

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

FERNANDO JOSÉ ROMAN

**CAVE desktop para
jogos 3D em primeira pessoa**

Trabalho de Graduação.

Prof. Anderson Maciel
Orientador

Prof. Luciana Nedel
Co-orientador

Porto Alegre, Junho de 2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquíria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. João César Netto

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

Agradecimentos

Agradeço a minha família que sempre me apoiou e me deu condições para que eu pudesse realizar meus sonhos.

Aquela que esteve ao meu lado desde o nosso sempre me fortalecendo.

Ao meu orientador e a minha co-orientadora, pela colaboração.

Agradeço ao Instituto de Informática da UFRGS pela excelente condição de aprendizagem e recursos.

SUMÁRIO

| | |
|---|-----------|
| LISTA DE ABREVIATURAS E SIGLAS..... | 7 |
| LISTA DE FIGURAS | 8 |
| LISTA DE TABELAS | 9 |
| RESUMO | 11 |
| ABSTRACT..... | 13 |
| 1 INTRODUÇÃO | 15 |
| 2 TRABALHOS RELACIONADOS | 17 |
| 2.1 Jogos..... | 18 |
| 2.2 Cave | 21 |
| 2.2.1 Características de uma CAVE | 23 |
| 2.3 Realidade Aumentada | 24 |
| 3 PROJETO DE UMA CAVE DESKTOP..... | 27 |
| 3.1 Visão Geral do Sistema | 27 |
| 3.2 Parâmetros de Qualidade para Displays..... | 28 |
| 3.2.1 Aspectos de Imersão..... | 28 |
| 3.2.1.1 Campo de Visão..... | 28 |
| 3.2.1.2 Panorama | 29 |
| 3.2.1.3 Intrusão | 30 |
| 3.2.2 Aspectos de Visualização | 30 |
| 3.2.2.1 Acuidade Visual..... | 30 |
| 3.2.2.2 Olhar ao redor | 31 |
| 3.2.2.3 Colaboração | 32 |
| 4 IMPLEMENTAÇÃO DO SISTEMA | 33 |

| | | |
|------------|---|-----------|
| 4.1 | Softwares Utilizados | 33 |
| 4.1.1 | ARToolKit | 33 |
| 4.1.2 | AssaultCube..... | 34 |
| 4.2 | Hardware Utilizado | 36 |
| 4.3 | Mudanças no Software do Jogo..... | 36 |
| 4.4 | Calibração dos Monitores | 41 |
| 4.4.1 | Rastreamento e identificação por marcadores fiduciais | 41 |
| 5 | EXPERIMENTOS COM USUÁRIOS | 45 |
| 5.1 | Objetivos e Hipóteses | 45 |
| 5.2 | Planejamento e Realização dos Testes | 45 |
| 5.3 | Resultados Obtidos e Discussão | 46 |
| 5.3.1 | Precisão..... | 48 |
| 5.3.2 | Número de vezes que o jogador morreu | 48 |
| 5.3.3 | Número de inimigos que o jogador matou | 48 |
| 5.3.4 | Número de tiros que o jogador levou | 48 |
| 5.3.5 | Movimentação horizontal..... | 49 |
| 6 | CONCLUSÃO..... | 51 |
| | REFERÊNCIAS..... | 53 |
| | APÊNDICE A TEXTO PARA O TESTE..... | 55 |
| | APÊNDICE B QUESTIONÁRIO TESTE | 56 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|---|
| UFRGS | Universidade Federal do Rio Grande do Sul |
| BD | Banco de Dados |
| MIT | Massachusetts Institute of Technology |
| DEC | Digital Equipment Corporation |
| PDP-1 | Programmed Data Processor-1 |
| RPG | Role Playing Game |
| NES | Nintendo Entertainment System |
| FPS | First Person Shooter |
| RTS | Real-Time Strategy |
| HDM | Head-Mounted Display |
| CAVE | Cave Automatic Virtual Environment |
| AR | Augmented Reality |
| RA | Realidade Aumentada |
| Full HD | Full High Definition |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 2.1: Modelo Utah Teapot. | 18 |
| Figura 2.2: Head-Mounted Display binocular (HMD)..... | 21 |
| Figura 2.3: Um possível ambiente de CAVE | 22 |
| Figura 2.4: Esquema do HybridDesk. | 23 |
| Figura 2.5: Continuum de virtualidade..... | 24 |
| Figura 2.6: Elementos virtuais inseridos na cena | 25 |
| Figura 3.1: Sistema desenvolvido com três monitores..... | 27 |
| Figura 4.1: Pipeline de funcionamento do ARToolKit..... | 34 |
| Figura 4.2: Cena do jogo AssaultCube..... | 35 |
| Figura 4.3: Divisão da tela em três viewports | 37 |
| Figura 4.4: Três viewports como uma imagem contínua | 37 |
| Figura 4.5: Definição das três viewports | 38 |
| Figura 4.6: Definição de frustum..... | 39 |
| Figura 4.7: Ângulo entre os monitores (60°) igual ao ângulo do campo de visão do observador no jogo (60°) | 39 |
| Figura 4.8: Ângulo entre os monitores (30°) diferente do ângulo do campo de visão do observador no jogo (60°) | 40 |
| Figura 4.9: Monitores com marcadores e webcam..... | 41 |
| Figura 4.10: Exemplo de marcadores do ARToolKit..... | 42 |
| Figura 4.11: Relação do marcador com as coordenadas | 42 |
| Figura 4.12: Matriz de transformação | 43 |
| Figura 5.1: Gráfico que representa a experiência de cada pessoa com jogos..... | 46 |
| Figura 5.2: Gráfico que representa a habilidade de cada pessoa com jogos tipo FPS.... | 47 |
| Figura 5.3: Gráfico que representa a experiência das pessoas com aplicativos 3D | 47 |
| Figura 5.4: Gráfico que representa o modo de jogo que as pessoas mais gostaram..... | 49 |
| Figura 5.5: Gráfico que representa o modo de jogo que o as pessoas acham que jogaram melhor..... | 50 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 4.1: Distâncias máximas de identificação dos marcadores conforme seu tamanho | 34 |
|--|----|

RESUMO

Os jogos estão por toda a parte, e com a sensível melhora nos gráficos nos últimos anos, um novo desafio é o de criar interfaces melhores que ampliem a experiência sensorial dos jogadores. Nesse contexto, este trabalho propõe um sistema de CAVE para desktop fazendo uso de três monitores com calibração dinâmica do posicionamento e da angulação entre eles. A hipótese estudada é de que tal sistema ofereça uma ampliação da visão periférica de jogadores em um jogo 3D de tiro em primeira pessoa, o que eventualmente traria benefícios à performance do jogador. Na implementação foram utilizadas técnicas de realidade aumentada e visão computacional para fazer a calibração dos monitores. Bibliotecas gráficas (ARToolkit e OpenGL) foram utilizadas para detectar e calibrar os monitores em um espaço 3D e descobrir os ângulos entre eles. O jogo aberto AssaultCube foi modificado para suportar os três monitores e os ângulos entre eles. Este jogo ofereceu diversão e com ele foram feitos testes de adaptação do sistema proposto com o jogador. Os testes evidenciaram uma melhora na performance ao constatarem que o jogador move menos o mouse para olhar para os lados e consegue identificar inimigos como mais facilidade, uma vez que o campo de visão é maior.

Palavras-Chave: realidade aumentada, jogos, CAVE, Jogo de tiro em primeira pessoa, AssaultCube, ARToolkit.

CAVE desktop for first person shooter 3D games

ABSTRACT

Games are everywhere and, with the sharp improvement of graphics in the later years, a new challenge is to create better interfaces to amplify the sensorial experience of game players. In this context, the present work proposes a desktop based CAVE system using three video monitors with dynamic position and angle between them. Our hypothesis is that such a system provide an improvement in the players' peripheral vision for 3D first person shooting games. This would even benefit the player performance. In our implementation we used augmented reality and computer vision techniques to calibrate the monitors. Graphics libraries (ARToolkit e OpenGL) have been used to detect and calibrate monitors within the same 3D space and calculate the angles between them. The open source game AssaultCube has been modified to support three monitors and different camera angles between them. The game have shown to be entertaining and has been used as a use case for user tests. Tests have shown that the desktop CAVE system allows for performance improvement as the players make significantly less look- around movements with the mouse while keeping the average number of kills and deaths favorable in relation to a conventional one monitor setup.

Keywords: augmented reality, games, CAVE, first person shooter games, AssaultCube, ARToolkit.

1 INTRODUÇÃO

Os vídeo games estão cada dia mais presentes na casa das pessoas. Vemos propagandas na televisão, vídeos na internet e ficamos maravilhados com mundos virtuais impressionantes. Os jogos hoje oferecem gráficos quase reais e ambientes dinâmicos e cheios de aventuras. Queremos fazer parte desses mundos e os vídeo games são a janela para esses lugares. Porém, para termos uma experiência satisfatória com os vídeo games precisamos de interfaces de visualização tão boas quanto os jogos. A presente tecnologia nos oferece televisões e monitores com grandes resoluções. No mercado, dispositivos com resolução Full HD (Full High Definition) ou 1080p, que corresponde a 1920 por 1080 pixels, são cada vez mais comuns de serem encontrados.

Todos esses avanços contribuíram para aumentar a imersão do jogador no mundo virtual apresentado. Mas existem outras interfaces que podem aumentar ainda mais a imersão do jogador, como os HMDs (Head-Mounted Display), que necessitam de outros dispositivos para funcionar adequadamente, o que não ajuda na popularidade desta interface e as CAVEs (Cave Automatic Virtual Display) que ainda são uma inovação computacional que depende de muitos recursos financeiros, tanto para custo da aparelhagem como para espaço físico, o que inviabiliza um jogador comum de ter uma CAVE dentro de casa.

Com essa visão, esse trabalho propõe a implementação de um sistema fazendo uso de três monitores, com calibração dinâmica do posicionamento e da angulação entre eles, com a finalidade de constatar se há um aumento da imersão e uma melhora na jogabilidade em jogos de tiro em primeira pessoa. São observados aspectos como precisão de tiro, quantidade de vezes que o jogador morre ou mata os oponentes e movimentação horizontal, que é o movimento de olhar para os lados feito com o mouse. Uma melhora é esperada, pois com o aumento do ângulo de visão, o jogador terá sua visão periférica amplificada e poderá perceber a presença de inimigos mais rapidamente

do que faria com a limitação de ver o mundo virtual através de um único monitor. Além disso, essa técnica deve aumentar a imersão, proporcionando uma experiência mais agradável ao usuário.

A implementação do sistema é feita usando três monitores LCD convencionais para compor um multi-display em forma de CAVE simplificada com um monitor frontal e dois laterais. Os displays são calibrados usando um sistema de câmeras e técnicas de visão computacional e é avaliado o desempenho de jogadores usando o sistema de três monitores em comparação ao sistema tradicional de um monitor.

