Aplicação da Inteligência Artificial na Acessibilidade do Jogo Minecraft: Desenvolvimento de um protótipo para Controle Facial e Sonoro *

Jakelyny Sousa de Araújo† Prof. Dr. Roberto Wiest‡
Prof. Me. José Antônio Oliveira de Figueiredo§

17 de dezembro de 2024

Resumo

Este artigo apresenta uma solução de acessibilidade em jogos eletrônicos direcionada a usuários com limitações motoras por meio do desenvolvimento de um sistema de controle facial e sonoro baseado em inteligência artificial. A aplicação foi implementada em Python e traduz comandos faciais e de voz em ações no jogo, substituindo controles tradicionais e permitindo que indivíduos com baixa mobilidade ou destreza manual interajam autonomamente. O sistema foi testado dentro do ambiente do jogo Minecraft, onde os participantes realizaram missões específicas: coleta de itens, criação de objetos, coleta de alimento e construção de abrigo. Essas tarefas permitiram avaliar a funcionalidade do sistema em diferentes cenários e coletar métricas sobre a precisão dos comandos, esforço físico e cognitivo, bem como a satisfação dos usuários. Os resultados apontaram uma precisão geral de 64,93%, com usuários com deficiência motora enfrentando maiores dificuldades em comandos faciais, enquanto comandos de voz apresentaram melhor desempenho. Todos os usuários concluíram as missões propostas, indicando a funcionalidade do sistema para diferentes perfis de jogadores. Embora os resultados tenham demonstrado a funcionalidade do sistema, foram identificadas oportunidades de aprimoramento, em particular na redução do esforço físico e na melhora da fluidez das funcionalidades de clique e controle facial. Este estudo

*Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense, Campus Passo Fundo, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação, na cidade de Passo Fundo, em 2024.

†Aluno do curso de Ciência da Computação, Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo (jakelynysousa.a@gmail.com).

‡Orientador do trabalho (Professor e Coordenador do curso de Ciência da Computação do Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo; robertowiest@ifsul.edu.br)

§Co-orientador do trabalho (Professor do curso de Ciência da Computação e Coordenador NAPNE do Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo; josefigueiredo@ifsul.edu.br)

reforça a viabilidade de sistemas baseados em controle facial e vocal como ferramentas inclusivas, ampliando o acesso e a interação de pessoas com deficiência motora em jogos eletrônicos.

Palavras-chaves: Acessibilidade. Jogos eletrônicos. Inclusão.

1 Introdução

A inclusão e a acessibilidade em jogos eletrônicos emergem como temas de interesse diante do avanço tecnológico e da crescente diversidade dos jogadores. Em um contexto onde os videogames se consolidam como uma forma de entretenimento e socialização, foi observamdo uma carência de acessibilidade para indivíduos com limitações motoras ou de destreza manual (SMITH; BASAK, 2023). Esse público, embora possua interesse e potencial para se beneficiar das experiências proporcionadas pelos jogos, frequentemente enfrenta barreiras impostas pelos controles tradicionais, que demandam habilidades motoras precisas (FAVA, 2008). A utilização de controles alternativos, que se adaptem a essas necessidades específicas, pode ser aplicado para democratizar o acesso ao universo dos jogos, proporcionando uma experiência inclusiva e equitativa.

A *Human-Computer Interface* (HCI) surge como uma ferramenta para enfrentar esses desafios, pois busca criar interfaces que maximizem a usabilidade e a acessibilidade dos sistemas digitais (MATHEW; HAJJ; ABRI, 2011). No contexto dos jogos, HCI explora formas de interação que vão além dos controles convencionais, integrando tecnologias como reconhecimento facial, comandos de voz e gestos para oferecer uma experiência de jogo mais inclusiva. Tais abordagens visam adaptar a interface ao usuário, considerando suas habilidades e limitações para promover um ambiente digital mais acessível (KOSCH et al., 2023).

O cenário atual revela que as soluções de acessibilidade ainda são limitadas e, muitas vezes, economicamente inviáveis para a maioria dos usuários (OSSMANN; ARCHAMBAULT; MIESENBERGER, 2008). A *Microsoft*, por exemplo, desenvolveu o Xbox *Adaptive Controller*, um dispositivo projetado para atender às necessidades de jogadores com deficiências motoras como é apresentado na Figura 1. Nele é possível conectar dispositivos externos como switches, botões, suportes e joysticks para criar uma experiência de controle personalizada (OMMEN et al., 2024), entretanto só é possível obtê-lo através de encomenda por vendedores certificados pela Microsoft.

Figura 1 – Jonah e seu terapeuta ocupacional Mike jogando com o Xbox *Adaptive Controller* - Divulgação Microsoft

Fonte: (Microsoft, 2018).

Além disso, jogos como *The Last of Us II* implementam configurações de acessibilidade, mas nem todos os desenvolvedores seguem esse modelo inclusivo, criando uma disparidade no acesso aos benefícios emocionais e sociais que os jogos podem oferecer (FAVA, 2008).

A problemática central deste trabalho reside na limitação imposta pelos controles tradicionais para jogadores com baixa mobilidade dos membros superiores. A questão norteadora é: *É possível desenvolver uma solução acessível e intuitiva, que permita a esses jogadores desfrutarem dos jogos eletrônicos?* A hipótese levantada é que, por meio da aplicação de reconhecimento facial e *Artificial intelligence* (AI), é viável oferecer um controle eficaz e acessível, eliminando a necessidade de controles físicos e aumentando a inclusão no ambiente de jogos (GUPTA et al., 2023).

Para responder à limitação imposta pelos controles tradicionais, este projeto propõe uma solução de interação em jogos baseada em Inteligência Artificial (IA) que elimina a necessidade de controles físicos, oferecendo uma interface acessível e intuitiva. A solução estrutura-se em torno de bibliotecas especializadas que aplicam diferentes técnicas de IA para o reconhecimento e interpretação de comandos faciais e de voz, permitindo uma interação adaptada e inclusiva para jogadores com mobilidade reduzida.

O Mediapipe, uma das tecnologias centrais, utiliza *Convolutional Neural Network* (CNNs) para a detecção e rastreamento de *landmarks* faciais, permitindo a interpretação de gestos e movimentos faciais em tempo real, que são então convertidos em comandos no jogo (LUGARESI et al., 2019). O OpenCV complementa essa funcionalidade com técnicas de visão computacional e suporte à integração de modelos de aprendizado profundo, como CNNs, para o processamento de imagens e detecção de movimentos faciais (UÇAR, 2017).

A biblioteca *SpeechRecognition* utiliza modelos de reconhecimento de fala, baseados em *Recurrent Neural Network* (RNNs) e arquiteturas mais avançadas, como *Transformers*, para interpretar comandos de voz. Esses modelos são treinados para identificar padrões acústicos e transformá-los em texto, permitindo que o sistema reconheça e execute comandos de áudio com precisão (KUMAR; VERMA; MANGLA, 2018). A combinação

dessas tecnologias de AI, incluindo CNNs para detecção de *landmarks*, redes recorrentes e *Transformers* para reconhecimento de fala, proporciona uma interface adaptável.

Este estudo contribui para o avanço das tecnologias assistivas em jogos eletrônicos ao propor uma aplicação que utiliza reconhecimento facial e inteligência artificial para oferecer uma interface de controle acessível. Dentre as principais contribuições, estão: uma análise das tecnologias de controle facial e AI com foco na acessibilidade em jogos, através do desenvolvimento de uma aplicação em *Python* que interpreta padrões faciais e sonoros e os converte em comandos de controle no jogo. O estudo inclui a realização de testes de usabilidade da aplicação, divididos em dois grupos: o público-alvo, composto por indivíduos com limitações motoras, composto por usuários sem tais limitações, com o objetivo de avaliar a eficácia e a funcionalidade da solução proposta. Ainda, o trabalho destaca a importância da inclusão digital e da democratização das experiências virtuais, fornecendo uma solução acessível que promove a participação de jogadores com limitações motoras em ambientes de jogos.

Este trabalho é estruturado em cinco seções. A próxima seção apresenta os trabalhos relacionados, destacando as soluções de acessibilidade e tecnologias em jogos eletrônicos. A seção de Metodologia descreve os métodos de desenvolvimento da aplicação proposta. A seção de Resultados apresenta os testes de usabilidade e análise dos resultados obtidos. Por fim, a seção de Discussão e Conclusões discute os principais achados e fornece recomendações para pesquisas futuras.

2 Trabalhos Relacionados

Esta seção apresenta uma revisão de estudos sobre acessibilidade e HCI em jogos. São analisadas abordagens focadas em reconhecimento facial, gestos e interfaces adaptativas com o objetivo de destacar avanços e limitações das soluções atuais.

A HCI baseada em reconhecimento facial e de gestos, a solução permite a detecção da posição dos olhos e boca, além de estimar a posição da cabeça, combinando sua orientação e gestos manuais para controlar dispositivos. Apesar de apresentar uma taxa de precisão de 93,6% no reconhecimento de gestos, o foco do estudo é limitado a controle por gestos, sem explorar acessibilidade para usuários com limitações motoras específicas (TU; KAO; LIN, 2013).

Para consoles de videogame Fanucci, Iacopetti e Roncella (2011) desenvolveram uma interface para acessibilidade em consoles de videogame, direcionada a pessoas com deficiências motoras. O dispositivo adapta o controle convencional permitindo a utilização de sensores comutáveis para controle de jogos. Embora seja eficaz para usuários com deficiências leves a moderadas, a solução depende de interfaces físicas e não considera comandos visuais ou auditivos, restringindo seu potencial de acessibilidade.

