

HoEM Run Derby

Bing Bhakdibhumi, Aiden Magee, Vivian Wang

Department of Electrical and Computer Engineering,
Carnegie Mellon University

Abstract—HoEm Run Derby is a two-player, baseball-themed tabletop arcade game designed to promote a fun social experience through immersive game design and mechanics. The players of the game shall take the roles of a pitcher and a batter. The pitcher, through a button console, can select from a set of preprogrammed pitch types, which induce variable pitch motions on the game ball via a magnetic gantry plate underneath the game board. The batter can then swing a physical plastic bat that is wirelessly connected to the game board, which actuates a pinball-style flipper on the game board to hit the pitch. On the game board, there are holes that determine the scoring of the hit, based on which hole the ball ends up in.

Index Terms—Arcade, Baseball, Embedded System, Gantry, Magnet, Microcontroller, Wireless

I. INTRODUCTION

Despite allowing society to communicate with each other at an unprecedented scale, modern technology has also been a force in physically disconnecting people from each other, resulting in an ever self-isolating population. This is also true for modern video games and entertainment, which have become increasingly single-player and screen-bound. As a result, many games become physically disengaging and discourage social interaction, rather than promoting togetherness. To combat this disconnect, we aim to deliver a physical game that offers immersive, engaging gameplay for all.

Our project, HoEM Run Derby, is an electromechanical, baseball-themed game that incorporates engaging physical gameplay and real-life moving elements into a portable tabletop arcade. The game is designed for two players, a pitcher and a batter, and aims to deliver the nostalgic mechanical feel of vintage arcade games. However, our project also offers novel gameplay mechanics beyond that of vintage arcade games, enabled by the latest iterations of gantry, microcontroller, and sensor technologies. The game's main features are: a playing surface with a magnetic ball and gantry system, which moves the ball towards the batter in a set of selectable preprogrammed pitch movements; a physical plastic baseball bat which, through the embedded microcontroller and IMU, detects when the batter swings it in real life and communicates the information to the game board; and a game board with different scoring holes, which computes the game state and provides real-time player feedback, facilitating an immersive gameplay experience for both players.

Currently, examples of existing games with real-life moving components include electromechanical games like pinball and

Whac-A-Mole, which have both remained staples in the arcade market. Both of these games, however, are not generally designed for two players and rely on simpler mechanical components, limiting the complexity and variation in gameplay. Our goal is to deliver a game that offers the same immersive gameplay of the existing games, while introducing novel gameplay mechanics of pitch movement control and wireless swing detection, and gearing it towards two-player play, in hopes of creating a social, immersive, and engaging experience for people who play our game.

II. USE-CASE REQUIREMENTS

We have identified seven quantitative use-case requirements for the project based on our qualitative goals. They are listed as follows:

Ease of Setup: The game unit should weigh less than 25kg total and have a setup time of less than 10 minutes. This ensures that users can enjoy the game without having to endure the hassle of a tedious setup.

Pitch Variety: The pitcher should be able to choose between at least 4 different pitch movements: fastball (fast down the middle), change up (slow down the middle), slider (curve to the left), and screwball (curve to the right). This adds a layer of complexity and skill expression to the gameplay, which in turn creates a satisfying experience.

Pacing: The average duration between pitch selection, pitch actuation, and hitting should be less than 6 seconds. This ensures that the players remain engaged with the game, and also adds a layer of skill expression.

Responsiveness: The delay between the swinging of the bat and the actuation of the flipper should be less than 100ms. This ensures the immersiveness and snappy feel of the game that is incredibly important for a fun playing experience.

Repeatability and Reliability: Repeating the same exact pitch should reliably result in a pitch ending up at a point at most 5mm away from the previous one in at least 95% of cases. This ensures that the gameplay remains constant, which could otherwise affect the enjoyability of the game.

Correctness: The game should record the correct result based on in-game outcomes over 99% of the time (e.g. a swing and miss should always be marked as a strike). This ensures that the game remains relatively mistake-free, which also could affect the enjoyability of the game otherwise.

Cost: The material costs of the entire project should be less than \$600. This is the budget constraint of this project; the low cost of building also makes the game more accessible.

In all, these requirements outline the aspects of the project that are necessary to create the social, immersive, and engaging experience that is our goal.

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION



Fig. 1. System rendering of the envisioned game board.

The playing system will consist of a wireless bat system and a game board (Fig 1), which houses a gantry that carries a retractable magnet (the coupling mechanism), controls for the gantry, a playing surface with scoring pockets that are embedded with sensors, and a centralized controller that maintains game state. The game board is tilted towards the batter for viewability and to require a clean hit for the ball to overcome the slope and fall into a scoring pocket.

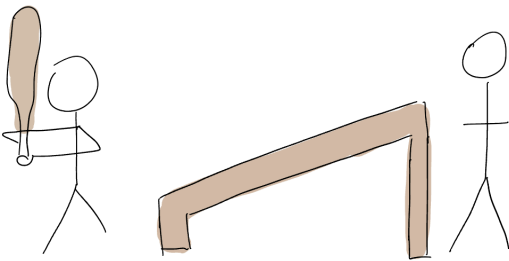


Fig. 2. Player positioning w.r.t. the board

Play begins with the pitcher (on the right of Fig. 2) selecting a pitch type by pressing one of three “Pitch Select” buttons on the side of the board facing them. Only once the pitch is selected can the pitcher select the “Pitch Now” button. Once the “Pitch Now” button is selected, the gantry below the playing surface drags the puck (ball) above the surface via a magnet. The puck is dragged in the selected path until just before the puck reaches the hittable region, at which point the gantry magnet is decoupled and the puck is now freely traveling towards the home plate.

