

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

DESENVOLVIMENTO DE UM PROTÓTIPO DE HARDWARE
E SOFTWARE PARA IMPLEMENTAÇÃO DE UM JOGO
COM PISTOLA

RULYE NAZÁRIO DE OLIVEIRA

BLUMENAU
2003

2003/2-36

RULYE NAZÁRIO DE OLIVEIRA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE HARDWARE
E SOFTWARE PARA IMPLEMENTAÇÃO DE UM JOGO
COM PISTOLA**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciência
da Computação — Bacharelado.

Prof. Paulo César Rodacki Gomes - Orientador

**BLUMENAU
2003**

2003/2-36

**DESENVOLVIMENTO DE UM PROTÓTIPO DE HARDWARE
E SOFTWARE PARA IMPLEMENTAÇÃO DE UM JOGO
COM PISTOLA**

Por

RULYE NAZÁRIO DE OLIVEIRA

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Paulo César Rodacki Gomes, Dr. – Orientador, FURB

Membro: _____
Prof. Miguel Alexandre Wisintainer, FURB

Membro: _____
Prof. Jomi Fred Hübner, FURB

Blumenau, 05 de dezembro de 2003

Dedico este trabalho a minha grande família e a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste.

Duas coisas te pedi. Não me negues antes de eu morrer. Não me dêes nem pobreza nem riquezas, para que eu não me farte e realmente te renegue, e diga: “Quem é Jeová?” e para que eu não fique pobre e realmente furte, e ataque o nome de meu Deus.

Provérbios 30: 7-9

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça.

À minha família, por tudo.

Ao meu colega Luis Augusto Spranger, pelas “consultorias” em eletrônica.

Aos dedicados monitores do curso Ciências da Computação, especialmente Leo Jonathan Faht, pelo apoio.

Ao meu orientador, Paulo César Rodacki Gomes, por ter acreditado na conclusão deste trabalho.

RESUMO

Este trabalho aborda jogos eletrônicos com pistola. Enfoca a relação entre o *hardware* e o *software* e propõe três métodos para detectar as coordenadas de alvos mirados pela pistola no monitor de vídeo. É apresentada também a especificação e a construção de um protótipo de pistola e a implementação de um protótipo de software para demonstrar o funcionamento do *hardware* desenvolvido e dos algoritmos propostos.

Palavras chaves: Jogos; Pistola; Multimídia.

ABSTRACT

This work approaches electronic games with light guns. It focuses on the relation between hardware and software and proposes three methods for the detection of target's coordinates in the video monitor. Also, the specification and construction of a prototype of light gun is presented along with the implementation of a software prototype which demonstrates the hardware as well as the proposed methods for target detection.

Key-Words: Games; Light Gun; Multimedia.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Funcionamento do jogo com pistola.....	17
FIGURA 2 – Exemplo de uma busca com algoritmo linear.....	18
FIGURA 3 – Exemplo de uma busca com algoritmo binário	19
FIGURA 5 - Circuito eletrônico da pistola	24
FIGURA 6 – Esquema de funcionamento da porta paralela	25
FIGURA 7 – Diagrama de classes do jogo	26
FIGURA 8 – Diagrama de seqüências do jogo	28
QUADRO 1 – Função de detecção com algoritmo de busca linear	29
QUADRO 2 – Função de detecção com algoritmo de busca binária	30
QUADRO 3 – Função de detecção com algoritmo de busca quaternária	30
QUADRO 4 – Função principal do jogo.....	31
FIGURA 9 – Diagrama de casos de uso do jogo.....	32
FIGURA 10 – Tela do protótipo desenvolvido	33
FIGURA 11 – Foto da pistola construída.....	34
FIGURA 12 – Número de passos necessários para os algoritmos detectarem cem quadros ...	35

LISTA DE TABELAS

Tabela 1 – Estatísticas da busca quaternária	36
--	----

LISTA DE SIGLAS

Hz – Hertz

h - hexadecimal

LISTA DE SÍMBOLOS

Ω - ohms

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO	12
1.2 ESTRUTURA DO TRABALHO	13
2 FUNCIONAMENTO DOS JOGOS COM PISTOLA.....	14
3 ALGORITMOS DE “DETECÇÃO DE TIRO”	17
3.1 BUSCA LINEAR	18
3.2 BUSCA BINÁRIA	19
3.3 BUSCA QUATERNÁRIA	20
3.4 BUSCA POR REFRESH DE VÍDEO.....	21
4 DESENVOLVIMENTO DA PISTOLA.....	22
4.1 REQUISITOS PRINCIPAIS DA PISTOLA.....	22
4.2 CIRCUITO ELÉTRICO DA PISTOLA.....	23
4.3 PORTA PARALELA	25
5 DESENVOLVIMENTO DO PROTÓTIPO.....	26
5.1 REQUISITOS PRINCIPAIS DO PROTÓTIPO	26
5.2 ESPECIFICAÇÃO	26
5.3 IMPLEMENTAÇÃO	28
5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	31
5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	32
5.4 RESULTADOS E DISCUSSÃO	34
6 CONCLUSÕES.....	37
6.1 EXTENSÕES	38
REFERÊNCIAS BIBLIOGRÁFICAS	39

1 INTRODUÇÃO

Desde que os computadores surgiram, há uma crescente procura por jogos eletrônicos. Eles estão cada vez mais complexos e interativos.

Os primeiros jogos apresentavam uma resolução gráfica não muito agradável com poucas cores e recursos limitados. Com a inovação dos computadores com suas placas de vídeo e recursos cada vez mais poderosos, os jogos também se inovaram. Há jogos com milhões de cores, imagens em três dimensões e com alta qualidade de vídeo e som que atraem a atenção dos jogadores.

Atualmente, os jogos estão cada vez mais interativos de tal forma que os jogadores não apenas vêem o monitor de vídeo, mas entram no jogo utilizando outros sentidos humanos além da visão e da audição. Há jogos com aparelhos que aguçam os sentidos humanos simulando a realidade. Por exemplo, os jogos com pistolas. Eles permitem que o usuário mire o alvo simulando o uso de uma pistola real.

Os jogos com pistola são um exemplo de multimídia. Segundo Vaughan (1994), multimídia é qualquer combinação de texto, arte gráfica, som, animação e vídeo transmitido pelo computador. Além disso, segundo Gibbis (1995, p. 5) a pistola pode ser classificada como um dispositivo de interação, pois abre as possibilidades de interagir com o sistema e fornece controle direto sobre a aplicação como outros dispositivos semelhantes, por exemplo *mouse* e *joystick*.

De acordo com 3Dzonemaster (2003), a maioria dos jogos opera em duas dimensões. O *mouse* permite mirar de cima e para baixo, de um lado para o outro. Da mesma forma o *joystick* também permite orientação para cima, para baixo, para esquerda e direita. Em jogos em três dimensões faz-se o uso de mais de um dispositivo. Com o teclado se faz a orientação e o controle da direção para caminhar no espaço. E com o *mouse* são feitos a mira e o tiro.

Alguns jogos usam uma pistola como dispositivo de interação para mirar e atirar. O usuário mira a arma na tela de vídeo e atira. Se o usuário acertou, o alvo explode. Segundo Howstuffworks (2003) para criar este efeito, a pistola interage com o computador para detectar a coordenada de vídeo de onde o usuário está mirando. Este dispositivo pode ser relacionado com o uso do *mouse*, onde o usuário aponta a seta no lugar desejado e pressiona o

botão do *mouse*. A pistola poderia ser substituída pelo uso do *mouse* para o usuário mirar os alvos e atirar, porém a pistola aguça a coordenação motora do usuário ao usar os braços para mirar e atirar.

A interação entre pistola e computador não pode não deve interferir na qualidade de interação do jogo com o usuário. Os atos de atirar e detectar o alvo devem ser feitos em tempo real, sem latência. Os gráficos dos jogos devem ser desenhados em alta velocidade, de modo a não se perceber o tempo que o computador leva para calcular as reações de todo jogo às ações do usuário e o tempo de cálculo e desenho dos gráficos em vídeo. Da mesma maneira, a interação entre pistola e jogo deve ser rápida suficiente para contribuir com o andamento satisfatório do jogo.

A idéia de desenvolver um jogo com pistola surgiu na disciplina de Multimídia do curso de Ciências da Computação da FURB. Como trabalho final da disciplina, foi solicitado dentre outras coisas o estudo de um dispositivo de multimídia, no caso a pistola eletrônica. Este trabalho motivou a implementação de *hardware* e *software* de um jogo com pistola para o trabalho final de curso.

O trabalho aborda o desenvolvimento do *hardware* e do *software* como proposta para utilização em jogos com pistola. Aborda também a comunicação do *hardware* usando a porta paralela do computador. Há a implementação de três algoritmos de detecção do tiro. E o software desenvolvido demonstra o funcionamento do jogo e a relação entre pistola e computador.

Para a construção da pistola, são utilizados componentes eletrônicos de fácil acesso nas lojas da região. Pode servir de referência para aqueles que se interessam em eletrônica construam a sua própria pistola. Além disso, o trabalho envolve noções de ferramentas para computação gráfica para o desenvolvimento do jogo.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um jogo com pistola.

Os objetivos específicos do trabalho são:

- a) construir um protótipo de uma pistola e especificar seu funcionamento;

- b) descrever como é detectada a coordenada no vídeo propondo algoritmos otimizados;
- c) construir um jogo básico que demonstre a interação com a pistola.

1.2 ESTRUTURA DO TRABALHO

O capítulo dois apresenta a fundamentação teórica e descreve como é detectada a coordenada no vídeo e a relação entre a pistola e o jogo. O capítulo três apresenta propostas de alguns algoritmos de detecção, suas vantagens e desvantagens e quando podem ser utilizados. O capítulo quatro mostra o funcionamento da pistola com uma breve descrição de cada componente utilizado. Em seguida, mostra como o jogo foi desenvolvido, seus requisitos e estrutura do *software* implementado. Por fim são apresentadas as conclusões e propostas para trabalhos futuros.

