

# Final Report

# ArduinoCade Lite

---

Kushagra Pant & Charlie Rickards

13th June, 2025

TEJ4M1-02

---

# Table of Contents

Last Year's Project .....	2
Implementing Improvements .....	3
Safety Information .....	4
List of Parts .....	5
Circuit Diagram .....	6
User Instructions .....	7
Breakdown of Code .....	9
Evaluation/Reflection .....	12
References .....	14

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

# Inspiration - Last Year's Project

The 2024 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. In 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 our old finished product.

# Implementing Improvements

The 2025 ArduinoCade “Lite” is an improved version of the original ArduinoCade. A video of the product can be found [here](#). The following are improvements that were implemented.

## New Game: Dodgers

To Start, a new game called Dodgers was added, the objective of which was to dodge the incoming bullets from the right, while only being able to move up and down. As the game progresses, the speed of the bullets increases. This game was synced to the rest of the project.

## Updated Home Menu

Since there are now 3 games, the home screen menu was updated to be scrollable. When loading up the ArduinoCade Lite, the first 2 games show up (Slither and Dot Defenders). Then, Dodgers showed up if right is continuously pressed (a.k.a. scrolling to the right). Slither reappears if scrolled to the left.

## Removed buttons, Changed All Controls to Joystick

The original game had one game with button controls and another with joystick controls. This was inconsistent for the players of our game, and so we decided to make the decision to make all games use Joystick. For this, the code of Dot Defenders and Dodgers had to be modified to use joystick input rather than button input. Additionally, the menu had to be changed to using the joystick, including its click which would serve as the select button.

## Updated Slither to use 2 screens

In the original game, Slither only used one screen. We updated our game to allow Slither to take advantage of the multiple screen layout we had by expanding it to a 16x8 map.

---

## Scoring for all Games

We also added a high score for every game. Previously, a high score only existed for Dot Defenders, but this was expanded in ArduinoCade Lite. To do this, EEPROM was expanded to store 12 characters instead of 4 (for the all-time high scores for each game), and the ShowFinalScore() method was expanded to accommodate all games, including passing in two parameters.

## Portability

Last but certainly not least, we changed the game to allow it to be portable. We did this by replacing the USB 2.0 AB with a 9 Volt Battery + Arduino Clip. We used a phone box to hold all the components, carving holes for the screens and joystick. This way the game is portable and handy.

## Safety Information

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

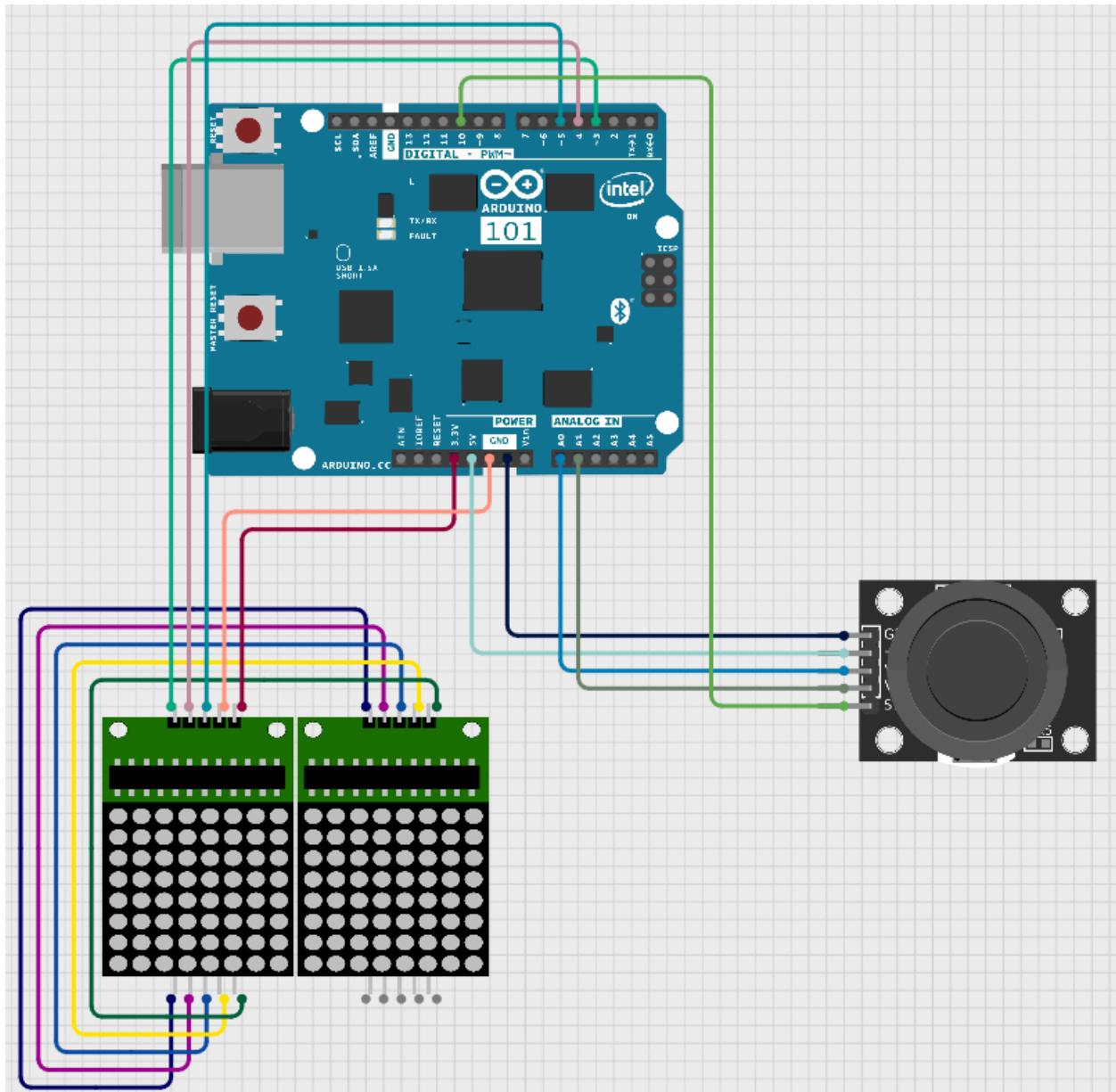
- Avoid touching the wiring or circuit components, especially when they are on
- Avoid consuming any of the components of the ArduinoCade, as they are inedible
- Avoid placing the ArduinoCade anywhere it could be tripped over
- Avoid using the ArduinoCade for anything other than its intended purpose 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	 Joystick x1	 M to M Jumper Wires x10
 F to F Jumper Wires x15	 LED Dot Matrix x2	 9 Volt Battery + Arduino Clip x1

