

TIAGO ALMEIDA BUNGENSTAB

DESENVOLVENDO APLICAÇÕES
MÓVEIS EM FLASH LITE

Assis
2009

DESENVOLVENDO APLICAÇÕES MÓVEIS EM FLASH LITE

TIAGO ALMEIDA BUNGENSTAB

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: LUIZ CARLOS BEGOSSO

Analisador: RITA DE CÁSSIA CASSIANO LOPES

Assis
2009

TIAGO ALMEIDA BUNGENSTAB

DESENVOLVENDO APLICAÇÕES
MÓVEIS EM FLASH LITE

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: LUIZ CARLOS BEGOSSO

Área de Concentração: _____

Assis
2009

AGRADECIMENTOS

Primeiramente, agradeço a Deus que providenciou tudo aquilo que me foi necessário, e sua misericórdia nunca faltou.

Agradeço aos meus pais, Tarcísio José Bungenstab e Clarice Almeida Bungenstab, pelo incentivo e pelas palavras de sabedoria.

Aos meus irmãos, amigos e colegas, pelo companheirismo e pelas diversas coisas que partilhamos.

A meu orientador Luiz Carlos Begosso, pelo direcionamento deste trabalho e aos demais professores, pelas disciplinas aplicadas no decorrer do curso.

E a todos os autores de livros e artigos citados nas referências desse trabalho, pela ajuda prestada e pelas ideias emprestadas.

RESUMO

O mercado de jogos para dispositivos móveis cresce a cada ano, devido ao aumento exponencial no número desses aparelhos. Este trabalho tem por objetivo: fazer um estudo exploratório na área de desenvolvimento de jogos, e estudar uma tecnologia de desenvolvimento para dispositivos móveis, o Flash Lite: Construir um jogo e demonstrar o seu desenvolvimento com o Flash Lite.

Palavras-Chave: Dispositivos móveis. Jogos. Flash Lite.

ABSTRACT

The market for mobile games grows every year due to the exponential increase in the number of such devices. This paper aims: to make an exploratory study in the development of games, and studying a technology development for mobile devices, Flash Lite. Build a game and show its development with Flash Lite.

Keywords: Mobile devices. Games. Flash Lite.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tabuleiro de Xadrez.....	24
Figura 2 - As Peças do Xadrez.....	25
Figura 3 - Movimento do Peão.....	26
Figura 4 - Movimento da Torre.....	26
Figura 5 - Movimento do Bispo.....	27
Figura 6 - Movimento do Cavalo.....	27
Figura 7 - Movimento da Rainha.....	28
Figura 8 - Movimento do Rei.....	28
Figura 9 - Regra de Captura do Peão.....	29
Figura 10 - Exemplo de Limite de Movimento 1.....	30
Figura 11 - Exemplo de Limite de Movimento 2.....	30
Figura 12 - Exemplo de Limite de Movimento 3.....	31
Figura 13 - Exemplo de Movimento do Cavalo.....	31
Figura 14 - Rei em Xeque.....	32
Figura 15 - Xeque-Mate do Rei.....	33
Figura 16 - Adobe Device Central.....	36
Figura 17 - Adobe Flash CS4 Professional.....	39
Figura 18 - Tela Inicial do Jogo.....	41
Figura 19 - Destaque das Casas de Possível Movimento.....	41

LISTA DE ABREVIATURAS E SIGLAS

J2ME	Java to Micro Edition
RPG	Role Playing Game
SDK	Software Development Kit
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
AS2	ActionScript 2
FLV	Flash Video
MC	MovieClip

SUMÁRIO

1. INTRODUÇÃO	10
1.1. OBJETIVOS.....	11
1.2. MOTIVAÇÃO	11
1.3. ESTRUTURA DO TRABALHO	12
2. DESENVOLVIMENTO DE JOGOS	13
2.1. CATEGORIAS	14
2.2. PÚBLICO-ALVO	16
2.3. PLATAFORMA	16
2.4. ETAPAS DE DESENVOLVIMENTO	17
2.4.1. Design Bible.....	17
2.4.1.1. Game Design.....	18
2.4.1.2. Game Play	18
2.4.1.3. Interface Gráfica	19
2.4.1.4. Roteiro	19
2.4.2. Criação Das Imagens E Áudio.....	20
2.4.3. Implementação	20
2.4.3.1. Codificação	21
2.4.3.2. Testes.....	22
2.4.3.3. Manutenção	22
3. PROPOSTA DO SOFTWARE.....	23
3.1. PUBLICO ALVO	23
3.2. PLATAFORMA	23
3.3. DESENVOLVIMENTO.....	24
3.3.1. Design Bible.....	24
3.3.1.1. Game Design.....	24
3.3.1.2. Game Play	33
3.3.1.3. Interface Gráfica	33
3.3.1.4. Roteiro	34
3.3.2. Criação Das Imagens	35
3.3.3. Implementação	35
3.3.3.1. Codificação	35
3.3.3.2. Testes.....	36
3.3.3.3. Manutenção	36
3.4. FLASH LITE.....	37
3.4.1. ActionScript.....	38
4. SOFTWARE XADREZ	39
4.1. MENU	40
4.2. JOGO.....	41
5. CONCLUSÃO	44
REFERÊNCIAS BIBLIOGRÁFICAS	45

1. INTRODUÇÃO

O mercado de desenvolvimento de jogos para dispositivos móveis tem crescido muito nos últimos anos. Mas apesar de esta área estar crescendo cada vez mais, o mercado ainda é muito pequeno aqui no Brasil. O que tem contribuído para este constante crescimento é o grande número usuários de dispositivos móveis (celulares, smartphones, PDAs, etc.), não só no Brasil, mas em todo o mundo.

A tecnologia mais utilizada neste mercado é a versão *Mobile* (versão para dispositivos móveis) do Java, da Sun Microsystems, *J2ME* (Java to Micro Edition). Para designar “dispositivos móveis”, opta-se por utilizar neste trabalho, a terminologia comumente empregada na área, *Mobile*. Mas não é a única tecnologia para desenvolvimento, existem outras como, a própria SDK (Software Development Kit) do fabricante dos aparelhos e outras de linguagem, entre elas C/C++, Python e Java. Existe também uma versão *Mobile*, assim como o J2ME, do Adobe Flash Player, da empresa Adobe Systems Incorporated, que é uma versão compacta do Flash Player para rodar em dispositivos móveis.

O Flash no início era somente um programa de animação, para desenvolvimento de aplicações Web, desenvolvido pela Macromedia. Hoje o Flash não é apenas uma forma de fazer animações, mas se tornou uma plataforma de desenvolvimento, graças a sua linguagem de programação, o Actionscript, atualmente na versão 3. O Actionscript contribuiu para o avanço de outras tecnologias derivadas dele, como Adobe Flex, Adobe Air, etc.

Em 2005 a Macromedia lança o Flash Lite, na sua versão *1.0*, que foi desenvolvido baseado no *Flash 4*. Alguns anos mais tarde é lançada a o *Flash Lite 2.0*, desta vez baseada no *Flash 7*. No ano de 2007 a Adobe anuncia o *Flash Lite 3.0* e em seguida o *Flash Lite 3.1*. Segundo a Adobe Systems, o Flash Lite já vem pré-instalado em mais de 400 modelos de aparelhos celulares, e são mais de 800 milhões de aparelhos em todo o mundo com suporte ao *Flash Lite*.

1.1. OBJETIVO

Este trabalho tem dois objetivos:

1. Realizar um estudo exploratório sobre o desenvolvimento de jogos para dispositivos móveis, utilizando a tecnologia Flash Lite. Para atender este objetivo serão abordadas questões relacionadas às vantagens e desvantagens do Flash Lite com outras tecnologias de desenvolvimento para dispositivos móveis;
2. Implementar uma aplicação móvel utilizando a tecnologia abordada neste trabalho, o Flash Lite. A aplicação consiste no desenvolvimento de um jogo para celular.

1.2. MOTIVAÇÃO

O *Flash Lite* é uma tecnologia nova, e vem se tornando cada dia mais popular e com mais dispositivos de suporte. Seguindo a tendência mundial, de migrar para a tecnologia móvel, o *Flash Lite* é uma tecnologia de grande potencial, e com muitos desenvolvedores ao redor do mundo.

