ELEC 3907 Group 2H

Project Proposal: Arcade-Style Game Console

February 1, 2026

Jason Alexander 101309867
Jason Alexander 101309867
Jason Alexander 101309867
Jason Alexander 101309867
Jason Alexander 101309867

Contents

1 Project Overview **1**

2 Design **1**
2.1 System-Level Description 1
2.1.1 The Hardware 1
2.1.2 The Software 1
2.2 Preliminary Specifications and Requirements 2
2.2.1 The Outputs 2
2.2.2 The Inputs 2
2.3 Design Approaches 3
2.3.1 The LED Matrix 3
2.3.2 The Game Controllers 3
2.3.3 The Divider 3
2.3.4 The Software 4
2.4 Integration 4

3 Risk and Safety Assessment **4**
3.1 Power delivery 4
3.2 Heat 4

4 Sustainability and Lifecycle **5**

5 Team Roles and Responsibilities **5**

6 Project Timing **6**

7 Conclusion **6**

A Supplemental materials **7**

1 Project Overview

The goal of this project is to provide a compact arcade-style game console designed to restore hands-on interaction commonly found in traditional arcade systems. The console would use a two-dimensional LED grid as the primary display as well as physical input devices and near-field audio output. It is designed to run multiple classic games such as Tron, Battleship, and Snake, all using a multipurpose hardware layout and modular software design.

2 Design

2.1 System-Level Description

The proposed console would be a game system where three different games can be played using a simple display and physical controls. It does not require an internet connection and can be played anywhere there is a source of power. It provides a compact, physical, and portable version of several arcade games such that people of all ages can enjoy using its simplistic controls and peripherals. Additionally, more games can be implemented and included on-board, allowing for different people with diverse preferences to enjoy the console.

2.1.1 The Hardware

The console itself (seen in Fig. 1) consists of multiple subsystems, including a 16x16 LED display with a divider, where each of the 256 pixels of the display can be individually addressed to display an almost arbitrary number of colours. There are two LCD screens for game selection, acting as a menu for players to interface with the console. These LCD screens will also act as instructional screens to guide players through the games. There are two removable and adjustable controllers, where the controller can be plugged into either side of the console to accommodate right- or left-handedness or for single and multiplayer options. The speaker provides sound effects for audio feedback and soundscapes during gameplay. There is also a freestanding divider that will be used during games that require the players' boards to be hidden from each other during gameplay—as in the case in battleship, for example. A microcontroller is programmed with the game logic, interfacing with the other components to provide a seamless experience while using the console. An additional microcontroller listens for audio playback cues, and drives a speaker accordingly, allowing for ansynchronous audio output. Finally, the power supply powers the display and the microcontrollers, the first of the latter of which subsequently powers the other components.

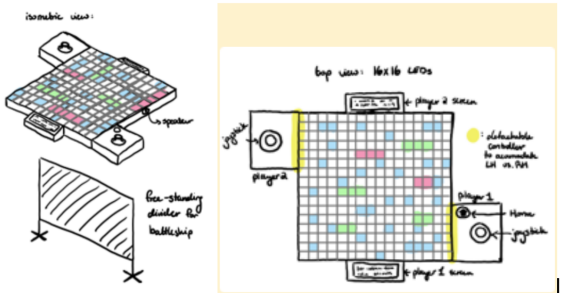


Figure 1: Orthographic views of the physical design

2.1.2 The Software

The software system must be able to interface with all the components outlined, taking input from the game controllers, controlling the LED matrix, using the LCD displays to relay information to each player, and playing audio. The system is constrained by the memory limitations of both microcontrollers, and the fact they are both single-threaded. To overcome scheduling issues, especially with continuous-delivery applications like audio playback, the second microcontroller is dedicated to receiving playback signals from the first microcontroller and controlling the speaker continuously. This second microcontroller can be much less powerful than the main one, essentially serving as an additional thread for the console, allowing it to avoid audio cut-outs and maintain responsiveness of the console proper.

