

Componentes utilizados

4 LED's

1 Buzzer

8 Resistores 220Ω

4 Botões

1 Protoboard

Descrição do projeto

O jogo é elaborado no arduino, desenvolvendo o raciocínio e a memorização. Simples e interessante envolvendo concentração para memorizar a ligação dos quatro leds correspondentes a quatro botões. Os botões serão acionados após o acendimento dos leds de acordo com a cor coincidente. Desenvolva habilidades de maneira divertida!

Programação do projeto

```
#define NOTE_D4 294
```

```
#define NOTE_G4 392
```

```
#define NOTE_A4 440
```

```
#define NOTE_A5 880
```

```
// criando o array para os 4 sons para sortear um som
```

```
int tons[4] = { NOTE_A5, NOTE_A4, NOTE_G4, NOTE_D4 };
```

```
// Nossa sequência de até 100 itens vai começar vazia.
```

```
int sequencia[100] = {};
```

```
// Indica a rodada atual que o jogo se encontra.
```

```
int rodada_atual = 0;
```

```
// Indica o passo atual dentro da sequência, é usado enquanto a sequência
```

```
// está sendo reproduzida.
```

```
int passo_atual_na_sequencia = 0;
```

```
/*
```

```
* Indica o pino de áudio, leds e botões.
```

```
* Os pinos de leds e botões estão em ordem, relacionados uns aos outros, ou
```

* seja, o primeiro led está relacionado ao primeiro botão. A ordem destas

* sequências também estão relacionadas a ordem dos tons.

*/

```
int pinoAudio = 6;
```

```
int pinosLeds[4] = { 8, 9, 10, 11 };
```

```
int pinosBotoes[4] = { 2, 3, 4, 5 };
```

```
// Indica se um botão foi pressionado durante o loop principal.
```

```
int botao_pressionado = 0;
```

```
// Flag indicando se o jogo acabou.
```

```
int perdeu_o_jogo = false;
```

```
void setup() {
```

```
    // Definindo o modo dos pinos dos Leds como saída.
```

```
    for (int i = 0; i <= 3; i++) {
```

```
        pinMode(pinosLeds[i], OUTPUT);
```

```
    }
```

```
    // Definindo o modo dos pinos dos Botões como entrada.
```

```
    for (int i = 0; i <= 3; i++) {
```

```
        pinMode(pinosBotoes[i], INPUT);
```

```
    }
```

```
    // Definindo o modo do pino de Áudio como saída.
```

```
    pinMode(pinoAudio, OUTPUT);
```

```
    // Inicializando o random através de uma leitura da porta analógica.
```

```
    // Esta leitura gera um valor variável entre 0 e 1023.
```

```
    randomSeed(analogRead(0));
```

```
}
```

```
void loop() {
```

```
// Se perdeu o jogo reinicializamos todas as variáveis.
```

```
if (perdeu_o_jogo) {  
    int sequencia[100] = {};  
    rodada_atual = 0;  
    passo_atual_na_sequencia = 0;  
    perdeu_o_jogo = false;  
}
```

```
// Toca um som de início para anunciar que o jogo está começando quando é  
a primeira rodada.
```

```
if (rodada_atual == 0) {  
    tocarSomDeInicio();  
    delay(500);  
}
```

```
// Chama a função que inicializa a próxima rodada.
```

```
proximaRodada();
```

```
// Reproduz a sequência atual.
```

```
reproduzirSequencia();
```

```
// Aguarda os botões serem pressionados pelo jogador.
```

```
aguardarJogador();
```

```
// Aguarda 1 segundo entre cada jogada.
```

```
delay(1000);
```

```
}
```

```
// Sorteia um novo item e adiciona na sequência.
```

```
void proximaRodada() {  
    int numero_sorteado = random(0, 4);  
    sequencia[rodada_atual++] = numero_sorteado;  
}
```

