

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

LUCAS LIPPERT IOCHPE REICHMANN GUERRA

**Interação gestual em jogos educativos
utilizando o sensor de movimentos Kinect**

Trabalho de Conclusão apresentado como
requisito parcial para a obtenção do grau de
Bacharel em Ciência da Computação

Prof^ª. Dr^ª. Luciana Nedel
Orientador

Porto Alegre, julho de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Guerra, Lucas Lippert Iochpe Reichmann

Interação gestual em jogos educativos utilizando o sensor de movimentos Kinect / Lucas Lippert Iochpe Reichmann Guerra. – Porto Alegre: PPGC da UFRGS, 2013.

78 f.: il.

Trabalho de Conclusão (graduação) – Universidade Federal do Rio Grande do Sul. Ciência da Computação, Porto Alegre, BR–RS, 2013. Orientador: Luciana Nedel.

1. Kinect. 2. Jogos eletrônicos educativos. 3. Educação infantil. 4. Vídeo games. I. Nedel, Luciana. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do curso: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“For small creatures such as we the vastness is bearable only through love.”

— CARL SAGAN

AGRADECIMENTOS

Agradeço à família e amigos por tornar esse trabalho possível, especialmente para minha mãe e irmã, ao propiciar o experimento com várias crianças e acima de tudo pelo grande apoio moral que me possibilitou passar por tudo isso.

Agradeço à Prof^ª Luciana Nedel pela sua orientação nesse trabalho.

Agradeço à UFRGS pelo ensino e pelos inúmeros desafios que precisaram ser transpostos para permitir a concepção desse trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	12
RESUMO	13
ABSTRACT	14
1 INTRODUÇÃO	15
2 VISÃO GLOBAL	17
2.1 Interação gestual por meio do Kinect	17
2.2 Jogos educativos	18
3 TRABALHOS RELACIONADOS	19
3.1 Jogos eletrônicos educativos	19
3.1.1 Big Brain Academy: Wii Degree	19
3.1.2 Genomics Digital Lab	21
3.1.3 Ko's Journey	21
3.1.4 Carmen Sandiego	21
3.1.5 The ClueFinders	23
3.1.6 Brain Age: Train Your Brain in Minutes a Day!	23
3.1.7 EcoQuest	23
3.1.8 GCompris	25
3.1.9 Super Solvers: Gizmos & Gadgets!	25
3.1.10 WolfQuest	25
3.1.11 Comentários finais	27
3.2 Jogos que utilizam interação gestual por meio do Kinect	29
3.2.1 Dance Central	29
3.2.2 Sesame Street: Once Upon a Monster	30
3.2.3 The Gunstringer	30
3.2.4 Kinectimals	30
3.2.5 Kinect Sports	32
3.2.6 Forza Motorsport 4	32
3.2.7 Kinect Disneyland Adventures	33
3.2.8 Kinect Adventures	33
3.2.9 Puss in Boots	35

3.2.10	Michael Jackson: The Experience	35
3.2.11	Comentários finais	37
4	DESENVOLVIMENTO DO JOGO	40
4.1	Histórico de desenvolvimento	40
4.2	Decisões de projeto	41
4.2.1	Interação por gestos	41
4.2.2	Elemento educacional	45
4.3	Resumo do funcionamento	48
4.4	Detalhes de implementação	49
4.4.1	Conversor de áudio	50
4.4.2	Rainbow Riders of Kinect	51
5	AVALIAÇÃO DA USABILIDADE DO SISTEMA	55
5.1	Hipóteses	55
5.2	Experimentos	55
6	RESULTADOS	58
6.1	Contexto dos experimentos	61
6.2	Constatações	63
7	CONCLUSÕES	64
	REFERÊNCIAS	66
	ANEXO I	69
	ANEXO II	74

LISTA DE ABREVIATURAS E SIGLAS

3D	Três Dimensões
DNA	Deoxyribonucleic Acid
DS	Dual Screen
FFT	Fast Fourier Transform
GUI	Graphical User Interface
HUD	Heads-up Display
PC	Personal Computer
PSP	Playstation Portable
RGB	Red Green Blue
SDK	Source Development Kit
USB	Universal Serial Bus
VGA	Video Graphics Array

LISTA DE FIGURAS

3.1	Resultado de um teste cerebral em “Big Brain Academy: Wii Degree”. Os 5 atributos são apresentados por meio de um gráfico radial e a nota final e “massa cerebral” do jogador são apresentadas (DESTRUCTOID REVIEW: BIG BRAIN ACADEMY: WII DEGREE, 2007).	20
3.2	Exemplo de desafio em “Big Brain Academy: Wii Degree”. Ao ordenar os números o jogo considera que está avaliando a capacidade de computação do jogador, um dos 5 atributos testados (Matt Casamassina, 2007).	20
3.3	Último estágio do jogo “Genomics Digital Lab”. O jogador deve saber a base de DNA a ser usada para parear com a base incompleta (Denise Deveau, 2010).	21
3.4	Livro com informações matemáticas, proveniente dos ancestrais da tribo de Ko em “Ko’s Journey”. O livro será a referência para qualquer cálculo matemático que venha a ser realizado ao longo do jogo (Scott Laidlaw, 2009).	22
3.5	Receita de um remédio feito para um lobo no jogo “Ko’s Journey”. É necessário misturar os ingredientes na proporção certa, além da quantidade estar de acordo com o peso do lobo (Scott Laidlaw, 2009).	22
3.6	Descrição da cidade de Atenas em “Where in the World is Carmen Sandiego?”, o primeiro jogo da série de Carmen Sandiego. Esse jogo se focou principalmente em apresentar ao jogador diferentes pontos no mundo durante à busca por Carmen Sandiego (Yid Yang, 2003).	23
3.7	Um dos diversos <i>puzzles</i> em “The ClueFinders 4th Grade Adventures: Puzzle of the Pyramid”, o segundo jogo da série The ClueFinders (CLUEFINDERS YEAR 4 ADVENTURES - PUZZLE OF THE PYRAMID, 2013).	24
3.8	Um dos desafios presentes em “Brain Age: Train Your Brain in Minutes a Day!”. O desafio deve ser realizado segurando o DS lateralmente, assim como os outros desafios presentes no jogo, o que difere do uso tradicional do console (BRAIN AGE: TRAIN YOUR BRAIN IN MINUTES A DAY IMAGES, 2006).	24
3.9	Tela do jogo “EcoQuest: The Search for Cetus” que descreve a baleia rei Cetus, o objetivo da busca (ECOQUEST THE SEARCH FOR CETUS, 2012).	25
3.10	6 exemplos de atividades presentes no jogo “GCompris” (Rafael Nink, 2008).	26

3.11	Labirinto de portas presente em “Super Solvers: Gizmos & Gadgets!”. Cada porta apresenta um desafio diferente para o jogador resolver e obter parte do veículo de corrida, o objetivo do jogo (FIVE GAMES THAT NEED TO BE ON WII U!, 2011).	26
3.12	Diagrama do veículo a ser construído em “Super Solvers: Gizmos & Gadgets!” (LET’S PLAY SUPER SOLVERS SERIES BY ICEPOTATO - PART 22: GIZMOS AND GADGETS PT 2, 2007).	27
3.13	Tela do jogo “WolfQuest” no modo <i>multiplayer</i> exibindo a HUD do jogo, com elementos como opções de diálogo, informação do seu e de outros lobos (WOLF QUEST GAME: 10. EPISÓDIO, 2013). . .	29
3.14	Imagem do jogo “Dance Central”, exibindo o formato de sua HUD, contendo exemplos de gestos usados como movimentos de dança, como “Lean Pop”, “Snake” e “Raise The Roof” (DANCE CENTRAL IMAGES, 2013).	30
3.15	Imagem do jogo “Sesame Street: Once Upon a Monster”, com os 3 monstros realizando um gesto que deve ser imitado pelo jogador para continuar (PLAYBOOK: SESAME STREET: ONCE UPON A MONSTER, 2013).	31
3.16	Imagem do jogo “The Gunstringer” apresentando o ambiente estilo velho oeste, a visualização estilo terceira pessoa, o cursor que se usa para selecionar os inimigos a atirar e o número de balas restantes (PREVIEW: THE GUNSTRINGER, 2011).	31
3.17	Uma pista de obstáculos para adestrar seu mascote em “Kinectimals” (Stephen Totilo, 2010).	32
3.18	Ilustração da imitação de movimentos realizada no jogo “Kinect Sports” ao jogar vôlei de praia (KINECT SPORTS PARA XBOX 360, 2013).	33
3.19	Imagem do jogo “Forza Motorsport 4” em sua tentativa de ser realista ao representar a aparência interna e externa dos carros, incluindo espelhos (Malcolm Owen, 2011).	34
3.20	Imagem do jogo “Kinect Disneyland Adventures”, exibindo a interação da personagem com uma personagem da Disney. Apesar do jogo ser supostamente em um local real, as personagens não são representadas por funcionários utilizando fantasias, mas sim por meio de suas representações em desenho animado (Jami Ferguson, 2011).	34
3.21	Um exemplo de navegação por menus de Kinect Disneyland Adventures. O cursor reage ao movimento da mão direita e os itens do menu são selecionados após manter a mão em posição por tempo suficiente, visto pela proporção em azul ao redor do cursor (SESSAO JOGUINHO: KINECT DISNEYLAND ADVENTURES, 2011).	35
3.22	Imagem do <i>minigame</i> “Rally Ball” de “Kinect Adventures”, em que o jogador deve manipular o corpo do avatar que imita seus movimentos para bloquear o maior número possível de bolas (IMAGENS: KINECT ADVENTURES , XBOX 360).	36
3.23	Imagem de uma luta de espadas no jogo “Puss in Boots”, que manuseia a espada imitando os movimentos realizados pelo jogador (James Newton, 2011).	36

3.24	Imagem demonstrando o funcionamento da tecnologia “Player Projection” em “Michael Jackson: The Experience”. O Kinect detecta as bordas que limitam a imagem do jogador em cada quadro capturado e os exibe na tela com uma filtragem de cores definida pelo jogador (MICHAEL JACKSON THE EXPERIENCE KINECT, 2013).	37
3.25	Imagem do jogo “Michael Jackson: The Experience” ao executar o movimento “The Lean”. Por se tratar de um movimento especial, o jogador precisa apenas realizar um gesto mais simples. Ao detectar esse gesto, o jogo altera a representação do jogador pelo “Player Projection” para a de Michael Jackson realizando o movimento como era realmente utilizado em seus shows (“SMOOTH CRIMINAL” MICHAEL JACKSON THE EXPERIENCE , XBOX360 KINECT).	39
4.1	Audiathon: o projeto original, adaptado para Kinaudiathon. Obstáculos azuis e vermelhos ocupam certa de 60 graus do tubo, sendo possível a colocação de 6 obstáculos na mesma seção do tubo.	41
4.2	Aeronave navegando em um tubo. Seu ângulo representa o ângulo da seta vermelha que parte do centro do tubo e vai ao encontro da aeronave.	42
4.3	Gesto volante: descartado pelas limitações de movimento e intuitividade.	43
4.4	Gesto com o cotovelo e mão: descartado por não possuir um movimento fluído.	44
4.5	Gesto com o ombro e mão: foi o gesto escolhido para o jogo Kinaudiathon e posteriormente Rainbow Riders of Kinect.	44
4.6	Variação do gesto com o ombro e mão. Ao articular o cotovelo é possível bloquear a vista do ombro, como visto na fotografia à direita.	45
4.7	Kinaudiathon: o projeto intermediário, adaptado para Rainbow Riders of Kinect.	46
4.8	Aparência do jogo Rainbow Riders of Kinect.	47
4.9	“Roda de cores” utilizada na teoria das cores. Exibe as 3 cores primárias (azul, amarelo e vermelho), as 3 cores secundárias (verde, violeta e laranja), assim como as 6 cores terciárias. A roda não exibe o resultado de misturas com a cor branca, presente na versão do jogo usada para os testes com o usuário (Cameron Chapman, 2010).	48
4.10	Primeiras linhas do arquivo <i>B</i> e arquivo <i>O</i> , gerados pelo conversor e utilizados como entrada para o jogo Rainbow Riders of Kinect.	52
6.1	Imagens presentes nos vídeos registrados durante o experimento. O jogo se mostrou atrativo para as crianças.	59
6.2	Estatística referente à frequência com que os usuários jogam vídeo games.	59
6.3	Estatística referente à experiência em jogos com o Kinect.	59
6.4	Estatística referente à satisfação dos usuários em relação ao jogo.	59
6.5	Estatística referente à auto-avaliação dos usuários do experimento.	60
6.6	O que foi mais “legal” dentre as 5 opções disponíveis, de acordo com os usuários do experimento.	60
6.7	O que foi mais “chato” dentre as 5 opções disponíveis, de acordo com os usuários do experimento.	60

6.8	O que foi mais difícil dentre as 4 opções disponíveis, de acordo com os usuários do experimento.	60
6.9	O que foi mais fácil dentre as 4 opções disponíveis, de acordo com os usuários do experimento.	60
6.10	Estatística sobre as médias e desvios padrão obtidos dos diferentes testes e diferentes faixas etárias.	61
6.11	Gesto utilizando ombro e mão utilizados de forma incorreta, diminuindo a distância entre os dois pontos pela perspectiva do Kinect e potencialmente afetando o posicionamento da aeronave.	62

LISTA DE TABELAS

3.1	Tabela comparativa contendo informações referentes aos jogos educativos estudados.	28
3.2	Tabela comparativa contendo informações referentes aos jogos que utilizam o Kinect estudados.	38

RESUMO

Neste trabalho é proposto um meio de interação gestual aplicável ao sensor de movimentos Kinect para ser utilizado por um jogo eletrônico educativo que foca o ensino da mistura de cores (utilizando o modelo subtrativo) para o público infantil.

Os gestos estudados tem como objetivo otimizar o controle da movimentação de um objeto no interior de uma superfície tubular de forma a tornar o mais intuitivo possível (e sem oferecer limitações) a seleção de outros objetos presentes nessa estrutura. O método de ensino do jogo também é estudado para um aprendizado mais produtivo.

Para a contextualização da proposta, também são apresentados: trabalhos relacionados aos temas propostos (jogos utilizando o Kinect e jogos eletrônicos educativos), etapas do desenvolvimento do jogo e sua arquitetura final, assim como experimentos com usuários a fim de provar sua usabilidade. Embora não tenha sido possível provar a eficácia no método de ensino aplicado, o jogo demonstrou ser envolvente e adequado para o público infantil.

Palavras-chave: Kinect, jogos eletrônicos educativos, educação infantil, vídeo games.

Gestural interaction in educational games using the Kinect motion sensor

ABSTRACT

In this work we propose an interaction technique based on gestures to be used in an educational electronic game. The game focus is to teach the mix of colors (using the subtractive color model) to children. Interaction is provided using Microsoft Kinect motion sensor.

The gestures proposed aim to optimize the control of the movement of two objects that move around a tubular surface. The interactive solution was conceived to be as intuitive as possible (and without any movement limitation). The teaching method used in the game is also studied, with the objective to improve learning productiveness.

In order to contextualize the proposal, this text also presents: related works on the themes proposed (games making use of Kinect and educational video games); the stages of our game development and its final architecture; and also the experiments with users we have conducted in order to verify its usability. Although it was not possible to prove the effectiveness of the teaching method applied, the game proved to be engaging and appropriate for children.

Keywords: Kinect, educational electronic games, early childhood education, video games.

1 INTRODUÇÃO

Jogos educativos estão entre os gêneros de jogos menos populares. Prover um misto entre educação e entretenimento de forma equilibrada e eficiente tem sido um grande desafio para diversas produtoras de jogos. Dentre os jogos desse gênero, alguns acabam por possuir uma grande carga de conhecimento sem encontrar um bom meio de transmiti-lo, enquanto que outros possuem um alto grau de entretenimento, porém com pouco conteúdo útil para o aprendizado do jogador. Desde o surgimento dos vídeo games e computadores domésticos, novos meios de aprendizado foram possíveis com jogos eletrônicos capazes de uma interação que jogos de tabuleiro ou de cartas não conseguem imitar. Ao longo de várias décadas, novas formas de interação foram propostas para vídeo games e computadores, porém sem obter sucesso, normalmente pela tecnologia não permitir ferramentas mais precisas, ou pela complexidade de processamento necessária para o funcionamento aceitável. Ao longo dos últimos 7 anos, notoriamente por meio do Wiimote¹, produtoras de consoles e jogos vêm obtendo maior sucesso com ferramentas de interação inovadoras, muito devido à repetição excessiva de controladores de jogo como *joypads* e teclado com mouse. No ano de 2010 foi lançada uma das grandes contribuições para esse despertar de novos meios de interação: o sensor de movimentos Kinect², a primeira ferramenta capaz de permitir a jogabilidade sem a necessidade do contato direto que obteve grande sucesso, interagindo apenas por meio da captura de movimentos do jogador.

Inicialmente utilizado apenas para jogos do console Xbox 360, bibliotecas de desenvolvimento para o Kinect foram disponibilizadas para o desenvolvimento em PCs, o que permitiu que grupos menores e programadores autônomos pudessem desenvolver jogos utilizando a tecnologia. O projeto Acessinha SP é uma consequência da maior acessibilidade de grupos de programadores de pequeno porte a plataformas de desenvolvimento compatíveis com essas novas formas de interação. Com mais de 2 milhões de cadastros e 700 postos ativos, o projeto Acessinha SP “busca garantir, por meio de acesso democrático à internet, a possibilidade de aprendizagem, lazer, desenvolvimento intelectual e social das crianças, juntamente com a família e a comunidade” (ACESSINHA - ACES-SASP, 2013). Ele também incentiva o envio de novos projetos que possam oferecer um atrativo para crianças.

Esse fator motivador resultou na decisão de reutilizar um projeto antigo e adaptá-lo para a meta proposta: criar um jogo eletrônico educativo para crianças. A reutilização permite um resultado final mais elaborado, porém limita o escopo de elementos educacionais possíveis, o que é um fator que afeta profundamente o potencial do jogo. O jogo original, apesar de ser facilmente adaptável para o Kinect por oferecer poucos graus de

¹Controlador do Nintendo Wii.

²Periférico do console Xbox 360. Mais detalhes na seção 2.1.

liberdade, pode perder seu elemento de entretenimento por essa transição, ou pelas adaptações necessárias para torná-lo educativo. Como já mencionado, esse é um dos grandes desafios no desenvolvimento de jogos educativos (sacrifício do entretenimento em prol da educação e vice versa). Esses dois fatores (entretenimento e ensino) devem portanto ser testados de forma a comprovar a qualidade do jogo.

Este trabalho está focado em crianças com idades entre 6 e 10 anos como faixa etária mais indicada para seu conteúdo, na tentativa de se adequar à faixa etária proposta pelo projeto Acessinha SP. Ao compartilhar esse jogo com o projeto Acessinha SP e possivelmente com qualquer outra organização que foque seus esforços na educação infantil por meio de jogos eletrônicos, espera-se que o jogo possa alcançar o maior número de crianças possível, de alguma forma contribuindo para o seu aprendizado, além da diversão proporcionada. O jogo, que foi produzido com a *engine*³ Unity⁴, pode também ser exportado para outros formatos permitindo outras formas de interação como dispositivos móveis ou Nintendo Wii, o que aumentaria o número de crianças com os meios para utilizar o jogo.

Este documento está organizado em 7 capítulos. O capítulo 2 introduz uma visão global dos conceitos foco desse trabalho: a interação gestual por meio do Kinect e os jogos educativos. O capítulo 3 apresenta uma gama de jogos relacionados à esses dois conceitos foco e os potenciais conceitos a serem implementados nesse trabalho. O capítulo 4 descreve a implementação realizada, desde seu estado inicial como projeto original até o estado final como trabalho adaptado aplicando os dois conceitos foco. O capítulo 5 oferece as definições de hipóteses e descrição de experimentos dos testes com o usuário. O capítulo 6 apresenta os resultados obtidos por meio desses testes. Por fim, o capítulo 7 apresenta as conclusões do trabalho e sugestões de melhoria.

³*Engine*, simplificado do termo *game engine*, ou motor de jogos: sistema designado para o desenvolvimento de vídeo games

⁴<http://unity3d.com/>.

2 VISÃO GLOBAL

Esse capítulo contém uma descrição dos elementos pertencentes aos dois tópicos principais do estudo abordado: interação gestual por meio do Kinect e jogos educativos. Apesar do foco do estudo ser mais específico em jogos eletrônicos educativos, a seção de jogos educativos (seção 2.2) abordará o conceito em geral, sem especificar se estes são jogados por meio de um vídeo game, computador ou qualquer outro meio eletrônico.

2.1 Interação gestual por meio do Kinect

O sensor de movimentos Kinect é um periférico do console Xbox 360 que contém diversos recursos permitindo que os jogadores sejam o controlador do jogo e sintam como se estivessem presentes em seu cenário. O software codificado no Kinect é um dos elementos que o tornam tão inovador. Seus desenvolvedores obtiveram grandes quantidades de dados capturando movimentos de pessoas em cenários da vida real. Ao processar todos esses dados utilizando um algoritmo de aprendizado, o Kinect pode mapear a informação visual que coleta em modelos tridimensionais representando pessoas de todos os tipos (variando idade, altura, gênero, formato do corpo, vestimenta utilizada, entre outros).

