

DUKITAN

GuardianBot

“A Defesa de sua Liberdade!”

GAMES

Fortaleza, 12 de Abril de 2006



Índice

Apresentação.....	3
Visão do Jogo.....	3
Descrição do Cenário.....	4
Descrição dos Personagens.....	4
Progressão.....	5
Interação.....	6
Regras do Jogo.....	6
Pontuação.....	7
Ranking.....	7
Softwares Utilizados.....	8
Ambiente MS-Windows ®.....	8
Ambiente GNU/Linux.....	9
Bibliotecas.....	9
Equipe.....	10
Cronograma.....	12
Orçamento.....	13
Proposta de Entrega.....	14
Glossário.....	15
ANEXO.....	17
ANEXO I – Tarefas.....	17
ANEXO II – Gráfico de Gantt.....	18
ANEXO III – Arte Conceitual.....	19

Apresentação

GuardianBot é um jogo que visa levar as pessoas de forma fácil e divertida o conhecimento sobre a filosofia Open Source e os principais softwares que a representam. Para isso, o jogador controlará o “Bot”, o personagem principal do jogo e o responsável por manter a segurança e estabilidade no sistema. Com este objetivo ele se deslocará dentro do computador, examinando cada pacote de software que chega, analisando se permitirá a instalação no computador.

Visão do Jogo

Este documento visa proporcionar uma visão do jogo GuardianBot, onde o jogador por meio da diversão conhecerá sobre softwares livres além de exercitar sua capacidade de percepção e resposta durante o jogo.

O jogo será desenvolvido integralmente com ferramentas Open Source, com o intuito de promover o uso do Software Livre para criação de jogos, bem como garantir a portabilidade do projeto entre sistemas operacionais, visto que o jogo deverá ser multiplataforma e rodará em ambientes MS-Windows® e GNU/Linux.

Para isso fará uso de técnicas consagradas mundialmente como “melhores práticas” no desenvolvimento de Sistemas Orientados a Objetos e da linguagem C++ com auxílio da biblioteca multimídia SDL e do framework brasileiro para jogos GBF.

Este projeto, destaca-se por atuar em diversas linhas tecnológicas, entre elas:

- ◆ **Software Livre**, pelo uso de ferramentas livre em todas as etapas do desenvolvimento;
- ◆ **Multiplataforma**, pelo principio de poder desenvolver e rodar o GuardianBot em MS-Windows® e GNU/Linux;
- ◆ **RUP**, por seguir as indicações da metodologia de processos do RUP;
- ◆ **Orientação a Objetos**, por projetar e analisar utilizando a modelagem UML;
- ◆ **Jogos**
 - ◆ **Educativos**, por ter como principio levar conhecimento sobre os softwares livres;

- ♦ **Raciocínio**, por exigir que o jogador avalie os softwares que estão no jogo;
- ♦ **Tiro 2D**, por permitir interação clássica de tiro;

Descrição do Cenário

O jogo ocorrerá dentro de um computador, ver “anexo III – Arte Conceitual” que tentará refletir de forma gráfica as idéias principais na montagem do cenário. Para melhorar o entendimento, podemos dividir o cenário do jogo em três entidades especiais sendo elas:

- ♦ **Trilho do Bot**, caminho pelo qual o “Bot” se desloca dentro do jogo, para poder examinar os pacotes de software.
- ♦ **Baias de Entrada**, abertura pela qual os pacotes de softwares entraram no cenário, localizam-se na parte superior ou inferior do cenário.
- ♦ **Barramento de Leitura**, abertura onde os pacotes de softwares serão instalados, localizam-se posicionadas próximas as baias de entrada.

Descrição dos Personagens

1. Bot

Descrição:

É o guardião da máquina, é uma espécie de mini-robô capaz de se deslocar sobre trilhos, para avaliar os pacotes de softwares que estão adentrando pelas baias de instalação. Possui um raio laser o qual é capaz de cancelar a instalação de qualquer software.

Objetivo:

Impedir a instalação de software não livres, ou seja, não permitir que softwares piratas e proprietários sejam instalados no computador.

Controlado por:

Jogador Humano, é o personagem controlado pelo jogador. Onde o jogador se colocará no papel do guardião evitando que problemas ocorram no computador.

