System design

System Requirements

Our project requirements are as follows:

Technical

- Our project must be able to transmit and receive data from bluetooth
- Our program must give real time advice at the right time.
- The user must be able to hear the advice from a bluetooth earpiece

Environmental

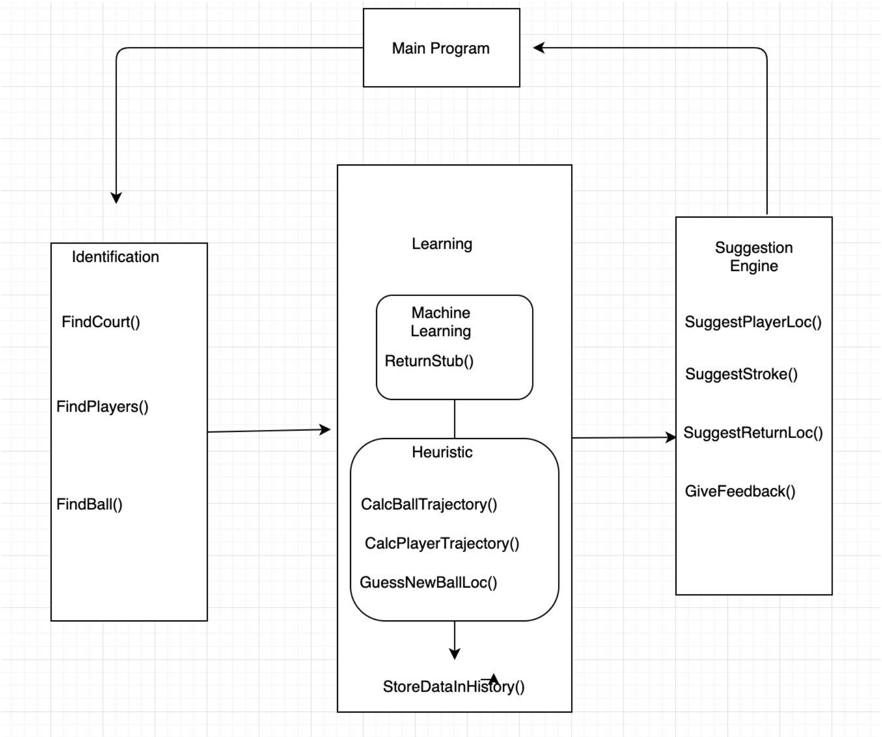
- The suggestion engine must be able to run off of a laptop set up near the court
- The video will be fed into the engine from an external webcam
- The engine will analyze live racquetball gameplay

Economic

- We need to be provided resources by the university
- We do not have a budget, so necessary equipment like video cameras and bluetooth will need to be lent to us

Functional Decomposition & System Analysis

Included below is a picture of our system diagram and the different components that make up our project.



Detailed Design

Our software design is split up between three sections as discussed above. The data flow will circle through these sections so in specifying the input/output I will discuss what each section is receiving and sending off.

Input/Output Specification

Main program (OVERALL INPUT/OUTPUT):

- Input: accepts live footage of badminton match
- Output: bluetooth audio of suggestion to players

Identification:

- Input: accepts a frame of the live footage at a time
- Output: outputs data of identified frame; where the birdie, players, and court are

Learning:

- Input: accepts all identified frame data to use for prediction (creates data model of player’s and birdie’s current and past locations and the bounds of the court)
- Output: outputs data model with predicted future trajectory of the birdie and players

Suggestion:

- Input: takes in future and current predictions of players and birdie
- Output: outputs a move suggestion which includes where a player should move to hit the birdie and where a player should aim to hit the birdie where it will be most difficult for opposing side to hit

User Interface Specification

Users will have two things provided to them; a video with the objects tagged and the suggested move for each move in the video.

Video with Identified Objects:

- A user can see a video with the identified objects and what the suggested move would have been for each move in the video after the fact and not live
- They can use this to improve their moves after the fact or also to look back on mistakes.

Audio Suggestions:

- Audio suggestions such as “Move to left, hit birdie to far right” (not final audio decision) will be provided to users after the opposing side has made a move
- These audio suggestions can be listened to via bluetooth audio

Hardware and/or Software Specifications

Users will need to setup two pieces of hardware as well as a computer running the software to use the decision engine.

Camera:

- A camera monitoring the live match at an above angle from the back of the court will need to be used by a user

Bluetooth earpiece:

- Users using the suggestion engine will need to have bluetooth earpieces to receive audio instructions

3. Testing and Evaluation plan

- Test plan (unit testing, integration testing, system testing, sample test cases)
- Simulation, modeling and/or prototyping (performance metrics, validation means, preliminary results if any)

Testing And Evaluation

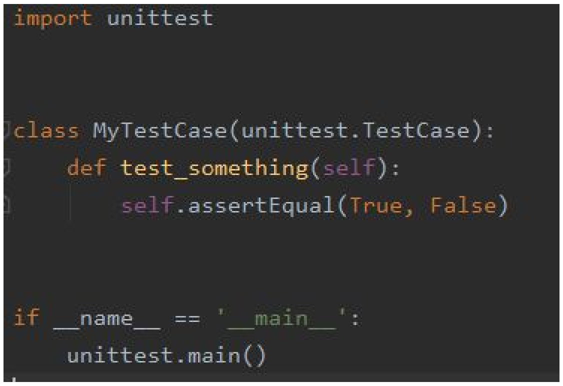
Test Plan

Our project will consist of two testing components: unit testing and integration testing. Our unit testing will cover our source classes, covering all non-trivial functionality. Our integration testing will mock a game and bluetooth controller. It will ensure that the application works when everything is connected, whereas our unit tests will test out individual components. We will use

the “unittest” framework for both unit and integration testing. This tool provides detailed information about what tests passed and failed. It also provides mocking tools.
We intend to write test cases after the main code is written. The reasoning behind this approach rather than test-driven-development is that the green state for our code will not come down to exact results. For example, we don’t know what bound to expect to capture around each player for our player identification. If we write the code first, we can find a practical tolerance for the location to set for unit tests.

Test Samples

Our first iteration is not complete yet, so we don’t currently have any tests. However, we do have templates to demonstrate the structure of our unit tests.



```
import unittest


class MyTestCase(unittest.TestCase):
    def test_something(self):
        self.assertEqual(True, False)


if __name__ == '__main__':
    unittest.main()
```

Performance Evaluation

Our application demands strict performance standards. We must be able to read the camera feed, analyze it through computer vision, determine what move to make, and then notify the player what to do via Bluetooth. We will have test cases specifically for performance once we determine our needs.

Basic Implementation

Basic Building Blocks

The basic building blocks for this project all come down to computer vision. We will be using computer vision to identify all key parts of the live badminton game. The main parts we will be identifying are the birdie, players, out of bounds lines and the stroke type of the players. We will also need to create a system for representing the location of these items. This will be used in the more advanced part of the project.

Familiarity To Platforms/Tools

Our team has experience with the majority of platforms and tools we will be using. The main tool we are using is python with OpenCV. Three out of the six-team members have used this in the past and the remaining three are in the process of learning it. One tool that will be new to all team members will be the library used to send messages to the Bluetooth earpiece. For this reason, this will be a more challenging part of the project.