



UNIVERSIDADE da MADEIRA

Departamento de Matemática e Engenharias

*Wow!Table – Framework de
visão computacional*

Francisco Norberto Sá Perestrelo

MESTRADO EM ENGENHARIA INFORMÁTICA

Junho 2008



Departamento de Matemática e Engenharias

*Wow!Table – Framework de
visão computacional*

Francisco Norberto Sá Perestrelo

MESTRADO EM ENGENHARIA INFORMÁTICA

Dissertação orientada pelo

Professor Doutor Pedro Filipe Pereira Campos

Junho 2008



Agradecimentos

A realização desta dissertação apesar de ser uma actividade individual só foi apenas possível com a colaboração, consciente ou inconsciente, de familiares, colegas e orientador. Por essa razão, desejo expressar os meus sinceros agradecimentos, pois sem eles a realização deste trabalho não seria possível.

Portanto não poderia deixar de mencioná-los, sendo estes:

Ao *Prof. Dr. Pedro Filipe Pereira Campos*, na qualidade de orientador pela oportunidade, apoio e colaboração que sempre me concedeu durante o desenvolvimento do presente trabalho.

Aos meus pais *José Perestrelo* e *Ilda Perestrelo* pela imensa compreensão, apoio, disponibilidade incansável demonstrada quer nos momentos bons, quer nos menos bons.

À minha irmã, *Rosa Perestrelo* agradeço de forma especial pela colaboração, apoio, carinho e compreensão que sempre me demonstrou em qualquer altura do meu percurso académico.

Aos meus colegas e amigos, *Carlos Ferreira, Magno Sousa, Pedro Abreu, Rubina Freitas, Victor Abreu*, entre outros, pela simpatia, amizade e encorajamento demonstrado no desenvolvimento deste trabalho.

A todos aqueles que directamente ou indirectamente contribuíram para a minha formação profissional. Desde já os meus sinceros agradecimentos.



Resumo

A construção de mesas interactivas hoje em dia já não é um conceito muito inovador, uma vez que já existem diversas empresas empenhadas nesta construção. A criação destas mesas apresenta a capacidade de fornecer ao utilizador uma experiência intuitiva e divertida. As mesas possuem diversos domínios de aplicação como por exemplo publicidade, promoção de um local turístico, entretenimento e fins educativos.

De maneira a se poder desenvolver aplicações confiáveis e económicas foi efectuada uma pesquisa sobre as aplicações interactivas existentes, principalmente mesas, que actualmente utilizam a detecção de movimento como meio de comunicação com o computador. Assim sendo, teve-se por objectivo a implementação de aplicações que efectuassem a detecção de movimento através de uma simples Webcam, com o intuito de ser economicamente viável. A utilização da Webcam nestas aplicações contribuiu para que deixasse de ser necessário a utilização do rato e teclado para interagir com a aplicação.

Palavras-Chave: Marketing; Detecção de movimento; Interacção com Webcam; Espaços interactivos.



Abstract

Nowadays, the construction of interactive tables is not an innovative concept since there are already several enterprises that engaged into their development. The conception of these tables has the ability to provide an intuitive and enjoyable experience to users. Tables are widely used in different application fields such as advertising, local tourism endorsement, entertainment and education.

In order to develop reliable and economical applications, research has been carried out on interactive applications available in market, especially tables which currently use motion detection as a way of communication with the computer. Therefore, in order to have an economically available application, the main objective was the implementation of applications to make motion detection through a simple webcam. The use of webcams on these applications contributed with the fact that it is no longer necessary not only to use the mouse and keyboard to interact with the application.

Keywords: Marketing; Motion detection; Webcam interaction; Interactive spaces.



Índice Geral

Agradecimentos	I
Resumo	II
Abstract.....	III
Índice de figuras	VII
Lista de abreviaturas	IX
Capítulo I – Introdução.....	1
1.1. Motivação	1
1.2. Objectivos	1
1.3. Contexto.....	1
1.3.1. Tecnologia Multi-Touch.....	1
1.4. Organização do relatório.....	3
Capítulo II - State of the art	4
2.1. YDreams (1)	4
2.1.1. yBillboards	4
2.1.2. yTable	5
2.1.3. yMirror	5
2.1.4. yStores	6
2.1.5. yWalk	6
2.2. EyeBoard (3).....	7
2.3. EyeStep (4)	8
2.4. Init-Lab “Red Ocean” (5).....	8
2.5. LCI Interactive (6)	8
2.6. Televisão de parede interactiva da Panasonic (7).....	9
2.7. HDTV: Mesa interactiva e inteligente da Panasonic (8)	9
2.8. Interactive LED Coffee Table (9).....	10
2.9. PingPongPlus (10)	10
2.10. iTable (11).....	11
2.11. Microsoft Surface (12)	11

2.12.	Natural Interaction SensitiveTable (13)	12
2.13.	SmartSkin (14)	12
2.14.	GestureTek (15).....	13
2.15.	Multi-Touch Table for virtual Factory (16)	14
2.16.	CUBIT (17)	14
Capítulo III - Implementação		16
3.1.	Linguagem de programação.....	16
3.2.	Hardware.....	17
3.2.1.	Montagem da mesa.....	17
3.3.	Técnicas	20
3.3.1.	Detecção de movimento	20
3.3.2.	Imagem de fundo	23
3.3.3.	Sistema de partículas	24
3.3.4.	Calibração da Câmara.....	26
3.4.	Dificuldades no desenvolvimento dos protótipos.....	27
Capítulo IV - Protótipos Desenvolvidos		29
4.1.	Body Mind Madeira.....	29
4.1.1.	Processo de desenvolvimento.....	30
4.2.	Wow!Bookstore.....	32
4.2.1.	Processo de desenvolvimento.....	32
4.3.	Wow!Tiles	33
4.3.1.	Processo de desenvolvimento.....	34
4.4.	Wow!Cinema	35
4.4.1.	Processo de desenvolvimento.....	36
4.5.	Wow!Track	38
4.5.1.	Processo de desenvolvimento.....	38
4.6.	Wow!Gallery	39
4.6.1.	Processo de desenvolvimento.....	40
4.7.	Wow!Album	41
4.7.1.	Processo de desenvolvimento.....	42
4.8.	Wow!Blackjack	43

4.8.1. Processo de desenvolvimento.....	44
4.9. Wow!Back_Office	45
4.9.1. Processo de desenvolvimento.....	46
Capítulo V - Testes e Resultados.....	56
Conclusões	61
Perspectivas Futuras	62
Bibliografia.....	63



Índice de figuras

Figura 1 - yBilloard da Nokia.....	5
Figura 2 - yTable.....	5
Figura 3 - yMirror.....	6
Figura 4 - yStore Shop Window.....	6
Figura 5 - yWalk. São Paulo fashion week.....	7
Figura 6 - EyeBoard.....	7
Figura 7 - EyeStep.....	8
Figura 8 - Init-Lab "Red Ocean".....	8
Figura 9 - LCI Interactive.....	9
Figura 10 - Televisão parede interactiva.....	9
Figura 11 - Mesa interactiva da Panasonic.....	10
Figura 12 - Interactive LED coffee Table.....	10
Figura 13 - PingPongPlus.....	11
Figura 14 - iTable.....	11
Figura 15 - Microsoft <i>Surface</i>	12
Figura 16 - SensitiveTable.....	12
Figura 17 - Estrutura 8x9 do smartSkin.....	13
Figura 18 - Operação com duas mãos.....	13
Figura 19 - GestureTek Multi-touch Illuminate Table.....	14
Figura 20 - Multi-Touch Table for virtual factory.....	14
Figura 21 - CUBIT.....	15
Figura 22 - Estrutura com o projector posicionado no tecto.....	17
Figura 23 - Estrutura com o projector sobre a mesa.....	18
Figura 24 - Montagem do simulador da mesa.....	19
Figura 25 - Detecção de movimento usando pixels.....	21
Figura 26 - Configuração das propriedades desejadas na câmara.....	22
Figura 27 - Touchlib Configapp.....	23
Figura 28 - As duas versões da técnica imagem de fundo.....	24
Figura 29 - Primeiro sistema de partículas.....	25
Figura 30 - Segundo sistema de partículas.....	25
Figura 31 - Terceiro sistema de partículas.....	26
Figura 32 - Janela de calibração da câmara.....	26
Figura 33 - Body Mind Madeira com uma ou duas folhas.....	29
Figura 34 - Diagrama de classes do Body Mind Madeira.....	31
Figura 35 - Wow!Bookstore.....	32
Figura 36 - Diagrama de classes Wow!Bookstore.....	33
Figura 37 - Wow!Tiles.....	34

Figura 38 - Diagrama de classes do Wow!Tiles.....	35
Figura 39 - Wow!Cinema.....	36
Figura 40 - Diagrama de classes do Wow!Cinema.....	37
Figura 41 - Wow!Track.....	38
Figura 42 - Diagrama de classes do Wow!Track.....	39
Figura 43 - Wow!Gallery.....	40
Figura 44 - Diagrama de classes do Wow!Gallery.....	41
Figura 45 - Wow!Album.....	42
Figura 46 - Diagrama de classes do Wow!Album.....	43
Figura 47 - Wow!Blackjack.....	44
Figura 48 - Diagrama de classes do Wow!Blackjack.....	45
Figura 49 - Interfaces do Wow!Back_Office.....	46
Figura 50 - Diagrama de casos de utilização.....	47
Figura 51 – Diagrama de actividades do caso de utilização “Calibrar câmara”.....	47
Figura 52 - Diagrama de actividades do caso de utilização "Apagar Projecto".....	48
Figura 53 - Diagrama de actividades do caso de utilização “Criar projecto”.....	48
Figura 54 - Diagrama de actividades do caso de utilização “Alterar projecto”.....	49
Figura 55 - Diagrama de actividades do caso de utilização “Configurar informações do projecto”.....	49
Figura 56 - Diagrama de actividades do caso de utilização “Carregar imagens”.....	50
Figura 57 - Janela para calibração da Câmara.....	51
Figura 58 - Interface de apagar projectos.....	52
Figura 59 - Definições do projecto.....	52
Figura 60 - Janela de configuração do Wow!Tiles.....	53
Figura 61 - Janela de configuração do efeito Body Mind Madeira.....	53
Figura 62 - Janela de configuração do efeito Wow!Bookstore.....	54
Figura 63 - Janela para carregar imagens.....	55
Figura 64 - Teste Body Mind Madeira.....	57
Figura 65 - Teste do Wow!Tiles.....	57
Figura 66 - Teste do Wow!Bookstore.....	58
Figura 67 - Gráfico referente ao teste realizado ao protótipo Wow!Track.....	59
Figura 68 - Gráfico referente ao teste do protótipo Wow!Cinema.....	60



Lista de abreviaturas

IV - Infravermelhos

MT - Multi-Touch

TS - Touch screen

SO - Sistema operativo



Capítulo I – Introdução

1.1.Motivação

Quer em aplicações de marketing interactivo, quer em museus e parques de ciência, as inovadoras aplicações baseadas em projecção e processamento de imagem em tempo real necessitam de melhores *frameworks* para a simples montagem, edição e adição de efeitos especiais. Uma *framework* deste género tem igualmente utilidade no que se refere à escalabilidade da solução, por exemplo juntando vários projectores aumenta-se a superfície de interacção, consequentemente sincronização entre aplicações.

1.2.Objectivos

A construção de mesas interactivas já não é um conceito assim tão inovador, ou seja, em termos de estrutura da mesa. Deste modo os objectivos deste projecto foram direccionados mais para a criação de efeitos e, posteriormente, a criação de uma *framework* para estes efeitos, nomeadamente:

- ❖ Criação de efeitos para a mesa interactiva;
- ❖ Desenhar e construir uma *framework* para efeitos visuais e gestão remota de superfícies interactivas, em particular, mesas interactivas;

1.3.Contexto

Este trabalho pode ser inserido em diversos contextos, tais como fins publicitários, promoção de locais turísticos, fins didácticos ou até mesmo apresentação de produtos. Este projecto tem igualmente como âmbito a tecnologia Multi-Touch (MT), uma vez que para uma mesa interactiva, este é um conceito fundamental.

1.3.1. Tecnologia Multi-Touch

O MT é uma tecnologia de interacção homem-computador onde o teclado e o rato são substituídos pela interacção directa com a tela do computador, mesa, parede, touchpad, entre outros. A tecnologia MT oferece uma experiência completamente diferente, pois através da combinação entre hardware e software, as telas MT são capazes de reconhecer toques simultâneos em diferentes pontos da superfície, em oposição ao touch-screen standard, como por exemplo: computador touchpad e ATM,

que reconhecem apenas um toque. Este efeito é conseguido através de uma variedade de meios, incluindo calor, pressão do dedo, alta taxa de captação da câmara, luz IV, captura óptica e captura da sombra, entre outros factores. A interpretação de padrões de toque permite que essas superfícies sejam capazes de compreender gestos e fornecer uma interacção rica com os objectos na tela, o que permite o utilizador a possibilidade de “manipular” os conteúdos.

Diversas empresas já aplicam esta tecnologia na produção de materiais. Uma grande exposição desta tecnologia ocorreu em 2007, quando a Apple anunciou o iPhone e a Microsoft apresentou o Surface. De seguida será realizada uma compilação de alguns usos desta tecnologia de MT em produtos desenvolvidos recentemente:

- ❖ Actualmente, a Apple já contém diversos produtos com esta tecnologia, nomeadamente iPhone, iPod touch, MacBook Air e MacBook Pro. A introdução do iPhone no mercado dos telemóveis foi importante, uma vez que foi o primeiro telemóvel a implementar a tecnologia MT o que permitia os utilizadores usarem o telemóvel sem recorrer a um teclado ou caneta. Esta tecnologia teve tanto sucesso que a Apple expandiu a sua utilização tanto para o novo iPod, como para o MacBook Air.

- ❖ O Microsoft Surface é um computador inserido numa mesa com uma grande tela “touch-responsive” no topo. A mesa utiliza pequenas câmaras (em oposição a pressão dos dedos ou calor) que permite reagir ao toque de qualquer objecto.

- ❖ A Asus incluiu funções MT no touchpad do EEE_PC 900.

Os investigadores desta tecnologia apresentam uma diversidade de aplicações, tais como:

- ❖ Maior experiência enquanto se jantam (encomendar comida, pedir por recargas, dividir a conta do restaurante, pagar contas, entretenimento).

- ❖ Usos governamentais

- ❖ Conceito de mapeamento

- ❖ Uma experiência multimédia.

A tendência é que cada vez mais equipamentos com esta tecnologia se encontrem disponíveis comercialmente, mas a possibilidade de esta substituir por completo a utilização do rato ou do teclado, como por exemplo no Surface e no iPhone, é improvável, pois para determinadas tarefas, tais como digitar textos, os dispositivos tradicionais apresentam um melhor desempenho.

Além disso, a tecnologia MT ainda precisa de tempo para ganhar a confiança do utilizador. O melhor exemplo disso é que o rato, que hoje é considerado indispensável, levou nada menos que 30 anos para se tornar popular, desde sua invenção em 1965.

Baseado neste cálculo, a tecnologia MT, que nasceu há pouco mais de 25 anos (início da década de 80), ainda tem pelo menos cinco anos pela frente para conquistar os corações e mentes do consumidor.

1.4. Organização do relatório

O capítulo I apresenta uma breve introdução ao projecto e define quais os objectivos que se pretende alcançar. Os capítulos que seguem têm a seguinte finalidade:

Capítulo II- State of the Art

Neste capítulo é analisado os diferentes tipos de soluções interactivas existentes, e materiais usados para a construção, tais como câmara IV, RFID, entre outros.

Capítulo III – Implementação

Neste capítulo será analisado todos os aspectos considerados durante a fase de implementação deste projecto. A linguagem seleccionada e as dificuldades encontradas são alguns dos tópicos abordados.

Capítulo IV – Protótipos Desenvolvidos

Neste capítulo é descrito todos os protótipos desenvolvidos durante este projecto.

Capítulo V – Testes e Resultados

Neste capítulo é efectuada uma análise sobre os testes realizados de modo a que fosse possível retirar conclusões sobre os protótipos.

Capítulo VI – Conclusão e Perspectivas Futuras

Neste capítulo é efectuado um resumo de todo o trabalho desenvolvido durante o projecto, sendo igualmente descrito as vantagens e desvantagens das selecções efectuadas durante o desenvolvimento dos protótipos. Assim como, perspectivas futuras tendo em conta o trabalho efectuado.



Capítulo II - State of the art

O desenvolvimento de soluções interactivas ainda é uma área relativamente recente, porém já se consegue aceder a diversas empresas cujo seu objectivo é desenvolver e vender estas soluções.

A maioria das soluções utiliza como input: câmara de infravermelho (IV), cartões RFID, *touch screen* (TS), entre outros. De seguida abordar-se-á sobre algumas destas soluções.

2.1. YDreams (1)

A *YDreams* é uma empresa Portuguesa fundada em Junho de 2000 por especialistas de renome internacional em tecnologia de informação, telecomunicações, processamento de imagem, sistemas de geo-informação e engenharia ambiental, com o intuito de apresentar soluções tecnológicas. A companhia tem como objectivo desenvolver tecnologia pioneira que se concentra principalmente nos meios de comunicação interactivos, à realidade aumentada e jogos para dispositivos móveis, entre outros.

Os produtos desta companhia dividem-se em quatro áreas comerciais, nomeadamente Ambiente, Educação & Cultura, Entretenimento e Publicidade.

Até ao momento esta companhia já colaborou com empresas de renome mundial, tais como Adidas, Nokia, Vodafone, entre outras. De seguida irei falar sobre alguns dos produtos da *YDreams*. (2)

2.1.1. yBillboards

Ao contrário dos meios de comunicação tradicionais como por exemplo televisão, rádio e imprensa, o *yBillboards* (Figura 1) consegue conjugar vídeo, animação, gráficos e texto com um novo factor decisivo, interactividade. Sensores incorporados na estrutura reagem à passagem dos transeuntes, tiram fotos e revelam as imagens automaticamente no ecrã, o qual permite ainda oferecer a possibilidade de fazer o download de conteúdos interessantes, nomeadamente imagens, vídeos e música para dispositivos móveis via *Bluetooth* ou tecnologia infravermelha.

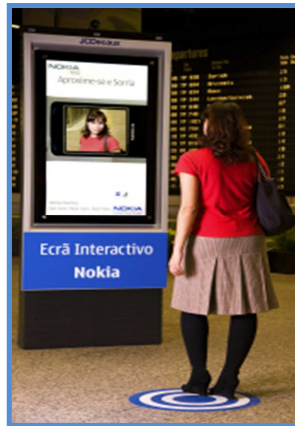


Figura 1 - yBillboard da Nokia.

2.1.2. yTable

A plataforma lisa altamente interactiva é ideal para apresentar todos os tipos de conteúdo de multimédia. Uma imagem temática é projectada numa mesa com um design especial. Os utilizadores usam marcadores especiais, ou mesmo as suas próprias mãos para explorar o conteúdo projectado, quer sejam texto, imagens, vídeo ou animações com som.

O *yTable* (Figura 2) introduz uma excitante aproximação que apela a todos os grupos etários. É intuitivo, fácil de utilizar e trabalhar quer como um instrumento didáctico poderoso, quer como instrumento de apresentação de eventos promocionais, feiras temáticas e exposições.



Figura 2 - yTable.

2.1.3. yMirror

O *yMirror* (Figura 3) produz ambientes imersivos onde os visitantes ao usar movimentos corporais podem interagir com a projecção. Os movimentos dos utilizadores fazem com que as imagens se movam ou se transformem em efeitos digitais que são inesperados e estimulantes.

Os utilizadores podem mover objectos virtuais, descobrir logótipos de uma marca escondidos ou produtos na tela. Os resultados, sempre surpreendentes e

divertidos, são um modo excelente de associar marcas ou produtos a experiências divertidas.



Figura 3 - yMirror.

2.1.4. yStores

A *YDreams* criou o conceito da loja inteligente, a *yStores*. Através da conjugação de vários tipos de soluções interactivas, que contribuem para que tanto a marca como o consumidor saiam a ganhar, provando que comprar não tem de ser uma experiência enfadonha.

O *yStores* (Figura 4) pode se dividir em três partes *Shop Window*, *Virtual Promoter* e *Kiosks*. O *shop window* é um filme semi-transparente com um sistema sensível ao toque, sendo este inserido na janela da loja, em conjunto com uma câmara discreta. Quando uma pessoa passa, um sensor de movimento detecta a sua presença e activa o promotor virtual. Este promotor virtual pode ser um vídeo de um actor ou então uma personagem 3D, que inicia uma conversa casual com o cliente convidando este a entrar na loja, tocar na janela e pesquisar o conteúdo multimédia ou um catálogo animado.



Figura 4 - yStore Shop Window.

2.1.5. yWalk

O *yWalk* (Figura 5) transforma qualquer espaço num espaço de exploração quer para crianças, quer para adultos. Os conteúdos digitais são projectados verticalmente no chão, que permite detectar a presença dos visitantes, activando a animação e o som. As

animações são provocadas por movimentos naturais e intuitivos, que interagem com o utilizador para criar um ambiente surpreendente.

As possibilidades de utilizar este produto são infinitas, pois tanto pode ser usado tanto para publicidade como para fins didácticos



Figura 5 - yWalk. São Paulo fashion week.

2.2. EyeBoard (3)

O *EyeBoard* (Figura 6) é um sistema de mesa interactiva que converte uma placa normal numa experiência divertida. Ao implementar a tecnologia reactiva *EyeClick*, cada dispositivo eléctrico do *EyeBoard* é especialmente adequado de tal forma que possa ser usado em vários campos tais como publicidade, apresentação de produtos, jogos interactivos, entre outros. Este sistema é activado pela detecção de movimento. Desenvolvido com uma perspectiva clara de consumismo, o *EyeBoard* oferece versatilidade incomparável.

