

Acessibilidade em Jogo Digital via Comandos de Voz para Pessoas com Deficiência Motora

Matheus de Medeiros Pereira
Univerisdade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, Brasil
matheusmmp@hotmail.com

Lucas Debatin
Laboratório de Inteligência Aplicada
Univerisdade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, Brasil
lucasdebatin@univali.br

ABSTRACT

Ensuring accessibility is a mandatory issue by law, both in everyday life and in the context of recreational activities. In electronic games this is no different, to ensure accessibility for people with disabilities (PwD), developers and research are constantly seeking to innovate in technical and methodological issues. This work will present an electronic game of the 2D platform genre for the computer, which offers the user the possibility of controlling through voice commands, generating a form of accessibility for motor PwD. During testing, the API achieved an average margin of error of 14.72% per recognition. Tests were carried out with 21 people without disabilities, who differed by their intimacy with digital games. Based on the results obtained, it was possible to verify that the players performance when using the standard control was superior to the performance using voice controls. This allowed us to analyze that, although voice control is an accessibility option, it is not efficient when compared to the standard options (mouse and keyboard), as the performance of users was superior.

KEYWORDS

Voice recognition, Accessibility, Games, Design

1 Introdução

A acessibilidade consiste na condição de quebra das barreiras de acesso e a efetiva participação de pessoas nos vários âmbitos da vida social. Portanto, para promover a inclusão social, a acessibilidade é uma condição fundamental e imprescindível. Para isso, existem leis que garantem direitos de inclusão em diversos âmbitos e condições, permitindo que pessoas tenham igualdade de participação de maneira plena e efetiva na sociedade, seja na locomoção, comunicação ou forma com que ocorre o acesso a sistemas interativos [1].

Sistemas interativos estão integrados na sociedade, sendo presentes em diversas tarefas do dia a dia, incluindo os jogos, que proporcionam entretenimento. Nesses sistemas, a acessibilidade garante que os benefícios do design de interação sejam acessíveis a todos [2].

No âmbito dos sistemas interativos, o jogo pode ser definido como uma atividade lúdica, sendo ainda, um ato voluntário concretizado como evasão da vida real, onde elementos de

interação compõem a imersão. Sendo assim, a ausência do fator imersão, acarreta uma influência negativa para com o sistema [3].

Em contrapartida, diversas desenvolvedoras estão buscando a criação de ferramentas para melhorar a acessibilidade em jogos, porém muitas não abrangem o público de usuários que possuem deficiência motora, visto que possuem limitação em executar certos padrões gestuais [4]. Baseado nisso, o presente trabalho desenvolveu um jogo plataforma 2D para computador que proporciona ao usuário a possibilidade de controlá-lo por meio de comandos de voz, gerando uma forma de acessibilidade para pessoas com deficiência (PcDs) motora, contribuindo assim com a área de Interação Humano-Computador. Ao fim, por meio desse jogo, foi executado testes com usuários para entender diferentes perspectivas e opiniões referentes aos controles, por meio da voz e utilizando mouse e teclado.

Este trabalho tem o intuito de permitir que mais PcDs motora consigam ingressar no meio do entretenimento eletrônico, por meio da implementação do reconhecimento de voz dentro do jogo, garantindo assim uma jogabilidade mais fluida e prazerosa. O jogo utiliza de tecnologias já existentes e pouco exploradas nessa área de experiência do usuário e seu desenvolvimento impulsiona o estudo da utilização de ferramentas de acessibilidade voltadas para o mercado do desenvolvimento de jogos.

O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta a fundamentação teórica. A seção 3 detalha os trabalhos existentes relacionados ao projeto. A seção 4 descreve questões de desenvolvimento, estruturação e funcionamento. A seção 5 mostra os resultados dos experimentos realizados. Finalmente, a seção 6 apresenta as conclusões e os trabalhos futuros.

2 Fundamentação Teórica

Para um melhor entendimento do jogo proposto, esta seção tem como objetivo apresentar informações sobre Jogos Digitais e Reconhecimento de Voz.

2.1 Jogos Digitais

O conceito de jogo implica numa atividade integrada a um conjunto de regras, que a partir delas é definido um vencedor ou perdedor. As regras “estão relacionadas aos problemas impostos ao jogador” [5], sendo assim, cabe ao jogador a tomada de decisão aos cenários apresentados.

Parte do que torna jogos digitais uma mídia imersiva é a possibilidade de uma experiência televisiva singular, com o convite à participação, ao interagir com o jogo por meio da interface interativa, utilizando periféricos, como, por exemplo, o mouse e teclado [6].

Os jogos podem ser divididos em dois grupos principais relacionados aos tipos e gêneros. O primeiro grupo corresponde à capacidade de simulação que o jogo promove, podendo ser de ação, aventura, simulação, educativo, estratégia, entre outros. Já o segundo está relacionado à tecnologia envolvida e às ferramentas que os jogos utilizam, podendo variar tanto das plataformas que o jogo foi planejado, como, por exemplo, consoles, aparelhos móveis ou arcade, quanto nas tecnologias que o jogo utiliza, tais como, câmeras de aparelhos móveis para realidade aumentada ou um ambiente 3D totalmente interativo por meio do uso de óculos de realidade virtuais [7].

