



Sistemas de Interação por Gestos e Voz aplicados a Jogos

PAULO FILIPE MELO RODRIGUES

Setembro de 2025

Sistemas de Interação por Gestos e Voz aplicados a Jogos

Paulo Filipe Melo Rodrigues

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Jogos, Sistemas Gráficos e Interativos**

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim. As exceções estão explicitamente reconhecidas na secção “Considerações éticas” do primeiro capítulo. Esta secção também declara como as ferramentas de IA foram utilizadas e para que finalidade.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 28 de setembro de 2025

Resumo

Este trabalho explora a utilização de sistemas de rastreamento de mãos e comando de voz aplicados em jogos com ênfase na acessibilidade e no estímulo de funções cognitivas (como memória e raciocínio) e terapêutica. Foi efetuada uma revisão do estado da arte e desenvolvidos dois protótipos funcionais: um baseado em rastreamento de mãos e outro em reconhecimento de voz, integrados em minijogos.

Realizaram-se dois ciclos de testes com utilizadores, que combinaram medições de desempenho dos protótipos, observação e inquéritos para recolher perceções subjetivas sobre usabilidade, intuitividade e potencial de aplicação. Os resultados indicam viabilidade técnica e interesse dos participantes, apontando o comando de voz como mais acessível em hardware comum e o rastreamento de mãos como promissor, embora sensível a condições de iluminação e qualidade de câmara. São discutidas limitações (amostra dos teste e necessidade de otimização de performance) e propostas futuras, como a junção dos dois sistemas, modos multijogador e aplicações dirigidas a públicos com necessidades especiais.

São discutidas limitações (amostra dos teste e necessidade de otimização de performance) e proposta futuras, como a junção dos dois sistemas, modos multijogador e aplicações dirigidas a públicos com necessidades especiais.

Palavras-chave: Rastreamento da Mão, Comando de voz, Acessibilidade, Jogos

Abstract

This dissertation investigates the potential of hand-tracking and voice-based interaction methods for games, focusing on accessibility, cognitive stimulation (memory and cognitive training), and therapeutics. After a structured literature review, two functional prototypes were implemented in Unity: (1) a hand-tracking variant that maps camera landmarks to in-game interactions (selection, dragging, and gesture confirmation) and (2) a voice-driven variant, using speech-to-text pipelines to trigger game actions and menu navigation. Both prototypes include a set of minigames (quizzes, color identification, reaction tasks, and simple 3D interactions) designed to test different interaction paradigms and cognitive demands.

Evaluation consisted of two user test rounds combining objective metrics (task completion, input latency observations) and subjective instruments (questionnaires and open feedback). Participants reported a clear interest in both modalities: voice commands proved generally more accessible on commodity hardware (ubiquitous microphones), while hand tracking offered higher immersion but required better camera quality and controlled lighting. Identified challenges included input latency/CPU load when running both systems concurrently, lack of robustness in noisy or low-light environments, and the need to refine selection affordances to reduce fatigue and misselections.

Contributions of this work are (a) an empirical assessment of gesture and voice-based controls in small game scenarios, (b) design recommendations for accessibility-oriented mechanics, and (c) two prototype references demonstrating practical integration strategies and trade-offs. Limitations include the sample size of tests (preventing strong statistical claims) and hardware variability across participants. Future work proposes hybrid modes (combined voice + gesture), multiplayer voice designs, extended clinical or longitudinal studies for cognitive rehabilitation, and exploration of alternative STT/TTS services to improve latency and accuracy.

Keywords: Hand-tracking, Voice command, Accessibility, Games

Agradecimentos

Em primeiro lugar, quero agradecer ao meu orientador, professor António Vieira de Castro, pelo convite para pertencer ao SIIS (Sistemas de Interação e Inovação Social), pela proposta de tese e pelo desafio, mas também pelo apoio, dedicação e disponibilidade ao longo de todo o projeto. Ao ISEP (Instituto Superior de Engenharia do Porto) e a todos os docentes pelos bons anos de percurso académico, que passaram a voar.

Ao SIIS (Sistemas de Interação e Inovação Social) pela disponibilização dos equipamentos e do espaço que permitiram que este projeto se concretizasse.

Ao meu colega João Lima por ter me ajudado a testar o protótipo a procura de possíveis problemas que pudessem prejudicar a experiência dos utilizadores durante os testes.

A todos os utilizadores que aceitaram testar os protótipos.

Por fim, mas não menos importante, a todos os colegas, amigos e familiares que de alguma forma contribuíram para o sucesso deste projeto.

Índice

1	Introdução	1
1.1	Enquadramento do tema	1
1.2	Motivação	2
1.3	Stakeholders	3
1.4	Planeamento	5
1.4.1	Metodologia	5
1.4.2	Cronograma do Projeto	7
1.4.3	Custos	9
1.5	Análise de Riscos	9
1.6	Considerações Éticas	10
1.7	Metodologia de investigação	11
1.7.1	Métodos de Revisão	11
1.7.2	Bibliotecas Digitais	14
1.7.3	Estratégia de Pesquisa	15
1.8	Estrutura do Documento	15
2	Estado da Arte	17
2.1	Acessibilidade e Usabilidade	17
2.1.1	Acessibilidade na vida	18
2.1.2	Acessibilidade digital e interativa	19
2.1.3	Acessibilidade em jogos	20
2.2	Rastreamento de gestos	21
2.2.1	Tecnologias e algoritmos de rastreamento de gestos	22
2.2.2	Rastreamento de gestos em jogos	23
2.2.3	Vantagens e desafios	25
2.3	Comando de Voz	26
2.3.1	Tecnologias e frameworks de reconhecimento de voz	26
2.3.2	Tecnologias e Frameworks de Text-to-Speech	28
2.3.3	Aplicações que usam Comando de Voz	28
2.3.4	Comando de voz em jogos	30
2.3.5	Vantagens e desafios	31
3	Tecnologias utilizadas	35

3.1	O Python.....	35
3.2	O Unity	36
3.3	O Microsoft Visual Studio Code.....	37
3.4	O Git e GitHub.....	38
3.5	O GitHub Desktop	38
3.6	O Figma.....	39
3.7	O Google Forms	40
3.8	O Chat GPT.....	40
3.9	A Adobe Illustrator e Photoshop	41
3.10	O Adobe Color	41
3.11	O Microsoft Clipchamp	42
4	Design.....	43
4.1	Wireframe.....	43
4.2	Paleta de cores.....	44
4.3	MockUp	45
4.3.1	O Rastreamento de Mãos.....	45
4.3.2	O Comando de voz	49
5	Desenvolvimento do protótipo.....	53
5.1	Configurações.....	53
5.1.1	Python	53
5.1.2	Unity	60
5.2	Desenvolvimento do Rastreamento de Mãos	67
5.3	Desenvolvimento do comando de voz	71
5.4	Desafios no desenvolvimento.....	75
6	Testes com utilizadores	77
6.1	1.º Teste com Utilizadores (Teste de Usabilidade).....	77
6.1.1	Perfil dos Participantes	78
6.1.2	Resultado e Análise	79
6.1.3	Conclusão do 1.º Teste de Utilizadores	86

6.2	2.º Teste de Utilizadores.....	87
6.2.1	Perfil dos Participantes.....	88
6.2.2	Resultados e Análises	89
6.2.3	Conclusão do 2.º Teste de Utilizadores	103
7	Conclusão e Trabalho Futuro	105
7.1	Planeamento final.....	105
7.2	Trabalhos Futuros	106
7.2.1	Junção dos dois sistemas.....	106
7.2.2	Multijogador local com comando de voz.....	107
7.2.3	Multijogador online	107
7.2.4	Projeto dirigido a público desfavorecidos (acessibilidade).....	107
7.2.5	Integração com assistentes virtuais	108
7.2.6	Exploração de STT e TTS alternativos.....	108
7.2.7	Escrita no ar.....	109
7.3	Escrita científica e evolução nos estudos	109
7.4	Conclusão final	110
	Referências	113
	Anexo A - WBS.....	121
	Anexo B - Cronograma Inicial.....	122
	Anexo C - Análise de Risco	124
	Anexo D - 1º Formulário.....	125
	Anexo E - 2º Formulário	129
	Anexo F - Instruções do 2º Teste	136
	Anexo G - Cronograma Final	139

Lista de Figuras

Figura 1 - Matriz de stakeholders	5
Figura 2 - WBS	7
Figura 3 - Cronograma inicial do projeto	8
Figura 4 - Cronograma inicial do projeto (Milestone).....	8
Figura 5 - Exemplo de semitransparência da mão em relação a visualização de objetos	23
Figura 6 - Exemplo do jogo "Cubism"	24
Figura 7 - Jogo "Suck Up"	31
Figura 8 - Wireframe do Rastreamento de Gestos	43
Figura 9 - Wireframe do comando de voz.....	43
Figura 10 - Paleta de cores principal	44
Figura 11 - Paleta de cores secundaria	45
Figura 12 - Mockup de Rastreamento de Mãos: Menu	46
Figura 13 - Mockup de Rastreamento de Mãos: Jogo das Cores	46
Figura 14 - Mockup de Rastreamento de Mãos: Quis.....	47
Figura 15 - Mockup de Rastreamento de Mãos: Pong Solo.....	47
Figura 16 - Mockup de Rastreamento de Mãos: Pong Classico	48
Figura 17 - Mockup de Rastreamento de Mãos: Jogo de Reação	48
Figura 18 - Mockup de Rastreamento de Mãos: Jogo da Matemática	49
Figura 19 - Mockup de Rastreamento de Mãos: Paint	49
Figura 20 - Mockup de comando de voz: Menu	50
Figura 21 - Mockup de comando de voz: Quis.....	50
Figura 22 - Mockup de comando de voz: Pong Solo	51
Figura 23 - Mockup de comando de voz: Pong Classico	51
Figura 24 - Mockup de comando de voz: Jogo da Memória	52
Figura 25 - Landmarks do mediapipe.....	54
Figura 26 - Comparação entre a detecção da mão utilizando cvzone e não utilizando, respectivamente	57
Figura 27 - Diagrama de comunicação entre Python e Unity	62
Figura 28 - Ilustração representativa da diferença entre os eixos de referência do OpenCV (origem no canto superior-esquerdo) e do Unity (sistema de coordenadas do motor)	64
Figura 29 - Esqueletos das mãos dentro do jogo	66

Figura 30 - Comparação entre os sistemas de seleção de objetos em rastreamento de mãos.	68
Figura 31 - Demonstração do jogo de Matemática.....	69
Figura 32 - Exemplo das mecânicas utilizadas no minijogo de Pintura.....	69
Figura 33 - Posicionamento da mão num ambiente 3D.....	70
Figura 34 - Comparação do tamanho do esqueleto da mão, antes e depois de fixar o tamanho	70
Figura 35 - Gráfico de resposta: "Como foi a sua experiência ao jogar os minigames?"	79
Figura 36 - Gráfico de resposta: "O tempo de resposta do jogo foi adequado?"	80
Figura 37 - Gráfico de respostas: "O controle por gestos pareceu intuitivo?"	81
Figura 38 - Gráfico de respostas: "Nas instruções, achas que devia ficar só texto a explicar a funcionamento do jogo, ou devia ter um GIF/vídeo explicativo"	82
Figura 39 - Gráfico de respostas: "Você vê este tipo de tecnologia sendo aplicada em quais áreas?"	83
Figura 40 - Gráfico de respostas: "Você teria interesse em testar versões futuras com novas mecânicas, como por exemplo, com comando de voz ou outros minijogos?"	85
Figura 41 - Gráfico de respostas: "Em termos das tecnologias implementadas considera terem potencial para uso em jogos?"	85
Figura 42 - Gráfico de respostas: "As regras e instruções foram claras?"	86
Figura 43 - Minijogos testados durante os testes	89
Figura 44 - Dificuldade sentida durante os testes.....	90
Figura 45 - Gráfico de respostas: "O tempo de resposta do jogo foi adequado?"	90
Figura 46 - Gráfico de respostas: "Sentiu-se confiante a usar os sistemas?"	91
Figura 47 - Gráfico de respostas: "Depois de jogar, sentiste desconforto físico?"	92
Figura 48 - Gráfico de respostas: "O controle por gestos/voz pareceu intuitivo?"	93
Figura 49 - Gráfico de respostas: "Nas instruções dos minijogos do rastreamento de mãos: achou útil ter um GIF explicativo a mostrar como se interage com o jogo?"	94
Figura 50 - Gráfico de respostas: "Em termos das tecnologias implementadas considera terem potencial para uso em jogos?"	100
Figura 51 - Gráfico de respostas: "Você vê este tipo de tecnologia sendo aplicada em quais áreas?"	100
Figura 52 - Gráfico de respostas: "De 0 a 10, quanto daria a sua experiência a utilizar o Rastreamento de Mãos/Comando de Voz para jogos?"	102
Figura 53 - Gráfico de respostas: "Comparação geral: Qual controlo preferes?"	102

Figura 54 - Cronograma Final	106
Figura 55 - Exemplo de integração da OCR para escrita no ar.....	109

Lista de Tabelas

Tabela 1 - Tabela dos <i>stakeholders</i> , níveis de poder/interesse e estratégias de envolvimento.	4
Tabela 2 - PICOCS Rastreamento de mão em Jogos	12
Tabela 3 - PICOCS comando de voz em Jogos.....	13
Tabela 4 - Ações possíveis ao receber informação de VoiceCommand.....	67
Tabela 5 - Tabela de resumo das mecânicas e finalidades de cada minijogo feito (rastreamento de mãos).....	71
Tabela 6 - Tabela de resumo das mecânicas e finalidades de cada minijogo feito (comando de voz).....	72
Tabela 7 - Dados sobre os participantes (primeiro teste).....	78
Tabela 8 - Tabela de respostas: "Você sentiu algum atraso ou imprecisão ao usar gestos para jogar?"	80
Tabela 9 - Tabela de respostas: "Que desafios técnicos você identificou na usabilidade do jogo?"	82
Tabela 10 - Tabela de respostas: "O que mais lhe chamou a atenção no protótipo?"	83
Tabela 11 - Tabela de respostas: "Quais melhorias técnicas você sugeriria para otimizar a experiência?"	84
Tabela 12 - Tabela de respostas: "Sugestões de melhoria"	86
Tabela 13 - Dados sobre os participantes e ambiente testado (segundo teste)	88
Tabela 14 - Tabela de respostas: "Você sentiu algum atraso ou imprecisão para jogar em algum dos protótipos? Se sim, pode descrever?"	92
Tabela 15 - Tabela de respostas: "Que desafios técnicos você identificou na usabilidade do jogo?"	95
Tabela 16 - Tabela de respostas: "O que achou sobre criar um jogo 3D utilizando rastreamento de mãos e uma câmara de computador normal?"	96
Tabela 17 - Tabela de respostas: "O que achou sobre criar jogos que precisa de ações rápidas utilizando comando de voz?"	97
Tabela 18 - Tabela de respostas: "Achas que este os jogos utilizarem comando de voz seriam ótimos para as pessoas com problemas visuais?"	98
Tabela 19 - Tabela de respostas: "O que mais lhe chamou a atenção no protótipo?"	99
Tabela 20 - Tabela de respostas: "Quais melhorias técnicas você sugeriria para otimizar a experiência?" e "Sugestões de melhoria"	101

Lista de Excertos de Código

Código 1 - Exemplo de configuração do <i>socket</i> pelo Python	54
Código 2 - Configuração do CVZone (em Python)	55
Código 3 - Configurações pelo MediaPipe (em Python)	57
Código 4 - Exemplo de configuração para STT utilizando API do Google (em python)	59
Código 5 - Exemplo de código de produção de conversão de texto em áudio utilizando gTTS e pygame (em python)	60
Código 6 - Script UDPReceive.cs (em C#)	62
Código 7 - Conversão de JSON para objeto C# no rastreamento de mãos (em C#)	63
Código 8 - Normalização das coordenadas (divisão por factor de escala) antes de aplicar aos objetos do Unity (em C#)	64
Código 9 - Aplicação das coordenadas aos objetos representativos dos <i>landmarks</i> (em c#) ...	65
Código 10 - Configuração das ligações do <i>landmarks</i> (em c#)	65
Código 11 - Conversão de uma String JSON em um objeto C# utilizando JsonUtility (em c#) ..	66
Código 12 - Código para definir a posição da mão no eixo dos Z	70
Código 13 - Código de configuração da verificação por similaridade	74

Acrónimos e Símbolos

Lista de Acrónimos

API	Aplication Programming Interface
ASL	Língua de Sinais Americana
ASR	Reconhecimento Automático de Voz
AWS	Amazon Web Services
BSL	Língua de Sinais Britânica
CNN	Redes Neurais Convulsional
CV	Computer Vision
EUA	Estados Unidos da América
GTTS	Google Text-to-Speech
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IoT	Internet das Coisas
ISEP	Instituto Superior de Engenharia do Porto
JS	JavaScript
NLU	Compreensão de Linguagem Natural
NPC	Non-Player Character
NSPE	National Society of Professional Engineers
OCR	Reconhecimento Óptico de Caracteres
PLN	Processamento de Linguagem Natural
PREPD	Preparação para Dissertação
PS	PlayStation
RA	Realidade Aumentada
RV	Realidade Virtual
SIIS	Sistemas de Investigação e Inovação Social
STT	Speech-to-Text
TTS	Text-to-Speech
UC	Unidade Curricular
UDP	User Datagram Protocol
UI	User Interface

UX	User Experience
VS Code	Microsoft Visual Studio Code
WBS	Work Breakdown Structure

1 Introdução

Este relatório apresenta o desenvolvimento de um projeto que explora a utilização de sistemas de rastreamento de mãos e comando de voz em jogos digitais, com foco na acessibilidade, estimulação cognitiva e inclusão social. Após um enquadramento teórico e a análise do estado da arte, foram criados dois protótipos funcionais e realizados testes com utilizadores para avaliar usabilidade, desempenho e potencial de aplicação. O documento encontra-se organizado em sete capítulos: introdução, revisão do estado da arte, descrição das tecnologias utilizadas, design, desenvolvimento do protótipo, testes com utilizadores e, por fim, conclusões e trabalhos futuros.

No presente capítulo, é feito o enquadramento do tema, justificando a relevância dos videojogos como meio de entretenimento e intervenção cognitiva, enquanto se reconhece a necessidade de interfaces mais inclusivas. São definidos os objetivos, *stakeholders*, planeamento e cronograma do projeto, assim como a análise de riscos e considerações éticas. Conclui-se com a metodologia de pesquisa adotada, destacando-se os métodos de revisão bibliográfico e as estratégias de investigação aplicadas.

1.1 Enquadramento do tema

Os videojogos são, atualmente, uma das formas de entretenimento e de aprendizagem mais difundidas na sociedade contemporânea. Contudo, a maioria das experiências lúdicas digitais, continua a privilegiar sedentarismo e limitando o potencial do jogo enquanto meio de promoção de atividade física e de intervenção cognitiva. Em paralelo, as tecnologias de interação, nomeadamente o rastreamento de mãos e os sistemas de comando por voz, têm evoluído rapidamente e oferecem novas possibilidades para projetar experiências mais imersivas, acessíveis e orientadas para objetivos terapêuticos ou educativos. (Gonçalves, 2022)

O presente estudo insere-se neste contexto e propõe analisar o potencial de utilização de gestos e de voz em videojogos com ênfase no desenvolvimento cognitivo e na recuperação de memória. Contrariamente a abordagens centradas exclusivamente na componente física, o trabalho foca-se em como mecanismos de interação natural (movimentos manuais e voz) podem suportar dinâmicas de jogos que estimulem funções cognitivas e como essas dinâmicas se traduzem em usabilidade, aceitabilidade e eficácia junto de utilizadores reais. (Chung, Griffin, Selezneva, & Gotz, 2018)

Para atingir este objetivo, o estudo combina investigação teórica e investigação experimental. Foi realizado um levantamento do estado da arte sobre interfaces gestuais e de voz aplicadas a aplicações e a jogos e intervenções cognitivas. Foram desenvolvidos dois protótipos funcionais: um baseado em rastreamento de mão e outro em comando de voz; e efetuados testes com utilizadores para avaliar a aplicabilidade, a experiência de utilização e potenciais benefícios na esfera cognitiva. A recolha de dados incluiu medidas objetivas de desempenho nos protótipos, observação e questionários para recolher percepções subjetivas e sugestões de melhoria. Este enquadramento justifica-se pela convergência entre avanços tecnológicos (sensores de mãos, modelos de reconhecimento de voz), necessidades sociais (procura por soluções lúdicas que promovam saúde cognitiva) e requisitos de acessibilidade (interfaces alternativas que podem beneficiar utilizadores com limitações motoras ou preceptivas) As contribuições esperadas deste estudo incluem:

- 1) Avaliação empírica da viabilidade de gestos e voz como mecanismos de interação para estímulos cognitivos em contexto de jogos;
- 2) Recomendações de design para integrar essas técnicas em aplicações educativas e de reabilitação;
- 3) Dois protótipos de referência que ilustram padrões de interação e possíveis caminhos para trabalhos futuros.

Em termos de delimitação, o estudo centra-se na conceção, implementação e avaliação inicial dos protótipos, não abrangendo neste momento ensaios longitudinais de reabilitação clínica. No final do trabalho será apresentada uma discussão crítica sobre limitações, implicações práticas e linhas futuras de investigação, incluindo estudos longitudinais, integração com realidade aumentada (RA) e adaptação para contextos clínicos.

1.2 Motivação

A escolha deste projeto surgiu do interesse pessoal em explorar novas formas de interação em videojogos. Ao deparar-me com a proposta, percebi que oferecia uma oportunidade única de unir duas áreas que considero estimulantes: a tecnologia e a acessibilidade. O tema motivou-me não apenas pela curiosidade em experimentar ferramentas inovadoras, como o rastreamento de mãos e o comando por voz, mas também pelo potencial impacto positivo que estas abordagens podem ter. A possibilidade de desenvolver soluções que promovam uma experiência de jogo mais inclusiva, capaz de reduzir o sedentarismo e, ao mesmo tempo, estimular capacidades cognitivas, reforçou a vontade de abraçar este desafio.

1.3 Stakeholders

Para o sucesso deste projeto é essencial identificar, compreender e envolver os todas as partes interessadas que podem ter um impacto ou serem impactadas pelo projeto (*stakeholders*) (Vaz, 2025), uma vez que desempenham papéis críticos no planeamento, implementação e no impacto social da iniciativa. Foram identificados seis *stakeholders* com níveis de poder e interesse distintos, o que condiciona as estratégias de comunicação e de envolvimento. Os cinco *stakeholder*:

- O **SIIS** (Sistemas de Investigação e Inovação Social) foi definido como *stakeholder* primário, por ter providenciado o contexto e recursos necessários ao estudo e desenvolvimento do projeto;
- Os utilizadores finais (**jogadores**) apresentam interesse elevado e poder intermédio, influenciando diretamente a aceitação e a evolução do produto através do seu feedback;
- **Instituições de saúde e reabilitação** constituem um *stakeholder* estratégico com interesse elevado e poder intermédio, dado o potencial de aplicação terapêutica dos protótipos;
- **Instituições de ensino e educadores** são outros *stakeholders* estratégico, mas com interesse médio e poder reduzido, representando uma oportunidade de disseminação no contexto educativo;
- As **empresas de tecnologia e hardware** detêm alto poder e interesse intermédio, sendo potenciais parceiras para fornecimento de dispositivos, suporte técnico e desenvolvimento conjunto;
- **Pessoas com necessidades especiais** (utilizadores com limitações motoras, visuais, cognitivas ou outras necessidades de acessibilidade) com um interesse elevado e poder baixo, ao nível individual, mas intermédio a elevado quando organizadas ou representadas por associações. Justifica-se a sua inclusão por constituírem um grupo-alvo das funcionalidades de acessibilidade e a sua participação é fundamental para validar a usabilidade, a eficácia e a ética das soluções propostas.

Com base nesta caracterização, definiram-se estratégias específicas de envolvimento para cada grupo, como testes com utilizadores e instituições de saúde até parcerias técnicas, de modo a maximizar o impacto e a viabilidade técnico-comercial do projeto. Na Tabela 1, podemos ver numa forma resumida os *stakeholder*, juntamente com os seus níveis de poder e

interesse e também uma estratégia de como seriam as comunicações com esses *stakeholder*. E na Figura 1 podemos ver uma matriz dos mesmos *stakeholders*, dando uma visão diferente.

Tabela 1 - Tabela dos *stakeholders*, níveis de poder/interesse e estratégias de envolvimento.

Stakeholder	Papel / Interesse principal	Poder	Interesse	Exemplo de como será o envolvimento
SIIS	Entidade anfitriã e patrocinadora do estudo; fornece contexto e recursos	Alto	Alto	- Relatórios periódicos; - Reuniões de acompanhamento mensais; - Validação de requisitos
Jogadores / Utilizadores finais	Teste de usabilidade, feedback e adoção do sistema	Médio	Alto	- Testes de utilizadores; - Questionários;
Pessoas com necessidades especiais	Grupo-alvo das funcionalidades de acessibilidade; Validação prática e ética	Médio	Alto	- Teste de usabilidade; - Inquéritos/questionários; - Sessões de jogo observadas;
Instituições de Saúde e Reabilitação	Aplicação terapêutica (fisioterapia, reabilitação cognitiva)	Médio	Alto	-Workshops clínicos; - Sessões de com terapeutas
Instituições de Ensino / Educadores	Uso pedagógico e integração curricular	Baixo	Médio	- Demonstrações em escolas; - Materiais pedagógicos; - Formações para professores
Empresas de Tecnologia / Hardware	Fornecimento de dispositivos, integração técnica, escalabilidade	Alto	Médio	- Parcerias técnicas; - Provas de conceito; - Acordos de suporte/fornecimento

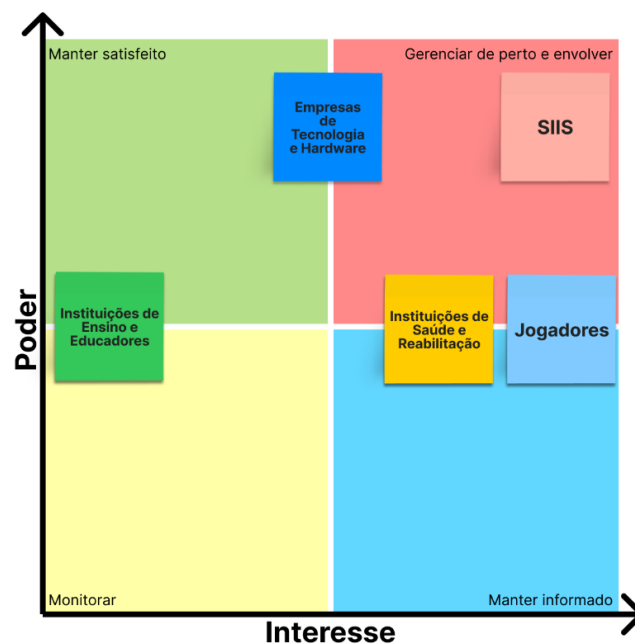


Figura 1 - Matriz de stakeholders

1.4 Planeamento

O planeamento é uma etapa fundamental para garantir uma organização e execução eficaz de um projeto. Este projeto tem como objetivo explorar o uso de tecnologias de rastreamento da mão e comando de voz como ferramentas para reduzir o sedentarismo e promover o desenvolvimento cognitivo e estimular a memória, para além de oferecer uma experiência de jogo mais acessível e inclusiva. O planeamento foi estruturado em diferentes fases, seguindo uma abordagem baseada em *Research Strategies*, adaptada para incluir pesquisas teóricas na etapa inicial do projeto. A estrutura organizacional foi feita numa WBS (*Work Breakdown Structure*), que identifica as atividades principais e os entregáveis essenciais para o sucesso do projeto.

1.4.1 Metodologia

O escopo deste trabalho está alinhado à metodologia de *Research Strategies*, que envolve as seguintes fases principais:

1. **Planeamento:** Inicialmente, serão definidos todos os requisitos a serem feitos do projeto, revisão da literatura que focam em estudos e relatórios de implementações (sendo que esta parte será a conclusão do que já foi começado em PREPD) e a

definição das tecnologias e linguagens de programação que serão usadas no decorrer do projeto.

2. **Design:** Com base nas pesquisas e planeamento, será feita uma pesquisa visual que explore diferentes aplicações de gestos e voz e será desenvolvido um conjunto de elementos visuais e protótipos não funcionais. Esses protótipos incluirão não só alguns elementos visuais, como os visuais dos menus e dos minijogos
3. **Desenvolvimento dos Protótipos:** Nesta fase será desenvolvido já os protótipos funcionais dos minijogos, tanto os que utilizam rastreamento de mão como os que utilizam comandos de voz, com as tecnologias e design definidos nas etapas anteriores.
4. **Avaliação e Análise de Resultados:** Esta fase serão feitos testes com utilizadores, juntamente com um questionário para ser possível recolher dados sobre a usabilidade dos jogos, como a satisfação dos jogadores com os jogos e se os jogam estão a cumprir o objetivo do projeto. Depois desses testes, com os dados obtidos, serão analisados para refinar os protótipos e validar a viabilidade das soluções propostas.
5. **Documentação e Conclusão:** E nesta fase final será focada a só trabalhar no relatório da dissertação, como descrever o desenvolvimento do projeto e dos resultados esperados comparando com os obtidos, entre outros tópicos. Nesta parte também se começará a realizar-se a apresentação e começar a treinar para ela para ter uma apresentação mais fluida e por fim a submissão do trabalho final.

Podemos ver a estrutura deste planeamento na Figura 2 que representa a estrutura WBS do projeto, onde é possível também ver que cada categoria foi dividida em sub-tarefas específicas para garantir que todos os aspetos do projeto sejam abordados de forma organizada. Este planeamento também serviu como base para a criação do cronograma detalhado do projeto.

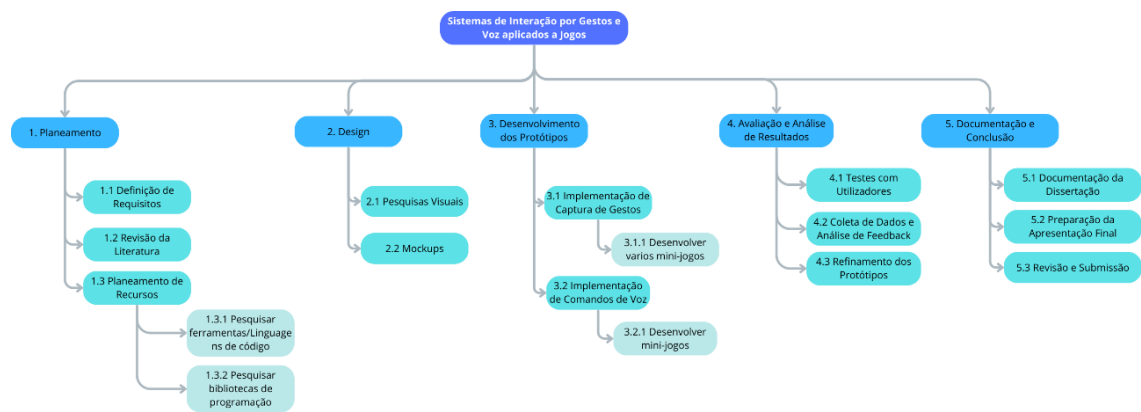


Figura 2 - WBS

1.4.2 Cronograma do Projeto

Após ter sido finalizado o WBS do projeto, passou-se para a criação do cronograma do projeto, fazendo um diagrama de *gantt* usando inicialmente o *Merlin Project*, que é uma ferramenta eficiente para gestão de projetos para o sistema operativo macOS. Mas que posteriormente foi alterado para o *ProjectLibre*, por motivos de ser gratuito e ser compatível tanto em Mac como em Windows, devido ao fato do projeto inicialmente estar a ser trabalhado nos dois sistemas operativos, mas principalmente em Windows.

O diagrama de *gantt* oferece uma representação visual das atividades e suas relações, facilitando o acompanhamento do progresso ao longo do tempo.

Tanto o *Merlin Project* como *ProjectLibre* ajudam-nos a identificar cada tarefa e as suas durações, facilitando uma distribuição eficiente de tempo e esforço para cada tarefa. Ele também oferece uma visualização do progresso, já que nele é possível meter percentagens de progresso de cada atividade e ao colocar as percentagens em atividade também mostra uma percentagem geral de como esta o progresso do projeto em si. E por fim, caso uma tarefa demore mais que o tempo previsto, ele deixa um alerta, em vermelho, que a tarefa demorou mais que o previsto e que as tarefas que estavam dependentes delas também começaram mais tarde que o previsto.

Este projeto terá uma duração prevista de 128, começando dia 03 de fevereiro de 2025 e um termino dia 30 de julho de 2025, sem contar com a parte já desenvolvida em unidade curricular (UC) de preparação para a dissertação (PREPD), que teve início no dia 16 de outubro de 2024 e vai durar até dia 04 de janeiro de 2025, contabilizando mais 58 dias de trabalho para o projeto.

Na Figura 3, podemos ver o cronograma inicial do projeto (feito em excel para ter uma melhor visibilidade), como a distribuição das tarefas por uma linha de tempo.

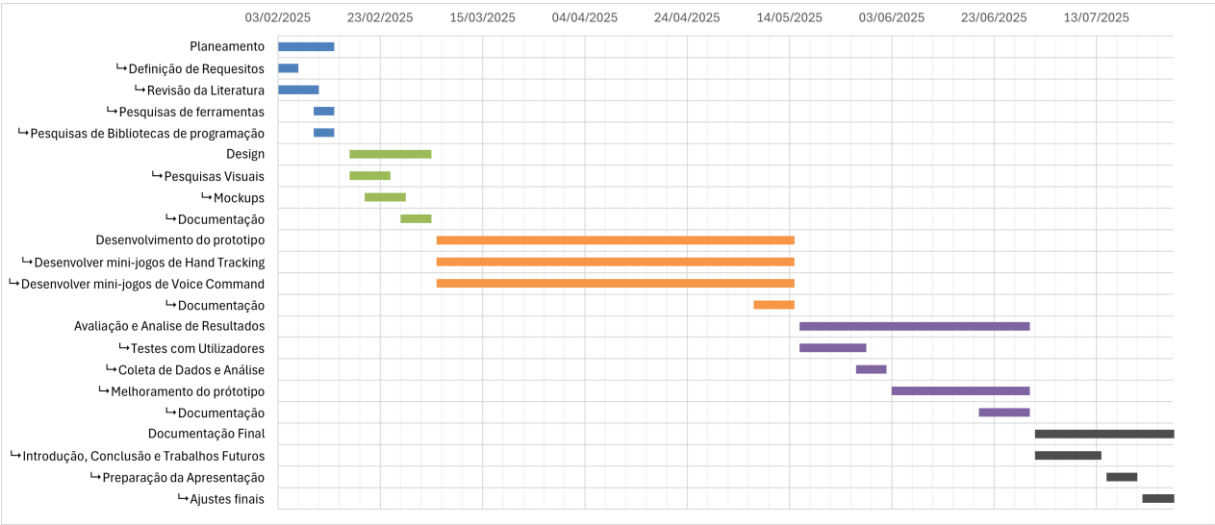


Figura 3 - Cronograma inicial do projeto

Na Figura 4 podemos ver as Milestone (ou “marco” em português) deste do cronograma inicial. As Milestone é um ponto de referência importante num projeto, como as definidas neste planeamento, os términos das partes do projeto e as revisões planeadas com o orientador do projeto. Não tem duração ou custo, servindo como um sinalizador de progresso e um ponto de controle essencial para monitorar o avanço do projeto. (Wikipédia, 2021)

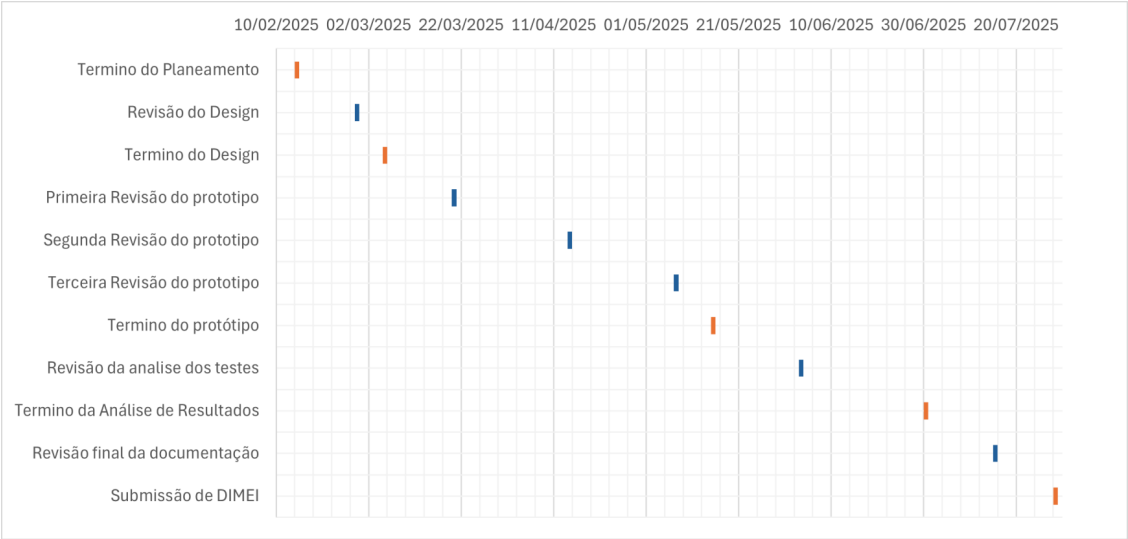


Figura 4 - Cronograma inicial do projeto (Milestone)

1.4.3 Custos

Por se tratar de um trabalho acadêmico, não houve custos associados ao pagamento de horas de trabalho, uma vez que o desenvolvimento foi realizado pelo aluno com a supervisão do orientador.

No que diz respeito aos equipamentos, também não foram gerados custos adicionais, dado que todos os dispositivos necessários já tinham sido previamente adquiridos e se encontram disponíveis para utilização. Relativamente aos softwares, foram utilizados exclusivamente ferramentas gratuitas, como *Python* e *Unity*.

Desta forma, o único “custo” envolvido no projeto foi o tempo investido pelo aluno e pelo orientador no planeamento, desenvolvimento e execução das atividades previstas.

1.5 Análise de Riscos

Para garantir o sucesso do desenvolvimento deste projeto, foi realizada uma análise de riscos detalhada com o objetivo de identificar possíveis obstáculos, avaliar suas causas e impactos e definir estratégias para mitigação. Esta abordagem permite antecipar problemas e estabelecer ações preventivas, reduzindo as chances de atrasos ou resultados insatisfatórios.

A análise consistiu na criação de uma tabela com os seguintes fatores para cada risco identificado:

- **Descrição:** Breve explicação sobre a natureza do risco;
- **Causa:** Principais motivos pelos quais o risco pode ocorrer;
- **Efeito:** Consequências esperadas caso o risco se concretize.
- **Responsável do Risco:** Responsável por monitorar e gerenciar o risco;
- **Probabilidade e Impacto:** Escala de 1 a 5 que avalia a chance de ocorrência e a gravidade do impacto, respetivamente;
- **Pontuação de Prioridade (PI Score):** Resultado da multiplicação da probabilidade pelo impacto, utilizado para priorizar a resposta;
- **Resultado Esperado Sem Ação:** O que aconteceria caso o risco não fosse tratado;
- **Tipo de Resposta ao Risco:** Estratégia a ser adotada (mitigação, aceitação, etc...);
- **Descrição da Resposta:** Medidas específicas para lidar com o risco identificado.

