

**INSTITUTO DE ENGENHARIA NUCLEAR**

**ANDRÉ COTELLI DO ESPÍRITO SANTO**

**ESTUDO DA VIABILIDADE DO USO DE REALIDADE VIRTUAL NA  
CRIAÇÃO DE UMA FERRAMENTA DE APOIO AO PLANEJAMENTO  
DE SEGURANÇA FÍSICA EM INSTALAÇÕES NUCLEARES**

**Rio de Janeiro  
2013**

**ANDRÉ COTELLI DO ESPÍRITO SANTO**

**ESTUDO DA VIABILIDADE DO USO DE REALIDADE VIRTUAL NA  
CRIAÇÃO DE UMA FERRAMENTA DE APOIO AO PLANEJAMENTO  
DE SEGURANÇA FÍSICA EM INSTALAÇÕES NUCLEARES**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia Nucleares do Instituto de Engenharia Nuclear da Comissão Nacional de Energia Nuclear como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Engenharia Nuclear – Acadêmico em Engenharia de Reatores.

Orientador: Prof. Antônio Carlos de Abreu Mól, D. Sc.

Rio de Janeiro  
2013

COTE Cotelli do Espírito Santo, André

Estudo da Viabilidade do Uso de Realidade Virtual na Criação de uma Ferramenta de Apoio ao Planejamento de Segurança Física em Instalações Nucleares / André Cotelli do Espírito Santo – Rio de Janeiro: CNEN/IEN, 2013.

92 f.

Orientador: Antônio Carlos de Abreu Mól

Dissertação (Mestrado em Engenharia de Reatores) – Instituto de Engenharia Nuclear, PPGIEN, 2013.

1. Realidade Virtual, 2. Segurança física, 3. Câmeras virtuais.

ESTUDO DA VIABILIDADE DO USO DE REALIDADE VIRTUAL NA  
CRIAÇÃO DE UMA FERRAMENTA DE APOIO AO PLANEJAMENTO  
DE SEGURANÇA FÍSICA EM INSTALAÇÕES NUCLEARES

André Cotelli do Espírito Santo

DISSERTAÇÃO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO  
EM CIÊNCIA E TECNOLOGIA NUCLEARES DO INSTITUTO DE  
ENGENHARIA NUCLEAR DA COMISSÃO NACIONAL DE ENERGIA  
NUCLEAR COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA  
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA  
NUCLEAR – ACADÊMICO EM ENGENHARIA DE REATORES

Aprovada por:

---

Prof. Antônio Carlos de Abreu Mól, D. Sc.

---

Prof. Cláudio Márcio Nascimento Abreu Perreira, D. Sc.

---

Prof. Roberto Schirru, D. Sc.

RIO DE JANEIRO, RJ - BRASIL

OUTUBRO DE 2013

## **AGRADECIMENTOS**

Ao meu orientador e amigo, Dr. Antônio Carlos de Abreu Mól, pelo apoio e orientação no desenvolvimento de todo trabalho.

Aos meus colegas e amigos de laboratório Daniel Chelles, Daniel Mól, Deise Galvão, Eugenio Rangel e Larissa de Oliveira, que participaram diretamente deste trabalho e me ajudaram em todos os momentos.

A todos os colegas e professores da pós-graduação em Ciência e Tecnologia Nucleares pelo convívio e aprendizado.

À Comissão Nacional de Energia Nuclear, que através do Instituto de Engenharia Nuclear, proporcionou a estrutura necessária para a conclusão desta dissertação.

Ao meu pai Carlos Alberto do Espírito Santo e à minha mãe Norma Cardoso Cotelli, pelo amor, educação e por sempre acreditarem em mim. A vocês, minha eterna gratidão.

E finalmente à minha querida noiva Joyce Pires, por seu amor, carinho, paciência e incentivo nos momentos de cansaço. Sou muito grato a Deus por tê-la na minha vida.

## RESUMO

Nos últimos anos observa-se a importância de aprimorar a segurança física em instalações nucleares, principalmente devido ao crescente avanço do programa nuclear brasileiro. Nesse sentido, o presente trabalho teve como objetivo desenvolver uma ferramenta que permitisse a visualização e o planejamento de estratégias de ação em um ambiente virtual, visando melhorar a segurança. Para tal, foi criado um modelo virtual do Instituto de Engenharia Nuclear (IEN), localizado na Ilha do Fundão – Rio de Janeiro – Brasil. Esse ambiente é um modelo tridimensional, com representações próximas da realidade, onde os personagens virtuais (avatares) podem transitar e interagir em tempo real. Situações diversas que afetam a visibilidade e detecção de adversários, como iluminação natural e artificial, fenômenos climáticos, sombras e outros podem ser simulados com alto grau de realismo. Além disso, a ferramenta possui um sistema de vigilância através de câmeras virtuais, possibilitando o monitoramento do ambiente. Deste modo, esse sistema possibilitará simular estratégias de abordagem, permitindo uma avaliação dos procedimentos efetuados, bem como auxiliar no treinamento de segurança em instalações radioativas e nucleares.

**Palavras-chave:** Realidade virtual, segurança física, câmeras virtuais.

## **ABSTRACT**

In recent years the importance of improving the physical security of nuclear facilities is increasing, mainly due to the advancement of the Brazilian nuclear program. In this sense, the present work aimed to develop a tool that allows the visualization and planning of action strategies in a virtual environment, in order to improve security. For such, a virtual model of the Instituto de Engenharia Nuclear (IEN), located on Ilha do Fundão – Rio de Janeiro – Brazil, was created. This environment is a three-dimensional model, with representations akin to reality, where virtual characters (avatars) can move and interact in real time. Situations that could affect adversary visibility and detection such as natural and artificial illumination, climate and shadows can be represented with realism. In addition, the tool has a virtual CCTV surveillance system that allows environment supervising. Thus, the system makes possible strategies simulation, allowing an evaluation of the performed procedures as well as assisting in the training of security personal in nuclear and radioactive installations.

**Keywords:** Virtual reality, physical security, virtual cameras.

## ÍNDICE

### CAPÍTULO 1

<b>INTRODUÇÃO</b> .....	<b>1</b>
1.1. Objetivos.....	5
1.1.1. Objetivo Geral .....	5
1.1.2. Objetivos Específicos .....	5
1.2. Descrição dos Capítulos.....	6

### CAPÍTULO 2

<b>REALIDADE VIRTUAL</b> .....	<b>7</b>
2.1. Realidade Virtual .....	7
2.2. Ambientes Virtuais.....	9
2.3. Avatares .....	9
2.4. Ferramentas de Modelagem.....	10
2.4.1. Autodesk 3ds Max.....	11
2.5. Núcleos de Jogos .....	12
2.5.1. CryEngine 3.....	13
2.5.2. Unreal Engine.....	14
2.5.3. Source Engine.....	15

### CAPÍTULO 3

<b>UNITY 3D</b> .....	<b>17</b>
3.1. Interface.....	18
3.1.1. Project .....	19
3.1.2. Hierarchy .....	20
3.1.3. Inspector.....	21
3.1.4. Toolbar .....	22
3.1.5. Scene .....	23
3.1.6. Game .....	24
3.2. Terreno .....	25

3.2.1. Usando o Terreno .....	25
3.2.2. Navegando no Terreno .....	25
3.2.3. Editando Terreno.....	26
3.2.3.1. Altitude .....	27
3.2.4. Textura .....	29
3.2.5. Árvores .....	30
3.2.6. Grama .....	31
3.3. Luzes.....	32
3.4. Assets: Importação e Criação.....	32
3.5. Asset Store .....	33
3.6. Prefabs .....	33
3.6.1. Herança.....	33
3.7. Scripts.....	34
3.7.1. Introdução ao C#.....	34
3.7.2. Criação de Scripts .....	35
3.7.3. Exemplos de Scripts.....	36
3.8. UniStorm.....	39

## **CAPÍTULO 4**

<b>MÉTODOS .....</b>	<b>40</b>
4.1. Modelagem do Ambiente.....	40
4.1.1. Modelagem das Edificações.....	40
4.1.2. Modelagem do Terreno .....	43
4.2. Implementação de Funcionalidades .....	47
4.2.1. Características do Avatar .....	47
4.2.2. Sistema de Dia e Noite.....	49
4.2.3. Iluminação Artificial .....	51
4.2.4. Câmeras.....	52

## **CAPÍTULO 5**

<b>TESTES E RESULTADOS .....</b>	<b>54</b>
5.1. Instituto de Engenharia Nuclear .....	54
5.2. Avaliação do Ambiente Virtual: Proporcionalidade das Dimensões Físicas .....	55

5.2.1. Proporcionalidade da Instalação .....	55
5.2.2. Proporcionalidade dos Prédios e Portarias .....	56
5.3. Avaliação do Ambiente Virtual Quanto aos Ângulos de Visão.....	61
5.3.1. Visão de Pontos Estratégicos.....	61
5.3.2. Visão de Câmera.....	63
5.4. Avaliação Quanto aos Tempos de Percurso .....	65

## **CAPÍTULO 6**

<b>CONCLUSÃO .....</b>	<b>70</b>
------------------------	-----------

6.1. Trabalhos Futuros.....	71
-----------------------------	----

<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>
--	-----------

## LISTA DE ILUSTRAÇÕES

Figura 2.1 - Avatares de pessoas normais (Augusto, 2008). .....	10
Figura 2.2 - Ferramenta de Modelagem <i>Autodesk 3ds Max</i> .....	12
Figura 2.3 - Núcleo de jogo <i>CryEngine</i> . .....	14
Figura 2.4 - Núcleo de jogo <i>Unreal Engine</i> .....	15
Figura 2.5 - Núcleo de jogo <i>Source Engine</i> .....	16
Figura 3.1 - Interface do <i>Unity 3D</i> . .....	19
Figura 3.2 - Janela <i>Project</i> . .....	20
Figura 3.3 - Janela <i>Hierarchy</i> . .....	21
Figura 3.4 - Janela <i>Inspector</i> .....	22
Figura 3.5 - Janela <i>Scene</i> . .....	23
Figura 3.6 - Janela <i>Game</i> .....	24
Figura 3.7 - Adicionando um terreno. ....	25
Figura 3.8 - Propriedades do terreno. ....	26
Figura 3.9 - <i>Inspector</i> do Terreno.....	27
Figura 3.10 - Elevações e depressões feitas no terreno. ....	28
Figura 3.11 - Montanhas feitas com limitação de altura. ....	28
Figura 3.12 - Montanhas suavizadas. ....	29
Figura 3.13 - Terreno com textura.....	30
Figura 3.14 - Cenário repleto de árvores.....	31
Figura 3.15 - Terreno com grama.....	31
Figura 3.16 - Criação de um <i>script</i> . .....	35
Figura 4.1 - Imagem topográfica adicionada no <i>3ds Max</i> .....	41
Figura 4.2 - Modelagem das edificações.....	41
Figura 4.3 - Edificações texturizadas no <i>3ds Max</i> . ....	42
Figura 4.4 - Terreno plano criado no <i>Unity</i> . ....	43
Figura 4.5 - Alteração das dimensões do terreno.....	44
Figura 4.6 - Terreno texturizado. ....	44
Figura 4.7 - Características geográficas do terreno. ....	45
Figura 4.8 - Modelos tridimensionais inseridos no <i>Unity</i> . ....	46

Figura 4.9 - Terreno com sua respectiva vegetação. ....	46
Figura 4.10 (a) - Avatar padrão do <i>Unity 3D</i> em primeira pessoa; (b) - Avatar padrão do <i>Unity 3D</i> em terceira pessoa. ....	47
Figura 4.11 - Avatar desenvolvido para a simulação.....	48
Figura 4.12 (a) - Visão do avatar em primeira pessoa; (b) - Visão do avatar em terceira pessoa.....	49
Figura 4.13 (a) - Ambiente virtual durante o dia, com o posicionamento do sol afetando as sombras dos objetos; (b) - Ambiente virtual à noite, contendo um sistema de fases lunares e estrelas. ....	49
Figura 4.14 - Condições climáticas afetando o cenário.....	50
Figura 4.15 - Menu de controle de tempo e das condições climáticas. ....	51
Figura 4.16 - Iluminação artificial do ambiente virtual. ....	52
Figura 4.17 - Central de controle dentro do ambiente virtual. ....	53
Figura 5.1 - Localização do Instituto de Engenharia Nuclear (IEN, 2013).....	54
Figura 5.2 (a) - Imagem topográfica do IEN; (b) - Instituto de Engenharia Nuclear virtual.....	56
Figura 5.3 (a) - Prédio da instalação; (b) - Prédio correspondente modelado virtualmente.....	57
Figura 5.4 (a) - Foto da portaria principal do IEN; (b) - Portaria principal na simulação. ....	59
Figura 5.5 (a) - Foto da portaria secundária do IEN; (b) - Portaria secundária na simulação. ....	60
Figura 5.6 (a) - Humano de estrutura mediana usado como referência; (b) - Avatar ajustado para manter a escala de um humano. ....	61
Figura 5.7 (a) - Ponto considerado relevante da instalação; (b) - Ponto correspondente no ambiente virtual. ....	62
Figura 5.8 (a) - Foto de um ponto estratégico da instalação; (b) - Vista do ambiente virtual durante o dia; (c) - Vista do ambiente virtual no início da noite; (d) - Vista do ambiente virtual sofrendo alteração climática.....	63
Figura 5.9 (a) - Câmera posicionada na instalação; (b) - Visão da câmera virtual. ....	64
Figura 5.10 – (a) Portaria Principal; (b) Portaria Secundária; (c) LABRV - Laboratório de Realidade Virtual; (d) Reator Nuclear Argonauta; (e) Depósito de Rejeitos Radioativos.....	66

## LISTA DE TABELAS

Tabela 5.1 - Rotas de deslocamento no instituto. ....	65
Tabela 5.2 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 1.....	66
Tabela 5.3 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 2.....	67
Tabela 5.4 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 3.....	67
Tabela 5.5 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 4.....	68
Tabela 5.6 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 5.....	68
Tabela 5.7 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 6.....	69
Tabela 5.8 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 7.....	69

## LISTA DE LISTAGENS

Listagem 1 - Código em C#.....	36
Listagem 2 - Script de movimentação. ....	37
Listagem 3 - Script de autodestruição. ....	37
Listagem 4 - Script de saída do programa. ....	38
Listagem 5 - Script de sonorização. ....	38

## LISTA DE ABREVIATURAS E SIGLAS

2D – Bidimensional

3D – Tridimensional

AIEA – Agência Internacional de Energia Atômica

API – Application Programming Interface (Interface de Programação de Aplicativos)

CNEN – Comissão Nacional de Energia Nuclear

COM – Component Object Model

COM+ – Evolução do Microsoft Component Object Model

DLL – Dynamic-link library (Biblioteca De Vínculo Dinâmico)

IA – Inteligência Artificial

IEN – Instituto de Engenharia Nuclear

OpenGL – Open Graphics Library

OpenGL ES – Open Graphics Library for Embedded Systems

PC – Personal Computer (Computador Pessoal)

PWR – Pressurized Water Reactors (Reator de Água Pressurizada)

RV – Realidade Virtual

RVI – Realidade Virtual Imersiva

RVNI – Realidade Virtual Não Imersiva

## CAPÍTULO

### 1. INTRODUÇÃO

Na acepção mais abrangente, o termo Segurança Nuclear refere-se a uma série de medidas implementadas por Estados e Organismos Internacionais para garantir a segurança no manuseio, uso e estocagem de materiais radioativos (IAEA, 2006). Órgãos internacionais, como a Agência Internacional de Energia Atômica (IAEA) agem em conjunto com os países signatários para elaborar normas e procedimentos para evitar que acidentes aconteçam prestar assistência caso aconteçam e também para prevenir o acesso a materiais radioativos por parte de Estados ou indivíduos com intenções suspeitas ou sujeitas a questionamento. A responsabilidade de aplicar essas normas e procedimentos recai sobre cada Estado, mas, dadas as consequências de um ataque nuclear, um arcabouço de instrumentos legais internacionais permite um trabalho conjunto entre Estados para garantir a segurança em termos transnacionais (Brooks, 2005; IAEA, 2009).

Em vista de ataques terroristas em diversos países, a comunidade internacional reconhece que medidas novas e mais poderosas devem ser tomadas para proteção e prontidão para uma gama de cenários de ataques que incluam o terrorismo nuclear (IAEA, 2007).

A pior hipótese neste cenário é a aquisição, por parte de terroristas, de uma arma nuclear. Uma arma nuclear é um artefato de enorme custo e tecnologia, no entanto, de posse de material radioativo, pode-se confeccionar uma artefato de dispersão radiológica, a chamada “bomba suja”.

Uma “bomba suja” é uma bomba química convencional que contém material radioativo. A bomba é usada para dispersar este material, mas causa também os seus efeitos inerentes. No entanto, é improvável que, além dos efeitos diretos da explosão, estes artefatos causem sérios prejuízos à saúde e ao ambiente. Mortes e ferimentos seriam causados pela explosão. O material dispersado provavelmente resultaria em baixas doses de contaminação e em um leve aumento no risco de longo prazo de desenvolver carcinogênese (U.S. Nuclear Regulatory Commission, 2002).

Embora não seja significativamente mais perigosa que uma bomba convencional, uma “bomba suja” pode ter um profundo efeito moral devido à desinformação do público em geral sobre efeitos biológicos da radiação. Custos econômicos, no entanto, poderiam ser altos, afetando os arredores do local da explosão, prejudicando comércio e valores de propriedades, além dos custos de descontaminação do local (Peruzzo, 2012).

A segurança de uma instalação que armazena ou produz materiais radioativos deve ser foco de grande atenção e estratégias de segurança devem ser elaboradas, testadas e aprimoradas de forma a garantir que esses materiais permaneçam guardados e contabilizados e que não corram risco de cair em posse de adversários (Peruzzo, 2012).

Na elaboração de uma estratégia de segurança muitos fatores devem ser considerados e a tarefa é complexa. Inicialmente deve-se definir o que se deseja proteger, que grau de proteção cada parte da instalação deverá possuir e de quem ou de que se quer proteger, ou seja, estimar os possíveis adversários e suas capacidades (Stackpole e Oksendahl, 2010).

Esta estratégia deverá ser testada em exercícios reais para saber se existe uma adequação do plano de defesa com as condições reais da instalação, tais como: topologia do terreno da instalação, visibilidade das fronteiras e do terreno, número e aparelhamento da equipe encarregada da defesa (agentes de segurança), equipamentos de detecção e tempos de acesso dos agentes do ponto central de segurança aos locais onde tenham sido detectadas atividades suspeitas (Stackpole e Oksendahl, 2010).

O processo de desenvolvimento de uma estratégia de segurança, portanto, demanda tempo, recursos financeiros e o envolvimento da equipe que elabora a estratégia, da que a executa, da direção da instalação e até de serviços externos à instalação, como polícia, corpo de bombeiros e hospitais.

A direção da instalação nuclear, autoridades governamentais, forças policiais e auxiliares (Corpo de Bombeiros, Defesa Civil, serviços de saúde, Guarda Costeira, transportes públicos etc.) devem, então, criar uma estratégia de planejamento de segurança que contemple todas as atividades relacionadas com a proteção de pessoas, patrimônio e materiais radioativos e que, ao mesmo tempo, permita que o funcionamento normal da instalação nuclear (Connors, 2007).

As principais áreas de responsabilidade de um planejamento abrangente são (Connors, 2007):

- Força de trabalho: Quantos agentes de segurança serão necessários para guarnecer os postos de segurança; quais as respectivas qualificações e/ou especializações; quantos turnos os postos deverão estar guarnecidos.
- Comunicações: Quais tecnologias irão atender as necessidades de comunicação entre os muitos setores e indivíduos envolvidos para coordenação de esforços e tempo de resposta adequado às ameaças.
- Controle de acesso: Como organizar o acesso de pessoas à instalação nuclear, a movimentação nos espaços intermediários e garantir a ordem no seu interior; como garantir o acesso e a segurança de locais onde se armazenam ou produzem materiais radioativos; como remover pessoas do local (emergências médicas ou policiais).
- Inteligência: Como usar as informações dos serviços de segurança e como elas afetariam a implementação dos procedimentos planejados.
- Credenciamento: Como fazer o cadastramento e identificação dos agentes de segurança; como fazer o cadastramento e identificação de outros trabalhadores.
- Suporte administrativo e logístico: Como implementar ações necessárias para o funcionamento do plano de segurança (serviços de informática, comunicação, compras, entregas, transporte, suporte jurídico, alimentação, seguros etc.).
- Proteção de infraestruturas e utilidades críticas: Como garantir o fornecimento de eletricidade, água, esgotos, redes de computadores/comunicação etc.
- Bombeiros e serviços de saúde: Como permitir o pronto acionamento para atendimento, em caso de necessidade.
- Materiais perigosos, armas e armas de destruição em larga escala: Como detectar, confiscar, desabilitar e dispor de materiais e artefatos perigosos.
- Suporte tático e gerenciamento de crises: Como proceder em face da necessidade de intervenção por parte das forças ou dos agentes de segurança.

- Treinamento: Como habilitar os agentes de segurança para as atividades que exercerão.
- Controle de multidões: Como lidar com grande número de pessoas em protestos, arruaças e demonstrações.
- Planejamento de evacuação: Como retirar os indivíduos em segurança da instalação, caso seja necessário.

As consequências postuladas da não prevenção às principais ameaças e categorias de risco são: proliferação de materiais radioativos; danos a pessoas; danos à propriedade; responsabilidade legal por negligência; perda de reputação. Em casos de atentados envolvendo materiais radioativos os custos humanos e econômicos são altos e conhecidos (U.S. Nuclear Regulatory Commission, 2011).

