



EGG101: Introduction to engineering Science

EGG 101: Challenge Project

12/10/17

Tanvir Khan and Spencer Cornelis

### **Operational Manual**

#### **Description of the goals and objectives of the game:**

The goals of the game are to test your reaction skills to get the highest score possible on a set difficulty of your choice. The arrows are the obstacles in the game and the x is the character that you control. There are two different aspects of the game. The main aspect of the game is how long you can last for. If the game is too easy at the current speed, then you can adjust the speed by the potentiometer. Your current speed will be indicated through the servo dial.

#### **Instructions for using the game:**

1. Hook up the 9-volt battery to the barrel jack or some other power source such as an ac adapter or USB connection to your computer.
2. Press the start button to begin playing.
3. **Hold the jump button to move x to the top row.**
4. Release the button to move x back to the bottom row.
5. To set a higher difficulty turn potentiometer counter-clockwise.
6. To set at lower difficulty turn potentiometer clockwise.
7. Once the game ends make note of your score and press the start button to play again.

#### **Description of the key technical elements:**

There were many key elements that worked together to make the game operate properly. The Main portion of the project takes place through the LCD. The LCD is wired into the Arduino displaying the actions of the code. Through the Arduino we have several push buttons, a potentiometer, servo motor, resistors, and capacitors that function together to make the game work. We wired this all into a battery to make the game portable.

The push buttons button state of either HIGH or LOW. If the button state is HIGH that means the button was pushed and you can tell the Arduino to do a certain function. If the

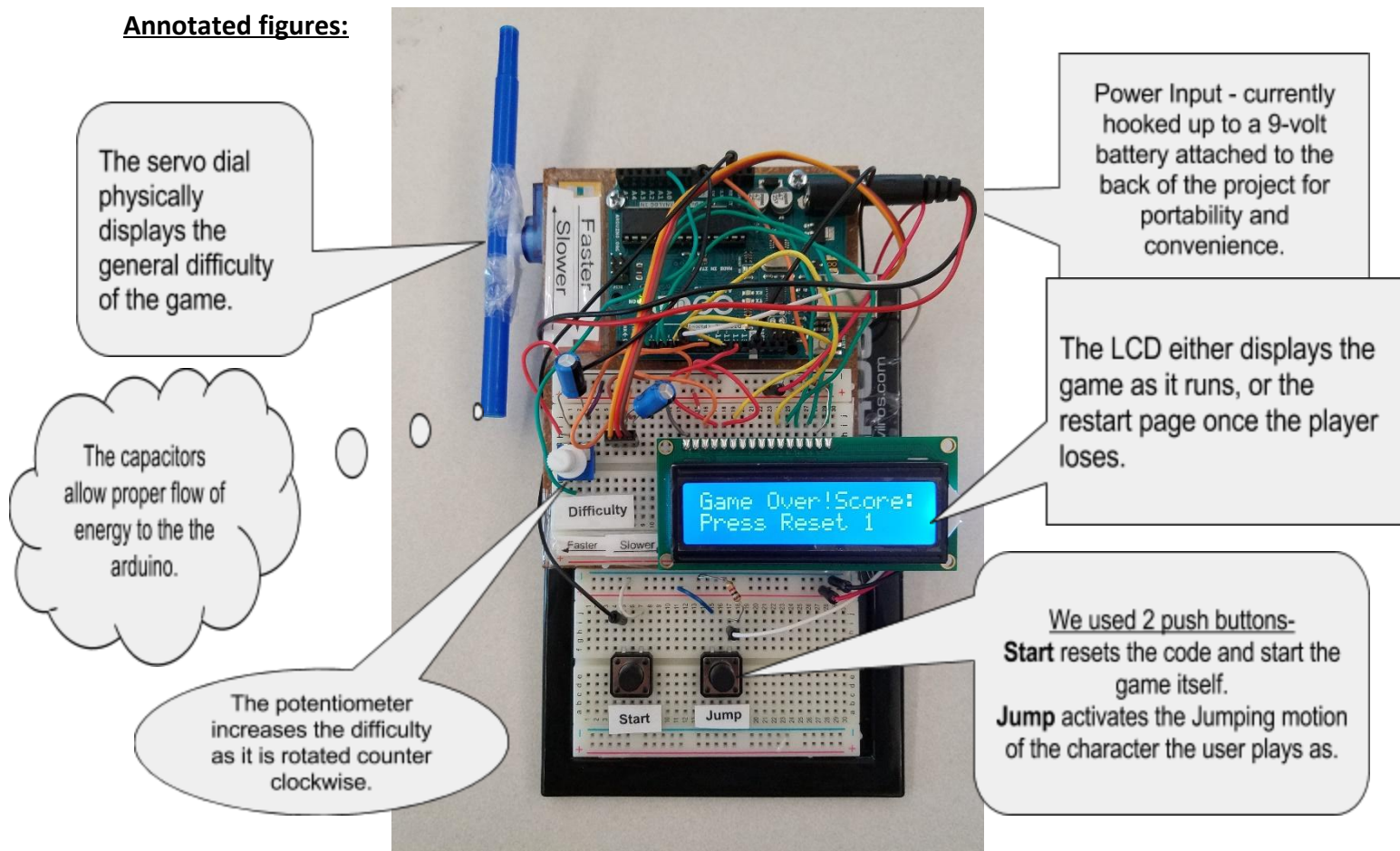
button state is LOW that means the button was not pushed and likewise tell the Arduino to do a certain function. We used 220 ohm resistors to wire our buttons into the breadboard.

The potentiometer is another source of input data the game interprets. The purpose of the potentiometer is to increase or decrease the difficulty of the game. We divided the value of the potentiometer by 1.8 in order to change the scale from 1-179 to a scale of 1-99. 99 is the base setting where the game runs at the easiest difficulty. 1 is the highest difficulty you can play at where the game processes so fast that one can barely react. This way, no matter how good the player is, the game can always be set to a speed that is challenging. We also added capacitors to the circuit to allow proper flow of electrical energy. .

We had to display the speed change in a way that the user could easily interpret. We wired a 180 degree servo into the arduino with a “pen needle” attached so that as the potentiometer value changed the speed, it also changed the angle of our servo. This way our servo could be used as a dial that physically displayed the current difficulty of the game. The user can speed up the projectiles either after or while playing. We added a capacitor to this segment of our circuit to allow proper flow of electrical energy throughout the project. We used a servo dial to provide a physical element of showing how fast or slow the arrows came to the x. When the dial moves clockwise, that indicates that the speed of the projectiles are moving fast and when the dial moves counterclockwise the projectiles are moving slow.

The LCD itself has 2 separate displays. Its primary objective is to display the actual game. The game consists of a player expressed as an “X” which can alternate between the top and bottom rows on the screen as you change the “Jump” buttons value from HIGH to LOW respectively. During the game the LCD also displays a projectile arrow “<-” that appears on the far right side of the screen. It is randomly decided through the code whether this arrow will appear on the top or bottom row of the screen. Then the arrow continuously updates its position so that it stays on whichever row it spawned into and glides across the screen horizontally. The rate at which it’s position updates is decided by the value set on the potentiometer. As the game runs it also displays a timer that is used to keep score. The game ends if the (x,y) positions of the arrow are both identical to the (x,y) position of the player. Once the game ends the LCD reverts to its secondary objective. It clears everything and displays a game over screen. This screen says “Game Over!” and displays your score as well as a restart option that is activated by pressing the start button.

**Annotated figures:**



**Figure 1:**

This is the final product after all the different aspects have been properly wired. It also briefly explains what each important component does.

### Flowchart:

