

Projeto Simon Arduino

Nesse projeto eu pensei em usar Sensores Ultrassônico, porque ele tem um controle bem melhor se a pessoa realmente escolheu aquela cor ou não, o sensor de presença não tem como controlar uma distância mínima, então qualquer coisa que passar na frente dele já conta como uma jogada, já o ultrassônico eu posso controlar isso, então se tiver alguém a 30 cm do sensor não irá contar como uma jogada mais quando chegar a 20 cm ou menos sim, por exemplo. Eu também usei um Piezo porque no vídeo que está na tarefa mostrando o jogo Simon, cada cor tem um som diferente então eu coloquei frequências diferentes para tocar em cada cor. A lógica que eu usei foi usar um vetor sem limite específico, porque se o cara que estiver jogando for uma Lenda ele pode continuar sem parar, assim eu gravo a sequência sorteada que deve mostrar para a pessoa nesse vetor e aumentar +1 a cada rodada e depois vou verificando a distância dos sensores e se for menor do que 20 cm e o sensor que ele passou a mão corresponde ao Led da ordem correta da sequência ele acerta e verifica a próxima cor, quando acabar a sequência sem ele ter errado nada, passa para próxima rodada acrescentando +1 na sequência, se ele errar perde saindo dos dois while e assim recomeça o jogo.

Eu usei 20 cm de distância para verificar a jogada porque no simulador é melhor para testar, já na vida real para ficar mais preciso esse valor poderia ser menor.

Código:

```
#include <Adafruit_NeoPixel.h>
```

```
#define PIN 2
```

```
Adafruit_NeoPixel Pixel = Adafruit_NeoPixel(4, PIN, NEO_GRB +  
NEO_KHZ800);
```

```
#define PinEchoVermelho 12
```

```
#define PinTrigVermelho 13
```

```
int pulsoVermelho = 0;
```

```
int distanciaVermelho = 0;
```

```
#define PinEchoAzul 10
```

```
#define PinTrigAzul 11
```

```
int pulsoAzul = 0;
```

```
int distanciaAzul = 0;
```

```
#define PinEchoVerde 8
```

```
#define PinTrigVerde 9
```

```
int pulsoVerde = 0;
```

```
int distanciaVerde = 0;
```

```
#define PinEchoAmarelo 6
```

```
#define PinTrigAmarelo 7
```

```
int pulsoAmarelo = 0;
```

```
int distanciaAmarelo = 0;
```

```
#define Piezo 4
```

```
int Erro = 0;
```

```
int Rodadas = 0;
```

```
int AcenderLed = 0;
```

```
int AcabouSequencia = 0;
```

```
void setup() {
```

```
    Pixel.begin();
```

```
    pinMode(PinEchoVermelho, INPUT);
```

```
    pinMode(PinEchoVerde, INPUT);
```

```
    pinMode(PinEchoAzul, INPUT);
```

```
pinMode(PinEchoAmarelo, INPUT);  
pinMode(PinTrigVermelho, OUTPUT);  
pinMode(PinTrigVerde, OUTPUT);  
pinMode(PinTrigAzul, OUTPUT);  
pinMode(PinTrigAmarelo, OUTPUT);  
pinMode(Piezo, OUTPUT);  
randomSeed(analogRead(0));  
Serial.begin(9600);  
}
```

```
void loop() {
```

```
    // começar a jogar
```

```
    Pixel.setPixelColor(0, 255, 0, 0);
```

```
    Pixel.show();
```

```
    delay(200);
```

```
    Pixel.setPixelColor(1, 0, 255, 0);
```

```
    Pixel.show();
```

```
    delay(200);
```

```
    Pixel.setPixelColor(2, 0, 0, 255);
```

```
    Pixel.show();
```

```
    delay(200);
```

```
    Pixel.setPixelColor(3, 255, 255, 0);
```

```
    Pixel.show();
```

```
    delay(200);
```

```
    Pixel.clear();
```

```
    Pixel.show();
```

```
delay(1500);

Erro = 0;
Rodadas = 0;
int SequenciaLed[] = {};

while(Erro == 0){

    SequenciaLed[Rodadas] = random(1, 5);
    Rodadas = Rodadas + 1;

    for(int i = 0; i < Rodadas; i++){
        // vermelho
        if(SequenciaLed[i] == 1){
            Pixel.setPixelColor(0, 255, 0, 0);
            Pixel.show();
            tone(Piezo, 262, 500);
            delay(500);
            Pixel.clear();
            Pixel.show();
            delay(200);
        }
        // verde
        if(SequenciaLed[i] == 2){
            Pixel.setPixelColor(1, 0, 255, 0);
            Pixel.show();
            tone(Piezo, 294, 500);
            delay(500);
            Pixel.clear();
        }
    }
}
```

```

    Pixel.show();
    delay(200);
}
// azul
if(SequenciaLed[i] == 3){
    Pixel.setPixelColor(2, 0, 0, 255);
        Pixel.show();
    tone(Piezo, 330, 500);
    delay(500);
    Pixel.clear();
    Pixel.show();
    delay(200);
}
// amarelo
if(SequenciaLed[i] == 4){
    Pixel.setPixelColor(3, 255, 255, 0);
        Pixel.show();
    tone(Piezo, 349, 500);
    delay(500);
    Pixel.clear();
    Pixel.show();
    delay(200);
}
}
AcenderLed = 0;
AcabouSequencia = 0;
while(AcabouSequencia == 0 && Erro == 0){
    // Distancia Vermelho:
    digitalWrite(PinTrigVermelho, LOW);

