Team B1: Passing Partner

John Pedraza, Miles Wandless, Andrew Wong

Use-Case/Application

Product

- Launcher that autotargets and shoots a Tennis ball without manually adjusting for different pass types and positions

Problem

- Athletes that train team sports often use commercial Ball Launchers when they don't have a partner who is capable of giving them a good pass. However these are difficult to use, you need a second person with experience aiming the launcher. Our product is designed to solve this issue by auto targeting using a tennis ball. Allowing athletes to finally fully eliminate the need for a passer.

Why a Tennis Ball?

- Physical/Budget Constraints. A tennis ball is the lightest ball that we can launch that is still heavy enough to have a predictable flight path similar to that of balls used in other sports. Best ball for a proof of concept
- Tennis balls are the most widely used training balls across all sports. Despite being proof of concept, it is still widely applicable

Use Case-Requirements

- Launch a tennis ball 20+ feet
- Sense and locate target at least 25 feet away
- Ability to turn left and right from 0 to 180 degrees
- Track individual within entire range of motion in front of launcher
- Margin of error: < 3.5 ft from chest of an individual on accurate shot
- Should be 80%. Accuracy of an untrained person when throwing a ball 20+ feet

Design Requirements

Ball Launching

- Launch a 58 gram tennis ball 20 + feet
    - Exit velocity of 7.73 m/s with optimal launch angle of 45 degree
    - At least 1.73 Joules of force
    - Assuming 24 inch friction wheels need an RPM of 1453.01
    - Taking 2 inch distance of contact with the ball we would have 0.8664Nm of torque
    - Motor attached to wheel needs to output 131.83 watts
- Motor Driver needs to be able to be able to support 2 motors using 132 Watts. Support 24 Volts and 30 Amps in order to drive the DC motors
- Should be recognizable/in line with commercial launcher design. Athletes should see design and have confidence in the ball launching mechanism

Design Requirements

Launcher Base

- Base needs to be able to rotate ~12 lbs which is the weight of our launcher

Camera Requirements

- Camera must be compatible with our micro controller in order for us to draw data, and should support CV
- Camera needs at least 25 feet of depth perception as this information is necessary in order to determine launch speed required to reach target

CV Requirements

- Microcontroller capable of running CV, minimum dual-core cpu, 4GB RAM
- Camera must be able to output ~30FPS to reasonably track a person

Solution Approach

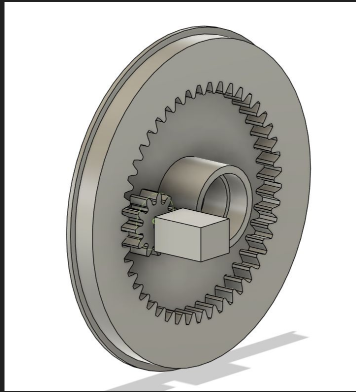
- Launching the Ball
  - Two high friction Polyurethane wheels
  - Wheels connected to two 24V ~18A DC motors with ~80% efficiency rotating in opposite directions. Each motor is capable of generating ~345 Watts, with a torque of 1.11Nm
  - Following launcher design of many current commercial launchers
  - Easier to control speed, possibly enable us to adjust launch angle and add spin later on
- Feeding ball to Launcher
  - Plan on having a stopping arm driven by a servo which will release the ball down a track into the wheels of our launcher
- Rotating Launcher (X-Axis)
  - Ring gear driven by stepper motor to rotate center of the base, which will sit on ball bearings
  - Stepper motor will drive the lid with a ~4:1 gear ratio, in conjunction with this gearing the stepper will have ~0.69Nm of torque, enough to turn the structure sitting on ball bearings