No campo de jogos que não buscam esses ultrarrealismos, a presença de um gráfico 3D se mostra não muito necessária, como em casos de jogos do gênero plataforma. Os jogos de plataforma com gráficos bidimensionais (2D) são um subgênero de jogos de ação caracterizado por um protagonista que se move e pula nas plataformas, coletando prêmios e eliminando inimigos de várias maneiras. Sendo assim, a presença de elementos ultrarrealistas não fazem parte dos requisitos obrigatórios para esse gênero [8].

Com base nisso, de acordo com Lewis et al. [9], o desenvolvimento de um jogo se baseia muito no uso da Engine, no qual se refere a uma coleção de módulos de código de simulação que não especificam diretamente o comportamento do jogo ou ambiente dele (dados do nível). As Engines incluem módulos de manipulação de entrada, saída (Renderização 3D, desenho 2D, som) e genéricos para física/dinâmica em diferentes cenários.

As diversas formas de controle correspondem aos inúmeros requisitos que os jogos digitais buscam cumprir, dentro dos convencionais, pode-se observar a preferência do uso de mouse e teclado como dispositivos de entrada de comando. Para casos em que é necessária maior precisão, a presença de controles com dois analógicos facilita o controle de câmera e movimentação.

2.2 Reconhecimento de Voz

A tecnologia de reconhecimento de voz está se tornando cada vez mais importante nas empresas de tecnologia, visto que a sua utilidade é melhorar as relações de usuários existentes e suportar novos meios da interação humana com computadores. A tecnologia de reconhecimento permite o uso “hands-free” (mãos livres) dos computadores, isto é, suporta o acesso fora da mesa de trabalho. Além disso, melhora a acessibilidade para usuários com inabilidade, podendo também reduzir o risco de ferimento repetitivo da tensão e outros problemas causados pelos meios de uso atuais, tais como, teclado e mouse [10].

Devido a vasta possibilidades de aplicação do reconhecimento de voz, sua metodologia foi subdividida em dois tipos, por meio da fala contínua ou da captação de palavras isoladas, a fim de otimizar o desempenho para tarefas específicas que não necessitam de um certo grau de complexidade. O que mais diferencia o sistema reconhecimento contínuo do isolado, é o

processo exercido pelo decodificador, no qual sua finalidade é procurar a melhor sequência de palavras num conjunto de hipóteses possíveis, de acordo com as características do sinal de voz, para formar uma frase, por exemplo. Esse tipo utiliza dois modelos: (i) acústico, que transforma o sinal que está sendo processado em palavras e sentenças; e (ii) de linguagem, que é responsável por identificar o idioma e reposicionar as palavras, excluindo frases gramaticalmente incorretas [11].

3 Trabalhos Relacionados

Esta seção tem como objetivo apresentar uma comparação entre os trabalhos de [12], [13], [14] e [15], expondo de forma sintética suas similaridades, diferenças e evidenciando a maneira com que o presente estudo se relaciona com eles.

Para selecionar esses trabalhos foram utilizadas palavras-chave com características semelhantes ao projeto, sendo elas: Speech Recognition, Accessibility e Game. Essas palavras-chave foram aplicadas nos seguintes repositórios Google Acadêmico e o IEEE Xplore.

Como é mostrado no Quadro 1, as metodologias utilizadas no reconhecimento de comandos por voz nos trabalhos não provocam variações na precisão. Os jogos utilizados para a implementação da acessibilidade por comandos de voz foram dos mais variados tipos, entretanto, o que eles têm em comum é a simplicidade dos controles por voz. Por fim, pode-se comprovar que todos os trabalhos localizados foram para o idioma inglês.

Quadro 1 – Metodologias dos trabalhos relacionados.

Refer.	Jogo	Metodologia	Melhor precisão	Idioma
[12]	Xadrez	SVM + HMM	95%	Inglês
[13]	Pong	RNC	99%	Inglês
[14]	Tetris	API MS SAPI 5.1	90%	Inglês
[15]	Diversos	N/A	N/A	Inglês
Este trabalho	EcoVoz	Google Cloud Speech	89%	Português Brasileiro

Dentre os trabalhos analisados, o de Piotrowski et al. [13] também utiliza da engine utilizada por este projeto. Porém a relação com a metodologia adotada para reconhecimento de voz para execução de comandos é completamente diferente à desse trabalho, pois ao adotar uma implementação de Rede Neural Convolutiva para reconhecimento de voz, permitiu que a própria rede acessasse o objeto durante o jogo e executasse o comando, dispensando algumas validações e processamentos que acontecem nesse atual trabalho.

Além disso, foi possível comparar que entre os métodos de reconhecimento de voz, como mostra no Quadro 1, esse é o único trabalho que reconhece outra língua sem ser a inglesa. Entre os trabalhos analisados é possível observar, que aqueles que optaram por implementar o reconhecimento por meio do uso de APIs externas obtiveram uma margem de precisão inferior.

