Computer Aided Design and Prototyping

(ME444 – SPRING 2021)

Mackey Mania

Design Report

Group No. 11

Trevor Ladner	Junior - ME
Peyton Young	Junior - ME
Mason Trenaman	Junior - MDE
Nitin Pauletti	Junior - MDE

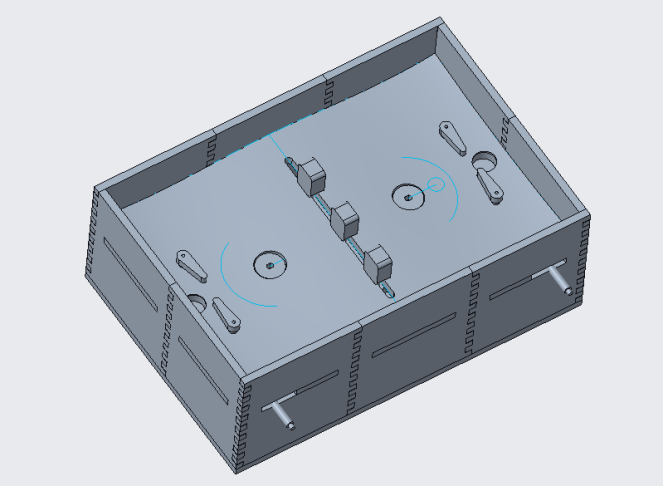
School of Mechanical Engineering
Purdue University

Revision History

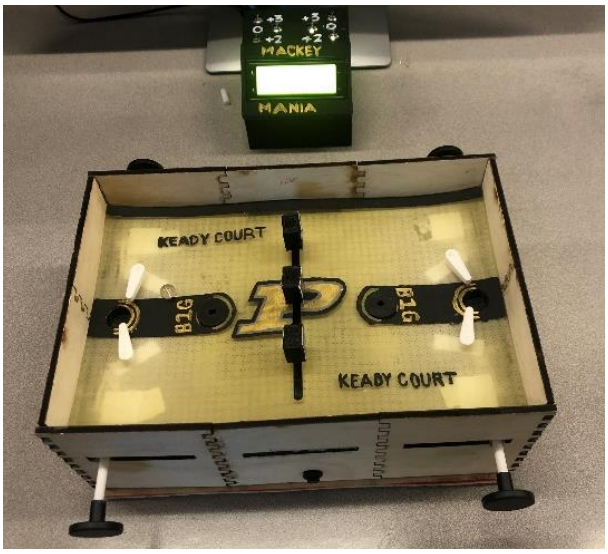
S. No.	Date	Revision ID	Revision Details (Page No., Paragraph, Line No. etc.)	Author
1.	5/2/21	Revision A	Initial Release	TL, PY, NP, MT
2.	5/4/21	Revision B	Final Version	TL, PY, NP, MT

Executive Summary

Mackey Mania is a two-player tabletop game meant to appeal to the memories of Purdue Basketball while combining the games of foosball and pinball. The game's goal is to defend your basket using two flippers that the user controls and score on your opponent's goal. In the center of the board is a sliding mechanism with three defenders: Purdue's Matt Painter, Carsen Edwards, and Tommy Luce. At the top of either free-throw lane is a spinning disk that redirects the game ball differently. Lastly, the game is made slightly more difficult by having a non-flat floor that guides the game ball towards the goal. This has a dual function to ensure that the game ball also does not get stuck somewhere while in play. The game is designed to look like the iconic Keady Court in Purdue's Mackey Arena.



CAD Assembly of Final Toy, Located in Action Toy → CAD → actiontoy.asm



Final Prototype Toy Seen From Above with Working Scoreboard

Table of Contents

Revision History 2
Executive Summary 3
1. Introduction 5
1.1 Project Backgrounds 5
1.2 Design Requirements & Constraints 6
2. Concept Generation and Selection 6
3. Design 7
3.1 Floor 7
3.2 Obstacles 7
3.3 Flippers 9
3.4 Scoreboard 11
3.5 Final Design 12
4 Prototyping and Testing 13
5. Results and discussion 14
5.1 Prototype Functionality 14
5.2 Future Improvements 14
5.3 Lessons Learned 15
6. Conclusion 16
References 18
Appendices 18

1. Introduction

1.1 Project Backgrounds

Boiler Up, Hammer down; just one phrase ingrained into every Purdue student's mind from day one. Therefore, logically when tasked to create a novel toy for a Purdue course, one thinks of an aspect of campus. With March Madness being hosted on campus this year, the famous Mackey Arena had garnered quite a bit of attention over the last few weeks. Wanting to create a basketball-themed game, inspiration was drawn from another game based on a sport, foosball. However, to ensure players would experience utterly new gameplay, the combination of a pinball playing field made the most sense.

Breaking the toy down, there are four main components. First is the playing field, sloped precisely to give players a challenge and ensure the ball never gets stuck. Second are the flippers, used to hit the playing ball to both score points and defend against shots from the opponent. Third are the two distinct obstacles, one moving linearly and the other spinning included to give the game another dimension of difficulty. Lastly, a scoreboard is implemented to introduce the game and keep track of scoring. The essential operation of the toy starts by plugging in the scoreboard for power and turning on the main game board. Once this is accomplished, play starts by dropping the ball into the main playing area. Players 1 and 2 must then try to score in their opponent's "basket" or defend theirs by hitting the ball by pushing in the flipper rods. Points are awarded based on two and three points for a ricochet shot and clean shot, respectively. After each score, each player can keep track of their score by pressing the button of the corresponding point they just scored. The first player to 24 points, the same number of championships won by Purdue, wins the game. Resetting the score back to zero is done via the "zero" button on the scoreboard.

In terms of marketability and target age range, this toy leaves little on the table. Due to the small ball used for playing, this toy is suited for children ages four and older to mitigate the risk of choking. As for market potential, Mackey Mania may limit itself due to niche branding. However, any Purdue fan would undoubtedly want to own a Purdue-themed toy. Another significant aspect of Mackey Mania is how unique the product is. While pinball and foosball are two well-known games, they can leave something to be desired; for example, there are very few, if any, two-player pinball games. Mackey Mania is innovative because it allows for a competitive, familiar atmosphere while introducing exclusive features like an LCD screen and moving

obstacles. Overall, Mackey Mania builds upon the success of other tabletop games to produce a fun, challenging, and competitive game that anyone, but especially Purdue fans, can come to love.

1.2 Design Requirements & Constraints

The final goal of this project was to design and build a working prototype of an action toy. The toy must have been innovative and significantly different from any toy currently available, and mechanically complex. Mechanical complexity has been defined for this project as having at least two non-trivial mechanisms or control systems. The toy must also be powered by hand or battery. The toy may be constructed using any tool or process available on Purdue's campus. The overall size of the toy had a soft cap of 10" X10" X10", and the total cost must be at or below 60 dollars.

2. Concept Generation and Selection

The team wanted to design a toy that was true to Purdue and themed it around as such. Early on, many ideas were thrown around, but two primary themes were centered on: Mackey Arena/Purdue Basketball and the famous Big Bass Drum.

The first idea that ended up being the final selected idea is 'Mackey Mania': a 2-player tabletop game that combines many favorites like pinball, foosball, air hockey, and basketball. The game was initially supposed to feature a round board and would have an entire shell to resemble the whole design of Mackey Arena at Purdue. The game would feature various obstacles on the board, including spinning disks that could redirect the ball somewhere random on the board and sliding obstacles that make it hard for the ball to reach the opponent's side. The game's goal is to protect your individual basket using flippers and score on your opponent's goal.

The second toy idea was briefly termed 'Hit That Big Bass Drum.' HTBBD was a hand crank toy that featured the Big Bass Drum spinning like it famously does, and the user would turn the crank to control the BBD crew member. The goal would be for the user to crank the toy so that the crew member would hit the drum's head in time. This was a much less complex toy than Mackey Mania as it was only single-player and dealt solely with timing the hit correctly.

Mackey Mania was ultimately chosen as the primary toy design as it had the most complexity and would produce the most fun for the consumer. It was known that it would be a challenge to make but would have the most extensive consumer base and was the most feasible.

3. Design

3.1 Floor

The floor was one of the most critical elements to the gameplay of Mackey Mania. It was designed to aid in the players' ability to score and ensure constant and nearly randomized gameplay. This was done by using a two-directional boundary blend to replace the surface of a solid rectangle. Running down the length of the floor, the boundary was the same on either side – a shape that somewhat reflects 2 periods of a cosine wave. Running the other direction, there were two paths to be considered – the center court and the end of the courts. The line that runs down the center of the court is completely straight and runs perpendicular to the face that the lengthwise path runs down. The ends of the court are simple parabolic. Figures 1 and 2 show the profile of each floor boundary used to create the boundary blend.

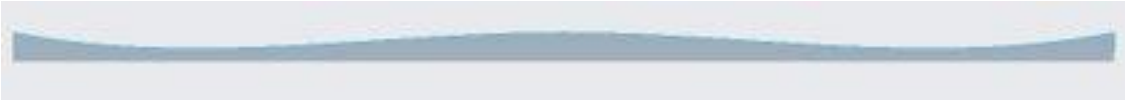


Figure 1: Lengthwise Boundary

Figure 2: Horizontal Boundaries

3.2 Obstacles

Several obstacles provide an added challenge to the gameplay of Mackey Mania. These obstacles are intended to disrupt the ball's path to add an element of difficulty or surprise. Two different obstacles were included in Mackey Mania. The first is a set of two spinning floor pieces. These are in the area that a jump ball circle would typically reside on a basketball court. The second obstacle is a Scotch Yoke mechanism located under the floor that operates 3 moving paddles above

the floor. All three of these obstacles (two spinning floors and paddle mechanism) are powered by hobby DC motors mounted to a custom 3D printed mount. This mount was then directly bolted to the bottom of the game box using M5 nuts and screws. This mounting technique can be seen in figure 3.