// Reproduz a sequência para ser memorizada.

```
void reproduzirSequencia() {  
    for (int i = 0; i < rodada_atual; i++) {  
        tone(pinoAudio, tons[sequencia[i]]);  
        digitalWrite(pinosLeds[sequencia[i]], HIGH);  
        delay(500);  
        noTone(pinoAudio);  
        digitalWrite(pinosLeds[sequencia[i]], LOW);  
        delay(100);  
    }  
    noTone(pinoAudio);  
}
```

// Aguarda o jogador iniciar sua jogada.

```
void aguardarJogador() {  
    for (int i = 0; i < rodada_atual; i++) {  
        aguardarJogada();  
        verificarJogada();  
  
        if (perdeu_o_jogo) {  
            break;  
        }  
  
        passo_atual_na_sequencia++;  
    }  
}
```

// Redefine a variável para 0.

```
passo_atual_na_sequencia = 0;  
}
```

```
void aguardarJogada() {  
    boolean jogada_efetuada = false;
```

```

while (!jogada_efetuada) {
    for (int i = 0; i <= 3; i++) {
        if (digitalRead(pinosBotoes[i]) == HIGH) {
            // Dizendo qual foi o botao pressionado.
            botao_pressionado = i;

            tone(pinoAudio, tons[i]);
            digitalWrite(pinosLeds[i], HIGH);
            delay(300);
            digitalWrite(pinosLeds[i], LOW);
            noTone(pinoAudio);

            jogada_efetuada = true;
        }
    }
    delay(10);
}

void verificarJogada() {
    if (sequencia[passo_atual_na_sequencia] != botao_pressionado) {
        // GAME OVER.
        for (int i = 0; i <= 3; i++) {
            tone(pinoAudio, tons[i]);
            digitalWrite(pinosLeds[i], HIGH);
            delay(200);
            digitalWrite(pinosLeds[i], LOW);
            noTone(pinoAudio);
        }

        tone(pinoAudio, tons[3]);
        for (int i = 0; i <= 3; i++) {

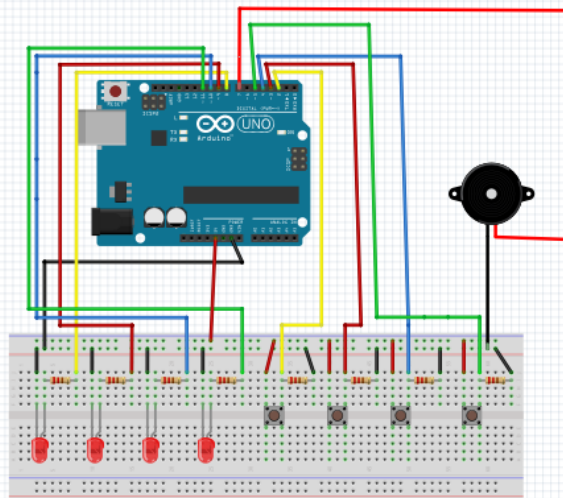
```

```
    digitalWrite(pinosLeds[0], HIGH);
    digitalWrite(pinosLeds[1], HIGH);
    digitalWrite(pinosLeds[2], HIGH);
    digitalWrite(pinosLeds[3], HIGH);
    delay(100);
    digitalWrite(pinosLeds[0], LOW);
    digitalWrite(pinosLeds[1], LOW);
    digitalWrite(pinosLeds[2], LOW);
    digitalWrite(pinosLeds[3], LOW);
    delay(100);
}
noTone(pinoAudio);

perdeu_o_jogo = true;
}
}

void tocarSomDeInicio() {
    tone(pinoAudio, tons[0]);
    digitalWrite(pinosLeds[0], HIGH);
    digitalWrite(pinosLeds[1], HIGH);
    digitalWrite(pinosLeds[2], HIGH);
    digitalWrite(pinosLeds[3], HIGH);
    delay(500);
    digitalWrite(pinosLeds[0], LOW);
    digitalWrite(pinosLeds[1], LOW);
    digitalWrite(pinosLeds[2], LOW);
    digitalWrite(pinosLeds[3], LOW);
    delay(500);
    noTone(pinoAudio);
}
```

Montagem do projeto



fritzing