2. Pacote de Software

Descrição:

É a representação de um programa de computador. O qual poderá representar:

1. *Software Livre*, será identificado por “ícones” de programas livres famosos como: Firefox, Linux, OpenOffice.
2. *Software de Origem Duvidosa*, será representado por um CD sem identificação, simbolizando a origem desconhecida de programas.
3. *Software Pirata*, será representado por um CD contendo a clássica imagem da caveira de pirata.
4. *Software Proprietário*, será representado por “ícones” desconhecidos, criados especialmente para representar qualquer programa proprietário, sem no entanto citar empresas e produtos.

Objetivo:

Seu único objetivo é chegar ao barramento de leitura, para que o processo de instalação possa ser concluído.

Controlado por:

Computador, é o conhecido “NPC” ou “CPU”, é personagem controlado de forma automática pelo computador.

Progressão

A evolução dentro do GuardianBot se dará pela quantidade de pacotes de softwares instalados. O jogador deverá instalar uma quantidade de pacotes de “softwares livres” pré-determinados em cada fase para avançar a fase seguinte.

O calculo de quantidade de pacotes de “software livres” necessário em cada fase será baseado na seguinte formula:

$$(b * c) * f = \text{quantidade}$$

Onde “*b*” é a quantidade de barramentos de leitura, “*c*” é a representação de uma constante definida pelo valor 10 e “*f*” é o número da fase atual.

Durante a evolução do jogo, as fases poderão sofrer variações das seguintes opções:

1. Posição

1. Posicionamento dos trilhos do Bot;

2. Posicionamento dos barramentos de leitura;
2. Velocidade
 1. Movimentação do Bot;
 2. Movimentação dos Pacotes de Softwares;
 3. Intervalo de lançamento dos Pacotes de Softwares;
3. Quantitativo
 1. Número de barramento de leitura;
 2. Número de Pacotes de Softwares;

Interação

Para interação homem-máquina será utilizado o teclado do computador, por ser um periférico básico e existencial em qualquer computador domestico.

Sendo assim, seguindo as convenções clássicas usadas pelo jogos de computador, elegeremos as seguintes teclas, afim de tornar o jogo o mais intuitivo possível:

<i>Tecla</i>	<i>Ação</i>
Seta <i>Direita e Esquerda</i>	Permitirá que o Bot se desloque sobre os trilhos, os quais serão posicionados horizontalmente dentro do cenário do jogo.
Setas <i>Cima e Baixo</i>	Permitirá que o Bot aponte seu raio laser para os pacotes de softwares que estão entrando pelas baías superiores ou inferiores do cenário.
Control (do lado esquerdo)	Permitirá que o Bot realize o disparo com seu raio laser, a fim de cancelar a instalação do pacote de software.
P	Será utilizado para efetuar uma pausa momentânea no jogo, assim como para continua-lo enquanto estiver em modo pausado.

Regras do Jogo

Avanço de Fase:

Para avançar nas fases o jogador deverá permitir a instalação dos pacotes de softwares classificados como livres, ou seja, em cada fase uma certa quantidade de softwares livres deve ser instalada para passar a fase seguinte.

Fim de Jogo:

Será declarado fim de jogo para o jogador que em qualquer momento obtiver as seguintes situações:

- Quando a quantidade de software *proprietário e pirata* instalados for igual ou superior ao quintuplo da quantidade de *barramentos de leitura*;
- *A pontuação for igual a zero (00), após a primeira fase;*

Finalização do Jogo:

O GuardianBot na sua versão final, será considerado como um jogo virtualmente ilimitado, visto que o seu fim será determinado pela agilidade do jogador. Sendo assim, uma característica marcante no jogo será o ranking local e online, o qual impulsionará o espírito de competição do jogador.

Pontuação

Para título de pontuação teremos os seguintes casos:

<i>Pacote de Software</i>	<i>Motivo</i>	<i>Valor</i>
Software Livre	Quando Instalado	10
Software Livre	Quando ocorrer o cancelamento da Instalação	(-) 02
Software Pirata	Quando Instalado	(-) 10
Software Proprietário	Quando Instalado	(-) 05
Software de Origem Duvidosa	Quando Instalado	5 a (-) 10

Ranking

Para estimular o espírito de competição de forma saudável, o jogo GuardianBot manterá um ranking com as melhores pontuações dos jogadores. Para isso propõe-se:

1. Ranking Local
Será mantido um ranking em cada máquina de jogador contendo os 10(dez) melhores jogadores.
2. Ranking Online
Será mantido no site do jogo uma página contendo as pontuações dos jogadores,

possibilitando assim a criação de estatísticas como:

1. Os 20(vinte) melhores jogadores;
 2. Os jogadores que mais pontuaram(somatório de pontuação);
 3. Os mais viciados (quantidade de vezes que publicaram a pontuação)
 4. Os mais novos jogadores;
-

Softwares Utilizados

O GuardianBot será desenvolvido em C++, seguindo os padrões de excelência para produção de softwares orientados a objetos, com o intuito de ser multiplataforma, permitindo assim uma maior difusão entre os jogadores brasileiros.