The batter (on the left of Fig. 2) can swing their bat, and as the bat is swung, the bat system sends a signal back to the game board for the central controller to actuate the flipper. swing is sensed by the bat system and communicated back to the game board controller.

If the swing is well-timed, the flipper will contact the puck cleanly and accelerate it enough to reach a scoring pocket. Otherwise, the puck will fall back into the pocket behind the bat, for which they are rewarded no points.

Beneath the scoring pockets, there is a simplified chute system, similar to that of a pool table, which will funnel the

puck into a consistent end location. The gantry is able to slide under this region and drag the ball up a ramp to the playing surface, where it is then returned to home plate. Section VI will go into further detail explaining this procedure.

The remainder of this page is intentionally left blank.

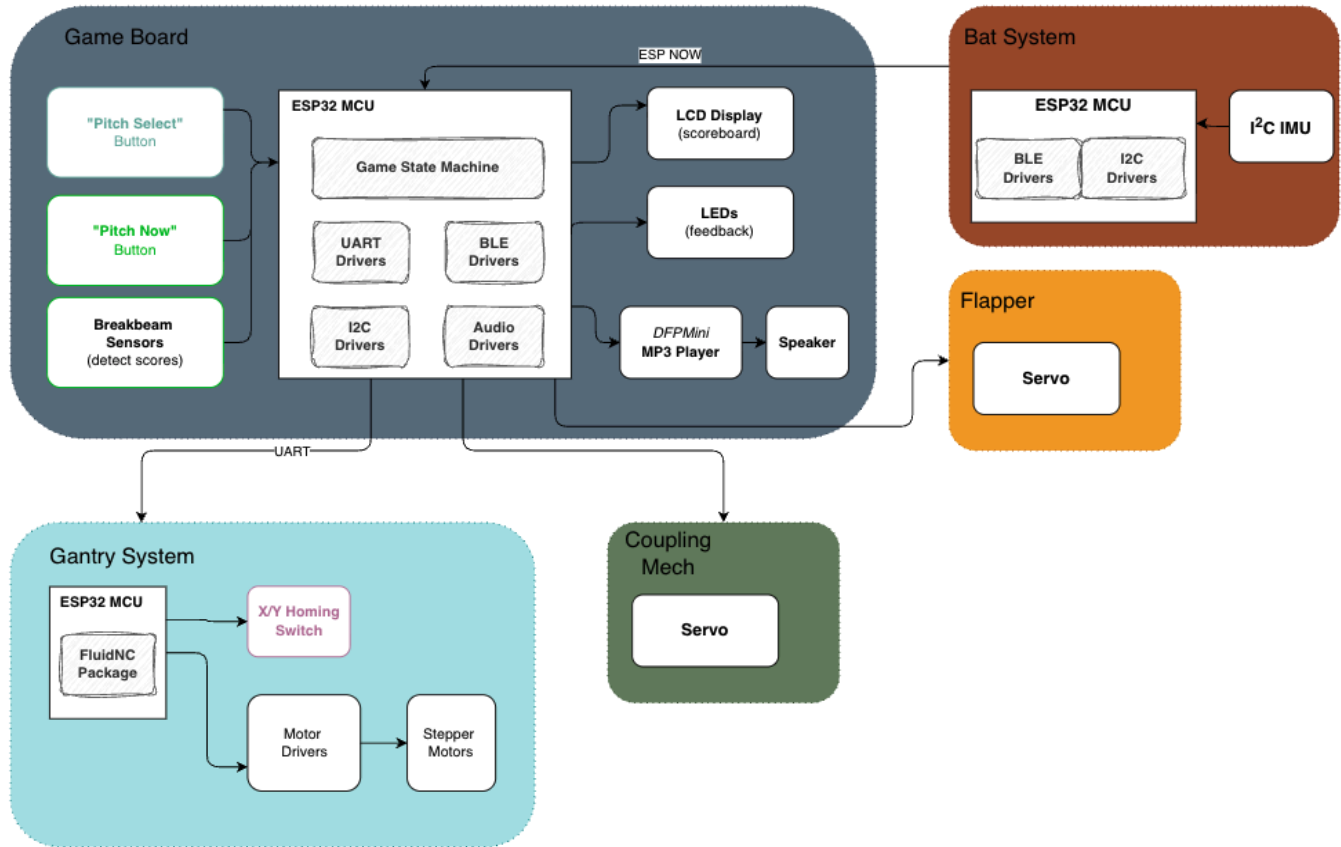


Fig. 3. The project is composed of three major and two minor subsystems: the game board, gantry, coupling mechanism, flipper (or “flapper”), and bat. Solid lines and hand-drawn lines indicate hardware and software components, respectively.

IV. DESIGN REQUIREMENTS

In conjunction with the requirements outlined in section II (Ease of Setup, Pitch Variety, Pacing, Responsiveness, Repeatability and Reliability, Correctness, and Cost), we have outlined these design requirements for the game which are crucial for the successful implementation of the project:

Gantry Motion: These are aspects of the gantry that we have identified to be crucial in order to satisfy the requirements for pitch variety, game pacing, and reliability.