A inteligência armazenada no sistema do Kinect é suficiente para analisar o que ele captura e alinhar essa informação com a coleção de estruturas de esqueletos armazenada para interpretar os movimentos do jogador. Assim que sua memória possui informações suficientes de suas partes do corpo, ela gera uma saída com essas informações em um formato 3D simplificado. Além de avaliar os movimentos do jogador, o Kinect também deve determinar as distâncias dos diferentes pontos do corpo do jogador ao longo de todo o curso do jogo. Para fazê-lo, ele utiliza uma série de sensores e analisa todos esses dados a uma taxa de 30 vezes por segundo.

O Kinect contém 3 partes vitais que funcionam juntas para detectar o movimento do jogador e criar sua imagem física na tela: uma vídeo câmera VGA em cores RGB, um sensor de profundidade e um microfone multivetorial. A câmera detecta os componentes de cor vermelha, verde e azul e possui uma resolução de 640 por 480 pixels e uma taxa de captura de 30 quadros por segundo. A qualidade do vídeo capturado auxilia a determinar o formato do corpo e a expressão facial do jogador. O sensor de profundidade contém um sensor monocromático e um projetor infravermelho que auxilia na criação de uma imagem tridimensional do conteúdo capturado pelos sensores. Ele também calcula a distância entre cada ponto do corpo do jogador ao transmitir uma luz infravermelha e calcular o tempo de ida e volta desses raios luminosos após refletir em outros objetos. O Kinect possui um sistema de microfones, composto de 4 microfones individuais que são capazes de isolar a voz do jogador de outros ruídos registrados, permitindo que os jogadores utilizem sua voz como um recurso extra para o controle. Unidos, esses componentes

podem detectar 48 pontos diferentes do corpo de cada jogador na taxa de captura de seus componentes individuais: 30 quadros por segundo.

Ao unir as capacidades de software e de hardware do Kinect, ele é capaz de gerar imagens em 3D e reconhecer seres humanos em seu campo de visão. Ele pode analisar a pessoa à sua frente e aplicar diversos filtros para determinar qual o formato da estrutura corporal do jogador dentre os tipos programados em seu sistema. Com as funcionalidades oferecidas, o Kinect se torna útil não só em jogos eletrônicos, mas também para qualquer aplicação que necessite do reconhecimento de pessoas em um ambiente 3D, sendo um sensor bastante preciso para o seu custo (Robert Cong, Ryan Winters, 2013).

2.2 Jogos educativos

Jogos educativos são jogos explicitamente desenvolvidos com propósitos educacionais ou que possuem valor educacional apenas em dados momentos ou contextos. Em geral, eles são produzidos para ensinar pessoas (em geral ou focado em grupos específicos) sobre certos assuntos, expandir ou reforçar conceitos, entender eventos históricos ou culturas, ou auxiliar o jogador no aprendizado de uma habilidade ao jogar. Qualquer tipo de jogo pode ser utilizado em um ambiente educacional, sendo os mais comuns jogos de tabuleiro, jogos de cartas e jogos eletrônicos.

Jogos dessa categoria possuem a intenção de produzir resultados na educação do jogador. O aprendizado por meio de jogos geralmente é desenvolvido de forma a equilibrar o assunto em questão entre jogabilidade e aprendizado, permitindo que o jogador aplique o conceito do assunto focado pelo jogo no mundo real.

Tradicionalmente, a tecnologia utilizada nas escolas opera de forma a realizar estudos de caso para apresentar os estudantes a certos conceitos, de forma a prepará-los para uma tarefa relevante no futuro que irá requerer esse conceito. No futuro, espera-se que o uso de jogos em ambientes simulados apresente o jogador a assuntos relevantes, de forma a tornar a experiência mais prática e útil. Em setores profissionais, tais como treinamento de voo, simulações já são utilizadas de forma a preparar pilotos para o treinamento com aeroplanos reais. Essas seções de treinamento são utilizadas para replicar situações críticas que ocorrem nos aeroplanos sem o fator de risco associado com o voo real.

Com o aumento da disponibilidade de dispositivos eletrônicos, vem ocorrendo uma mudança nos tipos de jogos praticados pelo público. Jogos eletrônicos (seja por um vídeo game ou por outro meio eletrônico) têm se tornado mais populares do que jogos de tabuleiro tradicionais. Esses jogos possuem a meta de envolver o jogador enquanto ensinam conceitos e habilidades. A capacidade de imersão dos jogadores (que normalmente requer um grande esforço no desenvolvimento do jogo para alcançá-la) facilita a “personificação empática”, que ocorre quando o jogador aprende a se identificar com a personagem que está utilizando e com o ambiente na qual está inserido (Sasha A. Barab, Brianna Scott, Sinem Siyahhan, Robert Goldstone, Adam Ingram-Goble, Steven J. Zuckerman, Scott Warren, 2009).

3 TRABALHOS RELACIONADOS

Este capítulo contém uma lista de trabalhos relacionados aos 2 tópicos principais do estudo abordado: jogos eletrônicos educativos e jogos que utilizam interação gestual por meio do Kinect. Ao estudar a respeito de vídeo games notáveis para sua época, elaborou-se uma lista com 20 jogos (sendo 10 jogos de cada categoria) que representam o gênero a qual pertencem, contendo a descrição dos elementos educacionais abordados e gestos utilizados, dependendo do tópico em questão. No final das descrições dos trabalhos de uma categoria, os dados obtidos são comparados e uma conclusão é tomada em forma de uma decisão para este trabalho.

3.1 Jogos eletrônicos educativos

Um jogo eletrônico educativo é um vídeo game que possui um valor educacional para o jogador. Jogos desse tipo focam em combinar educação e entretenimento em um único produto. A definição utilizada envolve jogos que possuem primariamente o objetivo do entretenimento, que eventualmente tendem a ensinar algum tipo de conteúdo ou realizar exercícios mentais de forma a estimular o cérebro do jogador. Softwares sob essa definição não são desenvolvidos baseado em algum currículo escolar, normalmente não envolvem a participação de orientadores educacionais e portanto não focam em habilidades dessa área, como alfabetização e introdução à matemática.

“Para um melhor entendimento dos métodos utilizados para ensino em jogos eletrônicos, esta seção contém um resumo de diversos jogos que foram considerados como educativos e notáveis” (LIST OF EDUCATIONAL VIDEO GAMES, 2013).

3.1.1 Big Brain Academy: Wii Degree

“Big Brain Academy: Wii Degree” é um jogo do console Nintendo Wii que tem a intenção de testar a capacidade cerebral do jogador, com desafios que requerem diferentes atributos do jogador, tais como análise, memorização, visualização, identificação e computação. Os resultados do teste são informados por meio de uma “massa cerebral” (em gramas) e uma avaliação por conceito (como C+, B- e A), como visto na figura 3.1. O jogador será exposto a uma série de pequenos desafios (com uma duração de cerca de 10 segundos cada) referentes aos atributos testados (utilizando o Wiimote) e será avaliado positivamente caso os resolva corretamente e o mais rápido possível. O objetivo é que o jogador ao praticar o jogo possa ter um pensamento mais rápido e preciso. Apesar disso, o jogo não ensina nenhum conceito específico, sendo apenas uma forma simples de exercitar o cérebro através da repetição dos testes. Um dos desafios do jogo pode ser visto na figura 3.2.

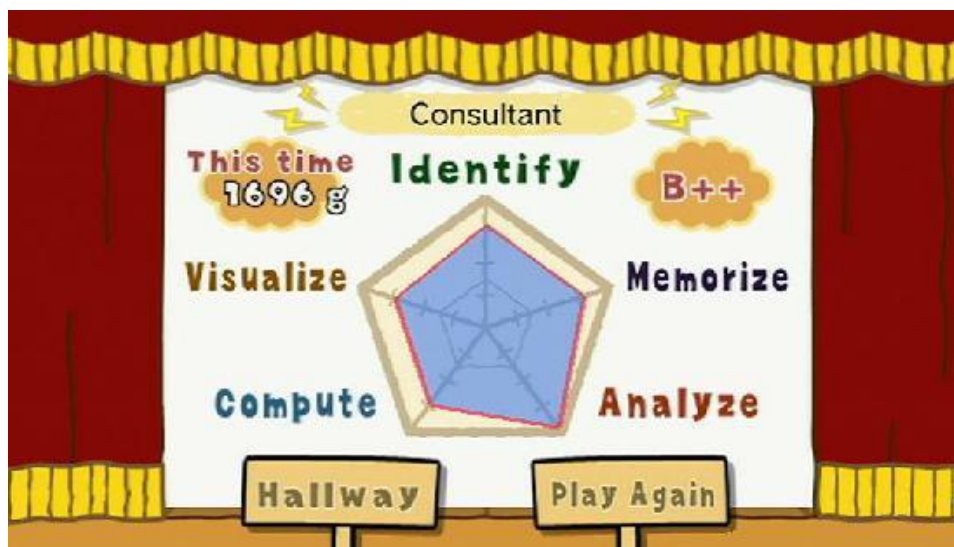


Figura 3.1: Resultado de um teste cerebral em “Big Brain Academy: Wii Degree”. Os 5 atributos são apresentados por meio de um gráfico radial e a nota final e “massa cerebral” do jogador são apresentadas (DESTRUCTOID REVIEW: BIG BRAIN ACADEMY: WII DEGREE, 2007).

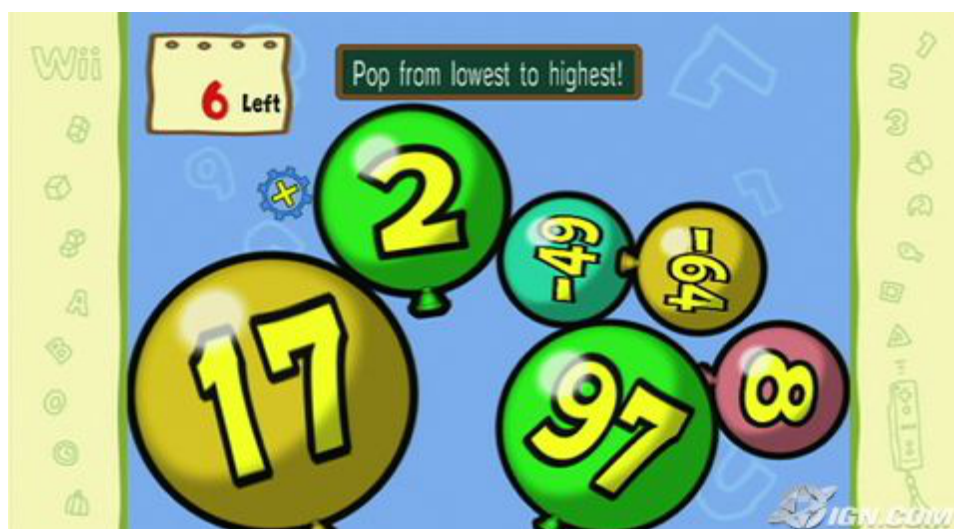


Figura 3.2: Exemplo de desafio em “Big Brain Academy: Wii Degree”. Ao ordenar os números o jogo considera que está avaliando a capacidade de computação do jogador, um dos 5 atributos testados (Matt Casamassina, 2007).



Figura 3.3: Último estágio do jogo “Genomics Digital Lab”. O jogador deve saber a base de DNA a ser usada para parear com a base incompleta (Denise Deveau, 2010).

3.1.2 Genomics Digital Lab

Trata-se de um jogo de PC que tem a intenção de ensinar o jogador o funcionamento da vida vegetal, exibindo e praticando os conceitos e elementos necessários para uma planta sobreviver. As atividades do jogo envolvem ajustar níveis de luz e líquidos, mover substâncias para simular o funcionamento das organelas das células vegetais, e até mesmo mover bases do DNA (como visto na figura 3.3). O foco principal é ensinar o jogador sobre os mecanismos presentes nas plantas.

3.1.3 Ko’s Journey

“Ko’s Journey” é um jogo de PC que tem a intenção de ensinar e incentivar o uso da matemática. O jogador se depara em um ambiente selvagem em que precisa resolver cálculos matemáticos utilizando um livro com os conhecimentos de sua tribo (como visto na figura 3.4). Esse conhecimento pode ser utilizado para determinar o tempo de viagem a algum lugar, orientar-se por meio de estrelas utilizando o plano cartesiano ou preparar um remédio ao calcular a proporção certa de ingredientes, visto na figura 3.5. O jogo já mostrou ter resultados positivos¹.

3.1.4 Carmen Sandiego

Trata-se de uma série de jogos de PC (17 jogos atualmente) que tem a intenção de ensinar diferentes matérias escolares para os jogadores, inicialmente ensinando apenas geografia e história (como visto na figura 3.6), expandindo seu escopo para matemática e inglês nas versões seguintes. O conceito é similar ao jogo detetive, em que o jogador deve escolher diferentes locais no mundo e buscar por pistas para encontrar a ladra Carmen Sandiego.

¹Uma demonstração do jogo “Ko’s Journey” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=FfwNqNK1JWk>.

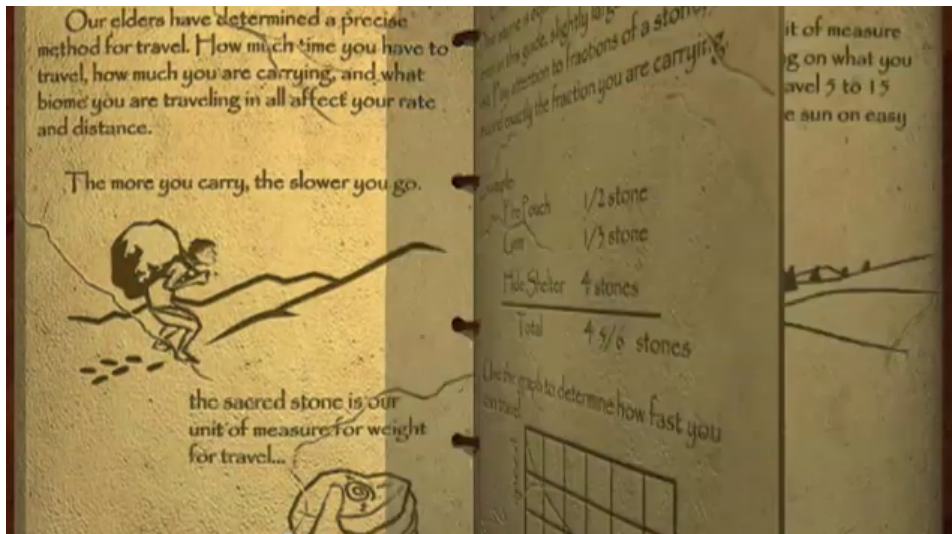


Figura 3.4: Livro com informações matemáticas, proveniente dos ancestrais da tribo de Ko em “Ko’s Journey”. O livro será a referência para qualquer cálculo matemático que venha a ser realizado ao longo do jogo (Scott Laidlaw, 2009).

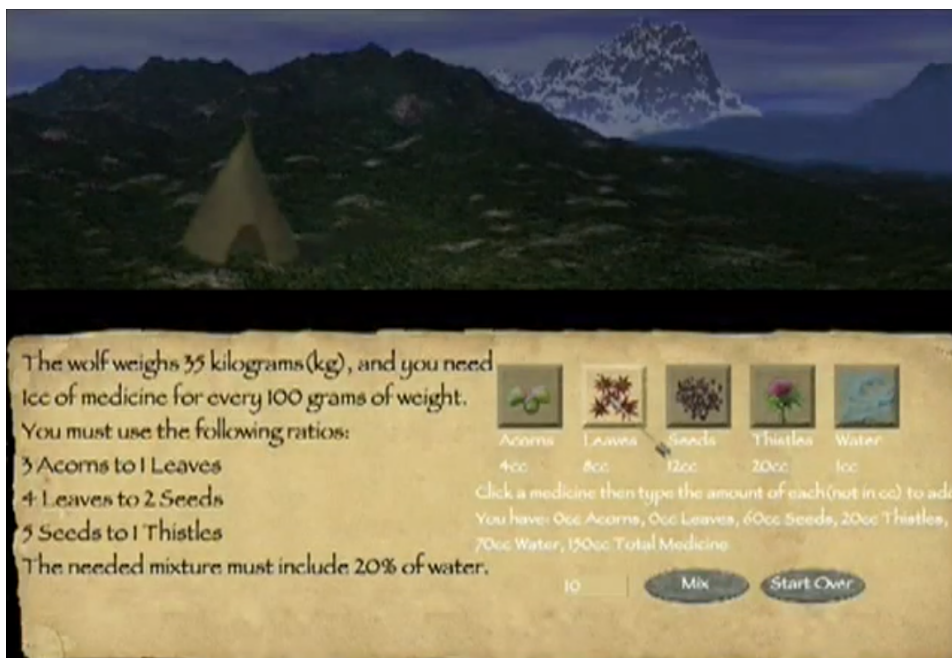


Figura 3.5: Receita de um remédio feito para um lobo no jogo “Ko’s Journey”. É necessário misturar os ingredientes na proporção certa, além da quantidade estar de acordo com o peso do lobo (Scott Laidlaw, 2009).



Figura 3.6: Descrição da cidade de Atenas em “Where in the World is Carmen Sandiego?”, o primeiro jogo da série de Carmen Sandiego. Esse jogo se focou principalmente em apresentar ao jogador diferentes pontos no mundo durante à busca por Carmen Sandiego (Yid Yang, 2003).

3.1.5 The ClueFinders

“The ClueFinders” é uma série de jogos de PC (10 jogos atualmente) em que o jogador deve participar de uma aventura em busca da solução de um mistério, utilizando temas como cavernas, pirâmides (visto na figura 3.7), no Himalaia ou em uma ilha com vulcão em erupção. Ao longo do curso do jogo, atividades educacionais variadas são os obstáculos a serem resolvidos para avançar. O jogo é repleto de animações no estilo de desenho animado, com desafios apropriados para crianças de 8 a 12 anos (dependendo do jogo).

3.1.6 Brain Age: Train Your Brain in Minutes a Day!

Trata-se de um jogo do console Nintendo DS que tem a intenção de desafiar o jogador com desafios mentais (como visto na figura 3.8) para o jogador exercitar o cérebro. Muito semelhante ao “Big Brain Academy: Wii Degree”, o jogo conta com pequenos cálculos matemáticos, exercícios de contagem, de observação e de memória². Parece mais focado ao público adulto/idoso como uma maneira de “manter o cérebro em forma” se utilizando da repetição (o subtítulo sugere que o jogo deva ser jogado alguns minutos por dia).

3.1.7 EcoQuest

A série EcoQuest é composta por 2 jogos de PC com a intenção de ensinar o jogador sobre a vida selvagem e o meio ambiente. No primeiro jogo explora-se o oceano em busca da baleia Cetus (como visto na figura 3.9) e foca-se na vida marinha. No segundo

²Uma demonstração do jogo “Brain Age: Train Your Brain in Minutes a Day!” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=NLLIQucI0xg>.

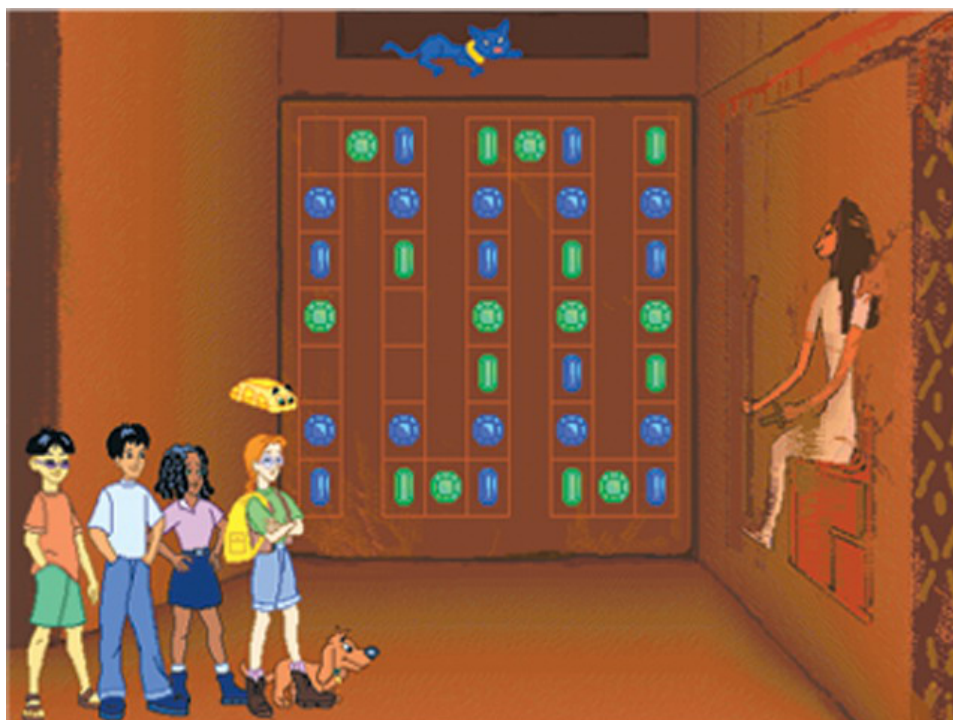


Figura 3.7: Um dos diversos *puzzles* em “The ClueFinders 4th Grade Adventures: Puzzle of the Pyramid”, o segundo jogo da série The ClueFinders (CLUEFINDERS YEAR 4 ADVENTURES - PUZZLE OF THE PYRAMID, 2013).