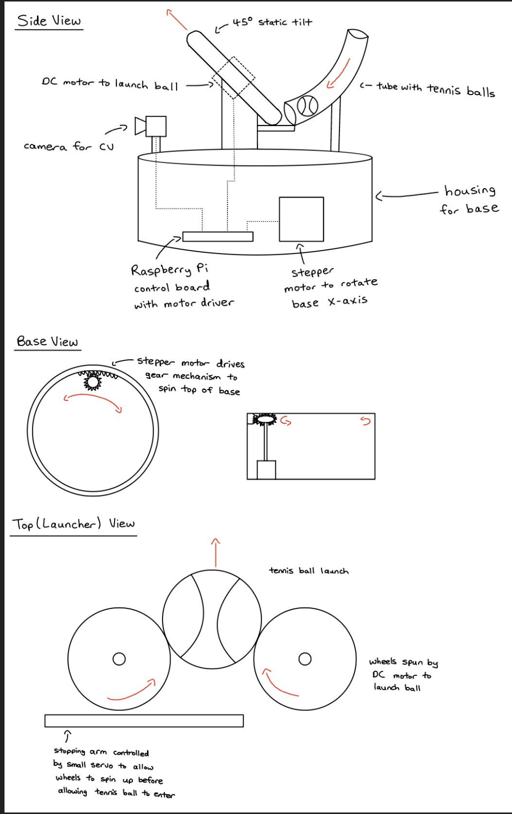
Solution Approach

- Computer Vision
  - Oak D Pro attached to the center of the base of the launcher
  - Camera has 36 feet of depth perception, compatible with Raspberry Pi, and CV support
  - Camera sends depth information, CV program identifies location of a person, RPi determines where to turn based on targets orientation with the center of the cameras view
- Projectile Motion/Targeting
  - From camera data, calculations on trajectory/speed done by Raspberry Pi
  - System will not track the person until the ball is launched
- Motor and Camera Control
  - Chose a motor driver that supports 60 Volts and 50A, which is enough to drive the 2 high power DC motors
  - Chose a Raspberry Pi 5 because we will be running CV and motor control from it, so it will be powerful enough to handle both
  - Raspberry Pi has multiple USB ports we can use to connect to the Oak D Pro and the Motor Driver

Physical Orientation and Solution Changes

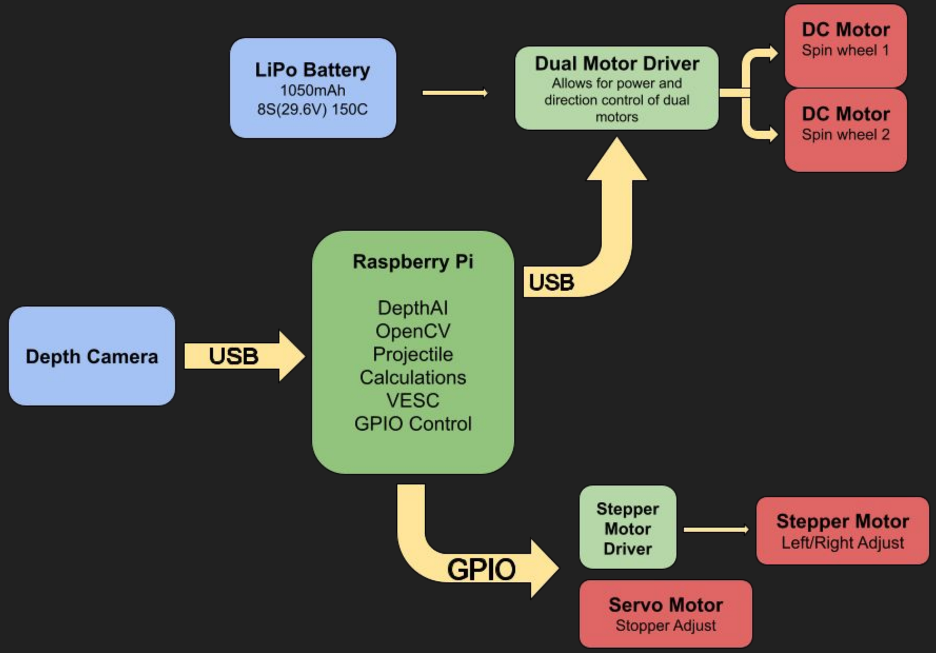
- Entire launcher will rotate along with base
- Camera will be placed in front of launcher on the lid of the base
- Raspberry Pi and motor controller located inside base
- Removed Vertical tilt
    - Unnecessary to achieve final goal
    - Having motors strong enough to tilt 12 lbs would stretch our budget





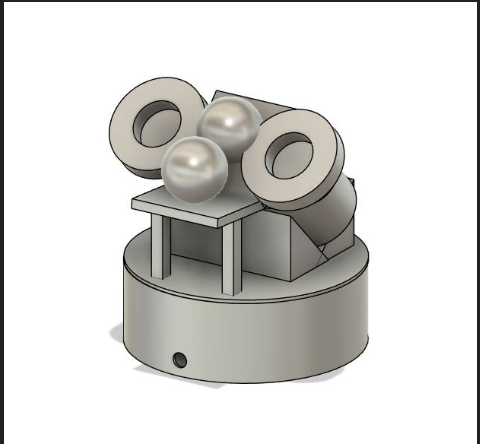
Block Diagram

- Camera provides visual information to RPi over USB
- RPi running DepthAi(depth camera library) and OpenCV
- RPi will use VESC to control DC motor speed with motor driver board over USB
- Battery will be connected directly to motor driver board
- RPi will use GPIO pins to control the stepper motor driver and servo motor



Implementation Plan

- Rotating base design took inspiration from another open source project online
- Most of the housing will be 3d printed in a few pieces and assembled
  - Likely the base will be one piece and the lid will be another
- Borrowed a RPi 5, Camera and DC Motor Driver
- We will be using pre-existing software libraries like OpenCV in addition to our targeting algorithm
- We will be buying wheels, motors, and battery



Testing

Basic Launch Test

- 20+ feet: Manually tilting launcher 45 degrees with max voltage
- Apply various voltages to motors to check max and minimum distance
- Perform this test a few times, simply want to ensure our launcher can reach proper distance
- Mitigate risk by getting motors stronger than required, have backup launching plan if fails