O mercado de jogos para dispositivos móveis cresce a cada dia, empresas de desenvolvimento vêm acumulando lucros. Pode-se citar como exemplo: a **Ovi Store**, loja virtual de aplicativos para celulares e smartphones, da Nokia.

1.3. ESTRUTURA DO TRABALHO

No primeiro capítulo, esta Introdução, são apresentados os objetivos e a motivação para a realização deste trabalho.

No segundo capítulo apresenta-se uma análise sobre o desenvolvimento de jogos, os tipos mais comuns, bem como as fases para seu desenvolvimento.

O terceiro capítulo aborda a proposta do software que será desenvolvido neste trabalho de conclusão de curso, e tece importantes comentários sobre a tecnologia utilizada, o *Flash Lite*.

No quarto capítulo apresenta-se o software desenvolvido utilizando a tecnologia estudada neste trabalho e um pouco da sua implementação.

No quinto capítulo são discutidas as conclusões obtidas a partir da realização deste trabalho.

2. DESENVOLVIMENTOS DE JOGOS

Os jogos eletrônicos sempre chamaram a atenção, através de cada vez mais interatividade com o jogador, fazendo com que o jogador passe horas jogando sem achar que esta seja uma atividade monótona. Isto ocorre desde antigos jogos clássicos, como Super Mario Bros (Nintendo), Sonic (Sega), e Pacman (Namco), até jogos mais recentes e complexos, alguns até sendo consideradas verdadeiras febres mundiais, assim como World of Warcraft (Blizzard), ou jogos da série de RPG Final Fantasy (Square Enix).

Mesquita Júnior (2007) descreve que é a partir desta capacidade de interação que se concentra o diferencial dos jogos eletrônicos, quando comparados com outras mídias de entretenimento. Desta maneira, os jogos eletrônicos caracterizam-se por sua linguagem hipertextual, capaz de unir texto, imagens, sons e narrativas, unindo o mundo real ao virtual, ampliando as fronteiras de um campo de estudos que embora, ainda incipiente, destaca-se, no cenário acadêmico, pela multiplicidade de abordagens que oferece.

De acordo com Bungenstab (2007) para um jogo ser sucesso no mercado, é necessário não só uma boa equipe de desenvolvimento e capital financeiro, mas também um bom planejamento, visando seu público alvo, e a plataforma para qual o jogo será desenvolvido.

Os jogos eletrônicos são classificados em categorias, de acordo com o gênero e tipo do jogo. Antes de mostrar as etapas de desenvolvimento de jogos, considera-se importante destacar as categorias dos mais populares entre os jogos atuais e uma descrição de cada uma delas.

2.1. CATEGORIAS

- *RPG*: que significa *Role Playing Game*, termo em inglês, que quer dizer, jogo de representação. No RPG o jogador assume o papel de um personagem em que, o este toma as decisões para a continuação da história. Cada decisão tomada pelo jogador pode acarretar em mudanças significativas no desenrolar da história. Esse jogo pode se passar em um mundo alternativo, medieval, futurístico ou uma mistura de ambos. Exemplo de famosas serie de jogos RPG *Breath of Fire (Capcom)* e *Final Fantasy (Square Enix)*.
- *Estratégia*: são jogos que, em geral, exigem do jogador pensamentos estratégicos e táticos para concluir-lo. Tem-se como exemplo o jogo *Warcraft III (Blizzard)*, em que a estratégia é controlar exércitos e armar esquemas táticos para se ter a vitória.
- *Corrida*: são jogos que antigamente eram apenas desenhos que se pareciam com os de um carro. Atualmente esses jogos possibilitam que o jogador escolha um carro, e o customize conforme suas preferências, para aumentar a potência do motor e competir em corridas. Um exemplo desse tipo de jogo é o clássico *Need for Speed (EA Games)*.
- *Shooter*: jogos do tipo “primeira pessoa”, ou seja, o jogador apenas visualiza a arma do personagem. Jogos desse tipo, na maioria das vezes, são jogos de guerra, em que os personagens são policiais contra terroristas e vice-versa. Um exemplo de jogo desta categoria, muito famoso, é *Counter Strike (Valve)*.
- *Luta*: esse tipo de jogo consiste em dois ou mais jogadores lutarem entre si, numa espécie de torneio. No início do jogo o jogador escolhe um personagem para ser o seu lutador no torneio. As séries *The King of Fighters (SNK)* e *Street Fighter (Capcom)* são bons exemplos de jogos desta categoria.
- *Aventura*: são jogos que tem uma história muito parecida com os jogos de RPG, mas, ao contrário desses jogos, a ênfase é dada ao enredo e não a parte gráfica ou ação. Normalmente são jogos em que se devem resolver enigmas e quebra-cabeças. Exemplo *Full Throttle (LucasArts)*.

- *MMO*: é uma sigla em inglês que significa *Multi Massive Online*, que significa Jogo Massivo Multiusuário Online. Esta categoria de jogo online suporta centenas, ou até dezenas de milhares de jogadores simultaneamente através da internet. Um exemplo é o jogo *World of Warcraft (Blizzard)*. Este por ser um RPG é chamado de MMORPG.
- *Simulação*: são jogos que simulam a vida real, com gráficos impressionantes, como nos jogos de corrida atuais. Os simuladores vão desde carros até aviões. Pode-se citar *Flight Simulator (Microsoft)* como exemplo de simulador de voo.
- *Plataforma*: são jogos em que o personagem aparentemente anda pela tela, mas na maioria das vezes o personagem fica no centro da tela e o cenário de fundo é que se movimenta. Exemplos desse tipo de jogos são os clássicos como *Super Mario Bros (Nintendo)* e *Sonic (Sega)*.
- *Esportes*: são simuladores de esportes em geral: basquete, futebol americano, hóquei, baseball, futebol, e outros tipos de esportes. Como exemplo de um jogo de futebol tem-se *Winning Eleven (Konami Sports)*.
- *Tabuleiro*: são jogos que possuem uma mecânica similar ao xadrez, damas, e outros jogos clássicos de tabuleiro. Esses jogos são baseados em turnos, ou seja, um jogador só poderá jogar após o outro executar algum movimento, e vice-versa.

2.2. PÚBLICO ALVO

Ao desenvolver um jogo, é preciso ter em mente o público alvo, ou seja, quem irá comprar o jogo a ser desenvolvido. A faixa etária dos compradores de jogos varia muito, mas geralmente está entre jovens de 16 a 25 anos. Sendo assim, antes de se produzir o jogo, é preciso definir o público alvo para qual o jogo será desenvolvido, e se poderá ser lançado possivelmente em outros países e em outras línguas, conforme destaca Bungenstab (2007).

Portando definir o público alvo antecipadamente é indispensável, para o sucesso do jogo.

2.3. PLATAFORMA

Definir a plataforma é tão importante quando a definição do público alvo. Saber se o jogo será sucesso em determinado console, ou se fará mais sucesso se lançado para computador, ou até ser lançado em várias plataformas simultaneamente.

Segundo Bungenstab (2007), tem acontecido um grande crescimento na área de jogos para dispositivos móveis, isso dado pelo exponencial aumento no número desses dispositivos ao redor do mundo. Número esse que tende a ultrapassar a quantidade de outras plataformas.

2.4. ETAPAS DE DESENVOLVIMENTO

Segundo Perucia e Berthem (2007), produzir um jogo eletrônico é como produzir um software qualquer. O processo de desenvolvimento é composto pelas fases de análise, planejamento, codificação e os testes, como todo software a ser produzido.

Pode-se afirmar que as etapas de desenvolvimento de um jogo são:

- Confecção do *Design Bible*;
- Criação das Imagens (arte dos personagens e cenários) e Áudio (efeitos de som e trilha sonora);
- Implementação.

2.4.1. Design Bible

O *Design Bible*, é o documento que contém todas as especificações do jogo a ser desenvolvido. Assim como um filme não pode ser gravado antes de ter um roteiro elaborado, um jogo também não pode ser desenvolvido sem ter um roteiro, atores, diretor, equipe de filmagem e edição. Sendo assim não apenas o *Roteiro*, mas também o *Game Design*, o *Game Play* e a *Interface Gráfica*, são as outras partes que compõem o *Design Bible*.