Nos próximos capítulos são apresentados os conceitos principais abordados no trabalho e como o sistema foi desenvolvido e avaliado. No capítulo 2 são apresentados trabalhos relacionados com as áreas de jogos, o sistema CAVE e realidade aumentada com um breve histórico e conceitos relacionados com essas áreas. No capítulo 3 é apresentado o sistema proposto neste trabalho, mostrando parâmetros qualitativos que guiaram o projeto e fazendo uma comparação entre o sistema proposto e outras interfaces de visualização. O capítulo 4 apresenta a implementação do sistema, mostrando os dispositivos de hardware necessários, como podemos usar os monitores para criar uma CAVE no desktop e como é feita a calibração dos monitores, bem como bibliotecas, jogo e ambiente de desenvolvimento utilizados. O capítulo 5 apresenta os testes realizados sobre o sistema, seus objetivos e como foram planejados e executados. Também possui uma análise dos resultados obtidos experimentalmente. Por fim, no capítulo 6 são apresentadas as considerações finais e os trabalhos futuros.

2 TRABALHOS RELACIONADOS

O desenvolvimento computacional gráfico iniciou há algumas décadas e vem evoluindo desde então. No mundo dos jogos essa evolução trouxe realismo para o mundo virtual, tornando esse mundo virtual cada vez mais realístico para o usuário. Isso é alcançado com gráficos mais fiéis e com a física dos objetos mais aprimorada. Além desses avanços surgiram interfaces de visualização e interação que vêm aumentando a imersão e a presença do usuário no ambiente virtual.

Na década de 50, pelo menos dois projetos foram de imensa importância para o surgimento da computação gráfica: o projeto Whirlwind, um computador desenvolvido no Massachusetts Institute of Technology (MIT) que operava em tempo real e mostrava a saída em um display de vídeo, e o projeto The Semi-Automatic Ground Environment (SAGE), um projeto militar americano de um sistema automatizado para procura e interceptação de bombardeiros inimigos. No final dessa mesma década, surgiu no MIT o computador TX-2 que integrava uma interface revolucionária para a época: um monitor CRT como entrada e saída de dados.

Porém, o termo *Computer Graphics* (Computação Gráfica em inglês) foi usado pela primeira vez por William Fetter em 1960. Este termo está relacionado com o processo de criação de imagens através do computador.

Em 1963, Ivan Sutherland, usando o computador TX-2, criou um software chamado Sketchpad, no qual uma caneta com luz poderia ser usada para desenhar figuras geométricas simples no computador. Este software fez de Ivan Sutherland o “pai” da computação gráfica e das interfaces gráficas.

Além de Sutherland, outros pesquisadores tiveram papel fundamental para o desenvolvimento da computação gráfica, como Jack Bresenham, o qual desenvolveu o algoritmo de traçado de linhas. Este algoritmo de linha determina quais pontos, em uma

matriz, devem ser preenchidos para formar aproximadamente uma linha reta entre dois pontos. A partir desse princípio, ele estendeu este algoritmo de linha para a construção de desenhos circulares. Depois desse advento, a computação gráfica evoluiu, passando de simples linhas e círculos para desenhos em duas dimensões.

John Warnock desenvolveu o algoritmo de determinação de faces ocultas, com o objetivo de identificar quais faces não são visíveis em um objeto quando uma terceira dimensão está presente. Com esse algoritmo em mãos foi possível o desenho de objetos em três dimensões. Um modelo criado em 1975 é muito conhecido na comunidade de computação gráfica: o Utah teapot ou Newell teapot (Figura 2.1), seu criador foi Martin Newell. Esse modelo é famoso pois na época que foi criado representava muitos desafios em um único objeto (superfícies curvas, bordas afiadas, furos, discontinuidades, etc.) e pesquisadores podiam usá-lo para testar seus sistemas gráficos em virtude de sua representação matemática estar aberta para o público.



Figura 2.1: Modelo Utah Teapot.

2.1 Jogos

O mercado de jogos, tanto para computadores, como para consoles, é um mercado bilionário. Cada vez mais pessoas e empresas tentam entrar neste ramo cada vez mais lucrativo.

Um grupo de estudantes também do MIT, usando um computador novo da época, o DEC PDP-1, criou o jogo Spacewar! em 1961, paralelamente a pesquisa de Sutherland, com seu Sketchpad,. Nesse jogo, dois jogadores humanos combatiam um contra o outro, cada um controlando uma nave capaz de lançar mísseis e entre os dois jogadores havia uma “estrela” para dificultar o combate. O jogo foi depois distribuído nos computadores

da DEC e se tornou o primeiro jogo que realmente influenciou pessoas não relacionadas com o projeto.

Já na década de 70, dita a “época de ouro”, os vídeo games realmente vieram para ficar. Em 1971 Nolan Bushnell e Ted Dabney criaram uma versão arcade do jogo Spacewar! e o chamaram de Computer Space. Foram fabricadas 1.500 máquinas arcade e o jogo foi lançado em novembro do mesmo ano. O jogo não teve muito sucesso devido à sua curva de aprendizado acentuada. Em 1972, foi lançado o Pong e esse sim é considerado o primeiro jogo arcade com uma grande popularidade. Pong é como um jogo de tênis em duas dimensões, onde dois jogadores tentam fazer com que o outro jogador erre a devolução da bola. Atari, a empresa que fez as máquinas de Pong, vendeu 19 mil delas. Também em 1972 foi lançado o Magnavox Odyssey, um console para ser usado em casa feito especialmente para jogos onde era necessário mudar chaves e jumpers para alterar a lógica do circuito. Esse tipo de console foi a primeira geração de consoles para jogos. Mas, a real “época de ouro” iniciou-se em 1978 com o lançamento de Space Invaders pela empresa Taito. Em contra partida, a Atari lançou em 1979 Asteroids tornando os jogos arcade muito populares. E eles se tornaram ainda mais populares com os jogos coloridos a partir de 1980, como Pac-Man. Esses já usavam consoles de segunda geração, em que o jogo era gravado em cartuchos com memórias ROM e então o console, com seu processador genérico, lia o jogo contido nesse cartucho.

A partir da década de 80 muitos gêneros de jogos apareceram no mercado. Jogos de aventura, jogos de luta, jogos de labirinto, jogos de plataforma como Donkey Kong, jogos de corrida, jogos tipo RPG, dentre outros. Não somente os jogos evoluíram como também as máquinas em que eles rodavam. Nessa mesma década, surgiram os computadores Commodore Vic20, Commodore 64, Apple II. Em 1984, os computadores tomaram o lugar dos consoles de segunda geração devido a falência de muitas empresas de jogos em 1983. O mercado de consoles reergueu-se em 1985 com a terceira geração de consoles com o lançamento do console de 8 bits da Nintendo: o Nintendo Entertainment System (NES). Ele vinha com o jogo Super Mario Bros e se tornou um sucesso imediato. Outros consoles também surgiram nesse momento, como o Sega Master System com o jogo Sonic, além do começo de muitas séries de jogos famosos como The Legend of Zelda e Final Fantasy. No final dos anos 80 e começo dos

anos 90, consoles de quarta geração foram apresentados, como o Sega Mega Drive e o Super NES ambos usando 16 bits. Neste momento os jogos tipo arcade começaram a cair na preferência da população, que preferia ficar em casa jogando em um console, uma vez que era mais barato e tinha menores dimensões.

Nos anos 90, os jogos sofreram uma revolução com a introdução de jogos com gráficos em três dimensões e o alavancamento de gêneros como First Person Shooter (FPS) ou tiro em primeira pessoa, Real-Time Strategy (RTS) ou estratégia em tempo real e outros. Muitos jogos foram lançados nesta década como SimCity, Ultima Online, Mortal Kombat e os precursores dos jogos tipo FPS: Wolfenstein 3D que foi lançado em 1992 pela idSoftware utilizando técnicas como *texture mapping* e Doom lançado em 1993 pela mesma idSoftware. No meio da década de 90, foram lançados os consoles de quinta geração com 32 e 64 bits como o Playstation e Nintendo 64 que aproveitavam o seu poder computacional para mostrar gráficos ainda melhores, como no jogo GoldenEye 007, o qual trouxe inovações para os jogos FPS, como mira telescópica, contagem de tiros na cabeça e missões baseadas em objetivos e Metal Gear Solid com seu rico enredo e pequenos filmes “in-game”.

No final dos anos 90, surgiram os consoles de sexta geração como o Dreamcast que trazia um modem embutido para jogar online. No ano de 2000, surgiu o Playstation 2 da Sony que se tornou o console mais vendido do mundo até hoje. Também surgiu o Xbox da Microsoft como concorrente direto do Playstation 2 e o GameCube da Nintendo. Diversos jogos foram fabricados para estes consoles, como Halo, Winning Eleven, Fifa 10, Call of Juarez, dentre muitos outros. Esse momento foi a grande arrancada dos jogos online, a Internet já estava espalhada pelo mundo todo e mais pessoas queriam poder jogar seus jogos preferidos com seus amigos, mesmo que estes estivessem muito longe fisicamente.

Os últimos consoles a chegarem no mercado foram o Xbox 360, o Playstation 3 e o Nintendo Wii, todos com poder de processamento gráfico muito grande, o que proporcionou aos criadores de jogos a oportunidade de criar jogos com qualidade superior. Gráficos quase reais, enredo rico e dinâmico, em que cada ação do jogador influencia no caminho seguido através do jogo.

2.2 Cave

Sutherland continuou sua pesquisa com computação gráfica e em 1966 criou o HMD (Head-Mounted Display) , um dispositivo eletrônico para ser usado na cabeça, como se fosse um capacete, com um pequeno visor para um olho (HMD monocular) ou para os dois olhos (HMD binocular) (Figura 2.2). Com esse aparelho ele criou a idéia de uma janela para um mundo virtual. Esta janela aumentou a imersão do usuário no mundo virtual. No entanto, sua desvantagem está na necessidade do uso de muitos dispositivos para funcionar, uma vez que, além do capacete, é necessário o uso de luvas para interagir com o mundo virtual, e também muitos fios para conectar o HMD e a luva.



Figura 2.2: Head-Mounted Display binocular (HMD).

Seguindo esta mesma idéia de janela para o mundo virtual, mas simplificando os dispositivos, Myron Krueger criou em 1975 o Videoplace, um laboratório onde a realidade virtual ficava ao redor dos usuários e respondia ao seus movimentos e ações. Tudo isso sem o uso de qualquer tipo de óculos ou HMD e de luvas especiais. Seu sistema usava projetores, câmeras e hardware especiais para projetar a silhueta do usuário no mundo virtual. As câmeras gravavam os movimentos dos usuários e essas imagens eram analisadas pelo computador e então suas ações eram transferidas para a silhueta no mundo virtual.

Porém, tanto o Videoplace como os HMD são limitados. O Videoplace mostra somente a silhueta do usuário e gráficos muito simples e o HMDs tem uma resolução muito baixa e permite somente um usuário interagir com o mundo por vez.

Pensando nessas limitações, no SIGGRAPH de 1992, foi anunciado a criação da primeira CAVE (Cave Automatic Virtual Display) pelo laboratório de visualização eletrônica da Universidade de Illinois em Chicago. Uma CAVE é uma pequena sala

onde três paredes e o chão funcionam como um monitor, assim imagens em alta definição podem ser exibidas e mais de um usuário pode interagir no mesmo mundo virtual.

Uma CAVE (Figura 2.3) é uma interface de realidade virtual (CRUZ-NEIRA, 1992) em que o usuário fica cercado por esse mundo virtual, ou seja, um ambiente em que três ou quatro paredes e o plano do chão, por vezes o teto, funcionam como um monitor. Isso concede ao usuário uma sensação de imersão muito grande no jogo, pois ele está cercado por esse mundo virtual. Seu design supera muitos problemas encontrados em outros ambientes de realidade virtual, como limitação de interação com somente um usuário e imagens em baixa definição. Outro ponto interessante é a possibilidade de ver os objetos por todos os lados.

