8-Ball Lifeguard

So you don't drown at the pool table

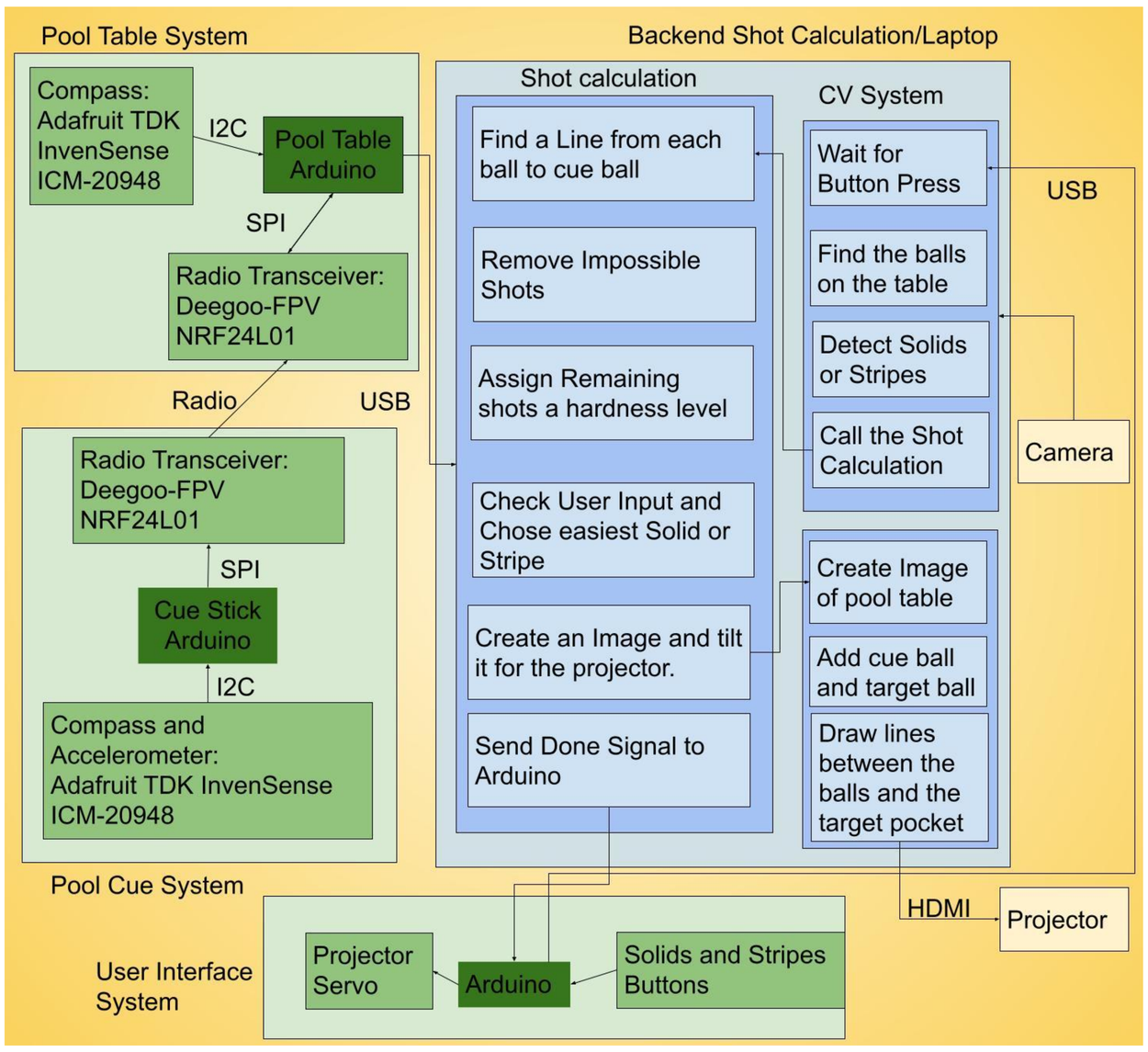
Team C7: Devank Agarwal, Jimmy Ray, Justin Rager
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

Our project serves as a way for beginners to practice and learn 8-ball pool. We project a shot for them to make, and provide feedback once they have made their shot. We want to give a **correct shot (<2° margin of error)** in an **efficient manner (<3 seconds)** with **accurate feedback (<0.5 m/s^2)**.

8-Ball pool is a hard game to learn. Additionally, in most places, coaching is expensive and inaccessible. The idea behind our project is to show people **what the best shot** for either a solid or a stripe is at any given game state, **help them make this shot**, and **give feedback** about their shot. Our best shot algorithm calculates the simplest shots to make and projects one onto the table. We do this to make it as simple as possible for users to pocket a ball, no matter what the game state is. We then take our collected data and project it onto the table as feedback for the user.