Para cumprir esta meta fará extenso uso de técnicas, tecnologias e projetos opensources.

Apresentaremos a seguir a relação de software e suas respectivas finalidades, a serem utilizados para o desenvolvimento do GuardianBot.

Ambiente MS-Windows ®

1. BROffice, ferramenta de escritório utilizada para preparar a documentação do projeto, é a ferramenta oficial para elaboração de qualquer documento offline e apresentações futuras sobre o projeto e seu andamento.
2. CodeBlocks, ambiente integrado de desenvolvimento, será a IDE padrão utilizada durante o desenvolvimento do jogo e demais artefatos de softwares necessários.
3. MinGW, versão do GCC (Gnu Compiler Collection) para o ambiente MS-Windows®, desta forma será seguido mais fielmente as definições estabelecidas pelo ANSI C++, visando a portabilidade do código fonte entre sistemas operacionais.
4. BoUML, ferramenta de modelagem UML para C++. Permitirá a elaboração de modelos mais fieis as necessidades e especificações previstas para a elaboração do jogo.
5. GIMP, ferramenta gráfica para criação, edição e retoques de imagens 2D. Será utilizada para o desenho das imagens do jogo.
6. GanttProject, ferramenta para gerenciamento de projetos, permitir elaborar cronogramas, definir recursos e acompanhar a conclusão das tarefas nas diversas fases do projeto.
7. InnoSetup, ferramenta para geração de programas instaladores, permitindo que o jogo seja

facilmente instalado ou removido do computador do jogador sem nenhum transtorno.

8. TortoiseCVS, ferramenta gráfica para uso do sistema de controle de versão CVS. Permitindo o controle sobre os artefatos de softwares criados.

Ambiente GNU/Linux

1. BROffice, ferramenta de escritório utilizada para preparar a documentação do projeto, é a ferramenta oficial para elaboração de qualquer documento offline e apresentações futuras sobre o projeto e seu andamento.
2. CodeBlocks, ambiente integrado de desenvolvimento, será a IDE padrão utilizada durante o desenvolvimento do jogo e demais artefatos de softwares necessários.
3. GCC, Gnu Compiler Collection, compilador para o ambiente GNU/Linux, desta forma será seguido mais fielmente as definições estabelecidas pelo ANSI C++, visando a portabilidade do código fonte entre sistemas operacionais, visto que os compiladores GNU são famosos por seguirem as convenções estabelecidas pelo ANSI e por possuírem versões para os mais diversos sistemas Operacionais.
4. BoUML, ferramenta de modelagem UML para C++. Permitirá a elaboração de modelos mais fieis as necessidades e especificações previstas para a elaboração do jogo.
5. GIMP, ferramenta gráfica para criação, edição e retoques de imagens 2D. Será utilizada para o desenho das imagens do jogo.
6. GanttProject, ferramenta para gerenciamento de projetos, permitir elaborar cronogramas, definir recursos e acompanhar a conclusão das tarefas nas diversas fases do projeto.
7. CVSGui, ferramenta gráfica para uso do sistema de controle de versão CVS. Permitindo o controle sobre os artefatos de softwares criados.

Bibliotecas

1. SDL, biblioteca multiplataforma opensource, utilizada para finalidades de multimídia em geral, bastante utilizada por aplicativos openources, dentre eles diversos jogos no mundo Linux.
2. SDL_mixer, módulo adicional da SDL para som. Permitindo manipulações avançadas sobre os arquivos de música e efeitos sonoros.
3. SDL_image, módulo adicional da SDL para imagens. Permitindo manipulações mais avançadas sobre diversos tipos de imagens, dentre elas o formato PNG.
4. SDL_net, módulo adicional da SDL. Permite a comunicação via rede/Internet entre outros jogadores ou servidores.

