



Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Gabriel Merêncio dos Santos

Hugo Mitsumori

Leon Henrique Pires de Brum

Lucas Hideki Sakurai

Shadow Struggles 2013

BlackJack

Volume único

São Paulo, 2013



Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Gabriel Merêncio dos Santos 1050371

Hugo Mitsumori 1050125

Leon Henrique Pires de Brum 1052322

Lucas Hideki Sakurai 1050052

Shadow Struggles 2013

BlackJack

Volume único

São Paulo, 2013



Projeto apresentado ao curso de
Ensino Médio integrado ao Técnico em
Programação e Desenvolvimento de
Sistemas do Instituto Federal de São Paulo,
Coordenadoria da Área de Informática,
como requisito parcial da disciplina de
Prática e Desenvolvimento de Sistemas

Professores Orientadores:

Ivan F. Martinez e Renato Fernandez



Resumo

O projeto *Shadow Struggles* 2013 propôs-se a desenvolver um jogo de RPG e estratégia multiplataforma (PC e Android), dando continuidade ao projeto do ano passado de mesmo nome. O projeto atende aos requisitos da disciplina de Prática e Desenvolvimento de Sistemas, lecionada no 4º ano para os alunos da turma do Ensino Médio Integrado ao Técnico em Programação e Desenvolvimento de Sistemas do IFSP. Durante o desenvolvimento, foram aplicados conhecimentos adquiridos ao longo do curso, tanto de programação quanto análise e engenharia de sistemas.

As principais tecnologias e ferramentas usadas no projeto foram a linguagem de programação Java, a IDE Eclipse e a biblioteca de desenvolvimento de jogos multiplataforma LibGDX.

Palavras-chave: Java, LibGDX, Jogo, Estratégia, RPG, Multiplataforma, Android, Desktop.



Abstract

The Shadow Struggles 2013 project proposed itself to develop a multiplatform (PC and Android) RPG and strategy game, continuing the project from last year by the same name. The project fulfills the requisites of the “Prática em Desenvolvimento de Sistemas” (Practice in Systems Development) subject, taught in the 4th year for students at “Ensino Médio Integrado ao Técnico em Programação e Desenvolvimento de Sistemas” (High School with a Systems Programming and Development Technician degree) from IFSP. During the development, knowledge acquired through the course, from programming and systems analysis and engineering, was applied.

The main technologies and tools used in the project were the Java programming language, Eclipse IDE and the multiplatform game development library LibGDX.

Tags: Java, LibGDX, Game, Strategy, RPG, Multiplatform, Android, Desktop.



Lista de Imagens

Figura 1 – Esboço interface Principal.....	18
Figura 2 - Legenda	18
Figura 3 – Tela Inicial.....	20
Figura 4 – Tela Inicial com Perfis.....	21
Figura 5 – Tela Principal	22
Figura 6 – Free Play	23
Figura 7 - Shop.....	24
Figura 8 – Edit Deck	25
Figura 9 – Troca de Perfil.....	26
Figura 10 - Settings.....	27
Figura 11 – Logo inicial.....	68
Figura 12 – Logo proposto.....	69



Índice de tabelas

Tabela 1 - Comparativo	32
Tabela 2 – Tarefas Relatório.....	50
Tabela 3 – Cronograma	65

Sumário

1 Introdução.....	13
2 Descrição Geral do Projeto	14
3 Caracterização do Projeto.....	15
4 Mecânica do Jogo.....	17
4.1 Batalha.....	17
4.2 Campanha.....	19
5 Navegação.....	20
Tela inicial.....	20
Menu principal.....	22
Free Play.....	23
Shop.....	24
Edit Deck.....	25
Troca de Perfil.....	26
Settings.....	27
6 Ferramentas e Tecnologias Utilizadas no Desenvolvimento	28
6.1 Ferramentas.....	28
Eclipse.....	28
LibGDX.....	28
Tiled Map Editor.....	28
Blender.....	28
ObjectAid UML Explorer	28
AmaterasUML.....	28
Window Builder.....	29
FindBugs	29
SlikSVN 1.6	29
6.2 Tecnologias.....	29
Java 6.....	29
JSON.....	29
JUnit	29
Java Logging	29
7 Metodologia de Gestão do Projeto.....	30
8 Análise de Nicho.....	32
9 Histórico das Atividades.....	33
10 Problemas.....	35
Manipulação de dados em arquivo JSON.....	35
Saída de membro da equipe.....	35
11 Escolhas e Descartes	36

11.1 Escolhas.....	36
Mecânica de jogo.....	36
Atualização das bibliotecas do LibGDX	36
Refatoração do Código	36
Encriptação de Arquivos	36
Sistema de <i>logging</i>	36
Teste Unitário	37
Metodologia de desenvolvimento.....	37
Interface	37
ClockingIT.....	37
11.2 Descartes	37
Themable Widget Library (TWL)	37
12 Lista de Recursos Faltantes.....	38
13 Conclusão.....	39
14 Links do Projeto	40
15 Referências bibliográficas	41
Apêndice A – Documento de Aprovação.....	42
Apêndice B – Resumo das Atas de Reuniões.....	47
Apêndice C – Postagens Semanais.....	50
Relatório de atividades (28/04 a 04/05)	55
Relatório de atividades (05/05 a 11/05)	56
Relatório de atividades (12/05 a 18/05)	56
Relatório de atividades (18/05 a 25/05)	58
Relatório de atividades (26/05 a 01/06)	58
Relatório de atividades (02/06 a 08/06)	58
Relatório de atividades (09/06 a 15/06)	58
Relatório de atividades (16/06 a 22/06)	59
Relatório de atividades (23/06 a 29/06)	59
Relatório de atividades (30/06 a 06/07)	59
Relatório de atividades (07/07 a 13/07)	60
Relatório de atividades (14/07 a 20/07)	60
Relatório de atividades (21/07 a 27/07)	60
Relatório de atividades (28/07 a 03/08)	60
Relatório de atividades (04/08 a 10/08)	61
Relatório de atividades (11/08 a 22/08)	61
Relatório de atividades (23/08 a 29/08)	61
Relatório de atividades (29/08 a 05/09)	61
Relatório de atividades (05/09 a 16/09)	62
Relatório de atividades (16/09 a 26/09)	62
Relatório de atividades (27/09 a 03/10)	62
Relatório de atividades (03/10 a 10/10)	63
Relatório de atividades (10/10 a 17/10)	63



Apêndice D – Cronograma de Desenvolvimento do Projeto.....	65
Apêndice E - Plano de Testes.....	66
Apêndice F – Análise de Código.....	67
Apêndice G – Logomarcas do Projeto	68
Apêndice H – Formulário de Aprovação.....	70



Lista de Abreviaturas e Siglas

IDE – Integrated Development Environment/Ambiente de Desenvolvimento Integrado.

IFSP – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

JRE – Java Runtime Environment/Ambiente de Execução Java.

JSON – JavaScript Object Notation/Notação de Objeto JavaScript.

RPG – Role-playing game/Jogo de Interpretação de Papéis.



Glossário

API (Application Programming Interface): Conjunto estruturado de funcionalidades que visa abstrair processos específicos facilitando o desenvolvimento de software.

Bug: Erro ou falha grave que impede um programa de computador de funcionar corretamente.

DAO (Data Access Object): É um objeto que é responsável por realizar a interface entre o banco de dados da aplicação e os demais objetos. Sendo assim, ele é responsável por fazer as chamadas do banco de dados e fornecer métodos para que outros objetos possam lidar com o banco de dados indiretamente.

Framework: É uma estrutura de suporte a partir da qual um projeto de software pode ser organizado e desenvolvido.

Log: Registro de eventos relevantes num sistema computacional.

Plug-in: Programa de computador usado para adicionar funções a outros programas maiores, provendo alguma funcionalidade especial ou muito específica.

Singleplayer: Modo de jogo em que há apenas um jogador, onde qualquer elemento alheio a ele é gerado pelo computador.

Software: Conjunto de programas e informações relacionadas responsáveis por definir operações a serem realizadas por um computador.

1 Introdução

É fato que o mercado de jogos na época informatizada atual é um dos mais movimentados e crescentes. Dezenas de jogos são lançadas a cada ano e as plataformas vão tendo constantes atualizações para melhorar a qualidade e imersão dos jogos das novas gerações.

Um dos mercados mais crescentes atualmente é o de jogos casuais para smartphones e tablets. Por terem simplicidade e diversão, conseguem abranger um mercado consumidor amplo, que inclui pessoas de todas as idades. No entanto, para um público *gamer* mais sério e dedicado, é esperado um tipo de jogo mais completo, com desafios mais diversificados, e temáticas mais dinâmicas.

O *Shadow Struggles* 2013 nasceu da vontade de criar um jogo estratégico com uma alta capacidade de envolver o jogador no contexto do jogo. Para tal, foram identificados três requisitos básicos: oferecer desafios interessantes, manter o jogador imerso no enredo e atingir um público-alvo abrangente. *Shadow Struggles* é, portanto, uma tentativa de aliar esses três fatores.

Seguindo a orientação dos professores da disciplina de PDS, a equipe visa, por meio desse projeto, se desenvolver ao máximo e aprender a lidar com diversos aspectos do desenvolvimento de projetos de software, desde fatores técnicos, quanto criativos e sociais.

2 Descrição Geral do Projeto

A disciplina de PDS tem como trabalho principal o desenvolvimento de um sistema (projeto), baseado em linguagens orientadas a objetos, e que está sujeito à aprovação dos professores avaliadores (clientes).

O objetivo do projeto é o desenvolvimento de um jogo do estilo RPG, em que o jogador explora as áreas do jogo e interage com os elementos através de batalhas, diálogos, missões, e eventos diversos. O jogo é baseado em um história de temática pós apocalíptica que mistura ciência e tecnologia. Ao longo do avanço do jogador na história, haverá decisões a serem tomadas que irão resultar em diferentes caminhos da história, seguindo um estilo de Visual Novel.

As batalhas são baseadas em uma mecânica de Tower Defense, em que o jogador deve defender sua base, ao mesmo tempo em que tenta destruir a base inimiga por meio da invocação de lutadores, efeitos e armadilhas, representados por cartas.

Ao decorrer do jogo, o jogador vai evoluindo o nível do personagem, melhorando os atributos de batalha e tendo a possibilidade de aquisição de novas cartas. Além disso, o jogador poderá montar diferentes estratégias de batalhas através da criação de decks variados com as diferentes cartas obtidas.