Additionally, since the hardware system is designed to be general-purpose, we have decided to leverage object-oriented programming techniques in C++ to model the hardware system and allow for composability. The hardware-software interface is designed in such a way that basic operations like controlling the LED matrix can be inherited and extended to implement game-specific behaviours, reducing unnecessary boilerplate code and further memory usage. Fig. 2 provides a block diagram outlining the general control flow of the system. Most elements included in this figure will have some class representation in software, allowing for simple control of components through their corresponding class methods.

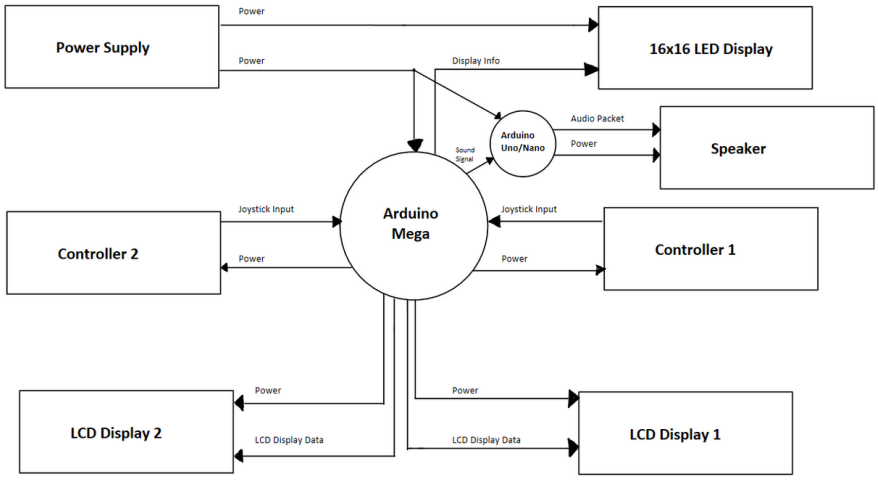


Figure 2: Block diagram of the system

2.2 Preliminary Specifications and Requirements

2.2.1 The Outputs

The display will be a 16x16 grid of LEDs which will server as the main display for the game console. An addressable GRB LED strip will be cut into 16 lines of 16 LEDs, the lines being wired together neatly on a piece of plywood. The LEDs will be controlled by an Arduino Mega and will require a power supply. In total, this component will require 256 addressable LEDs from a WS2812B LED strip, an Arduino Mega, a power supply, and physical interconnections—presumably wires, solder, etc. The enclosure will also need a 1x1 foot piece of thin plywood to serve as a backboard, a styrofoam sheet to make a grid separating the LEDs, and frosted acrylic glass to diffuse light. Most of these items can be found in the lab, with others being provided by group members. The only items to be purchased and included in our costs will be the LED strip at around $25, the plywood between $15 and $30, the styrofoam sheet around $2, and the acrylic glass around $5. The audio output will consist of two small speakers that were available in the lab, so they resulted in no additional cost.

2.2.2 The Inputs

The user inputs—one removable controller per player—will each consist of a joystick fitted into a block that can be attached to the console via interlocking rails. The controller would have metal contacts that, when slid into corresponding rails on the console, would establish a connection with the Arduino Mega. The rail would likely consist of guides that slide together to hold the controller firmly in place without too much stress on the contacts, but enough pressure to ensure a reliable connection. The materials required have been provided by group members or will be acquired when gathering the materials for housing the LED display.

2.3 Design Approaches

2.3.1 The LED Matrix

1. Test the LED strip to ensure the individual LEDs work and are addresable with the Arduino Mega
2. Measure the distance between LEDs to create a foam lattice reducing interference between pixels
3. Wire the 16 LED strip segments to form a uniformly-spaced pixel matrix on the plywood base
4. Test that the LEDs are still working and addressable with the Arduino
5. Construct the enclosure (styrofoam grid and acrylic glass)
6. Add ports for connection and ventilation
7. Install fans if needed

If, for instance, we are unable to control the LED matrix, an alternative last-minute solution would be to buy a premade LED matrix to server as the display.