First, the gantry movement should have the ability to reach a speed of at least 500mm/s and a lateral displacement of at least 150mm, in order to achieve the desired pitch movements. Additionally, the time between the start of the pitch to the decoupling of the ball should remain near constant for every repeated pitch, at most within ± 50 ms for each pitch type; the lateral position at which the gantry decouples with the ball should also be uniform throughout repeated pitches to within ± 2 mm for each pitch type.

Wireless Communication: The design specifications for the wireless communication between the bat and the game board satisfies the requirement for responsiveness.

Given the worst allowable latency between swing and flipper actuation is 100ms, we can roughly allocate the latency for each required step. First, the calculation for swing detection should take at most 20ms. The transmission of that information to the game board should take at most 50ms. The processing of that information on the game board should take at most 15ms. This gives a 15ms grace period for which the servo can start actuating movement in the flipper, maintaining latency below 100ms and allowing for smooth and responsive gameplay.

Game Board Functionality: The main functionality that concerns the game board are the scoring of plays and game state computation, which pertains to the requirement for correctness.

The sensor used in each hole should be triggered if and only a ball has fallen into that specific hole; misfires should only happen in less than 1% of cases. Once a scoring event has happened, the game state also needs to be updated accordingly. The software responsible for calculating the game

state must be accurate, correctly computing the score in at least 99% of all scoring events.

In lieu of these technical requirements, we are able to determine a clear plan forward for the implementation of the project which satisfies the preceding user-case requirements, and ultimately the goal that we have set out for our project.

V. DESIGN TRADE STUDIES

In designing our project, we encountered crossroads that required our careful judgement and thorough evaluation regarding what components and implementation techniques would best fit our vision. Design trade studies were conducted for the gantry hardware, scoring sensors, servos, and microcontrollers.

A. Gantry Hardware

We selected a belt-driven Cartesian gantry architecture to balance speed, programmability, and build complexity. Compared to CoreXY, a Cartesian configuration simplifies kinematics and debugging at the expense of some speed and precision. We deemed this an acceptable tradeoff for our MVP-oriented design, where ease of programming and predictable motion are preferred. Belt drive (GT2 timing belts and pulleys) was selected over lead screw/worm gear solutions because it supports higher linear speed and smoother motion at the target pitch speed range.

Each axis is built from common CNC components (2040 V-slot aluminum extrusion, gantry plates with V-wheels, NEMA17 stepper motors, GT2 belt, and corner brackets/T-nuts/spacers). This component set is widely used in the CNC hobby ecosystem (which is dominated by OpenBuilds), providing robust community documentation and replacement availability.

The gantry kit supports both a 20-tooth and 30-tooth pulley. The 30-tooth pulley offers increased linear travel per revolution at a reduced torque compared to the 20-tooth variant. Considering gantry speed is critical for our game to feel “snappy”, and our gantry carriage should be relatively light, we opted for the 30-tooth pulley.

B. Scoring Sensors: IR Breakbeam vs. Microswitch

We had two main candidates for detecting ball entry into scoring pockets, between IR breakbeam sensors and microswitches. The main requirements for the solution are that the sensor must achieve $\geq 99\%$ scoring accuracy, function within the funnel-lane geometry, tolerate vibration and bounce, and remain low-cost and easily replicable.

IR breakbeam sensors offer non-contact detection, eliminating mechanical wear and maintaining consistent performance over repeated trials. Although susceptible to false triggers from vibration or partial occlusion, these risks can be effectively mitigated through firmware controls such as minimum beam-break duration and state-based validation to prevent double-counting. Alignment sensitivity is moderate but manageable through rigid mounting within the lane

structure. The components are inexpensive and support modular installation without obstructing the ball path.

On the other hand, microswitches provide simple, low-cost detection but introduce reliability concerns. Contact bounce can produce multiple rapid transitions, particularly under high-speed impacts, and performance may degrade over time due to lever fatigue or deformation. While firmware debouncing can reduce false triggers, variability in impact angle and wear threatens long-term repeatability and the $\geq 99\%$ accuracy requirement. Mechanical integration also requires careful lane shaping to ensure consistent actuation without impeding ball motion.

With these considerations, we selected IR breakbeam sensors due to their superior reliability, minimal maintenance, and consistent long-term accuracy, making them better suited to meet project performance requirements.

C. Paddle Actuation: Servo Selection

Here, we evaluated servo options for the flipper mechanism that strikes the ball. To maintain responsiveness, the actuator must achieve approximately 0.05s per 60° of rotation so mechanical motion does not dominate system latency, while also providing sufficient torque to reverse and propel the ball toward target holes.

Using an impulse-based model, we calculated the average impact force given by:

$$F_{avg} \approx \frac{m_b \Delta v}{\Delta t}, \Delta v = v_{out} - v_{in}$$

And the corresponding torque about the pivot by:

$$\tau_{impact} \approx F_{avg} r = \left(\frac{m_b \Delta v}{\Delta t}\right) r$$

Using $m_b = 0.040\text{kg}$, $v_{in} = -0.45\text{m/s}$, $v_{out} = 2.0\text{m/s}$, $\Delta t = 0.005\text{s}$, and $r = 0.076\text{m}$, we obtained:

$$\tau_{impact} \approx 1.49 \text{ N} \cdot \text{m} \approx 15.2 \text{ kg} \cdot \text{cm}$$

Applying a 2x safety factor to account for flipper inertia, friction, and reduced torque at speed, we established a design target of 25–35kg·cm. After, comparing between a higher-torque ($\sim 35\text{kg}\cdot\text{cm}$), but slower servo ($\sim 0.07\text{s}/60^\circ$) and a faster high-voltage digital servo ($\sim 25\text{kg}\cdot\text{cm}$, at $0.05\text{s}/60^\circ$), we selected the faster servo to satisfy the requirement of achieving responsiveness while maintaining adequate torque margin. Integration testing will verify under-load performance, with contingency plans including a higher-torque servo or mechanical refinements if necessary.