O jogador terá toda a experiência de um RPG, com elementos de estratégia e uma grande interatividade com a história. A diversidade de itens e caminhos a serem tomados permitem ao jogador finalizar o jogo de diferentes formas, de modo a aproveitar ao máximo a diversão que o jogo proporciona.

3 Caracterização do Projeto

O que é

O projeto consiste em um jogo para as plataformas Desktop e Android, desenvolvido na linguagem Java, que poderá ser executado em qualquer sistema operacional que seja compatível com o Java Virtual Machine (JVM).

Modo de Funcionamento

O jogo poderá ser jogado por um jogador, sendo possível jogar na modalidade campanha ou na modalidade livre (Free Play).

Temática

Fictícia, futurística.

Premissa do Enredo

Em uma realidade alternativa, a alquimia desenvolve-se a tal ponto que todo o desenvolvimento científico-tecnológico passa por ela. Os humanos então iniciam uma exploração em massa do espaço e estabelecem um governo para gerenciar as metrópoles espaciais, deixando apenas os estudiosos para trás na Terra. Séculos após o início das viagens espaciais, espécies alienígenas colonizadas por terráqueos invadem o planeta desprotegido e tomam o controle para si, forçando os alquimistas humanos a trabalharem em seu prol.

O protagonista é um dos pesquisadores humanos sob o governo dos aliens, um alquimista em treinamento guiado por seu talentoso mentor. Após pesquisas no campo do ocultismo, o mentor abre um portal ligando o mundo real ao Ponto Ômega, a fonte de poder dos alquimistas. Isso provoca uma terrível catástrofe que destrói o laboratório e a cidade na qual ele está localizado, sobrevivendo apenas o mentor e o protagonista. Contudo, o incidente provoca habilidades inesperadas que permitem aos alquimistas darem vida a objetos inanimados.

O mentor passa a usar o conhecimento adquirido no contato com o sobrenatural à sua vantagem, a fim de ascender ao poder. Ao mesmo tempo, alquimistas do mundo todo encaram a situação como uma oportunidade para rebelarem-se contra a

dominação alienígena. O protagonista junta-se a um grupo rebelde, mas busca o poder necessário para impedir o mentor. Ele entra em contato com duas entidades cósmicas supremas do Bem e Mal e deve, então, escolher qual caminho trilhar para chegar ao Ponto Ômega.

Objetivo do Jogo

O Jogador deve explorar o mundo do jogo, interagindo com diversos elementos, como itens, cenas, NPC's, e vencendo batalhas. O personagem vai evoluindo e criando estratégias conforme o progresso do jogo e do enredo. O jogador “termina” o jogo quando chega ao fim de um enredo e vence todos os obstáculos. Devido ao modelo de história com alternativas, o jogador poderá progredir por diferentes caminhos do enredo.

4 Mecânica do Jogo

As duas principais modalidades de jogo são a batalha e o modo campanha/RPG, ambas descritas a seguir.

4.1 Batalha

O jogo é do gênero estratégia em tempo real, mesclando elementos de jogos estratégicos como Plants vs Zombies e Nine Heroes e jogos de tabuleiro (mais notadamente, duelos com cartas).

A tela de batalha está dividida em dois campos, cada qual com uma base a ser protegida e espaços (*slots*) para cartas serem posicionadas pelos jogadores. Há uma “zona neutra”, na qual não é possível posicionar cartas. Cartas podem representar unidades de ataque, unidades de defesa, efeitos especiais ou armadilhas.

O objetivo primário é defender a própria base e invadir a do adversário utilizando as cartas do baralho, mas as condições de vitória podem variar de acordo com a fase. Adicionalmente, o jogador perde quando não é mais possível sacar cartas ou o tempo da partida se esgota e a vitalidade de sua base é menor que a do adversário.

Durante o jogo, cartas podem ser enviadas do baralho à mão do jogador segundo um intervalo fixo de tempo, sendo também disponíveis ao inimigo. Adicionalmente, há um sistema de energia para limitar o uso de cartas: cartas consideradas fortes têm um custo de energia maior, e portanto não podem ser utilizadas com frequência. A energia é recuperada gradualmente durante a partida e também por meio de cartas de suporte.

Esboço da interface principal:

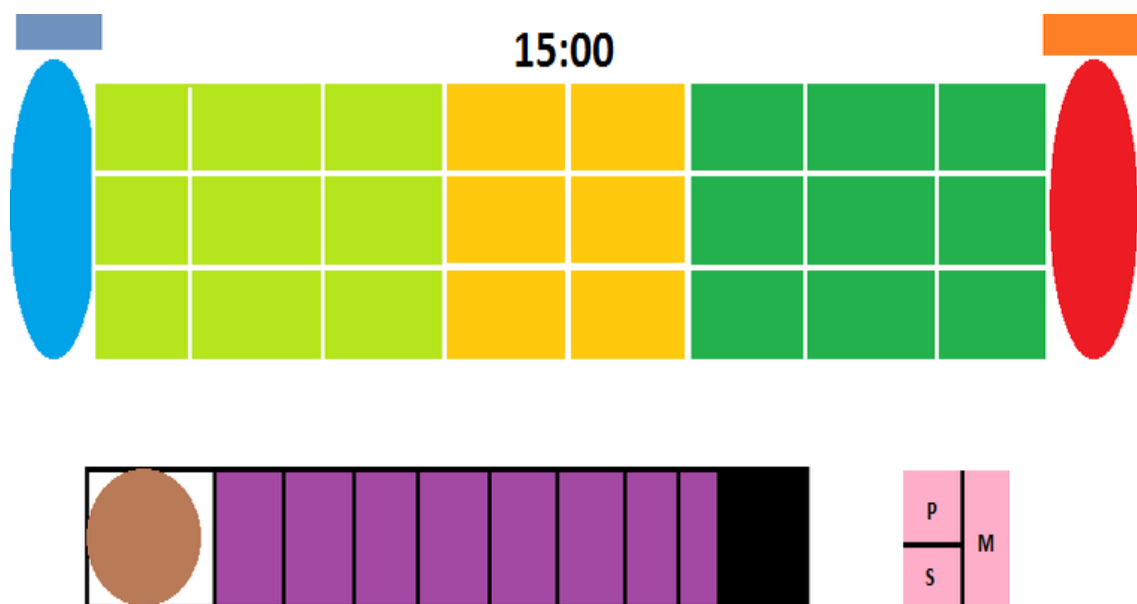


Figura 1 – Esboço interface Principal












Legenda	
	Base do jogador
	Status da base do jogador (Barra vazia = game over)
	Base do inimigo
	Status da base inimiga
	Campo do jogador
	Zona neutra
	Campo do inimigo
	Indicador de energia
	Cartas na mão do jogador
	Baralho do jogador
	Menu lateral: P - pausa o jogo, S - habilita/desabilita som, M - retorna ao menu principal

Figura 2 - Legenda

4.2 Campanha

O modo campanha segue uma narrativa fixa, mas com múltiplas possibilidades de desenrolar dependendo das escolhas feitas pelo jogador. A narrativa mescla um modo de visualização de texto com um RPG 2D.

Neste modo o jogador explora o mundo do jogo por meio de sua personagem. Ao explorá-lo o jogador tem a chance de interagir com inúmeros objetos, ativando mecanismos, abrindo baús, ou entrando em lojas por exemplo.

O objetivo do jogador neste modo é, além de navegar pelos variados ambientes do jogo, seguir o modo história até o final por meio da interação com os itens no mapa, vencendo as batalhas para derrotar seus oponentes.

Neste modo o jogador pode realizar diversas ações adicionais. Além disso, também pode interagir com personagens do próprio jogo, podendo comprar e vender determinados itens (como cartas, habilidades), iniciar batalhas, salvar o jogo, etc.

5 Navegação

Tela inicial

Na tela inicial, o usuário terá duas opções de jogar

- Continuar um perfil salvo;
- Criar um novo perfil;



Figura 3 – Tela Inicial

Caso o usuário escolha a opção de “Continue” aparecerá os perfis já salvos no lado direito.

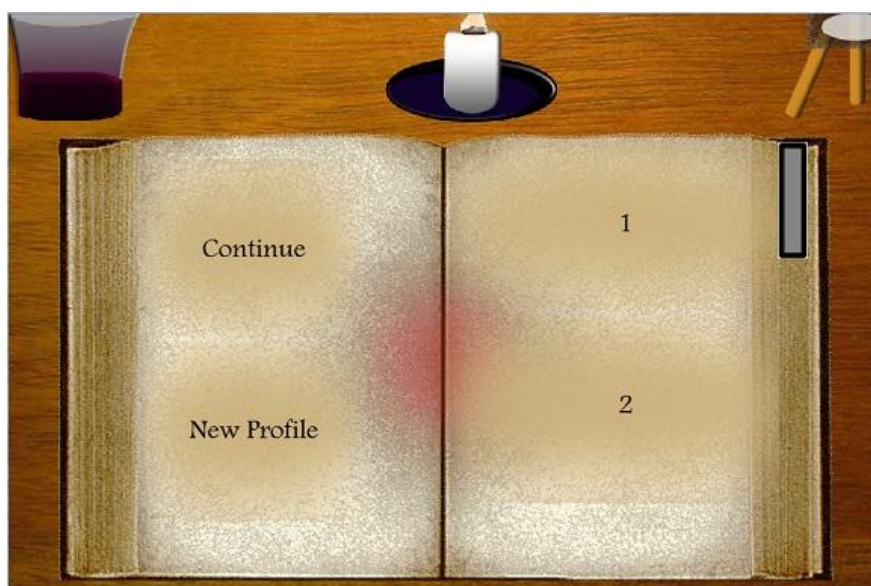


Figura 4 – Tela Inicial com Perfis

Menu principal

Após a tela inicial o menu principal irá dispor as principais opções:

- Modo Campanha;
- Free Play;
- Shop;
- Edit Deck;
- Trocar de perfil (canto inferior esquerdo);
- Configurações (canto inferior direito);

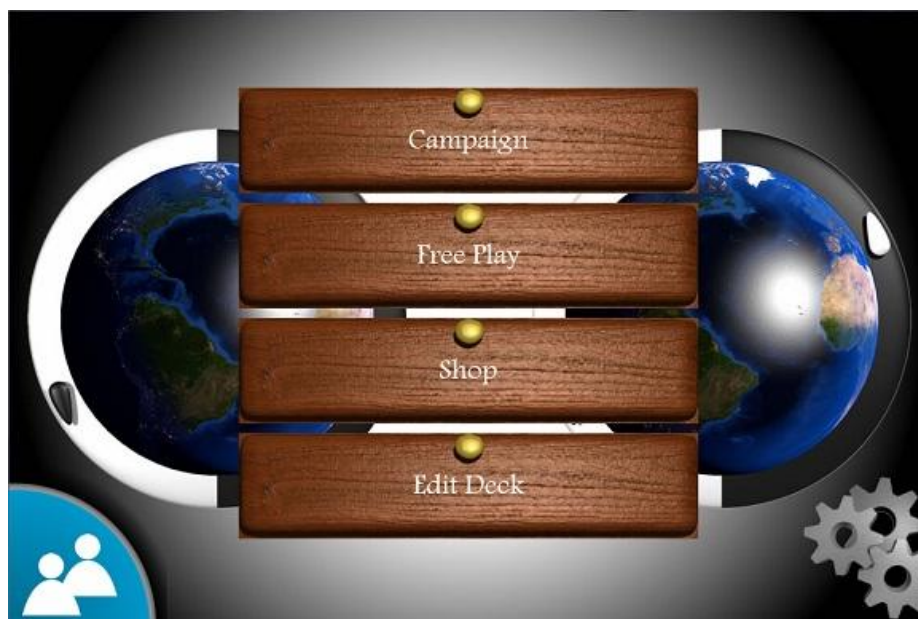


Figura 5 – Tela Principal

Free Play

Nessa tela o usuário pode escolher batalhas específicas para jogar.

- Menu lateral:
 - Select a Chapter: combobox que seleciona o capítulo desejado;
- Batalhas (região ao lado do menu lateral): aparece as batalhas disponíveis do capítulo selecionado e as informações da batalha (ícone, nome e recompensa);
- Setas laterais: navegação entre as páginas que dispõe as batalhas, quando houver mais que nove batalhas.

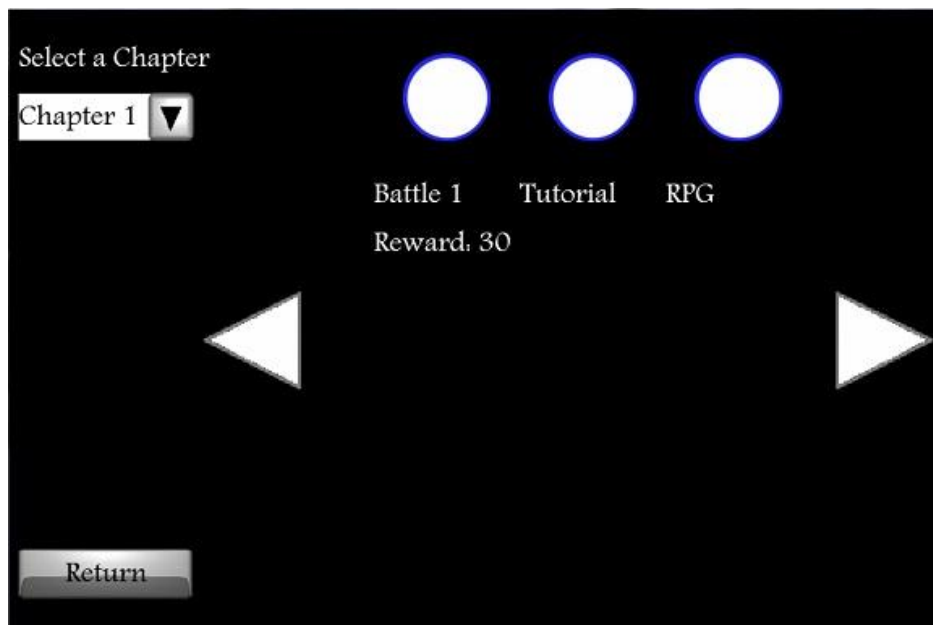


Figura 6 – Free Play

Shop

O Shop (loja) é onde o usuário poderá comprar e vender cartas. Nele é possível comprar diferentes pacotes. Haverá quatro opções:

- Packs: serão pacotes com uma quantidade de cartas e essas cartas serão aleatórias
- Extra: cartas extras
- Cards: compra de cartas avulsas
- Sell: venda de cartas que o usuário possui
- Setas laterais: navegadores entre páginas de cartas, caso haja mais de nove cartas a dispor, as setas irão dispor a página anterior ou posterior.

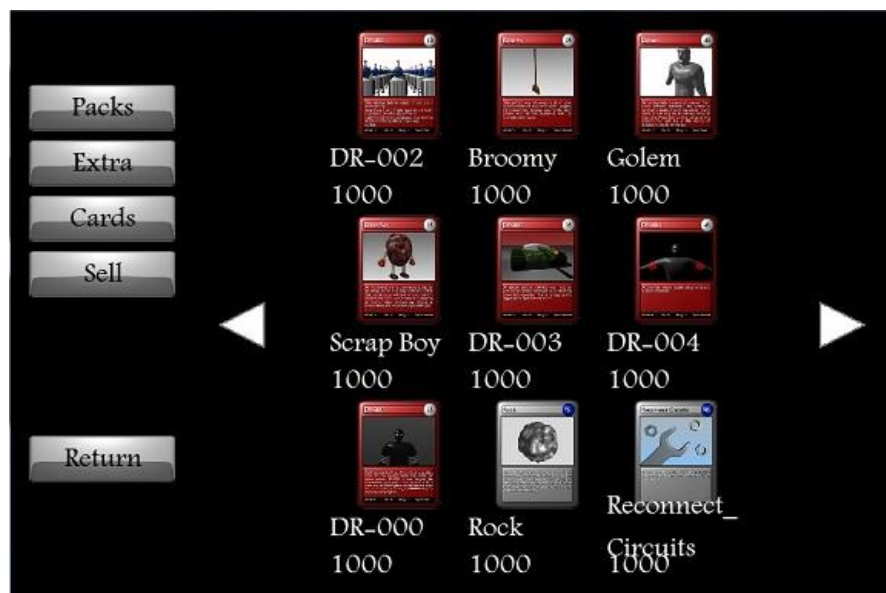


Figura 7 - Shop

Edit Deck

Nessa tela o usuário terá a opção de editar o(s) seu(s) deck(s).

- Menu lateral
 - Combobox: seleciona o deck que será alterado
 - New Deck: cria um novo deck
- Deck (região ao lado do menu lateral): área onde serão mostrados as cartas que o usuário possui no deck
- Available Cards: mostra as cartas disponíveis para serem adicionadas ao deck.

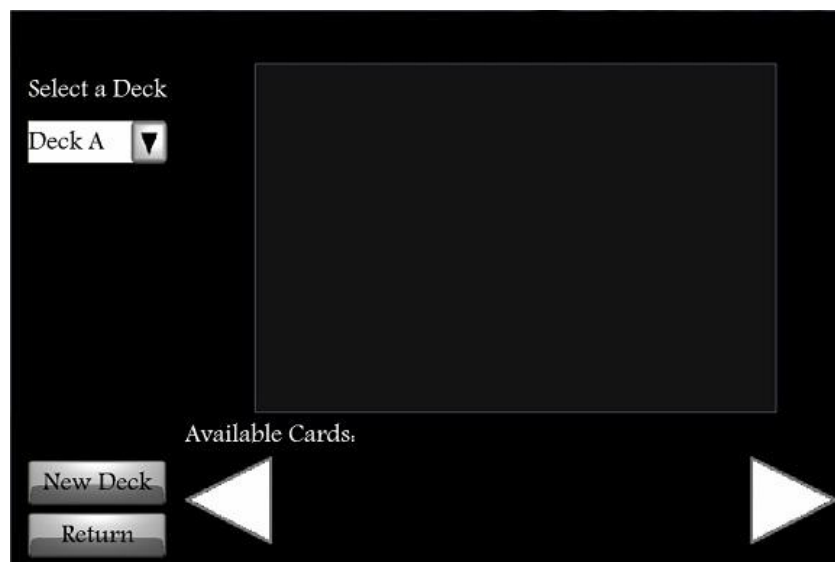


Figura 8 – Edit Deck

Troca de Perfil

Caso o usuário deseje jogar em outro perfil já salvo nessa tela ele terá a opção de realizar a mudança sem precisar reiniciar o jogo.

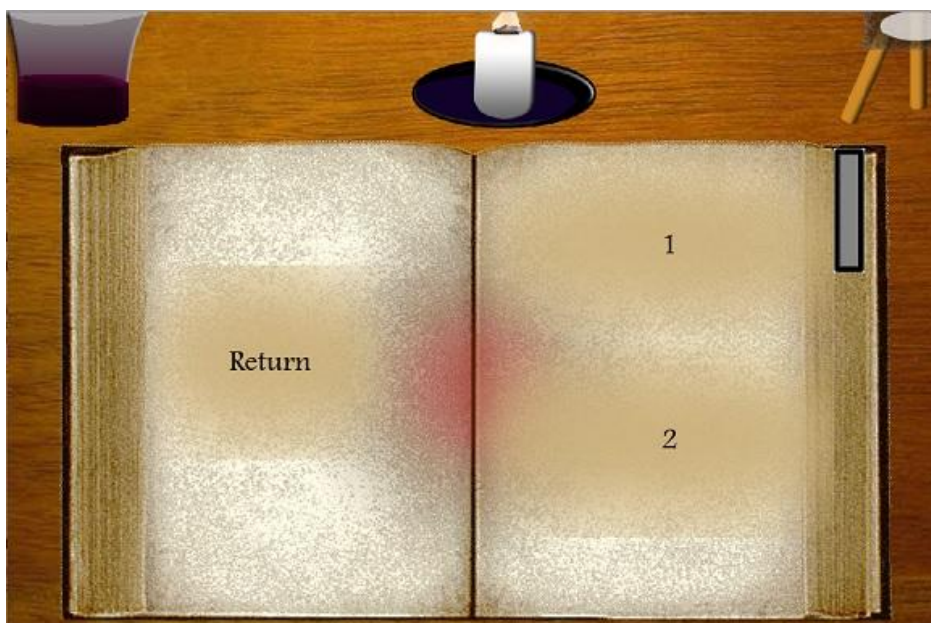


Figura 9 – Troca de Perfil

Settings

No settings (configurações) o usuário terá a seguinte tela

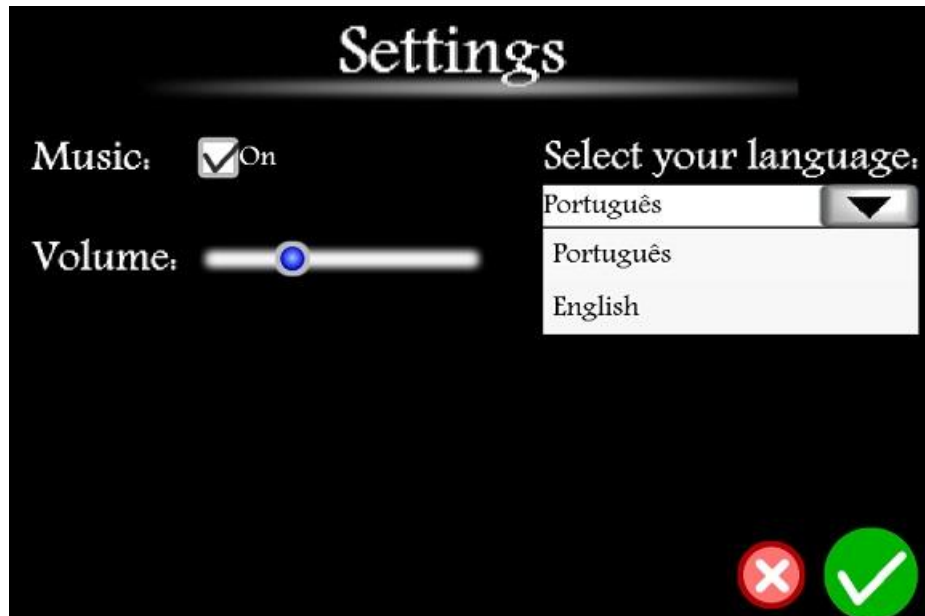


Figura 10 - Settings

Onde:

- Music On/Off: Ativa ou desativa a música;
- Volume: scroll para aumentar ou diminuir o volume
- Select your language: Seleciona a linguagem a ser usada no jogo (português ou inglês);
- Marca de checagem: Salva as alterações feitas;
- Marca X: Cancela as alterações;

6 Ferramentas e Tecnologias Utilizadas no Desenvolvimento

6.1 Ferramentas

Eclipse

IDE (Integrated Development Enviroment) open source para desenvolvimento em Java mais utilizado no mundo. Possui diversos plug-ins para atender as diversas necessidades de programadores.

LibGDX

Aplicação framework de desenvolvimento de jogos em Java com alguns componentes de C/C++ para código dependente de performance. Permite desenvolvimento de jogos para desktop e Android utilizando a mesma base de código.

Tiled Map Editor

Software de edição de mapas do tipo tile.

Blender

Software open source e gratuito de computação gráfica para criação de filmes, efeitos visuais, modelos 3D e videogames.

ObjectAid UML Explorer

Ferramenta de visualização do código Java e bibliotecas em diagrama classes UML e diagramas de seqüências UML que atualizam-se automaticamente à medida que o código é alterado.

AmaterasUML

Plug-in do Eclipse para edição de diagrama de classes UML, e diagrama de seqüências UML.

Window Builder

Plug-in do Eclipse composto por SWT Designer e Swing Designer para criação de aplicações Java GUI.

FindBugs

Programa open source que utilize análise estática para identificar centenas de tipos de potenciais de erros no código Java.

SlikSVN 1.6

Programa utilizado para gerar log do repositório do GoogleCode

6.2 Tecnologias

Java 6

Linguagem de programação orientada a objeto que é compilada para um bytecode que é executado por uma máquina virtual.

JSON

JavaScript Object Notation é um arquivo de texto utilizado para armazenamento de dados de forma simples e estruturada.

JUnit

Framework de teste unitário para programação em linguagem Java. Utilizado principalmente em TDD (Test Driven Development).

Java Logging

Recordação de atividades.

7 Metodologia de Gestão do Projeto

Scrum

Para a gestão no desenvolvimento do projeto, utilizamos a metodologia Scrum, com algumas adaptações às condições que possuíamos.

O Scrum é uma metodologia de desenvolvimento ágil de projetos baseada em ciclos iterativos, denominados de Sprint. Ele é usado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer.

Ao iniciar o projeto, é criado um *Product Backlog*, onde são definidos todos os requisitos do projeto. No início de cada Sprint, é feita uma reunião de planejamento, onde são definidas as principais tarefas do Backlog a serem implementadas na Sprint, de acordo com as prioridades selecionadas pelo *Project Owner*. Em cada Sprint, é criado um incremento do produto, ou seja, um produto funcional e testado com as novas funcionalidades.

Existem 3 papéis a serem cumpridos:

- **Product Owner:** responsável pela visão de negócios do projeto, é ele quem define e prioriza o *Product Backlog*. Na disciplina de PDS, o papel foi cumprido pelos professores Ivan Francolin Martinez e Renato Fernandez.
- **Scrum Master:** é uma mistura de gerente, facilitador e mediador. Seu papel é remover obstáculos da equipe e assegurar que as práticas de *Scrum* estão sendo executadas com eficiência. Inicialmente era cumprido pelo Gabriel Merêncio dos Santos. Após sua saída do IFSP, foi substituído pelo Hugo Mitsumori.
- **Equipe de Desenvolvimento:** responsável por entregar soluções e desenvolver os Sprints. Formada pelos membros: Gabriel Merêncio dos Santos, Hugo Mitsumori, Leon Henrique P. Brum e Lucas Hideki Sakurai.

Os Sprints do Projeto Shadow Struggles 2013 eram definidos mensalmente, com reuniões semanais nas aulas de PDS. Na maioria das vezes, acabávamos por cair no comodismo e ir direto para a continuação do desenvolvimento, sem consultar diretamente os *Product Owners*.

O *Product Backlog* foi definido no início do desenvolvimento utilizando a ferramenta **ClockingIT**, onde realizávamos o controle das tarefas e prazos de cada Sprint. Paralelamente, mantínhamos um cronograma geral em uma planilha do Excel, com o cálculo de pontos da Sprint, velocidades e progresso.

Ao iniciar o segundo semestre, com a saída do Gabriel Merêncio dos Santos do IFSP, o papel de *Scrum Master* passou a ser desempenhado pelo Hugo Mitsumori. Mas apesar disso, o Gabriel Merêncio dos Santos continuou participado da equipe de desenvolvimento, mesmo à distância.

Por decorrência de atrasos e imprevistos, as Sprints do mês de outubro em diante foram destinadas exclusivamente à Documentação do Projeto, pois não estávamos fazendo isso paralelamente ao desenvolvimento das funcionalidades.

8 Análise de Nicho

Atualmente o mercado de RPG encontra-se saturado, devido ao fato de ser um estilo de jogo muito explorado por desenvolvedores e que detém uma alta popularidade entre jogadores.

Comparativo

Na tabela abaixo temos um comparativo entre os principais títulos de jogos com estilo RPG lançados no ano de 2012.

Jogo	Plataforma	Quantidade de jogadores/vendas	Preço
Diablo III	Windows, Mac OS, PS3 e Xbox 360	14,5 milhões	\$ 42.21
Final Fantasy XIII-2	PS3 e Xbox 360	3,1 milhões	\$ 19.00
Mass Effect 3	Windows, WiiU, PS3 e Xbox 360	2 milhões	\$ 20.00

Tabela 1 - Comparativo

Podemos perceber que é um mercado bem diversificado, onde os grandes títulos mantêm a popularidade e tem um bom retorno.

9 Histórico das Atividades

O desenvolvimento deste projeto foi baseado da continuação um outro projeto que havíamos feito no ano passado (2012) na disciplina de Linguagem de Programação 2. Portanto, já tínhamos a lógica de batalha praticamente pronta. No entanto, ao longo do ano, tivemos que alterar o funcionamento de alguns elementos.

Após a apresentação da proposta e aprovação dos professores Ivan e Renato, o primeiro passo para começar o desenvolvimento foi a atualização das bibliotecas da API libGdx. Como é uma API em constante desenvolvimento, muitas funcionalidades foram alteradas e novos recursos foram adicionados. Houve alguns problemas para se adaptar às diferenças da atualização, e metade do grupo não estava familiarizado com a API por não ter participado diretamente no desenvolvimento do código. Portanto, foi necessário destinar o desenvolvimento inicial para o aprendizado da API e atualização das bibliotecas, além do planejamento das user stories e dos novos requisitos.

A atualização das bibliotecas foi feita paralelamente com a implementação de testes unitários no código existente através da plataforma JUnit. O novo modelo necessitou também de uma reformulação da interface de menus (adaptada às novas funcionalidades do modo RPG). Por ter gráficos e processamento pesados, estávamos tendo alguns problemas com smartphones de hardware menos robustos, e tentamos melhorar o gerenciamento de recursos na memória. Após diversas tentativas, chegamos ao consenso de limitar o jogo apenas para tablets e smartphones mais avançados,

Em abril, terminamos as correções no código para a atualização da biblioteca, e continuamos o processo de abrangência do código com os testes unitários, a reformulação da interface, e uma adaptação do roteiro original para o modelo RPG. Também foi iniciado o desenvolvimento de um tutorial interativo do sistema de batalhas e uma forma mais dinâmica de visualizar as informações das cartas.

Ao iniciar o segundo bimestre, em maio, o Gabriel começou a implementar o Editor de Entidades, que nos auxiliaria bastante na criação de novos itens, personagens, mapas, e o armazenamento em arquivos Json. Nessa primeira etapa,

iniciou-se a criação do modelo lógico do editor. Foi iniciada também a implementação da jogabilidade do RPG, com a movimentação do personagem pelo mapa (integrado com o editor de mapas Tiled Map Editor). O Hugo ficou responsável pela implementação lógica, e o Leon, pela interface e renderização. O Lucas continuou na reformulação da interface de menus.

Ao final de junho, conseguimos concluir o modelo e as funcionalidades utilizadas pelo Editor de Entidades e adaptamos o modelo do projeto principal para interagir com os arquivos criados pelo editor. Enquanto isso, continuamos o desenvolvimento do RPG, a reformulação da interface e a adaptação do roteiro.

Nas férias de julho tivemos pouca produtividade. Apenas concluímos algumas tarefas anteriores, como a movimentação do personagem pelo mapa, e iniciamos a implementação da interface do editor de entidades.

Com o início do segundo semestre tivemos a saída do Gabriel da instituição. Consequentemente elegemos o Hugo como novo gerente do projeto. Apesar do Gabriel não comparecer às aulas, ainda ajudava no desenvolvimento do projeto principal.

Em agosto por conta de todos os integrantes estarem fazendo cursinho, a velocidade do desenvolvimento caiu drasticamente, e fomos obrigados a diminuir o escopo do projeto. Ao invés de apresentar um jogo completo na apresentação final, resolvemos apresentar apenas uma versão DEMO, com todas as funcionalidades, mas com apenas um mapa jogável.

