

## RELATÓRIO DO PROJETO FINAL

### ESCOPO:

#### 1. Apresentação do projeto

O projeto SideSense tem como objetivo auxiliar pessoas a diferenciar os lados esquerdo e direito. Para isso, será desenvolvido um sistema que oferece duas funcionalidades principais:

Consulta: para identificação do lado pressionado pelo usuário.

Jogo interativo: para reforçar a memorização da direção correta.

#### 2. Título do projeto

SideSense

#### 3. Objetivos do projeto

Objetivo geral: Desenvolver uma ferramenta interativa e educativa para auxiliar pessoas a diferenciar o lado esquerdo do lado direito, promovendo o aprendizado dessa habilidade de maneira intuitiva e divertida, por meio de feedback visual e jogos interativos.

Objetivos específicos:

- Criar um sistema com botões de entrada para que o usuário possa indicar a direção (esquerda ou direita), recebendo feedback imediato através de um painel oled.
- Desenvolver um jogo interativo que auxilie na memorização das direções.
- Fornecer feedback imediato sobre acertos e erros.
- Facilitar a distinção entre os lados esquerdo e direito.
- Testar a eficácia da ferramenta com diferentes faixas etárias e pessoas com dificuldades de orientação espacial, para validar o impacto positivo do sistema.
- Utilizar a **BitDogLab** de forma otimizada, aproveitando os recursos da plataforma para garantir um sistema de baixo custo, robusto e fácil de ser replicado por outros desenvolvedores.

#### 4. Principais requisitos

Requisitos funcionais:

- O sistema deve ser capaz de ler as entradas dos botões de direção (esquerda/direita) pressionados pelo usuário.
- Após a pressão de um botão, o sistema deve fornecer um feedback imediato visual indicando se a direção escolhida está correta ou não.

- O sistema deve proporcionar ao usuário uma experiência de treinamento interativa por meio de um jogo dinâmico e intuitivo
- O sistema deve ser capaz de armazenar a pontuação do usuário de acordo com a quantidade de acertos durante a interação, para mostrar o desempenho ao final do jogo.
- O sistema deve utilizar a **BitDogLab** para integrar hardware e software, com display oled e botões.

Requisitos não funcionais:

- O sistema deve ser fácil de usar, permitindo que qualquer usuário, mesmo sem conhecimento técnico, consiga interagir sem dificuldades.
- O sistema deve garantir uma resposta rápida ao pressionar os botões, com um tempo de latência baixo (menos de 1 segundo) entre a ação do usuário e o feedback.
- O sistema deve ser acessível para pessoas com dificuldades de orientação espacial, proporcionando uma interface clara e feedbacks visuais.
- O sistema deve ser eficiente no consumo de energia, utilizando a **BitDogLab** de forma otimizada para garantir baixo consumo.
- O sistema deve ser facilmente escalável, permitindo a inclusão de novos jogos ou funcionalidades no futuro sem comprometer a estrutura existente.

## 5. Descrição do funcionamento

O funcionamento do projeto pode ser descrito pelas seguintes funcionalidades principais:

1- O sistema inicia com uma tela que permite escolher entre as funcionalidades de consulta e jogo, o usuário deve apertar qualquer botão até que a funcionalidade que ele deseja esteja indicada com letras maiúsculas, depois de 5 segundos, a funcionalidade é escolhida.

Funcionalidade de consulta:

1- A Funcionalidade conta com dois botões físicos, um para cada direção (esquerda e direita). O usuário deve pressionar o botão correspondente à direção que deseja identificar.

2- Após a pressão de um botão, o sistema fornece um feedback imediato por meio do display oled disponível na placa, indicando o lado que o usuário apertou.

Funcionalidade de jogo:

1- A funcionalidade permite que uma direção aleatória seja exibida no display, então o usuário deve apertar o botão correspondente a direção correta.

2- O jogo continua por 30 segundos se o usuário não errar.

3- O sistema registra o desempenho do usuário durante os jogos, exibindo uma pontuação final ao término de cada partida.

