

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO AUGUSTO GUIDO

**VISUALIZAÇÃO DE INFORMAÇÕES PARA CONSCIÊNCIA
SITUACIONAL NO DOMÍNIO MILITAR USANDO REALIDADE
VIRTUAL**

MARÍLIA

2016

BRUNO AUGUSTO GUIDO

VISUALIZAÇÃO DE INFORMAÇÕES PARA CONSCIÊNCIA
SITUACIONAL NO DOMÍNIO MILITAR USANDO REALIDADE
VIRTUAL

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador:
Prof. Ms. Allan Cesar Moreira de Oliveira

MARÍLIA

2016

Aos amigos pelo incentivo, especialmente ao meu amigo Sertão pela total ajuda e contribuição para a conclusão deste curso.

Aos meus pais e irmão, pelo apoio, compreensão e carinho durante todo este trajeto da minha vida.

A minha namorada Karen Isa, pelo seu amor e sua companhia.

Ao meu orientador Allan Oliveira, pela excelente orientação e paciência ao decorrer deste ano.

“Eu tentei 99 vezes e falhei. Mas na centésima tentativa eu consegui. Nunca desista de seus objetivos mesmo que eles pareçam impossíveis, a próxima tentativa pode ser vitoriosa”

Albert Einstein

RESUMO

Consciência situacional (SAW) é a percepção dos elementos do ambiente dentro de um volume de tempo e espaço, a compreensão de seu significado e a projeção de seu estado no futuro próximo. Sistemas com objetivo de auxiliar a consciência situacional são dependentes da qualidade da informação para que a informação fornecida aos operadores seja a melhor em relação ao cenário atual, auxiliando assim na cognição e tomada de decisão.

Para dar suporte à SAW, faz-se necessário utilizar técnicas que facilitem a visualização de informações (integrando-as em um ambiente virtual), de modo que elas sejam empregadas como forma de apresentar produtos e subprodutos de etapas de processamento da informação militar, etapas estas que são denominadas *situation assessment* (AS).

Desta forma, o presente trabalho descreve a incorporação de técnicas de Realidade Virtual (RV) à um framework que aborda questões como: qualidade da informação e qualidade de representação em avaliação de situações militares.

Um estudo de caso de visualização de informações militares será abordado para ilustrar a aplicabilidade da RV para a representação de informações para otimizar o nível de consciência situacional.

Palavras-Chave: realidade virtual, consciência situacional, tomada de decisão, imersão, visualização de informações.

LISTA DE ABREVIATURAS E SIGLAS

AS: Situation Assessment

ESAS – Emergency Situation Assessment System

IHC – Interação Humano Computador

PMESP – Polícia Militar do Estado de São Paulo

RV : Realidade Virtual

SAW : Situation Assessment Awareness

TI: Tecnologia da Informação

SUMÁRIO

INTRODUÇÃO	9
PROBLEMA	10
OBJETIVO.....	10
OBJETIVOS ESPECÍFICOS.....	11
METODOLOGIA DE TRABALHO	11
CAPÍTULO 1 - INTERFACE HOMEM X MÁQUINA.....	11
1. 1 Introdução	12
1.2 – Surgimento	13
1.3 Caracterização	13
1.4 Conceitos de Realidade Virtual.....	14
1.4.1 Espaço Cibernético.....	14
1.4.2 Estereoscopia.....	14
1.4.3 Graus de Liberdade (DOF).....	14
1.4.4 Presença.....	15
1.5 RV Imersiva e Não Imersiva.....	15
1.6 Tipos de Sistemas com Interfaces Não Convencionais	15
1.6.1 Sistema de Realidade Virtual	16
1.7 Configurações Genérica de Sistemas com Interfaces não Convencionais	17
1.8 – Modelos de Interação do Usuário Associados a Ambientes Virtuais.....	18
1.8.1 - Interação de um único usuário	18
1.8.2 – Interação de Vários Usuários	19
1.9 Formas de Realidade Virtual.....	20
1.9.1 Realidade Virtual de Simulação	20
1.10 Dispositivos para Realidade Virtual.....	21

1.10.1 Dispositivos de Entrada de Dados.....	21
1.8.2.2 Videocapacetes (HMDs)	23
1.8.2.3 Óculus Rift	23
CAPÍTULO 2 - CONSCIÊNCIA SITUACIONAL (SAW)	26
2.1 Percepção	25
2.2 Compreensão.....	25
2.3 Projeção.....	28
2.4 Trabalhos Relacionados	29
2.5 ESAS	30
CAPÍTULO 3 - FERRAMENTAS DE DESENVOLVIMENTO	38
3.1 – Unity 3D.....	36
CAPÍTULO 4 - PROJETO.....	42
4.1 Implementação do Mapa Panorâmico do Google Maps.....	40
4.1.1 – API GoogleMaps.....	40
4.1.2 – Movimentação dos Mapas.....	46
4.1.3 – Buscando Coordenadas em determinados Pontos do Mapa	50
4.1.4 – Buscando Informações das Denúncias através de Pacotes JSON	53
4.1.5 – Instanciando as denúncias no mapa panorâmico.....	57
4.2 – Implementação do StreetView	59
4.2.1 – Instanciando Modelos Tridimensionais na Cena do Crime.....	63
4.3 Realidade Virtual e Gestos de Movimento.....	66
RESULTADOS OBTIDOS.....	67
CONCLUSÃO	68
REFERÊNCIAS BIBLIOGRÁFICAS	69

INTRODUÇÃO

A representação gráfica de informações para a tomada de decisão no contexto militar apresenta desafios, devido à necessidade de fornecer aos especialistas tomadores de decisões subsídios para uma compreensão situacional de um cenário em tempo real (THOMSON et al, 2005). A quantidade e complexidade de elementos associados em ambientes dinâmicos somada ao curto tempo para agir em situações emergenciais é o principal problema enfrentado pelos especialistas que precisam de um bom nível de consciência situacional nestes cenários.

Consciência Situacional (SAW) é um modelo cognitivo que explica o entendimento da situação por um indivíduo para a tomada de decisão em sistemas dinâmicos e complexos.

O nível de consciência situacional (SAW) depende da percepção de um indivíduo de todos os elementos que compõem um determinado cenário em uma situação de emergência, a compreensão de significados e a projeção de um estado de coisas para um futuro próximo, antes de tomar uma decisão.

Para dar suporte à consciência situacional, faz-se necessário utilizar técnicas que facilitem a visualização de informações (integrando-as em um ambiente virtual), de modo que elas sejam empregadas como forma de apresentar produtos e subprodutos de etapas de processamento da informação militar, etapas estas que são denominadas *situation assessment* (AS). A cada conjunto de informações produzido, o especialista observa e se orienta à luz das visualizações, para então tomar uma decisão de acordo com seu nível de SAW.

Freitas distingue tais informações em três tipos de classes: unidimensional, bidimensional e tridimensional, definidas de acordo com a dimensão de um espaço onde tais elementos geométricos utilizados estejam localizados (FREITAS, 2001).

Sistemas que utilizam representação gráfica de informações tridimensionais podem ser construídos por meio de ambientes de realidade virtual com o uso de dispositivos imersivos, associados também a dispositivos de sensores de movimento que permitem a interação do usuário com o sistema. Assim, esta forma de visualização tem como fim facilitar ao especialista tomador de decisões analisar e obter compreensão dos dados, já que os mesmos podem ser visualizados de maneira intuitiva e interativa (ZORZAL et al, 2011).

Em suma, o presente trabalho incorpora técnicas de realidade virtual através de dispositivos estereoscópicos de imersão a um framework de visualização de informações, utilizando uma API do Google Maps e do Street View vinculadas a uma engine de motor de jogos. Para orientar a criação de ambientes imersivos de uma situação de emergência foi

utilizado como base um sistema de avaliação de situações (AS) chamado sistema ESAS, de autoria do grupo de pesquisa de IHC do laboratório COMPSI – UNIVEM – Marília-SP juntamente com o professor Leonardo Botega, e com o grupo de IHC da universidade de São Carlos (UFScar). Este trabalho possui vínculo com a polícia militar do estado de São Paulo (PMESP), assim, foi utilizado este sistema com o propósito de buscar todas as informações necessárias para simular uma situação de roubo para desenvolver o trabalho em questão.

PROBLEMA

Adquirir SAW é um processo desafiador. Estimular a percepção e promover a melhor compreensão dos elementos que compõem um cenário real muitas vezes culmina em um procedimento complexo. Em situações onde há risco à vida ou ao patrimônio, como em eventos de roubo, há necessidade de que o especialista conheça as entidades envolvidas e seus estados, para assim ter subsídios para uma tomada de decisão rápida e eficiente. Estes tipos de situações de emergência ensejam cenários muito complexos e dinâmicos na medida em que novas informações são geradas a todo momento, o que dificulta ao especialista atualizar-se rapidamente a este acúmulo de informações e conseqüentemente prejudicando assim a tomada de decisão rápida e eficaz.

OBJETIVO

O objetivo do trabalho consiste em desenvolver um sistema capaz de oferecer ao especialista fácil acesso à visualização de informações de uma situação, conseqüentemente, gerando otimização no nível de consciência situacional, utilizando a tecnologia de realidade virtual para imergir o especialista em um ambiente virtual referente à situação de emergência, juntamente com modelos tridimensionais com as informações referentes às entidades de uma situação.

Desta forma, os métodos de Realidade Virtual foram acoplados à um framework de visualização de informações, e inseridos como um módulo em um sistema completo de avaliação de situações, como mostra na **figura 1**.

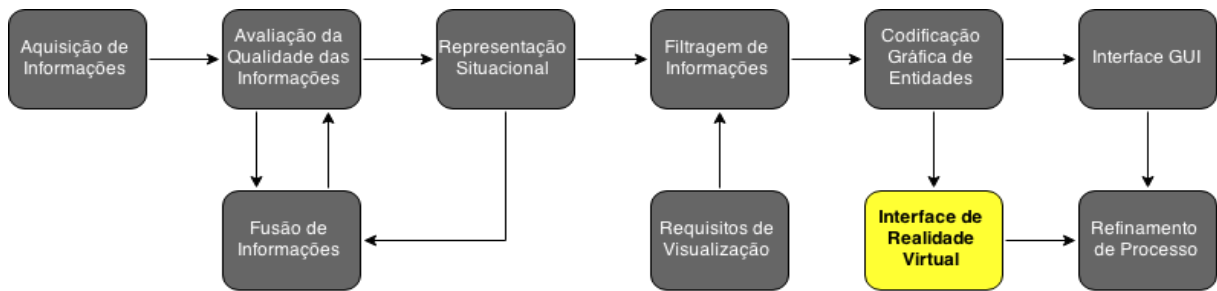


Figura 1 - Fluxo do sistema completo de avaliação de situações e o posicionamento do módulo de interface de RV para melhoria de SAW em destaque

OBJETIVOS ESPECÍFICOS

- Pesquisar e estudar conceitos de Realidade Virtual e Consciência Situacional;
- Realizar levantamento bibliográfico de artigos relacionados à Consciência Situacional no domínio militar;
- Estudar o framework existente para visualização de informações no contexto de SAW;
- Estudar engines gráficas como o UNITY3D e sua integração com o Óculus Rift e o Leap Motion;
- Implementar as técnicas de Realidade Virtual (imersão e interação) para o cenário da situação.

METODOLOGIA DE TRABALHO

O presente projeto visa atingir os objetivos estabelecidos através de três fases complementares:

1. Avaliação e sumarização do estado da arte em realidade virtual e consciência situacional, associado a estudos de trabalhos relacionados para justificar a inovação proposta por este projeto de pesquisa;
2. Desenvolvimento do mapa panorâmico juntamente com o ambiente virtual street view, utilizando como plataforma de desenvolvimento a engine motor de jogos Unity3D.
3. Prototipação das técnicas de realidade virtual identificadas como relevantes e incorporação de uma nova camada ao framework de visualização de informações para a consciência situacional;

CAPÍTULO 1 – Interface Homem x Máquina

1.1 Introdução

A tecnologia de realidade virtual (RV) oferece ao usuário de computador uma interface de realidade sintética, por meio de canais multissensoriais em ambiente 3D que possibilitam navegação imersiva e até mesmo interação em ambiente virtual. Essa tecnologia já vem sendo amplamente aplicada na maioria das áreas de conhecimento, com investimentos massivos das indústrias tanto de hardware quanto de software [Kirner et al, 2010], com perspectivas de crescimento bastante promissoras para diversos segmentos vinculados à área.

Sistemas que utilizam realidade virtual podem fornecer alta qualidade de imersão aos usuários, criando a sensação de estarem dentro de um ambiente virtual, uma interface homem-máquina bastante natural e amigável.

Para tal, as interfaces desses sistemas devem envolver um controle tridimensional altamente interativo de processos computacionais sincronizados com os movimentos naturais tridimensionais do corpo humano. A imersão em espaço virtual se torna mais viva na medida em que não apenas visualizamos, mas também exploramos e manipulamos os dados em tempo real (THOMSON et al, 2005). Requer-se apenas o conhecimento intuitivo do mundo físico para manipular o mundo virtual, com o uso de dispositivos, como capacete de visualização e controle, luvas e outros acessórios: o mesmo gesto de pegar um copo físico permite “pegar” um copo virtual e alterar o estado atual do ambiente, tal como se dá neste projeto.

Projetar um sistema de realidade virtual envolve estudos e recursos ligados à percepção humana, hardware, software, interface de usuário, fatores humanos e aplicações. Assim, domínios relacionados a dispositivos de E/S, computadores de alto desempenho e boa capacidade gráfica, sistemas paralelos e distribuídos, modelagem geométrica tridimensional, simulação em tempo real, navegação, detecção de colisão, avaliação, impacto social, projeto de interfaces e aplicações distribuídas em diversas áreas, são fundamentais para a elaboração de sistemas complexos de realidade virtual (THOMSON et al, 2005).

Ao longo deste capítulo serão apresentados a origem, os objetivos, as características da RV e suas formas, conceitos da realidade virtual, dispositivos não convencionais de E/S, a análise de algumas plataformas computacionais para realidade virtual, e por fim alguns ambientes nos quais a realidade virtual se aplica e suas otimizações.

1.2 – Surgimento

A realidade virtual já existe há mais de 40 anos, porém, só foi difundida após a popularização do computador pessoal, quando foi possível que um grande número de usuários pesquisassem e desenvolvessem programas utilizando esta nova tecnologia [KIRNER, 2006].

A realidade virtual começou a ser utilizada durante a Segunda Guerra Mundial para simulação de voos, preparando pilotos para a guerra, e em 1962 passou a ser utilizada para entretenimento. Um destaque desta época foi o *Sensorama*, uma cabine com imagem 3D com visão estereoscópica, som estéreo, vibrações, aromas e até vento produzido por ventiladores.

Outras criações para a utilização da realidade virtual foram: a caneta ótica, que projeta gráficos na tela, o *Videocapacete* e o *Super Cockpit*, uma cabine que lembra a de um avião real que permite ao piloto ou candidato treinarem a condução com seis graus de liberdade. Ainda foi criada uma luva com sensores, chamada de *Data Glove*, que identificava movimentos e inclinação nos dedos e passava estas informações para o computador. (KIRNER, 2006)

1.3 Caracterização

A realidade virtual possui diversas definições em parte por sua natureza interdisciplinar e sua evolução, com sua origem em sistemas computacionais de mesa, simuladores, sistemas de teleoperação, etc.

Existem três ideias básicas que compõem a RV: imersão, interação e envolvimento (THOMSON et al, 2005).

O conceito de imersão se caracteriza pelo sentimento de que o usuário esteja dentro do ambiente virtual, seja com dispositivos especiais de imersão como os capacetes de visualização, seja em salas de projeções das visões nas paredes, tetos e piso (THOMSON et al, 2005) conhecidos como cavernas ou “caves”. Para que o usuário vivencie maior sensação de imersão, usa-se o posicionamento automático da pessoa e dos movimentos da cabeça,

controles reativos, e etc., somados a estímulos de outros sentidos, como a audição, com sons 3D igualmente projetados (THOMSON et al, 2005).

O conceito de interação, uma das características mais marcantes, é a possibilidade de o usuário modificar instantaneamente o ambiente com base em seus movimentos e reações detectadas pelo computador, através dos dispositivos de entrada de dados, como nos videogames.

Já o conceito de envolvimento está ligado com o grau de motivação do usuário para com aquele sistema ou aquela determinada atividade específica, podendo ela ser tanto passiva (leitura de um livro, assistir televisão) como ativa (participar de um jogo com um amigo).

1.4 Conceitos de Realidade Virtual

Após termos conhecido alguns equipamentos utilizados na tecnologia da realidade virtual, é preciso discutir alguns conceitos relacionados ao tema, a fim de possibilitar a compreensão do funcionamento de algumas tecnologias. Conheceremos agora conceitos importantes para esse fim.

1.4.1 Espaço Cibernético

Esse conceito abrange uma representação gráfica criada com base em dados advindos de todos os computadores existentes. Através da internet, seria criado um sistema de rede integrado. Se os sistemas de RV conseguirem criar o Espaço Cibernético, seria possível interagir com tudo e com todos virtualmente. (MACHADO, 1995)

1.4.2 Estereoscopia

Ciência que estuda o olho humano, partindo-se do fato de que cada olho obtém uma imagem ligeiramente diferente do outro. O cérebro percebe essa diferença e, com isso, torna-se possível ter noção de distância e tamanho de um objeto, efeito este denominado paralaxe (MACHADO, 1995).