O *Design Bible*, que é como um manual de instruções para os futuros desenvolvedores, segundo Bungenstab (2007). Para este autor o Design Bible é composto elementos a seguir.

2.4.1.1. Game Design

O *Game Design* é a conceituação artística do jogo. É importante que parte deste documento seja escrita por um artista.

O *Game Designer* tem a função de criar personagens, cenários e fases do jogo. Sendo que este profissional pode desempenhar duas funções distintas: *Character Designer* e *Level Designer*.

- *Character Designer*: define o design dos personagens: os atributos físicos, psicológicos e outros. Ou seja, o *Character Designer* é quem modela os personagens.
- *Level Designer*: define os cenários e as fases do jogo, da mesma maneira que o *Character Designer* define os personagens.

Após esta etapa de *Design*, os modelos são passados para os artistas fazerem as concepções dos personagens, das interfaces, dos cenários e das fases do jogo.

2.4.1.2. Game Play

É a definição da jogabilidade do jogo. Isto se entende por regras e o balanceamento delas. Nesta parte do documento, deve ser descrito que o jogo será divertido e irá proporcionar bons desafios. Esta etapa é muito importante principalmente para os programadores, pois irá auxiliar na fase de desenvolvimento, afirma Bungenstab (2007).

Um jogo deve ter uma boa jogabilidade para se ter uma boa aceitação entre os jogadores, ou seja, deve ser divertido de se jogar, e não se tornar muito cansativo.

2.4.1.3. Interface Gráfica

Segundo Bungenstab (2007), este documento pode ser dividido em duas partes, *out game* e *in game*.

De acordo com BRANDÃO (2006).

A interface *out game* é referente aos menus presentes fora do jogo, como o menu inicial, novo jogo e opções. No planejamento desta, uma vez definidas todas as telas, deve-se relacionar as mesmas através de um fluxo de menus. A interface *in game* é aquela que exibe informações sobre o usuário em tempo de jogo. Além do enredo, deve-se ter conhecimento dos elementos e missões que deverão ser expostos e de que maneira isso se dará. (Brandão, 2006, p. 02)

O que se entende a partir desta citação é que a interface *in game* é tudo aquilo que é mostrado no jogo, ou seja, quando o jogador está jogando. E a interface *out game* são os menus e opções, fora do jogo, ou seja, antes do jogo se iniciar.

2.4.1.4. Roteiro

O roteiro de um jogo é similar ao roteiro de um filme e são cada dia mais parecidos, porém diferente em deixar um espaço para a interação do jogador. O roteiro é a narração do jogo. Nesta etapa o roteirista propõe a ideia, os personagens e a história. Com o passar do tempo, os roteiros dos jogos estão ficando cada vez mais envolventes e atraentes para os jogadores. Um bom roteiro aumenta a probabilidade de que um jogo faça sucesso junto ao seu público alvo.

2.4.2. Criação das Imagens e Áudio

Segundo Bungenstab (2007), ao se terminar a etapa do *Design Bible*, o desenvolvimento de um jogo se divide em duas partes distintas: criação artística e codificação, entretanto existindo ainda uma grande ligação entre elas. A criação artística pode ser entendida pela elaboração dos recursos do jogo, ou seja, os elementos que serão usados para seu desenvolvimento: sprites, cenários, terrenos, sons, músicas e arquivos de configuração.

A criação das imagens geralmente é feita por artistas, que são profissionais especializados nessa área. As imagens são criadas a partir dos modelos descritos pelos *Game Designers* no *Design Bible*.

Existem muitos programas específicos para criação e edição de imagens, um dos programas mais utilizados nesta área é o Adobe Photoshop. Mas dependendo do jogo a ser feito e da tecnologia a ser utilizada para o desenvolvimento, são utilizados outros programas. Pode-se citar como exemplo o Adobe Flash, que é amplamente utilizado na criação de imagens para o desenvolvimento de jogos.

Para a criação e edição dos efeitos sonoros e das músicas a serem utilizadas nos jogos, tem-se como exemplo o Sony Sound Forge, um software muito utilizado no cinema, na televisão e em vinhetas de rádios, fonte site oficial: www.sonycreativesoftware.com/soundforgesoftware.

2.4.3. Implementação

Segundo Junior, Nassu e Jonack (2002), o processo de implementação deve ser dividido em três partes: codificação, testes e manutenção. Sendo que a criação de jogos é muito próxima de uma produção artística, e dividir a implementação não será sempre possível.

2.4.3.1. Codificação

Esta fase é onde os programadores vão codificar as funcionalidades do jogo. É muito importante ao se criar o projeto de um jogo, já ter definido a tecnologia de desenvolvimento. A escolha da tecnologia leva em conta alguns fatores, como por exemplo, a plataforma.

Pode-se afirmar que este processo exige grandes conhecimentos e capacidade técnica por parte do programador. Algumas características geralmente são exigidas de um programador de jogos, como a capacidade de se adaptar com facilidade a tecnologias novas, saber trabalhar em projetos não detalhados, otimizar rotinas, e principalmente ter um bom relacionamento com os outros profissionais envolvidos no projeto como artistas, designers e escritores.

A programação de jogos não é algo que se pode chamar de “processo automatizável”, pois geralmente os algoritmos de jogos são verdadeiros desafios para os programadores, que sempre devem estudar novas maneiras de otimização do código.

Os autores Junior, Nassu e Jonack (2002) listam algumas características da equipe de programadores, para que o jogo seja um sucesso:

Integração artistas e designers;

- Conhecimento tecnológico;
- Criatividade;
- Grande capacidade de adaptação;
- Gosto pelo que faz.

2.4.3.2. Testes

Nesta parte do processo, são realizados testes e procurar por falhas, mas não apenas isto, também a jogabilidade e aceitação pelos jogadores devem ser salvos dos Testes. Por este motivo os testes são divididos em todas as fases do jogo.

Os testes, de funcionalidade e de fixação de *bugs* (erros), possuem as mesmas características clássicas da Engenharia de Software. Mas existe outra forma de testes que é particular ao desenvolvimento de jogos, o chamado *playtest*.

No *playtest* os programadores analisam a aceitação do jogo e a reação dos jogadores. O *playtest* é basicamente o jogo sendo testado por jogadores, e em algumas vezes por jogadores especializados, chamados de “*testadores profissionais*”, que além de encontrarem problemas, também mostram sugestões para a resolução dos mesmos.

2.4.3.3. Manutenção

Será difícil um jogo fazer sucesso se os jogadores não puderem jogá-lo. Por isso deve ser além de divertido e de fácil jogabilidade, o jogo deve ser de fácil instalação e totalmente compatível com a plataforma escolhida, para que funcione sem a ocorrência de problemas.

Os jogos online e multiusuários necessitam de manutenção constante, devido ao grande número de jogadores, por isso geralmente esse tipo de jogos tem atualização mensais e em algumas vezes até semanais, para garantir uma boa jogabilidade e correção eventuais erros.

3. PROPOSTA DO SOFTWARE

Este capítulo tem por objetivo, aliar a teoria proposta no capítulo anterior. A análise e o desenvolvimento de um jogo.

O jogo a ser implementado será um jogo de xadrez. A opção pelo o jogo de xadrez se deu pelo fato de ser um jogo clássico, e também por ser um jogo simples e de jogabilidade única que já inspirou a criação de vários outros jogos.

O desenvolvimento do jogo proposto neste trabalho, foi feito com base na metodologia de desenvolvimento de jogos citada neste trabalho.

3.1. PÚBLICO ALVO

Como foi dito anteriormente, a definição do público alvo é indispensável para o sucesso de um jogo.

Este jogo terá como público alvo, crianças e adultos de faixa etária entre 10 e 30 anos, brasileiros, pelo jogo ser feito em língua portuguesa.

3.2. PLATAFORMA

Assim como o público alvo, a definição da plataforma também é muito importante. Sendo assim, este jogo terá como plataforma qualquer celular com tela sensível ao toque e com suporte a Flash Lite 3.1, pode-se citar alguns celulares da Nokia, como por exemplo, *N97*, *5800 Xpress Music*, dentre outros.