E com a construção desta tabela foi possível identificar 5 principais riscos, sendo eles:

1. **Dificuldade na Integração de Gestos e Voz em Jogos:** Este risco apresenta uma alta prioridade (PI Score de 20), dado o impacto crítico no desempenho do protótipo. Para mitigar este risco, será necessário explorar e ajustar o escopo das tecnologias disponíveis.
2. **Falta de Padrões Claros para Jogos com Foco em Gestos e Voz:** Com PI Score de 12, este risco será tratado com criatividade e foco nos objetivos do projeto, já que a sua aceitação é mais viável do que a mitigação completa.
3. **Dificuldades na Realização de Testes com Utilizadores:** Também com PI Score de 12, este risco será mitigado através da preparação de alternativas, como testes online ou ampliação da base de participantes.
4. **Atrasos por Parte dos Utilizadores nos Testes:** Este risco, embora não seja crítico tendo um PI Score de 12 também, pode comprometer os prazos do projeto. Serão implementadas estratégias de comunicação e incentivos para manter os utilizadores engajados.
5. **Impossibilidade de Reunião com o Orientador em Momentos Cruciais:** Identificado como um risco moderado (PI Score de 9), este será mitigado com o agendamento de reuniões regulares e antecipadas.

Esta análise proporciona uma visão clara dos potenciais desafios do projeto e das medidas necessárias para minimizar os seus impactos. A revisão e atualização contínuas dos riscos identificados serão realizadas ao longo do desenvolvimento do projeto, garantindo assim que o projeto mantenha a sua trajetória planejada e seja possível a sua realização por completo.

No Anexo C – Análise de Risco podemos ver a tabela criada para a análise de risco.

1.6 Considerações Éticas

As considerações éticas são essenciais na investigação em engenharia informática, pois influenciam diretamente a integridade e o impacto social do trabalho. Esta seção aborda essa consideração tal como se aplicam à investigação e ao desenvolvimento deste projeto.

É fundamental realçar que este estudo foi conduzido em conformidade com as normas e diretrizes éticas referidas pela National Society of Professional Engineers (NSPE) (National Society of Professional Engineers, s.d.). A Declaração de Integridade constante deste documento foi preparada em cumprimento do Código de Boas Práticas e Conduta do P.Porto (Politécnico do Porto, 2020) que também foi integralmente respeitada.

Nos testes realizados com utilizadores, não foram solicitados dados pessoais identificáveis, podendo realizar os ensaios de forma totalmente anónima. Para fins demográficos foram recolhidos apenas a idade e o género.

1.7 Metodologia de investigação

A revisão de literatura é uma etapa essencial para fundamentar teoricamente o desenvolvimento deste projeto, que visa explorar a integração de rastreamento de mão e comandos de voz em jogos, com foco em acessibilidade e inovação. Por meio de uma análise criteriosa de estudos e publicações relevantes, busca-se compreender o estado da arte dessas tecnologias, seus benefícios, limitações e aplicações no contexto dos jogos digitais.

Nesta secção serão abordados os principais métodos de revisão de literatura, discutindo-se as diferenças entre abordagens tradicionais e sistemáticas. Além disso, serão apresentadas as ferramentas e estratégias utilizadas para identificar e analisar as fontes relevantes, incluindo o uso de bibliotecas digitais e tabelas PICOCS para a definição de critérios de pesquisa.

O objetivo é fornecer uma base sólida para responder às questões de investigação propostas e identificar direções promissoras para o desenvolvimento de jogos acessíveis e inovadores.

1.7.1 Métodos de Revisão

A revisão de literatura é uma etapa fundamental para contextualizar o projeto e identificar trabalhos prévios relacionados. Para este projeto, dois métodos principais foram considerados:

1. **Revisão Tradicional/Informal:** Este método permite uma abordagem mais flexível, proporcionando uma visão geral do tema sem necessidade de seguir um protocolo rígido. Ele é útil para explorar tópicos amplos e estabelecer os primeiros passos da pesquisa.
2. **Revisão Sistemática:** Este método, por outro lado, segue um protocolo estruturado para identificar avaliar e sintetizar estudos relevantes de maneira mais rigorosas. A revisão sistemática será aplicada para analisar estudos que abordem:
 - a. O impacto do rastreamento de mão e comando de voz em jogos.
 - b. Níveis de acessibilidade e usabilidade dessas tecnologias.
 - c. Benefícios e limitações dessas abordagens.

Para este projeto, também foram desenvolvidas tabelas PICOCS, já que elas são ferramentas cruciais neste contexto, permitindo uma seleção criteriosa de estudos que atendam aos critérios de inclusão e exclusão, como por exemplo:

- **Inclusão:** Estudos empíricos relacionados a jogos que utilizam rastreamento de mão e comando de voz.
- **Exclusão:** Artigos desatualizados ou que não avaliem diretamente a experiência do jogador e não serem artigos anteriores a 2015.

Inicialmente foram criadas duas tabelas PIPOCS com um foco em procurar jogos que utilizam sistemas de rastreamento de mãos e comando de voz, tendo em conta a experiência do utilizador final. Mas após algumas pesquisas realizadas com essas tabelas, identificou-se a necessidade de expandir o escopo da pesquisa. Assim as tabelas criadas anteriormente, foi adicionado que os focos não seriam apenas em jogos, mas também aplicações que utilizam esses sistemas. Essa aplicação permitiu explorar como as empresas, além daquelas das indústrias dos jogos, implementam estratégias para aumentar a acessibilidade de seus produtos rastreamento e comando de voz.

Tabela 2 - PICOCS Rastreamento de mão em Jogos

PICOCS	RQ part	I/E	Sub String
P	Jogos/Aplicações que utilizam sistemas de rastreamento de mão	I – Estudos que envolvem jogos/aplicações que utilizam rastreamento de mão. E - Estudos publicados antes de 2015	-Player immersion; -User experience -User satisfaction; -User feedback on motion controls;
I	Rastreamento de mão	I – Estudos que investigam a utilização de rastreamento de mão em jogos/aplicações. E – Estudos que se focam exclusivamente em tecnologia sem avaliar a experiência do utilizador.	-Hand Tracking in games or/and applications; -Motion detection in games or/and applications;;
C	---	---	---
O	Benefícios e limitações no uso de rastreamento de mão	I – Estudos que analisam o impacto na experiência do utilizador (imersão, envolvimento).	-Gaming/ Application performance with hand tracking; -User satisfaction in gesture-based;

			-Gameplay usability; - UI;
C	Indústria de jogos e Projetos de aplicações	I – Estudos relacionados à indústria dos videogames/aplicações. E - Estudos que não estejam relacionados com a indústria dos videogames/aplicações	- Industry industry innovations; - Development with tracking;
S	Estudo empírico com experimentos e inquéritos com utilizadores	I - Estudos empíricos, inquéritos, e estudos de caso sobre jogos/aplicações que utilizam rastreamento de mão. E - Artigos de opinião ou que não envolvam experimentação prática.	-Empirical study; -Case study; -User feedback; -Industry reports on gaming/application interaction technologies;

Tabela 3 - PICOCS comando de voz em Jogos

PICOCS	RQ part	I/E	Sub String
P	Jogos/Aplicações que utilizam sistemas de comando de voz	I – Estudos que envolvem jogos/aplicações que utilizam comando de voz. E - Estudos publicados antes de 2015	-Player immersion; -User experience; -User satisfaction; - User feedback on voice controls;
I	Comando de voz	I – Estudos que investigam a utilização de comando de voz em jogos/aplicações. E – Estudos que se foquem exclusivamente em tecnologia sem avaliar a experiência do utilizador.	-Voice-controlled in games/applications; -Speech recognition in gaming/applications;
C	---	---	---
O	Benefícios e limitações no uso de comando de voz	I – Estudos que analisam o impacto na experiência do utilizador (imersão, envolvimento).	-Effectiveness of voice commands in games; -Application performance with Voice command; -Voice command usability;

			-User satisfaction with Voice command; - UI;
C	Indústria de jogos	I – Estudos relacionados à indústria dos videogames e/ou projetos de aplicações. E - Estudos que não estejam relacionados com a indústria dos videogames e projetos de aplicações	- Industry innovations; -Voice command integration in games; - Development with Voice command;
S	Estudo empírico com experimentos e inquéritos com jogadores	I - Estudos empíricos, inquéritos, e estudos de caso sobre jogos/aplicações que utilizam comando de voz. E - Artigos de opinião ou que não envolvam experimentação prática.	-Empirical study; -Case study; -Player feedback; -Industry reports on gaming interaction technologies;

1.7.2 Bibliotecas Digitais

Para conduzir uma revisão de literatura abrangente e baseada em evidências, foram utilizadas, principalmente, plataformas de bibliotecas digitais como *IEEE Xplore*, *ACM Digital Library*, *ResearchGate* e *Google Scholar*, já que elas permitem o acesso a artigos científicos e relatórios de projetos relevantes.

Além disso, as *strings* de buscas definidas nas tabelas PICOCS, foram usadas para localizar estudos relacionados às questões de investigação, como:

- Qual é o nível de acessibilidade dos jogos que usam Rastreamento de Gestos e/ou Comandos de Voz?
- Que jogos ficariam interessantes utilizando Rastreamento de Gestos e/ou Comandos de Voz?

Para além das bibliotecas digitais, também se utilizou o motor de pesquisa do Chat GPT (versão 4o), já que ele é capaz de pesquisar muito mais rápido e em muitas mais plataformas do que uma pessoa normal, sendo só preciso passar os parâmetros feitos nas tabelas PICOCS e dar uma pequena explicação do que se esta a procura.

1.7.3 Estratégia de Pesquisa

A estratégia principal de pesquisa adotada foi a análise inicial das secções de Resumo, Introdução e Conclusão dos relatórios e artigos científicos. Essas secções oferecem uma visão geral condensa sobre o conteúdo abordado. Caso fosse identificado que o projeto ou estudo em questão utilizava rastreamento de gestos e comando de voz como foco principal, além de incluir análises realizadas com utilizadores ou estudos relevantes, o documento era selecionado para uma leitura mais detalhada no futuro. Caso contrário, era descartado imediatamente, evitando assim a perda de tempo com materiais que não contribuíam diretamente para os objetivos deste projeto.

1.8 Estrutura do Documento

O presente documento foi organizado em sete capítulos como a seguir se descreve.

O “Capítulo 1 – Introdução”, apresenta o enquadramento do tema, destacando a relevância dos videojogos como ferramentas de entretenimento, educação e intervenção cognitiva. Define os stakeholders, o planeamento, a metodologia de investigação, os riscos identificados e as considerações éticas. Conclui com a descrição da metodologia de pesquisa aplicada à revisão da literatura.

No “Capítulo 2 – Estado da Arte” explora-se o conhecimento existente sobre:

- Acessibilidade: conceitos gerais, acessibilidade digital e em jogos.
- Rastreamento de gestos: tecnologias, algoritmos, aplicações em jogos, vantagens e desafios.
- Comando de voz: frameworks de reconhecimento e síntese de fala (STT e TTS), aplicações práticas em jogos e outros contextos, bem como limitações.

Este capítulo constitui a base teórica que suporta o desenvolvimento do projeto.

O “Capítulo 3 – Tecnologias Utilizadas” apresenta as ferramentas e recursos aplicados durante o projeto, como Adobe Illustrator, Figma, GitHub, Python, Unity, Visual Studio Code, Google Forms, ChatGPT, entre outros. Cada tecnologia é justificada em função do seu papel no design, desenvolvimento e testes.

O “Capítulo 4 – Design” detalha o processo de design da aplicação:

- Wireframes e wireflows que estruturam os menus e minijogos;
- Paleta de cores escolhida com foco em acessibilidade;

- Mockups visuais, diferenciando as versões para rastreamento de mãos e para comando de voz.

O design privilegia clareza, acessibilidade e imersão.

O “Capítulo 5 – Desenvolvimento do Protótipo” Explica o processo de implementação:

- Configuração dos ambientes de Python e Unity;
- Desenvolvimento dos módulos de rastreamento de mãos e comando de voz;
- Integração com os minijogos;
- Principais desafios técnicos enfrentados, como latência, calibração e usabilidade.

O “Capítulo 6 – Testes com Utilizadores” apresenta a avaliação prática dos protótipos:

- 1.º teste de utilizadores: perfil, resultados e conclusões iniciais;
- 2.º teste de utilizadores: perfil, análises mais detalhadas, métricas de desempenho e percepções de usabilidade;
- Os testes comparam o impacto de cada modalidade (voz vs. gestos), identificando vantagens, limitações e sugestões de melhoria.

Finalmente, no “Capítulo 7 – Conclusão e Trabalhos Futuros” apresenta-se um resumo sobre os principais contributos do estudo e abordam-se algumas propostas de continuidade, incluindo:

- Integração híbrida de gestos e voz;
- Multijogador local e online;
- Aplicações voltadas para acessibilidade e reabilitação cognitiva;
- Integração com assistentes virtuais e exploração de novas tecnologias STT/TTS.

Conclui-se o capítulo com uma reflexão sobre a viabilidade técnica e social da abordagem.

2 Estado da Arte

Neste capítulo será explorado a literatura existente sobre acessibilidade, rastreamento de gestos e comandos de voz em jogos e aplicações digitais. São discutidas vantagens, limitações e desafios destas abordagens, bem como tecnologias e algoritmos utilizados. A análise evidencia o potencial destas soluções para promover experiências interativas mais inclusivas e inovadoras.

2.1 Acessibilidade e Usabilidade

A acessibilidade é um conceito que visa garantir que todas as pessoas, independentemente das suas limitações físicas, cognitivas, sensoriais ou tecnológicas, possam usufruir de produtos, serviços e ambientes de forma eficaz e independente. Na sua essência, a acessibilidade promove a inclusão, garantindo que as barreiras que limitam o acesso e a interação sejam minimizadas ou até mesmo eliminadas. A acessibilidade tem um foco em várias áreas, como em construções, transportes, educação e mais recentemente, tecnologias digitais.

Com o avanço da tecnologia e com o crescimento digital na vida das pessoas, a acessibilidade digital tornou-se um fator primordial. Ferramentas e interfaces digitais, como por exemplo: websites, aplicações moveis e jogos, precisam de ser desenhados de forma inclusiva para permitir que pessoas com diferentes tipos de deficiência (como visuais, auditivas ou motoras) possam utilizá-los plenamente. Este movimento em direção a um design inclusivo foi impulsionado por iniciativas globais, como as *Diretrizes de Acessibilidade para Conteúdo Web (WCAG)*, que fornecem orientações sobre como tornar o conteúdo digital mais acessível. (W3C, 2024)

No contexto dos jogos eletrônicos, a acessibilidade ganha ainda importância mais importância, pois muitas vezes, os jogos são uma forma essencial de entretenimento, interação social e até de desenvolvimento cognitivo. Jogos acessíveis não promovem apenas inclusão, mas também podem ser usados para criar experiências significativas para pessoas com deficiências, proporcionando uma oportunidade para todos se envolverem em atividades lúdicas, independentemente das suas capacidades físicas ou cognitivas.

Além da acessibilidade, o conceito de usabilidade é igualmente relevante para o desenho de interfaces e aplicações digitais. A usabilidade refere-se à facilidade com que utilizadores reais conseguem aprender a usar um sistema, executar tarefas com eficiência e alcançar os seus

objetivos, bem como ao nível de satisfação subjetiva da interação. Em particular, a usabilidade abrange critérios com eficiência (tempo e esforço para completar uma tarefa), eficácia (taxa de sucesso) e satisfação do utilizador, bem como práticas de design que reduzem erros e fornecem feedback claro. (Torres, 2025) (Silveira, 2024)

Num projeto que tem por objetivo tornar jogos e aplicações mais inclusivos, a usabilidade e a acessibilidade tornam-se complementares: uma solução pode ser tecnicamente acessível (por exemplo: suportar leitores de ecrã ou comandos de voz), mas só será realmente útil se essas opções foram intuitivas, eficazes e agradáveis de usar. Por esse motivo, este trabalho não só considera as normas de acessibilidade (por ex., WCAG) como também avalia aspetos de usabilidade nas interações (ex.: facilidade de aprendizagem dos minijogos, perceção de latência e precisão no reconhecimento por voz e hand-tracking, e satisfação dos utilizadores), de modo a orientar recomendações de design que resultem em experiências inclusivas e funcionais em contextos reais de jogo.

2.1.1 Acessibilidade na vida

A acessibilidade na vida real é um conceito abrangente que visa garantir que todas as pessoas possam ter acesso e utilizar os espaços, produtos, serviços na sociedade de uma maneira mais independente e segura. Este conceito é particularmente relevante em áreas como a arquitetura, transportes, comunicação e em ambientes digitais. A acessibilidade é um direito fundamental, assegurado em muitos países por legislações como a Convenção sobre os Direitos das Pessoas com Deficiência das Nações Unidas (ONU), que promove a inclusão e igualdade de oportunidades. (United Nations, 2016)

No contexto da arquitetura e dos espaços públicos, a acessibilidade envolve o design universal, que consiste na criação de ambientes que possam ser utilizados pelo maior número de pessoas possíveis, sem necessidades de adaptações. Um exemplo prático são as rampas para cadeiras de rodas, elevadores acessíveis, sinalizações em braille e sistemas de alerta visual e auditiva. O objetivo deste tipo de acessibilidade é eliminar barreiras físicas que possam impedir o acesso de pessoas com mobilidade reduzida ou outras limitações físicas, garantindo assim uma participação plena e equitativa na sociedade. (HAPI, 2015) (Pereira, et al., 2019).

Outra área que a acessibilidade também se aplica, são a serviços essenciais, como transportes. Veículos adaptados, paragens de autocarro acessíveis e sistemas de informação sonora em comboios são exemplos de como o setor dos transportes tem vindo a adotar práticas inclusivas

para garantir que pessoas com deficiência possam deslocar-se de forma independente para outros lugares.

Outro domínio relevante é a comunicação e o acesso à informação. Tecnologias assistivas como leitores de ecrã, legendas, audiodescrição e até Inteligências Artificiais (IA) com linguagem gestual são fundamentais para pessoas com deficiências sensoriais, como a surdez ou a cegueira. A acessibilidade digital tem tido um foco crescente, com padrões como as WCAG a delinear as melhores praticas para garantir que todos possam aceder à informação online.

A acessibilidade não é apenas uma questão técnica, mas uma questão de inclusão social e de direitos humanos. Ao eliminar barreiras e promover a acessibilidade em todas as esferas da vida, cria-se uma sociedade mais justa e equitativa, onde todos, independentemente das suas capacidades, podem participar plenamente em qualquer atividade.

2.1.2 Acessibilidade digital e interativa

A acessibilidade digital visa proporcionar acesso igualitário a informações e serviços online, permitindo que indivíduos com deficiências utilizem tecnologias assistias, como leitores de ecrãs e ampliadores, para navegar na internet. Para que um site seja acessível, é essencial que ele siga padrões e diretrizes estabelecidas, como as recomendações do W3C que orientam o desenvolvimento de conteúdos web acessíveis.

Estudos recentes destacam a importância de avaliar a acessibilidade em websites de comunicação científica, propondo etapas que incluem o cumprimento de padrões web, aplicação de diretrizes de acessibilidade e realização de avaliações específicas. Essas medidas são cruciais para garantir que a informação científica seja acessível a todos, promovendo a inclusão no ambiente acadêmico e científico. (Morais, Andrade, Sena, Segundo, & Amaro, 2023) Além disso, a acessibilidade digital não se limita apenas a websites, mas também se estende a aplicações móveis e outros serviços digitais. A implementação de práticas de design universal, que considerem a diversidade dos utilizadores desde a conceção dos produtos, é fundamental para assegurar que todos possam beneficiar-se das inovações tecnológicas. (Esperança, Sousa, Costa, Pereira, & Francisco, 2021)

Em resumo, a acessibilidade digital em aplicações e websites é uma área em constante evolução, impulsionada por avanços tecnológicos e por uma crescente conscientização sobre a importância da inclusão digital. A adoção de práticas acessíveis no desenvolvimento de

produtos digitais é essencial para construir uma sociedade mais justa e equitativa, onde todos tenham a oportunidade de participar plenamente do mundo digital.

2.1.3 Acessibilidade em jogos

No caso da acessibilidade nos jogos, ela tem recebido um aumento de atenção nos últimos anos, resultando em avanços significativos para tornar os jogos mais inclusivos para pessoas com deficiência.

Um exemplo notável é o jogo *The Last of Us Part II*, lançado em 2020, que incorporou uma ampla gama de opções de acessibilidade, incluindo recursos específicos para jogadores com deficiência visual e auditiva. Essas opções permitem personalizar a experiência do jogo, tornando-a mais inclusiva e adaptada às necessidades individuais.

Outro jogo que é um exemplo notável, é o *Forza Horizon 5*, lançado em 2021. O jogo foi concebido em estreita colaboração com a comunidade de jogadores com deficiência e as Diretrizes de acessibilidade da Xbox (Microsoft, 2021). O jogo estabeleceu um novo padrão para a acessibilidade sensorial, particularmente para jogadores com deficiência auditiva e visual. Um dos recursos mais inovadores é suporte para Língua de Sinais Americana (ASL) e Britânica (BSL) para todas as cinemáticas do jogo. Este recurso, com um intérprete em tempo real exibido em um pequeno ecrã, é um avanço significativo que permite aos jogadores surdos ou com deficiência auditiva total acompanharem a história e os diálogos de forma completa. Complemento disso, o jogo inclui um *Screen Reader Narrator* que lê em voz alta o texto de menus e outros elementos da interface, além de opções de *Text-to-speech* (TTS) e *Speech-to-text* (STT) para a comunicação em chat de voz, garantindo que os jogadores possam interagir socialmente e navegar no jogo de forma independente.

Além disso, iniciativas acadêmicas têm explorado soluções inovadoras para melhorar a acessibilidade em jogos. Um exemplo, é o estudo *“NavStick: Making Video Games Blind-Accessible via the Ability to Look Around”* (Nair, et al., 2021) da *Columbia University* de Nova Iorque, Estados Unidos da América (EUA), que propôs uma ferramenta baseada em áudio que permite a jogadores com deficiência visual explorar ambientes virtuais de forma mais intuitiva, aumentando sua autonomia e imersão nos jogos.

Outro exemplo é o estudo *“Accessibility in Video Games: A Review”* (Shrestha & Karki, 2022) da *Nepal College of Information Technology* que oferece uma revisão abrangente sobre práticas e tendências relacionadas à acessibilidade em jogos. Este trabalho identifica os principais desafios enfrentados por desenvolvedores ao implementar soluções acessíveis, como por exemplo “A

ausência de diretrizes padronizadas cria dificuldades para garantir consistência nas soluções de acessibilidade, o que pode levar a variações significativas na experiência do jogador com deficiência." (Shrestha & Karki, 2022)(tradução livre do autor), e destaca a importância de padrões de design inclusivos na indústria.

Esses exemplos ilustram o compromisso crescente da indústria de jogos e da comunidade acadêmica em promover a inclusão, garantindo que os jogos digitais sejam mais acessíveis e desfrutados por um público ainda mais amplo.

2.2 Rastreamento de gestos

O rastreamento de gestos (ou de mãos) é uma tecnologia que permite a detecção e interpretação de movimentos corporais, em particular das mãos, para interação com sistemas digitais. Esta abordagem oferece uma forma intuitiva e natural de controle, eliminando qualquer necessidade de dispositivos físicos, como ratos, teclados ou comandos. Baseando-se em câmaras, sensores de movimentos e algoritmos de IA, esta tecnologia identifica gestos específicos ou posições das mãos para traduzir as ações do utilizador em comandos digitais.

Esta tecnologia tem aplicações em diversas áreas, incluindo realidade virtual (RV), onde o rastreamento de mãos permite interações mais imersivas e naturais. Em dispositivos de VR, o rastreamento de mãos utiliza câmaras e sensores embutidos para capturar os movimentos das mãos em tempo real, analisando a posição e orientação das mãos e reconhecendo gestos específicos. (Realidade Virtual, s.d.)

A crescente adoção do rastreamento de gestos em diversas indústrias, desde o entretenimento e saúde até à educação, reflete o seu potencial em tornar as interações digitais mais acessíveis, inclusivas e eficazes. Por exemplo, um sistema baseado em gestos pode ser particularmente útil para pessoas com dificuldades motoras, permitindo-lhes interagir com interfaces sem a necessidade de periféricos tradicionais.

Além disso, o rastreamento de gestos tem sido explorado para o reconhecimento de linguagem gestual, permitindo uma comunicação mais eficaz entre humanos e máquinas. Por exemplo, algoritmos avançados têm sido desenvolvidos para reconhecer gestos com alta precisão, o que pode ser aplicado no reconhecimento de linguagem gestual. (Olhar Digital, 2019)

E, apesar dos avanços significativos, o rastreamento de mãos ainda enfrenta desafios, como a precisão em condições de iluminação diferentes e a complexidade dos gestos. No entanto, a investigação contínua e o desenvolvimento de novas técnicas, com o uso de IA, têm contribuído

para a melhoria desta tecnologia, tornando-a cada vez mais precisa e acessível para as pessoas. (Ceccon, 2020)

2.2.1 Tecnologias e algoritmos de rastreamento de gestos

O rastreamento de gestos é uma área de pesquisa significativa na interação humano-máquina, permitindo que os sistemas interpretem movimentos humanos como comandos. Diversas tecnologias e algoritmos têm sido desenvolvidos para aprimorar a precisão e a eficiência desse rastreamento.

Uma abordagem comum envolve o uso de sensores de profundidade, como o *Leap Motion* e o *Kinect*, que capturam dados tridimensionais dos movimentos das mãos. Esses dispositivos permitem a detecção precisa de gestos em tempo real, facilitando interações mais naturais usando sistemas computacionais. Por exemplo, o *Leap Motion* tem sido utilizado para reconhecer gestos personalizados da mão em tempo real, empregando aprendizado de métricas e ações para melhorar a precisão do reconhecimento. (Silva, 2017)

Além disso, algoritmos de visão computacional têm sido desenvolvidos para detectar e rastrear gestos sem a necessidade de dispositivos adicionais. Técnicas como a subtração de fundo e o uso de modelos de cor da pele, permitem a segmentação das mãos em imagens, possibilitando o reconhecimento de gestos em ambientes convencionais. (Truyenque, 2005)

Um exemplo é o artigo "*A Survey on Hand Gesture Recognition in Context of Soft Computing*" (Chaudhary, Raheja, & Das, 2010) que apresenta uma análise abrangente sobre os métodos de reconhecimento de gestos baseados em *soft computing*. A pesquisa destaca como técnicas de computação suave, como redes neurais, algoritmos genéticos e lógica difusa, são utilizadas para resolver problemas de incerteza e ambiguidade no reconhecimento de gestos. Por exemplo, algoritmos de aprendizado supervisionado são aplicados para treinar modelos a fim de classificar gestos em diferentes categorias, enquanto a lógica difusa é empregue para lidar com as variações nos movimentos das mãos, garantindo maior adaptabilidade em aplicações. Além disso, o artigo também salienta a importância de considerar o contexto em que os gestos são realizados, como iluminação e oclusões, para melhorar a robustez dos sistemas.

Outra abordagem envolve o uso de redes neurais artificiais para o reconhecimento de gestos estáticos. Modelos de aprendizado profundo, como as Redes Neurais Convulsionais (CNNs), têm demonstrado eficácia na classificação de gestos com alta precisão, aprendendo características discriminativas a partir de grandes conjuntos de dados. (Silva, 2017)

Resumindo, as tecnologias e algoritmos de rastreamento de gestos evoluíram significativamente, incorporando sensores avançados e técnicas de aprendizado de máquina para proporcionar interações mais intuitivas e naturais entre humanos e máquinas.

2.2.2 Rastreamento de gestos em jogos

Em questão de jogos, o rastreamento de gestos tem sido amplamente explorado, proporcionando interações mais naturais e imersivas para os jogadores. Diversos estudos científicos têm investigado a aplicação dessa tecnologia em ambientes de jogos e, por sua vez, a sua eficácia para a acessibilidade para os jogadores.

O estudo "*Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study*" (Masurovsky, et al., 2020) explora o uso de rastreamento de mãos sem necessidade de comandos para as tarefas, como por exemplo de agarrar e colocar objetos em RV imersiva, como podemos ver na Figura 5. Os autores analisaram diferentes elementos de design, como feedback visual e auditivo, com intuito de melhorar a precisão e naturalidade dos movimentos. Além disso, os estudos empíricos realizados indicam que o uso dessa tecnologia pode reduzir barreiras de entrada para novos utilizadores ao eliminarem as necessidades de aprender a mexer com dispositivos externos, tornando, assim, as interações mais intuitivas e acessíveis para um público amplo, incluindo pessoas com limitações motoras.

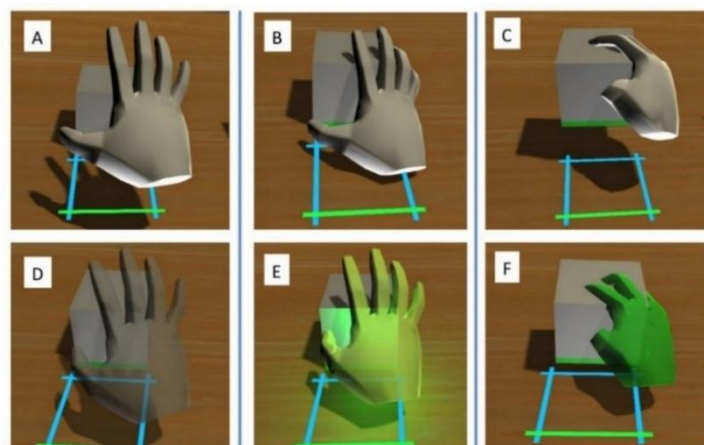


Figura 5 - Exemplo de semitransparência da mão em relação a visualização de objetos

De (Masurovsky, et al., 2020)

Outro estudo, é o "*Influence of Gameplay Duration, Hand Tracking, and Controller Based Control Methods on UX in VR*" (Kojić, et al., 2024) que investiga como diferentes fatores,

incluindo a duração do jogo e os métodos de controle influenciam na experiência do utilizador (UX) em ambientes de VR, a Figura 6 mostra um exemplo de um jogo feito por eles. Neste estudo foi observado que o rastreamento de mãos, quando comparado a comandos tradicionais, proporcionando uma maior sensação de imersão, embora possa exigir um período inicial de adaptação vinda da parte dos jogadores. Além disso, a duração prolongada do jogo demonstrou impacto na precisão e no conforto físico, destacando a necessidade de otimizar algoritmos para reduzir a fadiga dos utilizadores durante sessões de jogo mais longas.



Figura 6 - Exemplo do jogo "Cubism"

De (Kojić, et al., 2024)

Um terceiro estudo, é do artigo “*A Comparative Study of Hand-Gesture Recognition Devices for Games*” (Khalaf1, Alharthi, Alshehri, Dolgov, & Dugas, 2020) que realiza uma análise detalhada de dispositivos populares de reconhecimento de gestos, como o *Leap Motion*, *Intel RealSense* e *Microsoft Kinect*. A pesquisa avalia critérios como precisão, latência e conforto do utilizador em diferentes cenários de jogo. O estudo destaca que, enquanto o *Leap Motion* se sobressai em tarefas de alta precisão devido à sua capacidade de rastrear movimentos finos, o *Kinect* é mais eficaz em cenários de corpo inteiro, como jogos de dança e desportos. Já o *Intel RealSense* apresenta um equilíbrio entre versatilidade e custo-benefício, tornando-o mais adequado para desenvolvedores independentes ou jogos de médio porte.

Estes estudos enfatizam o papel fundamental do rastreamento de gestos como uma tecnologia promissora para melhorar a acessibilidade e a interatividade nos jogos. Eles também apontam desafios, como a necessidade de algoritmos mais robustos e de dispositivos que equilibrem precisão, custo e facilidade de uso, sugerindo caminhos futuros para o avanço da área.

2.2.3 Vantagens e desafios

Concluindo, o rastreamento de gestos oferece uma série de vantagens significativas que tornam essa tecnologia atraente para aplicações interativas e acessíveis. Uma das principais vantagens é a sua capacidade de proporcionar uma interação natural e intuitiva, eliminando a necessidade de controladores físicos. Isso simplifica a UX, especialmente para iniciantes ou pessoas com limitações motoras, permitindo que os gestos das mãos sejam utilizados como uma extensão direta de suas intenções. Além disso, o rastreamento de gestos é amplamente versátil, podendo ser aplicado em jogos, RV e aumentada, aplicações educacionais, aplicações para institutos de recuperação e até mesmo em ambientes industriais, como apontado em estudos que exploraram sua eficácia em tarefas específicas, como o agarrar e posicionar objetos em ambientes virtuais. (Masurovsky, et al., 2020)

No entanto, essa tecnologia também enfrenta desafios consideráveis que precisam ser superados para alcançar o seu verdadeiro potencial. Um dos principais obstáculos é a precisão do rastreamento, que pode ser afetada por fatores como iluminação inadequada, oclusão das mãos ou movimentos rápidos e complexos. Estudos, como o comparativo entre dispositivos de rastreamento (Khalaf¹, Alharthi, Alshehri, Dolgov, & Dugas, 2020) mostraram que, embora dispositivos como *Leap Motion* e *Kinect* ofereçam vantagens específicas, nenhum deles é perfeito para todas as situações. Além disso, a fadiga física decorrente de movimentos repetitivos ou de longa duração acaba por ser um desafio também, principalmente em jogos ou aplicações intensivas, como destacado no estudo sobre duração de jogabilidade e UX. (Kojić, et al., 2024)

Outra questão relevante é o custo e a acessibilidade desses dispositivos. Enquanto alguns, como o *Kinect*, oferecem soluções para aplicações de corpo inteiro, outros, como o *Leap Motion*, são mais acessíveis, mas limitados a rastreamento de mãos. A integração desses dispositivos em plataformas de desenvolvimento também pode exigir habilidades técnicas avançadas, dificultando sua adoção por desenvolvedores iniciantes ou equipes pequenas. Além disso, algoritmos de rastreamento precisam ser constantemente otimizados para lidar com variabilidade entre utilizadores, como tamanhos de mãos, gestos únicos e contextos culturais diversos.

Apesar desses desafios, o rastreamento de gestos continua a ser uma área promissora, como potenciais inexplorados em acessibilidade e interatividade. A combinação de avanços tecnológicos, como algoritmos mais robustos e sensores mais precisos, com estudos que

priorizem a usabilidade e o conforto do utilizador, pode transformar essa tecnologia em um padrão para uma ampla gama de aplicações.

2.3 Comando de Voz

O **comando de voz** é uma tecnologia que permite a interação entre o utilizador e os dispositivos através de instruções verbais. Esta abordagem elimina a necessidade de controladores físicos ou interfaces táteis, utilizando a fala como meio principal de entrada. Com o avanço significativo em reconhecimento de voz e processamento de linguagem natural (PLN), o comando de voz tem-se tornado cada vez mais presente em diversas áreas, como em assistentes virtuais, dispositivos IoT (Internet das Coisas), aplicações empresariais e jogos interativos.

O objetivo principal desta tecnologia é criar experiências de interação mais naturais e acessíveis, especialmente para utilizadores com limitações físicas, visuais ou motoras. Graças ao seu carácter intuitivo, o comando de voz reduz a curva de aprendizagem em muitas aplicações, sendo uma solução poderosa para democratizar o acesso às tecnologias. No entanto, o sucesso dessa abordagem depende de fatores como precisão no reconhecimento, suporte a múltiplos idiomas e robustez para operar em ambientes ruidosos. (Hoy, 2018)

A aplicação de comandos de voz em contextos interativos, como jogos, também se destaca como uma ferramenta promissora para ampliar a acessibilidade e criar experiências imersivas.

2.3.1 Tecnologias e frameworks de reconhecimento de voz

Nos últimos anos, o reconhecimento de voz tornou-se uma das áreas mais promissoras no campo das interações humano-máquina. Tecnologias avançadas e *frameworks* específicos possibilitam o desenvolvimento de aplicações que interpretam comandos de voz com alta precisão. Nesta secção exploraremos algumas das principais soluções disponíveis atualmente, destacando assistentes de voz populares e *frameworks* de desenvolvimento.

2.3.1.1 Assistentes de Voz

Sempre que quando se fala em comando de voz, a primeira coisa que passa pela cabeça das pessoas são os assistentes de voz, já que eles se tornaram o principal exemplo do uso de reconhecimento de voz no dia a dia. Entre os mais conhecidos, destacam-se:

- **Amazon Alexa (Echo Devices):** Desenvolvida pela Amazon, Alexa é uma plataforma de IA que permite interações multimodais, com foco em comandos de voz. Através

dos dispositivos Echo, os utilizadores podem controlar dispositivos domésticos inteligentes, obter informações, ou executar tarefas como criar listas de compras e definir alarmes. Alexa utiliza o **Amazon Lex**, um serviço que combina reconhecimento automático de voz (ASR) e compreensão de linguagem natural (NLU). (Amazon, s.d.)

- **Google Assistant:** Incorporado em telemóveis androids e dispositivos Google Nest, o *Google Assistant* é uma solução robusta que usa algoritmos avançados para compreensão contextual. Baseado no **Google Cloud STT API**, ele suporta múltiplos idiomas e oferece integração com aplicações de terceiros. (Google, s.d.)
- **Apple Siri:** Apenas disponível para dispositivos Apple, Siri utiliza a *framework Speech Framework* do iOS para transformar comandos falados em ações. A Siri também se destaca por integrar informações do utilizador em dispositivos sincronizados. (Apple, s.d.)
- **Microsoft Cortana:** Embora tenha sido gradualmente descontinuado em dispositivos pessoais, Cortana ainda é utilizada em ambientes empresariais, integrando-se com o Microsoft 365. A tecnologia subjacente inclui o **Azure Cognitive Services**, que oferece APIs para reconhecimento de fala e tradução. (Microsoft, s.d.)

2.3.1.2 Frameworks e APIs de Reconhecimento de Voz

Para além dos assistentes de voz, existe *frameworks* e APIs que permitem o desenvolvimento de aplicações que utilizem comandos de vozes, como por exemplo:

- **SpeechRecognition (Biblioteca de Python):** É um pacote *open-source* que permite a integração de diferentes APIs de reconhecimento de voz, incluindo Google, IBM Watson e Microsoft Azure. É amplamente utilizado para protótipos devido à sua simplicidade. (Zhang, et al., 2025)
- **CMU Sphinx:** É um sistema de reconhecimento de voz *open-source* amplamente utilizado em pesquisas e que oferece ferramentas para modelagem acústica e processamento de linguagem natural. (CMUSphinx, 2023)
- **IBM Watson Speech to Text:** É um serviço robusto que permite transcrição e análise de fala em tempo real, frequentemente utilizado em contextos empresariais. Inclui suporte para personalização de modelos de fala. (IBM, s.d.)
- **Kaldi:** É um *toolkit open-souce* usado em pesquisas académicas para reconhecimento de fala, conhecido por sua flexibilidade e suporte para sistemas de reconhecimento híbrido. (Kaldi, s.d.)