O *EyeBoard* é um sistema desenhado para detectar utilizadores, pelo toque na exposição assim como objectos colocados na mesa, nomeadamente canecas, placas ou mãos. O *EyeBoard* suporta definições de MT e suporta também todas as aplicações *EyeStep*.

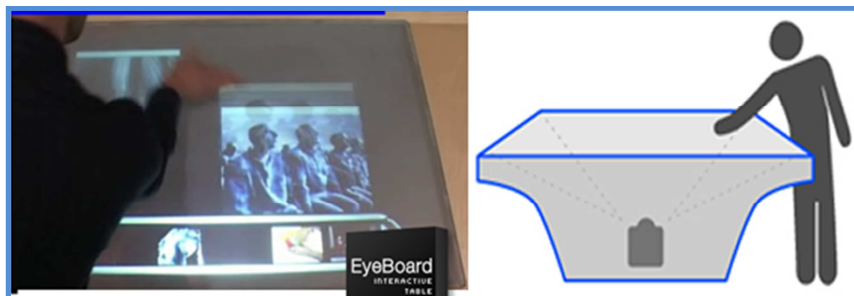


Figura 6 - EyeBoard.

2.3. EyeStep (4)

O *EyeStep* (Figura 7) produz a interacção de corpo inteiro, o qual combina o movimento humano com um chão reactivo. O *eyeStep* converte qualquer área aberta em uma experiência contínua de movimento, acção, diversão e excitação. Deste modo, integra um rico conjunto de efeitos e templates, incluindo animações, vídeo, aplicações gráficas que permitem que o *EyeStep* seja simples de usar.

O *EyeStep* é uma plataforma de movimento interactiva. Uma imagem é projectada do tecto para o chão. Uma “white sticker” ou “coating” pode ser aplicada no chão caso este seja muito escuro. Uma câmara é colocada além do projector para capturar os movimentos dos utilizadores.

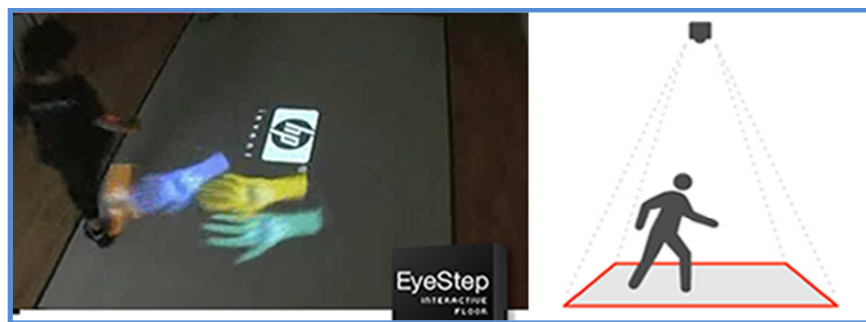


Figura 7 - EyeStep.

2.4. Init-Lab “Red Ocean” (5)

O “*Red Ocean*” (Figura 8) é uma aplicação que possui um sistema de detecção de movimento que pode ser posto em exibição em diferentes locais, que contribui para que a audiência possa experimentar a tecnologia *high-tech*. É efectuada uma projecção vertical cuja imagem se encontra projectada no chão. A imagem projectada é composta de vários ícones que se distanciam quando é detectado movimento das pessoas.

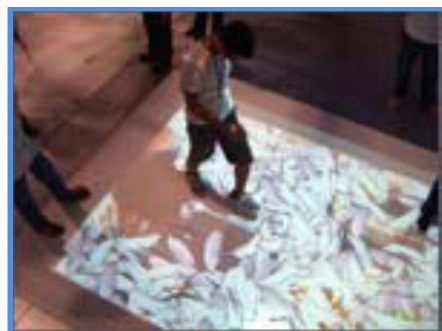


Figura 8 - Init-Lab "Red Ocean".

2.5. LCI Interactive (6)

O vídeo interactivo pode ser usado no chão ou na parede. Os visitantes podem interagir com o vídeo. Isto pode ser utilizado numa grande variedade de funções, como

por exemplo publicidade, divulgação de algum produto, atracções, entre outros. A tecnologia interactiva pode ser usada na combinação com a projecção ou outras tecnologias de tela.



Figura 9 - LCI Interactive

2.6. Televisão de parede interactiva da Panasonic (7)

Como já se sabe, quer a televisão, quer a internet tem um papel importante na vida das pessoas. A Panasonic teve isto em consideração para desenvolver uma “parede televisão” interactiva TS. Esta “parede” permite não só ver televisão, mas também navegar pela internet, aceder correio electrónico, ouvir música e até mesmo ver quem está a tocar na campainha. Como é uma tela TS, dá-lhe a interactividade para jogar basquetebol, tocar piano, entre outras funções. Para controlar uma televisão tão excepcional, é preciso um comando de idêntica qualidade como pode se observar na Figura 10.



Figura 10 - Televisão parede interactiva.

2.7. HDTV: Mesa interactiva e inteligente da Panasonic (8)

Após a Panasonic ter apresentado a televisão de parede interactiva, recentemente apresentou a televisão mesa interactiva (Figura 11). Ao colocar um telefone na mesa e com o auxílio de um “pequeno” agente, a mesa pode ler a informação, nomeadamente dados pessoais, agenda e contactos, presentes no seu telemóvel. Este agente tem outras

funções tais como controlo da temperatura, televisão, entre outras. Para comunicar com este agente, uma vez que esta mesa é TS, basta o utilizador tocar na mesa o que permite usufruir das diferentes funções que o agente dispõe. O ecrã não é igual aos dos *PDA*'s, é muito mais sólido de forma a possibilitar que seja possível por objecto no ecrã, sem provocar qualquer tipo de dano.



Figura 11 - Mesa interactiva da Panasonic.

2.8. Interactive LED Coffee Table (9)

Desde o momento que acontece um movimento, a mesa vem à vida. Os led's só são activados quando é detectado movimento. Usando uma rede de 32 sensores ópticos de IV activos e passivos, ele detecta o movimento em cima da mesa e responde com iluminação.



Figura 12 - Interactive LED coffee Table.

2.9. PingPongPlus (10)

O PingPongPlus (Figura 13) é uma versão digital do jogo clássico de ping-pong. É jogado com raquetes normais, bolas e uma mesa “reactiva” que incorpora sensibilidade, som e tecnologias de projecção. Os projectores exibem padrões de luz e sombra sobre a mesa. Quando a bola toca na mesa deixa imagens de ondulação e o ritmo do jogo é acompanhado por efeitos musicais e visuais.

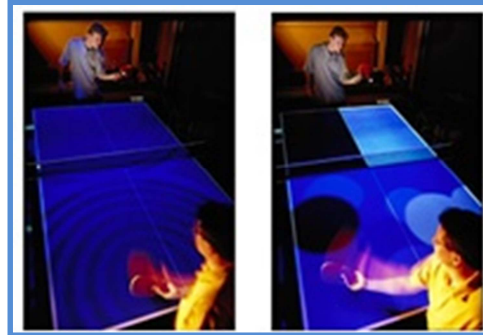


Figura 13 - PingPongPlus

2.10. iTable (11)

Ao usar as capacidades de gerência única da Catchyoo, a Catchyoo *iTable* (Figura 14) contribui para a interactividade de qualquer mesa, com a projecção descendente. A biblioteca FX da Catchyoo pode ser instalada em clubes, bares, museus e lugares de recreação para produzir experiências únicas e espantosas. Como é o caso do Catchyoo *playground*, o *iTable* pode produzir uma *network* de publicidade. Múltiplas *iTables* podem ser anexadas de forma a criar um espaço interactivo.



Figura 14 – iTable.

2.11. Microsoft Surface (12)

O Microsoft *Surface* (Figura 15) é destinado a parceiros comerciais, nomeadamente hotéis, restaurantes ou locais de entretenimento público que estão a procura de oferecer ao seu convidado uma excelente experiência interactiva. A empresa exibiu um computador em formato de mesa, que fez com que o rato e o teclado sejam substituídos por meios naturais de interacção como a voz, uma caneta ou mesmo a mão. O Microsoft *Surface*, que oferece uma tela de 30 polegadas sob uma tampa de plástico resistente, permite que as pessoas toquem e movimentem objectos na tela para as mais diversas tarefas, por exemplo desenho digital, montagem de quebra-cabeças virtuais ou mesmo pedidos de comida no cardápio online de um restaurante. O computador também

reconhece e interage com outros aparelhos colocados sobre a sua superfície, de modo que os utilizadores de telemóveis poderão comprar *ringtones* facilmente ou um grupo de pessoas sentadas em torno de um *Surface* poderão olhar as fotos armazenadas em uma câmara digital colocada sobre a superfície.

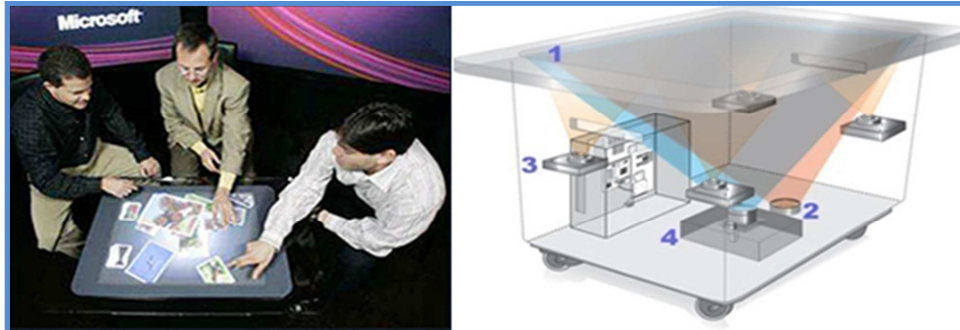


Figura 15 - Microsoft *Surface*.

2.12. Natural Interaction SensitiveTable (13)

A SensitiveTable (Figura 16) é uma mesa MT que detecta mãos e objectos em contacto com ela a 60 frames por segundo. A aplicação de software permite a criação de experiências naturais. A mesa está equipada com microfones e antenas RFID nos seus bordos.

O sistema consiste numa mesa interactiva “todo-em-um”, com o tamanho de 54”, um computador escondido e uma superfície MT óptica que abrange toda a área visível, proporcionando, desta forma uma manipulação directa dos conteúdos digitais com as mãos. O sistema fornece uma peça de mobiliário para espaços públicos, de forma compatível com o meio envolvente, que pode suportar interacção para a manipulação de conteúdos digitais.

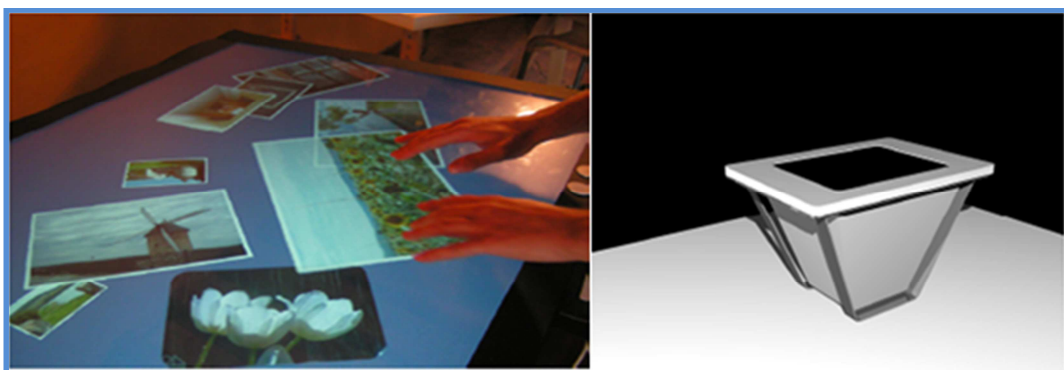


Figura 16 - SensitiveTable.

2.13. SmartSkin (14)

SmartSkin é uma nova arquitectura de sensores de modo a fazer superfícies interactivas que são sensíveis a mãos e gestos humanos. Este sensor reconhece múltiplas posições de mão e determina a distância entre a superfície e a mão usando sensores

capacitivos e uma malha em forma de antena. Em contraste com sistemas de reconhecimento de movimento baseado na câmara, todos os elementos sensíveis podem ser integrados na superfície, fazendo assim que não haja problemas de iluminação e oclusão.

A Figura 17 mostra o esquema de uma mesa interactiva com um sensor smartSkin 8x9.

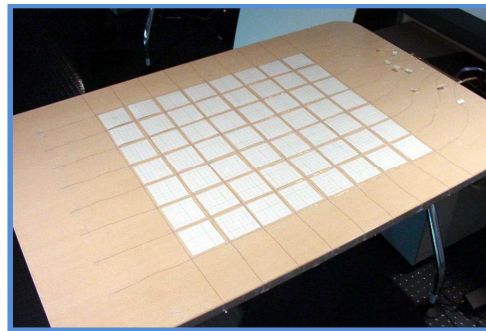


Figura 17 - Estrutura 8x9 do smartSkin.

A capacidade de múltiplas mãos pode ser usada para melhorar a manipulação dos objectos. Por exemplo, um utilizador pode mover objectos independentemente com uma mão. Pode também “concatenar” dois objectos usando as duas mãos, como ilustrado na Figura 18.

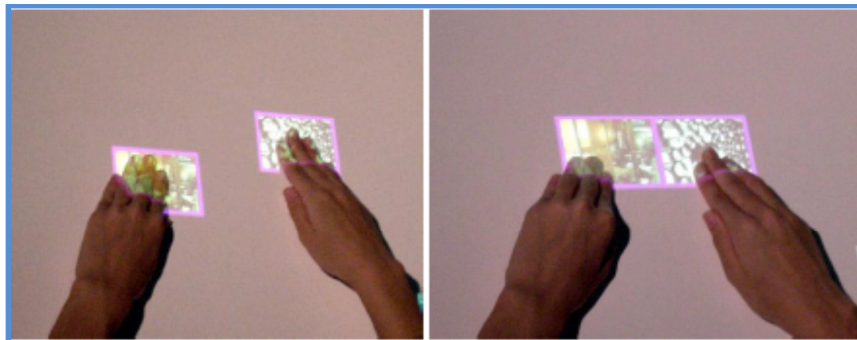


Figura 18 - Operação com duas mãos.

2.14. GestureTek (15)

A tecnologia MT da GestureTek tem aplicações ilimitadas que vão desde stands, centros de apresentação, discotecas, restaurantes, hotéis, atracções turísticas, entre outros. Usando tecnologia de câmara especializada e tecnologia de controlo gestual, a GestureTek tem disponibilizado, ao longo dos anos, mesas interactivas. A tecnologia MT permite as pessoas interagirem com os conteúdos multimédia, como por exemplo aceder a informação, jogar, criar efeitos especiais, manipular fotografias, ate mesmo ver publicidade de uma forma verdadeiramente única e atraente.

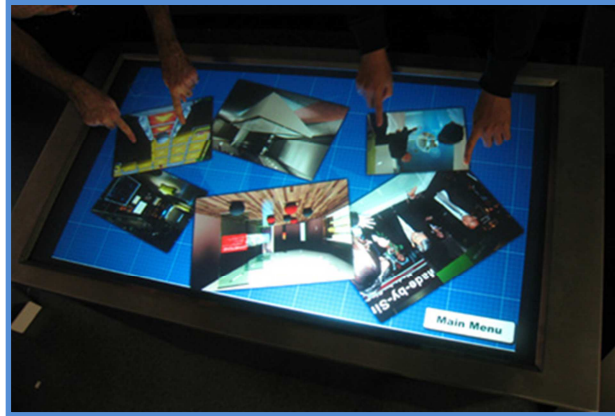


Figura 19 - GestureTek Multi-touch Illuminate Table.

2.15. Multi-Touch Table for virtual Factory (16)

A Fraunhofer-IGD criou uma mesa MT (Figura 20) para interacção com ambientes virtuais, cuja apresenta alta resolução e visualização de dados em tempo real com um tamanho de 150x90 cm. Um método óptico é usado para detectar os dedos do utilizador, no qual recorre a uma câmara para observar a superfície da mesa, sendo utilizado para tal um software especial baseado na visão denominado de *VisionLib*.

Tendo por um lado o MT e por outro a visualização de dados em 3D, o utilizador pode tocar o conteúdo em 3D e interagir com ele sem problemas. A técnica de interacção é intuitiva, fácil de aprender e de baixa complexidade de modo a garantir uma boa e fácil utilização.

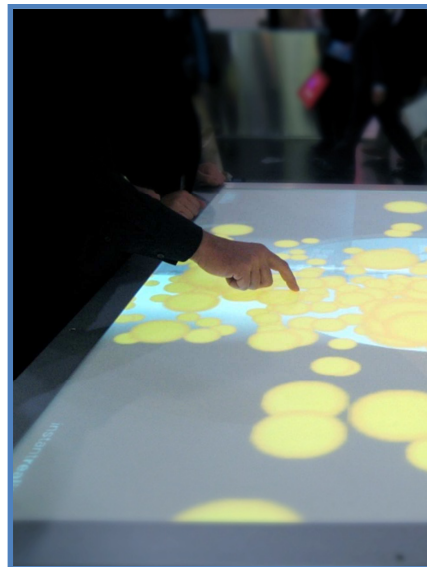


Figura 20 – Multi-Touch Table for virtual factory.

2.16. CUBIT (17)

A CUBIT é uma superfície interactiva para interacções MT. Foi projectada com o intuito de redefinir a computação visual e distanciar o paradigma do ponteiro através

do rato. Os dedos são vistos como pontos de localização de área de contacto. Baseado nesses inputs sensoriais tenta ilustrar gráficos que se comportam como noções humanas de objectos físicos.



Figura 21 - CUBIT.



Capítulo III - Implementação

Neste capítulo será abordado alguns aspectos gerais dos protótipos em fase de conclusão, em cada secção será discutido as selecções em termos de *software*, *hardware* e técnicas de programação efectuadas, de modo a construir os respectivos protótipos.

3.1. Linguagem de programação

A linguagem de programação seleccionada para implementar este projecto foi o *Processing* (18) (ferramenta *open source*). O *Processing* é uma linguagem que tem por base o lado gráfico da linguagem *Java*, o qual simplifica e origina novas funcionalidades. As principais razões que contribuíram para a selecção do *Processing* foram as seguintes:

- ❖ A nível de programação é idêntico ao *Java*;
- ❖ É mais eficaz no processamento de imagem em tempo real;

O *Processing* relaciona conceitos de software com princípios de forma visuais, movimento e interacção. Este integra uma linguagem de programação, ambiente de desenvolvimento e metodologia de ensino num sistema unificado. O *Processing* foi criado para ensinar fundamentos de programação informática dentro de um contexto visual, servir como um software *sketchbook* e ser usado como ferramenta de produção. Estudantes, artistas, profissionais de design e investigadores utilizam-no para aprendizagem, prototipagem e produção de aplicações.

Embora a linguagem vulgarmente utilizada durante a implementação deste projecto fosse o *Processing*, outras linguagens como o Flash (19) foram igualmente analisadas. O Flash é a tecnologia mais utilizada na Web, pois permite a criação de animações vectoriais. O Flash possui uma linguagem de programação, o *ActionScript*, que se encontra sempre em evolução, tanto que já se encontra disponível o *ActionScript3*, o que faz com que cada vez mais esta deixe de ser vista como apenas uma ferramenta de recurso para design.

No desenvolvimento deste projecto, utilizou-se o programa *Adobe Fireworks* para manipulação de imagens. Este programa foi fundamental no desenvolvimento do projecto, uma vez que para todos os protótipos foi necessária a manipulação de imagens. Assim sendo, do tempo dispensado para a realização de cada protótipo, uma

parte do tempo foi dispensado para a manipulação das imagens necessárias, uma vez que a experiência na manipulação desta ferramenta era reduzida.

3.2. Hardware

Em termos de hardware, foi utilizado um computador, uma câmara de IV, um projector e uma mesa. A câmara de IV utilizada, não é mais do que uma webcam normal com duas películas de negativos de rolo fotográfico no interior. Foi seleccionado este material para a detecção, visto que é um produto económico e de fácil manutenção.

De modo a apresentar os nossos protótipos utilizou-se o projector e a mesa. O projector tem como objectivo projectar o protótipo e a mesa será a superfície onde este será projectado.

3.2.1. Montagem da mesa

Para a estrutura da mesa interactiva teve-se em conta duas estruturas de modo a obter uma interacção o mais natural possível. Ambas as estruturas são constituídas pela mesa, um projector, uma câmara de IV e um computador, sendo que a principal diferença entre elas é a posição onde está colocado o projector e a câmara.

Na primeira estrutura o projector e a câmara encontram-se posicionados no tecto, o que permite efectuar uma projecção de *top-down* como é ilustrado na figura 22.

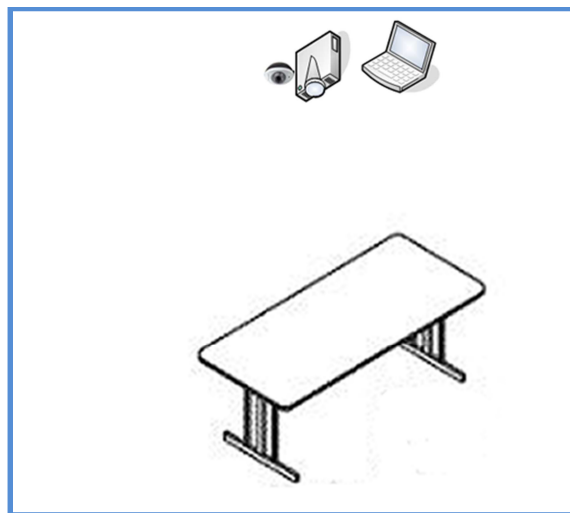


Figura 22 - Estrutura com o projector posicionado no tecto.

