

Mini ‘Follow & Fool’ Arcade (Arduino Project)

Project Information

1. Team Members

Isaac Sanchez

Sebastian Ho

2. Project Title

Mini Follow & Fool Arcade (originally titled “Mini Simon Says Arcade”)

3. Project Abstract

A small arcade-like machine where two players can play a simple game, similar to the Simon Says children’s game where players must follow the commands of the CPU-controlled leader character, using buttons and a joystick. The game is played through one controller Arduino, and visually displayed through a display Arduino connected to an LCD screen, with a buzzer attached to the display as well.

4. System Description

The purpose of this system is to simulate an arcade-style environment that’s more portable than normal arcade cabinets. The system is meant to host a simple game that plays like Simon Says, with two players competing against each other. One Arduino acts as the visual display and emits the game sounds. The other Arduino acts as the controller for both players.

Design

5. Description of Subsystems

The first subsystem is titled “display”. This subsystem displays the game’s visuals through the LCD screen and plays sounds depending on the game state.

The second subsystem is titled “controller”. This subsystem allows two players to control the game using three buttons and a joystick for each player.

6. Description of Input and Output Devices per Subsystem

The Display Output Devices

Device	Behavior	Output
TFT Screen	While the game is operating, the TFT screen shows the current state	Graphics

	of the game to the players.	
buzzer	When the leader says to perform an action, when a player wins a game, or when a tie game occurs, the buzzer will play a small sound effect or, in the case of a player winning, a jingle.	Sound/Music

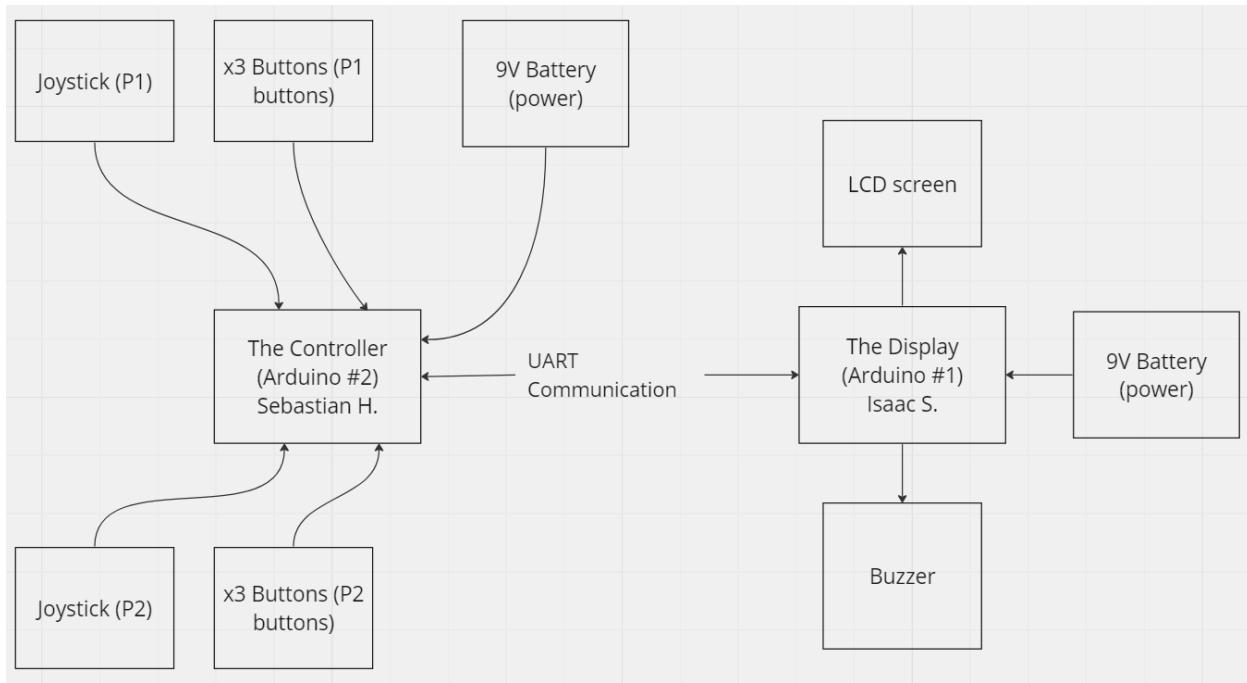
The Controller Input Devices

Device	Behavior	Input
Joystick	When the leader says to input an arrow direction, each player will use the joystick to input an arrow direction to match the leader.	Player Input
Button	When the leader says to use a button, the player will use the left button to match the leader. The middle button will be used to fool the other player with the leader displaying an incorrect action. The right button will be used to send the player's last input to the display.	Player Input