Por conta de algumas dificuldades que estávamos tendo e pela necessidade do outro grupo (Shadow Struggles Online) de alterar o modelo do esquema de batalha, fizemos uma série de reuniões para definir um novo modelo e um fluxo de dados mais coerente.

Para continuar a implementação, estávamos muito dependentes da criação dos arquivos Json para codificar o uso dos recursos o fluxo de dados. Portanto, fomos obrigados a dar foco total na interface do projeto auxiliar (Editor de Entidades), que no momento, só continha o modelo.

Com o início do quarto bimestre, fomos direcionados pelos professores a terminar a documentação, que deveria ser entregue dentro de um mês.

10 Problemas

Ao longo do desenvolvimento do projeto o grupo deparou com alguns problemas que serão descritos a seguir

Atualização da biblioteca LibGDX

A primeira decisão do grupo para o projeto foi a atualização da biblioteca utilizada na versão anterior do jogo (desenvolvida em 2012). A nova versão trouxe diversos problemas no código do jogo em geral, assim foi necessário consertar os bugs causados pela atualização. O processo de correção foi realizado com o auxílio do membro Felipe que era um dos integrantes do grupo anterior que desenvolveu a versão do ano de 2012 do projeto.

Utilização da Themable Widget Library (TWL)

A intenção inicial do projeto era utilizar a TWL para desenvolver a interface dos menus do jogo, porém a incompatibilidade do mesmo com a biblioteca LibGDX obrigou o grupo a descartar o seu uso.

Manipulação de dados em arquivo JSON

Para a implementação de novas cartas, para teste e complementação do jogo, o grupo lidou com problemas relacionados à manipulação de dados no JSON, principalmente na leitura dos arquivos, uma vez que a estrutura dos mesmos acabou tornando-se muito extensas para serem desenvolvidas manualmente como ocorria no período de desenvolvimento. Para tanto o grupo optou por focar o desenvolvimento no editor do jogo.

Saída de membro da equipe

No início do terceiro bimestre o integrante do grupo Gabriel precisou retirar-se parcialmente do projeto por motivos pessoais, foi necessário uma reestruturação nas funções do grupo.

11 Escolhas e Descartes

11.1 Escolhas

Mecânica de jogo

O modo Campanha será do tipo RPG (Role Playing Game) em 2D com uma visão aérea. Dentro do RPG o personagem terá atributos que o jogador poderá customizar:

- Quantidade de energia;
- Velocidade de regeneração de energia durante as batalhas;
- Tamanho do baralho;
- Quantidade de vida;
- Pontos de baralho (cada carta vai ter sua quantidade de pontos, dependendo do quão forte for; o número máximo de pontos permitido em um baralho será dado pelos pontos de baralho);
- Probabilidade de saque duplo.

Atualização das bibliotecas do LibGDX

Atualização das bibliotecas do LibGDX.

Refatoração do Código

Organizar melhor o código (de 2012), pois este ficou muito difícil de se trabalhar.

Encriptação de Arquivos

Optamos pela encriptação de arquivos relacionados com as cartas do jogo, assim evitaríamos possíveis trapaças por parte do usuário. Tal arquivo poderá ser alterado apenas por desenvolvedores do jogo.

Sistema de *logging*

Utilização do Java Logging, oferecendo um tratamento de erros adequados.

Teste Unitário

Abranger a maior parte do código com testes unitários utilizando JUnit.

Metodologia de desenvolvimento

Metodologia Scrum, desenvolvimento feito em Sprints, cada uma com um enfoque (individual ou coletivo) em algum aspecto do projeto.

Interface

Utilização do TableLayout, que permite descrever uma hierarquia de elementos da GUI usando uma notação de tabelas e é suportado pelo libgdx.

ClockingIT

Organizador online, utilizamos para designar as tarefas e os prazos para cada integrante do grupo durante cada Scrum.

11.2 Descartes

Smartphones

O desenvolvimento para smartphones Android foi descontinuado devido à lentidão apresentado pela maioria dos dispositivos testados, o jogo será voltado para desktop e tablets.

Themable Widget Library (TWL)

Inicialmente optamos por utilizar a biblioteca TWL devido ao fato desta apresentar diversos recursos para interface de jogos, porém devido à incompatibilidade com a nova versão do LibGDX não foi possível continuar com a utilização dessa biblioteca e então descartamos o seu uso.

12 Lista de Recursos Faltantes

Apesar do comprometimento do grupo em relação ao projeto, não foi possível concluir alguns recursos previamente propostos, sejam estes por necessidade de prazo, sejam por outros problemas, não foi possível a entrega do projeto com esses recursos. Nesta parte são descritos os recursos faltantes, qual foi a causa para não concluir este, o que conseguimos implementar e o que falta.

NPCs

- Descrição: Non-Playing-Characters, fazem parte do jogo em RPG na qual o usuário poderia interagir com os personagens do ambiente do jogo.
- Causa: Problemas com a parte de inteligência artificial, pois alguns personagens teriam estratégias próprias de batalha
- Implementado: Nada foi implementado.
- Falta: O recurso em si.

Enredo

- Descrição: A história na qual o jogo se baseia.
- Causa: Muito a ser implementado e pouco foco nessa parte
- Implementado: Foram escritos alguns capítulos do enredo e discutimos os finais possíveis para a história, porém não foi possível finalizá-lo.
- Falta: Finalizar e implementar completamente.

Diálogos

- Descrição: Ato de conversar dos personagens do jogo
- Causa: O projeto apresentou diversas falhas em outros aspectos além do RPG e o grupo precisou desviar o foco do desenvolvimento para reparar os problemas.
- Implementado: Nada foi implementado.
- Falta: O recurso em si.

Cartas

- Descrição: Elementos para o usuário utilizar no seu deck
- Causa: Pouco foco.
- Implementado: Algumas cartas já estão criadas

Falta: Implementar o restante

13 Conclusão

Após quase 10 meses de empenho no desenvolvimento dessa nova etapa do projeto, a equipe teve um grande aprendizado de como funciona o processo de criação de um software. Antes desse desafio, a maior parte da equipe conhecia apenas superficialmente as etapas de desenvolvimento. Muito do que sabíamos fazer era só abrir a IDE e começar a codificar o que “pode funcionar”.

Apesar de termos partido de um projeto desenvolvido no ano passado, os focos foram diferentes, a própria API havia passado por muitas mudanças desde então. Pudemos nos preocupar com muitos mais detalhes do funcionamento e do desempenho do jogo e enxergar novos aspectos do desenvolvimento, como metodologias de desenvolvimento, ferramentas de gestão, entre outros.

Muitas dificuldades surgiram ao longo do desenvolvimento, tanto em âmbito técnico quanto social, e tivemos de lidar com nossos próprios obstáculos pessoais. Foi um grande aprendizado de trabalho em equipe e organização de prioridades que iremos levar adiante nos próximos projetos. Para alguns membros que irão seguir a área de desenvolvimento, já é uma boa carga de experiência e disciplina, considerando que foi muito próximo de um desenvolvimento com requisitos profissionais.

Mesmo vivendo uma época conturbada com múltiplas tarefas do dia-dia, a equipe foi capaz de se organizar e vencer muitos desafios através do Shadow Struggles 2013. A grande expectativa é de continuar nos desenvolvendo e melhorando o jogo e, quem sabe, lançar uma versão oficial no mercado de jogos futuramente.

14 Links do Projeto

- **Blog do Projeto:** <http://studioblackjack.blogspot.com.br/>
- **Endereço do Repositório do IFSP:**
<https://fw.lab.sp.cefetsp.br/svn/a6pgp/A2013-PDS/ShadowStruggles/>
- **Endereço do Repositório do Google Code:**
<https://shadowstruggles.googlecode.com/svn/>
- **Canal do Youtube da Equipe:**
<https://www.youtube.com/user/TheStudioBlackJack>
- **ClockingIT:** <http://blackjack.clockingit.com/>
- **LibGdx:** <http://libgdx.badlogicgames.com/>

15 Referências bibliográficas

- **Requisitos do projeto para a Disciplina de PDS – Acessado em 22/10/13:**
<http://dicas.ivanfm.com/>
- **Desenvolvimento Ágil – Scrum – Acessado em 22/10/13:**
<http://desenvolvimentoagil.com.br/scrum/>
- **Aspiring Craftsman – Interactive Architecture Patterns – Acessado em 22/10/13:**
<http://aspiringcraftsman.com/2007/08/25/interactive-application-architecture/>
- **OpenGLized – Reading and Parsing JSON – Acessado em 22/10/13**
<http://openglized.blogspot.com.br/2013/06/reading-and-parsing-json-file-with.html>

Apêndice A – Documento de Aprovação

Projeto: Shadow Struggles

Equipe: BlackJack

- Gabriel Merêncio dos Santos
- Hugo Mitsumori
- Leon Henrique Pires de Brum
- Lucas Hideki Sakurai

Objetivos:

- Dar continuidade ao projeto Shadow Struggles iniciado em 2012, completando os seguintes itens:
 - Loja de itens: compra, venda de cartas, e acessórios
 - Edição de baralho: trocar cartas entre o deck em uso e o estoque do jogador
 - Correção de bugs (problemas com gerenciamento de memória)
- Desenvolver produto finalizado na forma de um RPG 2D:
 - Nesta modalidade, o jogador poderá explorar os cenários, em complemento à narrativa
 - Vários inimigos, missões e itens espalhados pelo ambiente
 - Quantidade de experiência e níveis

- Distribuição de habilidades para cada novo nível que o jogador atingir permitindo que ele melhore seus atributos

Plano de atividades

Núcleo principal do jogo:

- Colocar tutorial interativo dentro do jogo
- Terminar modo edição de baralho
- Terminar loja de itens
- Reformular criação de cartas, tornando a adição de novas cartas mais simples
 - Estabelecer alguns efeitos/ações pré-programados
 - Uma nova carta selecionará um conjunto desses efeitos e o configurará de modo específico
- Reformular criação de inimigos, tornando a adição de novos inimigos mais simples
 - Similar à criação de cartas: estabelecer rotinas básicas pré-programadas e permitir que uma IA seja configurada a partir de um arquivo usando essas rotinas
- Otimizar gerenciamento de memória para Android
- Utilizar TWL para reformular a interface dos menus
- Implementar modo RPG
 - Interpretar mapas construídos pelo Tiled Map Editor
 - Interpretar NPCs e objetos diversos nos mapas
 - Reformular a construção de cenas para permitir cenas mais complexas