Esta estrutura apresenta as seguintes vantagens:

- ❖ Como está posicionado no tecto não necessita de um espelho para projectar para a mesa. E devido à distância entre o tecto e a mesa, faz com que o tamanho da projecção não fique pequeno.

Como desvantagens apresenta os seguintes aspectos:

- ❖ O utilizador ao interagir com a mesa faz com que a projecção deixe de ser projectada na mesa, mas sim na mão e no braço do utilizador.

- ❖ Uma vez que o projector e a câmara encontram-se localizados no tecto, reduz a curiosidade de o utilizador quanto à funcionalidade da mesa.

Na segunda estrutura, tanto o projector como a câmara encontram-se na parte inferior da mesa, como é ilustrado na Figura 23. Embora na imagem não ilustre, o protótipo é projectado na mesa através de um espelho, ou seja, o projector encontra-se em paralelo com a mesa, tendo um espelho que se encontra num ângulo de 45° , sendo assim projectado o protótipo na mesa.

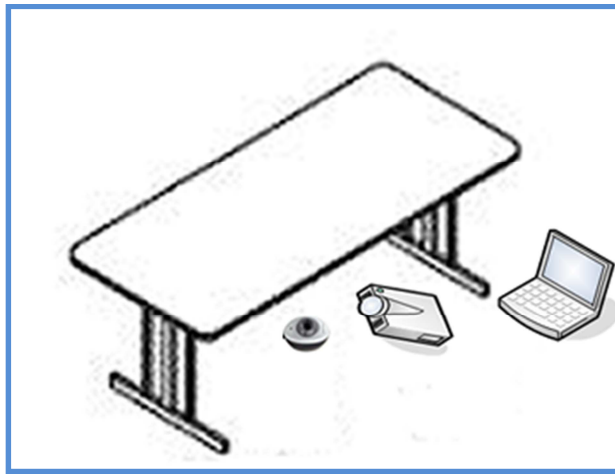


Figura 23 - Estrutura com o projector sobre a mesa.

As vantagens desta estrutura são as seguintes:

- ❖ O utilizador ao interagir com a mesa não interrompe a projecção, ou seja, como o projector se encontra debaixo da mesa o utilizador quando interage, não fica com a mão entre a projecção e a mesa.

- ❖ Como o projector e a câmara se encontram por debaixo da mesa, faz com que estes estejam escondidos do olhar do utilizador, o que contribui para que a mesa pareça mais natural.

Como desvantagens esta estrutura apresenta:

- ❖ O tamanho da projecção depende do tamanho do espelho e da distância a que se encontra o projector do espelho. Uma vez que ambos se encontram sob da mesa, para além do computador, faz com que o espaço seja reduzido, conseqüentemente a projecção será pequena.

- ❖ Uma vez que se utiliza o espelho para reflectir a imagem na mesa, este tem de estar num ângulo perfeito, cerca de 45° , caso contrário a projecção é em forma de trapézio isósceles em vez de rectangular.

Uma vez que não era possível estar constantemente a testar todos os protótipos na mesa, foi necessário desenvolver uma alternativa para simular a mesa. Para tal foi necessário utilizar o seguinte material:

- ❖ Caixa de cartão
- ❖ Mica
- ❖ Folha de papel
- ❖ Câmara IV.
- ❖ Fita-cola

A caixa de papelão tem a função de simular as paredes da mesa. Num dos lados foi efectuado um pequeno orifício de modo a passar o fio da câmara IV, sendo posteriormente a câmara IV fixada na base da caixa e no topo, de modo a simular o vidro ofusco da mesa, inseriu-se a folha dentro da mica e fixou-se a mica na parte superior da caixa, como demonstrado na Figura 24.



Figura 24 - Montagem do simulador da mesa.

3.3. Técnicas

Aqui será abordado as principais técnicas utilizadas no desenvolvimento dos protótipos, tais como detecção de movimento e sistema de partículas.

3.3.1. Detecção de movimento

A detecção de movimento é uma das técnicas mais importantes, uma vez que é utilizada em todos os protótipos. Por esta razão, na criação do código para a detecção de movimento foi necessário algum tempo e por consequência diferentes soluções.

A primeira solução criada não apresentava grande precisão, mas tendo em conta que parte dos protótipos produzidos não requeria alta precisão, era possível a sua aplicação. A função desta técnica é a detecção de todos os *globs* (pontos de movimento) sendo, posteriormente, efectuado a média desses pontos e devolvido as coordenadas X e Y.

O problema é que a detecção do movimento, nesta solução, é realizada segundo a detecção do branco e o movimento é dado pela média dos pontos onde ele detecta a diferença, que resulta no aparecimento de dois problemas:

- ❖ Nunca se consegue ter dois movimentos em simultâneo;
- ❖ Caso os pontos de movimento sejam nas duas extremidades da mesa, o movimento é detectado no centro;

A solução encontrada foi determinar a média dos pontos que se encontravam próximos uns dos outros. Assim sendo, conseguiu-se resolver o problema das extremidades. No entanto, os únicos problemas desta solução são os seguintes:

- ❖ Contínua a não ser muito precisa, embora melhor que a versão anterior;
- ❖ Para detectar movimento é necessário um movimento forte.
- ❖ Contínua a não se conseguir detectar dois movimentos em simultâneo.

Uma vez que a solução anteriormente referida não era uma solução global, ou seja, não funcionava perfeitamente para todos os protótipos, tentou-se encontrar uma solução de modo a detectar movimento de outra forma. Esta solução baseia-se na detecção de movimento por *pixels*. Ao utilizar os *pixels* produziu-se duas versões, uma das versões compara as últimas duas *frames* captadas pela Webcam; e a outra compara à última *frame* captada com uma *frame* inicialmente seleccionada para servir de comparação. Tanto uma como outra versão agarra nas duas *frames* e faz a comparação *pixel a pixel*. O resultado da comparação é colocado num *array* do tamanho da *frame* capturada, e caso os *pixels* comparados sejam iguais, no *array* é colocado que o *pixel*

em questão é preto. Caso contrário, quer dizer que houve movimento e o *pixel* é colocado a vermelho (Figura 25). Depois é só percorrer este *array* para se ter conhecimento onde houve movimento.

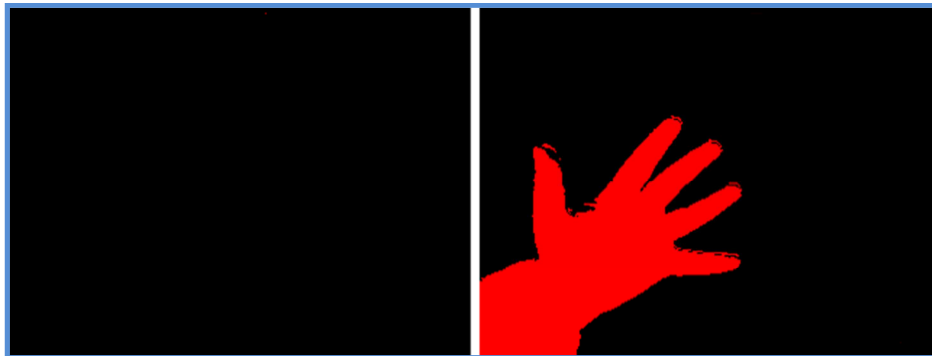


Figura 25 – Detecção de movimento usando pixels.

Todas as soluções anteriormente abordadas foram programadas em *Processing*, mas uma vez que estas detecções não conseguiam diferenciar os dedos do resto da mão, ou seja, detectava o movimento da mão inteira, tentou-se descobrir outras soluções para que ao detectar movimento se conseguisse detectar só os dedos.

Uma das soluções encontradas foi a biblioteca *Touchlib* (20), sendo esta utilizada com o intuito de produzir superfícies MT. A biblioteca *Touchlib* permite monitorizar os *blobs* de luz IV e envia para os programas estes eventos MT, tais como “dedo para baixo” e “movimento do dedo”. Esta biblioteca inclui uma aplicação de configuração e funciona com a maior parte dos dispositivos de captura de vídeo e Webcams.

Na versão actual, *Touchlib* pode transmitir eventos através do protocolo TUIO (que utiliza o OSC). Isto torna o *Touchlib* compatível com outras aplicações que suportam este protocolo, nomeadamente Flash, VVVV, *Processing* e *PureData*.

De modo a ser possível configurar os parâmetros da câmara, com intuito a que esta capture o movimento dos dedos é necessário executar o ficheiro *Configuration.bat* na pasta da biblioteca. Após esta acção aparecerá uma janela para definir as propriedades desejadas para a câmara, como ilustrado na Figura 26. Nota: Um Frame Rate mais elevado funcionará melhor que uma alta resolução (Output Size).

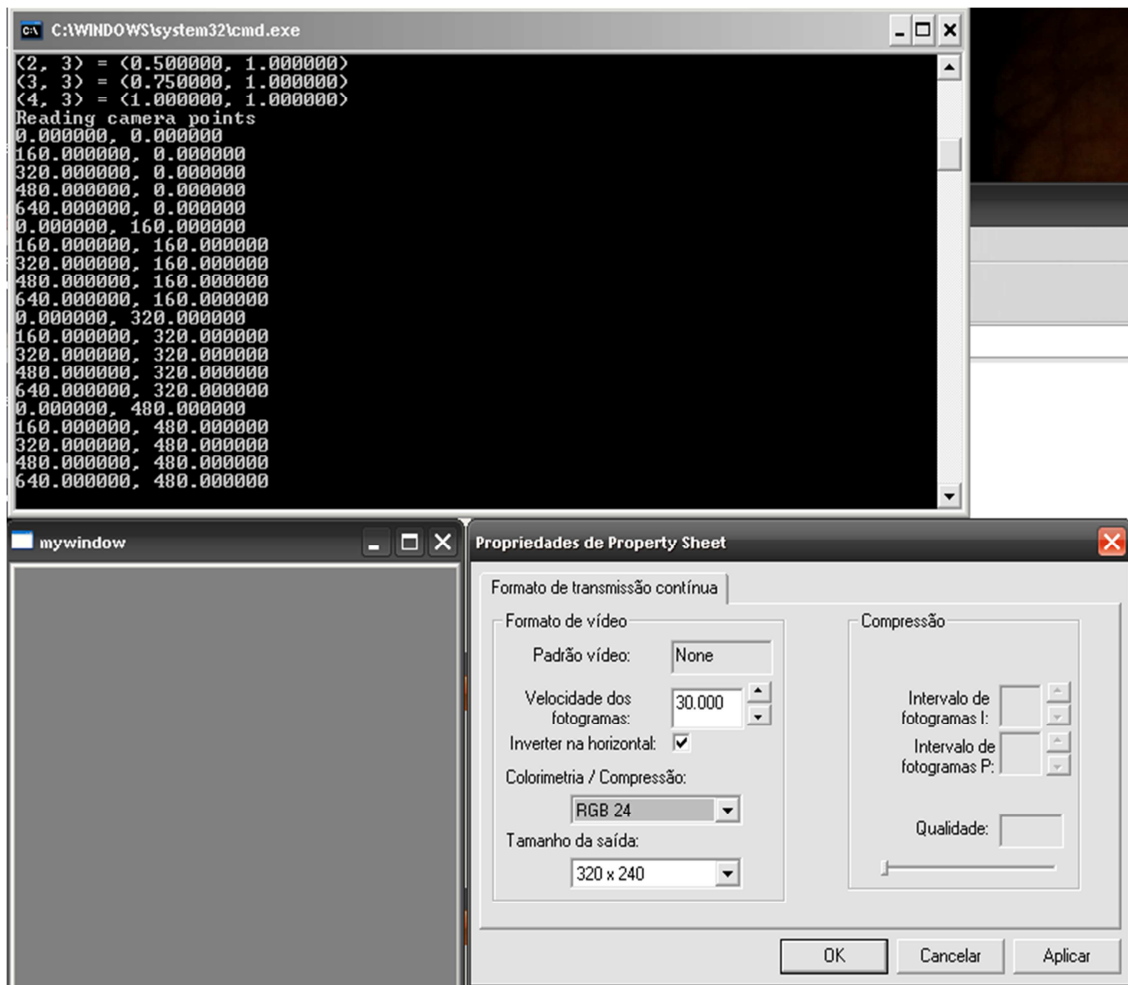


Figura 26 - Configuração das propriedades desejadas na câmara.

Múltiplas janelas irão aparecer com as imagens captadas pela câmara, como demonstrado na Figura 27. Depois devesse ajustar os “Sliders” até que a janela **rectify** apenas tenha os *blobs* provenientes dos dedos. Quando já tiverem sido efectuadas todas as configurações necessárias, basta premir “ESC” para sair da aplicação.

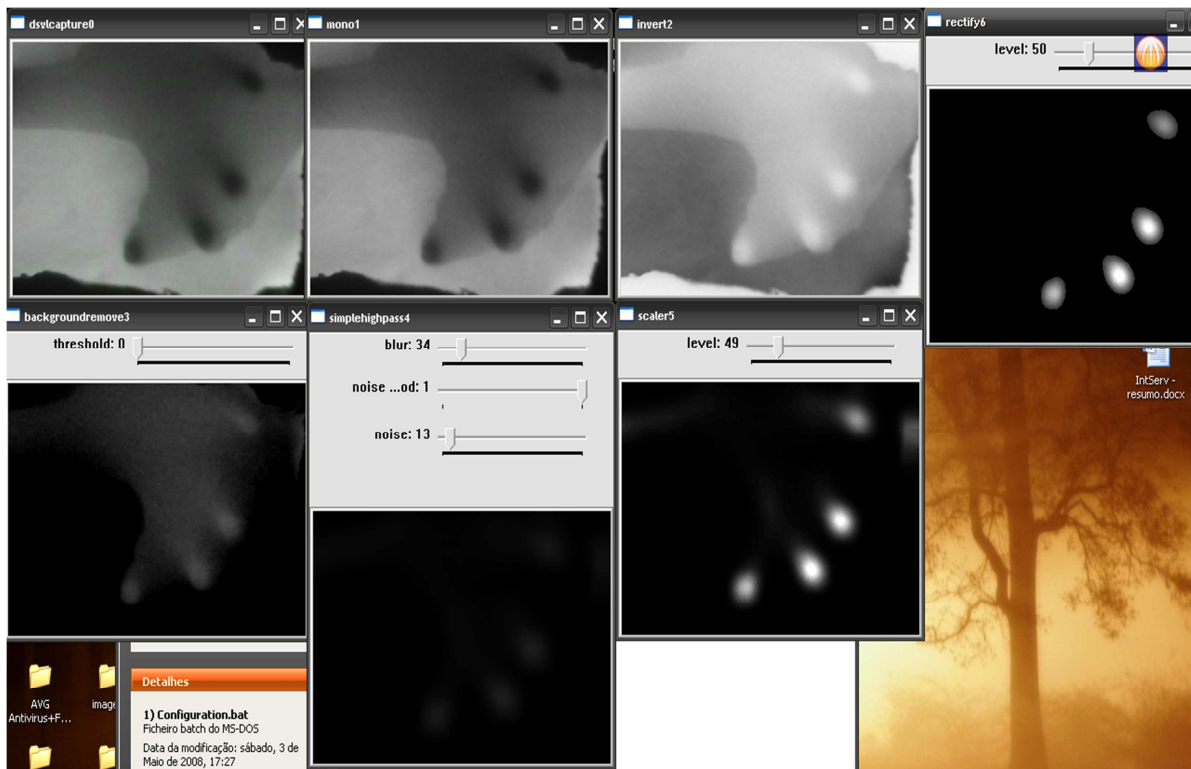


Figura 27- Touchlib Configapp.

De seguida será explicado o objectivo de cada janela:

- ❖ Dsvlcapture(0): é a janela onde é ilustrada a imagem captada pela câmara.
- ❖ Mono(1): modifica a imagem captada pela câmara numa imagem em escala de cinzento.
- ❖ Invert(2): inverte as cores da imagem da janela Mono1.
- ❖ Backgroundremove(3): subtrai o background da corrente imagem.
- ❖ Simplehighpass(4): só deixa que os pontos mais brilhantes passem.
- ❖ Scaler(5): amplifica o output do filtro anterior, ou seja, pontos brilhantes ficam mais intensos.
- ❖ Rectify(6): é a imagem final utilizada para a detecção de movimento.

3.3.2. Imagem de fundo

Esta técnica tem como finalidade ter um *array* de imagens. Esta é constituída por duas versões, sendo que a primeira apresenta cada uma das imagens durante 10 segundos e a segunda muda de imagem quando deixa de detectar movimento. Esta técnica suporta um número ilimitado de imagens. Quando chega à última imagem repete o ciclo. Esta técnica não é apresentada sozinha num protótipo, mas pode ser usada para acrescentar mais efeito ao protótipo. Contém igualmente a possibilidade de efectuar

com que a imagem apareça já do tamanho da janela ou ir aumentado até ao tamanho da janela (Figura 28).



Figura 28 – As duas versões da técnica imagem de fundo.

3.3.3. Sistema de partículas

Para a implementação desta técnica, utilizou-se a biblioteca *traer.physics* (21). Esta biblioteca tem por objectivo a criação de um sistema de partículas que está encarregue de tudo, como por exemplo: definição forças entre as partículas e o responsável pelo avanço da simulação. Ao criar o sistema de partículas pode-se definir forças como gravidade, *drag*, criação de partículas e criação de atracção (repulsão) entre as partículas. As partículas podem representar objectos ou imagens, cujas imagens podem apresentar sobre a extensão .png, .jpg e .gif. A atracção ou repulsão que actuam entre duas partículas e ambas aproximam-se ou afastam-se por aplicação de uma força em cada partícula.

Ao utilizar esta biblioteca criou-se três sistemas de partículas consoante o protótipo pretendido, os quais procederei à explicação de cada uma das versões:

O primeiro sistema de partículas com um número definido de partículas, após a definição que as partículas se repeliriam de uma partícula fixa, sendo esta partícula a posição onde era detectado o movimento. A Figura 29 exemplifica esta variação onde o ponto verde é a partícula fixa, ou seja, onde o movimento é detectado, consequentemente, as outras partículas repelam-se desse ponto.

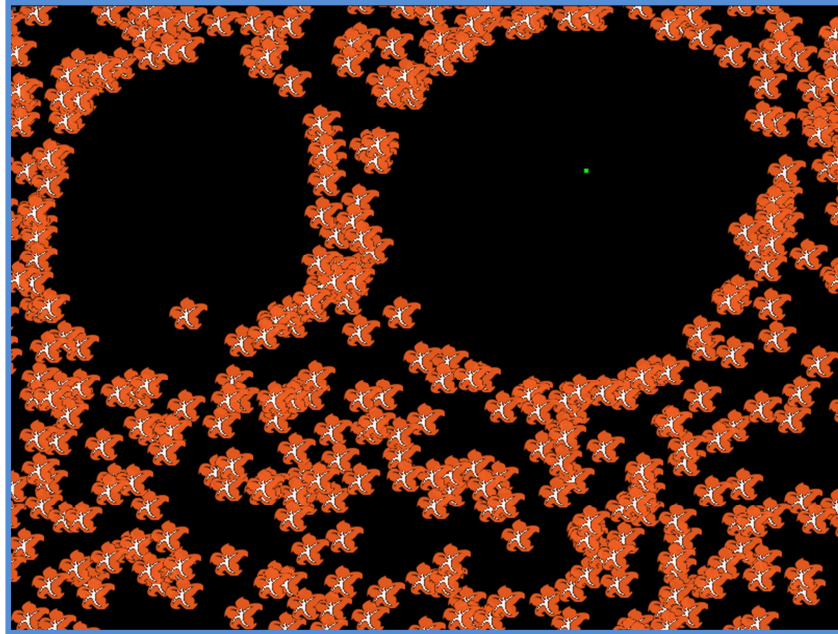


Figura 29 – Primeiro sistema de partículas.

O segundo sistema de partículas tem por objectivo colocar uma ou mais partículas onde é detectado movimento. Após um determinado tempo, o qual pode ser definido pelo programador, elas são “mortas”. Esta variação serve principalmente para realizar o efeito “rasto” consoante o movimento do utilizador. A Figura 30 demonstra esta versão do sistema de partículas.

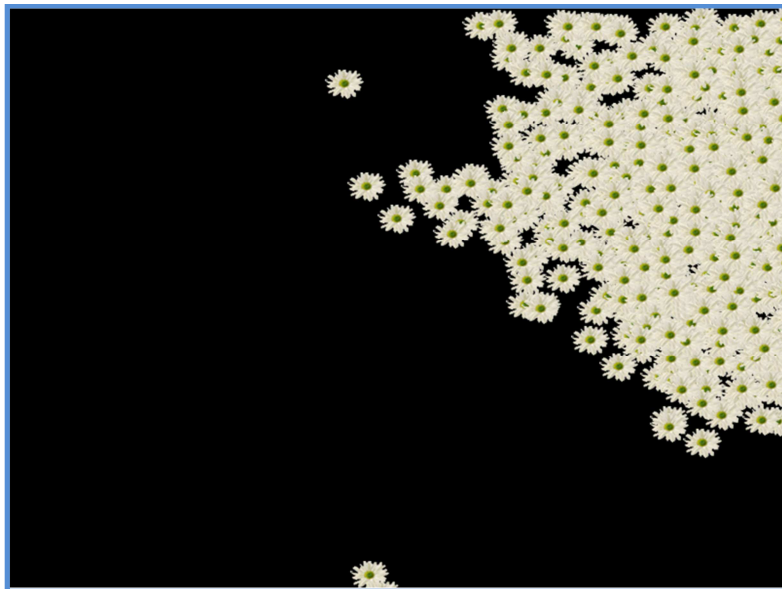


Figura 30 – Segundo sistema de partículas.