No que se refere ao reconhecimento de movimentos, Phan et al. (2022) introduziram o "Mosquito game" para reabilitação de membros superiores através da utilização do OpenCV para reconhecimento de movimentos. Testado com estudantes, o jogo demonstrou ser uma ferramenta motivacional para a reabilitação. No entanto, o sistema foca exclusivamente em suporte terapêutico e não foi projetado como uma interface acessível para jogos convencionais.

Mendes et al. (2023) apresentaram o RehaBEElitation, um jogo sério para avaliar sinais motores em pacientes com Doença de Parkinson. O usuário controla os movimentos

de uma abelha em um ambiente 3D. Embora a arquitetura seja modular e extensível a solução está focada em avaliação clínica e não se destina ao controle direto de jogos convencionais.

Westin (2004) estudou o desenvolvimento de um jogo acessível em 3D chamado "Terraformers", projetado para atender jogadores cegos, com baixa visão e jogadores sem deficiência visual. A interface de som do jogo foi ajustada para criar uma experiência imersiva que integra o mundo gráfico 3D e facilita a navegação auditiva. Embora tenha grande relevância para a acessibilidade, o foco no som limita sua aplicabilidade para usuários que dependem de outras formas de interação, como comandos de voz ou gestos.

Rand, Kizony e Weiss (2004) utilizando o *Sony PlayStation II EyeToy* para reabilitação de idosos com deficiências. O sistema, projetado originalmente como um ambiente de jogo utiliza a captura de vídeo para permitir a relação por movimentos. Em três estudos-piloto, o EyeToy foi comparado com o sistema *VividGroup Gesture Xtreme* (GX) VR, considerando fatores como presença, controle e esforço percebido. Embora apresente potencial como ferramenta de reabilitação a baixo custo, a solução não explora comandos por voz ou interfaces de jogo completas.

3 Metodologia

Esta seção descreve a metodologia para desenvolver o protótipo de controle baseado em movimentos faciais e comandos de voz. Cada subseção aborda um aspecto específico do sistema.

3.1 Tecnologias

A escolha das tecnologias baseadas na linguagem Python, como OpenCV para rastreamento e reconhecimento facial, MediaPipe para detecção de gestos, SpeechRecognition para integração de comandos de voz, PyAutoGUI para automação de interface gráfica, é estratégica devido à sua flexibilidade, robustez e ampla comunidade de desenvolvedores (ABADI et al., 2016). Essas bibliotecas são amplamente reconhecidas por sua capacidade de proporcionar soluções escaláveis e eficientes para problemas complexos de HCI (MAHMOOD; JABBAR; ABDALRAZAK, 2023), especialmente em ambientes de jogos acessíveis, o Python se destaca pela sua versatilidade e robustez em tarefas como processamento de imagens e visão computacional, através do uso de suas bibliotecas.

3.1.1 OpenCV

O reconhecimento facial é uma aplicação de biometria que utiliza características únicas do rosto humano para a identificação e verificação de indivíduos (TURK; PENTLAND, 1991). No presente estudo, a escolha do OpenCV para o desenvolvimento do sistema foi fundamentada na sua robustez e versatilidade no processamento de imagens. Reconhecido por sua eficiência em tarefas de visão computacional como detecção de objetos e reconhecimento facial em tempo real, devido à sua implementação de algoritmos otimizados e à sua extensa documentação (BRADSKI, 2000).

O OpenCV também permite a implementação de CNNs para reconhecimento facial por meio de sua interface de *Deep Neural Network* (DNN). As CNNs são treinadas utilizando um conjunto extenso de imagens faciais anotadas, ajustando os pesos dos neurônios da

rede para minimizar a função de perda, frequentemente representada pela entropia cruzada em problemas de classificação multiclasse (MALHOTRA et al., 2022).

Dessa forma, o OpenCV apresenta-se como uma solução eficiente para o desenvolvimento de sistemas baseados em visão computacional, viabilizando o uso de técnicas avançadas de aprendizado de máquina para o reconhecimento facial e permitindo sua aplicação em cenários que exigem alto desempenho e confiabilidade.

3.1.2 MediaPipe

O rastreamento de pontos faciais desempenha um papel fundamental na visão computacional, permitindo a identificação precisa de características anatômicas do rosto humano em tempo real (TURK; PENTLAND, 1991).

O MediaPipe é um *framework* de processamento de mídia de código aberto desenvolvido pelo Google que destaca-se como uma plataforma versátil para análise de vídeo em tempo real, oferecendo rastreamento preciso de pontos faciais e gestuais (LUNA-JIMÉNEZ et al., 2023). Utilizando modelos de aprendizado profundo otimizados para baixa latência, o MediaPipe tem sido amplamente adotado em aplicações que requerem interatividade, como realidade aumentada, HCI e desenvolvimento de jogos (GOOGLE, 2021).

Para o rastreamento facial, o MediaPipe *Face Mesh* fornece uma representação detalhada do rosto com 468 pontos distintos em 3D, mapeando regiões críticas da face com alta precisão. A pipeline de processamento do MediaPipe integra detecção facial, seguimento de pontos e reconstrução de superfície utilizando uma combinação de modelos de aprendizado de máquina e técnicas de otimização em tempo real (HANGARAGI; SINGH; NEELIMA, 2023).

O *framework* é projetado para operar de maneira eficiente em diversas plataformas, desde servidores de alta capacidade até dispositivos móveis, com suporte à aceleração por GPU para garantir processamento em tempo real. Por meio de seus índices de malha mapeados é possível executar comandos essenciais da aplicação como o deslocamento do cursor, cliques e a calibração de piscadas, a calibração baseia-se no deslocamento da malha e nos índices associados à região ocular. A integração do MediaPipe no desenvolvimento de jogos digitais amplia significativamente as possibilidades de interação, permitindo a implementação de controles baseados em movimentos e gestos naturais do usuário, promovendo experiências interativas e dinâmicas.

3.1.3 SpeechRecognition

O reconhecimento de fala nada mais é do que a conversão de sinais acústicos em texto compreensível por sistemas computacionais, essa tecnologia é fundamental não apenas para a comodidade e eficiência em tarefas cotidianas, como ditado de mensagens e controle por voz de dispositivos inteligentes, mas também em áreas como acessibilidade, tradução automática, transcrição de áudio e análise de sentimentos em mídias sociais (TURK; PENTLAND, 1991).

A escolha pelo SpeechRecognition no protótipo desenvolvido é justificada por sua capacidade de integração com diferentes fontes de áudio, incluindo arquivos, entrada por microfone e transmissão em tempo real que é o foco da aplicação (SPEECHRECOGNITION, 2023). O processamento de áudio na biblioteca SpeechRecognition ocorre por meio de algoritmos de *Automatic Speech Recognition* (ASR), que realizam desde o pré-processamento do sinal de fala até a decodificação do modelo acústico. Modelos acústicos baseados em

Hidden Markov Models (HMMs) ou redes neurais profundas (DNNs) são utilizados para mapear segmentos de fala em unidades fonéticas (MALHOTRA et al., 2022). O pré-processamento envolve a extração de características como os *Mel-Frequency Cepstrum Coefficients* (MFCCs), que capturam as propriedades espectrais da fala, permitindo maior precisão na interpretação de comandos.

A biblioteca SpeechRecognition também interage com APIs de reconhecimento de fala, como Google *Web Speech API* e *Microsoft Bing Voice Recognition*, ampliando a capacidade de resposta em diferentes ambientes (NASSIF et al., 2019). A utilização de comandos de voz no protótipo foi adotada para complementar as limitações do controle exclusivamente facial, permitindo a ampliação da gama de comandos disponíveis para interação no jogo, essa abordagem integrada fornece maior flexibilidade, promovendo uma experiência funcional.

3.1.4 Automação de Comandos

A automação de comandos é uma prática amplamente utilizada em cenários que demandam a execução programática de ações que emulam interações do usuário, como cliques e digitação. Essa técnica é essencial em tarefas repetitivas, como testes de software, simulações de interação de usuários, preenchimento automático de formulários e a criação de scripts para interações complexas em programas sem suporte a APIs acessíveis, essa automação permite a realização de testes automatizados e a análise de desempenho em cenários controlados.

3.1.4.1 PyDirectInput

O PyDirectInput é uma biblioteca Python projetada para simular dispositivos de entrada de forma direta, oferecendo precisão e baixa latência em relação a métodos tradicionais. Essa biblioteca interage com dispositivos enviando comandos de baixo nível ao sistema operacional, sendo adequada para aplicações que exigem responsividade em jogos eletrônicos (LEARNCODEBYGAMING, 2021). Sua utilização é frequente em ambientes de teste e simulação, onde a sincronização de comandos é necessária para atender aos requisitos de desempenho.

No protótipo desenvolvido, o PyDirectInput foi aplicado no controle de cliques direito e esquerdo acionados pelas piscadas dos olhos do usuário e na execução de comandos relacionados à movimentação em linhas específicas de interação facial. A biblioteca demonstrou eficiência em reduzir o tempo de resposta desses comandos básicos e na execução precisa das ações vinculadas às piscadas.

Entretanto, a biblioteca apresentou limitações ao lidar com o controle do cursor e outras teclas, impossibilitando sua aplicação nessas funcionalidade. Apesar de eficiente para cliques e ações pontuais, suas restrições exigiram a busca por alternativas para funções mais complexas.