Neste sentido, o planejamento das ações relacionadas às estratégias de segurança podem melhor ser executadas se treinamentos e simulações prévias destas ações puderem ser realizadas.

Nos últimos anos, tecnologias de Realidade Virtual vêm sendo usadas em diversas áreas do conhecimento para treinamento e simulação:

- Ambientes de salas de controle de plantas nucleares para estudos ergonômicos e de interfaces homem/sistema (Gatto, 2012).
- Avaliação (Mól et al., 2008a) e mapeamento (Freitas et al., 2010) de dose em sala de reator virtual, com alimentação online de níveis de dose de um reator real.
- Simuladores de instalação, configuração e utilização de equipamentos de segurança (Gonçalves et al., 2010), incluindo câmeras de vigilância e equipamentos de detecção de radiação.
- Mesa de controle virtual associado a um simulador de um PWR para treinamento simultâneo de operadores (Aghina et al., 2007). Esta aplicação inclui o uso de interfaces naturais (movimentos e voz) e visão estérea (Mól et al., 2008b; Jorge et al., 2007).
- O sistema VRdose (Louka et al., 2005) é uma ferramenta que permite a visualização da distribuição da radiação e as estimativas destas doses ocupacionais em instalações nucleares.

- Simuladores para estudos e treinamento de evacuação de ambientes de trabalho e industriais (Mól, Jorge e Couto, 2007; Mól et al., 2008c), operado por múltiplos usuários (cada usuário representado por um avatar) e com a presença de personagens autônomos.
- Treinamento e avaliação das defesas físicas e táticas de uma instalação nuclear (Augusto et al., 2009; Jorge et al., 2011).
- Desenvolvimento de ambientes tridimensionais para simulação e treinamento (Xi et al., 2009), permitindo reproduzir cenários operacionais e de emergência de uma usina de energia nuclear.

Neste contexto, é proposto neste trabalho a construção de um modelo computacional usando técnicas de realidade virtual que represente com o maior nível de detalhe possível uma instalação real, para que nesse modelo computacional seja possível realizar ensaios das estratégias de segurança física.

## **1.1. Objetivos**

### **1.1.1. Objetivo Geral**

O objetivo desta dissertação é verificar a viabilidade do uso de uma ferramenta de construção de jogos para criação de um modelo computacional (ambiente virtual) que sirva como ferramenta de auxílio para estudo e definição de estratégias de segurança física de instalações nucleares.

### **1.1.2. Objetivos Específicos**

- Avaliar o modelo computacional construído quanto ao grau de realismo de sua estrutura física ao ser comparado ao ambiente real;
- Avaliar o modelo computacional construído quanto algumas funcionalidades de segurança ao ser comparado ao ambiente real

## 1.2. Descrição dos Capítulos

O presente trabalho distribuiu-se em capítulos, sendo esta introdução o primeiro deles, apresentando da seguinte forma:

No Capítulo 2, é descrito o uso de realidade virtual, ambientes virtuais e avatares. Além disso, são apresentados as funções e benefícios das ferramentas de modelagem e núcleos de jogos, citando os principais disponíveis no mercado.

O Capítulo 3 aborda o núcleo de jogo utilizado nesta dissertação, o *Unity 3D*. As principais características desse núcleo e como utilizá-lo, desde a criação de terrenos até o desenvolvimento de scripts.

No Capítulo 4, são apresentados os métodos adotados para tornar o núcleo de jogo capaz de executar o tipo de simulação desejada, e as modificações e ajustes realizados para alcançar os objetivos.

No Capítulo 5, são discutidos os testes e resultados. Nele, fala-se do *Unity 3D* modificado e adaptado para a reprodução dos testes na instalação, o Instituto de Engenharia Nuclear – IEN, avaliando a proporcionalidade das dimensões físicas, os ângulos de visão e os tempos de percurso no ambiente virtual com o real.

Finalmente, o Capítulo 6 apresenta as conclusões e sugestões pra trabalhos futuros.

## CAPÍTULO

### 2. REALIDADE VIRTUAL

#### 2.1. Realidade Virtual

Realidade Virtual (RV) consiste no conjunto de técnicas e tecnologias destinadas a proporcionar ao usuário a sensação de integração com sistemas computacionais. Utilizando interfaces avançadas, objetiva-se inserir o usuário em um mundo virtual, possibilitando sua interação com este ambiente tridimensional de forma realista e em tempo real. Deste modo, RV pode ser vista como uma experiência do usuário com sistemas computacionais de três dimensões. Trata-se, portanto, de uma interface avançada de usuário para acessar aplicações computacionais em tempo real, permitindo a visualização, movimentação e interação nos ambientes sintéticos gerados pelo computador, utilizando para tal finalidade, dispositivos multi-sensoriais (Kirner e Siscoutto, 2007).

A utilização de Realidade Virtual começou com a Força Aérea dos Estados Unidos, no período pós Segunda Guerra Mundial, direcionando suas aplicações a construção de simuladores de voo. Devido à natureza desta aplicação inicial, a RV visava no seu início isolar, por completo, o usuário do mundo real. Atualmente encontra-se Realidade Virtual aplicada as mais diversas áreas do conhecimento, como experimentações virtuais, simuladores, treinamentos, maquetes virtuais, estudos ergonômicos, e, de forma acentuada, na indústria do entretenimento, com jogos, filmes e animações (Francis e Tan, 1999). Nessas novas aplicações, o conceito de Realidade Virtual não necessariamente está condicionado ao uso de artefatos como capacetes e luvas, alcançando, a contento, a finalidade de oferecer ao usuário a possibilidade de atuar virtualmente num ambiente sintético interativo.

Devido a grande variedade de aplicações, a RV foi classificada, segundo a forma como o usuário interage com o ambiente virtual, em Realidade Virtual Imersiva (RVI) e Realidade Virtual Não Imersiva (RVNI) (Burdea e Coiffet, 2003).

A RVI tem como objetivo inserir o máximo possível o usuário no ambiente do aplicativo, isolando-o do ambiente real ao seu redor. Para atingir esse objetivo, tecnologias especiais são utilizadas de modo a bloquear a percepção do usuário do

mundo real. Na RVI se procura interagir com o mundo virtual de forma mais intuitiva utilizar como, por exemplo, orientar-se olhando a sua volta com o movimento da cabeça, perceber o espaço como na visão binocular humana, reconhecer orientações sonoras a partir da localização da emissão de sua fonte (direção e distância), utilizar gestos e comandos de voz para comunicar-se (Haguenauer, Cunha e Filho, 2011).

Por RVNI o usuário tem acesso ao ambiente virtual sem isolar-se totalmente do ambiente real. Nestes casos, são utilizados dispositivos convencionais como monitor, teclado, mouse e caixas de som. Contudo, há interfaces nas quais o usuário é parcialmente isolado do ambiente real, num híbrido que mescla dispositivos convencionais e não convencionais. Nestes casos, é proposta uma classificação do tipo Realidade Virtual Semi-Imersiva (Burdea e Coiffet, 2003).

Entretanto, independentemente dessas classificações, há que destacar dois conceitos igualmente relevantes: Presença e Envolvimento. O conceito de presença refere-se ao estado de consciência, em se sentir estar no ambiente. Este sentido de presença decorre em grande medida do nível de imersão possibilitado pela interface, mas não deve ser exclusivamente creditado ao uso de tecnologias especiais como capacetes e luvas virtuais (Kirner e Siscoutto, 2007). O sentimento de presença está fundamentado no processo cognitivo do sujeito e de suas representações mentais do mundo e do ambiente por ele experimentado através de dada interface e, particularmente, no envolvimento com o aplicativo. De forma análoga com o que ocorre no campo da comunicação e da literatura, onde o leitor se sente vivendo as experiências do personagem, existem alguns aplicativos com baixo grau de imersão (sob o ponto de vista dos dispositivos especiais supracitados) alcançam um nível de envolvimento do usuário muitas vezes superior àqueles que utilizam aparatos tecnológicos avançados.

Não há dúvidas que o grau de envolvimento e o sentido de presença podem ser potencializados ou facilitados por dispositivos que aproximem o usuário da experiência do real, atuando sobre as sensações, que são os canais de comunicação e percepção humana do mundo (Goebbels, Lalioti e Göbel, 2003). Entretanto, sob o ponto de vista do envolvimento do usuário no ambiente, traduzido no sentido de presença, interfaces de Realidade Virtual não imersiva podem ser, se

bem utilizadas, mais eficientes que interfaces classificadas com maior grau de imersão.

## **2.2. Ambientes Virtuais**

Entende-se por ambientes virtuais os cenários reproduzidos por Realidade Virtual, onde são executadas todas as ações da aplicação. Estes ambientes são povoados por avatares que se deslocam e interagem com todos os elementos existentes (Stanney, 2002).

As simulações em Realidade Virtual são executadas nestes ambientes virtuais, que geralmente são construídos de acordo com ambientes reais, buscado estabelecer uma semelhança visual, respeitando proporções, disposição de objetos, metragens, etc. Segundo Augusto et al. (2007), estes cenários buscam respeitar as leis da física, garantindo que a interação do avatar com o ambiente virtual seja semelhante à interação humana com o ambiente real equivalente.

## **2.3. Avatares**

Denominam-se avatares, os personagens existentes dentro de ambientes virtuais, responsáveis por interagir com este mundo criado pela Realidade Virtual. Estas entidades transitam e controlam o cenário tridimensional em que estão inseridas, proporcionando ao usuário sua representação no ambiente em uso. Um avatar pode possuir qualquer forma, no entanto, aplicações que necessitam de forte identificação com a realidade utilizam-se de avatares com aparência humana (Figura 2.1). Para garantir uma maior verossimilhança, busca-se ainda, utilizar avatares caracterizados com o ambiente proposto, como por exemplo, utilizando uniformes, acessórios, e características físicas condizentes com a temática em questão (Augusto et al., 2007).

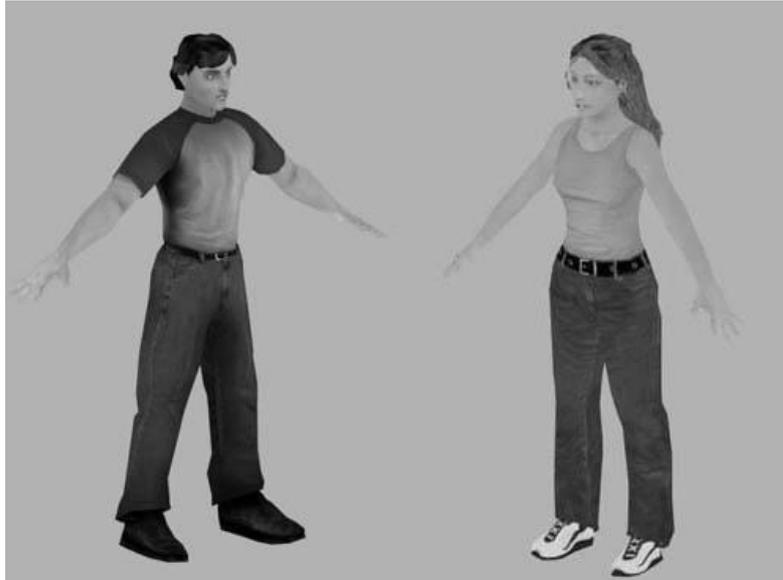


Figura 2.1 - Avatares de pessoas normais (Augusto, 2008).

## 2.4. Ferramentas de Modelagem

Modelagem tridimensional consiste no processo de desenvolvimento de representação gráfica computacional de superfícies ou objetos de três dimensões. Estes modelos produzidos podem ser estáticos ou animados, e podem ser desenvolvidos em diversos softwares especializados. Tais softwares são comumente denominados ferramentas de modelagem.

As ferramentas de modelagem têm por objetivo centralizar funções necessárias à construção de modelos tridimensionais, como, por exemplo, técnicas de construção de malhas (que dão formas aos objetos) por polígonos, vértices e bordas, facilitando o trabalho do desenvolvedor, que, desta forma, não precisa deter-se a representação matemática de cada modelo (Autodesk, 2010).

De forma simplificada, o processo de modelagem tridimensional consiste nas seguintes etapas: modelagem propriamente dita, dando forma ao objeto pretendido por meio de malhas; iluminação do modelo; texturização, correspondente a simulação de materiais e texturas sobre as superfícies dos modelos; renderização, que consiste na geração de imagem a partir de modelos do ambiente tridimensional, e, em alguns casos, animação (Calciolari, 2011).

### 2.4.1. Autodesk 3ds Max

O *Autodesk 3ds Max* consiste em uma ferramenta de modelagem que tem por objetivo ser uma solução completa não somente na modelagem como também na renderização, animação e simulações em três dimensões, atendendo assim as necessidades de diversos profissionais de computação gráfica no desenvolvimento de diversas aplicações, como jogos, maquetes virtuais, cinema, animações, etc (Harper, 2012). Com grande aceitação no atual mercado de computação gráfica, o 3ds Max, figura 2.2, é uma das principais ferramentas proprietárias para modelagem tridimensional.

Seus recursos de modelagem e texturização tridimensional fornecem grande flexibilidade à construção de malhas e superfícies, devido principalmente a sua característica de suporte a mapas vetoriais e a possibilidade de criação de objetos paramétricos. Segundo Calciolari (2011), por meio destes recursos é possível atribuir e editar as texturas aplicadas aos modelos, bem como aplicar sombreamento às criações.

No que concerne à animação 3D, esta ferramenta permite o controle e animação de personagens (avatares) para criação de jogos e vídeos. O *3ds Max* engloba por padrão diversas ferramentas auxiliares e suporte a *plugins*, que permitem, por exemplo, a inserção de gravidade, controle de colisões, inserção de vento, inserção de explosões, etc, fornecendo maior realidade às cenas geradas (Gahan, 2011).

De acordo com a *Autodesk* (2010) para garantir uma maior realidade aos modelos, este software permite ainda a aplicação de dinâmica e efeitos específicos em seus objetos, como dinâmica de fluxo de partículas, de movimentação de cabelos e pelos, movimentação de tecidos, e dinâmica de corpos rígidos.

Com formas flexíveis e integradas de renderização, o *Autodesk 3ds Max* permite ao desenvolvedor uma boa manipulação das imagens a serem produzidas.

Por fim, observa-se que a ferramenta disponibiliza uma interface de usuário amigável e configurável, possibilitando a organização do ambiente de desenvolvimento de acordo com as necessidades de cada profissional.

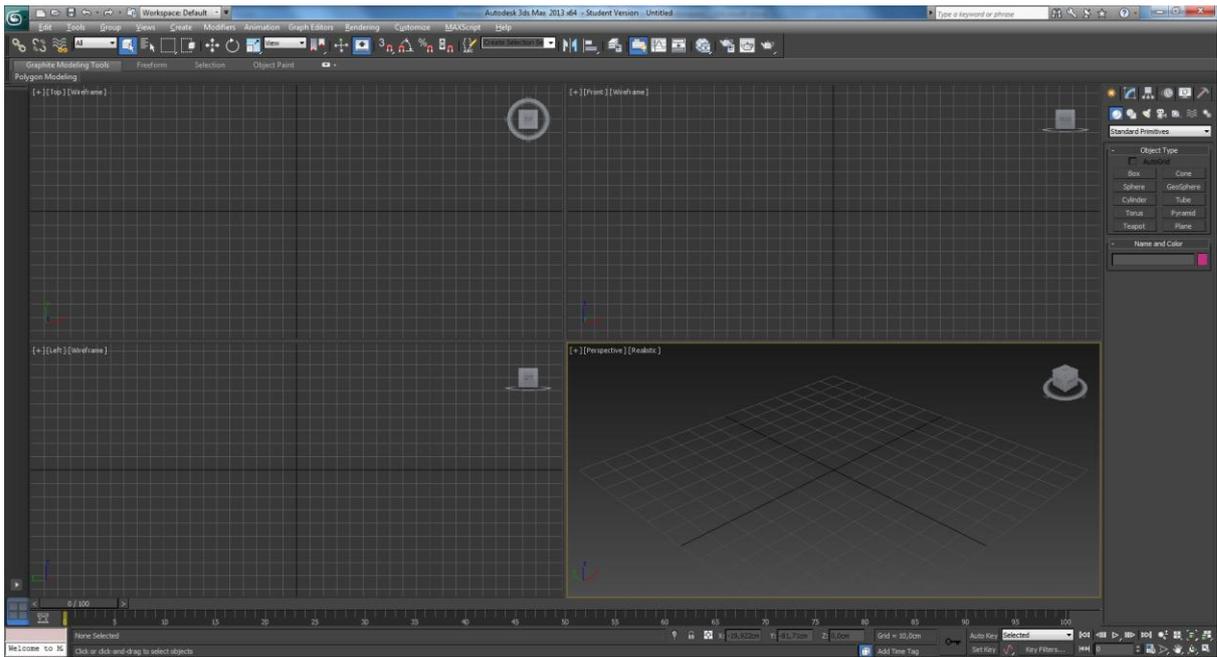


Figura 2.2 - Ferramenta de Modelagem Autodesk 3ds Max.

## 2.5. Núcleos de Jogos

Núcleos de jogos, também conhecidos como *game engines*, são programas destinados à produção de jogos eletrônicos. Para isso, esses programas centralizam diversas bibliotecas e funções necessárias ao desenvolvimento, proporcionando para o programador acesso rápido às ferramentas requeridas à criação destes jogos. Basicamente, tais softwares fornecem os recursos fundamentais para a criação de um jogo, principalmente um núcleo de renderização de gráficos em tempo real, acompanhado de funções de detecção de colisão e um núcleo de física, podendo também fornecer diversos recursos adicionais. Conforme Trenholme e Smith (2008), núcleos de jogos podem também ser adaptados e utilizados para o desenvolvimento de outras aplicações que necessitem de gráficos gerados em tempo real.

Visando fornecer ao desenvolvedor de jogos um ambiente completo e ágil, as *game engines* mais utilizadas são acompanhadas, por padrão, de ferramentas integradas de criação e edição de conteúdo, como editores de cenários e interpretadores de *scripts* para a programação de funções necessárias aos jogos e aplicações. Além destas, para caracterizar-se como um núcleo de jogos a ferramenta deve disponibilizar algumas funcionalidades fundamentais, como

reprodução de áudio, suporte a animações, reprodução de vídeos, capacidade de comunicação com outras instâncias da aplicação para realização de comunicação por rede (aplicações multiplayer), funções de inteligência artificial, acesso e interpretação de arquivos, gerencia de memória, etc (Augusto et al., 2007).

A utilização de *game engines* proporciona ainda outras facilidades, como reutilização de conteúdos desenvolvidos (objetos, scripts, avatares, etc.). A abstração de plataforma de desenvolvimento é outro importante benefício, pois permite que um mesmo jogo desenvolvido em um núcleo possa ser compilado para executar em diferentes computadores, vídeo games e arquiteturas.

Aplicações de Realidade Virtual *Desktop* baseadas em *Games* de Primeira Pessoa, embora não possam ser classificadas como RV imersiva, agregam outros elementos e valores com suas narrativas, possibilitando um alto grau de envolvimento. Este envolvimento está baseado na identificação dos usuários com seus avatares e com o contexto das ações e tarefas que se pretende realizar.

Atualmente, existem inúmeros núcleos de jogos no mercado, alguns deles gratuitos, já outros podem chegar a custar valores como U\$ 35.000. Dentre essas *game engines*, as principais disponíveis são: a *CryEngine 3*, o *Unreal Engine*, a *Source Engine* e o *Unity 3D*.

### **2.5.1. CryEngine 3**

A *CryEngine* é um poderoso núcleo de jogo desenvolvido nos últimos 10 anos pela empresa alemã *Crytek*. A desenvolvedora fez grandes alterações no núcleo original ao longo de vários anos e está atualmente na terceira versão, sendo por isso chamado de *CryEngine 3*.

De acordo com o Sean (2012), este núcleo de jogo inclui um conjunto integrado de ferramentas para manipular gráficos e elementos, permitindo a edição em tempo real. Uma das principais características dessa ferramenta é o desenvolvimento multiplataforma *PC*, *PlayStation 3* e *Xbox 360*. Com isso é possível que estudantes e desenvolvedores independentes construam aplicações gratuitas para computadores, desde que nada seja cobrado.

A *CryEngine 3* permite modelar e criar animações dentro do próprio editor ou importar de ferramentas de modelagem como o *Autodesk 3ds Max* e *Autodesk Maya*. O núcleo de jogo, figura 2.3, conta ainda com um grande número de recursos, como iluminação natural e sombras dinâmicas, reflexões em tempo real, renderização de águas e oceanos, entre outros.