4- Se o usuário aperta o botão errado, o jogo é encerrado e exibe a pontuação atingida até o momento.

## 6. Justificativa

Muitas pessoas, especialmente aquelas com dificuldades de orientação espacial, como dislexia, enfrentam desafios ao tentar distinguir esquerda de direita. Esse projeto visa

proporcionar uma ferramenta simples e eficaz para o desenvolvimento dessa habilidade de forma prática e divertida.

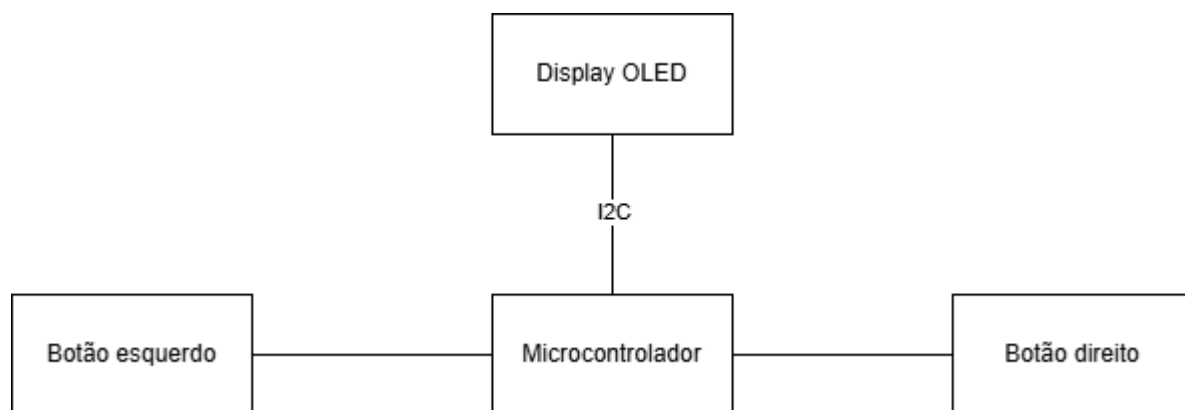
Com ele, será possível superar limitações e proporcionar uma vida mais independente às pessoas que o utilizam, ajudando principalmente no desenvolvimento cognitivo de crianças, no aprendizado de pessoas com dificuldades cognitivas, e auxiliando em tarefas que exigem rapidez e precisão, como dirigir, além de ajudar na prevenção da perda de memória e dificuldades de orientação em idosos.

## 7. Originalidade

Embora existam diversas ferramentas educacionais para auxiliar no aprendizado de direções, o SideSense se destaca pela interatividade e pelo uso de jogos, o que torna o processo de aprendizado mais dinâmico e atraente, consolidando a capacidade de distinguir esquerda e direita de maneira mais eficaz.

### HARDWARE:

#### 1. Diagrama em blocos



#### 2. Função de cada bloco

Microcontrolador: Controla o sistema, processa os dados e gerencia os periféricos.

Display OLED: Exibe mensagens e orientações para o usuário.

Botões: Captam a entrada do usuário para interagir com o sistema.

Interface I2C: Comunicação entre o microcontrolador e o display OLED.

#### 3. Configuração de cada bloco

Microcontrolador: Gerencia entradas e saída de dados via I2C.

Botões: Configurados como entradas com pull-up interno ativado.

Display OLED: Configurado para comunicação I2C.

#### 4. Especificações

O Raspberry Pi RP2040 foi escolhido como microcontrolador devido à sua alta performance e capacidade de gerenciar entradas e saídas digitais, atendendo aos requisitos

do jogo e ao feedback do display. O display OLED SSD1306 exibe as direções e a pontuação de forma clara e eficiente, enquanto os botões push-button permitem que o usuário interaja com o sistema de maneira simples e precisa. A comunicação entre os componentes é feita via I2C e GPIO, garantindo uma integração eficiente e de baixo custo. Esses componentes atendem tanto aos requisitos funcionais (como leitura de entradas e exibição de resultados) quanto aos não funcionais (baixo custo, consumo de energia otimizado e fácil implementação).

