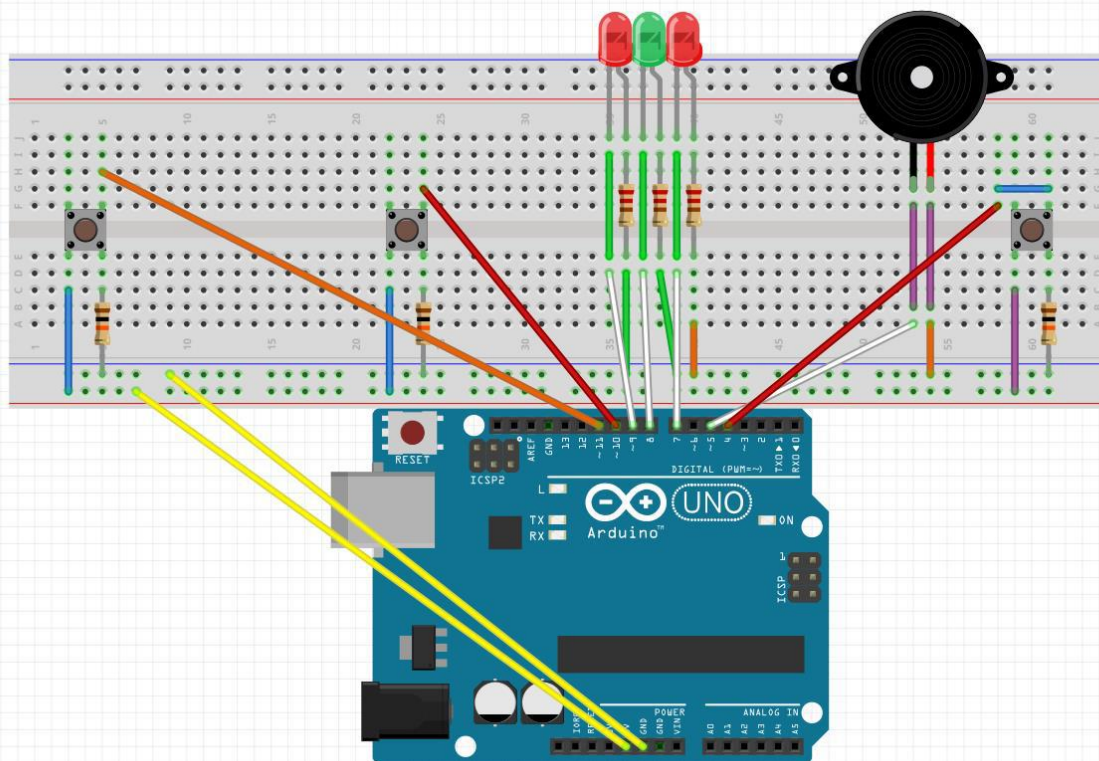


Space Squash Documentation

The “Space Squash” project is a game which has been built while utilizing an Arduino microcontroller, Arduino integrated development environment, and the Processing 3 development environment. Moreover, this project is a single player game which has similar characteristics to the retro classics Pong and Brick Breaker. The objective of this game is simple in that the player must try to keep the ball within the screen for as long as possible without letting drop through the boundaries of the paddle itself.

To begin with, the Arduino microcontroller and the breadboard acts as a controller for the game. The very left button shifts the player’s paddle to the very left, the very right button shifts the paddle to the very right, and the button towards the center resets the game for the player. The buzzer produces a small melody to indicate that the game is restarting whenever the center button is pressed. Finally, the LEDs in the middle indicate the state of the paddle: the left red LED lights up whenever the paddle is moving left, the green LED lights up whenever the paddle is stationary, and the right red LED lights up whenever the paddle is moving rightwards.

The full circuit of the controller is demonstrated below:



The code for the controller has been programmed through the Arduino IDE, which is an integrated development environment used to control components of the Arduino microcontroller. Moreover, the Arduino is controlled through a series of statements that either makes the component either turn on (represented by a 1 or HIGH) or turn off (represented by a 0 or LOW). The Arduino IDE is also used to collect data from the Arduino through the “Serial.print()” function. The example code from the Space Squash controller which achieves the data-input and output functionalities is shown below:

```
void loop() {
  // Read the data coming from the buttons, which is represented in int of either 0 or 1
  int leftMove = digitalRead(leftButton);
  int rightMove = digitalRead(rightButton);
  int reset = digitalRead(resetButton);

  if (leftMove == 1) {
    Serial.print("Left\n");
    ledStatus(1, 0, 0);
  }
  else if (rightMove == 1) {
    Serial.print("Right\n");
    ledStatus(0, 0, 1);
  }
}
```

The code for the game is written in the Processing development environment, which is an IDE designed to create visual multimedia while potentially utilizing the Arduino microcontroller. The code in Processing is mainly responsible for outputting data to the computer by reading the printed statements from the Arduino serial port. For example, whenever the leftMove button is pressed, the string “Left” followed by the next line sequence is executed. The example code from Processing, which is shown below, detects the string “Left” and instructs the program to tilt the paddle to the left.

```
switch(data) { // Alternate pre-declared boolean values corresponding to the input from the serial port
  case "Left\n":
    tiltLeft = true;
    tiltRight = false;
    break;
  case "Right\n":
    tiltLeft = false;
    tiltRight = true;
    break;
  case "Reset\n":
    gameReset();
  case "0\n":
    tiltLeft = false;
    tiltRight = false;
    break;
}
```

Aside from reading and outputting data, the code written by Processing is also responsible for sketching other aesthetic components of the games like the game objects and defining the logic of the game to make it entertaining while intuitive.

To play the game, the user must follow these steps:

1. Unzip the compressed “SeJin_SpaceSquash_Nov15” file
2. Build the Arduino circuit on the breadboard as shown in the diagram below
3. Open, compile, and upload the SpaceSquash_Controller file to the Arduino microcontroller through the Arduino IDE
4. Open and execute the SeJin_SpaceSquash file in the “SeJin_SpaceSquash” folder with the Processing development environment

Any knowledge acquired while constructing the project is from the Arduino¹ and Processing² Language References which can be found online. These references provide extremely comprehensive information on how to utilize each of the functions that are pre-written in the Arduino and Processing programming languages.

The video demonstration of this game can be found here: <https://youtu.be/B5gA0su6WxE>

Bibliography

1. “Language Reference.” *Arduino – Language Reference*, www.arduino.cc/reference/en/.
2. Fry, Ben, and Casey Reas. “Reference. Processing Was Designed to Be a Flexible Software Sketchbook.” *Language Reference*, Processing, processing.org/reference/.