3.1.4.2 Keyboard

A biblioteca Keyboard desenvolvida em Python, é utilizada para captura de eventos em tempo real e simulação de pressionamentos de teclas. Suas funcionalidades permitem a automação de interações com interfaces que dependem de entradas de teclado, sendo aplicada em scripts de automação e sistemas que demandam precisão na execução de

comandos (SMITH, 2018). A capacidade de capturar e manipular eventos do teclado em tempo real torna-a uma ferramenta útil em cenários que exigem rapidez na entrada de dados. Além disso, por ser multiplataforma, a biblioteca pode ser aplicada em diversos sistemas operacionais, ampliando seu uso em diferentes contextos.

No protótipo desenvolvido, a biblioteca Keyboard foi uma alternativa funcional aos comandos de voz, apresentando tempos de resposta menores em comparação ao PyDirectInput. Essa característica foi relevante para a implementação da hotbar do jogo, onde o reconhecimento entre numerais e strings era necessário para o funcionamento correto, porém, por ser especificamente projetada para manipulação de entradas de teclado, ela não oferece suporte nativo para movimentos ou posicionamentos do mouse. Ainda sim sua integração possibilitou um controle mais adaptável às demandas da aplicação.

3.1.4.3 PyAutoGUI

O PyAutoGUI é uma biblioteca Python projetada para a automação de GUI, permitindo o controle programático de dispositivos de entrada, como mouse e teclado. Essa funcionalidade possibilita a automação de interações com aplicativos que não oferecem APIs acessíveis, sendo utilizada em jogos eletrônicos para testes automatizados de interfaces, simulação de ações de jogadores e análise de desempenho em cenários controlados (SAFITLI; AMINUDDIN et al., 2024).A biblioteca oferece suporte a cliques, movimentos coordenados do mouse e automação de comandos de teclado, sua compatibilidade com sistemas operacionais *Windows*, *macOS* e *Linux* amplia sua aplicabilidade em diversos ambientes.

No protótipo desenvolvido, o PyAutoGUI foi utilizado devido sua capacidade de manipular comandos complexos e realizar o deslocamento correto do cursor, sendo a única biblioteca que atendeu a essa funcionalidade entre as usadas anteriormente. Embora eficiente em termos de abrangência de comandos, a biblioteca apresentou limitações relacionadas ao tempo de resposta, tornando-a menos indicada para operações que demandam alta velocidade de execução, apesar disso, sua flexibilidade e suporte a comandos avançados atenderam às necessidades do protótipo, especialmente no controle do cursor.

3.1.5 Tkinter

Em ambientes científicos, comerciais e educacionais as interfaces gráficas desempenham um papel crucial na facilitação da interação entre usuários e sistemas computacionais, fornecendo uma representação visual que simplifica a complexidade dos dados e funcionalidades subjacentes, permitindo a interação de forma intuitiva com o software (MYERS; HUDSON; PAUSCH, 2000).

A escolha pelo Tkinter é devido sua inclusão na biblioteca padrão do Python, garantindo disponibilidade sem a necessidade de instalações adicionais. Tkinter é o kit de ferramentas de interface GUI fornecido como um *wrapper* em torno da biblioteca Tcl/Tk. Ele é amplamente utilizado para desenvolver interfaces de usuário rápidas e simples, sendo particularmente eficaz em aplicações de desktop para programas de acessibilidade que não exigem gráficos avançados, mas priorizam a simplicidade e a facilidade de uso (LUNDH, 1999).

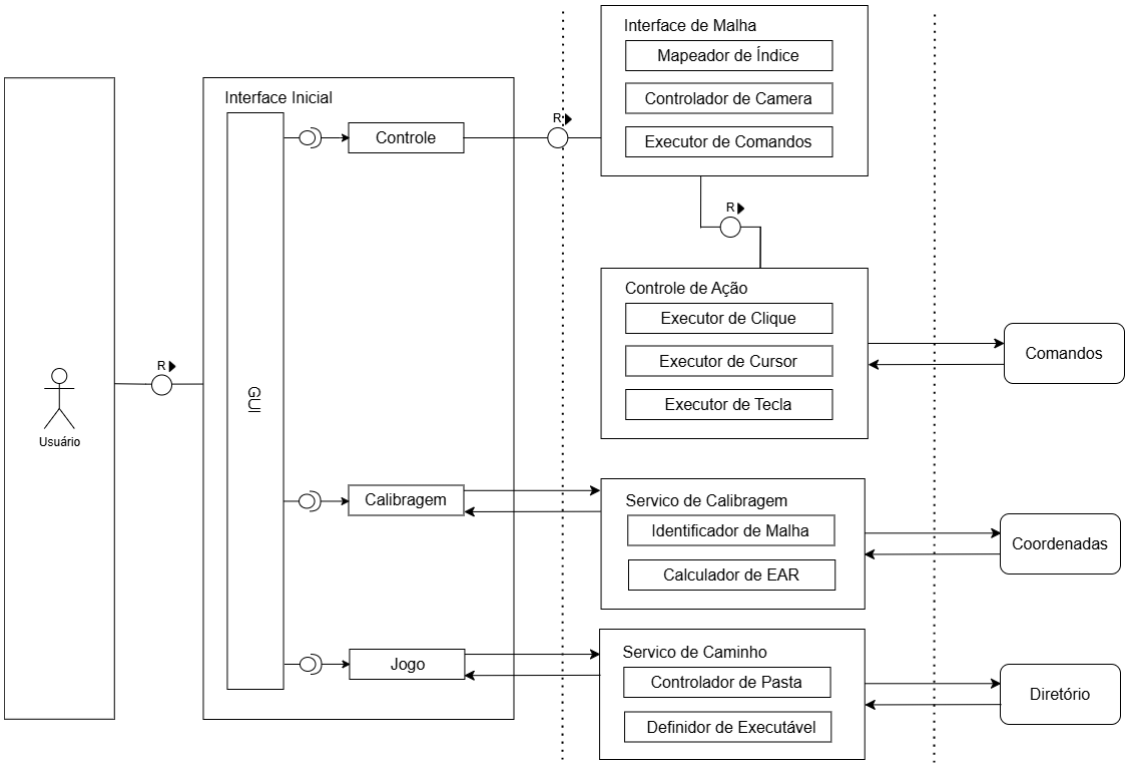
A utilização de Tkinter em aplicações de acessibilidade é valiosa devido à sua compatibilidade com as plataformas *Windows, macOS e Linux*, além de sua facilidade de integração com outros módulos Python. A biblioteca permite criar componentes de GUI

como botões, caixas de diálogo e menus de forma intuitiva. Essas características fazem de Tkinter uma ferramenta útil e flexível para o desenvolvimento de interfaces em programas de acessibilidade, tornando ela fácil de usar e manter.

3.2 Arquitetura

A interação do usuário é realizada por meio de piscadas, movimentos faciais e comandos de voz que são traduzidos em ações integradas diretamente ao jogo `Minecraft`. A escolha de trabalhar com o jogo Minecraft como estudo de caso para avaliação da usabilidade e eficácia da aplicação desenvolvida oferece uma oportunidade para testar a solução em um ambiente que é simultaneamente popular, acessível e versátil. A capacidade de moldar o jogo para incluir funcionalidades de acessibilidade específicas demonstra o potencial de Minecraft como uma ferramenta inclusiva para jogadores com deficiências físicas com baixo grau de destreza manual ou nenhuma. O jogo é conhecido por sua capacidade de engajar criativamente aos jogadores em um mundo aberto, servindo como um excelente teste para a aplicabilidade e robustez da tecnologia proposta (DUNCAN, 2011).

Figura 2 – Arquitetura Geral do ZenzeléPlay Baseada no Modelo TAM



Da autora, 2024

O sistema foi projetado com uma arquitetura modular utilizando a biblioteca Mediapipe para a detecção de landmarks faciais e o OpenCV para o processamento de imagens. Essa organização modular permite uma separação clara de responsabilidades entre os componentes do sistema, facilitando sua manutenção e escalabilidade. A modelagem foi estruturada com base na abordagem *Standardized Technical Architecture Modeling* (TAM), desenvolvida pela SAP para padronizar e facilitar a descrição de arquiteturas técnicas complexas (AG, 2004). A arquitetura está organizada em camadas funcionais bem definidas,

garantindo um fluxo eficiente de dados, comandos e conforme ilustrado na Figura 2, que apresenta os principais módulos, suas funções e as interações entre eles.

Na base da interação está o usuário, que utiliza uma *Graphical User Interface* (GUI) para enviar comandos ao sistema. Essa interface inicial captura entradas como piscadas, movimentos faciais e comandos de voz encaminhando essas informações para os módulos de controle. Em seguida, a interface de malha utiliza o *Mediapipe* para identificar *landmarks* faciais e processar as imagens capturadas pela câmera. Esse módulo é composto por três subcomponentes: o mapeador de índice, que identifica as *landmarks* faciais relevantes; o controlador de câmera, que gerencia o fluxo de imagens capturadas; e o executor de comandos, que transforma os dados processados em ações executáveis.

O controle de ações é responsável pela execução dos comandos baseados nas entradas do usuário. Esse módulo inclui o executor de cliques que interpreta piscadas para realizar cliques, o executor de cursor que traduz movimentos faciais em deslocamento do cursor, e o executor de teclas, que converte comandos de voz em entradas de teclado. Para melhorar a acurácia nas detecções, o serviço de calibragem realiza a calibração do sistema, esse serviço inclui o identificador de malha, que define as regiões faciais relevantes, e o calculador de *Eye Aspect Ratio* (EAR), que mede a proporção de abertura ocular para identificar piscadas com maior precisão.