D. Bat MCU: ESP32 S3 vs. Raspberry Pi 5

The choice of microcontroller we put on the bat is extremely important, as achieving the seamless, physical gameplay that we want would be dependent on the capabilities of the

microcontroller to process sensor data and communicate it with the board. Our options for the microcontroller unit were narrowed down to the ESP32 and the Raspberry Pi 5, both of which are highly popular cost-effective units for wireless projects.

Our three main requirements for the microcontroller is to: 1) support low latency wireless communication, as to meet our requirement for a snappy and immersive gameplay; 2) consume minimal power, given that the circuit will running for the entire duration of a game; and 3) familiar to and easy to set up, as to mitigate time spent learning the ins and outs of the device, allowing us to streamline the development of the MVP quicker.

While both the ESP32 and Raspberry Pi offer support for Bluetooth and WiFi, the ESP32 also offers ESP-NOW, a peer-to-peer protocol for ESP32 microcontrollers which has relatively low latency compared to Bluetooth and WiFi. ESP-NOW is also optimized for short-burst messaging, which fits into our desired application for wireless technology. In terms of power consumption, the ESP32 and Raspberry Pi draws 22mA and 2A of current respectively. Finally, given that everyone in the group has had experience working with Arduino microcontrollers, we believe that the ESP32's compatibility with the Arduino IDE would best facilitate familiarity and ease of development. Upon these considerations, the ESP32 is our microcontroller of choice for this project.

VI. SYSTEM IMPLEMENTATION

Our system can be broken down into three major systems (and several minor ones):

A. Fabrication:

As mentioned in Section III and pictured in Figure 1, the game board houses the sensor-embedded playing surface, the central controller, and the pitch subsystem.

The playing surface will be a ~3x2 feet acrylic sheet, featuring cutouts for puck drop-holes. The drop-holes will have breakbeam sensors embedded underneath them to detect if the puck has passed. This signal is sent back to the central controller, which updates the game state accordingly.

Beneath the playing surface is a region designated for gantry travel, roughly 500x500 mm from the home plate outwards.

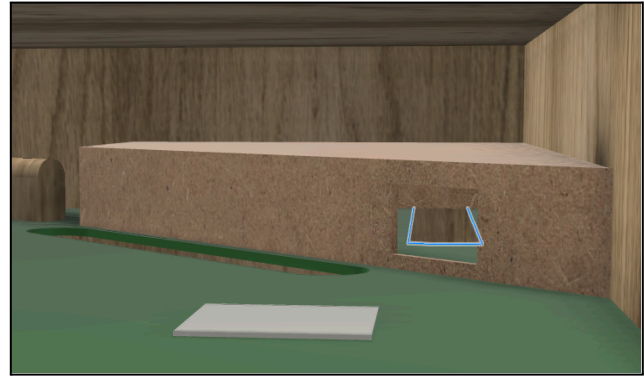


Fig. 4. Zoomed-in image of the right extrusion, which disguises a cutout in the playing surface for the return ramp.

Slightly outside and below the gantry designated area is an acrylic bottom floor that is tilted towards the right wall, where there is a chute. After falling into a pocket, the puck drops onto the floor and into the chute, sliding down alongside the right wall until it reaches the bottom-right. The gantry slides under the puck and couples, then glides it along a ramp back up to the surface. Once it reaches the surface, it pops out of the extrusion to the right of home plate and returns to the pitcher's mound for the next pitch.

Central Controller: The central controller is an ESP32 MCU running the Game State Machine that is described in Appendix 1. It interfaces with the bat subsystem via ESP-NOW, the flipper and scoring sensors via GPIO, and the gantry controller via UART.

Puck: The game "ball" is a compact steel puck that is roughly 1" in diameter and weighs 40g. The puck was loosely based on a pinball, and was chosen to be heavy enough to maintain stable motion on the playfield and to remain coupled while not being too heavy for the flipper servo.

B. Pitch (Gantry + Coupling)

The Pitch subsystem is pictured in Apdx. 2, and is split into three main parts as follows:

Gantry Hardware: The gantry kit consists of 2040 V-slot extrusions, a gantry plate with V-wheels, NEMA17 stepper motors for each axis, GT2 timing belts and pulleys, and additional brackets, spacers, and nuts.

Gantry Control: The gantry is controlled by an ESP32 running FluidNC, an open-source and feature-rich firmware that enables motion-planning without requiring low-level step generation. Two TMC2209 stepper drivers are used to drive the two axes. A 24V, 10A power supply should provide sufficient power for the motors (and game board), which draw at most 2.5 A each. Homing is implemented with two limit switches mounted at a known corner of the travel (bottom-right reference), which the gantry uses at startup to establish a repeatable coordinate frame.

Coupling Mechanism: The puck is coupled to the gantry using neodymium magnets mounted on the gantry carriage beneath the playfield. During the pitch segment, the magnets remain close to the playfield to maximize coupling force and reduce slip. About 100mm from the plate, the system will

decouple the puck by retracting the magnet assembly in the Z-axis by about 15mm. This decouple method provides a quick and precise retraction without adding significant weight to the gantry.