2.3.2 Tecnologias e Frameworks de Text-to-Speech

O **TTS** é uma tecnologia que converte texto em áudio falado, sendo amplamente utilizada em assistentes virtuais, softwares de acessibilidade e até na criação de conteúdo. Com o avanço da IA e das redes neurais, as soluções TTS estão cada vez mais naturais e adaptáveis, com suporte a múltiplos idiomas, sotaques e até mesmo entonações emocionais. As principais tecnologias e *frameworks* que utilizam TTS são:

- **Amazon Polly:** Converte texto em fala realista, com suporte a múltiplos idiomas e diferentes vozes, usada na Amazon Alexa. Ela utiliza redes neurais para criar vozes mais naturais e permite personalização, como ajustar o tom e o ritmo da sua fala. Polly é integrada à Amazon Web Services (AWS), tornando-a ideal para aplicações em nuvem. (Amazon, s.d.)
- **Google Cloud Text-to-Speech:** Este serviço o Google Cloud oferece suporte a vozes naturais baseadas em WaveNet, uma tecnologia que utiliza redes neurais profundas para criar uma fala muito realista. Ele também suporta a personalização d voz e integração com outros serviços da Google Cloud. (Google, s.d.)
- **Microsoft Azure Text-to-Speech:** Parte dos serviços da IA do Azure, o TTS da Microsoft suporta mais de 100 vozes em diferentes idiomas. Ele é amplamente utilizado em aplicações corporativas, como *chatbots* e assistentes virtuais, e oferece personalização avançada, incluindo a criação de vozes customizadas. (Microsoft, s.d.)
- **Microsoft Speech SDK:** Este kit de desenvolvimento permite a criação de aplicações que integram TSS de forma acessível, com opções avançadas de personalização e suporte a dispositivos IoT. (Microsoft, s.d.)
- **IBM Watson Text-to-Speech:** Oferece TTS como parte dos seus serviços de IA, com suporte a múltiplos idiomas e vozes naturais. A ferramenta permite integrar facilmente a funcionalidade em aplicações e dispositivos, sendo muito usada em soluções empresariais e educacionais. (IBM, s.d.)

2.3.3 Aplicações que usam Comando de Voz

Para além das tecnologias faladas nas duas seções anteriores, também existem aplicações que incorporam comandos de voz em seus sistemas, o que se têm tornado cada vez mais comum, proporcionando aos utilizadores interações mais naturais e eficientes com essas aplicações. Algumas das áreas mais abrangentes destas aplicações são de:

1. **Automação Residencial:** Estudos feitos mostram o quão pratico e útil aplicação que se podem serem controlados pela voz são em ambientes residências. Um exemplo é estudo de *“Uso da tecnologia de comandos de voz para domótica”* (CARVALHO & ROLAND, 2019) que explorou a utilização de comandos de voz para uso domótica, demonstrando como sistemas baseados em plataformas como o Arduino podem ser implementados para controlar dispositivos domésticos e como este tipo de aplicação facilita a interação dos utilizadores com o ambiente residencial, promovendo maior conforto e acessibilidade.
2. **RV:** O projeto *“Hands-Free VR”* (Fernandez, et al., 2024) explorou a criação de uma aplicação de RV controlada por comandos de voz. Esta aplicação permite que utilizadores interajam com ambientes virtuais sem a necessidade de controlos manuais, aumentando a imersão, acessibilidade e a eficiência em experiências de RV, como por exemplo na conclusão de tarefas.
3. **Controle de Drones:** O projeto *“Evaluating Voice Command Pipelines for Drone Control: From STT and LLM to Direct Classification and Siamese Networks”* (Simões, Rodrigues, Silva, & da Silva, 2024) focou-se em desenvolver e avaliar diferentes pipelines de comando de voz para controlar drones. Este projeto mostrou uma melhor precisão na execução dos comandos de voz, principalmente no mapeamento direto de voz, e também um menor tempo de inferência, tornando-o assim mais eficiente em situações que requeriam respostas rápidas.

Estes três projetos demostraram que aplicação que tem um sistema de comando de voz integrados melhoram a sua acessibilidade, sendo assim benéfico até para indivíduos com mobilidade reduzida, permitindo o controle de dispositivos sem a necessidade de interação física, melhorar a eficiência porque a interação por voz pode acelerar a execução de tarefas, eliminado etapas intermediárias e permitindo comandos mais diretos e a melhorar a UX, já que interfaces de voz proporcionam uma experiência mais natural e intuitiva, reduzindo a curva de aprendizagem e aumentando a satisfação do utilizador.

A integração de comandos de voz em diversas aplicações continua a crescer, impulsionada por avanços tecnológicos de reconhecimento de fala e IA ampliando as possibilidades de interação de humano-máquina.

2.3.4 Comando de voz em jogos

Parecido que tem acontecido nas aplicações, o comando de voz também tem tido uma evolução significativa em jogos eletrônicos, proporcionando interações mais naturais e imersivas entre jogadores e sistemas de jogo. Esta tecnologia permite que os jogadores utilizem comandos verbais para controlar, por exemplo, personagens, navegar em menus e interagir com o ambiente do jogo, enriquecendo a experiência de jogo e tornando-a mais acessível.

2.3.4.1 Exemplo de aplicações práticas do comando de voz em jogos

Comando de voz em jogos não é uma mecânica recente, isso é provado com o jogo *“Lifeline”* de 2003, que foi um jogo feito para PlayStation 2 (PS 2) e foi um dos principais pioneiros a permitir que os jogadores controlassem o protagonista através de comandos de voz, como por exemplo, com comandos como “andar”, “correr”, “parar” e “atirar”, mas embora a sua inovação para a época, o sistema de reconhecimento de voz de *Lifeline* recebeu diversas críticas devido à sua precisão limitada, deixando muitos dos seus jogadores frustrados quando os comandos não eram reconhecidos corretamente. (Mattey & Yuksel, 2018) (Dan, 2023)

Outro jogo foi o *“Seaman”* de 1999, para a consola Sega Dreamcast, que também implementou o sistema de comando de voz para os jogadores interagirem e cuidarem com uma criatura virtual chamada de Seaman. Quando os jogadores conversavam com o Seaman, ele respondia de maneira inteligente e, às vezes, de forma sarcástica. O jogo conseguiu reconhecer uma variedade de palavras e frases, permitindo diálogos que influenciavam no comportamento e no desenvolvimento da criatura. *Seaman* é reconhecido por sua originalidade e pelo uso avançado de reconhecimento de voz para a sua época. A interação verbal profunda proporcionou uma experiência envolvendo, destacando-se como um dos primeiros jogos a explorar a comunicação por voz de forma significativa. (Byford, 2019)

Para além dos exemplos anteriormente referidos, trabalhos recentes têm investigado possibilidades inovadoras de integração do comando de voz em jogos. O estudo *“Let’s Talk Games: An Expert Exploration of Speech Interaction with NPCs”* (Zargham, et al., 2023) analisou o uso de interações por voz para comunicação com NPCs (non-player characters), permitindo diálogos mais dinâmicos e respostas contextuais que potenciam a profundidade narrativa. Os autores concluíram que a integração de comandos de voz pode enriquecer a narrativa e proporcionar experiências de jogo mais imersivas, mas também destacaram desafios

significativos, como a necessidade de um reconhecimento de fala preciso e da interpretação contextual dos comandos dos jogadores.

É ainda enfatizada a importância de um design cuidadoso, de modo a garantir que as interações voz sejam intuitivas e complementem a jogabilidade existente, sem introduzir complexidades desnecessárias. Um bom exemplo de um jogo recente que utiliza interação com NPCs por comando de voz é o jogo **Suck Up!** (Proxima Enterprises Inc, s.d.), como se ilustra na Figura 7.



Figura 7 - Jogo "Suck Up"

De (Proxima Enterprises Inc, s.d.)

Tratasse de um jogo de aventura/gestão em que o jogador interpreta um vampiro e interage com uma pequena comunidade de NPCs recorrendo sobretudo ao comando de voz. As respostas dos NPCs, suportadas por mecanismos de IA, variam conforme o conteúdo, tom e intenção das falas do jogador, criando diálogos dinâmicos que influenciam relações, missões e o desenrolar da narrativa.

2.3.4.2 Conclusão

A integração de comandos de voz em jogos representa um avanço significativo na forma como os jogadores interagem com os mundos virtuais, oferecendo experiências mais imersivas e acessíveis. No entanto a implementação eficaz dessa tecnologia requer a superação de desafios técnicos e de design para garantir que a interação por voz seja intuitiva, responsiva e significativa dentro do contexto do jogo.

2.3.5 Vantagens e desafios

A implementação do comando de voz em aplicações interativas, como jogos, dispositivos IoT e assistentes virtuais, apresenta inúmeras vantagens significativas, mas também impõe desafios técnicos e de design que precisam ser superados.

Entre as suas principais vantagens, destaca-se a acessibilidade, que facilita a utilização de aplicações e de jogos, especialmente por utilizadores com limitações físicas, visuais ou motoras. Essa tecnologia permite uma interação por utilizadores com limitações físicas, visuais ou motoras. Essa tecnologia permite uma interação mais intuitiva, eliminando qualquer barreira imposta por interfaces tradicionais, como teclados ou ecrãs táteis. Além disso, a naturalidade e a imersão proporcionadas pelo uso da fala criam experiências interativas mais próxima da comunicação humana convencional, enriquecendo a narrativa e promovendo um maior envolvimento, especialmente em jogos. Outra vantagem significativa é a redução da curva de aprendizagem, uma vez que o comando de voz se baseia em linguagem natural. Isso facilita a adoção por uma ampla gama de utilizadores, incluindo crianças e idosos, tornando a interação mais inclusiva. Por fim, a eficiência na execução de tarefas também se destaca: comandos diretos, como “ligar a luz”, “atacar inimigo” ou indicar um destino a um GPS, sem precisar, por exemplo, estacionar o carro para digitar o seu destino, eliminando assim as etapas intermediárias, otimizando a interação e aumentando a produtividade dos seus utilizadores.

No entanto, essa tecnologia enfrenta também desafios notáveis. Um dos principais é o reconhecimento de fala em ambientes ruidosos. Em locais com ruído de fundo, como espaços públicos ou residências movimentadas, a precisão de reconhecimento pode ser comprometida, afetando diretamente a usabilidade e a experiência do utilizador. Outro desafio é a diversidade linguística e dialética, embora muitos sistemas suportem múltiplos idiomas, a variação de sotaques, dialetos e expressões regionais continua a ser um obstáculo. Tornar a tecnologia inclusiva para todas as variantes linguísticas é uma tarefa complexa. Além disso, existem limitações técnicas e contextuais, pois a interpretação precisa dos comandos de voz depende de avanços em PLN e de uma compreensão mais sofisticada do contexto da interação, evitando respostas inadequadas ou ações incorretas.

O design e a integração também apresentam desafios: comandos de voz devem principalmente complementar e não substituir as interfaces existentes. Implementações mal planejadas podem gerar frustrações, como comandos não reconhecidos ou interações limitadas, prejudicando a experiência do utilizador. Finalmente, as questões de privacidade e segurança exigem atenção especial. O uso contínuo de microfones para captar comandos levanta preocupações relacionadas à privacidade e sistemas vulneráveis podem ser alvo de comandos maliciosos, demandando robustez em termos de segurança.

Em conclusão, o comando de voz representa uma evolução significativa na interação humano-máquina, trazendo benefícios como acessibilidade, eficiência e imersão em diferentes

contextos. Contudo, para que o seu potencial seja plenamente aproveitado, é crucial superar os desafios técnicos e contextuais. Isso inclui avanços no reconhecimento de fala, melhor adaptação e ambientes diversos e um design centrado no utilizador. Dessa forma, a tecnologia pode continuar a democratizar o acesso às aplicações e promover experiências interativas inovadoras.

3 Tecnologias utilizadas

Para este tipo de projeto, as tecnologias utilizadas acabam por ser um dos fatores mais importantes para o desenvolvimento e as conclusões. Neste capítulo são as ferramentas e tecnologias utilizadas durante o desenvolvimento do projeto, como: Figma, Python, Unity, ChatGPT, entre outras. É detalhado como cada recurso contribuiu para o design, desenvolvimento e integração das funcionalidades de rastreamento e comando de voz, destacando a importância de ferramentas colaborativas e linguagens de programação adequadas ao contexto.

3.1 O Python

O Python é uma linguagem de programação de alto nível, interpretada e amplamente utilizada devido à sua simplicidade e versatilidade. Com uma sintaxe intuitiva e uma vasta variedade de bibliotecas, Python permite o desenvolvimento rápido de aplicações para diversas áreas, incluindo IA, visão computacional, processamento de áudio e comunicação em rede. (Python Software Foundation, 2025)

Neste projeto, o Python foi utilizado para lidar com a detecção de gestos e comandos de voz, além de facilitar a comunicação entre diferentes componentes do sistema.

Para este projeto foram utilizadas as seguintes bibliotecas:

- **OpenCV (Open Computer Vision):** Biblioteca de visão computacional usada para processar imagens e vídeos em tempo real. No projeto, é utilizada para capturar a imagem da câmera do jogador e ajudar na detecção das mãos (Senyayev, Asmorkalov, Sergregory, & Skvark, 2025);
- **Cvzone (Computer vision zone):** Esta biblioteca facilita a utilização do OpenCV, fornecendo funções de alto nível para o rastreamento de mãos (sendo uma junção do OpenCV com o Mediapipe), reconhecimento fácil e manipulação de imagens. Esta biblioteca foi usada para simplificar a detecção das mãos e extrair suas coordenadas (Vizdx LLC, 2024);
- **Speech_recognition:** Esta biblioteca permite a conversão de áudio em texto. No projeto, é responsável por interpretar comandos de voz do jogador, possibilitando a interação com o jogo (Zhang, et al., 2025);

- **gTTS (Google Text-to-Speech):** Converte texto em fala, tornando possível fornecer feedback sonoro ao jogador. A biblioteca gera arquivos de áudio a partir do texto processado (pndurette, 2024);
- **Pygame:** Uma biblioteca voltada para desenvolvimento de jogos, utilizada neste projeto para ajudar a reproduzir os áudios gerados pelo gTTS diretamente pelo código, sem precisar de ficheiros de audio (illume & pygameci, 2024);
- **Socket:** Permite a comunicação entre sistemas via rede. No projeto, é utilizada para estabelecer uma conexão entre Python e Unity, garantindo a transmissão de informações entre o sistema de detecção (Python) e o motor de jogo (Unity) (Python Software Foundation, 2025);
- **Flask:** Serve como servidor Hypertext Transfer Protocol (HTTP) leve que expõe endpoints para coordenação, como foi utilizado neste projeto, entre o Python e o Unity. No projeto é usado para ativar/desativar a partilha de imagem (POST) e para servir o último *frame* em JPEG (GET), permitindo ao Unity pedir a visão da câmara quando necessário (Pallets, 2025).

O uso destas tecnologias permite a criação de uma interface acessível e interativa, onde o jogador pode interagir com o jogo através de gestos e comando de voz, promovendo uma experiência inclusiva e dinâmica.

3.2 O Unity

O Unity é um motor de jogo (*game engine*) amplamente utilizado para desenvolvimento de jogos e aplicações interativas. Criado pela Unity Technologies, ele oferece um ambiente integrado que permite criar experiências e, 2D, 3D AR e VR, com suporte para várias plataformas, como Windows, macOS, consolas, dispositivos móveis e navegadores. (Unity Technologies, 2025)

O Unity funciona baseado numa arquitetura orientada a componentes, onde os *GameObjects* representam os elementos da cena e os Componentes adicionam funcionalidades a esses objetos. O motor utiliza a linguagem C# para programação de scripts, permitindo criar comportamentos personalizados e interações dinâmicas. Além disso, o Unity inclui um sistema de física, motor gráfico otimizado, suporte a animações e ferramentas de renderização avançadas. (Unity Technologies, s.d.)

A interface do Unity é composta por painéis como o *Scene View* (visualização do ambiente do jogo), *Game View* (pré-visualização do jogo em execução), *Inspector* (edição de propriedades

dos objetos) e *Hierarchy* (estrutura hierárquica dos *GameObjects*). O motor também oferece suporte a bibliotecas e pacotes adicionais para melhorar o desenvolvimento dos projetos, como, por exemplo os que foram utilizados neste projeto, *Python Scripting* que ajuda a integrar códigos feitos na linguagem Python nos seus projetos de Unity

No contexto deste projeto, o Unity foi escolhido devido à sua flexibilidade para desenvolver jogos com rastreamento de gestos e comando de voz e a sua compatibilidade com diversas tecnologias de acessibilidade.

3.3 O Microsoft Visual Studio Code

O Microsoft Visual Studio Code (VS Code) é um editor de código-fonte leve e altamente extensível, desenvolvido pela Microsoft. Suporta diversas linguagens de programação e oferece funcionalidades avançadas, como realce de sintaxe, depuração integrada, controle de versões com Git e um vasto ecossistema de extensões. (Microsoft, 2025)

O VS Code foi utilizado neste projeto devido à sua flexibilidade e eficiência no desenvolvimento de aplicações. Algumas das suas principais funcionalidades incluem:

- **Suporte a múltiplas linguagens:** Permite a edição de código em Python, C#, JavaScript (JS) e muitas outras linguagens de programação, sendo ideal para um ambiente de desenvolvimento híbrido, como este projeto que envolveu Python e Unity (C#);
- **Extensões:** Através do Marketplace de extensões, é possível adicionar funcionalidades como suporte aprimorado ao Unity, depuração para Python e integração com GitHub;
- **Integração com Git:** O VS Code possui um sistema de controle de versões integrado, permitindo clonar repositórios, fazer commits e gerir branches diretamente no editor;
- **Depuração e IntelliCode:** O IntelliCode fornece sugestões de código inteligente e auto completar, direcionado para a linguagem de programação C#, Python e JS, enquanto o depurador integrado facilita a identificação e correção de erros (Microsoft, 2024);
- **Tabnine:** O Tabnine é uma extensão baseada em IA que melhora a sugestão de código, oferecendo auto completar mais preciso e eficiente. O Tabnine vai analisando o código que esta a ser desenvolvido e dando sugestões ajustadas para o código em questão. No projeto, ele auxiliou na escrita de código em Python e C#, tornando o desenvolvimento mais rápido e reduzindo erros de sintaxe (Tabnine, s.d.);
- **Terminal embutido:** Facilita a execução de comandos do Git, Python, Unity CLI e outras ferramentas sem precisar alternar entre janelas. Neste projeto, o terminal foi

maioritariamente utilizado para o desenvolvimento do código Python, analisando os dados que estavam a ser gerados por ele e erros cometidos durante a suas execuções. No contexto deste projeto, o VS Code foi utilizado para desenvolver os códigos em Python responsáveis pela deteção das mãos e da voz, desenvolvimento do código em C# para o Unity e bem como a comunicação entre o Python e Unity via sockets. A sua interface intuitiva, combinada com extensões permitiram um fluxo de trabalho mais produtivo e eficiente.

3.4 O Git e GitHub

O Git é um sistema de controlo de versões distribuído que permite gerir e acompanhar alterações em projetos de software de forma eficiente. Ela possibilita a colaboração entre vários desenvolvedores, garantindo que as mudanças sejam registadas se necessário e integradas de maneira organizada. O Git funciona através de comandos que permitem criar repositório, realizar commits (salvar alterações), criar branches (ramificações para desenvolvimento paralelo) e fazer merges (junções de versões ou branches). (GitHub, Inc., 2025)

Enquanto o GitHub é uma plataforma baseada em nuvem que utiliza Git para armazenar, gerir e compartilhar repositórios de código. Além de servir como um repositório remoto, o GitHub oferece ferramentas para colaboração, como pull requests (solicitações de alterações), issues (relatórios de problemas ou melhorias) e GitHub Actions (automatização de processos). (GitHub, Inc., 2025)

No contexto deste projeto, o Git e GitHub foram utilizados para:

- **Controle de versões:** Manter um histórico das alterações feitas no código, facilitando a reversão de mudanças e o rastreamento do progresso.
- **Backup:** Garantir que o código esteja armazenado de forma segura e acessível de qualquer lugar.
- **Gestão de desenvolvimento:** Utilização de branches para testar novas funcionalidade sem comprometer a versão principal do projeto.

A utilização destas ferramentas proporcionou um fluxo de trabalho mais seguro e organizado, garantindo a integridade do código e facilitando futuras melhorias no projeto.

3.5 O GitHub Desktop

O GitHub Desktop é uma aplicação gráfica que facilita a utilização do Git e a gestão de repositórios hospedados no GitHub. Ele oferece uma interface intuitiva que permite aos

utilizadores interagir com o Git sem necessidade de comandos no terminal, tornando o fluxo de trabalho mais acessível e eficiente. (GitHub, Inc, 2025)

No contexto deste projeto, o GitHub Desktop foi utilizado para:

- **Facilitar commits e merges:** A interface gráfica ajuda a visualizar o histórico de alterações, como o que foi alterado no código desde último commit, e gerir diferentes versões do código.
- **Acompanhar o histórico do projeto:** Exibe um log com detalhes das modificações feitas ao longo do tempo, facilitando o rastreamento de mudanças e correções.
- **Gerir repositórios Git de forma visual:** Permite clonar repositórios, criar branches e sincronizar alterações com o GitHub de maneira mais simples.

A escolha do GitHub Desktop para este projeto deve-se à sua praticidade e integração direta com o GitHub, permitindo uma gestão mais eficiente do repositório sem a necessidade de comandos manuais.

3.6 O Figma

O **Figma** é uma ferramenta de design colaborativa baseada em nuvem, amplamente utilizada para criação de interfaces, prototipagem e desenvolvimento de experiências de utilizador (UX/UI). Ele permite que designers e desenvolvedores trabalhem em conjunto em tempo real, facilitando a criação e validação de conceitos visuais. (Figma, Inc, s.d.)

No contexto deste projeto, o Figma foi utilizado para:

- **Wireframes:** Criar os primeiros esboços da interface, definindo a estrutura e disposição dos elementos antes da implementação.
- **Mockups:** Refinar o design visual, aplicando estilos, cores e tipografia para representar com maior precisão a aparência final da aplicação.
- **Protótipo não funcional:** Desenvolver um modelo interativo da interface para demonstrar como ia ser a experiência do utilizador, sem necessidade de programação. Permitindo assim, testar a navegação e o fluxo de interação antes do desenvolvimento.

A escolha do Figma deveu-se à sua facilidade de uso, a possibilidade de partilhar com outras pessoas para testes, o que acelerou o processo de design e permitiu validar as decisões visuais de forma eficiente.

3.7 O Google Forms

O **Google Forms** é uma ferramenta online que permite a criação de formulários e questionários de forma intuitiva e acessível. Ele é amplamente utilizado para recolher dados, realizar pesquisas e obter feedback de utilizador. (Google, s.d.)

Neste projeto, o Google Forms foi utilizado para a criação de **formulários** que foram entregues juntamente com o protótipo funcional para os testes com utilizadores, e, por sequencia, recolher as opiniões e dados relevantes dos mesmos, como dados sobre usabilidade, acessibilidade e eficácia da interação do projeto. O Google Forms permite também visualizar respostas em tempo real e exportar os dados para uma análise pos-conclusa dos testes.

A escolha do Google Forms deve-se à sua facilidade de uso e de envio para os utilizadores testes.

3.8 O Chat GPT

O **Chat GPT** é um **modelo** de IA baseado em linguagem natural, desenvolvido para auxiliar na geração de texto, pesquisa e suporte técnico. Ele poder ser utilizado para responder perguntas, auxiliar na escrita e até mesmo ajudar na resolução de problemas de programação. (OpenAI, 2022)

Neste projeto o Chat GPT foi utilizado para:

- **Pesquisa de artigos científicos:** Apoiou na busca de artigos científico sobre rastreamento e comando de voz, tanto em aplicações como em jogos, como mencionado na seção 1.7.2.
- **Correção e formalização do relatório:** Auxiliou na revisão de texto, garantindo uma estrutura clara e coesa, além da correção de erros gramaticais e aprimoramento da formalidade.
- **Resolução de erros de código:** Ajudou a identificar e a sugerir soluções para problemas encontrados durante o desenvolvimento.
- **Otimização de código:** Propôs algumas melhorias na estrutura e eficiência do código em C# e Python, incluindo boas práticas e otimização do protótipo.

O uso do Chat GTP permitiu acelerar o processo de desenvolvimento, facilitar a pesquisa de informações relevantes e aprimorar a qualidade técnica e formal do relatório.

3.9 A Adobe Illustrator e Photoshop

O Adobe Illustrator é uma ferramenta de design gráfico vetorial amplamente utilizada para a criação de ilustrações, logótipos, ícones e elementos gráficos que necessitam de alta qualidade e escalabilidade. O Illustrator trabalha com gráficos vetoriais, o que permite que as imagens criadas possam ser redimensionadas sem perda de qualidade, sendo ideal para a criação de elementos visuais limpos e precisos. (Adobe, 2025)

Já o Adobe Photoshop é uma aplicação focada na edição e manipulação de imagens rasterizadas (baseadas em pixels), sendo uma referência no mercado para retoques fotográficos, criação de composições visuais e ajustes de imagens, como correções de cor, recortes, aplicação de filtros e tratamentos mais detalhados que exigem a manipulação direta dos pixels. (Adobe, 2025)

No âmbito deste projeto, o Adobe Illustrator foi utilizado principalmente para a criação de ícones personalizados que compõem a interface do protótipo, garantindo um aspeto visual consistente e adaptado às necessidades da aplicação. Alguns exemplos incluem os ícones para as regras e instruções presentes antes dos minijogos.

Por outro lado, o Adobe Photoshop foi utilizado para realizar ajustes em imagens que o Illustrator, por trabalhar apenas com vetores, não conseguia tratar de forma eficiente. Foram efetuadas correções como, por exemplo, remoção de fundo das imagens.

A utilização conjunta destas duas ferramentas permitiu obter uma identidade visual mais polida e adaptada ao projeto, contribuindo para uma melhor experiência de utilizador.

3.10 O Adobe Color

O **Adobe Color** é uma ferramenta online desenvolvida pela Adobe que permite criar, explorar e guardar paletas de cores com base em diferentes regras de harmonia cromática, como análogas, complementares, tríades, entre outras. Através de círculo cromático interativo, o utilizador pode ajustar as cores manualmente ou gerar combinações automaticamente. A ferramenta também oferece funcionalidades como a extração de paletas a partir de imagens e a possibilidade de verificar a acessibilidade das combinações para diferentes tipos de daltonismo. (Adobe, 2025)

No âmbito desta tese, o Adobe Color foi utilizado para auxiliar nas definições das paletas de cores da interface do jogo. A ferramenta foi particularmente útil na escolha de cores que se destacassem entre si, garantindo um contraste visual eficaz entre os elementos, o que é essencial para acessibilidade e clareza visual, especialmente num jogo que visa incluir jogadores

com diferentes necessidades. As combinações criadas serviram de base para garantir uma identidade visual coerente e funcional ao longo do desenvolvimento.

3.11 O Microsoft Clipchamp

O Microsoft Clipchamp é um editor de vídeo intuitivo, baseado em interface gráfica de arrastar e largar com uma linha tem portal (*timeline*) que simplifica operações comuns de pós-produção. Permite importar vários formatos de ficheiro, cortar e unir clips, ajustar a resolução e taxa de frames, aplicar transições, sobrepor texto e legendas, e trabalhar o áudio (volume, fades, corte). Oferece também ferramentas rápidas de exportação com opções de compressão e formatos adaptados para web e redes sociais, facilitando a criação de vídeos curtos, tutoriais e activos multimédia otimizados para diferentes plataformas. (Microsoft, 2025)

No projeto, o **Clipchamp** foi utilizado, principalmente, para editar os vídeos que foram convertidos em GIFs usados nos tutoriais. A edição desses vídeos foi principalmente, cortar cenas irrelevantes, reorganização em pequenas sequências e ajuste de resolução para obter ficheiros leves e fluídos, fundamentais para os tutoriais.

Além disso, o **Clipchamp** foi utilizado no processamento dos efeitos sonoros do projeto: foram importados ficheiros de áudio, recortados e eliminadas as partes desnecessárias, ficando assim com as partes sonoras mais relevantes, melhorando assim o feedback auditivo.

4 Design

Neste capítulo, será abordado o processo de desenvolvimento do design do projeto, desde a criação dos Wireframes iniciais até à definição da paleta de cores, até à elaboração dos Mockups finais que representam a interface e a experiência visual pretendida.

4.1 Wireframe

Para o desenvolvimento dos Wireframes, foi utilizado a ferramenta Figma, conforme mencionado na secção 3 Tecnologias utilizadas. Estes Wireframes representaram a primeira versão visual do protótipo, servindo como base para explorar a estrutura das interfaces dos menus e dos minijogos. Nesta fase, não foram aplicadas cores, uma vez que o objetivo principal era definir o posicionamento dos elementos em cada interface da aplicação. Na Figura 8 e Figura 9 podemos ver o resultado dos Wireframes criados para os protótipos de rastreamento de gestos e comando voz, respetivamente.

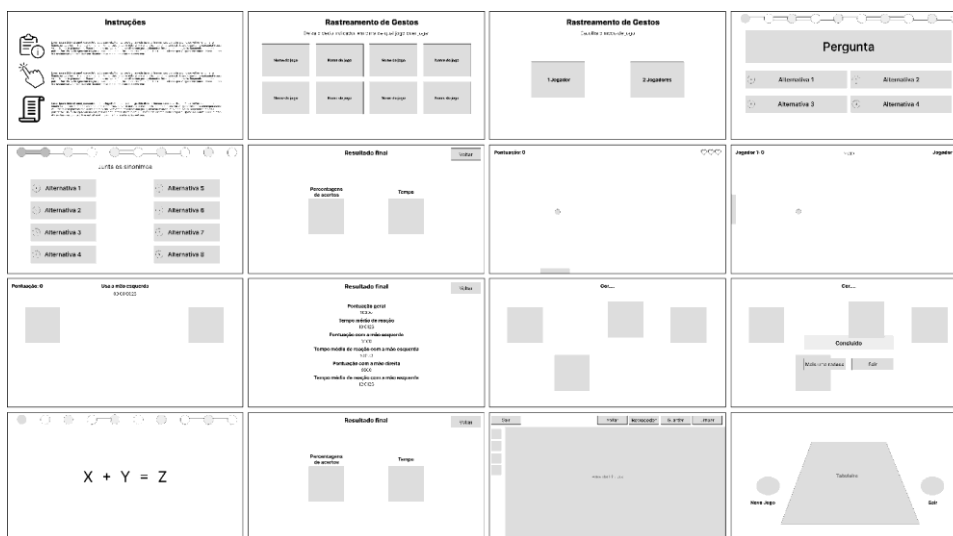


Figura 8 - Wireframe do Rastreamento de Gestos

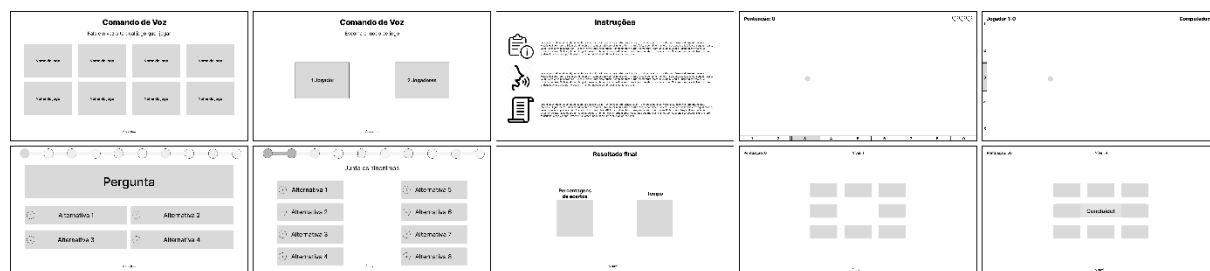


Figura 9 - Wireframe do comando de voz

4.2 Paleta de cores

Na definição da paleta de cores para este projeto, optou-se por utilizar cores visualmente distintas entre si, com o objetivo de facilitar a percepção e a identificação dos elementos do jogo por parte do utilizador. Esta abordagem visa, sobretudo, melhorar a clareza visual e a acessibilidade, particularmente nos minijogos que envolvem rastreamento de gestos, onde é fundamental que o utilizador consiga distinguir com facilidade quais os elementos com os quais pode interagir.

Desta forma, foram selecionadas duas paletas de cores: uma paleta principal, que abrange as cores predominantes na interface e nos elementos centrais dos jogos. E também, uma paleta secundária, utilizada como complemento visual, para reforçar a distinção entre elementos ou contextos específicos.

A Figura 10 apresenta a paleta principal, composta pelas seguintes cores:

- **Azul Brilhante (#007BFF)** – Aplicada como cor de fundo nos vários cenários do jogo.
- **Laranja Forte (#FF7300)** – Utilizada principalmente nos botões interativos, quer para comandos por gestos, quer por voz.
- **Cinza Médio (#A0A0A0)** – Aplicada no destaque de textos informativos, como instruções ou eventos que ocorrem durante os minijogos.
- **Verde Lima (#32CD32)** – Utilizada para representar progresso ou feedback positivo, sinalizando conquistas ou ações bem-sucedidas do utilizador.
- **Dourado (#FFD700)** – Reservada para elementos de destaque, como por exemplo a pergunta principal num quiz, de modo a chamar a atenção do utilizador.



Figura 10 - Paleta de cores principal

E na Figura 11 podemos ver a paleta secundária, que é composta pelas seguintes cores:

- **Aqua/Ciano (#00FFFF)** – Esta cor foi é uma cor de destaque, que mostra algumas escolhas do utilizador, ou até o próprio utilizador.
- **Indigo Azul-Escuro (#1E3A8A)** – Esta cor, igualmente como o Azul Brilhante na paleta principal, também é usado como cor de fundo, como por exemplo nas instruções ou em elementos que precisaram de destaque.

- **Verde Néon (#39FF14)** – Esta cor foi usada como destaque, como nas linhas que formaram as linhas no esqueleto da mão do utilizador
- **Azul Violeta (#8A2BE2)** – Esta cor também foi usada como destaque no esqueleto da mão do jogador, mas neste caso para destacar o ponto da mão que tem influência no jogo em questão
- **Vermelho (#FF0000)** – Igualmente as cores anteriores, esta cor também foi usada como destaque, como os pontos das mãos do utilizar (chamadas de *landmark*) ou até em ações erradas feitas pelo utilizador.

Nota: O esqueleto da mão pode ser encontrado na seção 5.1.2.1 Rastreamento de Gestos



Figura 11 - Paleta de cores secundaria

4.3 MockUp

Após a criação dos Wireframes, WireFlow e na escolha das patetas de cores, foram desenvolvidos os Mockups utilizando também a ferramenta Figma. Estes Mockups representaram uma versão visual, mas próxima do produto final, com a aplicação de cores, ícones e outros elementos gráficos. O objetivo desta fase foi traduzir a estrutura funcional definida anteriormente numa interface visualmente apelativa e consistente.

Para além de definir o aspeto visual da aplicação, os Mockups também serviram como base para validar aspetos relacionados com a legibilidade, contraste e hierarquia da informação. Esta etapa foi fundamental para garantir que os elementos gráficos não apenas fossem esteticamente agradáveis, mas também funcionais e acessíveis, contribuindo para uma melhoria na UX.

4.3.1 O Rastreamento de Mãos

Esta secção apresenta os *Mockups* finais do protótipo dos jogos com sistema de rastreamento de mãos.

Na Figura 12 podemos ver a interface do Menu e a página “Sobre” do protótipo de rastreamento de mãos.



Figura 12 - Mockup de Rastreamento de Mãos: Menu

Na Figura 13 pode a interface do jogo das cores, apresentado por duas páginas de instruções (uma a explicar o funcionamento do minijogo e outra com um GIF demonstrativo), uma de escolha de modo de jogo e o jogo em si.

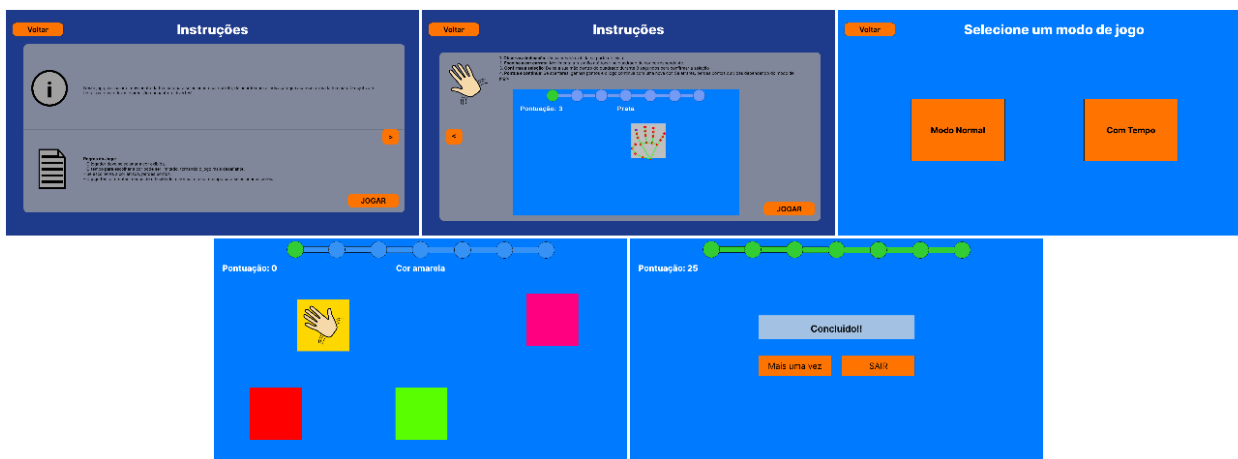


Figura 13 - Mockup de Rastreamento de Mãos: Jogo das Cores

Na Figura 14 podemos ver a interface do jogo de perguntas (quiz), composta por vários elementos, desde as instruções iniciais, a escolha do quiz, as duas formas de perguntas (alternativas e correspondência) e também a indicação de como é mostrado quando o jogador acerta ou erra a resposta.



Figura 14 - Mockup de Rastreamento de Mãos: Quis

Na Figura 15 temos a interface do minijogo do pong solo, nesta interface já podemos ver três páginas de instruções, as duas primeiras são parecidas com já mencionado anteriormente e a terceira explica um pouco as habilidades especiais que podem acontecer durante o jogo.



