**Jogo da Memória pelo Tinkercad**

# ARTHUR AUGUSTO DOMINGOS SILVA

02 de Julho de 2024

# RESUMO

Este projeto consiste em um jogo de memória utilizando a plataforma Arduino, LEDs, Botões e um buzzer. O objetivo é que o jogador reproduza uma sequência crescente de tons e luzes gerada aleatoriamente pelo sistema. A cada rodada, a sequência aumenta, aumentando também a dificuldade do jogo. O hardware utilizado inclui uma placa Arduino Uno, quatro LEDs, quatro botões e um buzzer, todos conectados a pinos específicos da placa. O código controla a geração da sequência, a reprodução para o jogador e a verificação das jogadas. Em caso de erro, o jogo termina e um sinal visual e sonoro indica a falha, reiniciando a sequência para uma nova tentativa.

# Palavras-chave: Arduino; Jogo de memória; LEDs; Buzzer; Projeto eletrônico.

**1. INTRODUÇÃO**

Este projeto foi concebido como parte do Trabalho Final da disciplina de “Laboratório de Iniciação á Programação” pela plataforma do Tinkercad, com o intuito de demonstrar as habilidades do grupo em programação e montagem de Arduinos. O objetivo desse projeto foi de criar um jogo de memória onde o jogador deve reproduzir uma sequência de tons de acordo com o que é gerado aleatoriamente. A sequência aumenta a cada rodada, elevando a dificuldade do jogo.O circuito inclui uma placa Arduino Uno, quatro LEDs, quatro botões e um buzzer. Os LEDs e botões estão conectados a pinos digitais específicos do Arduino, e o buzzer está conectado ao pino

12. O sistema gera e reproduz sequências de tons e luzes, que o jogador deve repetir pressionando os botões na ordem correta. Em caso de erro, o jogo reinicia automaticamente, sinalizando a falha com efeitos visuais e sonoros.

# PLANEJAMENTO

Objetivos Específicos

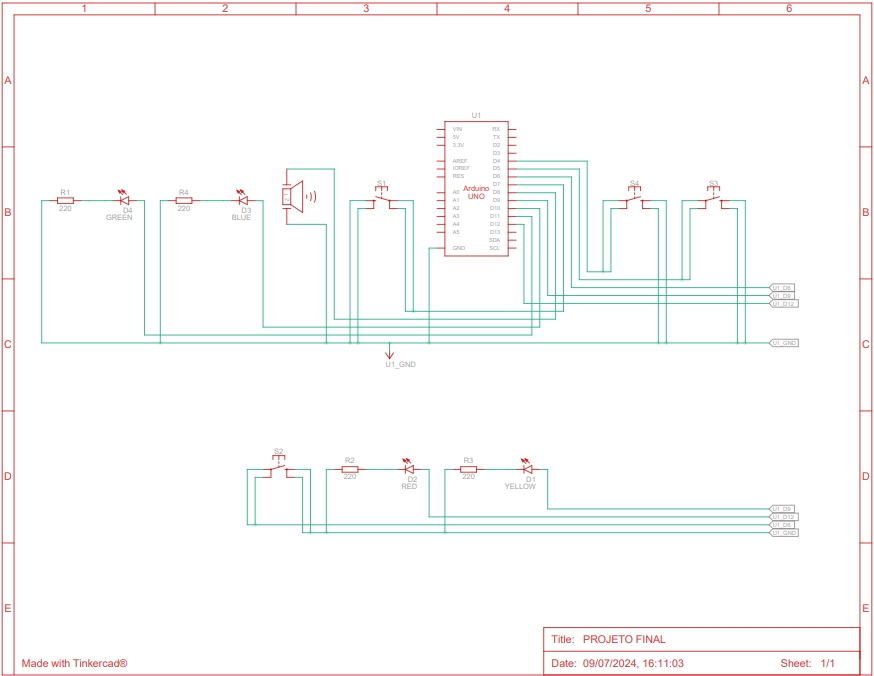
* 1. Desenvolver um jogo de memória utilizando a plataforma Arduino.
  2. Implementar a geração aleatória de sequências de tons e luzes.
  3. Integrar LEDs e um buzzer para feedback visual e sonoro das ações do jogador.
  4. Criar um sistema de verificação de sequências pressionadas pelo jogador.

|  |  |
| --- | --- |
| ETAPA | PRAZO |
| Planejamento | 18/06/2024 |
| Montagem | 25/06/2024 |
| Codificação | 25/06/2024 |
| Testes | 01/07/2024 |
| Documentação | 02/07/2024 |
| Apresentação | 04/07/2024 |

Recursos

* Arduino Uno
* LEDs (4 unidades)
* Botões (4 unidades)
* Buzzer
* Protoboard
* Jumpers
* Software Arduino IDE

# METODOLOGIA



O circuito é composto por uma placa Arduino Uno, LEDs, botões e um buzzer, cada um desempenhando funções específicas:

* Arduino Uno: Controla todo o sistema e executa o código para o jogo de memória.
* LEDs: Indicam visualmente a sequência de tons reproduzida pelo sistema.
* Botões: Permitem ao jogador interagir, reproduzindo a sequência de tons.
* Buzzer: Fornece feedback sonoro correspondente aos tons gerados.
* Resistores: controla o fluxo de corrente elétrica no circuito.