2 FUNCIONAMENTO DOS JOGOS COM PISTOLA

Os jogos são amplamente divulgados e distribuídos. Desde os jogos antigos até os dias de hoje, muitas melhorias ocorreram. Novos estilos de desenho e novos algoritmos para manipulação de personagens e efeitos especiais foram implementados para atrair os consumidores de jogos. Cada estilo de jogo quer atingir uma determinada massa consumidora, daí a escolha de som, cores e histórias apropriadas, bem como os tipos como jogos de estratégia, ação ou aventura. Há ainda os jogos que procuram melhorar a interação entre usuário e máquina.

Os jogos oferecem várias alternativas de interação. Atualmente, pode-se encontrar jogos eletrônicos que envolvem ação do jogador, como as pistas de danças eletrônicas. Segundo Vicaria (2002) as pistas de danças são um dos atrativos mais procurados nos fliperamas, cibercafés e danceterias. São máquinas que mesclam a tecnologia dos videogames com o agito das danceterias. O desafiante sobe num tablado equipado com sensores e recebe instruções sobre onde pisar. As ordens seguem o ritmo de uma música. O resultado é uma coreografia. Estas pistas são uma alternativa para os jogadores, procurar um jogo com interatividade relativamente mais ativa.

O presente trabalho aborda outra alternativa de interatividade nos jogos, como é o exemplo de jogos com pistola, onde o usuário usa sua coordenação motora para acertar os alvos. Embora a pistola seja um dispositivo popular para jogos eletrônicos, o seu funcionamento não é muito divulgado. Não foram encontrados trabalhos correlatos em buscas feitas na Internet e em livros sobre jogos ou dispositivos multimídia. Apenas são encontradas breves citações sobre o funcionamento da pistola. Estas citações foram suficientes para propor alguns algoritmos de detecção e a construção de um protótipo de pistola para este trabalho.

Segundo Mendes (1997), a pistola basicamente é constituída de um sensor óptico e outros dispositivos que interpretam os sinais deste sensor e transmitem ao jogo. A pistola funciona como um detector de luminosidade, pois há um sensor que detecta a quantidade de luz presente no ambiente. Dado o formato da pistola, com um tubo escuro, o sensor só recebe luz proveniente de um ponto apenas, ou seja, o alvo no monitor de vídeo. A pistola então capta o sinal do sensor e envia para o jogo que irá interpretá-lo.

Ao observar um jogo com pistola, através de experiência pessoal ao pesquisar empiricamente o funcionamento da pistola gravando em vídeo e assistindo em câmera lenta, pode-se notar que no instante que o usuário atira utilizando a pistola, o monitor de vídeo faz um sinal quase imperceptível. O monitor pisca, mas é tão rápido que é imperceptível. Alguns jogadores podem até pensar que a piscada é algum tipo de efeito especial para dar a sensação de que um tiro foi dado. Mas, na verdade, quando o monitor pisca, o jogo está detectando onde o usuário está mirando.

De acordo com Mendes (1997), um ponto luminoso varre seguidamente a tela. No exato instante que o usuário atira o monitor se apaga completamente; a tela fica completamente escura. Logo em seguida, aparece um pequeno quadrado no canto do vídeo que percorre, com o tempo, toda a superfície do vídeo. Mas todo esse processo é tão rápido que quase não é percebido. O quadrado percorre toda a tela até chegar no exato lugar onde a pistola está mirando. De acordo com Mendes (1997), o ponto luminoso é detectado por uma fotocélula existente na pistola e comparado pelo computador com um padrão de coordenadas. A pistola envia o sinal que detectou um ponto luminoso o computador verifica a coordenada do ponto luminoso recém iluminado. Assim que o monitor detecta a coordenada, a tela do jogo volta ao normal e o jogo mostra se acertou o alvo ou não.

Howstuffworks (2003) mostra outra proposta de detecção de tiro. Basicamente, o computador apaga a tela e pinta então a tela inteiramente de branco. Faz-se o exame do tempo para o monitor de vídeo desenhar a imagem branca. Comparando o sinal que vem do sensor de luz com os sinais do redesenho do monitor de vídeo, o computador pode detectar onde o feixe de luz está na tela quando o sensor detecta primeiramente sua luz.

O jogo também deve detectar se o usuário está mirando para alguma área fora do monitor de vídeo. Por exemplo, no instante do disparo, o usuário poderia estar apontando a pistola para cima, para uma lâmpada ou para um lugar escuro. Se o usuário mirar em uma lâmpada qualquer, o jogo falharia indicando que o usuário estaria mirando em um ponto luminoso no vídeo. O jogo também deve tratar o caso de usuário mirar em algo escuro fora da área de vídeo, informando o caso ao usuário.

O jogo verifica se o jogador está mirando para uma lâmpada no primeiro estágio da detecção, isto é, quando a tela se apaga completamente e fica escura. Se o usuário está mirando para uma lâmpada, o computador recebe o sinal de que está mirando para um alvo

luminoso, alvo este que não está desenhado em qualquer parte do monitor. Não é necessário então continuar o teste de detecção. É certo que o usuário está mirando para um alvo fora do monitor.

Finalmente, deve ser detectado se o usuário está mirando para um lugar escuro fora do monitor. O jogo detectará que está fora do monitor, pois o quadrado percorrerá toda a área do vídeo e em momento algum o sensor irá detectar sua luminosidade. É desta maneira que o jogo detecta um ponto no monitor de vídeo utilizando sensores ópticos.

3 ALGORITMOS DE “DETECÇÃO DE TIRO”

Este capítulo descreve os métodos propostos neste trabalho para realizar a leitura de coordenadas de tiro da pistola. Ao final do capítulo é apresentado ainda um algoritmo de busca por *refresh* de vídeo, comumente utilizado em máquinas de jogos comerciais (do tipo console de fliperama).

O computador comunica-se com a pistola para detectar onde o usuário está mirando. Resumidamente, o computador controla as imagens no monitor iluminando regiões do vídeo. A pistola detecta a luminosidade do monitor e envia para o computador se está mirando em algo claro ou escuro.

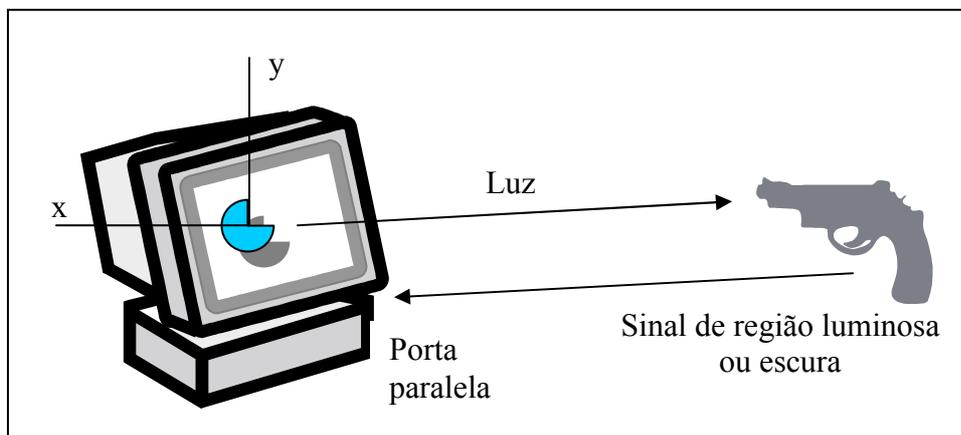


FIGURA 1 – Funcionamento do jogo com pistola

A lógica mais simples para detectar o tiro seria de iluminar apenas os alvos. Quando o usuário atirar, inicialmente a tela apaga-se e em seguida apenas o alvo existente se ilumina. Se a pistola devolver o sinal ao computador de que está mirando em uma região clara, conclui-se que acertou o alvo. No entanto, não será possível dizer com exatidão onde o usuário acertou, apenas será possível dizer que acertou. E muito menos poderá ser indicado onde o usuário atirou quando ele errou o alvo. Por isso, há a necessidade de utilizar algoritmos de busca que possam detectar qualquer ponto da área do jogo.

A seguir são propostos algoritmos com esta finalidade, o algoritmo linear, binário e quaternário. Todos os três algoritmos têm a seguinte característica em comum, o computador desenha quadros luminosos na tela e então compara o sinal proveniente da pistola verificando se a mira está em quadro luminoso ou escuro. No entanto, diferem no modo que são desenhados os quadros luminosos na tela. Finalmente é apresentado mais um algoritmo, busca

por *refresh*, com a mesma finalidade, mas que se caracteriza por não desenhar quadros no monitor de vídeo.

3.1 BUSCA LINEAR

A busca linear é um algoritmo relativamente simples. Uma área luminosa percorre seqüencialmente a tela até encontrar o alvo. A área luminosa pode iniciar a varredura da tela a partir do canto superior esquerdo, por exemplo, e percorrer toda a área do vídeo até chegar ao canto inferior direito.

O tamanho da área luminosa utilizada deve estar de acordo com a precisão permitida pelo *hardware* da pistola. Com componentes mais precisos e sensíveis para o hardware, pode-se utilizar áreas menores. A figura abaixo mostra um exemplo passo a passo do algoritmo de busca linear.