Figura 15 - Mockup de Rastreamento de Mãos: Pong Solo

Na Figura 16 podemos ver a interface do minijogo do pong clássico, que é muito semelhante ao pong solo, adicionado só a possibilidade de escolha de modo de jogo (jogador contra jogador ou jogador contra computador)

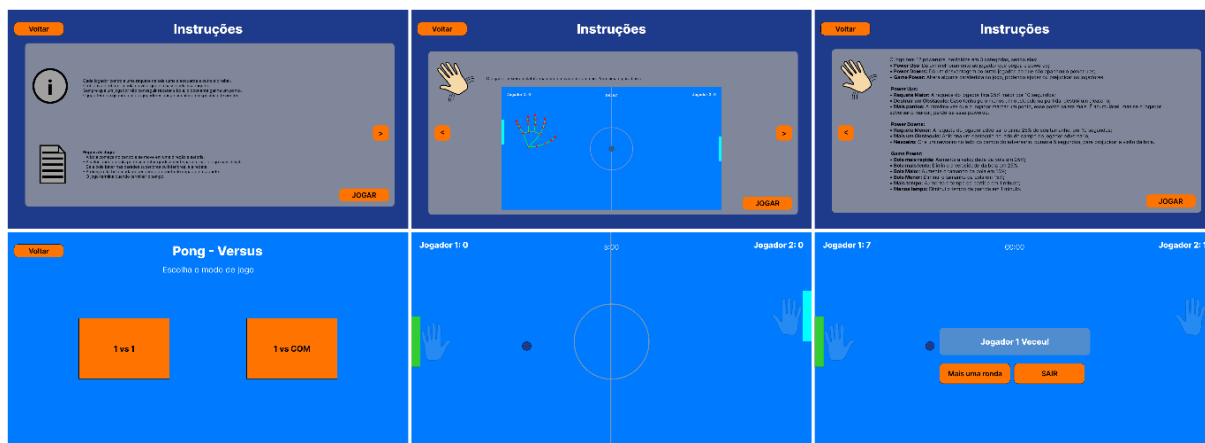


Figura 16 - Mockup de Rastreamento de Mãos: Pong Classico

Na Figura 17 temos a interface do jogo da reação, onde podemos ver uma interface simples, mas que no final mostra ao utilizador informações sobre o tempo médio de demora e a sua pontuação (total e de cada mão),

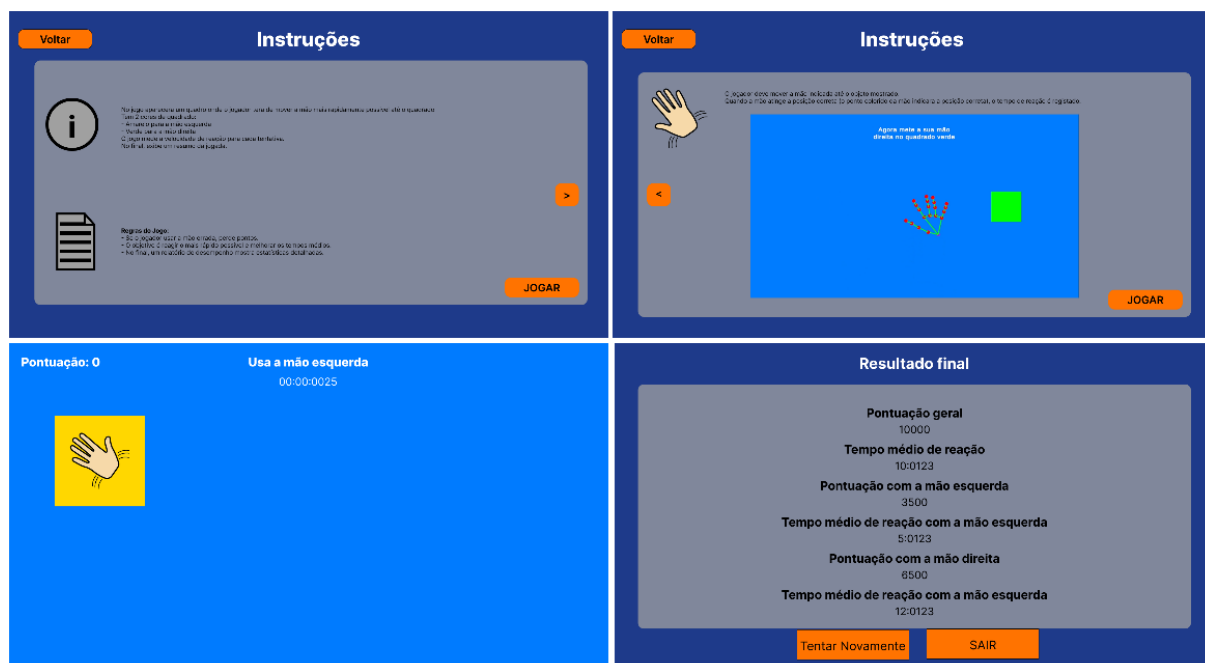


Figura 17 - Mockup de Rastreamento de Mãos: Jogo de Reação

Na Figura 18 podemos ver a interface do minijogo das perguntas de matemática, a interface é muito semelhante já apresentada na interface no quiz.

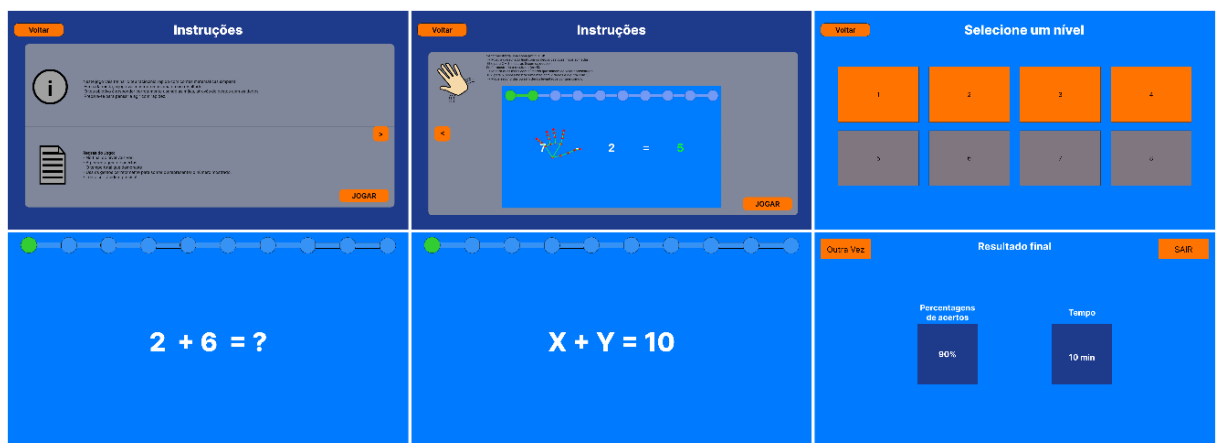


Figura 18 - Mockup de Rastreamento de Mãos: Jogo da Matemática

Na Figura 19, podemos ver a interface do minijogo de pintura (paint), onde pode-se ver uma lista de escolhas para desenhos e também uma área em branco onde o utilizador pode pintar livremente ou um desenho pré-selecionado.

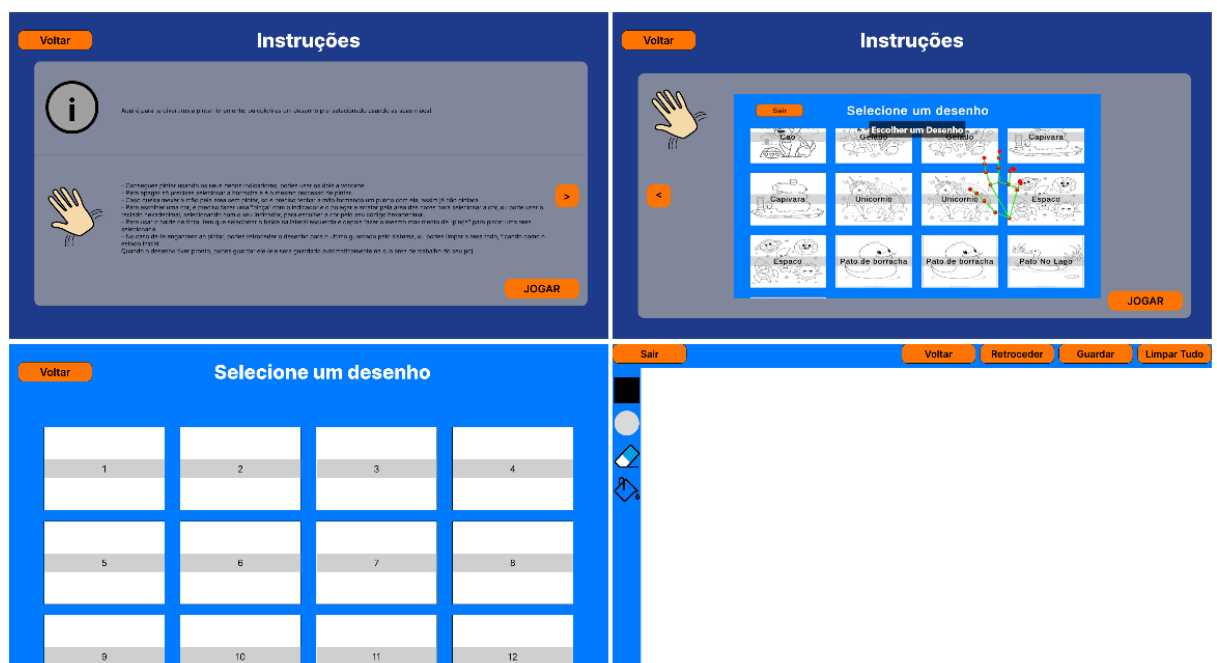


Figura 19 - Mockup de Rastreamento de Mãos: Paint

4.3.2 O Comando de voz

Esta secção apresenta os Mockups finais do protótipo dos jogos com sistema de comandos de voz.

Na Figura 20 podemos ver a interface do Menu e do “Sobre” do protótipo do comando de voz, que visualmente é muito semelhante ao do rastreamento de mãos.



Figura 20 - Mockup de comando de voz: Menu

Na Figura 21, podemos ver a interface do minijogo do quiz, que também é muito semelhante ao do rastreamento de mãos, só ajustada para o comando de voz, como as instruções que acabam por só ocupar uma página, pois não houve a necessidade de criar um GIF demonstrativo, adicionado assim só os comandos importantes que o utilizador precisaria dizer.



Figura 21 - Mockup de comando de voz: Quis

Na Figura 22 e na Figura 23 podemos ver as interfaces dos dois minijogos do pong (pong solo e pong clássico), que, diferentemente do protótipo de rastreamento de mãos, aqui houve a necessidade de colocar lugares pré-definidos e numerados para que utilizador dizendo apenas o número do lugar, possa mover a raquete para esse lugar.

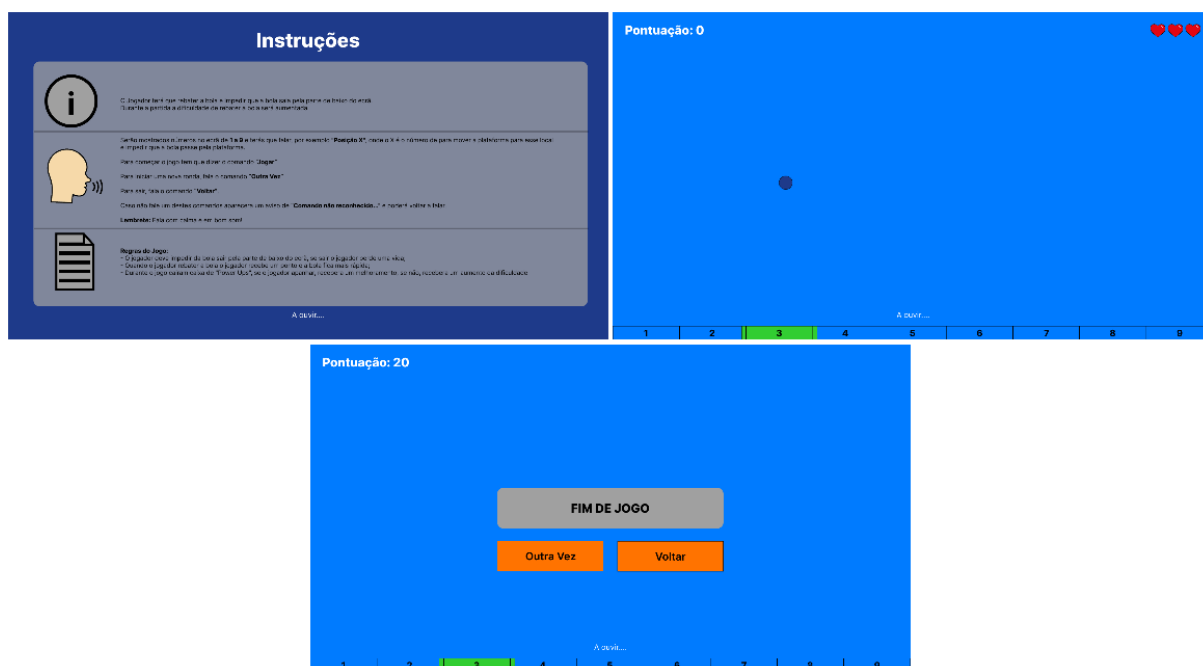


Figura 22 - Mockup de comando de voz: Pong Solo

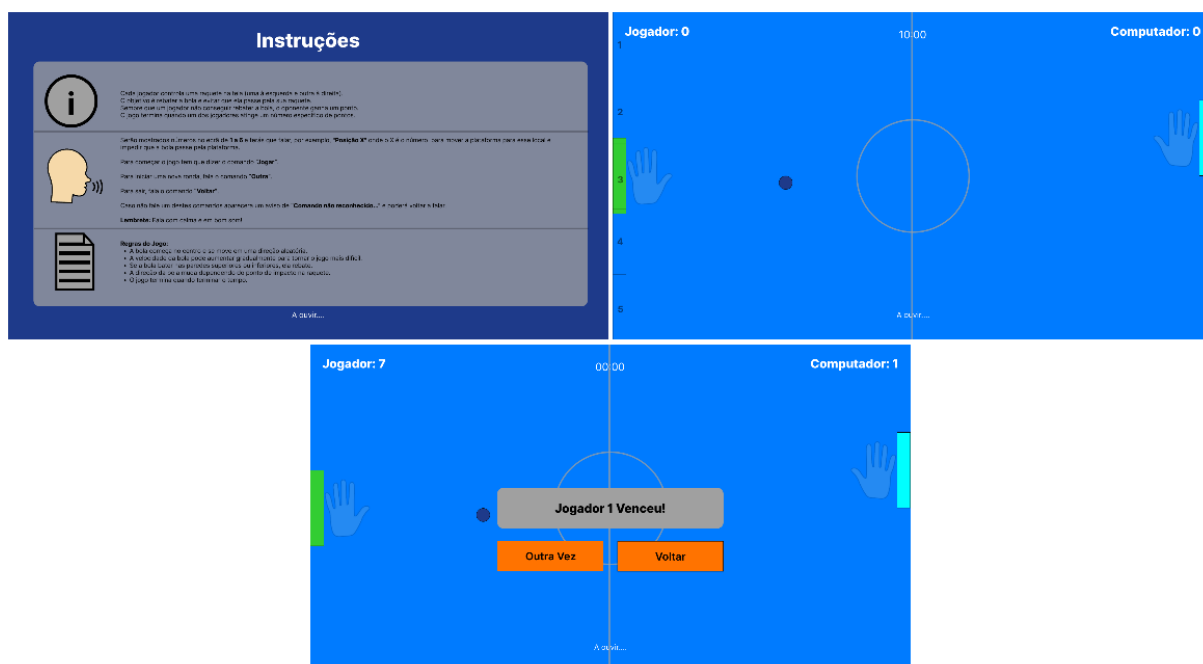


Figura 23 - Mockup de comando de voz: Pong Classico

Na Figura 24 temos a interface do minijogo da memória. É uma interface simples, mas mostra como é indicado as palavras disponíveis, qual palavra tem de se dizer (em amarelo), se a palavra esta correta (em verde) ou se a palavra não esta correta (em vermelho).

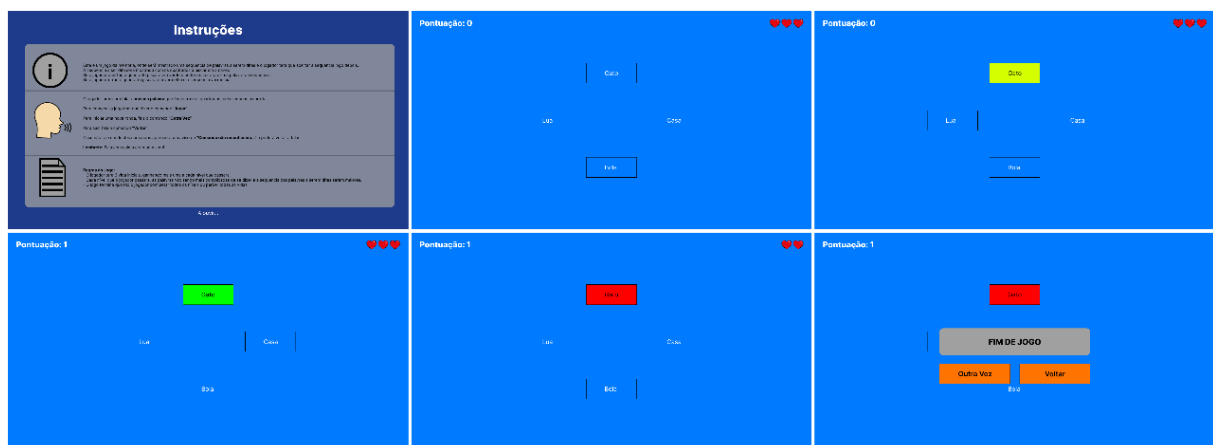


Figura 24 - Mockup de comando de voz: Jogo da Memória

5 Desenvolvimento do protótipo

No desenvolvimento deste trabalho, foi implementada uma prova-de-conceito que integra os mecanismos de rastreamento de mãos e reconhecimento de comandos de voz para controlar minijogos acessíveis. Esta secção descreve as configurações do ambiente (Python e Unity), as decisões técnicas e implementações dos módulos de rastreamento de mãos e de comandos de voz e dificuldades encontradas, justificando as escolhas e avaliando o seu impacto na usabilidade e acessibilidade do sistema.

5.1 Configurações

Nesta secção descrevem-se as configurações dos ambientes de desenvolvimento (Python e Unity) aplicadas para suportar o progresso do projeto de forma eficaz.

5.1.1 Python

Inicialmente configurou-se o sistema em Python. Esta linguagem foi escolhida por oferecer um ecossistema maduro de bibliotecas para visão computacional e áudio (por exemplo, OpenCV, MediaPipe, cvzone, NumPy e bindings para serviços cloud), boa documentação e facilidade para prototipagem rápida e testes experimentais. A integração com Unity foi feita através de uma ponte de comunicação utilizando socket e JSON, permitindo que Python trate a captura e processamento especializado, enquanto o motor gráfico lida com a lógica e apresentação. Estas características reduziram o risco técnico e aceleraram o ciclo de iteração, sem comprometer a capacidade de escalar ou substituir componentes por alternativas mais performance no futuro. No excerto de Código 1 podemos observar um exemplo como o socket e JSON foram utilizados para criar a ponte de comunicação entre o Python e Unity:

```

import socket
import json
# Parametros
address = "127.0.0.1"
port = 5052
# Comunicação (com unity) - UDP
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverAddressPort = (address, port)
# exemplo de um objeto:
hand = {
    "lm": [0, 1, 2, 3, 4, 5],
    "type": "left"
}
message = json.dumps(hand) # converter o objeto em uma string JSON
try:
    sock.sendto(message.encode("utf-8"), serverAddressPort)
except Exception:
    print("[Python] Erro ao enviar UDP:")
sock.close() # encerrar a comunicação do socket

```

Código 1 - Exemplo de configuração do *socket* pelo Python

5.1.1.1 Rastreamento de Gestos

Para o rastreamento de gestos foram utilizadas bibliotecas como o OpenCV, responsável pelo acesso à câmara e pelo processamento de imagem, e o cvzone, que funcionou como camada de suporte sobre funcionalidades de visão computacional providas pelo MediaPipe e pelo OpenCV. O MediaPipe fornece os modelos e a infraestrutura de detecção de mãos (*landmarks* e inferência) como dos representados na Figura 25, enquanto o cvzone encapsula essas funcionalidades em utilitários de nível mais alto, por exemplo: desenho de *landmarks* e normalização de dados, o que acelerou a implementação inicial do protótipo e reduziu a complexidade do código de integração. Pode imaginar-se o MediaPipe como uma biblioteca que disponibiliza vários módulos de detecção (mão, corpo, rosto, etc.) e o cvzone como um conjunto de “*helpers*” que facilita o acesso e a interpretação desses resultados.

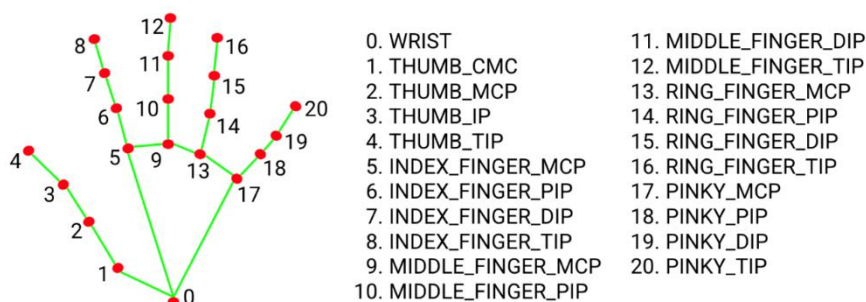


Figura 25 - *Landmarks* do mediapipe

(Google, 2025)

É importante notar que, embora o cvzone facilite o desenvolvimento de protótipos, projetos mais complexos podem exigir controle mais fino sobre parâmetros e inferência. Nesses casos, a utilização direta do MediaPipe ou a customização dos modelos é recomendada.

O MediaPipe fornece as posições normalizadas dos *landmarks* (coordenadas x, y, z) e a classificação de *handedness* (mão esquerda/direita). O cvzone aplica pós-processamento a esses dados para os tornar imediatamente utilizáveis na aplicação: converte as coordenadas normalizadas em pixels, organiza os pontos numa lista denominada de *lmList* e calcula valores derivados, como a caixa delimitadora (*bbox*) e o centro da mão (*center*). A indicação do tipo de mão (*type*: “Right”/“Left”) resulta da classificação de *handedness* do MediaPipe e é exposta pelo cvzone. Nos Código 2 e Código 3 apresenta-se, respetivamente, um exemplo de configuração com cvzone e um exemplo equivalente apenas com MediaPipe. A comparação ilustra como cvzone reduz a quantidade de linhas de código, permitindo focar mais cedo a lógica do jogo. A Figura 26 ilustra a diferença visual entre as duas abordagens.

```
# com cvzone
from cvzone.HandTrackingModule import HandDetector
import cv2
cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=2) # usa MediaPipe por baixo
try:
    while True:
        # Obter o quadro da webcam
        success, img = cap.read()
        if not success:
            break
        img = cv2.resize(img, (1280, 720))
        img = cv2.flip(img, 1) # Inverter a webcam (espelhar horizontalmente)
        hands, img = detector.findHands(img) # desenha landmarks automaticamente
        if hands:
            # Obter a lista de pontos de referência
            for hand in hands:
                print(hand) # dicionário com 'lmList', 'bbox', 'center', 'type', ...
                # exemplo de output = {'lmList': [[235, 330, 0]...], 'bbox': (202, 132,
150, 198), 'center': (277, 231), 'type': 'Right'}
                lmList = hand['lmList'] # [(x_pixel, y_pixel, z), ...]
                bbox = hand['bbox'] # (x, y, w, h)
                handType = hand['type'] # 'Right' / 'Left'
                cx, cy = hand["center"] # coordenadas do ponto central da mão
                # Desenha um círculo no centro da mão
                cv2.circle(img, (cx, cy), 10, (0, 255, 255), cv2.FILLED)
            # Exibir imagem
            cv2.imshow("Image", img)
finally:
    cap.release()
    cv2.destroyAllWindows()
```

Código 2 - Configuração do CVZone (em Python)


```

# Exemplo com MediaPipe (sem cvzone)
import cv2
import mediapipe as mp
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)
# Configuração do MediaPipe Hands
with mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=2,
    min_detection_confidence=0.6,
    min_tracking_confidence=0.5
) as hands:
    try:
        while True:
            success, img = cap.read()
            if not success:
                break

            # Resize e espelho para ficar semelhante ao comportamento do cvzone
            img = cv2.resize(img, (1280, 720))
            img = cv2.flip(img, 1)
            img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            results = hands.process(img_rgb)

            if results.multi_hand_landmarks:
                # results.multi_hand_landmarks e results.multi_handedness têm a mesma
                # ordem
                for hand_landmarks, handedness in zip(results.multi_hand_landmarks,
                                                       results.multi_handedness):
                    h, w, _ = img.shape
                    # Constrói lmList em píxeis (x_pixel, y_pixel, z_scaled)
                    lmList = []
                    for lm in hand_landmarks.landmark:
                        x_px = int(lm.x * w)
                        y_px = int(lm.y * h)
                        # lm.z é relativo – aqui escalamos para um valor inteiro similar
                        # ao exemplo (opcional)
                        z_scaled = int(lm.z * 100) # podes ajustar o factor se quiseres
                        # outro range

                    lmList.append([x_px, y_px, z_scaled])
                    # Calcula bbox (x, y, w, h) a partir dos landmarks
                    x_coords = [p[0] for p in lmList]
                    y_coords = [p[1] for p in lmList]
                    x_min, x_max = min(x_coords), max(x_coords)
                    y_min, y_max = min(y_coords), max(y_coords)

```

```

bbox = (x_min, y_min, x_max - x_min, y_max - y_min)
# Centro da bbox
center = ((x_min + x_max) // 2, (y_min + y_max) // 2)
# Tipo da mão (Right / Left) recebido do handedness
hand_type = handedness.classification[0].label # 'Right' ou 'Left'
# Monta dicionário semelhante ao do cvzone
hand_dict = {
    'lmList': lmList,
    'bbox': bbox,
    'center': center,
    'type': hand_type
}
# Mostra o dicionário (ou guarda/loga)
print(hand_dict)
# exemplo de output: {'lmList': [[399, 518, 0], [464, 495, -2],
[522, 463, -3], [564, 432, -4], [591, 400, -6], [467, 368, -2], 487..., 'bbox': (266,
214, 325, 304), 'center': (428, 366), 'type': 'Left']}
# Desenhar landmarks e conexões (semelhante ao cvzone)
mp_drawing.draw_landmarks(
    img, hand_landmarks, mp_hands.HAND_CONNECTIONS,
    mp_drawing.DrawingSpec(color=(0, 255, 0), thickness=2,
circle_radius=3),
    mp_drawing.DrawingSpec(color=(0, 128, 255), thickness=2)
)
# Desenha bbox e centro
x, y, w_box, h_box = bbox
cv2.rectangle(img, (x, y), (x + w_box, y + h_box), (255, 0, 255),
2)

cx, cy = center
cv2.circle(img, (cx, cy), 8, (0, 255, 255), cv2.FILLED)
# Mostrar frame
cv2.imshow("MediaPipe Hands", img)
finally:
    cap.release()
    cv2.destroyAllWindows()

```

Código 3 - Configurações pelo MediaPipe (em Python)

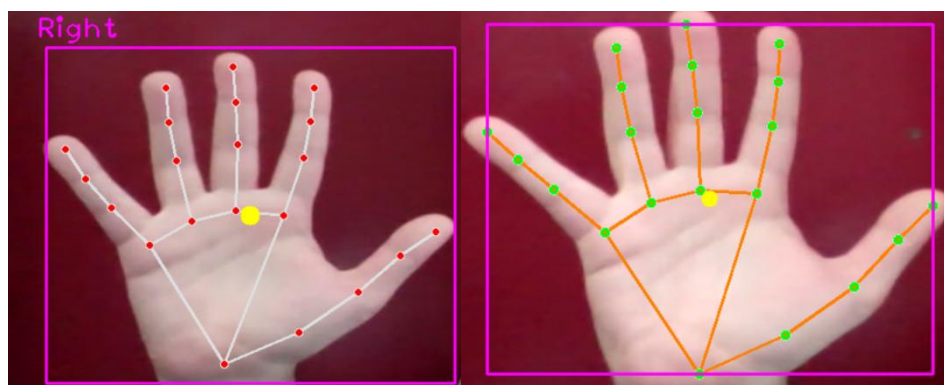


Figura 26 - Comparação entre a detecção da mão utilizando cvzone e não utilizando, respectivamente

Durante o desenvolvimento do protótipo surgiu a necessidade de integrar no Unity a visualização das imagens geradas pelo OpenCV, de modo a evitar a existência de uma janela

secundária e proporcionar ao utilizador um feedback visual direto sobre a área de deteção da mão. Para esse fim foi implementada uma ponte HTTP simples com Flask, onde o Unity envia um pedido indicando que pretende aceder à visão da câmara e módulo Python responde fornecendo o último *frame* processado codificado em JPEG. Esta abordagem (Unity -> Python -> Unity) permite que o utilizador visualize em tempo-real a imagem com os *landmarks* e *bounding boxes* desenhados pelo OpenCV, facilitando a validação do sistema (por parte do utilizador e do avaliador) e a resolução de possíveis problemas, como por exemplo: verificar se a mão está a ser detetada corretamente, sem recorrer a janelas externas ao do jogo.

5.1.1.2 Speech-to-Text

Para o reconhecimento de voz utilizou-se a biblioteca `speech_recognition` (Python) com o backend Google speech (Web Speech API). Esta opção deve-se à facilidade de integração multiplataforma e às funções de pré-processamento (redução de ruído, thresholds), bem como ao suporte a vários serviços de reconhecimento, o que permitiu prototipar rapidamente para interações em tempo real. O uso da Google Speech via esta biblioteca é útil para comandos curtos e vocabulários limitados de jogos mais simples, principalmente com um suporte a múltiplas línguas e facilidade de integração. No excerto de Código 4, podemos ver um exemplo de como foi configurado o STT.

```
import speech_recognition as sr # Audio para texto
import time

# Crie um objeto de reconhecimento de fala
recognizer = sr.Recognizer()

# Ajustes de sensibilidade (opcional, pode experimentar com os valores)
recognizer.energy_threshold = 399 # mais baixo = mais sensível
recognizer.pause_threshold = 0.95 # termina mais rápido a gravação
# Ajustar o ruído (recomendado fazer uma vez)
with sr.Microphone() as source:
    print("Calibrando microfone para o ruído ambiente.... aguarde")
    recognizer.adjust_for_ambient_noise(source, duration=2)
    print("Pronto! Microfone calibrado!")
```

```

# Função callback chamada automaticamente ao reconhecer áudio
def callback(recognizer, audio):
    try:
        text = recognizer.recognize_google(audio, language="pt-PT")
        print(f"Reconhecido: {text}")

    except sr.UnknownValueError:
        print("Não entendi o que foi dito.")
    except sr.RequestError as e:
        print(f"Erro ao obter resultado: {e}")

# Inicia escuta contínua em background
stop_listening = recognizer.listen_in_background(sr.Microphone(), callback)
print("Microfone aberto. Pressiona Ctrl+C para sair.")
# Loop principal (mantém o programa vivo)
try:
    while True:
        time.sleep(0.1)
except KeyboardInterrupt:
    print("Programa encerrado pelo utilizador")
    stop_listening(wait_for_stop=False)

```

Código 4 - Exemplo de configuração para STT utilizando API do Google (em python)

5.1.1.3 Text-to-Speech

Para complementar o comando de voz, foi integrado um sistema de TTS, responsável por converter texto em áudio de forma a fornecer uma orientação ao utilizador, como uma comunicação máquina-utilizador. Para o protótipo recorreu-se à biblioteca gTTS, que permite gerar ficheiros de áudio a partir de *strings* de texto com suporte a múltiplos idiomas e opções básicas de velocidade. Optou-se pelo gTTS pela sua API simples e por produzir vozes com qualidade aceitável para prototipagem. Em vez de salvar ficheiros MP3, o áudio é reproduzido diretamente em memória com Pygame, evitando I/O desnecessário. Esta combinação permite que o protótipo forneça feedback sonoro ao utilizar sem necessidade de armazenar ficheiros MP3 permanentes: o áudio é gerado e reproduzido de forma transitória (ficheiro temporário ou em memória), reduzindo I/O e simplificando o fluxo de integração com o Unity. No Código 5 temos um exemplo de uma configuração que utiliza a biblioteca gTTS juntamente com Pygames para produção de um áudio através de texto.

```

from gtts import gTTS
import tempfile
import os
import time
import pygame

def speak_text_tempfile(text, lang="pt"):
    # Inicializa o mixer do pygame (fazer uma vez no startup do app)
    if not pygame.mixer.get_init():
        pygame.mixer.init()
    # Cria ficheiro temporário
    tmp = tempfile.NamedTemporaryFile(delete=False, suffix=".mp3")
    tmp_name = tmp.name
    tmp.close()
    try:
        # Gera e salva o MP3 temporário
        tts = gTTS(text=text, lang=lang, slow=False)
        tts.save(tmp_name)
        # Reproduz com pygame (bloqueante - espera até terminar)
        pygame.mixer.music.load(tmp_name)
        pygame.mixer.music.play()
        # Espera enquanto estiver a tocar
        while pygame.mixer.music.get_busy():
            time.sleep(0.05)
    finally:
        # Remove ficheiro temporário
        try:
            os.remove(tmp_name)
        except OSError:
            pass

if __name__ == "__main__":
    speak_text_tempfile("Olá! Comando recebido com sucesso.", lang="pt")

```

Código 5 - Exemplo de código de produção de conversão de texto em áudio utilizando gTTS e pygame (em python)

5.1.2 Unity

Após a configuração em Python, implementou-se no Unity o recetor de dados enviado por sockets. O script UDPReceive.cs escuta datagramas UDP (User Datagram Protocol) e encaminha os dados para os módulos do jogo, assegurando que a interface gráfica e a lógica são atualizadas de forma consistente e eficiente. No Código 6 apresenta-se a implementação final do UDPReceive.cs.

```

using System.Net;
using System.Net.Sockets;
using System.Threading;
public class UDPReceive : MonoBehaviour
{
    private Thread receiveThread;
    private UdpClient client;
    private UdpClient senderClient;
    public int receivePort = 5052;
    public string address = "127.0.0.1";
    public bool startReceiving = true;
    public bool printToConsole = false;
    public string data;

    void Start()
    {
        receiveThread = new Thread(ReceiveData);
        receiveThread.IsBackground = true;
        receiveThread.Start();
        senderClient = new UdpClient(); // Inicializa o UdpClient
    }
    private void ReceiveData()
    {
        try
        {
            client = new UdpClient(receivePort);
            IPEndPoint anyIP = new(IPAddress.Any, 0);
            while (startReceiving)
            {
                try
                {
                    byte[] dataByte = client.Receive(ref anyIP);
                    data = Encoding.UTF8.GetString(dataByte);

                    if (printToConsole) { Debug.Log(data); }
                }
                catch (SocketException ex)
                {
                    Debug.LogError($"Socket Exception: {ex.Message}");
                }
            }
        }
        catch (Exception err)
        {
            Debug.LogError($"Error in UDP Receive: {err}");
        }
        finally
        {

```

```

        CloseSocket();
    }
}
private void CloseSocket()
{
    if (client != null)
    {
        client.Close();
        client = null;
    }
    if (senderClient != null)
    {
        senderClient.Close();
        senderClient = null;
    }
}
private void OnApplicationQuit()
{
    startReceiving = false;
    CloseSocket();
}
}

```

Código 6 - Script **UDPReceive.cs** (em C#)

Na Figura 27 podemos ver uma breve explicação através de um diagrama como funciona a comunicação entre o Python e o Unity para o envio de dados.

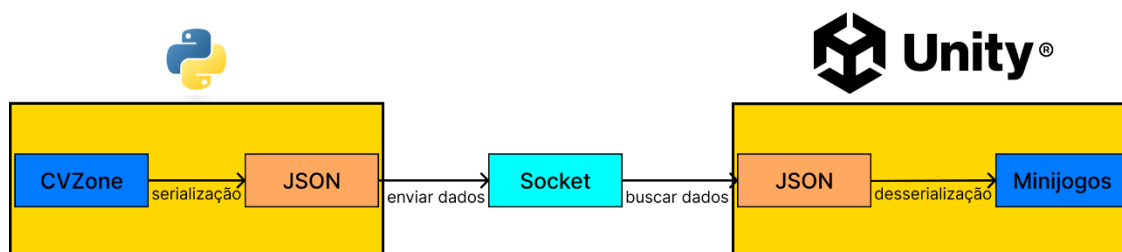


Figura 27 - Diagrama de comunicação entre Python e Unity¹

5.1.2.1 Rastreamento de Gestos

Para o rastreamento de gestos foi necessário converter os dados recebidos do módulo Python (em formato JSON) para objetos do C# de forma a facilitar o seu consumo pelos componentes do Unity. Para esse efeito utilizou-se a biblioteca de terceiros Newtonsoft.Json, que fornece funcionalidades avançadas para serialização e desserialização JSON. A função `JsonConvert.DeserializeObject` é usada para transformar os dados JSON num dicionário onde

¹ Este trabalho não é patrocinado nem afiliado à Unity Technologies nem à Python Software Foundation. "Unity" e o logótipo Unity são marcas comerciais da Unity Technologies. "Python" e o logótipo Python são marcas registadas da Python Software Foundation.

cada chave representa um tipo de mão e o valor da lista de *landmarks*. No excerto apresentado no Código 7 mostra-se essa conversão para o tipo Dictionary<string, HandData>.

```
using Newtonsoft.Json;
using System;
public class Utilities : MonoBehaviour
{
    ...

    public class HandData
    {
        public List<List<float>> LM { get; set; }
    }
    ...

    /// <summary>
    /// Obtém os pontos da mão a partir da mensagem UDP em JSON.
    /// </summary>
    public Dictionary<string, HandData> GetHandPoints()
    {
        string data = udpReceive != null ? udpReceive.data : null;
        if (!string.IsNullOrEmpty(data))
        {
            try
            {
                return JsonConvert.DeserializeObject<Dictionary<string,
HandData>>>(data);
            }
            catch (JsonException ex)
            {
                Debug.LogWarning($"[Utilities] Erro ao desserializar JSON de hand
points: {ex.Message}");
            }
        }
        return null;
    }
    ...
}
```

Código 7 - Conversão de JSON para objeto C# no rastreamento de mãos (em C#)

Após a conversão, os pontos dos *landmarks* foram transformados em posições visíveis no ambiente do jogo. Para tal, foram criados objetos (denominados “circles”) que recebem as coordenadas (x, y, z) correspondentes a cada *landmark*.

Mas, antes dessa implantação, convém notar uma diferença importante de convenções de coordenadas entre as ferramentas OpenCV: que utiliza coordenadas de imagem com origem no canto superior-esquerdo (x crescente para a direita e y crescente para baixo); e Unity: que usa um sistema de coordenadas 3D cuja origem e orientação podem diferir (por exemplo: no espaço do mundo, o eixo Y normalmente aponta para cima). Para alinhar o referencial vertical (OpenCV usa origem no canto superior-esquerdo), aplicou-se em Python a transformação:

$$y_unity = frame_height - y_opencv$$

Depois de invertido o eixo Y, as coordenadas foram normalizadas/escaladas para as unidades do mundo do Unity. A Figura 28 ilustra de forma representativa a diferença entre os sistemas de coordenadas.