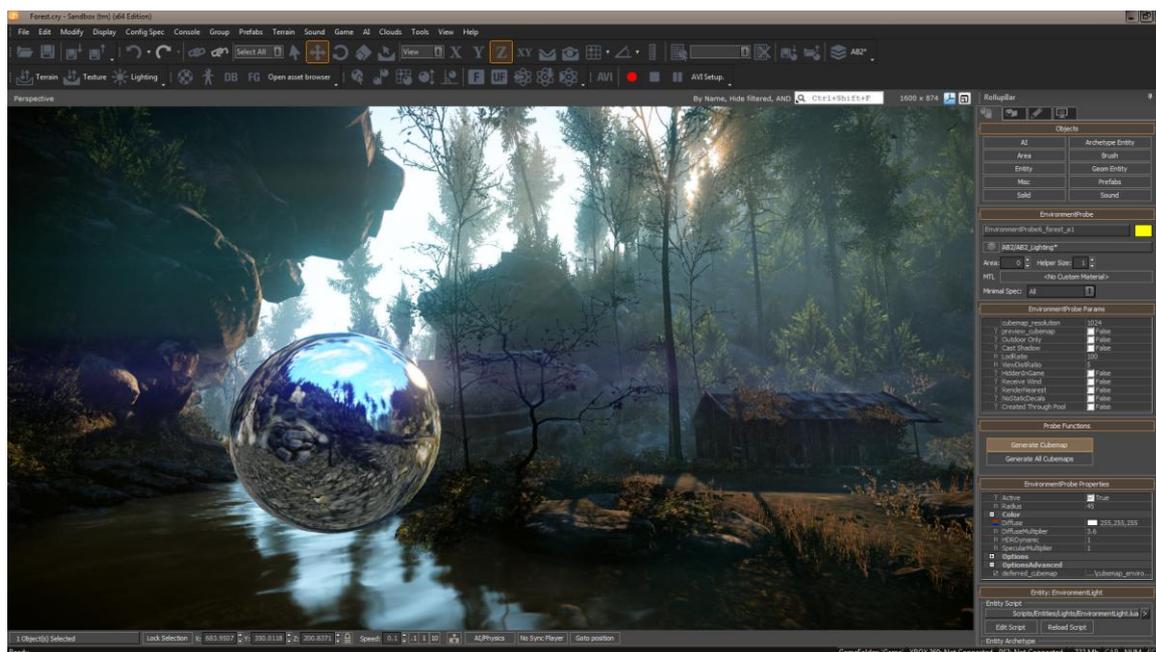


Figura 2.3 - Núcleo de jogo *CryEngine*.

## 2.5.2. Unreal Engine

O *Unreal Engine* foi desenvolvido pela *Epic Games* e usado pela primeira vez em 1998 no jogo *Unreal*. Desde então esse núcleo é uma plataforma completa de desenvolvimento de jogos, contendo visualizações avançadas e simulações tridimensionais ricas em detalhes. Este núcleo de jogo suporta múltiplas plataformas, incluindo *Windows*, *Linux*, *Mac OS X*, e os consoles *Xbox 360*, *PlayStation 3* e *Wii*. Tendo sido a base para diversas aplicações e games disponíveis no mercado, e estando atualmente na sua terceira versão (*Unreal Engine 3*).

O editor do núcleo de jogo tem todas as principais ferramentas de criação, oferecendo um sistema dinâmico de folhagem e iluminação com projeção de luzes em tempo real (Figura 2.4). Além disso, o editor ainda possui a ferramenta *Unreal*

*Content Browser*, que permite gerenciar todos os arquivos do projeto, como sons, animações e objetos, em uma única interface.

O *Unreal Engine 3* adicionou suporte ao *DirectX 11*, introduzindo uma técnica aprimorada de *anti-aliasing* para reduzir os serrilhados e suavizar a aparência dos objetos. Devido a isto, aumentando a qualidade gráfica e dando mais realismo as cenas (Mooney, 2012).

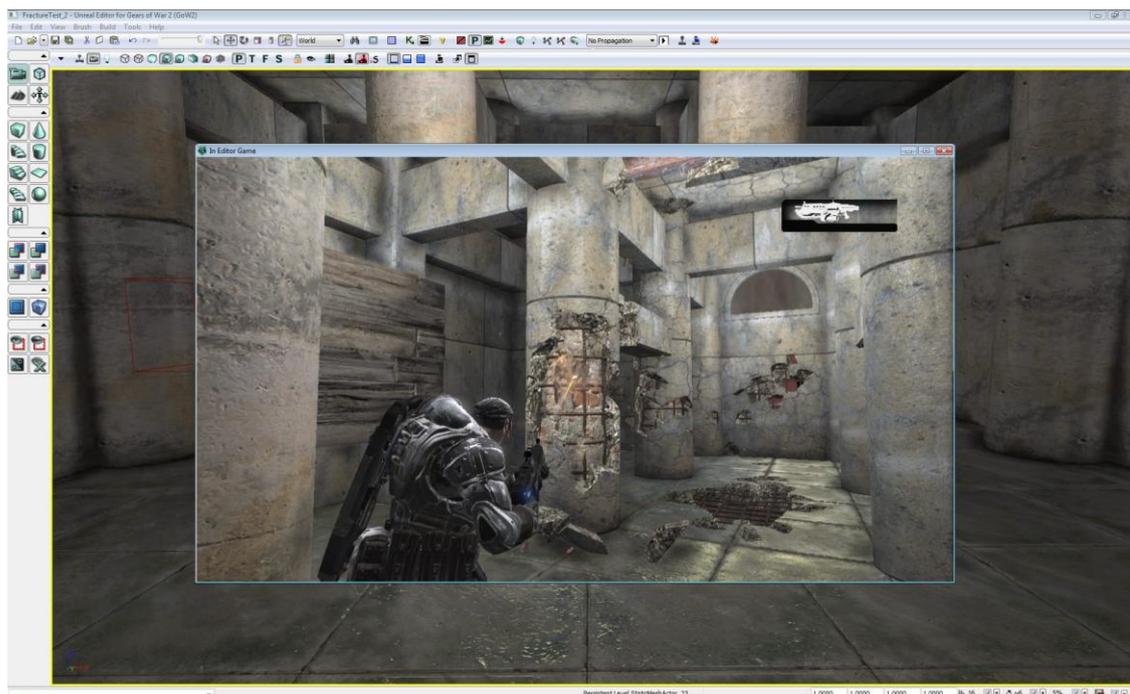


Figura 2.4 - Núcleo de jogo *Unreal Engine*.

### 2.5.3. Source Engine

A *Source Engine*, figura 2.5, é um núcleo de jogo 3D desenvolvido pela *Valve Corporation*. Suas características incluem um elevado grau de modularidade e flexibilidade, uma tecnologia de sincronia labial e expressão facial, um poderoso sistema de física e um eficiente modo online.

Este núcleo de jogo foi utilizado pela primeira vez no *game Half-Life 2*, lançado no final de 2004 (Saunders e Novak, 2012). Embora tenha passado anos desde o seu lançamento, o *Source Engine* continua sendo atualizado e está presente em alguns dos mais conhecidos títulos de jogos digitais da atualidade,

encontrando-se em diversas plataformas como *Windows*, *Linux*, *Mac OS X*, e nos consoles *PlayStation 3* e *Xbox 360*.

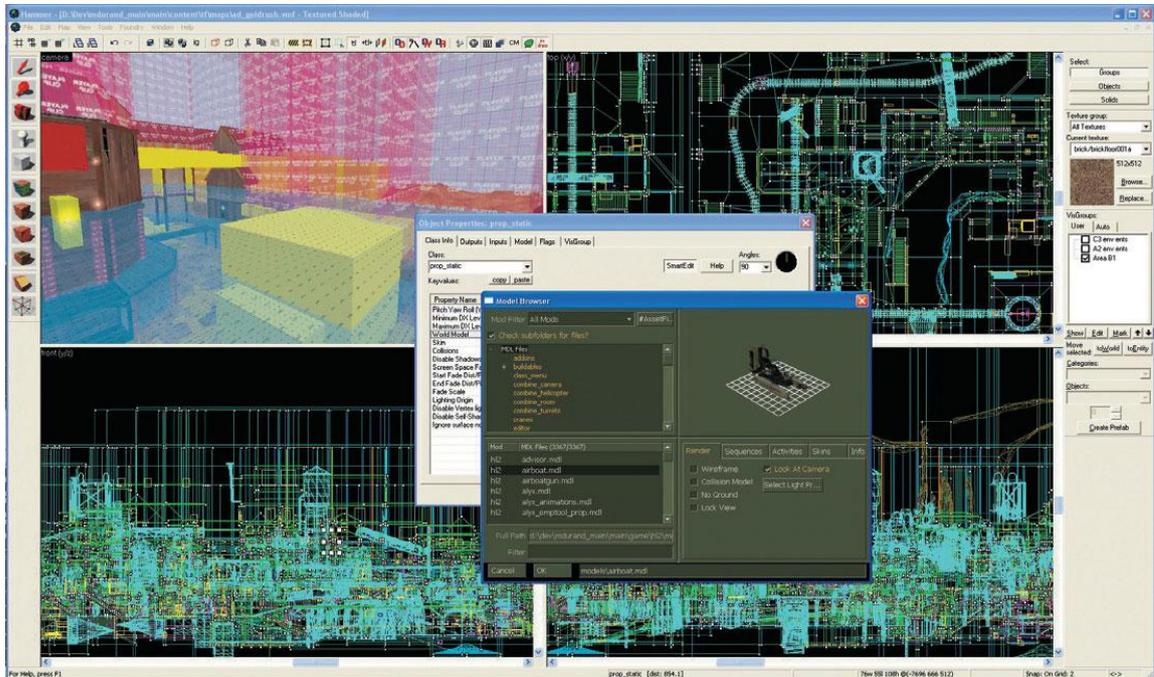


Figura 2.5 - Núcleo de jogo *Source Engine*.

## CAPÍTULO

### 3. UNITY 3D

O *Unity 3D* consiste em um robusto núcleo de jogo tridimensional, complementado por diversas funções necessárias ao desenvolvimento de jogos, como bibliotecas destinadas a inserção de física aos ambientes, iluminação e áudio interativo. Devido à alta qualidade de seus jogos, aliado a uma rápida curva de aprendizagem, esta ferramenta foi escolhida para cumprir um dos objetivos propostos nesta dissertação.

Prezando por portabilidade, este núcleo de jogo mantém como forte característica sua pluralidade de plataformas. Devido a isto, o desenvolvedor não precisa preocupar-se com as peculiaridades de cada sistema operacional ou *hardware*, visto que o *Unity* realiza todos os ajustes necessários a esta compatibilidade. Atualmente as plataformas suportadas, para quais os jogos desenvolvidos podem ser portados, são *iOS*, *Android*, *Windows*, *Blackberry 10*, *OS X*, *Linux*, *PlayStation 3*, *Xbox 360*, *Windows Phone 8*, *Wii* e *Wii U*. Além destas, os jogos podem ser portados para *Flash*, possibilitando sua execução em navegadores (web browsers) como *Chrome*, *Firefox*, *Safari* e *Internet Explorer* (Kyaw, Peters e Swe, 2013).

O *Unity* utiliza para renderização de seus objetos *Direct3D*, *OpenGL*, *OpenGL ES* e algumas APIs proprietárias. Sua renderização suporta mapeamento de relevo, reflexão e paralaxe, processamento de sombras dinâmicas, e efeitos pós-processamento. Uma importante característica da renderização no *Unity 3D* é a utilização do Método da Oclusão Implícita (*occlusion culling*), com a qual renderiza-se apenas os objetos que estão visíveis na cena, em tempo real, deixando todos os demais componentes do ambiente, que estão fora do alcance da câmera, de fora do processamento. Desta forma a evita-se a realização cálculos desnecessários, otimizando o processo e melhorando a performance do jogo (Blackman, 2011).

Este núcleo de jogo foi construído utilizando o framework de desenvolvimento Mono, implementação em código fonte aberto da plataforma .NET. Entretanto, para os desenvolvedores de jogos, o *Unity* apresenta-se como uma ferramenta multilinguagem de programação, suportando scripts em três linguagens: *JavaScript*

(linguagem de programação interpretada), *C#* (linguagem de programação orientada a objetos, desenvolvida pela Microsoft, nativa da plataforma .NET) e *Boo* (linguagem de programação multiparadigma que suporta orientação a objetos, programação imperativa e programação funcional, com sintaxe inspirada na linguagem *Python*). Tal característica fornece flexibilidade ao desenvolvimento, deixando a escolha de linguagem a cargo do programador (Goldstone, 2011).

Os desenvolvedores podem utilizar o *Unity 3D* sob duas formas de licenciamento, na versão gratuita, denominada apenas como *Unity*, e na versão proprietária, conhecida como *Unity Pro*, com um custo de \$ 1.500,00 (mil e quinhentos dólares). A versão proprietária tem como diferencial a disponibilização de recursos e funcionalidades adicionais, destinadas principalmente ao desenvolvimento de jogos profissionais, como filtros de áudio, streaming de vídeos, recursos adicionais de animação, suporte a textura 3D, etc.

### **3.1. Interface**

O núcleo de jogo *Unity 3D* possui uma interface simples e intuitiva, objetivando facilitar o desenvolvimento de jogos e simulações. Sua área de trabalho engloba janelas, onde cada uma possui uma função específica. Estas janelas são chamadas de *Views* e estão representadas na figura 3.1.

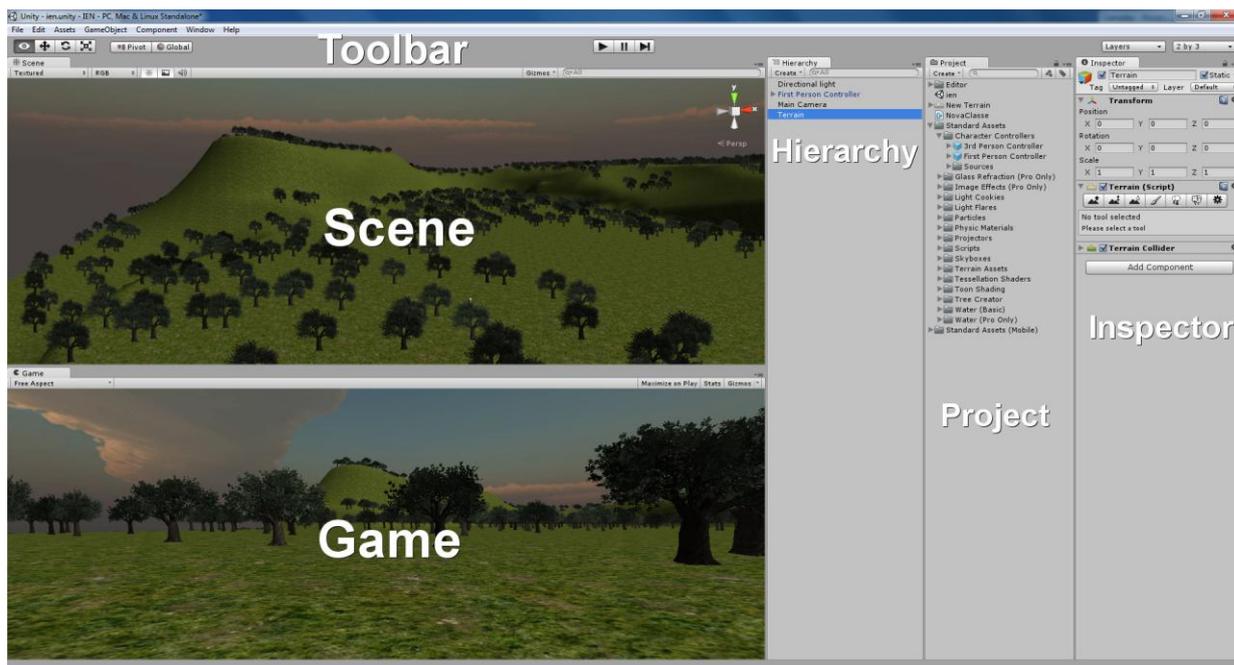


Figura 3.1 - Interface do *Unity 3D*.

### 3.1.1. Project

Na janela *Project View*, (Figura 3.2), são armazenados todos os arquivos (*assets*) que compõe o projeto. Esta é a interface para organização e manipulação de arquivos como cenas, *scripts*, modelos 3D, texturas, áudios, *Prefabs* e fontes. A estrutura exibida na janela é correspondente à subpasta *Assets* dentro da pasta do projeto.

Novos *assets* são adicionados ao projeto quando arrastados diretamente para o *Project*, ou então na aba *Assets* → *Import New Asset*. Alguns *assets* devem ser criados dentro do *Unity*, clicando com o botão no *Project* → *Create*.

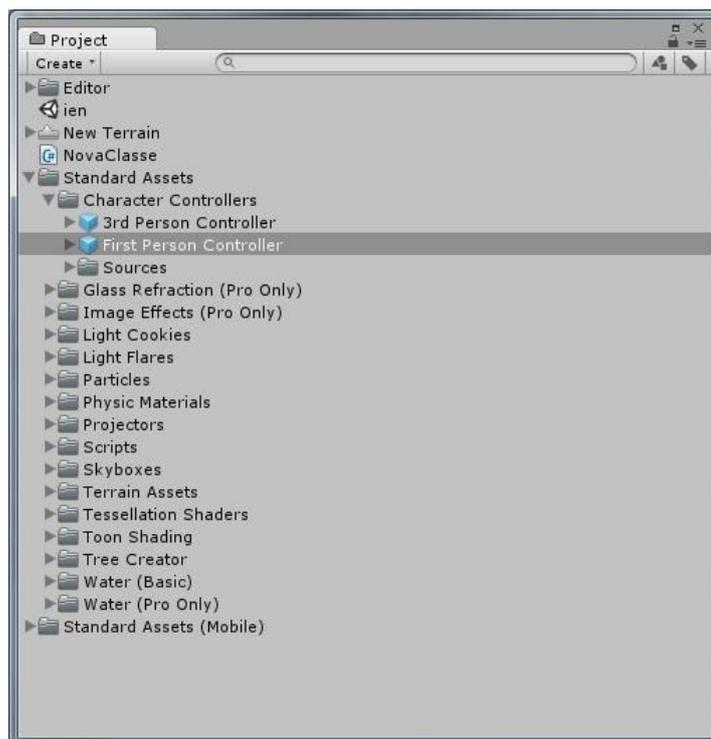


Figura 3.2 - Janela *Project*.

### 3.1.2. Hierarchy

A janela *Hierarchy View* engloba todos os elementos contidos no cenário (Figura 3.3). Através desta é possível organizar e visualizar a hierarquia entre os diversos objetos (*GameObjects*) que compõe a cena. Além de adicionar e remover objetos da cena, a aba permite que estes sejam conectados, esse conceito é conhecido como *Parenting*. Para criar um *GameObject* dependente (*child*), basta arrastá-lo para cima do objeto que o irá comandar (*parent*) e este irá herdar suas características.

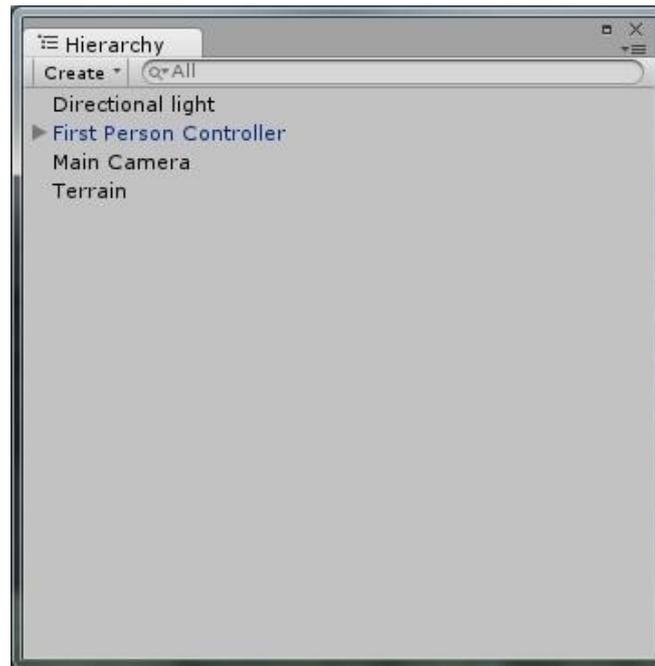


Figura 3.3 - Janela *Hierarchy*.

### 3.1.3. Inspector

No *Inspector* encontram-se todas as informações de configurações do *Game Object* selecionado, incluindo seus componentes e suas respectivas propriedades (Figura 3.4). As propriedades do objeto mostradas podem ser diretamente modificadas, inclusive variáveis de scripts. Além disso, permite ajustar atributos de componentes durante a execução da aplicação.

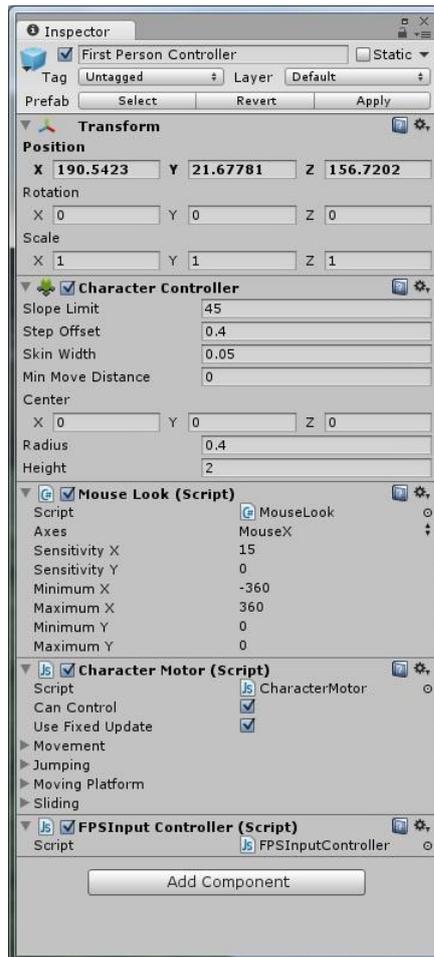


Figura 3.4 - Janela *Inspector*.

### 3.1.4. Toolbar

A barra de ferramentas é composta por cinco controles básicos. Cada parte corresponde a uma função diferente do editor.



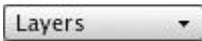
*Transform Tools*: ferramenta de transformação utilizada na *Scene View*.



*Transform Gizmo Toggles*: altera a exibição do *Scene View*.



*Botões Play, Pause e Step*: utilizados com o *Game View*.



View.

*Layers Drop-Down*: controla quais objetos são mostrados na *Scene View*.