1.4.3 Graus de Liberdade (DOF)

Esta definição está relacionada aos movimentos em relação aos eixos X, Y e Z. Normalmente uma pessoa pode deslocar-se para frente ou para trás em relação a estes três eixos, bem como rotacionar em relação a eles. Percebe-se, portanto, que esse conceito está

diretamente ligado à liberdade de movimento do usuário da realidade virtual (MACHADO, 1995).

1.4.4 Presença

Sensação de que a pessoa está no ambiente virtual. Esta sensação é conseguida através de estímulos principalmente à visão, audição e ao tato, sendo o olfato estimulado com menor frequência. A interação com o ambiente aumenta a sensação de presença no decorrer do experimento, porém, não é recomendado uso constante da realidade virtual, pois variando de pessoa para pessoa e de ambiente para ambiente, acometem-lhes sintomas como dor de cabeça, náuseas e tonturas. (MACHADO, 1995)

1.5 RV Imersiva e Não Imersiva

A realidade virtual pode ser tanto imersiva quanto não imersiva. Assim, exploraremos aqui algumas vantagens e desvantagens de cada uma delas.

Como citado anteriormente, a RV imersiva se caracteriza pela utilização de capacetes ou de salas de projeções nas paredes para “inserção” do usuário no ambiente 3D; já a RV não imersiva se vale de interfaces postas diante do usuário, como com o uso de monitores.

Embora a RV imersiva tenha evoluído nos últimos anos, a RV não imersiva apresenta ainda vantagens, como: utilizar todas as características do avanço da tecnologia das placas gráficas, evitar as limitações técnicas e problemas decorrentes do uso constante dos capacetes, e a facilidade de seu uso. Dependendo das necessidades, apenas o uso de monitores para a visualização de informações já é suficiente; porém, com a evolução da tecnologia da realidade virtual a tendência vem sendo o aumento de aplicações que utilizam capacetes ou salas de projeções (THOMSON et al, 2005).

1.6 Tipos de Sistemas com Interfaces Não Convencionais

Para entendermos o conceito e a definição de sistemas que apresentam interfaces de hardware e software que envolvem dispositivos e abordagens não convencionais, é importante verificar o relacionamento usuário/ambiente conforme a **figura 2**.

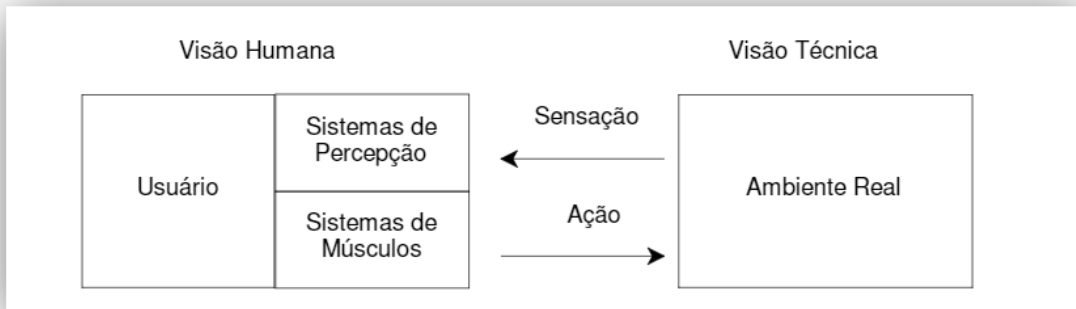


Figura 2 – Relacionamento Usuário/Ambiente

A **figura 2** mostra que o usuário utiliza dispositivos para aumentar o seu nível de imersão em um ambiente virtual, criando a sensação de presença, com uso de dispositivos sensores de movimento que capta gestos do usuário, para que o mesmo realize ações e modifique o estado do ambiente virtual.

Estes tipos de ambientes são compostos pelo espaço físico, funções, processos, equipamentos e conceitos. São responsáveis por responder às ações do usuário com os estímulos que provocam a sensação de imersão. O ambiente no contexto desses modelos compõem-se dos espaços reais e artificiais [KIRNER, 2006].

1.6.1 Sistema de Realidade Virtual

O sistema de RV é composto por um usuário, uma interface homem-máquina e um computador (**figura 3**). O computador gera um mundo virtual no qual o usuário imerge por meio de uma interface homem-máquina juntamente com o auxílio de dispositivos sensoriais de percepção e controle, realizando ações neste ambiente e modificando o estado de coisas. Esse ambiente virtual pode ser projetado para simular dois tipos de ambientes diferentes: ambiente imaginário ou ambiente real.

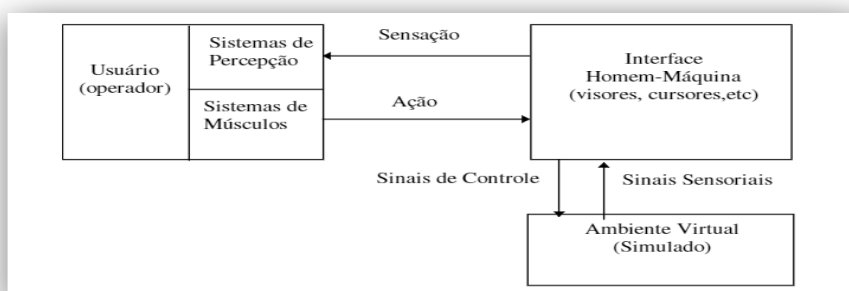


Figura 3 – Sistema de Realidade Virtual

1.7 Configurações Genérica de Sistemas com Interfaces não Convencionais

Os vários tipos de sistemas com interfaces não convencionais podem ser agrupados de uma forma genérica, como mostra a **figura 4**.

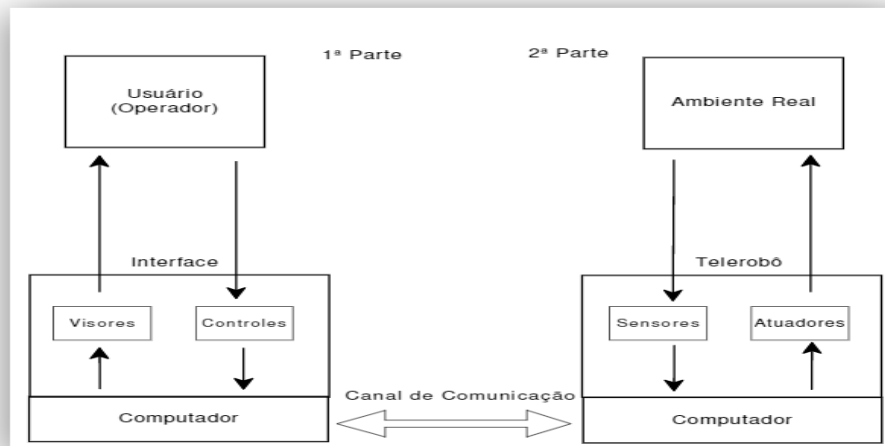


Figura 4 – Configuração Genérica dos Sistemas com Interfaces não Convencionais

Aqui, enquadram-se os sistemas de telepresença, realidade virtual, realidade aumentada e realidade melhorada. Em todos eles, o usuário vale-se de dispositivos eletrônicos de E/S para se projetar em um ambiente virtual e interativo. Tanto a interface homem-máquina quanto as experiências do usuário do mundo real são fundamentais para um melhor nível de desempenho em um sistema baseado em realidade virtual.

O texto que a seguir relaciona os quatro tipos de cenários nos quais a particularização da configuração mostrada na **figura 4** podem ser modeladas:

a-) Caso a primeira parte seja desprezada, a segunda parte poderá se transformar em um robô autônomo, considerando-se que o computador seja utilizado para realizar seu controle.

b-) Caso a segunda parte seja desprezada, porém o computador esteja sendo utilizado para gerar um ambiente virtual, o sistema passa a ser visto como um sistema de realidade virtual.

c-) Caso as duas partes estejam sendo desprezadas e o computador da primeira parte não crie ambientes virtuais, limitando-se a repassar sinais de controle e sensação, o sistema funcionará como um sistema de telepresença.

d-) Caso as duas partes sejam consideradas e o computador da primeira parte gere ambientes virtuais a serem misturados com visões reais, resultará um sistema de realidade aumentada para ambientes virtuais normais, ou um sistema de realidade melhorada para ambientes virtuais complementados com sinais de processamento de imagens e anotações.

1.8 – Modelos de Interação do Usuário Associados a Ambientes Virtuais

Em um cenário composto por um ou mais usuários, é possível que estes se comuniquem de várias maneiras com o mundo real e com o ambiente virtual equivalente (THOMSON et al, 2005). Assim, aparecem dois grupos distintos de interação virtual: um, onde o usuário participa isoladamente, e outro, onde há vários usuários interagindo entre si e com o ambiente (THOMSON et al, 2005). Segue a **figura 5** detalhando esse esquema:

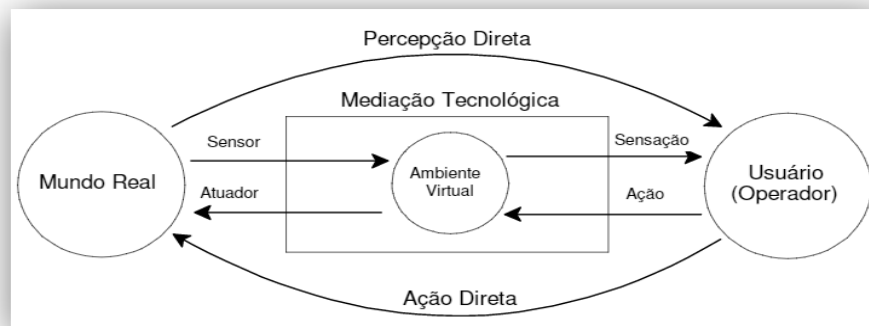


Figura 5 – Esquema de Interação com mediação tecnológica

1.8.1 - Interação de um único usuário

Esse tipo de interação ocorre de quatro maneiras: espectador, participação real, participação simulada, e sem participação ou possível supervisão. Conforme demonstra na **figura 6**

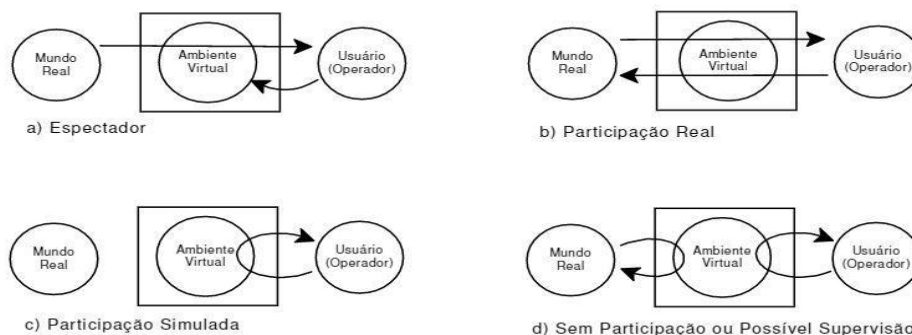


Figura 6 – Formas de Interação Mediada de um único usuário

Com exceção da participação simulada, onde o ambiente virtual pode ser tanto imaginário como a representação de um mundo real, nos outros casos o ambiente virtual deve representar fielmente o mundo real. É possível observar com base na **figura 6** que o espectador possui uma situação particular de sistema de telepresença, no qual realiza apenas inspeção, já no caso da participação real o usuário consegue não somente realizar inspeções ao ambiente virtual, como também interagir e modificar o seu estado, no caso da participação simulada o ambiente virtual pode ser imaginário ou simulando um cenário do mundo real, e no último caso corresponde a um exemplo de robô no ambiente virtual com possibilidade de supervisão, intermediando do mundo real ao usuário.

1.8.2 – Interação de Vários Usuários

Este tipo de Interação pode ocorrer de três maneiras: comunicação entre usuários, compartilhamento do ambiente virtual, e realização de trabalhos cooperativos no mundo real através do ambiente virtual compartilhado (**Figura 7**). ‘

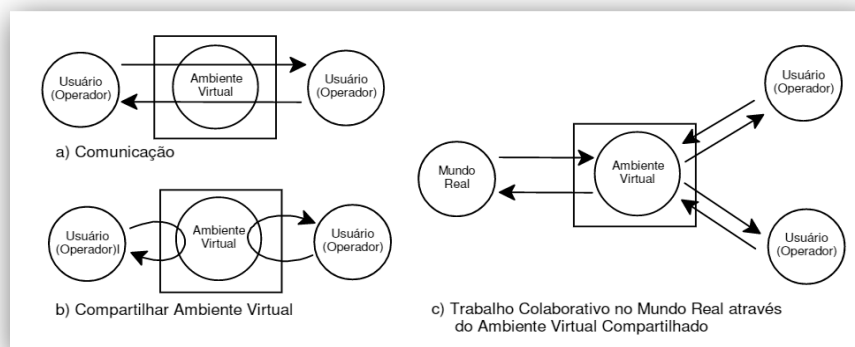


Figura 7 – Interação de Vários Usuários

Comunicação – Informações são trocadas através do ambiente Virtual

Compartilhamento do ambiente virtual – os usuários interagem entre si através do ambiente virtual

Trabalho Colaborativo – Cooperam entre si no mundo real através de representações de um ambiente virtual compartilhado, possibilitando o trabalho colaborativo ocorrer em um ambiente virtual imaginário sem vínculo direto com o mundo real (THOMSON et al, 2005), como no jogo de representação de vida que se passa em um ambiente virtual simulado, o Second Life. Grandes empresas como Amazon, Google e Ebay possuem suas próprias empresas nesse mundo virtual a fim de divulgar e vender seus produtos para seus clientes; e

ainda algumas empresas realizam até mesmo reuniões de negócios dentro destes ambientes virtuais.

1.9 Formas de Realidade Virtual

Para que a RV pudesse atender a diferentes necessidades e situações, fez-se necessário criá-la sob diferentes formas para cada caso. Alguns tipos existentes e mais utilizados são a RV de simulação, a RV de projeção, a Realidade Aumentada, a Telepresença, os Displays Visualmente Casados e a RV de mesa.

1.9.1 Realidade Virtual de Simulação

Esta é a primeira forma de realidade virtual, que foi utilizada de início para o treinamento de pilotos na Segunda Guerra Mundial. O usuário é levado a uma cabine onde tem acesso a um monitor e controles para que ele “dirija” um carro, uma lancha, um avião ou outro meio de transporte. Algumas possuem *feedback* tátil e sonoro, que propiciam uma maior sensação de realismo. Para isto são utilizados displays visualmente casados, com a função de enviar imagens e sons diretamente ao usuário geralmente através de um capacete, sendo este responsável por captar os movimentos da cabeça, e enviá-los ao computador, a fim de que sejam processados e gerados novos sons e imagens. (MACHADO, 1995)

Estes dispositivos, geralmente parecidos com um óculos, são o Lightweight, Óculus Rift, Google CardBoard(cuja tradução literal significa baixo peso) e os obturadores de cristal líquido (LCD) que comutam a projeção da imagem entre o olho esquerdo e direito, causando o fenômeno supracitado de estereoscopia. A **figura 8** demonstra um usuário utilizando o dispositivo de imersão *óculus rift* em um simulador de vôo.



Figura 8 – Simulador de vôo com realidade virtual

1.10 Dispositivos para Realidade Virtual

Para fazermos proveito da tecnologia de realidade virtual em um sistema, é preciso primeiramente utilizarmos certos dispositivos especiais de E/S de dados propriamente desenvolvidos a fim de aumentar o nível de imersão e realismo ao usuário.

1.10.1 Dispositivos de Entrada de Dados

Os dispositivos necessários para que se utilize a RV serão diferentes de acordo com o tipo escolhido para determinado caso, podendo-se em alguns casos fazer uso de mais de um tipo. Estes dispositivos têm contato direto com o usuário e transmitem suas informações do mundo real para o mundo virtual, e conseqüentemente causando modificações no estado deste ambiente.

1.10.1.1 Luvas de Dados

Estes dispositivos identificam o movimento dos dedos e das mãos através de sensores mecânicos ou, no caso das luvas de dados mais recentes como a *Data Glove*, utilizam-se sensores de fibra óptica. Com o movimento dos dedos, o cabo de fibra óptica é dobrado, reduzindo a passagem da luz. Essas variações da intensidade de luz são resumidas e transmitidas ao computador. Pela facilidade de utilização, ela aumenta a sensação de presença do usuário. A **figura 9** demonstra uma mulher utilizando este tipo de dispositivo. (MACHADO, 1995)



Figura 9 – Mulher usando Data-Gloves (Luvas de Dados)

Fonte: (<http://tecnologia.hsw.uol.com.br/realidade-virtual7.htm>)

1.10.1.2 Sensores de Movimento

Os sensores de movimentos atuais são capazes de captar movimentos sem o uso de luvas especiais, valendo-se apenas de sensores óticos. O *Leap Motion* (**Figura 10**), utilizado neste projeto, é um pequeno dispositivo com um sensor capaz de captar o movimento dos dez dedos das mãos e possui as dimensões de um pen drive.



