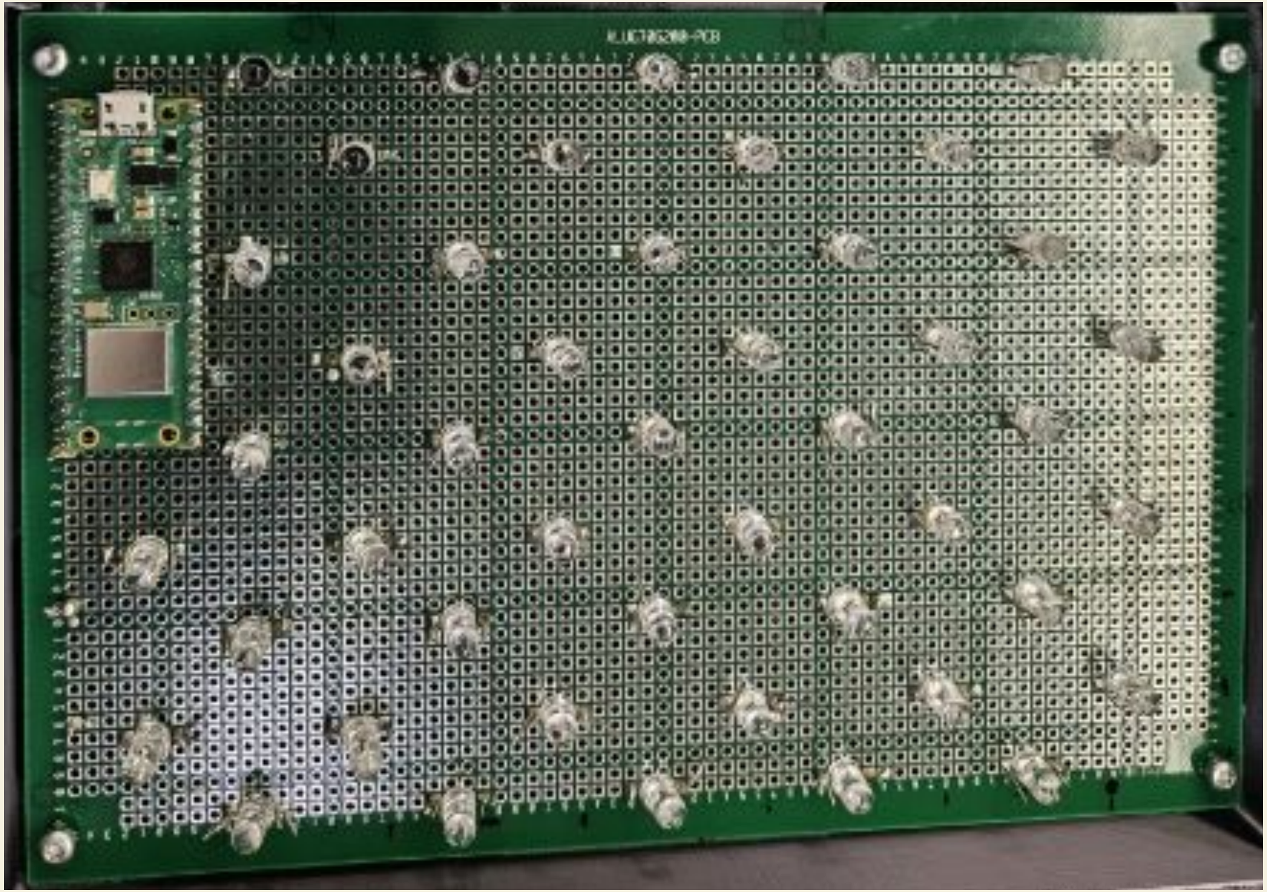
| Part Name | Cost ($) | Comments |
| --- | --- | --- |
| Raspberry Pi Pico | 4.00 | “Brain” of the module<br>Takes in data from the sensors |
| Pico shim | 8.25 | An external feature of the Pico |
| Photodiodes | 0.37/unit (47) | Accumulates data from laser fire |
| Resistors | 0.10 | Used in series with photodiodes and Pico input |
| Vibrating Motors | 1.95 | Used to inform the vibration detector of data |
| PCB | 24.7 | Houses all circuit components |
| Jumper Wires | 4.95 (75-pack) | To connect all components on the PCB |
| Battery | 5.50 | Sends power to the circuit |