## 5. Lista de materiais

Raspberry Pi Pico (1 unidade): Microcontrolador que gerencia os componentes do sistema, realizando o processamento e controle do hardware.

Display OLED SSD1306 (1 unidade): Display de 128x64 pixels utilizado para exibir as direções, pontuação e mensagens de feedback.

Botões push-button (2 unidades): Botões de pressão para captar a entrada do usuário, um para a direção esquerda e outro para a direção direita.

Resistores pull-up internos (2 unidades): Usados para garantir que os botões estejam corretamente configurados com resistores de pull-up.

Jumpers para conexão (1 conjunto): Fios de conexão para ligar os componentes do circuito, como os botões e o display, ao microcontrolador.

## 6. Descrição da pinagem usada

Botão Esquerdo: GPIO 5

Função: Entrada para detectar o pressionamento do botão de direção esquerda.

Botão Direito: GPIO 6

Função: Entrada para detectar o pressionamento do botão de direção direita.

I2C SDA: GPIO 14

Função: Pino de dados para comunicação I2C com o display OLED.

I2C SCL: GPIO 15

Função: Pino de clock para comunicação I2C com o display OLED.

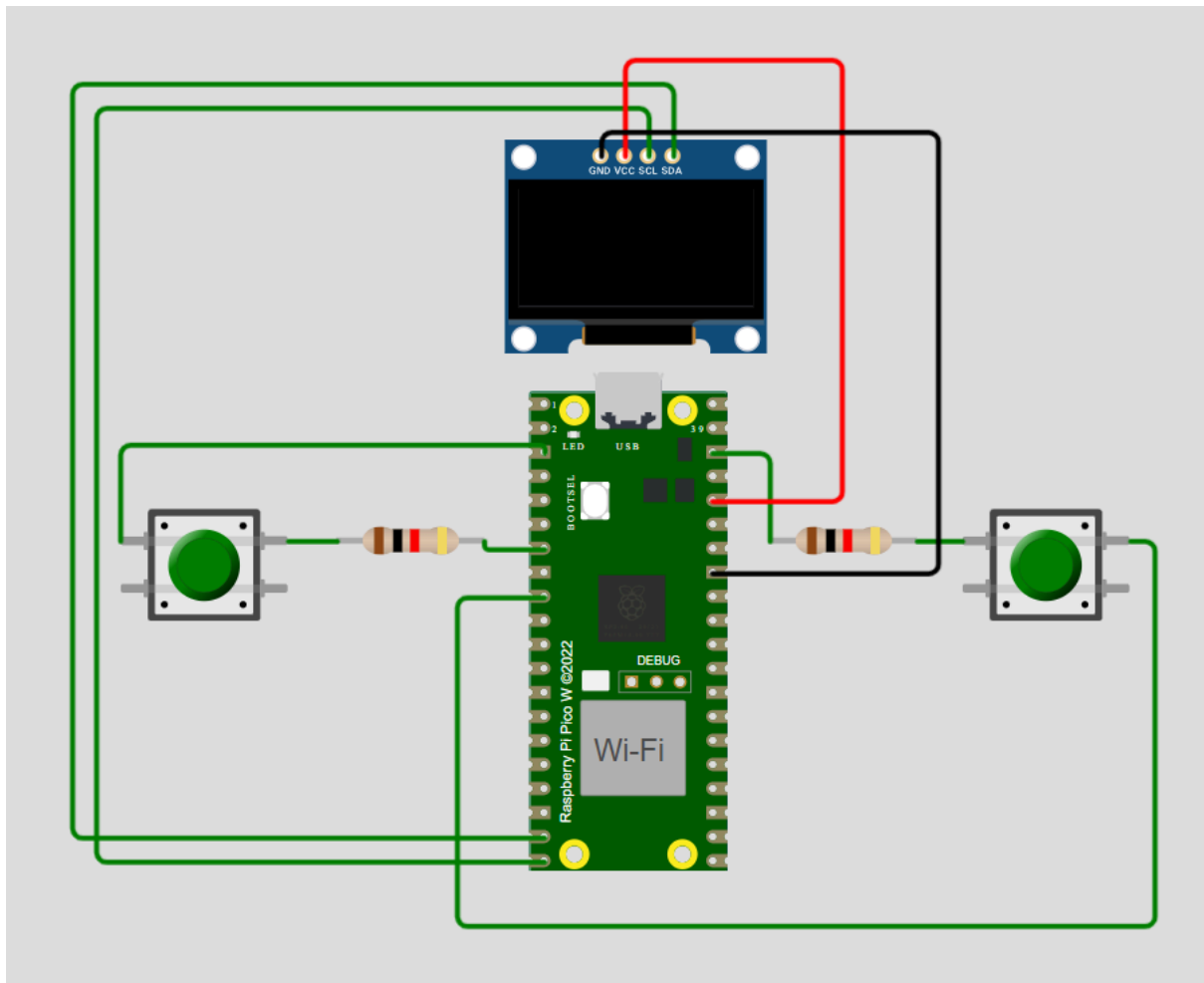
UART TX: GPIO 0