```

```
digitalWrite(PinTrigVermelho, HIGH);
digitalWrite(PinTrigVermelho, LOW);
pulsoVermelho = pulseIn(PinEchoVermelho,HIGH);
distanciaVermelho = pulsoVermelho/58;
// Distancia Verde:
digitalWrite(PinTrigVerde, LOW);
digitalWrite(PinTrigVerde, HIGH);
digitalWrite(PinTrigVerde, LOW);
pulsoVerde = pulseIn(PinEchoVerde,HIGH);
distanciaVerde = pulsoVerde/58;
// Distancia Azul:
digitalWrite(PinTrigAzul, LOW);
digitalWrite(PinTrigAzul, HIGH);
digitalWrite(PinTrigAzul, LOW);
pulsoAzul = pulseIn(PinEchoAzul,HIGH);
distanciaAzul = pulsoAzul/58;
// Distancia Amarelo:
digitalWrite(PinTrigAmarelo, LOW);
digitalWrite(PinTrigAmarelo, HIGH);
digitalWrite(PinTrigAmarelo, LOW);
pulsoAmarelo = pulseIn(PinEchoAmarelo,HIGH);
distanciaAmarelo = pulsoAmarelo/58;

// Verifica Vermelho
if(distanciaVermelho <= 20){
    Pixel.setPixelColor(0, 255, 0, 0);
        Pixel.show();
    tone(Piezo, 262, 500);
    if(SequenciaLed[AcenderLed] == 1){
```

```

    AcenderLed++;
}
else{
    Erro = 1;
}
while(distanciaVermelho <= 20){
    digitalWrite(PinTrigVermelho, LOW);
    digitalWrite(PinTrigVermelho, HIGH);
    digitalWrite(PinTrigVermelho, LOW);
    pulsoVermelho = pulseIn(PinEchoVermelho,HIGH);
    distanciaVermelho = pulsoVermelho/58;
}
delay(500);
Pixel.clear();
Pixel.show();
delay(500);
}
// Verifica Verde
if(distanciaVerde <= 20){
    Pixel.setPixelColor(1, 0, 255, 0);
    Pixel.show();
    tone(Piezo, 294, 500);
    if(SequenciaLed[AcenderLed] == 2){
        AcenderLed++;
    }
    else{
        Erro = 1;
    }
}
while(distanciaVerde <= 20){

```

```
digitalWrite(PinTrigVerde, LOW);
digitalWrite(PinTrigVerde, HIGH);
digitalWrite(PinTrigVerde, LOW);
pulsoVerde = pulseIn(PinEchoVerde,HIGH);
distanciaVerde = pulsoVerde/58;
}
delay(500);
Pixel.clear();
Pixel.show();
delay(500);
}
// Verifica Azul
if(distanciaAzul <= 20){
    Pixel.setPixelColor(2, 0, 0, 255);
        Pixel.show();
    tone(Piezo, 330, 500);
    if(SequenciaLed[AcenderLed] == 3){
        AcenderLed++;
    }
    else{
        Erro = 1;
    }
}
while(distanciaAzul <= 20){
    digitalWrite(PinTrigAzul, LOW);
    digitalWrite(PinTrigAzul, HIGH);
    digitalWrite(PinTrigAzul, LOW);
    pulsoAzul = pulseIn(PinEchoAzul,HIGH);
    distanciaAzul = pulsoAzul/58;
}
```



```

    delay(500);
    Pixel.clear();
    Pixel.show();
    delay(500);
}

// Verifica Amarelo
if(distanciaAmarelo <= 20){
    Pixel.setPixelColor(3, 255, 255, 0);
        Pixel.show();
    tone(Piezo, 349, 500);
    if(SequenciaLed[AcenderLed] == 4){
        AcenderLed++;
    }
    else{
        Erro = 1;
    }
}

while(distanciaAmarelo <= 20){
    digitalWrite(PinTrigAmarelo, LOW);
    digitalWrite(PinTrigAmarelo, HIGH);
    digitalWrite(PinTrigAmarelo, LOW);
    pulsoAmarelo = pulseIn(PinEchoAmarelo,HIGH);
    distanciaAmarelo = pulsoAmarelo/58;
}

delay(500);
Pixel.clear();
Pixel.show();
delay(500);
}

if(AcenderLed >= Rodadas){

```

```
        AcabouSequencia = 1;
    }
}
}
// Perdeu
Serial.println("Errou!");
tone(Piezo, 415, 1500);
delay(3000);
}
```

Print do circuito na próxima página.