*Layout Drop-Down*: altera a disposição de todos os *Views*.

### 3.1.5. Scene

A *Scene View*, figura 3.5, corresponde a parte interativa do projeto. Nela é possível editar e posicionar o ambiente, os personagens, as câmeras, e todos os elementos que compõe a cena. Nesta janela, pode-se manipular os objetos através das opções de arrastar e soltar com o mouse, semelhante à ferramentas de modelagem 3D.

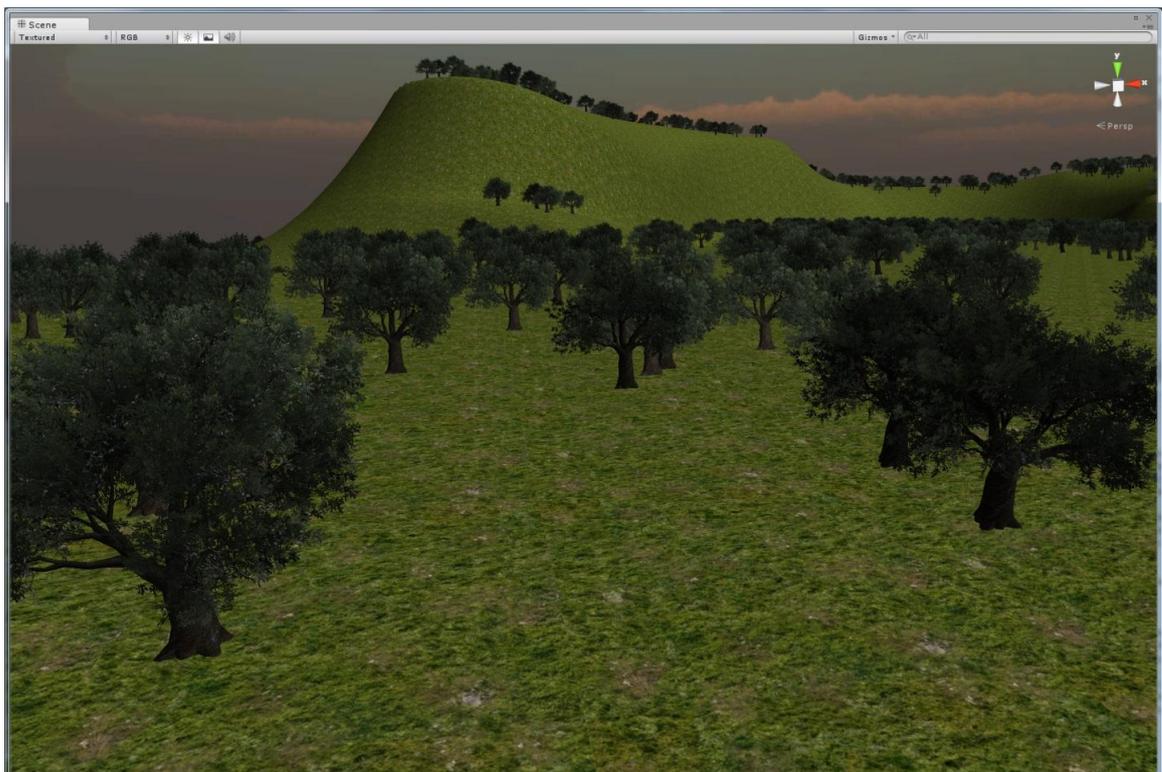


Figura 3.5 - Janela *Scene*.

A barra de controle da *Scene View* possui diversas formas de navegação e possibilita alterar a forma em que os objetos são renderizados no editor. Além de

contar com recursos como a visualização da iluminação e sons durante a construção da aplicação.

### 3.1.6. Game

O *Game View* é janela que renderiza a câmera presente no avatar (Figura 3.6). É responsável pela visualização do projeto na forma em que será exibido quando finalizado, funcionando como uma prévia do comportamento dos diferentes elementos que o compõe. A simulação é acionada ao selecionar o botão *Play* do *Toolbar*. Qualquer mudança feita durante esta execução é temporária, sendo resetada ao sair deste modo.



Figura 3.6 - Janela *Game*.

Na barra de controle do *Game View* pode se visualizar várias informações estatísticas sobre a simulação, tais como número de frames por segundo, tempo de processamento, memória de textura utilizada, entre outras. Estes dados são importantes para uma posterior otimização, caso seja necessário.

## 3.2. Terreno

### 3.2.1. Usando o Terreno

No *Unity* um terreno pode ser adicionado acessando na barra de menu: *GameObject* → *Create Other* → *Terrain*, figura 3.7. Desta forma o terreno é adicionado no cenário, aparecendo na janela de *Project* e *Hierarchy*.

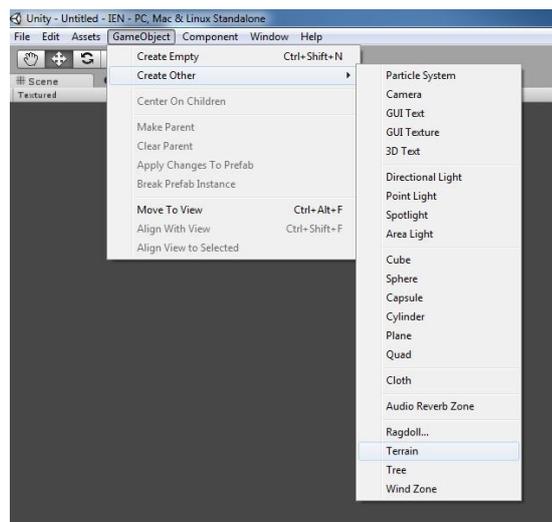


Figura 3.7 - Adicionando um terreno.

### 3.2.2. Navegando no Terreno

Terrenos funcionam de forma diferente do que outros *GameObjects*, enquanto estes podem sofrer transformações livremente, como translação, escalonamento e rotação. No *Unity*, somente é possível realizar a movimentação do terreno de acordo com as coordenadas (x, y e z), modificando os valores de *Transform Position*, figura 3.8.

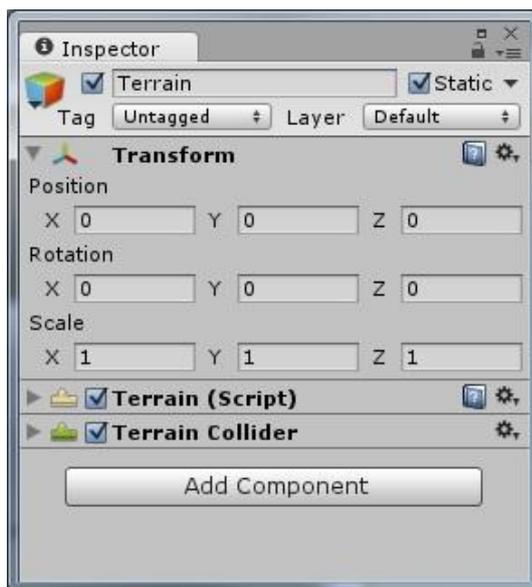


Figura 3.8 - Propriedades do terreno.

### 3.2.3. Editando Terreno

Na janela *Inspector* do terreno (Figura 3.9) são exibidas algumas ferramentas de edição. Cada ferramenta altera o terreno de uma forma diferente, podendo modificar a altura, pintar o mapa e colocar detalhes como árvore e rochas.

A maioria das ferramentas utiliza um pincel, exibido quando a ferramenta é selecionada. Após escolher a ferramenta e o pincel, entre os vários modelos disponíveis, é necessário posicioná-lo no terreno e pintar para iniciar as alterações em tempo real.

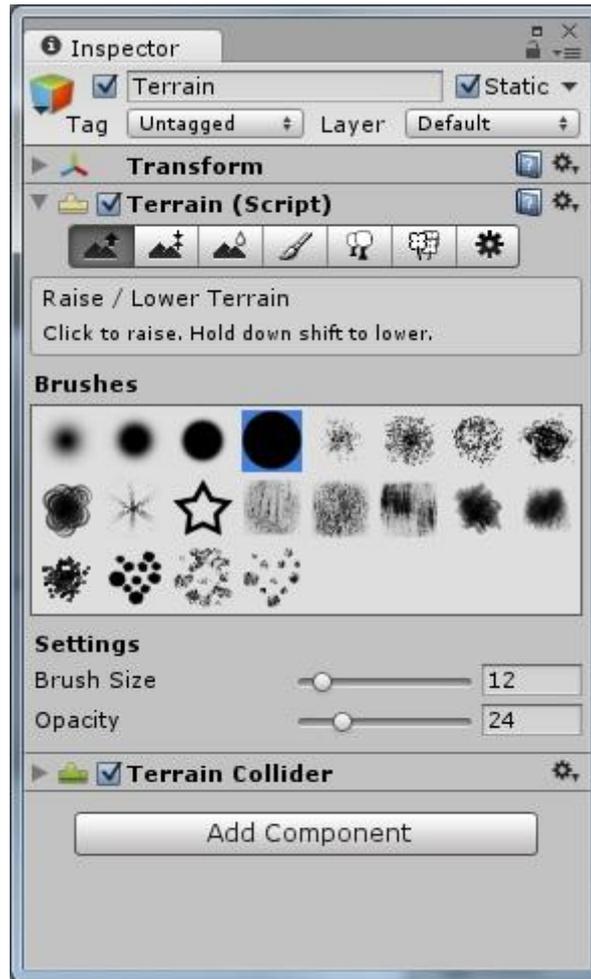


Figura 3.9 - *Inspector* do Terreno.

#### 3.2.3.1. Altitude

A ferramenta *Raise Height*, localizada no *Inspector*, é utilizada para modificar a altitude (Figura 3.10). Com esta ferramenta, o terreno sofre uma elevação ao clicar com o mouse, ganhando altura a cada clique. Para diminuir esta altura, simplesmente segure a tecla *Shift* e use o pincel.

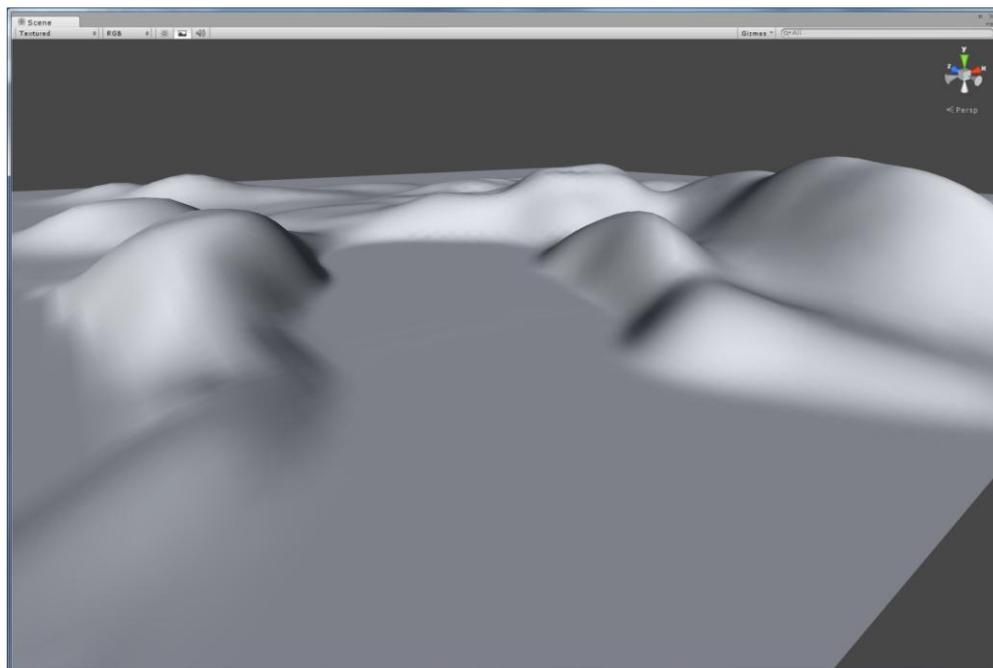


Figura 3.10 - Elevações e depressões feitas no terreno.

Existe outra ferramenta capaz de alterar o terreno, chamada *Paint Height*. Esta permite especificar a altura desejada e mover qualquer parte do solo para tal altura, figura 3.11.

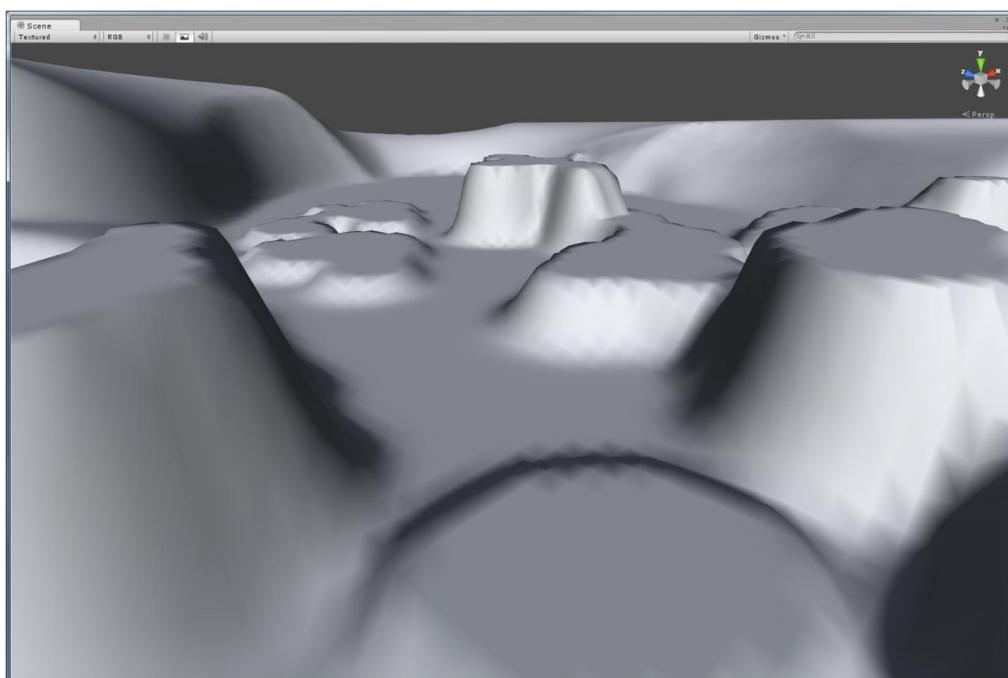


Figura 3.11 - Montanhas feitas com limitação de altura.

Outra ferramenta do terreno é a *Smoothing Height* que permite ao usuário, após definir a altura desejada, suavizar qualquer diferença de altura. Com isso deixando as arestas mais arredondadas, como mostrado na figura 3.12.

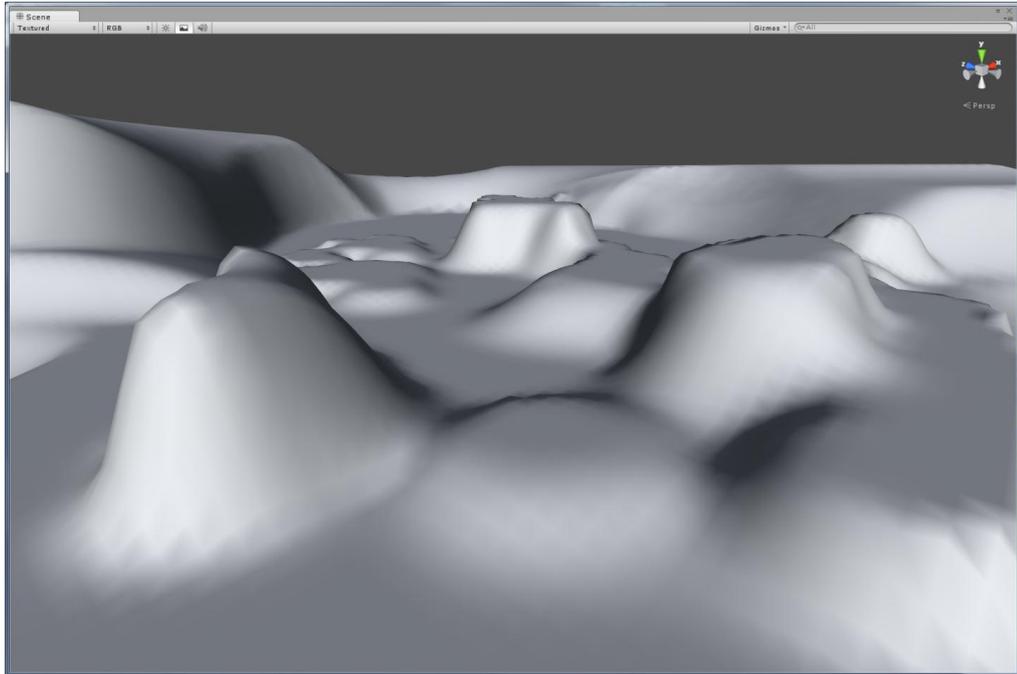


Figura 3.12 - Montanhas suavizadas.

#### 3.2.4. Textura

A textura é utilizada para decorar o terreno dando uma aparência real, apresentada na figura 3.13. Também conhecida como *Splat Maps*, possibilita utilizar uma mesma textura repetidamente no mapa e misturá-la fazendo suaves transições de um mapa para o outro. Para usar a textura faz-se necessário selecionar uma, ir até o terreno na *Scene View* e pintar.

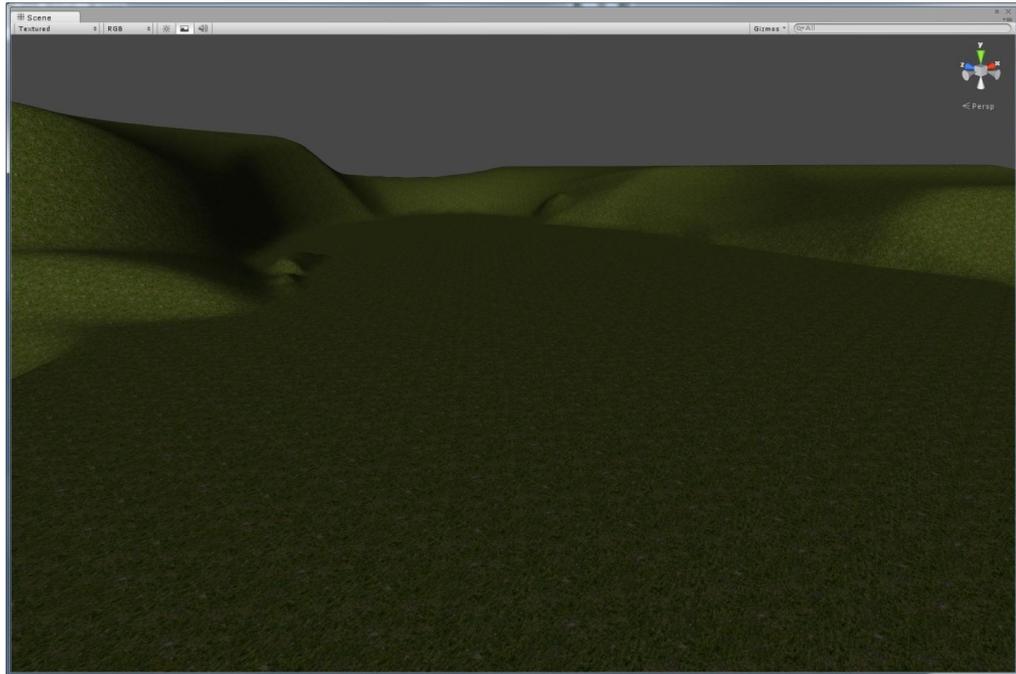


Figura 3.13 - Terreno com textura.

### 3.2.5. Árvores

Uma importante característica do *Unity* é seu suporte especial para árvores. Esse suporte funciona renderizando as árvores próximas da câmera em 3D e transformando as mais longe em 2D. Para que estas sejam vistas do ângulo certo, as árvores 2D são atualizadas automaticamente para se orientar em relação à câmera.

No *Inspector* existe uma ferramenta chamada *Place Trees*, que possibilita adicionar árvores pelo cenário. Após selecionar uma árvore da biblioteca disponível, basta clicar em qualquer local do terreno para colocá-la. Além disso, a ferramenta ainda conta com o botão *Mass Place Trees*, utilizado para adicionar uma floresta inteira, figura 3.14.

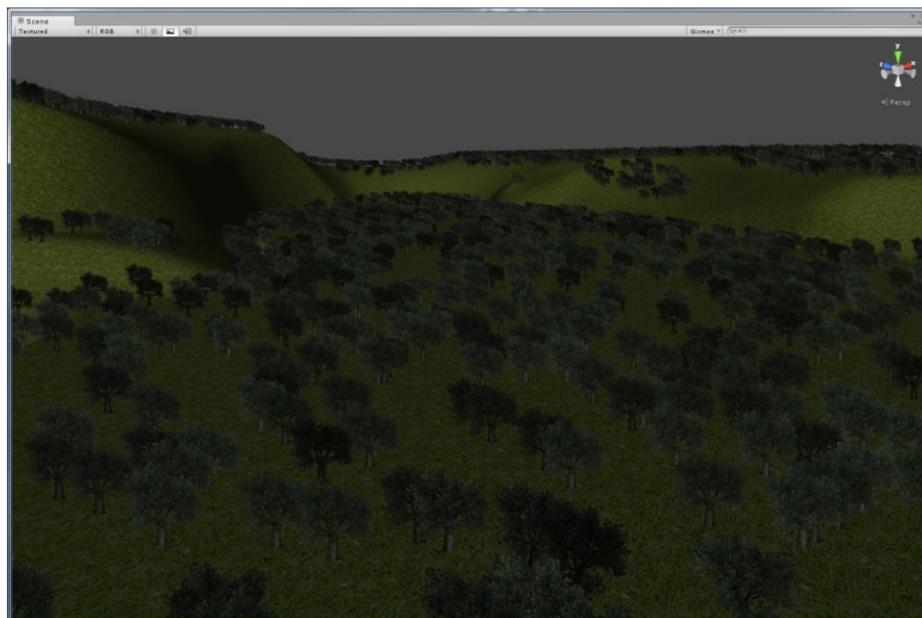


Figura 3.14 - Cenário repleto de árvores.