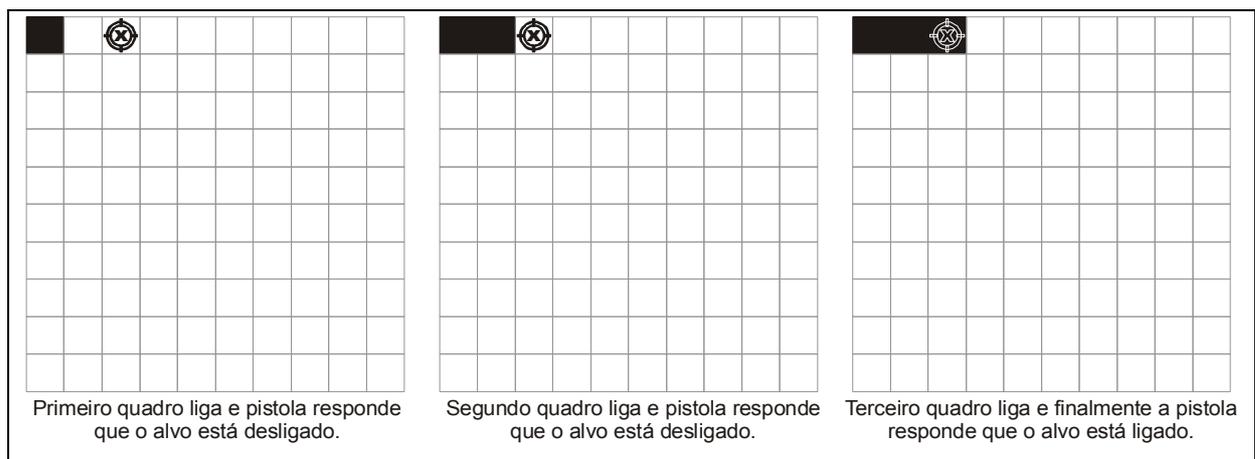


FIGURA 2 – Exemplo de uma busca com algoritmo linear

Por ser um algoritmo simples, possui desvantagens. Quanto mais próximo do ponto inferior direito do monitor de vídeo, maior será o tempo para detectar o alvo. Isso pode até tornar a busca inviável e comprometer o funcionamento do jogo, sendo que o usuário perde a sensação de um jogo próximo a realidade e começa a perceber o tempo de detecção de tiro.

Uma tentativa de amenizar o problema seria utilizar quadros maiores diminuindo consideravelmente a quantidade de quadros para percorrer. Porém a precisão da localização das coordenadas do tiro seria afetada, pois o jogo somente poderia detectar alvos grandes.

3.2 BUSCA BINÁRIA

Este algoritmo é inspirado nos algoritmos de busca binária e possui uma lógica mais elaborada para detectar o alvo. A área de varredura para detecção de tiro é inicialmente dividida pela metade de sua largura. Cada uma das metades é desenhada com uma cor clara (branco). A metade na qual a pistola está apontada é dividida recursivamente até chegar-se a um nível de recursão (ou largura da área iluminada) de acordo com uma precisão pré-estabelecida. Após isso, o mesmo processo é empregado subdividindo-se recursivamente a faixa vertical na metade de sua altura, conforme demonstrado na figura abaixo:

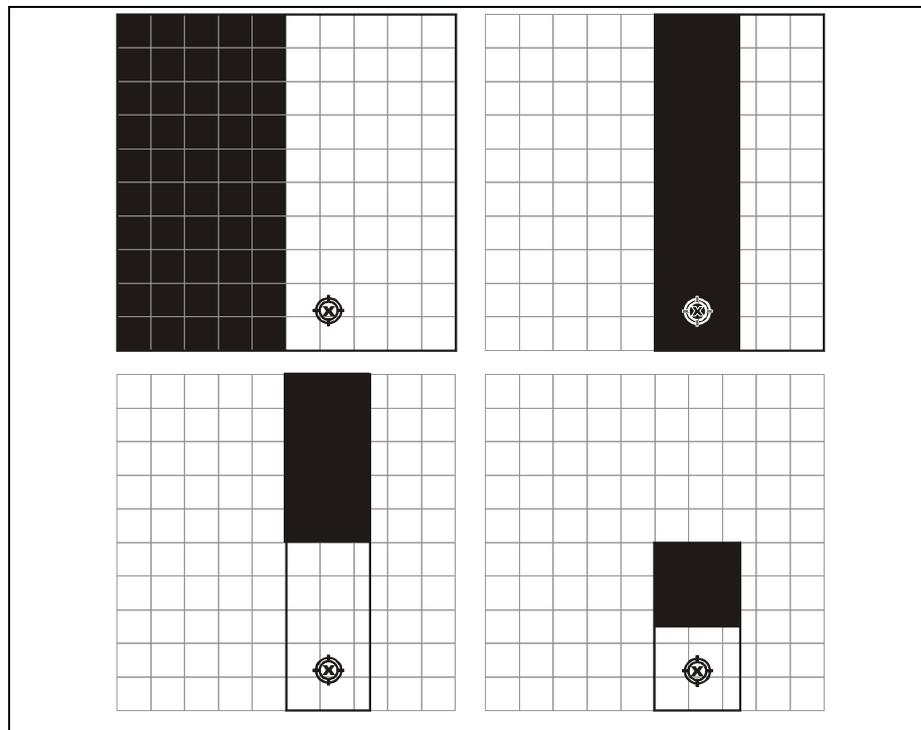


FIGURA 3 – Exemplo de uma busca com algoritmo binário

Por exemplo, considerando uma imagem com 100 quadros, ao dividir a imagem ao meio, metade clara e escura, é possível descartar já de início 50 quadros ao saber em qual metade o usuário está mirando. Estes 50 quadros seriam percorridos um a um na busca simples. Em seguida, a área onde o usuário está mirando é novamente dividida ao meio, podendo-se descartar mais 25 quadros. E o processo é repetido sucessivamente até a imagem ficar dividida no tamanho de um quadro.

Uma característica interessante deste algoritmo é que o de detecção de tiro, na maioria dos casos, é mais rápido que o tempo necessário para percorrer a tela seqüencialmente como

na busca linear. Somente para detectar os quadros mais próximos ao canto superior esquerdo que o algoritmo linear detectaria mais rapidamente.

3.3 BUSCA QUATERNÁRIA

A busca quaternária é baseada nos algoritmos *Quadtree*. A idéia fundamental do funcionamento *Quadtree* é a subdivisão sucessiva de um plano 2D para representação de um gráfico. Quando um *Quadtree* é usado para representar uma área em um plano, cada quadrante pode ser preenchido, parcialmente, completo ou vazio, dependendo da quantidade que o quadrante pertence à área da imagem que esta representa. Um quadrante preenchido é recursivamente subdividido em subquadrantes. Subdivisões continuam até todos os quadrantes ficarem homogêneos ou quando um quadrante está representando satisfatoriamente uma área. Maiores informações sobre o uso do algoritmo *Quadtree* podem ser encontradas em Foley (1990, p. 550).

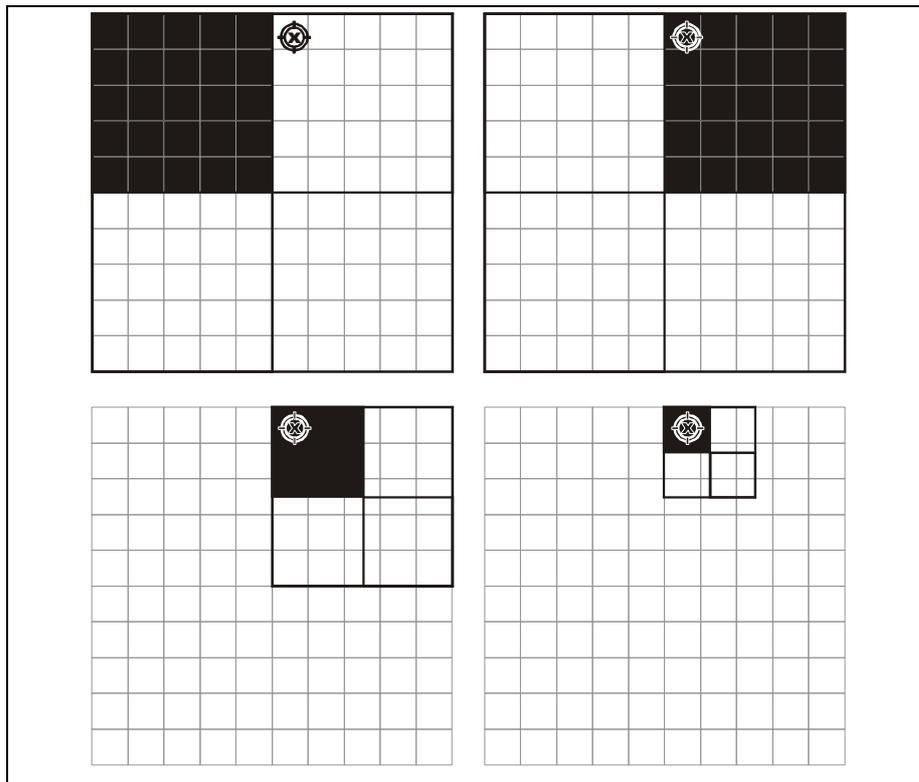


FIGURA 4 – Exemplo de uma busca com algoritmo quaternário

Como se pode observar na figura 4, no uso do algoritmo para detectar o tiro, a tela é dividida em quatro quadrantes e é detectado em qual quadrante o usuário está atirando. O

quadro detectado como alvo é recursivamente dividido em quatro. O processo é repetido até obter um quadro de tamanho mínimo pré-estabelecido que possa ser utilizado como referência para indicar a coordenada que o usuário atirou.

Exemplificando em um monitor com 100 quadros, ao dividir em quatro partes iguais, pode-se descartar 75 quadros ao detectar o quadro alvo. O algoritmo de busca binária precisaria dividir a tela duas vezes para descartar 75 quadros. Mas é preciso percorrer os quatro quadros seqüencialmente para detectar o quadro alvo. O que é uma desvantagem.

Este algoritmo pode ser mais bem aplicado quando se sabe que há uma maior probabilidade de o usuário mirar em determinado canto do monitor de vídeo. Assim, a relativa demora em detectar outros cantos, onde o usuário raramente atirará, pode ser compensada com a velocidade para detectar onde o usuário provavelmente atirará maior número de vezes.

3.4 BUSCA POR REFRESH DE VÍDEO

O monitor de vídeo possui uma frequência de amostragem. A cada segundo, as imagens são atualizadas dezenas de vezes. A busca por *refresh* de vídeo propõe utilizar as próprias atualizações das imagens no monitor para detectar o alvo. Segundo Howstuffworks (2003), ao mudar a imagem, o jogo comparava o comportamento do sinal da pistola com um padrão pré-definido de coordenadas. Comparando o sinal que vem do sensor de luz com os sinais do redesenho do monitor de vídeo, o computador pode detectar onde o feixe de luz estava na tela quando o sensor detectou primeiramente sua luz.

