

Universidade de Brasília - Faculdade Gama

Disciplina: Introdução aos Jogos Eletrônicos

Professor: Edson Alves

Alunos:

Tiago Gomes Pereira 09/0134222

Matheus Fonseca 10/0054650

Charles Oliveira 09/0043006

Pedro Zanini 10/47337

Ninja Siege

Documento de Design

Sumário

1	Introdução	3
2	Premissa e Ambientação do Jogo	3
3	Elementos do Jogo	3
3.1	Waves	3
3.2	Inimigos	3
3.2.1	Samurais	3
3.2.2	Piratas	4
3.3	Torres	5
3.3.1	Curto Alcance	5
3.3.2	Longo Alcance	5
3.3.3	Armadilhas	5
3.4	Percurso	5
3.4.1	Posicionamento das Torres	5
3.4.2	Relação Torre X Inimigo	5
3.5	Mapas	6
4	Mercado do jogo	6
4.1	Moeda	6
4.2	Evolução de unidades	6
5	Interfaces	6
5.1	Audio	6
5.2	Video	7
5.3	Controles	7
6	Plataforma	7
7	Audiência	7
8	Desenvolvimento	7
8.1	Linguagem compilada	7
8.2	Linguagem de script	7
8.3	API's	8
8.3.1	API Gráfica - SDL	8
8.3.2	API de Áudio	8
8.3.3	API de Rede	8
8.3.4	Manipulação de Arquivos	8
8.3.5	GUI	8
8.4	Documentação	8
8.5	Ferramenta de Controle de Versão	8
8.6	Editor de Texto	8

1 Introdução

Este documento aborda as características estruturais do jogo a ser produzido pela Z-One Team. Serão detalhados o ambiente do jogo, isto é, em que momento da historia ele se passa e os elementos fundamentais que definem o jogo. Dentre os elementos especificados estarão os inimigos, torres, mercado de jogo, mapas, etc.

2 Premissa e Ambientação do Jogo

Século XV, Japão Feudal, os ninjas de todo mundo se reuniram para reunião secreta mais importante dos ninjas, a escolha do novo Mestre Supremo dos Ninjas. A reunião é no topo da Torre da Escuridão, uma torre de 12 andares que é o centro de comando de todos os ninjas.

Mas os detalhes dessa reunião chegaram aos ouvidos dos maiores inimigos dos ninjas: os piratas e os samurais. Eles resolveram aproveitar essa oportunidade para fazer um cerco e atacar os ninjas enquanto estão dentro de sua torre e acabar de vez com as maiores lideranças ninjas.

Você, como o novo mestre supremo dos ninjas, **Sr. Myagi**, deve guiar os seus ninjas até a base da torre se defendendo dos piratas e samurais que tem como objetivo invadir e matar o **Sr. Myagi**.

3 Elementos do Jogo

3.1 Waves

Cada andar possui um número de ondas contínuas de inimigos, podendo variar dependendo do andar. No início do jogo o jogador pode tomar o tempo que for necessário para gastar os recursos iniciais construindo suas torres. Após construir suas torres e se sentir satisfeito ele inicia o jogo e as ondas de inimigos atacam continuamente com intervalos entre as ondas.

O jogador possuirá a opção de mandar a proxima onda mais cedo. E quando o jogo estiver pausado o jogador não poderá construir torres.

3.2 Inimigos

3.2.1 Samurais

- **Soldado (Ashigaru):** Samurai de classe baixa equipado com um par de facas.
 - Velocidade: Média
 - Vida: Baixa
 - Resistência: Média
- **Retalhador (Batosai):** Samurai com sede de sangue, extremamente veloz e fatal, ficando equipado com duas katanas.

