Hand Cricket

Description: The project is an electronic cricket game with a display and keypad, incorporating some unique modifications. The game begins by allowing the player to choose whether they want to bat or bowl. Once the choice is made, the match starts.

During the batting phase:

- The player selects a number between 0 and 6 using the keypad.
- The computer generates a random number within the same range.
- If the numbers do not match, the player's selected number is added to their total score.
- If the numbers match, the player is "out," and the batting phase ends.

The game lasts for three overs. After the first player is out, roles are switched: the player who was bowling now bats, and the same rules apply.

The objective:

- The batting player must achieve a higher score than the opponent's total.
- If the batter scores more, they win; otherwise, the bowler wins.

The display provides live updates on scores, while the keypad allows user input for selecting numbers. This interactive game combines strategy and chance, making it a fun and engaging experience!

Arduino 01 CODE:

```
//Tanzilur Rahman
//Student ID:200595789
//15/08/2024

#include <Wire.h>
#include <IRremote.h>
#define IR_RECEIVE_PIN 11

int led=12;
```

```
void setup()
{
  Serial.begin(9600);
  IrReceiver.begin(IR_RECEIVE_PIN);
  Wire.begin(); // join i2c bus (address optional for master)
  pinMode(led,OUTPUT);

}
int IR = -1;
byte x = 0;

void loop()
{if (IrReceiver.decode()) {
        IrReceiver.resume();
        IR = IrReceiver.decodedIRData.command;
        showled();
        x=IR;
        Wire.beginTransmission(4); // transmit to device #4
    Serial.println(x);        // print the integer
        Wire.write(x);           // sends one byte
        Wire.endTransmission();    // stop transmitting
        delay(500);
  }

}
void showled(){
digitalWrite(led,HIGH);
delay(200);
digitalWrite(led,LOW);
}
```