Este algoritmo teoricamente é mais rápido que os outros propostos, pois com apenas uma imagem será possível detectar o alvo. Não será preciso desenhar um quadro percorrendo a tela, pois o próprio monitor desenha uma imagem percorrendo toda a área de vídeo. A partir deste processo poderá ser indicado o alvo.

Esta proposta de algoritmo não foi implementada neste trabalho devido o uso de equipamento limitado. Para funcionamento exato da detecção, deve-se usar componentes de alta velocidade e precisão visto que os monitores atualizam as imagens em frequências acima de 50Hz. Ainda deve ser feito um estudo considerando o tempo que a pistola leva para detectar, calcular e enviar uma resposta ao computador.

4 DESENVOLVIMENTO DA PISTOLA

Este capítulo descreve as características gerais da pistola e os requisitos que levaram à sua construção. Também descreve seu funcionamento e uma proposta de construção que atende os requisitos.

A pistola basicamente é constituída de um sensor óptico e outros dispositivos que interpretam os sinais deste sensor e transmitem ao jogo, conforme Mendes (1997). A pistola funciona como um detector de luminosidade, pois há um sensor que detecta a quantidade de luz presente no ambiente.

Para a construção da pistola, foram levados em consideração requisitos como o uso de sensores ópticos, como fotocélulas ou equivalentes. Além disso, a pistola deve ter um dispositivo de calibração, pois os monitores podem ter luminosidade diferente. E para a comunicação com o computador, deve ser utilizada a porta paralela.

4.1 REQUISITOS PRINCIPAIS DA PISTOLA

O protótipo desenvolvido neste trabalho apresenta os seguintes módulos distintos: o *hardware* o *software*. O *hardware* trata-se da pistola eletrônica. O *software* refere-se o jogo com os algoritmos de detecção.

A pistola deve atender os seguintes requisitos:

- a) deve utilizar sensores ópticos, como fotocélulas ou equivalentes, para detectar a coordenada de vídeo (requisito funcional - RF);
- b) deve ser adaptável aos tipos de monitores. Os monitores podem ter intensidades, brilho e luminosidade diferentes entre si. Por isso, a pistola deve ter um dispositivo de calibração que se ajuste ao monitor que está sendo utilizado (requisito não funcional - RNF);
- c) deve ser utilizada a porta paralela do computador ao fazer a comunicação entre a pistola e o computador (RNF);

4.2 CIRCUITO ELÉTRICO DA PISTOLA

A figura 5 mostra o diagrama do circuito eletrônico da pistola construída como proposta para este trabalho. Como indicado no diagrama, a pistola possui as seguintes entradas e saídas:

- a) Alimentação de uma fonte externa representada através dos pólos $+V_{\text{fonte}}$ (pólo positivo da fonte) e $-V_{\text{fonte}}$ (pólo negativo da fonte)
- b) Tensões da porta paralela do computador, representada através de $+V_{\text{pp}}$ (tensão positiva) e GND (terra).
- c) Uma saída que retornará GND se a pistola está mirando num ponto escuro ou $+V_{\text{pp}}$ se for num alvo claro.

Para seu funcionamento, a pistola é ligada a uma fonte externa de 5V a 12V. Logo após a entrada da fonte, encontram-se dois resistores, R1 e R2, de mesmo valor e dispostos em série. Assim, entre estes dois resistores, pode-se obter a tensão média entre os dois pólos da fonte externa. Considera-se que a tensão entre os resistores é de 0V e este ponto é ligado ao GND da porta paralela.

Para detectar a luminosidade, é utilizada uma fotocélula como sensor de luz. O sensor pode ser comparado a um resistor que varia a sua resistência conforme a luminosidade incidida nele. Em ambientes escuros o sensor possui baixa resistência que tende a zero com a ausência de luz. Quanto mais claro for o ambiente, maior será a resistência, cerca de $1M\Omega$ em ambientes bem iluminados.

Visto que a luminosidade de um monitor pode variar de acordo com os modelos ou outras variáveis, é adicionado um potenciômetro ao circuito, identificado no diagrama como P1, que varia de 0 a $1M\Omega$, assim como o sensor ótico, mas conforme ajuste do usuário. O potenciômetro deve ser ajustado para ter uma resistência intermediária entre a resistência equivalente ao sensor mirando num alvo luminoso e um alvo escuro. Desta maneira, o potenciômetro servirá de referência para saber quando o sensor está mirando em um ponto luminoso, ou seja, quando o sensor possui resistência maior que a do potenciômetro. Se a resistência do sensor for menor que a do potenciômetro, pode-se concluir que o alvo é escuro. Portanto, o problema de detectar se o alvo é luminoso ou escuro se resume em saber se a resistência do sensor é maior ou menor que a do potenciômetro.

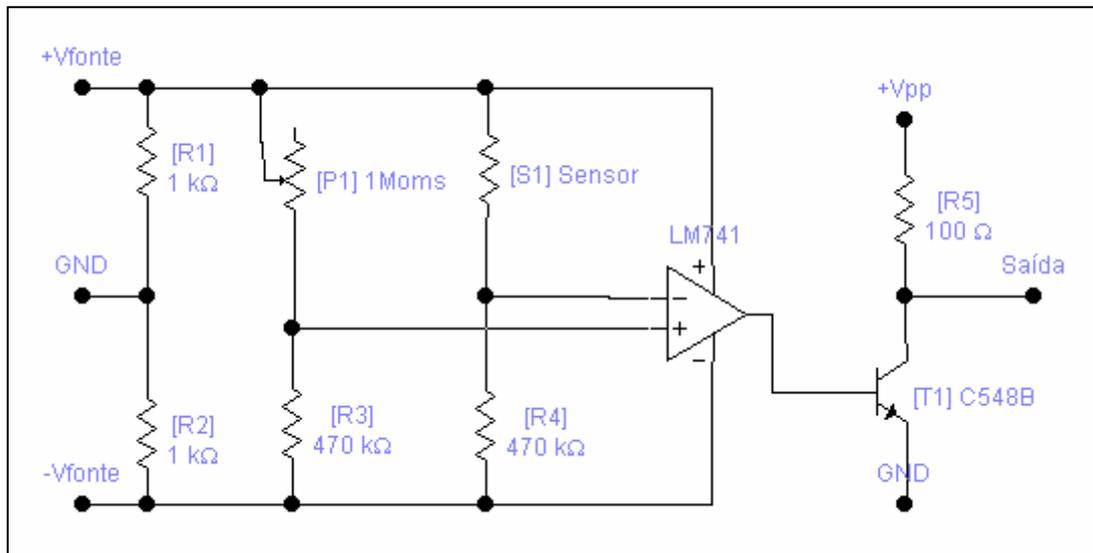


FIGURA 5 - Circuito eletrônico da pistola

Para comparar o sensor com o potenciômetro é utilizado um amplificador operacional, neste caso o modelo LM741 da *National* (National, 2003), que neste circuito está disposto como um comparador de tensões. As tensões usadas para comparação são as encontradas nos pontos intermediários entre o sensor, o potenciômetro e os seus respectivos resistores ligados em série, R3 e R4. Se o sensor ou o potenciômetro, ficarem com valores iguais aos seus resistores em série, a tensão no ponto intermediário será de 0V, como ocorre no caso dos resistores R1 e R2. Se a resistência do sensor, por exemplo, for maior que a do resistor R4, há maior impedimento para a tensão positiva da fonte que o resistor R3 tem para a tensão negativa da fonte. Assim a tensão neste ponto tenderá para valores aproximados à tensão negativa quanto maior for a resistência do sensor à tensão positiva. Por outro lado, se o sensor tender à resistência zero, a tensão aproximará à tensão positiva. Assim, obtém-se as tensões para comparação do amplificador operacional. Maiores detalhes sobre a tensão em associações de resistores podem ser obtidos observando as leis das tensões de Kirchof em Silva (1995).

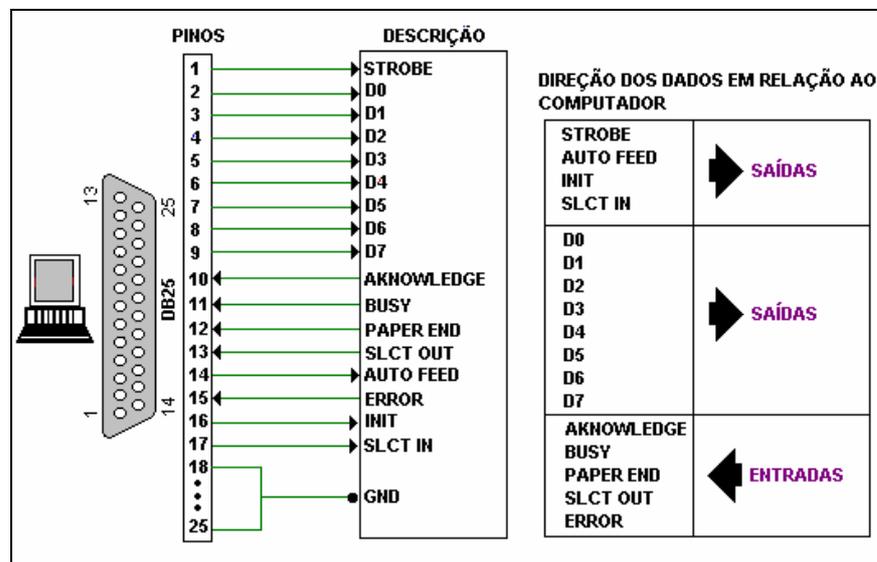
Resumidamente, o amplificador operacional funciona da seguinte forma: se a tensão do sensor for menor que a tensão do potenciômetro, apresentará na sua saída a tensão negativa. Caso contrário, apresentará a tensão positiva.