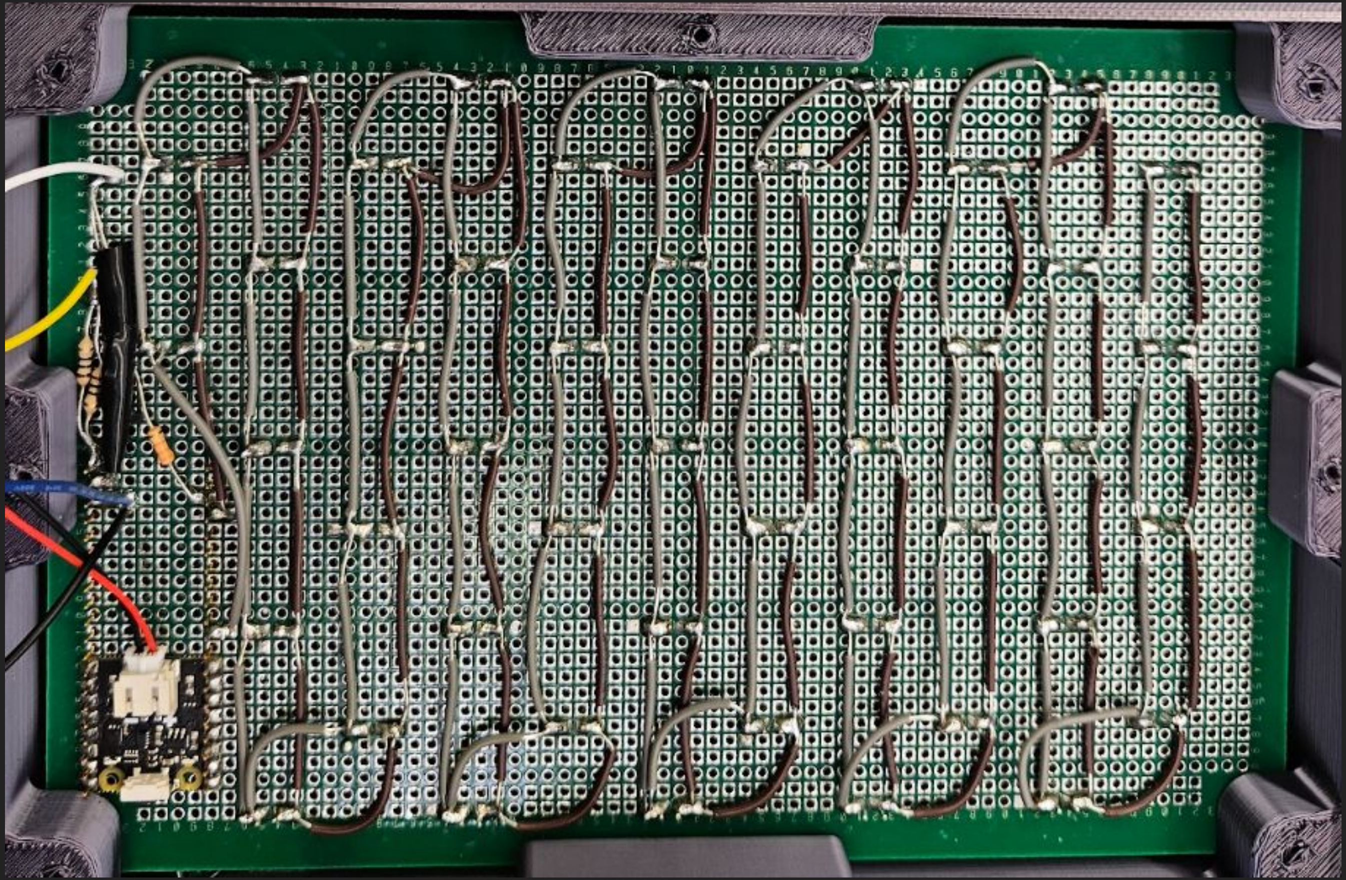
# Additional Materials

| Part Name | Comments |
|---|---|
| Acrylic Diffuser | (cost varies; avg. cost is $5 for 8x10-inch sheet)<br>Refracts light from laser for increased detection rate |
| Vibration receiver | (client’s part)<br>Reads from the motors to signal the flag |
| Flag Indicator | (client’s part)<br>Gets raised during laser detection |
| Target Housing and Stand | (client’s part)<br>Contains the PCB, diffuser, and motors inside |

# Hardware Features


Trigger Interactive 850 mAh Battery
(Batteries we currently use. Velcro to inside of housing)
PCB
(Digi-Key Part Number EXN-23409-PCB)
Lexan
(Self Cut)
Smart Target Attachment
(Velcro's on back of laser target housing)
Main Housing – 3D Printed
(Tyler can design and print)
Feet – 3D Printed
(Tyler can design and print)
Trigger Interactive Plastic Screws
(Tyler has these)





# Software Features

```python
while True:  # loop indefinitely
    sensor_value = sensor_pin.read_u16()  # read sensor value
    current_average = calculate_rolling_average(sensor_value)  # calculate rolling average
    data_line = 'sensor value: %d' % sensor_value  # create data line string
    print(data_line)  # print data line to console

    if sensor_value > current_average + alpha_rise:  # if a rising edge is detected
        if not hit_detected:  # if a hit has not already been detected
            spike_timer += 0.01  # increment the spike timer

            if spike_timer <= max_spike_duration:  # if the spike timer is within the maximum duration
                hit_detected = True  # a hit has been detected
                motor1_pin.on()
#                 motor2_pin.on()
                print('Hit')
                time.sleep(0.5)
        else:  # if a hit has already been detected
            if sensor_value <= current_average + alpha_fall:
                hit_detected = False  # reset the hit detection flag
                motor1_pin.off()
#                 motor2_pin.off()
                spike_timer = 0  # reset the spike timer

    else:  # if no rising edge is detected
         if hit_detected and sensor_value <= current_average + alpha_fall:  # if a falling edge is detected after a hit
            hit_detected = False  # reset the hit detection flag
            motor1_pin.off()
#             motor2_pin.off()
            spike_timer = 0  # reset the spike timer

    time.sleep(0.01)  # wait for 0.01 seconds before the next iteration
```