5. **GBF**, framework opensource brasileiro, desenvolvido para o desenvolvimento de jogos 2D multiplataforma. Será a base do desenvolvimento do GuardianBot, visto que é mantido por integrante do projeto.
-

Equipe

A equipe para participação do JogosBr edição 2005/2006 caracteriza-se pela opção de “*grupo independente*”, o qual visa adquirir experiências com este projeto para que em um período de médio a longo prazo possam buscar investimentos que levem a formalização de uma empresa dedicada ao desenvolvimento de jogos no Ceará.

A equipe, responderá informalmente pelo nome de “DukItan Games”.

Conforme estabelecido pelo regulamento do JogosBr edição 2005/2006, será seu representante o analista de sistemas “David de Almeida Ferreira”, o qual é o encarregado perante o concurso de efetivar a inscrição e manter as comunicações entre o concurso e a equipe.

Para a otimização do processo de desenvolvimento deste projeto, estabelecemos que este projeto deverá ter os seguintes papéis:

1. Gerente de Projeto
2. Analista de Sistemas
3. Designer Digital 2D/3D
4. Escritor
5. GameDesign

Como o grupo encontra-se em estado de formação inicial/independente não será possível ter dedicação exclusiva de um profissional em cada papel, sendo assim um profissional poderá atuar em mais de um papel, desde que o mesmo encontre-se capacitado e que não comprometa as demais atividades.

Para compor a equipe pretendemos alocar inicialmente quatro profissionais de T.I. Onde na presente fase temos como selecionados e confirmados os seguintes profissionais:

1. David de Almeida Ferreira

- Brasileiro, Solteiro, 25 anos.
- Analista de Sistemas, formado pela União Educacional de Brasília (UNEB/DF), graduado em Processamento de Dados no ano de 2002/01.
- Exerce desde 2004 o cargo de Analista de Sistemas na Secretária de Administração do Estado do Ceará (SEAD).
- Atua com análise e desenvolvimento de software para plataforma desktop, web e cliente-servidor em C/C++, Java, Delphi, Php, PL/SQL, Perl.
- Atuou como administrador de portais relacionados com o desenvolvimento de jogos sendo eles PDJ(2000-2001), ArsLudica(2002), PDJ (2003), Unidev(2004).
- Desenvolve desde 2001 jogos/demos em diversas tecnologias entre elas: C/C++, Allegro, DirectX e SDL.
- Atualmente realiza pesquisas com softwares livres, participando de projetos para educação e divulgação das filosofias opensource e do software livre, além de ser o mantenedor do framework GBF (Gamework's Brazilian Framework), o qual é destinado a jogos multiplataforma usando softwares livres e técnicas de análise e projeto Orientado a Objetos.

2. Walisson Ferreira Pereira

- Brasileiro, Solteiro, 24 anos.
- Bacharel em Ciências da computação, formado pela Universidade Estadual Vale do Acaraú (UVA/CE)
- Tecnólogo em Eletromecânica pelo instituto centro de ensino tecnológico – CENTEC
- Atualmente é aluno do terceiro ano do curso de engenharia civil da UVA.
- Exerce desde 2004 o cargo de Professor Universitário da UVA, onde leciona: Arquitetura e Organização de Computadores, Sistemas Operacionais, Engenharia de Software além de Linguagem de Programação I e II (C++ e Java).

3. Francisco Wendel Almeida Cavalcante

- Brasileiro, Solteiro, 23 anos.
- Cursando Bacharelado em Computação na Universidade Federal do Ceará - UFC
- Técnico em Informática Industrial pelo CEFET-CE
- Curso de japonês básico(6 semestres) concluído no Centro de Línguas da UECE

- Certificações: Sun Certified Java Programmer, Oracle Certified JPROG 5, Sun Certified Instructor for Java, Oracle Certified Instructor for Java
 - Atua com análise e desenvolvimento de software para plataforma desktop, web e cliente-servidor em C/C++, Java, JSP, JavaScript, PL/SQL
 - Atualmente realiza pesquisa em tecnologias ligadas ao desenvolvimento de jogos, entre elas: DirectX, SDL, J2ME, OpenGL, etc
 - Exerce desde 2003 o cargo de "Técnico Judiciário – Especialidade Programação de Sistemas" no Tribunal Regional Eleitoral do Ceará
-

Cronograma

1. Tarefas.

No “ANEXO I” citamos e estimamos as tarefas iniciais para o desenvolvimento do projeto com base na experiência adquirida pela equipe tanto em análise e gerenciamento de projetos, quanto a preceitos estabelecidos pelo RUP.

