Decision Support in Racket Games

Design Document

sdmay20-44
Client: Simanta Mitra
Adviser: Simanta Mitra

Team
Benjamin Kramer - Meeting Facilitator
Brian Guidarini - Test Manager
Katelyn Sinn - Schedule Manager
John Rachid - Delivery Manager
Aiden McMinimy - QA Manager
Christion Barnes - Report Manager

Table of Contents

Project and Report Requirements

Executive Summary

Will be written once the rest of the document is complete.

Project Statement

The main goal of this project is to provide real-time suggestions to badminton players to help them improve their game. The advice is given over a bluetooth earpiece as the player plays. To accomplish this, our application takes in a live feed of video, analyzes the frames, and suggests a course of action for the player.

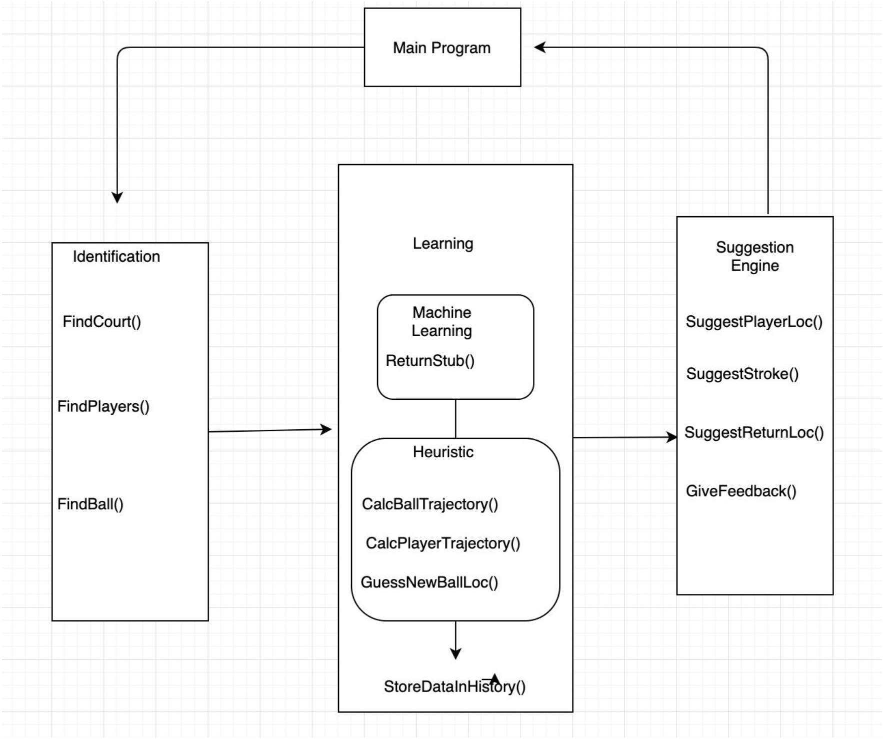
Project Goals & Deliverables

Goal	**Description**
Player tracking	Player locations are found in screen space.
Ball Tracking	The ball location is found in screen space.
Court Identification	The court boundaries are found in screen space.
Player Location	The location of the players are found in world space.
Ball Location	The location of the ball is found in world space
Court Location	A 2D representation is created of the court.
Ball Trajectory	The trajectory of the ball is calculated.
Player Trajectory	The trajectory of the players is found.
Player Location Suggestion	An ideal location is suggested for the player.
Stroke Suggestion	A stroke type (back hand, overhead, underarm, etc.) is suggested for the player.
Return Location Suggestion	An ideal place to hit the ball to is suggested.
Give feedback	The application suggests ways the player can improve.

Deliverable	**Description**
Video Analysis	Given a video, our application can give ball trajectory, player trajectory, locations and more.
Feedback from Video	Given a video of gameplay, our application can give suggestions on how to improve.
Real Time Feedback	Given a live stream of video, our application can suggest ideal player actions in real time.
Vocal Communication	Our application can give audio suggestions over a bluetooth earpiece.

