Final Report

ArduinoCade

Kushagra Pant & Charlie Rickards

14th June, 2024

TEJ3M1-02

Table of Contents

Note: Page numbers refer to the number at the bottom right

Overview

ArduinoCade (pictured right) is a portmanteau of “Arduino” and “Arcade” named as it is a classic arcade machine, but built with an Arduino Uno. The ArduinoCade uses two 8x8 LED dot matrices to form a 16x8 LED screen. On the ArduinoCade there are two games, Dot Defenders, based on the classic arcade game Space Invaders and Slither based on the more modern arcade style game Snake. In Dot Defenders, you control a small ship on the bottom of the screen which you move around and shoot bullets to defeat an oncoming wave of enemies. As you defeat more enemies, the enemies become faster and faster, making stopping the last few extra hard. As the rounds progress, the enemies get faster so the waves become harder to clear until you eventually lose. This game comes in with a built-in score counter which displays your score and the system’s high score after you are defeated. is Slither, you are a snake and you move using a joystick. The goal is to eat as many apples as possible, the catch being after eating each apple you get longer. The apples spawn at a random place on screen, and after you eat one a new one spawns in. If you run into yourself you die and must try again. You can also run off the edge of the screen, if you do so you will loop around and reappear on the other side, which you can use to your advantage to avoid yourself and eat more apples, and grow longer.

The games are originally separate. Our improvement was combining and syncing the games, as well as adding a home screen which, upon death in either game, the user will go back to. This is important as it puts both games into the same program, allowing them to be accessed simply using the controls given to them without access to the arduino.

Attached Here is a link to a video, showing off and explaining the finished product.

Research

Inspiration

Our very first idea was to make a Zoom Cube like device, a kids game where you try to hit the flashing light before the time runs out, with the times progressively getting faster. However, there was no existing code for this on Arduino, so we'd have to code it from scratch, which we felt would take too much time, so the idea was eventually scrapped.

We decided to settle on modifying a retro arcade game of which there was already Arduino code online to handle the logic. Arcade games are still fun and relevant to this day, and putting them on an 8x16 screen wouldn't compromise their quality, which is why we decided to go with this idea.

Modifications to Proposal

The original proposal we made simply suggested that we add a score counter to the Space Invader game. However, upon further consideration, we decided that it was too simple yet tedious of a task to do. This was because the source code already came with a mechanism to total the score, and there were also 10 pins on a score counter, which was a lot of connections we'd need to make. Instead, we decided to use our knowledge of both software and hardware to combine two retro games and combine them into a single device using an LED Matrix.

The two games we decided to combine were the aforementioned Space Invaders (based on Enjoy Mechatronics) as well as Snake (based on Davide Gatti). We decided to call our version of Space Invaders "Dot Defenders," and our version of Snake "Slither."

Other Sources

We also analyzed the code of the Arduino Labs, specifically Project 4 to help us understand how buttons work, Project 10 to help us understand how the Serial monitor works, and Project 17 to understand how the Dot Matrix works.

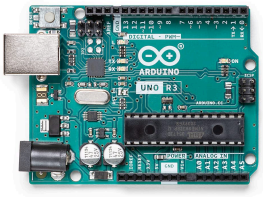
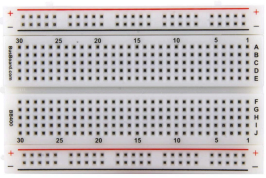
Safety Information

The ArduinoCade is a perfectly safe user experience, however to minimize risk please follow the following guidelines:

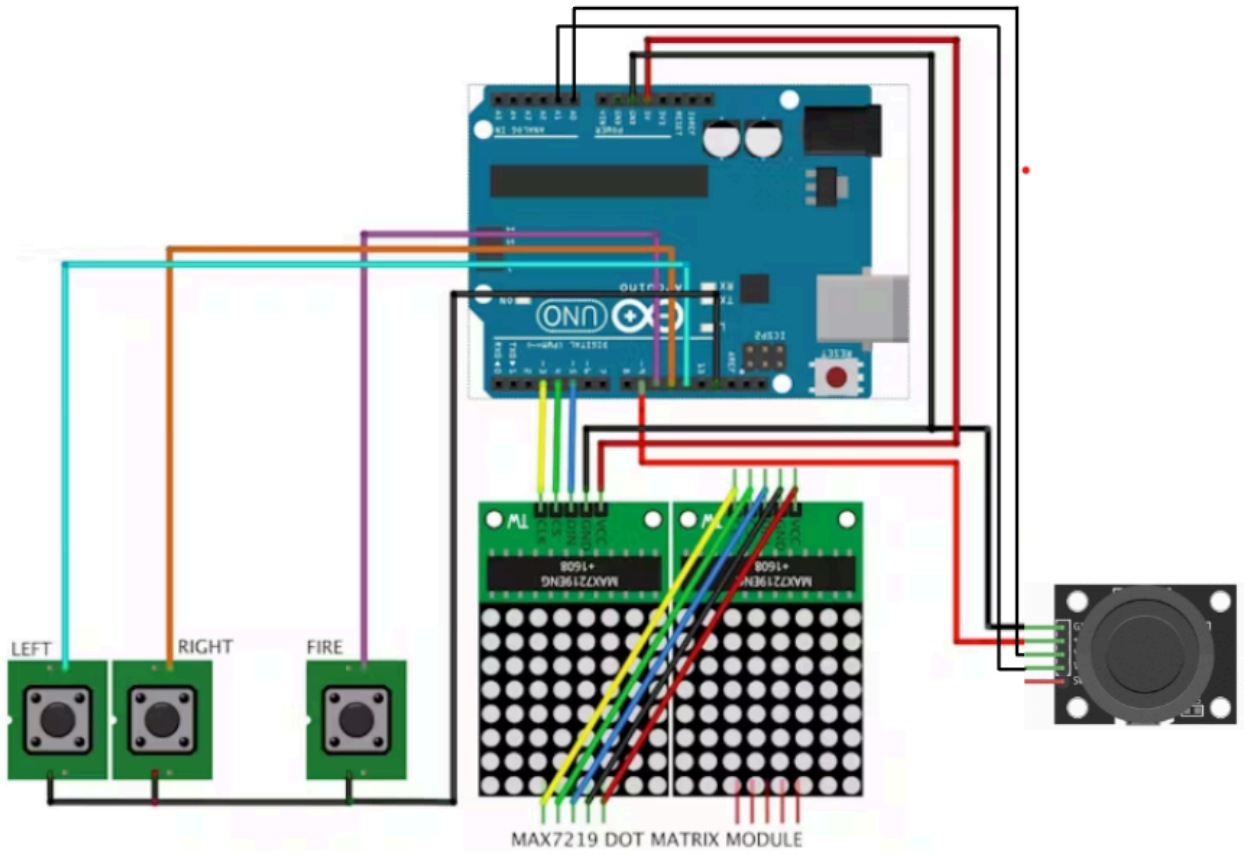
- Avoid touching the wiring or circuit components, especially when they are on
- Avoid consuming any of the components of the ArduinoCade, as they are inedible
- Avoid placing the ArduinoCade anywhere it could be tripped over
- Avoid using the ArduinoCade for anything other than its intended purpose of playing classic arcade games

List of Electronic Parts

This only contains the parts that are part of the device electronically, not for example the materials used to construct the physical box.

 Arduino Uno R3 x1	 M to F Jumper Wires x5	 M to M Jumper Wires x23
 F to F Jumper Wires x10	 LED Dot Matrix x2	 USB 2.0 AB x1
 Solderless Breadboard x1	 2 Pin Tactile Button x3	 Joystick x1

Circuit Diagram



User Instructions

Home Instructions

Use the ◂ ▸ buttons to select your game of choice.

The **left** side is Slither, while the **right** side is Dot Defenders.

Click the ▲ button to enter the game.

Slither Instructions

In Slither, use the joystick to control your snake. The direction you move your joystick in is the same direction the snake will turn.

Try to eat as many fruits as you can!

You die when you run into yourself.

Upon death you will return to the home screen, upon which you have to follow the Home Instructions.

Dot Defender Instructions

In Dot Defenders, use the ◂ ▸ buttons to move your ship left and right, and the ▲ button to shoot

Try and destroy as many aliens as you can before they reach you!

You die when the enemies reach the bottom 2 rows.

Upon death you will see a scoreboard. The first score, without the underline, is the score you achieved in the round you just played. The second, underlined score, is the all-time high score.

After viewing your score, press ▲ to return to the home screen, upon which you have to follow the Home Instructions.