Como pode ser observado na Tabela 1, entre os trabalhos selecionados, apenas dois executaram testes com usuários, sendo que nenhuma amostra de testes corresponde a usuários com deficiência. Esses testes permitem uma ampla perspectiva de usuários e possíveis resultados, permitindo traçar uma métrica comparativa sobre os dados levantados.

Tabela 1 – Amostra de teste dos trabalhos relacionados.

Referência	Amostra de teste
[12]	N/A
[13]	N/A
[14]	7
[15]	8
Este trabalho	21

4 Desenvolvimento

Foram implementados todos os níveis do jogo utilizando o Engine Unity3D. Após a implementação dos níveis, foi o início da implementação da integração com a API Google Cloud Speech, para realizar o reconhecimento de voz por palavras isoladas com o propósito de interpretar comandos que o usuário invoque no jogo.

Para essa integração com a API, o projeto faz uso de uma biblioteca pública chamada UnityGoogleStreamingSpeechToText. É utilizado seu código como base para montar uma interação entre o script que envia as requisições pra API e o controle de acesso aos componentes de áudios presentes na Engine.

Em seguida, foram adaptados os comandos no jogo para aceitar os controles por voz e a partir disso foram testados os níveis usando inteiramente o reconhecimento de voz, para que seja possível analisar o Level Design a fim de garantir uma jogabilidade mais fluida para usuários que utilizem do controle por voz.

Conforme os níveis foram validados, foi passado por uma fase de finalização e incrementando as cut scenes e diálogos presentes na história, bem como melhorias mecânicas e estéticas.

A metodologia aplicada para transformar um jogo plataforma em um jogo que permita os usuários controlarem por meio da voz, foi permitir que o usuário não tenha necessidade de controlar a distância percorrida no ângulo horizontal, podendo interagir apenas com a direção em que o personagem se move, garantindo assim que o jogador tenha tempo para decidir o momento do salto ou ações de precisão.

4.1 Estrutura do Jogo

A engine Unity3D trabalha com a estruturação de projetos baseados em objetos que participam de cenas, que por sua vez, esses objetos recebem componentes, que permitem o controle de recursos como reprodução de sons, animações e implementação de comportamentos por meio de scripts na respectiva cena.

A linguagem de programação para controle desses objetos é C#, que é integrada ao objeto por meio de um componente, proporcionando assim uma facilidade na inspeção de parâmetros nos métodos e comportamento dos objetos.

Os componentes oferecem, em sua maioria, funções nativas da própria engine, que abstraem a complexidade e permitem ao desenvolvedor o controle de parâmetro, tais como: formas geométricas básicas que servem de controle para colisão, reprodução sonora ou comportamento de gravidade. Ao implementar um componente de script, é possível acessar diversos componentes e fazer a manipulação desses parâmetros dentro do próprio objeto instanciado.

Toda cena o personagem acessa o script de controle de variáveis para inicializar tanto o contador de pontos de vida quanto o de coletáveis e forma de controle. Quando o personagem colide em um objeto identificado como “inimigo” ou “armadilha”, ele inicia uma rotina de verificação dos pontos de vida e uma animação de recebimento de “dano”. É possível visualizar essa rotina de animação na Figura 1.



Figura 1: Animação de recebimento de “dano”

O Unity3D possui uma função chamada Update, que é executada repetidamente durante a cena e é chamada basead numa taxa de frames variáveis. No início da cena toda vez que o personagem chamar a função Update, é feita uma checagem para validar se algum comando foi executado pelo usuário, por exemplo, quando o personagem é controlado por teclado, é verificado se o jogador pressionou a tecla para executar o comando de pular e se o personagem se encontra sobre o chão, para adquirir impulso.

Entretanto, ao controlar o personagem utilizando a voz, esse dado de entrada de áudio é enviado para a API e seu retorno é salvo numa variável em um objeto. Então toda vez que a função Update é chamada, é verificado se o valor nessa variável corresponde à alguma ação esperada no dicionário de comandos presentes no jogo, para assim executar a rotina do comando.

O que diferencia os inimigos, além dos elementos visuais, são seus comportamentos específicos, como por exemplo, tem um tipo de inimigo que salta sobre cenário e existe um inimigo que não executa o comportamento de patrulhar a cena. Como é possível identificar na Figura 2 dois inimigos distintos.



Figura 2: Comportamento de diferentes inimigos

Dentre a gama de inimigo, um em especial, executa uma rotina de animação de uma habilidade que o permite desativar o comportamento de recebimento de dano durante um período curto de 1 segundo. Esse comportamento é chamado de forma aleatória, dentro de uma margem de 0,06%.

O usuário também tem a possibilidade de encontrar objetos que permitem ao personagem modificadores durante aquela cena, como por exemplo, o modificador “Raio” que modifica o valor do parâmetro do salto durante 10 segundos, permitindo alcançar áreas diferentes do cenário. Como é possível observar na Figura 3, outro modificador é a adição de um projétil que verifica a direção do personagem e dispara um objeto do tipo “Bala” na horizontal, esse objeto contém um comportamento que ao colidir em um objeto classificado como “Inimigo”, causa a sua destruição.