Figura 10 - Dispositiva Leap Motion tamanho de um pen drive

Fonte: <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2014/05/o-que-e-leap-motion.html>

O *Leap Motion* utiliza infravermelho e câmeras para captar movimentos simultâneos com precisão de centésimos de milímetros, com latência inexistente para olhos humanos.

Para ser utilizado, um software instalado em um sistema operacional (Windows, MAC) transforma os dados de movimentos em comandos entendidos pelo S.O. Os controles são divididos em duas zonas: os movimentos de pouca profundidade, como navegação geral do cursor na tela e são captados mais próximos ao sensor, e os outros comandos, como cliques do mouse ou a ativação de botões, é realizado numa zona mais próxima ao monitor.

1.10.2 Dispositivos de Entrada-Saída de Dados

Os dispositivos de entrada-saída de dados são utilizados para aumentar o nível de imersão do usuário em um ambiente virtual, e alguns destes dispositivos são:

1.10.2.1 Dispositivos Visuais

Existem dois tipos de sistemas de imagens para a RV: a monoscópica e a estereoscópica. No primeiro, a mesma imagem é enviada para os dois olhos, enquanto que no segundo cada olho recebe uma imagem ligeiramente diferente, simulando o efeito de paralaxe na nossa visão. Existem também duas classes destes dispositivos, uma formada por vídeo

capacetes (HMD) e Head Coupled Display e outra formada por monitores e projeções. Para a RV, a exigência na exibição é de uma quantidade de 15 a 22 quadros por segundo. (MACHADO, 1995)

1.10.2.2 Videocapacetes (HMDs)

Possui duas pequenas telas de televisão e lentes especiais, que permitem que o usuário focalize as imagens corretamente. Rastreia-se a posição e a orientação da cabeça para determinar seu movimento e enviar esta informação ao computador. Serão enviadas ao usuário novas imagens de acordo com sua movimentação, dando-lhe a sensação de estar no mundo virtual. (MACHADO, 1995)

1.10.2.3 Óculus Rift

O *Óculus Rift* foi utilizado neste projeto, e é um equipamento de realidade virtual projetado especialmente para jogos eletrônicos (PANG et al, 1997), desenvolvido e fabricado pela Óculus.

É um sistema visual do tipo *Head-Mounted display*. Este equipamento possui um sensor externo de movimento para captar a movimentação do usuário em relação aos eixos x, y e z, através de três giroscópios, e possui ainda dois ecrãs OLED um para cada olho (por definição ecrã é um dispositivo usado para projetar e visualizar imagens). Estes ecrãs possuem uma resolução de 1080x1200 cada, uma taxa de atualização de 90hz (50% superior a um monitor convencional) e foi projetado para ocupar uma grande parte do campo visual humano. É recomendável que o usuário utilize um *headphone* para aumentar a sensação de imersão ao ambiente virtual simulando áudio 3D. A **figura 11** demonstra um usuário utilizando este dispositivo.



Figura 11 – Usuário utilizando o dispositivo Óculus Rift

Fonte: <http://www.techtudo.com.br/tudo-sobre/oculus-rift.html>

CAPÍTULO 2 – CONSCIÊNCIA SITUACIONAL (SAW)

Consciência Situacional (SAW) é um modelo cognitivo que explica o entendimento da situação por um indivíduo para a tomada de decisão em sistemas dinâmicos e complexos, e pode ser dividida em três níveis: percepção, compreensão e projeção.

O nível de percepção é caracterizado pela identificação de entidades relevantes do ambiente. O nível de compreensão vai além da percepção dos elementos presentes, incluindo compreender o significado e evolução dos estados de tais elementos. Já o nível de projeção se caracteriza pela habilidade de projetar ações dos elementos do ambiente, num futuro próximo (ENDSLEY, 1996).

Visto que atingir o estado de SAW é um processo que ocorre na mente do especialista, é de suma importância entender o nível de conhecimento obtido por ele na utilização de um sistema, para que, desta forma, falhas na cognição não prejudiquem o processo decisório. No nível representacional da informação, o objetivo é fornecer visualizações de informações que auxiliem o especialista a compreender um determinado cenário, e com base nestas informações e em seu nível adquirido de SAW, deverá realizar uma representação de um estado num futuro próximo para sequenciar atividades que possam vir a ocorrer (GOUIN et al, 2004).

Sistemas com o objetivo de auxiliar a consciência situacional dependem da qualidade da informação que são oferecidas para os especialistas. Estas informações fornecidas devem representar fielmente um cenário real, auxiliando na cognição e tomada de decisão, assim, o operador do sistema pode ser submetido à incerteza quando informações imperfeitas são fornecidas ao sistema, comprometendo o processo decisório.

Sabendo-se que a incerteza pode estar contida na informação, as visualizações de dados incertos devem fornecer ao usuário um recurso para a aquisição, manutenção e retomada da consciência situacional. O objetivo é fornecer ao tomador de decisão visualizações que auxiliem a abstração do estado atual da situação reduzindo o tempo de

análise de informações relevantes para que a projeção da sequência de atividades seja a mais correta possível em relação ao ambiente (GOUIN et al, 2004).

Em alguns casos, faz-se necessário o uso de apenas algumas informações relevantes de uma determinada situação para se obter um bom nível de consciência situacional.

Consciência Situacional é estudada em um vasto campo, incluindo a educação, condução, expedição de trem, manutenção, operação de usinas, previsão do tempo, operações militares, entre outros. A **figura 13** representa o modelo de Endsley referente ao processo de aquisição de SAW. Mica Endsley é uma engenheira e ex-cientista da força aérea dos Estados Unidos, autora de mais de 200 artigos científicos sobre consciência da situação, sendo reconhecida internacionalmente por seu trabalho pioneiro na concepção, desenvolvimento e avaliação de sistemas para apoiar a consciência da situação e tomada de decisão.

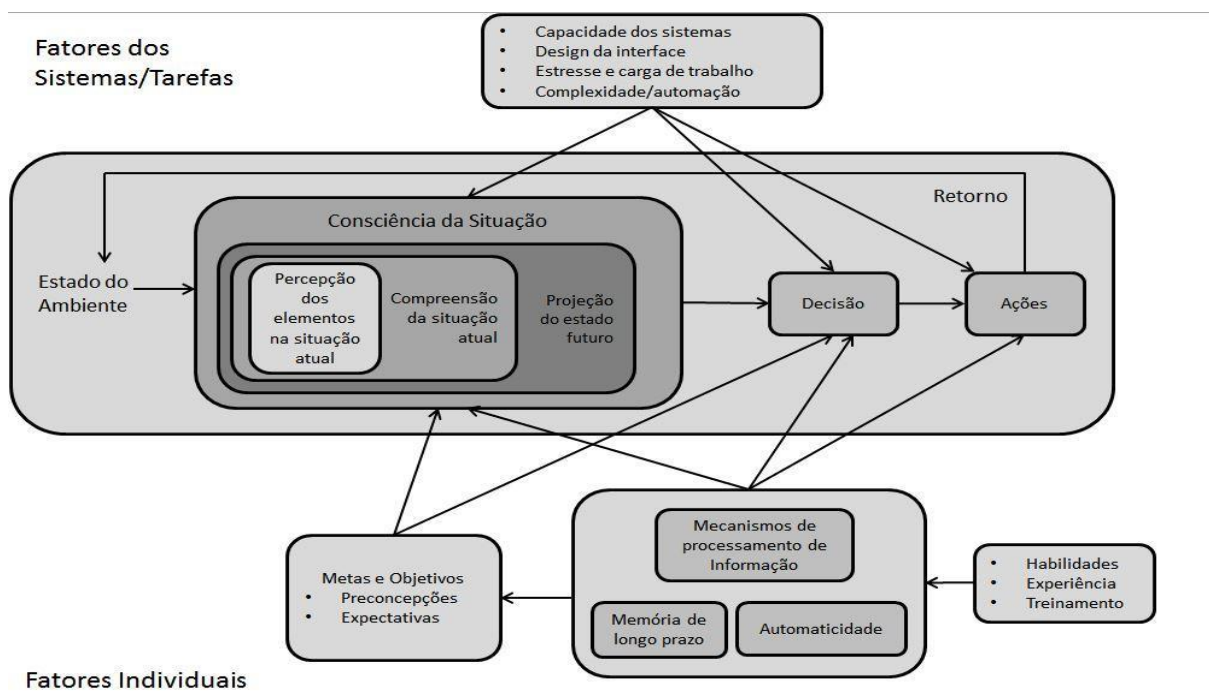


Figura 13 – Processo de Aquisição de SAW

2.1 Percepção

Para cada tipo de domínio as exigências para se obter SAW são bastante diferentes, porém, a percepção é o primeiro passo para se alcançar SAW, ou seja, perceber o estado, atributo e a dinâmica dos elementos relevantes no ambiente. Por exemplo, um piloto precisa perceber os elementos importantes para o voo, como: o status do sistema,

informações de latitude e longitude, condições climáticas, o terreno, juntamente com suas características relevantes.

Desta forma, manter a concentração em todos esses detalhes, não somente de sua cabine, mas também observando outras aeronaves, os dados de voo e de navegação podem tornar este processo algo muito desafiador e cansativo.

Outro exemplo é um oficial do exército que precisa detectar inimigos, posições civis e amigáveis, suas movimentações, características do terreno em que sua tropa se encontra, condições climáticas, obstáculos, entre outras características relevantes para se obter neste primeiro nível de SAW. Este nível pode vir tanto por meio visual, auditivo, tátil, gustativo, olfativos, ou até mesmo a combinação deles.

Outro exemplo é um produtor de vinho que consegue recolher informações muito críticas sobre o estado atual do processo de fermentação de um determinado vinho no processo de fabricação, através do gosto e do cheiro bem como através da inspeção visual, e assim determinar intervenções as mais adequadas. Este é um exemplo de como combinar vários meios para se obter este primeiro nível de SAW.

Um médico usa também todos os sentidos e as informações disponíveis para avaliar a saúde de seu paciente. Porém, um fator importante a se destacar neste primeiro nível é a experiência do usuário em uma determinada área: quanto mais experiente ele for, mais rápido e maior será seu campo de percepção dos elementos daquele cenário. Um cardiologista treinado pode ouvir pequenas diferenças no ritmo do batimento cardíaco ou ver pequenas diferenças em padrões significativos em uma impressão no monitor do computador, que um observador não treinado não conseguiria enxergar. Um piloto experiente pode saber que algo está errado em sua nave apenas ouvindo algum barulho no motor ou vendo o padrão de luzes em um campo de visão aérea.

Vale ressaltar que o indivíduo pode obter este primeiro nível apenas percebendo diretamente um ambiente, ao olhar pela janela ou ao sentir uma vibração, não estando este nível dependente de sistemas computacionais. Comunicações verbais e não verbais são tipos de fonte de informações que contribuem para este nível, como as situações anteriores à existência destes sistemas, como uma pessoa dirigindo um carro pela estrada, atenta a todo o momento a outros veículos na estrada e também a buracos ou animais na pista.

Uma das dificuldades e partes críticas neste nível, em muitos domínios, está associada ao nível de confiança de uma determinada informação.

Como em operações militares onde muitas vezes pode ser difícil avaliar todos os aspectos necessários da situação devido a problemas como escurecimento da visão, ruído, confusão, fumaça, e a dinâmica de uma situação que muda constantemente. O oficial do exército deve tentar determinar o estado atual dessas informações, enquanto as forças inimigas estão trabalhando a todo custo para reter essas informações ou fornecer informações falsas.

Jones e Endsley (1996) relatam que a maioria dos problemas com SAW no domínio de aviação ocorrem no nível 1, foi constatado que 76% dos erros de SAW em pilotos estavam relacionados a não perceber as informações disponibilizadas.

Em outros casos (cerca de 40%) este fato ocorreu pela falta de informação necessária ou por alguma falha no sistema fornecendo informações erradas. Por exemplo, uma transmissão de rádio com ruído, ou uma visão obscurecida do piloto pela mudança de clima dificultando a visão das linhas de uma pista de pouso.

Outros problemas (cerca de 20%) ocorrem pela perda da informação na memória, pelo fato do indivíduo obter novas informações. E em outros casos (cerca de 30%) muitas informações foram obtidas, menos a informação chave.

Todos esses problemas podem ocorrer pelo fato do especialista no momento de obter a informação ou de transmiti-las estar distraído por algum fator externo (por exemplo, um telefonema, ou uma conversa relacionada), porém o mais comum é o mesmo estar lidando com outras informações relevantes para tarefas competindo em seu ambiente de trabalho. Por exemplo, ele poderia estar olhando em outros monitores enquanto a informação chave estava presente em outro monitor.

Tendo tudo isso em mente, é possível entender as dificuldades e a responsabilidade em que o indivíduo tomador de decisão enfrenta neste primeiro nível. Uma profusão de informações simultâneas e precisar procurar somente pelas informações relevantes para aquela situação, ou seja, a informação chave.

2.2 Compreensão

O segundo passo para se obter SAW é compreender o significado e a importância das informações que foram obtidas no primeiro nível e combinar todas essas informações a fim de alcançar as metas e objetivos em questão.

O nível 2 é análogo a ter um alto nível de compreensão de leitura das informações adquiridas no primeiro nível.

Um exemplo simples, é considerar um motorista se aproximando de um cruzamento. O motorista então vê uma luz amarela no semáforo, por isso, ele entende que precisa proceder com cautela, em função da distância do cruzamento.

Sua percepção da taxa de desaceleração de seu carro em função da distância do cruzamento e tempo permite ao mesmo determinar se é mais viável parar ou acelerar. Essa compreensão que o motorista obteve em relação aos impactos daquela determinada situação e seus objetivos define o que é necessário para ter o nível 2 de SAW.

Outro exemplo é um problema durante uma decolagem em que o piloto ao ver as luzes de aviso piscando em seu painel deve rapidamente determinar a gravidade do problema em termos de aeronavegabilidade imediata da nave e combinar isso com a situação da pista, terreno, clima em que se encontra aquela situação atual, a fim de saber se é uma situação para abortar o voo.

Um piloto novato pode ser capaz de obter o mesmo nível de percepção que um piloto experiente, porém pode não conseguir compreender e entender o significado da quantidade de informações presentes em sua frente, dificultando assim a compreensão da situação e sua tomada de decisão. Em operações militares e de gerenciamento de emergência a falta deste segundo nível pode fazer com que situações que poderiam ter sido evitadas com uma boa tomada de decisão acabem em tragédia.

Para um comandante militar este nível pode significar compreender um relatório referente à movimentação das tropas inimigas, significando onde elas estejam, para onde vão ou estão se concentrando, para que seja possível tomar uma decisão correta. O comandante, apenas pelo fato de ver faixas de veículos na estrada já pode identificar quais os veículos e o número de tropas inimigas que estão se aproximando.

Estudos revelam que cerca de 19% dos erros cometidos em SAW no domínio aéreo envolvem problemas relacionados a este nível (Jonas & Endsley, 1996), ou seja, as pessoas são capazes de obter o primeiro nível tanto por meio visual quanto auditivo, porém não são capazes de interpretar o significado dessas informações.

2.3 Projeção

Uma vez que a pessoa já percebeu todos os elementos e as informações ao redor de sua situação e compreendeu o significado e a importância dos mesmos em relação à meta atual, o próximo passo de SAW é a capacidade de prever o que possa vir a ocorrer com os elementos percebidos no primeiro nível em um futuro próximo. Este é o 3º Nível de SAW. Uma pessoa só pode atingir este nível após uma boa compreensão da situação (Nível 2 SAW) e com um bom entendimento do funcionamento e da dinâmica do sistema que eles estão trabalhando.

Com base neste nível, o motorista do exemplo anterior sabe que se prosseguir para o cruzamento estará suscetível de provocar um acidente. Este Nível é fundamental para uma boa tomada de decisão, já que o tomador de decisão deve prever o que possa vir a ocorrer em relação alguma ação a ser tomada, se o efeito será positivo ou negativo.

Por exemplo, comandantes do exército podem projetar a direção em que as tropas inimigas irão se aproximar e os prováveis efeitos de suas próprias ações com base no segundo nível que ele desenvolveu.

Outro exemplo é no domínio aéreo, no qual pilotos e controladores de tráfego aéreo devem trabalhar ativamente para projetar os movimentos de outras aeronaves que se aproximam a fim de antecipar possíveis problemas.

Especialistas em vários domínios, não somente militares, dedicam uma boa parte de seus tempos livres para treinarem esse nível, a fim de se obter melhores resultados em relação a projeções em diversas situações [ENDSLEY, 2011].

Até mesmo na área de jogadores profissionais se utilizam SAW, como no xadrez, em que o jogador deve perceber o movimento das peças do oponente e tentar compreender sua estratégia antes de tomar alguma decisão. Também em outros jogos, como o pôquer, em que os jogadores utilizam os padrões de apostas dos adversários para tentar descobrir a sua

estratégia e com base nisto prever suas cartas com base em suas apostas, e tomar uma decisão correta.