- Implementar interação do jogador com o mapa (movimentação do personagem, falar com NPCs etc.)
 - Definir e implementar status do jogador
 - Implementar sistema de experiência e níveis
 - Implementar sistema de distribuição de pontos
 - Implementar inventário do jogador
- Tornar a jogabilidade mais intuitiva através de mais informações na tela de batalha
 - Mostrar informações das cartas durante a batalha
 - Exibir formas de representação alternativas das cartas para tornar o visual mais agradável
- Encriptar as informações dos arquivos
- Abranger todo o código existente com testes unitários
- Utilizar Eclipse Modelling Framework para transformar classes existentes em diagramas
- Atualizar as bibliotecas do libgdx

Ferramentas externas para a construção do jogo:

- Edição de cartas e baralhos
- Edição de inimigos
- Edição de cenas e fluxo de jogo
 - Por exemplo, alterar os “pontos de Luz”/”pontos de Trevas” após fazer uma determinada escolha
- Edição de mapas

- Adicionar NPCs e outros objetos, configurando seus comportamentos

Outros

- Adaptar o roteiro para o modo RPG
- Adaptar o GDD para o modo RPG
- Fazer a tradução para português
- Terminar a sonoplastia do jogo
- Reformular os cenários de batalha existentes
- Criar mapas do modo RPG
- Criar os personagens para o modo RPG

Tecnologias e ferramentas:

- LibGDX: Framework de desenvolvimento multiplataforma voltada para jogos
- Blender: Ferramenta 3D para criação dos personagens
- JUnit e Android JUnit: Tecnologias para testes unitários
- Logcat e Java Logging: Criação de logs
- Eclipse: IDE de desenvolvimento
- Tiled Map Editor: Editor de mapas (RPG)
- Eclipse Modelling Framework: Transformar diagramas em classes Java e classes em diagramas, permitindo uma maior sincronização entre a documentação e o código e facilitando tomadas de decisão em relação à arquitetura

Metodologias:

- **Gerenciamento:** Scrum
- **Desenvolvimento:** Test Driven Development

Apêndice B – Resumo das Atas de Reuniões

Reunião de 14/02

Foi realizada uma reunião antes da definição oficial dos grupos, com participantes tanto do projeto Shadow Struggles quanto do projeto Shadow Struggles Online: Felipe Brigalante, Gabriel Merêncio dos Santos, Hugo Mitsumori, Leon H. P. Brum, Lucas Hideki Sakurai, Victor de Medeiros Anderson e Victor Eiti Yamamoto. Foram discutidos diversos tópicos envolvendo *features* dos projetos, o processo de desenvolvimento e mudanças no código. Resumo dos resultados, tendo em vista exclusivamente o Shadow Struggles:

- Foi sugerido que a campanha se desse na modalidade de RPG 2D, com algumas características pré-definidas;
- O design foi apontado como uma característica essencial, então decidimos investir na produção de melhores interfaces;
- Optamos por gastar algum tempo do desenvolvimento atualizando as bibliotecas do LibGDX, a fim de obter melhores recursos e mais suporte;
- A metodologia de desenvolvimento Test Driven Development (TDD) foi escolhida, bem como a metodologia de gerenciamento Scrum;
- Foi sugerido que utilizássemos alguma ferramenta para modelar as classes, sincronizando o código com a documentação.

Link para a postagem completa relatando as conclusões da reunião:
<http://studioblackjack.blogspot.com.br/2013/02/conclusoes-da-reuniao-de-1402.html>.

05/07/2013 – Planejamento pré-férias

No dia 07 de julho, após o início das férias do IFSP, resolvemos fazer uma reunião por Skype entre os membros da equipe para fazermos um levantamento geral de tudo o que foi feito no primeiro semestre e o planejamento das férias e do segundo semestre. Foi objetivado o término da engine do jogo e do editor de arquivos para posteriormente ter tempo para dedicar ao conteúdo, design e testes no segundo semestre.

22/08/13 – Redefinições da mecânica de batalha

Por conta da dificuldade da equipe do Shadow Struggles Online de adaptar a estrutura de batalha atual para funcionar em rede, foi necessário redefinir as classes do modelo e da interface da mecânica de batalha.

A reunião do dia 22, realizada em conjunto com a equipe de programação do Shadow Struggles Online, serviu para retrabalharmos a Model da batalha. Utilizamos um plugin do eclipse chamado AmaterasUML para a modelagem dos dados e do fluxo de funcionamento do modelo de batalha.

29/08/13 – Modelagem da Viewer da batalha

Dando continuidade na reformulação da estrutura da batalha, destinamos a reunião do dia 29 para modelar as classes que farão a renderização e processamento dos métodos da interface da batalha.

Para a modelagem das classes da interface, também utilizamos o AmaterasUML como ferramenta de criação dos diagramas. Ao longo da reunião, surgiram algumas dúvidas, que seriam resolvidas durante a semana. O principal questionamento foi com relação à função da Controller, que no momento, só servia para acumular métodos, e que poderia facilmente ser eliminada ou resumida após uma organização melhor.

19/09/13 – Reunião com os Clientes (professores Renato e Ivan)

No dia 19 de setembro, ao término do terceiro bimestre, os professores se reuniram com cada grupo para ver o progresso do desenvolvimento e a documentação.

Mostramos o estado atual do projeto, o projeto Tools, e os problemas ocorridos no terceiro bimestre.

Como não estava sendo dada a atenção devida para a documentação do projeto, os professores nos direcionaram a utilizar o tempo de desenvolvimento na formulação da documentação do projeto e nos requisitos da disciplina, que constam no site

<http://dicas.ivanfm.com/aulas/pds>

A documentação deveria ser entregue no dia 24 de outubro, e como faltava pouco tempo, resolvemos dar uma parada no desenvolvimento do jogo e do projeto Tools para nos dedicar inteiramente na documentação faltante.

Apêndice C – Postagens Semanais

Seguem as postagens de relatório, com a ressalva de que muitas delas dependem de *links* externos para terem sentido completo, então é recomendado vê-las em suas páginas originais.

Relatório de atividades (03/03 a 09/03)

Nesta semana inaugural, estabelecemos as principais atividades do projeto no ClockingIT, site que permite o gerenciamento de tarefas dentro de uma equipe. Infelizmente, não conseguimos encontrar uma boa ferramenta gratuita voltada a Scrum, então vamos ter que adaptar o ClockingIT às nossas necessidades. Enfim, aqui está o que temos planejado para o primeiro *sprint*.

Tarefa	Dificultad e	Membros delegados à tarefa
Concertar erros causados pela atualização do libgdx	Normal	Todos
Retrabalhar interface dos menus	Crítico	Gabriel, Leon
Otimizar gerenciamento de recursos	Crítico	Gabriel, Hugo
Abranger o código existente com testes unitários	Normal	Todos
Adaptar o GDD ao modo RPG	Muito grave	Gabriel

Tabela 2 – Tarefas Relatório

PS: Os níveis de dificuldade do ClockingIT são "Trivial", "Menor", "Normal", "Principal", "Crítico" e "Muito grave".

O sprint está marcado para terminar em 31/03, quando esperamos poder lançar a versão 0.1 com algumas melhorias gráficas e suporte a português e a Android. A

partir dos resultados obtidos, poderemos avaliar a velocidade da equipe e planejar o próximo sprint de acordo.

- **Gabriel:** Correção de erros, algumas pesquisas para coletar ideias úteis (resultados descritos abaixo), início das interfaces.
- **Hugo:** Correção de erros, início de testes unitários.
- **Leon:** Correção de erros.
- **Lucas Hideki:** Correção de erros, leituras sobre testes unitários.

O Felipe, do Shadow Struggles Online, também nos ajudou em relação à atualização do libgdx.

Resultados da pesquisa

No que diz respeito ao gerenciamento de recursos, a solução parece ser juntar imagens individuais em um único atlas de texturas através da classe TexturePacker2 do libgdx; aqui está um tutorial e exemplo de utilização. É menos custoso para o sistema manter uma única textura na memória e procurar as texturas individuais dentro dela do que ficar trocando de texturas.

Para a interface, provavelmente utilizaremos TableLayout, que permite descrever uma hierarquia de elementos da GUI usando uma notação de tabelas e é suportado pelo libgdx. Adicionalmente, encontrei uma maneira mais simples de linkar recursos comuns entre o desktop e o Android do que eu anteriormente fazia.

Decisões

Por fim, tomamos mais algumas decisões em relação ao *game design*. A saber:

- O estio gráfico do modo RPG seguirá uma visão aérea, como em Pokémon.
- Os atributos do personagem, que o jogador poderá customizar, serão:
 - Quantidade de energia;

- Velocidade de regeneração de energia durante as batalhas;
- Tamanho do baralho;
- Quantidade de vida;
- Pontos de baralho (cada carta vai ter sua quantidade de pontos, dependendo do quão forte for; o número máximo de pontos permitido em um baralho será dado pelos pontos de baralho);
- Probabilidade de saque duplo.

Disponível em: <http://studioblackjack.blogspot.com.br/2013/03/relatorio-de-atividades-0303-0903.html>.

Relatório de atividades (10/03 a 16/03)

Semana sem muitas novidades, mas por enquanto as coisas estão sob controle.

- **Gabriel:** Refatoração de grande parte do gerenciamento de recursos e mais tentativas de implementar a nova interface através do TableLayout.
- **Hugo:** Tentativas de implementar um sistema de log e refatorações no código.
- **Leon:** Testes unitários.
- **Lucas Hideki:** Documentação de código.

Novamente, o Felipe do Shadow Struggles Online nos auxiliou no que diz respeito à manutenção do código. Adicionalmente, foram levantadas algumas observações quanto ao game design, das quais chegamos a duas decisões:

1. Haverá um sistema de "New Game+" quando o jogador chegar a algum final, permitindo que ele regresse a um estado antes de realizar uma escolha. Ele regressará com todos os seus atributos mantidos, mas terá que enfrentar os mesmos inimigos e ver as mesmas cenas novamente.
2. Haverá um item especial para limpar a distribuição de pontos realizada por um jogador, permitindo que eles os redistribua. Este item provavelmente será adquirido no Shadow Struggles Online, mediante compra por cash.

Relatório de atividades (17/03 a 23/03)

Com o sprint chegando ao fim, já podemos ver um caminho claro até onde queremos chegar. Enfim, ao relatório.