O terceiro sistema de partículas consiste em criar um sistema de partículas com gravidade. Este tem como finalidade criar partículas numa posição *random* e deixar que a gravidade faça o resto como ilustrado na Figura 31. Esta variação tem como função

incrementar mais “beleza” aos protótipos, porém, não é o único efeito. As partículas criadas podem ou não ser sensíveis ao movimento.



Figura 31 - Terceiro sistema de partículas.

3.3.4. Calibração da Câmara

A criação desta técnica teve como objectivo a possível calibração da câmara, ou seja, oferece a opção do utilizador seleccionar a zona em que ele deseja que a câmara capture movimento. Para tal, é fornecido ao utilizador a imagem capturada pela câmara e a opção do utilizador clicar em dois pontos extremos, ou seja, ele clica no primeiro ponto que deverá ser no canto superior esquerdo e o segundo ponto no canto inferior direito. Será efectuado um rectângulo que liga estes dois pontos e este rectângulo será a zona em que a câmara captará, como ilustrado na Figura 32. Como resultado é devolvido os valores dos pontos de modo a que seja possível inserir estes valores nas técnicas de detecção de movimento.

Esta técnica é utilizada antes de executar qualquer protótipo, de modo a obter os pontos em que será capturado o movimento.

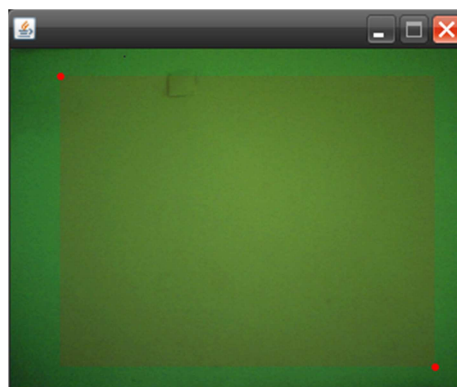


Figura 32 - Janela de calibração da câmara.

3.4. Dificuldades no desenvolvimento dos protótipos

No desenvolvimento dos protótipos as dificuldades globais encontradas foram as seguintes:

i. Arranjar forma de ter a aplicação em *full screen*. Para tal, tentou-se usar a biblioteca *OpenGL* (22), que se encontra incorporada no *Processing*, mas o que esta biblioteca simplesmente faz é criar uma janela do tamanho do ecrã. Caso se criasse o executável do protótipo e o corrêssemos então o *OpenGL* realizaria o *fullscreen* de forma correcta. No entanto, descobriu-se que se em vez de se executar o código da aplicação, executasse o código em modo de apresentação a aplicação ficava em *fullscreen*. Só que usando esta biblioteca, num sistema de partículas, cujo fundo é uma imagem este sobrepõe a imagem as “folhas”, mas caso o fundo fosse uma cor em vez de uma imagem, funcionava perfeitamente.

Encontrou-se outra biblioteca para o *fullscreen* (23) que tem como objectivo redimensionar o tamanho da janela para o tamanho do ecrã. O problema desta biblioteca é que entra em conflito com a classe para a reprodução de vídeo.

ii. Outra dificuldade encontrada foi no sistema de partículas. Mais concretamente no movimento das partículas. Ao usar a função *Attraction* da biblioteca *Traer physics*, quando era detectado o movimento as partículas se afastavam como desejado, porém na ausência de movimento elas simplesmente apareciam na posição inicial, em vez de voltarem ao lugar inicial de maneira fluida. Depois tentou-se usar a função *Spring* da mesma biblioteca, resolvendo assim o problema das folhas ao voltarem ao lugar na ausência de movimento, contudo criou o problema ao afastar as folhas na presença de movimento. Enquanto, o *Attraction* só cria atracção ou repulsão das partículas que se encontram mais perto do movimento e conforme as partículas estão mais afastadas cria menos repulsão, o *Spring* cria atracção ou repulsão a todas as partículas com a mesma intensidade quer estas estejam perto ou longe do movimento. Tentou-se variar os valores tanto do *Spring* como do *Attraction* de maneira a tentar resolver os problemas acima mencionados, contudo não se observou nenhuma evolução.

A solução encontrada foi usar a função *Attraction* para afastar as folhas e programar o seu retorno para o lugar inicial de maneira fluida.

iii. A utilização da câmara para detecção de movimento, no qual utilizou-se a biblioteca *JMyron* (24) foi outra das dificuldades encontradas. Ao colocar a câmara a

captar numa resolução superior a 640x480 a câmara deixava de funcionar correctamente e a captação aparecia em mosaicos na janela da aplicação. Descobriu-se depois que a câmara tem resolução máxima de 640x480, porém, mesmo nesta resolução, esta não funcionava perfeitamente. A melhor resolução para a captação era a de 320x240. A solução para este problema em concreto foi captar a 320x240 e depois multiplicar os valores recebidos da coordenada X e Y por 2.5, de modo a ter conhecimento onde exactamente era o movimento no tamanho da janela da aplicação com dimensões de 800x600 *pixels*.

Um outro problema da biblioteca *JMyron* é que durante a execução de um protótipo tenha-se de desligar e voltar a ligar a câmara, à terceira vez ocorria um erro na execução do protótipo. Para tentar resolver este problema tentou-se encontrar outra biblioteca para a utilização da câmara. A biblioteca encontrada foi a *libcv*, mas esta biblioteca não era fiável uma vez que nem sempre que executávamos o protótipo ela capturava as imagens da câmara. A solução final encontrada foi fazer com que a câmara nunca se desligasse nos quais se alterava os parâmetros da câmara sempre que necessário.

iv. Na leitura de ficheiros .txt. Pois as funções da linguagem de programação *Processing* para leitura de ficheiros usam *UTF-8 encoding*, o que quer dizer que a letra “ç” e letras com “˘ ~ ^”, não aparecem.

v. Outro dos problemas foi também reproduzir um ficheiro de vídeo usando o *Quicktime*, pois é a biblioteca de vídeo que vem incorporada no *Processing*, o que implica que os ficheiros teriam de ter a extensão .mov. O problema desta biblioteca é que no sistema operativo (SO) Windows reproduzia o vídeo, mas não o som. Tentou-se correr o protótipo no SO Mac e embora desse som, este não corria ao mesmo tempo que o vídeo, e caso se mudasse de vídeo daria o som do vídeo anterior e do vídeo actual. Descobriu-se depois uma classe denominada de *FasterMovie.pde* que embora usasse a mesma biblioteca resolvia o problema anteriormente descrito.

Capítulo IV - Protótipos Desenvolvidos

Depois de pesquisar alguns exemplos em que eram usados Webcams, iniciou-se a aprendizagem nos exemplos de código para *Processing* e começou-se a criar o próprio conjunto de protótipos e cada um deles foi o resultado de várias evoluções. Tentou-se fazer os protótipos de maneira a que a utilização fosse simples e interessante de modo a cativar os utilizadores.

Nesta secção será apresentado e descrito cada um dos protótipos desenvolvidos durante este projecto.

4.1. Body Mind Madeira

O protótipo Body Mind Madeira (Figura 33) utiliza um sistema de partículas, e que de acordo com o movimento do utilizador, as folhas afastam-se, ilustrando como *background* uma imagem da Ilha da Madeira. A imagem de *background* vai sempre modificando de modo a manter o utilizador cativado. Este protótipo possibilita que a existência de um ou dois tipos de folha, conforme o desejado.



Figura 33 - Body Mind Madeira com uma ou duas folhas.

Este protótipo tem várias áreas de aplicação, tais como promoção de um local turístico, como é o caso, marketing e para fins educacionais. Em termos de marketing um caso hipotético seria uma empresa de automóveis ter como objectivo a apresentação de um novo carro, então em termos de publicidade criava-se uma mesa onde as folhas neste caso seriam o símbolo da empresa, que ao detectar movimento afastavam-se mostrando como *background* diferentes imagens do respectivo carro. Em termos educacionais poderíamos

usar o tema da reciclagem e neste caso teríamos uma mesa onde as folhas seriam o símbolo da reciclagem e o *background* imagens com slogans a incentivar a reciclagem, como por exemplo: papel é no contentor azul e assim sucessivamente.

4.1.1. Processo de desenvolvimento

Este protótipo começou com o objectivo de criar um sistema de partículas que se afastariam conforme fosse detectado movimento. Para tal, utilizou-se as funções contidas na biblioteca *traer.physics*, com o intuito de criar a repulsão entre as partículas criadas e o movimento. De modo a se poder criar a repulsão entre as partículas e o movimento teve-se que criar uma partícula que era fixa e na qual a posição desta partícula era a posição onde era detectado movimento. Como, anteriormente já referido (Secção 3.4, pág. 27), o retorno das partículas não estava perfeito e mesmo usando as funções disponíveis da biblioteca não se resolveu o problema, a solução foi programar o retorno das partículas de maneira fluida.

A seguinte etapa foi introduzir uma imagem como fundo. Ao introduzir a imagem como *background* teria de se assegurar que a imagem era do tamanho da janela ou então daria erro. Deste modo decidiu-se que o *background* teria como cor de fundo preto e a imagem iria aumentando até chegar ao tamanho da janela da aplicação. O modo como se passava de imagem para imagem tinha duas versões, anteriormente descritas (Secção 3.3., pág. 20).

Como objectivo final para este protótipo era que sempre que existisse movimento reproduzisse-se um som, de modo a que o utilizador para além de sensação visual tivesse também uma sensação auditiva.

Este protótipo utiliza as seguintes técnicas referidas anteriormente (Secção 3.3, pág.20): detecção de movimento, imagem de fundo e sistema de partículas.

A Figura 34 ilustra o diagrama de classes deste protótipo. Em seguida irá ser apresentado uma pequena descrição e os respectivos objectivos de cada classe. A classe *Gere* é a classe principal, ou seja, é a classe que coordena tudo e onde é definida todas as variáveis gerais ao programa, sendo igualmente responsável pelo som. Tem na sua funcionalidade o método *draw()* que é responsável pela execução do efeito e de fornecer informação de qual a imagem de fundo a apresentar. A classe *BackgroundImage* armazena todas as imagens destinadas para o fundo num *array* e tem como objectivo apresentar uma imagem de fundo de cada vez, sendo que a posição do *array* da imagem a apresentar é fornecido pela classe *Gere*.

A classe *CameraMovement* tem como função capturar a imagem da câmara, efectuando depois a comparação entre a imagem captada (guardada no *array* *img*) e uma imagem inicialmente seleccionada (guardada no *array* *inicial*). A classe *Coordenadas* tem como objectivo a criação de um objecto composto pelas coordenadas X e Y, onde é detectado o movimento.

A classe *InicializarParticulas* tem como finalidade criação do sistema de partículas, criação das partículas e o estabelecimento da repulsão entre as partículas e a posição onde é detectado movimento. Esta classe tem a funcionalidade de associar uma partícula a uma imagem, e fazer o retorno das partículas na ausência de movimento.

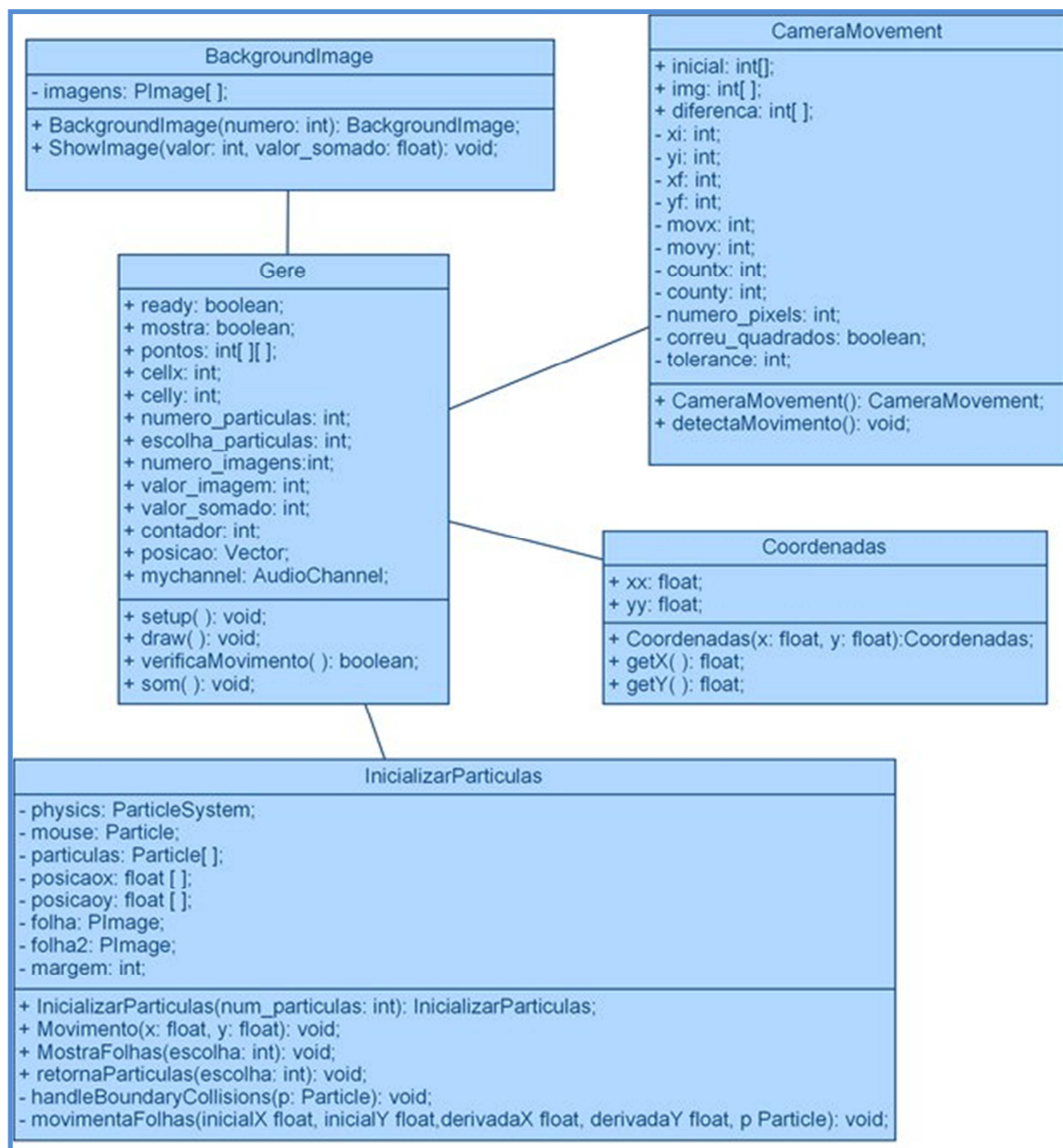


Figura 34 - Diagrama de classes do Body Mind Madeira.

4.2. Wow!Bookstore

O protótipo Wow!Bookstore tem como objectivo a divulgação de livros numa livraria. Em concreto este protótipo permite a exibição de livros e informação referente a cada um deles em simultâneo. A informação disponível é a seguinte nome do autor, editora, sinopse e o preço do livro. Ao detectar movimento num dos livros, este roda a imagem no eixo dos Y, procedendo a exibição de outro livro, como se pode observa na Figura 35.



Figura 35 – Wow!Bookstore.

4.2.1. Processo de desenvolvimento

Este protótipo iniciou com o objectivo de rodar uma imagem no eixo dos Y. Depois de se conseguir o pretendido, o objectivo foi passar a ter quatro imagens, de modo a preencher o tamanho da janela, podendo estas rodar cada uma independentemente.

A etapa seguinte foi que ao chegar ao meio da rotação do livro substitui esse livro por outro que não esteja no momento em exibição. Com o intuito deste protótipo ficar o mais geral possível foi elaborado de forma a ser configurável quantos livros se queriam em simultâneo em exibição. Por esta razão basta inserir o número de imagens que deseja no eixo dos X e o número de imagens que se deseja no eixo dos Y. E o número de imagens total na janela da aplicação será a igual ao número de imagens X vezes o número de imagens Y. Uma vez que os livros em exibição não se repetem, isto exige que o número total de livros seja superior ao número total de livros em exibição.

Como o número de livros em exibição é configurável, quer dizer que o tamanho de exibição dele pode ser variado, por esta razão foi necessária a criação de uma função que redimensionasse os livros para o tamanho correcto.

Neste protótipo a única técnica utilizada foi a detecção de movimento.

A Figura 36 apresenta um diagrama de classes deste protótipo. A classe *Livraria* é a classe principal, sendo responsável pela exibição do efeito. Além disto, é responsável pela definição do tamanho da janela entre outras variáveis globais. A classe *CameraMovement* tem os mesmos objectivos da classe *CameraMovement* referido no protótipo anterior (Secção 4.1.1 pág. 30), cuja diferença entre estas classes é que em vez de comparar a imagem capturada com uma inicialmente seleccionada, utiliza o método `differenceImage()`, da biblioteca *JMyron*, que retorna a diferença entre os últimos dois frames captados pela câmara.

A classe *ViraLivros* tem como principal finalidade seleccionar quais as imagens para a exibição, a rotação das imagens e o redimensionamento das imagens para o tamanho desejado, de modo a que não exista livros repetidos na janela.

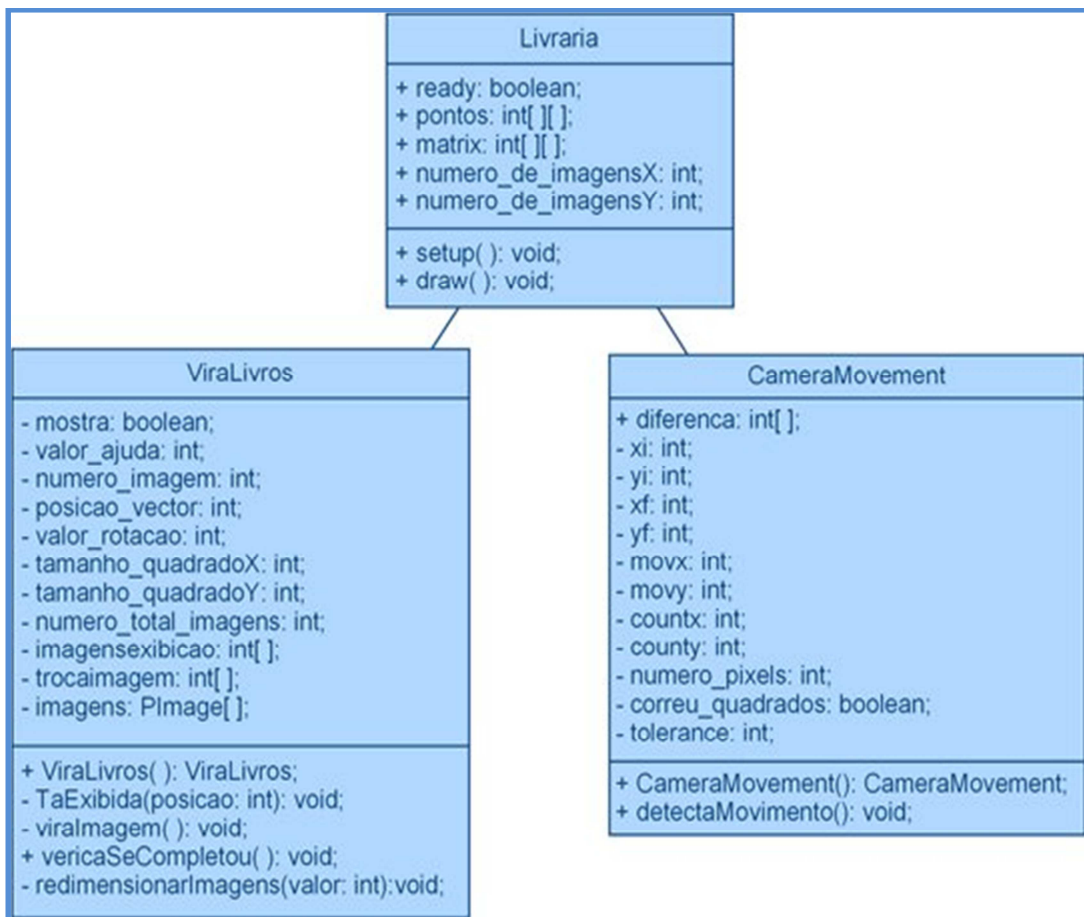


Figura 36 - Diagrama de classes Wow!Bookstore.

4.3. Wow!Tiles

O Wow!Tiles (Figura 37) tem por objectivo a promoção de algum local turístico. Assim sendo como fundo tem imagens referentes ao local turístico, cujas imagens encontram-se tapadas com azulejos. Este protótipo pode ter um ou dois tipos de azulejos, conforme o pretendido. Ao ser detectado movimento sobre os azulejos estes movimentam-

se deixando assim a imagem a descoberto. O objectivo principal deste protótipo é retirar todos os azulejos de modo a ver a imagem de fundo. Depois de certo tempo os azulejos voltam a reaparecer e a imagem de fundo é mudada.



Figura 37 - Wow!Tiles.

4.3.1. Processo de desenvolvimento

Com este protótipo o primeiro passo foi arranjar uma maneira de colocar os azulejos de modo a preencher a janela. O tamanho do azulejo não necessita de ser sempre o mesmo tamanho, mas caso se use dois tipos de azulejos convém que sejam do mesmo tamanho, pois o programa só faz o cálculo a partir do tamanho do primeiro azulejo.