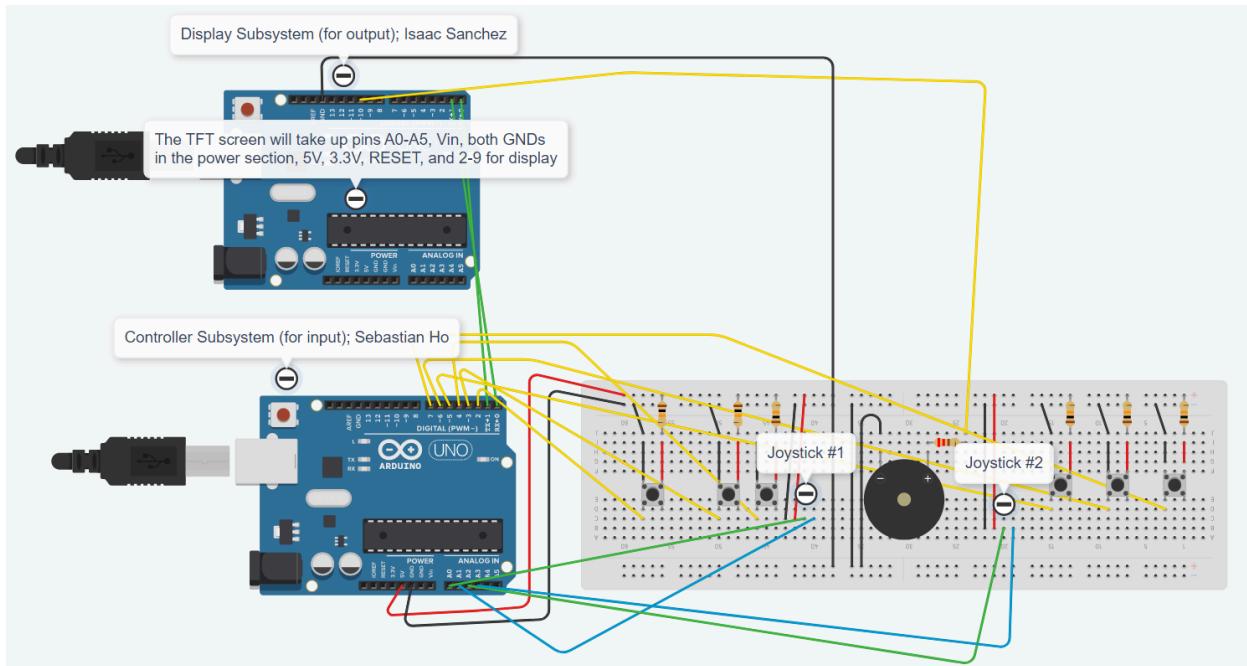
7. Communication Mechanism

We used UART serial communication between the two systems. The right button on the controller is used by players to send their last input to the display Arduino. The display Arduino checks for a message that's sent for every iteration of the loop function. If there are any available bytes, the display will process the input and progress the game accordingly. Each input a player can make is coded into a unique byte message which tells the display what command was received.

8. Block Diagram of the System (Big Picture Diagram)



9. Hardware Diagram



Guide to Building Project

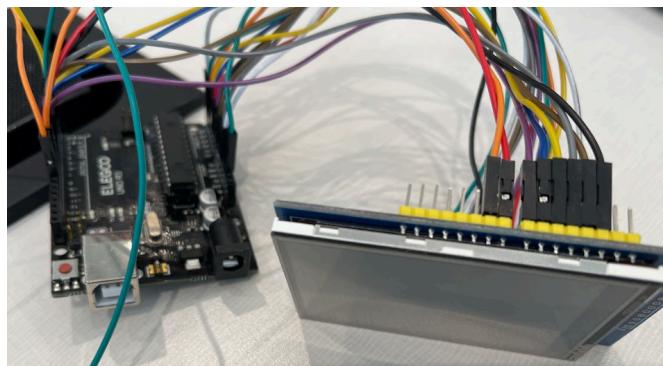
10. List of Materials

2x Arduino Uno (or equivalent)
6x Button
6x 10k Ohm Resistor
2x Joystick
1x Piezo Buzzer
1x 220 (or higher) Ohm Resistor
1x TFT Screen
Jumper Wires (male-to-female & male-to-male)
2x Breadboard

