Jogo Whack-a-Mole com Arduíno Uno

Simulação Disponível no: [Thinkercard](https://www.tinkercad.com/things/jvFg4etgcoY-whack-a-mole-game-with-arduino)

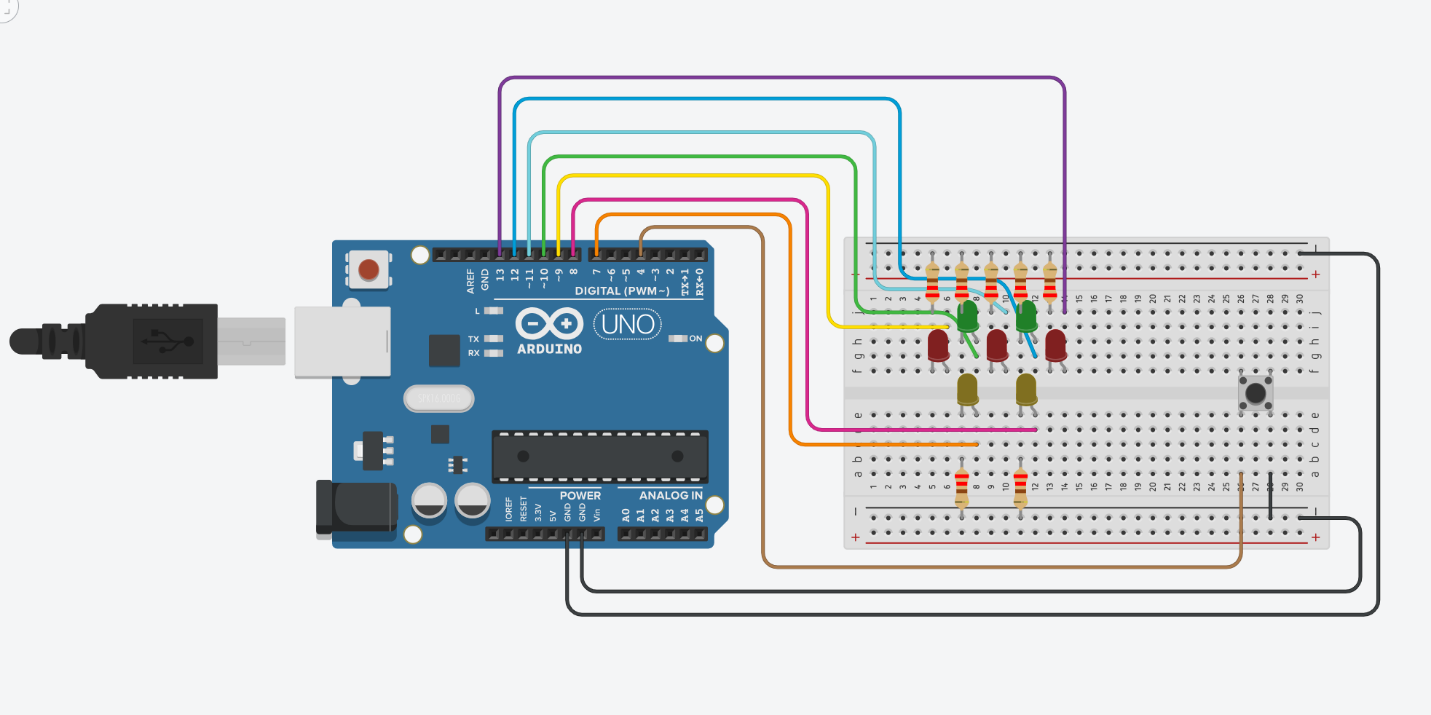
Concebido Por: Carlos Oliveira

# - Componentes:

# 

# - Circuito Elétrico:

# 



# - Funcionamento e Estrutura do Programa:

Este código implementa a lógica principal do jogo Whack-a-Mole utilizando um Arduino Uno. Ele verifica se um botão foi pressionado enquanto um LED aleatório está aceso e utiliza operações bitwise para gerir o estado dos LEDs.

O loop principal verifica continuamente se o botão foi pressionado durante o tempo em que o LED aleatório está aceso. Para isso, utiliza a função millis() para medir o tempo decorrido desde o início do acendimento do LED, armazenado na variável tempo\_inicio. O código implementa debounce para evitar leituras erradas do botão. Se o estado do botão, armazenado na variável leitura\_estado\_botao, mudar, o tempo do último debounce, armazenado na variável ultimo\_tempo\_debounce, é atualizado. Se o tempo decorrido desde o último debounce for maior que o tempo de debounce estabelecido, armazenado na variável tempo\_debounce, o estado do botão é atualizado.

Se o botão for pressionado (estado do botão igual a LOW) durante o tempo em que o LED está aceso, são realizadas operações bitwise para atualizar o estado dos LEDs. O LED atual, armazenado na variável led\_atual, é deslocado para a esquerda (<<) com base no número aleatório gerado, armazenado na variável led\_aleatorio. O estado dos LEDs acesos, armazenado na variável leds\_acesos, é atualizado utilizando a operação OR bitwise (|). No fim, se a variável leds\_acesos corresponder a um valor predefinido (valor correspondente a todos os LEDs estarem ligados) o jogo termina retornando a um estado IDLE, aguardando novamente uma pressão superior a dois segundos do botão. Mensagens de debug são impressas na comunicação Serial para monitorar o estado dos LEDs.

# - Observações Importantes:

1. O código do jogo foi concebido para ser utilizado com um Arduino Uno e foi testado com sucesso. No entanto, é possível que seja necessário fazer ajustes para que funcione corretamente com outros modelos de Arduino ou com outros microcontroladores.

2. O código do jogo foi concebido para ser corrido sem interrupções num prazo máximo de aproximadamente 49 dias devido à natureza do tipo de variável utilizada para armazenar o tempo e à maneira como o tempo é contabilizado. Caso seja necessário correr o código por um período de tempo superior é necessário fazer ajustes para evitar overflow na variável que armazena o tempo e na função que é responsável por o contabilizar (função millis).