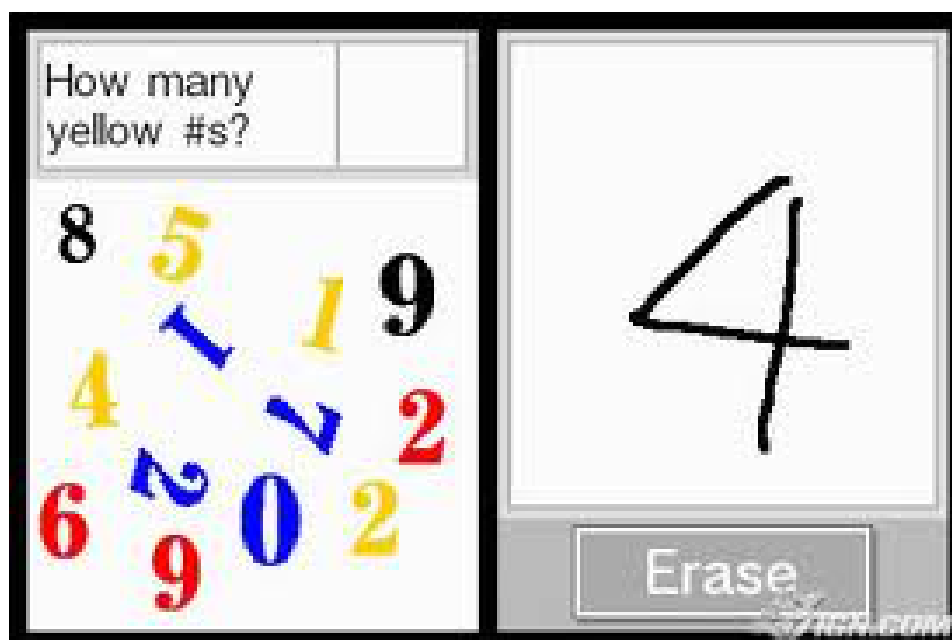


Figura 3.8: Um dos desafios presentes em “Brain Age: Train Your Brain in Minutes a Day!”. O desafio deve ser realizado segurando o DS lateralmente, assim como os outros desafios presentes no jogo, o que difere do uso tradicional do console (BRAIN AGE: TRAIN YOUR BRAIN IN MINUTES A DAY IMAGES, 2006).

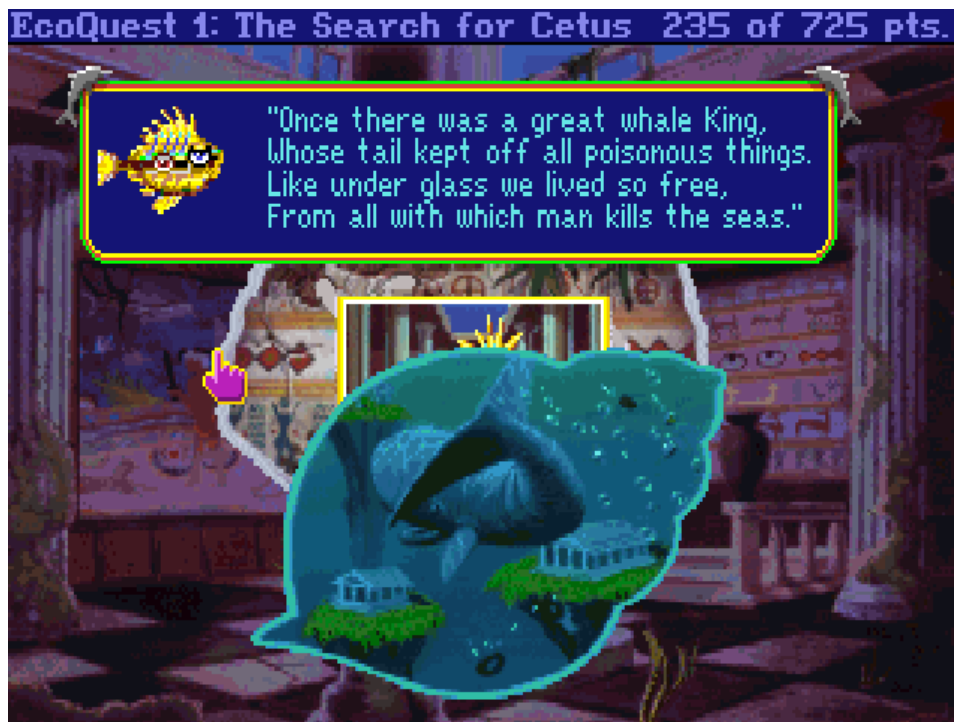


Figura 3.9: Tela do jogo “EcoQuest: The Search for Cetus” que descreve a baleia rei Cetus, o objetivo da busca (ECOQUEST THE SEARCH FOR CETUS, 2012).

jogo explora-se a floresta em busca da cura para uma doença que afeta os nativos da área, enquanto evita-se que a floresta seja destruída. É um jogo do tipo *point and click*, típico para a época do lançamento de seus jogos (1991 e 1993), provavelmente devido às limitações tecnológicas.

3.1.8 GCompris

GCompris é um jogo educativo que apresenta diferentes atividades para crianças de 2 a 10 anos de idade. Esse jogo possui uma vasta gama de atividades (130 atividades, 6 delas presentes na figura 3.10) dos mais variados assuntos, como aprender a manusear o computador, matemática, ciências, geografia, jogos clássicos como xadrez e memória, leitura, entre outros.

3.1.9 Super Solvers: Gizmos & Gadgets!

“Super Solvers: Gizmos & Gadgets” é um jogo educativo focado em conceitos de Física, como ímãs, mecânica, eletrônica básica e formas de energia. O objetivo do jogo é vencer corridas contra seus rivais construindo veículos para tal, como mostrado na figura 3.12. Isso é realizado movendo a personagem através de um labirinto repleto de portas que contém desafios (visto na figura 3.11), que ao serem completados irão render uma parte do veículo.

3.1.10 WolfQuest

WolfQuest é um jogo em 3D para PC que simula a vida selvagem ao fazer o jogador viver na pele de um lobo. O propósito principal do jogo é de ajudar os jogadores a entender os lobos e seu papel na natureza. O jogo consiste em saber sobreviver como um lobo, caçando suas presas ou procurando por carcaças, fugindo de ursos-pardos e coiotes



Figura 3.10: 6 exemplos de atividades presentes no jogo “GCompris” (Rafael Nink, 2008).



Figura 3.11: Labirinto de portas presente em “Super Solvers: Gizmos & Gadgets!”. Cada porta apresenta um desafio diferente para o jogador resolver e obter parte do veículo de corrida, o objetivo do jogo (FIVE GAMES THAT NEED TO BE ON WII U!, 2011).



Figura 3.12: Diagrama do veículo a ser construído em “Super Solvers: Gizmos & Gadgets!” (LET’S PLAY SUPER SOLVERS SERIES BY ICEPOTATO - PART 22: GIZMOS AND GADGETS PT 2, 2007).

ou formando um par com outro lobo para proteção e acasalamento. Há também o modo *multiplayer* (visto na figura 3.13) que permite até 5 jogadores simultâneos.

3.1.11 Comentários finais

Ao agrupar as informações dos jogos educativos mencionados, foi produzida a tabela 3.1.11 contendo os atributos mais relevantes para o estudo.

Entre outros atributos auto-explicativos, destacam-se os seguintes:

- Método: de que forma é apresentado o conteúdo do jogo;
- Abrangência: ensina algo específico, ou apenas pequenas charadas para o cérebro que são de conhecimento geral;
- Gênero: gênero do jogo, reflete o que será mais importante para obter bons resultados (por exemplo: reflexos ou pensamento rápido).

Jogos como “Big Brain Academy: Wii Degree” demonstram que o fator tempo pode tornar o jogo mais interessante quando trata-se de pequenos desafios que o jogador já conhece. Tais jogos não possuem o objetivo de ensinar, mas sim de treinar o jogador, de forma que os problemas se tornem triviais e possíveis de se realizar em menos tempo. Em contrapartida, em jogos como “Genomics Digital Lab” o objetivo é basicamente ensinar o jogador, não exigindo que as tarefas sejam realizadas rapidamente.

Entre jogos rápidos com pequenos desafios e jogos não dependentes do tempo ensinando novos conceitos, neste trabalho decidiu-se pelo primeiro por considerar mais fácil ter sucesso em treinar o cérebro do que ensinar algo completamente novo. O ensino deste provavelmente iria requerer uma abordagem mais específica para ativar o interesse do jogador ao desconhecido.

O conteúdo apresentado como uma história potencialmente atrai o jogador por mais tempo (com o objetivo de terminar o jogo), sendo um método que possui grande potencial. Apesar disso, algo mais simples como o método repetitivo de alguns dos jogos

Tabela 3.1: Tabela comparativa contendo informações referentes aos jogos educativos estudados.

Nome	Método	Abrangência Educacional	Gênero	Faixa etária	Tipo de interação	Ano
Big Brain Academy: Wii Degree	Repetição	Geral (lógica/matemática básica)	Ação (jogo contra o re-lógio)	Crianças e adolescentes	Wiiimote	2007
Genomics Digital Lab	Múltiplos estágios	Específica (Biologia/vegetais)	<i>Puzzle</i>	Adolescentes e adultos	Mouse e teclado	2009
Ko's Journey	<i>Storytelling</i>	Específica (matemática "focada")	<i>Puzzle</i>	Crianças e adolescentes	Mouse e teclado	2010
Carmen Sandiego	<i>Storytelling</i>	Geral	<i>Point and click</i>	Crianças e adolescentes	Mouse e teclado	Lançamentos entre 1985 e 2001
The ClueFinders	<i>Storytelling</i>	Geral	<i>Point and click</i>	Crianças e adolescentes	Mouse e teclado	Lançamentos entre 1998 e 2002
Brain Train Your Brain in Minutes a Day!	Repetição	Geral (lógica/matemática básica)	Ação (jogo contra o re-lógio)	Todas as idades	Caneta Stylus	2005 (Japão)
EcoQuest	<i>Storytelling</i>	Específica (vida marinha e meio ambiente)	<i>Point and click</i>	Crianças e adolescentes	Mouse e teclado	1991 e 1993
GCompris	Múltiplos estágios	Geral	Misto	Crianças	Mouse e teclado	2000, com última versão em 2012
Super Solvers: Gizmos & Gadgets!	Múltiplos estágios	Específica (física básica)	<i>Puzzle</i>	Crianças e adolescentes	Mouse e teclado	1993
WolfQuest	<i>Storytelling</i>	Específica (vida de um lobo)	Aventura	Adolescentes e adultos	Mouse e teclado	2007



Figura 3.13: Tela do jogo “WolfQuest” no modo *multiplayer* exibindo a HUD do jogo, com elementos como opções de diálogo, informação do seu e de outros lobos (WOLF QUEST GAME: 10. EPISÓDIO, 2013).

mencionados também possui potencial semelhante, além de requerer um menor esforço de desenvolvimento.

3.2 Jogos que utilizam interação gestual por meio do Kinect

O sensor Kinect foi lançado no dia 4 de novembro de 2010 nos Estados Unidos. Cerca de 127 jogos que utilizam o Kinect foram lançados desde seu lançamento até junho de 2013, sendo obrigatório o uso do sensor em alguns deles, enquanto que outros podem ser jogados por meios alternativos. Alguns desses jogos foram lançados em múltiplas plataformas, com o Kinect muitas vezes adaptado a partir de um estilo de jogo originalmente diferente.

Para um melhor entendimento dos gestos e conceitos utilizáveis no Kinect, esta seção contém um resumo de diversos jogos que utilizam total ou parcialmente o sensor Kinect como forma de interação com o jogo (LIST OF KINECT GAMES, 2013). Durante a descrição, quando mencionado um meio de interação ou gesto, será indicado entre parênteses qual o tipo de interação, seja seleção, manipulação ou navegação, assim como mímica, uma interação que pode ser expressa como uma combinação das 3 anteriores, porém não sendo evidente de qual forma.

3.2.1 Dance Central

“Dance Central” é um jogo musical e rítmico para o Xbox 360 que foi um dos lançamentos iniciais de jogos que utilizam o Kinect, sendo um dos primeiros jogos utilizando a tecnologia.

O jogo consiste em dançar conforme o ritmo da música (mímica), sendo que os movimentos são capturados pelo Kinect e representados na tela por meio de um avatar, sendo possível realizar 650 movimentos de dança diferentes (alguns exemplos podem ser visualizados na figura 3.14).



Figura 3.14: Imagem do jogo “Dance Central”, exibindo o formato de sua HUD, contendo exemplos de gestos usados como movimentos de dança, como “Lean Pop”, “Snake” e “Raise The Roof” (DANCE CENTRAL IMAGES, 2013).

3.2.2 Sesame Street: Once Upon a Monster

Trata-se de um jogo de Xbox 360 que utiliza o Kinect onde o jogador controla uma personagem da série Muppets e realiza pequenos *minigames* para avançar. O jogo é apresentado como um livro de histórias dividido em vários capítulos. Em cada capítulo há um monstro com algum tipo de problema a ser resolvido pelos Muppets³. Utilizados até mesmo nos menus, há diversos gestos suportados, como balançar os braços imitando os personagens do jogo (mímica, como visto na figura 3.15), mover as mãos na tela para pegar objetos (seleção) ou afastar as duas mãos na utilização do menu (seleção).

3.2.3 The Gunstringer

“The Gunstringer” é um jogo de tiro em terceira pessoa que utiliza o corpo do jogador para controlar o personagem. Com o tema do velho oeste, o jogador estará no controle do personagem Gunstringer, um xerife zumbi que possui fios como uma marionete. O jogo prossegue de forma similar a uma peça de teatro, sem a intenção de ser realista⁴. O jogador controla os movimentos do personagem com sua mão esquerda (navegação), enquanto que mira e atira com suas armas utilizando a mão direita. O jogador pode marcar múltiplos inimigos para atirar (seleção), e ao fazer o gesto para atirar (recuar o braço, simulando o efeito do recuo do revólver ao atirar), o personagem atira em todos os inimigos simultaneamente (manipulação da arma, sendo no máximo 6 tiros). A figura 3.16 dá uma ideia geral das características apresentadas.

3.2.4 Kinectimals

Trata-se de um jogo para o Xbox 360 (e também em outras plataformas, com a forma de controle diferente) em que o jogador interage com animais virtuais tratando-os como

³Uma demonstração do jogo “Sesame Street: Once Upon a Monster” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=Vr0kupO2Z-o>.

⁴Uma demonstração do jogo “The Gunstringer” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=Eb58beleiqg>.



Figura 3.15: Imagem do jogo “Sesame Street: Once Upon a Monster”, com os 3 monstros realizando um gesto que deve ser imitado pelo jogador para continuar (PLAYBOOK: SESAME STREET: ONCE UPON A MONSTER, 2013).



Figura 3.16: Imagem do jogo “The Gunstringer” apresentando o ambiente estilo velho oeste, a visualização estilo terceira pessoa, o cursor que se usa para seleccionar os inimigos a atirar e o número de balas restantes (PREVIEW: THE GUNSTRINGER, 2011).



Figura 3.17: Uma pista de obstáculos para adestrar seu mascote em “Kinectimals” (Stephen Totilo, 2010).

mascotes. Ao longo do jogo, o jogador utiliza o Kinect para alimentar, brincar e cuidar de seu mascote. O mascote imita seus movimentos, sendo possível realizar gestos para comandar ações como pular, sentar ou fingir de morto (manipulação do mascote). Esses truques serão usados sem objetivo específico (*free play*) ou em desafios como campos de obstáculos (como visto na figura 3.17).

3.2.5 Kinect Sports

“Kinect Sports” é um jogo que consiste em uma coleção de 6 simuladores de jogos esportivos (boliche, boxe, atletismo, tênis de mesa, vôlei de praia e futebol) e 8 *minigames*. O jogo foi desenvolvido para demonstrar as capacidades técnicas do Kinect (semelhante ao “Wii Sports” com o Wiimote), na qual é possível fazer gestos semelhantes aos dos esportes mencionados, como arremessar um dardo ou chutar uma bola de futebol (manipulação do dardo ou bola) e obter resultados semelhantes aos de realizar a ação na vida real (vide figura 3.18). O jogador utiliza seu avatar do Xbox 360 e ao final de cada evento o jogo determina os “melhores momentos” do jogador, permitindo que ele os salve ou até mesmo compartilhe pelo Facebook (indiretamente pelo Kinectshare)⁵.

3.2.6 Forza Motorsport 4

“Forza Motorsport 4” é o quarto jogo da franquia (com lançamentos para o Xbox e Xbox 360), sendo o primeiro a utilizar o Kinect como possível controle (porém opcional). Semelhante à série “Gran Turismo”, “Forza Motorsport” tenta simular o funcionamento de carros reais, sem fatores fictícios incluídos no jogo, incluindo uma visão realista da perspectiva do piloto (como visto na figura 3.19). O Kinect pode ser utilizado parcialmente, sendo o controle do carro ainda exercido pelo *joypad* do Xbox 360 e apenas utilizando a orientação da cabeça para alterar a direção da visão (manipulação da cabeça do avatar). O Kinect também pode ser utilizado totalmente, de forma que o jogador deva estender as mãos como se estivesse segurando um volante e girar seus braços imitando o giro desse

⁵Uma demonstração do jogo “Kinect Sports” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=Zw2Fo9k96hc>.



Figura 3.18: Ilustração da imitação de movimentos realizada no jogo “Kinect Sports” ao jogar vôlei de praia (KINECT SPORTS PARA XBOX 360, 2013).

volante (navegação). Neste caso, o carro irá utilizar câmbio automático e será acelerado automaticamente, sendo possível frear ao estender a perna direita para frente (manipulação do carro), semelhante a um piloto automático dos carros atuais⁶. É também possível utilizar a funcionalidade do microfone do Kinect para iniciar corridas e navegar pelos menus.

3.2.7 Kinect Disneyland Adventures

“Kinect Disneyland Adventures” é um jogo em que se explora a Disneylândia, participando de jogos temáticos e atrações presentes no parque real, além de permitir que o jogador capture fotografias de seu avatar no parque e interaja com os personagens da Disney (como visto na figura 3.20). O jogo consiste em passear pelo parque (navegação) e realizar pequenas tarefas e objetivos para obter créditos no parque. O controle é possível apenas por meio do Kinect. O jogador deve fazer movimentos como utilizar a mão como um cursor virtual para selecionar itens nos menus (seleção, vide figura 3.21), levantar o braço para responder uma pergunta de múltipla escolha (seleção) e abanar para iniciar ou encerrar uma conversa com alguém próximo (seleção).

3.2.8 Kinect Adventures

Trata-se de um jogo de ação para o Xbox 360, e está incluso com a compra do Kinect. O jogo é uma coleção de 5 *minigames* esportivos e de aventura que possuem duração de cerca de 3 minutos. O objetivo de cada *minigame* é obter o maior número de broches de aventura (semelhante a moedas) que definem quão boa foi a atuação dependendo do número obtido (o que resulta em medalhas dependendo da performance)⁷. O jogo captura o movimento de todo o corpo para exercer ações como inclinar para uma direção (mani-

⁶Uma demonstração do jogo “Forza Motorsport 4” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=BiewYJbUDF8>.

⁷Uma demonstração do jogo “Kinect Adventures” pode ser assistida pelo seguinte vídeo: http://www.youtube.com/watch?v=nee86i_vInE.



Figura 3.19: Imagem do jogo “Forza Motorsport 4” em sua tentativa de ser realista ao representar a aparência interna e externa dos carros, incluindo espelhos (Malcolm Owen, 2011).



Figura 3.20: Imagem do jogo “Kinect Disneyland Adventures”, exibindo a interação da personagem com uma personagem da Disney. Apesar do jogo ser supostamente em um local real, as personagens não são representadas por funcionários utilizando fantasias, mas sim por meio de suas representações em desenho animado (Jami Ferguson, 2011).



Figura 3.21: Um exemplo de navegação por menus de Kinect Disneyland Adventures. O cursor reage ao movimento da mão direita e os itens do menu são selecionados após manter a mão em posição por tempo suficiente, visto pela proporção em azul ao redor do cursor (SESSAO JOGUINHO: KINECT DISNEYLAND ADVENTURES, 2011).

pulação do avatar, para curvar um veículo ou mudar a posição do corpo, como visto na figura 3.22), pular para fazer o mesmo com sua personagem ou se agachar para desviar de obstáculos (idem ao anterior).

3.2.9 Puss in Boots

“Puss in Boots” (no Brasil, “O Gato de Botas”) é um jogo de ação baseado no filme homônimo, lançado para Xbox 360, Playstation 3, Wii e Nintendo DS. Permite o uso do Kinect e do Playstation Move para a versão de seus respectivos consoles, utilizando os sensores de movimento para imitar as habilidades em esgrima do Gato de Botas. O Gato de Botas deve limpar seu nome de todas as acusações contra sua pessoa (que o fez ser um fugitivo procurado), o que basicamente resulta em uma aventura repleta de lutas com espada. Gestos envolvem movimentação com os braços simulando uma luta com espada (vide figura 3.23), chutes, movimentar os braços simulando movimentos de arranhar para usar o ataque especial do Gato de Botas, além de movimentar o corpo para esquivar (manipulação do avatar ou espada)⁸.