A saída do amplificador operacional está ligada ao transistor T1 que funciona como uma chave. Se a tensão na sua base, que por sua vez está ligada à saída do amplificador operacional, for positiva, o transistor não fará resistência e abrirá passagem para a tensão

GND da porta paralela. Como há o resistor R5 dificultado a passagem da tensão +Vpp da porta paralela, a saída do circuito então será GND. Caso o amplificador operacional forneça à base do transistor uma tensão zero ou negativa, o transistor fará total resistência ao GND da porta paralela. A saída do circuito então será +Vpp.

4.3 PORTA PARALELA

A porta paralela é uma interface de comunicação entre o computador e um periférico. No caso do jogo, o periférico é a pistola que envia dados ao computador através da porta paralela, chamada de LPT1. A figura abaixo mostra em detalhes a porta paralela e seus pinos.



Fonte: Messias (2003)

FIGURA 6 – Esquema de funcionamento da porta paralela

Para o jogo ler os sinais elétricos que a pistola enviou pela porta, deve ser acessado o endereço de memória da porta paralela, que se refere aos pinos de entrada da porta LPT1. Para isso, os sinais elétricos devem estar em uma faixa estabelecida de tensão. Segundo Messias (2003), um sinal elétrico é considerado como nível lógico zero quando a tensão está entre 0V e 0,4V. Um pino está com sinal em nível lógico 1 quando a tensão está entre 3,1V e 5V.

O computador envia para a pistola os sinais GND e +Vpp, que é conseguida ligando a um pino de saída com sinal 1. O jogo deve enviar este sinal disponibilizando no endereço de saída porta LPT1. E para ler a pistola, os pinos 12 e 11 estão ligados à saída da pistola que receberão os sinais que poderão ser lidos no endereço de entrada da porta paralela.

5 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo será apresentado o protótipo que foi desenvolvido para demonstrar a detecção do tiro usando os algoritmos de detecção linear, binária e quaternária.

5.1 REQUISITOS PRINCIPAIS DO PROTÓTIPO

O software deve atender os seguintes requisitos:

- deve apresentar um desenho que servirá como alvo a ser atingido com a pistola.
Este alvo deve estar em movimento constante e aleatório através de toda a área do vídeo (RF);
- deve ser emitido um aviso sonoro ou visual quando a pistola for disparada (RF);
- deve possuir uma tela de calibração e configuração que auxiliará o usuário a ajustar a pistola ao monitor em uso (RNF);
- o tempo para o jogo detectar o ponto que a pistola está mirando deve ser menor ou igual a meio segundo (RNF).

5.2 ESPECIFICAÇÃO

A especificação da ferramenta é apresentada na forma de um diagrama de classes da UML. A ferramenta CASE utilizada para esta especificação foi o *Rational Rose 2000* da *Rational* (Rational, 2003). A seguir encontra-se o diagrama de classes do jogo.

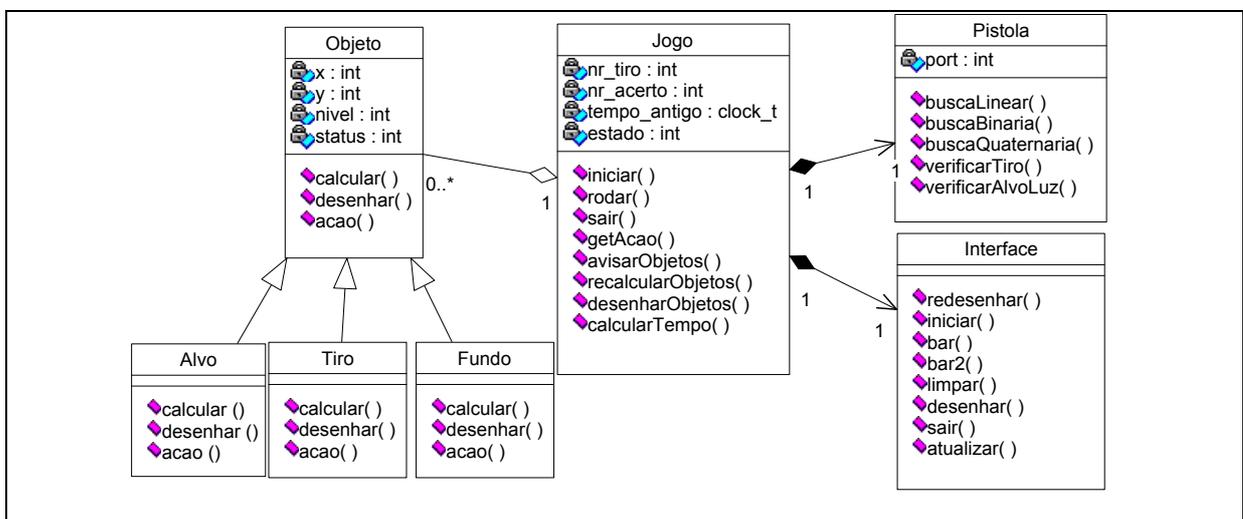


FIGURA 7 – Diagrama de classes do jogo

As classes criadas para o jogo estão descritas a seguir:

- a) Objeto: possui informações e métodos para os objetos ou personagens que fazem parte do jogo. Esta é uma classe abstrata, deverá ser implementada pelas suas subclasses. Possui atributos referentes ao posicionamento, ao estado do objeto e ao nível de posicionamento da tela. Quando houver dois ou mais objetos que deverão ser desenhados na mesma área, os objetos de menor nível serão desenhados ao fundo dos objetos de maior nível. Através do estado, o comportamento do objeto poderá ser calculado para reagir às ações do jogo e ser desenhado.
- b) Alvo: classe que herda e implementa a classe Objeto. O método calcular incrementa as posições para o alvo mover-se de modo constante e horizontal no vídeo. O método desenhar desenha um simples retângulo que servirá como alvo. O método ação recebe as mensagens do jogo e pode reagir destruindo o objeto alvo quando ocorrer um evento de tiro na mesma região do alvo. Se o alvo foi atingido, é disparada uma nova mensagem a todos os objetos do jogo para que possam reagir.
- c) Tiro: ao ser criado o objeto de tiro, é enviada uma mensagem aos objetos avisando que um tiro foi dado em uma coordenada detectada. O método desenhar desenha um pequeno círculo para identificar onde foi dado o tiro. Quando o método ação recebe a mensagem de que um alvo foi atingido, o objeto tiro é destruído para não ser desenhado na tela.
- d) Fundo: o método implementado é o desenhar. Este limpa a tela do jogo.
- e) Jogo: classe que gerencia e controla o jogo. Inicia a interface, controla as ações do jogo, dos objetos e a animação do jogo.
- f) Pistola: classe que controla o hardware do jogo e detecta onde o usuário atirou implementando todas as buscas propostas no trabalho.
- g) Interface: classe que controla a interface gráfica do jogo.

A figura abaixo apresenta o diagrama de seqüências. Este diagrama mostra a seqüência em que as ações ocorrem dentro do sistema para rodar o jogo.

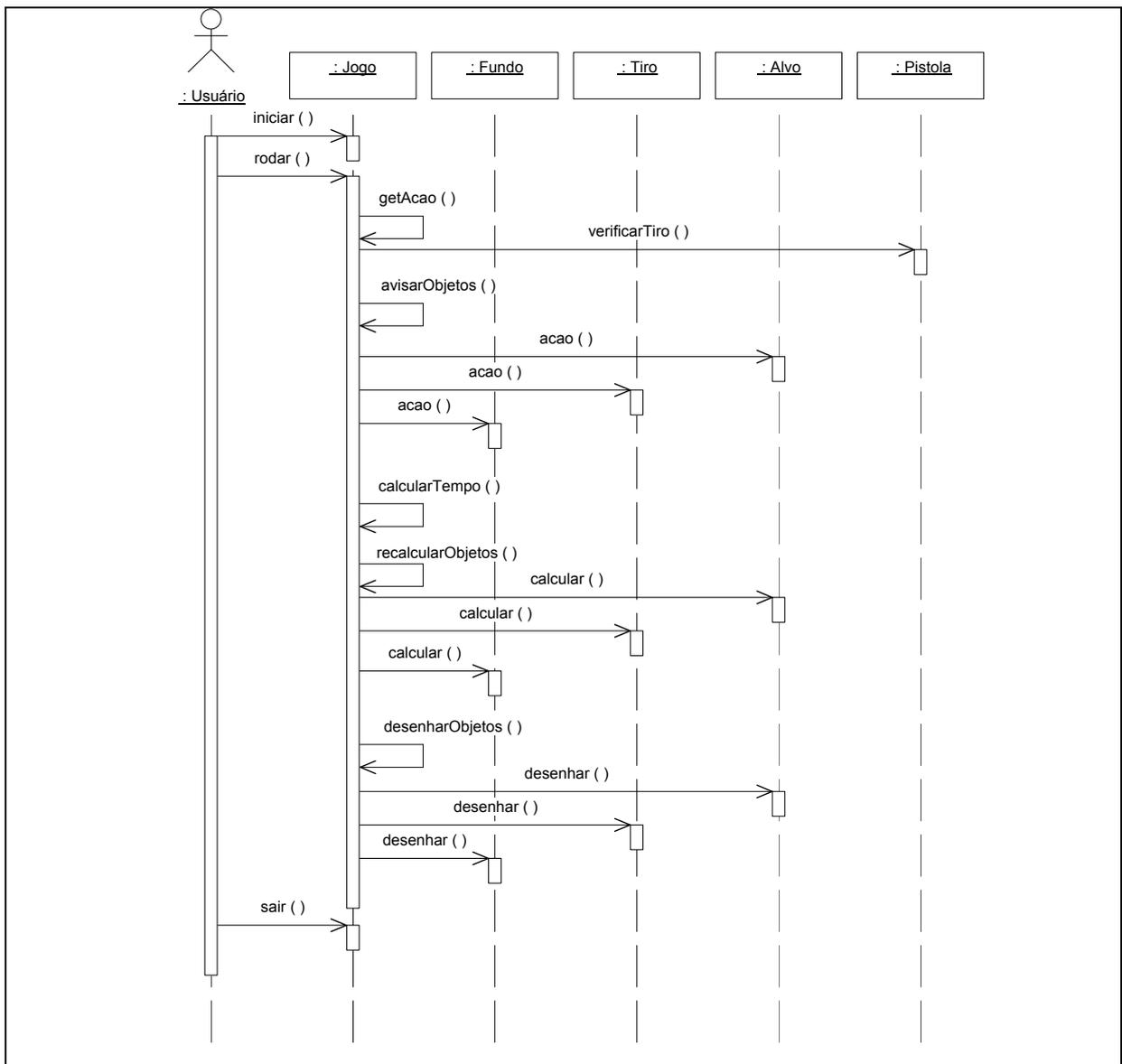


FIGURA 8 – Diagrama de seqüências do jogo

5.3 IMPLEMENTAÇÃO

Nesta seção, apresentam-se considerações sobre a implementação do jogo, a camada de *hardware* e interface. Possui trechos do código gerado e comentários sobre o funcionamento.