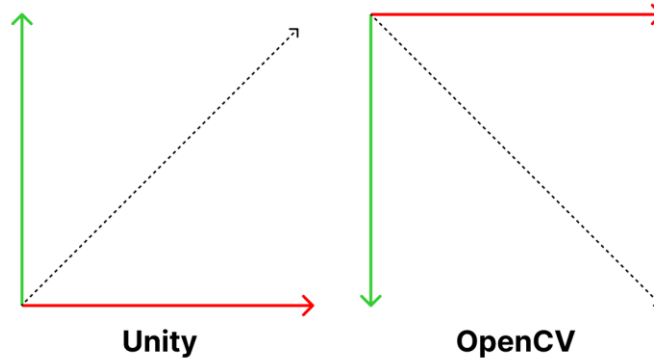


Figura 28 - Ilustração representativa da diferença entre os eixos de referência do OpenCV (origem no canto superior-esquerdo) e do Unity (sistema de coordenadas do motor)

Além da correção do eixo Y, foi necessário ajustar a escala das coordenadas: os valores provenientes do cvzone/OpenCV encontravam-se tipicamente na ordem das centenas (pixels), enquanto a cena do Unity foi dimensionada para trabalhar com magnitudes mais pequenas (ordem das dezenas/unidades). Para normalizar os valores, aplicou-se um fator de escalada, onde cada componente (x, y, z) de cada *landmark* foi dividido por uma constante **SCALE** antes de ser atribuído às posições dos objetos. O Código 8 ilustra esses ajustes de escala e a inovação da função que define a posição de cada ponto

```
for (int i = 0; i < scaledPoints.Count; i++)
{
    var x = scaledPoints[i][0] / SCALE;
    var y = scaledPoints[i][1] / SCALE;
    var z = scaledPoints[i][2] / SCALE;
    SetHandPoints(x, y, z, handIndex, i);
}
```

Código 8 - Normalização das coordenadas (divisão por factor de escala) antes de aplicar aos objetos do Unity (em C#)

Com as coordenadas corrigidas e escaladas, os pontos foram atribuídos aos objetos “circles” responsáveis pela visualização dos *landmarks*. O excerto do Código 9 mostra a função SetHandPoints, que atualiza a posição local dos objetos correspondentes a cada *landmark* para as duas mãos.

```

private void SetHandPoints(float x, float y, float z, int hand, int point)
{
    Vector3 position = new(x, y, z);
    if (hand == 0)
    {
        if (hand1Points[point] != null &&
hand1Points[point].transform.localPosition != position)
            hand1Points[point].transform.localPosition = position;
        ...
    }
    else if (hand == 1)
    {
        if (hand2Points[point] != null &&
hand2Points[point].transform.localPosition != position)
            hand2Points[point].transform.localPosition = position;
        ...
    }
}

```

Código 9 - Aplicação das coordenadas aos objetos representativos dos *landmarks* (em c#)

Para melhorar a legibilidade e dar a impressão de um “esqueleto” de mão (em vez de pontos soltos) implementou-se um conjunto de linhas que ligam pares de *landmarks*. Cada linha foi configurada com um *LineRenderer* e com um script de ligação que atualiza as posições de origem e destino a cada frame. O Código 10 mostra esse componente responsável por ligar dois pontos de *landmark* e manter a linha atualizada durante a execução. E com isso, quando o utilizador passar com a mão a frente da camara, já aparecia o esqueleto da mão, como podemos ver na Figura 29.

```

public class LineCode : MonoBehaviour
{
    LineRenderer lineRenderer;
    [SerializeField] private Transform origin; // landmark de origem
    [SerializeField] private Transform destination; // landmark de destino
    [SerializeField] private float lineWidth = 0.05f;

    void Start()
    {
        lineRenderer = GetComponent<LineRenderer>();
        if (origin == null || destination == null)
        {
            Debug.LogWarning("Origin ou Destination não estão atribuídos no LineConnector.");
            enabled = false; // desativa o script para evitar erros
            return;
        }
        lineRenderer.startWidth = lineWidth;
        lineRenderer.endWidth = lineWidth;
    }

    void Update()
    {
        lineRenderer.SetPosition(0, origin.localPosition);
        lineRenderer.SetPosition(1, destination.localPosition);
    }
}

```

Código 10 - Configuração das ligações do *landmarks* (em c#)

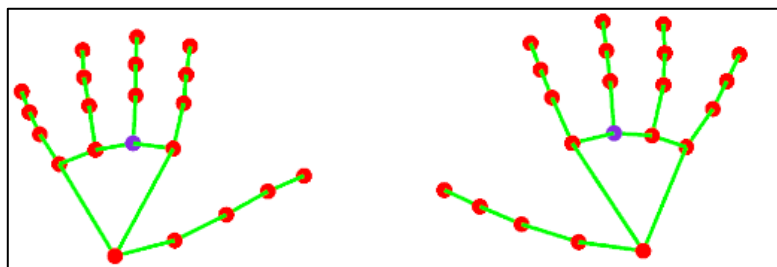


Figura 29 - Esqueletos das mãos dentro do jogo

5.1.2.2 Comando de voz

As configurações do sistema de comando de voz no Unity foram relativamente simples em comparação com o rastreamento de gestos, uma vez que a informação recebida do módulo Python é mais compacta (estado do reconhecimento e texto reconhecido). O Python comunica ao Unity três estados principais: *listening* (a aguardar fala), *command* (áudio convertido em texto) e *error* (problema na detecção ou no serviço de STT); junto com a *string* de texto quando aplicável. No lado do Unity a conversão dos dados JSON foi efetuada com JsonUtility do UnityEngine, como o apresentado no Código 11.

```

public class Speech
{
    public string type; // "listening" | "command" | "error"
    public string text; // texto reconhecido (quando type == "command")
}

void Update()
{
    string data = utilities?.GetVoiceCommand(); // obtém a string JSON vinda do
receptor UDP/queue

    if (string.IsNullOrEmpty(data)) return;
    Speech command;
    try
    {
        command = JsonUtility.FromJson<Speech>(data);
    }
    catch
    {
        Debug.LogWarning("[VoiceCommand] Erro ao interpretar comando de voz.");
        return;
    }
    if (!waiting) // esperar que o jogo tenha iniciado por completo

        HandleCommand(command);
}

```

Código 11 - Conversão de uma String JSON em um objeto C# utilizando JsonUtility (em c#)

Após a conversão, a aplicação procede à interpretação do tipo de mensagem e apresentar o feedback apropriado ao utilizado (por exemplo, texto no ecrã), além de encaminhar comandos válidos para os restantes módulos dos jogos. Na Tabela 4 - podemos ver um resumo de ações possíveis quando o Unity recebe a informação vindas do Python.

Tabela 4 - Ações possíveis ao receber informação de VoiceCommand

Tipo	Ação
<i>listening</i>	Mostra uma mensagem “Diga alguma coisa...” indicando que o sistema está a aguardar entrada do utilizador.
<i>command</i>	O utilizador falou; o texto reconhecido é apresentado como feedback e o comando é encaminhado para os scripts responsáveis por validar e executar a ação.
<i>error</i>	Ocorreu um problema na deteção/serviço de STT; apresenta mensagem de erro e, se necessário, sugere ao utilizador tentar novamente.

5.2 Desenvolvimento do Rastreamento de Mãos

No início do desenvolvimento do protótipo de rastreamento de mãos, desenvolveram-se dois jogos educativos simples: um minijogo de perguntas (Quiz) e um minijogo de identificação de cores (Jogo das Cores). O objetivo principal desta etapa foi validar as mecânicas de interação implementadas no ambiente de desenvolvimento Unity. No Quiz, recorreu-se inicialmente à seleção por contacto do dedo indicador sobre os objetos, mas para que não houvesse deleções involuntárias, fez com que o utilizador se precisa de manter o dedo em cima do objeto durante 2,5 segundos para confirmar a escolha. A mesma técnica foi aplicada no Jogo das Cores, mas em vez do indicador, utilizou-se o ponto central da mão, de forma a ter dois minijogos diferentes com sistemas semelhantes, apenas com variações no método de seleção de objetos para testar qual seria mais eficiente.

Posteriormente, realizou-se o primeiro teste com utilizadores (ver na seção 6.1), no qual os participantes indicaram que o tempo de seleção deveria ser menor e que era necessário um feedback visual que indicasse que a resposta estava a ser selecionada. Durante a análise dos resultados dos questionários do teste (ver seção 6.1.2), propôs-se reduzir o tempo de 2,5 segundos para 1 segundo e adicionar um indicador de “carregamento” para informar o utilizador. Mais adiante no desenvolvimento, adotou-se outra técnica de seleção baseada no fecho da mão (em forma de punho) para indicar ao sistema que o utilizador pretendia selecionar determinado objeto. Verificou-se que este método era mais prático e acabou por ser utilizado na maior parte

dos botões, mantendo-se a seleção por indicador apenas em botões específicos, por serem pequenos e nos quais o gesto de fechar a mão era menos adequado. Na Figura 30 apresenta-se uma comparação do antes e depois para melhor compreensão.

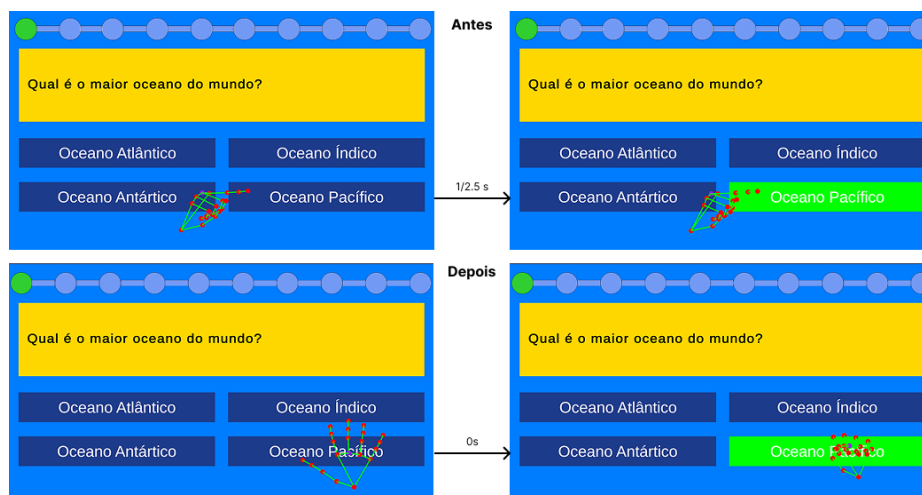


Figura 30 - Comparação entre os sistemas de seleção de objetos em rastreamento de mãos

Posteriormente, no desenvolvimento do protótipo, procurou-se testar várias formas de utilizar os sistemas de rastreamento de mãos e as finalidades a que esses sistemas poderiam dar origem em jogos. Por exemplo, nos dois minijogos de Pong (Pong Solo e Pong Clássico) testou-se o movimento de um objeto (neste caso, as raquetes) apenas pela posição da mão, bem como a possibilidade de implementar jogos multijogadores utilizando este sistema de rastreamento. Outro minijogo criado foi um de reação: tal como no Quiz e no Jogo das Cores, o utilizador tem de posicionar a mão sobre os objetos, mas, em vez de ter de fechar a mão ou esperar para confirmar a seleção, a escolha é efetuada instantaneamente.

No entanto, estes minijogos baseavam-se sobretudo na posição da mão, pelo que surgiu a ideia de criar um minijogo em que o utilizador teria de executar gestos manuais. Assim nasceu o Jogo de matemática, em que o utilizador indica o número que pretende selecionar através da quantidade de dedos levantados (ver Figura 31).

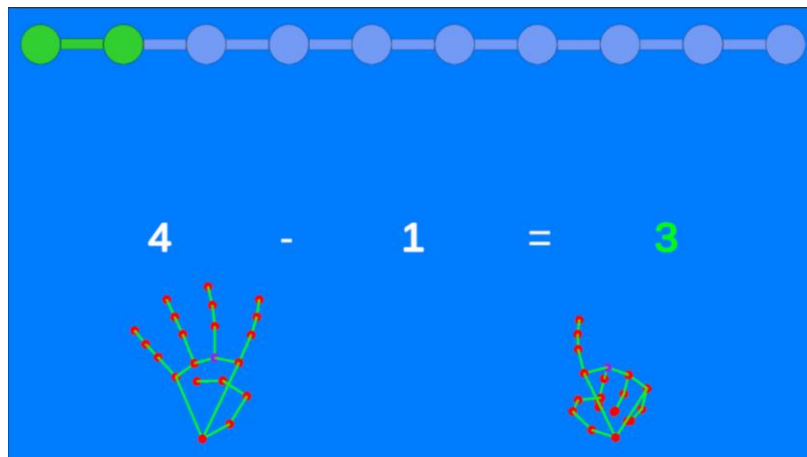


Figura 31 - Demonstração do jogo de Matemática

Seguindo essa lógica, foi também desenvolvido um minijogo de pintura (inspirado no Paint) que explora várias mecânicas: a principal é pintar usando a posição do indicador; para deixar de pintar dá necessário fechar a mão; e existe ainda um gesto de pinça (indicador + polegar) para, por exemplo, escolher a cor a utilizar (ver Figura 32).

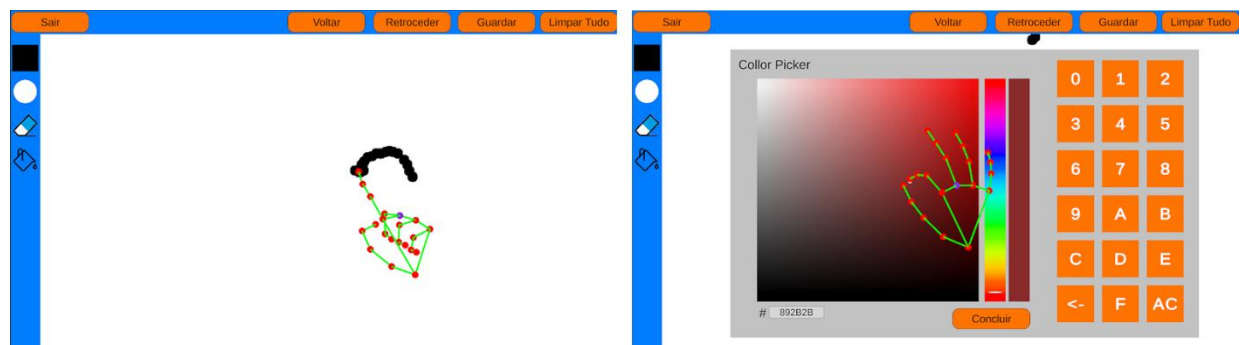


Figura 32 - Exemplo das mecânicas utilizadas no minijogo de Pintura

Por fim, considerou-se um jogo 3D para avaliar como um sistema de rastreamento de mãos 2D, suportado por uma câmara que captura imagens em 2D, se adapta a um ambiente tridimensional. Aqui começou o desafio: o MediaPipe (e, consequentemente, o CVZone) realiza a detecção da mão num plano 2D, pelo que foi necessário encontrar uma heurística para estimar a posição no eixo Z. Após pesquisa, optou-se por utilizar a “largura” da mão, medindo a distância entre o ponto na base do indicador e o ponto na base do mindinho, de modo a definir o posicionamento no eixo Z, quanto mais próximo a mão esta da câmara, maior é essa distância na imagem capturada. O processo encontra-se exemplificado no Código 12 e o resultado obtido pode ser visto no minijogo das Damas 3D na Figura 33.

```

float x0 = LM[5][0] - LM[17][0];
float y0 = LM[5][1] - LM[17][1];
float z0 = LM[5][2] - LM[17][2];

// float z_offset = 20 - (width_pix / 10);
// float hand_size = 200/width_pix;
// middle_width = 150, z = 0
// max_width = 550, z = 20
// Slope = (z1 - z2) / (x1 - x2)
// Slope = (0 - 20) / (150 - 550) = 0.05
// b = 150 * 0.05 + b = 0 <=> b = 7.5
float width_pix = Mathf.Sqrt(Mathf.Pow(x0, 2) + Mathf.Pow(y0, 2) +
Mathf.Pow(z0, 2));

float x_hand = 1.5f * (LM[0][0] / SCALE) - 22.5f;
float y_hand = 1.1f * (LM[0][1] / SCALE) - 2f;
float z_hand = 0.15f * width_pix - 20f;

hand.transform.position = new Vector3(x_hand, y_hand, z_hand);

```

Código 12 - Código para definir a posição da mão no eixo dos Z

(Ajraou, 2025)

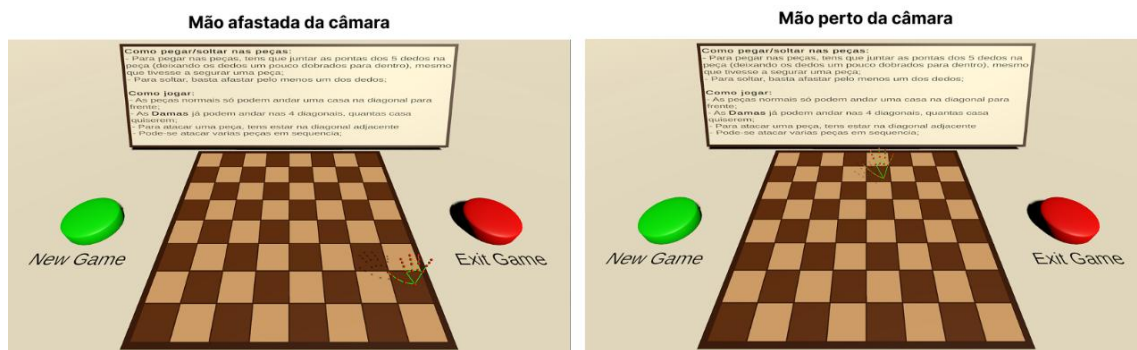


Figura 33 - Posicionamento da mão num ambiente 3D

Posteriormente, essa lógica também foi utilizada para fixar o tamanho do esqueleto da mão nos minijogos 2D: assim, dependendo da distância da mão à câmara, o esqueleto da mão mantinha aproximadamente o tamanho, como podemos ver na Figura 34.

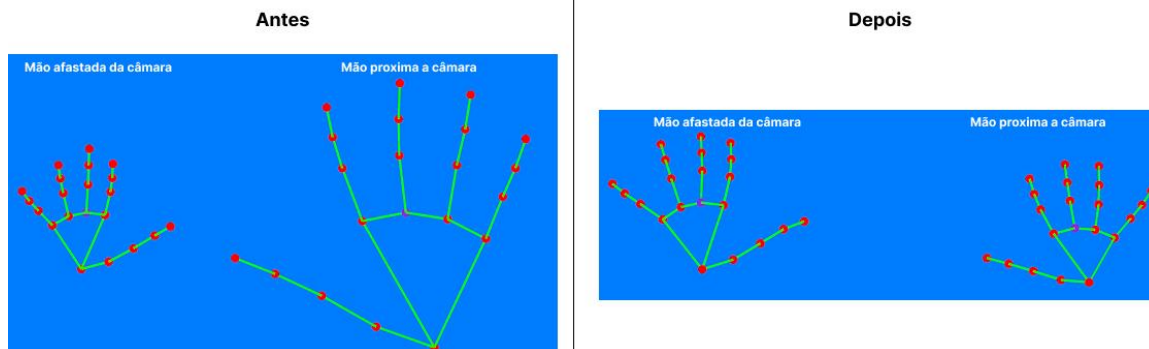


Figura 34 - Comparação do tamanho do esqueleto da mão, antes e depois de fixar o tamanho

Na Tabela 5 apresenta-se um resumo dos minijogos implementados, das mecânicas exploradas e das finalidades de cada um.

Tabela 5 - Tabela de resumo das mecânicas e finalidades de cada minijogo feito (rastreamento de mãos)

Jogo	Mecânica	Finalidade
Quiz	Selecionar um objeto com o fechar da mão	Educação
Jogo das Cores	Selecionar um objeto com o fechar da mão	Educação e análise de daltonismo em pessoas
Pongs	O movimento da mão movimentar um objeto junto e	Diversão, multijogador e análise de utilização do sistema para jogos mais dinâmicos
Jogo de Reação	Selecionar um objeto só tocando nele	Análise de reação em pessoas em reabilitação
Jogo de Matemática	Deteção de número através de dedos levantados	Educação
Paint	Pintar uma tela através de um ponto da mão e fazer gestos para ativar ações	Diversão
Damas 3D	Segurar objetos fazendo gestos parecidos com segurar objetos na vida real	Diversão e análise de como um sistema de deteção de mão 2D funciona para ambientes 3D

5.3 Desenvolvimento do comando de voz

Para o desenvolvimento dos protótipos que utilizaram o sistema de comandos de voz, optou-se por adaptar inicialmente o Quiz, já implementado para rastreamento de mãos, para que pudesse ser jogado através de comandos de voz. Este minijogo foi pensado, sobretudo, para o primeiro teste (ver seção 6.1), de modo a verificar se as configurações do STT estavam a funcionar corretamente e se adaptavam a ambientes ruidosos, tanto para comandos pré-configurados como para leitura de palavras.

Como resultado do primeiro teste (ver seção 6.1.2), os utilizadores referiam que o STT nem sempre reconhecia os comandos e quer era necessário repetir várias vezes até o sistema entender. Assim, pesquisou-se se a biblioteca utilizada tinha opções para ajustar o nível de captura e reduzir a sensibilidade a ruído de fundo. Nas documentações da biblioteca (Zhang, et

al., 2025) verificou-se que o SpeechRecognition dispõe de um parâmetro (*energy_threshold*) que permite ajustar a sensibilidade do STT, filtrando ruídos de fundo. Deve ter-se, no entanto, em conta que uma sensibilidade demasiado baixa pode prejudicar o reconhecimento de palavras curtas, por isso testaram-se vários valores até encontrar um compromisso entre rejeitar ruído e reconhecer palavras curtas.

Após esse ajuste, adaptou-se também os dois minijogos de Pong para suporte por comando de voz. Apesar das mecânicas básicas serem semelhantes às utilizadas no Quiz (comandos pré-configurados), estes minijogos permitiram avaliar a viabilidade de comandos de voz em jogos que exigem reação rápida, tal como no jogo *Lifeline* referido na seção 2.2.2, e verificar se persistiam os problemas de latência ou de má interpretação por parte do STT.

Por fim, criou-se um minijogo de Memória (jogo da memória) cuja ideia inicial era ser utilizado por utentes idosos em reabilitação para avaliar o estado das suas memórias. Mas depois de integrar TTS e efeitos sonoros, confirmou-se que o minijogo podia ser jogado integralmente sem recurso ao ecrã, o que motivou a utilização de comandos de voz em contextos de acessibilidade para pessoas com deficiência visual. No segundo teste (ver seção 6.2) foi incluído um desafio no qual os participantes jogassem o Jogo da Memória sem olhar para o ecrã, para avaliar essa possibilidade.

Na Tabela 6 apresenta-se um resumo dos minijogos implementados, das mecânicas exploradas e das finalidades de cada um.

Tabela 6 - Tabela de resumo das mecânicas e finalidades de cada minijogo feito (comando de voz)

Jogo	Mecânica	Finalidade
Quiz	Falar um texto curto e usar um comando fixo	Educação
Pongs	Comandos de posição	Diversão, multijogador e avaliação para jogos dinâmicos
Jogo da Memória	Repetir a palavra solicitada	Reabilitação de idosos e avaliação de acessibilidade para pessoas com deficiência visual

Durante o desenvolvimento e com a base nas análises do primeiro teste, surgiu a necessidade de melhorar a robustez do STT mediante uma verificação por similaridade: quando o STT interpretava incorretamente uma palavra (por exemplo, “janela” -> “canela”), o sistema comparava a transcrição com a palavra pretendida e, se a similaridade excedesse um limiar (por

exemplo, 65%), aceitava-se a correspondência com válida. Esta estratégia revelou-se especialmente útil para palavras longas cujas sílabas finais podem ser mal transcritas.

Para calcular a similaridade utilizou-se o algoritmo de Levenshtein, que mede a “distância de edição” entre duas sequências. Por exemplo, a distância de edição entre “janela” e “canela” é 1, pois basta uma substituição (j -> c) para transformar uma palavra na outra (Wikipedia.org, 2023). No excerto do Código 13 demonstra-se a utilização do algoritmo Levenshtein em conjunto com a normalização do texto (minúsculas e remoção de acentos) e o processo de reconhecimento.

```
/// Normalize texto: trim, lower, remove diacríticos e carecteres não necessários
private static string Normalize(string s)
{
    if (string.IsNullOrEmpty(s)) return "";
    s = s.Trim().ToLowerInvariant();
    if (s.Length > 0 && s[0] == '\uffeff') s = s[1..]; // remove BOM se existir
    // Normalize para FormD e remove diacríticos
    string formD = s.Normalize(NormalizationForm.FormD);
    StringBuilder sb = new StringBuilder();
    foreach (var ch in formD)
    {
        var uc = CharUnicodeInfo.GetUnicodeCategory(ch);
        if (uc != UnicodeCategory.NonSpacingMark) sb.Append(ch);
    }
    // só manter letras e número e espaços
    var cleaned = new StringBuilder();
    foreach (char c in sb.ToString())
        if (char.IsLetterOrDigit(c) || char.IsWhiteSpace(c)) cleaned.Append(c);
    return cleaned.ToString().Normalize(NormalizationForm.FormC).Trim();
}

/// API principal: devolve a melhor opção (ou null)
public string ProcessRecognized(string textRecognized, List<string> listOptions,
float minSimilarity = 0.65f)
{
    float bestSimilarity = 0f;
    if (string.IsNullOrEmpty(textRecognized) || listOptions == null ||
listOptions.Count == 0) return null;
    string normRec = Normalize(textRecognized);
    string bestOption = null;
    foreach (var option in listOptions)
    {
        string normOpt = Normalize(option);
        float sim = Similarity(normRec, normOpt);
        if (sim > bestSimilarity)
        {
            bestSimilarity = sim;
            bestOption = option;
        }
    }
    if (bestSimilarity >= minSimilarity)
        return bestOption;
    return null;
}
```

```

/// Calcula a similaridade entre 0 e 1, baseada em Levenshtein
private static float Similarity(string a, string b)
{
    if (a == b) return 1f;
    if (string.IsNullOrEmpty(a) || string.IsNullOrEmpty(b)) return 0f;
    int distance = Levenshtein(a, b);
    int maxLen = Mathf.Max(a.Length, b.Length);
    if (maxLen == 0) return 1f;
    float sim = 1.0f - (float)distance / maxLen;
    // Clamping por segurança
    if (sim < 0f) sim = 0f;
    if (sim > 1f) sim = 1f;
    return sim;
}

/// Algoritmo Levenshtein (iterativo, eficiente o suficiente para palavras curtas)
private static int Levenshtein(string s, string t)
{
    int n = s.Length;
    int m = t.Length;
    if (n == 0) return m;
    if (m == 0) return n;
    int[] prev = new int[m + 1];
    int[] cur = new int[m + 1];
    for (int j = 0; j <= m; j++) prev[j] = j;
    for (int i = 1; i <= n; i++)
    {
        cur[0] = i;
        for (int j = 1; j <= m; j++)
        {
            int cost = (s[i - 1] == t[j - 1]) ? 0 : 1;
            cur[j] = Mathf.Min(Mathf.Min(cur[j - 1] + 1, prev[j] + 1), prev[j - 1] + cost);
        }
        (cur, prev) = (prev, cur); // swap
    }
    return prev[m];
}

```

Código 13 - Código de configuração da verificação por similaridade

Ainda durante o desenvolvimento, foi-se alterando o idioma do ouvinte do STT entre português de Portugal (pt-PT) e português do Brasil (pt-BR) para analisar qual dos dois proporcionaria a melhor experiência aos utilizadores. Após alguns testes verificou-se que o pt-BR se revelou mais eficaz no reconhecimento de palavras, apesar das diferenças de sotaque entre pt-PT e pt-BR, já que no modo pt-PT observou-se por vezes que, quando uma palavra não era bem reconhecida, o sistema chegava a transcrever uma palavra em inglês (por exemplo: “janela” -> “chanel”), prejudicando a experiência do utilizador.

5.4 Desafios no desenvolvimento

Ao longo do desenvolvimento surgiram diversos desafios técnicos e organizacionais, dos quais o mais destacado foi a configuração e a integração dos diferentes sistemas. Nunca tinha utilizado mais do que uma linguagem de programação para o mesmo projeto, pelo que a comunicação entre componentes foi uma fonte constante de problemas de configuração em ambientes de teste. Por vezes era necessário iniciar e encerrar os protótipos com frequência e, em alguns casos, os mecanismos de comunicação não eram corretamente finalizados e ficavam à escuta, originando conflitos quando se tentava reiniciar. A resolução exigia, muitas vezes, um reinício completo do sistema. Este tipo de problema evidenciou a necessidade de mecanismos de encerramento controlado de processos.

Outro desafio prático foi optar por testar os minijogos sem recurso ao rato e teclado, de forma a validar continuamente os sistemas alternativos em condições reais de utilização. Isso implicou testar repetidamente várias formas de interação para determinar quais eram mais acessíveis e eficientes para utilizadores finais, o que aumentou o tempo de iteração e a complexidade dos testes. Onde surgiu a necessidade de criar uma explicação ao utilizador para conseguir utilizar o sistema nas melhores condições.

As restrições do ambiente de teste constituíram igualmente uma limitação significativa: os sistemas de reconhecimento de voz exigiram ambientes relativamente silenciosos e o rastreamento de mãos beneficiou sempre de boa iluminação. Essas exigências tornaram inviável realizar testes em locais públicos ou ruidosos, como bibliotecas ou cafeterias, e reduziram de cenários reais avaliados durante o desenvolvimento. Este facto salientou a importância de testar em condições diversas, de adotar técnicas de supressão de ruído e de explorar sensores menos dependentes da luz ambiente (por exemplo câmaras de profundidade) para aumentar a robustez.

Por fim, houve um desafio de natureza criativa: enquanto os comandos de voz já contam com vários exemplos inspiradores na indústria e na pesquisa, existiram muito menos referências de jogos baseados em rastreamento de mãos, o que dificultou a fase de ideação. No entanto, essa escassez de material transformou-se em oportunidade, pois permitiu experimentar e definir novas formas de interação, explorar mecânicas alternativas e propor soluções originais. A experiência mostrou a utilidade de envolver utilizadores em sessões de co-criação e testes iterativos desde cedo, para gerar ideias mais alinhadas com as necessidades reais e para validar rapidamente hipóteses de design.

6 Testes com utilizadores

No presente capítulo, são apresentados os dois ciclos de testes de usabilidade realizados com utilizadores. Os resultados revelam interesse pelos dois sistemas, apontando o comando de voz como mais acessível em hardware comum e o rastreamento de mãos como mais imersivo. São identificados problemas de usabilidade, como fadiga física e latência, mas também oportunidades de melhoria para futuras versões.

6.1 1.º Teste com Utilizadores (Teste de Usabilidade)

Durante a fase inicial do desenvolvimento dos protótipos, foi decidido realizar um teste de usabilidade preliminar com o objetivo de avaliar se a funcionalidade e o desempenho das configurações iniciais estavam alinhados com os objetivos propostos, ou se seriam necessários ajustes antes de dar continuidade ao desenvolvimento. (Hotjar Ltd, 2023)

Este teste foi conduzido presencialmente com os participantes e teve início uma breve apresentação, cujo propósito foi contextualizar os participantes e explicar as atividades que iriam realizar. De seguida, os participantes foram convidados a experimentar três minijogos protótipos disponíveis até ao momento: dois deles baseados em rastreamento de mãos e comando de voz (Quiz) e um baseado apenas em rastreamento de mãos (Jogo das Cores). Apesar da simplicidade destes minijogos, a experiência permitiu obter uma visão mais clara sobre os aspetos que necessitavam de ser melhorados ou ajustados, contribuindo assim para o desenvolvimento dos minijogos mais complexos nas fases seguintes.

Este teste ajudou a medir a facilidade com que os utilizadores conseguiram interagir com os protótipos e identificar problemas de usabilidade que passaram despercebidos durante as configurações.

Por motivos de legibilidade, as alternativas sem respostas (0%) não são representadas, igualmente as questões em que uma única alternativa obteve 100% das respostas são reportadas em texto, em vez de se apresentarem gráficos. A listagem completa das alternativas e o formulário original encontram-se no Anexo D – 1º Formulário

6.1.1 Perfil dos Participantes

Durante este primeiro teste, só houve participação de 10 indivíduos, no entanto eram todos estudantes de Engenharia Informática. O número de participante foi muito pequeno devido a um problema de exportação do jogo para seu executável, onde havia a necessidade de os jogadores terem um compilador de python instalado nos seus pc, fazendo só que pudessem fazer teste presencialmente. Mas este número de participantes já ajudou a ter uma melhor visão do que era preciso ser feito para melhorar o desenvolvimento do protótipo. Na Tabela 7 é possível verificar os dados dos participantes.

Tabela 7 - Dados sobre os participantes (primeiro teste)

Pergunta	Alternativas	N%
Idade	Entre 20 e 30	100%
Género	Masculino	100%
Formação académica (Maior frequentada)	Licenciatura	50%
	Mestrado	50%
Já trabalhou anteriormente com tecnologias de rastreamento de mãos ou comando de voz antes?	Sim	50%
	Não	50%
Qual o seu nível de familiaridade com estas tecnologias?	Iniciante	80%
	Intermediário	20%

Os participantes envolvidos neste primeiro teste ao protótipo situam-se todos na faixa etária dos 20 aos 30 anos, sendo todos eles do sexo masculino. No que diz respeito ao nível de escolaridade, todos eles já possuem um grau de ensino superior, até o momento da realização deste teste, sendo 50% o grau de licenciatura e os restantes 50% o grau de mestrado. Foi ainda questionado se já haviam tido contacto prévio com tecnologias semelhantes, tendo 50% dos participantes respondido afirmativamente e os restantes 50% negativamente.

Adicionalmente, foi solicitado que indicassem o seu nível de conhecimento na área, tendo 80% referido possuir um nível de conhecimento inicial e 20% um nível intermédio.

6.1.2 Resultado e Análise

Após a conclusão das sessões de teste, os participantes receberam um inquérito composto por 16 questões, destinado a recolher a sua opinião sobre os protótipos, propostas de melhoria e eventuais dificuldades.

As duas primeiras perguntas, destinadas a avaliar a perceção dos participantes sobre a utilidade do rastreamento de gestos e dos comandos de voz para jogos, obtiveram a seguinte resposta: nove dos dez participantes (9/10; 90% do total) responderam “Sim”. Um participante não respondeu a estas perguntas (1/10; 10% do total) e foi registado como “Não respondeu”. Entre os respondentes (n = 9), a escolha “Sim” correspondeu a 100% das respostas.

A terceira pergunta já teve como destino saber se os participantes tiveram uma experiência fácil ao testar os minijogos ou não. O objetivo desta questão era ter 100% dos participantes em um nível fácil ou muito fácil de utilização deste tipo de sistema, mas como podemos ver na Figura 35, 60% dos participantes indicaram que tiveram dificuldades ao experimentar o que indica que a aplicação precisa de melhorias em termos de UI/UX para que os utilizadores consigam mexer melhor nesta aplicação, sem sentirem nenhuma dificuldade.

COMO FOI A SUA EXPERIÊNCIA AO JOGAR OS MINIGAMES?

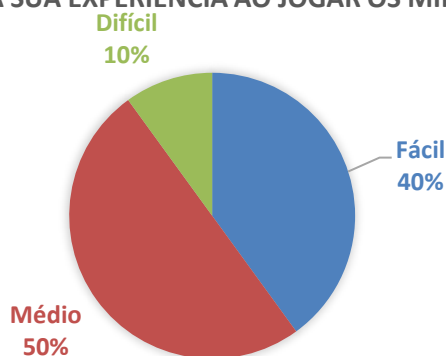


Figura 35 - Gráfico de resposta: "Como foi a sua experiência ao jogar os minigames?"

Na quarta pergunta, questionamos se os participantes sentiram alguma imprecisão ao usar os gestos para jogar, e houve 5 pessoas que responderam que sim, como mostra a Tabela 8, sendo praticamente toda envolta da deteção da mão em si, isto é devido que o MediaPipe é muito sensível a luz. Então, em ambientes que a luz não esteja bem distribuída, pode sim dificultar a deteção da mão e já era esperado que houve este problema durante este primeiro teste, já que é um protótipo muito inicial.

Tabela 8 - Tabela de respostas: “Você sentiu algum atraso ou imprecisão ao usar gestos para jogar?”

Questão	Respostas
Você sentiu algum atraso ou imprecisão ao usar gestos para jogar? Se sim, pode descrever?	R1 - Pequena dificuldade ao reconhecer a mão e o delay de seleção podia ser mais pequeno
	R2 - A distância teria de ser específica para a deteção da mão.
	R3 - Sim, quando o movimento da mão é muito rápido o sensor desaparece. Seria melhor ter um aviso nas regras sobre este acontecimento.
	R4 - Senti um atraso na seleção de respostas, ao ser selecionado a opção.
	R5 - Sim; demora demasiado tempo a detetar que o utilizador quer selecionar e tem alguma imprecisão quando a cor com o fundo e a mão é similar.

Outro problema foi no tempo respostas ao selecionar uma alternativa, mas como já se pensou que isso poderia ser um problema, a quinta pergunta já foi feita a pensar nisso, onde se perguntou se deveria manter o tempo, aumentar ou diminuir. E como mostra na Figura 36, 60% dos participantes disseram que o tempo devia ser mais pequeno. Como neste momento o tempo de seleção de alguma alternativa nos minijogos ou em botões nos menus é de 2.5 segundos, achou-se melhor abaixar para no máximo de 1 segundos.

O TEMPO DE RESPOSTA DO JOGO FOI ADEQUADO?

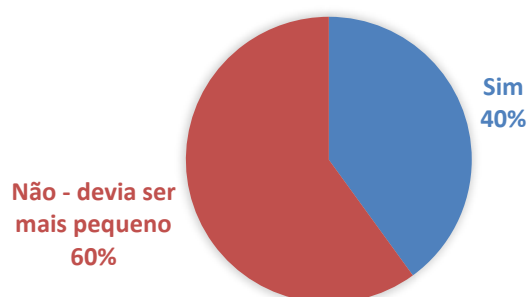


Figura 36 - Gráfico de resposta: "O tempo de resposta do jogo foi adequado?"

Na sexta pergunta, perguntou-se se o controlo por gestos, tantos nos menus como nos minijogos em si. E como é mostrado na Figura 37, nenhum participante respondeu que “não”,

que indica que este primeiro protótipo está no caminho certo, mas mesmo assim, é preciso melhoria já que 30% dos participantes responderam “parcialmente”.

O CONTROLE POR GESTOS PARECEU INTUITIVO?