3.2.10 Michael Jackson: The Experience

Trata-se de um jogo musical baseado nas músicas de Michael Jackson, lançado inicialmente em 2010 para Nintendo DS, PSP e Wii, teve uma versão melhorada desenvolvida para ser jogada pelo Kinect e o Playstation Move. A versão para Xbox 360 captura o movimento de todo o corpo, e também utiliza uma coreografia totalmente diferente dos lançamentos de 2010, além de outras mudanças e adições ao jogo original. O jogo contém os *hits* mais famosos de Michael Jackson, como “Thriller”, “Smooth Criminal”, “Bad”, “Billie Jean”, entre diversos outros. A versão para Kinect utiliza uma tecnologia chamada “Player Projection” (projeção do jogador), em que a captura de imagem do jogador é utilizada diretamente no jogo, porém com uma aura colorida envolvendo-a (como visto na

⁸Uma demonstração do jogo “Puss in Boots” pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=ibc6uS435pc>.



Figura 3.22: Imagem do *minigame* “Rally Ball” de “Kinect Adventures”, em que o jogador deve manipular o corpo do avatar que imita seus movimentos para bloquear o maior número possível de bolas (IMAGENS: KINECT ADVENTURES , XBOX 360).



Figura 3.23: Imagem de uma luta de espadas no jogo “Puss in Boots”, que manuseia a espada imitando os movimentos realizados pelo jogador (James Newton, 2011).

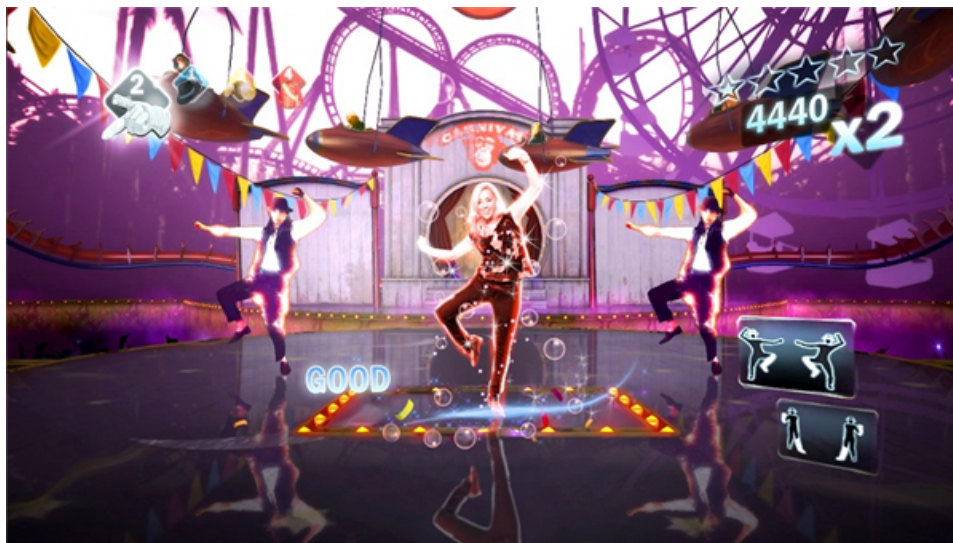


Figura 3.24: Imagem demonstrando o funcionamento da tecnologia “Player Projection” em “Michael Jackson: The Experience”. O Kinect detecta as bordas que limitam a imagem do jogador em cada quadro capturado e os exibe na tela com uma filtragem de cores definida pelo jogador (MICHAEL JACKSON THE EXPERIENCE KINECT, 2013).

figura 3.24). Os gestos interpretados são basicamente os do repertório de coreografias utilizados por Michael Jackson (mímica), com algumas modificações para movimentos que requerem condições especiais, como o “Moonwalk” e o “The Lean” (vide figura 3.25).

3.2.11 Comentários finais

Agrupando as informações dos jogos que utilizam o sensor Kinect que foram mencionados, foi elaborada a tabela 3.2.11 contendo os atributos mais relevantes para o estudo.

Entre outros atributos auto-explicativos, além dos atributos mais relevantes da tabela 3.1.11 descritos na subseção 3.1.11, destaca-se o seguinte:

- Exercício: nível de exercício físico exigido pelo jogo.

A maioria dos jogos que utilizam o Kinect predominantemente utilizam mímica, seja para o avatar imitar o jogador, ou o jogador repetir os movimentos do avatar (seguindo um conjunto de instruções de gestos, como em jogos de dança). Há também os jogos em que o Kinect é utilizado apenas como um auxiliar ou forma alternativa de jogo, como no jogo “Forza Motorsport 4” ou em lançamentos paralelos em outros consoles.

Ainda há muito a ser explorado a respeito de formas de se utilizar o Kinect, porém alguns conceitos puderam ser observados nesse estudo. Alguns dos exemplos incluem utilizar a mão como um cursor, controlar um personagem em terceira pessoa utilizando a mão esquerda para a movimentação e a direita para atacar, utilizar dos pés para pedais enquanto o jogador permanece sentado e navegar por menus de formas bem variadas.

Uma das formas de navegação de menus, vista em “Michael Jackson: The Experience” (vide subseção 3.2.10), permite a seleção de menus por meio do ângulo formado entre o ombro e a mão direita ou esquerda do jogador, e mostra ser uma forma de seleção adequada para a navegação em uma superfície tubular, que é a proposta do jogo desenvolvido neste estudo (vista no capítulo 4). Ao alterar a curvatura do ombro estendendo o braço, é possível fazer giros completos sem limitações físicas (ao contrário de utilizar

Tabela 3.2.: Tabela comparativa contendo informações referentes aos jogos que utilizam o Kinect estudados.

Nome	Gênero	Faixa etária	Tipo de interação	Método	Exercício	Ano
Dance Central	Dança/Ritmo	Todas as idades	Kinect	Múltiplos estágios	Alto	2010
Sesame Street: Once Upon a Monster	Aventura	Crianças	Kinect	<i>Storytelling</i>	Médio	2011
The Gunstringer	Aventura	Adolescentes e Adultos	Kinect	<i>Storytelling</i>	Baixo	2011
Kinectimals	Aventura	Crianças e adolescentes	Kinect	<i>Storytelling</i>	Médio	2010
Kinect Sports	Ação	Todas as idades	Kinect	Múltiplos estágios	Alto	2010
Forza Motorsport 4	Corrida	Todas as idades	Kinect e <i>joypad</i> do Xbox 360	Múltiplos estágios	Baixo	2011
Kinect Disneyland Adventures	Aventura	Crianças e adolescentes	Kinect	<i>Storytelling</i>	Baixo	2011
Kinect Adventures	Ação	Todas as idades	Kinect	Múltiplos estágios	Alto	2010
Puss in Boots	Aventura	Adolescentes	Kinect (versão do Xbox 360)	<i>Storytelling</i>	Alto	2011
Michael Jackson: The Experience	Dança/Ritmo	Adolescentes	Kinect (versão do Xbox 360)	Múltiplos estágios	Alto	2011 (versão do Xbox 360)



Figura 3.25: Imagem do jogo “Michael Jackson: The Experience” ao executar o movimento “The Lean”. Por se tratar de um movimento especial, o jogador precisa apenas realizar um gesto mais simples. Ao detectar esse gesto, o jogo altera a representação do jogador pelo “Player Projection” para a de Michael Jackson realizando o movimento como era realmente utilizado em seus shows (“SMOOTH CRIMINAL” “MICHAEL JACKSON THE EXPERIENCE”, XBOX360 KINECT).

as duas mãos simulando segurar um volante), além de evitar imprecisões caso a orientação fosse realizada por apenas um ponto (no caso, 2 pontos mapeados pelo Kinect serão utilizados: o ombro e a mão).

4 DESENVOLVIMENTO DO JOGO

Este capítulo contém a ideia (linha de raciocínio até o conceito final por dois ângulos diferentes, referentes aos tópicos focados pelo trabalho), o desenvolvimento (partindo do projeto inicial Audiathon, produzindo a versão intermediária Kinaudiathon e finalmente o jogo resultante Rainbow Riders of Kinect) e os detalhes da implementação do jogo produzido nesse trabalho.

4.1 Histórico de desenvolvimento

Audiathon é um jogo de ação em 3D, inspirado no jogo Audiosurf¹. O jogo consiste em pilotar uma aeronave dentro de um tubo enquanto desvia de obstáculos vermelhos e captura obstáculos azuis. O ambiente do jogo é totalmente dinâmico, sendo uma faixa de áudio responsável pela geração do tubo e dos obstáculos presentes nele (sensíveis à intensidade da faixa utilizada). O jogador possui a liberdade de mover a aeronave na superfície interna do tubo, sendo possível realizar translações completas nele. O controle do jogo pode ser exercido por meio de um mouse ou por um Wiimote adaptado para controlar o cursor do mouse (a opção com Wiimote era meramente por meio de um aplicativo que mapeia a posição do Wiimote para uma posição de cursor do Windows). O jogo requer o uso de um conversor para gerar os arquivos necessários para seu funcionamento a partir da faixa de áudio. O jogo possui uma alta densidade de obstáculos, tornando o jogo um simples desafio de reação rápida e precisa². Uma imagem do jogo Audiathon pode ser vista na figura 4.1.

O cálculo do movimento consistia no produto escalar entre o vetor tangente à posição da aeronave em relação ao tubo e o vetor de deslocamento do cursor no último quadro. Apesar desse método de controle ser bastante confiável, uma de suas grandes falhas ocorre devido à utilização do mouse e Wiimote, que são incapazes de realizar movimentos circulares de forma mais livre. Para que a aeronave completasse uma translação com apenas um giro com o mouse, era necessário que o usuário realizasse um giro que formasse um círculo exatamente do tamanho definido pelo programa. A realização de um giro que formasse um círculo de tamanho diferente ao esperado poderia resultar na aeronave presa em algum ângulo ou até mesmo movendo-se na direção oposta ao desejado, devido à natureza do produto escalar.

No decorrer do desenvolvimento deste trabalho foram realizadas diversas mudanças no trabalho original do Audiathon, com uma versão intermediária do trabalho intitulado

¹<http://www.audio-surf.com/>.

²O executável do jogo Audiathon pode ser baixado pelo seguinte link: <https://www.dropbox.com/s/id7lw7eniia3tbr/Audiathon.zip?dl=1>.

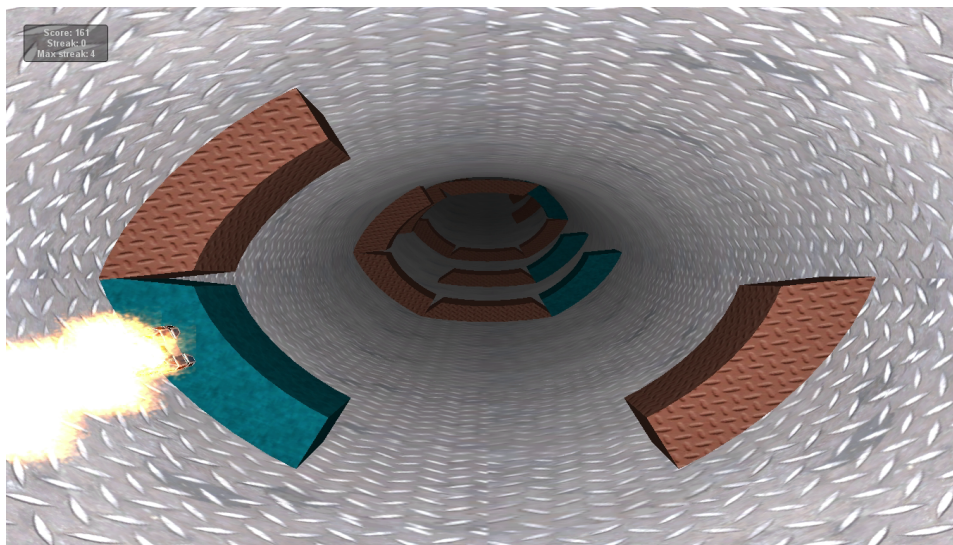


Figura 4.1: Audiathon: o projeto original, adaptado para Kinaudiathon. Obstáculos azuis e vermelhos ocupam certa de 60 graus do tubo, sendo possível a colocação de 6 obstáculos na mesma seção do tubo.

de Kinaudiathon. Esta versão basicamente alterou a forma de controle ao utilizar apenas o Kinect para tal, além de adicionar o cursor como um elemento extra de seleção para o jogo. O jogador deveria guiar a aeronave e o cursor ao objeto correspondente do elemento. Com apenas uma combinação correta entre 36 possíveis, o princípio do jogo começara a ficar mais complexo, sendo necessária a diminuição da densidade de objetos no tubo (nível semelhante ao jogo atual) e da velocidade da aeronave para permitir tempos de reação mais aceitáveis. A versão atual, intitulada *Rainbow Riders of Kinect* difere principalmente no cursor que agora é fixo à translação do tubo (antes sendo um cursor com a liberdade de um cursor em um sistema de janelas) e estilo de jogo, que foca no conceito do aprendizado sobre mistura de cores.

4.2 Decisões de projeto

Como visto na seção 4.1, a implementação usada na versão atual (intitulada *Rainbow Riders of Kinect*) partiu da alteração de um projeto anterior, se tratando de uma adaptação. Devido à motivação do trabalho como parte do projeto *Acessinha SP*, essa adaptação foi planejada de forma a exigir duas novas metas a serem atingidas pelo trabalho final: o controle do jogo ser realizado pelo sensor de movimentos Kinect e a adaptação para um jogo educativo. Essas duas novas metas levaram a diversas questões em relação ao objetivo do trabalho e os meios para alcançá-lo.

4.2.1 Interação por gestos

Para utilizar uma forma de interação gestual, primeiramente seria necessário determinar um meio de mapear os movimentos do jogador, transformando-os em um ângulo entre 0 e 360 graus, representando a posição da aeronave (o único grau de liberdade presente no jogo original *Audiathon*). Inicialmente surgiu a ideia de realizar uma simples transformação de coordenadas do mouse (da versão antiga) para coordenadas mapeadas de uma das mãos, agindo como um mero cursor, mantendo toda a lógica de cálculo de posicionamento idêntica a versão inicial vista na seção 4.1. O revés desse método, no entanto,

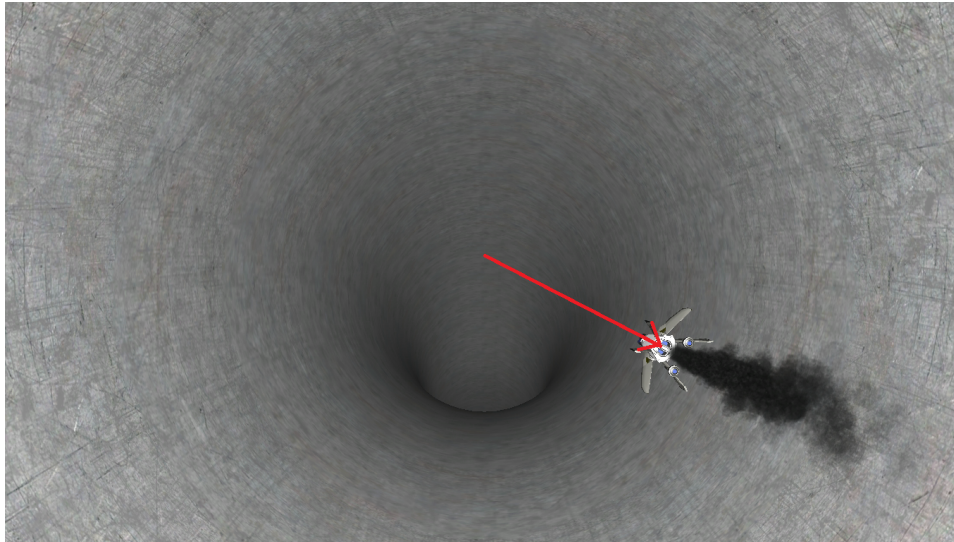


Figura 4.2: Aeronave navegando em um tubo. Seu ângulo representa o ângulo da seta vermelha que parte do centro do tubo e vai ao encontro da aeronave.

seria exatamente o mesmo do uso com o mouse: o movimento é relativo ao deslocamento do cursor, resultando em posições diferentes da aeronave para uma mesma posição do cursor, dependendo do movimento realizado nos momentos anteriores. Para superar esse problema seria necessário encontrar um gesto capturável pelo Kinect que representasse exatamente um ângulo entre 0 e 360 graus em qualquer contexto, um mapeamento absoluto. Para isso seria necessário o mapeamento de pelo menos 2 pontos capturados pelo Kinect, para então definir o ângulo da aeronave a partir do ângulo formado entre esses dois pontos e de algum referencial simples, como o eixo X ou Y referente aos valores capturados. A figura 4.2 apresenta a noção de ângulo no contexto do jogo.

O primeiro gesto considerado foi o da mímica de um volante, mapeando a posição da mão direita e esquerda e calculando o ângulo entre eles, transformando esse ângulo diretamente para o ângulo da aeronave. O grande revés desse gesto é a limitação da translação de forma fluente para qualquer direção, pois ao incrementar esse ângulo seria necessário o movimento de ambos os braços para a mesma direção, o que eventualmente resultaria no cruzamento dos dois braços (como visto na figura 4.3). Esse gesto é inviável para esse contexto por não ser possível realizar um movimento fluente como girar constantemente no sentido horário, além de não representar uma posição intuitiva para um dado ângulo do volante ao jogador (por exemplo, a posição da aeronave ao manter as duas mãos como na fotografia do meio na montagem da figura 4.3 não é intuitiva, pois não fica claro em qual canto a aeronave deveria estar).

Para resolver o problema da posição intuitiva para o jogador, considerou-se a utilização de alguma extremidade do corpo como primeiro ponto e o ponto mais próximo ligado à ele, representando a articulação que controla esse extremo. Apenas 5 extremidades são registradas pelo Kinect: os dois braços, as duas pernas e a cabeça. Pela maior precisão e conveniência foi decidido utilizar um dos braços. O ângulo então, seria formado entre a mão e o cotovelo de um dos braços, o que é intuitivo para jogador determinar o ângulo que representa (por exemplo, se a mão está verticalmente acima do cotovelo, é intuitivo concluir que a aeronave será mapeada para a parte de cima da tela). Por outro lado, a incapacidade de rotacionar de forma fluente se repete como um revés desse gesto. Seria necessário utilizar a combinação do ponto mapeado pela mão com algum outro ponto de

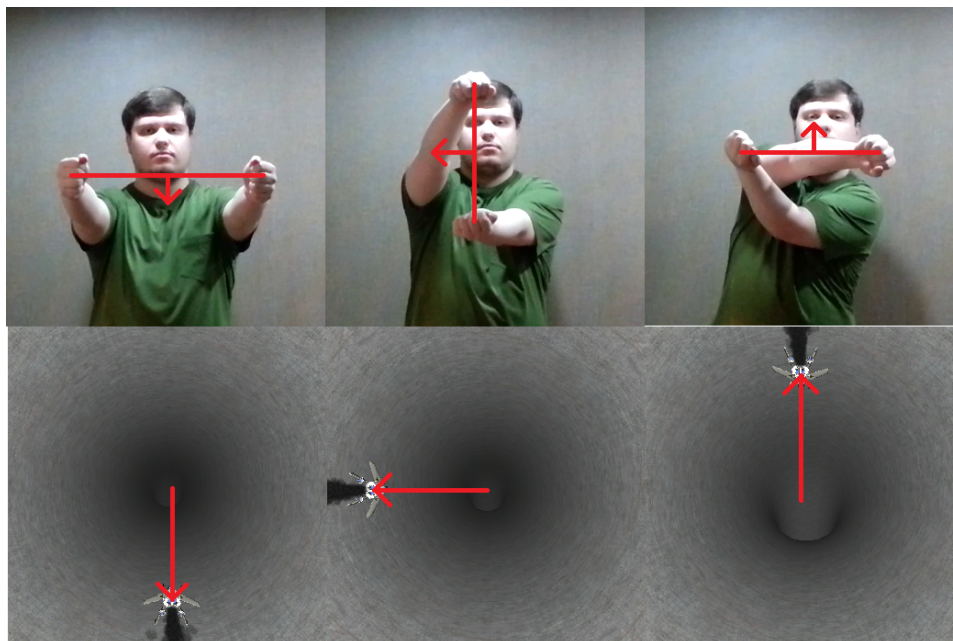


Figura 4.3: Gesto volante: descartado pelas limitações de movimento e intuitividade.

forma a alcançar um mapeamento absoluto, visualmente intuitivo e fluente para o jogador. A figura 4.4 ilustra o gesto com o cotovelo e mão.

O gesto utilizado na versão final do *Rainbow Riders of Kinect* então foi decidido a partir da escolha do segundo ponto para formar um ângulo com a mão de um dos braços: o ombro. Diferente do cotovelo, o ombro permite um movimento fluído para qualquer sentido, além de possuir os demais atributos positivos dos gestos anteriores, sendo assim definido como o gesto a ser utilizado no jogo para o controle da aeronave (visto na figura 4.5). Outro gesto proporcionado pelo mapeamento é por meio da articulação do cotovelo, formando ângulos sem necessariamente estender completamente o braço, bastando apenas uma distância considerável entre o ombro e a mão (visto pela câmera do Kinect) o que permite várias posições de jogo diferentes (embora apesar da conveniência, é possível que o braço do jogador bloqueie a vista do ombro, visto na figura 4.6, potencialmente causando imprecisões no cálculo de posição). Finalmente, uma das vantagens mais importantes desse gesto, assim como o anterior, é que pode ser utilizado duas vezes: um em cada braço. Isso resultou na ideia de utilizar outro braço para o controle de outra entidade, criando um segundo grau de liberdade ao jogo.