Figura 3: Execução do disparo de projétil horizontal

Durante todo o jogo, o usuário pode adquirir coletáveis que somam em um contador, que é apresentado no final do jogo, juntamente com o número de tentativas dele.

4.2 Integração com API

Durante as cenas em que o usuário tem o controle do personagem, é instanciado um objeto chamado API, composto apenas de um componente para entrada de áudio e um script que controla toda conexão com a API.

A API funciona por meio de requisições que são enviadas ao servidor da Google que possui mecanismos de inteligência artificial dedicada ao reconhecimento da voz, que retornam exclusivamente uma transcrição do que foi reconhecido. Isto é, a API detecta o microfone presente no computador e, ao iniciar uma cena, ela é chamada por meio de um script que envia o áudio por

meio da requisição baseada nas configurações presentes no cabeçalho, retornando a sua transcrição e, após isso, executando o comando no personagem.

Dentro do corpo da requisição estão dispostos objetos que recebem desde uma autenticação gerada manualmente dentro da plataforma de serviços Cloud da Google, como também configurações para o reconhecimento em streaming. O Quadro 2 representa o código-fonte que descreve a configuração utilizada no projeto.

Quadro 2 - Parâmetros do objeto de configuração da requisição da API Google Speech.

```
await _streamingCall.WriteAsync(new
StreamingRecognizeRequest()
{
    StreamingConfig = new
StreamingRecognitionConfig()
{
    Config = new RecognitionConfig()
    {
        Encoding =
RecognitionConfig.Types.AudioEncoding.Linear16,
        SampleRateHertz =
audioConfiguration.sampleRate,
        LanguageCode = "pt-BR",
        MaxAlternatives = 1
    },
    InterimResults = returnInterimResults,
}
});
```

Esse reconhecimento de streaming só é possível por meio da utilização do modelo de Chamada remota de procedimento (RPC, acrônimo de Remote Procedure Call), no qual tem acesso ao serviço disponibilizado no endereço speech.googleapis.com, em que faz uso do método `StreamingRecognize` que permite executar reconhecimento de voz por streaming bidirecional, recebendo resultados provisórios enquanto ainda está enviando os áudios.

Em um RPC de streaming bidirecional, o cliente inicia a chamada e o servidor recebe metadados do cliente, então o servidor tem a autonomia de enviar metadados de retorno ao cliente, ou esperar o cliente inicie o streaming de mensagens. Isso se deve ao fato de que o RPC de streaming bidirecional possui dois fluxos independentes, então o cliente e o servidor podem ler e gravar mensagens em qualquer ordem e momento, permitindo assim, que uma API possa retornar um dado provisório enquanto está processando a requisição em tempo real.

Tanto as configurações da requisição quanto os bytes de dados de áudio são codificados e encapsulados dentro de um objeto chamado `RecognitionConfig`. Este por sua vez, possui 4 parâmetros: `Encoding`, `SampleRateHertz`, `LanguageCode` e `MaxAlternatives`.

O campo do `Encoding` descreve o formato da codificação de dados de áudio enviados em todas as mensagens do método `RecognitionAudio`. De acordo com a documentação do Google Speech, para melhores resultados, a fonte de áudio deve ser

capturada e transmitida sem perdas. Para garantir isso, foi optado por usar o Linear16 que garante uma amostra não compactada de 16 bits.

O SampleRateHertz corresponde a taxa de amostragem em Hertz dos dados de áudio que são enviados em todas as mensagens do método RecognitionAudio. Os valores válidos dessa taxa variam entre 8000-48000 Hz. O recomendado pela documentação é deixar com 16000 Hz.

O LanguageCode serve como parâmetro para identificar o idioma que a requisição se trata, seguindo o padrão da IETF language tag, para o projeto, foi selecionado o idioma português do Brasil, que corresponde ao código “pt-BR”.

Por último, o parâmetro de MaxAlternatives descreve o número máximo de hipóteses de reconhecimento a serem retornadas. Especificamente, o número máximo de mensagens em cada SpeechRecognitionResult. Dentre os valores válidos que vão de 0-30, para cumprir o escopo do projeto de abordar comandos para executar no jogo, o valor selecionado foi 1. Após encaminhar a requisição para a API, o script aguarda o retorno.

Ao adquirir o retorno, é convertido para uma variável string o

valor encapsulado dentro do objeto transcript, que ao final da requisição é encaminhado para o objeto “personagem”. Esse retorno passa por um método de comparação, que caso contenha a mesma palavra que corresponde a determinado comando, o personagem inicia uma rotina de execução do comando.

5 Resultados

Essa seção apresenta e discute os resultados do projeto, permitindo analisar a sua contribuição. Na subseção 5.1 é apresentado uma comparação entre as palavras selecionadas. A subseção 5.2 apresenta os dados retirados dos formulários, bem como os parâmetros e métricas de comparação. Na subseção 5.3 é apresentado a discussão sobre os resultados obtidos.