Rotational Test

- Add 1.5x weight of theoretical launcher weight (18lbs)
- Test rotational speed and accuracy
- Mitigate risk by getting motors stronger than necessary, can change gear ratio for more torque if failed

Computer Vision Test

- Ensure camera has a 95% accuracy rate at detecting a person within 6-25 ft range
- Ensure depth perception works between 6- 25 feet
- Redo tests above with different levels of lighting
- Mitigate risk by having camera range longer than launch range

Testing

Calculation Test

- Confirm algorithm correctness by manually moving our camera
- Successful test indicated by target being in line with the center of the camera after we follow its rotational directions
- This test will be done many times at different distances and angles from the camera
- If failed, simply adjust parameters

Integration Tests

- After combining all parts, move to different areas within receiving area. Make sure that the camera rotates the base to be facing you. Should do this many times, Draw a straight line with a tape measure from the middle of the camera. If the target is within 3.5 feet of the target then this test is a success
- Risk is mitigated by previous unit testing, if failed likely an only interconnection issue

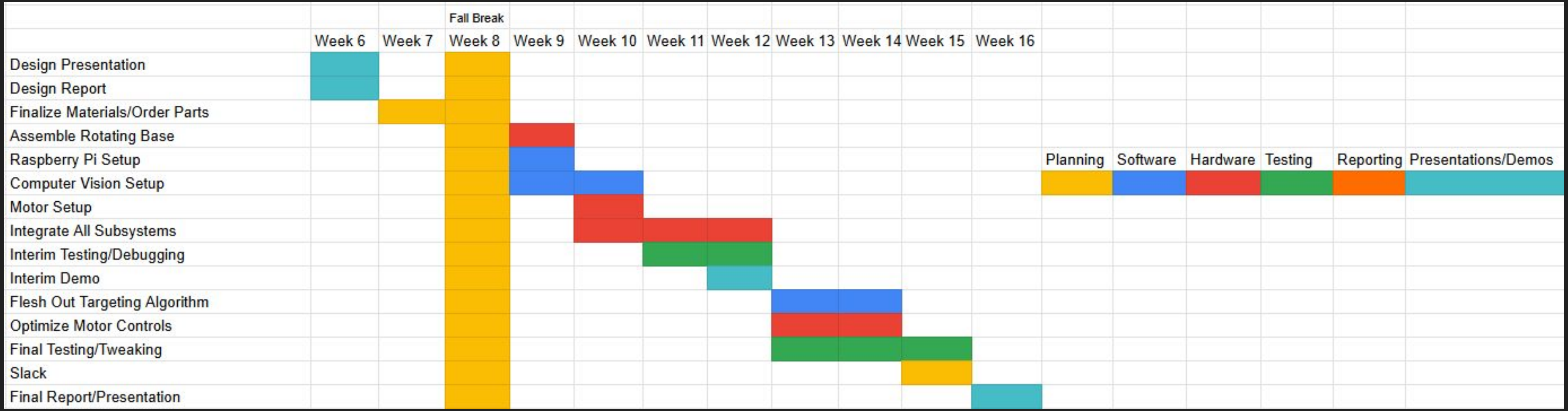
Live Test

- Launch a live ball. Mark where you are standings, as well as where the middle of your chest is. Move to catch the ball after launch and stand still
- Risk is mitigated by all previous testing

Division of Labor

John	Andrew	Miles
CV ● Person Identification/tracking ● Identifying targeting height ● Depth perception algorithm ● Identifying and determining rotational information **PI interfacing** ● Handle Camera to Pi communication/connection ● Translating targeting algorithm by andrew to be used for motor control **Structure Assembly** ● Testing and tweaking physical build of the rotational base **Testing** ● Calculation tests ● Integration tests ● Computer Vision Test	**Targeting/Physics Calculations** ● Identify RPM required to launch ball ● Determine Wattage needed for each throw **PI interfacing** ● Develop algorithm to do calculations based on CV inputs **Circuit Design** ● Build any circuit that may be required between devices **Structure Assembly** ● Determining and tweaking orientation of wheels in launcher ● Build the ball release system **Testing** ● Basic launch Test ● Calculation test	**Motor Control** ● Use motor driver board to control dual DC motors ● Use stepper motor driver to control stepper motor ● Use direct GPIO pins to control servo **PI interfacing** ● Connect Pi to main motor board via USB and other motors via GPIO ● Convert projectile calculations into motor movement and power **Structure Assembly** ● Design and print 3d model of structure **Testing** ● Rotational Test ● Integration Test ● Live Test

Gantt Chart


Fall Break
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
Week 16
Design Presentation
Design Report
Finalize Materials/Order Parts
Assemble Rotating Base
Raspberry Pi Setup
Computer Vision Setup
Motor Setup
Integrate All Subsystems
Interim Testing/Debugging
Interim Demo
Flesh Out Targeting Algorithm
Optimize Motor Controls
Final Testing/Tweaking
Slack
Final Report/Presentation
Planning
Software
Hardware
Testing
Reporting
Presentations/Demos