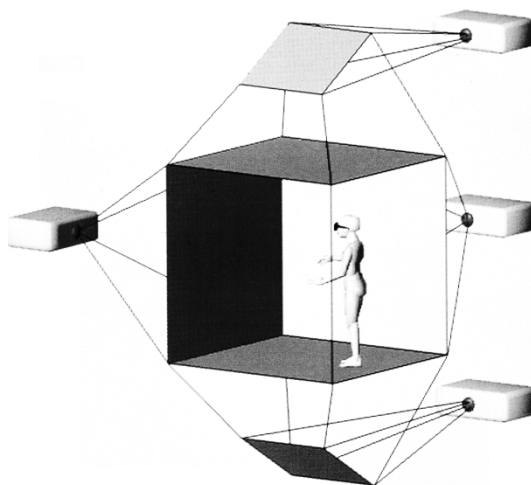


Figura 2.3: Um possível ambiente de CAVE (CRUZ-NEIRA, 1992)

Baseando-se nesse projeto de CAVE, o HybridDesk (Figura 2.4) partiu da motivação de criar um ambiente de trabalho que agregasse os recursos das tradicionais interfaces *WIMP* (*Windows, Icons, Menus and Pointer*) mais o acréscimo de outros recursos (hardware e software) que viabilizasse a realização de tarefas interativas 3D (CARVALHO, 2010).

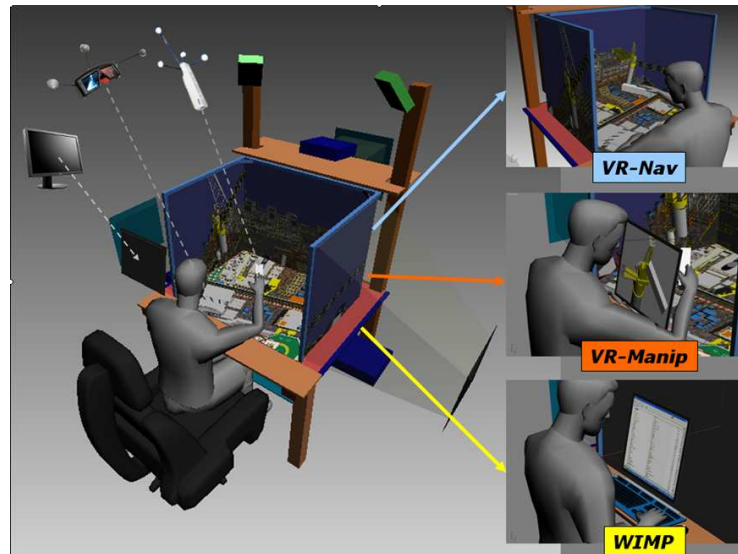


Figura 2.4: Esquema do HybridDesk (CARVALHO, 2010).

2.2.1 Características de uma CAVE

Cada parede é na verdade uma tela de retroprojeção e o usuário pode usar óculos 3D ou *shutter glasses* com imagens projetadas em formato estereoscópico, ou seja, são mostradas duas imagens para cada quadro da projeção, uma para o olho direito e outra para o olho esquerdo. Os óculos são sincronizados com a frequência da projeção, mostrando cada uma das imagens para o respectivo olho, bloqueando a outra. Com isso obtém-se a ilusão de uma imagem em três dimensões. É interessante ressaltar que usando imagens estereoscópicas é possível andar ao redor de objetos que estarão flutuando no ar.

Algumas idéias estão ligadas ao ambiente de CAVE. A idéia de suspensão de descrença é uma delas (CRUZ-NEIRA, 1992). Esse termo surgiu dos críticos do cinema e é definido como a vontade do espectador de acreditar nas imagens como sendo reais e desconsiderar o meio que o cerca. O espectador pensa dessa forma para substituir o julgamento da verdade pelo entretenimento. Outra idéia é a perspectiva centralizada no observador (CRUZ-NEIRA, 1992). Essa idéia é baseada na posição da câmera ao longo de um eixo estendido perpendicularmente a partir do centro da tela. A perspectiva centralizada simula a visão da posição do observador. Para mantê-la, um sensor deve constantemente mandar informações de posicionamento afim da simulação. Essas características são fundamentais para uma maior imersão no mundo virtual. Sem elas a CAVE não tem sentido.

Caves são usadas por várias áreas e estão presentes em muitas universidades pelo mundo. Muitos engenheiros usam Caves para melhorar o desenvolvimento de produtos. Por exemplo, para projetar uma fábrica ou prototipar uma peça, onde usa-se um ambiente de Cave para simular a fábrica ou peça, e dar uma visão mais precisa do resultado final, melhorando assim a confiabilidade no projeto. Ou projetar um carro e poder vê-lo pela perspectiva do motorista e do passageiro, bem como poder observar o seu redor e melhorar seu visual. Tudo isso sem a fabricação física de qualquer peça.

2.3 Realidade Aumentada

Devido à grande difusão de interfaces digitais como a televisão, os computadores, os celulares e *Smart Phones*, os vídeos digitais estão cada vez mais presentes em nosso cotidiano: vídeos capturados de eventos reais ou produzidos pelo computador. Em muitos deles temos a parte virtual misturada com imagens reais.

Realidade aumentada (RA) pode ser definida de várias formas. Algumas delas estão descritas como se segue.

Segundo Insley, RA é uma melhoria do mundo real com textos, imagens e objetos virtuais, gerados por computador (INSLEY, 2003).

Paul Milgram propôs um *continuum* de virtualidade (Figura 2.5), que varia entre um ambiente totalmente real e um ambiente totalmente virtual no outro extremo. Na verdade uma escala contínua inexistente que envolve todas as possibilidades de inserção de elementos virtuais no mundo real ou de elementos reais no mundo virtual (MILGRAM, 1994).

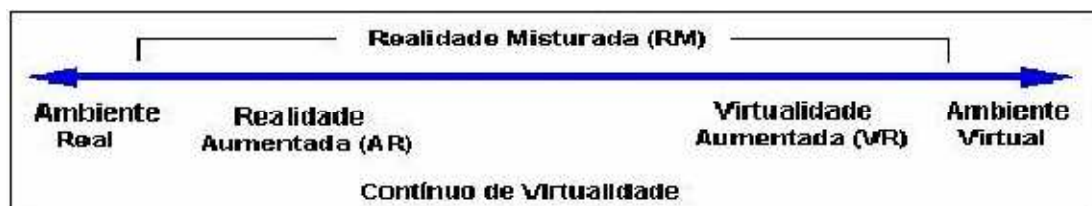


Figura 2.5: Continuum de virtualidade (MILGRAM, 1994).

A área entre os dois extremos é chamada de realidade mista ou misturada, que consiste na sobreposição de elementos virtuais gerados por computador com o ambiente físico, exibido ao usuário através de alguma interface de visualização. Essa realidade misturada pode ser dividida em realidade aumentada e virtualidade aumentada.

Realidade Aumentada consiste na inserção de objetos virtuais no ambiente físico, exibido ao usuário, em tempo real, com o apoio de algum dispositivo tecnológico, usando a interface do ambiente real (KIRNER, 2007). O termo realidade aumentada foi cunhado no começo da década de 90 por Thomas Preston Caudell enquanto trabalhava na Boeing. Ele foi responsável por criar um sistema em que o operário usava um HMD e mostrava os esquemas de fios de cada avião, substituindo assim as caras placas de madeira com os esquemas.

Virtualidade Aumentada é a inserção de representações de elementos reais no mundo virtual, usando a interface que permite ao usuário interagir com o ambiente virtual (KIRNER, 2007).

Já a definição feita por Azuma (AZUMA, 2001), fala que RA é um sistema que suplementa o mundo real com objetos virtuais gerados por computador, parecendo coexistir no mesmo espaço e apresentando as seguintes propriedades:

- combina objetos reais e virtuais no ambiente real;
- executa interativamente em tempo real;
- alinha objetos reais e virtuais entre si;
- aplica-se a todos os sentidos, incluindo audição, tato, força e cheiro.

Assim, podemos concluir que realidade aumentada permite um enriquecimento do ambiente real com elementos virtuais (Figura 2.6), esses elementos inseridos na cena podem ser de qualquer natureza, desde objetos tridimensionais que representam outra pessoa, por exemplo, a uma imagem representando o placar de uma partida de futebol.



Figura 2.6: Elementos virtuais inseridos na cena (KATO, 2000).

3 PROJETO DE UMA CAVE DESKTOP

Neste capítulo é descrito o sistema desenvolvido neste trabalho. Inicialmente será dada uma visão geral da concepção e funcionamento do sistema. Após, será feita uma comparação qualitativa entre o sistema proposto neste trabalho e algumas interfaces de visualização como um sistema com somente um monitor, um HMD e uma CAVE.



Figura 3.1 – Sistema desenvolvido com três monitores.

3.1 Visão geral do sistema

O sistema é constituído de três monitores LCD idênticos (Figura 3.1) e um computador com duas placas de vídeo. Duas placas são necessárias, pois atualmente somente é possível conectar dois monitores por placa de vídeo. Com os monitores conectados, configura-se o sistema operacional para considerar todos os monitores

como uma área de trabalho contínua. Depois, configura-se a resolução do jogo para utilizar todos os monitores. Isso é feito colocando a resolução horizontal como três vezes a resolução de um monitor e a resolução vertical permanecendo a mesma.

Com essas configurações feitas é preciso calibrar os monitores. A calibração é feita usando uma webcam. Posiciona-se a webcam em frente aos monitores de modo que ela consiga filmá-los por inteiro. Nos monitores são mostrados marcadores que serão identificados pelo software e assim são identificados os ângulos entre os monitores. Esses ângulos, então, são passados para o jogo, que se ajusta para mostrar a parte correta do mundo virtual.

3.2 Parâmetros de Qualidade para Displays

Todos os produtos no mercado têm parâmetros de qualidade normalmente testado por empresas especializadas. Num sistema de display, como uma CAVE, isso não é diferente. A motivação para sua construção está diretamente relacionada com a qualidade oferecida em comparação com outras interfaces de visualização virtual.

3.2.1 Aspectos de Imersão

Estes aspectos medem o nível de simulação visual que uma interface de realidade virtual pode proporcionar ao usuário: o nível de suspensão de descrença (CRUZ-NEIRA, 1992). Alguns itens para criar uma suspensão de descrença satisfatória são descritos a seguir.

3.2.1.1 Campo de Visão

O campo de visão (*FoV*) é o ângulo máximo que o observador enxerga sem mover a cabeça. Considerando uma tela com *W* polegadas de largura e sendo *D* a distância do observador até a tela, pode-se obter o campo de visão com a fórmula:

$$FoV = 2 \tan^{-1} \frac{W}{2D}$$

Se compararmos as principais interfaces de visualização: monitores LCD, um Head-Mounted Display (HMD), uma CAVE e três monitores interligados veremos cenários distintos. Um monitor LCD tem um campo de visão variado mas muito limitado, depende do seu tamanho e da distância do observador. Vendo um monitor de 19 polegadas a 18 polegadas de distância produz-se um campo de visão de

aproximadamente 45°. Em um HMD o campo de visão é fixo para cada olho, e os ângulos de 100° a 140° são comuns para esse tipo de interface. O campo de visão em uma CAVE é dependente da posição do observador em relação a cada tela, mas toda a interface oferece um campo de 360°, sendo esse o campo máximo de visão possível, o qual pode ser limitado se usar dispositivos como *shutter glasses*.