The Z-retraction mechanism is implemented using a small servo-driven rack-and-pinion linear actuator module, which was created by user “potentprintables” on Thingiverse.

C. Bat and Flipper System

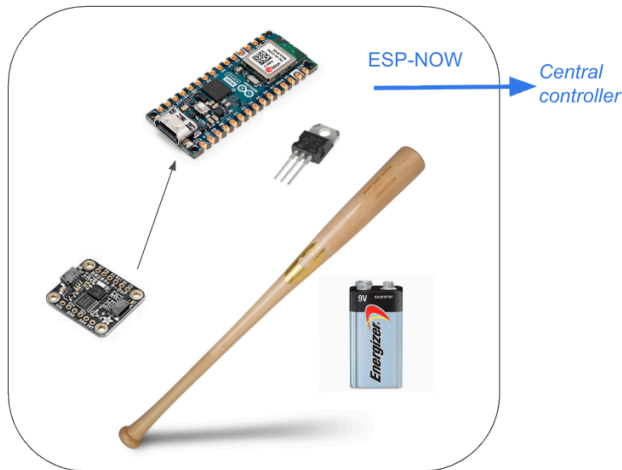


Fig. 5. Components of the Bat System

The bat and flipper subsystem is split into three main parts as follows:

Swing detection (IMU): The ESP32 microcontroller mounted on the bat is connected to a BNO085 IMU through a wired I²C protocol. The IMU allows for 9 degrees of freedom in the 3 dimensions, which is used to determine whether the bat has been swung or not. From the batter’s perspective looking at the pitcher, we let the y-direction parallel to the general pitch movement, and the x-direction to be the left and right of the pitcher. To determine whether the bat has been swung, we take into consideration the bat’s movement in the xy-plane and its speed. When the tip of the bat reaches a speed of above ~10 m/s (around half of the average little league bat speed) and subsequently breaks the plane of home plate, a swing is registered, and the microcontroller transmits this information to the game board.

Wireless protocol (ESP-NOW) Swing events are transmitted from bat to board using ESP-NOW to minimize latency. Each message includes a swing event flag and a timestamp. The board-side controller validates packets and forwards a single swing trigger to the flipper control logic.

Board-side servo wiring, power, and mechanical mount (portable main supply). The flipper actuator servo is mounted under the board near the flipper pivot using a rigid bracket to reduce compliance and improve repeatability. The servo is powered from a portable main power supply to a

dedicated servo power rail with capacitors near the actuator to mitigate voltage droop during fast actuation. The board-side controller provides the servo PWM control signal, and a common-ground reference is maintained between the controller and servo rail. Mechanically, the servo horn is attached via screws to a ~3-inch 3D printed flipper.

VII. TEST, VERIFICATION AND VALIDATION

A. Tests for Design Specification: Gantry Motion

We will verify gantry motion performance using scripted pitch trajectories and repeated trials while logging timing and position. For speed verification, we will run 20 trials of the fastball profile and measure the time required to traverse the pitch segment; we will compute average gantry speed and confirm it meets or exceeds 500mm/s. For curve verification, we will run 20 trials each of slider and screwball and measure net lateral displacement during the pitch segment using either (1) commanded gantry position logs or (2) a marked grid/video overlay, and confirm lateral displacement is at least 150mm.

B. Tests for Design Specification: Repeatability

To verify repeatability-related requirements, we will run 20 repeated trials per pitch type and record both the time from pitch start to decouple and the lateral position at decouple. We will confirm the decouple timing remains within ± 50 ms for each pitch type, and that the decouple lateral position remains within ± 2 mm for each pitch type. These tests directly verify the technical bounds required to achieve pitch variety, pacing, and repeatability.

C. Tests for Design Specification: Latency

We will verify responsiveness by timestamping, in the Arduino IDE, the swing detection to packet transmission, and packet receipt on the game board to flipper actuation. We will run at least 200 hit events and compute latency distributions for each stage. For each hit, we expect the measured latency to be below 100ms.

D. Tests for Design Specification: Game Board

We will verify scoring and state update correctness with controlled scoring trials and log-based checking. For each scoring hole, we will perform at least 50 controlled events and confirm the sensor for that hole triggers if and only if the ball enters that hole, with misfires occurring in less than 1% of the trials.

To verify the correctness of game state computation, we will feed the gameboard a known sequence of scoring events and compare the computed game state to a manual ground truth log. We will confirm the game state computation is correct in $\geq 99\%$ of scoring events, and investigate any failures as either sensor classification errors or FSM logic defects.

VIII. PROJECT MANAGEMENT

A. *Schedule*

Our tentative project schedule is shown in Appx. 4. To summarize our process: we are planning to work on building working prototypes for the bat and flipper subsystems through the first week of March. In the two weeks following, we will then integrate the bat and the flipper subsystems, while working on the build of the game board at the same time.

We hope to have a completed game board before the last week of March, at which we will fully integrate all of the parts together, achieving a minimum viable product (MVP) by the start of April.

After achieving MVP, we plan to spend the rest of April working on fine tuning details and hashing out any existing problems, especially those related to gameplay experience, at which point, we hope to have a polished product ready for the final demonstration in early May.

B. *Team Member Responsibilities*

Bing is primarily responsible for the bat subsystem and its wireless integration with the game board. This includes assembling the circuit and physical components that make up the bat, and fine tuning the software for swing detection and wireless communication in order to ensure smooth gameplay. He is also responsible for working with his teammates to integrate subsystems together.

