Jumpman LCD Game

By: San San Goh

Description

Goal:

- Allow the user to move the character up and down to avoid obstacle
- A buzzer will sound if the user encounters the obstacle

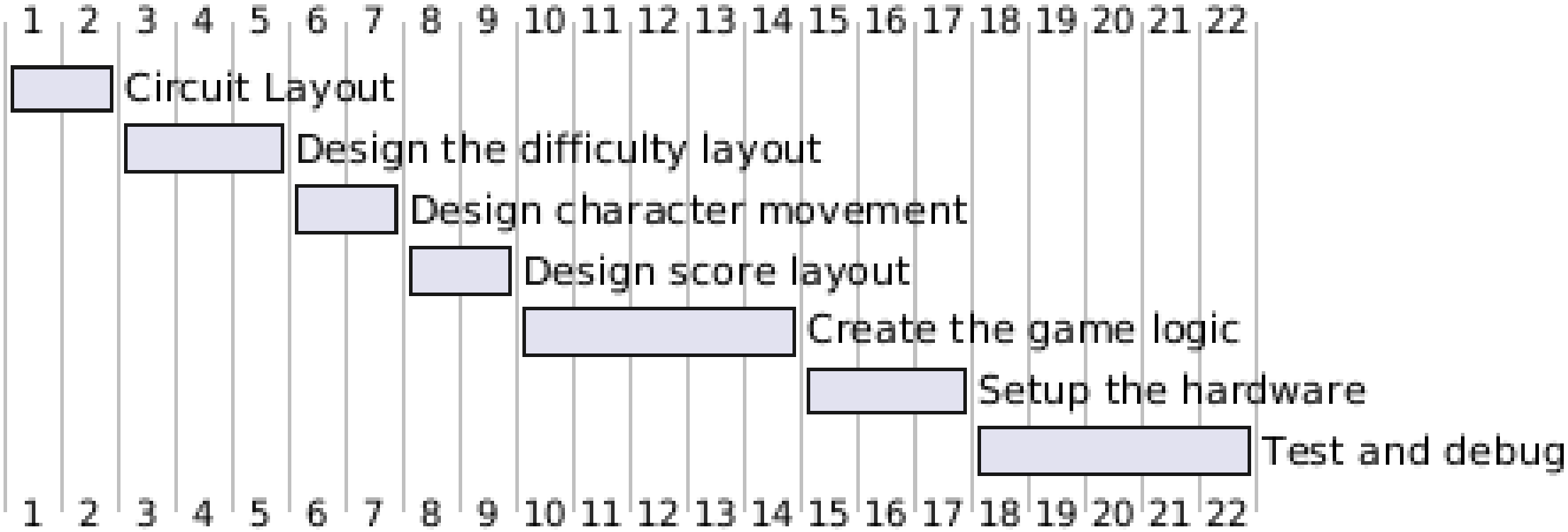
Requirement

Requirements:

- Allowing the charcter to jump, collect items and avoid obstacle [function]
- Store information regarding the placement of obstacles, items and layout [data]
- Keeping track of the score [function]
- LCD screen to display the visual and the buzzer if the user lose the game [usability]
- The IDE software used to code the circuit [environemnt]
- The components to create the circuit [environment]
- Implementing difficulty level [function]
- Storing the score [data]
- The joystick to change character movement [usability]
- Push button to start the game and select difficulty level [usability]
- The game gives off a retro nostalgic vibe [context]
- The target audience who might be participating in this game [context]
- Potentiometer to adjust the contrast of the LCD screen [usability]