2. Gráfico de Gantt

No “ANEXO II”, com base no levantamento inicial das atividades para a conclusão do projeto foi elaborado um gráfico de Gantt, visando facilitar a visualização de iniciação e finalização de cada tarefa assim como pontos de sobrecarga, afim de melhor dimensionar os riscos tornando-os mínimos durante a execução do projeto.

3. Observações

Com relação a locação de recursos humanos, de acordo com a caracterização da equipe, ou seja, atuando na forma de “grupo independente”, as tarefas definidas no cronograma, tomam como base a unidade em “dias”, no entanto a equipe envolvida tem uma disponibilidade parcial, fazendo com que o cronograma considere como dia de trabalho o equivalente a quatro horas corridas, ou seja, um dia de trabalho é definido pela quantidade de quatro horas diárias.

Orçamento

Para efeito de calculo de orçamento, estaremos levantado os gasto envolvidos na aquisição de produtos, despesas diversas e quadro de profissionais.

1. Aquisição Única

<i>Aquisição de Softwares</i>	<i>Motivo</i>	<i>Valor</i>
Microsoft Windows XP Professional Pt-Br	Licenciamento para poder efetuar compilação e testes em um ambiente legalizado.	480,00
Total		480,00

2. Quadro de Profissionais

(Observação: Os proventos para o quadro de profissionais é baseado na carga horária de vinte horas por semana)

<i>Perfil</i>	<i>Função</i>	<i>Valor</i>
Gerente de Projeto / Analista	Gerenciar prazos, atividades e recursos. Elaborar, definir casos de uso e demais artefatos UML.	1.000,00
Desenvolvedor / Programador	Desenvolver as ferramentas e bibliotecas necessárias para o desenvolvimento do jogo. Programar o jogo de acordo com as definições estipuladas pelo Analista e o Escritor.	1.000,00
Designer 2D/3D	Criar e editar as imagens a serem utilizadas pelo jogo, baseado nas definições do Escritor. Criação de material para divulgação. Criação de arte gráfica para confecção do manual e site.	800,00
Roteiro(Escritor) / Game Design	Elaborar a história do jogo, bem como o seu amadurecimento. Criação do Manual do jogo. Elaboração do conteúdo para o site do jogo.	800,00
Total		3.600,00

3. Despesas Diversas

<i>Reserva de Recursos</i>	<i>Valor</i>
Duas passagens terrestres de “ida e volta”, partindo de Fortaleza - Sobral	400,00
Uma passagem terrestre de “ida e volta”, partindo de Sobral – Fortaleza	200,00
Consultoria para o desenvolvimento do jogo e ferramentas (cinco horas)	500,00
Registro de Domínio e Hospedagem por 12 meses	500,00
Material de Apoio	400,00
Custos de Expediente	600,00
Total	2.600,00

<i>Recurso</i>	<i>Quantidade</i>	<i>Valor</i>	<i>Valor Total</i>
Aquisição de Softwares	1	480,00	480,00
Quadro de Profissionais	5	3.600,00	18.000,00
Reserva de Recursos	1	2.600,00	2.600,00
Total			21.080,00

Proposta de Entrega

Para participação do concurso JogosBr na sua edição 2005/2006, propõem-se a entrega de um demo jogável com quatro (04) fases, conforme data prevista no cronograma, com versões compiladas para MS-Windows® e GNU/Linux, desde que os prazos acordados nesta fase de iniciação do projeto, sejam honrados pelas partes envolvidas nas etapas de seleção, julgamento, divulgação e premiação dos projetos, assim como no repasse do custeio para produção do projeto de demo jogável, ou conforme acordo firmado entre as partes envolvidas.