O braço alternativo ao usado para controlar a aeronave poderia ser utilizado. Inicialmente, a ideia (que foi mantida até o fim do desenvolvimento do *Kinaudiathon*) seria utilizar o outro braço de forma diferente à do braço da aeronave, desafiando o jogador a realizar tarefas simultâneas e diferentes. A forma de utilização decidida foi a mais simples possível, o mapeamento da mão como um cursor como o mouse, permitindo o movimento total sobre a tela. A ideia de jogo era simples: o jogador deveria combinar a seleção de um entre vários objetos que estariam fluando na tela por meio de um HUD com um dos obstáculos selecionáveis pela aeronave, limitando o número de combinações corretas entre os 2 elementos. Essa aproximação se mostrou interessante até a ideia da utilização das cores como meio de aprendizado (visto detalhadamente na subseção 4.2.2), o que levou a descartar o conceito de utilizar dois gestos diferentes, utilizando ambas as mãos da mesma forma (com um ângulo calculado a partir da posição da mão e ombro).

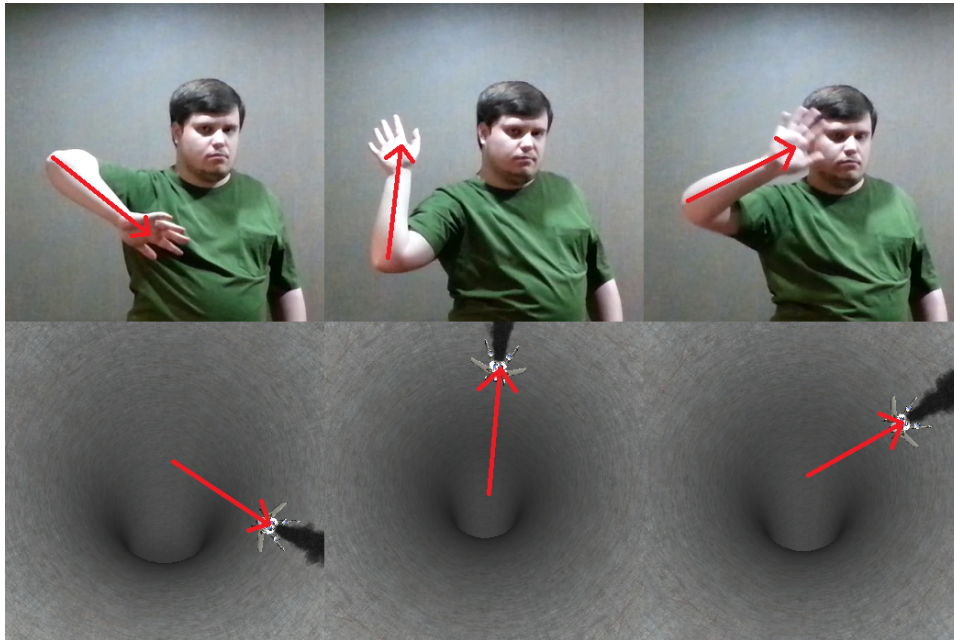


Figura 4.4: Gestor com o cotovelo e mão: descartado por não possuir um movimento fluído.

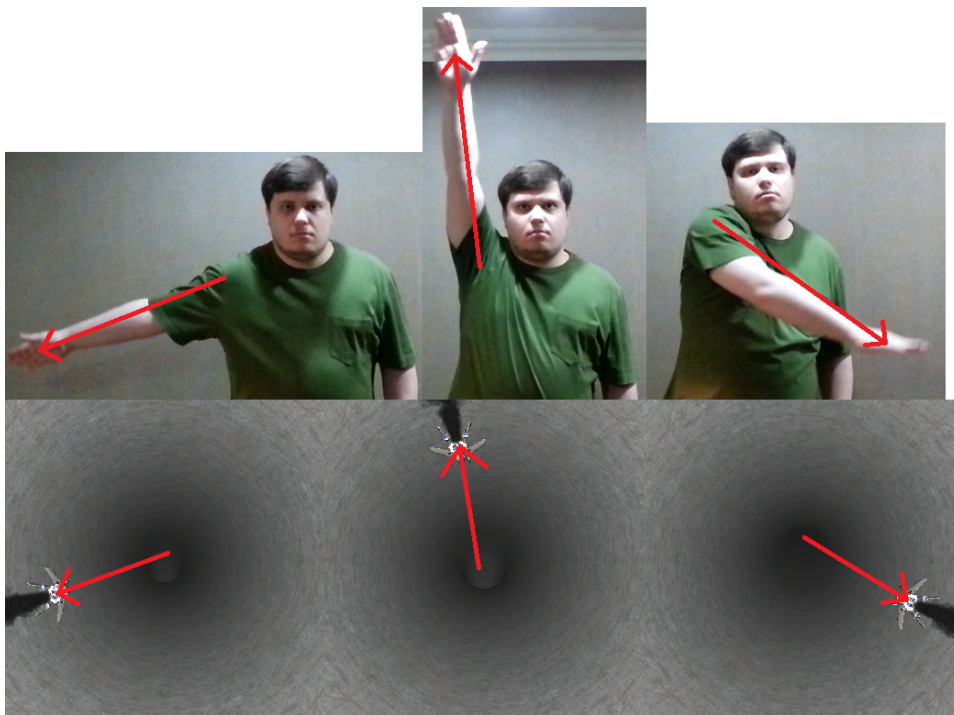


Figura 4.5: Gestor com o ombro e mão: foi o gestor escolhido para o jogo Kinaudiathon e posteriormente Rainbow Riders of Kinect.

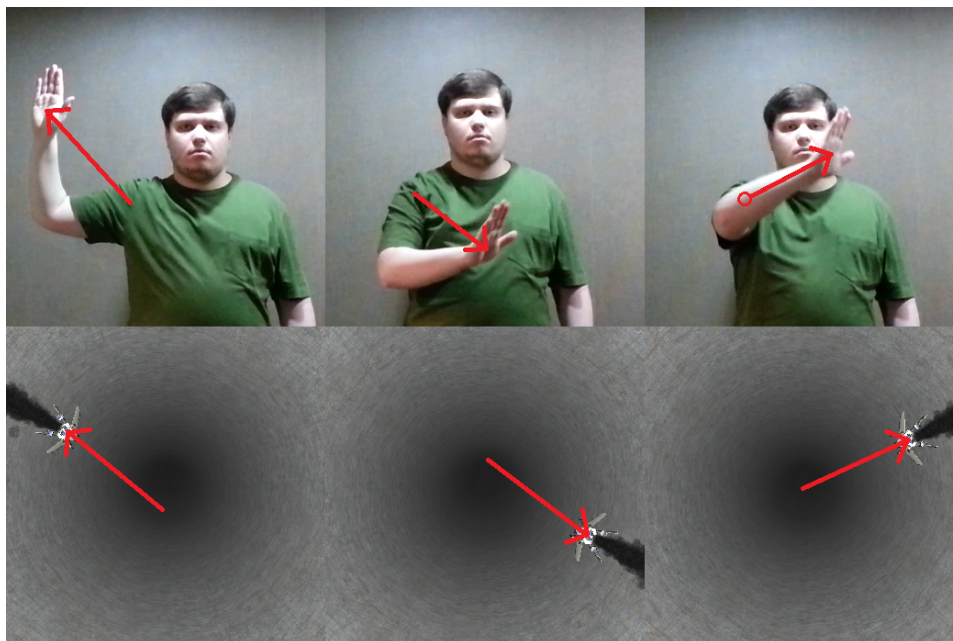


Figura 4.6: Variação do gesto com o ombro e mão. Ao articular o cotovelo é possível bloquear a vista do ombro, como visto na fotografia à direita.

Para a utilização do cursor dependendo apenas de um ângulo, seria necessário limitar sua posição de forma similar a aeronave, movendo-se em torno de uma região circular. A posição desse cursor foi utilizada na versão final do jogo, girando ao redor de um anel composto de cores, sendo que o centro do cursor representa o ângulo utilizado e a cor sob este centro é a cor selecionada por ele. Como não há motivos para utilizar algum dos braços para controlar alguma entidade em particular, foi desenvolvida uma tela anterior ao início do jogo em que o jogador deve selecionar qual braço será responsável por qual entidade (aeronave ou cursor). O gesto utilizado para a seleção resume-se em levantar o braço (coordenada Y da mão ser maior que a do ombro) a ser utilizado para controlar a aeronave por cerca de 1 segundo e meio. Como uma pequena adição ao jogo, também é possível jogar com duas pessoas em simultâneo ao levantar ambos os braços nessa tela, permitindo que cada jogador controle uma entidade com seu braço direito.

Para uma interação que permita o uso dos gestos abordados, desde o início do desenvolvimento do projeto foi decidido o uso do sensor Kinect. A biblioteca disponibilizada para o Unity utilizando os recursos do Microsoft SDK permite a conversão dos diferentes pontos capturados pelo Kinect em objetos do jogo, com suas respectivas coordenadas espaciais representando o jogador registrado.

4.2.2 Elemento educacional

Partindo do projeto original Audiathon, ele não contém nenhum elemento educacional aplicado. Devido à necessidade desta adição, algumas ideias foram exploradas.

Com o conceito de controlar 2 entidades adicionado, a primeira ideia consistiu no jogo agir como um exercício mental com múltiplas tarefas (ao controlar a aeronave e o cursor simultaneamente), com um conceito mais focado no aprimoramento do que o aprendido, similar ao jogo “Big Brain Academy: Wii Degree”, visto na seção 3.1.1. O controle de duas entidades simultâneas necessitou a diminuição significativa da quantidade de obstáculos, assim como da velocidade da aeronave, transformando o jogo que era mais focado

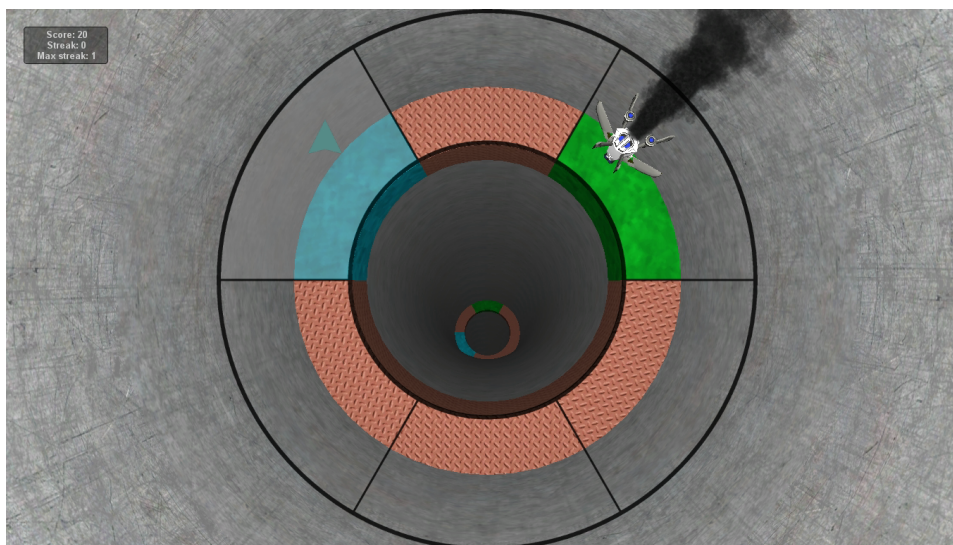


Figura 4.7: Kinaudiathon: o projeto intermediário, adaptado para Rainbow Riders of Kinect.

em reflexos rápidos em um jogo de coordenação e raciocínio (visto ao comparar a figura 4.1 do projeto original Audiathon e a figura 4.8 da versão final do jogo desenvolvido). A ideia foi considerada simples demais, necessitando algum elemento extra além da combinação e movimentação simultânea de entidades. A versão que utiliza apenas o exercício com múltiplas tarefas é a Kinaudiathon³. Uma imagem do jogo Kinaudiathon pode ser vista na figura 4.7.

O elemento considerado foi o casamento de padrões entre elementos selecionados pelo anel de texturas em que o cursor localiza-se e elementos selecionados pela aeronave. A proposta resumia-se ao primeiro grupo desses elementos representar algo que pudesse ser relacionado com o segundo, como forma de aprendizado infantil ao apresentar padrões que possuam uma relação importante entre eles. Vários pares de padrões foram propostos: “Animal macho e animal fêmea”, “ambiente e animal”, “operação matemática e resultado” e “figura de um objeto e palavra correspondente” (sugerindo objetos como animais, cores e formas geométricas). Todos esses conceitos foram considerados muito simples, mesmo para o âmbito infantil, e o meio de aplicá-los consistiria em utilizar texturas identificando tais conceitos, o que se tornaria problemático pelo pequeno espaço ocupado na tela pelos elementos em questão (texturas na HUD e obstáculos). Foi então decidido pela utilização de cores por ter seu valor identificável considerando o tamanho dos elementos. O próximo passo seria determinar que tipo de atividade educativa poderia ser realizada apenas com cores.

O resultado foi decidido como sendo a aritmética de cores, ou simplesmente a mistura de cores (cor 1 mais cor 2 igual à cor 3) seguindo a teoria das cores (modelo subtrativo). Originalmente planejava-se ilustrar a cor 1 e 2 por meio do anel de texturas que o cursor se posiciona, fazendo com que cada um desses elementos possuísse duas cores. Decidiu-se que tal formato seria visivelmente confuso, por resultar em muitas cores diferentes na tela. A decisão de usar outro elemento como o representante de uma dessas cores foi então decidida, com uma cor sendo definida automaticamente pelo jogo e pintada no cursor. Utilizar uma das cores como independente da escolha do jogador se mostrou ser

³O executável do jogo Kinaudiathon pode ser baixado pelo seguinte link: <https://www.dropbox.com/s/xjpay5rd221vsxy/Kinaudiathon.zip?dl=1>.

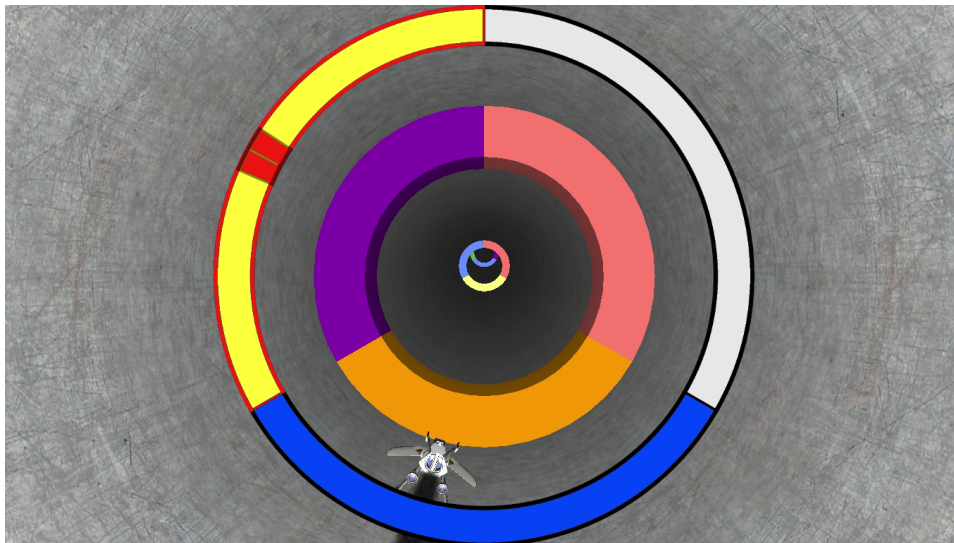


Figura 4.8: Aparência do jogo Rainbow Riders of Kinect.

uma decisão de projeto interessante, por forçar o jogador a tentar mais combinações por possuir uma cor fora do controle do jogador. Para um incentivo maior, foi decidido alterar a posição de todos os elementos a cada rodada, resultando em um cursor com uma cor sempre diferente da anterior, um anel de texturas selecionando novas cores em diferentes posições a cada rodada e um anel de obstáculos também com posições de cores diferentes da rodada anterior.

Herdeado das versões anteriores, o jogo portanto consistia de rodadas em que o jogador deveria selecionar uma cor entre 6 cores no anel de texturas, combiná-la com a cor do cursor e finalmente controlar a aeronave até a cor que corresponde à mistura (de acordo com o modelo subtrativo) dessas duas cores. O grande revés dessa versão é o alto número de cores diferentes (herdeado do projeto intermediário Kinaudiathon), consistindo de 7 cores para os dois termos da aritmética de cores (1 cor para o cursor mais 6 para o anel de texturas), totalizando 21 cores como combinações dessas 7 cores iniciais, sendo algumas delas repetidas. Apesar das cores repetidas nunca ocorrerem em uma mesma rodada, ainda assim muitas das cores de uma mesma rodada tendiam a ser visualmente muito semelhantes umas às outras, tornando a identificação da solução mais difícil, especialmente por se tratar de um jogo infantil. Dessa forma, decidiu-se utilizar apenas 4 cores como termos, sendo elas as cores primárias da teoria das cores (azul, amarelo e vermelho) com a adição da cor branca, resultando em 6 cores como combinações dessas 4 cores iniciais. Essa mudança foi considerada adequada, pois apesar do jogo apenas ser capaz de ensinar 6 combinações em sua versão atual, seria trivial alterar as cores utilizadas para ensinar outras combinações em particular⁴. Uma imagem do jogo Rainbow Riders of Kinect pode ser vista na figura 4.8.

O modelo final do jogo portanto permite o aprendizado sobre a mistura de 4 cores, que ao explorar todas as combinações em quantidade suficiente leva o jogador a não apenas aprendê-las, como também exercita essa informação de forma a otimizar a rapidez com que o jogador a acessa em sua mente (pelo limite de tempo imposto para as respostas e pela prática por meio da repetição).

⁴O executável do jogo Rainbow Riders of Kinect pode ser baixado pelo seguinte link: <https://www.dropbox.com/s/q56baqy9864rfh8/RRoK.zip?dl=1>.

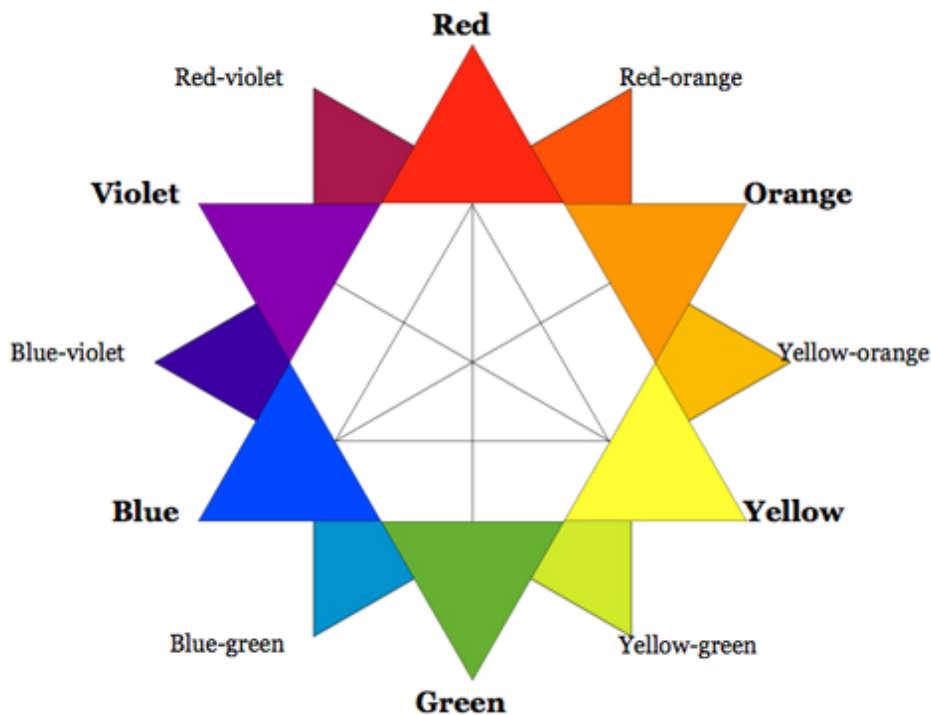


Figura 4.9: “Roda de cores” utilizada na teoria das cores. Exibe as 3 cores primárias (azul, amarelo e vermelho), as 3 cores secundárias (verde, violeta e laranja), assim como as 6 cores terciárias. A roda não exibe o resultado de misturas com a cor branca, presente na versão do jogo usada para os testes com o usuário (Cameron Chapman, 2010).

4.3 Resumo do funcionamento

O jogador ao iniciar o programa deve selecionar a resolução de tela (procedimento padrão dos aplicativos compilados em Unity). Ao abrir a janela do jogo, uma tela instrui o jogador a decidir com qual braço controlará a aeronave e o cursor. A escolha é feita levantando o braço a ser utilizado para controlar a aeronave por cerca de 1 segundo e meio. É também possível levantar ambos os braços para escolher o modo de dois jogadores, no qual cada jogador controla um dos elementos apenas com o braço direito. O jogo tem início logo em seguida, informando o modo de jogo selecionado.

Rainbow Riders of Kinect é um jogo em que o objetivo é combinar cores e escolher uma cor resultado que representa a mistura das cores selecionadas seguindo o modelo subtrativo. A mistura seguindo esse princípio consiste basicamente na forma considerada a mais natural, que se baseia na mistura de pigmentos de diferentes cores, seguindo os tons da “roda de cores”, como visto na figura 4.9. Essa é a forma mais comum da mistura de cores, normalmente ensinada às crianças, pois representam o resultado aproximado da mistura de 2 tons de tinta em uma atividade de pintura (BASIC COLOR THEORY, 2013).