Função: Pino de transmissão de dados para comunicação serial (não utilizado diretamente no projeto, mas reservado para possíveis expansões).

UART RX: GPIO 1

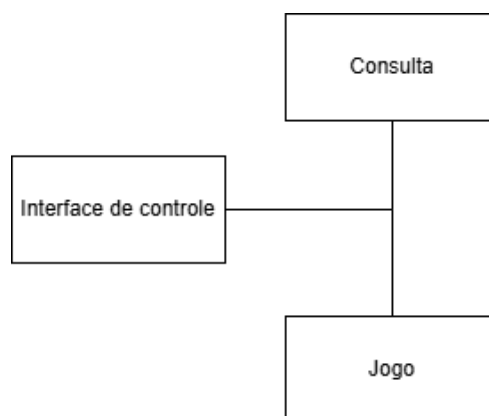
Função: Pino de recepção de dados para comunicação serial (também reservado para futuras expansões, se necessário).

## 7. Circuito completo do hardware



## SOFTWARE:

### 1. Blocos funcionais



### 2. Descrição das funcionalidades

Interface de Controle: Permite ao usuário escolher entre o modo "consulta" e o "jogo" ao pressionar os botões associados.

Consulta: Exibe uma mensagem indicando "O BOTÃO CORRESPONDE À DIREÇÃO", seguida por "ESQUERDA" ou "DIREITA", conforme a ação do botão pressionado pelo usuário.

Jogo: Exibe aleatoriamente as palavras "ESQUERDA" ou "DIREITA" e espera que o usuário pressione o botão correto. O objetivo é acumular pontos respondendo corretamente durante um tempo de 30 segundos.

### 3. Definição das variáveis

BTN\_E\_PIN, BTN\_D\_PIN: Define os pinos dos botões de controle (Esquerda e Direita).

UART\_ID, UART\_TX, UART\_RX: Configurações da UART para comunicação serial.

I2C\_PORT, I2C\_SDA, I2C\_SCL: Configurações da interface I2C para controle do display OLED.