Além disso, o sistema conta com um serviço de caminho, responsável por gerenciar dados relacionados ao jogo. Esse módulo permite salvar e carregar trajetos personalizados, contando com o controlador de pasta, que organiza os diretórios de armazenamento, e o definidor de executável, que garante o acesso correto aos arquivos do jogo. Todos os dados gerados pelos serviços de calibragem e controle de ações são encaminhados aos módulos de comandos e diretórios, permitindo sua integração direta com o jogo configurado.

Essa organização modular e hierárquica assegura que as funcionalidades do sistema sejam executadas de maneira eficiente e estruturada, proporcionando uma interação fluida entre o usuário e o ambiente de jogo.

3.2.1 Fluxo do Usuário

A modelagem do sistema foi baseada nos princípios da metodologia Prometheus, utilizada para a definição e estruturação de sistemas multiagentes. Essa metodologia fornece diretrizes para a descrição de componentes e suas interações, promovendo a padronização no desenvolvimento de sistemas (WINIKOFF; PADGHAM, 2004).

O fluxo do usuário foi projetado para garantir uma interação contínua e fluída com o sistema. O processo se inicia com a etapa de calibração das piscadas, um ajuste para parametrizar as características individuais do usuário, como a taxa de fechamento ocular e o padrão de piscadas. Em seguida, o sistema captura e interpreta os movimentos faciais e os comandos de voz do usuário, convertendo-os em ações específicas para o controle do jogo. Esse fluxo operacional permite que as entradas do usuário sejam processadas em tempo real, proporcionando uma experiência interativa e responsiva.

A Figura 4 apresenta o fluxo de interação do usuário no sistema, evidenciando as conexões entre os componentes e o trânsito de dados entre eles. O fluxo começa com o usuário, que interage com o sistema por meio da GUI, utilizando piscadas, movimentos do rosto e comandos de voz. Essas entradas são capturadas e encaminhadas aos módulos responsáveis pela interpretação das ações.

Na modelagem do sistema, os agentes, representados por símbolos padronizados de

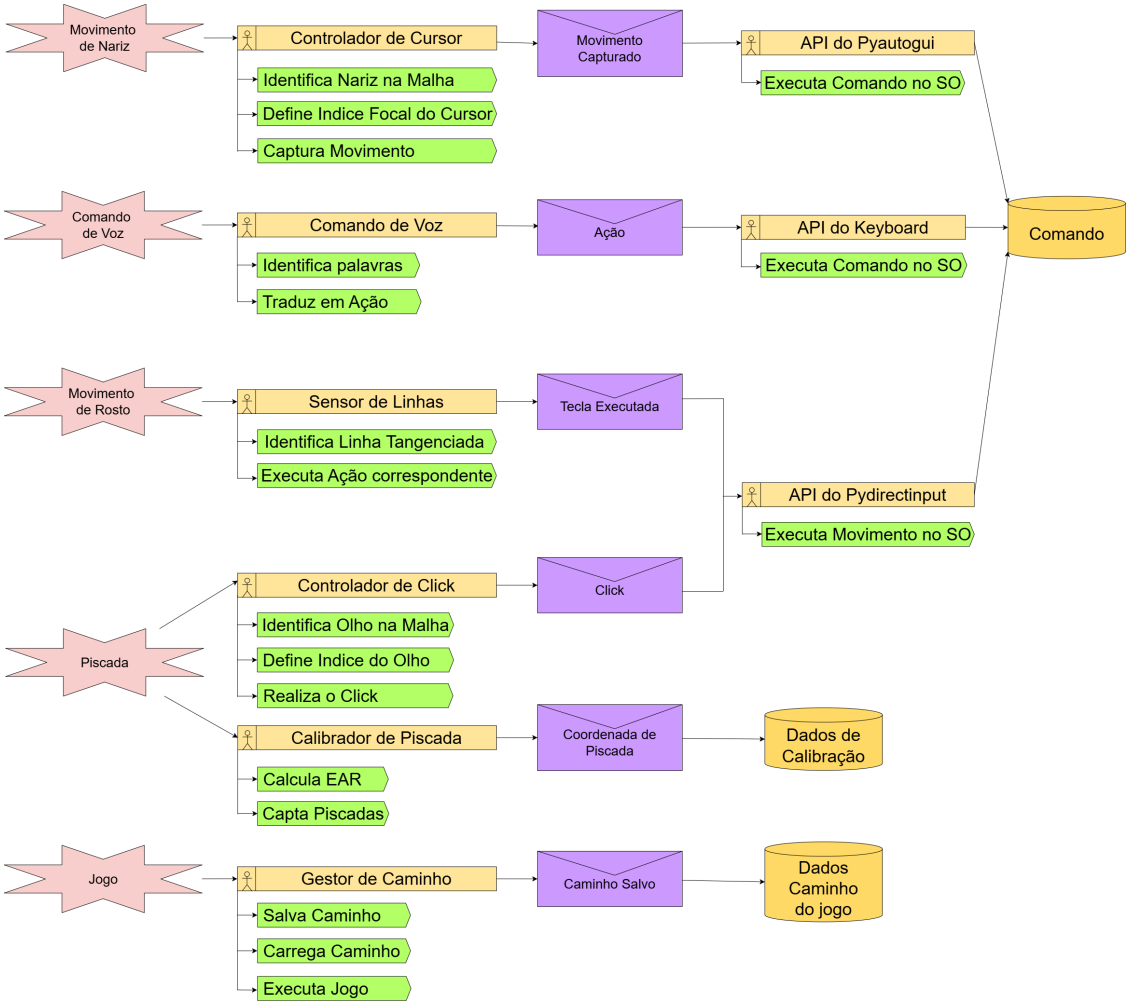
acordo com a notação da Figura 3, desempenham um papel central na coordenação do fluxo de informações. As percepções capturadas do usuário, como comandos de entrada, estão simbolizadas por elementos em formato de estrela na cor vermelha, constituindo o ponto de partida do fluxo. Esses dados iniciais são processados pelos agentes, que traduzem as percepções em comandos específicos, representados por elementos em formato de losango verde, e enviam essas ações para o sistema operacional ou diretamente ao jogo configurado. As mensagens trocadas entre os agentes, que estruturam a comunicação e o fluxo de dados, são representadas por envelopes na cor roxa.

Figura 3 – Notação utilizada no Diagrama de Visão Geral do Sistema

Fonte: (WINIKOFF; PADGHAM, 2004)

Essa organização destaca a progressão lógica do fluxo de dados, desde a captura inicial até a execução de comandos no ambiente do jogo. A integração entre os módulos é estruturada de maneira a garantir que cada etapa do processo seja realizada de forma sincronizada. A representação gráfica na Figura 4 evidencia a dinâmica das interações entre os componentes, bem como a relevância de cada etapa no fluxo de interação do usuário.

Figura 4 – Fluxo de Interação do Usuário no Sistema ZenzeléPlay



Da autora, 2024.

3.2.2 Interface do Usuário

A interface do usuário foi desenvolvida utilizando a biblioteca `Tkinter`, com o objetivo de proporcionar uma interação intuitiva e simplificada para o controle do sistema de reconhecimento facial no contexto de jogos. A interface inclui três botões principais para funções de calibração, carregamento do jogo e configuração do caminho do jogo, além de uma área de visualização da câmera, que permite ao usuário monitorar o enquadramento e o funcionamento da detecção facial, como evidenciado na Figura 5.

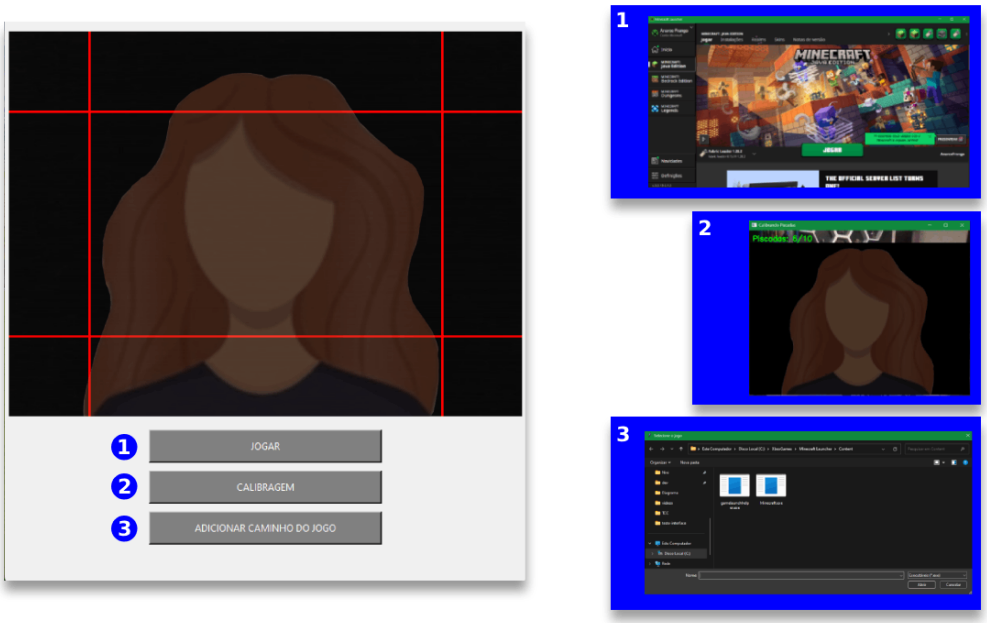
Na primeira execução da aplicação, o processo de calibração é iniciado automaticamente, focando na detecção de piscadas para estabelecer um padrão personalizado para o usuário. Esse processo utiliza o cálculo do EAR para identificar a taxa de abertura e fechamento dos olhos, com o objetivo de evitar o reconhecimento de piscadas que estejam fora do padrão natural do usuário. Esse ajuste é essencial para garantir que apenas os movimentos faciais intencionais sejam interpretados como comandos de jogo, evitando interferências indesejadas.