O cursor do jogo inicia aleatoriamente com uma das 3 cores primárias (azul, amarelo ou vermelho) além da cor branca. O jogador deve então controlar esse cursor até uma das cores selecionáveis no anel de texturas (em que o cursor gira ao redor) e ao mesmo tempo selecionar a cor resultante, controlando a aeronave até o obstáculo de cor que represente a mistura das duas cores selecionadas (totalizando 6 cores possíveis de obstáculo). O jogo possui a duração da faixa de áudio que é executada ao longo do jogo, e seus obstáculos, assim como o tubo na qual a aeronave viaja, são gerados de acordo com o áudio (a gera-

ção é pré-processada por um programa externo), reagindo conforme sua intensidade. O controle é realizado através do Kinect, na qual o jogador deve controlar a aeronave e o cursor utilizando ambos os braços (cada um controla uma entidade). A posição de cada entidade é definida a partir do ângulo entre a mão e o ombro do braço, com o cursor e a aeronave imitando o mesmo ângulo do braço correspondente.

Inicialmente, o jogo considera um acerto com o valor de 10 pontos. A cada acerto sucessivo, o jogo adiciona um bônus de $2(s-1)$ pontos, sendo “s” o número de combinações consecutivas acertadas, com um bônus máximo de 10 pontos, totalizando em uma pontuação máxima por acerto de 20 pontos. Ao cometer um erro, o bônus é desconsiderado, voltando ao valor 0 para o próximo acerto.

O jogo conta com 2 graus de liberdade (1 para cada braço). Cada um representa o ângulo de cada entidade: um valor entre 0 e 360 graus⁵. A interação no jogo consiste do seguinte:

- Quando o jogador controla a aeronave ele está selecionando obstáculos por meio dela;
- Quando o jogador controla o cursor ele está selecionando texturas coloridas;
- A aeronave está navegando no tubo;
- O cursor está navegando no anel de texturas.

4.4 Detalhes de implementação

O projeto do jogo Rainbow Riders of Kinect envolve diversos compiladores, bibliotecas e linguagens de programação. Para que o jogo funcione corretamente, o computador utilizado precisa do sensor Kinect conectado a ele utilizando o adaptador com plugue para fonte de energia (a entrada USB não fornece energia para o Kinect, apenas serve como uma ponte para a troca de dados), assim como precisa possuir a instalação do Microsoft SDK de versão beta 2 ou superior⁶. Finalmente, o jogo requer 2 arquivos para o seu funcionamento: o arquivo contendo a estrutura do tubo em que a aeronave navega e o arquivo contendo a posição dos anéis de obstáculos presentes no tubo. Esses arquivos são gerados por um programa auxiliar, que dado o áudio utilizado no jogo, produz os 2 arquivos em questão. Os arquivos precisam provir do mesmo arquivo de áudio utilizado no jogo, pois a estrutura do tubo e posicionamento dos obstáculos são dependentes das amostras presentes nesse arquivo (o tubo reage ao ritmo da música presente no arquivo e os obstáculos tendem a se posicionar em batidas na percussão ou no início de nota musical). Dessa forma, Rainbow Riders of Kinect pode ser compilado por meio da engine Unity com qualquer música desejada.

A engine Unity “é um ecossistema de desenvolvimento: um poderoso motor de renderização totalmente integrado com um conjunto completo de ferramentas intuitivas e com um rápido fluxo de trabalho para criar conteúdo interativo em 3D” (CREATE THE GAMES YOU LOVE WITH UNITY, 2013). O desenvolvimento típico de uma aplicação em Unity consiste na colocação de objetos (*game objects*) em um cenário tridimensional (uma aplicação pode possuir diversos cenários). Cada *game object* é independente dos

⁵Duas amostras de jogo do Rainbow Riders of Kinect com músicas diferentes podem ser assistidas pelo seguinte vídeo: <http://www.youtube.com/watch?v=dA4j6XjccWI>.

⁶Uma versão do Microsoft SDK compatível pode ser obtida pelo seguinte link: <http://wiki.etc.cmu.edu/unity3d/images/4/4c/KinectSDK64.zip>.

demais (sendo possível afiliar um *game object* a outro, o que mantém sua posição relativa ao objeto “pai” como uma simples hierarquia em árvore) sendo possível a adição de componentes que caracterizam um objeto (apesar de que todos os *game objects* devem possuir o componente transform, que representa sua posição em relação ao cenário ou ao objeto pai). Alguns exemplos de componentes envolvem renderizadores de malhas, emissores de partículas, colisores de malhas, câmeras, emissores de áudio e execução de *scripts*⁷. Como cada *script* é dependente de um *game object* (apesar de poder possuir variáveis estáticas, que independem dele), o fluxo de execução da aplicação pode variar enormemente dependendo dos *game objects* ativos e a ordem de execução dos *scripts*. *Scripts* no Unity podem conter quaisquer funções ou métodos, porém existem nomes de funções que são reservados ao contexto de execução, sendo apenas essas as funções diretamente executadas pelo Unity (outras funções devem ser chamadas através delas). Alguns nomes reservados incluem *Awake*, *Start*, *Update* e *LateUpdate*, sendo as duas primeiras executadas apenas uma vez (ao ativar o *game object* com o *script* embutido) e as duas últimas executadas a cada quadro da aplicação (sendo que funções *LateUpdate* só serão executadas após todas as funções *update* terem sido executadas naquele quadro). Por padrão a ordem de execução de funções do mesmo tipo ou prioridade é arbitrária, porém é possível configurar uma ordem de execução de cada *script* ou *game object*, o que normalmente resulta em uma queda de desempenho da aplicação (SCRIPT EXECUTION ORDER SETTINGS, 2011).

4.4.1 Conversor de áudio

O conversor de arquivos de áudio é responsável pela geração do tubo e dos obstáculos presentes no jogo. Ele foi desenvolvido na linguagem C++ e utiliza a biblioteca *libsndfile* para a leitura das amostras de áudio presentes no arquivo de entrada. Os dois arquivos de saída gerados pelo conversor são vitais para funcionamento do jogo, por representarem todos os elementos do ambiente dele: o tubo e os obstáculos. O arquivo B (*board*) e o arquivo O (*obstacle*) usados com o sufixo do nome do arquivo de entrada (utilizado no Rainbow Riders of Kinect como *input*, o que resulta nos arquivos *inputB* e *inputO*), contém todas as informações necessárias para a formação do cenário do jogo, com exceção das cores do cursor, anel de texturas e anel de obstáculos, que são gerados pelo jogo.

O programa consiste de 5 funções, sendo que o núcleo da lógica da conversão encontra-se em 3 funções: *generateBoardFile*, *generateObstacleFile* e *main*. Basicamente a rotina principal abre o arquivo de entrada e o prepara de forma a gerar uma estrutura legível para as outras duas funções, que geram a informação necessária para o arquivo do tubo e o arquivo dos obstáculos. O tratamento da rotina principal consiste simplesmente em fundir os dois canais de um som estéreo (caso seja estéreo) em apenas uma faixa e obter a função derivada das amostras, registrando a diferença da magnitude do valor entre a amostra atual e a anterior: o “delta” positivo da faixa (que é considerado como sendo a intensidade de som no instante da amostra). A rotina também gera estatísticas da média do valor das amostras obtidas, assim como o valor mínimo e máximo das amostras, permitindo determinar o volume máximo do arquivo, sendo possível tolerar arquivos de entrada com um volume baixo.

A rotina *generateBoardFile* é responsável pela geração do arquivo contendo o tubo presente no jogo (arquivo B). Ela basicamente separa a sequência de amostras recebida em seções, calculando a média de intensidade sonora presente nelas, sendo esse o único valor necessário para a geração do tubo. As coordenadas de cada ponto do tubo são cal-

⁷*Scripts* são programas que podem interpretar e automatizar a execução de tarefas.

culadas a partir da intensidade sonora da seção, gerando pontos mais distantes do ponto anterior (que será traduzido no jogo como uma seção que deve ser percorrida em velocidade maior) e uma inclinação gradual do tubo para baixo em caso de uma seção com alta intensidade sonora. Caso a intensidade sonora seja baixa ou nenhuma, são gerados pontos próximos do ponto anterior (resultará em uma seção que deve ser percorrida lentamente) e uma inclinação gradual do tubo para cima. O tubo nesse momento é representado como uma lista de pontos, que ligados representam o centro do tubo em toda a sua extensão. O arquivo *B* (utilizado no Rainbow Riders of Kinect como *inputB.dat*) armazena as coordenadas de cada ponto gerador do tubo (exceto coordenadas *Z* por sempre serem 0). A primeira linha contém o cabeçalho do arquivo, informando o número de vértices declarados, o valor abstrato de precisão do tubo (maior precisão resulta em mais vértices e uma velocidade maior para a aeronave) e o número de vértices antes de iniciar a música (período preparatório para o jogador). Após o cabeçalho, segue a declaração dos dados, que são as coordenadas *X* e *Y* do centro do tubo.

A rotina *generateObstacleFile* é responsável pela geração do arquivo contendo os anéis de obstáculos presentes no jogo (arquivo *O*), meramente determinando a posição deles no tubo, sem detalhar a quantidade de obstáculos e suas cores. Ela basicamente divide as amostras em seções e calcula a FFT⁸ de cada uma (por meio da rotina *four1*), calcula a magnitude dos números complexos gerados e finalmente interpola as seções para obter o espectro de um dado momento do arquivo de áudio. Esse espectro é então comparado com o último espectro calculado de forma a tentar detectar um aumento súbito de energia no trecho (comparação realizada na rotina *getOutputObstacle*), o que resulta em obstáculos ocorrendo em momentos relevantes do áudio, como uma batida na percussão ou no início de uma nota musical. É relevante salientar que esse cálculo não ocorre em toda a extensão da música, pois ao gerar um ponto representando o anel de obstáculos, o programa é impedido de gerar outros anéis após um dado instante, que na configuração utilizada para os testes se traduz em 3 segundos⁹. O arquivo *O* (utilizado no Rainbow Riders of Kinect como *inputO.dat*) armazena a posição dos anéis de obstáculos presentes no tubo. Na primeira linha está o cabeçalho contendo a precisão utilizada para os obstáculos, que significa a quantidade de posições de anéis de obstáculos que podem ser inseridas entre 2 pontos do tubo (declarados no arquivo *B*), portanto sua precisão depende da precisão do tubo. Em seguida, a declaração de dados contém os anéis de obstáculos posicionados utilizando a medida abstrata da precisão dos obstáculos presente no cabeçalho. Por exemplo: um anel de obstáculos na posição 443 com uma precisão de obstáculos de 2 irá se posicionar entre os vértices do tubo de número 221 e 222. Um exemplo de arquivo *B* e arquivo *O* é ilustrado na figura 4.10¹⁰.

4.4.2 Rainbow Riders of Kinect

Rainbow Riders of Kinect é compilado utilizando diversos *scripts* na linguagem Javascript da *engine* Unity (Unityscript) para a lógica geral do jogo, além de *scripts* na linguagem C# como parte da biblioteca de desenvolvimento com o Kinect para Unity

⁸FFT ou *Fast Fourier Transform*, algoritmo de complexidade $O(N \log N)$ que calcula a transformada de Fourier.

⁹O executável do conversor de áudio pode ser baixado pelo seguinte link: <https://www.dropbox.com/s/71vymb7kv62g635/RRoK%20Converter.zip?dl=1>.

¹⁰Amostras de arquivos *B* e *O* podem ser obtidas pelo seguinte link: <https://www.dropbox.com/s/ikpwx64qkksxzc/Arquivos%20Input%20RRoK.zip?dl=1>.

Arquivo B	Arquivo O
header{661,7,60}	header{2}
data{	data{
0 (0.000000,0.000000)	122
1 (0.200000,0.000000)	188
2 (0.400000,0.000000)	251
3 (0.600000,0.000000)	313
4 (0.800000,0.000000)	374
5 (1.000000,0.000000)	443
6 (1.200000,0.000000)	507
7 (1.400000,0.000000)	568
8 (1.600000,0.000000)	630
9 (1.800000,0.000000)	716
10 (2.000000,0.000000)	778
11 (2.200000,0.000000)	840
12 (2.400000,0.000000)	904

Figura 4.10: Primeiras linhas do arquivo *B* e arquivo *O*, gerados pelo conversor e utilizados como entrada para o jogo Rainbow Riders of Kinect.

(que foi adaptada para o jogo), a “Kinect Wrapper Package for Unity3D”¹¹. A lógica do jogo se divide em vários pequenos *scripts* que controlam cada elemento do jogo, porém o núcleo do jogo encontra-se em 3 *scripts*: *Board*, responsável pela geração do tubo em que a aeronave navega a partir do arquivo de dados gerado pelo conversor; *Obstacle*, responsável pela geração e gerenciamento dos anéis de obstáculos (gerado pelo conversor), além de manter a pontuação do jogo e conseqüentemente o desempenho do jogador; *Ship*, responsável pelo movimento da aeronave e de tudo que é afetado por esse movimento (exceto o que já é gerenciado pelo *script Obstacle*).

O *script Board* é responsável pela geração do tubo em que a aeronave navega. Basicamente, o *script* abre o arquivo *B*, lê seu conteúdo, registra cada um dos vértices declarados e gera o tubo a partir desses pontos, dado um número de vértices por anel do tubo e seu raio. O tubo completo então é gerado, após a geração de todos os seus vértices, do vetor normal e vetor tangente de cada um desses vértices, da coordenada de textura de cada um desses vértices e dos triângulos declarados utilizando os vértices. O tubo é então dividido em seções menores para simplificar o processamento durante a execução do jogo, sendo enfim declarado como um *game object* do Unity, tornando cada seção do tubo como parte do cenário. O *script* ativa as duas primeiras seções do tubo, pois serão as duas únicas seções visíveis durante o trajeto da aeronave ao longo da primeira seção. A ativação e desativação de outras seções é realizada pelo *script Ship*. Nota-se que a ordem das operações poderia ser revertida, inicialmente repartindo os pontos declarados pelo arquivo *B* e então gerando a estrutura do tubo em seus respectivos *game objects*, o que pouparia processamento.

O *script Obstacle* posiciona os anéis de obstáculos declarados no arquivo *O*, assim como os colore e atualiza seus estados, além de armazenar e atualizar a cor do cursor e anel de texturas e calcular a pontuação do jogo. O *script* inicialmente lê o arquivo *O*, registra seu conteúdo e determina a pontuação máxima obtível pelo número de anéis de obstáculos presentes. Ele também calcula aleatoriamente as cores do cursor, anel de

¹¹A biblioteca “Kinect Wrapper Package for Unity3D” pode ser obtida pelo seguinte link: <http://wiki.etc.cmu.edu/unity3d/images/b/bd/KinectWrapperPackage.zip>.

texturas e obstáculos de forma a sempre existir uma cor do obstáculo que corresponda a mistura da cor do cursor e de uma das cores do anel de texturas. Na rotina de atualização que é executada a cada quadro, o *script* gerencia colisões com obstáculos, que ao detectadas iniciam novas rodadas (atualização de cores e atualização da pontuação do jogador). Ele também atualiza a visibilidade dos obstáculos dependendo da posição da aeronave (recurso para simplificar o processamento semelhante ao mencionado no *script Board*), tornando visível obstáculos que se aproximam da aeronave e destruindo obstáculos já ultrapassados por ela e fora do campo de visão. Nota-se que o *script* é responsável por diversas tarefas pois armazena variáveis altamente dependentes delas, porém o *script* poderia ter uma rotina semelhante ao *script Board*, apenas gerando as estruturas, designando o processamento durante o jogo para outro *script*.

O *script Ship* é responsável pelo movimento da aeronave e suas implicações. A cada quadro, o *script* é responsável por calcular o ângulo formado no braço da aeronave capturado pelo Kinect e alterar a posição da aeronave para esse ângulo. Ele também calcula o deslocamento longitudinal ao longo do tubo a partir do tempo transcorrido de forma que cada segmento entre 2 pontos do tubo (pontos originais do arquivo *B*) seja percorrido no mesmo intervalo de tempo. O resultado é um aumento da velocidade da aeronave em segmentos longos, que foram anteriormente calculados para serem longos devido à intensidade sonora do instante no arquivo de áudio (o que resulta na sensação da aeronave estar seguindo o ritmo da música). O *script* também sincroniza a posição da aeronave com o tempo de execução do áudio, aumentando ou diminuindo a velocidade da aeronave em caso de perda de sincronia. Ele também é responsável pelo gerenciamento das seções do tubo geradas no *script Board*, ativando seções que se aproximam e omitindo seções que já foram percorridas.

O jogo conta com outros 8 *scripts* não ligados à integração com o Kinect:

- *ControlSetType* é responsável pela tela inicial do jogo, gerenciando o conteúdo exibido nela e registrando a configuração escolhida pelo jogador (qual braço controla qual entidade, ou modo de dois jogadores);
- *CameraControl* é responsável pelo posicionamento da câmera, que basicamente segue a aeronave, além de possuir como objetos dependentes (exatamente na mesma posição da câmera) a fonte de iluminação e a fonte de áudio utilizada durante o jogo;
- *CursorControlGUI* é responsável por determinar a posição em que o cursor se posiciona a partir do braço do cursor do jogador, além de posicionar o anel de texturas corretamente na tela (centralizado e com o tamanho da menor das dimensões da tela);
- *MessageControl* é um simples *script* que exibe parte do texto visível no final do jogo (“Sua pontuação foi...”);
- *ScoreControl* exibe a pontuação do jogador com a cor de seu prêmio (nenhum, bronze, prata ou ouro), além de executar um som único para cada prêmio;
- *MedalControl* exibe qual o prêmio foi obtido, caso exista;
- *StreakControl* exibe o maior número de combinações corretas consecutivas;
- *InfoControl* informa o jogador que ao pressionar qualquer tecla o jogo será encerrado.

Os 11 *scripts* restantes são parte da biblioteca de desenvolvimento do Kinect para o Unity. Dos 11 é interessante destacar a função dos seguintes *scripts* (pois foram diretamente atribuídos a um *game object*):

- *DeviceOrEmulator* controla se o jogo irá receber entradas do sensor Kinect ou de um arquivo de registro de movimento (emulado);
- *KinectSensor* é responsável pelo funcionamento do sensor Kinect (camada mais externa do gerenciamento);
- *KinectEmulator* é responsável por executar arquivos de registro de movimento, sem a necessidade do sensor Kinect;
- *SkeletonWrapper* é responsável por gerar as informações de posição do jogador (seu esqueleto) a partir das imagens capturadas pelos 3 sensores;
- *DepthWrapper* obtém a imagem capturada pelo sensor de profundidade para um formato utilizável pelo Unity;
- *KinectRecorder* registra a posição do esqueleto em cada quadro quando ativado, armazenando em um arquivo de registro de movimento.

Também é interessante salientar o *script KinectInterop*, embora não atribuído a um *game object*, é responsável pela obtenção de recursos provenientes do Kinect SDK, sendo acessado indiretamente por outros *scripts* (MICROSOFT KINECT - MICROSOFT SDK, 2012).

5 AVALIAÇÃO DA USABILIDADE DO SISTEMA

De forma a expor os indivíduos testados à mesma quantidade de informação, um protocolo de testes foi desenvolvido, promovendo condições de igualdade a todos. Hipóteses a respeito da usabilidade do jogo foram levantadas, assim como variáveis dependentes e independentes do experimento.

5.1 Hipóteses

Para demonstrar a usabilidade do jogo, foram levantadas 3 hipóteses:

- *H1*: O usuário aprenderá as combinações de cores apresentadas com o jogo;
- *H2*: O usuário irá se divertir com o jogo;
- *H3*: Crianças entre 6 e 10 anos serão capazes de compreender e ter coordenação suficiente para utilizar o jogo.

Para que um jogo educativo obtenha sucesso é necessário obter um balanço entre o elemento educacional presente nele e o entretenimento provido por ele. As duas primeiras hipóteses tratam desse aspecto e tem o objetivo de determinar se o jogo atinge um patamar mínimo em cada um dos atributos, suficiente para ser considerado um jogo (no contexto do entretenimento) e ser educativo. Por se tratar de um assunto elementar, o foco do jogo consiste em ensinar crianças. O desempenho das crianças será comparado com adultos para verificar se elas mantêm um desempenho semelhante ao dos adultos no jogo e que corresponda ao conhecimento do assunto (verificado por questionários).

5.2 Experimentos

Inicialmente é aplicado um questionário com perguntas a respeito de informações pessoais que podem afetar no desempenho do jogo, seguido de um questionário de múltipla escolha perguntando sobre todas as seis combinações de cores utilizadas na versão do jogo utilizada para o teste. Em todos os testes realizados, o questionário foi aplicado oralmente, não permitindo muito tempo para o raciocínio¹.

Após esta etapa é exibido um vídeo explicando os elementos do jogo, assim como a forma de jogar. O vídeo é verbalmente detalhado, porém constitui de apenas imagens do

¹O questionário de introdução pode ser visualizado no anexo I e também acessado pelo seguinte link: https://docs.google.com/forms/d/1H50_Cr5r4S091pUbK_CIZeYulM7t701esMJfb71dd-I/viewform.

jogo capturadas na tela, com algumas pequenas e grosseiras edições para salientar elementos em particular. Dessa forma, o vídeo não exibe um modelo visual de jogabilidade, cabendo ao usuário interpretar e intuitivamente encontrar um meio funcional de atingir a meta do jogo².