A falta de projeção com precisão (Nível 3) pode ocorrer devido à insuficiência de recursos mentais (caso a pessoa esteja sobrecarregada com outro processamento de informações, por exemplo) ou devido a conhecimento insuficiente do domínio.

Somente 6% das falhas em relação à SAW no domínio aéreo se enquadram neste nível (Jonas & Endsley, 1996), devido a dificuldades de se obter os níveis anteriores já citados, e assim não progredindo ao terceiro nível.

2.4 Trabalhos Relacionados

Nos últimos anos a necessidade de tomadas de decisões em crises e os suportes de sistemas de simulação vêm crescendo (BOIN, 2009), porém, as pesquisas na área dedicaram pouca atenção ao uso de ferramentas de simulação de tecnologia da informação (TI) (DUGDALE et al, 2010) até os últimos anos. Hoje em dia, com o aumento da criminalidade e acidentes, exige-se o suporte de tecnologias como a realidade virtual em vários projetos, os quais serão citados no decorrer deste tópico. A tecnologia de RV é utilizada para criar uma interface mais amigável aos usuários que utilizam sistemas de visualização de informações para cenários de emergência, otimizando assim o conhecimento e a consciência da situação (SAW) (ENDSLEY, 1996).

Em (PARAVATI et al, 2012) foi desenvolvido um sistema inteligente de iluminação pública para reduzir os níveis de consumo de energia em diferentes domínios. O projeto apresenta um quadro de monitoramento o qual oferece suporte e controle à iluminação das ruas através de simulações, estabelecendo condições ambientais e de iluminação. O sistema também utiliza recursos avançados de 3D *vision* e interação, e uma interface virtual capaz de recriar o efeito de iluminação nos ambientes da cidade virtual, como por exemplo, os efeitos de escurecimento de luz. Além disso, técnicas estereoscópicas imersivas foram utilizadas juntamente com dispositivos especiais de imersão, que fazem com que os operadores explorem melhor esses ambientes através de navegações virtuais, fornecendo uma melhor sensação de realismo. Comandos de vozes, poses e gestos também foram utilizados e reconhecidos pelo sistema, detectados por sensores de dispositivos especiais de interação.

O sistema é capaz de detectar se existem pessoas em um determinado local, e automaticamente ativar a iluminação naquela área a fim de melhorar a segurança pública. O sistema também possui uma opção de auto programar os horários e os níveis de iluminação para determinados pontos, nos quais há mais movimentações.

(BROEK et al, 2011) desenvolveu um sistema para domínios militares marítimos, onde ocorrem atos de pirataria, tráfico de drogas entre outros atos criminosos no meio marítimo. O objetivo do projeto foi melhorar a consciência de situações de ameaças em cenários marítimos através de combinação de informações com base em sensores contendo dados e informações de outras fontes.

O sistema detecta os padrões de navegação de determinados navios e localidades, e suas atividades diárias, gerando assim informações que são fundidas e armazenadas com base em ontologias de domínio. Uma análise com base nestas informações foi realizada para gerar contraste com comportamentos suspeitos de algum navio. O sistema então utilizou várias técnicas aplicadas a estas ontologias para informar o grau de suspeição e o nível de ameaça deste navio.

Consciência situacional não existe somente em domínios militares, mas também em áreas de saúde, como em (BUI et al, 2010), onde foi desenvolvido um sistema de suporte médico neurologista para detectar a evolução de doenças como tumores.

Diante de sistemas de informações médicas que acumulam uma grande quantidade crescente de dados e de complexidade, buscando encontrar dados de forma rápida e eficiente, o EMR (BUI et al, 2010) teve como principal objetivo facilitar o uso destes sistemas. A visualização da EMR foi adaptada ao usuário, de forma que tornou-se possível buscar e encontrar rapidamente as tarefas e os problemas médicos de forma abrangente.

O sistema armazena informações pessoais do paciente, como nome, idade, gênero, entre outros, como também a descrição detalhada da doença que o paciente possui, acompanhada do histórico detalhado da doença em uma linha temporal, para que o médico tenha rápida leitura sobre se a doença está evoluindo ou regredindo.

A Realidade Virtual é utilizada também para muitas formas de treinamento, como na utilização de simuladores de treinamento em cenários 3D de emergência virtual. (CHITTARO et al, 2014) desenvolveu um sistema que tem como principal objetivo aumentar a segurança das pessoas ao capacitá-las a tomar decisões corretas em situações com risco de morte que

poderiam enfrentar na vida real, aprendendo a evitar erros comuns e escolhendo ações corretas durante a simulação.

Foi utilizada uma simulação interativa 3D de um cenário de aviação em uma situação de emergência no qual o avião está a afundar no mar após queda, e os usuários podem interagir com o sistema a todo o momento de forma realista. Durante a simulação, informações sobre a situação são mostradas numa tela, juntamente com ações entre as quais os usuários podem escolher. Ações recomendadas para cada situação também são mostradas dependendo da decisão tomada, e uma nova animação é demonstrada na simulação. Em casos de uma ação correta, os usuários veem consequências positivas em termos de progressos para a sobrevivência. E por fim, para que estas simulações sejam cada vez mais realistas foram alteradas as características visuais e auditivas específicas do ambiente, aumentando-se assim o nível de medo despertado nos usuários.

No projeto (JULIEN et al, 2003) um ambiente virtual de formação e treinamento para o Corpo de Bombeiros foi desenvolvido em colaboração com o departamento de bombeiros de Atlanta. Esse ambiente virtual permite ao usuário navegar em torno do ambiente, visualizar de qualquer ângulo uma casa modelada em 3D em chamas; permite executar e visualizar comandos referentes às ações do corpo de bombeiros (desenvolvidas junto com a equipe e os oficiais); e o ambiente é modificado conforme cada ação realizada, assim, a fumaça e o fogo é visto de forma realista pelo usuário do sistema.

O sistema é utilizado para que o comandante treine estagiários, que conduzirão bombeiros virtuais na execução de ações necessárias para cada tipo de situação diferente de incêndio virtual. A sequência correta de comandos executados com êxito irá apagar a chama com o mínimo de perigo possível para os bombeiros e o mínimo de dano causado à casa. Para realizar esta simulação, este projeto utilizou uma biblioteca de ambiente virtual simples (SVE), uma estrutura extensível para a construção de aplicações que utilizam ambientes virtuais.

ViPR é um tipo de planejamento de salas virtuais. Em (BROUGHTON ET AL, 2006) foi desenvolvido um ambiente de visualização 3D para informações hierárquicas. O centro de análise laboratorial (FOCAL) em defesa da organização de defesa e tecnologia da Austrália (UFPR), visa explorar novos paradigmas para a Consciência Situacional e comando e controle (C2), visando aperfeiçoar o futuro das operações nos centros de comando militar,

fazendo uso de novas tecnologias desenvolvidas como a realidade virtual e animação 3D em tempo real.

Este trabalho recente incluiu o desenvolvimento de um projeto e protótipo conceitual do planejamento de salas virtuais (ViPR), onde o usuário pode visualizar informações de multimídia referentes a uma determinada situação nas paredes dessas salas virtuais imersivas. Não somente visualizar, o usuário pode também interagir com essas informações. Quartos octogonais foram criados cada um possuindo uma gama de informações de diferentes níveis como tabelas, gráficos, imagens, textos. É possível ao usuário navegar por este ambiente através de portas que dão acesso a quartos contíguos, explorando assim diferentes níveis de abstração.

Uma opção para acessibilidade seletiva em todo o ambiente também foi possível através da interação com um mapa 3D. Este mapa possui um gráfico representando uma árvore ontológica no qual informações referentes ou semelhantes interligam conexões que possibilitam ao usuário acesso direto à informação desejada.

Um sistema chamado 3DNSITE foi desenvolvido em (PINTORE et al, 2012), esse sistema tem como funcionalidade simplificar o reconhecimento de locais na gestão de crise através de um sistema de visualização 3D interativa em rede, utilizando modelos tridimensionais juntamente com fotografias incorporada em dispositivos com capacidade de hardware heterogêneos e interligadas em diferentes tipos de redes.

Além disso, o sistema apresenta uma interface de navegação simples e intuitiva não necessitando que o usuário tenha um conhecimento hábil em sistemas que possuem interações com realidade virtual e objetos tridimensionais.

2.5 ESAS

Neste projeto foi utilizado uma arquitetura e um protótipo de sistema de avaliação de situações de emergência, denominada ESAS (*Emergency Situation Assessment System*) para realizar toda a busca de dados referentes às situações de emergência. Este sistema tem como principal objetivo armazenar todas as informações das denúncias realizadas pelas testemunhas. Por exemplo, uma situação de emergência caracterizada como roubo possui basicamente quatro entidades: um ou mais criminosos, objeto roubado, local do crime e uma ou mais vítimas.

Para este projeto foi utilizado um cenário de roubo, sendo este um caso real, obtido a partir de registros da PMESP (Polícia Militar do Estado de São Paulo), e foi desenvolvido um exemplo completo do processo de avaliação de situações a partir de um estudo de casos.

A **figura 14** apresenta a interface do protótipo sistema ESAS:

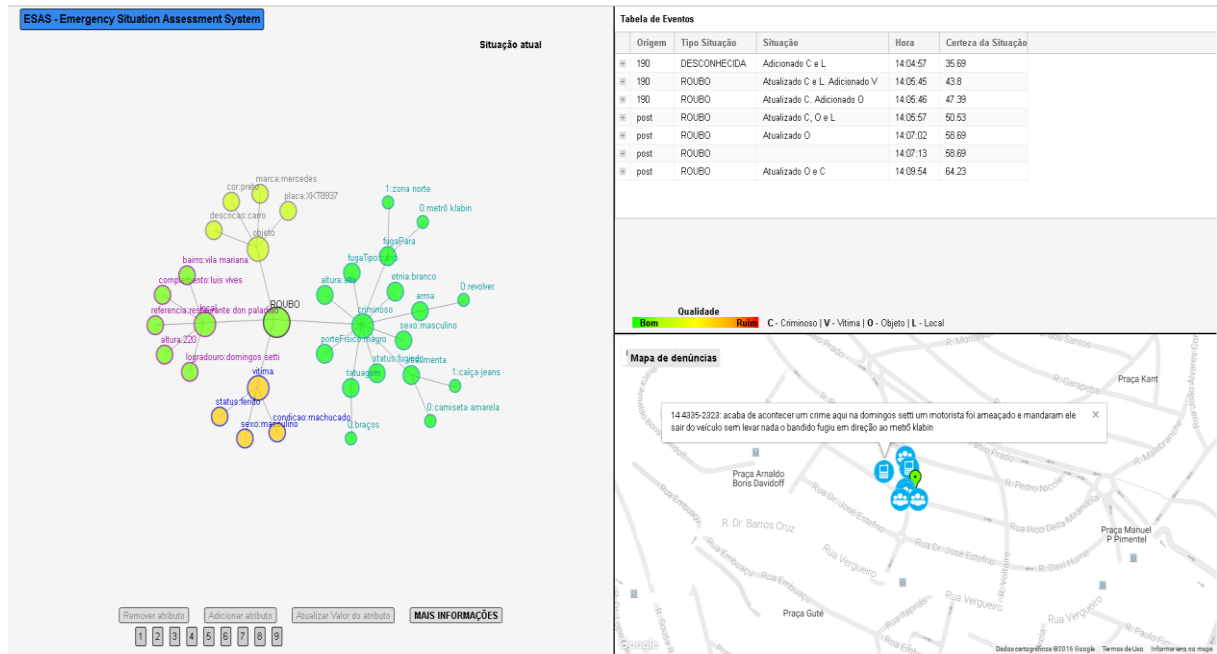


Figura 14 - Interface do Protótipo Sistema ESAS

O processo de avaliação e construção de tais informações referentes a uma situação, desde sua aquisição até a gestão de interface, como mostra a **figura 14**, segue uma metodologia de avaliação de qualidade de dados e informações. Nesta seção será explicado brevemente o funcionamento do sistema ESAS.

Foram realizadas requisições a este sistema para a busca das informações referentes à situação de roubo, e por fim estas informações foram utilizadas na realização do trabalho. É possível observar na **figura 14** que a primeira denúncia foi realizada por meio de uma ligação ao serviço de atendimento da PMESP (190). Assim, o sistema ESAS transcreve a mensagem da denúncia como mostra na **figura 15**.

14 4335-2323: acaba de acontecer um crime aqui na domingos setti um motorista foi ameaçado e mandaram ele sair do veículo sem levar nada o bandido fugiu em direção ao metrô klabin

Figura 15 – Denúncia recebida pela vítima e transcrita pelo sistema ESAS

A denúncia da **figura 15** foi realizada por meio de “Aquisição de Dados HUMINT”, ou seja, por uma pessoa. O sistema ESAS realiza, a partir destes dados, uma classificação dos objetos e dos atributos presentes na denúncia, armazenando os mesmos em formato JSON, como mostra a **figura 16**.

```

{id":1,
"categoria":"Situação Desconhecida",
"denuncias":{
  "id":1,
  "categoria":"DESCONHECIDA",
  "fonte":"190",
  "horaInicio": "2015-05-29 14:03:27",
  "horaFin": "2015-05-29 14:04:57",
  "hora":"14:04:57",
  "gps":["-23.593611", "-46.627527"],
  "origemMensagem":"14 4335-2323",
  "texto":"acaba de acontecer um crime aqui na domingos setti um motorista foi ameaçado e mandaram ele sair do veículo sem levar nada o bandido fugiu em direção ao metrô Klabin",
  "eventos":"Adicionado C e L",
  "certeza":35.69
"tempo":{
  "quantidade": "0",
  "tipo": "2015-05-29 14:03:49",
"situacao":{
  "criminoso":{
    "sexo": "masculino",
    "status": "fugindo",
    "fugaTipo": null,
    "fugaPara": "metrô klabin",
    "descricao":null,
    "idade":null,
    "altura":null,
    "condicao":null,
    "etnia":null,
    "porteFisico":null,
    "corCabelo":null,
    "tipoCabelo":null,
    "vestimenta":null,
    "marcaVestimenta":null,
    "arma":null,
    "tatuagem":null,
    "palavraRef":null,
    "completude":23.80,
    "consistencia":null},
    "local":{
      "logradouro": "domingos setti",
      "tipologradouro":null,
      "complemento":null,
      "bairro":null,
      "altura":null,
      "direcao":null,
      "sentido":null,
      "referencia":null,
      "numero":null,
      "gps":["-23.593680", "-46.627397"],
      "completude":33.33,
      "consistencia":null}},
    "objeto":{
      "descricao":null,
      "condicao":null,
      "marca":null,
      "modelo":null,
      "tamanho":null,
      "cor":null,
      "ano":null,
      "placa":null,
      "porte":null,
      "completude":0,
      "consistencia":null}},
    "vitima":{
      "palavraRef": null,
      "sexo":null,
      "status":null,
      "idade":null,
      "altura":null,
      "condicao":null,
      "etnia":null,
      "porteFisico":null,
      "corCabelo":null,
      "tipoCabelo":null,
      "corVestimenta":null,
      "marcaVestimenta":null,
      "completude": 0,
      "consistencia":null},

```

Figura 16 – Dada do sistema ESAS referente às instâncias de uma denúncia em formato JSON

A partir da denúncia demonstrada na **figura 15**, o sistema ESAS consegue interpretar e armazenar informações como: status do Criminoso, o local de onde o criminoso está fugindo, o logradouro do local, o horário e a data da denúncia, a fonte da denúncia, o número de origem da denúncia como também sua mensagem.

O sistema, posteriormente, oferece serviços para acesso às informações de cada denúncia.

CAPÍTULO 3 – FERRAMENTAS DE IMPLEMENTAÇÃO

Neste capítulo é abordado sobre a ferramenta utilizada para o desenvolvimento do projeto, o Unity3D é uma plataforma de motor de jogos, no qual possui muitos componentes que facilitaram na implementação do trabalho, ao decorrer do capítulo é explicado o funcionamento de alguns desses componentes.

3.1 – Unity 3D

Para explicar como foi implementado cada etapa do projeto é preciso entender como funciona a mecânica do unity, os games do unity são baseados em cenas, assim os *Game Objects* são praticamente todos os objetos dentro da cena. São todos os elementos posicionados dentro da cena através de um sistema de coordenadas, sejam em duas ou três dimensões: Câmeras, modelos, luzes, sistemas de partículas: todos são *GameObjects*. Um game pode possuir várias cenas, e cada cena pode possuir diferentes *gameobjects*, *câmeras*, *terrenos* entre outros objetos, como por exemplo, as fases de um jogo podem ser construídas em cenas diferentes. Todos esses objetos podem se movimentar dentro da cena, o unity para isso o utiliza um conceito chamado de *Transform*, que por sua vez contém as coordenadas para a posição, rotação e escala do objeto, como pode ser visto na **figura 17**, a rotação é expressa em coordenadas retangulares através dos chamados ângulos Euler, que variam de 0 a 360°.

