SLAVE CODE FOR SPARKFUN SET

```
#include <LiquidCrystal.h>
#include <Wire.h>
//for the LED screen
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//for the LED lights that change color
int redPin = 7;
int greenPin = 9;
int bluePin = 8;
const int pressurePad = 10;
//for the game stuff
boolean gameMode = false;
boolean mode, game;
long previousmillis = 0;
long currentmillis, previousmillis3;
int num;
int speakerPin = 13;

void setup() {
  Wire.begin(9);
  Wire.onReceive(receiveEvent);
//set up pins for light stuff
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
    // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  //for pressure pad
  pinMode(pressurePad, INPUT);
  //for game stuff
  randomSeed(analogRead(0));
  Serial.begin(9600);
  pinMode(speakerPin, OUTPUT);
}
void receiveEvent(boolean gameModeOutside){
  gameMode = gameModeOutside;
}
void loop() {
  if(gameMode == true){
    playGame();
    exit(0);
  }
```

```
  delay(10);
}
//code for the game. When it is called, it starts looping
void playGame(){
  game = true;
  int interval = 30000;
  mode = false;
  boolean mode2 = false;
  currentmillis = millis();
  int orimillis = currentmillis;
  while(game == true){
    mode2 = false;
    num = random(4);
       //changes the color of the LED
    if(num == 0){
      currentmillis = millis();
      previousmillis = currentmillis;
      previousmillis3 = currentmillis;
      mode = true;
      color();
      lcd.noDisplay();
      lcd.display();
      lcd.setCursor(5,0);
      lcd.print("Color");
      while(mode == true){
        currentmillis = millis();
        if(currentmillis - previousmillis >= interval){
          previousmillis = currentmillis;
          mode = false;
        }
        if(pressure() == false){
          mode = false;
        }
        if(endGame() == false){
          mode = false;
          game = false;
        }
        if(currentmillis - previousmillis >= interval - 5000){
          mode2 = true;
        }
        if(mode2==true && currentmillis-previousmillis3 >= 1000){
          previousmillis3 = currentmillis;
          tone(speakerPin, 262, 250);
        }
```

```
  }
}
//changes the pattern of the flashing
else if(num == 1){
  currentmillis = millis();
  previousmillis = currentmillis;
  int previousmillis2 = currentmillis;
  previousmillis3 = currentmillis;
  mode = true;
  int smallInterval = blinks();
  lcd.noDisplay();
  lcd.display();
  if(smallInterval == 0){
    lcd.setCursor(4,0);
    lcd.print("Filling:");
    lcd.setCursor(6,1);
    lcd.print("Full");
  }
  else if(smallInterval == 1000){
    lcd.setCursor(4,0);
    lcd.print("Filling:");
    lcd.setCursor(6,1);
    lcd.print("Empty");
  }
  while(mode == true){
    currentmillis = millis();
    if(smallInterval == 0){
      setColor(0,0,255);
    }
    if(smallInterval == 1000){
      setColor(0,0,0);
    }
    if(currentmillis - previousmillis >= interval){
      previousmillis = currentmillis;
      mode = false;
    }
    if(pressure() == false){
      mode = false;
    }
    if(endGame() == false){
      mode = false;
      game = false;
    }
    if(currentmillis - previousmillis >= interval - 5000){
```

```
      mode2 = true;
    }
    if(mode2==true && currentmillis-previousmillis3 >= 1000){
      previousmillis3 = currentmillis;
      tone(speakerPin, 262, 250);
    }
  }
}
//for displaying the number
else if(num == 2){
  currentmillis = millis();
  previousmillis = currentmillis;
  previousmillis3 = currentmillis;
  mode = true;
  lcd.noDisplay();
  lcd.display();
  lcd.setCursor(7,0);
  lcd.print(String(randNumber()));
  while(mode == true){
    currentmillis = millis();
    if(currentmillis - previousmillis >= interval){
      previousmillis = currentmillis;
      mode = false;
    }
    if(pressure() == false){
      mode = false;
    }
    if(endGame() == false){

      mode = false;
      game = false;
    }
    if(currentmillis - previousmillis >= interval - 5000){
      mode2 = true;
    }
    if(mode2==true && currentmillis-previousmillis3 >= 1000){
      previousmillis3 = currentmillis;
      tone(speakerPin, 262, 250);
    }
  }
}
//for displaying the shape
else if(num == 3){
  currentmillis = millis();
```

```
    previousmillis = currentmillis;
    previousmillis3 = currentmillis;
    mode = true;
    lcd.noDisplay();
    lcd.display();
    lcd.setCursor(4,0);
    lcd.print(displayShape());
    while(mode == true){
      currentmillis = millis();
      if(currentmillis - previousmillis >= interval){
        previousmillis = currentmillis;
        mode = false;
      }
      if(pressure() == false){
        mode = false;
      }
      if(endGame() == false){
        mode = false;
        game = false;
      }
      if(currentmillis - previousmillis >= interval - 5000){
        mode2 = true;
      }
      if(mode2==true && currentmillis-previousmillis3 >= 1000){
        previousmillis3 = currentmillis;
        tone(speakerPin, 262, 250);
      }
    }
  }
  endGame();
  mode2 = false;
  lcd.noDisplay();
  lcd.clear();
  setColor(0,0,0);

  //signifies that the next condition has been shown
     //another stimulus base checked :D
  tone(speakerPin, 362,500);
  delay(1000);
  if(currentmillis-orimillis >= 180000){
    game = false;
  }
}
lcd.display();
```

```
  lcd.print("GAME OVER!!!!!!!");
  delay(10000);
  lcd.clear();
  lcd.noDisplay();
}


//methods for each of the modes
void setColor(int redValue, int greenValue, int blueValue){ //255 for each
  analogWrite(redPin, redValue);
  analogWrite(bluePin, blueValue);
  analogWrite(greenPin, greenValue);

}
void color(){
  int randoColor = random(3);
  if(randoColor == 0){ //red
    setColor(255,0,0);
  }
  if(randoColor == 1){ //green
    setColor(0,255,0);
  }
  if(randoColor == 2){ //purple
    setColor(128,0,128);
  }
  return;
}
int blinks(){
  int rando = random(2);
  int intervals = 0;
  if(rando == 0){
    return intervals;
  }
  else{
    intervals = 1000;
    return intervals;
  }
}
int randNumber(){
  int rando = random(3);
  return rando+1;
}
String displayShape(){
  int rando = random(3);
```

```
  if(rando == 0){
    return "Oval";
  }
  else if (rando == 1){
    return "Swiggly";
  }
  else{
    return "Diamond";
  }
}
boolean endGame(){
  if(gameMode == false){

    return false;
  }
  else{

    return true;
  }
}
boolean pressure(){
  int pressureState = digitalRead(pressurePad);
  if(pressureState == LOW){
    return false;
  }
  return true;
}
```