Wireless Pacman-Inspired Tag Game

School of Engineering and Applied Sciences
Harvard University

Cambridge MA.

Description: A 3D Pacman game using remote controllers to navigate 2 players (the Pacman and 1 ghost) throughout the maze

By: Audrey Cheng, John Bugeraha, Miranda Morales, Zaria Ferguson

Class: Engineering Sciences 50
Instructor: Marko Loncar

May 2024

ABSTRACT
This project is a 3D version of the Pacman game using remote controllers to navigate 2 players (the Pacman and 1 ghost) throughout the maze. The goal of this project is to create a user-friendly game that is both challenging and fun by having the Pacman player navigate its way to the other end of a complicated maze while the ghost player is trying to tag them before they can escape the maze.

INTRODUCTION
This is a Pacman-inspired tag game. It is a two-player game, with one player acting as the ghost and the other acting as Pacman. Each player is represented by a small robot that can move in four ways: backwards, forwards, turning right, and turning left. The robots are both controlled wirelessly via bluetooth. The players will maneuver around a small maze as the ghost player attempts to tag the Pacman player. Two lasers were placed on either end of the ghost robot so that when it gets close enough to the Pacman, photoresistors on the Pacman will sense the increase in light and turn on a neopixel to signal that it has been tagged . We chose this idea because we wanted to create an interactive game that would engage our audience. We decided to modify Pacman, since it is a classic and very recognizable game. We are particularly interested in robots and wanted to use light and photoresistors, so we designed our game to incorporate these elements.

DESIGN
In developing our Bluetooth-controlled PacMan project, we encountered several issues during the process that led to the design decisions we made.

The first problem was that without proper containment, the wiring and batteries kept on falling off the robot during movement, causing interruptions and malfunctions. Additionally, without controlled lighting, it impacted the accuracy of the laser-tagging mechanism due to the photoresistors being exposed to the ambient light.

To address these problems, we had to 3D-print a casing that we attached on top of the robot to secure the battery and wiring components, as well as to block the surrounding light, preventing it from affecting the photoresistors. We left small openings at the front and back of the casing to allow for the lasers to flash and the photoresistors to receive light from the flash effectively.

During initial tests, the robots also flipped frequently due to rapid or sharp movements. To address this, we added weights to the base of the vehicles to lower the center of gravity and stabilize the robots.

For the controllers, we had four buttons for directional movement, ie. Forward, back, left, and right. We arranged the four buttons on the breadboard and used it as a controller. We placed the forward and backward buttons on one side, and the left and right buttons on the other side to allow for easy control.

Also, the wire colors we used on the controllers matched the color of the PacMan/Ghost paper we placed on top of the particular robot. This color-coding helped us identify which controller belonged to which robot.

For the maze, we initially used a PacMan maze we found online as inspiration for our maze designs. However, given the constraints on the wood available to us we made a few changes for the final design of our maze. We were mindful of the width of each mini robot car, 10.5 cm, in determining the width of the lanes in the lanes. We ensured that the lanes were wide enough for the cars to move in all four directions and minimize bumping into the walls but also narrow enough that both robot cars would not be able to fit side by side. The images of our initial and final designs are in the appendix.

PARTS LIST

Part/Material	Quantity	Unit Cost	Extended Cost	Link
Mini Round Robot Chassis Kit - 2WD with DC Motors	2	$19.95	$39.90	https://www.adafruit.com/product/3216
Arduino Nano 33 BLE	2	$26.30	$52.60	In Stock
Arduino MKR Wifi 1010	2	$38.60	$77.20	In Stock
Adafruit DRV8833 DC/Stepper Motor Driver Breakout Board	2	$5.95	$11.90	In Stock
Lasers	2	$5.95	$11.90	In Stock
Photoresistors	2	$0.95	$1.90	In Stock

NeoPixel Stick - 8 x 5050 RGB LED with Integrated Drivers	1	$5.95	$5.95	In Stock
Lithium Ion Battery - 3.7V 2000mAh	4	$12.50	$50.00	In Stock
Push Buttons	8	$0.13	$1.04	In Stock