Breakdown of Code

The two games code have each been taken from two separate sites, then stitched together using our group's knowledge of code.

- Slither was derived from Enjoy Mechatronics
- Dot Defenders was derived from Davide Gatti

Below will be segments explaining each section of the code. The full, annotated code can be found here.

Fields and Constructors

All of the fields are initialized at the start of the file. First the ones that are used in both games are declared, with the arrays used by both ending this segment, then the ones that are just used in Slither are declared, followed by the ones just used in Dot Defenders. It concludes with the arrays used in Dot Defenders.

After this, the constructors are declared. Ending the segment, a Snake and Apple object are created, which are used in the Slither game.

Original Methods

This section contains methods written entirely by us. This includes the homeScreen(), setup(), and loop() methods. The setup() method is used to start the LED Matrices up, set their brightness, start up the Console, and set each of the inputs to their respective modes. The homeScreen() method lights up the 2 LED Matrices according to the logos of each game. The loop() method serves as the main execution cycle. It calls homeScreen(), allowing the user to choose either between Slither or Dot Defender. It also calls the respective game method for when a game is selected, and sets the screen back to the home screen when the game is over. A more in-depth, line by line explanation can be found in the code document.

Slither Methods

These methods are used in the Slither game, and include the SnakeGame(), calculateDeltaTime(), Update(), Render(), and removeFirst() methods. They were primarily sourced from Davide Gatti's code, with a major change being the SnakeGame() function. Since originally, the game had its own setup() and loop(), as it was a stand-alone game, we had to change this. So we made SnakeGame(), a function with a while loop cutting it in half. Before the while loop is the original setup() function, and within the while loop is the original loop() function. Another change made is that, upon dying, instead of immediately resetting, the game resets back to the home screen by exiting the while loop.

Dot Defender Methods

These methods are used in the Dot Defender game, and include the methods DotDefenders(), EEPROMReadlong(long address), EEPROMWritelong(int address, long value), MainScreen(), CalcMaxAlien(), NextLevel(), GameInit(), PrintCannon(byte pos), PrintAliens(byte x, byte y,byte ox, byte oy), GameOver(), ShowScore(), SetLed(byte X,byte Y, boolean OnOff), and PrintNumber(long num, boolean underline). Similarly to Slither, the game had its own setup() and loop() methods, so we made DotDefenders() containing the former setup() method, with a while loop containing the former loop() method. Upon death, it exits the while loop, ending the game and sending the user back to the home screen.

Evaluation/Reflection

As mentioned above our very first idea was to make a Zoom Cube like device. However, after coming to some problems with discovering that there was no existing code online for us to improve on, as well as our own lack of 3d printing experience (which would have been required to make the cube) we eventually decided on making an arcade machine.

Our original plan was to make Space Invaders and Atari Breakout as the two games on the ArduinoCade, using this code for Breakout. Unfortunately, we were unable to synchronize the two together in time, as the Breakout used multiple files which was very difficult to make compatible using our level of programming knowledge. So we had to make a last-minute switch to a different game once it was clear that Breakout wouldn't work. These things are impossible to plan for, and only become apparent as we build the project to see what works. Overall, we feel that we did the project to the best of our ability in the given time frame.

We learned a lot from this experience, not only the obvious like how to program better in Arduino IDE and build real-world devices, but also to plan realistically and not be afraid to try new things. We were so persistent on doing Breakout until very late, but had we tried new games such as Snake sooner and not tried to beat the dead horse, we could have reduced the burden of the task.

It also taught us to do more extensive research before carrying out tasks. We spent the first week coding the logic behind Dot Defenders, which can be found here, but it wasn't very compatible with the Arduino as that uses a modified version of C++, plus we didn't familiarize ourselves with the methods of the LED Matrix. Then, after researching a bit more, we found out that we could've just used an already coded Arduino-friendly version of the game, which would have saved us time. All in all, had we done extensive research we could have recovered a significant amount of time and effort.

References

1. Gatti, D. (2022, January 14). *Space Hinvaders*. Hackster.io. https://www.hackster.io/davide-gatti/space-hinvaders-c48731#code
2. Enjoy-Mechatronics. (2021, July 27). *Arduino-snake-game* https://github.com/Enjoy-Mechatronics/Arduino-Snake-Game/blob/main/Arduino-Snake-Game.ino