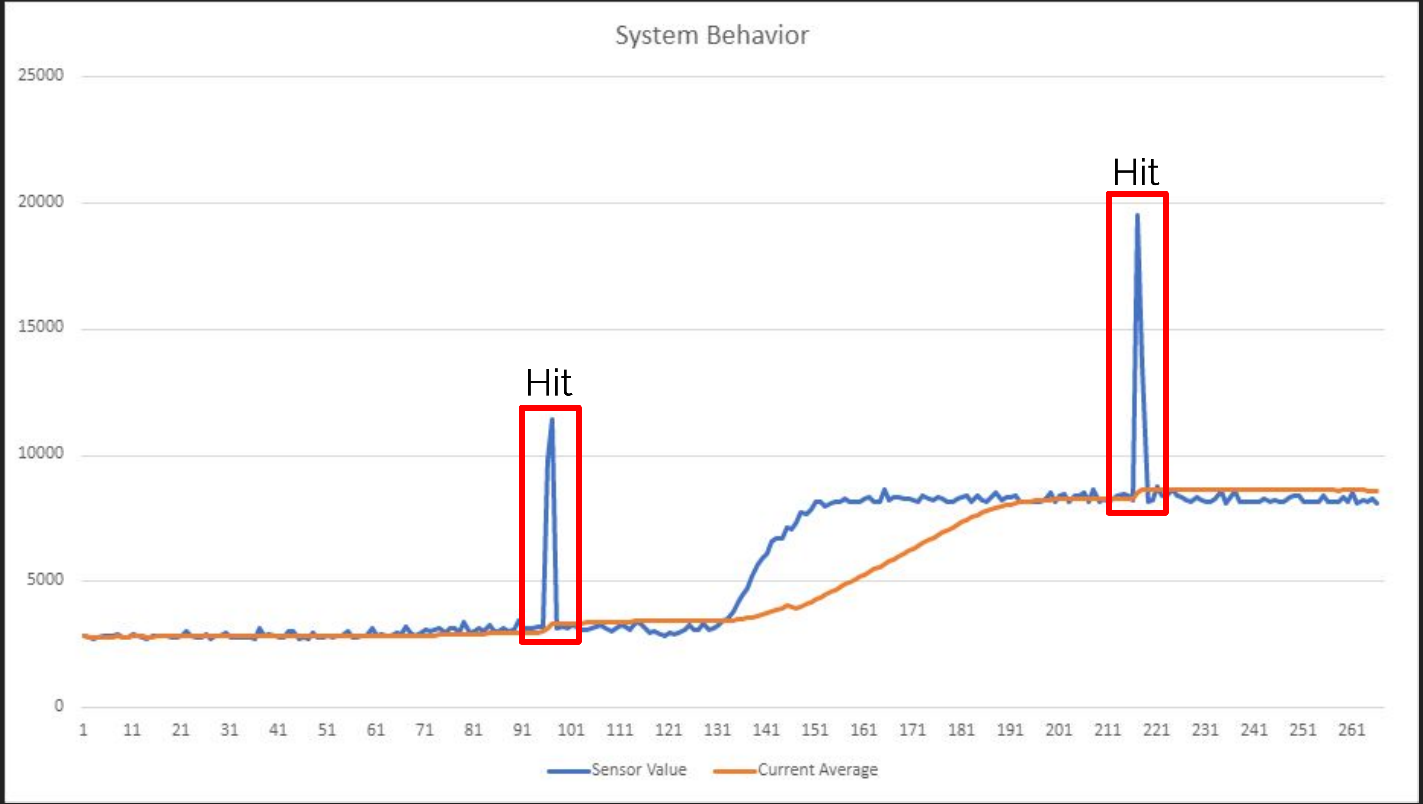
Main Code:

- All code under the while loop
- Runs whenever the Pico is powered on
- Is able to detect the difference between spikes (hits) or the change of ambient lighting

Rolling Average Algorithm:

- Used to find the light intensity in the user's current setting to establish it as the norm.

```python
def calculate_rolling_average(new_value):  # define function to calculate the rolling average
    global count, cursor, data, rolling_average  # use global variables
    rolling_average *= count  # multiply rolling average by number of data points
    if count < data_set_size:  # if the data set is not full yet
        count += 1  # increment the count
    else:
        rolling_average -= data[cursor]  # subtract the oldest value from the rolling average
    data[cursor] = new_value  # add the new value to the data set
    rolling_average += new_value  # add the new value to the rolling average
    cursor = (cursor + 1) % data_set_size  # update the cursor position
    if count > 0:  # if there are data points in the data set
        rolling_average /= count  # divide rolling average by number of data points
    return rolling_average  # return the rolling average value
```


System Behavior
25000
20000
15000
10000
5000
0
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231
241
251
261
Hit
Hit
Sensor Value
Current Average

# Accomplishments and Milestones

1. Transliteration of Arduino code to MicroPython for the Pico
2. Increased laser detection rate
   - ➢ more concentration of photodiodes and better diffusion
3. Extensive software threshold testing to ensure that user input is detected
   - ➢ Also to avoid false detections
4. Translation of laser input into vibration that the receiver can detect
5. Interaction with the Trigger Interactive app

# Accomplishments and Milestones

6. Analysis of Raspberry Pi Pico power limitations
   - ➢ Voltage and current testing between Pico outputs and motors
   - ➢ Determination of Pico power output with many loads
7. Fully integrated a full target system with a permanent PCB board layout.
   - ➢ Soldering components and ensuring good connections
8. Total of 3 target modules assembled and soldered