- **Gabriel:** Reorganização dos recursos em atlas de texturas, concertos em arquivos e animações, reestruturação da documentação de game design.
- **Hugo:** Reestruturação de parte do código, testes unitários.
- **Leon:** Testes unitários, início da otimização das imagens.
- **Lucas Hideki:** Testes unitários.

Uma das tarefas do sprint já foi concluída; a atualização das bibliotecas do libgdx foi realizada com sucesso, de modo que o jogo roda da mesma forma que rodava antes da atualização. Os testes unitários e processo de otimização para Android também estão próximos do fim. O game design deve estar pelo menos majoritariamente pronto até o final do sprint, enquanto as interfaces talvez não possam ser elaboradas e/ou implementadas a tempo.

Relatório de atividades (24/03 a 31/03)

O primeiro *sprint* foi concluído, mas infelizmente não foi possível cumprir a proposta inicial de ter uma nova demo para divulgação. De toda forma, vamos planejar os detalhes dos rumos do projeto nesta semana. Quanto ao que foi feito na semana:

- **Gabriel:** Conclusão da reorganização do gerenciamento de recursos, mas ainda falta testar no Android; início da implementação da primeira tela.
- **Hugo:** Auxílio da construção da interface através do TableLayout, reestruturação do código.
- **Leon:** Redimensionamento das imagens para otimização.
- **Lucas Hideki:** Testes unitários.

Relatório de atividades (01/04 a 06/04)

Revisando o primeiro *sprint* e realizando o planejamento do segundo, vimos que os dados indicam que vamos conseguir concluir todas as tarefas a tempo. Por enquanto, ainda vamos nos focar em concertar e implementar aspectos legados do ano passado e reformular o visual.

- **Gabriel:** Game design.
- **Hugo:** Alterações estruturais no código.
- **Leon e Lucas Hideki:** Interface dos menus.

Relatório de atividades (06/04 a 13/04)

Mais uma semana normal. Encontrei o ObjectAid UML Explorer como uma solução para converter o código em modelos e mantê-los sincronizados com as mudanças.

- **Gabriel:** Game design.
- **Hugo:** Tentativas de implementar o tutorial.
- **Leon e Lucas Hideki:** Interface dos menus.

Relatório de atividades (14/04 a 20/04)

Nessa semana, verificamos que o máximo de otimização que conseguimos alcançar só nos permite obter um bom desempenho em tablets ou smartphones de boa performance. Considerando também aspectos de design e jogabilidade, foi decidido que o Shadow Struggles só irá suportar PC e tablets (ainda a partir de Android 2.2).
Sumário das atividades:

- **Gabriel:** Game design, otimização de recursos e detalhes na interface.
- **Hugo:** Implementação do tutorial.
- **Leon e Lucas Hideki:** Interface dos menus.

Relatório de atividades (21/04 a 27/04)

Nessa semana, os professores Ivan e Renato realizaram uma avaliação dos grupos, que deveriam relatar como o projeto estava progredindo em linhas gerais. Em nosso projeto, foram ressaltados pelos professores aspectos referentes à nossa organização interna através do ClockingIT e a planilha com o cronograma geral, como algumas atividades que devem ser melhor detalhadas, mas no geral parecemos estar bem encaminhados. Quanto às tarefas individuais:

- **Gabriel:** Game design, atualizações na documentação geral, proposta de solução ao problema com testes envolvendo arquivos¹.
- **Hugo:** Implementação do tutorial e correção de erros em arquivos de texto².
- **Leon e Lucas Hideki:** Interface de menus.

¹A atividade de abranger o código legado com testes unitários havia sido provisoriamente suspensa devido ao fato de termos encontrado dificuldade em testar módulos dependentes de arquivos. Com a solução proposta, devemos ser capaz de retomar essa atividade no futuro ou delegá-la ao grupo do Shadow Struggles Online.

²Devido a conflitos de codificação, havia alguns erros em arquivos de texto que comprometiam a leitura. Esse era o último detalhe que faltava para o jogo enfim possuir suporte pleno a português.

Relatório de atividades (28/04 a 04/05)

Início do mês, início de um novo *sprint*. Nosso objetivo para maio é estabelecer a base para o modo RPG, incluindo as ferramentas de implementação do jogo. A ideia é terminar a programação geral o quanto antes possível para podermos nos concentrar no game design em si, realizando testes e fazendo ajustes. Segue o resumo das atividades realizadas pelos membros da equipe na semana:

- **Gabriel:** Game design, modelagem de dados, auxílio à implementação do tutorial.
- **Hugo:** Implementação do tutorial.
- **Leon e Lucas Hideki:** Interface de menus.

Relatório de atividades (05/05 a 11/05)

Nessa semana, começamos a implementar o RPG e desenvolver as ferramentas de auxílio na construção do jogo.

- **Gabriel:** Implementação dos modelos nas ferramentas.
- **Hugo:** Implementação do RPG (interpretação/renderização do mapa e movimentação do jogador).
- **Leon:** Interface de menus.
- **Lucas Hideki:** Pesquisas sobre serialização para lidar com possíveis problemas de interoperabilidade entre os arquivos gerados pelas ferramentas e o jogo.

Relatório de atividades (12/05 a 18/05)

Houve alguns avanços bem significativos no projeto Tools, as ferramentas de implementação do jogo, embora ainda não haja uma interface gráfica. O RPG também está indo bem, mas ainda restam alguns desafios que podem tomar um tempo.

- **Gabriel:** Construção das ferramentas (modelos e gerenciamento de arquivos).
- **Hugo e Leon:** Implementação do RPG (interpretação/renderização do mapa e movimentação do jogador).
- **Lucas Hideki:** Interface de menus.

Para fins de documentação, a seguir vou comentar brevemente sobre o problema de compatibilidade entre os arquivos e como resolvemos.

No projeto, utilizamos a biblioteca JsonBeans para a serialização e desserialização de objetos em arquivos JSON devido ao fato de ser compatível com o LibGDX. A biblioteca funciona bem quando é utilizada em um único sistema, mas encontramos uma dificuldade ao lidar com dois projetos distintos, com pacotes e classes diferentes: listas. Observe, por exemplo, a seguinte classe, com getters, setters e construtores omitidos :

```
public class Foo {  
    private String name;  
    private ArrayList<Integer> fooBars;
```



```
}
```

Um objeto dessa classe serializado possui a seguinte representação em JSON, por padrão:

```
{
  "name": "Name"
  "fooBars": [
    { "class": "java.util.Integer",
      "value": 2
    },
    { "class": "java.util.Integer",
      "value": 40
    }
  ]
}
```

Como listas podem armazenar objetos de quaisquer tipos, é necessário fazer referência ao nome completo da classe, incluindo pacote (no caso, `java.util.Integer`). Isso é um problema no nosso caso, uma vez que temos classes idênticas, mas em pacotes distintos (por exemplo: `"com.example.Card"` e `"com.example.tools.Card"`). Para contornar a situação, criamos em ambos os projetos uma classe que inicializa um objeto `Json` e atribui *tags* comuns às classes:

// Projeto 1:

```
Json json = new Json();
json.addClassTag("Card", com.example.Card.class);
```

// Projeto 2:

```
Json json = new Json();
json.addClassTag("Card", com.example.tools.Card.class);
```

Logo, cada um dos projetos lida com suas próprias classes e pacotes, mas a interação entre os sistemas não é prejudicada devido ao uso de *tags* compartilhadas.

Relatório de atividades (18/05 a 25/05)

- **Gabriel:** Gravação de arquivos no editor do jogo e esboço da interface do editor.
- **Hugo:** Finalização do tutorial.
- **Leon:** Implementação do RPG (movimentação do jogador).
- **Lucas Hideki:** Interface de menus.

Relatório de atividades (26/05 a 01/06)

Semana de feriado, sem grandes evoluções no projeto. Adicionamos os módulos cliente e servidor do Shadow Struggles Online ao nosso repositório, assim todos têm acesso fácil tanto ao Shadow Struggles Online quanto o "Shadow Struggles Offline". Eu (Gabriel) trabalhei no editor e implementei algumas funcionalidades básicas, como gravar e procurar arquivos e dados de objetos.

Tecnicamente, já quase está tudo pronto, testado e documentado (com exceção da interface), mas o processo de edição e remoção de dados de objetos ainda é bem mais complicado do que deveria ser.

Relatório de atividades (02/06 a 08/06)

Nesta semana, teremos mais uma apresentação para mostrar os avanços no projeto até agora. Pretendemos nos focar nos progressos de jogabilidade (tutorial, início do modo RPG), design (interfaces de menu) e no editor. Os slides serão disponibilizados no próximo relatório. Agora, às atividades!

- **Gabriel:** Finalizações no editor (sem interface gráfica), adaptação do projeto principal ao novo modelo de dados e início da implementação de um algoritmo de pathfinding (algoritmo A*) para o RPG.
- **Hugo e Leon:** Implementação do RPG.
- **Lucas Hideki:** Interfaces.

Relatório de atividades (09/06 a 15/06)

Essa última semana foi dedicada à apresentação parcial do projeto.

Relatório de atividades (16/06 a 22/06)

Nessa última semana de aula antes das férias, foi apresentado o progresso dos grupos restantes e tivemos que nos focar nas provas e trabalhos finais, então não houve muito avanço no projeto. Pretendemos, contudo, continuar o desenvolvimento durante as férias.

Relatório de atividades (23/06 a 29/06)

Neste final de sprint, o projeto principal agora já está adaptado para receber os dados gerados pelo editor. Por enquanto ainda não há grandes mudanças que impactam diretamente o jogador, mas, vamos tentar terminar essa etapa mais preparatória durante as férias. Às tarefas:

- **Gabriel:** Adaptação do projeto principal ao novo modelo de dados, início da roteirização.
- **Hugo e Leon:** Implementação do RPG.
- **Lucas Hideki:** Interfaces.

Relatório de atividades (30/06 a 06/07)

Fizemos uma reunião na sexta, 05/07, para esclarecer alguns pontos do projeto e nos planejarmos para as férias. Nosso objetivo primário é concluir boa parte da programação principal do jogo (a engine do RPG e o editor) para podermos nos dedicar ao conteúdo, design e testes do jogo em si no segundo semestre. As tarefas realizadas durante a semana foram:

- **Gabriel:** Movimentação de personagem, interfaces de menu, roteiro.
- **Hugo:** Interface do editor.
- **Lucas Hideki:** Interfaces de menu.
- **Leon:** Movimentação de personagem.

