

# Fruit Ninja AR 🍉 - Arthur Micha, Ishaan Jaffer, Logan Snow

“An arcade-style game using wearable technology as controller input and haptic feedback”

ECE Areas:

- Software Systems
- Circuit Design
- Signal Processing



# Use Case

## Fruit Ninja:

- Has already been adapted to VR, but not AR
- Not everyone can afford an Oculus/VR Headset
- Fun immersive experience when fatigued from Zoom meetings
- A familiar game with intuitive controls (Over 1 Billion downloads)



To showcase our glove, we will develop a Fruit Ninja-style arcade game that utilizes the glove as its sole input mechanism.

# Why a Glove?

## Current Motion Tracking in Video Games

- Motion-Tracking remotes which are bulky and unnatural to hold
- The Xbox Kinect which requires a significant amount of space
- Virtual Reality remotes are the newest improvement, but still require the user to hold something



# Requirements - Glove

Controller for this game

- **Wireless (for mobility and usability)**
  - Battery on the glove
- **Communicates with Laptop**
  - Arduino on glove
  - Unity sends signal to arduino



# Requirements - Fruit Ninja Environment

- **Game Functionality**
  - Players cut fruits, avoid bombs
  - If fruit missed, lose 1 life
  - If bomb cut, lose game
  - As game progresses, more fruits appear
  - Multiple fruits can be sliced at once (for score combos)
  - Object Physics (arc motion)
- **3D Designs**
  - Fruits
    - 3-4 regular fruits and 1-2 special fruits for MVP
  - Bombs
  - Sword
    - What the glove will be seen as



# Technical Challenges

## 1. Game Environment Challenges:

- Translating 2D game visuals to 3D
- Redesigning the game to work in 3D space. Including building 3D assets for the game
- Concurrently calculating positions of multiple fruit in real time

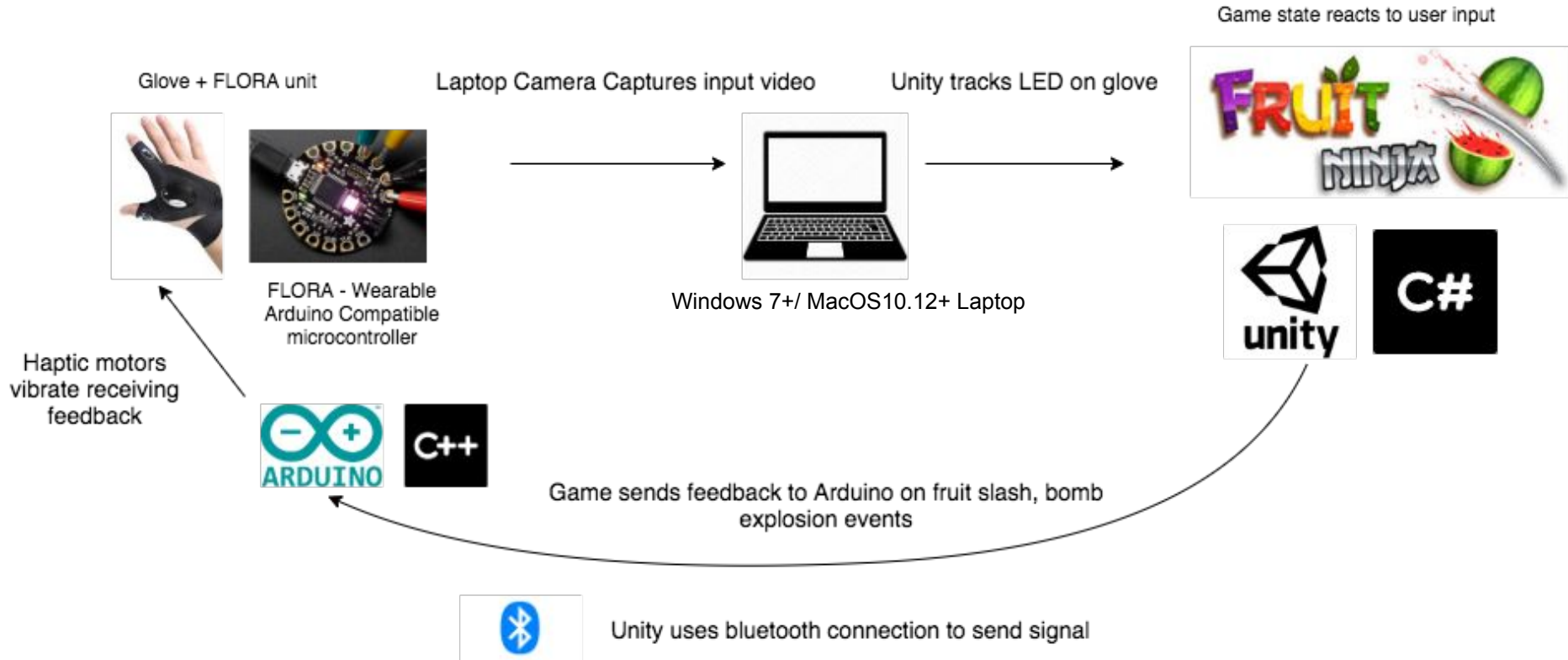
## 2. Game-Glove Communication Challenges