Neste trabalho o campo de visão é maior do que de um monitor LCD, pois serão usados três monitores lado a lado, aumentando, dessa forma, o ângulo de visão. Se compararmos o sistema com três monitores interligados com uma CAVE, o campo de visão será pior, pois não tem como ter um campo de visão de 360° com apenas três monitores. Porém, se mais monitores forem usados é possível se aproximar do campo de visão de uma CAVE. Os três monitores darão um campo de visão similar ao encontrado no HMD. No entanto, será dependente da posição do observador em relação a cada monitor, como em uma CAVE.

3.2.1.2 *Panorama*

A imersão também está diretamente ligada a capacidade da interface de cercar o usuário com o mundo virtual. Essa idéia é chamada de panorama e difere do campo de visão, uma vez que o movimento da cabeça é considerado.

Em uma interface tipo um monitor, a idéia de panorama não é muito aplicável, pois o monitor é mais uma janela para um mundo virtual e não cerca o observador de nenhuma maneira. Em um HMD o panorama é muito forte, pois tudo que o usuário observa é a simulação. Para qualquer lado que o observador se vire ele verá uma parte do ambiente virtual. Isso também acontece com uma CAVE, pois ele está cercado pelas projeções. Nesse caso, a sensação de imersão é melhorada, pois se o usuário movimentar a cabeça, a projeção já estará na tela, sem ter a necessidade de muitos cálculos. Já em um HMD é necessário uma quantidade maior de cálculos, para rastrear o movimento do usuário e saber qual parte do mundo mostrar ao usuário, e a renderização dessa parte, o que pode gerar alguns pulos nas imagens da simulação impactando negativamente na sensação de imersão.

Usando três monitores, o panorama será melhor do que apenas um monitor, mas ainda será restrito, pois será apenas uma janela maior para o mundo virtual.

3.2.1.3 Intrusão

A intrusão está ligada ao quão restritos os sentidos ficam ao usar uma determinada interface. O HMD é o mais intrusivo de todos, pois ele isola o usuário de todo o ambiente real. Se ele der um passo a frente, não sabe onde pisará. Alguns HMDs podem renderizar o ambiente real juntamente com o virtual, o que diminui a intrusão da interface, mas esta modificação restringe o campo de visão. O monitor e a CAVE são os menos intrusivos, pois o usuário pode mover-se livremente e manter consciência dos aspectos reais e virtuais do ambiente.

Usando o sistema de três monitores interligados, a intrusão se mantém ao mínimo, pois o usuário continuará com plena consciência do ambiente ao seu redor.

3.2.2 Aspectos de Visualização

Estes aspectos medem quão efetiva é a interface de visualização. Essa medida é importante pois a interface deve oferecer uma experiência visual satisfatória. Alguns destes aspectos são descritos a seguir.

3.2.2.1 Acuidade Visual

A qualidade de um *display* é geralmente medida pela sua resolução, ou seja, o número de *pixels*, ou pontos, que compõem a imagem. A qualidade de uma interface de realidade virtual é melhor medida usando-se a combinação de resolução e de campo de visão. Esta medida é chamada de acuidade visual do *display*.

Uma maneira de fazer esta medida é usando a resolução e a distância do observador até o centro da tela. Uma tela com H pixels de resolução horizontal e W polegadas de largura mostra um *pixel pitch* de $P=W/H$ polegas por pixel. Considerando uma distância D, o ângulo que um pixel representa na retina é $\tan^{-1}P/D$ e é medido em minutos. Assim, na acuidade visual, a porção de um pixel que ocupa um minuto da retina é dada pelo inverso desse ângulo: $1/\tan^{-1}P/D$ (CRUZ-NEIRA, 1992).

Outra medida utilizada para medir a visão é a fração de Snellen, 20/X. Essa fração mostra que um observador a 20 pés pode ver o que um observador com visão normal pode ver a X pés a olho nu. Por exemplo, uma pessoa com visão 20/40 pode ver os mesmos detalhes que uma pessoa normal veria a 40 pés, só que a 20 pés, ou seja, ela tem uma visão pior que a maioria das pessoas.

Se pegarmos, por exemplo, um monitor de 19 polegadas com resolução de 1280 x 1024 pixels e uma distância de aproximadamente 18 polegadas da tela, teremos uma acuidade visual de 20/45. Comparando com o código nacional de trânsito é uma acuidade visual insatisfatória para dirigir veículos leves, categoria “B”, pois a acuidade necessária é de 20/30. Em uma CAVE, se levamos em conta a resolução de cada projetor como 1280 pixels e uma tela de 7 pés, teremos uma acuidade visual de 20/110. Mas, se aumentarmos a resolução de cada projetor, teremos uma acuidade visual bem melhor. Se usarmos um HMD teremos uma piora significativa da acuidade visual. Passando para 20/425 devido ao tamanho da tela usada em um HMD e, devido também a sua resolução que é muito baixa. Uma CAVE não é tão ruim como um HMD, mas também não é tão boa quanto um monitor.

Se usarmos três monitores de 19 polegadas teremos uma acuidade visual igual à de um monitor, pois mesmo tendo aumentada a resolução total, aumentamos também o tamanho total da tela.

3.2.2.2 Olhar ao redor

Esta é a possibilidade de um observador de mover-se ao redor de um objeto, vendo-o por diferentes ângulos (CRUZ-NEIRA, 1992). Essa é uma propriedade muito útil em várias áreas. Ela torna possível uma melhor visualização do projeto antes que qualquer gasto com material físico seja feito.

Em um monitor LCD, a propriedade de olhar ao redor não funciona bem. Se formos para o lado do monitor veremos muito pouco da tela e se formos mais ainda para o lado a tela não poderá ser vista. Já usando um HMD isso não ocorre, pois quando o observador se move para o lado, o sistema recalcula sua posição em relação ao mundo virtual e mostra a parte correta. Em uma CAVE, essa propriedade é plena. Ao mover-se, o observador desloca seu campo de visão para diferentes áreas da projeção fazendo com que veja elementos do mundo virtual por uma perspectiva associada espacialmente com sua posição real. E se for usado *shutter glasses* a sensação é mais intensa pois os objetos estarão flutuando no ar, sendo possível vê-los por todos os lados.

No entanto, se usarmos três monitores, como este trabalho propõe, a propriedade de olhar ao redor fica melhor do que se for usado somente um monitor. Mas ainda assim se o observador se mover para o lado verá somente um monitor, se o ângulo entre eles for de 90°, ou verá apenas um pedaço de um dos monitores.

3.2.2.3 Colaboração

Um dos principais aspectos de visualização é a colaboração entre observadores, ou seja, a interface de visualização deve permitir que mais de um usuário use o ambiente virtual ao mesmo tempo. Em uma interface tipo monitor isso é possível, mas a perspectiva será a de apenas um dos usuários. O mesmo ocorre com uma CAVE, mas se forem usados *shutter glasses*, tanto na CAVE como no monitor é possível que cada observador veja o ambiente de sua perspectiva. Já em uma interface tipo HMD é possível haver colaboração caso haja mais de um dispositivo conectado no mesmo ambiente. No caso, é necessário que cada usuário esteja munido de um HMD e que o sistema possa gerar o rendering individual de cada usuário em tempo hábil. Por fim, se usarmos uma interface com três monitores, o nível de colaboratividade será como se estivéssemos usando apenas um monitor.

4 IMPLEMENTAÇÃO DO SISTEMA

4.1 Softwares Utilizados

Neste trabalho foram utilizados softwares auxiliares que não fazem parte do objetivo deste trabalho, mas são fundamentais para o funcionamento do sistema.

4.1.1 ARToolKit

A ARToolKit é uma biblioteca de desenvolvimento de aplicações bastante popular na comunidade de RA. É amplamente utilizada por ter o código aberto a alterações, incentivando os usuários a executarem, estudarem e modificarem os exemplos disponíveis de acordo com as suas aplicações desejadas.

Implementada em C e C++, oferece suporte a programadores para o desenvolvimento de aplicações de RA com baixo custo computacional. Ela utiliza o rastreamento óptico, o qual implementa técnicas de visão computacional para identificar e estimar em tempo real a posição e a orientação de um marcador em relação ao dispositivo de captura de vídeo, sendo possível desenvolver aplicações que o utilizam para posicionar e orientar elementos virtuais na cena do mundo real, tornando-se um meio de interação do usuário com essas aplicações. Para a implementação de uma aplicação simples de RA, o funcionamento da ARToolKit consiste em:

- iniciar a configuração do vídeo, ler o arquivo de cadastramento dos marcadores e ler os parâmetros da câmera;
- capturar um quadro do vídeo;
- detectar e identificar os marcadores;
- calcular a transformação do marcador relativa à câmera;
- desenhar o objeto virtual no marcador.

Essas etapas estão ilustradas no pipeline de funcionamento do ARToolKit na figura 4.1.

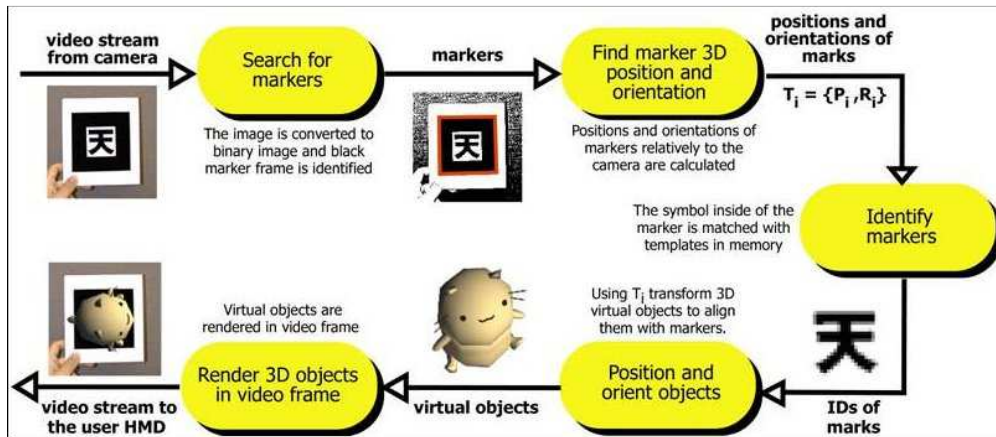


Figura 4.1: Pipeline de funcionamento do ARToolKit (KATO, 2010).

Essa biblioteca possui algumas limitações. Se a câmera estiver muito longe do marcador a biblioteca não conseguirá identificá-lo. Se a câmera estiver muito perto, não conseguirá filmar o marcador por inteiro e a identificação ficará comprometida. Assim, quanto maior o marcador, mais longe a câmera poderá ficar. A tabela 4.1 mostra algumas distâncias máximas em relação ao tamanho do lado do marcador.

| Tamanho do lado do marcador (centímetros) | Distância máxima para identificação (centímetros) |
|---|---|
| 7 | 40 |
| 9 | 63,5 |
| 10,8 | 86,36 |
| 18,7 | 127 |

Tabela 4.1: Distâncias máximas de identificação dos marcadores conforme seu tamanho (KATO, 2010).