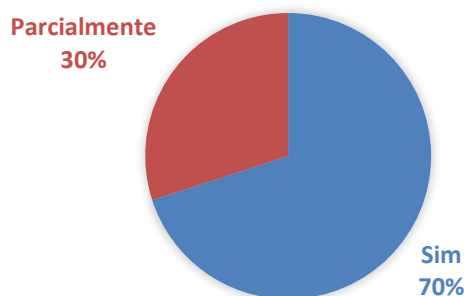


Figura 37 - Gráfico de respostas: "O controle por gestos pareceu intuitivo?"

Na sétima pergunta, o objetivo foi já perceber a opinião dos participantes sobre se este tipo de tecnologia pode tornar as interações como jogos (e não só) mais natural, já que retira a necessidade de objetos de controle, com o rato, teclado e comando para interagir com os jogos. E todos os participantes concordaram com esta linha de pensamento, tendo 100% votos na alternativa "Sim", achando também que estes tipos de tecnologias podem deixar as interações mais naturais.

Na oitava pergunta, já teve o objetivo de perceber quais desafios os participantes tiveram durante o teste. Na Tabela 9 podemos ver que esta pergunta complementou a da Tabela 8, já que dois participantes descreveram mais dois problemas na detecção da mão, mas como explicado anteriormente, estes dois erros indicados também envolvem a iluminação do local que está a ser utilizado o MediaPipe. Os outros dois erros tiveram a ver com o comando de voz, já que visto durante o teste, os participantes tiveram de falar várias vezes para a aplicação conseguir detetar o que o participante disse. Uma das causas possíveis para isto pode ter sido o ruído, já que estes tipos de sistemas são sensíveis a sons externos que podem prejudicar a interpretação pela parte da aplicação. Como este problema afeta diretamente a experiência dos jogadores podendo levar a perda de interesse em mexer neste tipo de aplicações.

Tabela 9 - Tabela de respostas: "Que desafios técnicos você identificou na usabilidade do jogo?"

Questão	Respostas
Que desafios técnicos você identificou na usabilidade do jogo?	R1 – Reconhecimento de voz
	R2 – Dificil reconhecimento da mão com fundos de cor parecida à mão/situações de pouca luz.
	R3 – Por vezes o jogo não identificava o áudio direito, a pessoa tinha de se repetir várias vezes.
	R4 - Precisão da mão, mais especificamente no polegar.

A nona pergunta teve como objetivo recolher a opinião dos participantes relativamente à clareza das instruções fornecidas para os minijogos. Durante a realização dos testes, as instruções apresentadas consistiam apenas num texto explicativo que descrevia as regras e o funcionamento de cada jogo. No entanto, procurou-se perceber se a adição de elementos visuais, como um GIF ou um vídeo, poderia facilitar a compreensão por parte dos utilizadores. Como podemos ver na Figura 38, 80% dos participantes indicaram que a presença de um GIF ou vídeo explicativo contribuiria para uma melhor compreensão do funcionamento dos minijogos. Com base nesse feedback, decidiu-se integrar animações explicativas e um tutorial introdutório para orientar o utilizador na aprendizagem dos jogos. No entanto, importa referir que estas melhorias não serão tratadas como requisitos prioritários no desenvolvimento atual, dado que a atenção principal será direcionada para outros aspetos considerados mais relevantes para os objetivos desta dissertação.

NAS INSTRUÇÕES, ACHAS QUE DE VIA FICAR SÓ TEXTO A EXPLICAR A FUNCIONAMENTO DO JOGO, OU DE VIA TER UM GIF/VÍDEO EXPLICATIVO

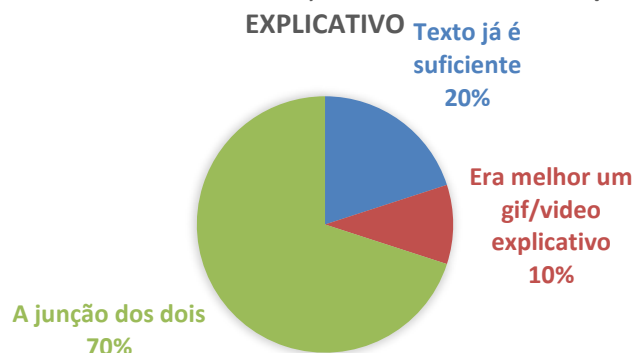


Figura 38 - Gráfico de respostas: "Nas instruções, achas que devia ficar só texto a explicar a funcionamento do jogo, ou devia ter um GIF/vídeo explicativo"

A decima pergunta já teve como objetivo ver a opinião sobre que áreas estas tecnologias podem ser aplicadas e ter uma ideia de que tipos de minijogos poderíamos focar depois deste teste. Como podemos ver na Figura 39, todos os participantes disseram que estas tecnologias podem ser utilizadas para a Educação, mas para este caso tem se atenção que o fato dos minijogos apresentados neste teste serem voltados para essa área, pode ter afetado na escolha por parte de todos participantes. Para além da educação, a maioria dos participantes também disseram que estas tecnologias podem ser para “Reabilitação média”, “Entretenimento” e “Exercício Físico”, mostrando que era ser boa ideia explorar jogos nestas áreas, como já planeado antes deste teste.

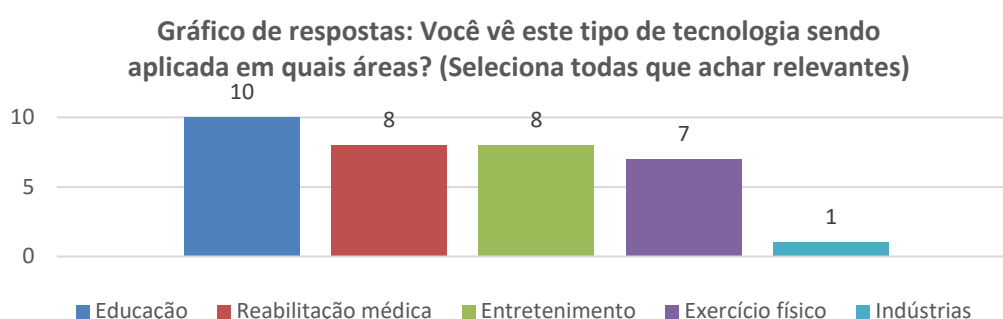


Figura 39 - Gráfico de respostas: “Você vê este tipo de tecnologia sendo aplicada em quais áreas?”

A próxima pergunta pediu para os participantes indicarem o que lhes chamou mais atenção e como podemos ver na Tabela 10, a maioria indicou o fato de utilizar a câmara do portátil para captar a mão e mostrar o esqueleto no ecrã a indicar a posição da mão. Outra resposta importante, é do participante que gostou da utilização de cores diferentes para indicar ao utilizador, no caso do quiz, se tinha resposta certa ou errada.

Tabela 10 - Tabela de respostas: "O que mais lhe chamou a atenção no protótipo?"

Questão	Respostas
O que mais lhe chamou a atenção no protótipo?	R1 – O reconhecimento de mão
	R2 – O feedback com diferentes cores para realçar escolha selecionada e resposta certa/errada
	R3 – As tecnologias de interação.
	R4 – A utilização da câmara para identificar a mão.
	R5 – O uso de <i>hand-tracking</i> para responder as opções.
	R6 – A apresentação do formato da mão

Tendo em conta os elementos que mais chamaram a atenção dos participantes, foi também incluída uma pergunta destinada a recolher sugestões sobre possíveis melhorias na experiência do utilizador. As respostas, apresentadas na Tabela 11, revelam que a maioria das sugestões centraram na implementação de feedback adicional. Como o uso de feedback sonoro nos minijogos controlados por comando de voz e a introdução de um indicador visual “*loader*” nos de rastreamento de mão, com o objetivo de sinalizar que uma opção está a ser selecionada. Adicionalmente, um dos participantes sugeriu substituir o esqueleto da mão por um cursor. No entanto, esta proposta entra em contradição com algumas das preferências ditas na pergunta anterior. Assim, optou-se por uma solução intermédia: alterar a cor do ponto da mão responsável pela interação com os objetos nos minijogos, de forma a melhorar a visibilidade e a perceção de seleção sem comprometer a representação visual já existente.

Tabela 11 - Tabela de respostas: "Quais melhorias técnicas você sugeriria para otimizar a experiência?"

Questão	Respostas
Quais melhorias técnicas você sugeriria para otimizar a experiência?	R1 – Diminuir o delay de seleção.
	R2 – Utilizar som como uma camada extra de feedback, diminuir o input lag no reconhecimento da voz.
	R3 – Um design mais visualmente moderno, como os jogos mais recentes.
	R4 – Alterar o esqueleto da mão para um cursor. Ter feedback visual na seleção, por exemplo um loader à volta.
	R5 – Otimizar a receção da resposta da voz, otimizar a identificação do indicador.
	R6 – Detecção de braço

Já a próxima pergunta foi com objetivo de ver se caso houvesse um segundo teste, poderia contar com a participação dos mesmos. E como podemos ver na Figura 40 a maioria disse que sim, garantindo já alguns participantes para um possível futuro teste.

VOCÊ TERIA INTERESSE EM TESTAR VERSÕES FUTURAS COM NOVAS MECÂNICAS, COMO POR EXEMPLO, COM COMANDO DE VOZ OU OUTROS MINIJOOS?

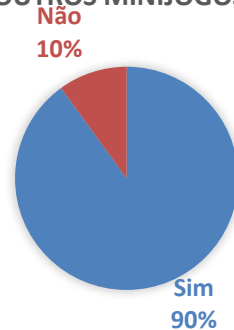


Figura 40 - Gráfico de respostas: "Você teria interesse em testar versões futuras com novas mecânicas, como por exemplo, com comando de voz ou outros minijogos?"

A próxima pergunta, já teve como objetivo ver a opinião dos participantes em utilizar esta tecnologia para jogos. O objetivo desta pergunta era a maioria dos participantes entre as opções "Bastante" e "Muito", mas como podemos ver na Figura 41, os participantes acabaram por ficar divididos entre as opções "Médio" e "Bastante", o que mostra que é preciso melhorar a experiência de utilizador neste protótipo para que os utilizadores vejam as vantagens de jogar jogos com estas tecnologias.

EM TERMOS DAS TECNOLOGIAS IMPLEMENTADAS CONSIDERA TEREM POTENCIAL PARA USO EM JOGOS?

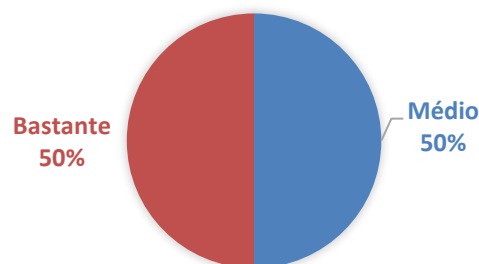


Figura 41 - Gráfico de respostas: "Em termos das tecnologias implementadas considera terem potencial para uso em jogos?"

A próxima pergunta já teve o foco de ver se adicionar uma barra de progresso para alguns minijogos seria interessante para os utilizadores. E todos os participantes responderam que "Sim" (100% de respostas), o que indica que essa adição visual é interessante para os utilizadores, em vez de só mostrar um número do seu progresso.

A próxima pergunta já teve o objetivo de ver se as regras e instruções dos minijogos eram claras. E como podemos ver na Figura 42, a maioria disse que “Sim”, mas houve 10% que disseram que “Não”, este pode ser o caso, que já foi analisado anteriormente, que adicionar um tutorial e/ou um GIF informativo já ajuda na melhor percepção das regras.

AS REGRAS E INSTRUÇÕES FORAM CLARAS?

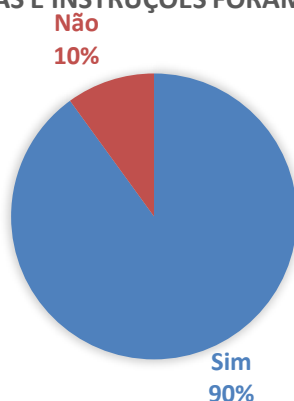


Figura 42 - Gráfico de respostas: "As regras e instruções foram claras?"

E na última pergunta, pediu-se aos participantes para darem algumas sugestões de melhoria para o protótipo. Como podemos ver na Tabela 12, duas delas foram a diminuição do tempo que o jogo demora a confirmar a resposta do jogador, tanto no Rastreamento de Gesto como no de comando de voz, e outras duas foram mais voltadas para o UX, como melhorar o design.

Tabela 12 - Tabela de respostas: "Sugestões de melhoria"

Questão	Respostas
Sugestões de melhoria	R1 – Instruções sucintas e diminuição do <i>delay</i> de seleção
	R2 – As perguntas e respostas serem <i>random</i> . O design ser mais apelativo.
	R3 – Apesar de ser apenas um protótipo, algo que deveria ser melhorado é uma interface mais apelativa ao utilizador.
	R4 – Otimização em geral.
	R5 – Diminuir tempo de seleção e reconhecimento de voz

6.1.3 Conclusão do 1.º Teste de Utilizadores

A realização do primeiro teste com utilizadores mostrou-se fundamental para compreender melhor as dificuldades e as oportunidades de melhoria do protótipo desenvolvido até o momento. De formal geral, os resultados foram bastantes positivos, destacando o potencial das tecnologias de Rastreamento de Gestos e comando de voz na área de jogos.

Verificou-se que a maioria dos participantes reconheceu a utilidade destas tecnologias e considerou que podem tornar as interações mais naturais, eliminando a necessidade de dispositivos físicos como rato e teclado. No entanto, surgiram também desafios importantes: 60% dos utilizadores relataram dificuldades na utilização destas tecnologias ao jogarem os minijogos, apontando para a necessidade de melhorias na UI/UX.

As principais limitações identificadas incluíram imprecisões na deteção de gestos, devido à sensibilidade do sistema à iluminação, atrasos na confirmação de seleções e dificuldades na utilização de comando de voz em ambientes ruidosos. Estas questões já eram esperadas, considerando o estágio inicial do protótipo e reforçam a necessidade de otimizações técnicas. O feedback recolhido também evidenciou o interesse dos participantes em implementadas melhorias, como a introdução de tutoriais mais visuais e na adição de feedback sonoro e visual durante as interações.

Importa ainda destacar que, apesar das dificuldades sentidas, os participantes mostraram-se interessados em participar em futuros testes e demonstraram abertura para utilizar estas tecnologias em jogos, desde que a UX seja melhorada.

Concluindo, este primeiro teste permitiu validar o conceito e identificar pontos críticos que serão fundamentais para guiar as próximas fases do desenvolvimento. As melhorias propostas serão fundamentais para tornar a aplicação mais acessível, intuitiva e atrativa para os utilizadores, alinhando-se com os objetivos iniciais.

6.2 2.º Teste de Utilizadores

No final da fase de desenvolvimento dos protótipos, realizou-se o segundo teste de utilizadores com o objetivo de avaliar o cumprimento das metas da tese e recolher a experiência dos participantes com os sistemas implementados. O teste consistiu no envio do protótipo final a um conjunto de participantes, acompanhado de instruções (ver Anexo F – Instruções do 2º Teste). Não foi necessário realizar ensaios presenciais, pois a necessidade da instalação do interpretador do Python já se encontrava resolvida.

Tal como na análise do primeiro teste, as alternativas sem respostas (0%) não são representadas graficamente, igualmente as questões em que uma única alternativa obteve 100% das respostas são reportadas em texto, em vez de apresentarem gráficos. A listagem completa das alternativas e o formulário original encontra-se no Anexo E – 2º Formulário.

6.2.1 Perfil dos Participantes

No segundo teste participaram oito indivíduos, um número reduzido para análises estatísticas robustas. Ainda assim, a amostra permitiu obter indicações preliminares acerca da utilização prática dos sistemas. Na Tabela 13 é possível observar o perfil dos oito participantes deste segundo teste, os dispositivos utilizados e o contexto ambiental dos participantes, a existência de eventuais limitações e a acessibilidade dos protótipos.

Tabela 13 - Dados sobre os participantes e ambiente testado (segundo teste)

Pergunta	Alternativas	N%
Idade	Entre 20 e 29	100%
Género	Masculino	87,5%
	Feminino	12,5%
Formação académica (Maior frequentada)	12º ano	12,5%
	Licenciatura	25%
	Mestrado	62,5%
Já trabalhou anteriormente com tecnologias de rastreamento de mãos ou comando de voz antes?	Sim	50%
	Sim, mas só Comando de voz	12,5%
	Sim, mas só Rastreamento de mãos	12,5%
	Não	25%
Qual o seu nível de familiaridade com a tecnologia rastreamento de mãos?	1 – Iniciante	37,5%
	2	25%
	3	12,5%
	4	12,5%
	5 – Avançado	12,5%
Qual o seu nível de familiaridade com a tecnologia comando de voz?	1 – Iniciante	12,5%
	2	12,5%
	3	50%
	4	12,5%
	5 - Avançado	12,5%
Dispositivo utilizado no teste	Portátil com câmara integrado	25% (n=2)
	Portátil com microfone externo	12,5% (n=1)
	Computador com câmara externa	37,5% (n=3)
	Computador com microfone externo	75% (n=6)

Condições de iluminação do ambiente (n = 6)	Bom	50%
	Médio	33,3%
	Fraco	16,7%
Condições de ruído do ambiente	Muito bom (Condições ideais)	62,5%
	Bom	37,5%
Distância aproximada entre o utilizador e o computador/portátil (n = 7)	Menos que 0.5m	57,1%
	Entre 0.5 a 1m	42,9%
Tens alguma limitação que afetou a experiência (motora, visual, auditiva, outra)?	Sim	12,5%
	Não	87,5%
Se "Sim": Que tipo?	Visual	100%
O protótipo foi acessível para ti?	Sim	100%

6.2.2 Resultados e Análises

Após a conclusão das sessões de teste, os participantes tiveram acesso a um inquérito composto por 25 questões, destinado a recolher a sua opinião sobre os protótipos, propostas de melhoria e eventuais dificuldades.

Nas duas primeiras perguntas, os participantes foram solicitados a indicar que minijogos testaram, de modo a contextualizar a origem das experiências relatadas. A frequência de utilização de cada minijogo é apresentada na Figura 43. Destaca-se que quatro participantes testaram apenas o protótipo de comando de voz: três por não dispor de câmara nos seus dispositivos e um devido a problemas de interação com o sistema.



Figura 43 - Minijogos testados durante os testes

Como terceira e quarta pergunta, foi pedido para os participantes indicarem como foi a sua experiência com cada protótipo. Conforme ilustrado na Figura 44, a maioria dos participantes classificou a experiência entre “Muito Fácil” a “Médio”, o que é indicativo positivo. Registraram-se, no entanto, algumas dificuldades: um participante avaliou como “Muito Difícil” o rastreamento de mãos. A análise individual de cada participante, revelou que tal classificação se deveu a problemas técnicos desse participante com o sistema de rastreamento, justificando a sua avaliação

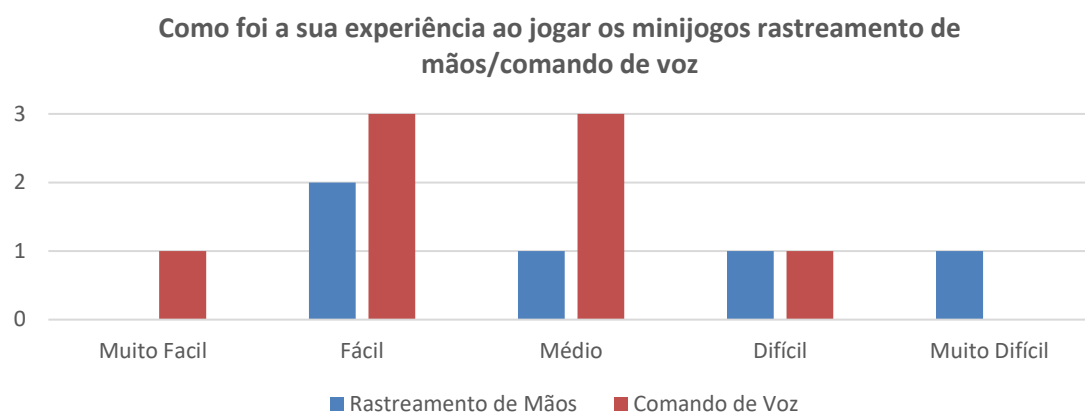


Figura 44 - Dificuldade sentida durante os testes

Como quinta pergunta, foi perguntado se o tempo de resposta foi adequado, e como podemos ver na Figura 45, as respostas dividiram-se entre “Sim” e “Devia ser mais curto”, indicando que o tempo de resposta é ainda excessivo. Ao contrário do primeiro teste, em que as queixas eram sobretudo direcionadas ao rastreamento de mãos, neste ensaio as reclamações centraram-se no comando de voz. Isto sugere uma melhoria no tempo de resposta do rastreamento de mãos, enquanto o comando de voz continua a apresentar latência relevante.

O TEMPO DE RESPOSTA DO JOGO FOI ADEQUADO?

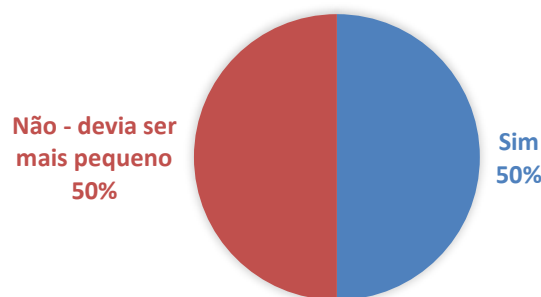


Figura 45 - Gráfico de respostas: "O tempo de resposta do jogo foi adequado?"

Na sexta pergunta pediu-se para os participantes avaliarem a sua confiança de 1 a 5 ao utilizar os sistemas. E como podemos ver na Figura 46, metade dos participantes atribuiu a nota 2 à sua confiança ao usar os sistemas, o que indica um nível baixo de confiança geral, já que o objetivo era que a grande maioria (por volta de 80%) terem dado pontuação de 3 para cima. Esta observação pode refletir a falta de familiaridade com tecnologias de rastreamento de mãos e reforça a necessidade de incluir tutoriais mais detalhados e um período de adaptação antes do uso experimental. E outra análise é de que este tipo de sistemas ainda tem muito para melhorar para que as pessoas possam utilizá-los sem nenhuns problemas.

SENTIU-SE CONFIANTE A USAR OS SISTEMAS?

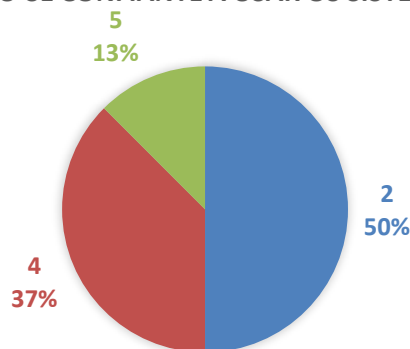


Figura 46 - Gráfico de respostas: "Sentiu-se confiante a usar os sistemas?"

A sétima pergunta já foi só voltada para o rastreamento de gestos, mas que desse também para o comando de voz, onde se perguntou se a pessoa ficou “cansada” depois de testar os jogos. E como podemos ver na Figura 47, dois participantes referiram sentir-se “um pouco” cansados após o teste. Tal resultado é expectável nos minijogos que usam rastreamento de mãos, dado que exigem manter a mão elevada durante períodos prolongados. Este facto sugere que os jogos não devem ser demasiado longos ou deverão incluir pausas estratégicas para evitar fadiga pela parte dos jogadores.

CONFORTO FÍSICO — DEPOIS DE JOGAR, SENTISTE DESCONFORTO FÍSICO (CANSAÇO/OMBROS/PESCOÇO/OLHOS)?

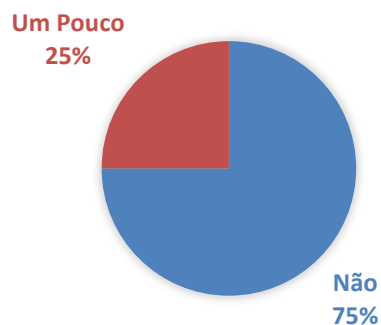


Figura 47 - Gráfico de respostas: "Depois de jogar, sentiste desconforto físico? "

A oitava pergunta já teve com objetivo de saber se as pessoas descreverem os problemas de atraso, que foi perguntado na quinta pergunta, e também falaram de alguma imprecisão que tiveram durante os testes. E como podemos analisar com as respostas na Tabela 14, A maioria dos participantes reportou a demora do comando de voz na interpretação e execução dos comandos; alguns referiram igualmente falhas na captação de áudio, em que o sistema não registou o que foi dito. Ainda podemos ver que duas pessoas que comentaram sobre o rastreamento de mãos, sendo uma delas comentou que o sistema era impreciso, mas quando se analise (mais uma vez) as respostas de cada participante, individualmente, verifica-se que essa pessoa tinha colocado que a condições de iluminação do ambiente que se encontrava era fraco, o que reforça o que já tinha sido analisado, que para o sistema conseguir detetar a mão, a pessoa precisa de estar num ambiente com boa iluminação, caso contrario o sistema vai ter problemas. A última pessoa comentou que teve problemas com a profundidade no jogo das Damas 3D, mas essa pessoa também deixou a indicação que a sua câmara é uma câmara já antiga, o que pode justificar essa imprecisão, mas como não houve mais pessoas que indicaram como eram as suas câmaras, não há uma conclusão final, só uma atenção para a possibilidade de camaras mais antigas terem problemas com este tipo de sistemas.

Tabela 14 - Tabela de respostas: "Você sentiu algum atraso ou imprecisão para jogar em algum dos protótipos? Se sim, pode descrever?"

Questão	Respostas
Você sentiu algum atraso ou imprecisão para jogar em algum dos protótipos? Se sim, pode descrever?	R1 – Disse 6 e ele demorou 10 segundos para responder.
	R2 – Senti atraso no <i>voice command</i> onde o jogador dizia as palavras e ele não reagia se as entendeu ou não.
	R3 – Sim, por vezes o jogo não captava o microfone.

	R4 – Sim, na maioria. No caso do pong solo, onde a rapidez é necessária, é preciso adivinhar quando a bola vai ainda a subir para conseguir ouvir e registrar a posição dita antes da bola conseguir alcançar. Neste caso passou a ser um jogo de sorte.
	R5 – Sim, o tracking das mãos era bastante impreciso, dificultando a navegação pelos jogos.
	R6 – Jogo das Damas (<i>Hand-tracking</i>) não consegui mexer em nenhuma peça depois de alguns minutos a tentar com várias distâncias. Parte do problema pode ser a câmara pois é uma 720p externa já com alguns anos.

Na nona pergunta, perguntou-se se os controles apresentados nos protótipos eram intuitivos. Como podemos ver na Figura 48, a grande maioria respondeu sim, para o comando de voz, mas já no rastreamento ficou dividido entre 50% “Sim” e os outros 50% entre “Não” e “Parcialmente”, mas sendo esses 50% que não indicaram “Sim”, um deles é que teve problemas com o sistema e os outros dois foram o que tinham a iluminação fraca e a camera antiga, sendo justificável a opinião desses três participantes, já que eles não tiveram as condições ideais para testar o rastreamento de gestos.

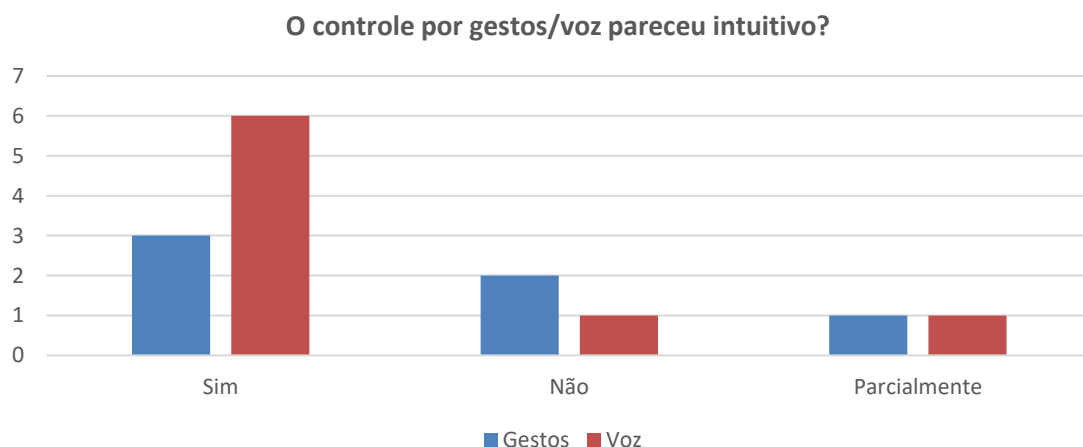


Figura 48 - Gráfico de respostas: "O controle por gestos/voz pareceu intuitivo?"

Depois a decima pergunta, foi questionado se estas tecnologias podiam ser melhoradas para interação mais natural, onde 100% dos participantes responderam que sim, confirmando que todos concordam com o fato que estas tecnologias tem o potencial de serem utilizadas por pessoas como tivessem de interagir naturalmente com outra pessoa (no caso, máquina).

Na decima primeira pergunta, perguntou-se se as regras e instruções tinham sido claras sobre que tinham de fazer nos minijogos, e 100% também responderam que sim, indicando que pode ter havido melhoria a comparar com o primeiro teste, já que lá não 100% das pessoas que não acharam regras e instruções claras, mas não sendo 100% conclusivo, já que no primeiro teste houve mais participantes que neste segundo. Nota: As pessoas que não testaram o rastreamento de mãos por não terem câmara ou ter tido problemas, responderam na mesma “Sim”, mas nesse caso entende-se que só estavam a referir só ao comando de voz.

Na decima segunda pergunta, teve com objetivo o mesmo que a decima primeira pergunta, mas com o foco nos GIFs utilizados nas instruções para o rastreamento de gestos tinham sido uteis para mostrar como o jogo era jogado, e como podemos ver na Figura 49, 100% dos participantes disseram que “Sim”, mas tendo 1 que disse que só com texto (igual utilizado só no comando de voz). O que indica que em comparação com o primeiro teste, houve sim uma melhoria nas instruções, já que lá grandes partes tinham dito que acrescentar GIFs demonstrativos iria ajudar na compreensão das instruções.

NAS INSTRUÇÕES DOS MINIJOGOS DO HAND TRACKING: ACHOU ÚTIL TER UM GIF EXPLICATIVO A MOSTRAR COMO SE INTERAGE COM O JOGO? (N=6)

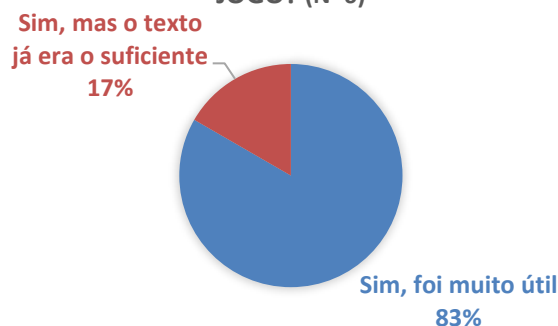


Figura 49 - Gráfico de respostas: "Nas instruções dos minijogos do rastreamento de mãos: achou útil ter um GIF explicativo a mostrar como se interage com o jogo? "

Na decima terceira pergunta tentou-se saber quais desafios técnicos o participantes tiveram durante o teste, e como podemos ver na Tabela 15 que aquele participante que teve problemas com o sistema, indicou que não conseguiu selecionar as caixas, o que por si só não é muito explicativo, porque não se sabe se foi um problema mesmo com o sistema, ou se foi o utilizador que não estava a fechar a mão como era pretendido, mas no caso, se for a segunda hipótese, indica que se houve um GIF demonstrativo para essa parte, provavelmente esse problema não existiria. Outro participante que indicou um desafio técnico, foi em relação ao selecionar o conteúdo no minijogo inspirado no Paint, apesar dessa participante ter sido o que indicou que

as condições de iluminação não eram das melhores, mostra que para sistemas de rastreamento de mãos deve ser ter botão com tamanho consideráveis para que não haja necessidade de uma boa fonte de iluminação para conseguir os selecionar. As restantes respostas foram em volta dos problemas de deteção do sistema de comando de voz utilizado, como já tinha sido verificado anteriormente.

Tabela 15 - Tabela de respostas: "Que desafios técnicos você identificou na usabilidade do jogo?"

Questão	Respostas
Que desafios técnicos você identificou na usabilidade do jogo?	R1 – O <i>hand-tracking</i> não funcionou, não conseguia selecionar as caixas
	R2 – Identifiquei no <i>voice command</i> em que certas palavras/frases o sistema não entendia ou entendia mal, fazendo com que fosse difícil responder a certas perguntas.
	R3 – O tempo de processamento da voz ser muito longo, as vezes a voz também não é detetada ou dava erro de não reconhecer as palavras. Diversas vezes as palavras eram também transformadas em outras minimamente parecidas, como "janela" passar para "chanel".
	R4 – Não conseguia selecionar o conteúdo a maior parte das vezes, principalmente no Paint.
	R5 – Microfone não reconheceu à primeira e tive de reiniciar para reconhecer sem ter de mudar nada nos settings.

Na decima quarta pergunta perguntou-se a opinião de criar jogos 3D usando este sistema de deteção de mão em 2D e falarem um pouco das suas experiências ao jogar as Damas 3D. Como podemos analisar na Tabela 16, 2 participantes reportaram que tiveram problemas com a profundidade e um quarto reportou que teve problemas ao jogar, mas que podia ser devido a câmara. O problema da percepção de profundidade era previsível, dado que se recorre a deteção em 2D para inferir informação 3D. A precisão final depende fortemente da qualidade da câmara e das condições de iluminação, pelo que a reprodução espacial por ser limitada com equipamentos convencionais. Mas como um dos participantes mencionou, com a evolução das câmaras disponíveis no mercado, cada vez vai haver portáteis com câmaras com melhor qualidade (já que neste momento, muitos portáteis não tem câmara com qualidade muito alta, já que essas câmaras muitas das vezes são só utilizadas para vídeos-chamas rápidas, tipo reunião, e não para fotografar como uma câmara de um telemóvel bom) e também o preço de câmaras com visão laser, que conseguem ver na profundidade, serão cada vez mais acessíveis

para utilizadores comuns (sendo que neste momento o custo de uma câmara com visão 3D de boa qualidade anda por volta de 300€-1500€). Mostrando, assim, o quanto este tipo de sistemas pode evoluir com os anos, e como uma das respostas, ter potencial para criar um modelo imersivo online.

Tabela 16 - Tabela de respostas: "O que achou sobre criar um jogo 3D utilizando rastreamento de mãos e uma câmara de computador normal?"

Questão	Respostas
O que achou sobre criar um jogo 3D utilizando rastreamento de mãos e uma câmara de computador normal? Descreva a sua experiência no minijogo Damas 3D	R1 – Achei a ideia interessante! Só teve o problema com a profundidade, mas considerando os avanços que vão ser efetuados nas câmaras dos portáteis, este problema poderá ser reduzido ou até mesmo eliminado futuramente.
	R2 – A camara não conseguia detetar a profundidade eficientemente, dificultando o posicionamento das peças, passado para baixo do tabuleiro a posição da mão
	R3 – Interessante e com bastante potencial para transferir jogos de tabuleiro tradicionais para um modelo imersivo online.
	R4 – Os outros 2 jogos de <i>hand-tracking</i> funcionaram relativamente bem só tendo falhas quando mexia as mãos demasiado rápido. O Damas 3D foi o único dos que joguei que não consegui testar direito porque a câmara não tinha precisão suficiente para mexer uma peça.

Na decima quinta pergunta perguntou-se a opinião sobre ter jogos que precisão de reação rápida, usando comando de voz e, parecido com a pergunta anterior, pediu-se a experiência dos participantes com os jogos de Pong. E como podemos analisar na Tabela 17, a maioria das respostas reclama que as vezes o comando demora a sair e que as vezes precisa de prever para onde a bola vai para, tornado assim um jogo de previsão, mas também os participantes acharam a ideia interessante se não tivesse essa questão da demora do STT a processar o que foi dito

pelo utilizador. Ainda dando um destaque ao comentário R2 onde o participante destacou a ideia de utilizar o rastreamento de mãos juntamente com o comande voz em jogos.

Tabela 17 - Tabela de respostas: “O que achou sobre criar jogos que precisa de ações rápidas utilizando comando de voz?”

Questão	Respostas
O que achou sobre criar jogos que precisa de ações rápidas utilizando comando de voz? Descreva a sua experiência nos minijogos de Pong utilizando o comando de voz	R1 – Acho bem, mas eram lentos a responder.
	R2 – Divertido e, numa futura versão, seria interessante usar a tecnologia de <i>hand-tracking</i> para uma versão alternativa.
	R3 – Embora seja uma ideia interessante, é muito difícil orientar a raquete para onde a bola está a dirigir por voz, sem mencionar o problema técnico, que foi supramencionado.
	R4 – O tempo de registar uma resposta também como o tempo entre registar várias respostas não é rápido suficiente para criar jogos que precisam um nível elevado de rapidez e precisão
	R5 – Caso o jogo capte o microfone, creio que seja uma boa mecânica para ser implementada em jogos futuros.
	R6 – O tempo de espera entre falar e o jogo processar a mudança torna este tipo de jogos difíceis de jogar, mas consigo ver a tecnologia a ser utilizada melhor noutros tipos de jogos.
	R7 - Acho que deveria ser usado para jogos sem limite de tempo já que o tempo de ação é muito demorado.
	R8 - O <i>voice command</i> é bastante lento na identificação da instrução, dificultando a jogabilidade dos jogos, principalmente em ações rápidas.

Na decima sexta pergunta, teve como objetivo em analisar o potencial da utilização de jogos que utilizam STT, TSS e feedback sonoro para pessoas com problemas visuais. Então foi pedido aos participantes para jogarem o minijogo da memória sem olharem para o ecrã, só jogando através dos sons oferecidos pelo jogo. E como podemos analisar na Tabela 18, os quatro participantes que fizeram esse desafio acharam uma boa ideia e que sim, este tipo de era uma

grande valia para pessoas com problemas visuais para que consigam interagir com a aplicação com mais naturalidade. Ainda houve um participante que destacou a vantagem destas tecnologias serem usadas por pessoas com limitações motoras, já que este tipo de interação elimina a necessidade de usar o teclado ou o rato, sendo que no minijogo da memória, só faltou encaixar um TTS para as instruções, como dito na resposta R3, para que o minijogo fosse totalmente possível ser jogado por pessoas com deficiências visuais.

Tabela 18 - Tabela de respostas: "Achas que este os jogos utilizarem comando de voz seriam ótimos para as pessoas com problemas visuais?"

Questão	Respostas
Pergunta para quem fez o desafio de jogar o minijogo da memória sem olhar para o ecrã Achas que este os jogos utilizarem comando de voz seriam ótimos para as pessoas com problemas visuais? Dê um pouco da sua visão, quando descreve a sua experiência com o minijogo da Memória	R1 – Sim, a tecnologia <i>text to speech</i> ajuda muito as pessoas com problemas visuais
	R2 – Acho que é uma boa ideia. Na prática o <i>voice</i> deveria ser feito por pessoas para ficar mais fácil de reconhecer as palavras. Acho também que deveria de ser apenas uma palavra em vez de 2 como aconteceu com o caso de "o sol" que estava a registar palavras distintas pois pensava que era apenas uma. Tive problemas também a reconhecer algumas vezes que falei, que dava erro de não reconhecer o que foi dito.
	R3 – Após mais adaptações como um tutorial adaptado sim seria interessante para alguém com problemas visuais
	R4 – Sim, este tipo de jogos é realmente muito acessível para pessoas com problemas visuais. Ao testar, tornou-se um pouco cansativo pelo tipo de jogo que é, mas em nenhum momento tive de olhar para o ecrã. Para além disso, como também não usa rato ou teclado, pode ser jogado por pessoas com diversas limitações, incluindo deficiências motoras, o que o torna ainda mais inclusivo.
	R5 – Não fiz sem olhar para o ecrã, mas por vezes não entendia o que era falado, não registando a resposta.