Project and Report Requirements

Proposed Design/Method



Above is all of the planned aspects of our project. As of right now we have implemented the identification aspects of the plan. We are currently working on transforming this information so it can be used in the learning aspect. This will allow the two aspects to work with each other. This involves translating the video coordinates to real world coordinates that we can use to give the player a recommendation.

Design Analysis

The advantage of this solution is that the main functions are clearly separated, keeping main functionality organized. It is not hard to figure out where certain functions are just based on the functionality alone. Another strength is that because everything is so separated, it's easy for people to work on different parts of the project without interfering with anyone else's work.

One disadvantage is that piecing everything together is not always easy. Extra work has to be put in to making everything as modular as possible so that functions aren't doing the work of other parts of the program. This means the project will require quite a bit of architectural work.

Highlight the strengths, weaknesses, and your observations made on the proposed solution

Development Process

Our group uses Agile software development practices. We develop two week deliverable schedules, deciding upon what we will work on in the next two weeks. Those main tasks get broken down for each group member to work upon. We inform our client what our deliverable schedule is as well. We use two week sprints because as students it is better to have a larger window due to our busy schedules.

Design Plan

This is a design document of our project. Each module captures different parts of our overall flow for interactive suggestion. It begins with identifying aspects of the live match through video input. These objects are the court corners, the player locations, and the location of the birdie in 3D space. This data will be transformed into a 3D coordinate system and passed off to learning. Learning calculates movement of the players and birdie in a set time, calculating where the players and birdie are moving based past identified information. The Suggestion Engine then takes these locations and predicted movements to suggest a move to the players. Each module depends on the last for data in order to serve its purpose. Here's an outline of the input/output of each module:

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 Engine:

- 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

Statement of Work/Project Plan

Previous Work

Previous work and literature

Technology Considerations

The Language we chose to work in was Python. This was chosen due to:

- Easy integration with machine learning libraries
- Easy integration with computer vision libraries
- Familiarity

The alternative language we were comparing to was Java. However Java has the following drawbacks:

- Does not have state of the art machine learning libraries
- Unfamiliar with java computer vision libraries

We decided to use Open-CV for the following reasons:

- Familiarity
  - 3 members of the team have taken graduate courses with Open-CV
  - Open Source

Task Decomposition

Task/subtask decomposition with dependencies

Risks and Risk Management

Did you identify risks and risk management approaches

MileStones and evaluation Criteria

Did you identify Milestones and evaluation criteria for different tasks

Milestones

Milestone	Evaluation Criteria
Player tracking	Player is tracked with >= 90% accuracy. Each player can be detected without hard-coding locations.
Ball Tracking	Ball coordinates are tracked with an accuracy level to be decided.
Court Identification	The court is tracked consistently and accurately.
Player Location	Player location is accurately given according to the scale of the court.
Ball Location	Ball location is accurately given according to the scale of the court.
Court Location	Court location is provided accurately.
Ball Trajectory	The trajectory of the ball can be meaningfully used in the feedback engine.
Player Trajectory	The trajectory of the player can be meaningfully used in the feedback engine.
Player Location Suggestion	The location suggestions help the user.

Stroke Suggestion	The stroke suggestions help the user.
Return Location Suggestion	The return location suggestions help the user.
Give feedback	The feedback is useful to the player, and leads to improvement.

Project Tracking Procedures

How will you track progress? Process and metrics.

Outcomes & Validation

Each aspect of the project has a different set of expected outcomes and validation criteria. As shown, we separated the project into three sections. For Identification we expect the outcome to be an accurate estimation of the 3D location of the two players and birdie with respect to the badminton court. For Learning we expect the outcome to be a reasonable prediction of where the birdie will go based on the birdie's previous position and with more time a prediction of how the players will move based on previous data of specific players. For Suggestion we expect the outcome to be a recommendation of how a player should have or should react based on the Learning predictions.

Project Timeline, Resources, and Challenges

Project Schedule

Schedule, Gantt chart

We began our project on September 10. Our current progress goes up through November 3.

