

Game Project – Arduino Program and Progress

Rough estimates of time spent on this week's project efforts:

Name: Ousmane Toure 3 hrs

Name: Zaki Lodi 3 hrs

Name: Michael Nicholas 3 hrs

Deliverables:

All parts will be submitted together as a single PDF file.

6 Part 1. (15 points) Arduino Code and Wiring Diagram

Our code is fairly simple but effective, we basically started off by declaring our variables and setting up an array to hold all of our minigames. Next, we initialized our main and created a while loop so that the code is always running. We set up two if statements that when either button was pressed would run their associated code. Our dice button would roll a dice between 1 and 7 not including 7 and display it to the LCD screen. The game dice would randomly select one of the variables in the array, games in this case, and display it onto the lcd screen. Those were the only two inputs that would change the LCD screen itself. If players wanted to reset the game for any reason, they would press the reset button on the arduino itself. But resetting the code would only display the Welcome screen, nothing else really changes as it is all random.

Code:

```
#include <time.h>
#include <LiquidCrystal.h>           //the liquid crystal library contains commands
                                     for printing to the display

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);    // tell the RedBoard what pins are
                                             connected to the display

int Dice = 2;           //the pin the buttons connected to
int Game = 4;           //the pin the buttons connected to
int D1;                 //declaring dice
bool playing = true;    //declaring playing loop
const int arraySize = 10; //array size
const char* games[arraySize] = {"Coin Flip", "Rochambeau", "Tic Tac Toe", "Thumb War",
                                "Cup Pong", "Flip Cup", "Coin Spin", "Stare Contest", "Paper Football", "Hand Slap",
                                //array full of our games
};

//end of random order
int main() {
    pinMode(Dice, INPUT_PULLUP);    //checks the state of the pin
    pinMode(Game, INPUT_PULLUP);

    lcd.begin(16, 2); //turns on the lcd and sets its size parameters
    lcd.setCursor(3,0);
    lcd.print("Welcome to"); //prints text onto the lcd
    lcd.setCursor(3,1);
    lcd.print("Gamception");
```

```

while (playing = true) {    //while still playing loop through these if statements
    if (digitalRead(Dice) == LOW ) {    //if the dice button is pushed
        srand ( time(NULL) ); //random time
        D1 = random(1,7);    //the minimum and max value of the dice
        lcd.setCursor (5,0); //sets the lcd start at 0,0
        lcd.clear();        //clears screen
        lcd.print ("You rolled a " );    //prints out the value of the dice
        lcd.print (D1);

    }

    if (digitalRead(Game) == LOW ) { //if the game button is pushed
        srand ( time(NULL) ); //random time
        int RandIndex = random(1,10) ; //generates a random number between 1 and
10
        lcd.clear();
        lcd.setCursor (1,0);
        lcd.print ("Your Game Is: ");
        lcd.setCursor (2,1);
        lcd.print (games[RandIndex]); //prints a random game from the array based
off of the random number generated above

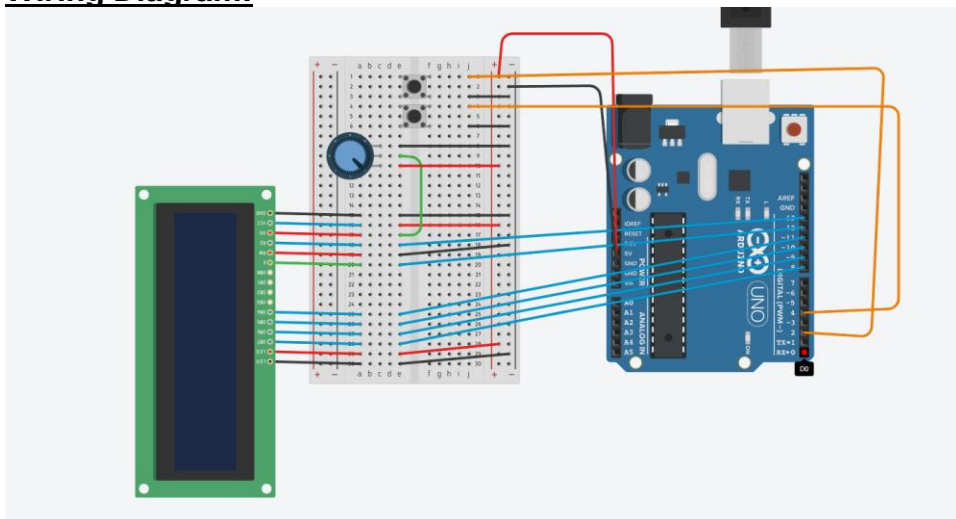
    }

}

return 0;
}

```

Wiring Diagram:



Jumper Wires:

[GND to GND(-)] [5V to 5V+] [D2 to J1] [D4 to J4] [J3 to GND(-)] [J6 to GND(-)] [E8 to GND(-)] [E9 to E17] [E10 to 5V+] [E15 to GND(-)] [E16 to 5V(+)] [E18 to D13] [E19 to GND(-)] [E20 to D12] [E25 to D11] [E26 to D10] [E27 to D9] [E28 to D8] [E29 to 5V(+)] [E30 to GND (-)]

LCD Screen: [A15 – A30] (pin 1 to A15)

Potentiometer: [A8+A9+A10]

Buttons [E1/E3 to G1/G3] [E4/E6 to G4/G6]

Part 2. (15 points) What Remains

In terms of design of our board, we are considering either reprinting the board with new pieces of wood, this would mean designing more spots and creative art on the empty parts of the board. If we had more time to do so, this would take 5-6 hours to redesign, acquire material, and laser cut. For coding, if we had more time, we would want to incorporate more elements, designs that would flash on the lcd and even games that can be played on the breadboard. This would take somewhere between 10 to 12 hours in total, because coding is super hard. Other than that, making our game look better is the primary focus as we want it to look more appealing. This week we could finalize that and end up with a fairly good-looking game.