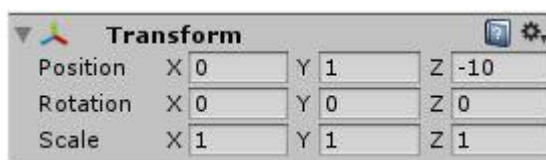


Figura 17 – Transform GameObjects Unity3D

Para realizar a movimentação de um objeto na cena do unity, basta utilizar um script para modificar as posições desse objeto, alterando assim as coordenadas do objeto. Para rotacionar um objeto é um pouco mais complexo, as rotações do unity são controladas por uma classe chamada *Quaternion*, algum dos métodos que essa classe possui é o *Slerp* (*from, to, time*) e o *Lerp* (*from, to, time*) que servem para interpolar o objeto de uma *rotation* (*from*) para outra (*to*) por um tempo *time*.

Outros elementos que foram utilizados no projeto e são muito importantes dentro das cenas do unity são os modelos, materiais e texturas (*Models, Materials e Textures*). Os modelos são elementos gráficos dentro de uma cena. Esses elementos são modelados utilizando materiais, texturas e os chamados *shaders*. Os materiais são baseados em *texturas* e *shaders*: a *textura* diz o que é desenhado na superfície do material, enquanto os *shaders* dizem como será desenhado.

Cada *GameObject* possui o seu tipo específico de como será a forma de seu objeto, esses tipos são chamados de *Mesh*, dentre eles são: Cubo (Cube), Esfera (Sphere), Quadrado (Quad), Cilindro (Cylinder), Plano (Plane) entre outros.

Outro elemento muito complexo que o unity permite a criação são os chamados prefabs (pré-fabricados), estes objetos são complexos, normalmente constituídos por um conjunto de vários outros objetos, que são montados de uma determinada forma e salvos para serem reutilizados nas cenas do seu game a qualquer momento, estes objetos podem ser instanciados, na verdade o que é instanciado é um clone dos objetos referente a estes *prefabs*.

No caso deste projeto foi construído um objeto *prefab* para cada Modelo 3D que foi instanciado na cena de crime dentro do ambiente *streetview*, em que cada um representa uma entidade da situação. Foi utilizado um *prefab* em formato de Cubo (Cube) com uma imagem referente a cada tipo de denúncia, caso tenha sido uma denúncia via 190, uma imagem referente a um celular é desenhado neste objeto, caso contrário utiliza-se uma imagem referente a rede social. Cada *prefab* possui um componente de texto também, sendo possível armazenar as informações referente as mensagens das denúncias recebidas de cada vítima, a **figura 18** demonstra estes *prefabs* intanciados em uma cena vazia do projeto.

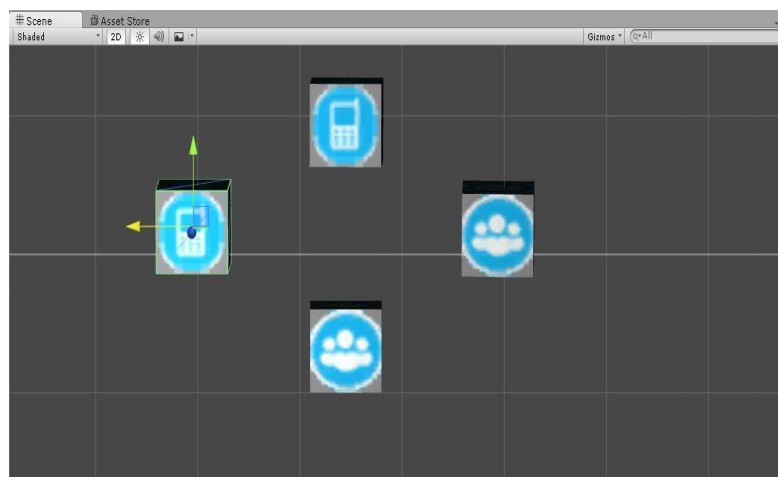


Figura 18 – Prefabs Instanciados em uma cena vazia no Projeto

Por fim os *scripts* do Unity são elementos que são associados aos *GameObjects* definindo tal comportamento para o mesmo. Esses scripts podem ser criados por algumas linguagens de programação tais como: C# (utilizado no projeto), Javascript ou BooScript. O Unity oferece inúmeras classes que cuidam de translação, rotação, colisão, animação, entre outros, para os *GameObjects* dentro da cena do Game.

CAPÍTULO 4 - PROJETO

Este capítulo descreve todas as etapas realizadas para a conclusão do trabalho, descrevendo todos os passos realizados em cada uma delas, da implementação do sistema até a incorporação da tecnologia de realidade virtual. Este capítulo foi dividido em duas etapas:

1) A primeira etapa descreve como foi implementado o mapa panorâmico do Google Maps no Unity3D, onde o usuário tem a visualização das informações de todas as denúncias recebidas pelas testemunhas geolocalizadas em determinados pontos do mapa em que cada denúncia foi efetuada, tanto por via 190 quanto por redes sociais. O usuário tem a opção também de navegar livremente pelo mapa através de gestos que são captados e interpretados por um dispositivo de sensor de movimento. Desse modo, o usuário pode navegar e visualizar qualquer rua ou local a partir daquele ponto específico da situação em questão.

2) A segunda etapa descreve como foi implementado o *street view* no Unity3D, onde o usuário tem a opção de visualizar um determinado ponto específico clicando no mapa panorâmico, como também visualizar a cena de um determinado crime contendo modelos 3D estrategicamente posicionados no *street view*. Estes modelos são referentes a uma determinada situação de emergência, como um Modelo 3D de uma vítima ferida ou um criminoso fugindo de moto. Por fim, cada Modelo 3D possui suas informações específicas que foram registradas no sistema com base nas denúncias recebidas.

A **figura 19** mostra o fluxograma dessas etapas e a sequência em que cada uma foi realizada de acordo com o desenvolvimento do trabalho.

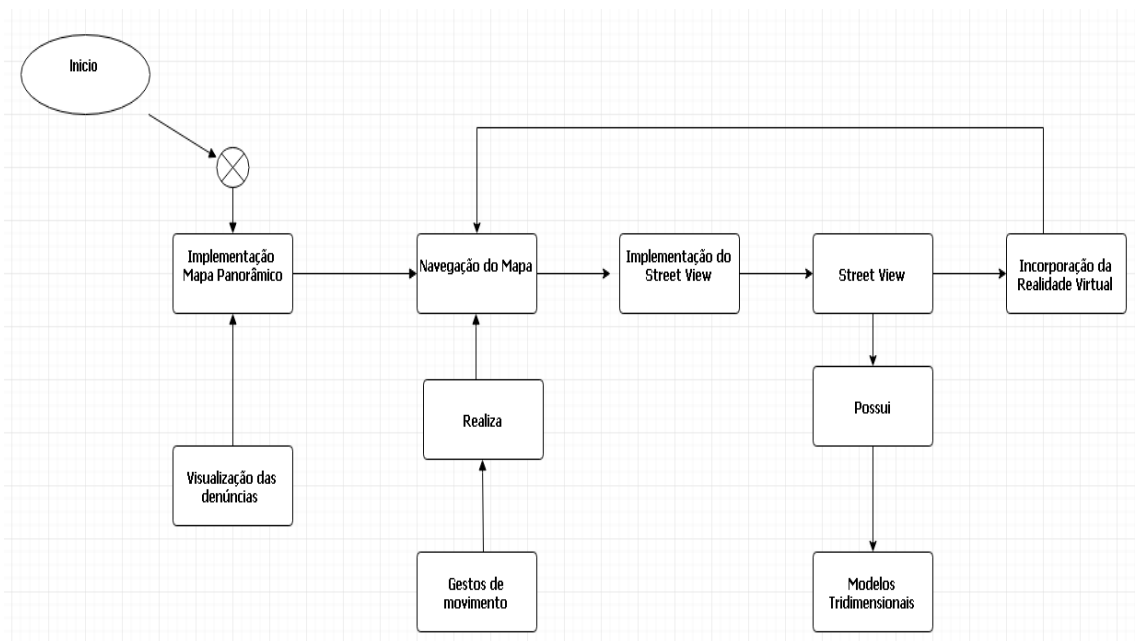


Figura 19 – Diagrama de Blocos do Sistema

4.1 Implementação do Mapa Panorâmico do Google Maps

Depois de ter explicado o funcionamento de alguns dos elementos que foram utilizados no projeto, este capítulo descreve as etapas realizadas para a implementação do mapa panorâmico, utilizando a ferramenta Unity3D.

4.1.1 – API GoogleMaps

A primeira etapa foi criar um script para que seja possível conectar-se à API do Google Maps utilizada no projeto. O objetivo é realizar requisições via *webservices* ao servidor do Google Maps, solicitando assim imagens de um determinado ponto (Local) a partir de dados como: Endereço (Address) ou Latitude e Longitude enviados por parâmetros ao servidor do Google Maps. Neste projeto os dados enviados por este parâmetro são obtidos através de uma requisição ao sistema ESAS, que possui todas as informações necessárias da situação. Vejamos a **figura 20**.



Figura 20 – Parâmetros Google Maps

O primeiro parâmetro em amarelo representa o endereço, o segundo e o terceiro em vermelho representa a latitude e longitude, e por último o nível de zoom. Esse *Script* fornece outros parâmetros também como *Zoom* do mapa, tal como o tipo do mapa: Híbrido (*Hybrid*), Terreno (*Terrain*), Satélite (*Satellite*) ou *RoadMap*. O tipo de mapa utilizado para o projeto foi o Híbrido, a **figura 21** mostra esse script no projeto.

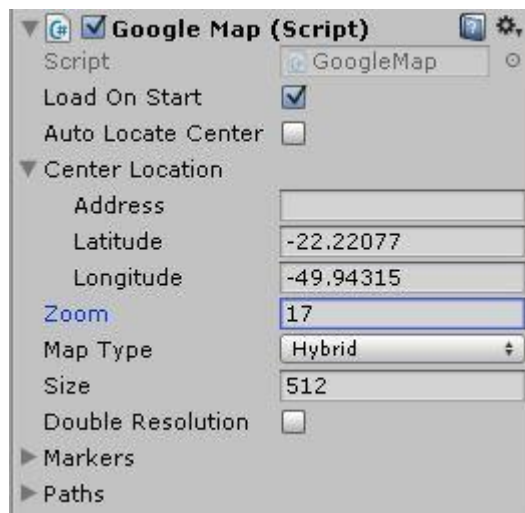


Figura 21 – Script Google Maps

Esse *Script*, após retornar a resposta de uma requisição, utiliza um componente de renderização (*Renderer*) próprio do Unity para transformar a imagem retornada em uma textura (*Texture*) a ser desenhada em um *material* do *GameObject*.

Foram criados nove *GameObjects* do tipo *Mesh* plano (*Plane*) renomeados como: Mapa (1), Mapa (2), Mapa (3) assim sucessivamente, sendo o mapa do meio o Mapa Central (5), e os demais adjacentes são complementares. Dessa forma, cada um destes objetos representam uma parte do mapa panorâmico, todos incluídos em um *GameObject* denominado “Mapa_Principal”, assim, ao realizar a movimentação do “Mapa_Principal”, o restante dos mapas também são movimentados juntamente.

Por fim, foi alterada a posição de cada mapa para que todos se encaixem conjuntamente. Como resultado, temos a sensação de estarmos visualizando um único mapa panorâmico, como na **figura 22** onde se demonstra em uma visão 2D os mapas acoplados uns aos outros, e os quadrados azuis demonstram alguns mapas propositalmente selecionados para uma melhor exemplificação do resultado. No projeto final foi alterado o modelo do mapa, para algo parecido com um modelo curvo, assim, ao utilizarmos o dispositivo de imersão

(*Oculus Rift*), temos a sensação de estarmos olhando para algo parecido com um “telão” de cinema, como mostra a **figura 23**.

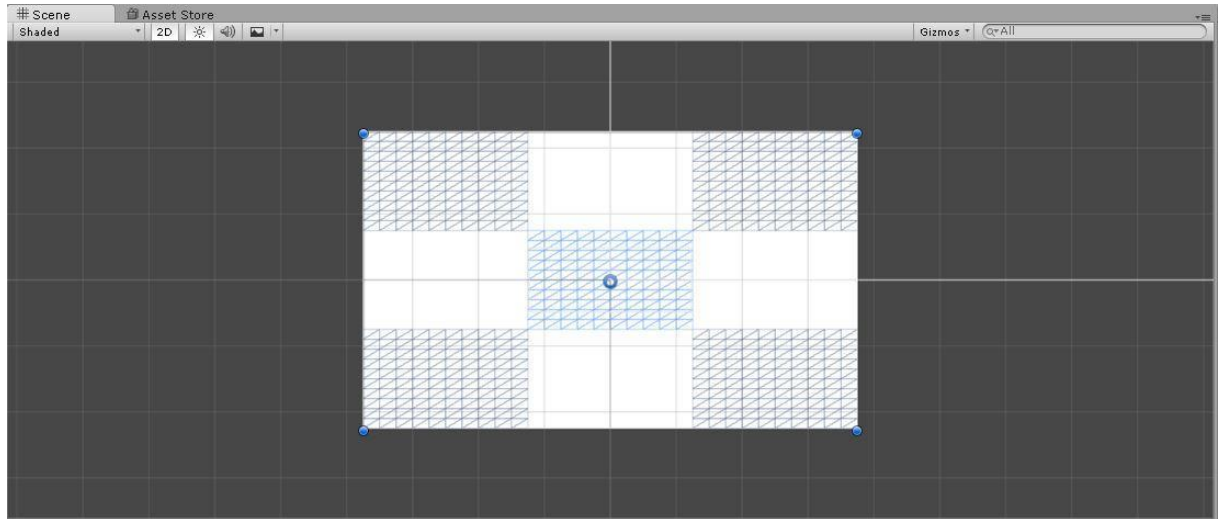


Figura 22 - Todos *GameObjects* mapas encaixados um ao outro

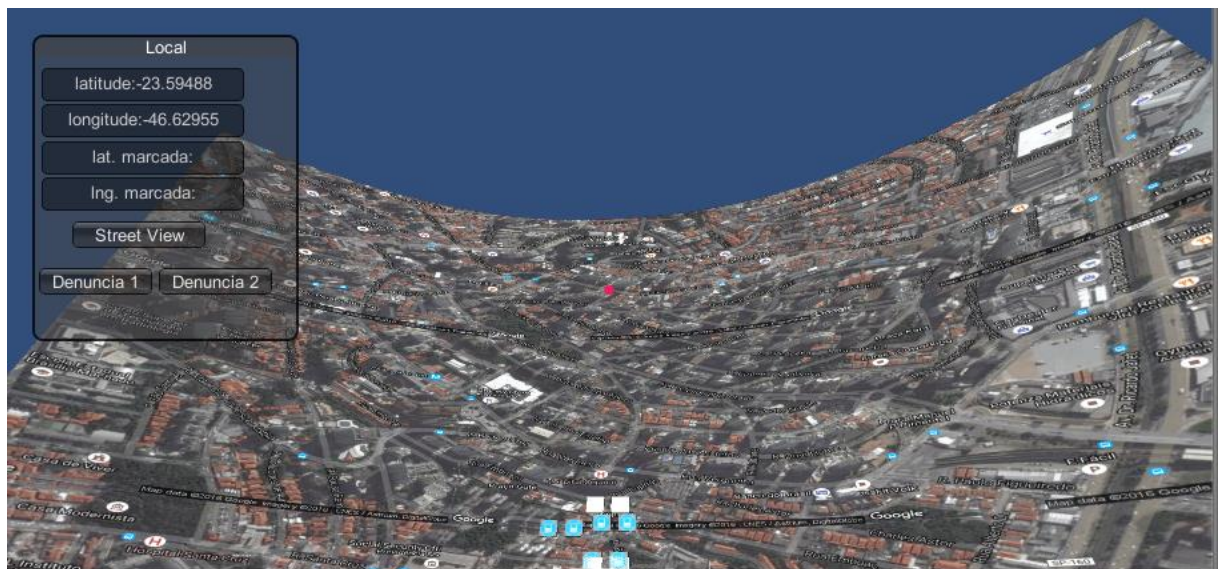


Figura 23 – Mapa Panorâmico no ambiente virtual

O próximo passo foi incluir em cada mapa um *script* da API GoogleMaps (**Figura 24**) para que, ao iniciarmos o projeto, o *script* associado a cada mapa faça uma requisição ao servidor do GoogleMaps solicitando uma imagem para aquele mapa. Para isso, são utilizadas os dados referentes à latitude e longitude de cada mapa, e por fim é utilizado um Componente de renderização próprio do Unity, transformando assim a resposta da requisição em uma

textura (*texture*) da imagem do local solicitado, preenchendo assim o material de cada mapa como mostra na **figura 25**

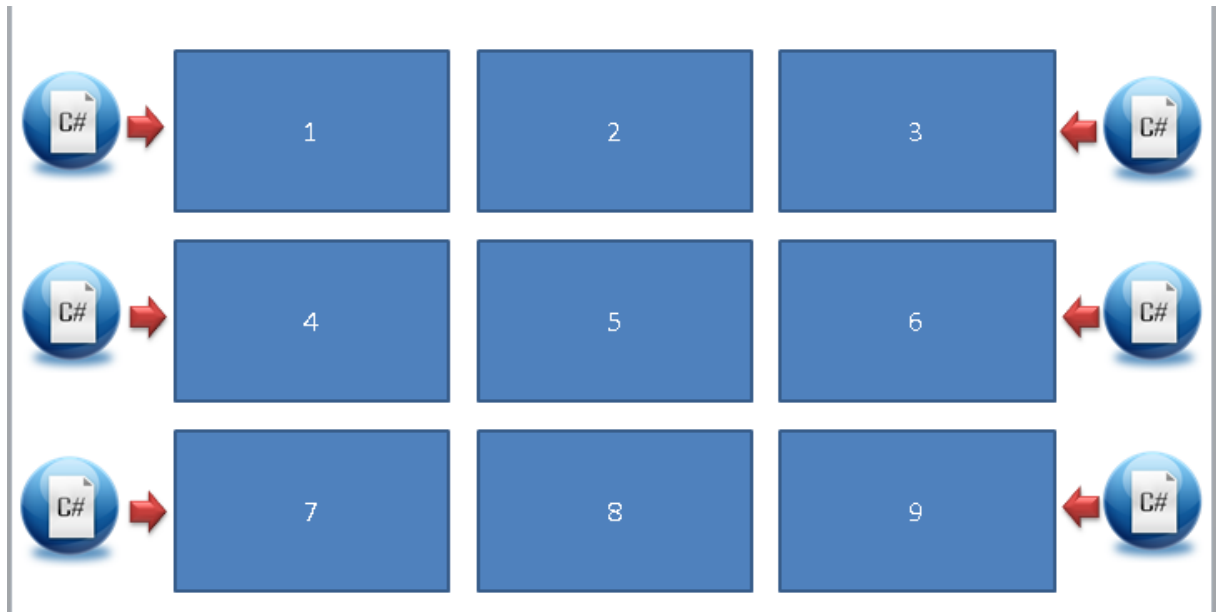


Figura 24 – Scripts API Google Maps associados a cada mapa



Figura 25 – Mapas com as imagens retornadas do Google Maps

Ao “encaixarmos” todos eles, temos a impressão de estarmos visualizando apenas um único mapa grande, com uma visão muito mais ampla ao redor, como mostra na **figura 26**.



Figura 26 – Todos os mapas acoplados juntamente

Para que todos os mapas se encaixem de tal maneira a não haver partes de imagens dos locais repetidas em mais de um mapa, ou espaços vazios, foi preciso implementar um *script* cuja função é realizar a interação de um mapa com outro, buscando seus dados de latitude e longitude, e com base no *mapa central* (5), calcular quais são as coordenadas dos mapas adjacentes.

Para tal foi utilizado o próprio GoogleMaps em um browser, e realizada a navegação do mapa através das setas do teclado, obtendo como resultado uma média para realizar os cálculos das coordenadas de latitude e longitude de um mapa a outro. A diferença foi um valor muito pequeno, começando a partir da 3^a casa decimal como mostra a **figura 27**.

/@-22.2356674,-49.9717246,16.5z

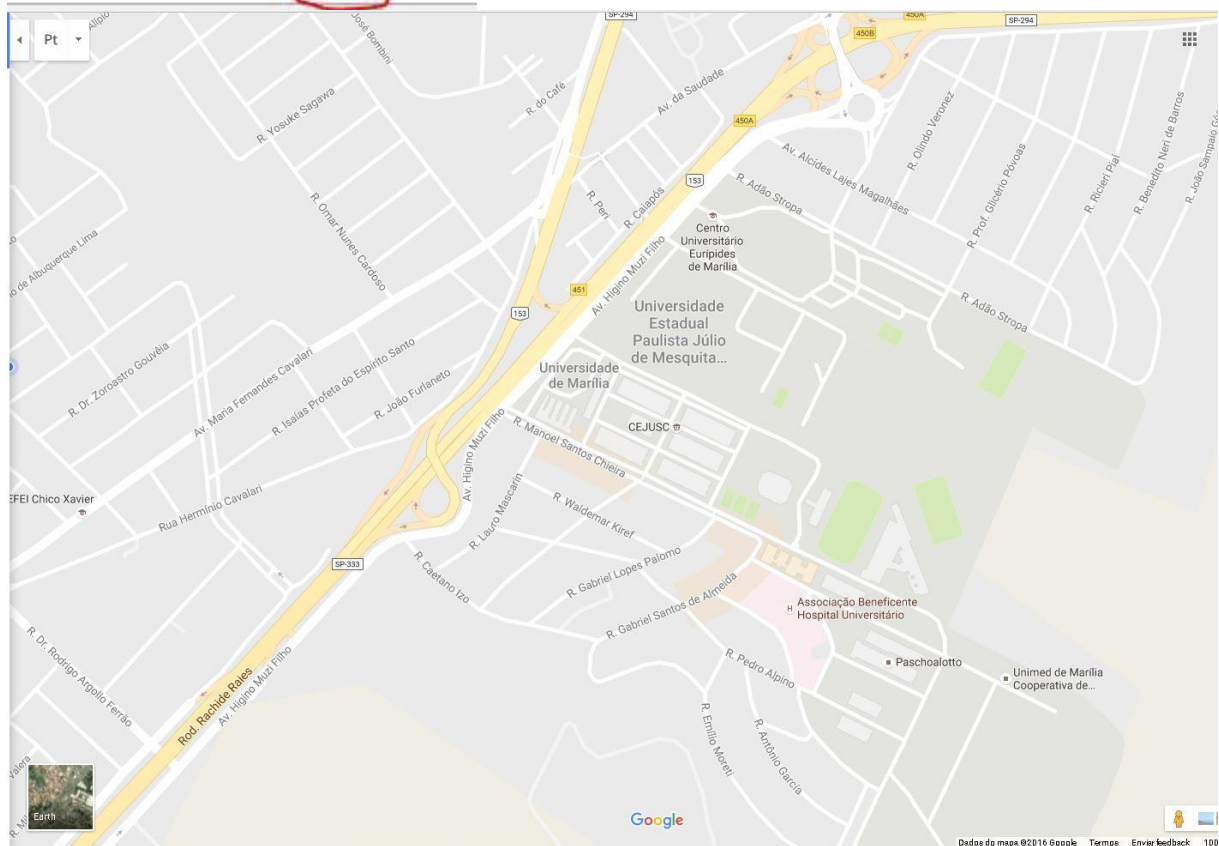


Figura 27 – Parâmetros de Coordenadas Google Maps

Foi possível entender que os mapas da terceira coluna possuem um valor maior de longitude em relação aos mapas da coluna central, e estes maior que os mapas da primeira coluna. Do mesmo modo, os mapas referentes à primeira linha possuem um valor maior de latitude do que os mapas da linha central, e estes da linha inferior. Para isto, é necessário que os mapas interajam uns com os outros, buscando dados de coordenadas e realizando cálculos de soma ou subtração para gerar novos valores para cada mapa.

A **figura 28** exemplifica melhor essa ideia, mostrando o mapa panorâmico já implementado na cena do Unity no projeto e os valores referentes às alterações de coordenadas de um mapa à outro. Por exemplo, o **Mapa 3** possui uma alteração de 0.0055 somada em sua longitude relativamente ao **Mapa 2**, e este 0.0055 relativamente ao **Mapa 1**, enquanto que o **Mapa 9** possui uma alteração de 0.0049 subtraída em sua latitude em comparação ao **Mapa 6**, e este igualmente ao **Mapa 3**.

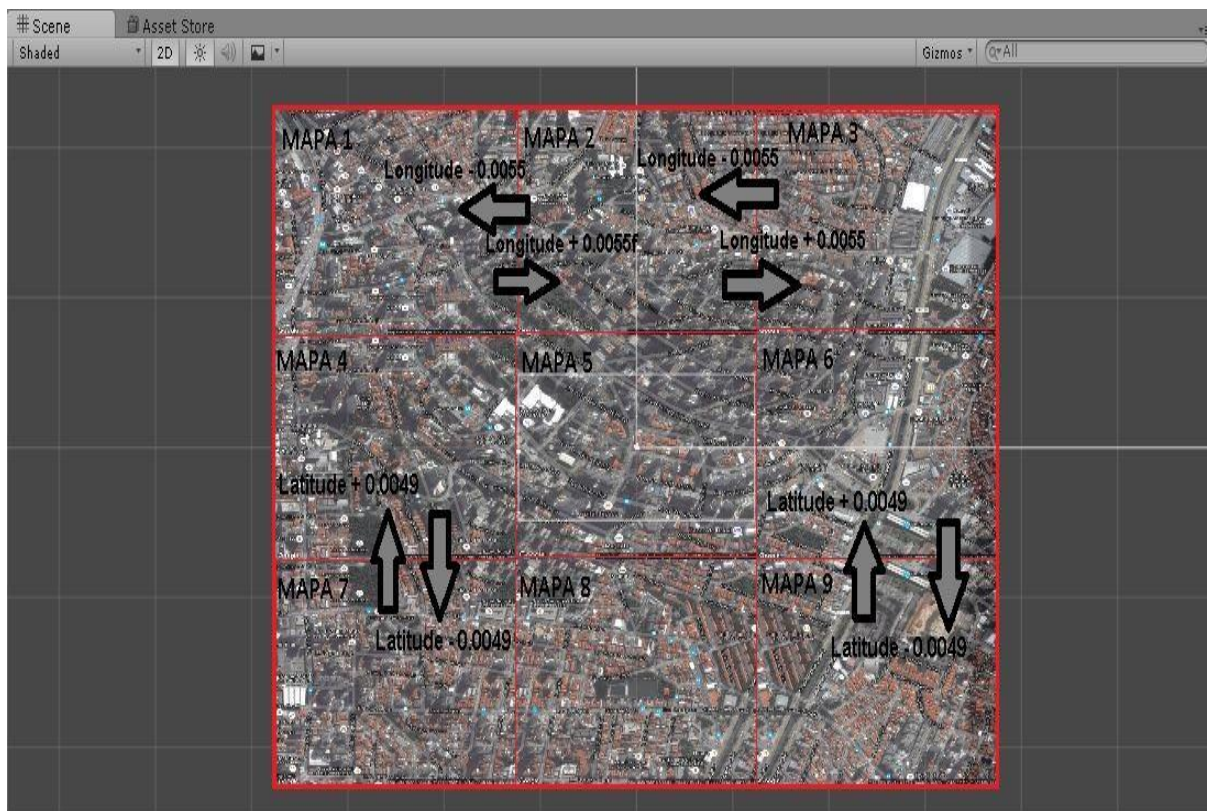


Figura 28 - Cálculo Coordenadas Mapas

No próximo subcapítulo será abordada a implementação da movimentação dos mapas, possibilitando assim, a interação do usuário ao sistema realizando a navegação ao mapa panorâmico.

4.1.2 – Movimentação dos Mapas

O primeiro passo para realizar a movimentação entre os mapas foi implementar um *script* associado ao Mapa Principal (5) para que seja possível identificar qual a direção para a qual o usuário está navegando naquele momento (direita, esquerda, cima ou baixo). Para isso ser possível, foi preciso utilizar como base o raio de visão da câmera principal com projeção Ortográfica direcionada ao centro do mapa panorâmico (mapa 5). Para obtermos o campo de visão atual a que o usuário esteja visualizando foi preciso também utilizar os dados referentes à posição do mapa panorâmico, para que seja possível identificar o momento em que os mapas precisem ser realocados, dando continuidade à navegação do usuário.

Assim, quando o campo de visão da câmera do usuário estiver chegando ao fim de algum dos mapas adjacentes ao mapa central, realiza-se a realocação destes mapas, onde os 3 mapas mais distantes são descartados, os 3 centrais são afastados e 3 novos são chamados para preencher o caminho da navegação.

Esta navegação foi implementada através de uma função em um script associado ao objeto “Mapa_Principal”. De início as setas do teclado do computador foram utilizadas como entrada de dados (*Input*) para realizar a movimentação do mapa panorâmico, porém, no projeto final estes *Inputs* foram substituídos por gestos e movimentos captados e interpretados por um dispositivo de sensor de movimento acoplado ao *Óculus Rift*. A implementação destes gestos de movimentos captados ao sistema pertencem a um trabalho paralelo de um aluno do grupo do mesmo projeto. No caso desse projeto foi utilizado o “Leap Motion”, e pelo fato deste ser um dispositivo pequeno foi possível encaixar o mesmo na parte da frente do *Óculus Rift* como mostra a **figura 29**;



Figura 29 – Dispositivo Leap Motion acoplado ao *Óculus Rift*

Ao utilizar as setas do teclado ou gesticular no *Leap*, os valores da posição referente ao objeto “Mapa_Principal” são modificados de acordo com o sentido que o usuário esteja navegando, ou seja, caso o usuário esteja navegando no sentido horizontal, estes valores são alterados em relação ao eixo ‘x’, sendo que para esquerda esse valor é subtraído, e para a direita esse valor é somado.

O objeto câmera permanece fixo na posição central (0,0), com o raio da câmera de visão fixo no mapa central, assim, quando uma navegação é realizada horizontalmente no sentido à direita, por exemplo, quando este raio de visão estiver quase atingindo o final de algum dos mapas da última coluna, o usuário estará próximo de visualizar uma cena vazia do

projeto. Como mostra na **figura 30**, onde chegamos ao momento de realocar os mapas e prosseguir para a navegação do usuário.



Figura 30 – Usuário visualizando uma cena vazia do mapa

Foi definido um valor limite referente à posição do Mapa_Principal, para assim ser possível identificar o exato momento em que é preciso realizar um reposicionamento dos mapas, definido ao observar atentamente o raio de visão da câmera nos quatro sentidos possíveis de navegação. Como vemos na (**Figura 31**) e (**Figura 32**).

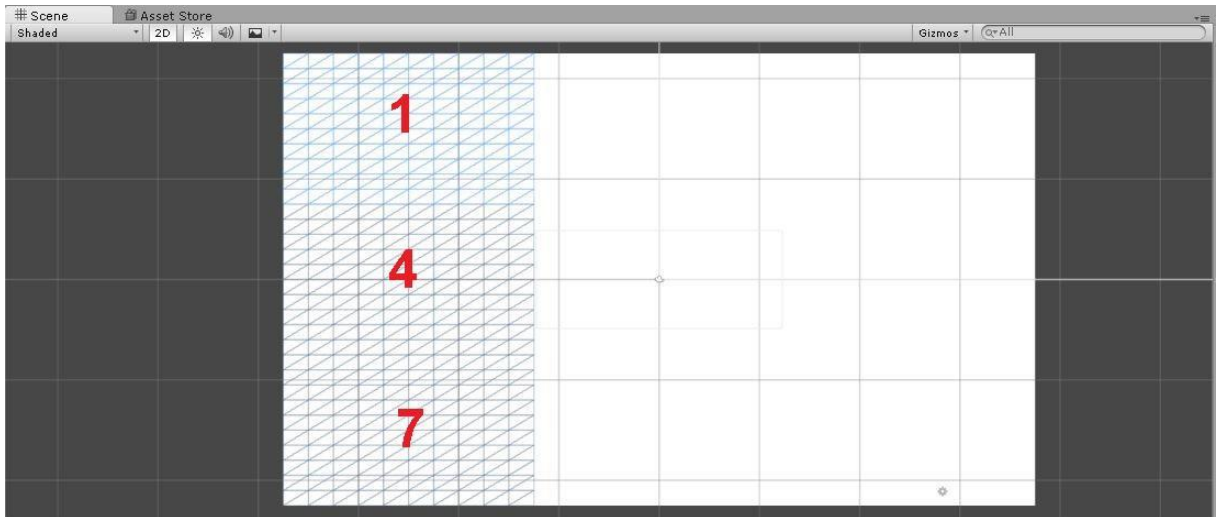


Figura 31 - *GameObjects* mapa (1), mapa (4), mapa (7) referentes à primeira coluna do Mapa Principal.

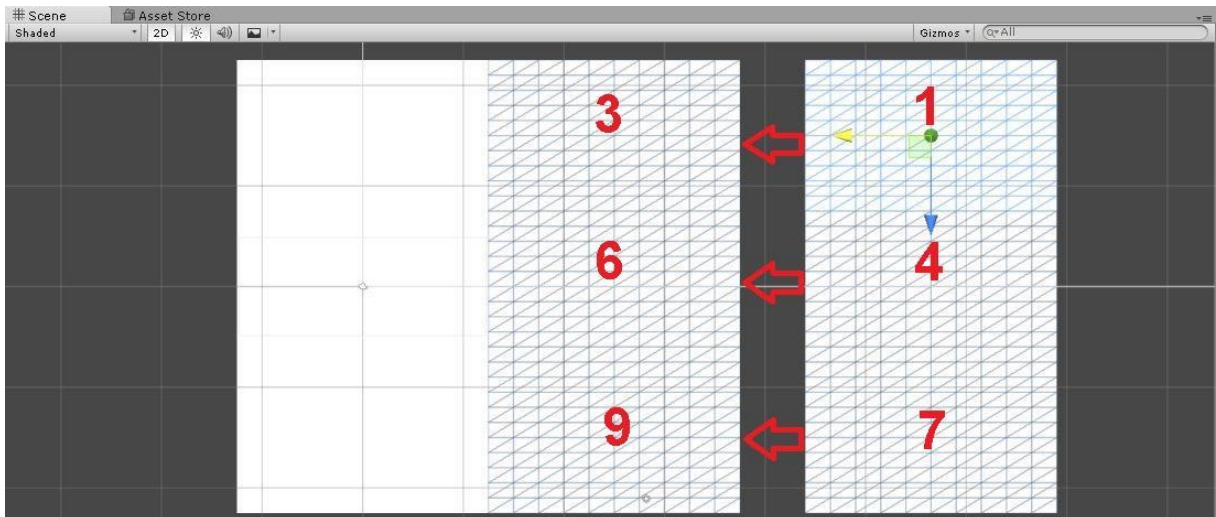


Figura 32 - Encaixando os mapas da primeira coluna à última coluna

Dessa forma é possível dar continuidade ao mapa visualizando novos pontos a partir de novas informações (latitude, longitude) enviadas ao servidor do *GoogleMaps* e retornadas como imagens.

