User Documentation

Suleiman Farah

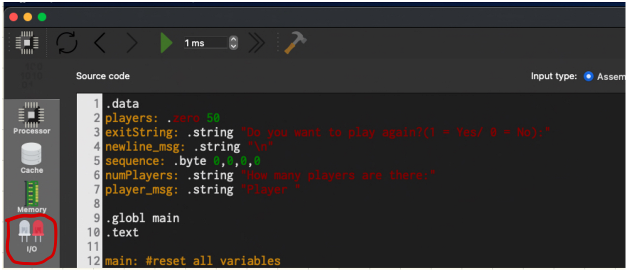
March 2023

Introduction

The program referenced in this user documentation is based on a memory game called Simon. Simon challenges the player's ability to recall a sequence of touch-sensitive buttons that light up in different colors. In each round, the LED device will display a sequence of button colors that the player must then replicate by touching the buttons in the same order. If the player is able to match the sequence accurately and quickly enough, they will progress to the next round, where the sequence will become longer. As the game continues, the sequences will become increasingly challenging, until the player is unable to replicate the sequence correctly, at which point the game will end.

How to Start

To compile and engage with the program, the user must have the latest version of Ripes installed. Assuming that Ripes is downloaded, the user needs to setup the proper I/O (input/output) devices in the I/O tab (see figure 1 for reference). To load the program, the user must download the "starter.s" source file and select the file via Load Program selection in the Files tab (located on the topleft on IOS).



```
.data
players: .zero 50
exitString: .string "Do you want to play again?(1 = Yes/ 0 = No):"
newline_msg: .string "\n"
sequence: .byte 0,0,0,0
numPlayers: .string "How many players are there:"
player_msg: .string "Player "

.globl main
.text

main: #reset all variables
```

Figure 1: Location of I/O tab in Ripes

The program requires two I/O devices, one 2x2 LED Matrix, and one D-pad. To setup the LED Matrix, double-click the "LED Matrix" option under Devices located in the upper-left corner. The LED Matrix 0 specifications should appear in the upper-right corner. Set the value of width and height to 2, this creates a 2x2 grid of RGB LEDs.

Next, double-click the ”D-Pad” selection to create the D-pad, the process is the same as the LED Matrix creation. Do not modify any of the D-pad specifications as it may cause the program to crash. Also, the user is recommended to separate the D-pad and LED Matrix, respectively, to see the LEDs flash while maintaining access to the console. Otherwise, the user will need to switch between the Editor and I/O tabs during program execution, which could impede the game experience.

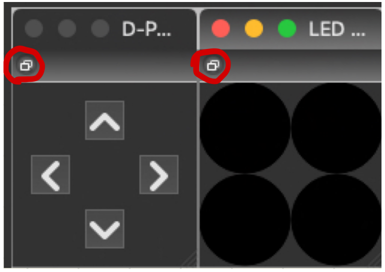


Figure 2: Location of window button for I/O devices in Ripes

Gameplay

On any execution of the program, the user will be prompted for the number of players in the Simon game. After each turn, the score of each player will be calculated and outputted to the console. The console will display the current player’s turn. The user will be prompted to play again at the end of each round, the multiplayer feature allows each players score to carry over between multiple rounds. A round ends once all players have taken a turn.

On the execution of a round, four random numbers between 0 and 3 inclusive will be chosen and will be mapped to each LED in the matrix. The random number sequence will be stored as bytes in adjacent points in memory to mimic an array. For convenience, each LED in the matrix corresponds to a different colour.

RANDOM NUMBER SEQUENCE TO LED MAPPING (COLOUR AND POSITION)

- 0 = RED (0, 0)
- 1 = GREEN (1, 0)
- 2 = WHITE (0, 1)
- 3 = BLUE (1, 1)

A player will be able to guess the order of the LED sequence after the LEDs stop flashing via mouse click on the D-Pad. All of the LEDs will flash green on a correct guess and red otherwise. If a player guesses any of the sequence incorrectly by pressing the wrong D-Pad mapping, the player's turn will immediately end and will be receive 0 points for that round. Whereas, a player who guesses the full sequence correctly will be receive 1 point on the completion of their turn.

LED MATRIX TO BUTTON MAPPING

- LED TOPLEFT = DPAD UP
- LED TOPRIGHT = DPAD DOWN
- LED BOTTOMLEFT = DPAD LEFT
- LED BOTTOMRIGHT = DPAD RIGHT

As the number of rounds increases, the difficulty will also increases in the form of faster LED sequence output. The LEDs in the sequence will appear faster as the delay between each LED is reduced by 25 milliseconds each round. Eventually, there will be no delay and the user will not be able to see the LEDs flicker as they switch too quickly between a color and then black to reset.

Ending the Game

The game can only be ended after all the players have taken a turn. If there is only one player, then the user will be prompted to play again at the end of their turn. To exit, the program takes in an integer value where 1 indicates the player wants to start another round and 0 indicates the player wants to exit the program.