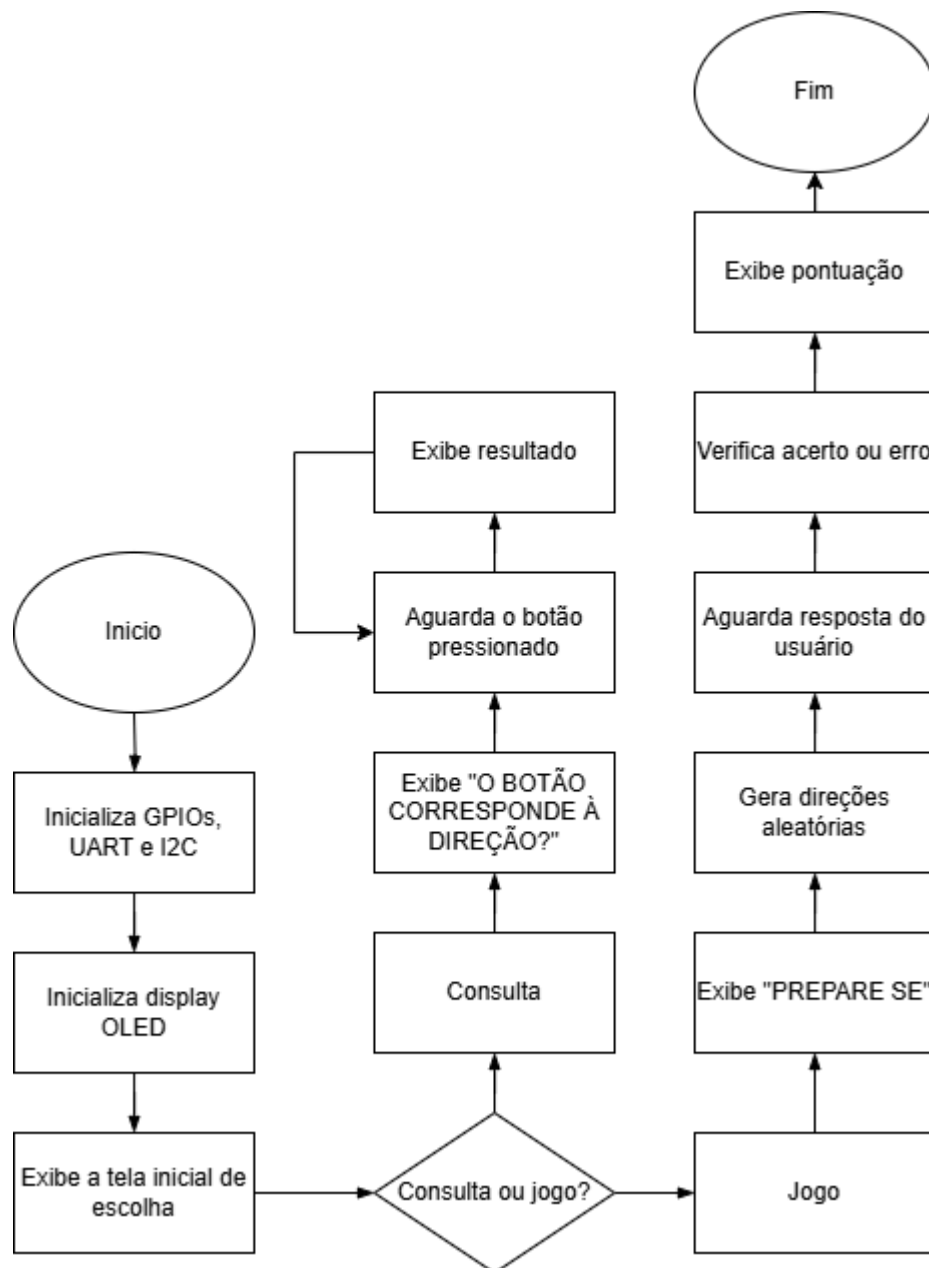
endereço: Endereço do display OLED (0x3C).

input\_string: Armazena a string exibida no display durante a consulta.

contador: Variável que mantém a contagem de pontos no jogo.

tempo\_inicio: Marca o tempo de início de uma partida de jogo para controlar o tempo de duração.

### 4. Fluxograma



## 5. Inicialização

A inicialização do sistema é realizada na função main():

- Inicialização dos pinos GPIO para os botões.
- Configuração da UART e I2C para comunicação.
- Inicialização e configuração do display OLED.

## 6. Configurações dos registros

GPIO: Configuração dos pinos dos botões para entrada com resistores pull-up ativados.

UART: Configuração do baud rate de comunicação com a UART.

I2C: Inicializa o barramento I2C para comunicação com o display OLED.

SSD1306: Configuração do display para exibir texto.