Ao realizar um reposicionamento dos mapas, é alterado também o valor da posição destes mapas em uma variável do tipo *matriz*, que tem a função de armazenar o posicionamento correto dos nove mapas do mapa panorâmico. Assim, é possível ter um controle de cada mapa individualmente, como também identificar os mapas que precisam ser reposicionados a cada navegação do usuário. Desta forma a **figura 33** demonstra como os

mapas ficam reorganizados após uma navegação a direita do mapa. Por fim a navegação para o sentido vertical funciona da mesma forma, mudando apenas os mapas que serão reposicionados. No caso de uma navegação para cima, os mapas a serem reposicionados são os mapas referente a última linha.

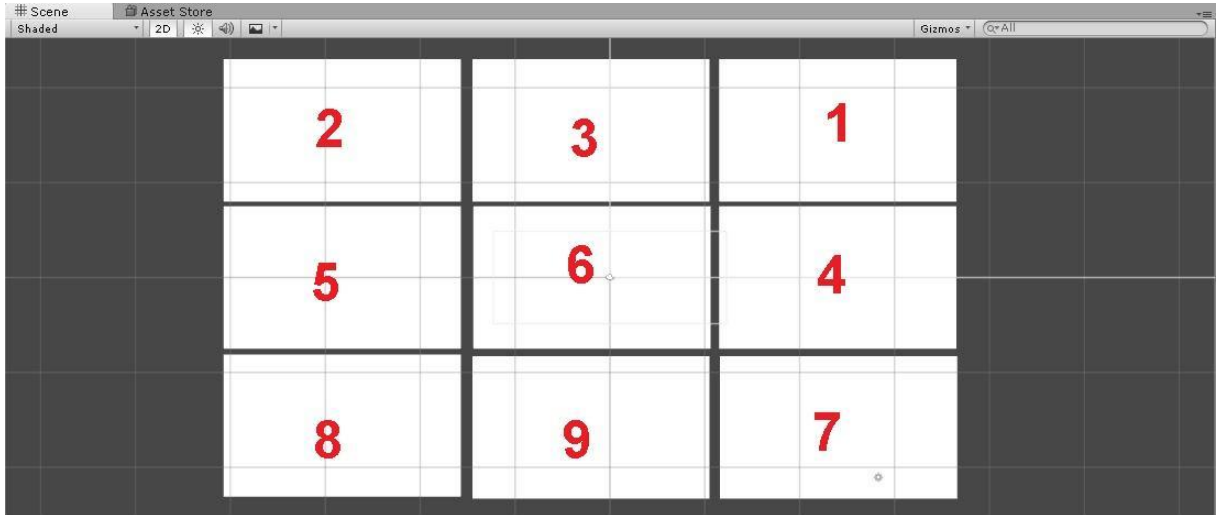


Figura 33 - Mapa Panorâmico após uma modificação de mapas à direita

4.1.3 – Buscando Coordenadas em determinados Pontos do Mapa

A próxima etapa implementada foi trazer informações referentes à latitude e longitude do Mapa Panorâmico a partir de qualquer ponto. Inicialmente foi utilizado o mouse para realizar esta função. O objetivo desta etapa foi tornar possível que o usuário interaja com o sistema.

O primeiro passo foi estudar como funciona o plano cartesiano no Unity3D. O sistema de coordenadas da tela do computador no Unity trabalha de uma maneira diferente, sendo que o ponto inicial (0,0) em relação aos eixos x e y da tela são localizados no canto inferior esquerdo da cena do jogo, e a altura e largura máxima da tela do computador são localizadas no canto superior direito da cena do jogo, como mostra na **figura 34**.



Figura 34 - Ponto inicial e máximo no sistema de coordenadas da tela no Unity

Foram realizados alguns testes referentes a determinados pontos clicados na cena do projeto, a partir da resolução da tela de um notebook. Para tal foi criado uma variável denominada *mousePos* do tipo Vetor2D (Vector2) próprio do unity, o qual inicia seus valores recebendo as coordenadas atuais em pixel do mouse a cada *frame*, através do uso de uma classe denominada *Input.mousePosition*. Assim, quando o usuário clicar na tela é chamado uma função padrão do Unity da classe *Input* que identifica que houve um evento de clique do mouse. Serão então retornadas no console do Unity as coordenadas x e y referentes à variável *mousePos*.

Para que o ponto central das coordenadas da tela se tornem o valor (0,0) quando realizado um clique do mouse ao meio da tela, foi preciso utilizar as variáveis “width” e “height” padrão da classe *Screen* do Unity, cuja função é buscar a largura e altura máxima referente ao monitor, e através de um cálculo matemático que compara a posição atual do mouse em relação ao eixo das coordenadas centrais da tela.

Com base nesses cálculos foi possível identificar o exato ponto da tela clicado pelo usuário referente ao Mapa Panorâmico, porém o objetivo desta etapa do projeto é retornar as informações do mapa referente à latitude e longitude em um determinado ponto clicado pelo mouse, sendo assim, o próximo passo então foi desenvolver esta função.

Cada *GameObject* mapa possui seus dados de latitude e longitude, os quais foram solicitados e retornados ao servidor do *Google Maps* para o preenchimento das texturas destes objetos. Estes valores se referem ao ponto central destes mapas, ou seja, retorna no meio da tela (0,0).

Para isto, foi utilizado como base uma variável responsável por determinar um limite de distância do ponto central do mapa em relação ao mouse, e toda vez que o valor deste limite for atingido, um cálculo somatório ou de subtração da latitude/longitude do mapa será executado. Subsequentemente, sempre que o usuário clicar em um ponto da tela, tendo comparadas as distância do ponto do mouse até o centro do mapa, é possível identificar quantos cálculos de soma ou subtração precisam ser realizados para retornar as coordenadas do novo ponto. Como vemos na **figura 35**.

Foram utilizados como base para esses valores e distâncias, os quadrados referentes à cena do Unity3D. No exemplo abaixo, estamos apenas utilizando o conceito para o eixo 'x'. O usuário clicou em um ponto à direita do mapa, sendo assim, o cálculo de longitude deve ser somado com a longitude central do mapa.

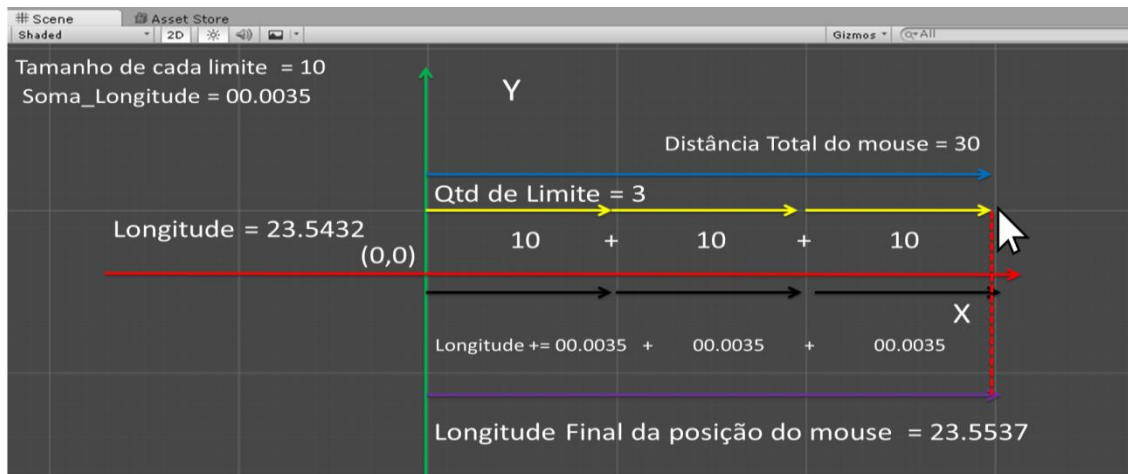


Figura 35 - Cálculo matemático para se obter o resultado de uma nova coordenada

O cálculo da **figura 35** está associado a um *script* relacionado ao objeto “Mapa_Principal” sendo executado a cada quadro da aplicação. Com o simples fato do usuário passar o mouse em um determinado ponto do mapa, as coordenadas referentes àquele ponto são retornadas em tempo real. Quando o usuário clicar em um determinado ponto do mapa, as informações referentes a estas coordenadas também são armazenadas em uma caixa (Box), implementada no canto superior esquerdo do mapa, como mostra na **figura 36**

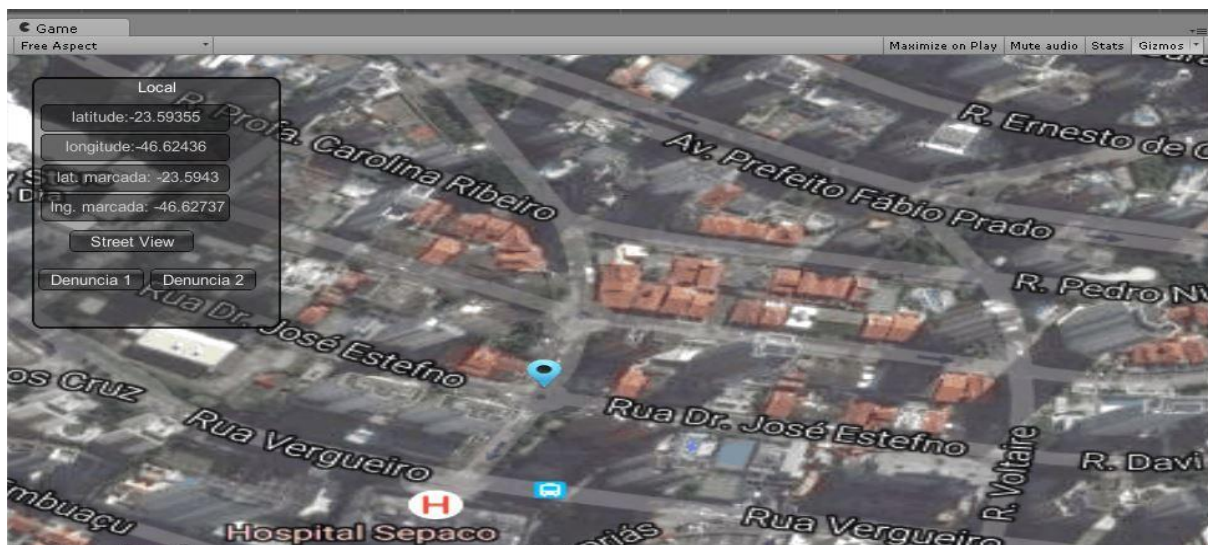


Figura 36 - Ponto clicado no mouse pelo usuário e retornado as informações de latitude e longitude na caixa (Box) no canto superior esquerdo ao mapa

As duas primeiras caixas de texto referem-se às coordenadas em tempo real no momento em que o usuário está passando o mouse sobre um ponto do mapa, e as duas caixas de texto abaixo se referem às coordenadas do ponto em que o usuário clicou com o mouse no mapa.

O próximo subcapítulo irá descrever o processo da busca das informações da situação do crime, realizando requisições ao sistema ESAS desenvolvido pelo grupo de IHC do COMPSI-UNIVEM.

4.1.4 – Buscando Informações das Denúncias através de Pacotes JSON

O próximo passo do projeto foi realizar a busca das informações das diversas denúncias realizadas pelas testemunhas através das ligações ao 190 ou por postagens em redes sociais. Estas informações contém dados como: número do celular da testemunha, a categoria da denúncia, o horário da denúncia, a localização, a mensagem da denúncia, etc.

Cada denúncia é armazenada no sistema ESAS, um sistema de visualização de informações desenvolvido pelos alunos do grupo IHC (Interação-Humano-Computador) do laboratório de Pesquisa COMPSI – UNIVEM – MARÍLIA, juntamente com o professor Leonardo Botega, e o grupo de IHC da faculdade de São Carlos (UFScar), trabalho vinculado à polícia militar do estado de São Paulo (PMESP) .

A **figura 37** demonstra a tabela de eventos deste sistema, onde todas as informações referentes às denúncias das testemunhas ficam armazenadas e organizadas. É possível

observar que a segunda denúncia, por possuir mais informações referentes à situação, possui um valor de certeza maior referente a primeira denúncia.

Tabela de Eventos						
	Origem	Tipo Situação	Situação	Hora	Certeza da Situação	
-	190	DESCONHECIDA	Adicionado C e L	14:04:57	35.69	
Mais Informações: acaba de acontecer um crime aqui na domingos setti um motorista foi ameaçado e mandaram ele sair do veículo sem levar nada o bandido fugiu em direção ao metrô klabin						
-	190	ROUBO	Atualizado C e L. Adicionado V	14:05:45	43.8	
Mais Informações: um cara foi roubado aqui na minha frente por alguém armado foi tudo muito rápido foi na domingos setti perto do restaurante don paladino o bandido tinha um revolver era alto e tinha tatuagens nos braços ele deu uma coronhada no homem e ele parece estar bem machucado						

Figura 37 - Interface Sistema de Denúncias

Os dados referidos na **figura 37** representam apenas informações relacionadas à denúncia de cada testemunha, com base nas mensagens de cada denúncia, informações referentes à descrição do criminoso, como: altura, tatuagem, cor do cabelo, e outras informações como o local para o qual o criminoso esteja fugindo, entre outras informações relacionadas a um determinado crime, são armazenadas separadamente ao sistema, sendo assim, os dados referentes a qualquer entidade são atualizados assim que uma nova denúncia conter novas instâncias ainda não armazenadas na entidade em questão.

Todas estas informações estão no formato JSON, assim é possível realizar a busca dessas informações fazendo uma requisição ao servidor, que nos retorna um pacote JSON contendo todas as informações necessárias. A **figura 38** demonstra estes dados em formato JSON.

```

{
  "id": "1",
  "fonte": "190",
  "horainicio": "2015-05-29 14:03:27",
  "horaFim": "2015-05-29 14:04:57",
  "gps": ["-23.593611", "-46.627527"],
  "origemMensagem": "14 4335-2323",
  "eventos": "Adicionado C e L",

  "tempo": {
    "quantidade": "0",
    "tipo": "2015-05-29 14:03:49"
  },

  "criminoso": {
    "sexo": "masculino",
    "status": "fugindo",
    "fugaTipo": null,
    "fugaPara": "metrô klabin",
    "descricao": null,
    "idade": null,
    "altura": null,
    "condicao": null,
    "etnia": null,
    "porteFisico": null,
    "corCabelo": null,
    "tipoCabelo": null,
    "vestimenta": null,
    "marcaVestimenta": null,
    "arma": null,
    "tatuagem": null,
    "palavraRef": null,
    "completude": 27.72,
  },

  "objeto": {
    "descricao": null,
    "condicao": null,
    "marca": null,
    "modelo": null,
    "tamanho": null,
    "cor": null,
    "ano": null,
    "placa": null,
    "porte": null,
    "completude": 10,
  },

  "vitima": {
    "palavraRef": null,
    "sexo": null,
    "status": null,
    "idade": null,
    "altura": null,
    "condicao": null,
    "etnia": null,
    "porteFisico": null,
    "corCabelo": null,
    "tipoCabelo": null,
    "corVestimenta": null,
    "marcaVestimenta": null,
    "completude": 10,
  },

  "local": {
    "logradouro": "domingos setti",
    "tipoLogradouro": null,
    "complemento": null,
    "bairro": null,
    "altura": null,
    "direcao": null,
    "sentido": null,
    "referencia": null,
    "numero": null,
    "gps": ["-23.593680", "-46.627397"],
    "completude": 30.76,
    "certeza": 35.69,
    "completudeTemporal": 100,
    "eventos": ["addCriminoso", "addLocal"],
    "ignorarFlag": false,
    "ignorarMensagem": null
  }
}

```

Figura 38 – Dados em Formato JSON armazenados no sistema ESAS

Fonte: Modelo de Fusão Dirigido por Humanos e Ciente de Qualidade de Informação- UFSCAR – São Carlos - (Leonardo C. Botega).

É possível observar na **figura 38** que cada instância refere-se a uma determinada classe JSON separada, e possui seus próprios atributos. Por fim, todas elas pertencem a uma mesma classe denominada “situação”, não demonstrada na figura acima. Todas as denúncias pertencem a uma mesma classe, sendo assim, possuem os mesmos atributos, e o que diferirá uma da outra são os valores contidos em cada um deles para cada denúncia, como a sua mensagem, origem, fonte, entre outros.