# Circuit Diagram



---

# User Instructions

## Home Instructions

Use the joystick to select your game of choice.

The left side is slither, the middle is Dot Defenders and the right is Dodgers.

Click down the joystick to enter the chosen 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 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, click the joystick to return to the home screen, upon which you have to follow the Home Instructions.

## Dot Defender Instructions

In Dot Defenders, use the joystick to move your ship left and right, click the joystick 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, click the joystick to return to the home screen, upon which you have to follow the Home Instructions.

---

## Dodgers Instructions

In Dodgers you use the joystick to move up and down.

Try to dodge the incoming projectiles and survive for as long as possible!

You die when you are hit by a projectile

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, click the joystick to return to the home screen, upon which you have to follow the Home Instructions.

# Breakdown of Code

The code of the three games have each been taken from three 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](#)
- [Dodgers was derived from Joshua Weßner](#)

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 all 3 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, concluding with the variables used in Dodgers. The purpose of each variable is written in the code.

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.

## Dodger Methods

There are only two methods in Dodger, which are dodger() and dodgerLoop(). These were the former setup() and loop() methods of the independent game created by Joshua Weßner. Despite only having two methods, the code for these methods are incredibly comprehensive, as they contain all the inner workings of the Dodger Game. The original game had many additional methods and fields, mostly for sound and displaying the numbers for the highscores. However, we did not need this, as we would integrate the scoring system of Dodgers with the displaying score system of Dot Defenders.

---

## Code for Scoring

An upgrade to the game that we made was implementing highscores for each game. The ShowFinalScore() method was updated to allow for any score and highscore to be displayed rather than just Dot Defender's. Most of the same code from the original highscores was reused, but the score and highscore methods were changed from the global variables of Dot Defenders to parameters that would be passed through the method. Additionally, more global variables were added to keep track of the highest scores of Slither and Dodgers. As well, EEPROM was changed from storing 4 characters to 12 characters. From 0 to 3 is the high score of Slither, from 4 to 7 is the high score for Dot Defenders, and from 8 to 11 is the score for Dodgers.

## Evaluation/Reflection

As mentioned above our plan for this project was to improve on our invention from last year. We enjoyed the challenge of our project last year and were very happy with the end result, so we wanted to see how our new skills could improve it.

We set out with 2 goals for how to improve on the original ArduinoCade: First we wanted to add some new games to allow for more variety and a more enjoyable playing experience. Second we wanted to refine the controls to allow them to all use the joystick instead of being split across different inputs for each game.

Adding new games required us to find a new project we were interested in adding, which led us to [flappy bird](#). We spent a few days trying to implement this game into our system, but unfortunately were unable to. This is because the flappy bird code uses the LEDMatrixDriver library to control the dot matrices instead of the LedControl library we use for our other games. After realizing this we decided to implement a different game instead of trying to recode the game we had found to use our LedControl system. This led us to asteroids, which we modified to use the joystick, and implemented into our game system.

Switching our games to all use the joystick was not a difficult task, but turned out to be the most important for our final product. It required a few modifications to the code, and a small bit of rewiring, that once finished allowed us to skip the need for a breadboard. No longer needing a breadboard our project took up much less space which gave us the idea to make our new game portable, which was the birth of the ArduinoCade Lite. We were able to make a box to store it in and hook the arduino to battery power allowing it to run without needing to be plugged into anything, and the box had a form factor that made it easy to carry around while you played.

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. We were so persistent on doing Flappy Bird until very late, but had we tried new games such as Asteroids sooner and not tried to beat the dead horse, we could have reduced the burden of the task.

It also taught us to not be afraid to try new things and see where it takes you. After rewiring everything to run through the joystick, we only had 2 wires plugged into our breadboard,

---

when we had the idea to try and get rid of the need for it all together. Rewiring so the screens and joystick power are both plugged directly into the arduino instead of joining on the breadboard allowed us to do this and lead to the idea to make our system portable.

All in all, we believe we did a good job creating a new and improved version of the ArduinoCade that maintains the feel of the original, but still brings new things to the table.

## 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>
3. kreck2003. (2018, February 4). MAX72XX LED matrix display asteroids game [Arduino Project Hub]. Arduino.  
<https://projecthub.arduino.cc/kreck2003/max72xx-led-matrix-display-asteroids-game-fc085b>