Essa distância também é influenciada pela complexidade do marcador. Quanto mais simples o marcador, melhor é para sua identificação. Marcadores com grandes áreas pretas e brancas são mais eficientes. Se mudarmos o marcador de tamanho 10,8 centímetros por um de mesmo tamanho mas com uma complexidade maior, reduziremos a distância máxima de detecção de 86,36 centímetros para 38,1 centímetros (KATO, 2010).

4.1.2 AssaultCube

AssaultCube (Figura 4.2) é um jogo de tiro em primeira pessoa com o código-fonte aberto e baseado na engine e no jogo Cube. O jogo começou em 2004, com o nome de

ActionCube, por alguns membros da comunidade Cube. O lançamento oficial se deu em novembro de 2006, e em maio de 2007 o nome do jogo foi trocado para AssaultCube para evitar confusão com o nome de outro jogo existente, o Action Quake. O jogo foi desenvolvido para ser mais realístico que o jogo Cube original mas mantendo sua simplicidade e sua velocidade de jogo.



Figura 4.2: Cena do jogo AssaultCube.

Existem duas facções no jogo, os Cubers Liberations Army (CLA) e os Rabid Viper Special Forces (RVSF), que o jogador deve escolher quando começa um jogo. O jogador pode tanto jogar online como jogar contra inimigos virtuais (bots). Existem doze modos diferentes de jogos:

- Capture the Flag (Captura da bandeira)
- Keep the Flag (Ache a bandeira e tente mantê-la sobre seu poder o máximo de tempo)
- Team Keep the Flag (Mesmo que Keep the Flag, mas com times)
- Deathmatch (Cada um por si)
- Team Deathmatch (Times contra times)
- One Shot One Kill (Usa-se somente rifles de atiradores de elite)
- Team One Shot One Kill (Um tiro uma morte com times)
- Last Swiss Standing (Usa-se somente facas e granadas)

- Survivor (Luta-se até sobrar somente um jogador)
- Team Survivor (Mesmo que Survivor mas com times)
- Pistol Frenzy (Usa-se somente pistolas, facas e granadas)
- Hunt the Flag (Resgate de reféns)

Dentre esses doze tipos de jogos, no modo contra os bots é possível somente jogar o Deathmatch, Team Deathmatch ou One Shot One Kill.

4.2 Hardware Utilizado

Para o desenvolvimento deste trabalho foram utilizados três monitores LCD, todos do mesmo tamanho e de mesma resolução. No caso foram utilizados três monitores de 19 polegadas com uma resolução de 1280 por 1024 pixels.

Também foi utilizado um computador com duas placas gráficas, sendo uma placa da marca Nvidia do modelo GeForce 9600 e outra também da marca Nvidia, mas o modelo GeForce 8600. Duas placas gráficas idênticas são recomendadas por motivos de compatibilidade, mas isso não é uma real necessidade.

Para a calibração dos monitores, foi utilizada uma webcam de 1,3 megapixels de resolução. Esta resolução é recomendada, pois webcams de menor resolução podem ter dificuldades em detectar os marcadores apresentados.

4.3 Mudanças no Software do Jogo

Algumas mudanças no código-fonte do jogo foram necessárias para que ele pudesse suportar os três monitores e suas calibrações. Isso foi feito em duas etapas. Primeiramente, a janela foi dividida em três viewports idênticas, replicando duas vezes a câmera virtual que representa a visão do jogador. Ao final dessa etapa temos a visão do jogador repetida em cada viewport (Figura 4.3).



Figura 4.3: Divisão da tela em três viewports.

Para termos a visualização esperada de uma CAVE, ou seja, uma imagem contínua, é preciso rotacionar as câmeras das viewports laterais usando o ângulo do campo de visão. Assim, teremos uma visualização ampliada, que fornece uma visão com três vezes o campo de visão original (Figura 4.4). Foi necessário também informar qual viewport está sendo renderizada para que o mini mapa, a vida do jogador e a quantidade de munição fosse renderizada somente na viewport central.

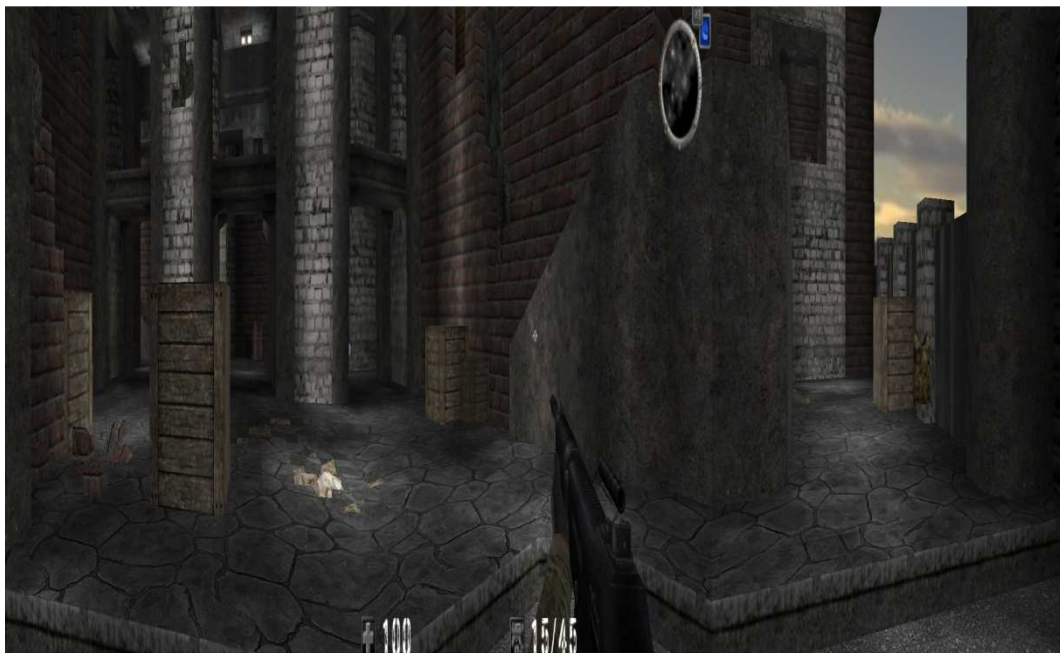


Figura 4.4: Três viewports como uma imagem contínua.

O código-fonte é mostrado na figura 4.5, o qual define cada uma das viewports, rotaciona a câmera e renderiza o frame chamando a função `gl_drawframe` (seta em vermelho).

```
//FIRST VIEWPORT
glViewport(0, 0, screen->w/3, screen->h);
viewportNum = 1;
player1->yaw = player1->yaw-dynfov();
computeraytable(camera1->o.x, camera1->o.y, dynfov());
⇒ gl_drawframe(screen->w, screen->h, fps<lowfps ? fps/lowfps : (fps>highfps ? fps/highfps : 1.0f), fps, viewportNum, anglesViewPort1);
if(frames>4) SDL_GL_SwapBuffers();
player1->yaw = player1->yaw+dynfov();

//THIRD VIEWPORT
glViewport(2*screen->w/3, 0, screen->w/3, screen->h);
viewportNum = 3;
player1->yaw = player1->yaw+dynfov();
computeraytable(camera1->o.x, camera1->o.y, dynfov());
⇒ gl_drawframe(screen->w, screen->h, fps<lowfps ? fps/lowfps : (fps>highfps ? fps/highfps : 1.0f), fps, viewportNum, anglesViewPort3);
if(frames>4) SDL_GL_SwapBuffers();
player1->yaw = player1->yaw-dynfov();

//SECOND VIEWPORT
glViewport(screen->w/3, 0, screen->w/3, screen->h);
viewportNum = 2;
computeraytable(camera1->o.x, camera1->o.y, dynfov());
⇒ gl_drawframe(screen->w, screen->h, fps<lowfps ? fps/lowfps : (fps>highfps ? fps/highfps : 1.0f), fps, viewportNum, anglesViewPort2);
if(frames>4) SDL_GL_SwapBuffers();
```

Figura 4.5: Definição das três viewports.

Deste modo, quando conectamos os três monitores e configuramos a resolução do jogo corretamente, temos, então, o jogo rodando em todos os monitores. Porém, os monitores devem estar posicionados de modo que o ângulo entre eles seja igual ao ângulo do campo de visão.

Para tornar o jogo compatível com calibração dinâmica dos ângulos entre os monitores, que será explicada na próxima seção (4.4 Calibração dos Monitores), foram necessárias alguma mudanças na função *setPerspective*, que se encontra no arquivo *rendergl.cpp*.

Nesse método é definido o view frustum ou volume de visualização (Figura 4.6) que é a visão do personagem do jogo. Somente os objetos dentro do volume de visualização serão renderizados. Neste trabalho teremos três câmeras e cada uma delas tem o seu próprio volume de visualização e um campo de visão. Assim, para termos a maior visualização possível com o campo de visão de cada uma, a soma dos ângulos entre os monitores deve ser igual ao do campo de visão (Figura 4.7). Mas se os ângulos entre os monitores forem diferentes do campo de visão, o observador verá somente um pedaço do que seria possível ver (Figura 4.8).

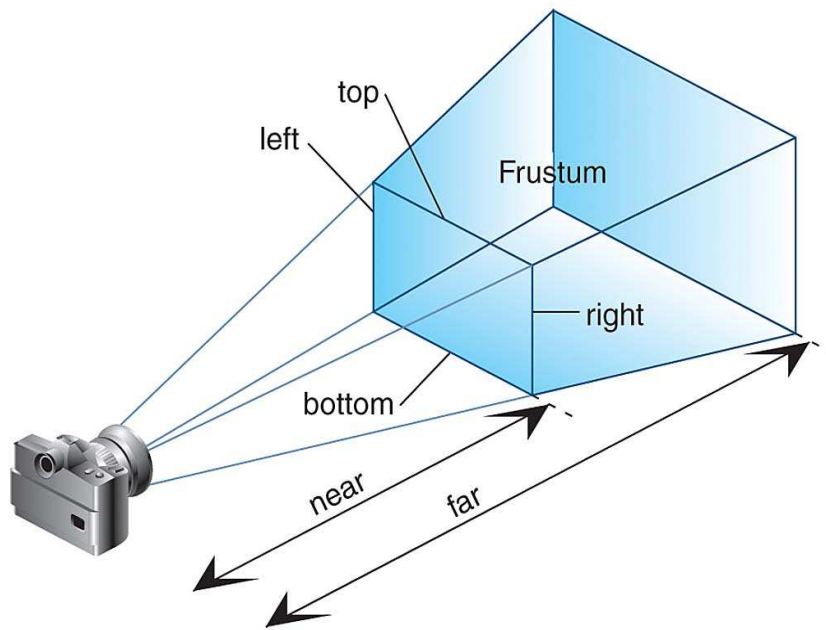


Figura 4.6: Definição do frustum.