3.3. DESENVOLVIMENTO

Com a *Plataforma* e o *Público Alvo* definidos, dá-se início a fase de desenvolvimento, iniciando pela criação do Design Bible, seguido pela criação artística, ou seja, a criação das imagens e dos efeitos sonoros a serem utilizados no jogo, e por fim a implementação.

3.3.1. Design Bible

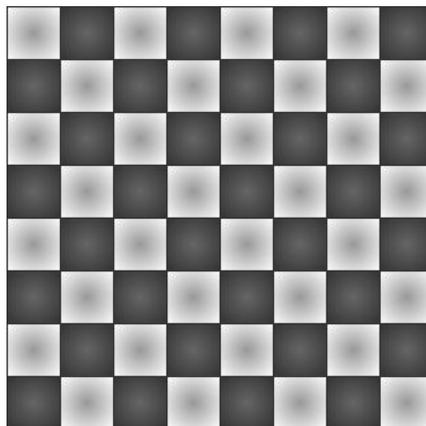
Nesta parte deste trabalho será proposto a análise do jogo a ser desenvolvido, no caso um jogo de *Xadrez*. Como base para esta análise, foi utilizado o livro *Xadrez Básico* (D'Agostini, 1954).

3.3.1.1. Game Design

Os elementos que compõem o jogo de xadrez são: o tabuleiro e as peças.

O Tabuleiro

O tabuleiro consiste em um quadrado, dividido em sessenta e quatro casas, alternando entre as cores: branco e preto. Como ilustra a figura 1.



**Figura 1 - Tabuleiro de Xadrez
(Adaptado de D'Agostini, 1954, p. 18)**

Uma sucessão vertical de oito casas é chamada de *Coluna*. Já uma sucessão horizontal é chamada de *Horizontal*. O tabuleiro é formado por oito *Colunas* e oito *Horizontais*. As sessenta e quatro casas do tabuleiro formam oito *Colunas* e oito *Horizontais*.

Chama-se de *Diagonal*, o conjunto de casas da mesma cor, que cruza o tabuleiro em linha reta, formando um ângulo de 45º em relação às *Colunas* e *Horizontais*.

O tabuleiro deve dispor-se de forma em que cada jogador tenha uma casa branca, na última casa a sua direita.

As Peças

Cada jogador controla 16 peças de uma respectiva cor, sendo oito peões, duas torres, dois cavalos, dois bispos, uma rainha e um rei, totalizando trinta e duas peças. Como ilustra a figura 2.

Pretas		Brancas	
1 Rei		1 Rei	
1 Rainha		1 Rainha	
2 Torres		2 Torres	
2 Cavalos		2 Cavalos	
2 Bispos		2 Bispos	
8 Peões		8 Peões	

Figura 2 - As Peças do Xadrez
(Adaptado de D'Agostini, 1954, p. 19)

Movimento das peças

Cada peça tem um movimento próprio no tabuleiro. Que faz cada peça ter seu papel na estratégia do jogo.

- **Peão**

O peão se movimenta somente nas verticais e para frente. O peão é a única peça que não se movimenta para traz e seu movimento é limitado a uma casa por jogada, somente quando o peão esta em sua posição inicial pode-se optar por movê-lo duas casas de uma só vez, como ilustra a figura 3.

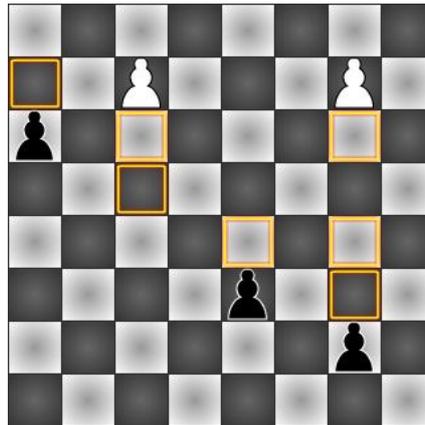


Figura 3 - Movimento do Peão
(Adaptado de D'Agostini, 1954, p. 22)

- **Torre**

A torre se movimenta pelas horizontais e verticais, para frente ou para traz, sem limite de casas, como ilustra a figura 4.

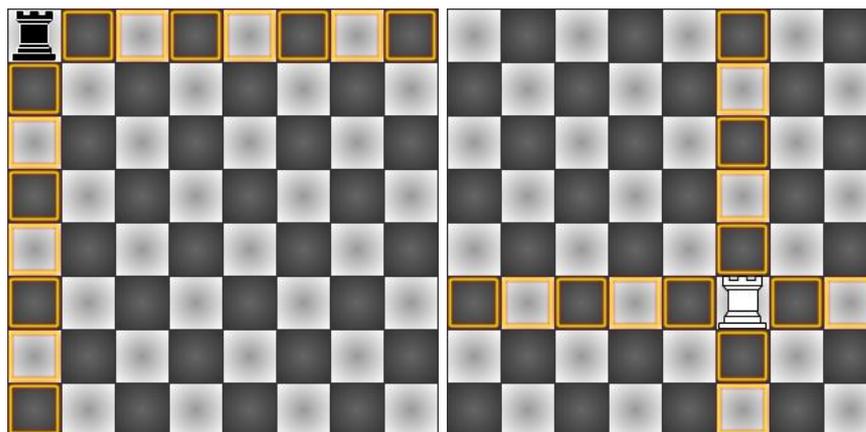


Figura 4 - Movimento da Torre
(Adaptado de D'Agostini, 1954, p. 20)

- **Bispo**

O bispo se movimenta somente nas diagonais do tabuleiro, sem limites de casas. O bispo nunca troca a cor da sua casa, cada jogador tem um bispo que se inicia em uma casa branca e em uma casa preta. A figura 5 ilustra o movimento do bispo.

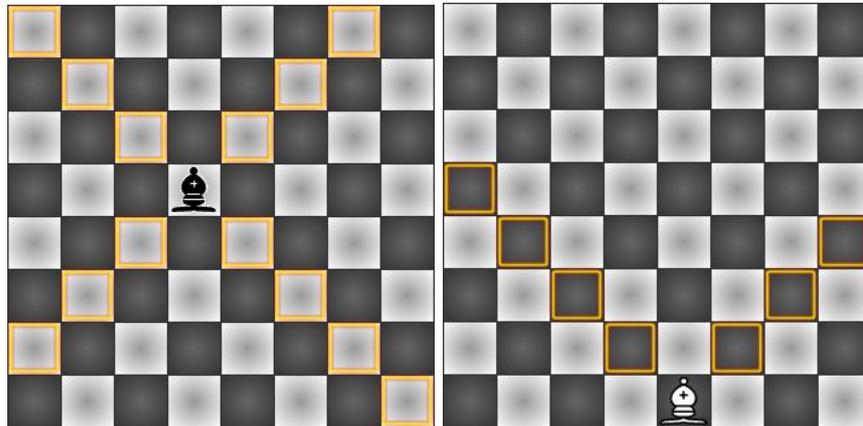


Figura 5 - Movimento o Bispo
(Adaptado de D'Agostini, 1954, p. 20)

- **Cavalo**

O cavalo se movimenta uma casa como torre e uma casa como bispo, ou seja, uma casa para qualquer direção e uma na diagonal, fazendo um movimento em **L**, como ilustra a figura 6.

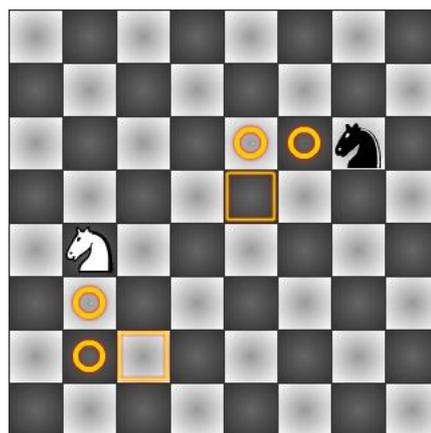


Figura 6 - Movimento do Cavalo
(Adaptado de D'Agostini, 1954, p. 22)

- **Rainha**

A rainha se move como o bispo e a torre juntos, ou seja, se move pelas horizontais, verticais e diagonais, sem limites de casas, como ilustra a Figura 7.