# Demonstration

# Key Contributions

**Lincoln Khongmaly**

- Translated Arduino code to micropython for the Raspberry Pi
- Aided in soldering PCB components
- Helped test photodiodes and assemble the board

**Akashkumar Patel**

- Assisted with translating micropython code for Raspberry Pi
- Modified code to store necessary data to txt file for testing purposes
- Assisted with board assembly

**Elijah Bryant**

- Assisted in hardware development and component research
- Modified code to to improve system accuracy
- Assisted in circuit assembly and design

# Key Contributions

**Neftali Medina**

- Soldered the jumper wires and photodiodes onto the final prototype PCB
- Aided in troubleshooting power issues with the Pico
- Helped to test the photodiode threshold and laser detection rate

**Adam Runde**

- Contributed to hardware acquisition and layout design
- Circuit and hardware troubleshooting

**Sidney Stowe**

- Contributed to hardware design and component research
- Contributed to software implementation
- Aided in hardware troubleshooting

# Challenges and Remedies

**Challenge 1:**

Translating the Arduino code into the functioning language of the Raspberry Pi

- ❖ Determining the appropriate syntax for MicroPython
- ❖ Ensuring that all aspects of the software were unchanged in the translation

**Solutions:**

- ❖ Careful analysis of each line of code
- ❖ Testing with a trial circuit to validate functionality

# Challenges and Remedies

**Challenge 2:**

Determining how to supply enough voltage to the vibrating motors to indicate a detection on the client's receiver

- ❖ Power output limitations found on the Pico during testing
- ❖ One motor was hardly vibrating because of the low output current
- ❖ Many motors placed in parallel or in various input pins would not function at all

**Solutions:**

- ❖ Placing a voltage regulator at the output did not solve the power issue
- ❖ Opted to use one motor and remove the power LED to decrease losses
- ❖ Strapped motor directly to flag module for max vibration

# Challenges and Remedies

**Challenge 3:**

Increasing the detection rate of the module and eliminating dead zones

- ❖ Even with diffusion, the photodiodes were unable to cover the entire board
- ❖ The digital threshold could not be lowered, since shadows and minute light changes would trigger the module

**Solutions:**

- ❖ Placed more diodes on the PCB, thus increasing the detection density
- ❖ Integrated data wave analysis in code to obtain better readings

# Challenges and Remedies

**Challenge 4:**

Assembling the necessary amount of targets requested by the client.

- ❖ With only a few weeks left, the client requested that we make 3 total copies of our module
- ❖ At that time, we were still perfecting the first module

**Solutions:**

- ❖ Delegated soldering work to all team members (1 member per board)
- ❖ Checked in frequently to ensure progress was being made

# Potential Future Implementations

1. Replacing the Pico with an IC that can output more power to its pins on a consistent basis
    a. Pico output limitations: 51 mA total output draw; 16 mA per pin
    b. Greater vibration of motors as a result
2. Substitution of motor with direct software implementation
    a. Bluetooth serves as communication between Pico and app
    b. Power output of Pico is not an issue
3. Reducing amount of photodiodes needed by improving laser diffusion
4. Implementation of digital potentiometer for easy sensitivity adjustment.

# Summary

- Our client desired a safer and more cost-effective target system
  - Would respond to laser beams in lieu of live fire
  - Integration with client's existing live fire system
- We first conceptualized ideas for an initial design
  - Photodiodes as laser receivers
  - Acrylic material to diffuse
  - Arduino as the data receiver and interpreter
- Conducted testing to determine alteration to the design
  - Tested different acrylic shapes and textures
  - Developed code to accurately detect a hit during light emission

# Summary

- Altered the design in various ways
  - PCB in place of breadboard and Pico in place of Arduino for size reduction
  - Modified code and increasing photodiode count to improve detection
  - Vibrating motor used instead of bluetooth communication to app
- Troubleshooted issues of the prototype
  - Low power output of Pico
  - Dead zones on PCB
- Replicated design to develop three total targets for the client
  - All targets connected via the app to form a small target practice field