Aiden is primarily responsible for the pitch system, consisting of the gantry and coupling mechanism. This includes gantry assembly and control, and coupling mechanism fabrication and control. He will also help with the game board design and fabrication.

Vivian is primarily responsible for designing and building the flipper subsystem. This entails modeling and 3D printing the flipper component, building the central controller to flipper control logic, and writing the software for it, too. She will also assist with the game board development and integration of the project's subsystems.

C. *Bill of Materials and Budget*

Please refer to Appx. 4 for the project's bill of materials as of February 27th, 2026.

D. *TechSpark Use Plan*

The process of building the game board will require the use of some TechSpark resources, mainly the laser cutter, 3D printer, and the machine shop. Beyond that, we do not foresee using TechSpark in any other capacity.

E. *Risk Mitigation Plans*

The biggest challenge for us going into this project is working with the gantry system. Apart from being highly mechanical, none of us had experience working with gantries

before, let alone building one for our own specific application and incorporating a magnetic decoupling system to top it off. To mitigate the complications that may arise, we are cautiously moving along with the implementation; by building up from a simple single-axis system, we hope to be able to familiarize ourselves with the technology. We have also developed backup plans just in case implementing the pitch delivery with a gantry proves to be too big of a hurdle for the timespan of the semester.

Another challenge we are being considerate of is the enjoyability of the gameplay mechanic itself. Since the gameplay requires the batter to control the flipper on the board by physically swinging, the timing of the bat, ball, and flipper must be exactly right in order to deliver the aspect of immersion that we are trying to achieve. Of course, the only way to test this out would be to play the game for ourselves. Thus, we hope to have an MVP ready for testing as early as possible, which then leaves us with ample time to work on the fine-tuning of the gameplay.

IX. RELATED WORK

While doing research for the ideation phase of this project, we encountered a product that almost perfectly captures the idea of the gameplay that we had in mind. The Williams Deluxe Official Baseball is a pinball-style arcade game released by Williams Electronic Games in 1960. The gameplay itself is simple: a pinball is dropped down the middle and the player bats it with a flipper, like a game of pinball. Yet still, the game boasts many mechanical features and quirks, which of course were all implemented with technology from over half a century ago. It is quite fascinating to see the complexity of the craft, and the ingenuity that the design radiates.

The remainder of this column is intentionally left blank.