Timing

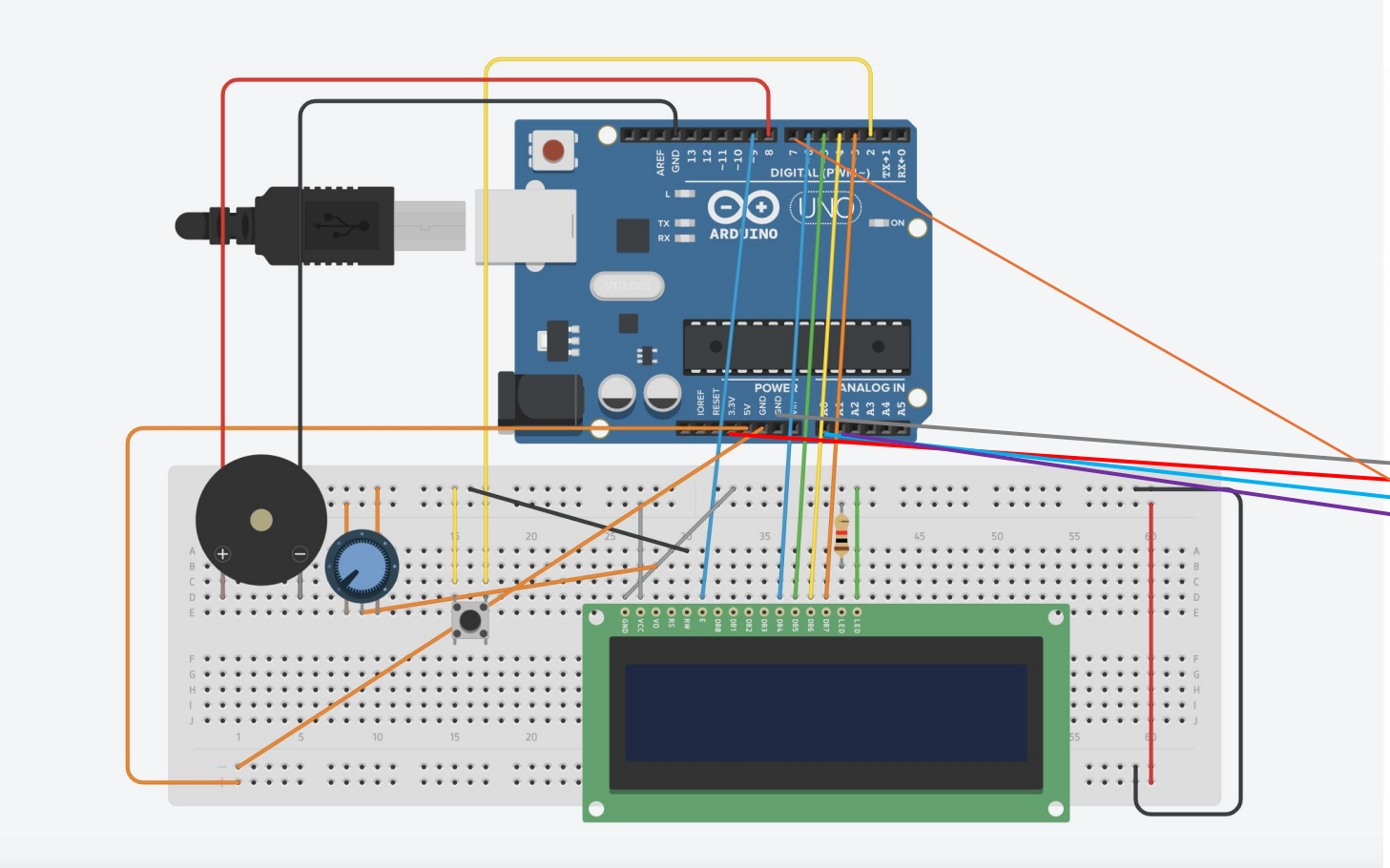
- Gant chart:

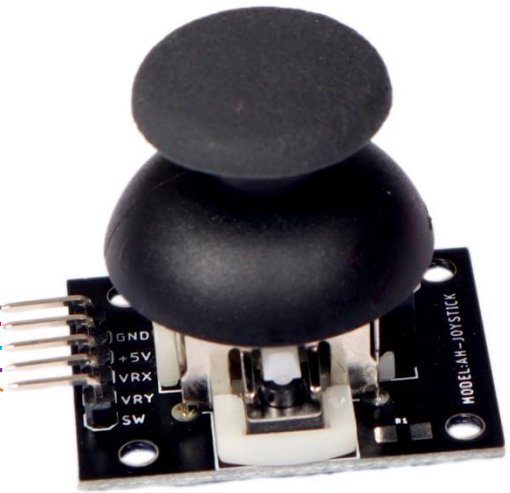


Components

- Breadboard
- LCD screen
- Joystick
- Buzzer
- Arduino Mega
- Push button
- Resistor
- Potentiometer

Circuit





GND connected to GND
5V connected to 3.3V
VRX connected to A0
VRY connected to A1
SW connected to digital pin 7

Code Architecture

Hardware Setup

- **LCD**: Displays the game, including player, terrain, and score.
- **Joystick**: Controls the player’s jumping action (Y-axis).
- **Button**: Starts the game and triggers jumps.
- **Buzzer**: Provides feedback on collisions.

Game States

- **Menu**: Displays "Press Start" until the button is pressed.
- **Playing**: The player runs and jumps through scrolling terrain. The game ends if the player collides with solid terrain.

Graphics and Terrain

- Terrain scrolls from right to left, with blocks randomly generated.
- Player sprites are defined for running and jumping, using LCD characters

Player Movement

- The player can run or jump based on joystick input (Y-axis for jump) and button presses.
- Player positions are updated in stages: running (lower/upper) and jumping (upward/downward).
- The game continuously updates the player's position, moves the terrain, and checks for collisions. If the player collides, the game resets to the menu.

Collision Detection

- The player collides with terrain if moving into a non-empty terrain block, ending the game.

Terrain Scrolling

- The terrain moves left each frame. New blocks are generated randomly as the old ones shift off the screen.

Input Handling

- **Joystick**: Detects upward movement to trigger jumps.
- **Button**: Uses an interrupt to trigger jumps when pressed.