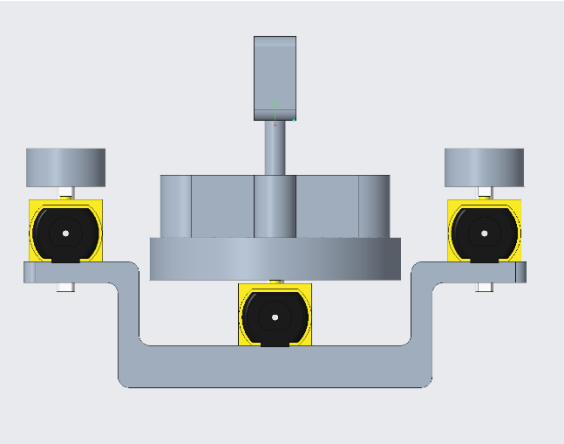


Figure 3: Mounting Bracket for Obstacles

The Scotch Yoke mechanism was designed, so the team to use a rotary motor to generate linear motion. A crank attached to the DC motor would spin and cause a slotted slider to move back and forth with the rotation of the crank. Three paddles were mounted on top of the slider. These paddles protrude from the game floor through a slot in the center where a midcourt line would be present. The slot on the floor acts as the linear guide for the mechanism. As the crank spins, the paddles move left and right at a high rate of speed to make a direct shot from one end of the court to the other more difficult. Rebounds off these paddles also require quick reaction times from the players, making the game more fast-paced and exciting. These paddles were also decaled with two former famous Purdue Players, Tommy Luce and Carsen Edwards, and Purdue's coach, Matt Painter. These decals add to the overall theme of the toy. Figures 4 and 5 show the geometry and positioning of these obstacle mechanisms.

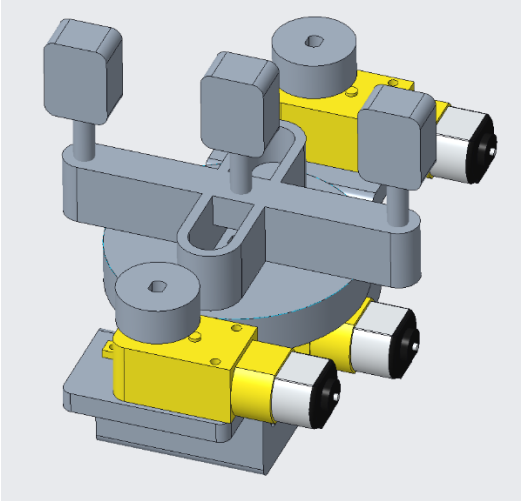


Figure 4: Scotch Yoke Paddle Mechanism and Spinning floors

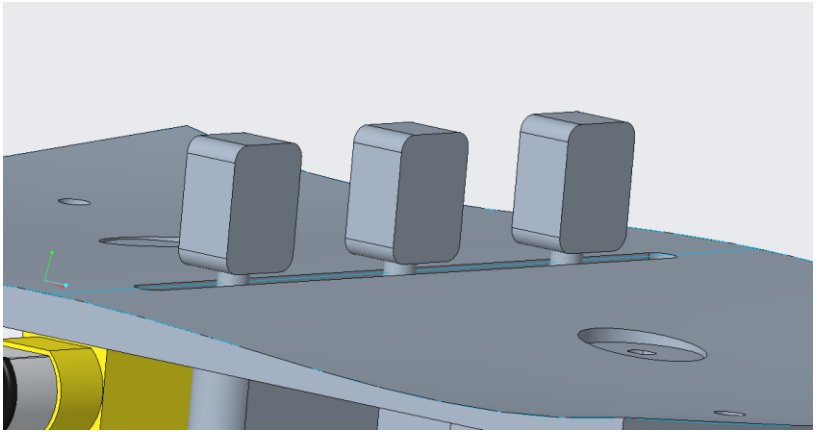


Figure 5: Obstacles as Seen Above Game Floor

3.3 Flippers

During the early concept generation phase, the team wanted to implement mechanically actuated flippers to simplify and minimize cost. An early design was based on a schematic provided to Pinterest for a wooden pinball flipper mechanism. The team was drawn to this design because it did not need any electrical components, instead of using a rubber band that functioned somewhat like a spring that would pull the flipper mechanism back to a datum position when activated. That existing design (which can be found in the project report references) was changed to work better with PLA filament and meet the project size and material constraints. This changed design became the final design for the flipper mechanism of the project and contained 4 small

parts. A small rod was cylindrically constrained into a square housing with a hole, and when pushed upon, would exert force near the edge of a rectangular block. The top of the block contained a stepped shaft, the larger of which was cylindrically constrained to a hole in the pinball floor and the smaller of which to the flipper part. The assembled mechanism is pictured in the following figure:

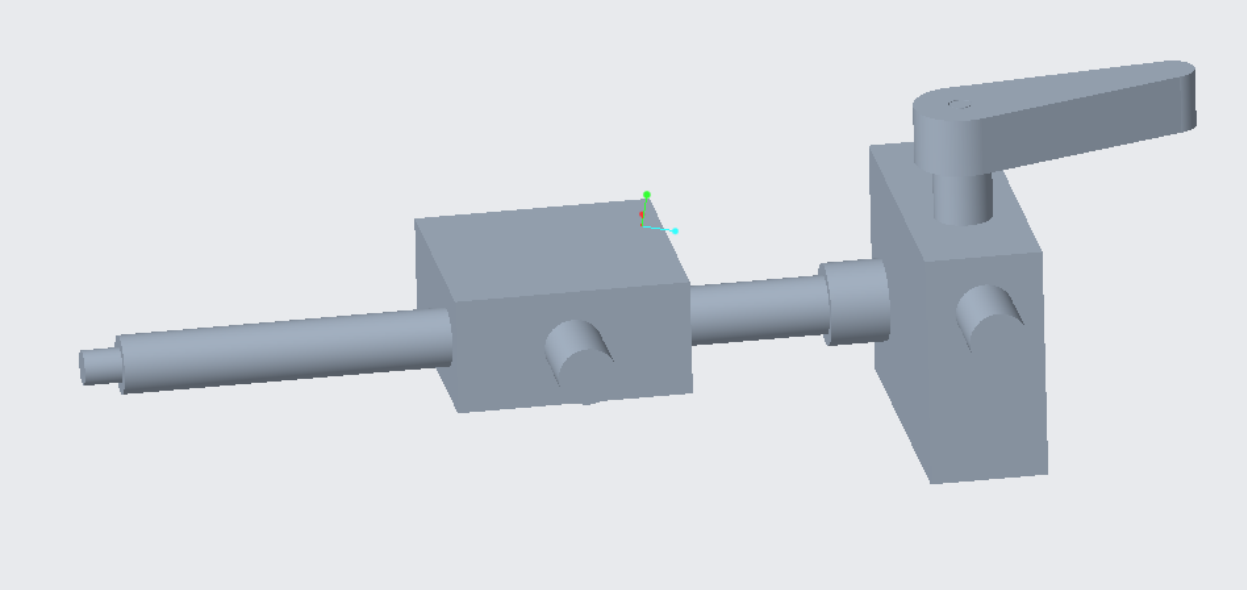


Figure 6: Flipper Mechanism assembly

When assembled to the main assembly, first, the rod housing was glued to the underside of the pinball floor. The final pushrod design, which contains a lip to help actuate the rectangular flipper connector block, was then inserted into the housing. Finally, the flipper connector was slotted into a hole in the floor and then capped with a flipper. The radial part of the flipper connector was sanded at the connection point to allow for adequate rotation when the rod was pushed against the connector. The assembly was spaced so that a small rubber band would be at its minimum tensile length when attached to the radial outcroppings shown above. Each part of this assembly was designed to be symmetrical for DFA considerations, which allowed the same 4 parts to be used for each of the four necessary flippers.

3.4 Scoreboard

During the planning phase, it was discussed how scoring should be implemented into Mackey Mania. Early on, the team agreed that an LCD scoreboard is the best way to integrate this system. The early design of such a system involved possible pressure sensors or IR sensors to recognize when the ball entered the "basket." Unfortunately, after realizing the time required to print the wall to which the screen was to be mounted, the decision was made to create a separate housing for the scoreboard. Thus, having sensors that needed to connect to the scoreboard via wires was not a feasible option. Instead, the team opted to have three buttons for each player; one button for each of the scores possible, two and three points, and the last to reset the player's score back to zero. The housing itself was designed to hold the I2C LCD screen, buttons, and Node32s. The LCD screen is placed at an angle to ensure the best viewing angle for both players. The buttons are placed on top for easy access, and the Node32s with all the wires are housed inside.

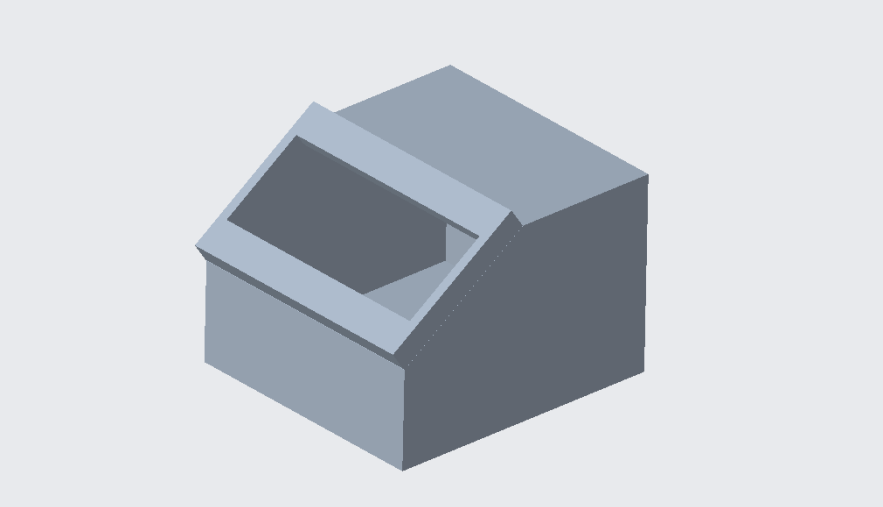


Figure 7: CAD model for the scoreboard housing

As mentioned earlier, the LCD and buttons are being powered and controlled via Node32s running a custom Arduino script (see appendix). As the name suggests, the scoreboard had to display the score and accurately track it throughout the game. To accomplish this task, the following logic was written into the code.