Inicialmente ao ser detectado movimento sobre os azulejos estes fugiam na diagonal, mas para dar um efeito mais expressivo, fez-se que ao fugirem, embora na diagonal, rodassem.

As técnicas utilizadas neste protótipo foram as seguintes: imagem de fundo e detecção de movimento.

Na Figura 38 está ilustrado o diagrama de classes deste protótipo. A classe *azulejos* é classe principal e por consequência a responsável pela representação do efeito. A classe *CameraMovement* e a classe *BakgroundImage* têm a mesma funcionalidade que a referida na Secção 4.2.1 (pág. 32) e Secção 4.1.1 (pág. 30), respectivamente. No entanto, a classe *BakgroundImage*, difere da anteriormente referida, devido ao método `ShowImage(valor: int)`, pois uma vez que é mudada a imagem de fundo, esta é sobreposta pelos azulejos, logo não há necessidade de fazer com que a imagem vá aumentando até chegar ao tamanho da janela.

A classe *RetiraAzulejos* tem como objectivo a gestão dos azulejos, ou seja, faz com que os azulejos se movam, verifique se todos os azulejos já foram tocados e se já se

encontram “fora” da janela da aplicação. Caso seja este o caso, os azulejos são recolocados e a imagem de fundo alterada. Esta classe tem ainda um método que define desde o início a direcção que os azulejos irão seguir.

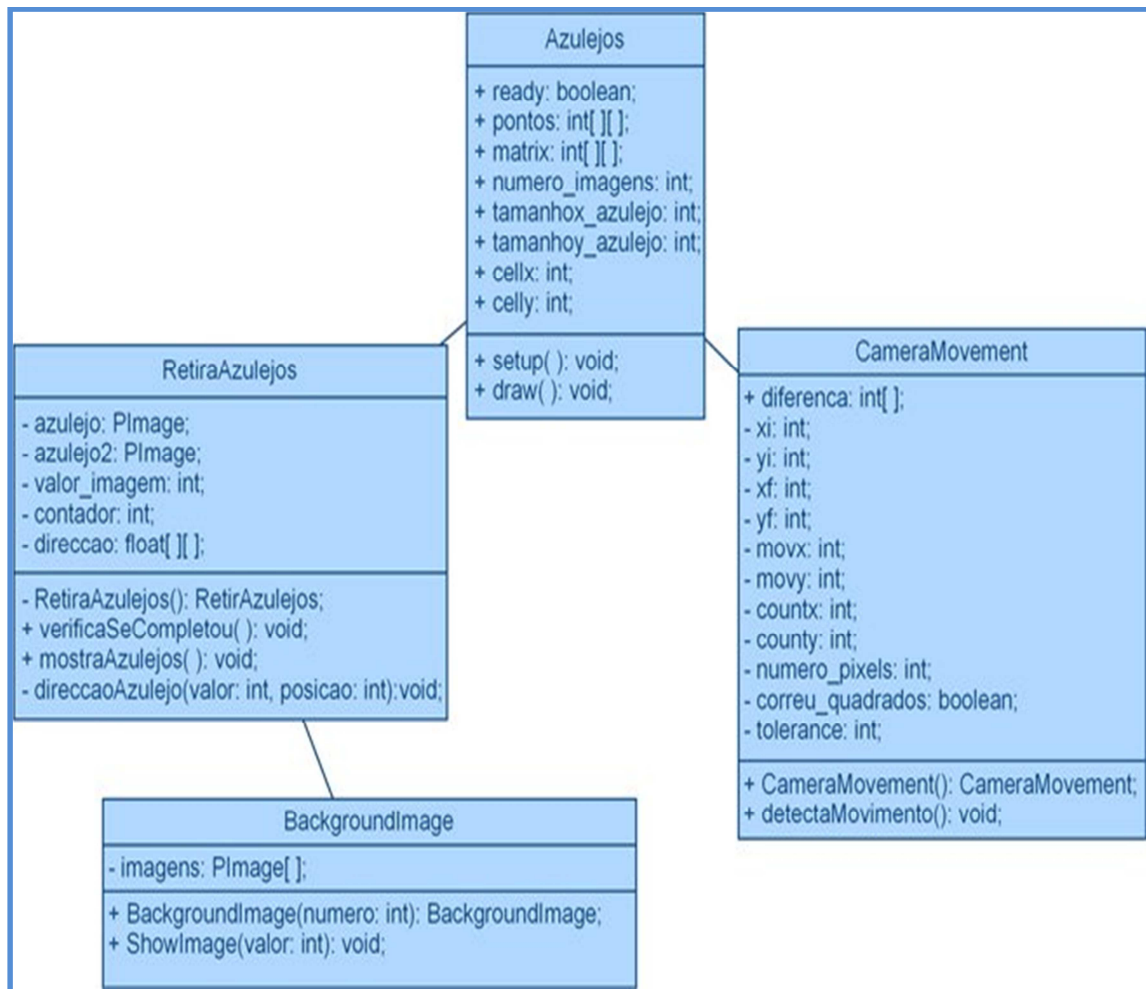


Figura 38 - Diagrama de classes do Wow!Tiles.

4.4. Wow!Cinema

Este protótipo foi efectuado a pensar nas pessoas que vão ao cinema mas ainda não têm uma ideia definida de qual o filme que vão assistir. Assim sendo este foi projectado com o intuito de ilustrar os *trailers* dos filmes em exibição, no qual contem informação sobre o respectivo filme, nomeadamente nome do filme, actores, hora de exibição, género de filme e sala, como ilustrado na Figura 39.

O utilizador pode passar de filme para filme usando os botões e de modo a que o utilizador tenha a percepção que tocou, o botão ao ser clicado muda de cor.



Figura 39 - Wow!Cinema.

4.4.1. Processo de desenvolvimento

Este protótipo teve como objectivo principal conseguir reproduzir um ficheiro .mov. Como anteriormente referido (Secção 3.4, pág.27), no SO Windows reproduzia o filme, mas não o som e no SO Mac embora desse o som este não acompanhava o vídeo e o protótipo consumia muitos recursos do computador. A solução apareceu quando descobriu-se a classe *FasterMovie.pde*, uma vez que esta classe não só resolveu o problema do som, como tornou o protótipo mais “leve”.

A etapa seguinte consistia em ler a partir de um ficheiro .txt a informação referente ao filme. A estrutura do ficheiro é definida de forma que para o programa a primeira linha é o nome do filme, a segunda linha os actores, sucessivamente. De modo a se saber qual o ficheiro de texto para um determinado filme, este teria de ter o mesmo nome que o filme, como por exemplo: caso o filme se denominasse teste.mov o ficheiro de texto seria teste.txt.

Procedeu-se em seguida à implementação dos botões de maneira a que fosse possível passar de filme para filme. Só para adornar o protótipo incluiu-se um sistema de partículas que tem por objectivo deixar “cair” partículas em posições *random* com o símbolo do cinema, neste caso o símbolo é o logótipo da Castello Lopes.

Caso o protótipo fosse adquirido por alguma empresa e caso esta quisesse mudar ou mesmo inserir mais um filme não necessitava de nos contactar, uma vez que o programa foi programado, de modo ao programa aceder a uma pasta específica que contém todos os

filmes disponíveis. Assim sendo, o cliente só necessita de inserir todos os filmes e seus ficheiros de texto nessa pasta e o programa realiza o resto.

As técnicas utilizadas neste protótipo foram: detecção de movimento e sistema de partículas.

A Figura 40 demonstra um diagrama de classes deste protótipo. A classe *Cinema* é a classe responsável não só pelo funcionamento do protótipo, mas também pela reprodução dos filmes e a passagem de filme para filme. A classes *FasterMovie* é a classe responsável pela manipulação dos vídeos, a qual possui os métodos como `start()`, `stop()`, entre outros. A classe *CameraMovement* tem o mesmo objectivo que a classe descrita na Secção 4.1.1. (pág. 30). A classe *SistemaParticulas* tem por finalidade a criação de um sistema de partículas com gravidade, sendo igualmente responsável pela adição ou remoção de partículas do sistema de partículas.

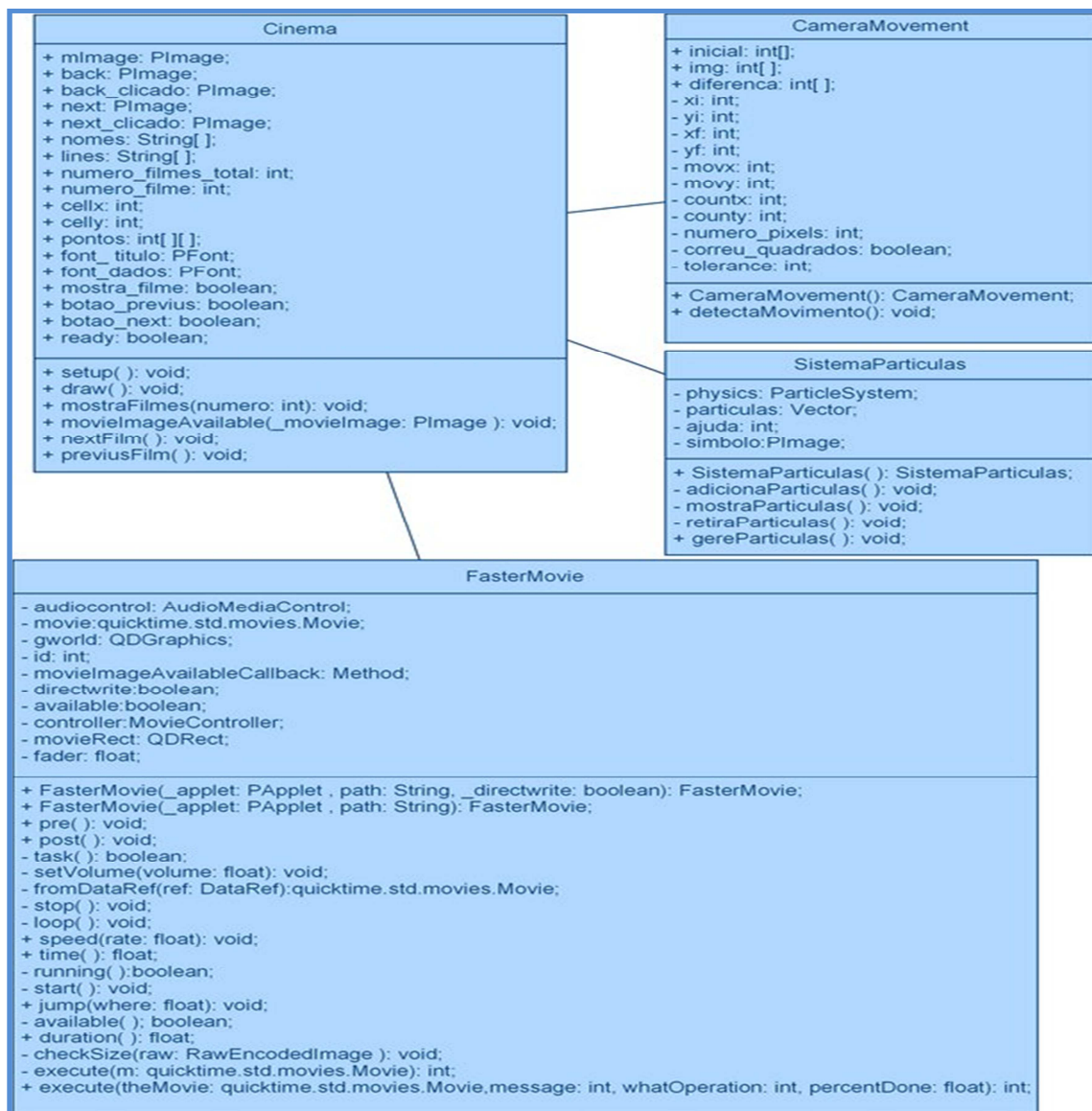


Figura 40 - Diagrama de classes do Wow!Cinema.

4.5. Wow!Track

Este protótipo tem várias áreas de aplicações, nomeadamente publicidade, fins educacionais ou até mesmo simplesmente para o entretenimento, uma vez que este protótipo tem como principal função criar um rasto de partículas consoante onde é detectado movimento. Estas partículas podem ser uma única imagem ou diferentes imagens, como por exemplo diversos tipos de flores. Como fundo pode ter uma sequência de imagens, um filme ou simplesmente um fundo a preto.



Figura 41 - Wow!Track.

4.5.1. Processo de desenvolvimento

Com o objectivo de criar o efeito rasto após a detecção do movimento, foi usado *pixels* para a detecção. Dividiu-se a imagem captada pela câmara numa grelha de 20x20, e correu-se cada quadrado da grelha, sendo depois verificado se havia ou não movimento. Caso se verificasse, inseria-se uma partícula nesse quadrado. De modo a que as partículas aparecessem, sem que fosse em forma de grelha, decidiu-se que a posição da partícula seria uma posição no interior do quadrado da grelha e não na posição inicial do quadrado.

Depois o objectivo foi fazer o fundo, desta maneira podia-se ter uma imagem estática ou então um sequência de imagens que passariam de uma para outra a cada 10 segundos.

Como objectivo final foi fazer com que as partículas fossem imagens diferentes, como na Figura 41. Para tal foi produzida uma classe que guardava toda a informação da partícula, como por exemplo a imagem que essa partícula representava.

As técnicas utilizadas neste protótipo foram detecção de movimento, imagem de fundo e sistema e partículas.

A Figura 42 ilustra o diagrama de classes do protótipo Wow!Track. A classe *Rasto* é responsável pelo funcionamento do protótipo e definição das variáveis globais, tanto a

classe *BackgroundImage* como a classe *CameraMovement* já foram anteriormente definidas na Secção 4.1.1. (pág. 30). A classe *SistemaParticulas* tem como objectivo a criação de um sistema de partículas sem gravidade, onde são inseridas partículas consoante o movimento detectado, sendo esta classe responsável pela adição e remoção de partículas. A classe *InformacaoParticula* tem como objectivo criar um objecto com a informação referente à partícula, nomeadamente a própria partícula, o número da imagem que a partícula representa e o nome do ficheiro da imagem.

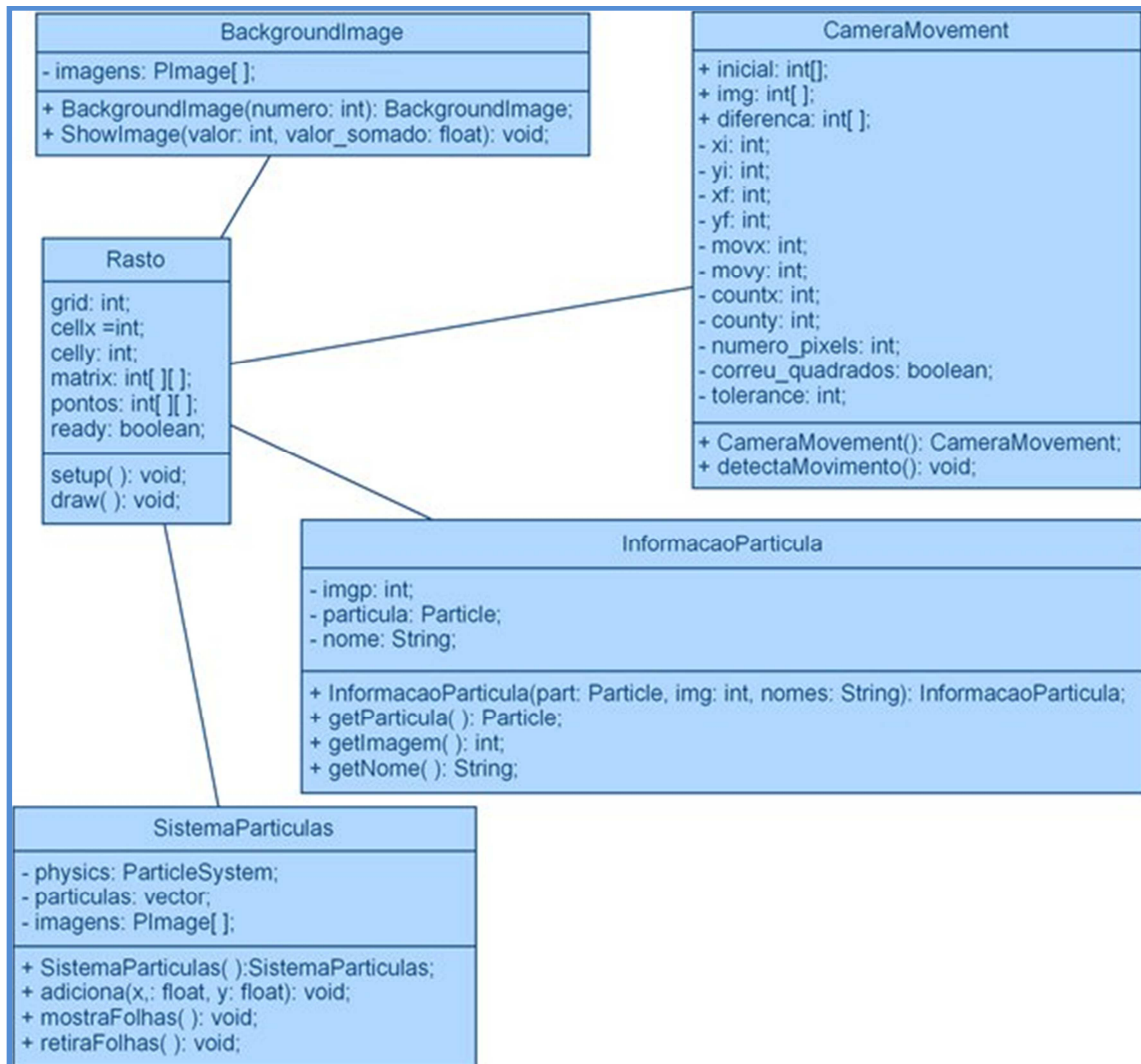


Figura 42 - Diagrama de classes do Wow!Track.

4.6. Wow!Gallery

O objectivo principal do Wow!Gallery (Figura 43) é a promoção de um local turístico. Este consiste basicamente numa imagem em tamanho grande com o nome desse local no topo da respectiva imagem. Na parte inferior da janela contém em miniatura, a imagem que antecede e sucede a respectiva imagem apresentada, assim como os botões que permitem a passagem de imagem para imagem.

Este protótipo tem o seu interesse pois pode auxiliar os visitantes a escolher quais os lugares que pretendem visitar.



Figura 43 - Wow!Gallery.

4.6.1. Processo de desenvolvimento

Este protótipo começou com o objectivo de colocar uma imagem no centro do ecrã de tamanho fixo e inserir o nome da localidade referente à imagem.

Com intuito de passar de imagem para imagem foi inserido dois botões de modo a facilitar o utilizador. Entre os botões encontram-se três miniaturas de imagens, sendo estas a imagem prévia e a próxima referente à imagem em exibição, sendo que a imagem em exibição encontra-se rodeada por um quadrado amarelo, de modo a servir de referência. Por fim, inseriu-se um sistema de partículas para ornamentar o protótipo, ou seja, dar um pouco de movimento ao protótipo.

As técnicas utilizadas neste protótipo foram as seguintes detecção de movimento e sistema de partículas.

A Figura 44 ilustra o diagrama de classes deste protótipo. As classes *CameraMovement* e *SistemaParticulas* encontram-se descritas na Secção 4.4.1. (pág. 36). A classe *Galeria* é responsável pelo funcionamento do protótipo, além da gestão das imagens, isto é, a exibição e redimensionamento das imagens, entre outras.

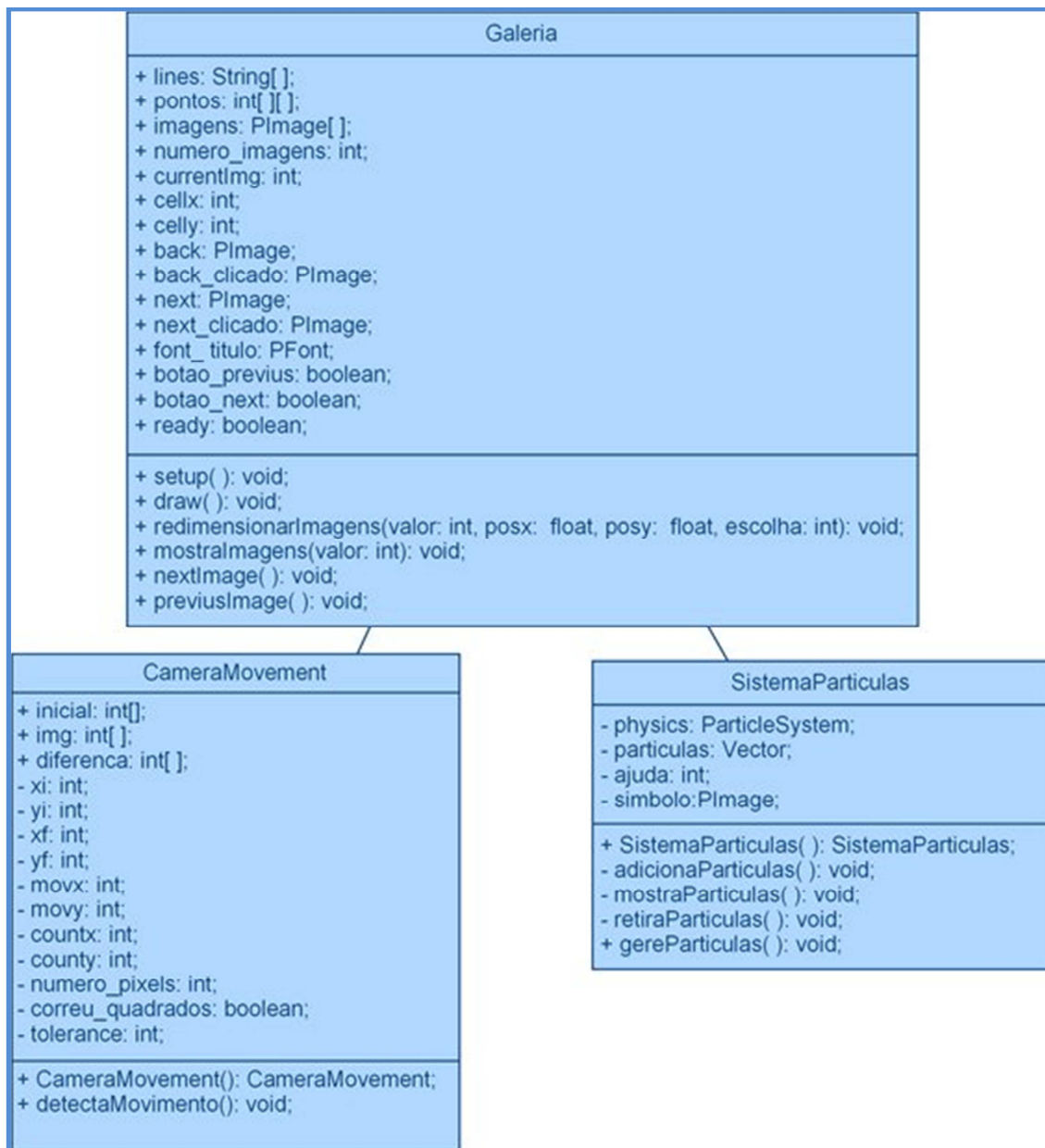


Figura 44 - Diagrama de classes do Wow!Gallery.