Score and Feedback

- Score increases with distance travelled.
- A buzzer sounds on collision to indicate game over.

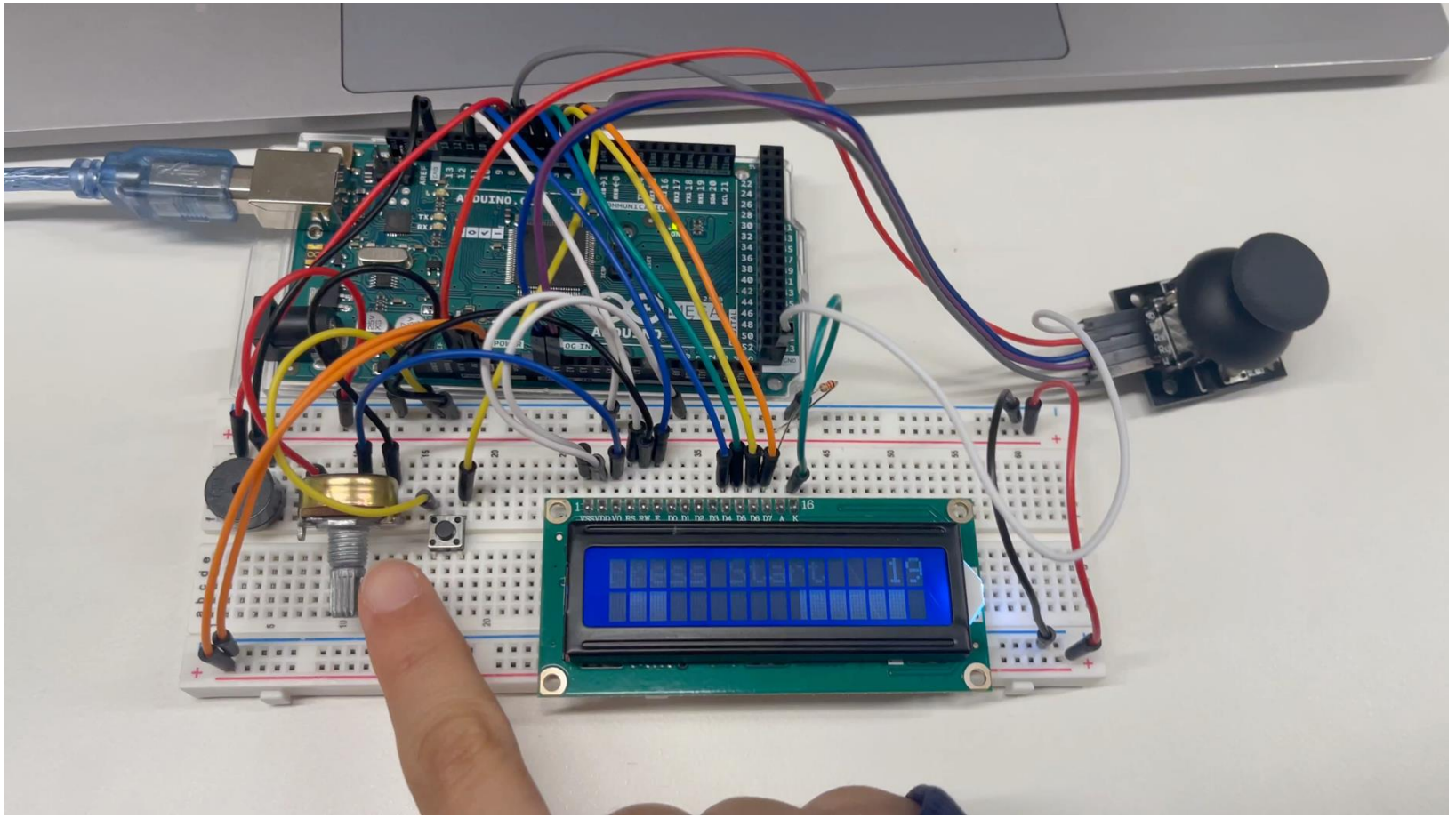
Key Functions

- initializeGraphics(): Sets up custom sprites for the player and terrain.
- advanceTerrain(): Shifts the terrain left and generates new blocks.
- drawHero(): Updates the player’s position and checks for collisions.
- buttonPush(): Interrupt function to handle button presses.

Game Logic

- The game continuously updates the player's position, moves the terrain, and checks for collisions. If the player collides, the buzzer will sound and the game resets to the menu.

Video



ENGR 101 Final Project Video.mov

References

1. https://projecthub.arduino.cc/crepeguy/jumpman-lcd-game-c9aea0
2. https://www.instructables.com/Jumpman-LCD-Arduino-Game/
3. https://zaitronics.com.au/blogs/projects/how-to-build-and-play-jumpman-a-fun-arduino-lcd-game-with-i2c-interface?srsltid=AfmBOopSVa1YRqbyYdAzOxQMtcHGUFJwepEyDGTyAqZWtNgkmPv6-fqd