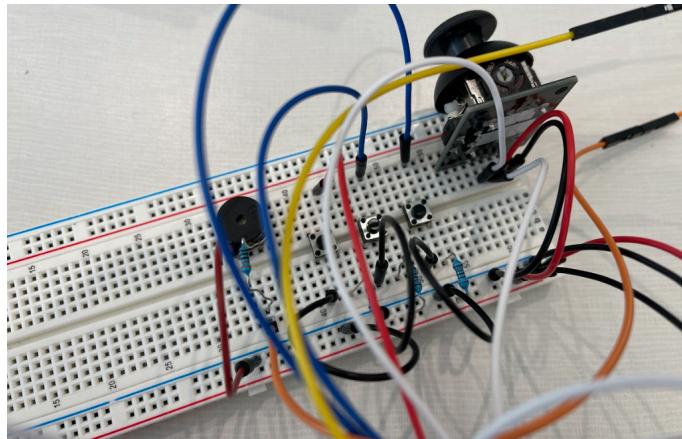
11. Steps to Build Project

* Use the hardware diagram as an example to help you build the project.
The display unit needs an Arduino Uno, a TFT Screen and a buzzer w/ 220 ohm resistor.

1. Insert the ends of two male-to-male wires into the TX and RX ports of the display Arduino.
2. Connect the remaining pins of the Arduino Uno to the TFT Screen using male-to-female jumper wires (this is done instead of inserting the TFT screen directly into the Arduino Uno so there's still access to the TX/RX ports for UART serial communication):
 - a. A5, A4, A3, A2, A1, A0
 - b. Vin
 - c. Both GND pins on the power section of the Arduino
 - d. 5V
 - e. 3.3V
 - f. RESET
 - g. Digital pins 2 to 9
3. Connect the free ends of the Display Arduino's TX and RX jumper wires into the opposite pin inserts of the Controller Arduino; TX connects with RX and vice versa.

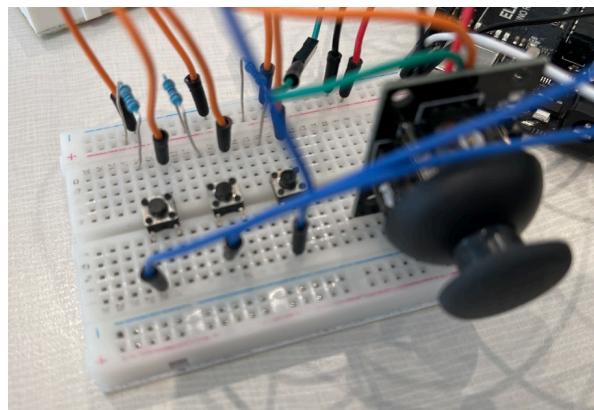


4. Place a piezo buzzer on a breadboard and connect it (with a 220 ohm or higher resistor in-between) to pin 10 and GND on the display Arduino. We placed the buzzer together with one of the player's controller parts on the same breadboard.



The controller unit is built with the remaining parts (with the breadboard that's housing the buzzer also housing one of the player's input components).

1. Connect three of the buttons to the controller Arduino Uno's 2, 3, and 4 pins, which are player one's left, middle, and right buttons, and connect three more buttons to pins 5, 6, and 7 for player two. Use a 10k Ohm resistor to complete the circuit for each button.
2. Connect the x-direction (VRx) of one joystick to A0 and y-direction (VRy) to A1 for player one (using male-to-male jumper wires). Then connect the x- and y-direction of another joystick to A2 and A3 respectively, for player two's directional input. NOTE: for adequate spacing of parts, one player's input components should be on the unused breadboard, and the other player's input components should be on the breadboard with the buzzer attached (unless you have a larger breadboard available to house more parts comfortably).



3. If not already done, connect the TX pin from the display arduino into the RX pin in the controller arduino. Then, connect the RX pin from the display arduino into the TX pin in the controller arduino.
4. Download the "controller" and "display" files found in the "Group34FollowAndFool" folder on [this github repo](#) and upload them to the corresponding controller and display Arduinos. The display project code requires "Adafruit_GFX.h" and "MCUFRIEND_kbv"

library files to be downloaded, through [the Arduino IDE's "Manage Libraries..." section](#) found in "Tools".

12. User Guide

The Mini Follow & Fool Arcade is a system that allows two players to compete against each other in a competitive game where players must follow the directions of the leader (in the middle of the screen) each round. The goal of the game is to last as many rounds as possible until one player runs out of lives.

Player input is not sent unless the right button is pressed. Players will need to make their input decision and send their input using the right button.

Before the start of each match, a ready up screen will show up. Each player will use the right button on the controller to ready up. A green 'R' will show up for a player that is ready, and when both players are ready, the game will start.



Every round, the leader will show a directional arrow or a circle to represent the move that the player will need to make. The arrow indicates that each player needs to move their joystick in the same direction as the arrow to send that input to the game with the right button and win the round, or fail and lose a life for that round.