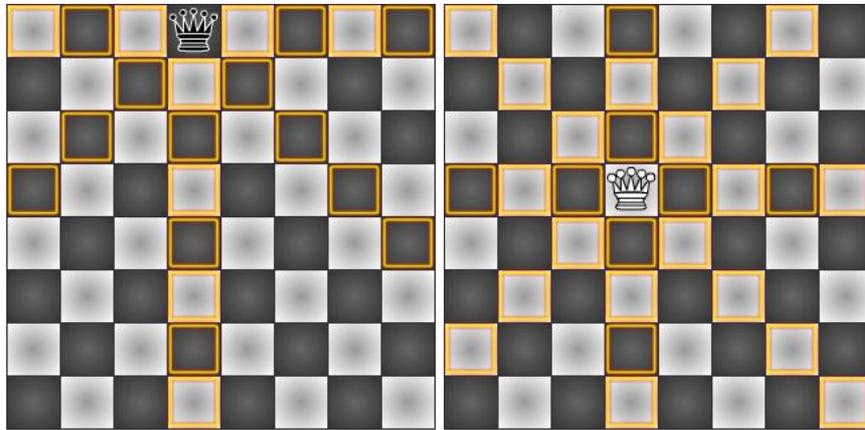


Figura 7 - Movimento da Rainha
(Adaptado de D'Agostini, 1954, p. 21)

- **Rei**

O rei movimenta-se de maneira similar a rainha, mas com o movimento limitado a uma casa por vez, como ilustra a Figura 8.

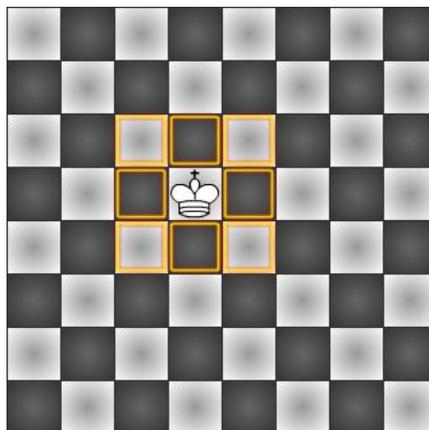


Figura 8 - Movimento do Rei
(Adaptado de D'Agostini, 1954, p. 21)

Regra de Captura

O jogo consiste em capturar as peças adversárias, isso ocorre quando uma peça se movimenta para uma casa já ocupada com uma peça adversária, esse movimento sendo chamado de captura, tomada ou então “comer” uma peça, como já citado acima. A única exceção de captura é o peão, que não captura as peças da mesma maneira que seu movimento. O peão se movimenta somente seguindo em frente, já o movimento para capturar peças adversárias é em diagonal também para frente. Como ilustra a Figura 9, o peão preto pode capturar algum dos dois peões brancos por estar nas diagonais a sua frente, ou então movimentar-se uma casa para frente e não realizar a captura de nenhuma peça.

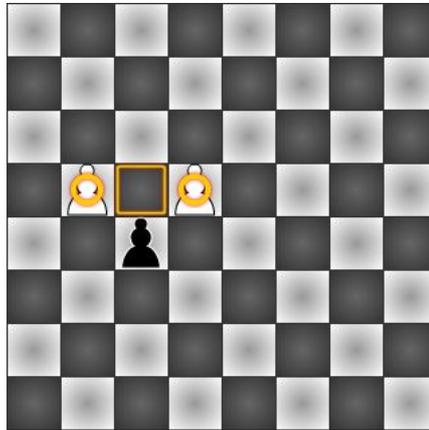


Figura 9 - Regra de Captura do peão
(Adaptado de D'Agostini, 1954, p. 27, p. 28)

Limite no Movimento das Peças

As peças têm um limite quanto ao seu movimento. Em alguns casos uma peça pode ficar sem movimento por existir outras peças impedindo esta peça de se movimentar, podem ser peças da mesma cor ou adversárias. Assim como a figura 10 ilustra, a torre preta está em sua posição inicial, e com peças da mesma cor ao seu redor, por isso está com seu movimento limitado.

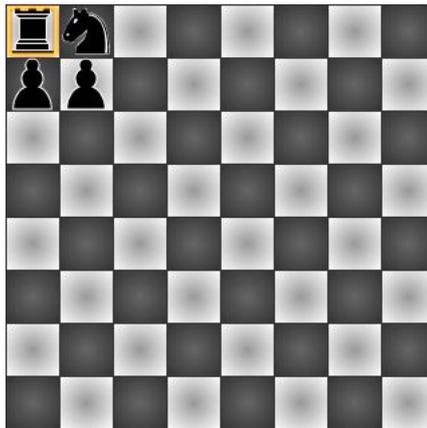


Figura 10 - Exemplo de Limite de Movimento 1

Por exemplo, um peão que fica frente a frente com outro peão, nenhum dos dois peões poderá se mover, até que apareça alguma peça em uma das diagonais à frente, para que estes realizem a captura e assim ficar livre novamente para se movimentar. Vejamos um exemplo na figura 11.

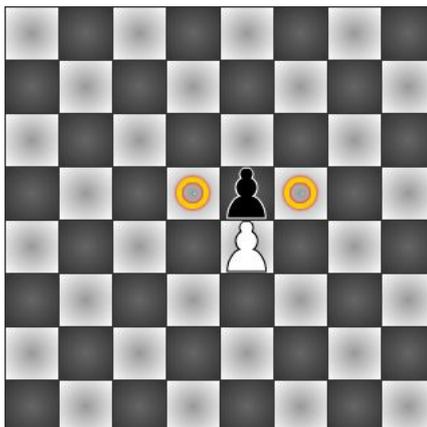


Figura 11 - Exemplo de Limite de Movimento 2

Se a peça que está no caminho for uma peça da mesma cor, o movimento fica limitado, se for uma peça adversária, pode-se capturar esta peça. Nenhuma peça

pode saltar sobre outra peça para se movimentar, como ilustra a figura 12, a torre preta deseja se movimentar para a casa selecionada em amarelo, mas existe um peão da mesma cor interrompendo o caminho, por isso o movimento não pode ser realizado.

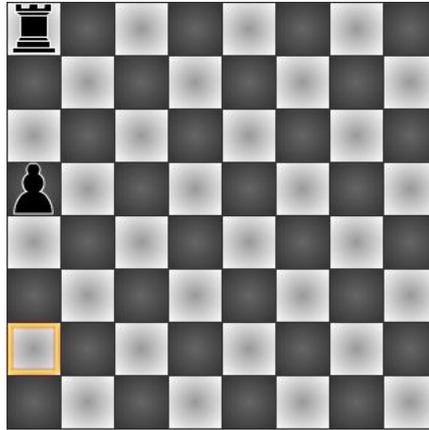
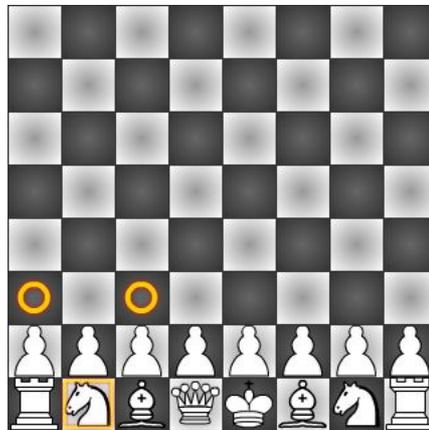


Figura 12 - Exemplo de Limite de Movimento 3

O Cavalo é Exceção

O cavalo é a única peça que faz exceção quanto a seu movimento, podendo assim “saltar” por cima das demais peças. Como mostra a figura 13.



**Figura 13 - Exemplo de Movimento do Cavalo
(Adaptado de D'Agostini, 1954, p. 26)**

Xequê

Diz-se que um rei está em xeque, quando o mesmo está sob um ataque. O xeque é uma advertência do adversário para o rei, dizendo que o rei está prestes a ser atacado, por isso o rei atacado deve se retirar da posição de ataque do adversário. Na figura 14 o rei branco está atacado por quase todos os lados, exceto pela casa selecionada em amarelo ao lado esquerdo do rei, que é a única opção de saída para o rei.



Figura 14 - Rei em Xeque
(Adaptado de D'Agostini, 1954, p. 30)

Quando o rei está em xeque existe três maneiras de procedência:

- Mover o rei para uma casa que não esteja sob o alcance do ataque do adversário.
- Proteger o rei com alguma outra peça, movendo a peça defensora no caminho entre o rei e o atacante.
- Capturar a peça atacante.