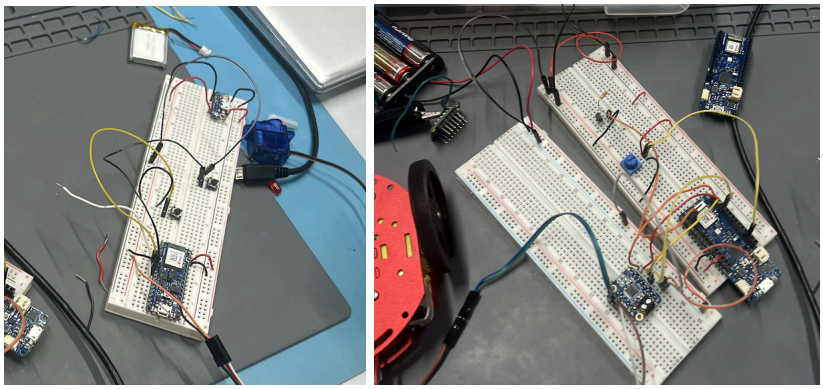
PROJECT IMPLEMENTATION

Assemble the robot cars

We used the Mini Round Robot Chassis Kit with DC motors from Adafruit as the frame for our robot. After the initial assembly, we later had to replace all four of the motors that came with the kit due to weak wire connections, but otherwise did not make any adjustments.

Set up Bluetooth Circuits & Bluetooth Arduino Code

When getting ready to set up our controllers, we quickly realized that the controllers we had ordered would not work the way we had originally planned. After consulting lab heads and TFs and a lot of brainstorming, we chose to continue working with Bluetooth and decided to design, build, and code our own controllers. We used a lab procedure from a graduate class provided to us as a foundation for learning about Bluetooth communication through Arduino boards and code. In completing this graduate student lab, we realized we could modify the circuits from the lab to build a controller circuit that could allow us to run the robot motors in a separate circuit.



Left: Initial controller circuit Right: Initial motor circuit

Based on the starter lab, we concluded we should use Arduino MKR Wifi 1010 and Arduino Nano 33 BLE boards for the two circuits that could communicate with each other. We decided to

use the MKR Wifi 1010 for the circuit that would go on the robot since these were closer to the Arduino boards we had previously used in lab and would allow us to add and control more elements to our circuit, such as the laser and neopixels. We also decided to use the BLE boards for the controller circuits as the inputs from this controller were the ones that needed to be communicated via Bluetooth. To avoid confusion and make troubleshooting easier, we decided to begin by developing one robot-controller set and ensuring it was successful and then replicate that to a second set afterward.

In designing our controller circuits, we decided we wanted our robot to be able to move forward and back and also turn either left or right. To do this we set up four pushbuttons on a breadboard in a layout that simulated a game controller. For each pushbutton, we connected one end to the breadboard's ground and the other end to a distinct pin on the BLE. To keep the controller circuit clean and connections secure without soldering, we did our best to keep our wiring as neat and short as possible. This circuit is referred to here on out as the Side B circuit.

Again to make it easier to troubleshoot our circuit while coding, we decided to begin our motor circuit by simply connecting each motor to the breadboard's ground and to a distinct pin on the Wifi 1010 board. After some initial trouble in getting our motors to run, we used a multimeter to measure voltage throughout our circuit and realized that the board was not providing enough power to get our motors to run. Thus, we redesigned our circuit by adding an Adafruit Motor Driver Breakout Board to provide the motors with enough power and to use the analog pins on the Wifi 1010 to control the direction of the motors. This circuit is referred to here on out as the Side A circuit.

Having set up both Side A (motor) and Side B(controller) circuits, we moved on to coding both sides. Following the general outline of the solution code for the graduate student lab we worked on before, we had a separate file for each circuit. We began by defining in the Side A code the pins used on the Wifi 1010 and in the Side B code the pins used on the BLE. Next, in the Side A code we defined a UUID for the BLE and each of the pushbutton input values so that there could be a connection established between the Wifi 1010 and BLE boards through which the input values from each of the pushbuttons in the controller circuit could be communicated. In the Side B code, we also defined the UUID for the connection between the boards and each of the pushbuttons.

To communicate the inputs of the controller, in the Side B code, the BLE was programmed such that while the BLE was connected to the Wifi 1010 board it would read the input of each pushbutton. More explicitly, each pushbutton was assigned an input value, *forwardPressed*, *backPressed*, *leftPressed*, or *rightPressed*. While the BLE and Wifi 1010 board were connected, each of these input values was read and assigned to a motor state value. These motor states, *forwardMotorState*, *backwardMotorState*, *leftMotorState*, and *rightMotorState*, were the values

that were communicated between the BLE and Wifi 1010 boards. In the Side A code, the Wifi 1010 board would continuously receive the values of these motor states. The board was programmed such that if a motor state value was 1(indicating that the corresponding button on the controller was pressed), the motors would spin such that the robot moved in the desired direction.