A primeira camada é a da interface gráfica representada através da classe Interface. Nela estão todos os métodos para a desenhar gráficos e outras funcionalidades como:

- a) iniciar: inicia e carrega os gráficos para ser utilizado durante o jogo;
- b) bar: desenhar um quadro branco com as coordenadas especificadas;

- c) bar2: desenha um quadro preto com as coordenadas especificadas;
- d) limpar: apaga todos os gráficos desenhados;
- e) atualizar: todos os gráficos são desenhados em uma região de memória temporária que não aparecerão na tela. Este método carrega todos os gráficos ali desenhados para então aparecer em vídeo.

A camada de *hardware* é acessada através da classe Pistola, onde são armazenadas informações como o endereço de memória da porta paralela, precisão e velocidade da detecção do tiro da pistola. A precisão é o tamanho do quadro que será desenhado na tela para a detecção da pistola. A velocidade refere-se a uma pequena pausa em milésimos de segundo que o algoritmo dará para a pistola detectar se o quadro é luminoso ou escuro. Assim pode ser configurado o tempo para ver o algoritmo passo a passo.

O algoritmo de busca linear é descrito no Quadro 1. Como pode-se observar, a lógica é simples, sendo que um quadro do tamanho configurado percorre toda a área de vídeo.

No Quadro 2 encontra-se a implementação do algoritmo de busca binária. Neste a tela é dividido ao meio verticalmente até formar uma faixa do tamanho da precisão configurada. Em seguida, esta faixa é dividida ao meio até encontrar o quadro onde o usuário atirou.

No Quadro 3 é descrito a busca quaternária. É uma lógica um pouco mais elaborada em relação aos outros algoritmos. Onde os quadros são divididos em quatro recursivamente até encontrar o quadro alvo com a precisão configurada.

```

Pixel Pistola::buscaLinear() {
    Pixel p; tela.limpar();
    if (verificaAlvoLuz()) { p.x=p.y=-1; return p; }

    for (float y = MIN_Y; y < MAX_Y; y=y+PRECISAO) {
        for (float x = MIN_X; x < MAX_X; x=x+PRECISAO) {
            tela.bar(x,y,x+PRECISAO,y+PRECISAO); tela.atualizar(); Sleep(VELOCIDADE);
            if (verificaAlvoLuz()) { p.x = x+PRECISAO/2; p.y = y+PRECISAO/2; return p; }
        }
    }
    p.x=p.y=-1;
    return p;
};

```

QUADRO 1 – Função de detecção com algoritmo de busca linear

```

Pixel Pistola::buscaBinario() {
    // limpa a tela e verifica se a mira está fora do monitor
    Pixel p; tela.limpar();
    if (verificaAlvoLuz()) {
        p.x=p.y=-1;
        return p;
    }
    double x0 = MIN_X, xf = MAX_X, y0 = MIN_Y, yf = MAX_Y;

    // varredura horizontal
    do {
        tela.limpar();
        tela.bar2(x0,MIN_Y, (x0+xf)/2,MAX_Y);
        tela.bar((x0+xf)/2,MIN_Y,xf,MAX_Y);
        tela.atualizar();
        Sleep(VELOCIDADE);
        if (verificaAlvoLuz()) x0 = (x0+xf)/2;
        else xf=(x0+xf)/2;
    } while (xf-x0 > PRECISAO);

    // varredura vertical
    do {
        tela.limpar();
        tela.bar2(x0,y0,xf, (y0+yf)/2);
        tela.bar(x0, (y0+yf)/2,xf,yf);
        tela.atualizar();
        Sleep(VELOCIDADE);
        if (verificaAlvoLuz()) y0 = (y0+yf)/2;
        else yf=(y0+yf)/2;
    } while (yf-y0 > PRECISAO);

    p.x=(xf+x0)/2;
    p.y=(yf+y0)/2;
    return p;
}

```

QUADRO 2 – Função de detecção com algoritmo de busca binária

```

Pixel Pistola::buscaQuaternaria() {
    // limpa a tela e verifica se a mira está fora do monitor
    Pixel p; tela.limpar();
    if (verificaAlvoLuz()) {
        p.x=p.y=-1;
        return p;
    }
    // varredura
    double x0 = MIN_X, xf = MAX_X, y0 = MIN_Y, yf = MAX_Y;
    do {
        tela.limpar();
        tela.bar2(x0,y0,xf,yf);
        tela.bar(x0,y0, (x0+xf)/2, (y0+yf)/2);
        tela.atualizar();
        Sleep(VELOCIDADE);
        if (verificaAlvoLuz()) {
            xf = (x0+xf)/2;
            yf=(y0+yf)/2;
        } else {
            tela.bar((x0+xf)/2,y0,xf, (y0+yf)/2);
            tela.atualizar();
            Sleep(VELOCIDADE);
            if (verificaAlvoLuz()) {
                x0 = (x0+xf)/2;
                yf=(y0+yf)/2;
            } else {
                tela.bar(x0, (y0+yf)/2, (x0+xf)/2, yf);
                tela.atualizar();
                Sleep(VELOCIDADE);
                if (verificaAlvoLuz()) {
                    y0 = (y0+yf)/2;
                    xf=(x0+xf)/2;
                } else {
                    y0 = (y0+yf)/2;
                    x0=(x0+xf)/2;
                }
            }
        }
    } while (xf-x0 > PRECISAO);
    p.x=(xf+x0)/2;
    p.y=(yf+y0)/2;
    return p;
}

```

QUADRO 3 – Função de detecção com algoritmo de busca quaternária

A classe base que gerencia e controla o jogo é a classe Jogo. Ela inicia a camada de interface e a camada de hardware, controla as ações, os objetos gráficos e a animação do jogo. A lógica principal está no método rodar que é o *loop* principal com os seguintes passos:

- a) detectar a ação do usuário, por exemplo, usuário atirou;
- b) avisar todos os objetos do jogo sobre a ação ocorrida para que possam reagir;
- c) calcular variação do tempo, em milésimos de segundo, que ocorreu desde a última vez que os objetos foram desenhados;
- d) recalculer objetos com o tempo calculado no passo anterior para que possam mover-se ou reagir de acordo com a variação do tempo. Com o cálculo dos objetos baseado no tempo em milésimos de segundo, não ocorrerá problema como objetos do jogo ficarem lentos em computadores com menos recursos ou rápidos demais em computadores mais potentes. Independente da velocidade do computador, os objetos reagirão e se movimentarão na mesma velocidade;
- e) desenhar os objetos. Todos os objetos, começando pelos objetos de nível de fundo, serão desenhados em uma área temporária de memória e em seguida aparecerão instantaneamente na tela, criando uma animação sem efeitos indesejados como objetos piscarem devido à constante atualização dos gráficos.

No Quadro 4 está o código principal do jogo.

```

void Jogo::rodar() {
    int acao, tempo=calcularTempo(0);
    do{
        acao = getAcao();
        avisarObjetos(acao);
        tempo = calcularTempo(tempo);
        recalculerObjetos(tempo);
        desenharObjetos();
    }while(acao!=0);
}

```

QUADRO 4 – Função principal do jogo.

5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

O trabalho foi desenvolvido com o ciclo de vida tradicional de software, inicialmente com análise dos requisitos, estudo das ferramentas disponíveis, projeto do sistema, implementação, testes e validação.

A *software* foi implementado no ambiente de programação *Visual C++ 6.0* da *Microsoft* (Microsoft, 2003), projetado para *Win32 Console Application*. No desenvolvimento da ferramenta foram utilizados os conceitos de orientação a objetos.

Para a visualização gráfica do jogo, os gráficos foram feitos utilizando a API do *OpenGL* (OpenGL, 2003), caracterizada pela alta performance e portabilidade. Para a animação do jogo foi utilizado o utilitário *GLUT* da *Silicon Graphics* (Glut, 1996). Para implementação da interface gráfica da aplicação, composta por botões e caixas de texto, foi utilizada o sistema de interfaces com o usuário *IUP* (Iup, 2003), desenvolvido no TeCGraF (Grupo de Tecnologia em Computação Gráfica), um laboratório do Departamento de Ciências da Computação na Pontifícia Universidade Católica do Rio de Janeiro. Suas principais vantagens são a portabilidade, alta performance, devido ao fato de que usa interfaces de sistemas nativos, e sua simplicidade.

Para comunicar-se com a porta paralela do computador, foi utilizada uma DLL (*Dinamic Link Libraries*) desenvolvida para ambientes *WindowsNT/2000/XP* disponível no site da *Logix4u* (Logix4u, 2003) pois estes ambientes bloqueiam o acesso direto às portas por medidas de segurança. Esta DLL possui funções de leitura e escrita na porta-paralela. A função de leitura disponibilizada pela DLL foi utilizada para implementar a função *verificaAlvoLuz()* do jogo, que lê o sinal de luminosidade proveniente da pistola.