- Velocidade: Alta
- Vida: Baixa
- Resistência: Baixa
- **Mestre do Dojo:** Samurai experiente em batalhas equipado com uma katana.
 - Velocidade: Baixa
 - Vida: Média
 - Resistência: Alta
- **Espada-Santa:** Samurai que possui técnicas supremas, secretas e sobre-humanas.
 - Velocidade: Média
 - Vida: Alta
 - Resistência: Média

3.2.2 Piratas

- **Saqueadores:** Pirata comum equipado com um mosquete (Estilo: Normal).
 - Velocidade: Média
 - Vida: Baixa
 - Resistência: Média
- **Corsários:** Pirata saqueador de grandes embarcações, equipado com um sabre.
 - Velocidade: Alta
 - Vida: Média
 - Resistência: Baixa
- **Perna de pau:** Pirata experiente com deficiência em uma perna, equipado com um punhal.
 - Velocidade: Baixa
 - Vida: Alta
 - Resistência: Alta
- **Capitão:** Pirata comandante, extremamente cruel e cauteloso, equipado com um gancho e um sabre (Estilo:Boss).
 - Velocidade: Média
 - Vida: Alta
 - Resistência: Alta

3.3 Torres

As torres serão os Ninjas. Desde antigamente os ninjas são mortais inimigos de samurais e piratas. Durante o jogo os Piratas e Samurais tentam chegar até o mestre supremo dos ninjas. Cada ninja pode ser evoluído, aumentando seus atributos, como o dano causado e a velocidade de ataque.

3.3.1 Curto Alcance

- Katana Ninja
- Nunchaku Ninja
- Manriki Ninja

3.3.2 Longo Alcance

- Kunai Ninja
- Shuriken Ninja
- Ninja Jogador de Bombas

3.3.3 Armadilhas

- Armadilha de Urso
- Parede com Espinhos

3.4 Percurso

Cada mapa terá um caminho fixo que os inimigos irão percorrer. Cada mapa representa um andar e o jogo terá no total 12 andares. Em cada mapa também será definido o número máximo de torres que podem ser instaladas e o número máximo de ondas.

3.4.1 Posicionamento das Torres

O posicionamento das torres será variável. O jogador poderá posicioná-las em qualquer lugar no mapa em que se possa construir torres, ou seja, qualquer lugar que não seja o caminho ou obstáculos (pilastras, mesas, outras torres, etc). Os lugares possíveis de posicionamento de torres serão específicos de cada mapa.

3.4.2 Relação Torre X Inimigo

Samurais e piratas não podem destruir ninjas, com exceção do ninja superior.

3.5 Mapas

O jogo possuirá 13 mapas, sendo que 12 mapa representará um ambiente da torre (um andar) e um ambiente externo, o jardim. Os ambientes serão:

1. Jardim externo MATHEUS
2. Saguão de Entrada TIAGO
3. Sala de Treinos (Dojo) CHARLES
4. Sala de jantar PEDRO
5. Sala de Armas PEDRO
6. Sala de banhos CHARLES
7. Biblioteca MATHEUS
8. Sala das Artes Negras TIAGO
9. Armazém (onde os mantimentos são guardados) PEDRO
10. Dormitório MATHEUS
11. Masmorras TIAGO
12. Sala da Guarda CHARLES
13. Sala de Reuniões TIAGO

4 Mercado do jogo

4.1 Moeda

Para compra de torres e aquisição de melhorias das mesmas será necessária o gasto de pontos de experiência "exp". A experiência é adquirida ao destruir uma unidade inimiga (varia em relação a dificuldade da unidade inimiga).

4.2 Evolução de unidades

Cada torre em jogo terá três níveis de evolução (o primeiro nível é o estado inicial). Cada evolução terá um custo de experiência variável, onde quanto maior o nível, maior é o custo.

5 Interfaces

5.1 Audio

Haverá música de fundo para navegação de menus e enquanto não tiver acontecendo ataques para cada fase. Para cada ataque das torres será tocado um som da arma usada. Ex: o som de uma lança ao se usar uma Katana Ninja.

5.2 Video

O jogo terá uma visão over-head. As torres/inimigos no entanto serão vistas de lado. A câmera será fixa, assim como o mapa do jogo, durante todas as etapas do jogo. Terá o tamanho de 800x600 pixels, ou seja, o tamanho do mapa é no máximo do tamanho da tela.