Já nas conclusões finais, na décima sétima e decima oitava foi perguntado se os participantes achavam que a interações apresentadas podiam ser úteis nos jogos, onde os 8 participantes responderam "Sim" em ambas as perguntas, mostrando um grande interesse na utilização destas tecnologias em possíveis jogos futuros.

Na decima oitava pergunta, já foi perguntado o que tinha chamado atenção para os utilizadores expressarem o que gostaram mais. E como podemos ver na Tabela 19, temos o R2 a falar que a mistura de rastreamento de mãos com um ambiente 3D aumenta o número de possibilidades, não só para o entretenimento como para a indústria. O R3 e R4 descreve que existem diversos benefícios e ao utilizar estes sistemas, criando jogabilidades inovadoras e que também ajuda na acessibilidade para que todas as pessoas possam ter a mesma experiência. E o R5 realça o potencial das tecnologias e como a fusão das duas tecnologias podiam criar jogos mais complexos (dando o exemplo da batalha naval).

Tabela 19 - Tabela de respostas: “O que mais lhe chamou a atenção no protótipo?”

Questão	Respostas
O que mais lhe chamou a atenção no protótipo?	R1 – As damas em 3D especialmente a parte de misturar o <i>hand-tracking</i> num ambiente 3D abre a porta a muitas possibilidades tanto para entretenimento como para outras indústrias.
	R2 – A ideia é bastante boa e existem diversos benefícios de usar <i>tanto hand-tracking</i> como <i>voice tracking</i> já no mercado direcionados normalmente a uma jogabilidade nova e inovadora. Ser possível usar para auxílio de certas condições é também muito útil e usado num jogo bom é apenas ganhar para todas as pessoas.
	R3 – A acessibilidade aos jogos por parte de ambos os sistemas. A navegação pelos menus e respetiva jogabilidade é intuitiva por parte de ambos os sistemas, podendo ser uma mais-valia para quem tem limitações.
	R4 – Potencial visto que a tecnologia está estabelecida permitindo expansão para mais jogos ou mesmo fusão de ambas as tecnologias em jogos mais complexos estilo batalha naval.

Na decima nona pergunta, pediu a opinião dos participantes o potencial do uso das tecnologias usadas nos protótipos terem potencial para os jogos e como podemos ver na Figura 50, os participantes ficaram divididos entre “Médio” e “Bastante”, o que não acaba por ser mau sinal, mas que mostra que ainda há muito para melhorar no protótipo, principalmente a velocidade de reconhecimento do STT, para que os utilizadores tenham melhores experiências ao jogar este tipo de jogos.

**EM TERMOS DAS TECNOLOGIAS IMPLEMENTADAS CONSIDERA TEREM
POTENCIAL PARA USO EM JOGOS?**

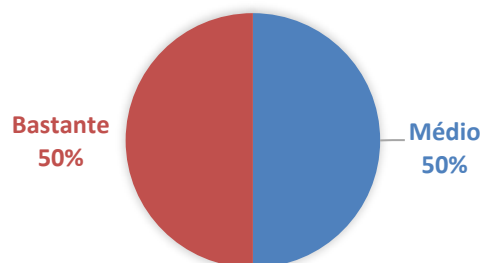


Figura 50 - Gráfico de respostas: "Em termos das tecnologias implementadas considera terem potencial para uso em jogos?"

Na vigésima pergunta, igualmente no primeiro teste, perguntou-se onde que os participantes achavam que esta tipo de tecnologias podia ser aplicada em quais áreas. E na Figura 51 podemos ver que os resultados já foram um pouco diferentes dos primeiros testes (ver Figura 39) onde aqui 100% das pessoas escolheram "Reabilitação Médica" e a "Educação" ficou só com 5 dos 8 votos possíveis, mostrando que a análise do primeiro teste estava correta, onde os participantes tinham sido influenciados com os minijogos apresentados.

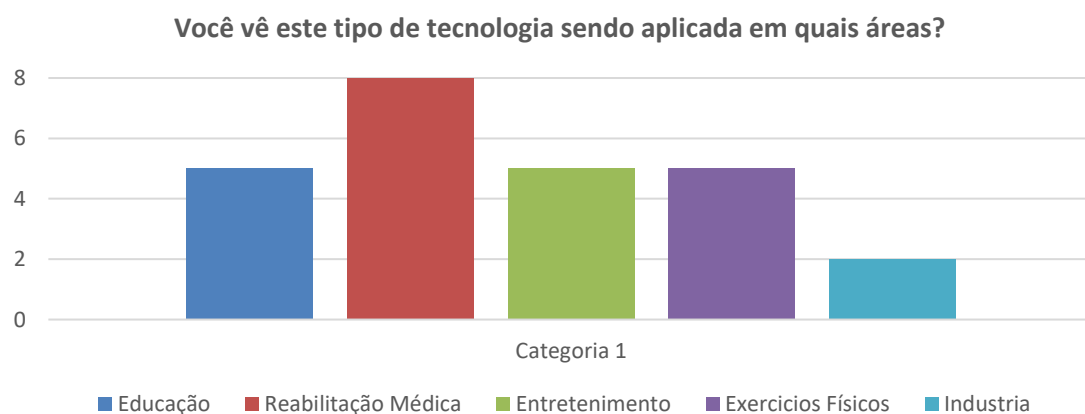


Figura 51 - Gráfico de respostas: "Você vê este tipo de tecnologia sendo aplicada em quais áreas?"

Na vigésima primeira e na vigésima segunda pergunta, pediu-se aos participantes para surgirem melhorias técnicas e outras melhorias em geral e, como podemos ver na Tabela 20, a grande parte fala em melhorar o comando de voz, para ter uma melhor detecção e ser mais rápido, como já comentando nas perguntas anteriores, e também otimizar o rastreamento das mãos,

para não depender da iluminação do ambiente para a localização da mão. Fora isso, tem uma sugestão de trocar o design entre os jogos para ter uma melhor variedade visual, o que se entende este comentário, já que nestes protótipos não se focou nada no UI e sim só no UX, e por fim novamente a ideia da junção das duas tecnologias.

Tabela 20 - Tabela de respostas: “Quais melhorias técnicas você sugeria para otimizar a experiência?” e “Sugestões de melhoria”

Questão	Respostas
Quais melhorias técnicas você sugeria para otimizar a experiência?	R1 – ser mais responsivo.
	R2 – Fazer com que o <i>voice command</i> entenda melhor a palavra a ser mencionada.
	R3 – Tentar ter menor tempo de resposta por parte do jogo, de modo que o jogador não fique prejudicado.
	R4 – Aumentar a velocidade que reconhece as palavras!
	R5 – Otimizar o tracking das mãos, visto que por vezes o <i>tracking</i> perdesse. Já os <i>voice commands</i> deveriam ser mais rápidos e deveríamos poder dizer apenas o número, sem dizer explicitamente "opção"
	R7 - Junção de ambas as tecnologias em novos jogos.
Sugestões de melhoria	R6 – Para além da <i>voice command</i> , se calhar outra melhoria que poderia ser feita, era o ambiente de fundo diferir entre jogos para dar maior variedade visual.

Na vigésima terceira e vigésima quarta pergunta pediu-se aos participantes darem uma nota de 0 a 10 das suas experiências com os protótipos (onde 0 é “Muito Insatisfatório” e 10 é “Muito Satisfatório”. E como podemos ver na Figura 52, o rastreamento de mãos apresentou avaliações díspares, com uma pontuação média de 5,17 (na escala 0-10), situando-se em patamar aceitável. Em contraste, o comando de voz registou uma média de 7,13/10, valor superior ao esperado. Apesar do comando de voz ter dado mais comentários negativos, teve uma pontuação media superior devia ao fato de as pessoas estarem mais habituadas a usarem aplicações que utilização algum tipo de comando de voz, como por exemplo a assistente do google em Androids, ou até mesmo a Siri em iOS, mas já não terem assim tanto contacto com aplicações que usam algum tipo de rastreamento, fazem assim que as mesmas pessoas, não tenham conhecimento prévio do funcionamento dessa tecnologia, acabando por ter algumas dificuldades em usar. Outro motivo acaba também por ser por haver mais aplicações no

mercado que utilizam comando de voz do que utilizam algum tipo de rastreamento, fazendo com que as tecnologias de voz estejam mais avançadas e acessíveis para as pessoas, do que as tecnologias usadas para o rastreamento de mãos. Este teste acaba por ser um exemplo disso, onde os 8 participantes tiveram microfones para testarem os minijogos, mas só 5 participantes tinham câmaras, e sendo que um deles teve problemas com o sistema e outro tinha uma câmara antiga que não acaba por ser a melhor opção para este tipo de jogos, sobrando assim 3 participantes com câmaras boas, mas ainda entra a questão que um desses tinha a iluminação fraca para o teste, aumentando mais uma vez o problema com a utilização deste sistema.



Figura 52 - Gráfico de respostas: "De 0 a 10, quanto daria a sua experiência a utilizar o Rastreamento de Mãos/Comando de Voz para jogos?"

Na última pergunta, solicitou-se aos participantes que escolhessem a tecnologia preferida. A maioria optou pelo comando de voz, o que sugere que, apesar das limitações identificadas, esta tecnologia é atualmente mais acessível e familiar para os utilizadores, facilitando a sua adoção em jogos e aplicações.

COMPARAÇÃO GERAL: QUAL CONTROLO PREFERES?

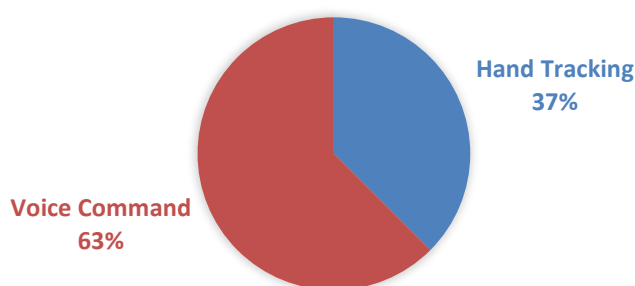


Figura 53 - Gráfico de respostas: "Comparação geral: Qual controlo preferes?"

6.2.3 Conclusão do 2.º Teste de Utilizadores

O segundo teste de utilizadores, embora de dimensão reduzida, forneceu informações complementares e relevantes para o desenvolvimento do protótipo. Em termos gerais, os resultados apontam para um padrão consistente: o comando de voz foi mais bem aceite e melhor avaliado pelos participantes (média 7,13/10), enquanto o rastreamento de mãos apresentou avaliações mais díspares (média 5,17/10) e problemas técnicos em alguns casos. A percepção média foi de 3,1 (escala 1-5), indicando um sentimento moderado de domínio sobre os sistemas e evidenciado a necessidade de melhorias na experiência do utilizador para aumentar a confiança.

O teste revelou também que, apesar da maior aceitação do comando de voz, este continua sujeito a latência em situações concretas: vários participantes referiam atrasos perceptíveis no sistema de reconhecimento de voz (STT), o que compromete a fluidez da interação. Por sua vez, o rastreamento de mãos, mostrou-se sensível às condições ambientais, como: fraca luminosidade e câmaras de baixa resolução reduziram a precisão da deteção, sobretudo na inferência de profundidade a partir de imagens 2D, aumentando assim a variabilidade das avaliações. A fadiga física foi mencionada por alguns participantes nos minijogos que exigem manter a mão elevada por períodos prolongados, o que sugere a necessidade de repensar a duração e o ritmo desses jogos ou de incluir pausas programadas.

A principal limitação deste teste é o reduzido tamanho da amostra (apenas 8 participantes), que torna as conclusões exploratórias e impede a generalização para o público-alvo. Além disso, a heterogeneidade do hardware dos participantes (tipos de câmara, qualidade do microfone, iluminação) introduz variabilidade que dificulta a atribuição inequívoca das causas dos problemas observados, já que não existem dados suficientes para perceber se é um problema com o utilizador em específico ou do equipamento/ambiente em que foi testado.

Apesar destas limitações, o segundo teste foi fundamental para identificar problemas reais (latência do STT, limiares de deteção), validar hipóteses de usabilidade (preferência pelo comando de voz) e orientar intervenções de melhoria (tutoriais, otimização do STT). Os resultados fornecem uma base prática para melhorias concretas no protótipo e para o planeamento de estudos subsequentes mais controlados e representativos.

7 Conclusão e Trabalho Futuro

Neste capítulo, são referidas conclusões do desenvolvimento do projeto, trabalho futuro e apreciações finais.

7.1 Planeamento final

Ao longo do decurso do projeto surgiram alterações ao nível do planeamento, sendo a principal a extensão da sua duração relativamente ao indicado na seção 1.4.2. Inicialmente, o projeto teria início a 3 de fevereiro de 2025 e término a 30 de julho de 2025 (total de 128 dias, sem contar com os fim-de-semanas). Na prática, o desenvolvimento começou a 10 de fevereiro de 2025 e terminou a 28 de setembro de 2025, totalizando um total de 165 dias (mais 37 dias do que o planeado inicialmente).

Este acréscimo de duração possibilitou um maior tempo de desenvolvimento dos protótipos, permitindo a criação de mais minijogos e o seu refinamento (correção de problemas que poderiam comprometer a experiência do utilizado), bem como um período de testes de utilizadores mais alargado e uma análise mais cuidada dos dados recolhidos.

Outra alteração relevante foi das reuniões com o orientador: no planeamento inicial, estava prevista uma reunião a fim de cada fase e, durante a fase de desenvolvimento, encontros quinzenais. Contudo, optou-se por reuniões presenciais semanais, de forma a permitir um acompanhamento mais próximo do progresso e a receção das orientações contínuas, sendo que este formato durou até finais de junho, onde a partir daí o projeto já teria encaminhado e não existiria a necessidade de reuniões semanais.

Outra alteração consistiu na realização de dois testes com utilizadores: um no início do desenvolvimento e outro no final. Inicialmente estava previsto apenas um teste no final do desenvolvimento, a inclusão de um teste inicial revelou-se muito útil, pois análises dele decorrentes permitiram verificar se as configurações iniciais eram adequadas e esclarecer várias dúvidas de UI/UX. Essas informações orientaram decisões de design que contribuíram para melhorar a usabilidade e a experiência global do protótipo.

Na Figura 54 apresenta-se o resumo do cronograma final do projeto.

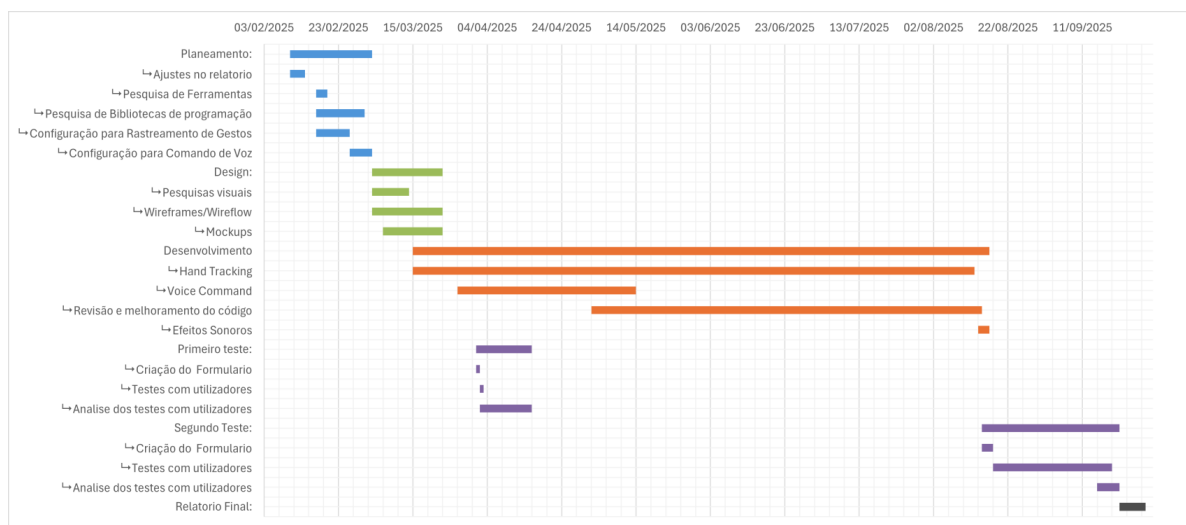


Figura 54 - Cronograma Final

7.2 Trabalhos Futuros

Apesar dos protótipos funcionais incluírem 8 minijogos baseados em rastreamento de mãos e 4 minijogos por comando de voz, permanecem várias linhas de investigação e melhorias técnicas/promocionais por explorar. Abaixo discute-se cada uma delas e propõem-se atividades experimentais e técnicas concretas.

7.2.1 Junção dos dois sistemas

A integração simultânea de rastreamento de mãos e comando de voz mostrou-se uma ideia promissora (apontada por um participante no segundo teste). Em jogos híbridos, por exemplo no *Pong*, em que a posição da raquete é controlada por gestos e comandos como “pausa” e “sair” por voz, a combinação pode aumentar a naturalidade e expressividade da interação.

Mas, antes dessa adição, é necessário ter atenção aos seguintes pontos de investigação e riscos técnicos:

- **Performance:** em máquinas antigas, a execução simultânea dos dois sistemas (rastreamento + STT) pode causar latência e consumo elevado de CPU/GPU. Seria necessária a realização de testes de perfilamento de performance, assim como a aplicação de estratégias de otimização (processamento em *threads* separados, redução de resolução/cadência de captura, uso de inferência em GPU/TPU quando disponível).
- **Gestão de conflitos:** definir regras de prioridade (quando o gesto e voz ocorrem ao mesmo tempo) e fornecer feedback multimodal para evitar ações indesejadas.

- **Mitigação:** oferecer modos configuráveis, por exemplo: “combinado”, “apenas voz” ou “apenas gestos”, para se adaptar às capacidades do hardware e às necessidades do utilizador. Para isso, podem ser realizados testes de desempenho (*benchmarks*) em hardware de três classes (baixo, médio e alto), medindo FPS, latência de input e uso de CPU/RAM.

7.2.2 Multijogador local com comando de voz

Criar um minijogo multijogador local por voz é viável e traz uma dinâmica social diferente, apesar do risco de um jogador atrapalhar o outro e da latência do STT. Em jogos mais simples, por exemplo um jogo do galo, onde cada jogador diz a sua jogada na sua respetiva vez, pode funcionar bem e oferecer uma interação alternativa entre os jogadores.

7.2.3 Multijogador online

Adicionar um modo online permitiria, por exemplo no jogo de Damas 3D com rastreamento de mãos, jogadores em residências diferentes jogarem entre si, ou até mesmo o *Pong* com comando de voz. Esta adição aumenta a componente social sem a necessidade de presença física. Um exemplo recente de jogo multijogador com comando de voz, é o ***Mage Arena***, que, durante o desenvolvimento deste projeto, encontra-se em acesso antecipado; nesse jogo, os jogadores controlam um mago numa arena (podendo ser 1 contra 1, ou em equipas até 4 jogadores cada) e utilizam STT para lançar magias (por exemplo: dizendo “*Fire Ball*” para lançar uma bola de fogo). (Wikipédia, 2025) (jrsjams, 2025)

7.2.4 Projeto dirigido a público desfavorecidos (acessibilidade)

Desenvolver minijogos pensados desde o início para utilizadores com défice visual, mobilidade reduzida ou outras necessidades, como foi analisado no segundo teste de utilizadores.

Por exemplo, o minijogo da memória já permitiu que alguns participantes jogassem sem precisar olhar para o ecrã. Esta experiência pode ser ampliada complementando as instruções com TTS, adicionando feedback sonoro para ações (por exemplo um som específico quando o jogador diz “Jogar”), lendo o que o STT interpretou e emitindo avisos explícitos quando o reconhecimento falha.

Além disso, pessoas com deficiência motora podem beneficiar do rastreamento de mãos ou de outros rastreamentos (por exemplo: rastreamento dos olhos), que removem a necessidade de usar teclado e rato para controlar ações nos jogos.

Devem ser realizados estudos com utilizadores reais com deficiência visual/motora, avaliando medidas de acessibilidade (efetividade, eficiência e satisfação) e o cumprimento de *guidelines* (por exemplo das WCAG, como mencionadas na seção 2.1).

7.2.5 Integração com assistentes virtuais

A utilização de dispositivos como a Alexa para controlar o jogo é uma interação interessante. A Alexa pode ser combinada com serviços de terceiros, como o *TriggersCMD*, que permite configurar e controlar ações no computador por voz. O *TriggersCMD* também possibilita pré-configurar comandos personalizados (através de ficheiros .bat ou .sh), que podem corresponder aos comandos do jogo (por exemplo: “Jogar ou “Sair”) ou às palavras que aparecem, por exemplo, no minijogo da memória. Esta integração adiciona uma mecânica alternativa de interação com os jogos. (Braga, Pedro, & Lima, 2023) (Augusto, 2020)

A utilização de assistentes virtuais traz alguns riscos, como latência adicional entre o assistente e o jogo, questões de autorização e segurança entre o assistente e o computador e a necessidade de suportar *cross-platform* (Windows/Linux/Mac). No entanto, pode melhorar significativamente a experiência do utilizador, como por exemplo: numa sala de aula o docente ou aluno poderia controlar um jogo educativo sem a necessidade de se deslocar até ao computador, projetando a imagem do ecrã, com o projetor, e colocando uma Alexa Echo Dot numa posição estratégica.

7.2.6 Exploração de STT e TTS alternativos

Investigar STT/TTS alternativos, incluindo versões pagas da API da Google (Speech-to-Text V1/V2) e outras soluções para analisar se tem uma melhor precisão e latência para melhorar a experiência dos jogadores. Estas APIs mencionadas, oferecem opções de residência de dados para multirregiões (e região única no caso da V2) e, no caso da V2, permitem o envio de resultados parciais para acelerar a interpretação dos comandos e tornar a interação mais dinâmica. (Google, 2025)

7.2.7 Escrita no ar

Explorar a possibilidade de criar um sistema de escrita no ar utilizando câmaras comuns dos portáteis. Utilizando, por exemplo, o minijogo de pintura, onde, no modo livre, uma pessoa pode escrever algumas palavras, e integrando uma IA capaz de leitura de escrita em imagens, poderia criar-se uma aplicação simples de escrita no ar. Poderia ser utilizado, por exemplo, o reconhecimento ótico de caracteres (OCR) da Google que, mediante um custo associado, tem a capacidade de reconhecer o texto presente em imagens e convertê-lo em texto codificado por uma máquina (OCR, 2025).

Esta ideia inspira-se no projeto de dissertação de 2012 do Mestrado em Engenharia Informática do ISEP, realizado por Pedro Leitão, no qual foi explorada a existência de uma interface que captava a caligráfica da escrita no ar (Leitão, 2012). Nesse projeto foi utilizado um Microsoft Kinect e a proposta aqui seria utilizar câmaras comuns dos portáteis, sendo que é mais comum uma pessoa ter uma portátil com câmara do que uma Kinect, aproveitando o minijogo de pintura e integrando-o com o OCR da Google, possibilitando outra variedade de jogos, como por exemplo: um jogo que peça ao utilizador para escrever uma palavra (como o seu nome) e em que a IA converta a escrita para texto e a guarde no sistema.

Na Figura 55 podemos ver um exemplo de como seria a integração do minijogo de pintura com o OCR.

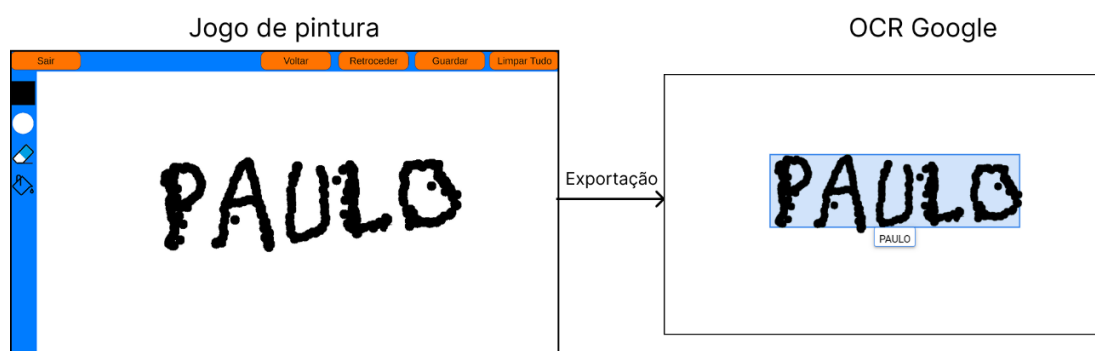


Figura 55 - Exemplo de integração da OCR para escrita no ar

7.3 Escrita científica e evolução nos estudos

Uma das linhas de continuidade natural deste trabalho passa pela divulgação científica dos resultados obtidos. A curto prazo, prevê-se a preparação de um ou dois artigos:

- Um artigo técnico descrevendo a engenharia dos protótipos (por exemplo: arquitetura Python–Unity, pipeline de hand-tracking e integração STT/TTS, métricas de desempenho), dirigido a conferências ou periódicos de interação humano-computador e jogos;
- Um artigo empírico focado nos resultados dos testes com utilizadores, análise da usabilidade e recomendações de design para acessibilidade em jogos. Estes dois tipos de publicações complementares permitirão tanto partilhar soluções técnicas como avaliar empiricamente o impacto das abordagens propostas.

A médio/longo prazo, a continuação desta linha de investigação pode evoluir para um projeto de doutoramento na área de Sistemas de Interação Avançados. Um doutoramento poderia aprofundar temas como:

- Métodos híbridos de interação (voz + gesto) robustos a ruído e variações ambientais;
- Adaptações personalizadas (DDA) para necessidades específicas de reabilitação cognitiva;
- Estudos longitudinais que avaliem ganhos cognitivos e de qualidade de vida. O doutoramento contribuiria não só para consolidar os resultados já obtidos, mas também para explorar com maior rigor experimental e clínico as aplicações terapêuticas e educativas destas tecnologias.

7.4 Conclusão final

Este projeto teve como objetivo analisar a utilização de sistemas alternativos, nomeadamente rastreamento de mãos e comando de voz, em jogos, com o intuito de potenciar benefícios cognitivos e terapêuticos e de reduzir o sedentarismo associado a jogos que exigem teclado e rato. Ao longo do desenvolvimento foram construídos protótipos que demonstraram a viabilidade destas abordagens em contextos lúdicos e educativos, mostrando que mesmo hardware comum, como um portátil, permite experiências interessantes sem necessidade de investimento elevado.

Os protótipos realizados cobriram desde de jogos simples, como um quiz para crianças, até propostas mais complexas em 3D. Esses protótipos evidenciaram que estas tecnologias podem tornar os jogos mais inclusivos e atrativos para públicos diversos (crianças, idosos ou utilizadores que procuram alternativas de interação), contribuindo tanto para o entretenimento como para aplicações pedagógicas e reabilitadoras. Apesar de os testes não terem contando com um número de participantes suficiente para análises estatísticas robustas,

os resultados qualitativos apontam para um interesse consistente dos utilizadores em experimentar jogos controlados por voz e gestos.

Os testes também permitiram identificar diferenças práticas entre as tecnologias: o comando de voz revela-se, atualmente, mais acessível na generalidade dos equipamentos, devido à ubiquidade de microfones integrados em telemóveis, auriculares, portáteis, entre outros. Por outro lado, o rastreamento de mãos exige normalmente uma câmara de qualidade e condições de iluminação, o que limita a sua adoção generalizada. Estas constatações sinalizam caminhos claros para trabalhos futuro, nomeadamente medidas de otimização, soluções nos dispositivos dos utilizadores e melhorias na robustez em ambientes ruidosos ou com pouca luz.

Para além do domínio dos jogos, as ferramentas estudadas têm um vasto campo de aplicação. O STT e TSS já estão a ser integrados em diversas aplicações, como assistentes virtuais (assistente da Google em smartphones), plataformas educativas (Duolingo) e experiências imersivas. Estas tecnologias podem ser combinadas com RA e modelos de IA para cenários comerciais, como provadores virtuais que permitem selecionar peças com gestos e visualizar a roupa aplicada a um avatar ou mesmo na imagem do cliente. Exemplos de marcas e experiências comerciais mostram que estas ideias estão a emergir no mercado, como a Burberry com sistemas RA (Burberry Group plc, 2022) e a Chanel com experiências multissensoriais (Hunger, 2022), reforçam o potencial de transferência das soluções aqui investigadas para contextos reais.

Existem, contudo, limitações técnicas relevantes: a latência de sistemas STT, as sensibilidades ao ruído ambientem a exigência de boa iluminação e de câmaras com resolução adequada para o rastreamento de mãos. Estes obstáculos são superáveis com investigação adicional, por exemplo: através de técnicas de supressão de ruído, uso de modelos com processamento parcial para reduzir latência, otimização do pipeline de visão e estratégias de design de interação que toleram imprecisões. Além disso, questões de privacidade e de segurança dos dados de voz e imagem devem sempre ser abordadas nas implementações para a segurança dos utilizadores.

Em resumo, este trabalho demonstra que rastreamento de mãos e comando de voz são alternativas promissoras para diversificar as formas de interação em jogos, com benefícios potencial para educação, reabilitação, bem-estar físico e novas experiências de entretenimento. A investigação realizada fornece uma base prática e experimental que justifica estudos mais amplos e protótipos mais refinados, e sugere que, num futuro próximo, estas tecnologias poderão tornar-se cada vez mais comuns e integradas no quotidiano dos utilizadores.

Referências

- Adobe. (2025). *Adobe Photoshop*. Obtido em 12 de Setembro de 2025, de <https://www.adobe.com/pt/products/photoshop.html>
- Adobe. (2025). *Adobe Color*. (Adobe) Obtido em 05 de Setembro de 2025, de <https://color.adobe.com/pt/>
- Adobe. (2025). *Adobe Illustrator*. Obtido em 12 de Setembro de 2025, de Adobe: <https://www.adobe.com/products/illustrator.html>
- Ajraou, H. (Março de 2025). *virtual_hand*. Obtido em 05 de Setembro de 2025, de Github: https://github.com/houdeyfa/virtual_hand
- Amazon. (s.d.). *Amazon Lex – Criador de bate-papo com IA*. Obtido em 23 de Dezembro de 2024, de aws.amazon: <https://aws.amazon.com/pt/lex/>
- Amazon. (s.d.). *Amazon Polly – Gerador de voz com IA*. (Amazon) Obtido em 23 de Dezembro de 2024, de <https://aws.amazon.com/pt/polly/>
- Apple. (s.d.). *Speech*. (Apple) Obtido em 23 de Dezembro de 2024, de documentation: <https://developer.apple.com/documentation/speech>
- Augusto, M. (16 de Junho de 2020). *Como controlar o computador pela Alexa*. Obtido em 23 de Setembro de 2025, de Medium: <https://medium.com/@matheusasg09/controlando-computador-com-a-alexa-d61a08e672ca>
- Braga, L., Pedro, W., & Lima, L. (Maio de 2023). *Como usar a Alexa no PC com Windows*. Obtido em 23 de Setembro de 2025, de tecnoblog: <https://tecnoblog.net/responde/como-usar-a-alexa-no-pc-com-windows/>
- Burberry Group plc. (07 de Junho de 2022). *Burberry launches new augmented reality shopping tool to celebrate the Lola bag*. Obtido em 25 de Setembro de 2025, de Burberry: 2022
- Byford, S. (06 de Setembro de 2019). *Seaman creator Yoot Saito on the fishy Dreamcast AI that was way ahead of its time*. Obtido em 28 de Dezembro de 2024, de TheVerge: <https://www.theverge.com/2019/9/6/20850674/yoot-saito-interview-seaman-sega-dreamcast-ai-20th-anniversary>
- CARVALHO, M. A., & ROLAND, C. d. (2019). *Uso da tecnologia de comandos de voz para domótica*. Artigo científico, Brasil. Obtido em 27 de Dezembro de 2024, de <https://ric.cps.sp.gov.br/handle/123456789/4955>

- Ceccon, D. (14 de Agosto de 2020). *Nova IA reconhece gestos com alta precisão*. Obtido em 21 de Dezembro de 2024, de <https://iaexpert.academy/2020/08/14/nova-ia-reconhece-gestos-com-alta-precisao>
- Chaudhary, A., Raheja, J., & Das, K. (2010). *A Survey on Hand Gesture Recognition in Context of Soft Computing*. India. Obtido em 21 de Dezembro de 2024, de https://www.researchgate.net/publication/215580014_A_Survey_on_Hand_Gesture_Recognition_in_Context_of_Soft_Computing
- Chung, A., Griffin, A., Selezneva, D., & Gotz, D. (2018). *Health and Fitness Apps for Hands-Free Voice-Activated Assistants: Content Analysis*. Casimiro Dias. doi:10.2196/mhealth.9705
- CMUSphinx. (28 de Dezembro de 2023). *OPEN SOURCE SPEECH RECOGNITION TOOLKIT*. (CMUSphinx) Obtido em 23 de Dezembro de 2024, de <https://cmusphinx.github.io/>
- Dan. (19 de Fevereiro de 2023). *Lifeline – Guide and Walkthrough*. Obtido em 28 de Dezembro de 2024, de GameFAQs: <https://gamefaqs.gamespot.com/ps2/589786-lifeline/faqs/29092>
- Esperança, C., Sousa, N., Costa, S., Pereira, I., & Francisco, M. (2021). *ACESSIBILIDADE WEB E DIGITAL*. Eletrónico, Politécnico de Leiria, Leiria. Obtido em 20 de Dezembro de 2024, de https://www.researchgate.net/publication/349442493_ACESSIBILIDADE_WEB_E_DIGITAL
- Fernandez, J. A., Lee, J. J., Vacca, S. A., Magana, A., Pes̃a, R., Benes, B., & Popescu, V. (2024). *Hands-Free VR*. Estados Unidos da America. Obtido em 27 de Dezembro de 2024, de <https://arxiv.org/abs/2402.15083>
- Figma, Inc. (s.d.). *What is Figma?* Obtido em 12 de Setembro de 2025, de <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>
- GitHub, Inc. (2025). *About GitHub Desktop*. Obtido em 12 de Setembro de 2025, de <https://docs.github.com/en/desktop/overview/about-github-desktop>
- GitHub, Inc. (2025). *github about*. Obtido em Setembro de 2025 de 2025, de 12
- GitHub, Inc. (2025). *About Git*. Obtido em 12 de Setembro de 2025, de <https://git-scm.com/about>
- Gonçalves, H. (2022). *Estudo do hábito para a reabilitação cognitiva suportado em jogos sérios*. Dissertação de Mestrado, Escola Superior de Média Arte e Design, Sistemas e Media Interativos, Porto. Obtido em 28 de Setembro de 2025, de <https://recipp.ipp.pt/entities/publication/49bb4173-c0ac-4a3e-8b73-0b760ae78123>

Google. (13 de Janeiro de 2025). *Guia de detecção de pontos de referência manuais*. Obtido em 28 de Agosto de 2025, de Google AI for Developers: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker?hl=pt-br

Google. (2025). *Transforme voz em texto com a IA do Google*. Obtido em 24 de Setembro de 2025, de Cloud Speech-to-Text: <https://cloud.google.com/speech-to-text>

Google. (s.d.). *Cloud Speech-to-Text*. (Google) Obtido em 23 de Dezembro de 2024, de cloud.google: <https://cloud.google.com/speech-to-text>

Google. (s.d.). *Google Forms*. Obtido em 12 de Setembro de 2025, de <https://workspace.google.com/products/forms/>

Google. (s.d.). *IA da Text-to-Speech*. (Google) Obtido em 23 de Dezembro de 2024, de <https://cloud.google.com/text-to-speech>

HAPI. (2015). *Mobility, Universal Design, Health, and Place*. Obtido em 19 de Dezembro de 2024, de https://research.gsd.harvard.edu/hapi/files/2015/11/HAPI_ResearchBrief_UniversalDesign-112315.pdf

Hotjar Ltd. (09 de Outubro de 2023). *Teste de usabilidade: uma introdução detalhada*. Obtido em 29 de Setembro de 2025, de Hotjar: <https://www.hotjar.com/pt-BR/teste-de-usabilidade/>

Hoy, M. B. (2018). *Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants*. Estados Unidos da América. doi:10.1080/02763869.2018.1404391

Hunger. (15 de Dezembro de 2022). *Chanel's immersive new exhibition, Le Grand Numéro de Chanel, is the perfect experience for fragrance lovers this festive season*. Obtido em 25 de Setembro de 2025, de Hunger: <https://huntermag.com/editorial/channels-immersive-new-exhibition-le-grand-numero-de-chanel-is-the-perfect-experience-for-fragrance-lovers-this-festive-season>

IBM. (s.d.). *IBM Watson Speech to Text*. (IBM) Obtido em 23 de Dezembro de 2024, de <https://www.ibm.com/products/speech-to-text>

IBM. (s.d.). *IBM Watson Text to Speech*. (IBM) Obtido em 23 de Dezembro de 2024, de <https://www.ibm.com/products/text-to-speech>

illume, & pygameci. (29 de Setembro de 2024). *pygame 2.6.1*. Obtido em 25 de Setembro de 2025, de pypi.org: <https://pypi.org/project/pygame/>

jrsjams. (15 de Setembro de 2025). *Mage Arena*. Obtido em 23 de Setembro de 2025, de Steam: https://store.steampowered.com/app/3716600/Mage_Arena/

Kaldi. (s.d.). *Kaldi*. (Kaldi) Obtido em 23 de Dezembro de 2024, de <https://kaldi-asr.org/doc/>

Khalaf1, A. S., Alharthi, S. A., Alshehri, A., Dolgov, I., & Dugas, P. O. (2020). *A Comparative Study of Hand-Gesture*. Estados Unidos da America. Obtido em 22 de Dezembro de 2024, de <https://par.nsf.gov/servlets/purl/10174270>

Kojić, T., Vergari, M., Knuth, S., Warsinke, M., Möller, S., & Voigt-Antons, J.-N. (2024). *Influence of Gameplay Duration, Hand Tracking, and Controller Based Control Methods on UX in VR*. Artigo, Technical University of BerlinGermany, Quality and Usability Lab, Alemanha. Obtido em 22 de Dezembro de 2024, de <https://arxiv.org/abs/2404.03337>

Leitão, P. M. (2012). *Interface Caligráfica de Escrita no Ar*. Dissertação de mestrado, Porto. doi:<http://hdl.handle.net/10400.22/4502>

Masurovsky, A., Chojecki, P., Runde, D., Lafci, M., Przewozny, D., & Gaebler, M. (2020). *Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study*. Artigo, Alemanha. Obtido em 22 de Dezembro de 2024, de <https://www.mdpi.com/2414-4088/4/4/91>