2.3.2 The Game Controllers

1. Test the joysticks, ensuring they communicate appropriately with the Arduino Mega
2. Determine dimensions for the controller based on the LED matrix's size
3. Contruct the controller housing that that will hold the joystick and connection terminals
4. Outfit the LED enclosure with contacts for controller interfacing
5. Install guide rails to slot the controller onto the display encloser
6. Test the controller maintains consistent connection with the Arduino

If there are technical or logistical difficulties with this approach, the controllers could be hard-wired to the LED display and affixed using other methods (screws, adhesive, etc.) to ensure working inputs for the system.

2.3.3 The Divider

1. Test variously sized dividers made of a temporary material like cardboard to determine the ideal dimensions. No player should have visibility of the other player's screen with the divider in place, but it should also be lightweight and relatively small.
2. Construct the divider with plywood
3. Add freestanding supports that hold the divider upright

If it is a challenge to make the divider stand upright, a slot can be made in the display housing to help hold the divider in place.

2.3.4 The Software

- Work together to create a generic and composable game board setup, including a virtual pixel matrix, inputs, menus, and audio cues that work for any game.
- Create a virtual representation of the game board for testing purposes when it is not realistic to flash code to the microcontrollers
- Implement the games based on the generic classes written for the game board
- Design music and sound effects for the Arduino Nano to play during games
- Ensure that the microcontroller handling audio correctly interprets audio playback cues from the main microcontroller

The software team will be working mostly independently on separate games, with input and help from each other as is necessary.

If there are problems with implementing working software on the Arduinos, it will likely be due to how memory is allocated. Given that there is a very limited amount of memory on the microcontrollers, alternate methods of allocating resources for each game may need to be considered (static allocation, predictable allocation and de-allocation per-game, etc).

2.4 Integration

The last step in the first iteration of the design process will be to ensure that the microcontrollers control the physical hardware as intended. If encountering problems at this stage, previous design steps will need to be revisited. Any problems at this point will be interfacing issues or limitations imposed by our hardware.

3 Risk and Safety Assessment

3.1 Power delivery

Supplying power to the microcontrollers and LED matrix poses inherent risks. The LED strip selected to build the display from has a considerable maximum power draw due to its individually addressable LEDs and the ability to control the colour of each LED. The recommended power supply from the vendor is 60 Watts (5 Volts at 12 Amps), however initial testing has shown that the actual power draw needed for this project is likely to be much less than that since all the LEDs in the matrix will not be at full load when portions of the display are off and only individual colours (i.e. red, blue, green) are used instead of combinations thereof. Still, supplying the required power to all 256 LEDs in the display will take a considerable amount of power. Risks associated with power delivery can be mitigated by ensuring there is no exposed wiring and implementing additional safety components like fuses to ensure fail-safety.

3.2 Heat

While power delivery has its safety risks, power draw also poses its own. Given that the LEDs and other components will be enclosed in a grid surrounded by foam and wood, the console may begin to heat up over time. If too much heat accumulates, the console may experience performance issue, fail, or even catch fire. If left unattended, the console could become a risk to its surroundings. To avoid these issues, action must be taken to dissipate heat and prevent heat from affecting other components of the game console. If heat becomes an issue, there are several options to manage it such as:

- Ventilation ports in the enclosure to allow for heat to escape
- Fans in the enclosure to allow for air to circulate
- Insulation like heat shrink, electrical tape, etc. to isolate components and prevent shorted connections

4 Sustainability and Lifecycle

The team has prioritized modularity of hardware, and composability of software—things that greatly contribute to the project's overall sustainability and longevity. Modular components like the controllers can be swapped out for new controllers if the existing controllers become damaged or begin to not work. Similarly, individual LEDs used in the display can be swapped out in the case that they begin to malfunction; each pixel of the display can be independently replaced, ensuring that the entire display does not need to be discarded when only a part of it begins to malfunction.