5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

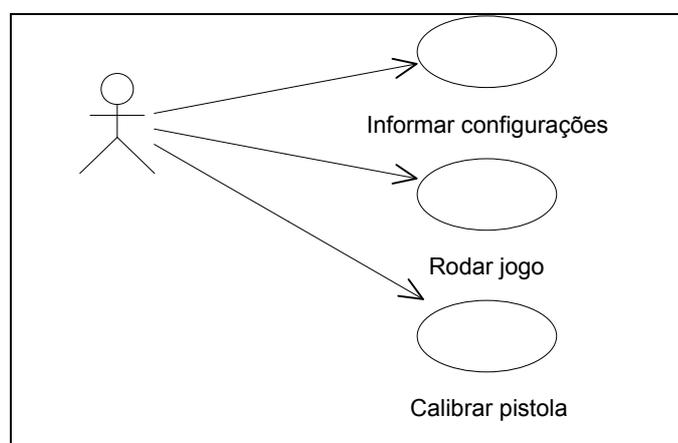


FIGURA 9 – Diagrama de casos de uso do jogo.

O jogo possui características didáticas que demonstram como é a detecção do tiro. Como mostra o diagrama de casos de uso acima, o usuário pode utilizar as seguintes funcionalidades:

- Informar configurações: informar a velocidade de detecção, a precisão do tiro através do tamanho dos quadros que serão detectados, endereço da porta paralela e qual algoritmo deseja visualizar em funcionamento quando a pistola atirar;
- Rodar o jogo: ao ativar esta opção o jogo entra em atividade apresentando objetos na tela, lendo a pistola e controlando as ações que ocorrem;
- Calibrar pistola: usuário tem um auxílio para ajustar sua pistola ao vídeo utilizado. Neste caso de uso, a tela fica cinza, o usuário deve mirar para a tela e seguir as instruções para ajuste do potenciômetro.

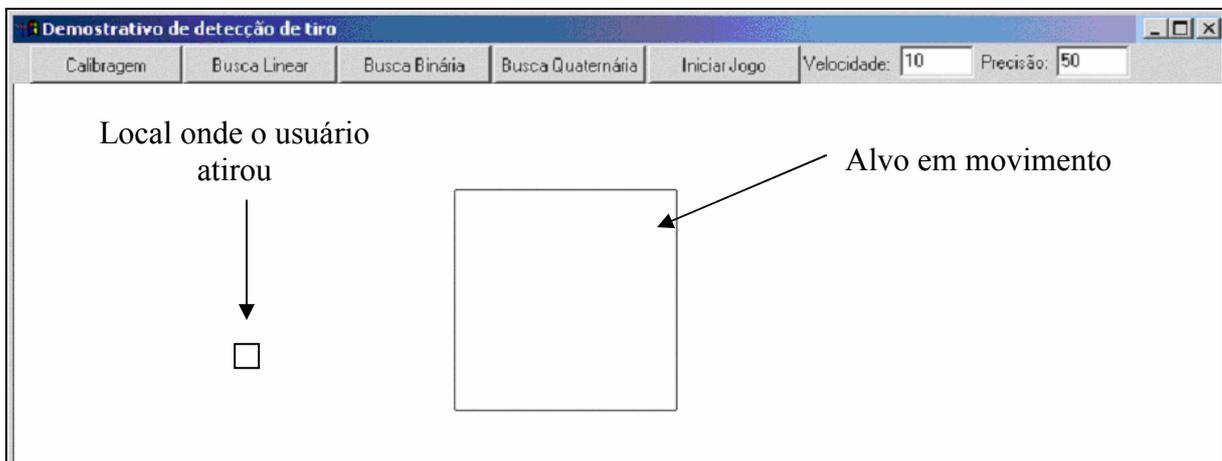


FIGURA 10 – Tela do protótipo desenvolvido

A figura 10 mostra a tela do jogo com um alvo em movimento representado na figura por um quadrado branco. Para iniciar o jogo, o usuário deve primeiro conectar a pistola na porta paralela do seu computador e clicar no botão Iniciar jogo. O jogo é composto por um alvo que percorre a área e o usuário deve atirar neste. É possível escolher também com qual algoritmo de busca o tiro será detectado com os botões de busca linear, busca binária e busca quaternária. Para configurar a velocidade e a precisão, o usuário pode digitar esses valores nos respectivos textos. Aumentando o valor da velocidade, o usuário poderá ver a detecção mais lentamente e observar os passos da detecção. A precisão indica o tamanho do alvo a ser acertado. Finalmente, o botão calibrar apresenta instruções para calibrar a pistola.

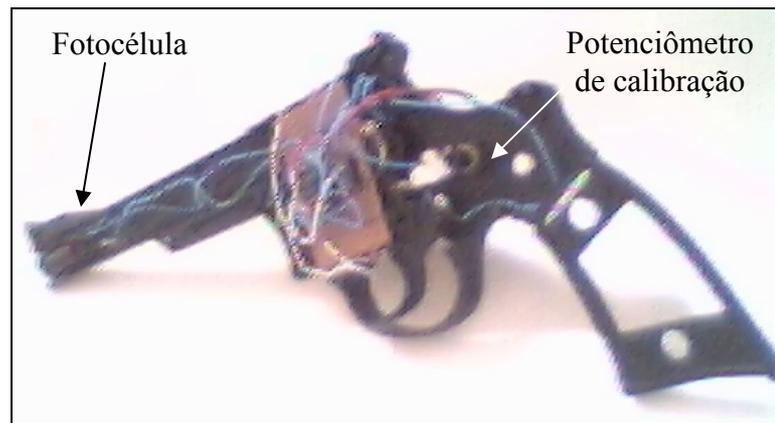


FIGURA 11 – Foto da pistola construída

A figura 11 mostra uma foto da pistola que foi construída. O sensor está encapsulado para que não a luminosidade do ambiente não afete na detecção.

5.4 RESULTADOS E DISCUSSÃO

O trabalho envolveu conhecimentos de várias áreas da computação. Envolve conhecimentos de eletrônica básica para a construção da pistola, bem como arquitetura de computadores para realizar a comunicação entre *hardware* e *software*. Além disso, utiliza conceitos de computação gráfica para o desenvolvimento do jogo.

Sobre a pistola, foi apresentada uma proposta de *hardware* para a solução do problema utilizando fotocélula e um potenciômetro para calibração do dispositivo. A calibração torna a pistola funcional, sendo que não é necessário “fechar as cortinas da sala” para que a pistola detecte precisamente a coordenada do tiro independente da luminosidade do ambiente ou do monitor de vídeo. Porém, há uma quantidade razoável de componentes utilizados na construção da pistola, o que pode tornar o projeto custoso para montagem em grande escala embora os componentes utilizados sejam de baixo custo. Portanto, deve ser estudada uma maneira de diminuir a quantidade de componentes, encontrando uma forma de eliminar talvez o potenciômetro ou o amplificador operacional. Ou ainda construir um novo circuito com o uso de fototransistores no lugar de fotocélulas e fazer uma comparação. Talvez possa haver uma mudança na velocidade ou no comportamento da pistola já que fotocélulas variam a resistência elétrica conforme a luminosidade, de outra maneira, fototransistores são transistores que variam com a luminosidade.

Com pesquisas em sites de busca na Internet, não foram encontrados trabalhos correlatos sobre jogos com pistolas com diagramas do circuito elétrico do *hardware*, embora jogos com pistola sejam populares como entretenimento. Com pouca divulgação do funcionamento do jogo, surgiram confusões no entendimento do seu funcionamento. A pistola levantou conhecimentos sobre detecção do vídeo. Estes conhecimentos podem ser aplicados para a construção de outros dispositivos semelhantes como *light-pens* e outros afins.

Foram também propostos algoritmos otimizados para a busca da coordenada. O algoritmo de busca linear pode demorar um tempo considerável para detectar pontos distantes. Com os algoritmos propostos este problema pode ser contornado.