Mattey, C., & Yuksel, B. (2018). *OverVoice: Voice Controlled Games*. University of San Francisco, San Francisco, CA 94117, USA. Obtido em 28 de Dezembro de 2024, de https://www.cs.usfca.edu/~byuksel/affectivecomputing/2018_examples/chai.pdf

Microsoft. (04 de Novembro de 2021). *Gaming for Everyone: The Accessibility Features of Forza Horizon 5*. (M. Brown, Editor) Obtido em 18 de Setembro de 2025, de Xbox Wire: <https://news.xbox.com/en-us/2021/11/04/gaming-for-everyone-the-accessibility-features-of-forza-horizon-5/>

Microsoft. (24 de Outubro de 2024). *IntelliCode*. Obtido em 12 de Setembro de 2025, de <https://marketplace.visualstudio.com/items?itemName=VisualStudioExptTeam.vscod-eintellicode>

Microsoft. (29 de Abril de 2025). *Visual Studio Code*. Obtido em 12 de Setembro de 2025, de <https://learn.microsoft.com/en-us/shows/visual-studio-code/>

Microsoft. (2025). *What is Clipchamp?* Obtido em 12 de Setembro de 2025, de <https://support.microsoft.com/en-au/topic/what-is-clipchamp-750e8940-cd76-4abf-9767-b34d3d3285d7>

Microsoft. (s.d.). *Azure AI Speech*. (Microsoft) Obtido em 23 de Dezembro de 2024, de <https://azure.microsoft.com/en-us/products/ai-services/ai-speech>

Microsoft. (s.d.). *Azure AI Speech*. (Microsoft) Obtido em 23 de Dezembro de 2024, de <https://azure.microsoft.com/en-us/products/ai-services/ai-speech/>

Microsoft. (s.d.). *Speech service documentation*. (S. s. documentation, Produtor, & Microsoft)
 Obtido em 23 de Dezembro de 2024, de <https://learn.microsoft.com/en-us/azure/ai-services/speech-service/>

Morais, C. T., Andrade, D. A., Sena, P. M., Segundo, W. L., & Amaro, B. (2023). *ACESSIBILIDADE EM WEBSITES DA COMUNICAÇÃO*. Obtido em 20 de Dezembro de 2024, de <https://cip.brapci.inf.br/download/227678>

Nair, V., Karp, J. L., Silverman, S., Kalra, M., Lehv, H., Jamil, F., & Smith, B. A. (2021). *NavStick: Making Video Games Blind-Accessible via the Ability*. Columbia University, New York.
 Obtido em 20 de Dezembro de 2024, de <https://arxiv.org/pdf/2109.01202>

National Society of Professional Engineers. (s.d.). *NSPE Code of Ethics for Engineers*. Obtido em 16 de Dezembro de 2024, de <https://www.nspe.org/resources/ethics/code-ethics>

OCR. (20 de Setembro de 2025). Obtido em 27 de Setembro de 2025, de Google Cloud: <https://cloud.google.com/use-cases/ocr>

Olhar Digital. (21 de 08 de 2019). *Algoritmo que rastreia mão pode reconhecer linguagem de sinais*. (Olhar Digital) Obtido em 21 de Dezembro de 2024, de Olhar Digital: https://olhardigital.com.br/2019/08/20/noticias/algoritmo-que-rastreia-mao-pode-reconhecer-linguagem-de-sinais/?utm_source=chatgpt.com

OpenAI. (30 de 11 de 2022). *Conheça o ChatGPT*. Obtido em 12 de Setembro de 2025, de <https://openai.com/pt-BR/index/chatgpt/>

Pallets. (19 de Agosto de 2025). *Flask 3.1.2*. Obtido em 25 de Setembro de 2025, de pypi.org: <https://pypi.org/project/Flask/>

Pereira, A. P., Lourenço, C., Boato, E., Ribeiro, E., Reis, H. S., Ramalho, H., . . . Felizardo, S. (2019). *Apontamentos de Educação Especial e Inclusiva*. Viseu, Portugal: Escola Superior de Educação de Viseu (ESEV). Obtido em 15 de Novembro de 2024, de https://repositorio.ipv.pt/bitstream/10400.19/6923/1/Ebook_ApontamentosEE-2019.pdf

pn durette. (10 de Novembro de 2024). *gTTS 2.5.4*. Obtido em 25 de Setembro de 2025, de pypi.org: <https://pypi.org/project/gTTS/>

Politécnico do Porto. (2020). *Código de Boas Práticas e de Conduta*. Despacho, Politécnico do Porto, Porto. Obtido em 16 de Dezembro de 2024, de <https://www.ipp.pt/comunidade/menu-comunidade/missao-equidade-diversidade-inclusao/DespachoP.PORTOP0402020CodigodeBoasPraticasedeConduta.pdf/view>

Proxima Enterprises Inc. (s.d.). *Suck Up!* Obtido em 09 de Setembro de 2025, de <https://www.playsuckup.com/>

Python Software Foundation. (18 de Setembro de 2025). *Python About*. Obtido em 21 de Setembro de 2025, de Python: <https://www.python.org/about/>

Python Software Foundation. (25 de Setembro de 2025). *socket — Low-level networking interface*. Obtido em 25 de Setembro de 2025, de Python docs: <https://docs.python.org/3/library/socket.html>

Realidade Virtual. (s.d.). *O que é rastreamento de mãos em realidade virtual?* (Realidade Virtual) Obtido em 21 de Dezembro de 2024, de <https://programae.org.br/virtual/glossario/o-que-e-rastreamento-de-maos-em-realidade-virtual/>

SenyaeV, A., Asmorkalov, Sergregory, & Skvark. (07 de Julho de 2025). *opencv-python 4.12.0.88*. Obtido em 25 de Setembro de 2025, de pypi: <https://pypi.org/project/opencv-python/>

Shrestha, A., & Karki, A. (2022). *Accessibility in Video Games: A Review*. Eletronico, Nepal College of Information Technology, Balkumari. Obtido em 21 de Dezembro de 2024, de https://www.researchgate.net/publication/360457520_Accessibility_in_Video_Games_A_Review

Silva, S. R. (2017). *Reconhecimento de Gestos Customizados da mão em tempo real usando aprendizado de métricas e grafos de ação*. Universidade Federal de Alagoas Instituto de Matemática, Alagoas. Obtido em 21 de Dezembro de 2024, de <https://www.repositorio.ufal.br/bitstream/riufal/2257/1/Reconhecimento%20de%20gestos%20customizados%20da%20m%C3%A3o%20em%20tempo%20real%20usando%20aprendizado%20de%20m%C3%A9tricas%20e%20grafos%20de%20a%C3%A7%C3%A3o.pdf>

Silveira, M. I. (27 de Outubro de 2024). *Usabilidade e acessibilidade em produtos digitais e físicos*. Obtido em 28 de Setembro de 2025, de Alura: <https://www.alura.com.br/artigos/usabilidade-acessibilidade>

Simões, L. E., Rodrigues, L. B., Silva, R. M., & da Silva, G. R. (2024). *Evaluating Voice Command Pipelines for Drone Control: From STT and LLM to Direct Classification and Siamese Networks*. Brasil. doi:<https://arxiv.org/abs/2407.08658>

Tabnine. (s.d.). *tabnine*. Obtido em 12 de 09 de 2025, de <https://www.tabnine.com/>

Torres, Y. (27 de Maio de 2025). *O que é usabilidade e por que ela é tão importante?* Obtido em 28 de Setembro de 2025, de Uol host: <https://uolhost.uol.com.br/blog/o-que-e-usabilidade/>

Truyenque, M. A. (2005). *Uma Aplicação de Visão Computacional*. Pontifícia Universidade Católica do Rio de Janeiro, DEPARTAMENTO DE INFORMÁTICA, Rio de Janeiro. Obtido em 21 de Dezembro de 2024, de <https://web.tecgraf.puc-rio.br/press/publication/Truyenque2005a/Truyenque2005a.pdf>

United Nations. (2016). *Convention on the Rights of Persons with Disabilities (CRPD)*. Obtido em 19 de Dezembro de 2024, de <https://social.desa.un.org/issues/disability/crpd/convention-on-the-rights-of-persons-with-disabilities-crpd>

Unity Technologies. (2025). *Unity Our Company*. Obtido em 21 de Setembro de 2025, de Unity: <https://unity.com/pt/our-company>

Unity Technologies. (s.d.). *Unity*. (Unity Technologies) Obtido em 23 de Março de 2025, de unity: <https://unity.com/pt>

Vaz, R. (14 de Janeiro de 2025). *O que são stakeholders e por que eles são importantes para as empresas?* Obtido em 28 de 09 de 2025, de docusign: <https://www.docusign.com/pt-br/blog/stakeholders>

Vizdx LLC. (31 de Março de 2024). *CVZone*. Obtido em 25 de Setembro de 2025, de Github: <https://github.com/cvzone/cvzone>

W3C. (12 de Dezembro de 2024). *Web Content Accessibility Guidelines (WCAG) 2.1*. Obtido em 19 de Dezembro de 2024, de <https://www.w3.org/TR/WCAG21>

Wikipédia. (09 de Outubro de 2021). *Milestone*. Obtido em 22 de Setembro de 2025, de Wikipédia: <https://pt.wikipedia.org/wiki/Milestone>

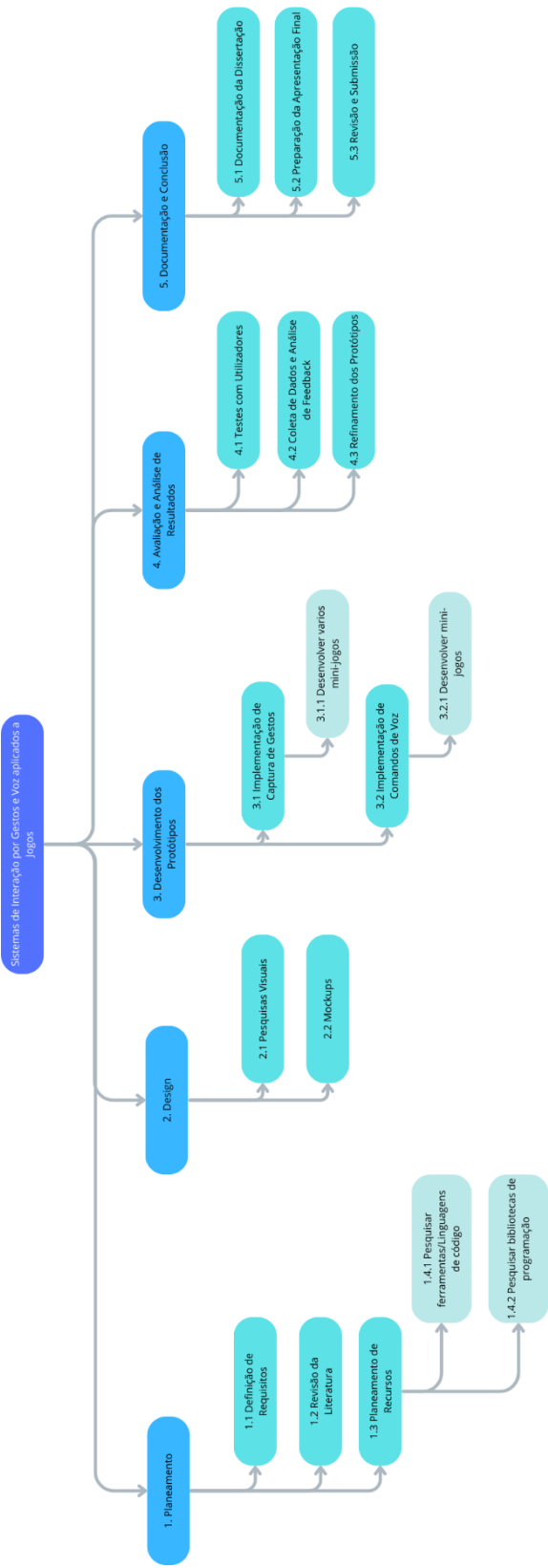
Wikipédia. (29 de Agosto de 2025). *Mage Arena*. Obtido em 23 de Setembro de 2025, de Wikipédia: https://en.wikipedia.org/wiki/Mage_Arena

Wikipedia.org. (22 de 10 de 2023). *Distância Levenshtein*. Obtido em 07 de Setembro de 2025, de Wikipedia: https://pt.wikipedia.org/wiki/Dist%C3%A2ncia_Levenshtein

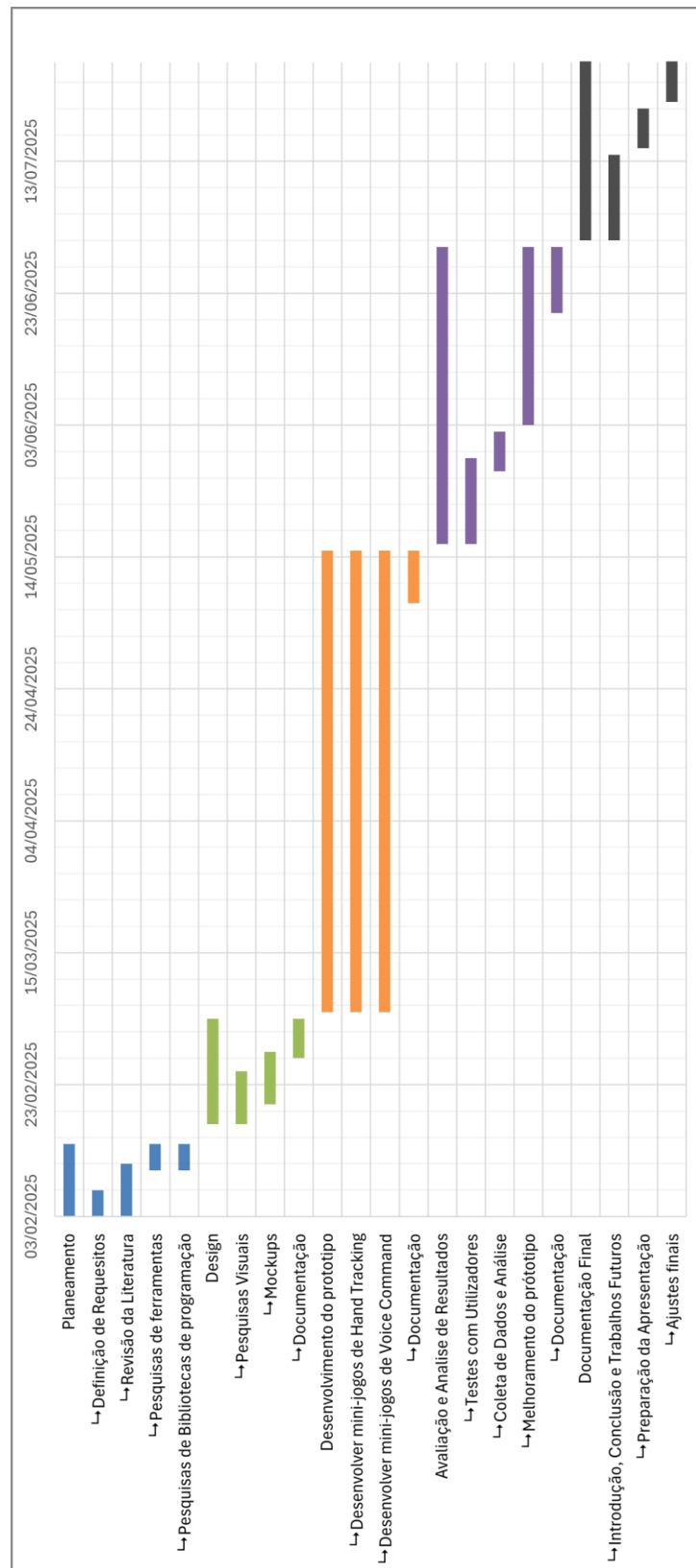
Zargham, N., Friehs, M. A., Tonini, L., Alexandrovsky, D., Ruthven, E. G., Nacke, L. E., & Malaka, R. (2023). *Let's Talk Games: An Expert Exploration of Speech Interaction with NPCs*. Research Article, International Journal of Human–Computer Interaction, Alemanha. Obtido em 28 de Dezembro de 2024, de <https://www.tandfonline.com/doi/full/10.1080/10447318.2024.2338666>

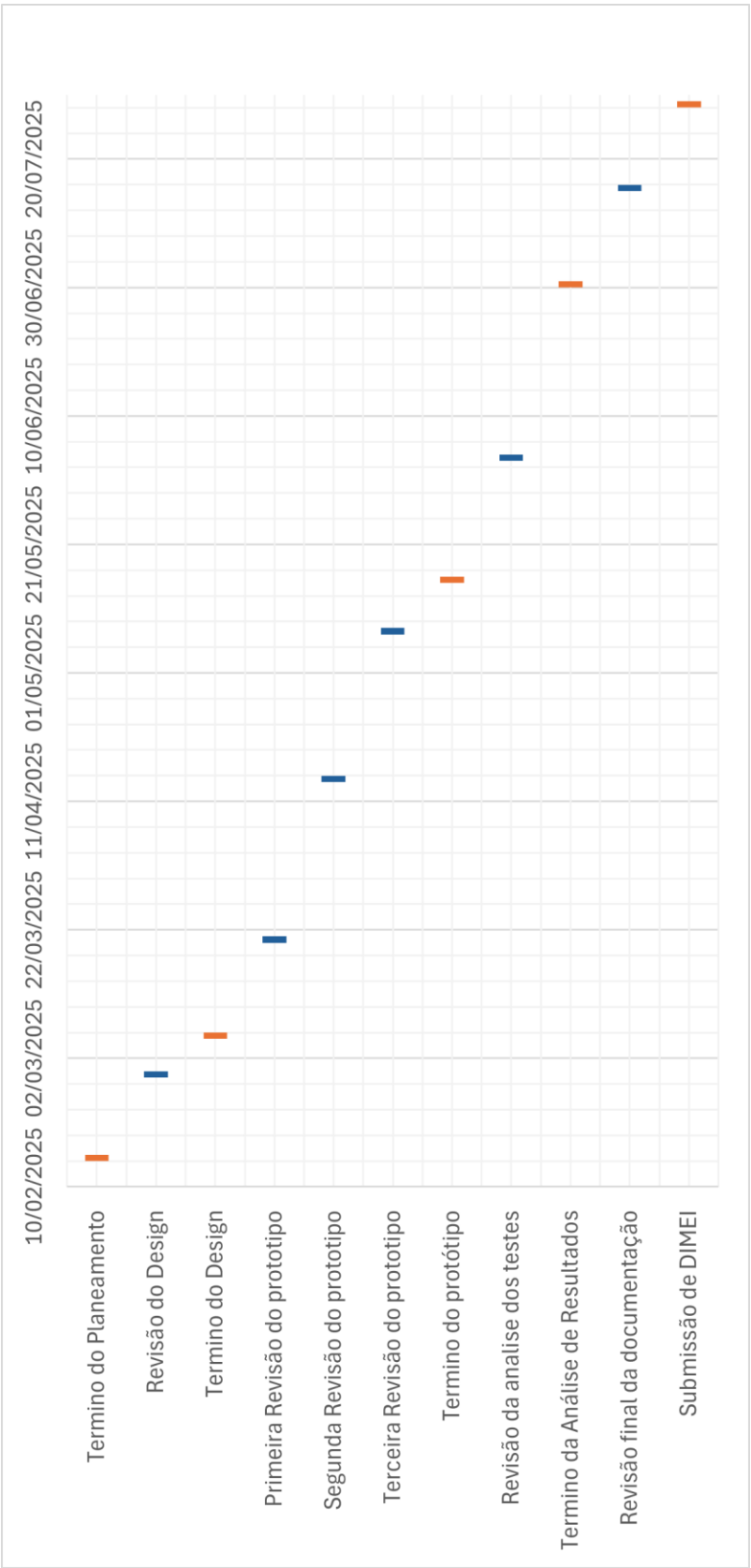
Zhang, A., Chembarpu, A., Smith, K., Hadenes, K., Braden, S., Turkynewych, B., . . . maverickagm. (13 de Maio de 2025). *SpeechRecognition 3.14.3*. Obtido em 05 de Setembro de 2025, de Python Package Index (PyPI): <https://pypi.org/project/SpeechRecognition/>

Anexo A - WBS



Anexo B – Cronograma Inicial





Anexo C – Analise de Risco

Projeto: Sistemas de Interação por Gestos e Voz aplicadas a Jogos

ID de risco	Descrição	Causa	Efeito	Proprietário do Risco	Probabilidade (1-5)	Impacto (1-5)	Pontuação PI	Resultado esperado, nenhuma ação	Tipo de resposta ao risco	Descrição da resposta
	Descrição do risco	Causa do risco	Efeito no projeto	Nome da pessoa que monitoriza o risco	O grupo obteve uma estimativa aproximada da probabilidade de tal	Estimativa aproximada de quão significativo é o impacto deste risco	Probabilidade multiplicada pelo impacto	O que acontecerá se o risco se tornar um problema e não foram tomadas medidas	Decisão tomada pelo grupo sobre como responder a este risco	Como saber que está na hora de colocar a resposta em prática
1	Dificuldade na integração de gestos e voz em jogos	Limitações das tecnologias de rastreamento e reconhecimento disponíveis.	Atrasos no desenvolvimento do protótipo e resultados insatisfatórios	Paulo Rodrigues	4	5	20	O protótipo não será funcional ou apresentará desempenho inferior, impactando os objetivos da dissertação	Mitigação	Iniciar testes preliminares com tecnologias disponíveis, identificar mais viáveis e ajustar escopo conforme
2	Falta de padrões claros para jogos com foco em gestos e voz	Maioria dos jogos não incorporam sistemas de comando de voz ou rastreamento de mãos	Curva de aprendizado maior e necessidade de desenvolver soluções próprias.	Paulo Rodrigues	3	4	12	O projeto pode não atingir os resultados desejados devido a limitações no design ou implementação.	Aceitação	Compensar a falta de padrões com criatividade e foco nos objetivos específicos do projeto
3	Dificuldade na realização de testes com utilizadores	Disponibilidade limitada dos participantes ou falta de recursos para organizar os testes	Atrasos na avaliação da solução e falta de dados para validar o protótipo.	Paulo Rodrigues	3	4	12	Sem os testes, os resultados do projeto podem ser questionáveis ou incompletos	Mitigação	Preparar testes alternativos (ex: online) ou ampliar a base de participantes com convites abertos e
4	Atrasos por parte dos utilizadores na realização dos testes do protótipo	Falta de comprometimento ou dificuldades na agenda dos utilizadores recrutados	O cronograma do projeto pode ser comprometido, atrasando a análise de resultados e decisões	Paulo Rodrigues	4	3	12	Resultados insuficientes ou atrasos podem comprometer a conclusão da dissertação dentro do prazo.	Mitigação	Estabelecer prazos claros e comunicar a importância dos testes. Oferecer lembretes e incentivos para incentivar a participação dentro do
5	Impossibilidade de reunir com orientador em momentos cruciais	Agenda cheia do orientador ou imprevistos	Decisões importantes podem ser atrasadas, impactando o cronograma do projeto	Paulo Rodrigues e Antonio Castro	3	3	9	O projeto pode perder o alinhamento com as expectativas do orientador ou ficar paralisado.	Mitigação	Estabelecer reuniões regulares com antecedência.

Anexo D – 1º Formulário

DIMEI

Este questionário faz parte do primeiro teste do protótipo desenvolvido para a cadeira de DIMEI, no âmbito do projeto "*Sistemas de Interação por Gestos e Voz aplicados a Jogos*". O objetivo é avaliar a experiência de utilização de hand tracking e voice command em minigames educativos e explorar possíveis melhorias e aplicações futuras.

As suas respostas serão essenciais para compreender a usabilidade, acessibilidade e potencial desta tecnologia, ajudando a direcionar os próximos desenvolvimentos. O questionário leva apenas alguns minutos e todas as respostas são anónimas.

Obrigado pela sua participação!

* Indica uma pergunta obrigatória

A interação por gestos por ser útil no jogos?

- ☐ Sim
- ☐ Não

A interação por audio pode ser útil nos jogos?

- ☐ Sim
- ☐ Nao

Gênero *

- ☐ Masculino
- ☐ Feminino
- ☐ Outro

Idade

- ☐ menos de 20
- ☐ Entre 20 e 30
- ☐ Acima de 30

Formação acadêmica (Maior frequentada)

- ☐ 12º ano
- ☐ Licenciatura
- ☐ Mestrado
- ☐ Doutorado

Já trabalhou anteriormente com tecnologias de hand tracking ou voice command * antes?

- ☐ Sim
- ☐ Não

Qual o seu nível de familiaridade com estas tecnologias? *

- ☐ Iniciante
- ☐ Intermediário
- ☐ Avançado

Como foi a sua experiência ao jogar os minigames? *

- ☐ Muito fácil
- ☐ Fácil
- ☐ Médio
- ☐ Difícil
- ☐ Muito difícil

Você sentiu algum atraso ou imprecisão ao usar gestos para jogar? Se sim, pode descrever?

A sua resposta

O tempo de resposta do jogo foi adequado? *

- ☐ Sim
- ☐ Não - devia ser mais pequeno
- ☐ Não - devia ser maior

O controle por gestos pareceu intuitivo? *

- ☐ Sim
- ☐ Não
- ☐ Parcialmente

Você acha que essa tecnologia pode ser melhorada para tornar a interação mais natural? *

- ☐ Sim
- ☐ Não

Que desafios técnicos você identificou na usabilidade do jogo?

A sua resposta

Nas instruções, achas que devia ficar só texto a explicar a funcionamento do jogo, ou devia ter um gif/vídeo explicativo *

- ☐ Texto já é suficiente
- ☐ Era melhor um gif/vídeo explicativo
- ☐ A junção dos dois, um gif/vídeo explicativo com um texto acompanhar

Você vê este tipo de tecnologia sendo aplicada em quais áreas? (Selecione todas que achar relevantes) *

- ☐ Educação
- ☐ Reabilitação médica
- ☐ Entretenimento
- ☐ Exercícios físicos
- ☐ Outra:

O que mais lhe chamou a atenção no protótipo?

A sua resposta

Quais melhorias técnicas você sugeriria para otimizar a experiência?

A sua resposta

Você teria interesse em testar versões futuras com novas mecânicas, como por exemplo, com voice comand ou outros minijogos? *

- ☐ Sim
- ☐ Não

Em termos das tecnologias implementadas considera terem potencial para uso em jogos? *

- ☐ Pouco
- ☐ Medio
- ☐ Bastante
- ☐ Muito

O tempo de respostas é adequado?

- ☐ Sim
- ☐ Não
- ☐ Podia ser mais rapido
- ☐ Podia ser mais lento

A barra de progresso no jogo foi útil? *

- ☐ Sim
- ☐ Nao

As regras e instruções foram claras? *

- ☐ Sim
- ☐ Nao

Sugestoes de melhoria

A sua resposta

Anexo E – 2º Formulário

DIMEI

Este questionário faz parte do segundo teste do protótipo desenvolvido para a cadeira de DIMEI, no âmbito do projeto "Sistemas de Interação por Gestos e Voz aplicados a Jogos". O objetivo é avaliar a experiência de utilização de hand tracking e voice command em minijogos e explorar possíveis melhorias e aplicações futuras.

As suas respostas serão essenciais para compreender a usabilidade, acessibilidade e potencial desta tecnologia, ajudando a direcionar os próximos desenvolvimentos. O questionário leva apenas alguns minutos e todas as respostas são anónimas.

Obrigado pela sua participação!

ATENÇÃO: Algumas das respostas poderão ser utilizadas durante a apresentação do projeto!

* Indica uma pergunta obrigatória

Informações pessoais

Aqui pedimos algumas informações pessoais, para saber a faixa etária, género e conhecimento sobre os sistema usado para fim de compreensão melhor dos possíveis resultado.
Não é pedido nome/email ou qualquer coisa que tire o anonimato para podem responder 100% a vontade

Gênero *

- ☐ Masculino
- ☐ Feminino
- ☐ Outro

Idade *

- ☐ <20
- ☐ 20-29
- ☐ 30-39
- ☐ 40-49
- ☐ >50

Formação académica (Maior frequentada) *

- ☐ 12º ano
- ☐ Licenciatura
- ☐ Mestrado
- ☐ Doutoramento
- ☐ Outra:

Ambiente e dispositivo

Dispositivo utilizado no teste *

- ☐ Portátil com câmara integrado
- ☐ Portátil com microfone integrado
- ☐ Portátil com câmara externa
- ☐ Portátil com microfone externo
- ☐ Computador com câmara externa
- ☐ Computador com microfone externo
- ☐ Outra:

Condições de iluminação do ambiente

- ☐ Muito bom (Condições ideais)
- ☐ Bom
- ☐ Médio
- ☐ Fraco
- ☐ Muito fraco (Condições péssimas)

Condições de ruído do ambiente

- ☐ Muito bom (condições ideais)
- ☐ Bom
- ☐ Médio
- ☐ Fraco
- ☐ Muito fraco (condições péssimas)

Distância aproximada entre o utilizador e o computador/portátil

- ☐ <0.5 m
- ☐ 0.5 - 1 m
- ☐ > 1m

Familiaridade com o sistema

Já trabalhou anteriormente com tecnologias de hand tracking ou voice command * antes?

- ☐ Sim
- ☐ Sim, mas só Voice command
- ☐ Sim, mas só Hand Tracking
- ☐ Não

Qual o seu nível de familiaridade com a tecnologia Hand Tracking? *

	1	2	3	4	5	
Iniciante	☐	☐	☐	☐	☐	Avançado

Qual o seu nível de familiaridade com a tecnologia Voice Command? *

	1	2	3	4	5	
Iniciante	☐	☐	☐	☐	☐	Avançado

Quais foram os jogos jogados durante o teste?

Hand Tracking *

- ☐ Quiz Game
- ☐ Pong Solo
- ☐ Pong Multi
- ☐ Jogo das Cores
- ☐ Jogo da Reação
- ☐ Jogo da Matemática
- ☐ Paint
- ☐ Damas 3D
- ☐ Nenhum porque tive problemas com o sistema
- ☐ Nenhum, não tenho câmara

Voice Command

- ☐ Quiz Game
- ☐ Pong Solo
- ☐ Pong Duelo
- ☐ Jogo da Memória
- ☐ Nenhum porque tive problemas com o sistema
- ☐ Nenhum, não tenho microfone

Experiência com os protótipos

Como foi a sua experiência ao jogar os minijogos Hand Tracking?

- ☐ Muito fácil
- ☐ Fácil
- ☐ Médio
- ☐ Difícil
- ☐ Muito difícil

Como foi a sua experiência ao jogar os minijogos Voice Command?

- ☐ Muito fácil
- ☐ Fácil
- ☐ Médio
- ☐ Difícil
- ☐ Muito difícil

O tempo de resposta do jogo foi adequado? *

- ☐ Sim
- ☐ Não - devia ser mais pequeno
- ☐ Não - devia ser maior

Sentiu-se confiante a usar os sistemas?

- | | | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Muito pouco confiante | ☐ | ☐ | ☐ | ☐ | ☐ | Muito confiante |

Conforto físico — Depois de jogar, sentiste desconforto físico (cansaço/ombros/pesçoço/olhos)?

- ☐ Não
- ☐ Um pouco
- ☐ Moderado
- ☐ Muito
- ☐ Outra:

Você sentiu algum atraso ou imprecisão para jogar em algum dos protótipos? Se sim, pode descrever?

A sua resposta

O controle por gestos pareceu intuitivo?

- ☐ Sim
- ☐ Não
- ☐ Parcialmente

O controle por voz pareceu intuitivo?

- ☐ Sim
- ☐ Não
- ☐ Parcialmente

Você acha que essa tecnologia pode ser melhorada para tornar a interação mais natural? *

- ☐ Sim
- ☐ Não

As regras e instruções foram claras? *

- ☐ Sim
- ☐ Sim, mas só no Hand Tracking
- ☐ Sim, mas só no Voice Command
- ☐ Não

Nas instruções dos minijogos do Hand Tracking: achou útil ter um gif explicativo a mostrar como se interage com o jogo?

- ☐ Sim, foi muito útil
- ☐ Sim, mas precisa de um melhoria para ter uma compreensão mais clara
- ☐ Sim, mas o texto já era o suficiente
- ☐ Não, a compreensão não foi muito clara

Que desafios técnicos você identificou na usabilidade do jogo?

A sua resposta

O que achou sobre criar um jogo 3D utilizando Hand Tracking e uma câmara de computador normal? Descreva a sua experiência no minijogo Damas 3D

A sua resposta

O que achou sobre criar jogos que precisa de ações rápidas utilizando Voice command? Descreva a sua experiência nos minijogos de Pong utilizando o Voice Command

A sua resposta

Pergunta para quem fez o desafio de jogar o minijogo da memória sem olhar para o ecrã

Achas que este os jogos utilizarem voice command seriam ótimos para as pessoas com problemas visuais? Dê um pouco da sua visão, quando descreve a sua experiência com o minijogo da Memória

A sua resposta

Acessibilidade

Esta seção é opcional, se não sentir confortável em responder, pode passar para a próxima

Tens alguma limitação que afetou a experiência (motora, visual, auditiva, outra)?

- ☐ Sim
- ☐ Não
- ☐ Prefiro não dizer

Se "Sim": Que tipo?

- ☐ Motora
- ☐ Visual
- ☐ Auditiva
- ☐ Cognitiva
- ☐ Outra:

O protótipo foi acessível para ti?

- ☐ Sim
- ☐ Parcialmente
- ☐ Não

Conclusões Finais

A interação por gestos por ser útil nos jogos?

- ☐ Sim
- ☐ Não

A interação por áudio pode ser útil nos jogos?

- ☐ Sim
- ☐ Não

Pergunta só para quem fez o primeiro teste

Durante este novo teste, sentiu melhoria significativas comparado com o teste feito em Abril?

- ☐ Sim
- ☐ Sim, mas não muito significativas
- ☐ Não

O que mais lhe chamou a atenção no protótipo?

A sua resposta

Em termos das tecnologias implementadas considera terem potencial para uso em jogos? *

- ☐ Pouco
- ☐ Médio
- ☐ Bastante
- ☐ Muito

Você vê este tipo de tecnologia sendo aplicada em quais áreas? (Selecione todas que achar relevantes) *

- ☐ Educação
- ☐ Reabilitação médica
- ☐ Entretenimento
- ☐ Exercícios físicos
- ☐ Indústria
- ☐ Outra:

Quais melhorias técnicas você sugeria para otimizar a experiência?

A sua resposta

Sugestões de melhoria

A sua resposta

De 0 a 10, quanto daria a sua experiencia a utilizar o Hand Tracking para jogos?



De 0 a 10, quanto daria a sua experiencia a utilizar o Voice Command para jogos?



Comparação geral: Qual controlo prefere? *

- ☐ Hand Tracking
- ☐ Voice Command
- ☐ Ambos
- ☐ Nenhum

Anexo F – Instruções do 2º Teste

Instruções — Teste de Utilizadores

Projeto: Sistemas de Interação por Gestos e Voz aplicados a Jogos (Disciplina: Dissertação — Mestrado em Engenharia Informática: Jogos, Sistemas Gráficos e Interativos)

Objetivo

Obrigado por participar neste teste. O objetivo é avaliar protótipos que utilizam rastreamento de mãos (hand tracking) e comandos de voz (voice command) para jogos, com foco em usabilidade e acessibilidade.

Conteúdo do pacote

Após descarregar e descompactar a pasta, irá encontrar duas subpastas:

- **Hand Tracking** — jogos que utilizam rastreamento de mãos (requer câmara do PC/portátil).
- **Voice Command** — jogos que utilizam comandos de voz (requer microfone do PC/portátil).

Dentro de cada pasta, o ficheiro que inicializa os jogos é **DIMEI.exe** — execute esse ficheiro para arrancar os protótipos.

Requisitos mínimos

- Sistema operativo: **Windows** (recomendado Windows 10 ou 11).
- Câmara integrada ou externa funcional (para jogos de Hand Tracking).
- Microfone funcional (para jogos de Voice Command).

Antes de começar — permissões e aviso do Windows

1. Ao iniciar o .exe, o Windows pode apresentar um aviso do tipo "**O Windows protegeu o seu computador**". Isto acontece porque se trata de um protótipo e o computador ativa o sistema de segurança para ver se tem certeza se querem abrir a aplicação. Para continuar: clique em "**Ver mais informação**" e depois em "**Executar mesmo assim**".
2. Na primeira execução o Unity (e o Python utilizado pelos protótipos) pode pedir permissões para aceder à câmara e/ou microfone. Conceda

"Permitir"; caso contrário, o jogo não terá acesso aos periféricos e não será possível jogar.

3. Se a câmara ou microfone estiverem a ser usados por outra aplicação (por exemplo, Zoom, Teams, Skype), feche essas aplicações antes de arrancar o protótipo. (Em princípio isso não deve dar problemas)

Passos para executar

1. Descarregue e descompacte a pasta com o protótipo.
2. Abra a pasta **Hand Tracking** ou **Voice Command** consoante o tipo de teste pretendido.
3. Execute **DIMEI.exe**.
4. Siga as instruções no ecrã e conceda as permissões solicitadas (câmara/microfone).
5. Feche outras aplicações que possam aceder à câmara/microfone, se necessário.

Se surgir algum erro inesperado, anote a mensagem de erro e o momento em que ocorreu para incluir no formulário de feedback.

Quais jogos deve testar (obrigatório)

Para garantir que conseguimos recolher respostas relevantes, pedimos que testem os seguintes minijogos obrigatórios:

- **Hand Tracking: Damas 3D** e mais **1 jogo à sua escolha** dentro da pasta Hand Tracking.
- **Voice Command: Um dos jogos do Pong e o Jogo da Memória.**

Estes 4 minijogos são os mínimos pedidos para poderem responder à maioria das perguntas do questionário. Se puderem testar mais jogos, agradecemos — isso ajuda a obter uma visão mais completa.

Desafio extra: No **Jogo da Memória**, tenta jogar sem olhar para o ecrã, só atrás dos áudio e efeitos sonoros para conseguir responder a uma pergunta no questionário

Prazo

- **Prazo ideal para resposta:** até **05 de setembro de 2025** (recomendado) — para nos permitir analisar com calma as respostas.
- **Data limite de disponibilidade do teste:** **07 de setembro de 2025.**

Link para o questionário

<https://docs.google.com/forms/d/e/1FAIpQLSdrNz1xTQecUJak70tTbB9pkGOvIcOuKCs62ekk2bm3ffaEYQ/viewform?usp=dialog>

Reportar problemas / Contacto

Se encontrar problemas técnicos graves, por favor anote exatamente o erro e em que momento ocorreu e entre em contacto connosco através do e-mail:

, ou se no caso tiver alguma forma direta de contactar o aluno responsável por este projeto, priorizem esse método.

Notas finais e agradecimento

Muito obrigado por dedicar tempo a testar estes protótipos. O seu feedback é essencial para melhorar a usabilidade e acessibilidade dos sistemas.

Tempo estimado de participação (opcional): 20–60 minutos dependendo do número de minijogos testados + 10-20 minutos a responder ao questionário.

Obrigado! 😊

Anexo G – Cronograma Final