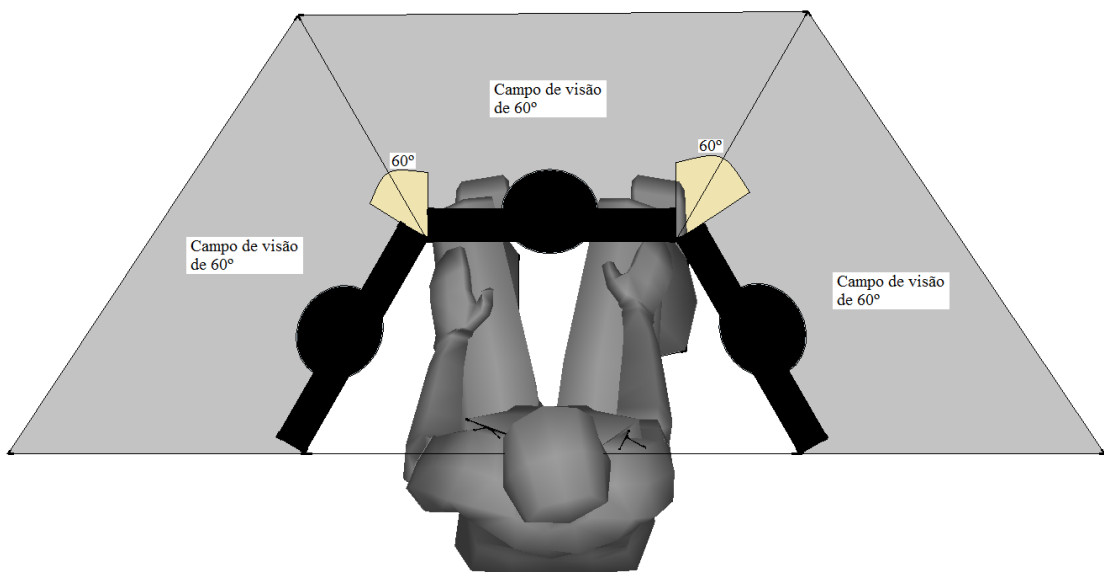


Figura 4.7: Ângulo entre os monitores (60°) igual ao ângulo do campo de visão do observador no jogo (60°).

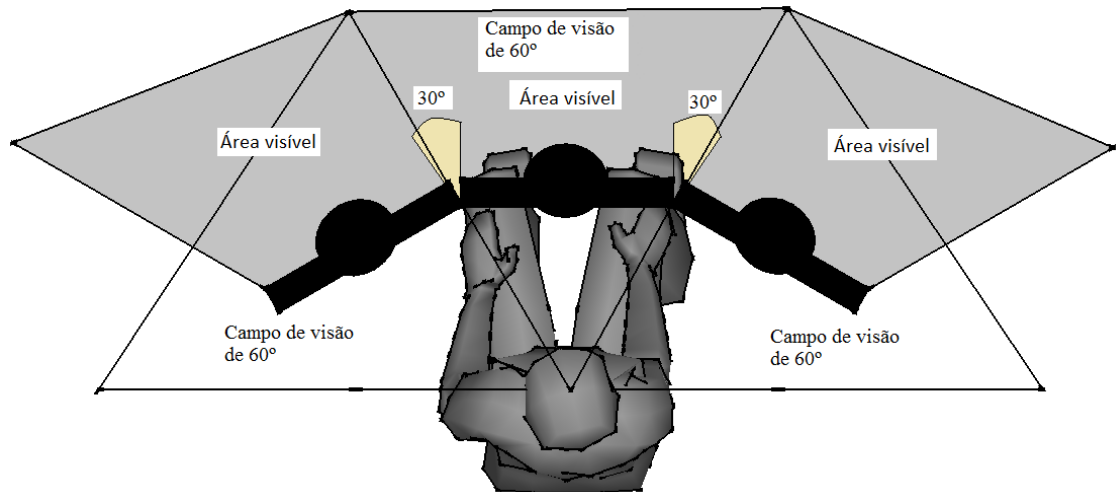


Figura 4.8: Ângulo entre os monitores (30°) diferente do ângulo do campo de visão do observador no jogo (60°).

Os ângulos entre os monitores foram passados para o jogo através de um arquivo texto e guardados em variáveis globais. Assim, no método `setPerspective`, é necessário verificar se os ângulos passados, os ângulos entre os monitores, são os mesmos que o ângulo do campo de visão. Se os ângulos forem iguais, então, chama a função `glFrustum` com os parâmetros inalterados. Entretanto, se os ângulos forem diferentes uma correção será feita para ter a correta visualização do ambiente.

A correção é feita aumentando o valor do parâmetro `left` do frustum na viewport mais da esquerda e diminuindo o valor do parâmetro `right` na viewport mais da direita. Para saber o quanto deve ser aumentando ou diminuído, é calculado o valor correspondente a um grau no frustum. Por exemplo, se os ângulos entre os monitores são de 30 graus, o frustum tem `left` igual a -50 e `right` igual a 50 e o ângulo do campo de visão é de 60 graus, cada grau neste campo de visão corresponderá a 100 dividido por 60 que é 1,667. Com esse valor é possível calcular qual o tamanho do frustum com um ângulo diferente. Multiplicamos, então, o ângulo entre os monitores (30°) com o valor referente a cada grau (1,667), assim teremos 50 que é o novo tamanho do frustum visível pelo monitor. Agora, para descobrir o `left` na viewport da esquerda diminuimos o `right` pelo novo tamanho encontrado, assim o novo `left` será 0. Se esse exemplo representasse a viewport da direita teríamos que descobrir o valor do `right`, que seria o `left` mais o tamanho, assim temos o valor do `right` igual a 0. Como toda tela do monitor é ocupada por essa área menor de visualização, a imagem fica esticada. Assim os objetos virtuais aparecem maiores nessas viewports. Se os ângulos entre os monitores

forem maiores que o do campo de visão, por exemplo 90 graus e 60 graus, respectivamente, acontece o mesmo que aconteceria em um ângulo entre os monitores de 30°.

4.4 Calibração dos Monitores

Para fazer a calibração dos ângulos entre os monitores foi criado um aplicativo utilizando a biblioteca ARToolKit. No aplicativo, o usuário deve utilizar uma webcam para filmar os monitores que apresentarão marcadores na tela (Figura 4.9) e o software identificará estes marcadores. Depois de identificados, o software, utilizando visão computacional, identificará os ângulos entre a webcam e os marcadores e com isso calculará os ângulos entre os monitores.



Figura 4.9: Monitores com marcadores e webcam.

4.4.1 Rastreamento e identificação por marcadores fiduciais

Os marcadores fiduciais são imagens que contêm características visuais fáceis de serem extraídas. Normalmente, são figuras geométricas quadradas, que contêm no seu interior símbolos para identificação (Figura 4.10). É necessário que as características visuais dessas marcas estejam plenamente visíveis para o processo de rastreamento e identificação ser realizado com sucesso. Se o marcador estiver obstruído parcial ou totalmente, o processo de rastreamento e identificação pode ficar comprometido.



Figura 4.10: Exemplo de marcadores do ARToolkit

O rastreamento é responsável pelo processamento da imagem, que extrai algumas informações com relação a detecção, e pela identificação de características dos marcadores, além de estimar sua posição e orientação. O reconhecimento de padrões é realizado pela identificação dos quatro vértices de regiões quadradas, contidas na imagem de vídeo, que é convertida em uma imagem binária (preto e branco). O símbolo no interior dos vértices é comparado aos gabaritos dos marcadores cadastrados pelo usuário (CLAUS, 2005). Caso a informação contida dentro do retângulo extraído seja semelhante com algum marcador cadastrado, o sistema passa a identificar o marcador.

Com o marcador identificado, a biblioteca passa a estimar sua posição e orientação. A obtenção da posição e orientação do marcador é realizada através da análise da imagem de vídeo, que estabelece o relacionamento entre as coordenadas do marcador e as coordenadas da câmera (Figura 4.11).

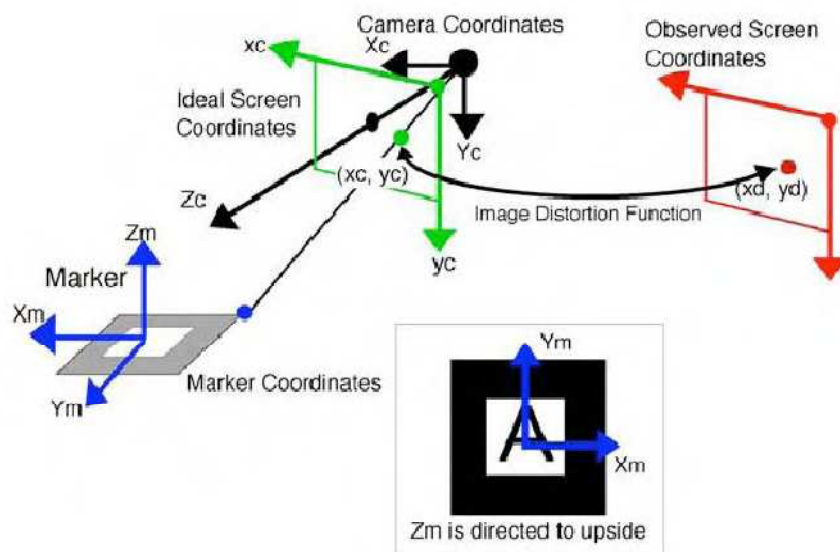


Figura 4.11: Relação do marcador com as coordenadas.

O relacionamento entre as coordenadas do marcador e as coordenadas da câmera é realizado por intermédio de uma matriz 4x4, denominada "matriz transformação" (Figura 4.12). A multiplicação de uma matriz transformação "T" por um ponto 3D no marcador (Xm,Ym,Zm), permite obter o ponto correspondente no sistema de coordenadas da câmera (Xc, Yc, Zc).

$$\begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix}$$

Figura 4.12: Matriz de transformação.

Com essa matriz é possível extrair o ângulo entre o marcador e a câmera. Assim, depois de armazenar os ângulos entre cada um dos três marcadores e a webcam, é possível descobrir os ângulos entre os três monitores.

5 EXPERIMENTOS COM USUÁRIOS

5.1 Objetivos e Hipóteses

Testes foram feitos para verificar se o sistema proposto, uma CAVE simplificada utilizando três monitores, melhora a jogabilidade e imersão dos jogadores. Verificando parâmetros como precisão nos tiros, número de vezes que o jogador morreu, número de inimigos que ele matou, número de tiros que o jogador levou e movimentação horizontal, espera-se que o jogador se sinta mais envolvido pelo jogo e apresente um melhor desempenho.

5.2 Planejamento e Realização dos Testes

Os testes foram realizados durante o evento Portas Abertas da UFRGS, no Instituto de Informática, no Campus do Vale. Pessoas aleatórias, integrantes ou não da comunidade da computação que participavam do evento, foram convidadas a realizar os testes. Esses testes foram repetidos posteriormente fora do evento com pessoas conhecidas.

Foi feito um texto explicativo dos testes (Apêndice A) contendo uma breve explicação sobre o teste. O texto apresenta o jogo que será jogado e indicações de como jogá-lo. Este texto foi lido antes da realização do teste. Um questionário também foi criado para ser respondido depois de o usuário realizar o teste (Apêndice B). Espera-se que esse questionário responda se o usuário teve uma melhora na imersão.

O teste consiste em um jogo de tiro em primeira pessoa em que uma pessoa joga o jogo Assault Cube. No teste, a pessoa tem um minuto para ambientação no jogo e, em seguida, para validação do teste e obtenção dos parâmetros, a pessoa joga dois minutos utilizando o sistema com três monitores. Depois o usuário faz outro teste, com mais dois minutos, mas com somente um monitor. A ordem do teste é aleatória, onde uma pessoa

pode fazer primeiro o teste com um monitor e depois com três monitores ou ao contrário. Foram escolhidos dois mapas do jogo para padronizar os testes, o mapa ac_gothic e ac_complex, e utilizados de maneira aleatória.