System Architecture

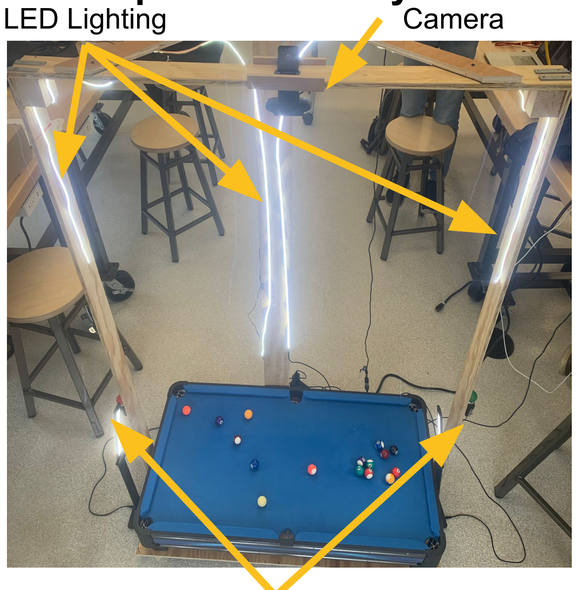


System Description

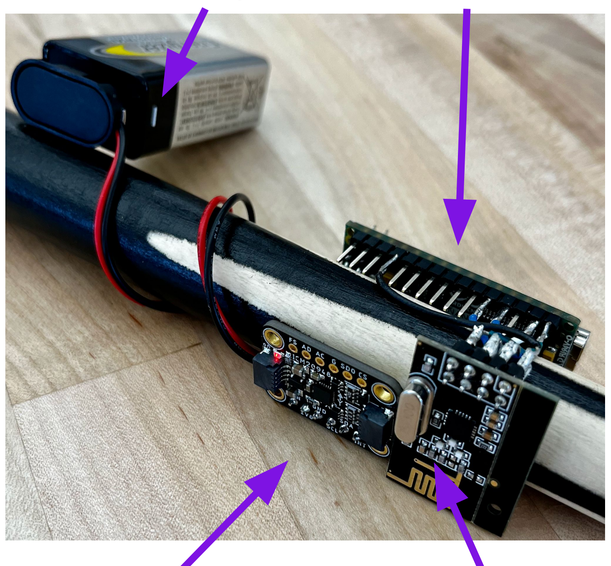
Our system has 5 main Systems:

- Computer Vision System
  - This will take a picture of a the pool table after the user presses a button to signal Solids or Stripes. It will then find the balls by looking for circular objects and calculate which balls are solids or stripes.
- Backend Shot Calculation
  - It will take in the positions of all of the balls and the target (Solids or Stripes) and calculate all possible shots. It then will calculate the easiest shot from the possible shots and send that shot to the projector system.
- Projector System
  - This will take in the best possible shot, create an image of all of the balls on the table, and then draw lines of the best possible shot so the user can follow the lines for their shot. It also will take in feedback for the user from the Pool Table System and project it onto the table.
- Pool Stick System
  - This will measure the direction the pool stick is facing as well as the acceleration of the pool stick. It will then send the data collected back to the Pool Table System
- Pool Table System
  - It takes in the raw data from the Pool Stick System and processes it so that it will determine when a shot has occurred. Once a shot has occurred it will send feedback of the how far off the angle of the user's shot versus the ideal shot was.

Computer Vision System



Pool Stick System



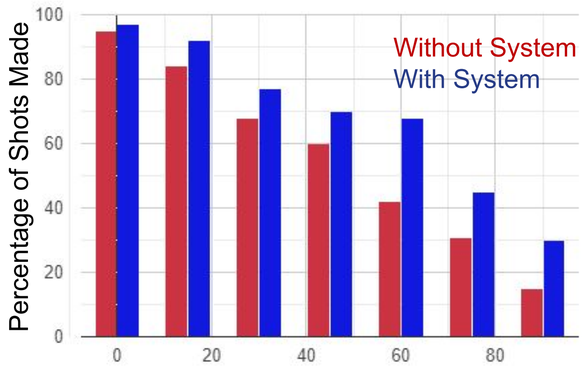
System Evaluation and Testing

We had two different testing methods:

1. System Performance Test Suite.
Over here, we measured the efficiency and correctness of our systems. We ran 5 main tests, the results of which are as follows:

Test	Result
End To End Latency (Time taken from press to projection)	1.5 seconds
Angle Deviation (Actual Shot vs Projected Shot)	1.5°
Camera Capturing (Percentage of Table Accurately Captured)	Ideal: 95% Not: 65%
Feedback Latency (Time from shot to feedback)	0.75 seconds
Feedback Correctness (Angle Deviation and Speed Differences)	2° 0.4m/s^2

2. User Test Suite:
The User test suite compared the performance of users with and without our system.



Conclusions & Additional Information

Through the course of these 13 weeks, we learnt a lot by doing this project. Our team picked up **Computer Vision** as a topic even though none of us had any experience with the same and we got to a point where we had a working CV system. We did not have a lot of experience with **woodworking** and were able to build a more reliable frame than we expected. **Microcontroller Communication** and **Game Based Physics and Maths** are not topics that anyone of us were an expert in. Due to delays in deliveries and unforeseen complications to the projects there are some aspects we wanted to integrate but did not have the time to, and moving forward the ways we could improve on this project are as follows:

1. Add **more features** to shot calculation to calculate **bounce and chained shots**.
2. Run a **Machine Learning model** on game states to better classify a **best shot heuristic**.
3. **Fine Tune Sensors** and **remove unwanted noise** from the Smart Cue system.

http://course.ece.cmu.edu/~ece500/projects/s23-teamc7/

Electrical & Computer ENGINEERING
Carnegie Mellon