A player can use the joystick to match the direction of output that the leader requests.



A player can press the middle button on their controller to make the leader perform a fake signal in a slightly different color than the leader's normal color. This is used to fool the other player into sending an erroneous input (which will make them lose a life).



At the end of each round, the game checks whether one or both players lost a life due to being unable to send the leader a correct command in time. Once a player makes a correct input, the other player can't send any more inputs, and that other player will lose a life at the end of the round. Once one or both players run out of lives, the game is over. The player who lasts the longest will win, or the game will end in a tie. Only one fake signal can be used per player, per game. After a player has won or the result is a tie, the game will return to the title screen and remain at the ready up screen for the next game.

13. Video of Final Project

<https://youtu.be/lIm0uMbzNAs>

Planning

14. Distribution of Work and Team Management

The experience of working as a team was beneficial for building a complex device involving both hardware and software. We met many times both in-person and online, which kept us on track with project deliverables. Overall, we're satisfied with the final result: a fun, competitive, arcade-style game.

We could have met more in-person for more testing or working on the project together. We could have also got some more work done independently before each of our meetings so that we could fix any issues that happened in the time that we worked independently.

Conclusions

15. Conclusions and Lessons Learned

I think we could have chosen different hardware at the start to proceed with (the LCD screen has a slow refresh rate). We should have also thought about more game ideas and done other side projects or tests with communication before deciding on the project's game. We should have also prioritized office hours assistance with issues that gave us a lot of trouble.

We could have also used actual arcade buttons and a joystick. Or we could have also made a casing for the project. But due to budget and time constraints we couldn't add them to our project.

We encountered issues having the Arduinos communicate with each other. We overcame this by attending office hours and making a piazza post to get additional help from some of the TAs. We also had an issue with buttons not working properly or not detecting input. This was caused by an improper wiring of each of the buttons and we were able to figure it out by going to office hours.

References and Attribution

16. Similar Projects (and why our Project is Unique)

The following projects only use one Arduino system, while our project has two Arduinos communicating with each other.

<https://maker.pro/arduino/projects/diy-handheld-game-console-with-arduino>

This project not only focuses on single-person play, but it lacks the joystick implementation that our project has. It uses a 6-button input design, where 4 of the buttons are used for directional input instead of using a joystick (Watson, 2023). This project uses a buzzer, similar to our project. The LCD screen is smaller than what we are using. This project also uses an Arduino Leonardo instead of an Arduino Uno. Of course, this system and our system are similar in that they are both portable gaming systems.

https://www.hackster.io/Arnov_Sharma_makes/arduino-game-controller-v2-15ee8c

This project lacks any built-in video games. In addition, this project uses its own custom PCB board for connections, as well as using a different Arduino board entirely, as they were using the SparkFun Pro Micro (Sharma, 2022).

<https://www.instructables.com/Arduino-Tamagotchi-Clone-Digital-Pet/>

Not only is this project meant for a single player, but it contains a game that is different from our own game that will be played in the mini arcade. This system uses the Arduino Pro Mini instead of the Arduino Uno (Koumaris, 2023).

17. Statement of Originality

Our project implements 2 Arduino subsystems, while most other implementations we've seen of game systems with Arduinos use only 1 subsystem. In addition, some projects that implement the controller subsystem, specifically the *Arduino Game Controller V2* by Arnov Sharma, do not implement the display subsystem. Finally, while other projects prioritize single player engagement with the system, our system will prioritize multiplayer engagement. For the gameplay, while the game is similar to Simon Says, it differs significantly due to the two-player aspect and the ability to fool another player by using the "false" signal with the middle button.

18. References

1. *From mind to design in minutes*. Tinkercad. (n.d.). Retrieved March 28, 2023, from <https://www.tinkercad.com/>
2. Koumaris, N. (2022, August 4). *Arduino Tamagotchi clone - digital pet*. Instructables. Retrieved February 17, 2023, from <https://www.instructables.com/Arduino-Tamagotchi-Clone-Digital-Pet/>
3. Miro. (n.d.). *Sign in: Miro: Online whiteboard for visual collaboration*. Miro. Retrieved February 17, 2023, from <https://miro.com/login/>
4. Sharma, A. (2022, December 11). *Arduino Game Controller V2*. Hackster.io. Retrieved February 17, 2023, from https://www.hackster.io/Arnov_Sharma_makes/arduino-game-controller-v2-15ee8c
5. Watson, R. (2023, February 17). *DIY handheld game console with an Arduino Leonardo: Arduino*. Maker Pro. Retrieved February 17, 2023, from <https://maker.pro/arduino/projects/diy-handheld-game-console-with-arduino>