5.3 Resultados Obtidos e Discussão

Vinte e cinco pessoas foram submetidas ao teste. Com o questionário foi constatado que a grande maioria das pessoas já possui experiência com jogos virtuais, e que mais da metade possui pelo menos 8 anos de experiência (Figura 5.1). O grupo de pessoas possui uma habilidade variada com jogos tipo FPS: 32% deles tem uma habilidade fraca, 24% possui habilidade normal, 24% possui uma boa habilidade e 20% é muito bom com jogos tipo FPS. Ninguém, neste quesito se acha o melhor em jogos tipo FPS (Figura 5.2).

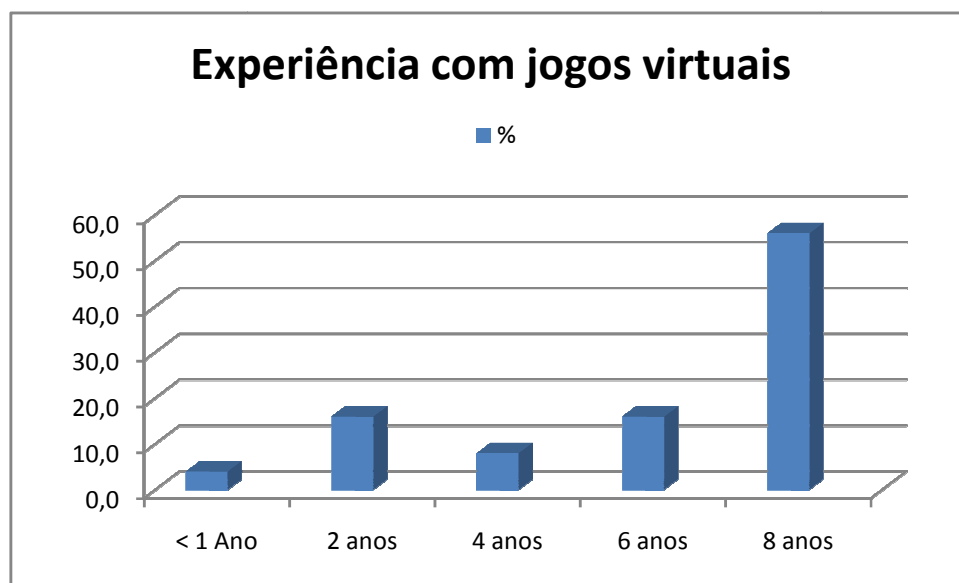


Figura 5.1: Gráfico que representa a experiência de cada pessoa com jogos.

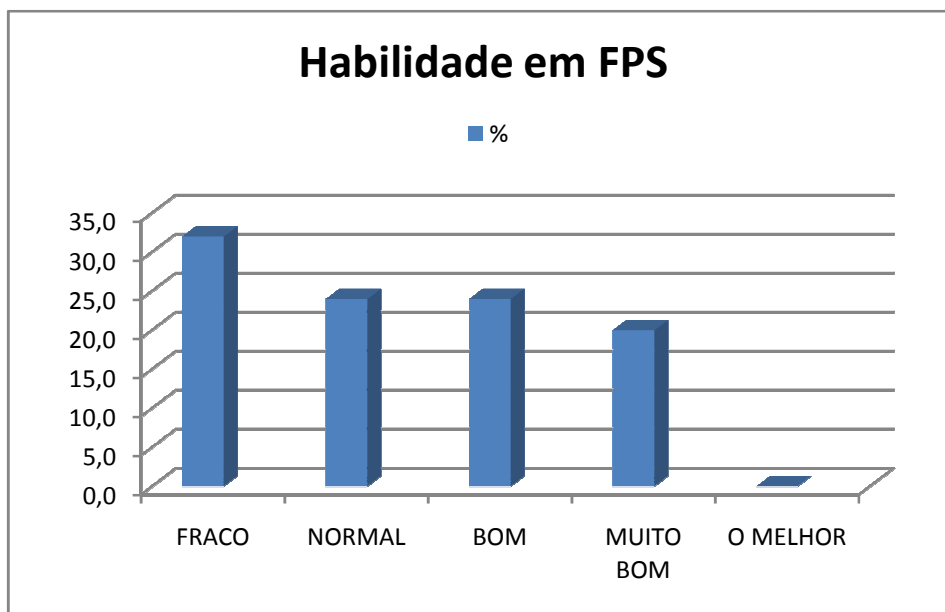


Figura 5.2: Gráfico que representa a habilidade de cada pessoa com jogos tipo FPS.

Mesmo que maioria das pessoas possua alguma experiência com jogos e com o uso do computador, a maioria delas tem pouca experiência com outros aplicativos 3D (Figura 5.3). Esses aplicativos são como as ferramentas de computer-aided design (CAD), para desenhos técnicos, ferramentas para criação de modelos em 3 dimensões como o Blender, 3DS, entre outros. Os que possuem alguma experiência com esse tipo de aplicativo são pessoas ligadas a computação que estavam no evento Portas Abertas e realizaram o teste.

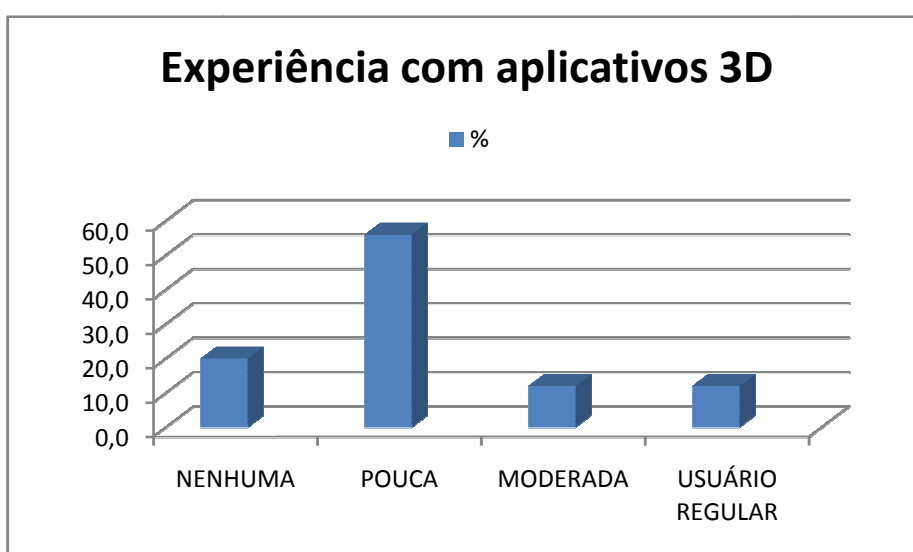


Figura 5.3: Gráfico que representa a experiência das pessoas com aplicativos 3D.

Para verificar se o usuário teve uma melhora no desempenho, foram analisadas cinco variáveis: precisão, número de vezes que o jogador morreu, número de inimigos que ele matou, número de tiros que o jogador levou e movimentação horizontal. Para todas elas se obteve médias melhores quando o jogador tinha a CAVE desktop com 3 monitores do que com apenas 1 monitor. Em seguida, foram feitos *testes-t* para verificar se esses resultados apresentavam significância estatística.

5.3.1 Precisão

Para a precisão com somente um monitor obteve-se a média de 21,75% tiros que acertaram o inimigo, já com o sistema de três monitores a média foi de 24,36%. Mesmo com a média maior com um monitor, o *teste-t* com $\alpha = 5\%$ resultou o valor F de 1,04 o que revela uma confiabilidade de 68,8%. Assim não temos significância estatística nesse parâmetro.

5.3.2 Número de vezes que o jogador morreu

Com esse parâmetro a média de mortes do jogador com um monitor ficou em 4,6 mortes por teste e com três monitores a média ficou em 3,36 mortes por teste. Porém o *teste-t* com $\alpha = 5\%$ resultou o valor F de 3,25 o que revela uma confiabilidade de 92,24%. Assim não temos confiabilidade estatisticamente significativa nesse parâmetro também.

5.3.3 Número de inimigos que o jogador matou

Com esse parâmetro a média de inimigos mortos pelo jogador com um monitor ficou em 3,04 mortes por teste e com três monitores a média ficou em 3,76 mortes por teste. Porém o *teste-t* com $\alpha = 5\%$ resultou o valor F de 1,03 o que revela uma confiabilidade de 68,52%. Assim não temos confiabilidade com significância estatística nesse parâmetro também.

5.3.4 Número de tiros que o jogador levou

Com esse parâmetro a média de tiros que o jogador levou com um monitor ficou em 21,88 tiros levados por teste e com três monitores a média ficou em 17,40 tiros levados por teste. Porém o teste t com $\alpha = 5\%$ resultou o valor F de 1,51 o que revela uma confiabilidade de 77,49%. Mais uma vez o teste não demonstrou significância estatística.

5.3.5 Movimentação horizontal

Com esse parâmetro a média de movimentação horizontal, que representa quanto o jogador virou para os lados com o mouse como se estivesse movendo a cabeça. Esse parâmetro foi capturado somando os graus cada vez que o jogador movia o mouse para qualquer um dos lados. Com um monitor a média ficou em 5676,41 graus por teste de dois minutos e com três monitores a média ficou em 3732,47 graus por teste. O *teste-t* revelou que esse parâmetro tem significância estatística com confiabilidade superior a 99.99% pois o *teste-t* com $\alpha = 5\%$ resultou o valor F de 28,88 o que representa essa confiabilidade. Esse resultado suporta a hipótese de que o sistema com três monitores melhora a visão periférica e conseqüentemente o desempenho do jogador.

Durante os testes, várias pessoas afirmaram que o sistema com três monitores causava um certo desconforto inicial, mas a medida que jogavam isso melhorava. Esse fato ocorreu porque os usuários ainda não estavam acostumados com mais de um monitor. Tanto que depois de realizar o teste muitos deles disseram que, primeiramente, não olhavam para os monitores laterais, mas quando algum outro personagem passava por algum dos monitores laterais eles imediatamente identificavam que havia alguém para aquele lado e a partir desse momento sempre olhavam para os outros monitores tentando identificar a presença de inimigos.

O questionário revelou também que 92% das pessoas que realizaram o teste gostaram mais de jogar com três monitores (Figura 5.4) e esse mesmo valor percentual de pessoas acredita que jogou melhor usando o sistema com três monitores (Figura 5.5).

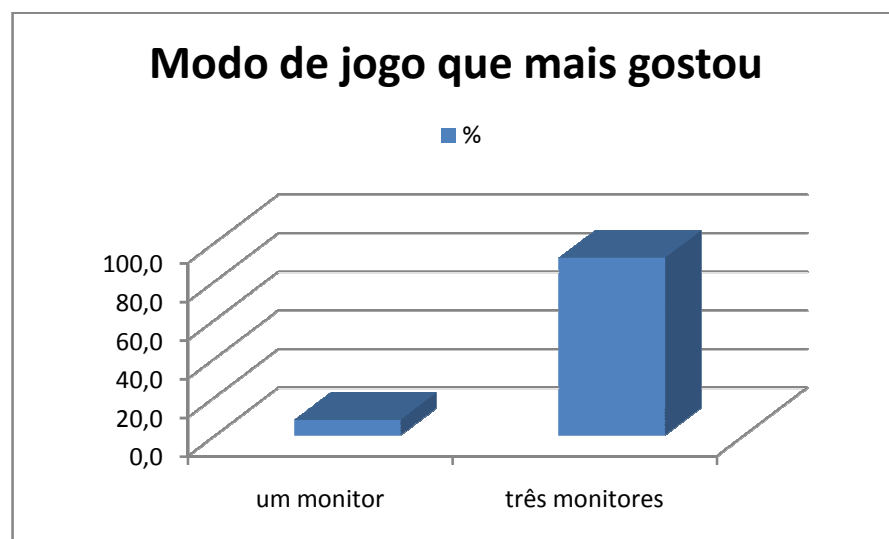


Figura 5.4: Gráfico que representa o modo de jogo que as pessoas mais gostaram.