- Calibrating glove with the 3D game environment ensuring accuracy of user input
- Developing Unity code to wirelessly send signals to Arduino
- Minimizing latency between game state changes and haptic feedback to the glove

## 3. Glove Challenges

- Minimizing delay between receiving signal and triggering haptic motors
- Designing glove to be lightweight and work wirelessly

# Solution approach



# Solution - Software

**Frameworks/Languages - Unity, C#, C++, OpenCV, UnityPhysics, UNet, UnityUI, Unity 2D, Unity Scripting, sketchup**  
**System requirements:** Windows 7, Windows 10(x86, x64), MacOS Sierra 10.12+ (x64)

## User Interface

- Fruit 3D models - **sketchup.com**
  - Pre-MVP using existing 3D assets to focus on game development
- Menu Screen, Game Screen buttons, Fruit Splashes built with **UnityUI**

## Gameplay

- Main functions - **Spawn Fruit, Detect Collision, Collision Handler, Input Handler, MoveFruit**
- Built with **Unity 2D, 3D and Unity Scripting**

## User Interactions

- Collision/Sword Detection - Unity Box Collider in **UnityPhysics**.
  - build working fruit ninja with laptop mouse
- **Unity + OpenCV** plugin to track LED on user glove

## Glove - AR Game Communication

- Unity + Sockets Script (**UNet**) - Send Wifi signal to glove
  - Unity GameObject controls the **.NET TCP socket**
  - Arduino **Wifi shield + FLORA** to received Haptic feedback



# Solution - Hardware

## General

- FLORA Microcontroller Unit
  - Run Arduino Code
- Adafruit Perma-Proto Quarter-sized Breadboard PCB - Single
- 5x LEDs
- 2x Gloves

## Haptic Feedback

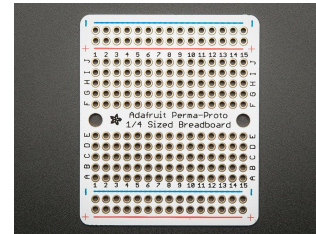
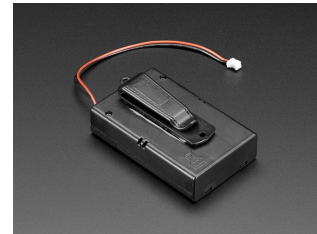
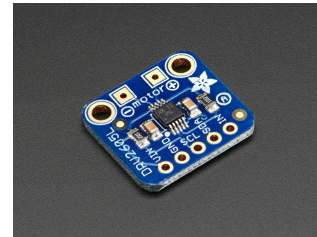
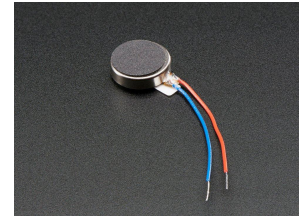
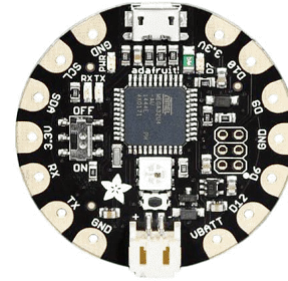
- Vibrating Mini Motor Disc
- Adafruit DRV2605L Haptic Motor Controller
- **Signal from FLORA -> Adafruit haptic motor controller -> vibrating mini motor discs**

## Power

- 3 x AA Battery Holder with On/Off Switch, JST, and Belt Clip

## Fabrication

- Velcro strips



# Testing

To test our project, the following metrics will be measured:

For the glove:

- Tracking rate (Hz) - rate at which our system can output tracking coordinates
- Latency (ms) - the delay from when motion occurs to being detected
- Precision (cm) - The resolution at which motion can be tracked

For the game:

- Collision Accuracy - when a fruit object is swiped, it should detect the collision reliably

# Division of Labor

## Arthur

- Assembling glove
  - LED and other sensors
  - Arduino code for haptic feedback communication
- Testing glove functionality
  - Test programs for sensors
- AR Game Environment
  - Object physics

## Logan

- 3D designs in Unity
  - Bombs
  - Sword
- AR Game Environment
  - Communication to Arduino for haptic feedback
  - Scoreboard
  - Game particles/effects
  - UI for selecting options in menu

## Ishaan

- 3D designs in Unity
  - Orange, Apple, Special bomb
- AR Game Environment
  - Home/Menu Screen
  - Calibrating + Tracking glove LED
  - Controlling fruits appearing on screen
  - Tracking fruit - sword interaction

# Schedule