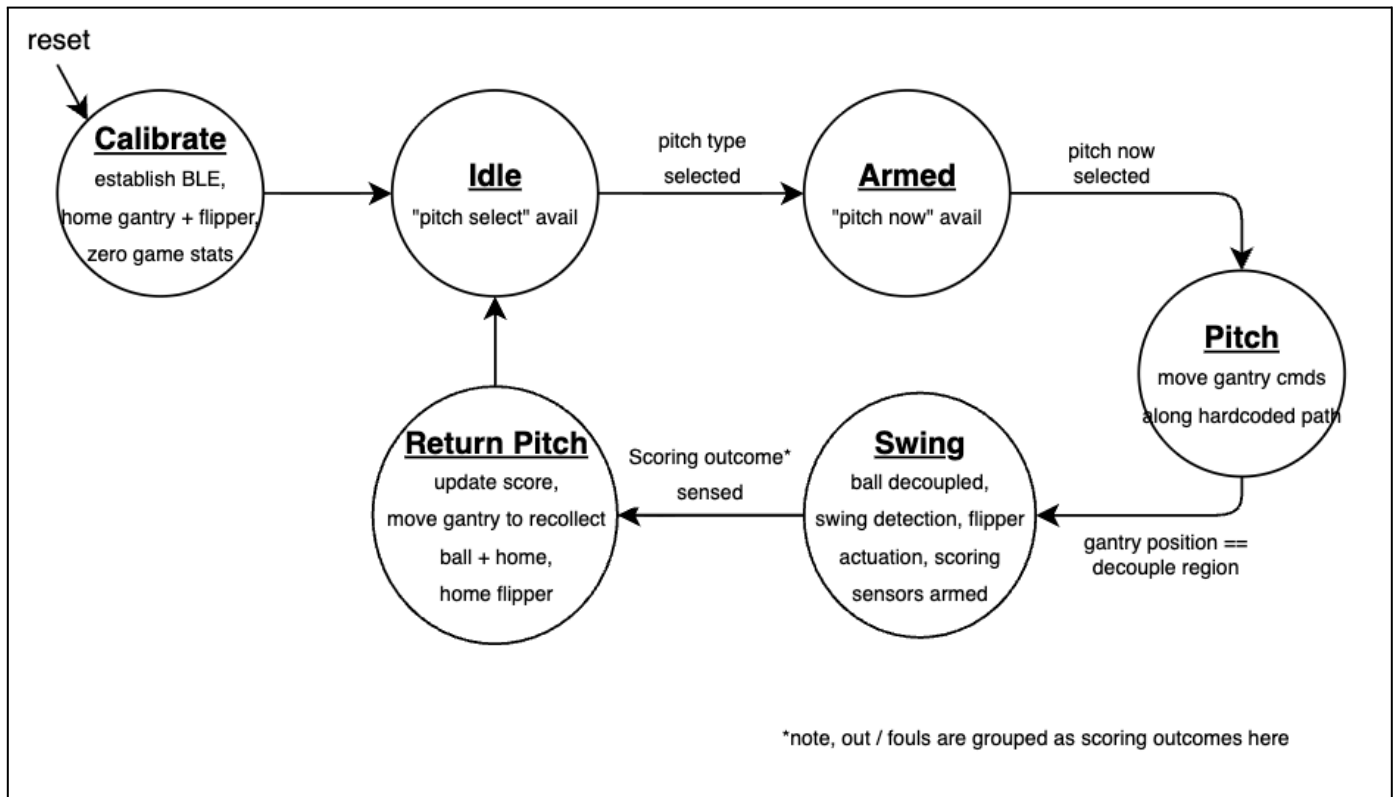
X. SUMMARY

In designing our project, we set out to address a central problem identified in our introduction: that modern entertainment increasingly isolates individuals behind screens, disconnecting people from each other. To combat this, we hope to build a two-player tabletop arcade experience that promotes meaningful social interaction, and delivers a fun and immersive experience for players. By combining a magnetic gantry pitch system, wireless bat swing detection, and sensor-based scoring, we recreate the timing, anticipation, and excitement of baseball in a compact, electromechanical format.

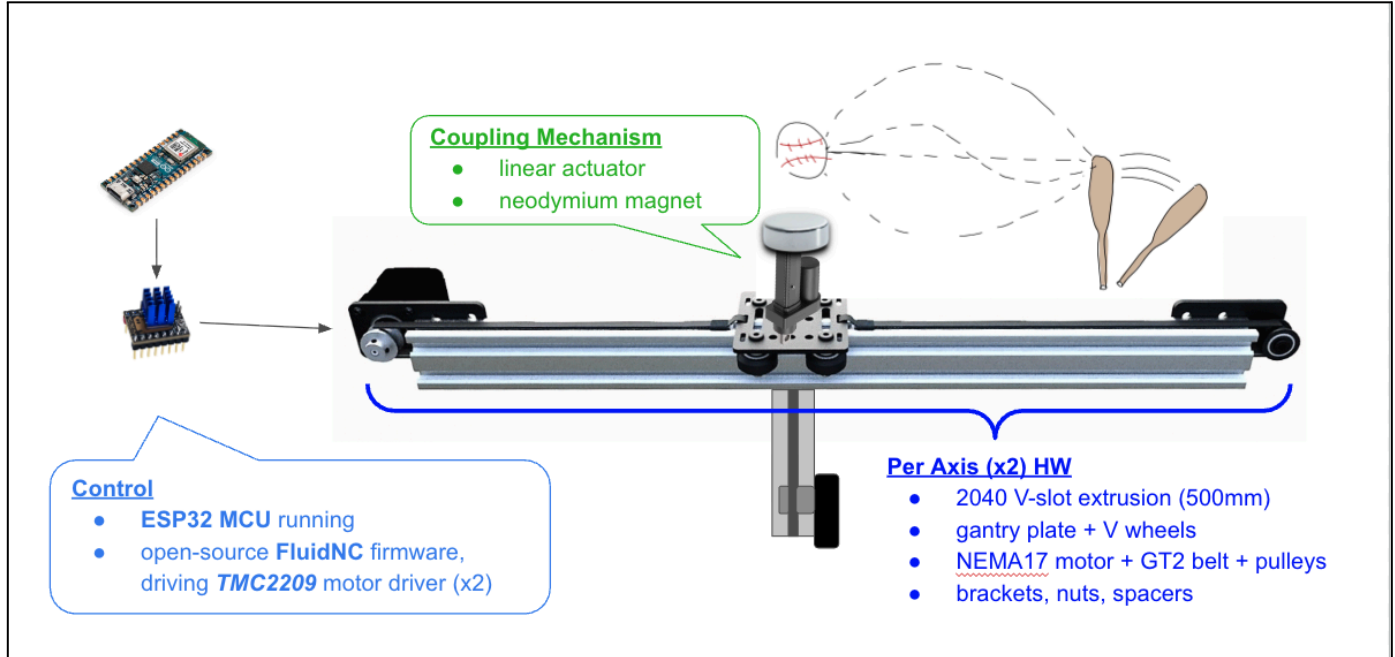
We recognize that there are a multitude of implementation challenges and road blocks that lie await, such as ensuring a reliable and snappy gameplay, integrating a multitude of moving parts together, or just having people find the game entertaining in general. We are, however, confident in our abilities to address these challenges, and work towards fulfilling our goal of creating a social, immersive, and engaging experience for all.

REFERENCES

- [1] "CoreXY explained: Comparison + strengths & weaknesses" Teaching Tech. YouTube.
<https://www.youtube.com/watch?v=ramiM3KHYE&t=559s>
- [2] OpenBuilds. Nema 17 V-Slot Belt Driven Actuator 500mm. GrabCAD.
<https://grabcad.com/library/nema-17-v-slot-belt-driven-actuator-500mm-1>
- [3] Adafruit Industries. Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout – BNO085 (BNO080) – STEMMA QT / Qwiic.
<https://www.adafruit.com/product/4754>
- [4] Raspberry Pi Foundation. Raspberry Pi 5 Product Brief.
<https://pip-assets.raspberrypi.com/categories/892-raspberry-pi-5/documents/RP-008348-DS-5-raspberry-pi-5-product-brief.pdf>
- [5] Arduino. Arduino® Nano ESP32.
<https://store-usa.arduino.cc/products/nano-esp32>
- [6] Pinside. Williams Deluxe Official Baseball.
<https://pinside.com/pinball/machine/deluxe-official-baseball>