## 7. Estrutura e formato dos dados

As strings exibidas no display OLED são definidas como arrays de caracteres (como `input_string` e `pontuacao`).

As pontuações são armazenadas como inteiros e formatadas como strings para exibição no display.

## 8. Organização da memória

- Endereço da memória EEPROM do Display: 0x3C (endereço do SSD1306)
- Pinos GPIO utilizados:
  - GPIO5 (entrada): Botão Esquerda
  - GPIO6 (entrada): Botão Direita
  - GPIO14 (SDA) e GPIO15 (SCL): Comunicação I2C
  - GPIO0 (TX) e GPIO1 (RX): Comunicação UART
- Áreas de memória alocadas dinamicamente:
  - Buffer de strings para exibição no display (`char input_string[20];`)
  - Variáveis de controle do jogo (`int contador`, `int direcao`)

## 9. Protocolo de comunicação

A comunicação entre o microcontrolador e o display OLED é realizada via I2C utilizando o protocolo para envio de dados gráficos e texto.

## 10. Formato do pacote de dados

Para o protocolo I2C, os dados enviados ao display são compostos por:

- Comandos de controle (por exemplo, para limpar o display).
- Dados de texto (strings como "ESQUERDA", "DIREITA", "VOCE ERROU", etc.).

O pacote de dados é enviado ao display com os comandos e dados a serem exibidos. Cada comando ou dado é seguido de uma sequência de bytes conforme o formato do display OLED.

## EXECUÇÃO DO PROJETO:

### 1. Metodologia

O projeto foi desenvolvido utilizando a plataforma Raspberry Pi Pico com a linguagem de programação C, juntamente com as bibliotecas Pico SDK, para controlar um display OLED (SSD1306) via comunicação I2C e processar entradas de dois botões conectados aos pinos GPIO. A metodologia de desenvolvimento adotada para o projeto seguiu as etapas abaixo:



### 1. Planejamento e Definição de Funcionalidades:

O projeto foi dividido em duas funcionalidades principais: consulta e jogo.

Na funcionalidade de consulta, o sistema exibe uma mensagem indicando qual botão (esquerda ou direita) deve ser pressionado.

No modo de jogo, a exibição alterna entre "ESQUERDA" e "DIREITA", e o jogador deve pressionar o botão correto para acumular pontos.

### 2. Desenvolvimento e Implementação:

As bibliotecas necessárias (Pico SDK, SSD1306) foram integradas ao código.

A comunicação I2C foi configurada para controle do display OLED.

A entrada dos botões foi configurada utilizando GPIO, e o sistema foi projetado para detectar a pressão de botões específicos e gerar respostas no display.

### 3. Integração e Testes:

Após o desenvolvimento das funcionalidades, a integração das funções de entrada e saída foi testada. O display OLED foi configurado para exibir texto e gráficos conforme o comportamento do sistema.

A lógica do jogo foi ajustada para garantir a pontuação correta e feedback visual imediato.

### 4. Validação e Ajustes Finais:

Os testes foram realizados em condições de uso real, com simulação de pressionamento de botões para garantir que o sistema respondesse corretamente às ações do usuário.

### 2. Testes de validação

Foram realizados testes de validação para garantir o funcionamento correto de cada funcionalidade do sistema. Os testes cobriram as seguintes áreas:

#### 1. Testes de Entrada (Botões):

Verificação da resposta do sistema aos pressionamentos dos botões de "esquerda" e "direita".

Teste de debouncing para garantir que o sistema não registrasse múltiplos pressionamentos de um botão devido a ruídos elétricos.

#### 2. Testes de Exibição (Display OLED):

Verificação se as mensagens "ESQUERDA", "DIREITA", "VOCE ERROU", e "VOCE CONSEGUIU" eram corretamente exibidas no display OLED.

Teste da atualização correta do display durante a execução do jogo, sem falhas de sincronização.

### 3. Testes de Lógica de Jogo:

O jogo foi testado para garantir que o tempo fosse contado corretamente, e que a pontuação fosse atualizada conforme o pressionamento correto ou incorreto dos botões.

