ECE 1000 Final Report: Joystick-Controlled Labyrinth Game

Eli DeSha and Jude Lee,

Tennessee Technological University

ecdesha42@tntech.edu, jelee43@tntech.edu

Abstract—The Joystick Controlled Labyrinth Game described within this paper uses many different concepts such as using a Raspberry Pi Pico, servo motor controls, and trigonometric equations to create a new experience out of an over 70-year-old game. With the use of two servo motors, a dual-axis analog joystick, and a Raspberry Pi Pico, the game board turns on both axes, and while difficult, it is playable. The following report outlines the creation, functionality, and usage of the game.

I. Introduction

Labyrinth has always been a simple yet captivating game. The way that only two knobs are required to do such complex motions is fascinating, and thus, this project was born. This project attempts to emulate that feeling, but instead of needing to use two knobs, only the use of one component is needed. And that component, is a simple joystick. While using much more complex components than just some simple knobs, this project attempts to emulate the simplicity of the original game, only using two servos, a joystick, and a Raspberry Pi Pico.

II. Background

During the creation of our joystick-controlled Labyrinth game, we used a variety of different resources, including YouTube, various websites, and GitHub. Since we are inexperienced and new to the field, much of the coding was sourced from these places. It helped us to understand the fundamentals of this project, and overall, helped us clean it up to its current state.

III. Project Description and Formulation

Materials:

1. Raspberry Pi Pico: This is used as the main basis of functionality, sending all instructions to respective parts.
2. Two Servo Motors: These are the components that physically turn the board.
3. Dual-Axis Analog Joystick: This component serves as the input from the user, allowing the player to control the board.
4. Labyrinth Game Box: The game itself, without any modifications.

Diagram: The diagram below shows an Arduino used instead of a Raspberry Pi Pico, due to the limitations of the Tinkercad software being used. Additionally, there is

an RGB LED in use, in place of the joystick for the same reason. The black wire represents ground, red represents 3.3V, and the orange and yellow wires represent the x and y axes of the joystick respectively.

Figure 1: Tinkercad simulated system

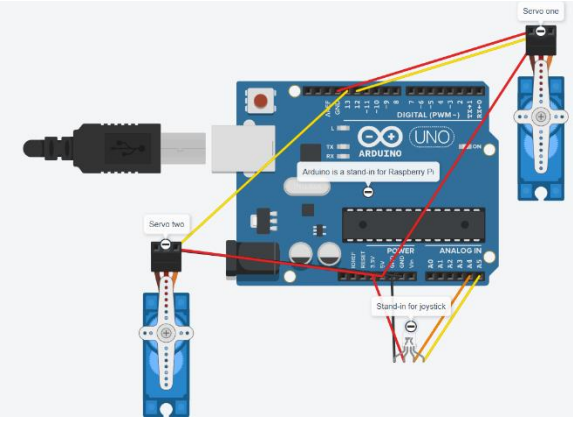


Figure 2: Components before assembly

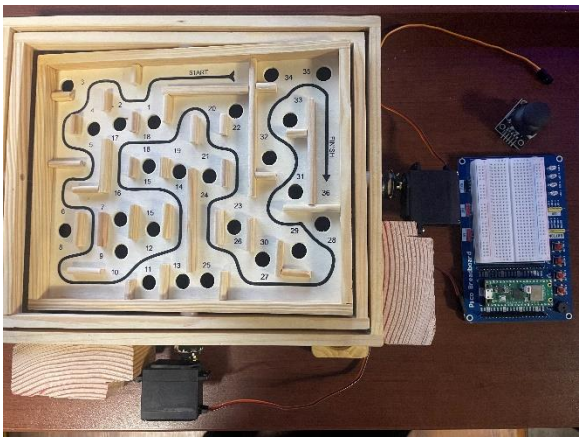
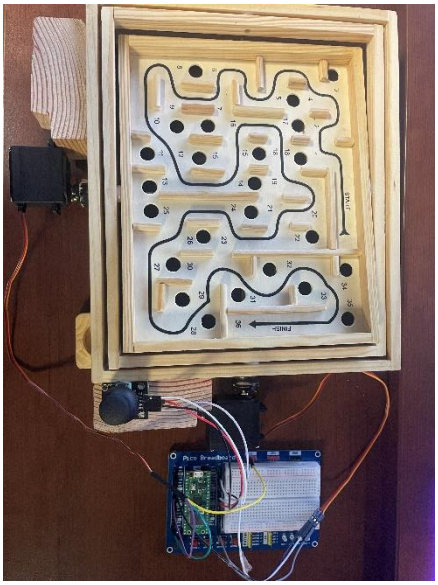


Figure 3: Fully assembled device



Functionality:

The components are all connected to various pins on the Raspberry Pi Pico, which include GPIO (General Purpose Input/Output), ground, and voltage pins. There are 4 GPIO pins used: 16 and 17 for the servo instructions, and 26 and 27 for the joystick. The pins used for the joystick are called ADC pins, since they convert the analog movement to a digital signal.

The Raspberry Pi checks for the x and y values of the joystick every tenth of a second, and outputs those values respectively. Alongside those values, a trigonometric function is used to calculate the angle of the joystick based on the x and y inputs. This angle is then translated into a frequency that can be read by the motor, to put it at that angle.

Both motors are programmed to operate on the same angle, thus both turning the same amount at the same time. This allows the ball to roll and be controlled by the player, as both the x and y axes of the board move.

IV. Discussion and Results

This project, while somewhat successful, still has much room to improve. The first improvement that we discovered should be made is that the motors need to operate independently of one another. Another is that the motors should have their resting state set to the point of equilibrium on the board, meaning that when the joystick is released, the motors return to a state where the board is centered and level.

Overall, we could work on implementing these other fixes, as well as new features such as an automatic ball return, a timer that detects when you have finished, and potentially even a mode with another board where to players race each other.

As for individual contributions:

Jude Lee provided all the tools necessary to mount the motors, as well as assisting with the mounting, coding, and implementation of the motors and joystick. His help was invaluable, as without his expertise of the use of tools, this project would have not been possible.

Eli DeSha assisted with the mounting of the motors, to a lesser degree than Jude, but mostly dealt with the research regarding the coding and implementation of the program.

While we did contribute in different ways, we both worked side-by-side throughout the entirety of the project, and this greatly hastened the development of it.

V. Conclusion

The most important part of this device is the Raspberry Pi Pico, since it handles all the calculations and instructions regarding joystick position and servo position. It is an invaluable component, since this makes the implementation of code, connections, and construction seamless in its operation. It also allows users to interact with the device effortlessly, without any prior knowledge of how the device works.

The Joystick-Controlled Labyrinth Game uses various components and concepts such as mechanical engineering, servo motors, Micropython, and joystick controls to create a playable remake of an old game, Labyrinth. These systems interact with one another to create the effect of moving an entire plate around with the flick of a finger, embodying the simplicity of its predecessor.

References

[1] John Caleb. (2024, November 22). Explanation for code by John Caleb. GitHub. https://github.com/JCWilliams1003/ECE-1000-Spring-2024-Final-Project-Insert-Project-Name.

[2] Paul McWhorter. (2024, February 27). Code sourced from this website. TopTechBoy. https://toptechboy.com/calibrating-joystick-to-show-positional-angle-in-micropython-on-the-raspberry-pi-pico-w/.

[3] Paul McWhorter. (2024, March 5). Instructions for servo code. YouTube. https://www.youtube.com/watch?v=ayY2wOJmrUE.