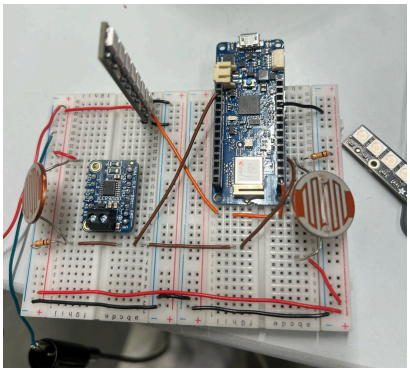
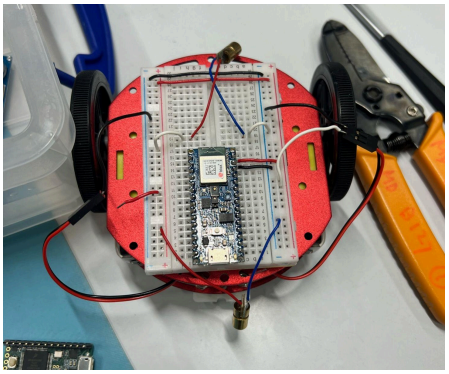
After testing our code and ensuring that the motors ran the way we wanted based on the controller input, we decided to get our circuits to run without being connected to a laptop as a power supply. We decided to use 4.5V battery packs as the power supply for the Side B circuit and to have a 3.7 lithium battery powering the Wifi 1010 board and a 4.5V battery pack powering the motor driver in the Side A circuit. With one controller-robot set completed, we replicated the same circuits and code(with distinct UUIDs for the BLEs and buttons) to get a second controller that would drive the second robot.

Success! Our circuits together with our code successfully allowed us to use a wireless battery-powered controller to drive our toy robots.

Set Up Laser Tag Circuits & Laser Tag Arduino Code

For the tagging mechanism, we added two laser diodes to the front and back ends of the ghost robot's motor control circuits. The pacman robot featured a circuit with two large photoresistors and a neopixel, which were connected to an Arduino MKR Wifi 1010. We initially designed the circuit based on the Serial Communication lab, but we ended up getting rid of the level shifter. We had also planned to use two small photoresistors, but decided to instead use larger photoresistors due to concerns that the circuit would have difficulty picking up the concentrated light from the laser diodes. As a result, we had to use two additional 10 kOhm resistors in series with the photoresistors. We then added the photoresistor and neopixel circuit to the breadboard used to control the pacman robot's motors.

The tagging mechanism only required code on the pacman circuit. Based on the Serial Communication lab, we wrote code that caused the neopixel to light up red when either one of the photoresistors were exposed to light. This involved some troubleshooting to determine the best resistance threshold received from the photoresistor to use when turning on the neopixel.

Left: photoresistor and neopixel circuit Right: LED diodes on ghost robot

3D Print Robot Casings

The casings were designed using Solidworks based on measurements of the robots using calipers. We designed and printed three different prototypes, changing the dimensions of the casings and adjusting different features to fit changes made to the circuits. These included adding cutouts to fit the specific breadboard orientations and adding coverage for the photoresistors. In terms of design, the casings are nearly identical besides minor differences to account for the particular breadboard layout of each robot. The pacman robot casing also has an additional slit so that the neopixel extends out of the casing and can be viewed by the players. Both casings have hexagonal openings on the top to allow for additional attachments. See appendix for part drawings.

The battery holders were also designed using Solidworks to fit the measurements of the batteries we used. They have hexagonal shafts attached to their base so that they can be easily attached to the tops of the robot casings. The holders also have circular, flat surfaces on which we added laser cut pieces with vinyl decals that resembled pacman and the ghost.

Build the Maze

Our maze was designed so that the lanes were wide enough for the cars to move in all four directions and minimize bumping into the walls but also narrow enough that both robot cars would not be able to fit side by side. To build the maze we used a wooden base as the base of our maze and wooden blocks to create the walls of our maze. We attached the wooden blocks to the base of the maze by first nailing the outer walls to each other at the corners and to the base of the maze. We then drilled the inner walls attached to these outer walls and drilled the remaining blocks into the base of the maze. The final maze design can be found in the appendix.

TEAM MANAGEMENT

We decided to distribute tasks according to everyone's interests and strengths. At the same time, our project required a lot of communication and interaction, so most of the time multiple members(if not all) were working together on the same task.