Interface *out game*

A interface *out game*, são os menus, opções do jogo, enfim, toda que o usuário ira interagir fora do jogo, ou antes, de se iniciar o jogo.

Ao se dar início ao jogo, o jogador verá uma rápida apresentação sobre os créditos do jogo, após isso aparecerá o *Menu Principal*, este contendo as opções:

- Jogar

Com esta opção dá-se início a jogo, o jogador com a cor branca sempre faz a primeira jogada.

- Sobre

Esta opção mostra os créditos do jogo.

- Ajuda

Esta opção mostra o básico das regras do xadrez, um pouco da jogabilidade, jogadas especiais.

- Sair

Esta opção fecha o jogo.

Interface *in game*

A interface *in game*, é o jogo propriamente dito, tudo que usuário vê e interage enquanto esta jogando.

Após o início do jogo, as peças são posicionadas em suas respectivas casas no tabuleiro. Na tela do jogo estarão o tabuleiro com as peças posicionadas, um botão para voltar ao *Menu Principal*.

3.3.1.4. Roteiro

O jogo de xadrez, por ser um jogo de tabuleiro, envolve somente estratégia, por isso não possui um roteiro, como um jogo de RPG por exemplo.

O que se pode citar como roteiro para o jogo de xadrez é que dois jogadores vão se enfrentar, em que o único objetivo mutuo é capturar o rei adversário.

3.3.2. Criação das Imagens

Com o *Design Bible* acabado, o desenvolvimento se divide em duas partes, a criação dos recursos gráficos do jogo, e na implementação.

Para desenvolvimento dos recursos gráficos deste jogo, será utilizado o *Adobe Flash CS4 Professional*, sendo o mesmo programa utilizado para a programação do jogo.

As imagens serão feitas em gráficos vetoriais, por terem um tamanho reduzido se comparado a outros formatos de imagens como, por exemplo: *jpeg* e *png*. E por serem criados pelo próprio Flash CS4.

Por ser um jogo para plataforma *Mobile*, não serão usados efeitos sonoros, para o jogo ficar o quanto mais compacto possível.

3.3.3. Implementação

A tecnologia a ser utilizada para a implementação, será o Flash Lite 3.1. O programa utilizado para o desenvolvimento nesta tecnologia será o *Adobe Flash CS4 Professional*, que utiliza a linguagem ActionScript 2.

3.3.3.1. Codificação

Para o desenvolvimento será usado a linguagem ActionScript 2, um poderosa linguagem baseada em ECMA Script, sendo uma linguagem orientada a objetos.

A codificação seguirá o seguinte método de programação:

- Uma matriz para controlar o tabuleiro
- Um evento para capturar a posição do mouse no momento do clique, para calcular em qual casa foi clicado
- Um método para o calculo da posição do tabuleiro em que o mouse foi pressionado

3.3.3.2. Testes

Os testes são feitos utilizando o emulador, disponibilizado pela própria Adobe, para o desenvolvimento *Mobile*. O *Adobe Device Central CS4*, um programa em que mostra a interface do celular e simula as aplicações na mesma interface do celular.

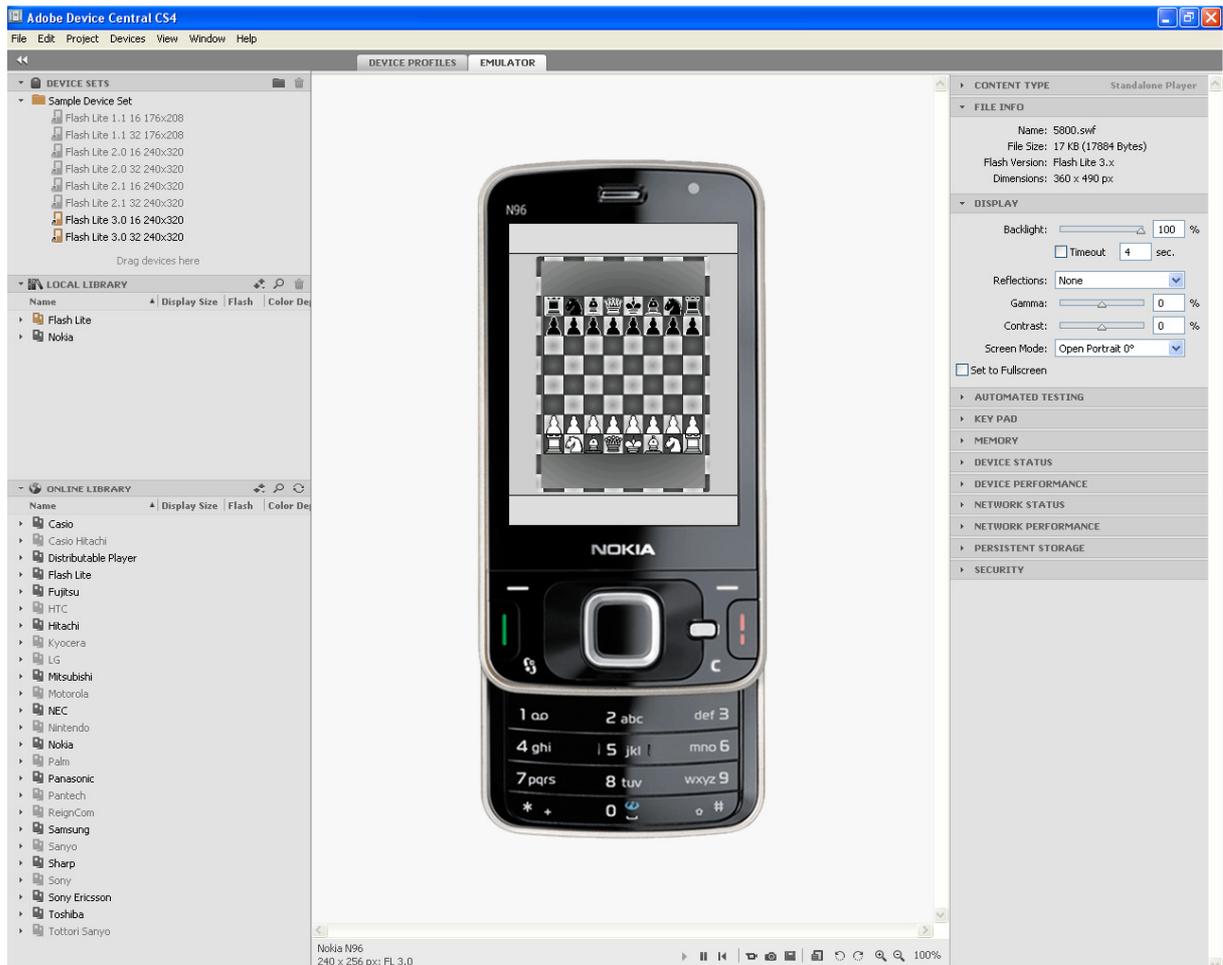


Figura 16 - Adobe Device Central

A figura 16 mostra o emulador usado para fazer os testes das aplicações desenvolvidas com o Flash Lite.

3.3.3.3. Manutenção

As manutenções da aplicação serão feitas apenas como melhorias e atualizações. Já que o jogo desenvolvido é totalmente compatível com a plataforma.

3.4. FLASH LITE

Flash Lite é o versão Mobile do Flash Player e está disponível em três versões:

- Flash Lite 1.x, baseado no Flash Player 4,
- Flash Lite 2.x, baseado no Flash Player 7,
- Flash Lite 3.x, baseado no Flash Player 8.

Flash Lite fornece scripts de animação e interatividade do usuário. O mundo Mobile irá desempenhar um papel enorme no futuro do Flash, nada atualmente rivaliza o Flash em termos de desenvolvimento e de valor acrescentado em aplicações e produtos. Felizmente, se tem a chance de pegar essa onda no início, quando a curva de aprendizado é menor e as oportunidades estão ainda a ser aproveitadas. Vendo que o mercado do Flash Lite está ainda se iniciando.