5.1 Palavras Selecionadas

Para otimizar o tempo de resposta, foram analisadas quais são as palavras, utilizadas nos comandos por voz, que se encaixam

melhor no escopo do jogo, e que não ocasionassem em um maior tempo de resposta, afetando negativamente a sua jogabilidade.

Dito isto, foram realizados ao todo 5 testes para cada palavra

que possivelmente poderiam ser utilizadas no jogo, então foram registradas seu tempo de execução em segundos. Por fim, foram analisadas e comparadas as médias de tempo em segundos de cada palavra que poderiam designar o mesmo comando. Essa metodologia permitiu então filtrar qual palavra desempenha um tempo de resposta mais favorável para o projeto.

Ao comparar todas as palavras que poderiam compreender o comando “Pular”, como pode ser visto na Tabela 2, é possível notar um desempenho superior da palavra “Top” comparando com as demais possibilidades. Porém, para garantir um entendimento melhor do usuário e transformar em um controle mais receptivo, foi optado por utilizar a palavra “Pula”, que foi a segunda palavra com o melhor tempo de resposta.

Tabela 2 - Desempenho para o comando “Pular”.

Palavra	Teste 1 (seg.)	Teste 2 (seg.)	Teste 3 (seg.)	Teste 4 (seg.)	Teste 5 (seg.)	Média
Top	1,3	1,49	1,46	1,56	1,28	1,3
Cima	1,83	1,69	1,83	1,73	1,85	1,83
Pula	1,46	1,86	1,65	1,69	1,56	1,46
Alto	1,63	1,81	1,83	1,9	1,88	1,63
X	1,63	1,88	1,73	1,72	1,53	1,63

Fazendo uma comparação de todas as possíveis palavras para designar o comando de movimentação horizontal para a direção esquerda, foram comparadas quatro palavras e a escolhida foi a palavra “Esquerda”, como descreve a Tabela 3, dentre as palavras com o tempo de resposta médio menores, a palavra “Esquerda” se mostrou aquela que se encaixa melhor no escopo de “indicar uma direção ao usuário”.

Tabela 3 - Desempenho para o comando “esquerda”.

Palavra	Teste 1 (seg.)	Teste 2 (seg.)	Teste 3 (seg.)	Teste 4 (seg.)	Teste 5 (seg.)	Média
Esquerda	1,46	1,45	1,39	1,34	1,46	1,42
Atrás	1,4	1,4	1,38	1,45	1,6	1,446
Volta	1,35	1,43	1,32	1,43	1,36	1,378
Anti-horário	1,89	1,4	1,65	1,61	1,66	1,642

A Tabela 4 representa os valores dos testes comparando o desempenho em segundos do reconhecimento do comando de movimentação para a direção direita utilizando as respectivas palavras listadas. É possível notar um desempenho superior ao utilizar a palavra “Direita” para indicar esse comando, tendo em sua média, 7 milissegundos de vantagem sobre a segunda melhor palavra.

Tabela 4 - Desempenho para o comando “direita”.

Palavra	Teste 1 (seg.)	Teste 2 (seg.)	Teste 3 (seg.)	Teste 4 (seg.)	Teste 5 (seg.)	Média
Direita	1,33	1,31	1,35	1,28	1,43	1,34
Frente	1,36	1,5	1,41	1,4	1,46	1,426
Horário	1,55	1,38	1,55	1,55	1,38	1,482
Vai	1,4	1,29	1,25	1,35	1,29	1,4

Por fim, para descobrir a melhor palavra para referenciar o comando de executar uma habilidade, foram comparadas as palavras citadas na Tabela 5, no qual apesar das palavras “Habilidade” e “Magia” terem conseguido um tempo de resposta médio superior, preferimos por designar esse comando com a palavra “Poder”, a fim de facilitar a pronúncia e o entendimento do usuário.

Tabela 5 - Desempenho para comando “poder”.

Palavra	Teste 1 (seg.)	Teste 2 (seg.)	Teste 3 (seg.)	Teste 4 (seg.)	Teste 5 (seg.)	Média
Poder	1,58	1,52	1,42	1,61	1,56	1,538
Habilidade	1,4	1,43	1,6	1,41	1,58	1,484
Atirar	1,62	1,63	1,48	1,66	1,58	1,594
Magia	1,55	1,52	1,33	1,33	1,39	1,424

5.2 Análise dos Testes Aplicados

Para realização dos testes, os usuários tiveram acesso a: (i) um link para download do jogo disponibilizado pelo Google Drive do autor¹; e (ii) o link do formulário de pessoas sem deficiência². Estes formulários estavam disponíveis para preenchimento on-line por meio da plataforma gratuita de formulários Google Forms. Também é possível acessar o link do formulário de pessoas sem deficiência por meio de um arquivo de texto presente no arquivo de download do jogo.

Os testes tiveram como objetivo a execução do jogo com ambos os controles, voz e padrão (mouse e teclado), oferecidos no projeto para apenas usuários sem deficiência. Ao todo, foram coletadas 21 amostras, sendo possível extrair informações sobre o desempenho dos usuários com todos os controles dispostos no jogo.