Após a calibração inicial das piscadas, o usuário deve configurar o caminho do jogo ao pressionar o botão “Adicionar Caminho do Jogo”. Esse botão abre uma janela de seleção

de arquivos, permitindo que o usuário localize e selecione o executável do jogo desejado. O caminho configurado é armazenado no sistema, facilitando o carregamento automático do jogo em futuras execuções e simplificando o processo de inicialização.

Com o caminho do jogo configurado e a calibração das piscadas concluída, o usuário pode iniciar o jogo ao pressionar o botão "Jogar". Neste ponto, o controle por voz é ativado, permitindo que o usuário interaja com o jogo utilizando tanto os movimentos faciais calibrados quanto comandos de voz. A interface permanece acessível durante toda a sessão, possibilitando ao usuário recalibrar ou ajustar o caminho do jogo conforme necessário, garantindo uma experiência de interação contínua e adaptável ao longo do uso.

Figura 5 – Usabilidade da interface



Fonte: Da Autora, 2024.

3.2.3 Calibração de Piscadas

A calibração das piscadas é uma etapa essencial para ajustar o limiar que define o fechamento dos olhos durante o uso do sistema. Este processo utiliza o cálculo do EAR, uma métrica que avalia a relação entre a largura e a altura dos olhos, permitindo identificar piscadas ou estados de olhos fechados em tempo real (MEHTA et al., 2019). O Algoritmo 1 descreve o fluxo de calibração, onde o sistema captura piscadas voluntárias do usuário, calcula o EAR médio e determina seu desvio padrão.

Algorithm 1 Calibração de Piscadas

```
function CALIBRATE_BLINKS(face_mesh, cap, left_eye_indices, right_eye_indices)
    blink_counter ← 0, ear_values ← [], eye_closed ← False
    while blink_counter < num_blinks do
        Capturar imagem da câmera
        Processar imagem com face_mesh
        if Landmarks faciais detectados then
            Calcular left_ear e right_ear com eye_aspect_ratio()
            ear ← (left_ear + right_ear)/2.0
            if ear < EYE_AR_THRESH e eye_closed ≠ True then
                Iniciar contagem de tempo da piscada
                eye_closed ← True
            else if ear ≥ EYE_AR_THRESH e eye_closed = True then
                Calcular duração da piscada
                if duração da piscada > MIN_BLINK_DURATION then
                    Incrementar blink_counter
                    Armazenar ear em ear_values
                end if
                eye_closed ← False
            end if
        end if
    end while
    return Média e Desvio Padrão de ear_values
end function
```

Durante o processo de calibração, o sistema realiza uma série de etapas para ajustar-se ao padrão natural de piscadas do usuário. Inicialmente, o sistema captura imagens da câmera e processa cada quadro utilizando o modelo `face_mesh`, responsável pela detecção de landmarks faciais. Em seguida, o método `EYE_AR_THRESH` calcula o EAR para cada olho, com base nos landmarks detectados nos olhos esquerdo e direito.

O valor médio do EAR é então comparado a um limiar predefinido (`EYE_AR_THRESH`) para determinar o estado dos olhos, indicando se estão abertos ou fechados. Quando o EAR fica abaixo do limiar, o sistema registra o início da piscada e inicia a contagem de tempo. Caso o EAR volte a ultrapassar o limiar, o sistema calcula a duração da piscada. Se essa duração exceder o tempo mínimo permitido (`MIN_BLINK_DURATION`), a piscada é registrada, o contador é incrementado e o valor do EAR é armazenado para análise posterior. Essa etapa assegura que o sistema ajuste suas detecções com base no padrão individual do usuário, permitindo uma interação mais precisa e adaptada às características de cada pessoa.

3.2.4 Controle de Movimento do Rosto

O controle de movimento do rosto é uma funcionalidade essencial para proporcionar uma experiência interativa aos usuários sem o uso das mãos. Utilizando *landmarks* faciais, o sistema consegue mapear os movimentos do nariz para controlar o cursor e ativar comandos de movimentação no jogo. Essa abordagem permite que o usuário controle seu avatar ou elemento do jogo de forma intuitiva e contínua, simplesmente movimentando o rosto.

Com o `Mediapipe` configurado, o sistema utiliza a função `move_cursor` para mapear o movimento do rosto ao cursor na tela. A função utiliza o método de média móvel exponencial para suavizar os movimentos do cursor, reduzindo o impacto de tremores ou movimentos abruptos do rosto, resultando em um controle mais natural e confortável para o usuário. O parâmetro `sensitivity` permite ajustar a intensidade dos movimentos, enquanto `alpha` controla a suavidade.

A função `move_cursor` calcula novas coordenadas para o cursor com base nos movimentos do rosto, aplicando a suavização e ajustes de sensibilidade, o que facilita o controle preciso. O uso da média móvel exponencial ajuda a eliminar ruídos de pequenos movimentos, como tremores, proporcionando uma experiência mais controlada e intuitiva.

3.2.5 Comandos de Direção

Além do controle de cursor, o sistema inclui a função `draw_border_commands`, que desenha linhas de borda na imagem da câmera para indicar áreas de ativação de comandos de movimento. Essas áreas de borda funcionam como "zonas de comando" que, ao serem atingidas pelos *landmarks* faciais, acionam comandos direcionais, permitindo ao usuário mover o avatar no ambiente do jogo.

Essa função avalia continuamente a posição dos landmarks faciais em relação às bordas da tela. Quando o usuário move o rosto para as áreas próximas às bordas, os comandos de movimentação (`W`, `A`, `S`, `D`) são acionados, permitindo controle direcional intuitivo. Este método de controle direcional usando zonas de borda é especialmente útil para movimentos mais amplos e pode ser ajustado para diferentes níveis de sensibilidade, dependendo das necessidades do usuário.

3.2.6 Comandos de Voz

O sistema utiliza uma função de reconhecimento de voz para interpretar comandos específicos e transformá-los em ações no jogo, tais como correr, andar e agachar. O reconhecimento de voz é executado continuamente em uma *thread* separada, permitindo uma resposta em tempo real às instruções do usuário.

A função monitora continuamente o áudio em busca de comandos predefinidos. Quando um comando é identificado, ele é traduzido em ações correspondentes no jogo, permitindo ao jogador controlar o personagem sem o uso das mãos, o que representa um avanço significativo em termos de acessibilidade para usuários com restrições motoras. Esses comandos incluem não apenas movimentos básicos, como andar e agachar, mas também ações contextuais mais complexas, como acessar o menu ou realizar um "pulo contínuo". Cada comando é projetado para facilitar a interação com o ambiente do jogo, permitindo que o usuário tenha uma experiência fluida e imersiva.

Esses comandos de voz foram projetados para simplificar a interação do usuário com o jogo, eliminando a necessidade de controles físicos e permitindo que jogadores com diferentes níveis de mobilidade participem de maneira autônoma e acessível. A variedade de comandos possibilita ao usuário realizar uma ampla gama de ações que cobrem desde movimentos básicos até comandos específicos, como acessar o menu e realizar movimentos estratégicos. Dessa forma, a interface de voz oferece uma alternativa eficaz e prática para a interação no jogo, ampliando as possibilidades de acessibilidade e inclusão digital.

4 Resultados

Os resultados obtidos foram analisados com base em duas abordagens principais: os dados quantitativos extraídos dos logs gerados durante os testes, que registraram o número de comandos emitidos, taxas de sucesso e falhas, e os dados qualitativos coletados por meio de um formulário aplicado aos participantes. Essa metodologia permitiu avaliar aspectos técnicos, como a precisão dos comandos bem como a experiência de uso relatada pelos participantes.

4.1 Usuários e Missões

O protótipo desenvolvido denominado ZenzeléPlay, cujo nome *Zenzelé* com origem na língua Zulu uma das principais línguas bantu da África do Sul, carrega como significado "faça você mesmo". No contexto do protótipo, o termo reflete a capacidade dos jogadores de agir de forma independente ao interagir com o ambiente do jogo, promovendo o controle por meio de comandos alternativos.

Os testes do protótipo envolveram dois grupos de usuários. O Grupo 1 foi composto por participantes com deficiência motora (PCD), sendo os Usuários 1, 2 e 3 que apresentavam condições como paralisia cerebral e dificuldades de mobilidade fina. Já Grupo 2 foi formado por participantes sem deficiência (não PCD), correspondendo aos Usuários 4, 5 e 6. Essa divisão permitiu avaliar o desempenho do sistema em diferentes condições de uso, bem como identificar as necessidades específicas de cada grupo de usuário.

Para avaliar a funcionalidade do sistema, foram propostas quatro missões no ambiente do jogo *Minecraft*:

- **Missão 1**: Coletar itens básicos, como madeira e pedra, utilizando comandos de movimento e interação.
- **Missão 2**: Criar itens específicos, como ferramentas ou blocos, utilizando comandos de menu e seleção.
- **Missão 3**: Coletar comida, interagindo com o ambiente do jogo para garantir recursos de sobrevivência.
- **Missão 4**: Construir um abrigo, tarefa opcional que exigia o uso de comandos avançados para posicionamento e interação.

