Dillon Roller

Mangesh Sakordekar

CSC 314

Program 2

Simon Says 3.0

The main objective of this program was to create a variation of the game “Simon Says” on a Raspberry Pi. Simon says works like this: A sequence of lights are displayed, along with their corresponding tones, and the user must repeat this sequence by pressing the buttons corresponding to the lights. Each time a sequence is completed successfully, the same sequence is repeated with an additional light at the end. This repeats until the user hits the wrong button anywhere in the sequence

The original Simon Says game features just one game mode, the one mentioned previously, along with 4 buttons, and each of its corresponding LEDs and tones. For our version, we decided to implement two additional modes: a fast and a hard mode. We also implemented a scoring system. A point is scored by completing a round. One round is a sequence of button presses. Usage for this program is provided in the manual section.

Game Modes

Normal: One random button is added to the end of each sequence.

Advanced: Time between the lights that are shown in the sequence are dramatically reduced.

Pro: Follows the same rules as the normal mode, but each time a sequence is completed, the number of lights that are added to the end of the sequence increases by 1 each time. For example,

the first sequence might be [B], then the second would be [B, R, G], followed by [B, R, G, G, B, Y], becoming even harder for each round completed.

C-code functions

mode1: This function runs the normal mode for the Simon Says Game.

mode2: This function runs the advanced mode for the Simon Says Game. It does this by using the normal mode, but reducing the time the light is displayed, and reduces the time between each successive display of the light. The audio is also removed to make it a bit harder.

mode3: This function runs the pro mode for the Simon Says Game. It does this by using the normal mode, but when adding a new button press to the end of each sequence, it looks at a counter and adds a new button press for each: I = 0, I < count. This counter is then incremented at the end of each round.

count_score: This function handles outputting the users score at the end of each game. The output is provided verbally.

high_score: Each time a game mode is completed, the score to compared to the high score of that mode. If it is better, it becomes the new high score, and the program outputs verbally “New high score.”

Assembly functions

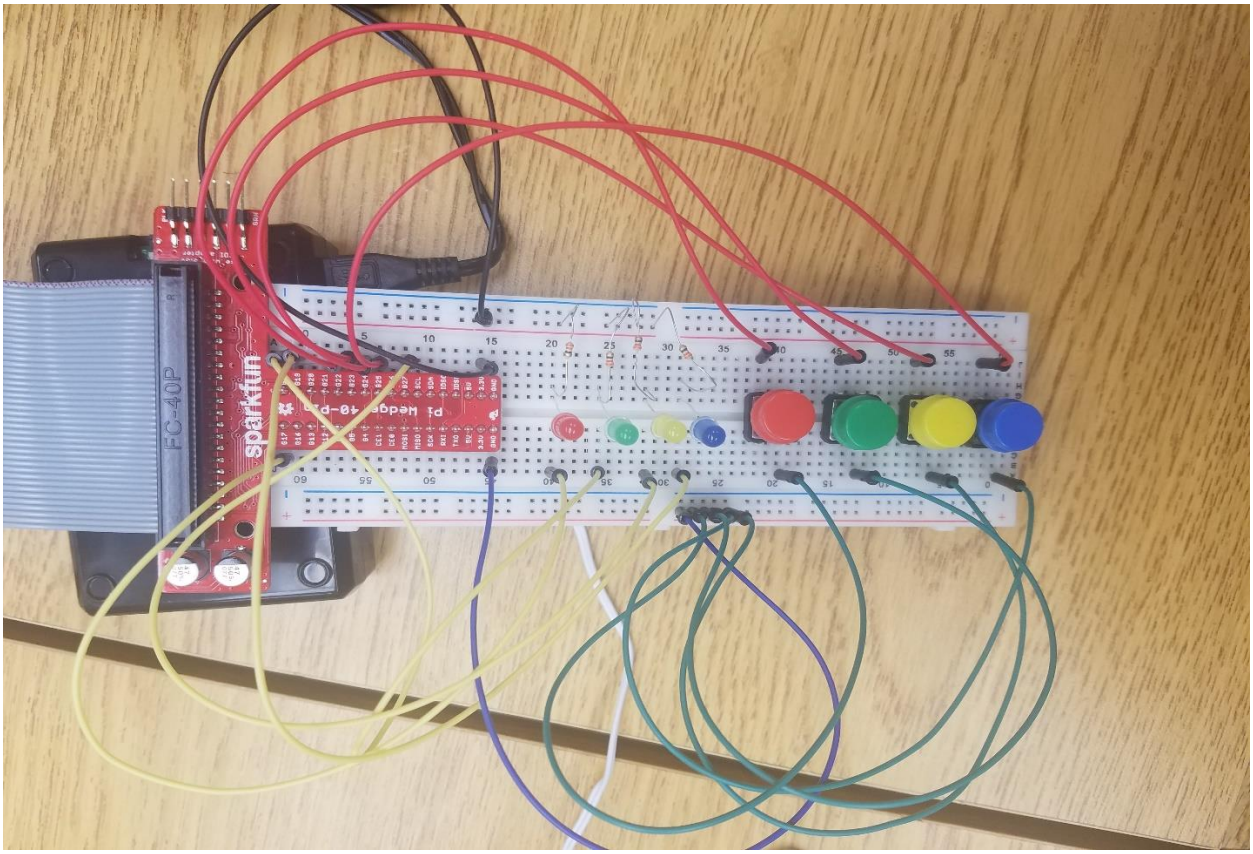
button_led: Used in the startup portion of the game. Greets the user, rapidly blinks all the lights a few times, and then after a short pause, blinks each individual light along with its respective tone.

lightup: Used to display the sequence of moves and output sounds for mode1 and mode3.

lightup2: Used to display the sequence of moves and output sounds for mode2.

read_button: This function reads in the button pressed for the game, and returns an integer (1-4)

Internals Structure of Simon Says Game



LED 1 – GPIO17

LED 2 – GPIO18

LED 3 – GPIO27

LED 4 – GPIO19

BUTTON1 – GPIO22

BUTTON2 – GPIO23

BUTTON3 – GPIO24

BUTTON4 – GPIO25

**All audio credit goes to their respective owners. All of the sounds we used for this program were posted to the public, and specifically stated it could be used for personal reasons.