### 3.2.6. Grama

A ferramenta *Paint Foliage* permite adicionar gramas, pedras ou outras decorações ao longo do cenário (Figura 3.15). Para tal, é preciso selecionar a grama desejada e pintar no terreno, seguindo os mesmos procedimentos das texturas e das árvores.

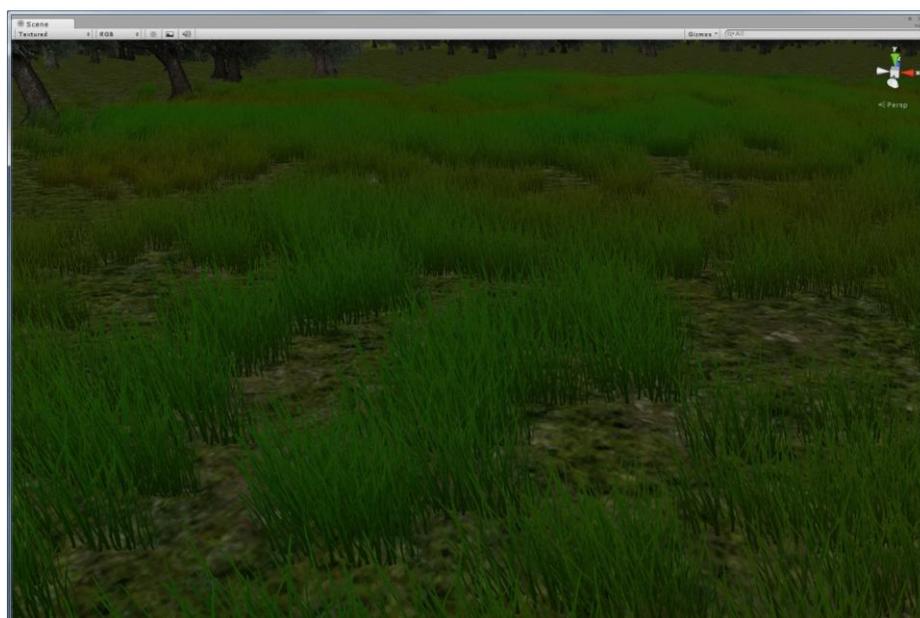


Figura 3.15 - Terreno com grama.

### 3.3. Luzes

Luzes são partes essenciais de todas as cenas. Enquanto a malha e a textura definem o formato e aparência, as luzes definem a cor e o clima do ambiente. Geralmente para obter uma boa iluminação são utilizadas mais de uma luz em cada cena, essa interação entre as luzes pode trazer resultados surpreendentes. No *Unity* existem quatro tipos de luzes: *Directional Light*, *Point Light*, *Spot Light* e *Área Light*.

1. *Directional Light*: usada principalmente em ambientes externos, simulando o sol ou a luz da lua. Essa luz afeta toda a superfície dos objetos em cena.
2. *Point Light*: brilha de um ponto para todas as direções igualmente, sendo normalmente usada em explosões, lâmpada, etc.
3. *Spot Light*: ilumina somente o que está na frente, brilhando apenas em uma única direção. São perfeitas para faróis de carro, lanternas ou postes de luz.
4. *Area Light*: emite luz de um lado de plano retangular, iluminando todas as direções daquele lado.

### 3.4. Assets: Importação e Criação

Uma das principais vantagens do *Unity* está relacionada com a utilização e manuseio de *assets*, tais como texturas, modelos 3D, sons e scripts. Utilizando o *Project View* é possível ter acesso rápido a todos os arquivos.

O *Unity* possui uma forma muito simples e eficiente de importação, bastando apenas mover o arquivo para dentro da pasta “*Assets*” do projeto. Ao efetuar este procedimento, a importação é realizada automaticamente, ficando imediatamente disponível para ser utilizada pelo usuário. Além disso, permitindo a visualização em tempo real de qualquer alteração feitas nos mesmos.

### 3.5. Asset Store

A *Asset Store*, também conhecida como Loja de Recursos, é uma biblioteca crescente de *assets* gratuitos e comerciais, criados tanto pela equipe do *Unity* quanto por membros da comunidade (Goldstone, 2011). Uma ampla variedade de *assets* são disponibilizados, contendo desde texturas, modelos e animações até exemplos de projetos completos e tutoriais. Esses recursos são acessados de uma interface simples presente no editor e são baixados e importados diretamente para seu projeto.

### 3.6. Prefabs

Um *Prefab* é um tipo de *asset*, reutilizável que fica guardado no *Project View*. Quando um *Prefab* é adicionado ao cenário é criada uma instância para ele, fazendo assim com que todos os objetos instanciados sejam ligados ao original, sendo basicamente seus clones. Não importa quantas instâncias existam no projeto, quando uma mudança for realizada no *Prefab* também é aplicada em todas as suas instâncias.

Para criar um *Prefab* basta arrastar o objeto criado na cena para o *Project View*, transformando seu nome em azul para mostrar que agora é um *Prefab*. Depois disso, o elemento e seus dependentes são copiados, podendo ser reutilizado a qualquer momento pelo usuário.

#### 3.6.1. Herança

Herança significa que sempre que houver uma mudança no *Prefab* original, estas mudanças são aplicadas em todos os *GameObjects* ligados a ele. Entretanto é possível mudar propriedades de uma única instância mantendo as outras intactas. Ao alterar qualquer propriedade da instância, o nome da variável ficará em negrito e esta será substituída. Qualquer variável que sofreu esta alteração não é afetada por mudanças do *Prefab* original.

### 3.7. Scripts

A programação é um ingrediente essencial em todos os projetos no *Unity*. Mesmo os mais simples utilizam scripts para responder a entrada do usuário e reproduzir eventos durante sua execução. Além disso, os scripts podem ser usados para controlar o comportamento físico dos objetos ou até mesmo implementar um sistema de inteligência artificial (IA).

No *Unity*, os scripts podem ser implementados em qualquer uma de três linguagens (*JavaScript*, *C#* ou *Boo*), sendo possível usar mais de uma delas em um mesmo projeto. Esses scripts são acoplados como componentes de objetos, ganhando assim flexibilidade na sua reutilização.

#### 3.7.1. Introdução ao C#

O *C#* (pronuncia-se "C sharp") é uma linguagem de programação orientada a objetos e que faz parte do .NET Framework (lê-se dot NET Framework) da Microsoft responsável por fornecer os serviços necessários, para construção e carregamento de aplicações para *Windows* e *Web*. As várias inovações no *C#* permitem o desenvolvimento rápido de aplicações, mantendo a expressividade e a elegância das linguagens C-style (Lotar, 2007).

A linguagem *C#* foi influenciada por várias linguagens, como por exemplo, *JAVA* e *C++*. Isto significa que elementos como declaração de variáveis, métodos e estruturas de controle (*if*, *loops*) são muito semelhantes ao *C++* (Sharp, 2011). Ela tem como objetivo ser uma junção das principais vantagens dentre essas linguagens, melhorando suas implementações e adicionando novos recursos.

Possui sintaxe simples, evitando características complexas encontradas no *C++*, e em outras linguagens semelhantes. Fornecendo recursos poderosos, como tipos de valor nulo, enumerações, delegações, expressões lambdas e acesso direto a memória, suporte a métodos e tipos genéricos, que possibilitam ao programador um desenvolvimento com segurança de tipo e desempenho.

Por ser uma linguagem orientada a objetos, ela possui todas as capacidades e características da orientação a objetos, como a herança, o polimorfismo, o encapsulamento, entre outras. Sendo uma das suas principais características à

tipagem forte, ou seja, o tratamento dos dados é realizado com a definição explícita de seus tipos. Além disso, possuindo suporte a *DLL's*, *COM* e *COM+* e *case-sensitive*.

Esta linguagem permite aos seus desenvolvedores criarem uma infinidade de aplicativos compatíveis com o .NET framework, como por exemplo aplicações tradicionais do *Windows*, *web services*, componentes distribuídos, aplicativos cliente-servidor e aplicativos com integração de banco de dados, entre outros tipos.

Os recursos citados fazem do C# uma linguagem simples de aprender e de usar, robusta e com um bom desempenho (Oberg, 2001). Em conjunto com os demais recursos da arquitetura .NET, o C# é a linguagem ideal para a criação de uma nova categoria de programas que aproveitam as oportunidades trazidas pela Internet.

### 3.7.2. Criação de Scripts

O script pode ser criado acessando a barra de menu: *Assets* → *Create* → *C# Script*, figura 3.16. Nesse campo também é possível selecionar qualquer uma das linguagens disponíveis.

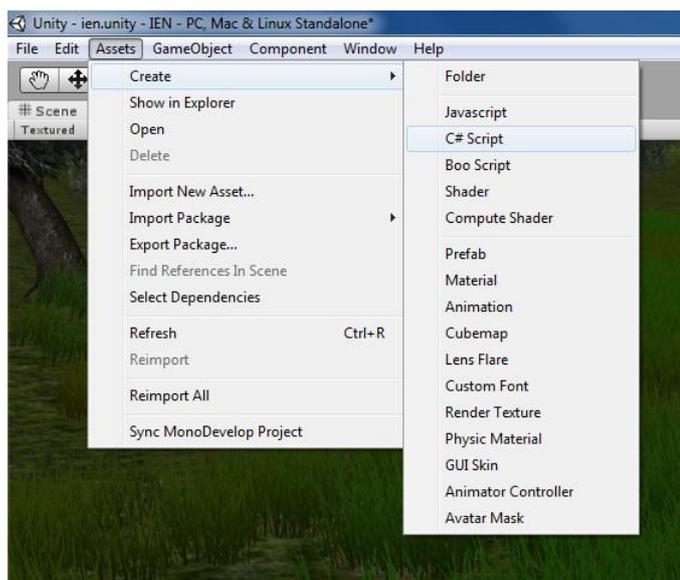


Figura 3.16 - Criação de um *script*.

Dentro da pasta “Assets” é criado um arquivo .cs, que precisa ser nomeado. Ao clicar duas vezes nesse arquivo é aberto o programa de edição padrão *MonoDevelop*, incluso na instalação do *Unity*. A edição desses arquivos de programação é feita em um programa externo e não diretamente no editor. Na listagem 1 é exibido o código padrão ao ser criado uma classe.

Listagem 1 - Código em C#.

```
using UnityEngine;
using System.Collections;

public class NovaClasse : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Existem algumas funções básicas de programação no *Unity*, como as funções: *Start*, *Update* e *OnCollisionEnter*. A função de atualização (*Update*) é a responsável por lidar com atualização de quadros do objeto, sendo carregada uma vez a cada frame. Isso pode incluir movimento, desencadeando ações e responder a entrada do usuário, basicamente qualquer coisa que precisa ser tratada ao longo do tempo durante a execução. Já a função de início (*Start*) é chamada antes de qualquer outro método, executando assim que o programa for iniciado. Por fim, a função de colisão (*OnCollisionEnter*) é acionada sempre que houver contato entres dois objetos com colisão adicionada.

### 3.7.3. Exemplos de Scripts

A listagem 2 mostra um script de movimentação básica, que permite movimentar qualquer objeto que possua esse script utilizando os botões direcionais

do teclado. Após aplicar o valor da velocidade é inserido um “*Time.deltaTime*”, isso faz com que o frame rate (quadros pro segundo) da aplicação seja independente, evitando assim variações de velocidade de um computador para o outro.

Listagem 2 - Script de movimentação.

```
using UnityEngine;
using System.Collections;

public class Movimentacao : MonoBehaviour {

    // Update is called once per frame
    void Update () {

        transform.Translate(transform.forward * 15 * Time.deltaTime *
Input.GetAxis("Vertical"));
        transform.Translate(transform.right * 15 * Time.deltaTime *
Input.GetAxis("Horizontal"));

    }
}
```

O script de autodestruição, listagem 3, elimina os *GameObjects* depois de um determinado tempo, evitando que estes se acumulem na cena. Para tal, este deve ser ligado ao elemento, que de acordo com esse código é destruído após 5 segundos.

Listagem 3 - Script de autodestruição.

```
using UnityEngine;
using System.Collections;

public class AutoDestruicao : MonoBehaviour {

    // Use this for initialization
    void Start () {

        GameObject.Destroy(gameObject,5);

    }
}
```

O *Unity* não tem, por padrão, uma tecla para fechar o executável do projeto, sendo necessário o desenvolvimento de um script para sair do programa. O script, listagem 4, fecha a aplicação ao pressionar a tecla *Escape* (Esc) do teclado.

Listagem 4 - Script de saída do programa.

```
using UnityEngine;
using System.Collections;

public class Sair : MonoBehaviour {

    // Update is called once per frame
    void Update () {

        if (Input.GetKeyDown(KeyCode.Escape))
            Application.Quit();

    }
}
```

No script de sonorização é emitido um som ao acertar um elemento e este é destruído (Listagem 5). Para reproduzir o áudio, o objeto que contém o script deve possuir um arquivo de som dentro dele. Com este código, quanto a velocidade da colisão for maior que seis, o som é acionado e o objeto desaparece do cenário.

Listagem 5 - Script de sonorização.

```
using UnityEngine;
using System.Collections;

public class Som : MonoBehaviour {

    void OnCollisionEnter(Collision collision) {

        if(collision.relativeVelocity.magnitude>6)
        {
            audio.Play();
            GameObject.Destroy(gameObject);
        }