September 10-24: Basic Identification of Court, Players, and Birdie

- To begin, our team split into groups to tackle identifying certain objects
- They were rough estimations to begin with

September 24-October 8: Improvements and 2D Player Transformation

- While improvements to birdie identification continued, court identification combined found corners and player tracking to create a 2D bird's eye view of the player locations

October 8-22: 3D Birdie Tracking Research

- Improvements to 2D bird's eye view continued
- Research into how to know the 3D coordinates of the birdie began, meeting with others and researching online

October 22-November 5: Improvements

- Improvements to architecture of our code
- 3D transformation of birdie camera data research continued

Feasibility Assessment

Feasibility assessment

Personnel Effort

Personnel effort task by task

Other Resources

Other resources task by task

Financial Resources

Financial resources. All of the software and hardware we have decided to use is free to download or available for free because a developer already owns it or we know someone that is willing to provide it. PyCharm is the IDE we use which is free with a commercial edition as we are all Iowa State University Students. All developers own computers to write and run our project on. Cameras can just be phone camera setup using tripods. Therefore no expenses are reported for this project.

Testing and Implementation

Hardware/ Software

Hardware/tools or software used for trials. Explain utility. Further details decided by team/advisor/client.

Users will need to set up three pieces of hardware as well as a computer running the software to use the decision engine.

Back 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

Side Camera

- A camera monitoring the live match from the side will need to be set up to capture birdie location as well

Bluetooth earpiece:
Users using the suggestion engine will need to have bluetooth earpieces to receive audio instructions

Interface Specifications

Any hardware/software interfacing. This is determined by team/advisor/client. Optional if this is not a part of your project work, but should be obvious that this is the case

Functional Testing

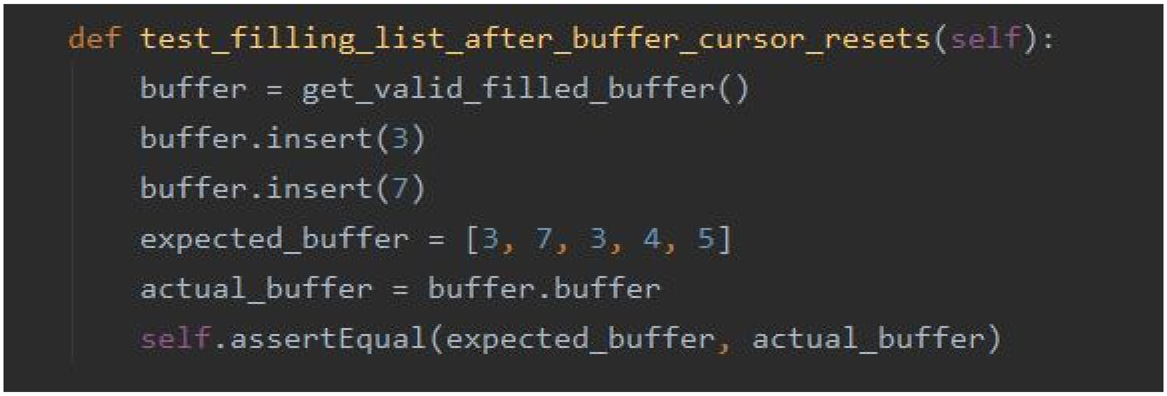
Examples include unit, integration, system, acceptance testing

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

The test shown below is what a typical test case looks like in our project. We use highly descriptive names in order to maintain clarity.



```
def test_filling_list_after_buffer_cursor_resets(self):
    buffer = get_valid_filled_buffer()
    buffer.insert(3)
    buffer.insert(7)
    expected_buffer = [3, 7, 3, 4, 5]
    actual_buffer = buffer.buffer
    self.assertEqual(expected_buffer, actual_buffer)
```

Non-functional Testing

Example include testing for performance, security, usability, compatibility

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.

Process

- Functional
  - Create a test class for each real class
  - Write enough tests to reach full functionality
- Non-functional
  - Measure performance and assert it falls within an acceptable range.

Results

Testing results

All unit tests currently pass.

Other Requirements

Nature of Content

Document explains the design specifications well and is full of visuals

Technical Approach

Ideas presented represent valid design specifications that the team is working on

Process Details

Specification distinguishes between design details for present project version and later stages of project

Design Testing/ Implementation

The document clearly describes the progress done by the team and the approach they are following

Conclusions

Summarizes functionality

References