4.7. Wow!Album

O Wow!Album (Figura 45) tem como finalidade ajudar a venda de álbuns musicais em centros comerciais. Este protótipo consiste principalmente, na queda em posições aleatórias de imagens, que não são mais do que as capas de álbuns. Quando é detectado movimento sobre um destes aparece o videoclip do álbum no meio da janela. Ao tocar no videoclip, volta a ser possível a selecção de um novo álbum para ouvir.

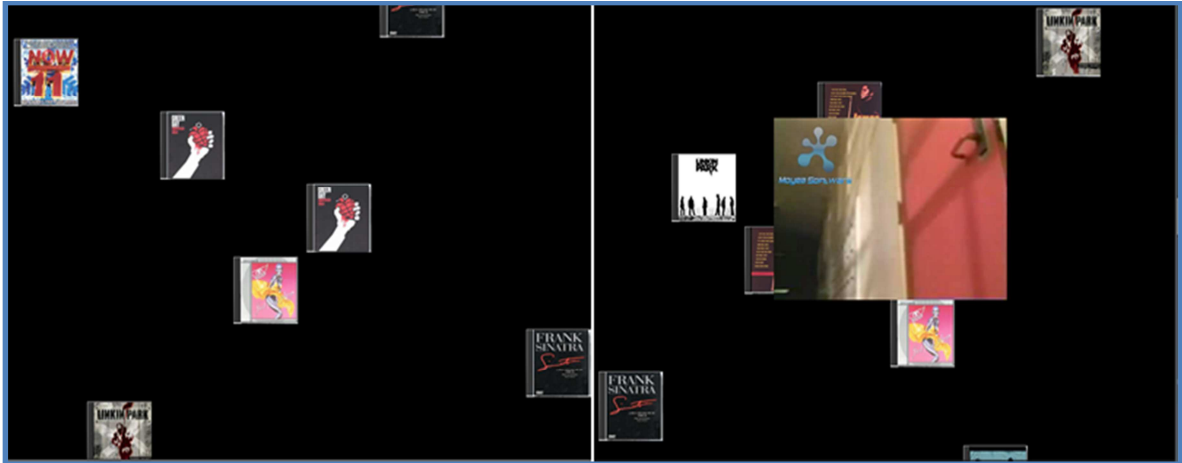


Figura 45 - Wow!Album.

4.7.1. Processo de desenvolvimento

O principal objectivo deste protótipo foi fazer com que as partículas fossem criadas em uma posição aleatória no eixo dos X, sendo que a posição no eixo dos Y seria negativa de modo a que na janela da aplicação a partícula não aparecesse lá vinda do nada. Com intuito, de contornar o problema de dois álbuns caírem juntos e ao tentar clicar no desejado ele detecta-se o movimento noutra álbum, dividiu-se a janela em oito colunas e fez-se com que as partículas fossem caindo aleatoriamente nestas colunas. De modo, a que as partículas não caíssem sequencialmente, fez-se de maneira que entre a criação de uma e outra partícula existisse um pequeno tempo de espera.

Depois se ter conseguido o objectivo anteriormente mencionado, a próxima etapa foi fazer com que ao ser detectado movimento sobre a capa do álbum este reproduzisse o videoclip no centro da janela. Caso estivesse a ser reproduzido um videoclip e fosse detectado movimento sobre este, a reprodução era interrompida e voltaria a estar disponível a opção de seleccionar outro álbum.

A Figura 46 demonstra um diagrama de classes deste protótipo. As classes *FasterMovie*, *CameraMovement* e *SistemaParticulas* já foram anteriormente referidos na Secção 4.4.1. (pág. 36) e a classe *InformacaoParticula* na Secção 4.5.1. (pág. 38). A classe *Albuns* é a classe principal e por consequência a responsável pelo funcionamento do protótipo, sendo igualmente responsável pela reprodução de um videoclip e pela verificação de se está ou não a ser reproduzido um videoclip no momento.

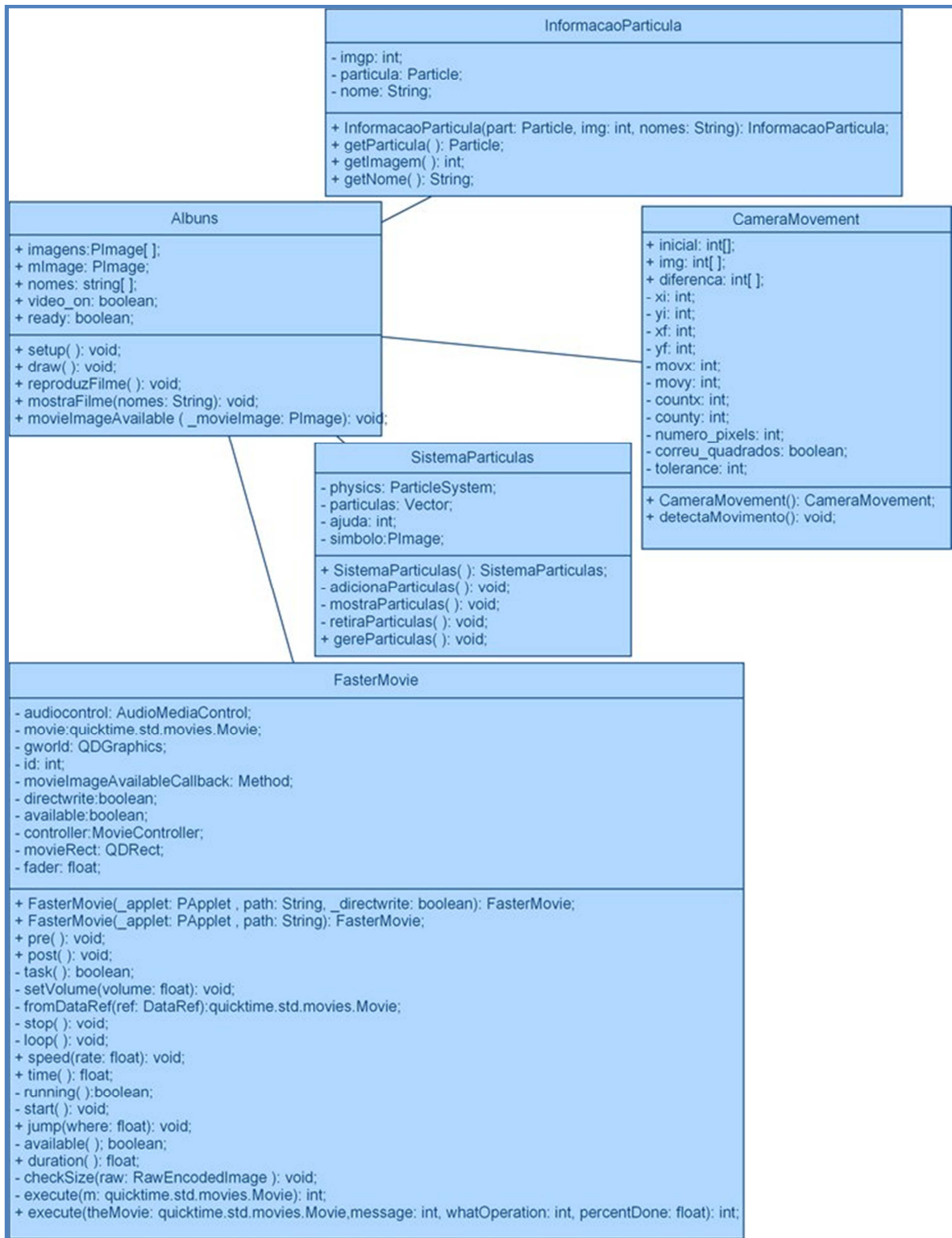


Figura 46 - Diagrama de classes do Wow!Album.

4.8. Wow!Blackjack

O Wow!Blackjack (Figura 47) tem como finalidade o entretenimento, ou seja, o objectivo primordial é fazer com que o utilizador jogue Blackjack de maneira interactiva. Este protótipo só tem interesse para utilizadores que tenham conhecimento do jogo e das suas regras, de modo a poderem usufruir de todas as potencialidades do protótipo.

Este protótipo, para além do entretenimento, pode ser utilizado para a publicidade uma vez que a mesa do jogo pode publicitar alguma marca, evento, etc. Assim sendo, este protótipo pode ser instalado não só num Casino para publicidade do jogo, mas também numa inauguração ou apresentação de algum produto.



Figura 47 - Wow!Blackjack

4.8.1. Processo de desenvolvimento

O objectivo primordial na implementação deste protótipo foi inserir as cartas e os botões na mesa numa posição específica. Ao se conseguir este objectivo, procedeu-se à implementação das regras do jogo e as funcionalidades dos botões, sendo que numa fase inicial os botões reagem ao rato. Finalizou-se o protótipo inserindo a detecção de movimento de modo a que o utilizador pudesse utilizar as mãos para interagir com o protótipo.

A Figura 48 ilustra o diagrama de classes deste protótipo. A classe *CameraMovement* já foi anteriormente referida na Secção 4.5.1. (pág. 38). A classe *Black_Jack* é a responsável pelo funcionamento do protótipo, sendo igualmente, responsável pela inserção das cartas na mesa, verificação das regras de jogo, decisão de qual a próxima carta a ser entregue e verificação de quem é o vencedor do jogo.

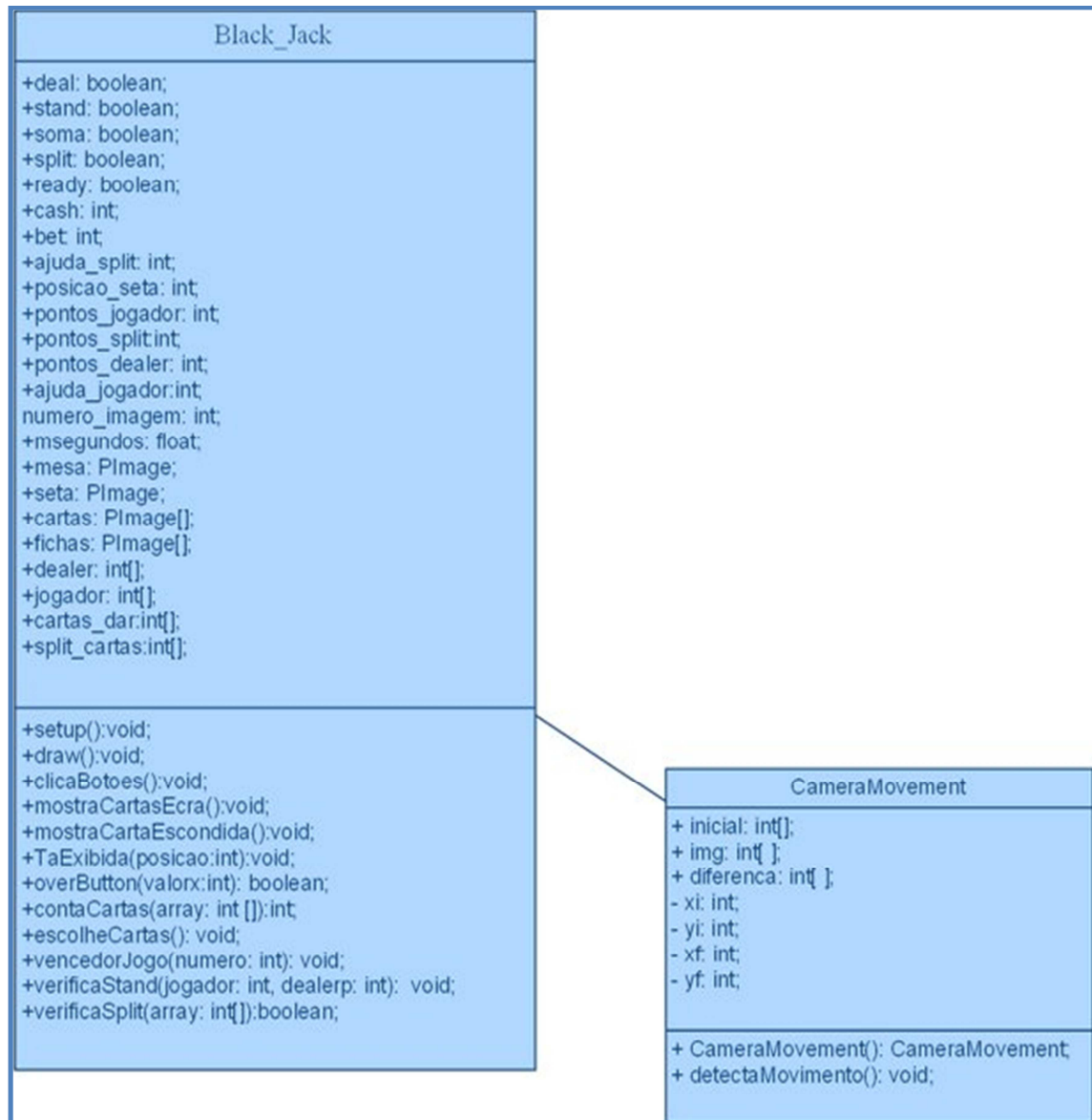


Figura 48 - Diagrama de classes do Wow!Blackjack.

4.9. Wow!Back_Office

A criação deste protótipo teve como finalidade a criação de uma interface gráfica para a configuração de alguns dos protótipos referidos no Capítulo 4. Com este protótipo a criação ou alterações de projectos foi simplificada, uma vez que retirou a necessidade de efectuar alterações no código. Este protótipo não é mais do que uma junção do código dos protótipos anteriores, sendo este alterado com o objectivo de se poder fornecer igualmente a opção de calibrar a câmara para que esta só detecte movimento na área desejada.

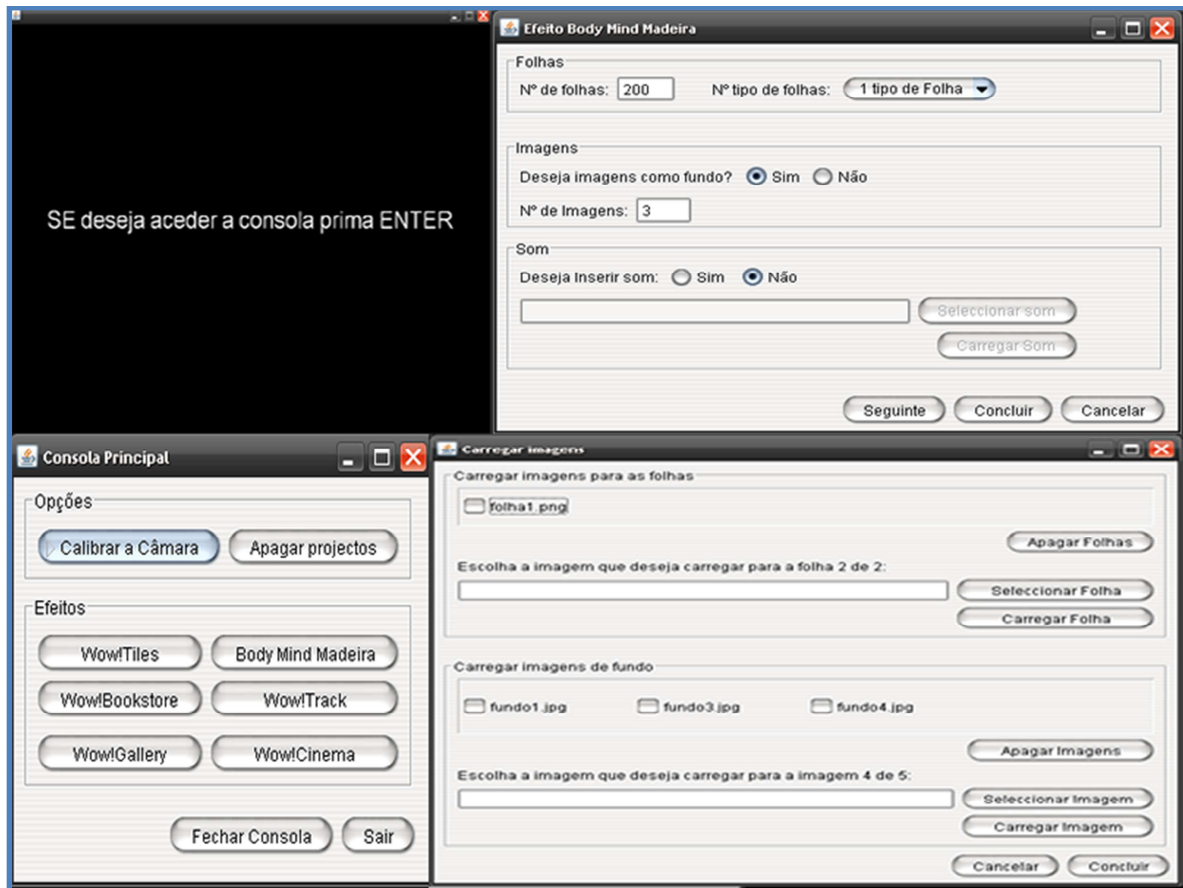


Figura 49 - Interfaces do Wow!Back_Office.

4.9.1. Processo de desenvolvimento

O objectivo inicial deste protótipo foi a implementação das interfaces gráficas. O programa *NetBeans 6.0.1* foi usado para o desenho das interfaces, passando-se posteriormente o código destas para o programa *Processing* que foi utilizado para a implementação deste protótipo. As informações que são configuradas para cada projecto irão ser guardadas dentro de uma pasta específica designada por *Processefeitos*. Dentro desta pasta encontra-se uma pasta para cada efeito. E cada projecto será guardado dentro da pasta do efeito.

Este protótipo tem como casos de utilização os demonstrados na Figura 50, ou seja, “Calibrar câmara”, “Apagar Projecto”, “Criar Projecto”, “Alterar Projecto”, “Configurar informações do projecto” e “Carregar imagens”. Todos os casos de utilização estão ligados ao utilizador embora o “Configurar informações do projecto” e “carregar imagens” se encontrem indirectamente ligados. Estes casos são extensões de outros casos (“Criar Projecto” e “Alterar Projecto”), ou seja, ao realizar um dos outros casos, faz com que estes sejam igualmente executados.

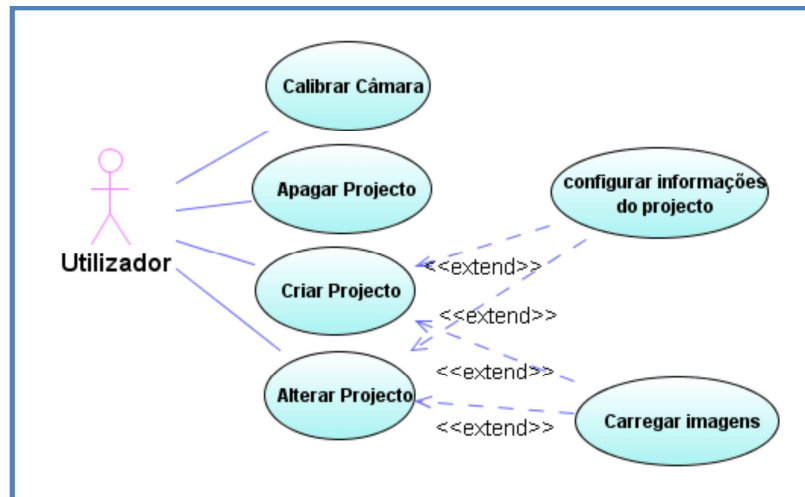


Figura 50 - Diagrama de casos de utilização.

O diagrama de actividades do caso de utilização “Calibrar Câmara” (Figura 51) representa as etapas seguidas pelo utilizador para calibrar a câmara para que esta capture o movimento na zona que ele deseja.