Relatório de atividades (07/07 a 13/07)

Por motivos de viagem, o Leon não pôde fazer mudanças no projeto, mas ele está encarregado da interpretação de eventos no mapa. Quanto às atividades dos demais membros:

- **Gabriel:** Interfaces de menu.
- **Hugo:** Interface do editor.
- **Lucas Hideki:** Interfaces de menu.

Relatório de atividades (14/07 a 20/07)

Com as férias chegando ao fim, vamos ao resumo das atividades realizadas na semana:

- **Gabriel:** Ajustes no editor e roteiro.
- **Hugo:** Interface do editor.
- **Lucas Hideki:** Testes de batalha no projeto principal.

O Leon ainda está viajando.

Relatório de atividades (21/07 a 27/07)

As aulas oficialmente começaram na última quinta, mas o Hugo e o Leon não puderam comparecer. Enquanto isso, eu e o Hideki finalizamos a tarefa de reformulação da interface dos menus do jogo e comecei a implementar a interpretação de eventos no mapa do RPG.

Relatório de atividades (28/07 a 03/08)

- **Gabriel:** Finalização da interpretação de eventos, mudanças na movimentação do personagem, sistema de log, compressão/descompressão de arquivos e gerenciamento de recursos.
- **Hugo:** Mudanças na interface de batalha.
- **Leon:** Implementação dos eventos de Save e teletransporte.
- **Lucas Hideki:** Testes unitários.

Relatório de atividades (04/08 a 10/08)

- **Gabriel:** Mudanças no sistema de batalha e implementação inicial do sistema de itens e o novo sistema de IA.
- **Hugo:** Interface de batalha.
- **Leon:** Sistema de eventos.
- **Lucas Hideki:** Testes unitários.

Relatório de atividades (11/08 a 22/08)

Verificamos que o projeto estava difícil de implementar devido às constantes mudanças no modelo e na forma como a batalha estava esquematizada.

Resolvemos fazer uma reunião (realizada no dia 22/08) e definimos o modelo e o esquema da batalha.

Está planejado reunir novamente na semana que vem para fazer as definições da Viewer e da Controller da batalha.

Até lá, procuraremos implementar em código o modelo definido em diagrama.

Relatório de atividades (23/08 a 29/08)

Estamos continuando a reformulação da arquitetura da batalha que estava difícil de fazer manutenção.

Fizemos uma reunião no dia 29/08 e definimos a Viewer da battle em formato de diagramas. Ficaram algumas discussões pendentes que serão resolvidas ao longo da semana. (Como por exemplo, a função da Controller na batalha).

Relatório de atividades (29/08 a 05/09)

Resolvemos nos focar no desenvolvimento do projeto do Editor de arquivos para facilitar a criação de cartas, eventos, cenas, e outros elementos que serão armazenados em arquivos JSON.

Temos a model funcional, mas ainda há dificuldade em como implementar uma interface visual para o desenvolvedor.

Relatório de atividades (05/09 a 16/09)

Durante esse período, procuramos dar um foco no desenvolvimento do Editor de Arquivos. Pela falta de tempo, estamos cogitando a ideia de reduzir o escopo para um versão DEMO, ao invés de uma versão final completa.

- Hugo: Editor de Recursos, Interface Geral, Editor de Inimigos (a concluir)
- Leon: Editor de Eventos (a concluir)
- Hideki: Editor de Decks e Cartas (a concluir)
- Gabriel: Sistema de Quests e Ações Condicionais dos Eventos

Relatório de atividades (16/09 a 26/09)

Após avaliação dos professores, fomos direcionados a mudar o foco do momento para a documentação, que ainda está com pouco conteúdo.

Cada membro ficará responsável por uma leva de tópicos primeira parte da documentação:

Lucas Hideki: Análise de Nicho, Descrição da Navegação, Ferramentas e Tecnologias, Testes

Leon: Mecânica do jogo, Escolhas e Descartes, Caracterização do Projeto, Problemas

Hugo: Epígrafe, Armazenamento de Dados, Conclusão e Histórico das Atividades

O prazo previsto para terminar essa primeira parte é até o dia 03/10/13

Relatório de atividades (27/09 a 03/10)

Continuamos o foco na documentação do projeto.

Objetivos Terminados:

- Descrição da navegação e Análise de Nicho (Hideki);
- Armazenamento de Dados e Histórico das Atividades;

Objetivos em andamento:

- Descrição das Ferramentas e Tecnologias (Hideki);
- Relatório das publicações semanais (Hugo);
- Mecânica do jogo, Escolhas e Descartes, Caracterização do Projeto, Problemas (Leon);

Objetivos para o dia 10/10/13:

- Hugo: terminar o relatório das publicações, terminar o histórico de reuniões, cronograma e manual técnico

- Hideki: terminar a Descrição das Ferramentas e Tecnologias, Escolhas e Descartes, e Relatório dos problemas ocorridos durante o desenvolvimento;

- Leon: terminar a descrição das Ferramentas e Tecnologias e a Caracterização do Projeto

Relatório de atividades (03/10 a 10/10)

Nessa última semana terminamos a primeira parte mais geral do projeto.

- Descrição das Ferramentas e Tecnologias;
- Relatório das publicações semanais ;
- Descrição da Mecânica do jogo;
- Relatório de Escolhas e Descartes;
- Caracterização do Projeto;
- Histórico de Reuniões;

O Hideki ficou de terminar o Relatório de Problemas;

Para a próxima semana iremos nos focar no manual técnico, que dividimos em 3 partes:

- Manual Técnico Geral do Jogo (Menus, Funcionalidades) - Hideki
- Manual Técnico do modo RPG (Movimentação, eventos, cenas) - Leon
- Manual Técnico da Batalha (Invocações, Inteligência Artificial, Fluxo de Batalha) – Hugo

Relatório de atividades (10/10 a 17/10)

Nessas últimas semanas, estamos nos concentrando em terminar a documentação do projeto.

O foco da semana foi o manual técnico do projeto.

Eu terminei os casos de uso da parte da batalha,

o Hideki terminou a parte da navegação,
e o Leon está finalizando a parte do RPG

Iniciamos nessa semana a descrição dos pacotes e a descrição técnica geral do projeto. O planejamento é terminar também os DFD's e a descrição das classes principais.

O manual técnico deve ser terminado até o dia 21/10, para termos tempo de imprimir as três vias e fazer os ajustes necessários

Apêndice D – Cronograma de Desenvolvimento do Projeto

Tarefas concluídas (total ou parcialmente): em azul.

Tarefas em andamento: em verde.

Tarefas não cumpridas: em vermelho.

	mar	abr	mai	jun	jul	ago	set	out	nov
Adaptar o GDD para o modo RPG	X	X							
Otimizar o gerenciamento de recursos	X	X							
Retrabalhar interface de menus	X	X	X	X	X				
Abranger código com teste unitários	X	X							
Atualizar bibliotecas do LibGdx	X	X							
Seleção de Idioma		X							
Tutorial In-Game		X	X						
Visualização Rápida das Cartas		X							
Tools – salvar informações			X						
Tools – Editar Informações			X	X					
Tools – Remover Informações			X	X					
Movimentação do Personagem			X	X	X	X			
Adaptar modelo e gerenciamento de dados				X					
Tools – Interface Gráfica					X	X	X	X	X
Sistema de Eventos						X			
Interface de Batalha						X			
Sistema de Ítems						X			
Novo sistema de IA						X			
Tools – Editor de recursos							X		
Sistema de Quests							X		
Ações Condicionais							X		
Relatório de Desenvolvimento								X	X

Tabela 3 – Cronograma

Apêndice E - Plano de Testes

Inicialmente, a proposta era de utilizar a metodologia de desenvolvimento *Test Driven Development*, em que antes das funcionalidades, são implementados os testes unitários. No entanto, a equipe acabou falhando em alguns momentos, e muitas classes acabaram ficando sem testes unitários.

Outra alteração das definições iniciais foi com relação ao JUnit, que seria a plataforma de testes unitário, mas como estávamos tendo problemas com a leitura de dados nos testes devido a um mecanismo do LibGdx, acabamos por montar nossa própria rotina de testes, de uma forma mais manual.

Ao executar o projeto, existem 3 modos de execução: DEBUG, TESTS e RELEASE.

- O modo DEBUG resulta em uma execução com alguns traços de referência para auxiliar na visualização dos resultados por parte do desenvolvedor. É o modo de execução enquanto o projeto estiver em desenvolvimento
- O modo RELEASE deve ser configurado ao finalizar a aplicação e gerar o executável para distribuição. Representa a aplicação final como deve ser vista pelo usuário.
- O modo TESTS faz com que, ao invés de executar a rotina normal do jogo, seja executada a rotina de testes de cada classe.

Assim como no JUnit, é criada uma classe de teste para cada classe do projeto, cada uma com os testes dos respectivos métodos. No término da execução dos testes, ao invés de gerar um relatório final com os resultados dos testes, os mesmos são lançados na saída de dados (System.out.print).

A implementação dos testes em si não fica muito diferente de testes unitários do JUnit, mas obviamente, ficamos sem alguns recursos que a ferramenta oferece, como exportação dos resultados, por exemplo.

Apêndice F – Análise de Código

Durante o desenvolvimento do projeto Shadow Struggles 2013, utilizamos uma ferramenta chamada FindBugs para a análise estática do código.

O FindBugs faz uma análise estática para identificar uma porção de erros potenciais no código Java, desde erros conceituais a más praticas de programação.

A utilização da ferramenta se deu em uma fase já avançada do desenvolvimento, de tal modo que foram identificadas algumas dezenas de erros, e não foi possível a correção de todos.

Alguns bugs encontrados, acabamos considerando como alarmes falsos, ao passo que alguns alertas nos ajudaram a corrigir erros no jogo. Apesar da utilidade, a maior parte dos erros que causavam dificuldade para a equipe não foi detectada com a ajuda do software, e boa parte dos alertas de más praticas de programação acabou não sendo corrigida por falta de tempo.

Apêndice G – Logomarcas do Projeto



Figura 11 – Logo inicial



Figura 12 – Logo proposto

Apêndice H – Formulário de Aprovação

Gabriel Merêncio dos Santos

Hugo Mitsumori

Leon Henrique Pires de Brum

Lucas Hideki Sakurai

Documento de aprovação do Projeto Shadow Struggles 2013

Objetivo: Aprovação na matéria de Prática e Desenvolvimento de Sistemas – 2013

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

Ensino Médio Integrado ao Técnico em Programação e Desenvolvimento de Sistemas

Data de aprovação:___/___/___

Comissão Examinadora:

Nome do Membro:_____

Nome do Membro:_____

Nome do Membro:_____