1. Include custom library for I2C LCDs
2. Set pin on Arduino for button
3. Initialize two variables for a "current" and "previous" state of the button
4. In a loop, compare current state to previous state (via *digitalRead* function)

    a. If different (i.e., the button was pressed), then increment the counter, otherwise continue in the loop
5. Save current state as the previous state for next time through the loop
6. Print the current value of the counter (for each player)
7. Repeat

Lastly, in addition to showing the current game score, the LCD of the scoreboard was used to display a welcome message at startup. In summary, the scoreboard is an I2C LCD connected to a Node32s that uses buttons to increase and reset score while also displaying a welcome message on startup.

3.5 Final Design

The final design required the integration of multiple sub-assemblies. The scoreboard system and holder were separate and came together with ease. However, the floor, obstacles, and flippers were much harder to integrate. The floor had to sit within the outer shell well but also had to make it so that the obstacle would mesh well with their locations on the floor. This was a meticulous process to ensure that the scotch-yoke and rotating disks would line up. If they did not line up, they would likely end up not working fully as planned or even getting stuck in the slot and not functioning at all. The flipper also had to line up with the lower mechanism that controlled its actuation. However, there was room for error with how the flipper mechanism connected on the lower side, which helped with the ease of assembly.

With the exception of the actuation rod, the flipper mechanism had to be assembled on the floor before it could be placed within the outer shell of the game. The obstacle mechanism was bolted into the bottom of the game shell. After that, lining up the floor was the next step. Once this was done, the only part left was adding the opponents into the scotch-yoke mechanism through the top of the floor. The assembly was rather tedious in some ways but was optimized to the best ability able.

4 Prototyping and Testing

This prototype was constructed using a variety of different fabrication methods. The walls of the box were laser cut to fit together like puzzle pieces. These were designed to fit snugly, so there was an added dimension of +.01in to the pieces' puzzle teeth to account for the laser's kerf. The game floor was SLA printed to ensure that it had the necessary print resolution to ensure the ball would travel smoothly and as intended. The flippers, obstacles, support for the floor, and scoreboard holder were FDM 3D printed using both PLA and PETG. FDM parts that required close fit with other components such as the motor crank and spinning disks were designed with a -.2in tolerance to account for the swelling that occurs when a part is 3D printed. These parts were also designed to have a 3D printer-friendly geometry to reduce print failures. The game base was hand cut out of ¼" plywood using tools from the Mallot Prototyping Suite.

Many issues were found and corrected throughout the assembly and testing phase. Several of the FDM parts had to be reprinted numerous times to hone the fit tolerance. PLA and PETG are also not the most robust materials, so parts had to be reprinted when they broke deformed. The geometry of these parts was also updated to help combat part failures. One major issue that was discovered was that the bottom base of the toy was not perfectly cut to the intended dimensions. Because of this, the game would sit crooked, and the slanted floor would cause the Scotch Yoke mechanism to jam. To alleviate this deformity, a quarter-inch of plywood was cut and adhered to the base to provide an even platform for the game. There was also extensive testing that went into how the obstacle motors were wired. It was determined that parallel wiring produces the most reliable results, so the motors were soldered accordingly.

The BOM underwent several iterations as the scope and implementation of the prototype changed. The most considerable change in scope was the decision to no longer seal the top of the toy with acrylic. As this decision was made later in the design process, the material had already been purchased. All other purchased materials were successfully utilized for this project, and the prototype stayed under the 60-dollar budget, as evidenced by the Bill of Materials included in the appendix.

5. Results and discussion

5.1 Prototype Functionality

Given that testing results indicate Mackey Mania is primarily working as intended, Team 11 is overall pleased with the performance of their toy. The physical prototype functions well, all things considered, but there are many observations and analyses that the team made towards improving the overall design that warrants discussion.

Firstly, an analysis of the functional ability of the physical prototype is in order. All functional components worked, but some had unique issues that were not thought about when considering their design in CAD.

For the scotch yoke assembly, unwanted extra frictional resistance (from rubbing against the base) occurred in the groove for the player pieces if the floor and walls were not aligned perfectly to the base, causing the yoke to catch at the ends of the groove and stopping the mechanism from moving correctly. Considerable care had to be made to make sure the alignment was perfect in the final product.

For the flipper mechanisms, the side holes in the laser-cut walls of the assembly were not designed with the pushrods in mind. As a result, they were not aligned to the sidewall openings, and an additional hole had to be drilled to improve this. Had we thought about the correct placement, this step could have been altogether avoided.

Although not a moving part, the base warped after printing to the point that a ball in the arena could sometimes come to a stop in an area where nothing could keep it moving. This consequence is one of the effects of the SLA printing process that our team failed to consider from the CAD alone.

5.2 Future Improvements

The rapid prototyping process naturally creates room for errors and mistakes, as thoroughness in design validation is limited due to project scope. If our team had another prototyping cycle, we believe we could refine the initial design by adding the following scrapped features that we initially included in the project scope but had to drop due to time limitations.

1. A rail or chute that would use gravity to move the ball to an open location.

This inclusion would likely call for an FDM printed chute assembled under the hole for scoring, one for each side. The chutes would be curved and would lead to an opening in one of the walls

containing an open area, also likely FDM printed, that would serve as a repository for scored balls. A player could easily retrieve the scored balls.