Additionally, since the codebase if built on generic classes controlling the hardware, games can be added further down the line to improve replayability by flashing new programs to the microcontrollers. For sake of argument, if the console were to be a mass-produced product, users can learn to implement their own games as they gain programming experience, thus allowing the product to grow with the user. The basic game board implementation is kept separate from game-specific logic by design, so vital code is protected from accidental changes during experimentation.

5 Team Roles and Responsibilities

The team roles and responsibilities have been split into three main categories: (A) hardware, (B) software, and (C) integration.

A. Marlon and Luc will be focused on hardware, both working on the LED display. Marlon will also focus on power and data delivery and construction of the game controllers, while Luc will work on the display enclosure.

B. Sophia and Jason will focus on software, creating the virtual game board but also working to implement Battleship and Snake respectively.

C. Immanuel will maintain strong communication between the hardware and software teams. He will assist in troubleshooting across the system when issues arise, and ensure the team has the necessary tools and resources during working sessions.

6 Project Timing

The project design cycle will be a standard engineering one. The following work breakdown structure lists all foreseen tasks with major development phases in chronological order. A more in-depth Gantt chart with task designations and progress is listed in Appendix A.

- Initiation
  - Define goals
  - Project outline
  - Project overview presentation
  - Project proposal
  - Project design review presentation
- Planning and design
  - Define scope
  - Identify parts
  - Tabulate cost
  - Acquire parts
- Execution
  - Part testing
  - Software-hardware interface
  - Board development
  - Game development
  - LED screen
  - Housing and divider
  - Game controllers
  - Implementation and testing
- Evaluation
  - Monitor progress
  - Evaluate progress
  - Consider problems
  - Address problems
  - Gather feedback

7 Conclusion

By combining modular hardware and composable software designs, the proposed game console system is able to support multiple classic games whilst remaining flexible with further expansion possible in the future. The proposed approaches balance performance, cost, and safety within the constraints of the microcontrollers. Identified risks, particularly power and thermal control, are mitigated through ventilation, insulation, and protective circuitry. Overall, the design demonstrates a sustainable and scalable system meeting project goals and encouraging creativity and longevity.