Glossário

<i>Termo</i>	<i>Significado</i>
Bibliotecas	Conjunto de funções/operações que adicionam recursos extra para linguagem de programação.
C++	Linguagem de programação de uso geral. Caracterizada pelo uso de múltiplos paradigmas como: Programação Genérica, Orientada a Objetos, Procedural e Abstrata.
Dispositivo de Entrada	São dispositivos que fornecem dados para operações em um programa, também chamados de unidades de entrada (no inglês input/output - I/O) Alguns dispositivos de entradas de dados são: teclado, mouse, scanner, joystick, microfone.
Framework	Conjunto de funções e/ou bibliotecas que se colaboram internamente, criando um conjunto de regras e normas, as quais devem ser seguidas para desenvolver o software. Tipicamente, um framework pode incluir programas de apoio, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto.
GBF	Gamework's Brazilian Framework, framework brasileiro desenvolvido em C++ e SDL visando a criação de jogos multiplataforma com técnicas de Orientação a Objetos.
GNU/Linux	GNU/Linux ou simplesmente Linux, é um popular sistema operacional livre, composto pelo núcleo (kernel) Linux e pelas bibliotecas e ferramentas do projeto GNU.
Multiplataforma	Software multiplataforma, é o software que possui versões para diferentes sistemas operacionais.
Open Source	O software chamado Open Source, ou em português código aberto, é um tipo de software cujo código fonte é público. O software de código aberto respeita as quatro liberdades definidas pela Free Software Foundation.
Orientação a Objetos	A Orientação a Objeto é um paradigma de análise, projeto, modelagem e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas objetos.
Portabilidade	Portabilidade para a informática é a capacidade que um programa (software) tem de ser compilado ou executado em diferentes arquiteturas de sistemas computacionais (diferentes arquiteturas de hardware ou de sistema operacional).
RUP	O RUP, abreviação de Rational Unified Process (ou Processo Unificado da Rational) é um processo de Engenharia de software que provê técnicas a serem seguidas pelos membros da equipe de desenvolvimento de software com o objetivo de aumentar a sua produtividade. O RUP usa a abordagem da orientação a objetos em sua concepção, e é

<i>Termo</i>	<i>Significado</i>
	projetado e documentado utilizando a notação UML (Unified Modeling Language) para ilustrar os processos em ação.
SDL	SDL é a abreviatura de Simple DirectMedia Layer, uma biblioteca de funções de programação multimídia, que permite acesso as interfaces de som, teclado, mouse, joystick e vídeo 2D (framebuffer) e 3D (OpenGL). O SDL é particularmente popular em sistemas como o GNU/Linux ou FreeBSD onde é usado por muitas aplicações, desde programas para tocar vídeo e DVD até jogos.
Software Livre	Um software é considerado como livre quando atende as quatro liberdades defendidas pela Free Software Foundation, as quais são : execução, estudo, redistribuição e modificação. Tais liberdades não fazem nenhuma referência aos custos envolvidos. O fato de se cobrar ou não pela distribuição ou pela licença de uso do software não implica diretamente em ser o software livre ou não.
UC	“Caso de Uso” ou “Use Case” (UC), é um tipo de classificador representando uma unidade funcional coerente provida pelo sistema, subsistema, ou classe manifestada por seqüências de mensagens intercambiáveis entre os sistemas e um ou mais atores.
Versão Beta	É um produto que ainda está sujeito ao desenvolvimento mas que é lançado para um publico de forma restrita ou não, com o intuito de refinamento e testes de usabilidade.

Observação:

Alguns dos significados deste glossário foram extraídos da Wikipédia “A enciclopédia Livre”, localizada em:
<http://www.wikipedia.org>