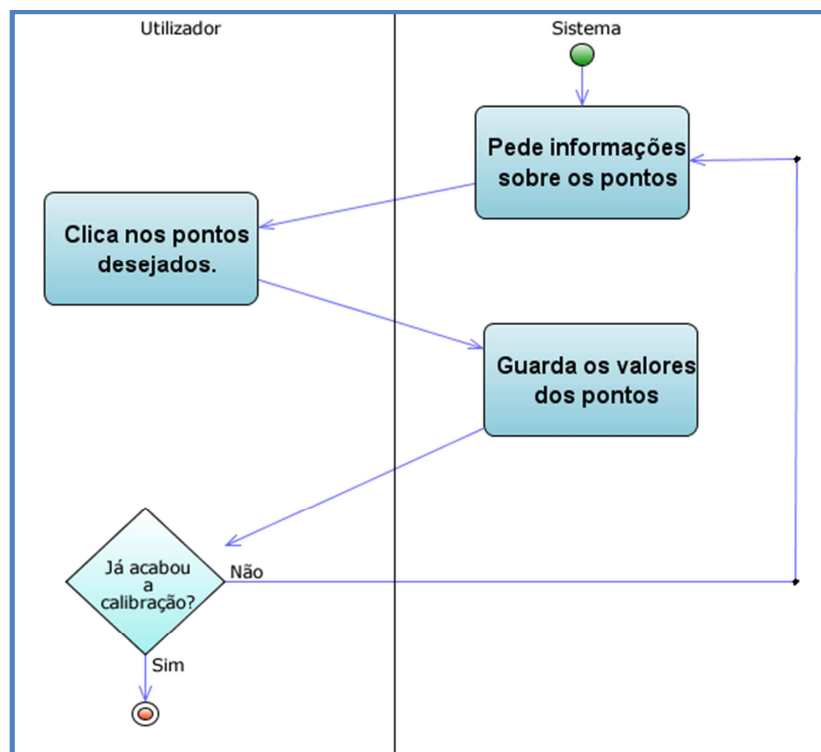


Figura 51 – Diagrama de actividades do caso de utilização “Calibrar câmara”.

O diagrama de actividades do caso de utilização “Apagar Projectos” (Figura 52) ilustra a sequência de etapas seguidas para que o utilizador apague os projectos, que já não tenha utilidade.

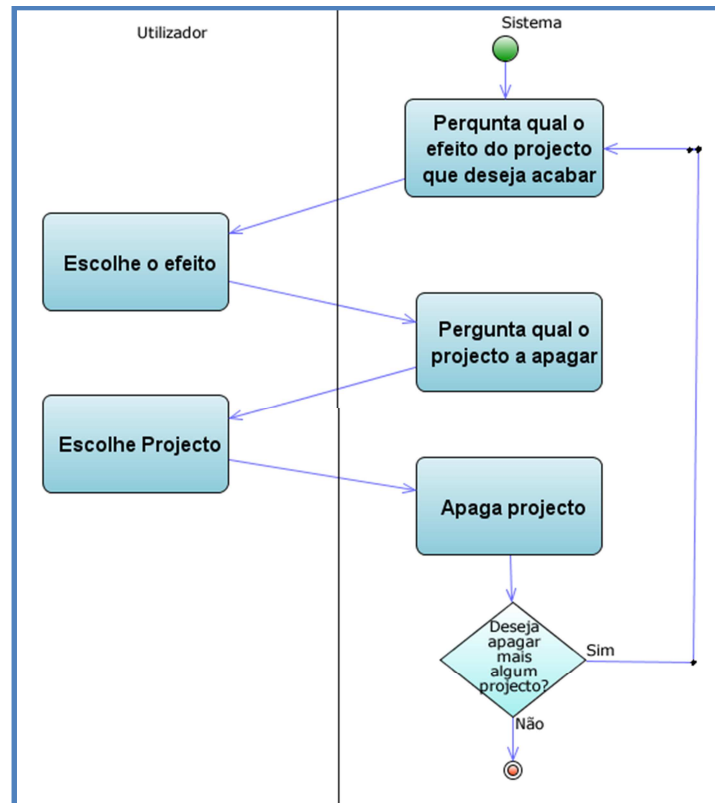


Figura 52 - Diagrama de actividades do caso de utilização "Apagar Projecto".

A Figura 53 representa o diagrama de actividades do caso de utilização "Criar projecto". Este diagrama ilustra a sequência de etapas que o utilizador tem de efectuar para a criação de um novo projecto.

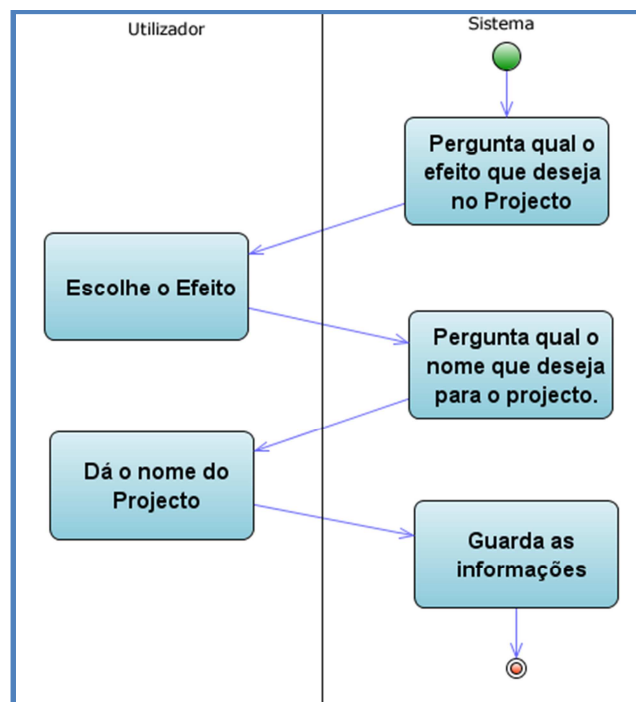


Figura 53 - Diagrama de actividades do caso de utilização "Criar projecto".

A Figura 54 ilustra o diagrama de actividades do caso de utilização “Alterar projecto”. Este diagrama é semelhante ao anterior, com a pequena diferença que em vez do sistema “pedir” o nome do novo projecto, pergunta qual o projecto existente que deseja alterar.

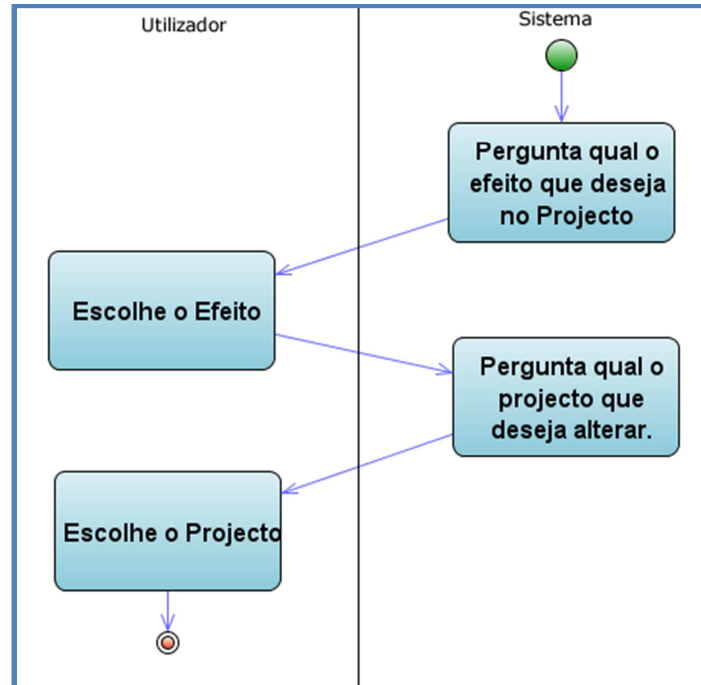


Figura 54 - Diagrama de actividades do caso de utilização “Alterar projecto”.

A Figura 55 exemplifica o diagrama de actividades do caso de utilização “Configurar informações do projecto”. Este representa as etapas seguidas pelo utilizador de modo a se poder configurar todos os dados do projecto. Este caso de utilização é invocado para que o utilizador possa concluir com sucesso tanto a criação como a alteração de um projecto.

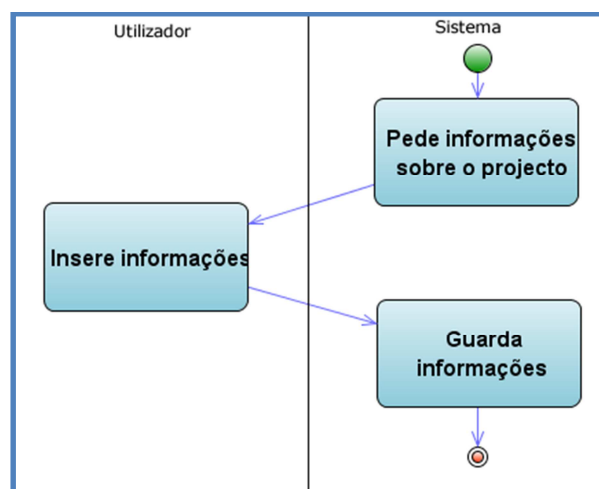


Figura 55 - Diagrama de actividades do caso de utilização “Configurar informações do projecto”.

O diagrama de actividades do caso de utilização “Carregar imagens” (Figura 56) ilustra a sequência de passos seguidos para que o utilizador carregue todas as imagens necessárias para o funcionamento do projecto. Este caso de utilização é invocado sempre que exista a criação de um novo projecto e na alteração de algum projecto caso seja necessário carregar novas imagens.

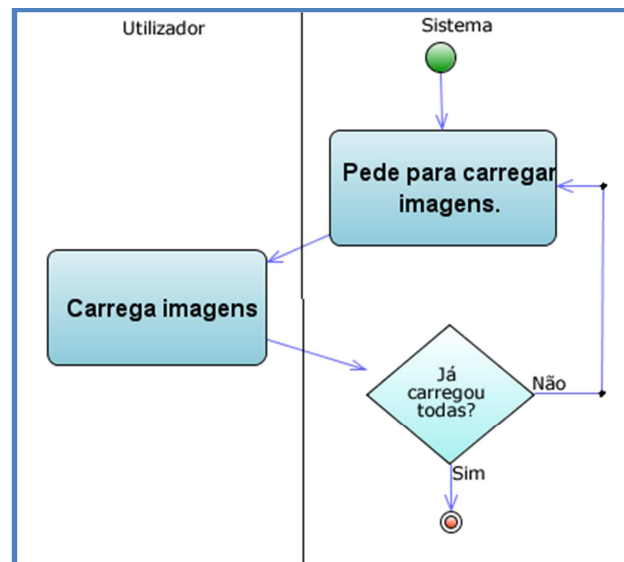


Figura 56 - Diagrama de actividades do caso de utilização “Carregar imagens”.

Assim sendo, o protótipo foi projectado de modo a que ao se executar aparecesse uma janela com o aviso “Se deseja aceder a consola prime ENTER”, como demonstrado na Figura 56. Ao clicar no botão em questão, aparece a Consola Principal com as opções que o utilizador dispõe, ou seja, a opção de calibrar a câmara, de apagar algum projecto anteriormente criado e criar ou alterar algum efeito.

Ao calibrar a câmara será apresentada uma janela com a imagem da câmara, sendo depois pedido para clicar no ponto inicial e final de onde deseja que comece e finalize a captura, como demonstrado na Figura 57. O quadrado a vermelho representa a zona que a câmara captará. Ao clicar nestes pontos, estes são guardados na pasta principal com o nome *contas.wow* que servirá para quando se executar um efeito, estes valores serão usados para a detecção de movimento. Decidiu-se por uma extensão desconhecida de modo a que se alguém fosse a pasta não tentasse abrir e alterar o ficheiro. Para voltar a consola principal basta clicar no ENTER.



Figura 57 - Janela para calibração da Câmara.

Caso o utilizador deseje apagar algum projecto apenas é necessário clicar no botão “Apagar projectos” e será apresentada a interface ilustrada na Figura 58. Nesta interface o utilizador selecciona o efeito do projecto que deseja apagar, sendo posteriormente seleccionado na ComboBox o projecto desejado. Depois basta clicar no botão “Apagar” e o projecto será apagado. Quando o utilizador tiver acabado de apagar todos os projectos desejados clica em “Concluir” para retornar a Consola Principal.

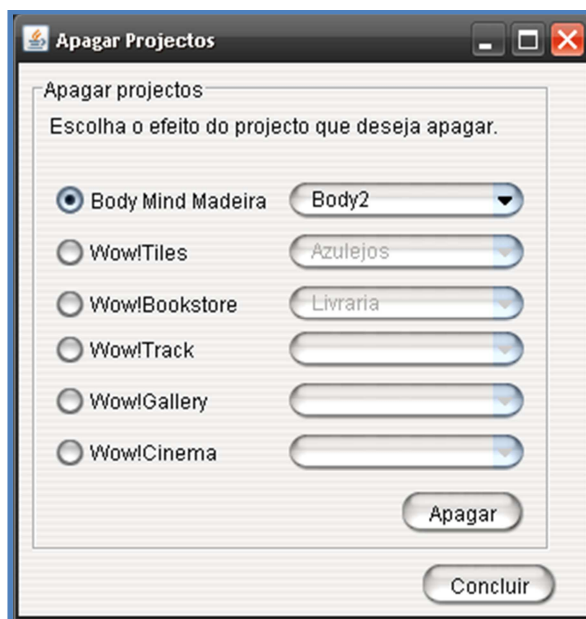


Figura 58 - Interface de apagar projectos.

Ao clicar em qualquer um dos efeitos será apresentada uma interface geral para todos (Figura 59), que consiste em saber se o utilizador pretende realizar um novo projecto do efeito em questão ou então alterar um já existente. Depois desta selecção clica-se em seguinte para continuar a configuração do projecto ou em cancelar para voltar a Consola Principal. Ao ser criado um novo projecto, é criada uma pasta com o nome dado ao projecto na pasta do efeito.

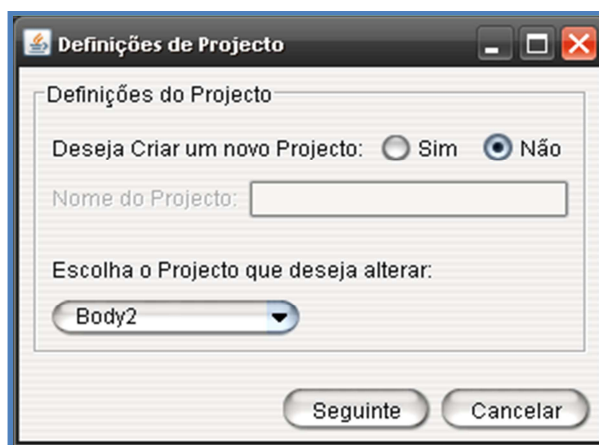


Figura 59 - Definições do projecto.

A interface seguinte é diferente para cada efeito, assim será efectuado uma breve descrição sobre algumas. Todas estas informações obtidas a partir destas interfaces são guardadas num ficheiro com o nome *Valores.wow* na pasta do projecto.

Nas configurações do *Wow!Tiles* (Figura 60) é pedido informação sobre o número de tipos de azulejos e de imagens de fundo desejados. É igualmente questionado se deseja inserir música de fundo, em caso afirmativo pede para seleccionar qual a música que

deseja. A música ao ser carregada será copiada para a pasta do projecto. No caso de estar a alterar um dos projectos, ao tocar no botão Concluir ele executará o protótipo. No caso de clicar no botão seguinte irá para a interface onde será carregadas as imagens dos azulejos e as imagens de fundo.

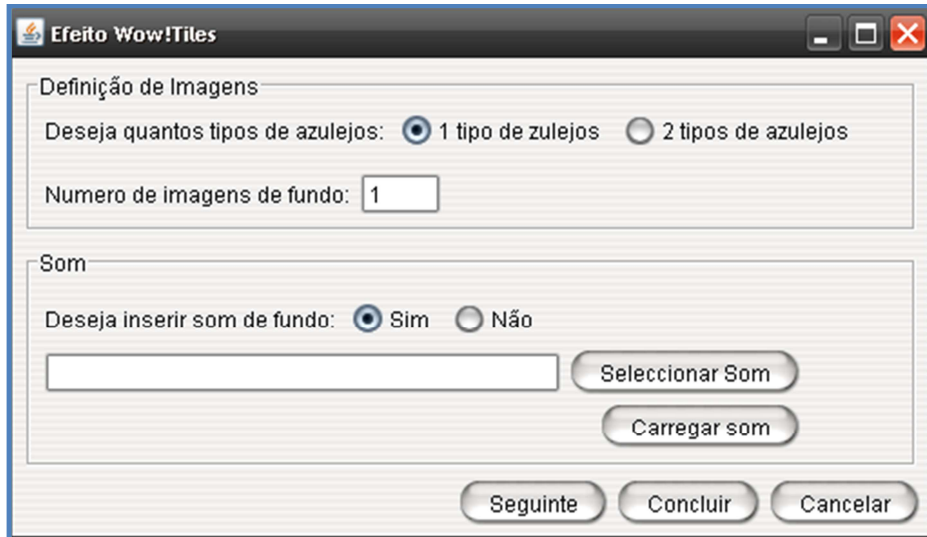


Figura 60 - Janela de configuração do Wow!Tiles.

Nas configurações do efeito *Body Mind Madeira* (Figura 61) é solicitada a informação sobre o número desejado de folhas, se deseja um ou dois tipos de folhas, se deseja imagens de fundo, e se sim qual o número de imagens de fundo que deseja carregar. Questiona igualmente se deseja som quando detectado movimento, em caso positivo pede para seleccionar e carregar os sons.



Figura 61 - Janela de configuração do efeito Body Mind Madeira.

Nas configurações do efeito *Wow!Bookstore* (Figura 62) é solicitada informações sobre quantas imagens deseja no eixo dos XX, quantas no eixo dos YY e quantas imagens no total deseja carregar. Ao guardar a informação o painel “Carrega Imagens” permite o utilizador carregar as imagens e se desejar apagar alguma anteriormente carregada. Todas as imagens carregadas são copiadas para a pasta do projecto. O nome das imagens será guardado num ficheiro designado *imagens.wow*. Quando o *Wow!Bookstore* for executado será criado um *array* que terá a informação lida deste ficheiro, ou seja, cada posição do *array* será um nome de uma imagem.

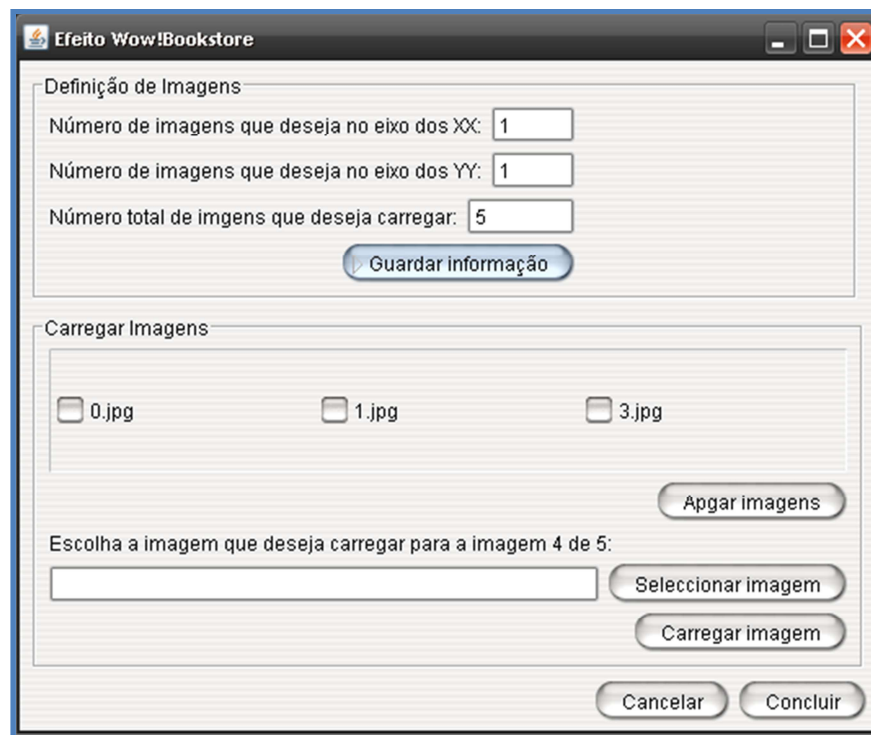


Figura 62 - Janela de configuração do efeito *Wow!Bookstore*.

Depois de configurar todas as informações tanto no *Wow!Tiles*, como no *Body Mind Madeira* tem-se que carregar as imagens desejadas, para tal criou-se esta interface (Figura 63) com esse objectivo. Ao carregar as imagens para as folhas, os nomes delas são guardados num ficheiro de texto com o nome *folhas.wow* e as imagens de fundo são guardadas num ficheiro de texto com o nome *imagens.wow* ficheiros estes que serão usados mais tarde quando for executado o protótipo. Todas as imagens carregadas são copiadas para a pasta do projecto e como a imagem continua com o mesmo nome, não é possível carregar duas imagens com o mesmo nome. Ao clicar em concluir será executado o projecto. Ao clicar em Cancelar volta-se a Consola Principal.