Game Console

Group 2H Immanuel, Sophia, Jason, Luc, Marlon

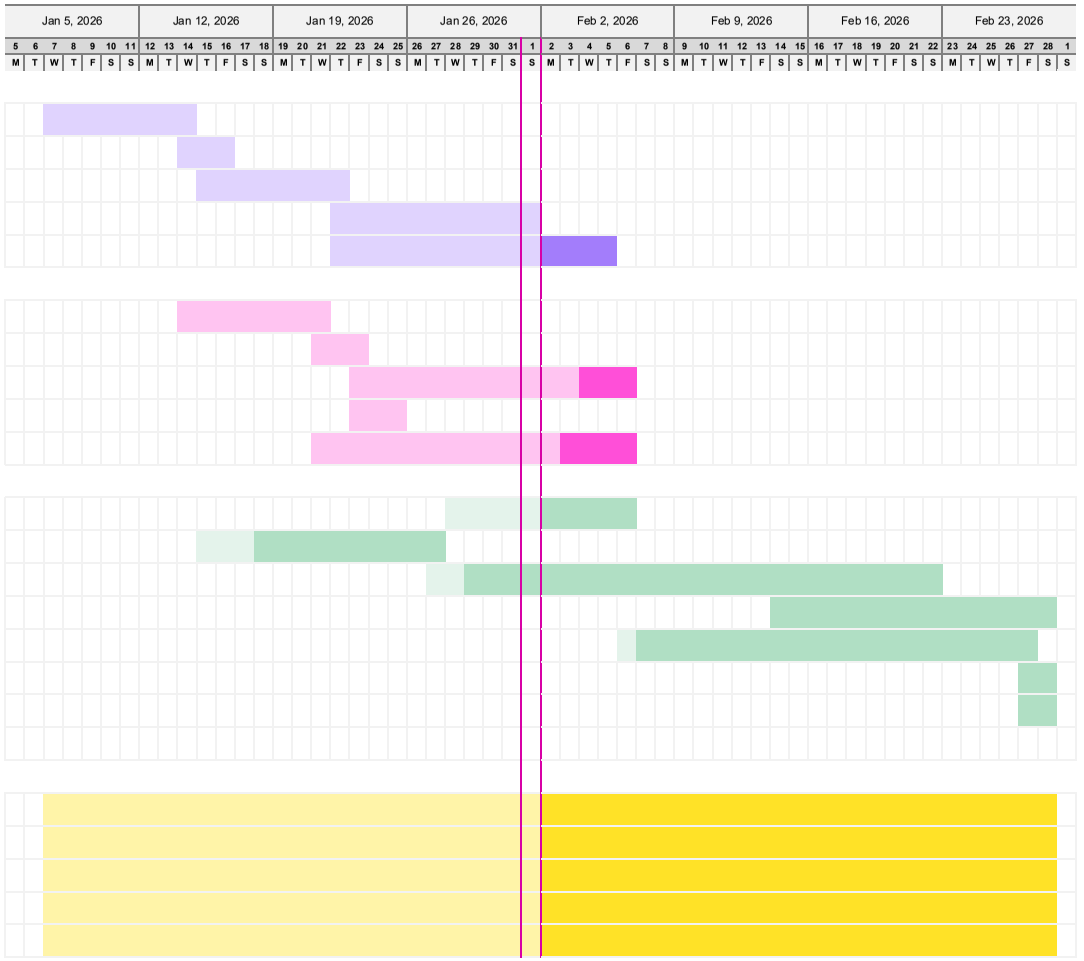
Project start: Wed, 1/7/2026

Display week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Define goals	The Group	100%	1/7/26	1/14/26
Project Outline	The Group	100%	1/14/26	1/16/26
Project Overview Presentation	The Group	100%	1/15/26	1/22/26
Project Proposal	The Group	100%	1/22/26	2/1/26
Project Design Review Presentation	The Group	75%	1/22/26	2/5/26
Planning and design				
Define Scope	The Group	100%	1/14/26	1/21/26
Identify Parts	The Group	100%	1/21/26	1/23/26
Tabulate Cost	The Group	80%	1/23/26	2/6/26
Define Scope	The Group	100%	1/23/26	1/25/26
Aquire Parts	Sophia, Marlon and Luc	80%	1/21/26	2/6/26
Execution				
Part Testing	The Group	50%	1/28/26	2/6/26
Software for Hardware Compone	Sophia and Jason	25%	1/15/26	1/27/26
Board Development	Sophia and Jason	10%	1/27/26	2/22/26
Game Development	Sophia and Jason	0%	2/14/26	3/8/26
LED Screen	Marlon and Luc	5%	2/6/26	2/27/26
Housing and Divider	Immanuel, Marlon and Luc	0%	2/27/26	3/6/26
Controller	Immanuel, Marlon and Luc	0%	2/27/26	3/6/26
Implementation and Testing	The Group	0%	3/6/26	3/20/26
Evaluation				
Monitor progress	Immanuel	36%	1/7/26	3/20/26
Evaluate Process	Immanuel	36%	1/7/26	3/20/26
Consider Problems	The Group	36%	1/7/26	3/20/26
Address Problems	The Group	36%	1/7/26	3/20/26
Gather Feedback	The Group	36%	1/7/26	3/20/26

Insert new rows ABOVE this one



A Supplemental materials