Inicialmente é possível analisar o perfil dos usuários da amostra por meio da classificação de 0 a 5, sendo 0 correspondente a primeiro contato com jogos digitais e 5 representando contato diário com jogos digitais. A partir disso pode-se traçar um comparativo entre as experiências prévias com jogos digitais com o teste do jogo do projeto. A Figura 4 demonstra que 57,14% dos usuários se consideram usuários diários de jogos digitais.



Figura 4: Frequência dos usuários com jogos digitais.

Com essa informação é possível então avaliar o comparativo dos usuários que conseguiram finalizar o jogo em ambos os controles, como demonstrado na Figura 5 em comparação com a Figura 6. No controle padrão todos os usuários finalizaram, em contraponto, no controle por voz 33% não conseguiram finalizar. Ao filtrar por esses jogadores que não conseguiram finalizar, é possível notar que aproximadamente 70% desses usuários

¹ Disponível em: <https://bitly.com/dVTTH>

² Disponível em: <https://forms.gle/s4CjBwm2VTbvocCr6>

classificaram seu contato com jogos digitais entre 2 ou inferior na questão da Figura 5.

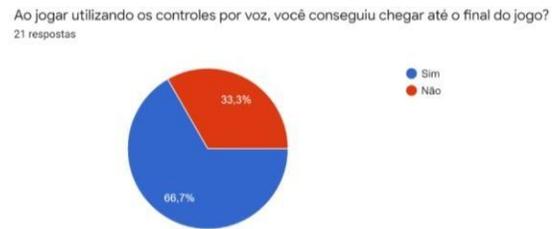


Figura 5: Finalização do jogo com Controles por Voz.

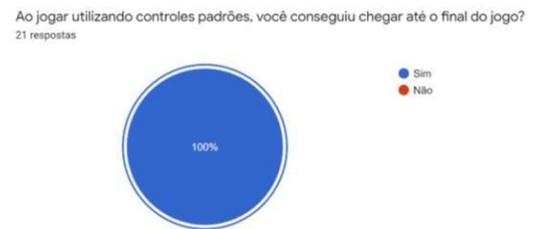


Figura 6: Finalização do jogo com Controles Padrões.

Para ter uma real noção de quantas tentativas os usuários tiveram para finalizar o jogo, foi disposta no formulário um campo pedindo por esse número ao término do jogo. Como é demonstrado na Figura 7, é possível observar que a média de tentativas ao jogar com controles por voz alcançou a marca de 12 tentativas por usuário. Enquanto na Figura 8, aproximadamente 47% dos jogadores finalizaram em apenas uma tentativa ao jogar com controles padrões, deixando uma média de tentativas em 1,5 tentativas.

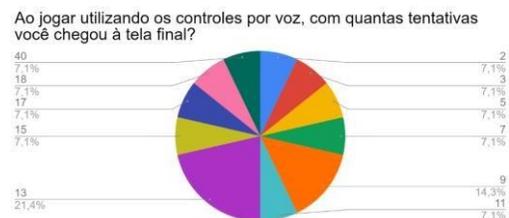


Figura 7: Número de tentativas até o final com Controles por Voz.

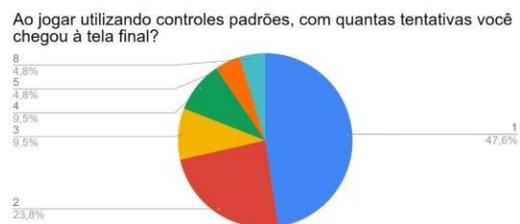


Figura 8: Número de tentativas até o final com Controles Padrões.

Ao final, o usuário pôde classificar alguns elementos presentes no jogo de 1 a 5, sendo 1 representado como pouco satisfatório e 5 como muito satisfatório. Como mostra a Figura 9 aproximadamente 52% dos jogadores classificaram o level design, que aborda assuntos como balanceamento, mecânicas e construção dos níveis, como muito satisfatório. Enquanto isso aproximadamente 38% dos usuários consideram a dificuldade e a portabilidade do jogo para a plataforma PC, muito satisfatório. Aproximadamente 47% dos jogadores consideram a jogabilidade como 3, o meio termo entre os dois extremos.

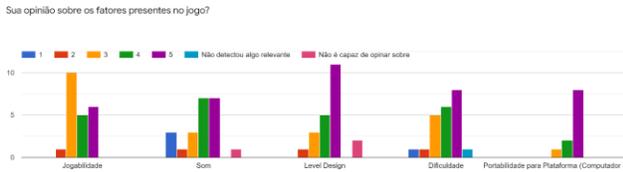


Figura 9: Gráfico comparativo entre elementos do jogo.

Por fim, foi questionado qual a preferência dos usuários em relação aos controles, e como pode ser visto na Figura 10, 40,9% preferiram utilizar os controles por voz e dentro dessa amostra, aproximadamente 66% obtiveram o número de tentativas igual ou superior a 13.