    }

}
```

### 3.8. UniStorm

Desenvolvido pela *Black Horizon Studios*, a *UniStorm* é um *plugin* para aplicação de sistema meteorológico em jogos desenvolvidos no núcleo de jogo *Unity 3D*. Este sistema possibilita a inserção de condições climáticas e relações temporais nos jogos desenvolvidos, de forma realista e pouco custosa ao desempenho da aplicação.

Disponível nas linguagens de programação *C#* e *JavaScript*, disponibiliza diversos tipos de clima, como, por exemplo, céu parcial ou totalmente nublado, tempo com neblina, tempo com ventania ou nevando, queda de folhas, clima chuvoso, relâmpagos, tempestades com chuva forte, etc, todos gerados aleatoriamente, para fornecer maior realismo às cenas (*Black Horizon Studios*, 2013). A *UniStorm* fornece também um controle de noite e dia de 24 horas, permitindo um sistema de tempo realista ao jogo, com aparição de sol, lua e estrelas, bem como um sistema de fases da lua, nascer e pôr do sol, com efeitos de movimentação de nuvens, etc. Este *plugin* possui um editor personalizado e de fácil utilização que permite o controle de todos os componentes disponíveis.

## CAPÍTULO

### 4. MÉTODOS

O método proposto nessa dissertação se divide em duas etapas. A primeira delas, denominada “Modelagem do Ambiente”, consiste na construção do ambiente virtual propriamente dito. Para este fim é utilizado o *Unity* para criação do terreno e o *software Autodesk 3ds Max* para criação de objetos, como por exemplo, as edificações. A segunda etapa, denominada “Implementação de Funcionalidades”, consiste na inserção de funcionalidades ao núcleo do *Unity*, como por exemplo, a criação de câmeras virtuais, de modo a permitir que o mesmo possa ser utilizado para o fim proposto. Estas funcionalidades são incorporadas por meio de configuração de parâmetros e introdução de novos códigos utilizando a linguagem *C#*.

#### 4.1. Modelagem do Ambiente

##### 4.1.1. Modelagem das Edificações

A modelagem das edificações foi feita pelo *software Autodesk 3ds Max*. Uma imagem topográfica foi adicionada e serviu como referência da área modelada (Figura 4.1). A altura pôde ser estimada posicionando-se uma pessoa de um metro e setenta ao lado de cada prédio e então foi feito o registro fotográfico. Dessa maneira, chegava-se a altura que era transportada para o *3ds Max*.

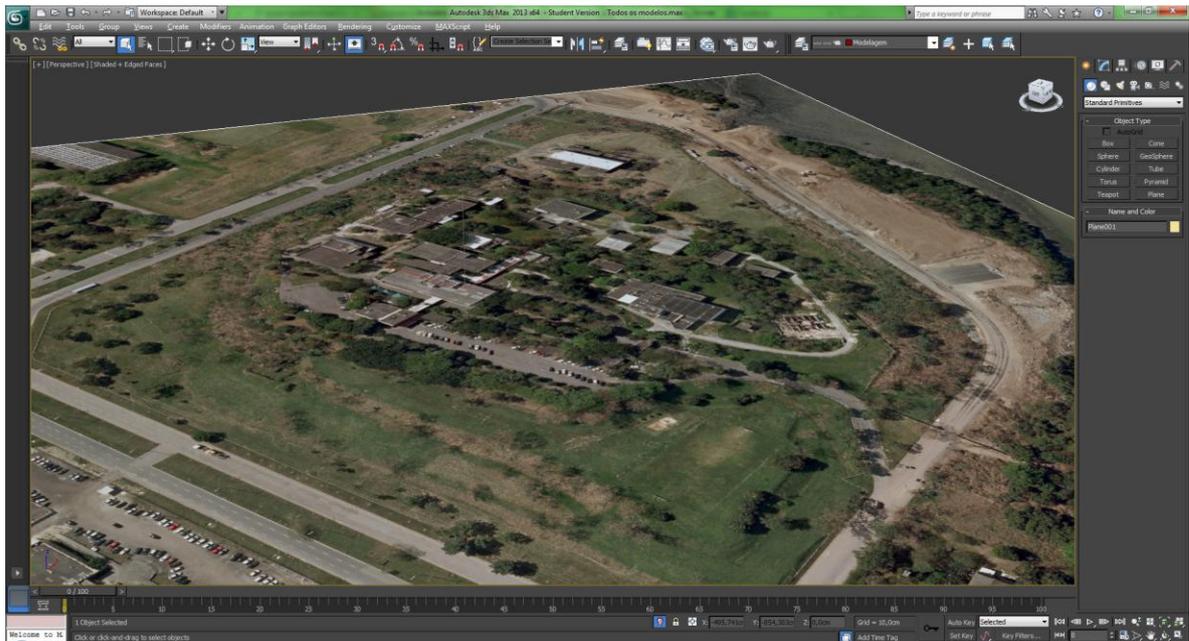


Figura 4.1 - Imagem topográfica adicionada no 3ds Max.

Apenas as fachadas das edificações foram modeladas. A técnica utilizada foi o *Poly Modelling*, que consistiu em adicionar uma primitiva *Box* ao espaço virtual e em seguida, através das ferramentas oferecidas pelo *software*, foi esculpida a forma até o estágio em que o modelo virtual transmitisse a credibilidade de ser uma representação da realidade (Figura 4.2).

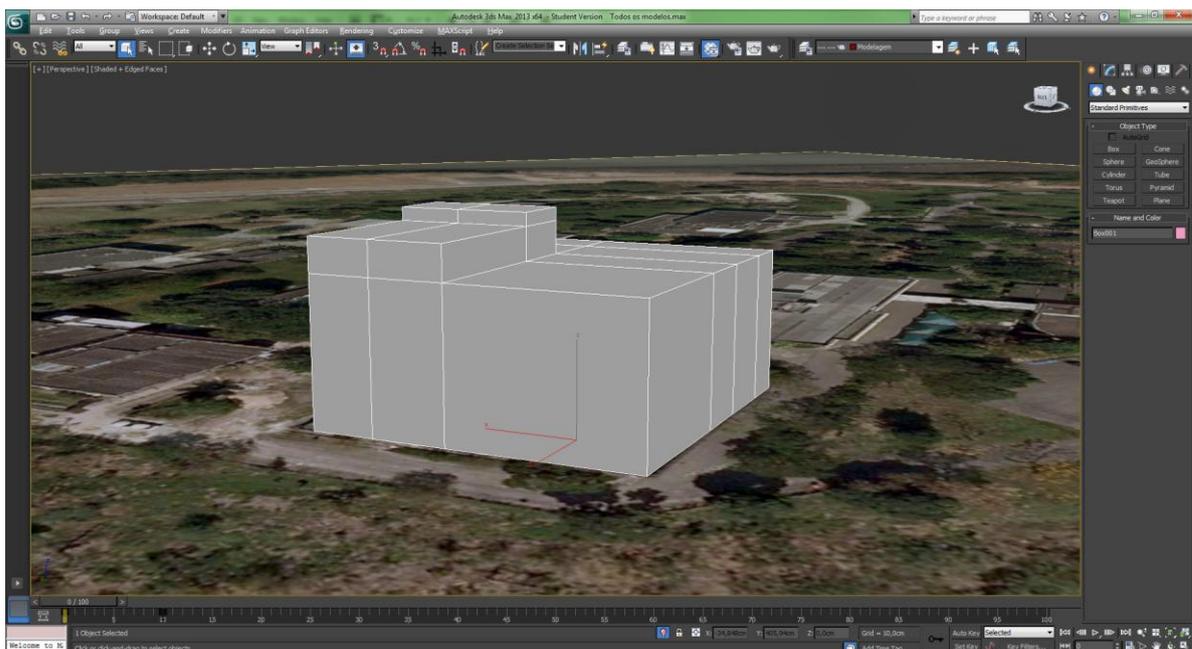


Figura 4.2 - Modelagem das edificações.

Houve também a necessidade de construir virtualmente o interior das portarias devido ao alto fluxo de pessoas durante todo o dia. Neste caso, para a modelagem das áreas internas, foram feitas fotos e visitas ao local. Objetos genéricos como cadeiras, mesas de computador, ar condicionado, pia e vaso sanitário, foram adicionados ao ambiente através de bancos de objetos disponibilizados em sites especializados. Objetos específicos como armários e janelas, também foram modelados através da técnica de *Poly Modeling*.

A segunda etapa consistiu no processo de texturização. Ainda no *3ds Max*, o objeto teve sua malha planificada e posteriormente transformada em uma figura. Foram aplicadas as camadas de textura, as quais foram baseadas em fotos colhidas no próprio local. Nas texturas foram concedidos os aspectos do estado de conservação e efeitos naturais causados pela umidade, poeira e etc.

A figura 4.3 mostra a textura planificada aplicada ao modelo, e cada uma das faces texturizadas coincidentes com a face do modelo que está posicionado no espaço virtual.

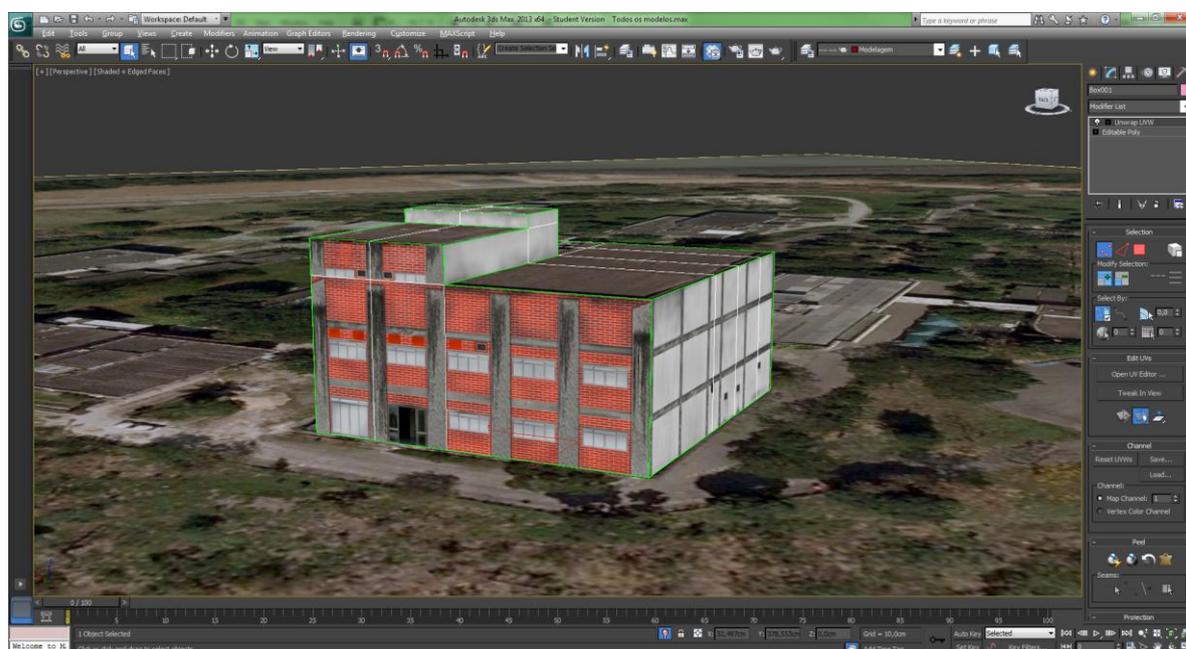


Figura 4.3 - Edificações texturizadas no 3ds Max.

#### 4.1.2. Modelagem do Terreno

O núcleo de jogo *Unity 3D* foi utilizado para realizar a modelagem do terreno da instalação nuclear estudada neste trabalho. Em um primeiro momento, criou-se um terreno padrão da ferramenta, sem formas, relevos ou texturas, destinado a servir de base para as construções e alterações posteriores (Figura 4.4).

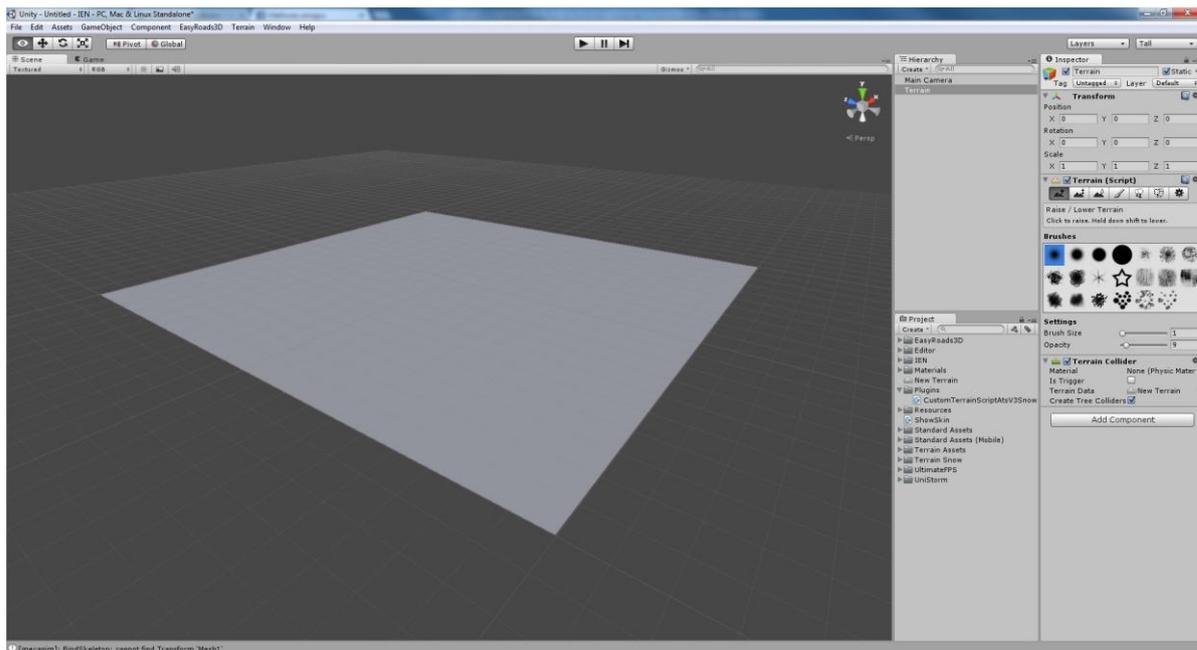


Figura 4.4 - Terreno plano criado no Unity.

Utilizando as medidas e proporções oriundas da imagem topográfica da instalação nuclear, no que concernem as extensões e escalas do cenário, dimensionou-se o terreno padrão do *Unity 3D* (Figura 4.5). Assim, as formas, grandezas de largura e comprimento do terreno em desenvolvimento foram adequadas às reais encontradas no sítio nuclear.

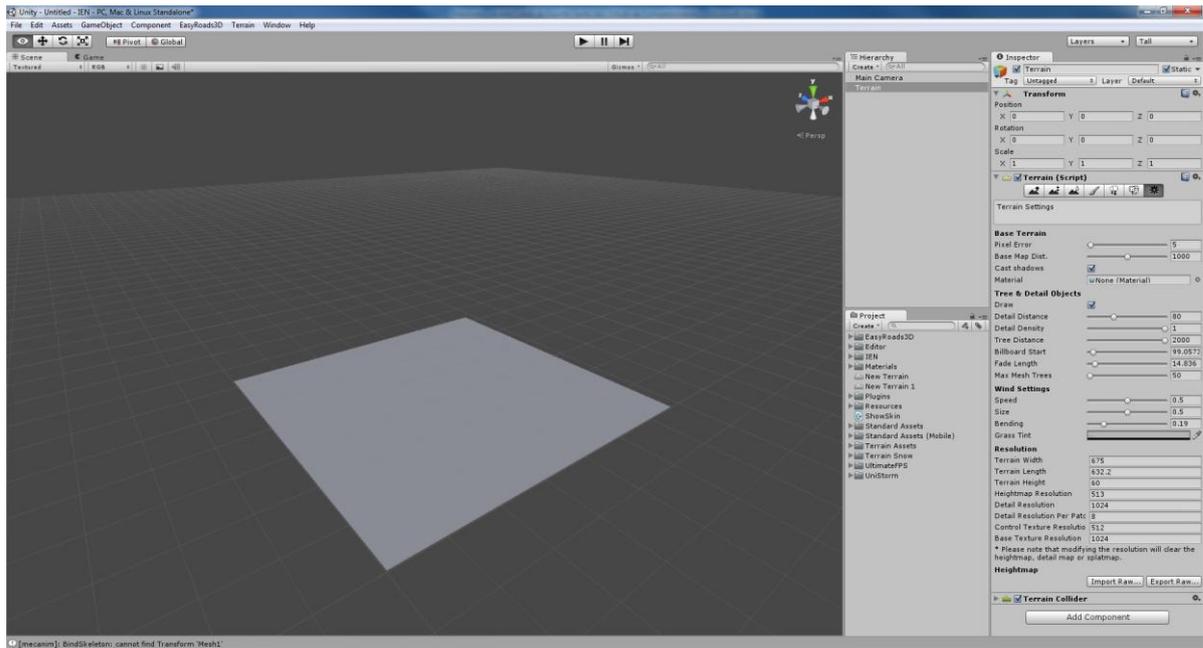


Figura 4.5 - Alteração das dimensões do terreno.

Após ajustar o terreno virtual em relação às formas e escalas do ambiente real, aplicou-se a mesma imagem topográfica utilizada para referência como textura deste terreno, apresentada na figura 4.6. Com as proporções já equivalentes, esta textura, ainda planificada, ajustou-se corretamente ao ambiente modelado, possibilitando a visualização dos pontos necessários à inserção de relevos e dos exatos locais onde as edificações deveriam ser posicionadas.

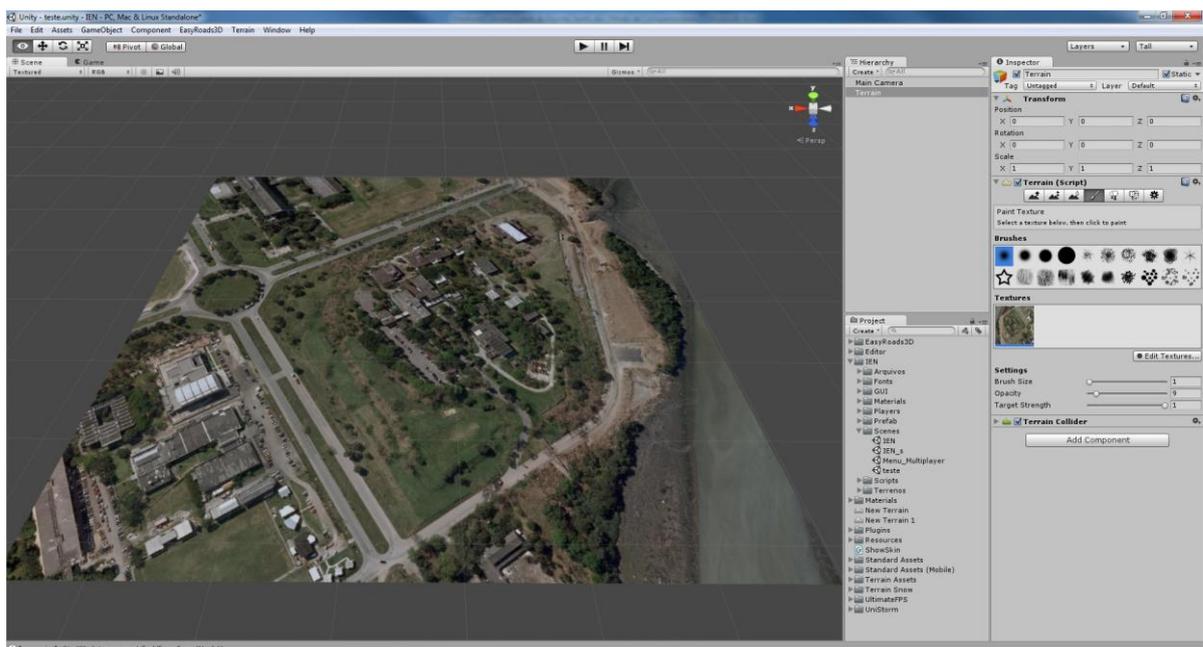


Figura 4.6 - Terreno texturizado.

A partir do terreno virtual modelado (em suas formas e dimensões de largura e comprimento) e já texturizado, foram realizadas as devidas alterações pertinentes ao relevo desta área. A figura 4.7 mostra a aplicação das características de acidentes existentes no terreno, sendo estas naturais ou oriundas de construções existentes.

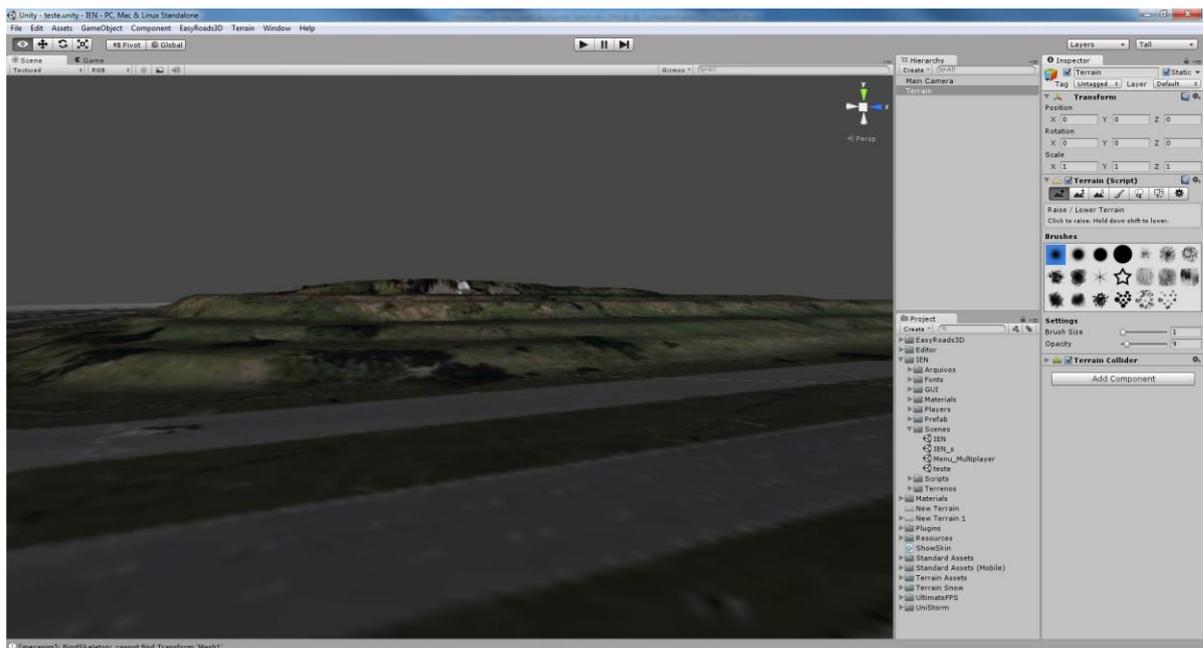


Figura 4.7 - Características geográficas do terreno.

Com o terreno virtual já desenvolvido, iniciaram-se as inserções de objetos e edificações existentes no ambiente real. Os objetos 3D produzidos no *Autodesk 3Ds Max*, conforme já abordado no tópico 4.1.1, foram importados para o terreno do *Unity 3D*. Estes elementos foram posicionados de acordo com a verdadeira disposição das edificações do sítio, e receberam escalonamento dentro do núcleo de jogo, para respeitar as proporções existentes, figura 4.8. No *Unity* uma unidade métrica equivale a 1 (um) metro, sendo necessário um ajuste de parametrização para que os objetos importados conservem esta escala.

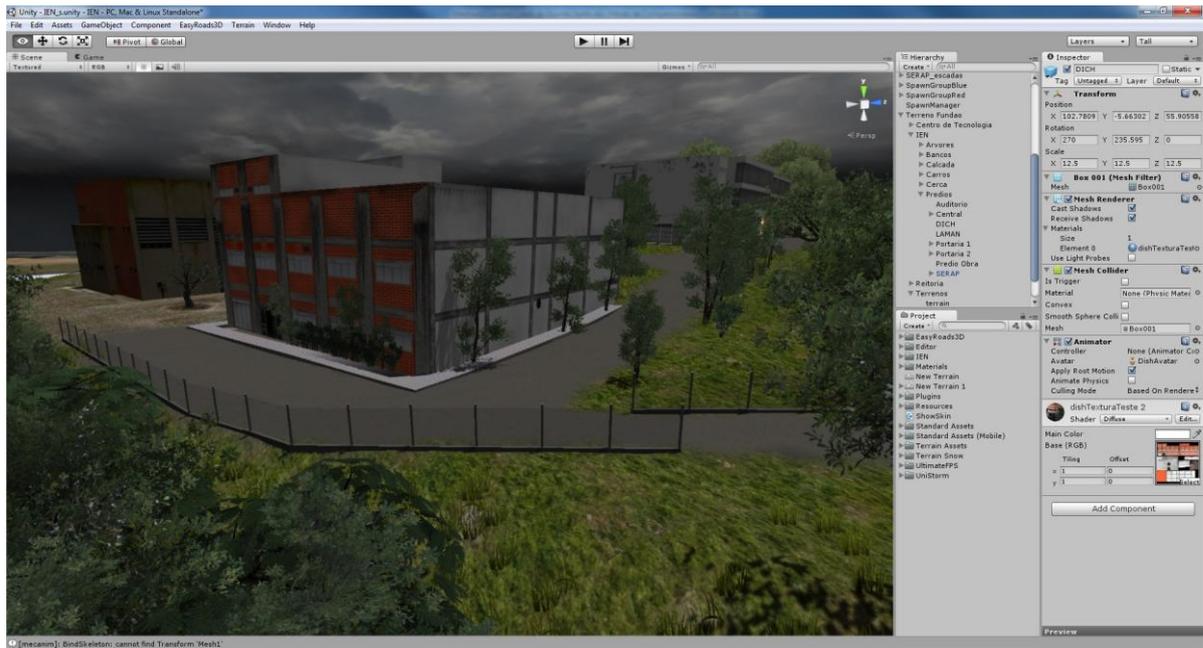


Figura 4.8 - Modelos tridimensionais inseridos no Unity.

Visando fornecer maior realismo ao cenário virtual, inseriu-se também as características de vegetação encontradas na instalação, como gramíneas, arbustos e árvores (Figura 4.9).

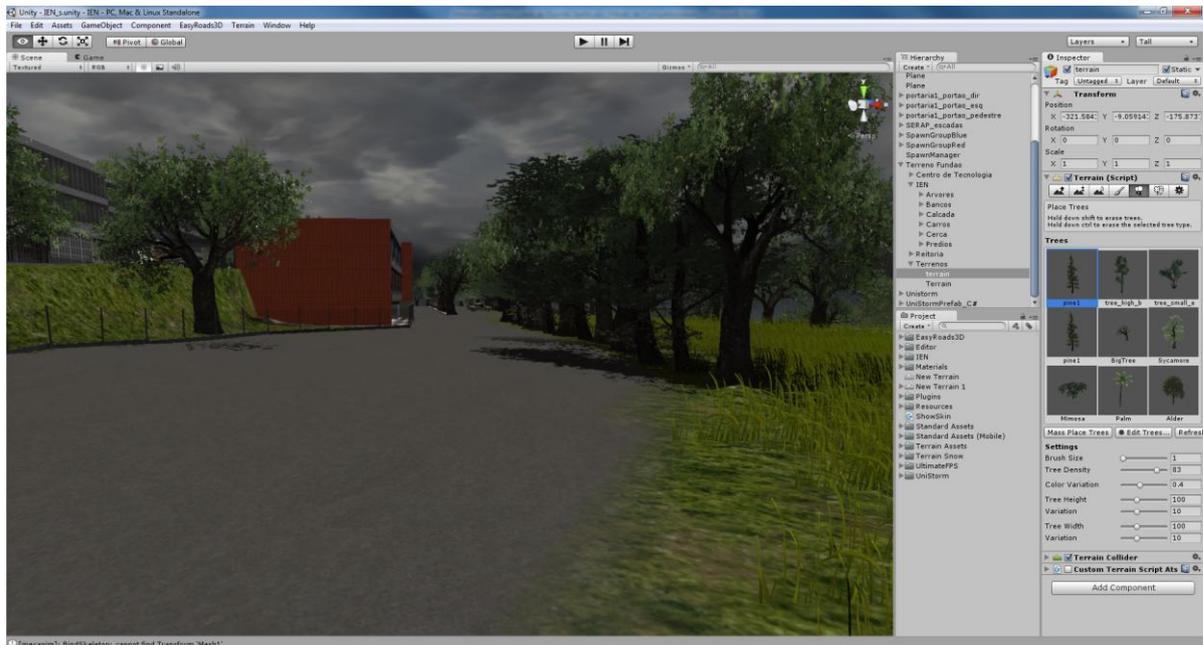


Figura 4.9 - Terreno com sua respectiva vegetação.

Por fim, para possibilitar a interação com o cenário desenvolvido, foram inseridos avatares, personagens virtuais, controlados pelo usuário da aplicação. As figuras 4.10 (a) e (b) mostram os módulos de avatares, tanto em primeira como em terceira pessoa, respectivamente. Um avatar em primeira pessoa é aquele que visualiza o ponto de vista do protagonista, sendo o usuário do sistema e o avatar o mesmo observador, permitindo ao usuário uma percepção de estar inserido no ambiente. Já o avatar em terceira pessoa é aquele que durante a utilização da aplicação pode ser visto pelo usuário, como sendo outra pessoa controlada pelo mesmo.



(a)



(b)

Figura 4.10 (a) - Avatar padrão do *Unity 3D* em primeira pessoa; (b) - Avatar padrão do *Unity 3D* em terceira pessoa.

## 4.2. Implementação de Funcionalidades

### 4.2.1. Características do Avatar

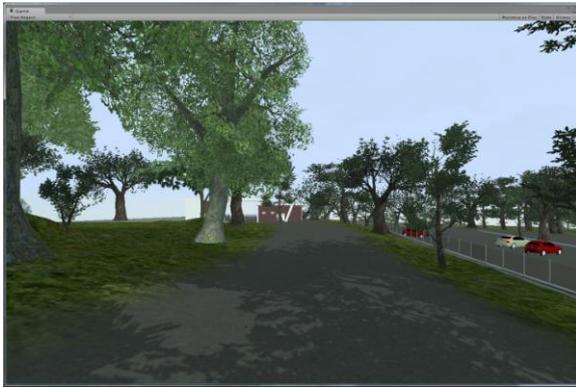
Para fornecer maior realismo ao sistema proposto, realizaram-se alterações nas características dos avatares presentes no ambiente virtual. Primeiramente alterou-se as características visuais destes personagens, evitando uma aparência infantil das animações (como é identificada no avatar padrão do *Unity 3D*), e proporcionando ao usuário uma identificação visual com personagens semelhantes aos que comumente são encontrados na instalação nuclear (Figura 4.11). Além de roupas e expressões alterou-se a altura dos avatares para 1 m e 70 cm, de acordo com a altura média de um homem adulto.



Figura 4.11 - Avatar desenvolvido para a simulação.

Foram ajustadas as velocidades de deslocamento dos personagens, para refletir a verdadeira velocidade de deslocamento de um ser humano dentro da instalação nuclear. Realizou-se medições de velocidades médias de pessoas caminhando e correndo dentro sítio, e tais medidas foram aplicadas aos avatares. Também se aferiu a altura de possíveis saltos que os indivíduos poderiam realizar dentro do ambiente, sendo tais parâmetros aplicados aos personagens virtuais.

Foi também implementada neste trabalho a alternância entre avatares de primeira e terceira pessoa. As figuras 4.12 (a) e (b) mostram a diferença de visão entre tipos de avatares. Desta maneira, o usuário do sistema sempre poderá escolher, de forma dinâmica, durante a utilização da aplicação, qual visualização melhor atenderá suas necessidades.



(a)



(b)

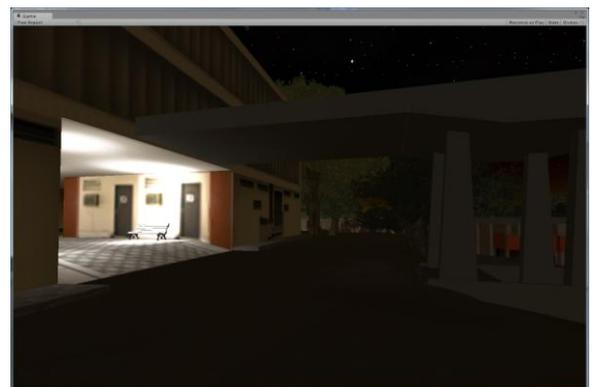
Figura 4.12 (a) - Visão do avatar em primeira pessoa; (b) - Visão do avatar em terceira pessoa.

#### 4.2.2. Sistema de Dia e Noite

Para possibilitar a realização de simulações com diferentes condições climáticas nas mais distintas horas do dia, a partir do *plugin UniStorm*, desenvolveu-se um sistema para controle de dia e noite no ambiente virtual (Figuras 4.13 (a) e (b)). Esta funcionalidade permite a aplicação no cenário de todas as características naturais ao longo das horas, como iluminação, sombras, posicionamento do sol, aparição de estrelas, fases lunares, etc.



(a)



(b)

Figura 4.13 (a) - Ambiente virtual durante o dia, com o posicionamento do sol afetando as sombras dos objetos; (b) - Ambiente virtual à noite, contendo um sistema de fases lunares e estrelas.

O sistema desenvolvido permite também a modificação de características meteorológicas, como por exemplo, dias nublados e ensolarados, chuvas, ventos, neve, etc. A combinação destas diferentes possibilidades permite a realização de simulações nos mais diferentes cenários influenciados por condições de clima e tempo, apresentada na figura 4.14.



Figura 4.14 - Condições climáticas afetando o cenário.

Implementou-se também um menu de controle para gerência das características meteorológicas do ambiente virtual (Figura 4.15). Por meio desta listagem de opções, o usuário pode determinar as condições de clima e tempo em seu cenário proposto, bem como escolher a hora do dia por meio de uma barra de rolagem disponível na aplicação. As características meteorológicas podem também ser escolhidas dentre diversas opções disponíveis, como: nublado, ensolarado, chuva, tempestade, etc.



Figura 4.15 - Menu de controle de tempo e das condições climáticas.

### 4.2.3. Iluminação Artificial

Utilizando fotos e realizando verificações presenciais na instalação em estudo, foram identificados os pontos reais de iluminação, onde as luminárias externas estavam instaladas. A seguir, foi inserido no ambiente virtual desenvolvido um sistema de iluminação artificial. Buscou-se reproduzir no sistema estas mesmas iluminações, compatíveis não só em posição, mas também em intensidade. A figura 4.16 apresenta esta iluminação artificial do cenário virtual.

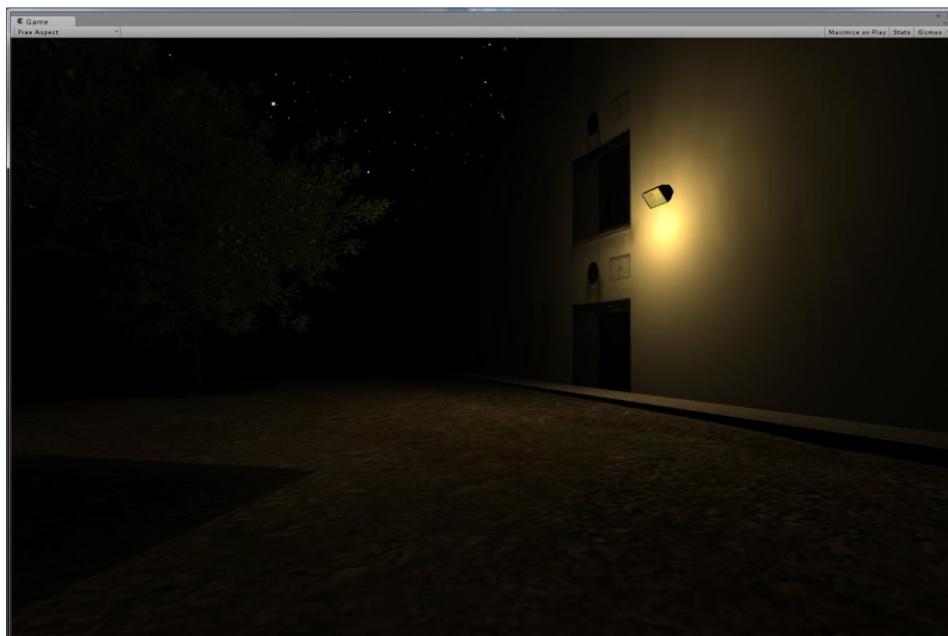


Figura 4.16 - Iluminação artificial do ambiente virtual.

Este sistema de iluminação do cenário virtual foi programado para acender (às 18h00min) a apagar (às 6h00min) automaticamente, de forma semelhante ao funcionamento real. No entanto, disponibiliza-se ao usuário uma opção *blackout*, na qual o mesmo pode optar por apagar todas as luzes da iluminação artificial de uma única vez.

#### 4.2.4. Câmeras

Para possibilitar simulações de segurança física, utilizando monitoramento por câmeras de vídeo, foram implementadas, neste trabalho, câmeras virtuais para monitoramento de pontos críticos da instalação nuclear. Tais câmeras foram posicionadas em pontos estratégicos, onde seria provável a passagem de qualquer possível invasor. A determinação de tais pontos foi baseada na avaliação de figuras topográficas, plantas baixas do sítio nuclear e avaliações presenciais. Nas avaliações presenciais realizou-se observação dos principais pontos de acesso, análise do perímetro e identificação de pontos com maior fluxo de pessoas.

Para possibilitar ao usuário uma avaliação mais completa do monitoramento por meio de câmeras de segurança, criou-se no cenário virtual uma central de

controle, que permite a visualização de todos os pontos monitorados simultaneamente (Figura 4.17). Com a utilização desta central de controle obtêm-se uma visão geral de toda a instalação, permitindo a otimização do posicionamento das equipes de segurança e evitando deslocamentos desnecessários dentro do terreno.