Miranda: assembled robots, set up the bluetooth circuits and code to control the robots (huge!), designed the maze, constructed the maze
Audrey: assembled robots, set up laser-photoresistor tagging mechanism circuits and code, assisted with bluetooth circuits, designed and 3D printed robot casings and battery holders, constructed the maze
John: designed & constructed the maze, set up bluetooth code to control the motors, assisted with the Bluetooth circuits
Zaria: designed & constructed the maze, set up laser-photoresistor tagging code, helped with troubleshooting the wire connections for the motors on the robot cars

OUTLOOK AND POSSIBLE IMPROVEMENTS

Reflecting upon the finished project, one of the most obvious areas of improvement is adjusting the Arduino code for the Pacman robot so that the motors and neopixel circuits can work simultaneously as intended. Adjusting the 3D printed robot covers would also help improve the functionality of the game. The battery holders currently attach to the top of the casings, which makes the robots top heavy and prone to tipping over. This issue could be solved by either adding weights to the base of the robot or moving the battery holder to give the robot a lower center of mass. Another possible improvement would be to lower the robots' speed to make them easier to control. They move very quickly as is, which makes them difficult to maneuver around the maze. A final potential improvement would be to design more user-friendly casings for the controllers. This would include designing a more stable method for attaching the power supply to the breadboards that were used as controllers.

ACKNOWLEDGEMENTS:
We would like to thank Professor Loncar, James Quigley, Champa Guruaj, and Leo Gomez for all of their help implementing bluetooth as the control mechanism for our robots and constant support throughout this project, as well as Ben Fichtenkort for his help with writing Arduino code for bluetooth and with motor drivers. We would also like to thank our TFs Oriana Ginji, Bella Pignataro, and Roderick Shumba who continued to encourage when parts of circuits did not work as we expected them to.

DISCLAIMER

Our report, code, and videos of our project are open to sharing.

REFERENCES:
We referenced a grad student lab guide (EE.L2 - Bluetooth Low Energy) as well as the corresponding solution code provided by James as a starting point for the Bluetooth control system.

We also used the following websites to become familiar with the BLE boards and code for the Bluetooth:
https://developer.android.com/develop/connectivity/bluetooth/ble/ble-overview
https://punchthrough.com/manage-ble-connection/

We used the following website to generate UUIDs for the code: https://bleid.netlify.app/
We used the ES50 Serial Communication lab guide as a starting point for the tagging mechanism.

