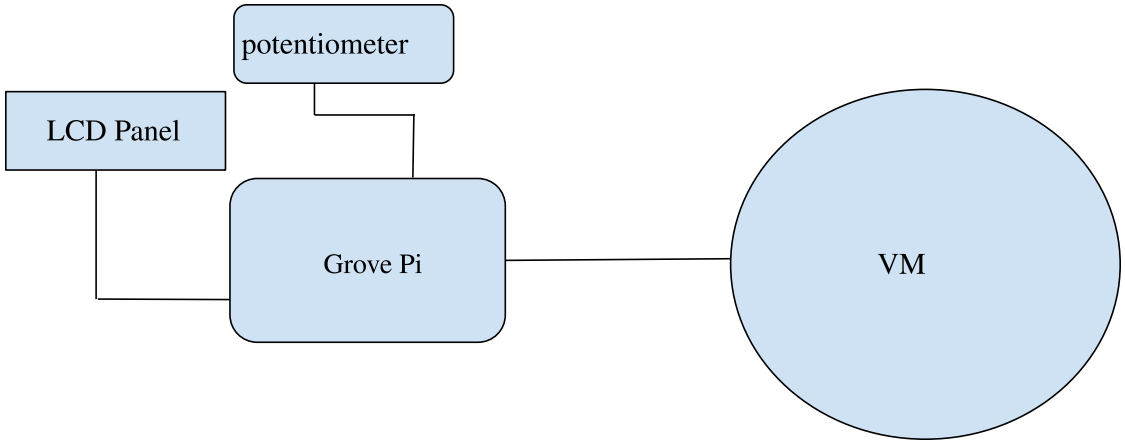
Team members: Donya BadamchiMelissa Shun

250 Final project

05 May 2023

Blinking Bird

Our project is a two-player game which allows a user to move a bird “@” from one side of an LCD panel to another while dodging obstacles “!”. The objective of the game is for two people to take turns and see which player can complete the most consecutive rounds without losing. The winning player is announced after both players have a chance to play and their scores are tallied and sent from the Grove Pi to the player's VM.



Our project included the use of the Virtual Machine, RPI, Potentiometer, LCD Panel, and Grove Pi. Data is collected from the potentiometer and dictates the movement of objects shown on the LCD panel. Information is sent as a dictionary pair to the virtual machine using API POST and GET requests and added to a json file with other winners and their scores. A

leaderboard with the top 5 players is then printed on the terminal window. We used the following libraries: grovpi, time, random, and grove_rgb to create the main code of our game. The game is based on the placement of obstacles and the player's movement of the bird icon "@". We use string manipulation and a random number generator to create obstacles at random on the screen. We use the data from the potentiometer and if and else statements to dictate where the bird would be on the screen. The code works in three different tiers controlled by a "cnt" variable. Reset mode, which resets the player's scores and begins a new game session, set up mode, which initializes all the variables for the beginning of every round, and play mode, which is when the game has started and the players are playing. We use a series of if and else statements to switch between these modes. Once the player has started a game there are a series of checks that happen every time the bird moves: 1st if the player has hit an obstacle, 2nd if the player has reached the end of the LCD panel. Depending on which of those are true and which player is playing (1 or 2) the game adds to a player's score, switches to the next player, or announces a win or tie sends information to the VM and resets the game.

Some of our problems included not knowing how to fill an empty JSON file other than doing it manually and also running into problems trying to use the ultrasonic sensor. Some of the limitations of our device were that you must have 2 players playing and you cannot switch to a single-player or more than two. Another limitation is that the game is very repetitive.