Arduino 2 CODE:

```
//Tanzilur Rahman
//Student ID:200595789
//15/08/2024
#include <Wire.h>
#include <LiquidCrystal.h>
#define buzzerPin 6
#include <math.h>
LiquidCrystal lcd(12,11, 5, 4, 3, 2); // These pin numbers are hard coded on the serial backpack board

int target = -1;
int computerscore = 0;
int player = -1;
int score = 0;
int currentscore = 0;
int toss = 0;
int state = 0;
double ball=0.0;
const int irCommands[] = {12,16,17,18,20,21,22};
const int scores[] = {0,1,2,3,4,5,6};
bool PisBatting=false;
bool gameEnded = false;
bool secondInnings = false;
bool out=false;
bool Out = false;

int randomNum = -1;

int x = -1;

bool resume=true;



void setup()

{

  lcd.begin(16, 2); // Initialize the LCD.

  Wire.begin(4);                // join i2c bus with address #4

  Wire.onReceive(receiveEvent); // register event

  Serial.begin(9600);           // start serial for output

  pinMode(buzzerPin, OUTPUT);

  randomSeed(analogRead(0));

}




void loop()

{

  if (gameEnded) {

    return;

  }

  if (state == 0) {

    tossprint();

  } else if (state == 1) {

    scoreprint();

    if(x > -1) {

      PlayerUse();
    }
  } else if (state == 2) {
    over();
    computer();
    checkOut();
                scoresaving();
    state = 1;
  }
  if (secondInnings) {
    checkWin();
  }

}





// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany)
{

  x = Wire.read();    // receive byte as an integer
  Serial.println(x);
   delay(100);
   Serial.print("Recevied");
  resume=true;
   playBuzzer();
// print the integer
```

```
}

void tossprint() { // Toss: player chooses to bat or bowl first
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Choose your side");
    lcd.setCursor(0, 1);
    lcd.print("1.BAT  2.BALL");
    delay(1000);

    if (x>10) {
        toss = x;
        lcd.clear();
        delay(300);
        state = 1; // Proceed to the next state

        if (toss == 16) {
            PisBatting = true;
        } else if (toss == 17) {
            PisBatting = false;
        } else {
            state = 0; // Invalid choice, reset state
        }
    }
    toss = -1; // Reset the toss variable
      x = -1; // Reset the toss variable



}
```

```
void PlayerUse() { // Player inputting their turns
  int number = -1;
        number=x;

  for (int i = 0; i < 7; i++) {
    if (number == irCommands[i]) {
      lcd.setCursor(3, 1);
      lcd.print(scores[i]);

      player = irCommands[i];
      Serial.print("Player ");
      Serial.println(player);
      break;
    }
  }

  number = -1;
  state = 2;

}
void computer() { // Computer batting or balling through random
  randomNum = random(7); // Generate a random index from 0 to 6
  currentscore = irCommands[randomNum];

  lcd.setCursor(3, 0);
  lcd.print(randomNum);
  Serial.print("Computer ");
```

```
    Serial.println(currentscore);

    delay(500);
}

void scoresaving() { // Saving the score for team if not out
    if (!out) {
        for (int i = 0; i < 7; i++) {
            if (PisBatting && player == irCommands[i]) {
                score += scores[i];
                break;
            } else if (!PisBatting && currentscore == irCommands[i]) {
                computerscore += scores[i];
                break;
            }
        }

        player = -1;
        currentscore = -2;
        x=-1;

    }
}

void checkOut() { // Checking if out happened
    if (currentscore == player) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("!!!OUT!!!");
    delay(3000);

    endInnings();
  } else {
        out = false;
  }
 x=-1;
}

void endInnings() { // After out, ending innings
  lcd.clear();

  if (!secondInnings) {
    target = (PisBatting ? score : computerscore) + 1;
    score = 0;
    computerscore = 0;
    secondInnings = true;

    Serial.print("target: ");
    Serial.println(target);
    lcd.print("Target: ");
    lcd.print(target);

    delay(1000);
    ball = 0.0;
    PisBatting = !PisBatting;
    out = true;
    lcd.clear();
  } else {
      Out = true;
    }
}

void checkWin() {//after 1st inigs checking if the team wins or not
  if ((target >= 0) && secondInnings ) {
    if (PisBatting && score >= target) {
      gameEnded = true;
      delay(800);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Player Wins!");
      delay(5000);
    } else if (!PisBatting && computerscore >= target) {
      gameEnded = true;
      delay(800);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Computer Wins!");
      delay(5000);
    } else if ( ball <= 3.0 && score < target && PisBatting && Out) {
      gameEnded = true;
      delay(800);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Computer Wins!");
      delay(5000);
    } else if (ball <= 3.0 && computerscore < target && !PisBatting && Out) {
      gameEnded = true;
```

```
   delay(800);
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("Player Wins!");
   delay(5000);
  }
 }
}


void over(){//counting and running over{6 balls a over}


  ball+=0.1;
         if (fmod(ball, 1.0) >= 0.6) {
  ball = ceil(ball);
 }
 if(ball >= 3){
  endInnings();
  lcd.clear();
 }
  delay(500);
  lcd.setCursor(8, 1);
  lcd.print("OVER:");
  lcd.print(ball,1);

}
void scoreprint(){//printing the score
 //lcd.clear();
```

```
   lcd.setCursor(0, 0);
   lcd.print("C:");
         lcd.setCursor(0, 1);
   lcd.print("P:");
  lcd.setCursor(8, 1);
   lcd.print("OVER:");
   lcd.print(ball,1);


  if(PisBatting){
         lcd.setCursor(6, 0);
               lcd.print("P'score:");
         lcd.print(score);



  }
  else{
         lcd.setCursor(6, 0);
               lcd.print("C'score:");
         lcd.print(computerscore);


  }
 delay(500);
}
void playBuzzer() {
  digitalWrite(buzzerPin, HIGH);  // turn the buzzer on
  delay(250);  // wait for 250 milliseconds
  digitalWrite(buzzerPin, LOW);  // turn the buzzer off
}
```