2. A closed top for the toy with an opening for the ball.

In section 4, a sheet of clear acrylic was bought to serve as a "dome top" for the prototype. Our team never got to implement this part due to a mixture of being unclear about shaping the acrylic into a dome properly and the time constraint. With another design cycle and the right manufacturing capabilities, we could implement a domed top, or more likely due to its simplicity, a flat closed top by fastening the acrylic to the wood walls and drilling out a reasonably large hole for the ball. Although the latter option could have been implemented currently, not having a rail or chute for the ball meant that there would not be a way to get the ball once someone scored; therefore, closing the top was abandoned.

3. A lip on flipper parts to prevent the rubber band from slipping.

This inclusion should have been implemented in the initial design but was not considered until the final assembly. The basic premise is simple, adding a small lip to the radial outcroppings of each flipper assembly, which would keep the rubber band in place when actuated. For the sake of time, our team simply hot glued the rubber bands to mechanisms. Having more time would allow us to add this feature, which is a better design overall.

5.3 Lessons Learned

The overall project experience was very valuable to the team and generated significant learning outcomes that should be applied to other projects going forward.

The strength of 3D printed materials is not considered in the CAD design of a product. Due to these parts being made from plastic, they can break. The team learned this the hard way, breaking multiple parts during the final assembly. Extra parts had to be printed, and if they were not printed and on hand during the assembly, the project timeline could have been significantly delayed. A suggested best practice that the team has learned is always to print extras of small breakable parts (if the budget allows for this).

Tangential to prototype functionality is that parts that work on paper or CAD may not work in an actual product. Our team encountered an example for an initial design on the scotch yoke mechanism. The initial design had the middle of the three-player pieces rotating and the edge player pieces moving linearly. This design involved a circular slot in the pinball floor in addition

to the linear one. However, doing this created an issue where two small parts of the floor between the circular and linear slot had no possible place for support. Although this design worked in CAD, these two small floor parts would have been floating in the air if that design were actualized. As a result, the team had to change the yoke design to the current iteration. This example is indicative that sometimes the design must be altered on the fly in the rapid prototyping process. Being able to adapt to this is beneficial.

A tried-and-true lesson learned is the adage to "measure twice and cut once." Our team made a few dimensioning errors when producing the physical product from the CAD, such as cutting the wood yoke base half an inch too short on one side and mounting the yolk half an inch too far. Had our team properly checked our measurements, we could have avoided additional work needed to correct these measurement errors.

Electronic components can be unpredictable in behavior when not assembled well. The team designed a circuit for the yoke mechanism, but it triggered only some motors of the mechanism during implementation. This issue was due to the connections not being utterly secure in some areas for our team, and careful attention was needed when soldering. One of the wires to a motor also popped out during final assembly, and the team had to resolder that part onto the motor. Having extra care with these kinds of electric components can go a long way to prevent more of these types of issues from happening.

6. Conclusion

Team 11 is pleased with their current prototype for Mackey Mania, and we feel that we learned a plethora of valuable knowledge, experiences, and practices for the rapid prototyping process. Specifically, the components of the project timeline that the team felt went most well were their time management, Arduino code implementation, overall project aesthetics, and final product functionality. These were all critical design parameters that the team focused on during the prototyping process.

The team also recognized specific factors that they believe to be missteps in the prototyping process. These include manufacturing the wall pieces of the toy, which were initially 3D printed and proved to a hindrance and time sink due to the poor-quality prints that resulted from their printers. If not for last-minute laser cutting, the final toy may have ended up being unfinished.

Additionally, the overall design stability was not considered in the project CAD. The resulting Mackey Mania toy is relatively fragile, which became a constant worry that the team had to consider during final assembly.

Finally, the team believes that they were somewhat overambitious in visual design, incorporating many additional features and buying some parts, like the acrylic, that they ended up not using or implementing in the final design due to the quick nature of the project timeline.

Team 11 hopes to have provided a detailed account of their experience designing, manufacturing, and building Mackey Mania from the insights mentioned above.

References

flipper mechanism: Pinball diy, Pinball, Wood games. Pinterest. (n.d.). https://www.pinterest.com/pin/568227677971174891/.

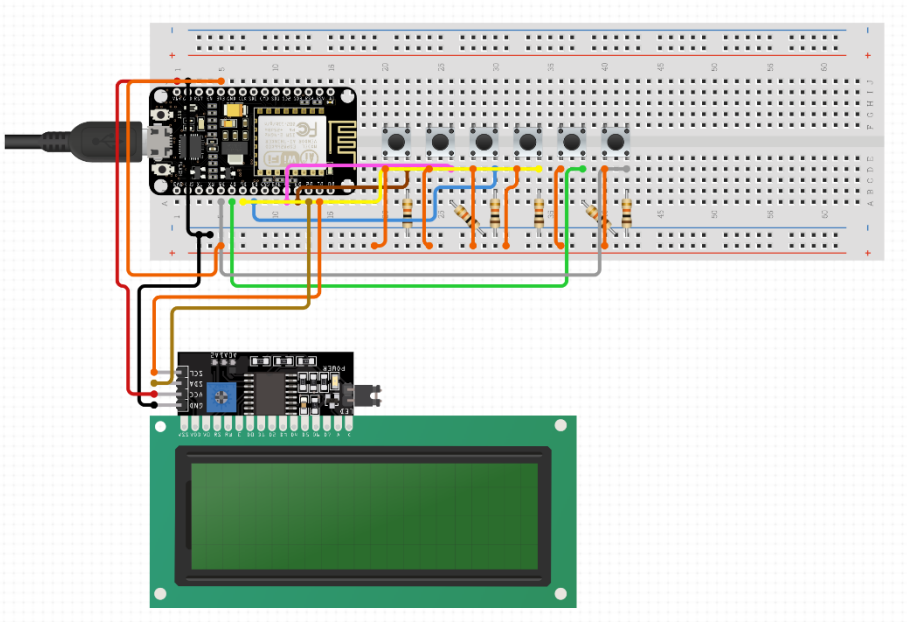
Ratansi, A., Santos, S., Moya, M., Osorio, R., Iasi, I. de, Hogan, K., … Puttskii. (2019, April 2). *I2C LCD with ESP32 on Arduino IDE - ESP8266 compatible*. Random Nerd Tutorials. https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/.

Appendices

Part Components	Component Type	Quantity	Specifications	Unit Cost	Total Cost	
Base	COTS	1	12" X 8.5" X 1/4" ply	7.59	7.59	
Game Board	Printed	1	SLA printed		Printing Budget	
Spinning Disk	Printed	2	PETG printed		Printing Budget	
Arena Top	COTS	1	12'' X 8.5''	11.98	11.98	
Arena Wall	COTS	8	Laser Cut ply			
People	Printed	3	PETG printed		Printing Budget	
Flippers	Printed	4	PLA printed		Printing Budget	
Flipper Mounts	Printed	4	PLA printed		Printing Budget	
Motor Mount	Printed	1	PETG Printed		Printing Budget	
Slider	Printed	1	PETG printed		Printing Budget	
Crank Arm	Printed	1	PETG printed		Printing Budget	
Handle Knobs	COTS	1	repurposed cabinet knobs	9.99	9.99	
9V battery	COTS	1			Donation	
Motor	COTS	3	dc motor		Donation	Prof. Liu Gave 2 Extra
Wires	COTS	1	for connecting to motor		Donation	
Screws	Purchase	1	M5 Screws	9.99	9.99	
Buttons	Purchase	1		8.99	8.99	
Scoring Display	Purchase	1		7.99	7.99	
Marble	Purchase	1	Clear bead, .45" diam		Donation	
				Complete	56.53	

Appendix A: Bill of Materials

Appendix B: Circuit Diagram



Appendix C: Arduino Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);  // set the LCD address to 0x27 for a 16 chars and 2 line display

// this constant won't change:

const int P1two_buttonPin   = 17;   // 17the pin that the pushbutton is attached to
const int P1three_buttonPin = 16;   //16
const int P1reset_buttonPin = 4;    //4

const int P1two_buttonPin2   = 9;   // the pin that the pushbutton is attached to
const int P1three_buttonPin2 = 10;
const int P1reset_buttonPin2 = 11;


// Variables will change:
//PLayer 1
```

```
int buttonPushCounter = 0;   // counter for the number of button presses
int up_buttonState = 0;        // current state of the up button
int up_lastButtonState = 0;     // previous state of the up button
int P1reset_buttonState = 0;
int P1reset_lastbuttonState = 0;
int down_buttonState = 0;        // current state of the up button
int down_lastButtonState = 0;     // previous state of the up button
bool bPress = false;

//Player2
int buttonPushCounter2 = 0;   // counter for the number of button presses
int up_buttonState2 = 0;        // current state of the up button
int up_lastButtonState2 = 0;     // previous state of the up button
int P1reset_buttonState2 = 0;
int P1reset_lastbuttonState2 = 0;
int down_buttonState2 = 0;        // current state of the up button
int down_lastButtonState2 = 0;     // previous state of the up button
bool bPress2 = false;
```

```
void setup()

{
 Serial.begin(9600);

 pinMode( P1two_buttonPin , INPUT_PULLUP);
 pinMode( P1three_buttonPin , INPUT_PULLUP);
 pinMode( P1reset_buttonPin, INPUT_PULLUP);

 pinMode( P1two_buttonPin2 , INPUT_PULLUP);
 pinMode( P1three_buttonPin2 , INPUT_PULLUP);
 pinMode( P1reset_buttonPin2, INPUT_PULLUP);

 lcd.init();                  // initialize the lcd


 // Print a message to the LCD.