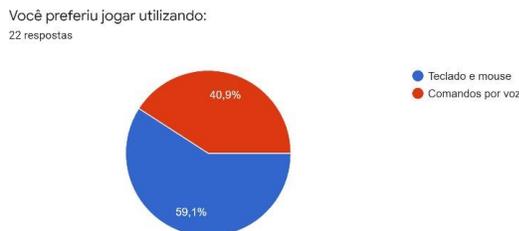


Figura 10: Preferência de controle.

Para a análise da precisão do reconhecimento de voz, foram utilizados os serviços de visualização e medição de dados por meio da plataforma Google Cloud Platform (da API Google Speech). Nesse painel de controle, é possível filtrar os dados por serviços e por métodos consumidos. Para a análise de dados, foi filtrado a execução da API durante o período de testes com usuários. Como pode ser observado na Figura 11, o gráfico detalha a porcentagem da margem de erro das requisições executadas durante o período dos testes. Nesse gráfico é possível observar que a margem de erro é diretamente proporcional ao número de acessos ao serviço, no qual alcançou uma média geral de 14,72%.



Figura 11: Proporção da margem de erro das requisições da API durante os testes.

5.3 Discussão

Nessa seção foram apresentados os resultados das análises de das palavras reconhecidas da API Google Speech e os dados coletados por meio dos formulários de pesquisa de opinião dos usuários. Essas análises permitiram um direcionamento nas configurações de aplicações do reconhecimento de voz integrado no jogo, pois foram selecionadas as configurações que obtiveram o melhor desempenho, levando em consideração as métricas supracitadas.

Dentre os trabalhos similares, foi possível notar também que os trabalhos com a menor margem de precisão são aqueles que utilizam de APIs externas para executar o reconhecimento de voz. Como é possível observar por este trabalho que utilizou a API Google Speech-To-Text para a língua português Brasil, e no melhor dos casos obteve uma taxa de precisão de 89%. Já o trabalho de Sporka et al. [14] obteve a melhor taxa de precisão com 90%, utilizando a API MS SAPI 5.1 para a língua inglesa.

Além disso, vale destacar que este trabalho coletou dados de testes executados com 21 usuários, algo que também é presente nos trabalhos de Harada et al. [15] e Sporka et al. [14], em que, respectivamente, testaram com 8 e 7 usuários. Por meio dos dados coletados, foi possível notar um interesse nítido em tecnologias de reconhecimento de voz aplicadas no meio dos jogos digitais, principalmente entre usuários que não se consideram jogadores assíduos.

Os testes com usuários sem deficiência permitiram traçar um comparativo entre o controle por voz e o controle padrão, por meio de mouse e teclado. Em todos os testes, os usuários tiveram um desempenho superior, utilizando como base a métrica do número de coletáveis comparada com o número de tentativas, ao jogar usando a controle padrão. Quando questionados sobre justificativa da diferença de desempenho entre os controles, a maioria dos usuários pontuaram que a lentidão no tempo de resposta dos controles era um fator decisivo que afetava o desempenho do usuário.

Ao final, foi possível identificar que usuários que não se consideram jogadores assíduos tiveram um desempenho inferior comparado aos usuários que se consideram experientes em jogos, porém esses jogadores não assíduos classificaram o jogo de maneira positiva e não se sentiram punidos mesmo ao falhar repetidas vezes.

6 Conclusões e Trabalhos Futuros

Esse trabalho teve como objetivo principal o desenvolvimento de um jogo que permita acessibilidade para pessoas com deficiência motora por meio do uso de inteligência artificial aplicada no reconhecimento de voz.

Por meio de uma pesquisa bibliográfica, foram levantadas as principais técnicas de reconhecimento de voz e tecnologias de interfaces de programação de aplicações que poderiam ser consultadas para a execução do reconhecimento de comandos por voz e seus requisitos. Além disso, por meio da leitura dos artigos também foi possível entender o funcionamento e direcionamento do desenvolvimento dos jogos, e quais ferramentas se encaixavam melhor no escopo do projeto.

No intuito de consumir a API de reconhecimento de voz selecionada, no caso o Google Cloud Speech-To-Text, foram buscadas as melhores configurações que se encaixassem no escopo do projeto, para isso a API foi configurada para executar o reconhecimento por meio do streaming de dados de áudio na requisição, garantindo um retorno instantâneo da API e obtendo em média 14% de margem de erro.

Por fim, 21 pessoas sem deficiência testaram o jogo e preencheram um formulário descrevendo seu desempenho e impressões. Todos os usuários tiveram que testar as duas formas de controle do jogo, por voz e padrão (teclado e mouse). Ao traçar uma comparação do número de tentativas adquiridos por usuários utilizando as duas formas de controle, é possível notar um desempenho superior ao fazer uso do controle padrão. A média de número de tentativas com controle padrão foi inferior que a média de controle por voz, mostrando-se que o controle por voz pode não ser eficiente ao primeiro contato com usuários sem deficiência.

Apesar dos resultados mostrarem uma perda de eficiência ao utilizar o reconhecimento da voz como forma de controle, também é possível observar um interesse nesse recurso em jogos digitais.