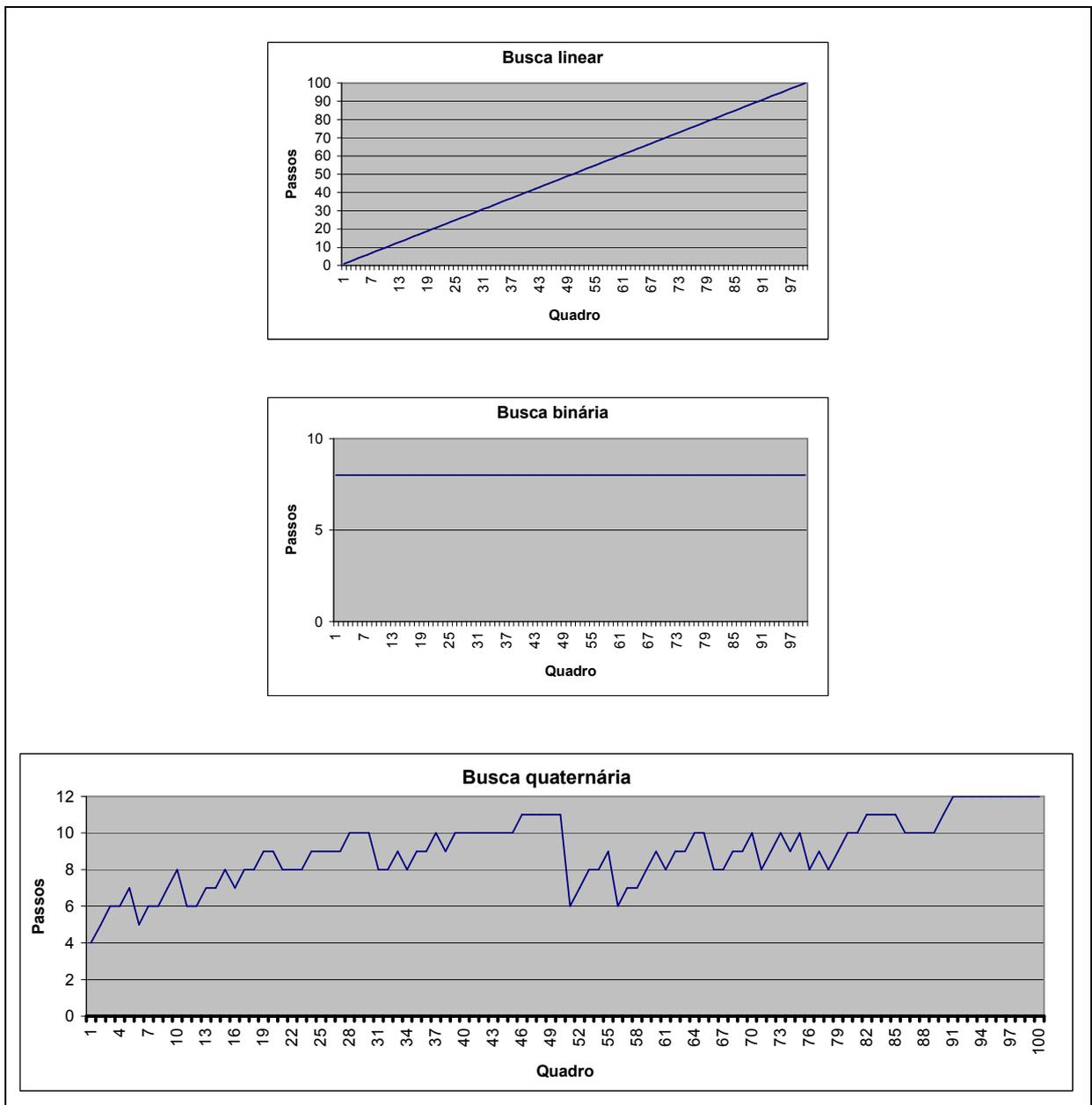


FIGURA 12 – Número de passos necessários para os algoritmos detectarem cem quadros

Foram realizados testes para comparação de desempenho dos diferentes algoritmos de busca. Para isso, foi simulado um usuário dando cem tiros, considerando que o monitor estava dividido em cem quadros de mesmo tamanho que abrangem toda a área de vídeo. O primeiro tiro é dado primeiro quadro no canto superior direito da tela, o segundo tiro no quadro ligeiramente ao lado, e assim sucessivamente até atirar em todos os quadros até o último no canto inferior direito do vídeo.

A figura 11 mostra testes para analisar o comportamento dos algoritmos desenvolvidos. Observa-se que o algoritmo de busca linear detecta mais rapidamente nos primeiros quadros que estavam no canto superior esquerdo, justo onde o algoritmo começa a percorrer o ponto luminoso. Porém o tempo necessário para detectar cresce linearmente à medida que os tiros são dados próximos ao canto inferior direito, onde o algoritmo termina de percorrer toda a área de vídeo.

A busca linear mostrou um aspecto constante. Para detectar qualquer tiro em qualquer posição no vídeo, foram necessárias oito unidades de tempo. O algoritmo de busca linear mostrou ser mais rápido nos primeiros quadros, porém os tiros nos outros quadros superaram muito o tempo da busca em relação à busca binária. Na grande maioria dos casos, a busca binária é mais rápida que a linear.

Tabela 1 – Estatísticas da busca quaternária

Tempo	Tiros detectados com o tempo
4	1
5	2
6	8
7	8
8	19
9	20
10	22
11	10
12	10
Soma	100

Como se pode observar na tabela 1, na busca quaternária algumas regiões são detectadas mais rapidamente e outras não. A tabela anterior mostra quantas vezes um quadro foi detectado com determinado tempo. Como se observa o algoritmo de busca quaternária mostrou-se mais eficiente em apenas 38% dos casos, onde a busca detectou em menos passos que o algoritmo binário detecta.

6 CONCLUSÕES

O trabalho apresentou o desenvolvimento de um protótipo de jogo com pistola. Apresentou-se a interação da pistola com o computador e como é detectado o alvo. Foi desenvolvido um hardware e implementado um software básico cumprindo os objetivos propostos.

O trabalho é multidisciplinar. Englobou conhecimento de algumas áreas da computação como eletrônica e computação gráfica. E o funcionamento como é o processo de detecção de tiro é pouco conhecido, apesar de estes jogos com pistola serem facilmente encontrados em casas de jogos.

Foram utilizados ferramentas portáteis e que oferecem muitos recursos para desenvolvimento de interfaces gráficas. Uma das dificuldades encontradas é o acesso à porta paralela em sistemas operacionais recentes como o Windows 2000. Estes sistemas bloqueiam o acesso para a leitura dos sinais da pistola. Por isso, foi utilizada uma DLL desenvolvida para estes ambientes para contornar este problema. Porém o uso desta DLL influencia na velocidade da leitura dos dados. A partir do momento que o jogo acessar com mais frequência os sinais da pistola, a velocidade de todo o jogo pode ficar comprometida.

Para o desenvolvimento da pistola, foram utilizados componentes de baixo custo. O componente mais caro utilizado foi o amplificador operacional que custou US\$ 0,17 sendo que a construção de toda a pistola não passou de R\$ 4,00. Assim, o trabalho possui algumas limitações devido ao uso de dispositivos de baixo custo, diminuindo a precisão e a velocidade de detecção da coordenada do alvo. A fotocélula utilizada neste trabalho leva 150 milissegundos para estabilizar a resistência em função da luminosidade detectada, o que deixa a detecção lenta para um jogo, pois ela pode ser percebida pelo usuário. A precisão, ou seja, o tamanho do alvo a ser detectado e a distância da pistola até o monitor, depende do formato da pistola e se a luz incide paralelamente no sensor sem interferência da luminosidade do ambiente. Porém estas limitações não comprometem os objetivos propostos para o trabalho que era o desenvolvimento de um jogo com pistola como protótipo.

6.1 EXTENSÕES

Sugere-se para trabalhos futuros o desenvolvimento de uma pistola com dispositivos mais rápidos e precisos como o uso modelos melhores de fotocélulas, ou substituir por fototransistores. Pode ser feito um estudo sobre lentes e como podem ser úteis para aumentar a precisão. O dispositivo também pode ser melhorado diminuindo a quantidade de componentes como o amplificador operacional, potenciômetro e utilizando talvez um microprocessador digital com conversão analógica digital, com a intenção de deixa-lo mais rápido e fácil de montar.

O jogo pode ser ampliado usando pacotes de animação gráfica 3D, som ou permitir que vários usuários possam jogar simultaneamente em rede ou no mesmo computador. Para estes tipos de jogos, podem ser usadas placas aceleradoras 3D, desenhando as imagens em alta velocidade para a detecção rápida ou ainda implementando todo o processo de detecção em linguagem de máquina.

O trabalho ainda pode servir como base para construção de outros dispositivos de multimídia, como *light-pens*, que podem substituir o tradicional *mouse*. Os algoritmos propostos podem ser úteis na detecção da coordenada no vídeo com tais dispositivos.

REFERÊNCIAS BIBLIOGRÁFICAS

- 3DZONEMASTER. **Game Controllers Enter A New Dimension**. Disponível em: <<http://www.gamesdomain.com/gdreview/zones/reviews/hardware/jan98/3dz.html>>. Acesso em: 5 nov. 2003.
- VICARIA, Luciana. Coreografias de fliperama. **Época**, São Paulo, v. 1, n. 212, jun. 2002.
- FOLEY, James D., et al. **Computer graphics** : principles and practice. 2.ed. Reading : Addison-Wesley, c1990. xxiii,1175p.
- GIBBS, Simon J. **Multimedia programming, objects, environments and frameworks**. Boston: Addison-Wesley Publishing Company, 1995.
- GLUT, OPENGL UTILITY TOOLKIT. **Glut API** – version 3. Disponível em: <<http://www.opengl.org/developers/documentation/glut/spec3/spec3.html>>. Acesso em: 5 nov. 2003.
- HOWSTUFFWORKS. **How does the light gun for a video game work?** Disponível em: <<http://www.howstuffworks.com/question273.htm>>. Acesso em: 5 nov. 2003.
- IUP, PORTABLE USER INTERFACE. **IUP – Interface com o usuário portátil** – versão 2.0.1. Brasil, [jul. 2003]. Disponível em: <<http://www.tecgraf.puc-rio.br/iup/>>. Acesso em: 5 nov. 2003.
- LOGIX4U. **Inpout32.dll for WIN NT/2000/XP**. Disponível em: <<http://www.logix4u.net/inpout32.htm>>. Acesso em: 5 nov. 2003.
- MENDES, Carlos Pimentel. **Por dentro do computador** - dispositivos básicos de entrada. Disponível em: <<http://www.novomilenio.inf.br/ano97/cpmcom03.htm>>. Acesso em: 12 maio 2003.
- MESSIAS, Antônio Rogério. **Porta Paralela**. Disponível em: <<http://www.rogercom.com/pparalela/introducao.htm>>. Acesso em: 5 nov. 2003.
- NATIONAL, SEMICONDUCTOR. **National P/N LM741** - Operational Amplifier. China, [2003]. Disponível em: <<http://www.national.com/pf/LM/LM741.html>>. Acesso em: 5 nov. 2003.
- OPENGL, HIGH PERFORMANCE GRAPHICS. **OpenGL Documentation**. Disponível em: <<http://www.opengl.org/developers/documentation>>. Acesso em: 5 nov. 2003.
- QUADTREE. **Quadtree**. Disponível em: <<http://www.geocities.com/grupotrees/quadtree/quadtree.htm>>. Acesso em: 5 nov. 2003.

RATIONAL, SOFTWARE CORPORATION. **IBM Rational Software Support - Documentation**. EUA, [2003]. Disponível em: <<http://www.rational.com/support/documentation/manuals/rose.jsp>>. Acesso em: 5 nov. 2003.

SILVA, Ricardo Pereira e. **Eletrônica básica: um enfoque voltado à informática**. Florianópolis: UFSC, 1995.

VAUGHAN, Tay. **Multimídia na prática**. São Paulo: Makron, 1994.