 //Player 1
 lcd.backlight();
 lcd.setCursor(5,0);
 lcd.print("Welcome to");
 lcd.setCursor(4,2);
 lcd.print("Mackey Mania!");
 delay(4000);
 lcd.clear();

 lcd.setCursor(0,0);

 lcd.print("Player 1");

 lcd.setCursor(3,2);

 lcd.print(buttonPushCounter);
```

```
 //Player 2
  lcd.setCursor(12,0);
  lcd.print("Player 2");
  lcd.setCursor(15,2);
  lcd.print(buttonPushCounter2);
}
void loop()
{
   checkUp();
   checkDown();
   checkResetP1();
   if( bPress){
       bPress = false;
      lcd.setCursor(3,2);
      lcd.print("   ");
      lcd.setCursor(3,2);
      lcd.print(buttonPushCounter);
   }
    if( bPress2){
       bPress2 = false;
      lcd.setCursor(15,2);
```

```
    lcd.print("   ");

    lcd.setCursor(15,2);

    lcd.print(buttonPushCounter2);
  }
}

void checkUp()
{
 up_buttonState = digitalRead(P1two_buttonPin);
 up_buttonState2 = digitalRead(P1two_buttonPin2);

 // compare the buttonState to its previous state

 if (up_buttonState != up_lastButtonState) {

  // if the state has changed, increment the counter

  if (up_buttonState == LOW) {

     bPress = true;

    // if the current state is HIGH then the button went from off to on:

    buttonPushCounter = buttonPushCounter + 2;

    Serial.println("on");

    Serial.print("number of button pushes: ");

    Serial.println(buttonPushCounter);

  } else {

    // if the current state is LOW then the button went from on to off:
```

```
   Serial.println("off");

  }

  // Delay a little bit to avoid bouncing

  delay(50);

}

// save the current state as the last state, for next time through the loop

up_lastButtonState = up_buttonState;

if (up_buttonState2 != up_lastButtonState2) {

  // if the state has changed, increment the counter

  if (up_buttonState2 == LOW) {

     bPress2 = true;

    // if the current state is HIGH then the button went from off to on:

    buttonPushCounter2 = buttonPushCounter2 + 2;

    Serial.println("on");

    Serial.print("number of button pushes: ");

    Serial.println(buttonPushCounter2);
  } else {

    // if the current state is LOW then the button went from on to off:

    Serial.println("off");
  }
```

```
   // Delay a little bit to avoid bouncing

   delay(50);
  }

  // save the current state as the last state, for next time through the loop

  up_lastButtonState2 = up_buttonState2;

}

void checkDown()
{

  down_buttonState = digitalRead(P1three_buttonPin);
  down_buttonState2 = digitalRead(P1three_buttonPin2);

  // compare the buttonState to its previous state

  if (down_buttonState != down_lastButtonState) {

    // if the state has changed, increment the counter

    if (down_buttonState == LOW) {

       bPress = true;

      // if the current state is HIGH then the button went from off to on:

      buttonPushCounter = buttonPushCounter + 3;

      Serial.println("on");

      Serial.print("number of button pushes: ");

      Serial.println(buttonPushCounter);
```

```
  } else {
    // if the current state is LOW then the button went from on to off:
    Serial.println("off");
  }
  // Delay a little bit to avoid bouncing
  delay(50);
}
// save the current state as the last state, for next time through the loop
down_lastButtonState = down_buttonState;
if (down_buttonState2 != down_lastButtonState2) {
  // if the state has changed, increment the counter
  if (down_buttonState2 == LOW) {
     bPress2 = true;
    // if the current state is HIGH then the button went from off to on:
    buttonPushCounter2 = buttonPushCounter2 + 3;
    Serial.println("on");
    Serial.print("number of button pushes: ");
    Serial.println(buttonPushCounter2);

} else {
```

```
    // if the current state is LOW then the button went from on to off:
    Serial.println("off");
   }
   // Delay a little bit to avoid bouncing
   delay(50);
  }
  // save the current state as the last state, for next time through the loop
  down_lastButtonState2 = down_buttonState2;
}

void checkResetP1()
{
  P1reset_buttonState = digitalRead(P1reset_buttonPin);
  P1reset_buttonState2 = digitalRead(P1reset_buttonPin2);

  // compare the buttonState to its previous state
  if (P1reset_buttonState != P1reset_lastbuttonState) {
   // if the state has changed, increment the counter
   if (P1reset_buttonState == LOW) {
     bPress = true;
    // if the current state is HIGH then the button went from off to on:
    buttonPushCounter = 0;
```

```
    Serial.println("on");
    Serial.print("number of button pushes: ");
    Serial.println(buttonPushCounter);
   } else {
    // if the current state is LOW then the button went from on to off:
    Serial.println("off");
   }
   // Delay a little bit to avoid bouncing
   delay(50);
  }
  // save the current state as the last state, for next time through the loop
  P1reset_lastbuttonState = P1reset_buttonState;
if (P1reset_buttonState2 != P1reset_lastbuttonState2) {
   // if the state has changed, increment the counter
   if (P1reset_buttonState2 == LOW) {
      bPress2 = true;
    // if the current state is HIGH then the button went from off to on:
    buttonPushCounter2 = 0;
    Serial.println("on");
```

```
    Serial.print("number of button pushes: ");

    Serial.println(buttonPushCounter2);

   } else {

    // if the current state is LOW then the button went from on to off:

    Serial.println("off");
   }

   // Delay a little bit to avoid bouncing

   delay(50);
  }

  // save the current state as the last state, for next time through the loop

  P1reset_lastbuttonState2 = P1reset_buttonState2;

}
```