5.3 Controles

Haverá o posicionamento das torres através da movimentação do mouse. O jogador mais experiente pode usufruir das teclas de atalho para selecionar uma torre e coloca-la no mapa apertando letras do teclado, assim a torre pode ser posicionada com o auxílio do mouse.

6 Plataforma

O Ninja Seige rodará apenas em ambiente Linux.

7 Audiência

O jogo foca pessoas de idade entre 10 a 30 anos de idade não importando o gênero. Jogadores com qualquer renda, que dispuserem de um computador, poderão se divertir com o Ninja Seige. O jogo também é voltado para o perfil mais casual, pois tem o objetivo de entreter a pessoa com um passatempo divertido.

8 Desenvolvimento

8.1 Linguagem compilada

Será utilizado o C++ como linguagem, pois contempla todos os recursos de baixo nível da linguagem C, como gerenciamento manual de memória, manipulação de ponteiros e linguagem próxima do nível de máquina, ao mesmo tempo em que contempla recursos de linguagem de alto nível, como orientação a objetos e uso de namespaces. Em concomitância com o C++ será utilizada a biblioteca SDL 1.2.15 para Linux. O ambiente de desenvolvimento será Ubuntu. O compilador a ser utilizado será o g++.

8.2 Linguagem de script

Será utilizado o Python 2.7 como linguagem de script pelo domínio que os integrantes de programação já possuem da linguagem e por já ser nativa de sistemas Linux. Python é uma linguagem multiparadigmas que permite programação estruturada, orientada a objetos e ainda permite programar com paradigma funcional, este que é visto em linguagens de nível bastante alto. Como é uma linguagem interpretada, questões que poderão sofrer grandes alterações durante o desenvolvimento (ex: Inteligência Artificial) serão desenvolvidas na linguagem Python, eliminando o tempo de re-compilação em cada modificação.

8.3 API's

8.3.1 API Gráfica - SDL

Simple Directmedia Layer (SDL): uma biblioteca multimedia e multiplataforma escrita em C, mas funciona nativamente com C++, a linguagem utilizada para desenvolver o jogo. Provê acesso ao audio, teclado, joystick e hardware de video de forma simples. <http://www.libsdl.org/>

8.3.2 API de Áudio

SDL Mixer: Biblioteca de audio utilizada em conjunto com SDL. Permite o desenvolvedor utilizar de forma simples várias amostras de som em conjunto da musica no jogo e simplifica a manipulação e execução de sons de vários formatos. http://www.libsdl.org/projects/SDL_mixer/docs/

8.3.3 API de Rede

Não será utilizado uma API de rede, pois não será necessário, visto que o jogo não possui modo multiplayer ou acesso a internet para realizar transações.

8.3.4 Manipulação de Arquivos

Imagens: *SDL_image, bibliotecaparamanipulaodeimagensutilizadaemconjuntocomSDL.* http://www.libsdl.org/projects/SDL_image/docs/

8.3.5 GUI

wxWidgets: biblioteca multiplataforma para confecção de GUI's com suporte para diversas linguagens, entre elas C++. www.wxwidgets.org/

8.4 Documentação

O Doxygen é uma ferramenta para documentação de código bastante completa e pode ser usada em C, C++, java, Python entre outras. Ele permite a criação de páginas html que documentam todo o código de uma forma elegante e completa como referência cruzada entre entidades e estruturas do código como também mostra partes do código em si. O doxygen ainda permite a geração de páginas man e documentos latex.

8.5 Ferramenta de Controle de Versão

O SVN é uma ferramenta para desenvolvimento de código em equipe utilizada para manter versões e builds do código em um único lugar. A utilização do SVN evita repetição e perda de versionamento de código, aspecto fundamental em projetos em grupo.

Utilizaremos o Google Code como repositório do código <<http://code.google.com/p/z-one/>>.

8.6 Editor de Texto

O editor de texto a ser utilizado será o gedit, no ambiente Linux, com os seguintes plugins:

- Autocomplete

- Class Browser
- Code Comment
- Gemini
- Snippets
- Terminal

A utilização dos plugins tem como objetivo auxiliar na tarefa de programação trazendo diversas funções de grande utilidade na hora de programar.