ANEXO

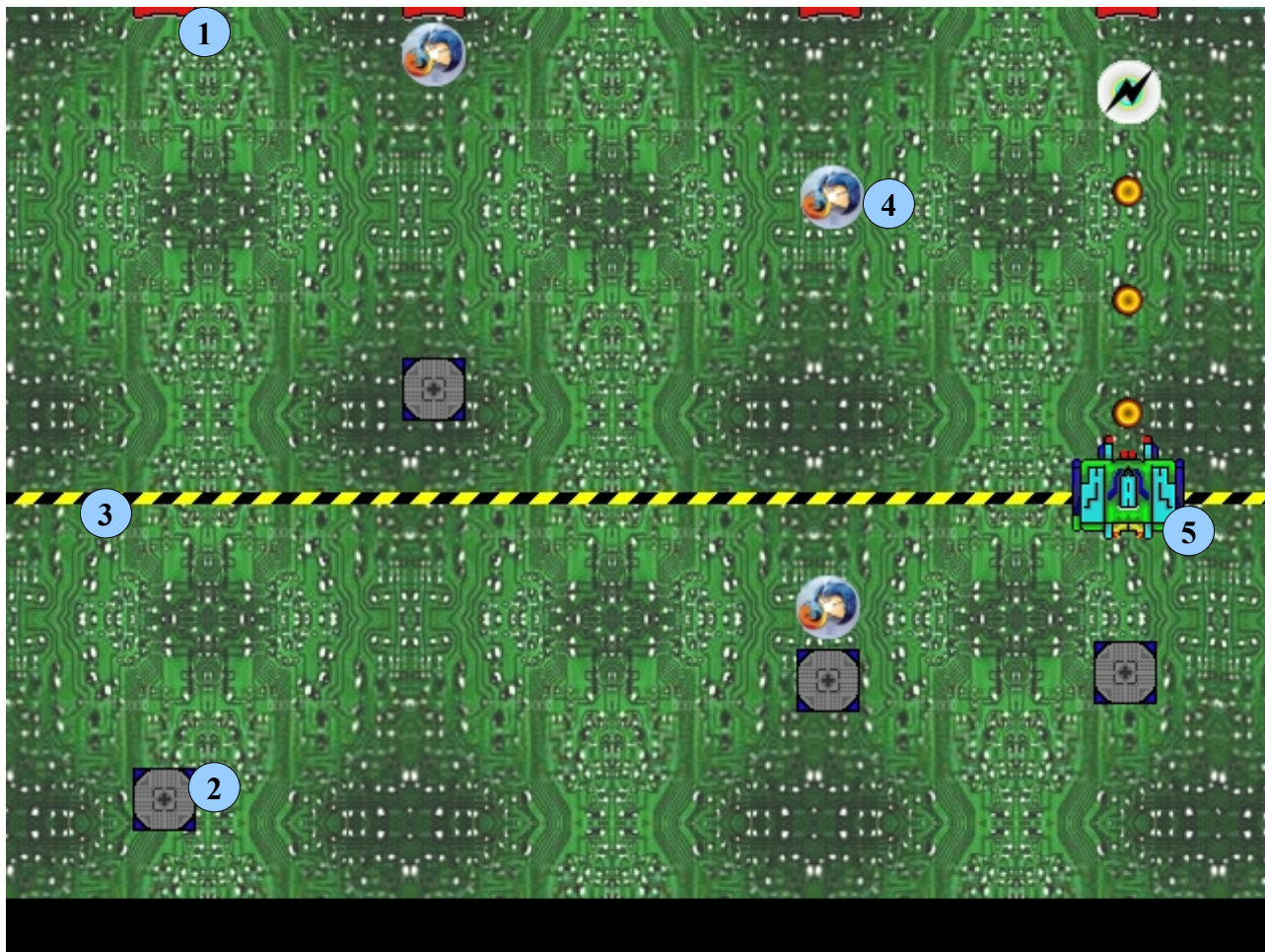
ANEXO I – Tarefas

<i>Nome</i>	<i>Início</i>	<i>Fim</i>	<i>Conclusão (%)</i>
GuardianBot	29/03/06	28/10/06	12
Iniciação	29/03/06	03/05/06	100
Definição do Roteiro	29/03/06	12/04/06	100
Documento de Visão	12/04/06	03/05/06	100
Elaboração	29/05/06	30/06/06	0
Definição dos Casos de Uso	29/05/06	30/06/06	0
Diagrama de Seqüência	12/05/06	30/06/06	0
Construção	16/06/06	29/09/06	0
Release 1	26/06/06	16/08/06	0
PKG-CDU01 - Implementação	26/06/06	05/08/06	0
PKG-CDU01 - Testes	03/07/06	16/08/06	0
Release 2	16/08/06	21/09/06	0
PKG-CDU02 - Implementação	16/08/06	14/09/06	0
PKG-CDU02 - Testes	21/08/06	21/09/06	0
Release 3 – Demo	21/09/06	27/10/06	0
Ajustes Necessários	21/09/06	27/10/06	0
Transição	03/08/06	28/10/06	0
Criação do Manual do Usuário	03/08/06	19/08/06	0
Criação do Site	17/08/06	07/09/06	0
Distribuição da Versão Beta	01/09/06	28/10/06	0

ANEXO II – Gráfico de Gantt



ANEXO III – Arte Conceitual



“Esta figura é apenas uma ilustração de como poderá ser o jogo, os elementos aqui apresentados são fictícios e são apenas para embasamento da conceituação do jogo.”

Legenda	
1	Baia de Entrada
2	Barramento de Leitura
3	Trilho do Bot
4	Pacote de Software
5	Bot