O teste prossegue para a etapa de treinamento por meio de um teste livre. O teste livre consiste em basicamente permitir ao jogador realizar tentativas utilizando exatamente o mesmo jogo que no teste final (também com a mesma música, formato do tubo e ordem dos obstáculos e cores). Após realizada a tentativa (não oficial) é perguntado se o usuário quer realizar o teste oficial ou se quer jogar mais uma rodada. Após decidir realizar o teste oficial, o usuário joga pela última vez, com esse resultado sendo o único a ser considerado.

Após os testes com o jogo, é apresentado o questionário de satisfação, contendo questões de opinião sobre a qualidade e dificuldade do jogo, assim como a auto-análise de desempenho do jogador (que pode ser inferida pelo sistema de “prêmios” apresentado no final do jogo, exibindo bronze, prata ou ouro dependendo do desempenho do jogador). Finalmente, o mesmo questionário de múltipla escolha sobre mistura de cores é aplicado para ser comparado ao desempenho no questionário de introdução³.

Para esse teste, correspondem as seguintes variáveis independentes (entre parênteses a hipótese correspondente):

- O teste realizado: introdução ou satisfação (*H1*);
- A faixa etária do jogador: criança ou adulto (*H3*).

Ao realizar o teste de satisfação, o jogador deve pontuar mais nas questões referentes à mistura de cores que em seu teste de introdução de forma a provar *H1*. A faixa etária do jogador é o fator crucial para a prova de *H3*, pois é um meio eficaz de determinar a capacidade das crianças testadas. Por esse motivo o número de crianças e adultos testados deve ser semelhante para uma prova mais precisa⁴.

Algumas variáveis dependentes também foram levantadas, entre elas destaca-se o desempenho nos questionários realizados e a pontuação no jogo, que são cruciais para a prova de *H1* e *H3*. A comparação entre o jogador antes de testar o jogo e depois de testá-lo é importante para *H1*, e a comparação de desempenhos entre crianças e adultos é importante para *H3*. As respostas em relação à qualidade do jogo e os vídeos de registro dos testes com a reação dos jogadores testados contribuíram para a prova de *H2*.

Para *H1* e *H3*, também é possível conceituar os devidos grupos de controle e grupos experimentais. Sendo *H1* uma comparação entre o desempenho inicial e final no questionário sobre mistura de cores, o grupo de controle consiste nas entradas dos questionários de introdução, representando o conhecimento dos usuários antes da experiência com o jogo. O grupo experimental portanto consiste nas entradas dos questionários de satisfação, representando o jogador após o conhecimento adquirido pela experiência. Dessa forma, o grupo de controle e experimental em relação à *H1* contém “n” entradas cada, sendo “n” o número de pessoas testadas, resultando em “2n” entradas. Sendo *H3* uma

²O vídeo demonstrativo exibido aos usuários pode ser acessado pelo seguinte link: http://www.youtube.com/watch?v=2wY_4AsD-GM.

³O questionário de satisfação pode ser visualizado no anexo II e também acessado pelo seguinte link: https://docs.google.com/forms/d/11CZA-9J9H1pJLcc_Vdmq6dV71_HMGI51KDOWLUGmk5M/viewform.

⁴Uma amostra demonstrando o protocolo que foi de fato utilizado no experimento pode ser assistida pelo seguinte vídeo: www.youtube.com/watch?v=wOiu2NbIOU.

comparação entre o desempenho relativo de crianças e adultos testados, o grupo de controle consiste nos adultos testados e o grupo experimental nas crianças testadas, resultando em “n” entradas, porque cada pessoa representa apenas uma entrada.

6 RESULTADOS

O jogo *Rainbow Riders of Kinect* foi desenvolvido e testado conforme planejado. Com o funcionamento descrito no capítulo 4, foram testadas 14 pessoas, sendo 7 crianças e 7 adultos. Uma ideia geral dos ambientes de teste e dos indivíduos testados é ilustrada pela figura 6.1.

Considerando as hipóteses propostas na seção 5.1 e os grupos de teste propostos na seção 5.2, podemos concluir que foram obtidas 28 entradas, sendo 14 de cada grupo para a verificação da hipótese $H1$, e foram obtidas 14 entradas, sendo 7 de cada grupo para a hipótese $H3^1$.

Dos dados obtidos, é relevante mencionar os seguintes (entre parênteses os resultados obtidos pelo experimento):

- Idade (metade possui idade menor que 10 anos e metade possui idade maior que 21 anos);
- Gênero (8 pessoas do gênero masculino, 6 pessoas do gênero feminino);
- Mão dominante (12 destros e 2 canhotos);
- Frequência que joga vídeo games (figura 6.2);
- Experiência com o Kinect (figura 6.3);
- Desempenho nos questionários de introdução e satisfação (figura 6.10);
- Número de tentativas antes de realizar o teste final (3 pessoas preferiram uma segunda tentativa, o restante realizou apenas uma);
- Pontuação no jogo após as tentativas (figura 6.10);
- Mão usada para controlar a aeronave (apenas uma pessoa controlou-a com a mão esquerda, o restante com a direita);
- Opinião sobre a qualidade do jogo (figura 6.4);
- Auto-avaliação do jogador a respeito do jogo (figura 6.5);
- O que foi mais “legal” e mais “chato” no jogo (figura 6.6 e 6.7);
- O que foi mais difícil e mais fácil no jogo (figura 6.8 e 6.9).



Figura 6.1: Imagens presentes nos vídeos registrados durante o experimento. O jogo se mostrou atrativo para as crianças.

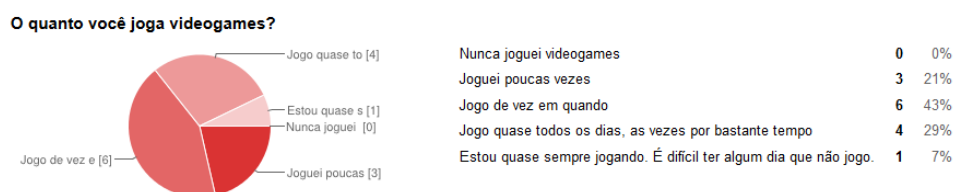


Figura 6.2: Estatística referente à frequência com que os usuários jogam vídeo games.

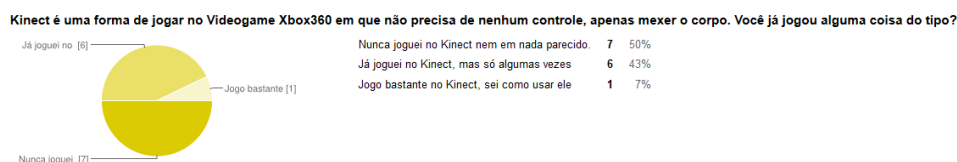


Figura 6.3: Estatística referente à experiência em jogos com o Kinect.

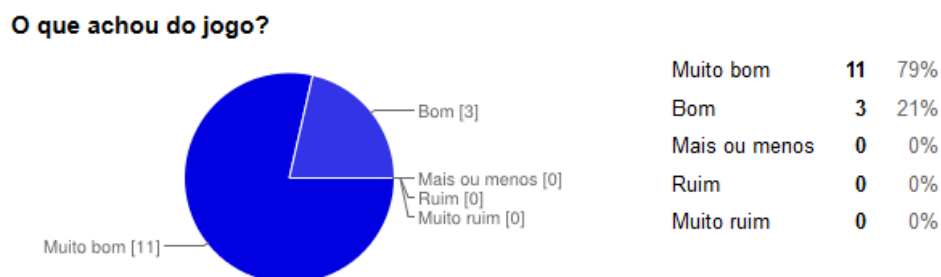


Figura 6.4: Estatística referente à satisfação dos usuários em relação ao jogo.

Como você achou que foi no jogo?

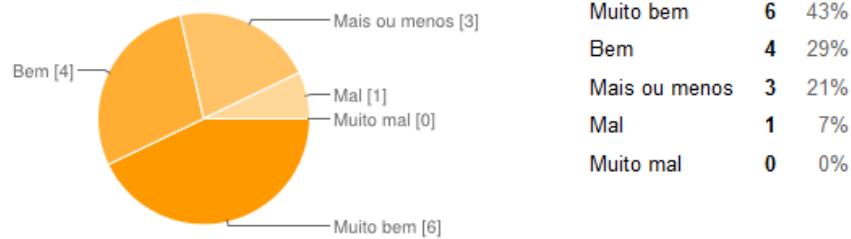


Figura 6.5: Estatística referente à auto-avaliação dos usuários do experimento.

Qual a coisa mais legal que achou no jogo?

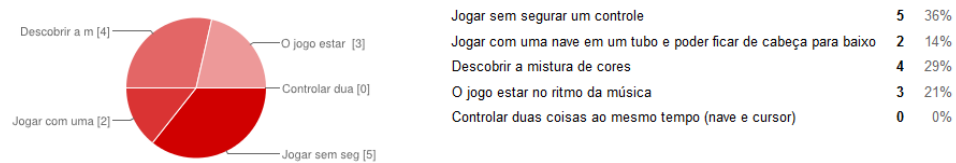


Figura 6.6: O que foi mais “legal” dentre as 5 opções disponíveis, de acordo com os usuários do experimento.

Qual a coisa mais chata que achou no jogo?

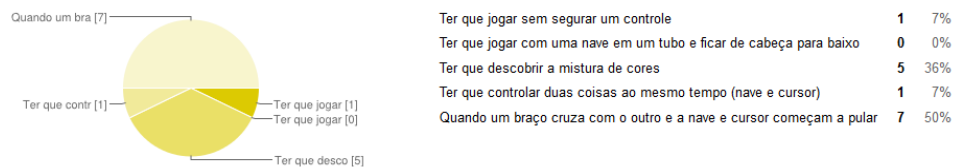


Figura 6.7: O que foi mais “chato” dentre as 5 opções disponíveis, de acordo com os usuários do experimento.

O que você achou mais difícil no jogo?

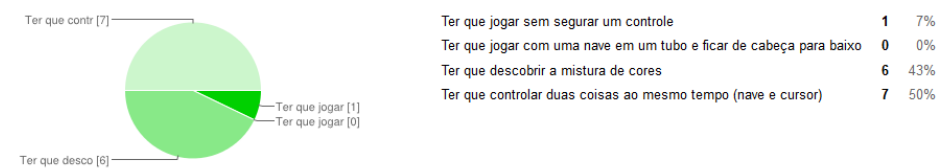


Figura 6.8: O que foi mais difícil dentre as 4 opções disponíveis, de acordo com os usuários do experimento.

O que você achou mais fácil no jogo?

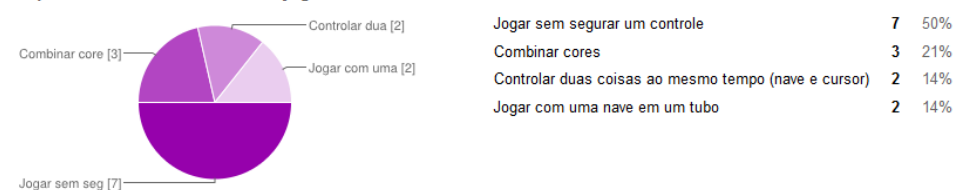


Figura 6.9: O que foi mais fácil dentre as 4 opções disponíveis, de acordo com os usuários do experimento.

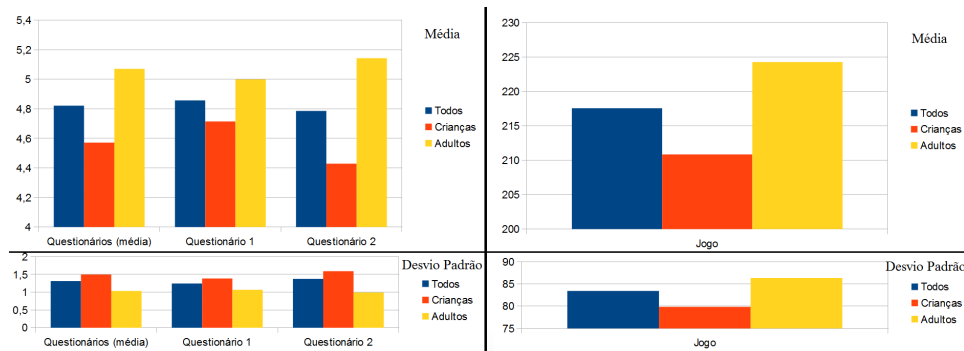


Figura 6.10: Estatística sobre as médias e desvios padrão obtidos dos diferentes testes e diferentes faixas etárias.

Além dos dados dos questionários, também foram registrados vídeos de cada teste realizado, assim como o registro de seus movimentos pelo Kinect para futura emulação²³.

Os dados de desempenho obtidos foram separados em 4 categorias: média dos 2 questionários, média do questionário 1, média do questionário 2 e média do jogo. Cada um desses valores foi determinado para todos os jogadores e para o grupo dos adultos e crianças individualmente. Essas informações podem ser visualizadas na figura 6.10⁴.

6.1 Contexto dos experimentos

O ambiente de teste não foi o mesmo para todos os usuários. 5 dos 7 testes realizados em crianças foram realizados em uma casa de festas, o que resultou em muito ruído sonoro e visual, somado ao contexto de descontração do evento. Por esse motivo foi decidido o uso de fones de ouvido para todas as etapas do teste com presença de áudio (vídeo explicativo, testes livres e o teste oficial), o que possivelmente removeu a liberdade de movimentos dos usuários (pela presença dos cabos dos fones de ouvido). Para manter a consistência, o uso de fones de ouvido foi aplicado mesmo em ambientes sem poluição sonora. Adicionalmente, a poluição visual ao redor do usuário (capturada pelo Kinect) pode também ter influenciado nos testes, tornando mais difícil a delimitação do jogador, assim como o mapeamento dos pontos relevantes do corpo a serem calculados pelo Kinect.

Observando os dados em vídeo, assim como os registros de movimento pelo Kinect pode-se chegar às seguintes constatações:

- Jogadores tendem a escolher uma cor em particular (selecionando-a sempre que possível) e acabam evitando exercitar as outras combinações existentes, sendo evidente a preferência da cor branca, por possuir cores resultantes mais fáceis de se induzir;

¹A tabela com as informações dos testes dos 14 participantes pode ser acessada pelo seguinte link: <https://docs.google.com/spreadsheet/ccc?key=0Aj2N6uGtIHWpdHhoYUhSNWU5UWZCY2owX2ZubnlpaHc&usp=sharing>.

²Uma montagem com alguns dos usuários testados pode ser assistida pelo seguinte vídeo: <http://www.youtube.com/watch?v=rrE12FgrP10>.

³Os vídeos brutos de 9 dos 14 usuários testados podem ser obtidos pelo seguinte link: <https://www.dropbox.com/s/cbdpr59m19h35nb/Usuarios%20Experimento.zip?dl=1>.

⁴As estatísticas brutas dos experimentos podem ser visualizadas pelo seguinte link: <https://www.dropbox.com/s/pitghtm4ixb88hy/Estatisticas%20Experimento.txt?dl=1>.

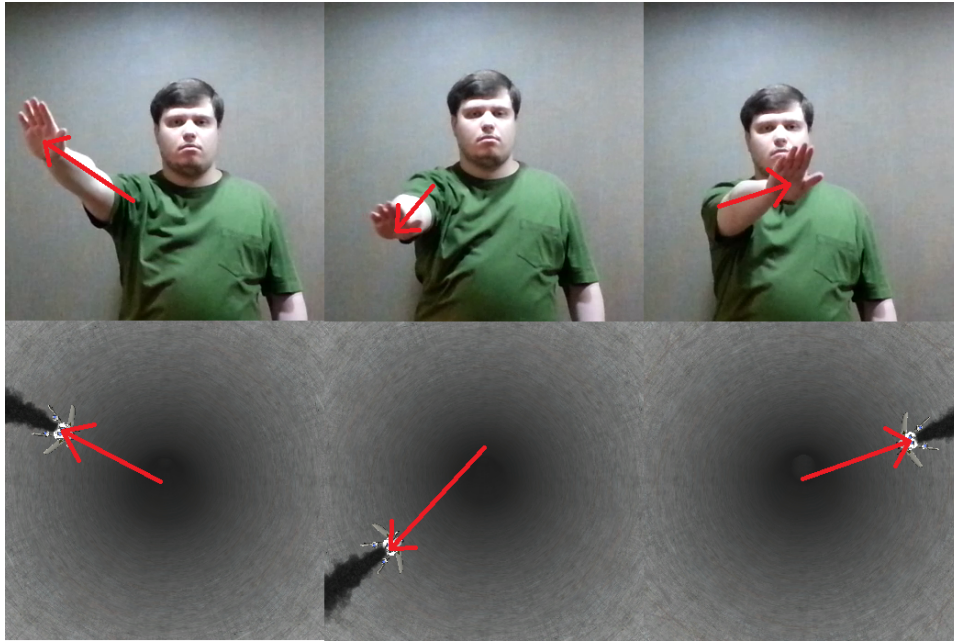


Figura 6.11: Gestos utilizando ombro e mão utilizados de forma incorreta, diminuindo a distância entre os dois pontos pela perspectiva do Kinect e potencialmente afetando o posicionamento da aeronave.

- Obstáculos gerados para a música em particular facilitaram ainda mais o uso da cor branca, sendo ela durante vários anéis de obstáculos permanecendo como textura inferior a ser selecionada pelo cursor (ou sendo a cor do cursor), o que não incentiva o movimento da mão que o controla;
- Houve também uma preferência a uma posição específica para o cursor, resultando em um baixo movimento do braço que controla o cursor;
- Mesmo em condições ideais, o Kinect não consegue posicionar a aeronave e o cursor corretamente quando algum dos braços bloqueia a visualização de um dos ombros ou do outro braço, muitas vezes resultando na aeronave e cursor se movendo de forma irregular na tela;
- Uma das consequências de não exibir uma ilustração de como jogar foi o fato de alguns jogadores não estenderem os braços para os lados de forma que o Kinect identificasse corretamente a posição das mãos e dos ombros (resultando na posição do ombro e da mão muito próximos na perspectiva do Kinect, como visto na figura 6.11).
 - Idealmente tanto a mão quanto o ombro deveriam pertencer a um mesmo plano XY , com a coordenada Z (profundidade detectada pelo Kinect) constante além de não flexionar o cotovelo para uma melhor precisão do sensor.

As observações realizadas acima se confirmam ao comparar o questionário, que somado ao fato de ter sido realizado oralmente, também resultou em uma média de acertos menor após o jogo. Isso provavelmente se deve pelo não exercício de todas as combinações de cores possíveis, assim como pelo tempo limitado de uso e exercício da mistura de cores.

6.2 Constatações

Com a análise estatística dos dados obtidos, nota-se o efeito oposto do esperado entre as 14 entradas do questionário de introdução e de satisfação, o que não prova *H1*. A versão atual do jogo mostrou-se ineficiente em ensinar a mistura de cores seguindo o modelo subtrativo. A queda de desempenho dos indivíduos testados é pequena, sugerindo um desempenho similar, não significando uma queda significativa de rendimento. Razões para esse resultado foram discutidas, e assume-se que o jogo foi incapaz de incentivar o jogador a explorar todas as combinações de cores possíveis, resultando no jogador apenas escolhendo combinações já conhecidas por ele. Ao ser forçado a exercitar todas as combinações de cores em quantidades significativas, somando ao *feedback* sonoro de acerto e erro de combinações, espera-se que o jogador possa eventualmente aprender todas as combinações utilizadas no jogo de forma intuitiva.

A partir das observações realizadas, o jogo resultou em reações mistas dos adultos, porém é visível pelo registro em vídeo que o jogo possui um fator de entretenimento positivo com as crianças. Na pergunta realizada no questionário de satisfação a respeito da qualidade do jogo, a predominante maioria classificou o jogo como “muito bom”, o melhor item da escala. A partir desses dados, é possível provar parcialmente *H2* para indivíduos de qualquer faixa etária, porém de acordo com os testes é possível afirmar que *H2* é verdadeiro especificamente com as crianças.

De acordo com os dados exibidos na figura 6.10 percebe-se que o desempenho no jogo das crianças foi pior em cerca de 6% em relação aos adultos, o que é uma diferença pequena considerando que seu conhecimento sobre a mistura de cores foi sensivelmente menor (desempenho pior em cerca de 10% em relação aos adultos). Das crianças testadas, uma alcançou a pontuação máxima em seu teste definitivo (além de uma segunda criança ter obtido a pontuação máxima em sua tentativa preliminar, não considerada em nenhuma das estatísticas) sugerindo que o jogo é jogável por crianças da faixa etária analisada: entre 7 e 9 anos (que pode ser estendida para 7 e 10 anos). Consequentemente, *H3* é parcialmente provada, ao não ser possível concluir de acordo com o teste que uma criança de 6 anos seja capaz de compreender os conceitos do jogo e ter coordenação suficiente para jogá-lo.

7 CONCLUSÕES

Este trabalho apresenta um meio de interação gestual aplicável ao sensor de movimentos Kinect para a seleção de elementos dispostos de forma circular (no caso, no interior de um tubo), assim como um meio de ensino infantil sobre a mistura de cores através de um jogo eletrônico educativo. Os resultados do experimento com os usuários (vistos no capítulo 6) demonstraram o elemento de entretenimento presente no jogo, assim como sua compatibilidade com o público infantil. Apesar disso, o jogo ainda requer uma série de modificações para ser efetivo no ensino sobre a mistura de cores (ao incentivar ou forçar o jogador a realizar todas as combinações presentes da forma mais homogênea possível). O estudo dos gestos para o controle no interior de uma superfície tubular (visto na subseção 4.2.1) se mostrou efetivo em demonstrar as restrições de gestos possíveis para obter uma translação ou rotação fluente e intuitiva para o usuário através de gestos realizados com os braços.