Figura 4.17 - Central de controle dentro do ambiente virtual.

## CAPÍTULO

### 5. TESTES E RESULTADOS

Como estudo de caso, foi escolhido o Instituto de Engenharia Nuclear (IEN), uma unidade pertencente à Comissão de Energia Nuclear (CNEN). O IEN, localizado na Ilha do Fundão, no Rio de Janeiro, Brasil, é o principal órgão estatal responsável pelo gerenciamento de rejeitos do estado do Rio de Janeiro (Figura 5.1).

No presente trabalho optou-se modelar e aplicar os testes aqui propostos nesta instalação nuclear devido a suas características física e a facilitação de acesso ao ambiente e às documentações necessárias. Além disso, o autor desta dissertação está vinculado ao instituto por meio do Laboratório de Realidade Virtual (LABRV), pertencente ao IEN.



Figura 5.1 - Localização do Instituto de Engenharia Nuclear (IEN, 2013).

#### 5.1. Instituto de Engenharia Nuclear

A CNEN é um órgão vinculado ao Ministério da Ciência, Tecnologia e Inovação (MCTI), tendo por objetivo licenciar e fiscalizar instalações nucleares e radioativas (instalações que utilizam radioisótopos, fábricas do ciclo de combustível, reatores, etc). Fundado em 1962, o IEN, pertencente ao CNEN, contribui, desde então, com a área nuclear e áreas afins dentro do território nacional. O IEN

consolidou-se ao longo das últimas décadas como instituição dedicada à pesquisa, desenvolvimento e inovação da área nuclear. Importantes atividades são realizadas pelo IEN, como pesquisa e fornecimento de radiofármacos, ensaios e análises de materiais, armazenamento de rejeitos radioativos, licenciamento de tecnologias, etc.

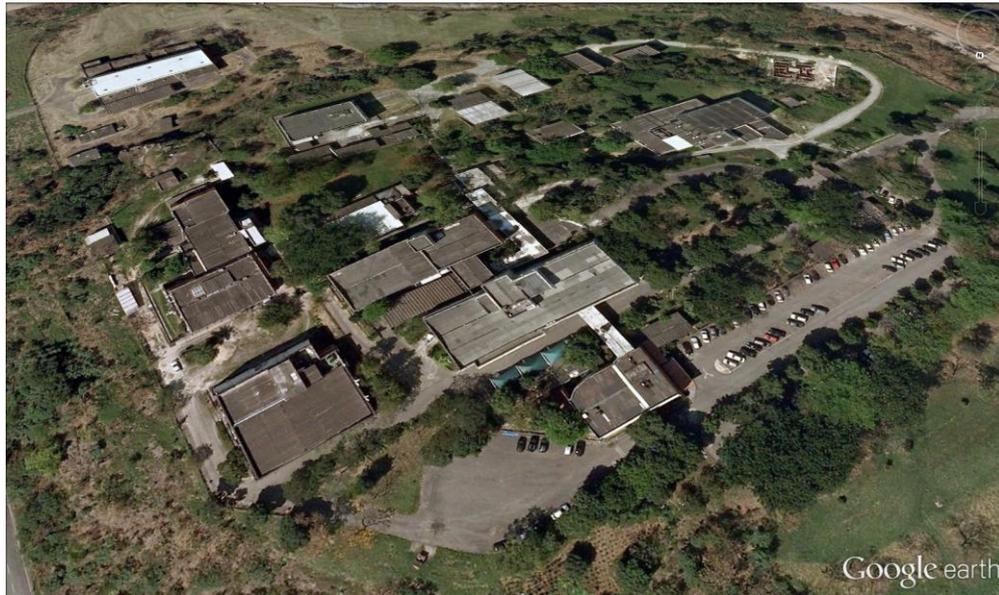
Dentre as principais atividades realizadas pelo IEN, a produção de radiofármacos, a armazenagem de rejeitos e o reator de pesquisa em atividade, foram determinantes na escolha deste sítio nuclear para desenvolvimento destes estudos. As instalações que abrigam tais atividades guiaram a modelagem do ambiente virtual.

## **5.2. Avaliação do Ambiente Virtual: Proporcionalidade das Dimensões Físicas**

O ambiente virtual desenvolvido neste trabalho primou por manter uma semelhança com as instalações do Instituto de Engenharia Nuclear. Pautando suas medidas e proporções em mapas, plantas, fotos e medições manuais das edificações reais do IEN, desenvolveu-se um cenário virtual para realização de todas as simulações aqui propostas.

### **5.2.1. Proporcionalidade da Instalação**

Visando proporcionar ao usuário uma forte identificação visual com as construções existentes neste sítio nuclear, foram mantidas as reais posições destes prédios no mapa do instituto. Deseja-se que um usuário conhecedor das instalações do IEN possa identificar facilmente sua representação virtual na aplicação durante as simulações realizadas, bem como, realizar o deslocamento por calçadas, ruas e portarias sem estranheza de sua localização virtual. Para este fim, posicionou-se não somente os prédios, mas também árvores, canteiros, cercas, bancos e demais objetos que compõem a composição visual do instituto. A figura 5.2 (a) mostra uma imagem topográfica da instalação em estudo, enquanto a figura 5.2 (b) apresenta essa mesma instalação desenvolvida virtualmente.



(a)



(b)

Figura 5.2 (a) - Imagem topográfica do IEN; (b) - Instituto de Engenharia Nuclear virtual.

### 5.2.2. Proporcionalidade dos Prédios e Portarias

A modelagem das edificações existentes respeitaram as medidas e proporções reais deste sítio. Considerando a unidade métrica do *Unity 3D* como relativa à 1 m, manteve-se a proporcionalidade de todos os objetos presentes no

cenário, convertendo-se todos para tal escala. Nas figuras 5.3 (a) e (b) é possível verificar a proporcionalidade do ambiente real com o ambiente virtual modelado neste trabalho.



(a)



(b)

Figura 5.3 (a) - Prédio da instalação; (b) - Prédio correspondente modelado virtualmente.

Dois dos principais pontos estratégicos do IEN foram modelados de forma mais detalhada, externa e internamente. Em tais pontos mostrou-se necessário garantir maior fidelidade de proporções, pois, nestes, o avatar realiza um maior número de interações, sendo obrigado a passar por um deles para ter acesso à instalação. Foram realizadas medições prévias, para embasar a modelagem, e posteriores, para atestar a proporcionalidade dos objetos virtuais.

Um dos principais pontos estratégicos do instituto é a portaria um, também chamada de portaria principal, destinada a recepção dos visitantes. A figura 5.4 (a) mostra uma foto tirada da portaria um e na figura 5.4 (b) é possível visualizar esta mesma portaria no ambiente virtual.



(a)



(b)

Figura 5.4 (a) - Foto da portaria principal do IEN; (b) - Portaria principal na simulação.

A portaria dois, conhecida como portaria secundária, é destinada à entrada de funcionários e colaboradores, sendo considerada assim outro ponto estratégico da instalação. As figuras 5.5 (a) e (b) mostram respectivamente uma foto da portaria secundária e esta desenvolvida no cenário virtual.



(a)



(b)

Figura 5.5 (a) - Foto da portaria secundária do IEN; (b) - Portaria secundária na simulação.

Configurou-se também o avatar do ambiente virtual para manter a escala de um ser humano de estatura mediana (aproximadamente 1 m e 70 cm). Suas interações com os objetos e edificações presentes nas simulações foram também avaliadas por meio de fotos e medições extraídas do IEN, figuras 5.6 (a) e (b).



(a)



(b)

Figura 5.6 (a) - Humano de estrutura mediana usado como referência; (b) - Avatar ajustado para manter a escala de um humano.

### 5.3. Avaliação do Ambiente Virtual Quanto aos Ângulos de Visão

Em uma instalação real, diversos são os ângulos de visão e avaliação do ambiente. Podem-se inspecionar as dependências por meio de rondas, câmeras de segurança e pontos fixos de observação. Estes ângulos podem ainda ser alterados de acordo com diferenças de iluminação e condições meteorológicas que influenciem na visibilidade do ambiente.

#### 5.3.1. Visão de Pontos Estratégicos

Algumas localizações no terreno do Instituto de Engenharia Nuclear podem ser consideradas críticas, pois se tratam de pontos estratégicos para garantia da segurança física da instalação nuclear. Os pontos considerados como prioritários durante o desenvolvimento do presente estudo foram priorizados durante a modelagem virtual do ambiente, dedicando-se a fornecer uma intensa imersão para o usuário do sistema. As figuras 5.7 (a) e (b) a seguir apresentadas fornecem um

comparativo de pontos estratégicos reais em paralelo com a mesma visão por meio do ambiente virtual desenvolvido.



(a)



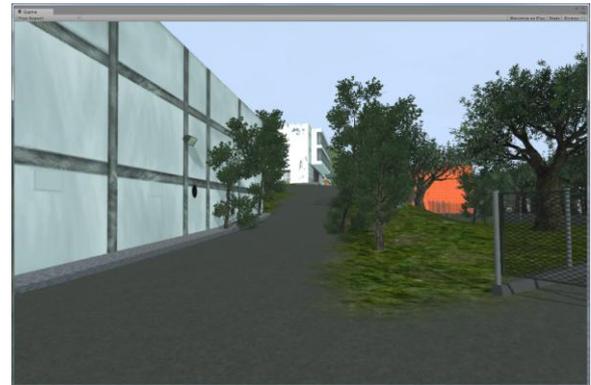
(b)

Figura 5.7 (a) - Ponto considerado relevante da instalação; (b) - Ponto correspondente no ambiente virtual.

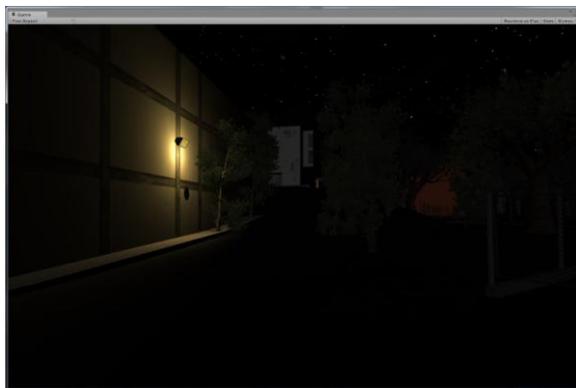
Buscou-se também possibilitar ao usuário a visualização destes pontos estratégicos com a interferência de iluminação natural e condições climáticas adversas, apresentadas nas figuras 5.8 (a), (b), (c) e (d). Para isso, aplicou-se o sistema de dia e noite aqui desenvolvido, baseado na *UniStorm*, para possibilitar a manipulação de tais características. Com esta implementação possibilitou-se a realização de comparativos das visualizações em diversos cenários de testes, bem como ocorre em simulações verdadeiras.



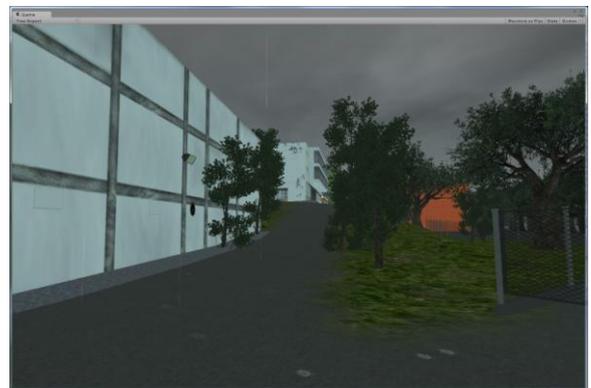
(a)



(b)



(c)



(d)

Figura 5.8 (a) - Foto de um ponto estratégico da instalação; (b) - Vista do ambiente virtual durante o dia; (c) - Vista do ambiente virtual no início da noite; (d) - Vista do ambiente virtual sofrendo alteração climática.

### 5.3.2. Visão de Câmera

Para validar o posicionamento de possíveis câmeras reais no IEN, com objetivo de possibilitar uma comparação com as câmeras virtuais desenvolvidas,

posicionaram-se nos mesmos pontos escolhidos câmeras reais para obtenção de imagens. A partir destas imagens, realizaram-se comparações e validações das imagens capturadas no cenário virtual. Como pode verificar-se nas figuras 5.9 (a) e (b), constata-se grande semelhança entre as imagens do cenário real, base de comparação, e as imagens obtidas no ambiente desenvolvido.



(a)



(b)

Figura 5.9 (a) - Câmera posicionada na instalação; (b) - Visão da câmera virtual.

#### 5.4. Avaliação Quanto aos Tempos de Percurso

Para validar a movimentação dos avatares dentro do ambiente virtual desenvolvido, realizou medições dos tempos de deslocamento dentro do cenário real e dentro do ambiente virtual, nos mesmos percursos. Os caminhos escolhidos para tais procedimentos, apresentados na tabela 5.1, foram alguns daqueles que fazem parte dos pontos críticos do instituto.

Tabela 5.1 - Rotas de deslocamento no instituto.

<b>ROTA</b>	<b>DESCRIÇÃO</b>
Rota 1	Deslocamento da portaria principal até a portaria secundária
Rota 2	Deslocamento da portaria principal até o prédio do reator nuclear presente no IEN
Rota 3	Deslocamento da portaria principal até o prédio do depósito de rejeitos
Rota 4	Deslocamento da portaria principal até o prédio onde está localizado o LABRV
Rota 5	Deslocamento da portaria secundária até o prédio do reator nuclear
Rota 6	Deslocamento da portaria secundária até o prédio do depósito de rejeitos
Rota 7	Deslocamento da portaria secundária até o LABRV

A figura 5.10 mostra a localização dos prédios dentro do Instituto de Engenharia Nuclear, estes foram utilizados como referência para medir calcular o tempo de deslocamento no ambiente virtual.

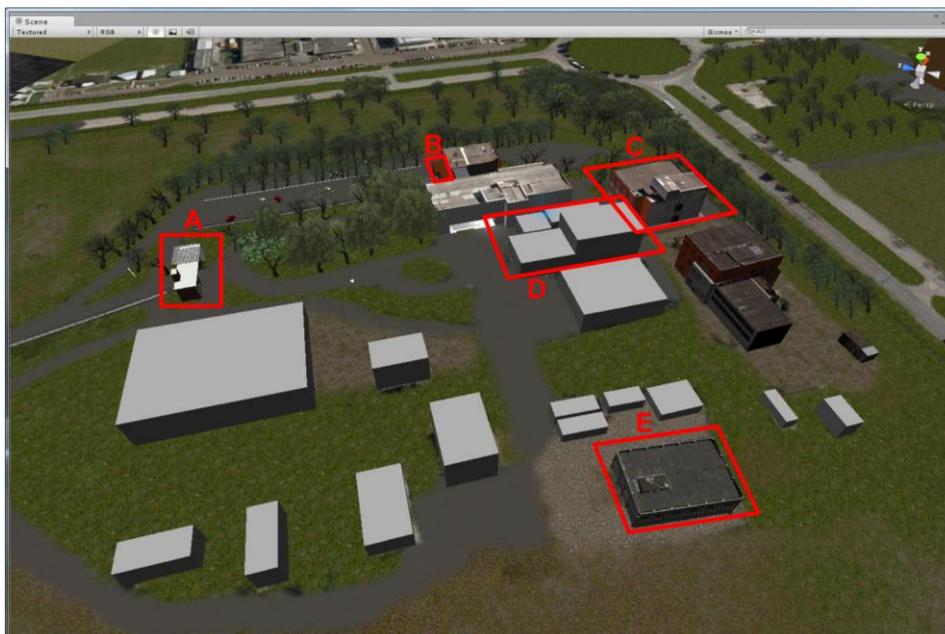


Figura 5.10 – (a) Portaria Principal; (b) Portaria Secundária; (c) LABRV - Laboratório de Realidade Virtual; (d) Reator Nuclear Argonauta; (e) Depósito de Rejeitos Radioativos.

Foi contabilizado o número de passos, necessário para tal deslocamento, quando realizado por um adulto de estatura mediana. Verificou-se o tempo, em segundos, para realização deste trajeto com o usuário caminhando e posteriormente com o mesmo usuário correndo em velocidade média de 12 km/h. As tabelas a seguir apresentadas abordam os resultados obtidos em cada um destes experimentos.

A tabela 5.2 apresenta as medições obtidas referentes à rota 1 (um). Para concluir este trajeto, que liga a portaria principal até a portaria secundária, é necessário percorrer 125 metros, podendo estes ser representados por números de passos, aproximadamente 179 caminhando.

Tabela 5.2 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 1.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	96,3	102,1	92	95	95,7	96,2	<b>Caminhando</b>
	Virtual	91,6	91,1	90,3	88,9	92,7	90,9	5,5
Correndo	Real	39,4	42,8	40,9	38,1	44,3	41,1	<b>Correndo</b>
	Virtual	38,3	39,4	37,7	38,8	38	38,4	6,5

As medições obtidas referentes à rota 2 (dois), que liga a portaria principal até o prédio do reator nuclear presente no IEN, são apresentadas na tabela 5.3. Nesse trajeto é necessário percorrer 136 metros, aproximadamente 191 passos caminhando.

Tabela 5.3 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 2.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	97,1	94,9	104,2	96,7	100,9	98,8	<b>Caminhando</b>
	Virtual	93,2	94,7	91,9	91,8	93,6	93,0	5,8
Correndo	Real	39,1	39,9	43,3	43,9	42,1	41,7	<b>Correndo</b>
	Virtual	40,1	39,7	40,4	41,3	38,9	40,1	3,8

A tabela 5.4 apresenta as medições obtidas referentes à rota 3 (três). Este trajeto liga a portaria principal até o prédio do depósito de rejeitos, sendo necessário percorrer 207 metros ou aproximadamente 288 passos caminhando para concluí-lo.

Tabela 5.4 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 3.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	150,2	165,3	158,3	155,4	162,8	158,4	<b>Caminhando</b>
	Virtual	145,5	148,1	147,2	147,9	145,8	146,9	7,3
Correndo	Real	67,6	69	70,7	67,8	71,1	69,2	<b>Correndo</b>
	Virtual	60,7	61,7	63,4	61,4	61,3	61,7	10,9

A tabela 5.5 apresenta as medições obtidas referentes à rota 4 (quatro). Para concluir este trajeto, que liga a portaria principal até o prédio onde está localizado o Laboratório de Realidade Virtual, é necessário percorrer 230 metros, podendo estes ser representados por números de passos, aproximadamente 312 caminhando.

Tabela 5.5 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 4.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	167,9	171	163,6	167	165,7	167,0	<b>Caminhando</b>
	Virtual	161,2	163	160,8	160,9	162,4	161,7	3,2
Correndo	Real	74,2	76,4	76,1	77	74,9	75,7	<b>Correndo</b>
	Virtual	71,9	70,4	72,7	73,4	73,8	72,4	4,3

As medições obtidas referentes à rota 5 (cinco), que liga a portaria secundária até o prédio do reator nuclear, são apresentadas na tabela 5.6. Nesse trajeto é necessário percorrer 65 metros, aproximadamente 92 passos caminhando.

Tabela 5.6 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 5.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	48,3	50,1	46,3	47,7	50,5	48,6	<b>Caminhando</b>
	Virtual	44,7	44,4	42,1	45,7	46,1	44,6	8,2
Correndo	Real	24,7	22,1	23,4	27,1	25,9	24,6	<b>Correndo</b>
	Virtual	22,3	21,7	24,7	22,7	22	22,7	8,0