A principal contribuição dessa pesquisa foi na área de Interação Humano-Computador (IHC), por meio do desenvolvimento de um jogo digital que utiliza uma API para o reconhecimento de voz para o português brasileiro, promovendo recursos de acessibilidade em um jogo digital.

Esse jogo que faz uso do reconhecimento de voz, para identificar comandos, poderá ser utilizado para: (i) servir de base para um jogo que deseja implementar acessibilidade para pessoas com deficiência; (ii) o demonstrativo das aptidões da utilização de bibliotecas externas em jogos; e (iii) uma demonstração das capacidades do reconhecimento de voz e vantagens de sua aplicação para a área do IHC.

Por fim, ao longo do desenvolvimento deste trabalho, puderam ser identificadas algumas possibilidades de melhoria e de continuação a partir de futuras pesquisas, as quais incluem:

1. Analisar o desempenho de outra metodologia de reconhecimento de voz, inicialmente, utilizando rede neural convolucional;

2. Aplicar testes em usuários PcDs a fim de conseguir coletar dados de suas impressões e analisar o uso do jogo como recurso de acessibilidade na prática;
3. Adaptar o jogo para plataformas WEB e dispositivos móveis, permitindo facilidade de acesso do usuário, e diversidade de aplicações e testes, escalonando o escopo do projeto;
4. Implementação de comandos em outras línguas, garantindo que possa haver uma comparação entre a velocidade de reconhecimento multilíngüístico e transformar o jogo mais acessível para mais usuários;
5. Implementação de customização de comandos de voz, permitindo ao usuário flexibilidade e conforto para jogar utilizando as palavras das quais ele melhor consiga pronunciar;
6. Extrair os dados referentes a performance do usuário ao testar o jogo no formato de log, permitindo maior fidelidade na captação das informações e diminuindo a margem de dados imprecisos.

REFERÊNCIAS

- [1] Presidência da República. 2015. *Lei nº 13.146, de 6 de julho de 2015. Lei Brasileira de Inclusão da Pessoa com Deficiência*. Brasília: Secretária-geral. Link: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/13146.htm.
- [2] B. S. Silva, S. D. J. Barbosa. 2010. *Interação humano-computador*. Rio de Janeiro: Elsevier Brasil. 408 p.
- [3] J. Huizinga. 2008. *Homo ludens: proeve eener bepalng van het spel-element der cultuur*. Amsterdam University Press.
- [4] S. Banks, C. Craven. 2018. *Can I Play That?* Link: <https://caniplaythat.com/>.
- [5] Gonzalo Frasca et al, (ed.). 2003. *Simulation versus Narrative: Introduction to Ludology*. Link: https://ludology.typepad.com/weblog/articles/VGT_final.pdf.
- [6] P. M. Greenfield, P. Dewinstanley, H. Kilpatrick, D. Kaye. 1994. *Action video games and informal education: effects on strategies for dividing visual attention*. *Journal of Applied Developmental Psychology*, 15:105-123.
- [7] Rodrigo Lyra. 2013. *Jogo Musical para auxiliar o exercício de execução rítmica*. Trabalho Técnico-científico de Conclusão de Curso (Graduação em Ciência da Computação) – Escola do Mar, Ciência e Tecnologia, Universidade do Vale do Itajaí, Itajaí, 2013.
- [8] Emanuel Reyno Montero et al. 2008. *Model Driven Game Development: 2D Platform Game Prototyping*. Link: https://www.researchgate.net/publication/221024315_Model_Driven_Game_Development_2D_Platform_Game_Prototyping.
- [9] Michael Lewis et al. 2002. *Game engines in scientific reserach*. Link: <https://www.cse.unr.edu/~sushil/class/gas/papers/GameA1p27-lewis.pdf>.
- [10] Josemar Barone Marangoni et al. 2006. *Reconhecimento e Sintetização de Voz usando Java Speech*. Link: http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/bjMnA2Zwc9685z8_2013-5-27-15-40-25.pdf.
- [11] Lucas Debatin, Aluizio Haendchen Filho, Rudimar Luís Scaranto Dazzi, Hércules Antonio do Prado, Edilson Ferneda. 2021. *Offline continuous speech recognition for mobile devices*. *International Journal of Development Research (IJDR)*, vol. 11, issue, 11, pp. 52248-52256, November, 2021.
- [12] Bharathi, B. et al. 2017. *Speech recognition based chess system for visually challenged*. *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 1797-1801.
- [13] Piotrowski, J., Yfantis, E., Campagna, A., Cornu, Q., Gallitano, G. 2019. *Voice Interactive Games*. 0780-0784. 10.1109/UEMCON47517.2019.8993015.
- [14] Sporka, A.J., Kurniawan, S.H., Mahmud, M., Slavík, P. 2006. *Non-speech input and speech recognition for real-time control of computer games*. *Assets '06*.
- [15] Harada, Susumu, Wobbrock, Jacob, Landay, James. 2011. *Voice Games: Investigation Into the Use of Non-speech Voice Input for Making Computer Games More Accessible*. 11-29. 10.1007/978-3-642-23774-4_4.