Verificação de que o sistema exibia a pontuação final após o término do tempo e que o jogo era reiniciado automaticamente.

### 4. Testes de Comunicação I2C:

A comunicação entre o microcontrolador e o display foi testada para garantir que os dados (texto e comandos de controle) fossem transmitidos corretamente pela interface I2C.

### 5. Testes com o público alvo:

Teste com pessoas reais com dificuldade de orientação espacial e com pessoas de idade avançada.

### 3. Discussão dos resultados

Os testes de validação confirmaram que o sistema funciona conforme o esperado, com algumas observações:

#### 1. Desempenho e Estabilidade:

O sistema se mostrou estável, com a comunicação I2C funcionando de maneira eficiente. O display OLED exibiu as informações sem falhas de sincronização ou problemas de atualização.

A detecção de botões foi precisa, com a lógica de debouncing funcionando corretamente, evitando múltiplos registros de pressões rápidas.

#### 2. Precisão no Jogo:

A funcionalidade de jogo funcionou bem, com o contador de pontos sendo atualizado corretamente. No entanto, o tempo de resposta de 30 segundos do jogo poderia ser ajustado para um valor maior, dependendo da preferência do usuário, para dar mais tempo ao jogador para acumular pontos.

#### 3. Usabilidade:

O modo "consulta" foi simples, mas eficaz, fornecendo ao usuário um feedback claro sobre qual botão pressionar.

O modo "jogo" é funcional e desafiador, mas poderia incluir um ajuste de nível de dificuldade (por exemplo, aumentando a velocidade do jogo conforme o progresso).

#### 4. Oportunidades de Melhoria:

Para aprimorar a experiência do usuário, seria interessante adicionar sons ou efeitos visuais adicionais ao jogo para aumentar a imersão.

O projeto poderia ser expandido para incluir um sistema de pontuação de longo prazo, onde os usuários poderiam comparar suas pontuações.

O projeto poderia também ter ajuda visual para pessoas que não conseguem ler, incluindo figuras como setas para orientar o usuário.

Em resumo, o projeto alcançou os objetivos iniciais de desenvolver um sistema interativo simples utilizando o Raspberry Pi Pico, botões e um display OLED, com um funcionamento estável e interatividade eficiente.

#### VÍDEO DO PROJETO:

Link para o vídeo no YouTube

<<https://youtube.com/shorts/YpuCiZEI4aA?feature=share>>

#### REFERÊNCIAS:

**Raspberry Pi Foundation. (2021).** *Raspberry Pi Pico: Datasheet*. Recuperado de: [https://www.raspberrypi.org/documentation/pico/datasheets/rp2040\\_datasheet.pdf](https://www.raspberrypi.org/documentation/pico/datasheets/rp2040_datasheet.pdf)

**Raspberry Pi Foundation. (2021).** *Pico SDK Documentation*. Recuperado de: <https://www.pico-sdk.io/>

**Adafruit. (2021).** *Adafruit SSD1306 OLED Driver for Arduino*. Recuperado de: <https://learn.adafruit.com/adafruit-ssd1306-oled-display/overview>

**Arduino. (2021).** *Debouncing a Button in Arduino*. Recuperado de: <https://www.arduino.cc/en/Tutorial/Debouncing>

**Pico SDK Examples. (2021).** *Using I2C with the Raspberry Pi Pico*. Recuperado de: <https://github.com/raspberrypi/pico-examples/tree/master/i2c>

**Chun, W. (2020).** *Programming Raspberry Pi Pico: Getting Started with the Raspberry Pi Pico using C/C++*. 1st Edition. O'Reilly Media.

**Pico SDK Documentation (2021).** *gpio.h - Raspberry Pi Pico GPIO Interface*. Recuperado de: <https://pico-sdk.readthedocs.io/en/latest/>

**Julius, G. (2019).** *C Programming for Raspberry Pi: C Programming for Beginners*. 2nd Edition. Apress.