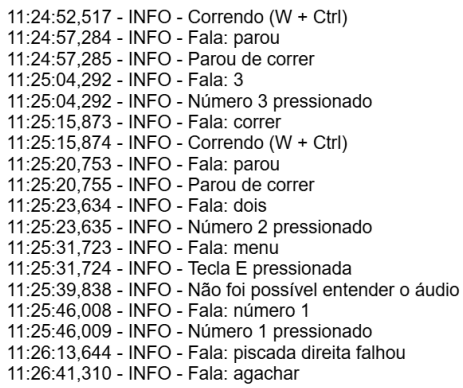
Todos os participantes concluíram as missões propostas, incluindo a construção do abrigo considerada opcional. Durante os testes foram avaliados aspectos como eficácia, precisão e usabilidade do sistema. Além de coletar feedback sobre os desafios encontrados e sugestões para melhorias futuras a análise desses dados permitiu verificar a adaptação dos jogadores ao controle alternativo bem como identificar o potencial da aplicação para uso regular em ambientes de entretenimento.

4.2 Desempenho Técnico

Os dados quantitativos registrados nos logs foram sintetizados na Tabela 1, que apresenta o número de comandos emitidos, as execuções bem-sucedidas e as falhas observadas. Esses registros foram captados durante os testes abrangendo as ações realizadas pelos

participantes, em que cada comando acionado foi registrado incluindo casos de sucesso e falha, como a Figura 6 apresenta.

Figura 6 – Registro do log



Fonte: Da Autora, 2024.

No entanto, as interações relacionadas às piscadas, movimentação por linhas e controle do cursor foram reportadas verbalmente pelos usuários considerando a impossibilidade de registro automático para esses casos específicos. No Grupo 1 composto pelos participantes com deficiência motora, os Usuários 1 e 3 enfrentaram dificuldades significativas na utilização de cliques e movimentos por linhas, o que limitou a validação dessas ações.

Tabela 1 – Precisão por Usuário

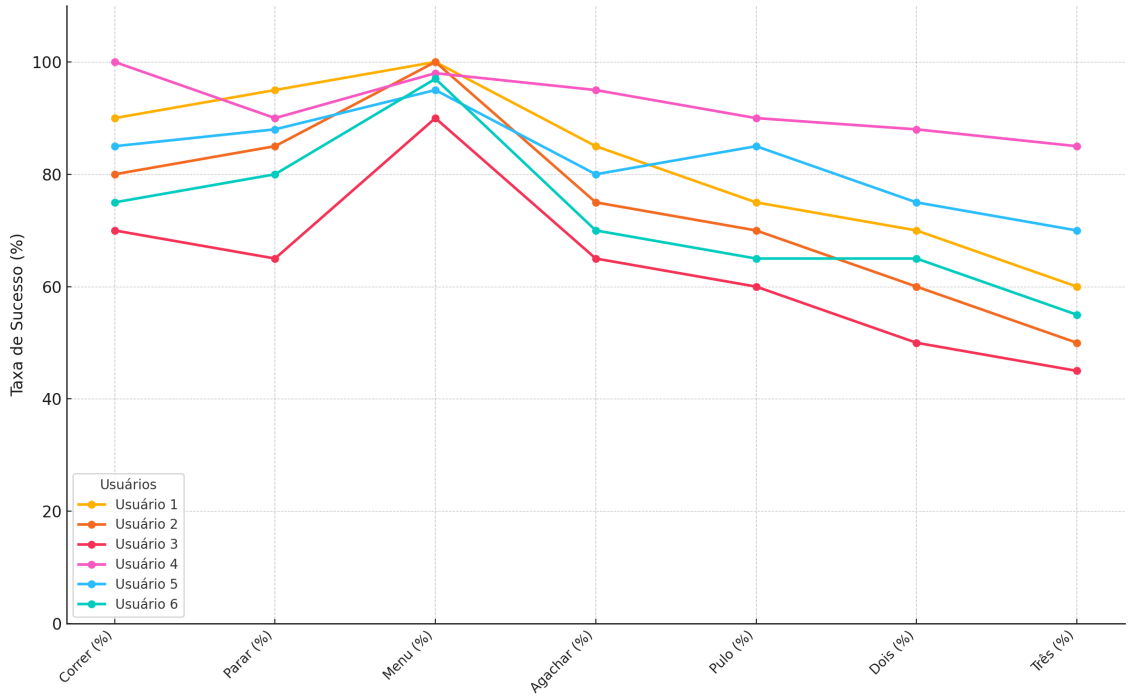
Usuário	**Comandos**	**Sucessos**	**Falhas**	**Precisão (%)**
PCD				
Usuário 1	36	24	12	66,67
Usuário 2	32	21	11	65,63
Usuário 3	45	15	30	33,33
Não PCD				
Usuário 4	88	70	18	79,55
Usuário 5	45	29	16	64,44
Usuário 6	42	28	14	66,67
Total	288	187	101	64,93

Fonte: Da autora, 2024.

Conforme ilustrado na Figura 7, a taxa de precisão variou significativamente entre os usuários, refletindo diferentes níveis de interação com os comandos implementados. O Usuário 3 apresentou a menor precisão (33,33%), indicando dificuldades relacionadas a limitações motoras que afetaram diretamente a execução de comandos, especialmente os que exigiam maior precisão, como cliques e comandos faciais. Por outro lado, o Usuário 4

obteve a maior taxa de sucesso (79,55%), evidenciando uma adaptação mais eficiente ao sistema e maior facilidade na execução dos comandos.

Figura 7 – Taxa de Sucesso por Tipo de Comando



Fonte: Da autora, 2024.

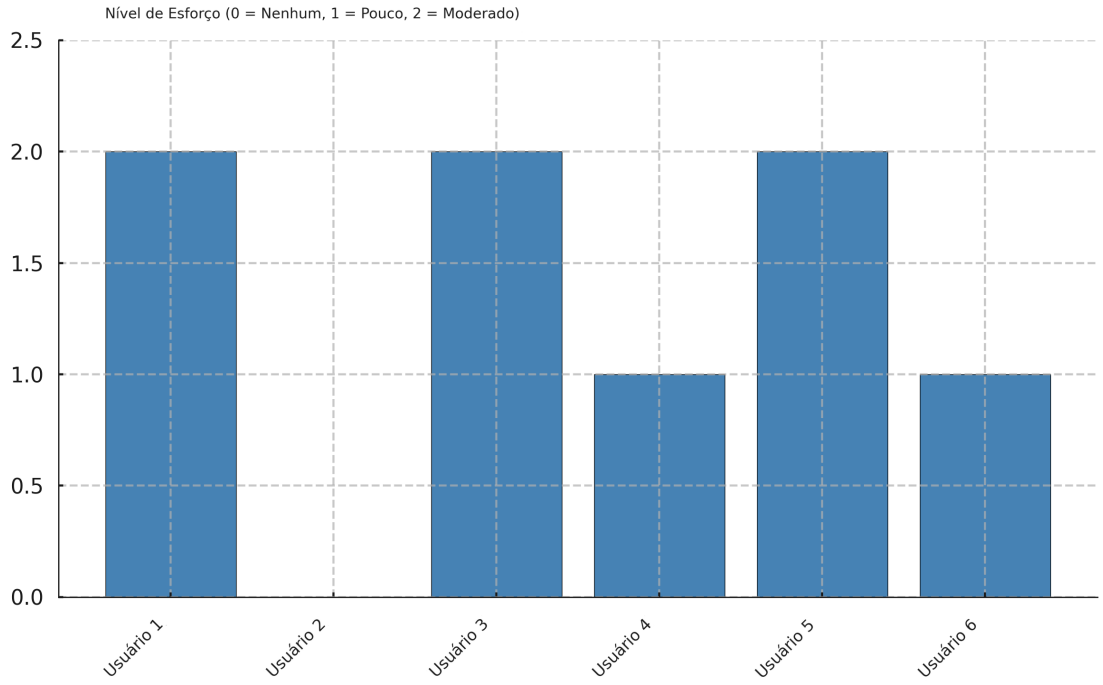
4.3 Esforço e experiência de uso

Os resultados qualitativos, avaliados por meio do formulário aplicado, forneceram uma análise do esforço físico e cognitivo necessário para o uso do sistema, bem como da facilidade de interação percebida pelos participantes. A Figura 8 apresenta os níveis de esforço relatados pelos usuários, revelando disparidades significativas entre os participantes com deficiência (PCD) e aqueles sem deficiência (não PCD).

Os dados indicam que os usuários PCD relataram níveis de esforço físico e concentração mais elevados, principalmente em comandos que exigiam maior precisão ou movimentos repetitivos, como cliques e comandos faciais. Essa tendência reflete a necessidade de ajustes ergonômicos e melhorias na calibragem, especialmente para atender às limitações motoras e cognitivas desse grupo. Em contrapartida, os usuários não PCD relataram esforço moderado ou baixo.

O nível de esforço, conforme ilustrado na Figura 8, variou significativamente entre os participantes, com destaque para os Usuários 1 e 3 (PCD), que enfrentaram maiores dificuldades em tarefas relacionadas à detecção de piscadas e ao controle facial. Esses usuários relataram cansaço físico, especialmente em sessões prolongadas de uso, indicando a necessidade de reduzir a carga cognitiva associada à interação.