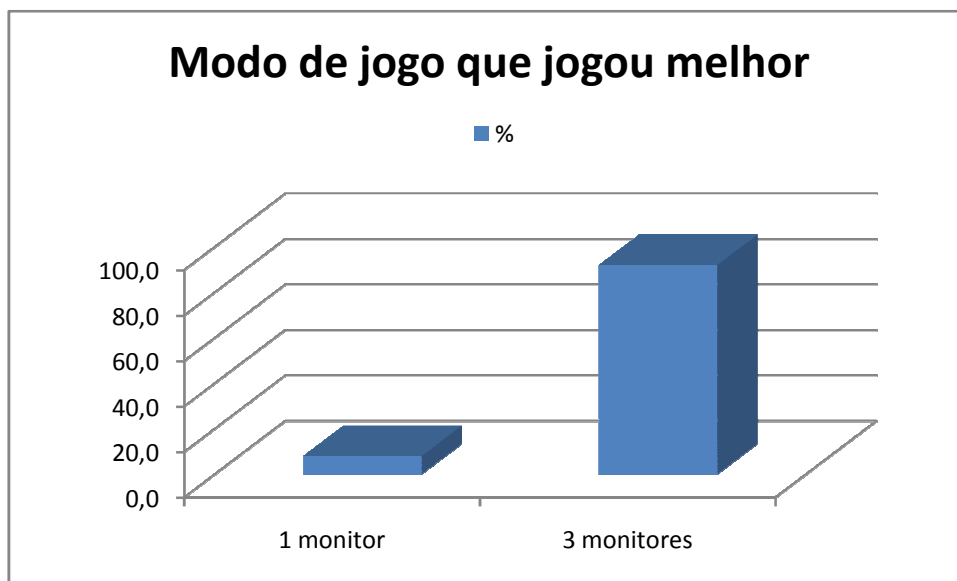


Figura 5.5: Gráfico que representa o modo de jogo que o as pessoas acham que jogaram melhor.

6 CONCLUSÃO

Analisando diferentes aspectos como precisão, quantidade de vezes que o jogador morre ou mata os oponentes, movimentação horizontal e dano tomado, observou-se que o jogador se sentiu mais envolvido pelo jogo e apresentou melhor desempenho. Também foi observado que o sistema proposto aumentou a imersão no ambiente virtual, o que proporcionou uma experiência mais satisfatória ao usuário.

A melhora no desempenho foi comprovada no aspecto de movimentação horizontal, que mostrou que o jogador move menos o mouse para observar o ambiente e, pelo fato de se ter um campo de visão maior, o jogador consegue identificar os inimigos com mais facilidade. Os outros parâmetros medidos, ainda que indicassem melhora no uso de três monitores ao invés de um, não tiveram significância estatística suficiente para provar a hipótese. No entanto, o conjunto desses resultados traz fortes evidências a favor da hipótese de que o sistema com três monitores melhora o desempenho do jogador.

O aumento de imersão foi constatado uma vez que as pessoas sentiram que jogaram melhor com o sistema de três monitores. Os dados mostram que algumas delas tiveram um desempenho pior em quesitos como inimigos mortos e vezes que morreu, mas, mesmo assim, elas ainda sentiram que jogaram melhor com o sistema proposto neste trabalho.

O sistema ainda possui melhorias a serem feitas. Uma delas é a calibração da posição dos monitores em relação ao jogador e a passagem desses dados para o jogo, a fim de posicionar as câmeras virtuais de acordo com a posição dos monitores e do jogador. Neste trabalho, somente os ângulos relativos dos monitores foram passados para o jogo. Outra melhoria seria a calibração durante o jogo, mas para isso seria necessário rastrear jogador e monitores o tempo todo, por exemplo, com o jogador

usando uma webcam sobre sua cabeça em um boné ou chapéu e que marcadores sejam exibidos na tela durante o jogo.

Em relação aos testes com usuários, acredita-se que usando uma amostragem maior de jogadores, ou separando os jogadores voluntários em grupos com perfis similares, os efeitos randômicos dos testes se diluam e se alcance significância estatística em todos os parâmetros. Isso deve ocorrer porque todos os parâmetros analisado estão fortemente relacionados com a aquisição de habilidades para o jogo desenvolvidas pelo jogador.

REFERÊNCIAS

AZUMA, R. et al. **Recent advances in augmented reality**. IEEE Comput.Graph. Appl., Los Alamitos, CA, USA, v.21, n.6, p.34–47, 2001.

BUXTON, William; et. al. **Large Displays in Automotive Design**. IEEE. Julho/Agosto 2000.

CAKMAKCI, Ozan; ROLLAND, Jannick. **Head-Worn Displays: A Review** Journal of Display Technology, Vol. 2, N^o. 3, SEPTEMBER 2006.

CLAUS, D.; FITZGIBBON, A. W. **Reliable automatic calibration of a marker-based position tracking system**. In: wacv-motion '05: proceedings of the seventh IEEE workshops on application of computer vision (wacv/motion'05) - volume 1, 2005.

CARVALHO, Felipe. **Um Sistema Híbrido Semi-Imersivo de Baixo Custo para Interações 2D-3D**. XII Symposium on Virtual and Augmented Reality. Natal, RN, Brasil - Maio 2010.

CAWOOD, Stephen; FIALA, Mark. **Augmented Reality: A Practical Guide**. Pragmatic Bookshelf, 2008.

CRUZ-NEIRA, Carolina; et. al. **The Cave – Audio Visual Experience Automatic Virtual Environment**. Siggraph Junho 1992.

FOLEY, James D.; et. al. **Computer Graphics: Principles and Practice**. Segunda Edição. Addison-Wesley, 1995.

GRANDI, Jerônimo G. **Visualização Espaço-consciente Colaborativa de Imagens Médicas** Caxias do Sul, 2009.

HILL, Francis S. **Computer Graphics**. Segunda Edição. Prentice Hall, 2001.

INSLEY, S. **Obstacles to general purpose augmented reality**. Disponível em: <http://www.yellowdocuments.com/14014933-obstacles-to-general-purpose-augmented> [s.n.], 2003. [Online; acessado em: maio-2010].

KATO, H.; et. al. **Virtual Object Manipulation on a Table-Top AR Environment**. In proceedings of the International Symposium on Augmented Reality, pp.111-119. Munich, Germany. (ISAR 2000).

KATO, H. **ARToolKit Documentation**. University of Washington. Washington, DC. Disponível em: < <http://www.hitl.washington.edu/artoolkit/documentation/>> Acessado em: abril 2010.

KIRNER, C.; KIRNER, T.G., **Virtual Reality and Augmented Reality Applied to Simulation Visualization**. In: El Sheikh, A.A.R.; Al Ajeeli, A.; Abu-Taieh, E.M.O.. (Org.). *Simulation and Modeling: Current Technologies and Applications*. 1 ed. Hershey-NY: IGI Publishing, 2007, v. 1, p. 391-419.

LI, Zhiyun; VARSHNEY, Amitabh. **A Real-time Seamless Tiled Display System for 3D Graphics**. *Department of Computer Science and UMIACS-University of Maryland at College Park*.

MAIER, Patrick; et. al. **What Do You Do When Two Hands Are Not Enough? Interactive Selection of Bonds Between Pairs of Tangible Molecules**. IEEE Symposium on 3D User Interfaces 2010 20-21 Março, Waltham, Massachusetts, USA.

MILGRAM, P. et. al. “**Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum**”. *Telemanipulator and Telepresence Technologies*, SPIE, V.2351. 1994.

MCCONNELL, Jeffrey J., **Computer Graphics Theory into Practice**. Surbury, MA, USA. Volume 2005.

REINALDO, Gelson C.; et. al. **A Dynamic Multi-display System Approach**. X Symposium of Virtual and Augmented Reality. Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS) Porto Alegre - RS – Brazil. 2008.

RIBOLDI, Gabriel. **Sistema Interativo em Realidade Aumentada: Permeando o Passado Arquitetônico na Praça Dante Alighieri de Caxias do Sul**. Centro de Computação e Tecnologia da Informação Bacharelado em Tecnologias Digitais. Caxias do Sul, 2009.

APÊNDICE A TEXTO PARA O TESTE

CAVE desktop para jogos 3D em primeira pessoa

Obrigado por participar. Este teste tem por objetivo medir a sua habilidade dentro de um jogo tipo FPS (First Person Shooter) ou tiro em primeira pessoa. Sua habilidade será medida contando o número de inimigos que você matar, o número de mortes suas e a sua precisão (tiros que acertaram/total de tiros). Seu objetivo será matar o máximo de inimigos possível em 2 minutos de teste. O teste será repetido duas vezes, uma usando um monitor e outra usando três monitores (ou vice-versa).

O jogo utilizado é o Assault Cube, jogo open source que utiliza a cube engine e foi lançado em 2007. Para jogar use as teclas W, A, S, D para a movimentação, o mouse para olhar para os lados, para cima e para baixo, e o botão esquerdo do mouse para atirar. Caso você morra clique novamente o botão esquerdo do mouse para voltar ao jogo. Os inimigos quem morrerem também voltam a aparecer em uma nova posição no mapa do jogo.

APÊNDICE B QUESTIONÁRIO TESTE

1. Nome (opcional): _____
2. Idade: _____
3. Profissão (se estudante, qual o curso e ano) _____
4. Joga atualmente algum jogo? () sim () não
Se jogava no passado responda quanto tempo faz que não joga.
() menos de 1 ano () 1 ano () 2 anos () 3 anos () mais de 4 anos
5. Há quantos anos joga? Ou jogou por quantos anos?
() menos de 1 ano () 2 anos () 4 anos () 6 anos () mais de 8 anos
6. Qual o tipo de jogo que mais joga?
() FPS (tiro)
() RPG (interpretação de personagem)
() RTS (estratégia)
() outro _____
7. Em se tratando de FPS (tiro), como avalia sua habilidade?
() fraco () normal () bom () muito bom () o melhor
8. Em média, quantas horas por semana você joga?
() menos de 2 horas () de 2 a 7 horas () de 7 a 20 horas () mais de 20 horas
9. Como avalia sua experiência com aplicativos gráficos em 3D? Ex: CAD, 3DS, Blender. () nenhuma () pouca () moderada () usuário regular
10. Já tinha experimentado aplicações de realidade virtual ou realidade aumentada?

() sim () não Se sim, descreva em uma frase _____

11. No teste de hoje: qual modo de jogo você gostou mais?

() com 1 monitor () com 3 monitores

12. No teste de hoje: em qual modo de jogo você acha que jogou melhor?

() com 1 monitor () com 3 monitores