Segundo a página do Flash Lite no site da Adobe Systems, www.adobe.com, o conteúdo criado com a tecnologia Flash atinge mais de 98% dos computadores com Internet. Hoje são mais de 30 milhões de arquivos SWF online. E mais de 80% dos vídeos na Internet está no formato FLV (Flash Vídeo), formato muito utilizado em sites como Youtube, MySpace, Google Video.

São mais de 400 modelos de dispositivos móveis com o Flash Lite. De acordo com a Strategy Analytics, o número de dispositivos com Flash Lite, atingirá um bilhão em 2009 e mais de 2,5 bilhões até o final de 2010, fonte: site oficial - www.adobe.com/products/flashlite.

3.4.1. ActionScript

O ActionScript sofreu muitas transformações e mudanças, desde que foi introduzido pela primeira vez com o Flash 2 em 2000. A Macromedia adicionou suporte para novas construções, objetos e propriedades com cada versão. A nova cara do ActionScript é uma das mais limpa, mais rápido. Com o Flash 4, o ActionScript deu aos projetistas a capacidade de adicionar lógica para os seus sites e começaram a usar a matemática e física, para fazerem paginas de conteúdos dinâmicos. Este lançamento foi seguido pelo ActionScript 1, para o Flash 5, uma linguagem de programação flexível, baseado em ECMA Script Language Specification, edição 3 (ECMA-262), a especificação da linguagem usada também pelo Java Script 1.5. Com base nesta norma significava que os desenvolvedores poderiam rapidamente se acostumar com a sintaxe. Eles logo começaram a desenvolver extensões para o núcleo de objetos, para oferecer mais funcionalidades.

O ActionScript 2 levou os designers ainda mais longe, permitindo uma verdadeira programação orientada a objetos, com classes, herança, interfaces, exceções e níveis de acesso para os métodos e variáveis de membro, tornando-o mais familiar para programadores Java e C#.

O ActionScript 2 é uma implementação que permite plenamente programação orientada a objeto. A maior parte do código escrito é visto como Java, salvo algumas poucas diferenças de sintaxe. Muitas das práticas associadas com desenvolvimento Java ou C#, também podem ser portadas para ActionScript. Na verdade, o AS2 oferece uma nova maneira de trabalhar. Sendo possível a importação de código separado. Como arquivos, desde o Flash 5, mas com AS2 você pode construir grandes bibliotecas de código reutilizáveis que podem ser compartilhados entre muitos desenvolvedores e projetos, gerenciados com sistemas de controle de origem, e estendido, sem medo de que funcionalidades subjacentes sejam alteradas, diz Legget, Boer e Janousek, 2006.

4. SOFTWARE XADREZ

Neste capítulo será mostrado o jogo desenvolvido utilizando a metodologia de desenvolvimento de jogos e a tecnologia proposta neste trabalho.

O jogo foi desenvolvido utilizando a tecnologia Flash Lite 3.0, e o software de desenvolvimento utilizado, o *Adobe Flash CS4 Professional*.

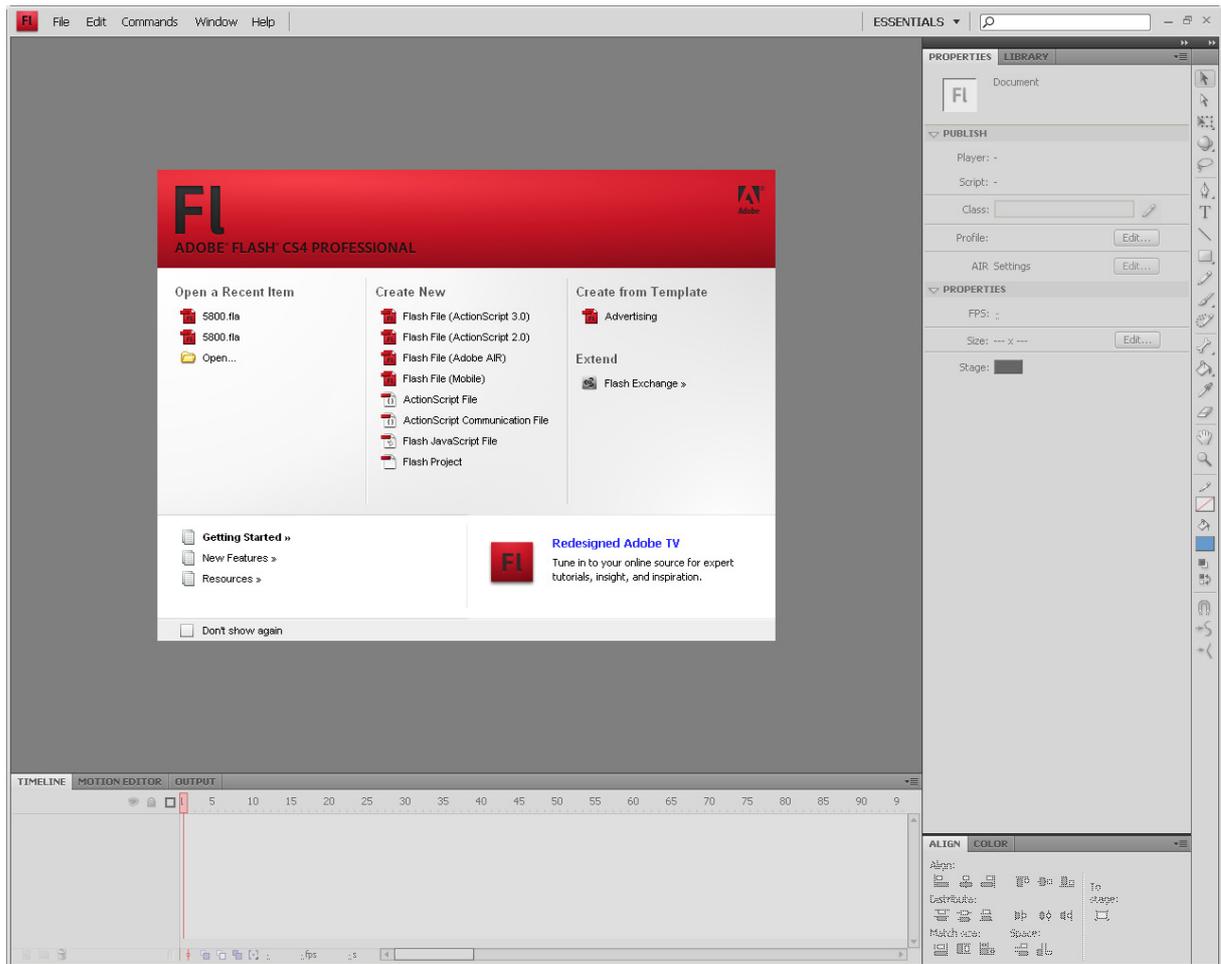


Figura 17 – Adobe Flash CS4 Professional

A figura 17 ilustra a interface inicial software de desenvolvimento o *Adobe Flash CS4 Professional*.

Os elementos de um tabuleiro de xadrez são: as peças e o tabuleiro. Para controlar o tabuleiro foi usada uma matriz de dimensões 8x8. Sendo que cada posição da matriz pode conter uma peça ou caso contrario não contendo nada, a casa esta vazia.

As peças são *MovieClips* (clipes de filme criados com o Flash), mais chamados de *MC*, que são inseridos dinamicamente na palco do Flash. O comando `attachMovie` adiciona os MC ao palco.

Exemplo:

```
_root.attachMovie ("mc","mc",this.getNextHighestDepth (), {_x:10, _y:10});
```

Assim são adicionados ao palco o tabuleiro e as peças, no início do jogo.

Quando uma peça for tomada, é necessária retirá-la do palco, ou seja, remover o *MC* do palco. É preciso usar o comando `removeMovieClip`, para remover um *MC* do palco.

Exemplo:

```
mc.removeMovieClip();
```

4.1. MENU

Ao se iniciar a aplicação a primeira tela a ser exibida é o *Menu* de opções.

As opções são:

- **Jogar**

Inicia o jogo. Com a opção de escolha de lado, branca ou preta.

- **Ajuda**

Mostra as principais regras do jogo de xadrez.

- **Sobre**

Mostra os créditos do jogo, o desenvolvedor e agradecimentos.

- **Sair**

Fecha o jogo.

Como ilustra a figura 18, a tela inicial da aplicação.