APPENDIX

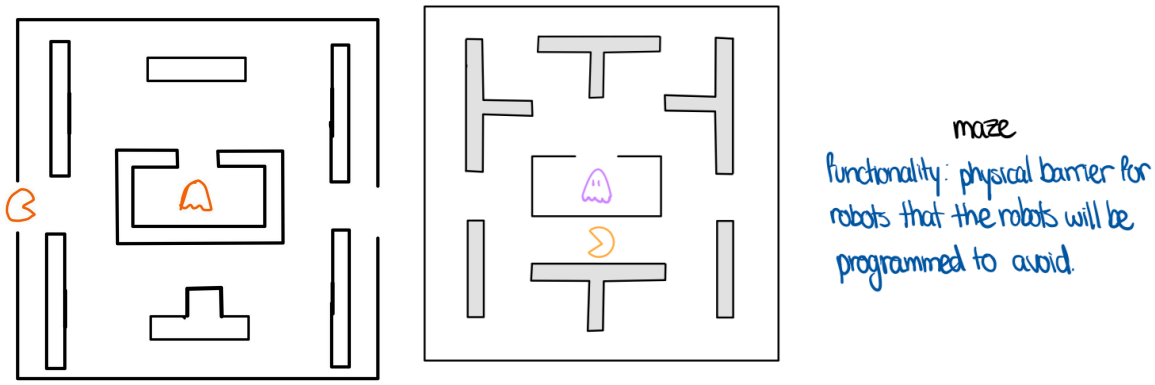


Image 1: Preliminary maze designs

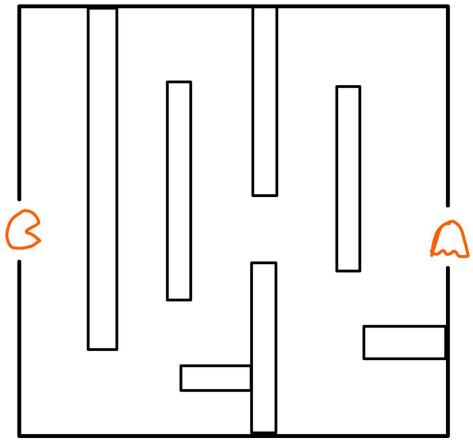


Image 2: Final Maze design

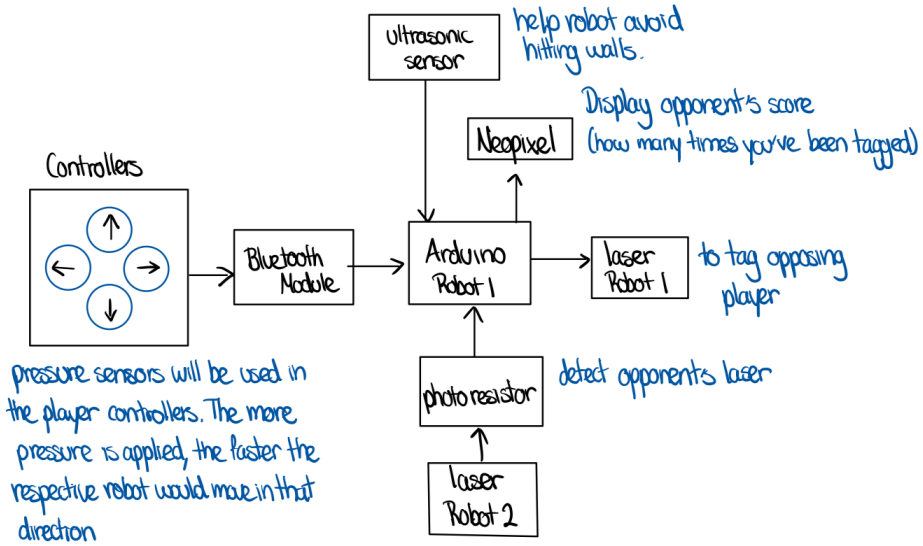
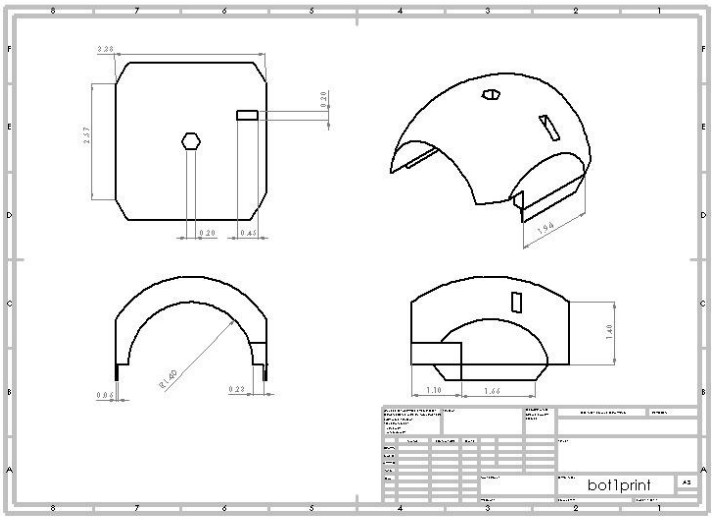
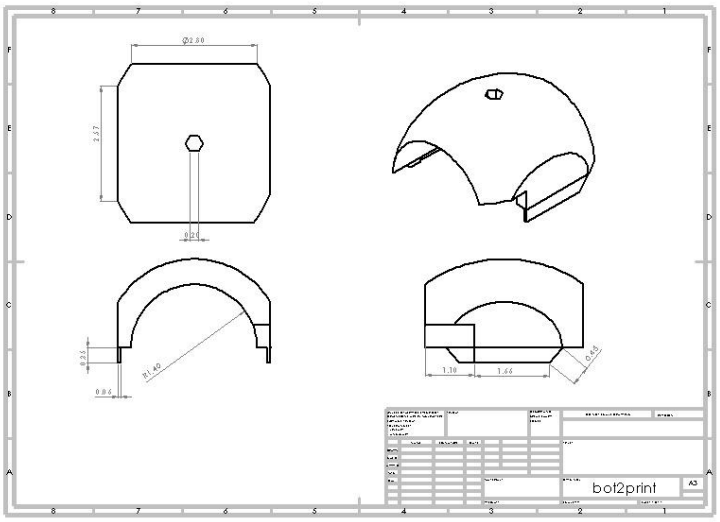


Image 3: Block Diagram





Images 4 & 5: Part drawings of 3D printed covers