Figura 8 – Nível de Esforço Necessário para o Uso da Aplicação

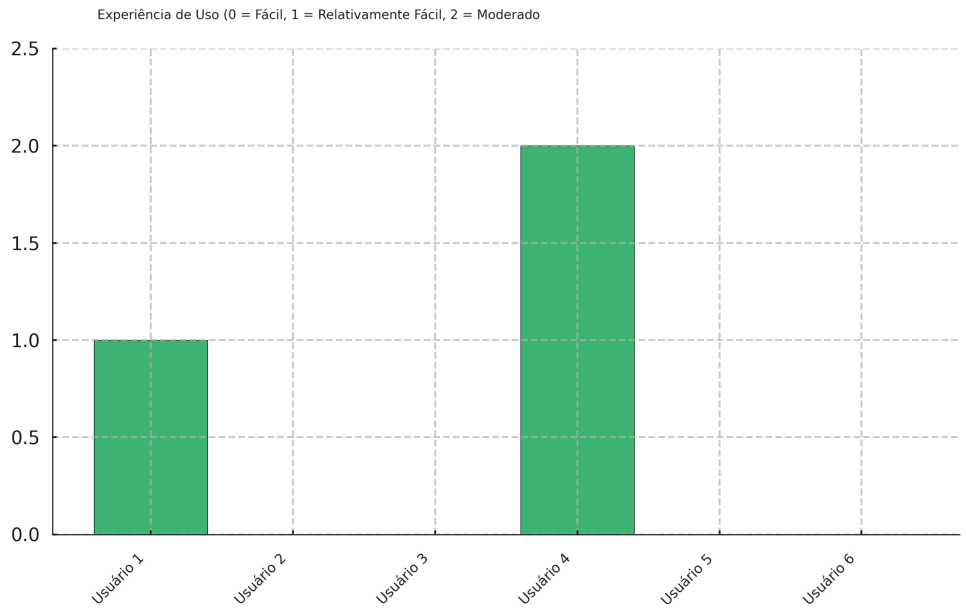


Fonte: Da autora, 2024.

A análise da experiência de uso classificada em “fácil”, “relativamente fácil” e “moderado”, revelou resultados interessantes. Conforme apresentado na Figura 9, os usuários PCD (Usuários 1, 2 e 3) avaliaram a aplicação como “fácil”, enquanto os usuários não PCD (Usuários 4, 5 e 6) classificaram a experiência como “relativamente fácil” ou “moderado”. Esse resultado indica que, embora os comandos faciais e de voz tenham sido projetados para oferecer acessibilidade, os usuários sem deficiência enfrentaram desafios inesperados na interação com o sistema.

Esse comportamento pode ser atribuído à adaptação mais rápida dos usuários PCD às ferramentas assistivas, possivelmente devido à familiaridade prévia com soluções acessíveis, conforme evidenciado no estudo de Goldau et al. (2024), em que avaliou um controle adaptativo em aplicações assistivas para indivíduos com mobilidade limitada nos membros superiores. Os resultados indicaram que usuários com experiência anterior em tecnologias assistivas demonstraram uma curva de aprendizado mais rápida ao interagir com novos sistemas. Em contrapartida, usuários não PCD podem ter encontrado maior dificuldade em utilizar uma interface que se afasta dos controles tradicionais, o que reforça a importância de considerar a curva de aprendizado associada a essas soluções.

Figura 9 – Classificação da Experiência de Uso

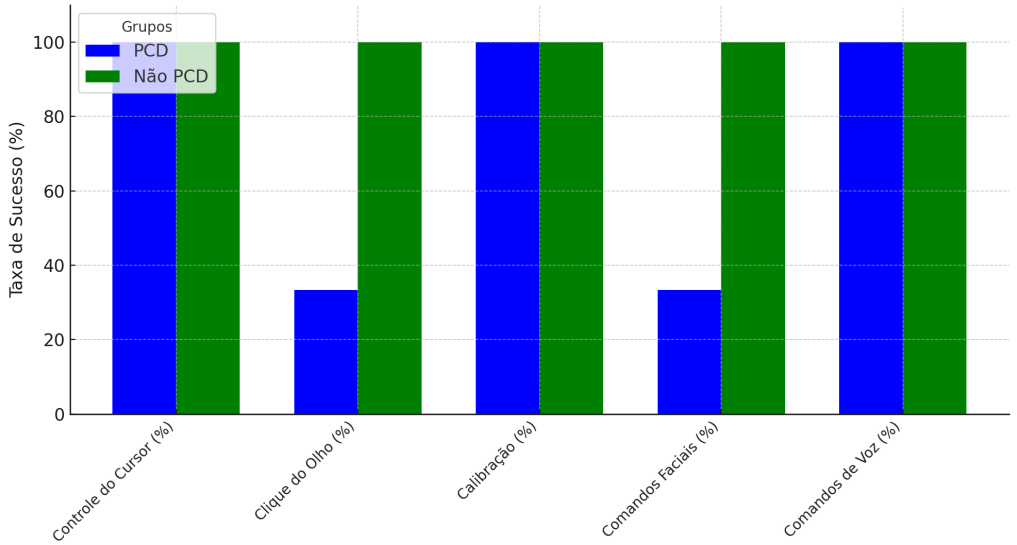


Fonte: Da autora, 2024.

4.4 Comparação de comandos

A Figura 10 apresenta a análise da precisão por tipo de comando. Os comandos de voz obtiveram maior precisão e facilidade de execução, enquanto comandos faciais como piscadas apresentaram maior dificuldade, especialmente para os usuários PCD.

Figura 10 – Taxa de Precisão por Tipo de Comando



Fonte: Da autora, 2024.

5 Discussão e Conclusões

Os resultados obtidos indicam que o protótipo desenvolvido apresenta potencial para ser utilizado como ferramenta de acessibilidade em jogos eletrônicos. Contudo, foram identificadas limitações técnicas em algumas funcionalidades que demandam aprimoramentos, a análise da taxa de sucesso por comando conforme ilustrado na Figura 7, demonstrou maior eficácia nos comandos vocais em comparação aos comandos faciais. Problemas como atraso na resposta e dificuldades na execução de cliques evidenciam a necessidade de ajustes técnicos no sistema.

Os logs registrados indicaram que comandos como *parar*, *correr* e acesso ao *menu* foram os mais utilizados pelos usuários, sendo fundamentais para o controle do jogo e navegação em ambientes virtuais. Apesar de todos os comandos estarem disponíveis e terem sido acionados ao menos uma vez durante os testes, a predominância do uso desses comandos mais básicos reflete sua relevância para uma interação prática e intuitiva no contexto do jogo. Esses dados apontam para a necessidade de aprimorar a fluidez e confiabilidade desses comandos básicos, garantindo uma experiência mais funcional para o usuário.

Além disso, a análise do esforço e da experiência de uso, apresentada nas Figuras 8 e 9 revelou que os comandos faciais exigiram maior esforço físico e cognitivo dos usuários PCD, reforçando a importância de ajustes ergonômicos e tecnológicos no sistema. Essas dificuldades destacam a necessidade de reduzir a carga de interação associada a esses comandos para torná-los mais inclusivos.

A Tabela 2 apresenta uma comparação entre a aplicação desenvolvida e outras soluções de acessibilidade em jogos. O protótipo se diferencia por integrar reconhecimento facial, comandos por gestos e comandos de voz em uma interface personalizável para jogos eletrônicos. Apesar disso, a precisão geral média de 64,93% (Tabela 1) indica que há espaço para melhorias, especialmente nas funcionalidades dependentes de comandos faciais que enfrentaram maior taxa de falhas.

Tabela 2 – Comparação das características de acessibilidade das soluções analisadas

Solução	**Reconhecimento Facial**	**Comandos por Gestos**	**Voz**	**Interface de Jogo**	**Customizável**
Westin (2004)	Não	Não	Não	Sim	Não
Rand, Kizony e Weiss (2004)	Não	Sim	Não	Não	Não
Fanucci, Iacopetti e Roncella (2011)	Não	Não	Não	Sim	Parcial
Tu, Kao e Lin (2013)	Sim	Sim	Não	Não	Não
Phan et al. (2022)	Não	Sim	Não	Não	Não
Mendes et al. (2023)	Não	Sim	Não	Não	Parcial
Trabalho Desenvolvido	**Sim**	**Sim**	**Sim**	**Sim**	**Sim**

Fonte: Da autora, 2024.

As análises realizadas permitem concluir que o sistema proposto alcançou os objetivos da pesquisa, ao oferecer uma alternativa acessível para usuários com diferentes graus de limitações motoras. No entanto, os dados também evidenciam que melhorias são necessárias para ampliar a eficiência e a experiência de uso, especialmente em comandos que exigem maior precisão e na redução do esforço físico associado às interações. Essas melhorias poderão contribuir para consolidar o protótipo como uma ferramenta mais robusta e inclusiva no contexto dos jogos eletrônicos.

5.1 Trabalhos Futuros

Recomenda-se os seguintes aprimoramentos para aumentar a eficiência e acessibilidade do protótipo:

- Implementação de algoritmos mais robustos para reconhecimento facial e de voz, visando maior precisão e consistência;
- Otimização do processo de calibragem para reduzir o esforço físico e cognitivo dos usuários, tornando a aplicação mais intuitiva e eficiente;
- Expansão dos testes para um público mais diversificado, contemplando diferentes tipos de limitações motoras e perfis de experiência;
- Redução do atraso nos comandos faciais, promovendo uma interação mais fluida e responsiva;
- Adaptação da aplicação para suportar outros jogos além do *Minecraft*, ampliando as possibilidades de uso e inclusão;
- Implementação de um sistema de edição de comandos, permitindo que todos os comandos sejam personalizáveis e opcionais, garantindo flexibilidade para jogar diferentes tipos de jogos e se adaptar aos usuários com diferentes necessidades.

Essas iniciativas têm como objetivo contribuir para o desenvolvimento de uma ferramenta mais eficiente e inclusiva, promovendo maior acessibilidade e personalização no contexto dos jogos eletrônicos.