A tabela 5.7 apresenta as medições obtidas referentes à rota 6 (seis). Este trajeto liga a portaria secundária até o prédio do depósito de rejeitos, sendo necessário percorrer 160 metros ou aproximadamente 219 passos caminhando para concluí-lo.

Tabela 5.7 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 6.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	115	112,1	113,3	115,3	114,5	114,0	<b>Caminhando</b>
	Virtual	109,1	108,7	109,3	110,8	109,4	109,5	4,0
Correndo	Real	60,1	65,7	63,4	63,1	62,8	63,0	<b>Correndo</b>
	Virtual	60,7	60,4	61,5	62,1	62,3	61,4	2,6

A tabela 5.8 apresenta as medições obtidas referentes à rota 7 (sete). Para concluir este trajeto, que liga a portaria secundária até o prédio onde está localizado o LABRV, é necessário percorrer 130 metros, podendo estes ser representados por números de passos, aproximadamente 180 caminhando.

Tabela 5.8 - Ações, Tempo Real, Tempo das Simulações, Tempo Médio das Simulações e o Erro Relativo para a rota 7.

Ação		Simulação 1 (s)	Simulação 2 (s)	Simulação 3 (s)	Simulação 4 (s)	Simulação 5 (s)	Tempo Médio das Simulações (s)	Erro Relativo (%)
Caminhando	Real	99,8	99,1	96,7	95,1	100,7	98,3	<b>Caminhando</b>
	Virtual	94,2	95,7	92,9	96,7	95,1	94,9	3,4
Correndo	Real	44	41,9	46,4	39,8	41,5	42,7	<b>Correndo</b>
	Virtual	41,9	44,3	42,8	45,6	44,2	43,8	2,4

## CAPÍTULO

### 6. CONCLUSÃO

Esta dissertação propôs verificar a viabilidade do uso de uma ferramenta de construção de jogos para criação de um modelo computacional (ambiente virtual) que sirva como ferramenta de auxílio para estudo e definição de estratégias de segurança física de instalações nucleares.

Neste sentido, o primeiro objetivo deste trabalho foi avaliar o modelo computacional construído quanto ao grau de realismo de sua estrutura física ao ser comparado ao ambiente real. Por último, avaliar o mesmo quanto algumas funcionalidades de segurança ao ser comparado ao ambiente real.

Para esse fim, mensurou-se o grau de realismo do ambiente desenvolvido por meio de comparações de proporções e análises de tempos de deslocamentos. Primeiramente, através de imagens foi realizada a comparação da instalação e seus prédios virtuais (visão de câmera) com o ambiente real (fotos). Observou-se que as proporções do ambiente real e dos objetos nele inseridos em relação aos seres humanos foram equivalentes as proporções dos avatares inseridos no ambiente. Esta aferição pautou-se na comparação de imagens e cenas reais reproduzidas virtualmente, que preservaram a equivalência em suas dimensões.

Utilizando como parâmetro as medições de tempos de deslocamento em trajetos realizados no IEN foram avaliadas as velocidades imprimidas pelos avatares ao deslocar-se no ambiente virtual. Para este fim foi realizado o mesmo trajeto no ambiente virtual e real, ao compará-los constatou-se que os tempos dependidos reais e virtuais foram bem próximos.

Sendo assim, os resultados alcançados na avaliação quanto à qualidade da reprodução permitiram concluir que a ferramenta de RV utilizada para modelar o cenário virtual e interagir com o mesmo foi capaz de criar uma instalação nuclear com bom grau de realismo, indicando boa semelhança entre o modelo virtual e um ambiente real.

Para possibilitar esta avaliação de estratégias em diferentes cenários, no sistema aqui desenvolvido foram inseridos diferentes controles para ajuste de configuração do ambiente virtual, como variação de condições meteorológicas,

alteração de iluminação real e artificial e diferentes ângulos de observação do sistema, utilizando câmeras de vigilância. Desta forma, assim como ocorre em treinamentos e simulações reais, torna-se possível avaliar a reação dos agentes de segurança diante de situações e condições adversas.

Sendo assim, os resultados alcançados na avaliação quanto à relevância da reprodução permitiram concluir que as situações reproduzidas por meio da ferramenta de realidade virtual têm potencial para serem utilizadas para auxiliar no planejamento e definição de estratégias de segurança física de instalações nucleares e, desta forma, otimizar o processo e o tempo de execução de tarefas.

Finalmente, pelos resultados apresentados, o trabalho atingiu seu propósito de servir como ferramenta complementar ao planejamento da segurança física de sítios nucleares, possibilitando a realização de simulações e treinamentos sem interferir no funcionamento da instalação, e sem expor os indivíduos a riscos de contaminação radiológica, diante de diversos possíveis cenários.

## **6.1. Trabalhos Futuros**

Para trabalhos futuros, recomenda-se:

1. A modelagem interna das edificações existentes no Instituto de Engenharia Nuclear, permitindo a realização de simulações que envolvam deslocamento dentro dos ambientes fechados.
2. A modelagem do ambiente ao redor do IEN, como vias de acesso, demais construções existentes na Ilha do Fundão, etc., possibilitando avaliações de segurança no que concernem tentativas de acesso indevido ao terreno do sítio, rotas de fuga de possíveis invasores, etc.
3. Implementação da comunicação entre os avatares, possibilitando a troca de informações e cooperação entre eles.
4. Desenvolvimento de um sistema de avaliação que permita mensurar a qualidade das ações de cada personagem inserido nas simulações, de acordo com objetivos propostos e metas atingidas.

## REFERÊNCIAS BIBLIOGRÁFICAS

AGHINA, M. A. C.; MÓL, A. C. A.; ALMEIDA, A. A. H.; PEREIRA, C. M. N. A.; VARELA, T. B.; CUNHA, G. G. *Full Scope Simulator of a Nuclear Power Plant Control Room Using Virtual Reality 3D Stereo Technics for Operators Training*. International Nuclear Atlantic Conference, 2007.

AUGUSTO, S. C.; MÓL, A. C. A.; JORGE, C. A. F., et al. *Use of virtual reality to estimate radiation dose rates in nuclear plants*. International Atlantic Conference – INAC 2007, Santos, São Paulo, Brasil, Outubro 2007.

AUGUSTO, S. C. *Utilização de Ambientes Virtuais na Estimativa de Dose de Radiação em Instalações Nucleares*. 2008. 81f. Dissertação (Mestrado em Ciências em Engenharia Civil) - Coordenação Dos Programas De Pós-Graduação de Engenharia, Universidade Federal do Rio de Janeiro. Orientadores: Luiz Landau e Antônio Carlos de Abreu Mól. 2008.

AUGUSTO, S. C.; MÓL, A. C. A.; COUTO, P. M.; SALES, D. S. *Using Virtual Reality in the Training of Security Staff and Evaluation of Physical Protection Barriers in Nuclear Facilities*. International Nuclear Atlantic Conference, 2009.

AUTODESK. *Guia Oficial 3ds Max*. 1. ed. Rio de Janeiro: Elsevier - Campus, 2010, 648p.

BLACKMAN, S. *Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development*. Nova Iorque: Apress Academic, 2011, 970p.

BLACK HORIZON STUDIOS. *Developer of UniStorm Weather System*. Disponível em: <<http://blackhorizonstudios.webs.com/>>. Acesso em: 26 junho, 2013

BROOKS, L.F. *National Nuclear Security Administration; Preventing Nuclear Terrorism: Towards an Integrative Approach*; em Nuclear Security: Global Directions for the Future. Proceedings of an International Conference. Londres, 16–18 Março 2005.

BURDEA, G. C.; COIFFET, P. *Virtual Reality Technology*. 2. ed. Nova Jérsei: Wiley-IEEE Press, 2003, 464p.

CALCIOLARI, F. *3ds Max 2012 - Modelagem, Render, Efeitos e Animação*. 1. ed. São Paulo: Erica, 2011, 424p.

CONNORS, E. *Planning and Managing Security for Major Special Events – Guidelines for Law Enforcement*. U.S. Department of Justice, E. U. A., 2007.

FRANCIS, G. A.; TAN, H. S. *Virtual Reality as a training instrument*. The Temasek Journal, v. 7, p.4-15, 1999.

FREITAS, V. G. G.; PEREIRA, C. M. N. A.; MÓL, A. C. A.; JORGE, C. A. F. *Radiation Dose Rate Map Interpolation in Nuclear Plants Using Neural Networks and Virtual Reality Techniques*. Annals of Nuclear Energy, 2010.

GAHAN, A. *3ds Max Modeling For Games*. 2. ed. Waltham: Focal Press, 2011, 432p.

GATTO, L. B. S. *Realidade Virtual na Avaliação Ergonômica de Salas de Controle de Plantas Nucleares*. 2012. 121f. Dissertação (Mestrado em Ciências em Engenharia Nuclear) - Comissão Nacional de Energia Nuclear/Instituto de Engenharia Nuclear, Conselho Nacional de Desenvolvimento Científico e Tecnológico. Orientadores: Antônio Carlos de Abreu Mól e Isaac José Antonio Luquetti dos Santos. 2012.

GOEBBELS, G.; LALIOTI, V.; GÖBEL, M. *Design and evaluation of team work in distributed collaborative virtual environments*. ACM Symposium on Virtual Reality Software and Technology, Osaka, Japão, 2003.

GOLDSTONE, W. *Unity 3.x Game Development Essentials*. 2. ed. Nova Iorque: Packt Publishing, 2011, 488p.

GONÇALVES, J. G. M., et al. *Virtual Reality Based System for Nuclear Safeguards Applications*. IAEA Symposium on International Safeguard, 2010.

HAGUENAUER, C.; CUNHA, G. G.; FILHO, F. C. *Realidade Virtual Aplicada ao Ensino*. 1. ed. Curitiba: EDITORA CRV, 2011, 180p.

HARPER, J. *Mastering Autodesk 3ds Max 2013*. Camp Hill: Sybex, 2012.

INTERNATIONAL ATOMIC ENERGY AGENCY. *Nuclear Security - Measures to Protect Against Nuclear Terrorism*. Annual Report, 2006.

INTERNATIONAL ATOMIC ENERGY AGENCY. *Promoting Nuclear Security: What the IAEA is doing*. Division of Public Information 06-30961/FS Series 1/03 Rev.1/E, 2007.

INTERNATIONAL ATOMIC ENERGY AGENCY. *Nuclear Security Plan - 2010-2013*. 2009.

INSTITUTO DE ENGENHARIA NUCLEAR. *O Instituto*. Apresenta informações sobre a instalação. Disponível em: <<http://www.ien.gov.br/oinstituto/local.php>>. Acesso em: 17 julho, 2013.

JORGE, C. A. F.; AGHINA, M. A. C.; MÓL, A. C. A.; PEREIRA, C. M. N. A. *Man Machine Interface Based on Speech Recognition*. International Nuclear Atlantic Conference; 2007.

JORGE, C. A. F.; MÓL, A. C. A.; COUTO, P. M.; FREITAS, V. G. G. *Virtual Simulation for Training Personnel in Emergency and Security Preparedness and Counteraction*. SIGE, 2011.

KIRNER, C.; SISCOOTTO, R. *Realidade Virtual e Aumentada Conceitos, Projeto e Aplicações*. Mostra de Produtos e Protótipos de RV e RA – IX Symposium on Virtual and Augmented Reality – SVR, 2007.

KYAW, A. S.; PETERS, C.; SWE, T. N. *Unity 4.x Game AI Programming*. Nova Iorque: Packt Publishing, 2013, 232p.

LOTAR, A. *Como Programar com ASP.Net e C#*. São Paulo: Novatec Editora, 2007, 608p.

LOUKA, M. N.; GUSTAVSEN, M. A.; EDVARDSEN, S.; OLSEN, A.; RINDAHL, G. *The Halden Viewer, A Tool for Virtual Walkthrough, Annotation, Radiation Visualisation, and Dose Evaluation (HWR-803)*. Halden, Norway: OECD Halden Reactor Project, 2005.

MÓL, A. C. A.; JORGE, C. A. F.; COUTO, P. M. *Estudo do Uso de Núcleos de Jogos na Criação de Ambientes Virtuais para Suporte ao Planejamento de Evacuação de Prédios e Circulação em Áreas Sujeitas a Radiação*. Mostra de Produtos e Protótipos de RV e RA – IX Symposium on Virtual and Augmented Reality – SVR, 2007.

MÓL, A. C. A.; JORGE, C. A. F.; COUTO, P. M.; AUGUSTO, S. C., CUNHA, G. G.; LANDAU, L. *Virtual Environments Simulation for Dose Assessment in Nuclear Plants*. Progress in Nuclear Energy, 2008a.

MÓL, A. C. A.; JORGE, C. A. F.; AGHINA, M. A. C.; COUTO, P. M.; CUNHA, G. G.; Landau, L.; PEREIRA, C. M. N. A. *Virtual Control Desks for Nuclear Power Plant Simulation: Improving Operator Training*. IEEE Computer Graphics and Applications, v.28, p.6 – 9, 2008b.

MÓL, A. C. A.; JORGE, C. A. F.; COUTO, P. M. *Using a Game Engine for VR Simulations to Support Evacuation Planning*. IEEE Computer Graphics and Applications, v. 28, p.6 – 12, 2008c.

MOONEY, T. *Unreal Development Kit Game Design Cookbook*. Nova Iorque: Packt Publishing, 2012, 544p.

OBBERG, R. J.; *Introduction to C# Using .NET*. 1. ed. Nova Jérsei: Prentice Hall, 2001, 480p.

PERUZZO, J. *Física e Energia Nuclear*. 1. ed. São Paulo: Livraria da Física, 2012, 376p.

SAUNDERS, K.; NOVAK, J. *Game Development Essentials: Game Interface Design*. 2. ed. Stamford: Cengage Learning, 2012, 384p.

SEAN, R. P. T. *CryENGINE 3 Game Development: Beginner's Guide*. Nova Iorque: Packt Publishing, 2012, 354p.

SHARP, J. *Microsoft Visual C# 2010 Passo a Passo*. São Paulo: Bookman, 2011, 776p.

STACKPOLE, B.; OKSENDAHL, E. *Security Strategy: From Requirements to Reality*. 1. ed. Nova Iorque: Auerbach Publications, 2010, 346p.

STANNEY, K. M. *Handbook of virtual environments: design, implementation and applications*. New Jersey, USA: Lawrence Erlbaum Associates, 2002, 1232p.

TRENHOLME, D.; SMITH, S. P. *Computer Game Engines for Developing First-Person Virtual Environments*. Virtual Reality 12 (3), 181 – 187. 2008

U.S. NUCLEAR REGULATORY COMMISSION. *Fact Sheet on Dirty Bombs*. 2002. Disponível em: <<http://www.au.af.mil/au/awc/awcgate/nrc/dirty-bombs.htm>>. Acesso em: 17 julho, 2013.

U.S. NUCLEAR REGULATORY COMMISSION. *Fukushima Daiichi TEPCO Nuclear Power Plant Accident: Comprehensive Coverage of Every Aspect of the Worst Nuclear Crisis Since Chernobyl*. Washington: Progressive Management, 2011. DVD-ROM.

XI, C.; WU, H.; JOHER, A.; KIRSCH, L.; LUO, C. *3-D Virtual Reality for Education, Training and Improved Human Performance in Nuclear Applications*. Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology. Knoxville, Tennessee, 2009.