Figura 18 – Tela Inicial do Jogo

4.2. JOGO

Ao dar início a partida de xadrez, as peças são posicionadas no tabuleiro, e o jogador branco realiza o primeiro movimento.

Quando as peças são capturadas, elas deixam o tabuleiro, e o seu respectivo *MC* é removido do palco.

Ao se clicar em uma peça para movimentá-la, ela é selecionada e são destacadas as casas que em o movimento é possível. Como ilustra a figura 19, o bispo com suas possíveis casa para se movimentar destacadas em bolinhas amarelas.

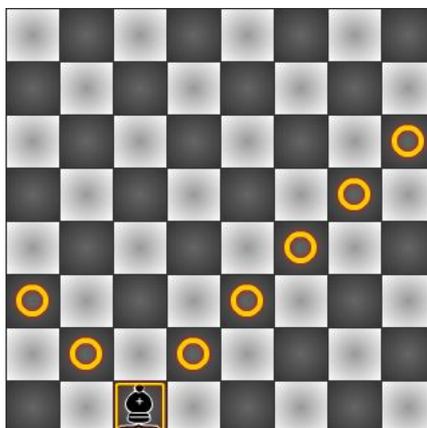


Figura 19 – Destaque das Casas de Possível Movimento

Para realizar o cálculo da casa selecionada, foi usada um método que recebe como parâmetro os valores das posições do eixo **X** e **Y** do clique na tela do dispositivo, e retorna o número da casa do tabuleiro em que foi pressionada a tela. Melhor dizendo, o método retorna a posição da matriz de controle do tabuleiro em que a tela fora pressionada.

Método `getPosition`:

```
function getPosition (numx:Number, numy:Number):Number {
    var posMat:Number;
    var linePos:Number;
    var colPos:Number;
    if ((numx >= minx && numx <= maxx) && (numy >= miny && numy <= maxy)) {
        linePos = Math.floor ((numy - miny) / 40);
        colPos = Math.floor ((numx - minx) / 40);
        posMat = (linePos * 8) + colPos;
    } else posv = -1;
    return posMat;
};
```

O evento de clique na tela é capturado através de um evento chamado *listener*. Esse evento está associado à tela, e captura todas as ações feitas na tela do dispositivo, como clique, movimento, etc. Este sendo adicionado como evento do mouse, mas no dispositivo a tela sensível ao toque que executa a função como mouse.

Exemplo:

```
var mouseListener:Object = new Object ();
Mouse.addListener (mouseListener);
mouseListener.onMouseDown = function () {
    //corpo da função é implementado aqui
};
```

A primeira linha cria uma variável do tipo *Object*, para ser o *listener* do mouse. A segundo adiciona a variável do tipo *Object* como um listener do mouse, com o comando *addListener*. O restante do código é a função *onMouseDown*, que

responde pelos eventos de quando o mouse for pressionado, que no caso será a tela a ser pressionada.

Para fazer interpolação de movimento, alterar valores de um ponto inicial a um ponto final, dos *MC* no palco, foi usada uma biblioteca de tweening, chamada TweenLite. Esta biblioteca fora criada pelo desenvolvedor e evangelista do ActionScript, Jack Doyle, que criou inicialmente a classe *TweenLite*, para acelerar seu processo de desenvolvimento para um cliente, visando uma melhora na performance em relação à classe nativa do Flash, a Tween Adobe, fonte site oficial: blog.greensock.com.

Quando uma peça é movimentada, a peça “*anda*” de sua casa inicial para a sua casa de destino. Esse movimento é mostrado para o jogador utilizando a biblioteca de tweening citada acima a *TweenLite*. Em que com uma linha de código é possível realizar interpolação de movimento, o que dá um efeito de que a peça está realmente “*andando*”.

Exemplo:

```
TweenLite.to (mc, 1, {_x:40, _y:40});
```

5. CONCLUSÃO

O mercado *Mobile* tende a crescer e ultrapassar em numero as outras plataformas, segundo expectativas de mercado. Observando este crescimento, investir tempo e dinheiro em tecnologias para tal, pode e deve render lucros, não falando em valores econômicos, mas em valores profissionais e acadêmicos.

O Flash Lite é uma tecnologia que vem crescendo junto com o mercado *Mobile*, e vem se tornando uma tecnologia robusta e de grande potencial, pelo conteúdo Flash ser pequeno em tamanho, numero de bytes, e também não exige muita capacidade de processamento e memória, que são coisas muito limitadas em dispositivos móveis. E também muito utilizada para o desenvolvimento de jogos, pois possui muitos recursos gráficos e métodos prontos, ideais para o desenvolvimento de jogos.

A metodologia de desenvolvimento de jogos citada neste trabalho, leva o desenvolvimento de jogos a um patamar diferente do desenvolvimento de software convencional, pois essa metodologia muda a forma do analista pensar, no caso o *Game Designer*. Já que pensando de forma diferente o *Game Designer*, não se prende a técnicas de engenharia de software, mas deixa a criatividade e a imaginação artística tomarem frente. Dessa forma o desenvolvedor ganha tempo, tendo assim vantagem em tempo de análise e desenvolvimento se comparado a outras metodologias.

Conclui-se que unindo a metodologia de desenvolvimento e a tecnologia citadas neste trabalho, o desenvolvedor ganha muito tempo, pois a metodologia faz com que a análise seja feita rapidamente, e a tecnologia possui uma linguagem de programação poderosa, de fácil uso e com muitos recursos visuais, que são muito utilizados no mundo dos jogos.

REFERÊNCIAS BIBLIOGRÁFICAS

MESQUITA JÚNIOR, D. S. O corpo plugado: interatividade nos videogames. III Seminário Jogos Eletrônicos Educação e Comunicação - Construindo Novas Trilhas, 01, 2005. Campina Grande: UEPB, 2007.
Acessado: 06/2009

BUNGENSTAB, DIOGO A. - Ferramentas para Desenvolvimento de Jogos - 2007
Acessado: 04/2009

PERUCIA, A. S.; BERTHÊM, A. C.; BERTSCHINGER, G. L.; MENEZES, R. R. C.. “Desenvolvimento de Jogos Eletrônicos - Teoria e Prática”. 2007
Acessado em: 06/2009

BRANDÃO, A.; TOBALDINI, M.; TORTELLI, D. M.; BRANCHER, J.. PLANEJAMENTO DA INTERFACE DE UM RPG EDUCACIONAL PARA COMPUTADORES. Em: Gamepad Seminário de Games, Comunicação e Tecnologia, 2006, Novo Hamburgo.
Disponível em: <<http://www-usr.inf.ufsm.br/~brandao/index.html>>
Acessado em: 06/2009

JUNIOR, ADEMAR S. R.; NASSU, BOGDAN T.; JONACK, MARCO A.. Um Estudo Sobre os Processos de Desenvolvimento de Jogos Eletrônicos (Games) (2002)
Acessado em: 05/2009

LEGGET, R.; BOER, D., W.; JANOUSEK, S. Foundation Flash Applications for Mobile Devices. New York: 2006.
Acessado em: 04/2009

JACK DOYLE – Tweenlite, Greensock Tweening Platform
Site oficial: <http://blog.greensock.com/>
Acessado em: 03/2009

FEDERAÇÃO INTERNACIONAL DE XADREZ - FIDE
Site oficial: <http://www.fide.com/>
Acessado em: 08/2009

D'AGOSTINI, DR. ORFEU GILBERTO - Xadrez Básico, Edição Revisada
Acessado em: 08/2009

ADOBE FLASH LITE - Página do Produto no site Oficial

Site oficial: <http://www.adobe.com/products/flashlite/>

Acessado em 03/2009

EDER LIMA - CONHECENDO A TWEENLITE (GREENSOCK TWEENING PLATAFORM) 12 Março 2009

Disponível em: <http://www.ederlima.com.br/blog/conhecendo-a-tweenlite/>

Acessado em: 05/2009

OVI STORE – Loja de aplicativos da Nokia

Site oficial: <https://store.oivi.com/>

Acessado em: 08/2009

SOUND FORGE SOFTWARE – Software para edição e criação de efeitos Sonoros

Site oficial: <http://www.sonycreativesoftware.com/soundforagesoftware>

Acessado em: 05/2009