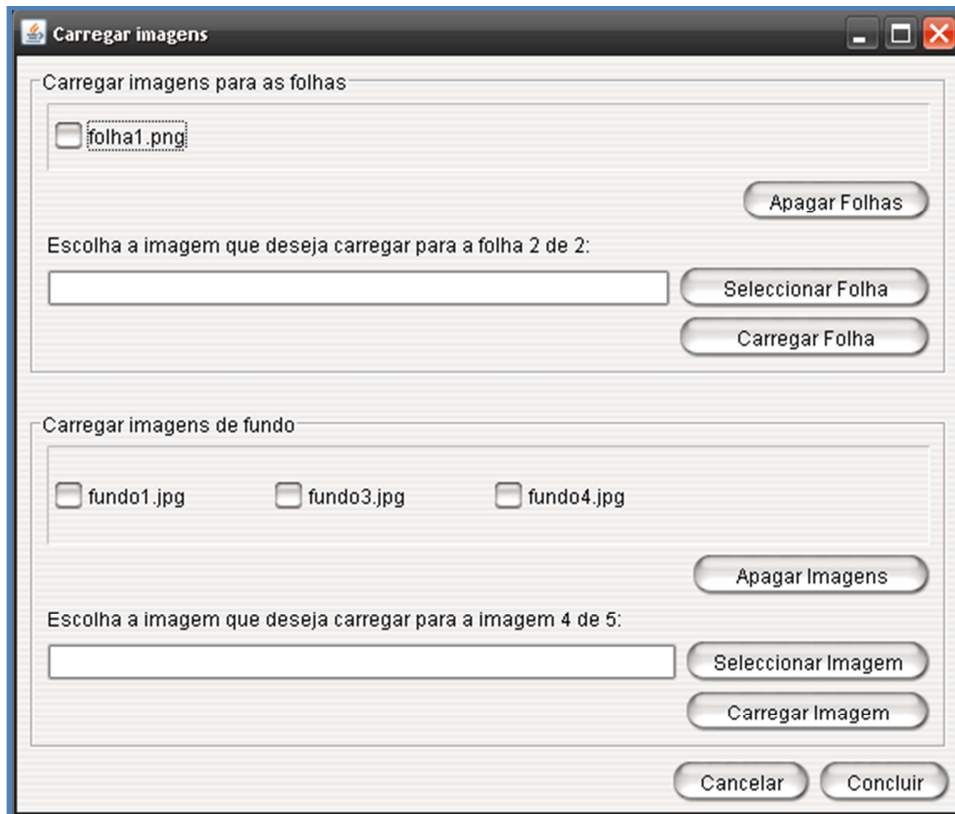


Figura 63 - Janela para carregar imagens.



Capítulo V - Testes e Resultados

Neste Capítulo será efectuada uma avaliação dos testes realizados e dos dados obtidos através desses testes. Uma vez que não foi possível realizar testes num lugar público, ou seja, implementação de uma mesa interactiva num local onde fosse possível observar e tirar informações sobre as reacções das pessoas em contacto com a mesa, assim sendo optou-se pela criação de testes informais.

Assim o primeiro teste consistiu em efectuar a montagem real da mesa com intuito de testar os primeiros protótipos e obter informações sobre como os utilizadores reagiriam a estes. Uma vez que não se conseguiu testar os protótipos com o público em geral, utilizou-se uma amostra representativa, sendo esta composta por pessoas de diferentes faixas etárias e com diferentes conhecimentos sobre a tecnologia e materiais utilizados na implementação deste projecto.

Uma das conclusões iniciais que se retirou foi que a mesa em si não era intuitiva, ou seja, o utilizador ao chegar perto da mesa não tinha noção de que deveria interagir com a mesa utilizando as mãos ou objectos como meio de comunicação. Outra conclusão foi o facto de que quanto menos os utilizadores soubessem sobre o funcionamento da mesa, mais impressionados ficavam e mais questões efectuavam sobre o funcionamento desta.

Um dos primeiros protótipos testados foi o Body Mind Madeira (Figura 64). Uma das conclusões que se extraiu deste protótipo foi que quando o utilizador percebia que ao colocar a mão sobre a mesa as folhas se afastavam, este tentava afastar todas as folhas do ecrã com a “desculpa” de poder observar a imagem de fundo. Quando já não existia folhas, o utilizador deixava de efectuar movimento o que desencadeava a resposta do protótipo, ou seja, mudava a imagem de fundo e fazia com que as folhas retornassem aos lugares iniciais, o que por sua vez estimulava o utilizador a iniciar novamente a “brincadeira” de afastar as folhas.

Uma outra conclusão retirada foi a reacção do utilizador quanto à reprodução do som uma vez que era detectado movimento. Inicialmente a reprodução de som criava um pouco de receio pois não entendiam o que desencadeava o som, mas ao perceberem que o som era reproduzido quando era detectado movimento, até achavam interessante e engraçado.



Figura 64 - Teste Body Mind Madeira.

Outro dos protótipos testados foi o Wow!Tiles (Figura 65). A principal conclusão retirada foi que o utilizador inicialmente tentava retirar todos os azulejos de maneira a poder aceder a imagem de fundo, contudo após um tempo já tentavam brincar com os azulejos, ou seja, retirar um de cada vez de modo a poderem analisá-los a saírem.

A desvantagem deste protótipo tem a ver com a inexistência de informação sobre as imagens de fundo, ou seja, qual a localização da paisagem. Isto pode ser considerado um problema, uma vez que este protótipo tem como função principal a promoção de fins turísticos, assim sendo é de todo o interesse que estas venham acompanhadas de informação, de modo a que se o utilizador caso pretenda visitá-la ter conhecimento de pelo menos o nome da localização.



Figura 65 - Teste do Wow!Tiles.

O Wow!Bookstore (Figura 66) foi igualmente um dos protótipos testados numa mesa real. Um dos problemas iniciais deste protótipo foi que este não era muito intuitivo, ou seja, o utilizador frente a este protótipo não tinha a noção que ao clicar num livro este efectuava movimentos rotacionais, permitindo outro livro ser exibido. Mas uma vez este conhecimento adquirido os utilizadores achavam interessante e já se divertiam a mudar os livros.

Outro dos problemas neste protótipo foi que caso o utilizador desejasse virar o livro superior, devido a técnica de detecção de movimento, o livro inferior também rodaria visto que a técnica de detecção de movimento detectava movimento sobre a mesa e não o toque na mesa.

Em geral os utilizadores que testaram este protótipo acharam este protótipo interessante, uma vez que permitia ao utilizador poder seleccionar o livro a comprar e em simultâneo aceder a informação essencial sobre o livro em questão.



Figura 66 - Teste do Wow!Bookstore.

Em relação aos outros protótipos não foi possível testá-los numa mesa real, nos quais seria usado uma amostra representativa de modo a obter resultados. Devido a esta impossibilidade, os testes foram realizados com recurso a uma caixa de cartão utilizada inicialmente para se testar o funcionamento dos protótipos (Secção 3.2.1, pág. 17).

Um dos testes realizados foi analisar como seria o comportamento do protótipo Wow!Track no ramo da publicidade. Para tal construiu-se o Wow!Track (pág. 38), de modo a que passassem como imagens de fundo um modelo da Nissan. E quando se detectasse movimento sobre a mesa, fosse possível observar um rasto de símbolos da Nissan. A Figura 67 representa um gráfico com os resultados obtidos. Os resultados foram divididos em duas perguntas: Se tinham conhecimento da marca que estava na publicidade? e Se conheciam o modelo de automóvel que se encontrava nas imagens?.

Estes dados foram obtidos através das perguntas anteriormente efectuadas, num intervalo de tempo (cerca de 10 minutos), após o utilizador ter interagido com o protótipo. Ao observar o gráfico concluiu-se que a taxa de *brand recall* da marca é elevada, ou seja, 9 em 10 utilizadores lembraram-se que marca presente no protótipo era da Nissan. Quanto a pergunta se o utilizador tinha conhecimento do modelo presente nas fotos, a taxa foi significativamente mais reduzida, uma vez que também se deve ao facto de não existir

informação permanente sobre o modelo, ou seja, só em algumas imagens era referenciado o modelo do carro em questão.

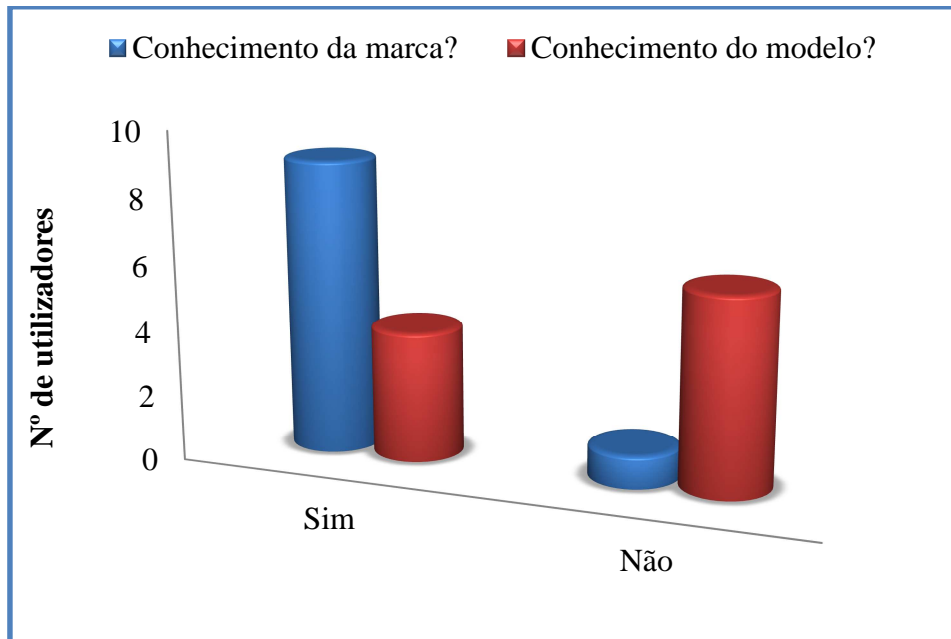


Figura 67 - Gráfico referente ao teste realizado ao protótipo Wow!Track.

Em todos os protótipos dois dos aspectos mais importantes e dos quais se tentou medir foram o interesse e a utilidade do protótipo em relação ao utilizador. Estes aspectos tem um impacto fundamental, pois seja qual for o domínio em que o protótipo seja aplicado, o grau de interesse do utilizador determinará o sucesso do protótipo e indirectamente poderá determinar, consequentemente, se este tem ou não utilidade. A forma utilizada para avaliar o interesse do utilizador foi principalmente o tempo em que o utilizador interagiu com o protótipo, embora se tivesse em consideração a resposta do utilizador referente a esse aspecto. A maneira utilizada para medir a utilidade do protótipo, com base no domínio de aplicação, se o utilizador aprendeu ou adquiriu alguma informação ao interagir com o protótipo.

Assim sendo, estes aspectos foram avaliados no protótipo Wow!Cinema (pág. 35). Uma vez que num caso real este protótipo tinha como objectivo ser implementado na entrada de uma sala de cinema, tentou-se medir a utilidade do protótipo, tendo como parâmetro a ajuda na selecção do filme a assistir. De maneira a avaliar o interesse teve-se em conta tanto a resposta do utilizador como também o tempo em que o utilizador interagiu com o protótipo. Os dados obtidos neste estudo encontram-se ilustrados na Figura 68.

Ao analisar o gráfico pode-se concluir que a percentagem de interesse dos utilizadores em relação ao protótipo é positiva, cerca de 70 %, sendo que a percentagem de

utilidade deste protótipo seja um pouco menor, cerca de 50 %, pois segundo os utilizadores embora este protótipo fosse interessante, eles quando iam ao cinema já iam com a ideia definida de qual o filme que iriam assistir. Embora existisse hipoteticamente uma mesa na entrada da sala de cinema, mesmo ao interagirem com esta, provavelmente não os iriam mudar de ideia em relação ao filme a assistir, porém teriam conhecimento de todos os filmes em exibição e os futuros a exhibir.

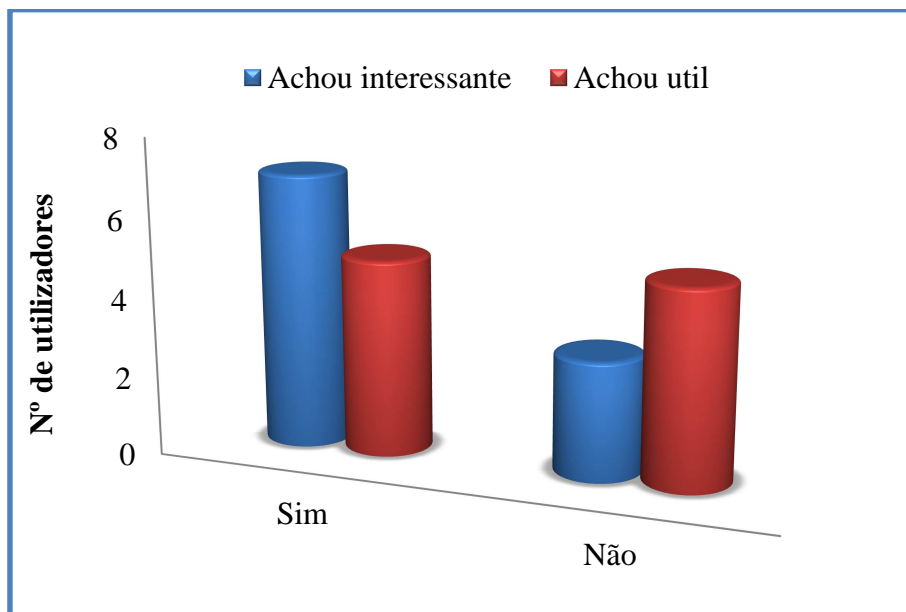


Figura 68- Gráfico referente ao teste do protótipo Wow!Cinema.

Outras conclusões que se pode retirar da realização deste teste foi que a utilização deste protótipo era intuitiva, ou seja, uma vez que a maneira de passar de filme para filme era através de botões o utilizador teve a percepção imediatamente do funcionamento deste protótipo, o que facilitava a interação entre o utilizador e o protótipo.



Conclusões

Como conclusões finais pode-se afirmar que a detecção de movimento pela câmara de IV tem como vantagens a fácil instalação e utilização, contudo uma das principais desvantagens inicialmente era a detecção de movimento, uma vez que esta era um pouco imprecisa. Este problema foi resolvido quando se começou a utilizar a detecção de movimento através dos pixéis. Ao utilizar esta técnica resolveu-se o problema da precisão, porém em contrapartida era detectado movimento mesmo que não se tocasse na mesa, ou seja, ao passar a mão por cima da mesa o movimento já era detectado. Concluiu-se que este pequeno problema não tinha muita relevância, uma vez que os protótipos criados não necessitavam de diferenciar o movimento quando tocado na mesa do movimento de quando passada a mão sobre a mesa.

Na parte final do projecto encontrou-se uma biblioteca, *touchlib*, que conseguia diferenciar os dedos do resto da mão, o que contribuiu para uma detecção “perfeita”. Devido a recente descoberta desta biblioteca não foi possível implementá-la em nenhum protótipo, porém testou-se os códigos de demonstração para perceber o potencial da biblioteca.

O trabalho realizado durante a implementação deste projecto permite uma interacção do utilizador com a mesa, sendo para tal usado não só as mãos como também objectos colocados sobre esta, em oposição aos instrumentos regularmente utilizados (rato e teclado), possibilitando assim uma aproximação do utilizador com a realidade, permitindo a este a ideia que tem o poder de controlar tudo.

Uma outra conclusão foi que a mesa não era intuitiva ao início, ou seja, o utilizador ao se deparar pela primeira vez com a mesa não tinha a noção que o objectivo era ele interagir com a mesa através de movimentos. Mas ao adquirir este conhecimento começava a interagir com a mesa normalmente.

Em relação aos protótipos as conclusões principais foram que em geral os utilizadores acharam-nos interessantes e que tinham utilidade no domínio aplicado. Um exemplo disso foi a aplicação do Wow!Track no domínio da publicidade e foi possível observar que a taxa de *brand recall* da marca, ou seja, a capacidade de se lembrarem da marca em exposição era alta.



Perspectivas Futuras

Num futuro próximo tem-se como objectivo estudar e descobrir todas as potencialidades da biblioteca *touchlib*, de modo a poder criar protótipos com uma precisão “perfeita”. Assim sendo, com recurso a esta biblioteca tem-se igualmente como objectivo a criação de protótipos com este tipo de detecção.

Uma das futuras criações será a construção de um protótipo para a manipulação de fotos ou imagens, ou seja, na mesa estarão distribuídas fotos de forma aleatória e o utilizador poderá movimentar as fotos para a posição desejada, sendo possível igualmente aumentar ou diminuir as imagens. Quando este objectivo estiver concluído, a etapa seguinte será realizar o mesmo que se efectuava com as fotos mas com vídeos, dando a opção do utilizador reproduzir e parar o vídeo quando desejado. Este protótipo estará mais direccionado para a promoção de um local turístico, embora possa ser igualmente utilizado para publicidade ou entretenimento.

Outro dos objectivos é a criação de um protótipo para realizar pinturas, ou seja, o utilizador poderá pintar ou desenhar o que desejar clicando simplesmente na mesa. Terá a opção de seleccionar a cor desejada e poder mudar de cor quando desejar. O principal objectivo deste protótipo será o entretenimento, embora possa ser igualmente utilizado para publicidade.

Um dos objectivos principais encontra-se igualmente em efectuar um teste real dos protótipos, isto é, tentar implementar uma mesa num lugar público de modo a ser possível analisar as reacções das pessoas que com ela interagirem e a partir desses dados ver algo nos protótipos que possa ser alterado de modo a estes ficarem mais intuitivos.



Bibliografia

1. YDreams: Home. *YDreams*. [Online] [Citação: 11 de Outubro de 2007.] http://www.ydreams.com/ydreams_2005/index.php?version=1&lang=en.
2. YDreams: YDREAMS ADVERTISING. *YDreams*. [Online] [Citação: 11 de Outubro de 2007.] http://www.ydreams.com/ydreams_2005/index.php?page=6.
3. EyeClick - EyeBorad- Turn your tables towards tomorrow. *EyeClick*. [Online] [Citação: 11 de Outubro de 2007.] http://www.eyeclick.com/products_501.html.
4. EyeClick - EyeStep - Step into action. *EyeClick*. [Online] [Citação: 11 de Outubro de 2007.] http://www.eyeclick.com/products_500.html.
5. Init-Lab - Works/Innovation. *Init-Lab*. [Online] [Citação: 11 de Outubro de 2007.] http://www.init-lab.com/works_6.php.
6. LCI - Interactive. *LCI - Homepage*. [Online] [Citação: 11 de Outubro de 2007.] <http://www.lci-uk.com/filter.php?filter%5B%5D=interactive>.
7. Panasonic's interactive TV wall, the demo. *Welcome to Akihabara news*. [Online] [Citação: 13 de Outubro de 2007.] <http://www.akihabaranews.com/en/review-63-X.html>.
8. HDTV: The interactive and intelligent table by Panasonic. *Welcome to Akihabara News*. [Online] [Citação: 13 de Outubro de 2007.] <http://www.akihabaranews.com/en/news-11387-HDTV:+The+interactive+and+intelligent+table+by+Panasonic.html>.
9. Interactive LED Coffee Table - The Wave. *We make great things and we make things great*. [Online] [Citação: 13 de Outubro de 2007.] <http://www.becausewecan.org/node/533>.
10. MIT Tangible Media Group | Projects- PingPongPlus. *MITMedia Lab - Tangible Media Group*. [Online] [Citação: 14 de Outubro de 2007.] <http://tangible.media.mit.edu/projects/pingpongplus/>.
11. *CATCHYOO - Official Website*. [Online] [Citação: 14 de Outubro de 2007.] <http://www.catchyoo.com/>.
12. *Microsoft Surface*. [Online] [Citação: 16 de Outubro de 2007.] <http://www.microsoft.com/surface/>.
13. *Natural Interaction SensitiveTable*. **Valli, A.** s.l. : ACM, 2008.
14. *SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces*. **Rekimoto, J.** s.l. : ACM, 2002.
15. Gesturetek || Illuminate Series. [Online] [Citação: 27 de Abril de 2008.] http://www.gesturetek.com/illuminate/productsolutions_illuminatetable.php.
16. IGD A4 - Virtual and Augmented Reality - Projects. *IGD A4 - Virtual and Augmented Reality*. [Online] [Citação: 3 de Maio de 2008.] <http://a4www.igd.fraunhofer.de/projects/48/>.

17. NOR_D. [Online] [Citação: 22 de Maio de 2008.] <http://nortd.com/cubit/>.
18. Processing 1.0 (BETA). [Online] [Citação: 27 de Setembro de 2007.] <http://processing.org/>.
19. Adobe Flash CS3 Professional, Interactive Multimedia, Interactive Design. [Online] [Citação: 22 de Novembro de 2007.] <http://www.adobe.com/products/flash/?promoid=BPDEE>.
20. NUI Group. [Online] [Citação: 2 de Maio de 2008.] <http://www.nuigroup.com/>.
21. traer.physics. [Online] [Citação: 2 de Outubro de 2007.] <http://www.cs.princeton.edu/~traer/physics/>.
22. FullScreenApp \ Learning \ Processing 1.0 (BETA). [Online] [Citação: 2 de Novembro de 2008.] <http://www.processing.org/learning/3d/fullscreenapp.html>.
23. FullScreen API For Processing. [Online] [Citação: 10 de Novembro de 2007.] http://www.superduper.org/processing/fullscreen_api/.
24. Myron (WebCamXtra) - Computer Vision & Well Connected Motion Tracking. [Online] [Citação: 2 de Outubro de 2007.] <http://webcamxtra.sourceforge.net/>.
25. Processing 1.0 (Beta). [Online] <http://www.processing.org/>.
26. SurfaceLib. [Online] [Citação: 3 de Janeiro de 2008.] <http://www.eskimoblood.de/surfacelib/>.
27. 180. [Online] [Citação: 12 de Fevereiro de 2008.] <http://www.timroth.de/180/index.html>.