# MONTAGEM

* 1. Conectar LEDs e botões: Ligar os LEDs aos pinos 12, 11, 10 e 9, e os botões aos pinos 7, 6, 5 e 4 da placa Arduino Uno.
  2. Instalar o buzzer: Conectar o buzzer ao pino 8 da placa Arduino.
  3. Montar no Tinkercad: Seguir o diagrama para montar fisicamente os componentes no ambiente virtual do Tinkercad.

# CÓDIGO EM C++

|  |
| --- |
| #define led1 10  #define led2 11  #define led3 12  #define ledPortao 9  #define botao1 5  #define botao2 6  #define botao3 7  #define botaoPortao 4  #define piezo 8  #define nota\_C4 262  #define nota\_E4 330  #define nota\_G4 392  #define nota\_A4 440  void output1();  void output2();  void output3();  void outputPortao();  void memoryClear();  void start();  void gameOver();  void complete();  void check(byte number);  byte memory[20];  byte count = 0x00;  int level = 0x01;  boolean wait = 0x00;  void setup() {  pinMode(led1, OUTPUT);  pinMode(led2, OUTPUT);  pinMode(led3, OUTPUT);  pinMode(ledPortao, OUTPUT);  pinMode(botao1, INPUT\_PULLUP);  pinMode(botao2, INPUT\_PULLUP);  pinMode(botao3, INPUT\_PULLUP);  pinMode(botaoPortao, INPUT\_PULLUP);  pinMode(piezo, OUTPUT);    digitalWrite(led1, LOW);  digitalWrite(led2, LOW);  digitalWrite(led3, LOW);  digitalWrite(ledPortao, LOW);  noTone(piezo);    memoryClear();  }  void loop() {  start();  for(int i = 0; i < level; i++) {  byte rand = random(1, 5);  memory[i] = rand;    switch(rand) {  case 0x01: output1(); break;  case 0x02: output2(); break;  case 0x03: output3(); break;  case 0x04: outputPortao(); break;  }    wait = 0x01;  }  while(wait) {  if(!digitalRead(botao1)) {  output1();  check(0x01);  }  if(!digitalRead(botao2)) {  output2();  check(0x02);  }  if(!digitalRead(botao3)) {  output3();  check(0x03);  }  if(!digitalRead(botaoPortao)) {  outputPortao();  check(0x04);  }  }  }  void output1() {  digitalWrite(led1, HIGH);  tone(piezo, nota\_C4);  delay(300);  digitalWrite(led1, LOW);  noTone(piezo);  delay(300);  }  void output2() {  digitalWrite(led2, HIGH);  tone(piezo, nota\_E4);  delay(300);  digitalWrite(led2, LOW);  noTone(piezo);  delay(300);  }  void output3() {  digitalWrite(led3, HIGH);  tone(piezo, nota\_G4);  delay(300);  digitalWrite(led3, LOW);  noTone(piezo);  delay(300);  }  void outputPortao() {  digitalWrite(ledPortao, HIGH);  tone(piezo, nota\_A4);  delay(300);  digitalWrite(ledPortao, LOW);  noTone(piezo);  delay(300);  }  void start() {  for(int j=0; j<5; j++) {  digitalWrite(led1, HIGH);  digitalWrite(led2, LOW);  digitalWrite(led3, LOW);  digitalWrite(ledPortao, LOW);  delay(100);  digitalWrite(led1, LOW);  digitalWrite(led2, HIGH);  digitalWrite(led3, LOW);  digitalWrite(ledPortao, LOW);  delay(100);  digitalWrite(led1, LOW);  digitalWrite(led2, LOW);  digitalWrite(led3, HIGH);  digitalWrite(ledPortao, LOW);  delay(100);  digitalWrite(led1, LOW);  digitalWrite(led2, LOW);  digitalWrite(led3, LOW);  digitalWrite(ledPortao, HIGH);  delay(100);  }    digitalWrite(ledPortao, LOW);  delay(500);  }  void gameOver() {  start();  level = 0x01;  wait = 0x00;  count = 0x00;  }  void memoryClear() {  for(int i = 0; i < 20; i++) memory[i] = 0;  }  void complete() {  if(level == count) {  wait = 0x00;  count = 0x00;  memoryClear();  level++;  }  }  void check(byte number) {  if(memory[count] == number) {  count++;  complete();  }  else gameOver();  } |

# CONCLUSÃO

O projeto foi bem-sucedido, atingindo os objetivos propostos. O circuito foi montado corretamente, e o código desenvolvido funcionou conforme esperado. O resultado final foi um sistema funcional capaz de gerar sequências aleatórias de tons e cores, que o jogador deve replicar corretamente. Através deste projeto, diversos conceitos de eletrônica e programação foram explorados e aplicados na prática.