Dos pontos negativos destacados nos capítulos anteriores, sugere-se mudanças a fim de incentivar o jogador a explorar todas as combinações possíveis de cores (de preferência da forma mais homogênea possível), de forma a intuitivamente aprendê-las. Primeiramente, a combinação de cores utilizando branco (e possivelmente outras cores que venham a ser utilizadas) tornam o resultado visualmente mais simples para o jogador alcançar (não havendo desafio). Uma sugestão para evitar esse problema seria que cada cor possuísse chances diferentes de serem escolhidas como a cor do cursor, evitando cores com combinações consideradas mais fáceis (como as que combinam com o branco) de serem frequentemente selecionadas.

O principal revés do jogo parece ser sua insensibilidade ao contexto de jogo do jogador, ao não registrar combinações mais utilizadas e ser incapaz de detectar que o jogador está limitando suas escolhas a uma pequena gama de combinações dentre as possíveis. Um meio de evitar esse problema é colher estatísticas das combinações escolhidas, assim como das posições do cursor e aeronave selecionadas. Utilizando as estatísticas, uma ação possível seria remover de 1 a $n - 1$ cores do anel de texturas em que o cursor se posiciona (sendo “ n ” o número máximo de texturas no anel de texturas). Isso forçaria a seleção de combinações a exigir (eventualmente) todas as combinações possíveis ao limitar o jogador a escolher combinações nunca usadas (pela ausência de outras texturas selecionáveis no anel de texturas). Ao remover as posições mais utilizadas pelo jogador ao controlar o cursor, também é possível incentivar o jogador a mover o braço do cursor mais frequentemente ao apenas permitir mover o cursor para posições menos utilizadas. Uma probabilidade P de não desenhar a textura de cor C na posição N poderia ser aplicada para o desvio da frequência caso a seleção da cor ou da posição não fosse homogênea (por exemplo: com 3 texturas utilizadas, a frequência homogênea é de $1/3$, o que resultaria em frequências de escolha da cor C e frequências de escolha da posição N , caso superiores a

1/3, possuam uma chance de não ocorrerem). Ao incentivar o jogador a escolher combinações menos frequentes, espera-se que o jogo possua um valor educacional relevante e possa de fato ensinar as combinações àquele que o jogue, além de incentivar o movimento do braço do cursor, que na maioria dos testes realizados pouco se moveu.

A mistura de cores como elemento educacional do jogo também permite a modificação do modelo que a mistura é submetida. *Rainbow Riders of Kinect* foi originalmente planejado para utilizar a mistura de cores natural (resultado ao misturar pigmentos, modelo subtrativo), porém também pode ser utilizado para ensinar a mistura de cores seguindo o modelo aditivo (ao misturar a luz de diferentes cores ao invés de pigmentos) ou qualquer outro modelo concebível. Isso é possível porque as cores resultantes não são calculadas automaticamente, permitindo ao desenvolvedor definir qualquer cor como o resultado de 2 outras cores. Outro recurso que pode ser aproveitado é a capacidade de criar novos estágios de jogo ao utilizar músicas diferentes, o que permite que o usuário possa usufruir do jogo por mais tempo antes de se tornar repetitivo. O jogador poderia escolher jogar entre músicas de sua preferência em uma lista ou até mesmo utilizar o seu arquivo de áudio como entrada e decidir ouvir a música que quiser enquanto desafia um ambiente de jogo que reage a ela.

Há também a possibilidade de integrar o jogo ao Kinect 2, assim que suas bibliotecas para desenvolvimento sejam disponibilizadas. Seu uso permitiria uma maior precisão nos movimentos, mais detalhes nos modelos gerados, maior campo de visão (permitindo gestos como caminhar, que é mais conveniente devido ao aumento de espaço mapeável), entre outros melhoramentos em relação ao Kinect utilizado para esse trabalho (Kyle Wagner, 2013).

REFERÊNCIAS

ACESSINHA - AcessaSP. <http://www.acesasp.sp.gov.br/acesinha/>.

BASIC Color Theory. <http://www.colormatters.com/color-and-design/basic-color-theory>.

BRAIN Age: train your brain in minutes a day images. <http://www.ign.com/images/games/brain-age-train-your-brain-in-minutes-a-day-nds-740437/4fa6ca2dc388ed13ece723>.

Cameron Chapman. **Color Theory for Designers, Part 1: the meaning of color.** <http://www.smashingmagazine.com/2010/01/28/color-theory-for-designers-part-1-the-meaning-of-color/>.

CLUEFINDERS Year 4 Adventures - Puzzle of the Pyramid. <http://www.avanquest.com/UK/software/cluefinders-year-4-adventures-puzzle-of-the-pyramid-51591>.

CREATE the games you love with Unity. <http://unity3d.com/unity/>.

DANCE Central Images. <http://www.ign.com/images/games/dance-central-xbox-360-77445>.

Denise Deveau. **Spongelab goes for the love of the biology game.** <http://www.cbc.ca/money/smallbusiness/story/2010/09/08/f-videogames-spongelab-gdl.html>.

DESTRUCTOID review: big brain academy: wii degree. <http://www.destructoid.com/destructoid-review-big-brain-academy-wii-degree-32966.phtml>.

ECOQUEST The Search for Cetus. http://sis.mob.org/game/ecoquest_the_search_for_cetus.html.

FIVE Games That Need to be On Wii U! <http://geekpreview.blogspot.com.br/2011/10/five-games-that-need-to-be-on-wii-u.html>.

IMAGENS: kinect adventures (xbox 360) - uol jogos. <http://jogos.uol.com.br/xbox360/galerias/kinect-adventures.htm>.

James Newton. **New Puss in Boots Trailer Rubs Up Against You.** http://www.purexbox.com/news/2011/09/new_puss_in_boots_trailer_rubs_up_against_you.

Jami Ferguson. **D23 Expo: kinect disneyland adventures.** <http://whysoblu.com/d23-expo-kinect-disneyland-adventures/>.

KINECT sports para xbox 360. <http://www.dicas10.com/kinect-sports-para-xbox-360/>.

Kyle Wagner. **[Vídeo] Kinect 2, passo a passo: o xbox vê você como nunca antes.** <http://gizmodo.uol.com.br/video-kinect-2-passo-a-passo/>.

LET'S Play Super Solvers Series by IcePotato - Part 22: gizmos and gadgets pt 2. <http://lparchive.org/Super-Solvers-Series/Update%2022/>.

LIST of educational video games. http://en.wikipedia.org/wiki/List_of_educational_video_games.

LIST of Kinect games. http://en.wikipedia.org/wiki/List_of_Kinect_games.

Malcolm Owen. **E3 2011 - Forza Motorsport 4 Fact Sheet.** <http://www.gametactics.com/2011/06/e3-2011-forza-motorsport-4-fact-sheet/>.

Matt Casamassina. **Hands-on Big Brain Academy: wii degree - the hit ds franchise makes its way to home console, but what're the differences?** <http://www.ign.com/articles/2007/05/25/hands-on-big-brain-academy-wii-degree>.

MICHAEL Jackson The Experience Kinect. <http://www.familyfriendlyvideogames.com/MichaelJacksonTheExperienceKinect#.Uc3GK210n3Q>.

MICROSOFT Kinect - Microsoft SDK. http://wiki.etc.cmu.edu/unity3d/index.php/Microsoft_Kinect_-_Microsoft_SDK.

PLAYBOOK: sesame street: once upon a monster. <http://learningworksforkids.com/playbooks/sesame-street-once-upon-a-monster/>.

PREVIEW: the gunstringer. <http://www.destructoid.com/p-the-gunstringer-195007.phtml>.

Rafael Nink. **GCompris versão para Windows.** <http://rafaelnink.com/blog/2008/02/22/gcompris-versao-para-windows/>.

Robert Cong, Ryan Winters. **How It Works: xbox kinect.** <http://www.jameco.com/Jameco/workshop/howitworks/xboxkinect.html>.

Sasha A. Barab, Brianna Scott, Sinem Siyahhan, Robert Goldstone, Adam Ingram-Goble, Steven J. Zuiker, Scott Warren. Transformational Play as a Curricular Scaffold: using videogames to support science education. **Journal of Science Education Technology**, [S.l.], v.18, n.4, p.305–320, August 2009.

Scott Laidlaw. **Ko's Journey Math Game.** <http://www.youtube.com/watch?v=FfwNqNK1JWk>.

SCRIPT Execution Order Settings. <http://docs.unity3d.com/Documentation/Components/class-ScriptExecution.html>.

SESSAO Joguinho: kinect disneyland adventures. http://www.youtube.com/watch?v=t2_WrgBeN4s.

"SMOOTH CRIMINAL"Michael Jackson The Experience (Xbox360 Kinect) - Mighty-MeCreative. 2011.

Stephen Totilo. **Review:** kinectimals. <http://www.kotaku.com.au/2010/11/review-kinectimals/>.

WOLF Quest game: 1o. episódio. <http://wolfquestgame.blogspot.com.br/p/2-episodio.html>.

Yid Yang. **Where in the World is Carmen Sandiego?** http://www.mobygames.com/game/genesis/where-in-the-world-is-carmen-sandiego_/screenshots/gameShotId,40396/.

ANEXO I

Questionário de introdução

Questionário de Introdução ao RRoK

Rainbow Riders of Kinect

* Required

Vamos conhecer um pouco sobre você

Responda as seguintes perguntas:

Quantos anos você tem? *

Você é... *

- Menino ou homem
- Menina ou mulher

Qual a mão que usa para escrever? *

- Direita
- Esquerda
- As vezes uma, as vezes outra (ambidestro)

O quanto você joga videogames? *

Qual a resposta que chega mais perto?

- Nunca joguei videogames
- Joguei poucas vezes
- Jogo de vez em quando
- Jogo quase todos os dias, as vezes por bastante tempo
- Estou quase sempre jogando. É difícil ter algum dia que não joga.

Kinect é uma forma de jogar no Videogame Xbox360 em que não precisa de nenhum controle, apenas mexer o corpo. Você já jogou alguma coisa do tipo? *

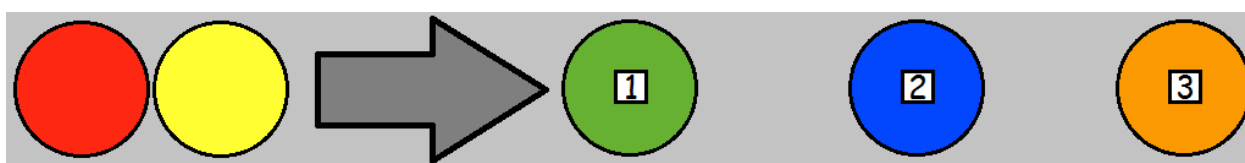
Qual a resposta que chega mais perto?

- Nunca joguei no Kinect nem em nada parecido.
 - Já joguei no Kinect, mas só algumas vezes
 - Jogo bastante no Kinect, sei como usar ele
-

Vamos ver o que você sabe sobre mistura de cores.

Qual o número no quadrado com a cor que você acha que é o que se consegue ao misturar as duas cores antes da seta na mesma linha?

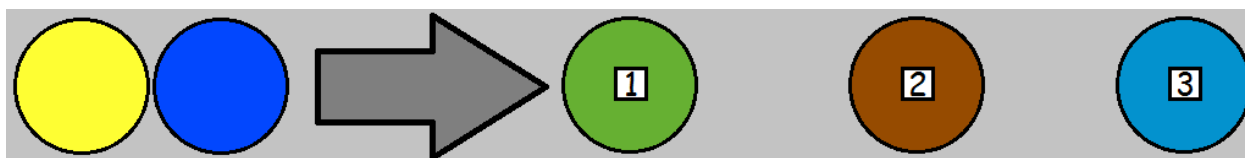
Vermelho + Amarelo



*

- 1
 - 2
 - 3
-

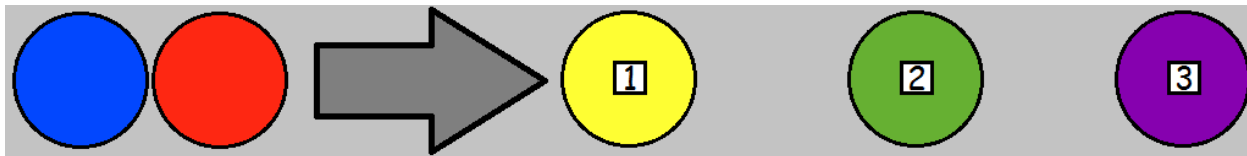
Amarelo + Azul



*

- 1
 - 2
 - 3
-

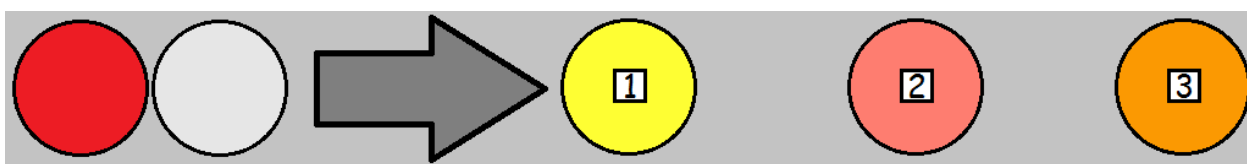
Azul + Vermelho



*

- 1
- 2
- 3

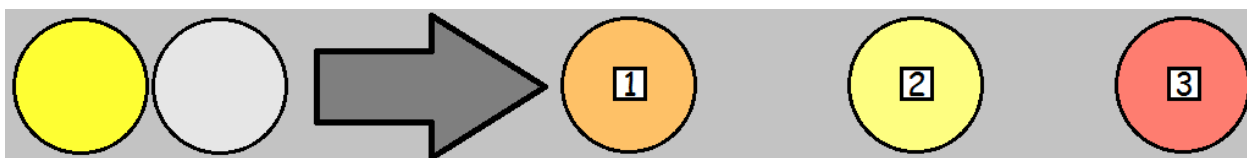
Vermelho + Branco



*

- 1
- 2
- 3

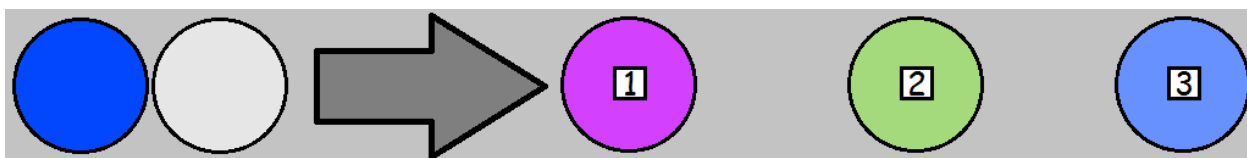
Amarelo + Branco



*

- 1
- 2
- 3

Azul + Branco



*

- 1
- 2
- 3

Submit

Never submit passwords through Google Forms.

Powered by


This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

ANEXO II

Questionário de satisfação

Questionário de Satisfação do RRoK

Rainbow Riders of Kinect

* Required

Dê sua opinião sobre o jogo

Responda as seguintes perguntas:

O que achou do jogo? *

- Muito bom
- Bom
- Mais ou menos
- Ruim
- Muito ruim

Como você achou que foi no jogo? *

- Muito bem
- Bem
- Mais ou menos
- Mal
- Muito mal

Qual a coisa mais legal que achou no jogo? *

Se não tiver o que pensou, escolha a mais legal das 5

- Jogar sem segurar um controle
- Jogar com uma nave em um tubo e poder ficar de cabeça para baixo
- Descobrir a mistura de cores
- O jogo estar no ritmo da música
- Controlar duas coisas ao mesmo tempo (nave e cursor)

Qual a coisa mais chata que achou no jogo? *

Se não tiver o que pensou, escolha a mais chata das 5

- Ter que jogar sem segurar um controle
- Ter que jogar com uma nave em um tubo e ficar de cabeça para baixo
- Ter que descobrir a mistura de cores
- Ter que controlar duas coisas ao mesmo tempo (nave e cursor)
- Quando um braço cruza com o outro e a nave e cursor começam a pular

O que você achou mais difícil no jogo? *

Se não tiver o que pensou, escolha o mais difícil dos 4

- Ter que jogar sem segurar um controle
- Ter que jogar com uma nave em um tubo e ficar de cabeça para baixo
- Ter que descobrir a mistura de cores
- Ter que controlar duas coisas ao mesmo tempo (nave e cursor)

O que você achou mais fácil no jogo? *

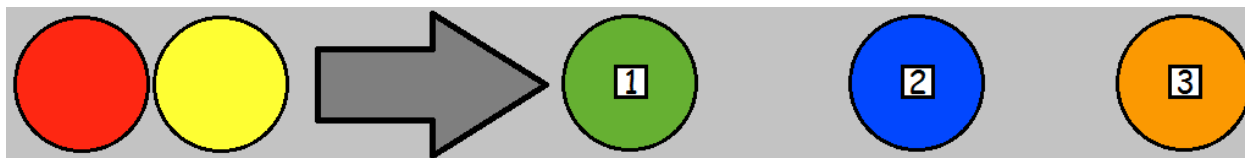
Se não tiver o que pensou, escolha o mais fácil dos 4

- Jogar sem segurar um controle
- Combinar cores
- Controlar duas coisas ao mesmo tempo (nave e cursor)
- Jogar com uma nave em um tubo

Vamos ver o que você aprendeu sobre mistura de cores.

Faça o mesmo que da primeira vez.

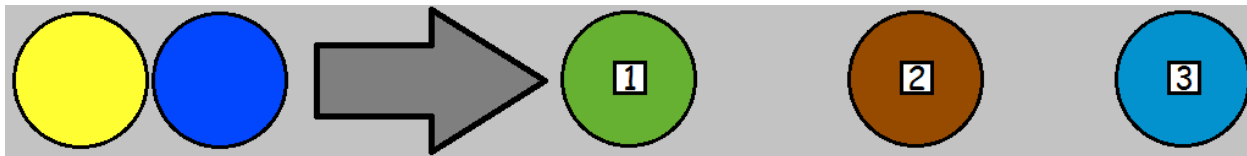
Vermelho + Amarelo



*

- 1
- 2
- 3

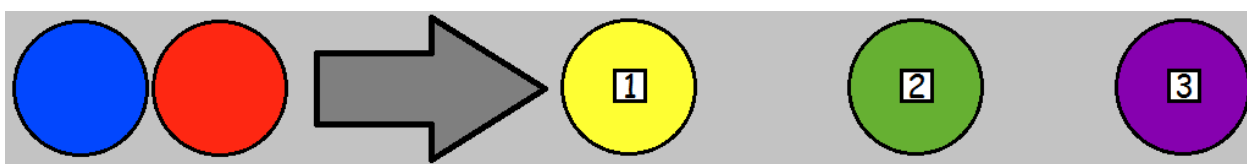
Amarelo + Azul



*

- 1
- 2
- 3

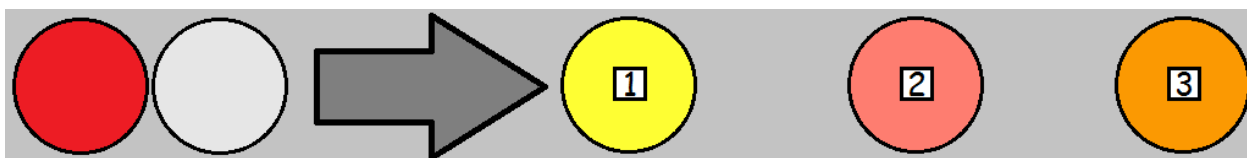
Azul + Vermelho



*

- 1
- 2
- 3

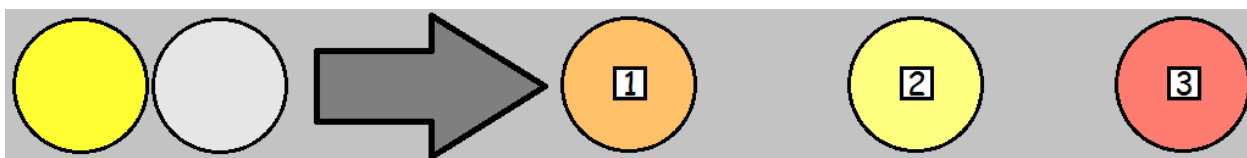
Vermelho + Branco



*

- 1
- 2
- 3

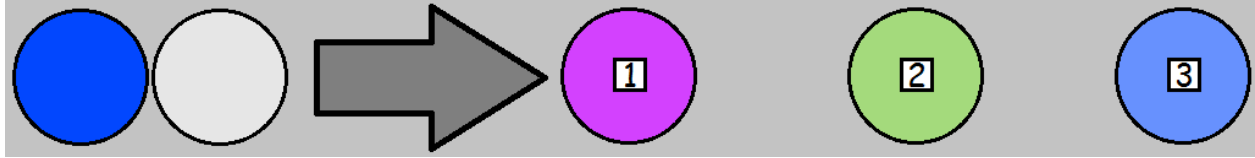
Amarelo + Branco



*

- 1
- 2
- 3

Azul + Branco



*

- 1
- 2
- 3

Submit

Never submit passwords through Google Forms.

Powered by
Google Drive

This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)