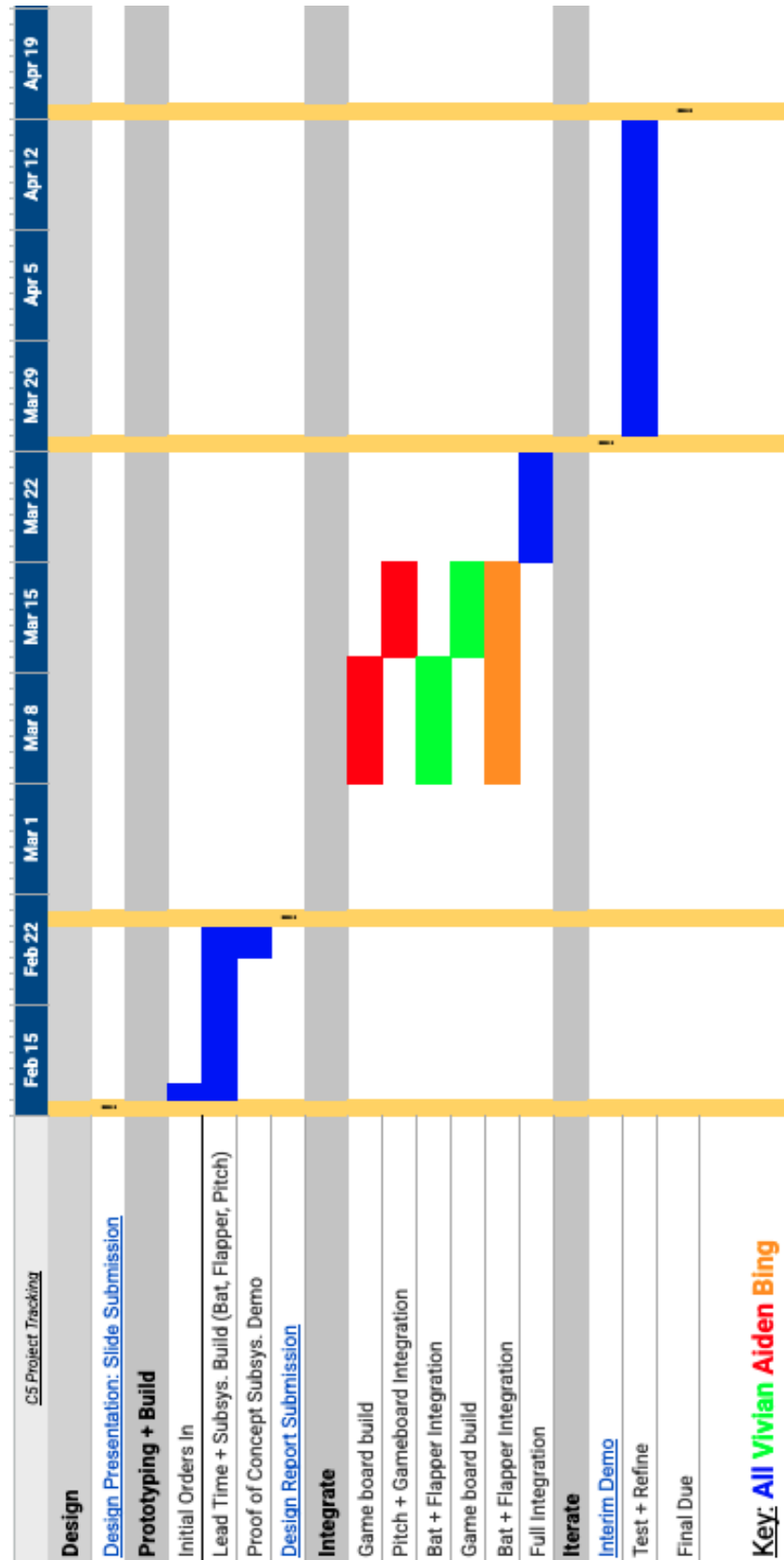
Appendix 1: Game System Machine Diagram



Appendix 2: Gantry Design

Subsystem	Name	Quantity	Price/Unit	Vendor	Total Spent		
Ball	Steel Rod	1	\$12.00	McMaster-Carr	\$12.00		
Bat	I2C IMU	1	\$31.94	Adafruit	\$31.94		
Bat	MG90S	1	\$10.00	Miuzei	\$10.00		
Bat	9V Battery	2	\$4.00	Home Depot	\$8.00		
Bat, Pitch, Gameboard	ESP32 MCU	3	\$18.30	Arduino	\$54.90		
Coupling	Continuous Rotation Servo (higher torque)	2	\$12.00	Adafruit	\$24.00		
Coupling	1/4"x1/16" Nd. Disc Magnets	15	\$1.00	n/a	\$15.00		
Coupling	PLA Filament	1	\$15.00	SUNLU	\$15.00		
Flapper	BLS3225 Servo Motor	1	\$36.00	ANNIMOS	\$36.00		
Gameboard	1/8", 3'x2' Acrylic Sheet	2	\$50.00	Home Depot	\$100.00		
Gameboard	1/2", 4'x8' Plywood Sheet	1	\$50.00	Home Depot	\$50.00		
Pitch	Belt-driven Linear Actuator Kit	2	\$70.00	bulkman3d	\$140.00		
Pitch	TMC2209 Motor Drivers	3	\$6.67	BIGTREETECH	\$20.00		
Pitch, Gameboard	24V, 10A Power Supply	1	\$60.00	KAINSC Products	\$60.00		
						Total	\$576.84
						Remaining	\$23.16

Appendix 3: Bill of Materials



Appendix 4: Tentative Gantt Chart