Application of Artificial Intelligence in the Accessibility of the Game Minecraft: Development of a prototype for Facial and Voice Control

Jakelyny Sousa de Araújo* Prof. Dr. Roberto Wiest†
Prof. Me. José Antônio Oliveira de Figueiredo‡

17 de dezembro de 2024

Abstract

This article presents an accessibility solution for electronic games aimed at users with motor limitations through the development of a facial and sound control system based on artificial intelligence. The application was implemented in Python and translates facial and voice commands into in-game actions, replacing traditional controls and allowing individuals with limited mobility or manual dexterity to interact autonomously. The system was tested within the Minecraft game environment, where participants carried out specific missions: collecting items, creating objects, foraging and building shelter. These tasks made it possible to evaluate the system's functionality in different scenarios and collect metrics on command accuracy, physical and cognitive effort, as well as user satisfaction. The results indicated an overall accuracy of 64.93%, with users with motor disabilities experiencing greater difficulties with facial commands, while voice commands demonstrated better performance. All users completed the proposed missions, indicating the system's functionality for different player profiles. Although the results demonstrated the system's functionality, opportunities for improvement were identified, particularly in reducing physical effort and improving the fluidity of click and facial control functionalities. This study reinforces the feasibility of systems based on facial and vocal control as inclusive tools, expanding access and interaction for people with motor disabilities in electronic games.

Key-words: Accessibility. Electronic Games. Inclusion.

*Aluno do curso de Ciência da Computação, Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo (jakelynysousa.a@gmail.com).

†Orientador do trabalho (Professor e Coordenador do curso de Ciência da Computação do Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo; robertowiest@ifsul.edu.br)

‡Co-orientador do trabalho (Professor do curso de Ciência da Computação e Coordenador NAPNE do Instituto Federal de Educação Ciência e Tecnologia, Sul-rio-grandense, Campus Passo Fundo; josefigueiredo@ifsul.edu.br)

Referências

ABADI, M. et al. {TensorFlow}: a system for {Large-Scale} machine learning. In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. [S.l.: s.n.], 2016. p. 265–283. Citado na página 5.

AG, S. *SAP TAM: Technical Architecture Modeling Standard*. 2004. Disponível em: <https://www.fmc-modeling.org/download/fmc-and-tam/SAP-TAM_Standard.pdf>. Citado na página 9.

BRADSKI, G. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, Miller Freeman Inc., v. 25, n. 11, p. 120–123, 2000. Citado na página 5.

DUNCAN, S. C. Minecraft, beyond construction and survival. *Well Played Journal*, Carnegie Mellon University, 2011. Citado na página 9.

FANUCCI, L.; IACOPETTI, F.; RONCELLA, R. A console interface for game accessibility to people with motor impairments. In: *2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*. [S.l.: s.n.], 2011. p. 206–210. Citado 2 vezes nas páginas 4 e 21.

FAVA, F. Jogando com o ar: o sopro como instrumento de acessibilidade nos jogos eletrônicos. *SBGAMES*, v. 7, p. 115–121, 2008. Citado 2 vezes nas páginas 2 e 3.

GOLDAU, F. F. et al. Adaptive control in assistive application–a study evaluating shared control by users with limited upper limb mobility. *arXiv preprint arXiv:2406.06103*, 2024. Citado na página 19.

GOOGLE. *MediaPipe: Cross-platform, customizable ML solutions for live and streaming media*. 2021. Disponível em: <https://mediapipe.dev/>. Acesso em: junho de 2024. Citado na página 6.

GUPTA, S. et al. The future is yesterday: Use of ai-driven facial recognition to enhance value in the travel and tourism industry. *Information Systems Frontiers*, Springer, v. 25, n. 3, p. 1179–1195, 2023. Citado na página 3.

HANGARAGI, S.; SINGH, T.; NEELIMA, N. Face detection and recognition using face mesh and deep neural network. *Procedia Computer Science*, Elsevier, v. 218, p. 741–749, 2023. Citado na página 6.

KOSCH, T. et al. A survey on measuring cognitive workload in human-computer interaction. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 55, n. 13s, jul. 2023. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3582272>. Citado na página 2.

KUMAR, A.; VERMA, S.; MANGLA, H. A survey of deep learning techniques in speech recognition. In: IEEE. *2018 international conference on advances in computing, communication control and networking (ICACCCN)*. [S.l.], 2018. p. 179–185. Citado na página 3.

LEARNCODEBYGAMING. *PyDirectInput: Python mouse and keyboard input automation for Windows using Direct Input*. 2021. Disponível em: <https://github.com/learncodebygaming/pydirectinput>. Citado na página 7.

LUGARESI, C. et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. Citado na página 3.

LUNA-JIMÉNEZ, C. et al. Interpreting sign language recognition using transformers and mediapipe landmarks. In: *Proceedings of the 25th International Conference on Multimodal Interaction*. [S.l.: s.n.], 2023. p. 373–377. Citado na página 6.

LUNDH, F. An introduction to tkinter. *URL: www. pythonware. com/library/tkinter/introduction/index. htm*, v. 539, p. 540, 1999. Citado na página 8.

MAHMOOD, N. T.; JABBAR, M. S.; ABDALRAZAK, M. A real-time hand gesture recognition based on media-pipe and support vector machine. In: SPRINGER. *International Conference of Reliable Information and Communication Technology*. [S.l.], 2023. p. 263–273. Citado na página 5.

MALHOTRA, P. et al. [retracted] deep neural networks for medical image segmentation. *Journal of Healthcare Engineering*, Wiley Online Library, v. 2022, n. 1, p. 9580991, 2022. Citado 2 vezes nas páginas 6 e 7.

MATHEW, A. R.; HAJJ, A. A.; ABRI, A. A. Human-computer interaction (hci): An overview. In: *2011 IEEE International Conference on Computer Science and Automation Engineering*. [S.l.: s.n.], 2011. v. 1, p. 99–100. Citado na página 2.

MEHTA, S. et al. Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio. In: *Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India*. [S.l.: s.n.], 2019. Citado na página 13.

MENDES, L. C. et al. Rehabeelitation: the architecture and organization of a serious game to evaluate motor signs in parkinson's disease. *PeerJ Computer Science*, PeerJ Inc., v. 9, p. e1267, 2023. Citado 2 vezes nas páginas 4 e 21.

Microsoft. *Xbox Adaptive Controller*. 2018. <https://www.xbox.com/pt-BR/accessories/controllers/xbox-adaptive-controller>. Citado na página 3.

MYERS, B.; HUDSON, S. E.; PAUSCH, R. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM New York, NY, USA, v. 7, n. 1, p. 3–28, 2000. Citado na página 8.

NASSIF, A. B. et al. Speech recognition using deep neural networks: A systematic review. *IEEE access*, IEEE, v. 7, p. 19143–19165, 2019. Citado na página 7.

OMMEN, C. V. et al. Xbox adaptive controller oobe: Creating a fun and accessible video gaming experience for all! 2024. Citado na página 2.

OSSMANN, R.; ARCHAMBAULT, D.; MIESENBERGER, K. Accessibility issues in game-like interfaces. In: SPRINGER. *Computers Helping People with Special Needs: 11th International Conference, ICCHP 2008, Linz, Austria, July 9-11, 2008. Proceedings 11*. [S.l.], 2008. p. 601–604. Citado na página 2.

PHAN, K. N. et al. Game for upper limb rehabilitation using opencv. In: *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*. [S.l.: s.n.], 2022. p. 1–4. Citado 2 vezes nas páginas 4 e 21.

RAND, D.; KIZONY, R.; WEISS, P. Virtual reality rehabilitation for all: Vivid gx versus sony playstation ii eyetoy. *Virtual Reality Assoc. Tech*, 01 2004. Citado 2 vezes nas páginas 5 e 21.

SAFITLI, M. J.; AMINUDDIN, F. huda et al. Controlling computer device cursors using hand gestures by utilizing opencv, mediapipe, and pyautogui. *Intellecta: Journal of Artificial Intelligence*, v. 1, n. 1, p. 1–16, 2024. Citado na página 8.

SMITH, C. *Keyboard: A Python library for keyboard input automation*. 2018. Disponível em: <https://github.com/boppreh/keyboard>. Citado na página 8.

SMITH, E. T.; BASAK, C. A game-factors approach to cognitive benefits from video-game training: A meta-analysis. *PLoS One*, Public Library of Science San Francisco, CA USA, v. 18, n. 8, p. e0285925, 2023. Citado na página 2.

SPEECHRECOGNITION. *SpeechRecognition*. 2023. <https://pypi.org/project/SpeechRecognition/>. Citado na página 6.

TU, Y.-J.; KAO, C.-C.; LIN, H.-Y. Human computer interaction using face and gesture recognition. In: *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. [S.l.: s.n.], 2013. p. 1–8. Citado 2 vezes nas páginas 4 e 21.

TURK, M.; PENTLAND, A. Eigenfaces for recognition. *Journal of cognitive neuroscience*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 3, n. 1, p. 71–86, 1991. Citado 2 vezes nas páginas 5 e 6.

UÇAR, A. Deep convolutional neural networks for facial expression recognition. In: IEEE. *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. [S.l.], 2017. p. 371–375. Citado na página 3.

WESTIN, T. Game accessibility case study: Terraformers - a real-time 3d graphic game. 01 2004. Citado 2 vezes nas páginas 5 e 21.

WINIKOFF, M.; PADGHAM, L. The prometheus methodology. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, Springer, p. 217–234, 2004. Citado 2 vezes nas páginas 10 e 11.