Todas as denúncias estão acopladas a um array de Objetos da classe *Denúncia*, sendo assim, é possível buscar cada denúncia específica procurando pelo seu atributo identificador ‘id’, como também por uma determinada posição referente a este *array*.

O próximo passo foi criar uma classe. No caso deste trabalho foi utilizada a linguagem C# para cada entidade, contendo os mesmos atributos pertencentes das entidades. A requisição ao servidor retorna um Pacote JSON contendo todas estas informações, assim, uma variável do tipo *string* foi criada para armazenar estas informações.

Após termos uma variável com todos os dados que precisamos, o próximo passo foi separar essas informações, de tal forma a armazená-las corretamente para suas devidas classes C#.

Para separar todas estas informações foi criada uma classe C# chamada JSON, esta classe é responsável por realizar a requisição do pacote JSON ao sistema ESAS. Feito isso, os dados são armazenados na variável *string*, e em seguida utiliza-se métodos de *manipulação de strings* para que seja possível dividir estas informações em vários “pedaços” e armazená-los em variáveis *strings* separadas para cada entidade. Os métodos utilizados de *manipulação de string* incluem: busca por índices de palavras, copiar uma determinada parte de uma string a partir de um índice, etc.

O índice é um valor inteiro referente à posição que uma determinada palavra se encontra em uma string, após encontrar e armazenar o índice da instância “situação” busca-se o próximo índice, referente à próxima palavra (entidade) desejada, no caso, “criminoso”. Porém, como precisamos obter apenas os atributos referentes às entidades, foi utilizada uma outra lógica na *manipulação de strings* para copiar os dados a partir do caracter ‘{’ e finalizando antes do próximo caracter ‘}’. Foram realizadas estas etapas para buscar as informações de todas as entidades, e como resultado obtivemos uma variável do tipo *string* para cada uma delas, contendo seus devidos dados (atributos) retornados e desempacotados do pacote JSON.

Por fim foi utilizado um método próprio do Unity responsável pela conversão de strings em objetos JSON e enviados os dados para suas devidas classes C#. A **figura 39** exemplifica todas as etapas deste processo.

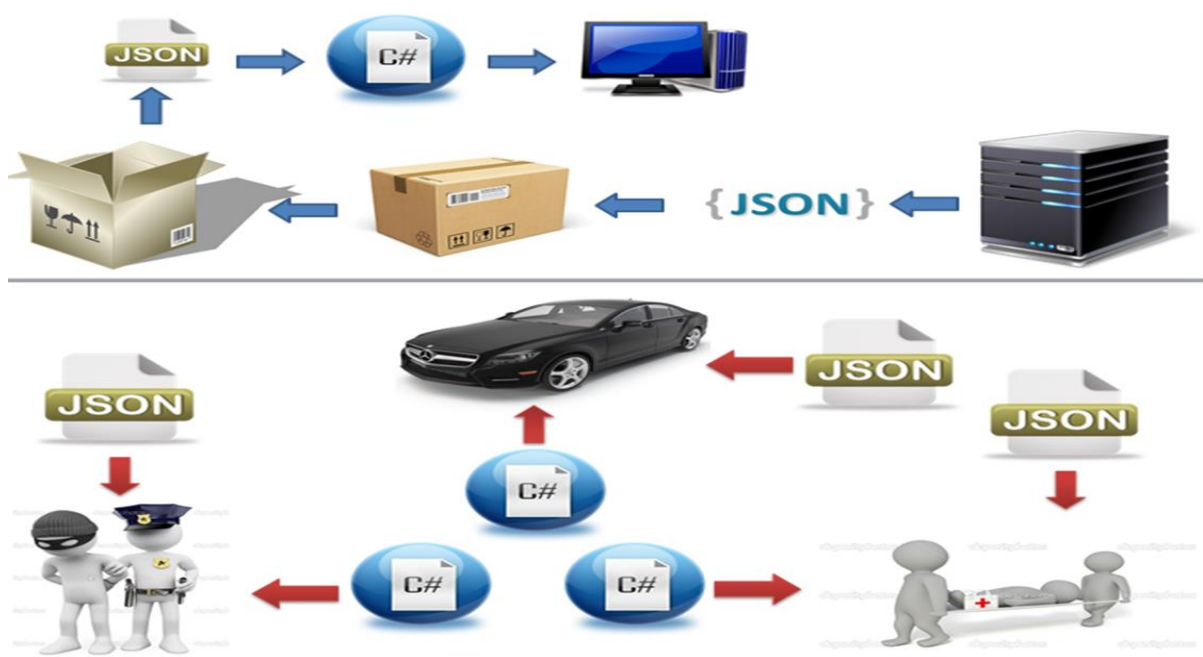


Figura 39 – Etapas para a busca das informações ao sistema ESAS

Já para a instância de denúncias, como são várias denúncias e a todo o momento chegam novas denúncias, utiliza-se a mesma lógica de *manipulação de strings*, mudando apenas a forma como os dados serão armazenados, já que cada denúncia irá ocupar um espaço em uma variável do tipo *array* da classe “Denúncia”.

4.1.5 – Instanciando as denúncias no mapa panorâmico

O próximo passo foi instanciar no mapa panorâmico os ícones referentes a cada denúncia realizada. Para tal foram utilizados dois objetos da Unity no formato de Cubo (Cube), em que cada um desses objetos contém uma imagem referente à fonte de cada denúncia. Ou seja, caso a denúncia feita pela vítima seja através do 190 é instanciado no mapa o objeto com um ícone de celular, caso seja por rede social, um ícone com uma imagem representando a rede social, como mostra a **figura 40**.

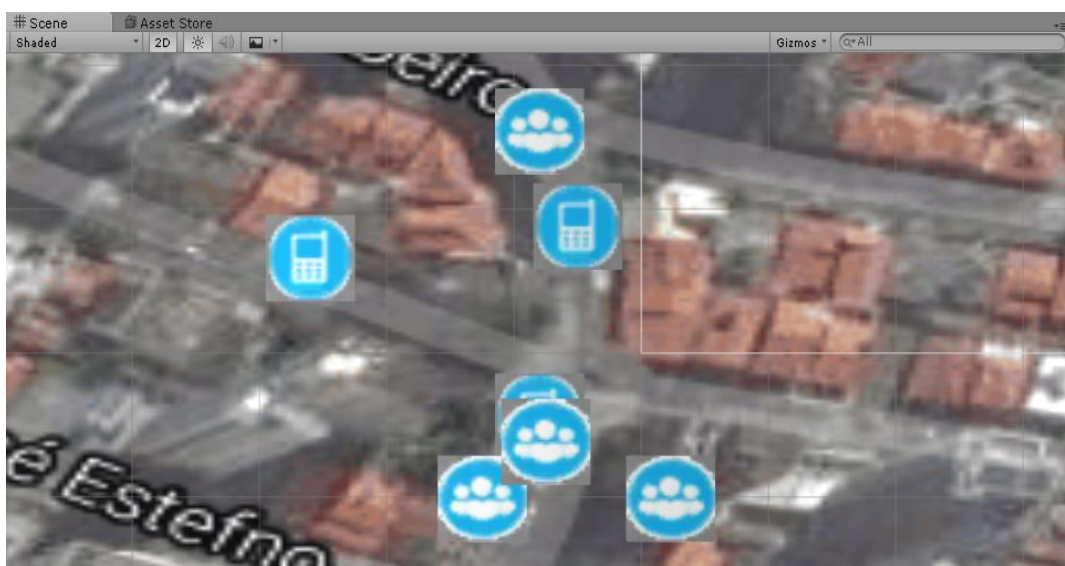


Figura 40 – Denúncias instanciadas no Mapa Panorâmico

Como é possível observar na figura acima, cada denúncia está posicionado em um determinado ponto no mapa representando o local em que cada denúncia foi realizada, para isso ser possível foi utilizado as coordenadas da tela do monitor com base nas informações referente à latitude e longitude de cada denúncia. Para ter acesso a todas as informações referente a cada denúncia foi implementado uma função na classe responsável por conter os dados pertencentes ao pacote JSON, no qual o objetivo é enviar todas as informações necessárias de cada denúncia através de um laço de repetição.

O acesso a informações referente à localização de cada denúncia foi realizado através do atributo “gps”, referente à classe “Denuncias” contido em cada denúncia, outra informação necessária a ser lida desta classe, é referente ao atributo “fonte” também pertencente à classe de denúncias como já mostrado na **figura 38**, assim é possível identificar qual a fonte de cada denúncia, seja por via ‘190’ ou por ‘post’, para que assim seja instanciado ao mapa o objeto referente a esta fonte. A **figura 41** demonstra o fluxograma deste processo.

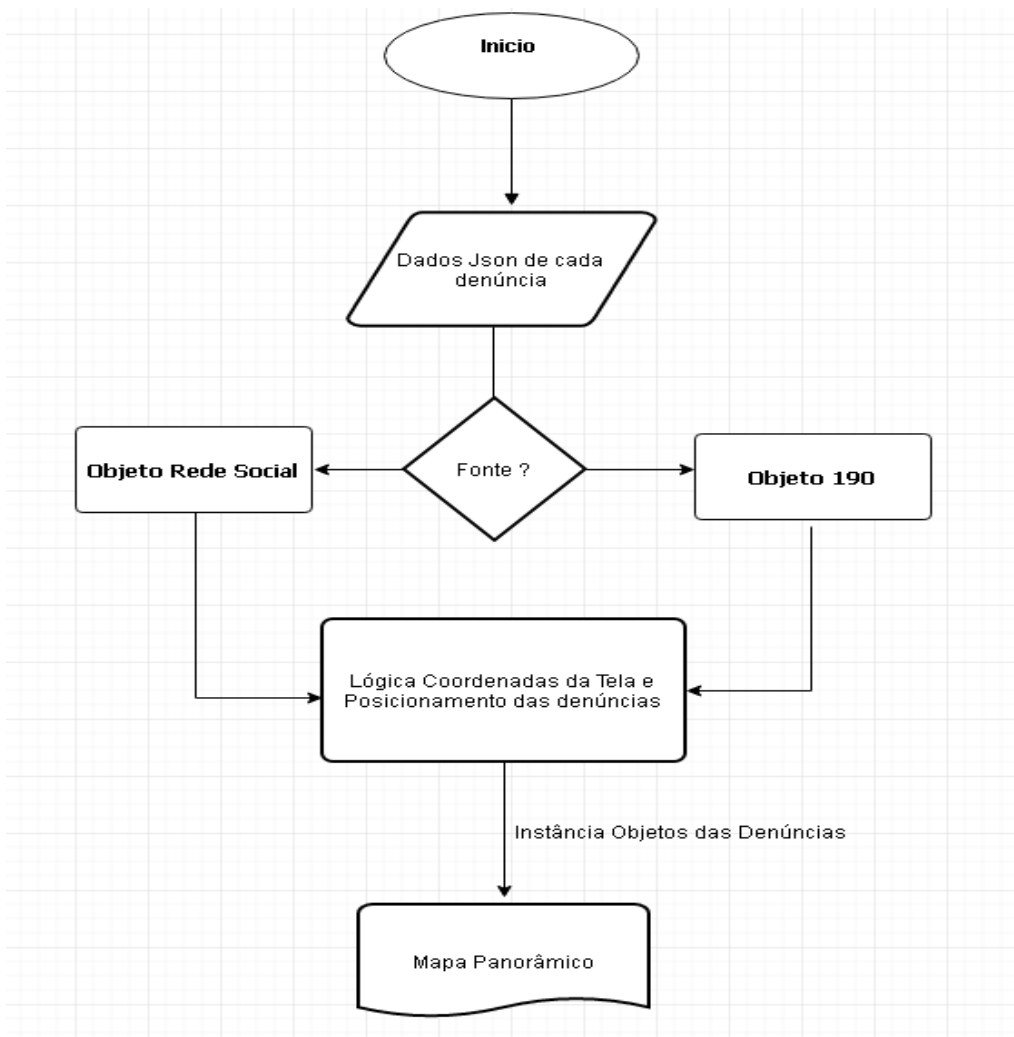


Figura 41 – Fluxograma do processo de instanciação das denúncias

O próximo passo foi buscar as informações referentes à mensagem da vítima para cada denúncia, e associá-las aos objetos instanciados no mapa panorâmico. Para isso foi preciso criar um script denominado “MensagemDenúncia”, onde sua função é receber os dados referentes à mensagem de cada denúncia por parâmetro, através de um método implementado e chamado por outra classe, classe esta que possui os dados referentes ao pacote JSON retornado do servidor.

Essa informação referente à mensagem da denúncia é obtida através do atributo “texto” referente à classe “Denuncias”, outra informação a ser enviada por parâmetro para este método é a informação referente à origem da mensagem. Ou seja, o número do celular da vítima, como também o seu perfil de rede social, caso a fonte da mensagem seja por postagem - esta informação é possível buscar através do atributo “origemMensagem” também referente à classe “Denuncias”.

Por fim, estas informações são armazenadas em um campo de texto de um componente próprio do Unity chamado “Text Mesh”, este componente associado a cada objeto de denúncia possui algumas propriedades, tais como: Text, alinhamento do texto, a cor do texto, o estilo da fonte, a fonte, o tamanho do texto entre outros.

Por último foi implementado no script “MensagemDenúncia” associado a cada denúncia, uma função que identifica o *input* do usuário sobre o objeto denúncia instanciado no mapa, onde vemos a mensagem referente a cada denúncia. Isto é possível ativando e desativando o componente “Text Mesh” associado a cada denúncia, como mostra na **figura 42**.



Figura 42 - Visualização de mensagem no mapa panorâmico referente às denúncias das vítimas

4.2 – Implementação do StreetView

Após a implementação da visualização dos mapas, da navegação entre os mapas, e da interação do usuário aos pontos de coordenadas do mapa, o próximo passo foi possibilitar ao usuário navegar pelo modo *streetView* em qualquer ponto clicado pelo mouse.

Para este projeto foi utilizado uma API do *streetView*. Ele funciona do mesmo modo que o GoogleMaps do mapa panorâmico, fazendo requisições ao servidor do Google StreetView enviando os parâmetros que seguem descritos:

Size – Especifica o tamanho da saída da imagem em pixels. Este tamanho é especificado como {width} x {height}; por exemplo, size = 600x400 retorna uma imagem com 600 pixels wide, e 400 high.

Location – a localização (Chagrin Falls, OH) pode ser enviada tanto em texto como em valores de latitude e longitude (-23.55532, -49.32454), assim a API do streetView irá buscar e encaixar um panorama mais próximo desse local.

Heading – Indica a direção da bússola da câmera. Os valores aceitos são de 0 a 360° (ambos os valores indicam Norte, com 90 indicando Leste, e 180 Sul).

Fov - Determina o campo de visão horizontal da imagem. O campo de visão é expresso em graus, com um valor máximo permitido de 120°. Ao lidar com uma viewport de tamanho fixo, como uma imagem Street View de um tamanho de conjunto, o campo de visão em essência representa zoom, e números menores indicam um maior nível de zoom.

Pitch - Especifica o ângulo para cima ou para baixo da câmera em relação ao veículo Street View. Valores positivos representam o ângulo da câmera para cima (90°, apontando para cima) e valores negativos representam o ângulo da câmera para baixo (-90°, apontando para baixo).

A **figura 43** mostra esses parâmetros em um navegador. Como resposta da requisição ao servidor do StreetView, uma imagem estática do local é retornada ao navegador.

<https://maps.googleapis.com/maps/api/streetview?size=600x300&location=-22.2259334,-49.946514&heading=151.78&pitch=-0.76>

Figura 43 - Parâmetros no navegador fazendo uma requisição ao servidor StreetView

Para entender o funcionamento do projeto é preciso explicar como funciona a classe API do streetView utilizada. Esta classe utiliza uma variável do tipo *Lista* de uma classe chamada *Direction*, que tem como principal função armazenar os valores referentes às direções da câmera para cada imagem fotografada pelo veículo do streetView. Sendo assim, ela possui três variáveis: *Name* do tipo *string*, representando nome da direção da câmera, a variável *heading* do tipo *double*, e a variável *pitch* também do tipo *double*. Assim, esta variável do tipo *List*, ao ser instanciada deve conter seis tipos de direções: Frente (heading=0, pitch=0), trás (heading=180, pitch=0), esquerda (heading=90, pitch=0), direita (heading=270, pitch=0), cima (heading=0, pitch=90) e baixo (heading=0, pitch=-90).

Quando o usuário clica em um determinado ponto do mapa e clica no botão “StreetView” do box, localizado no canto superior esquerdo da tela, os valores referentes à latitude e longitude armazenados nas duas últimas caixas de texto do box são enviadas a uma função desta classe do *StreetView*. Em seguida, uma pasta denominada *textures* é criada na pasta padrão *Assets* do Unity, que tem como função armazenar todas as seis imagens que serão retornadas da requisição ao servidor do StreetView daquele exato ponto clicado pelo usuário no mapa.

Para que seja possível retornar as imagens referentes às seis direções, é utilizada a variável *List<Direction>* em uma função de repetição, realizando as requisições ao servidor do StreetView e enviando por parâmetro os valores referentes ao *pitch* e *heading*, até que todas as direções tenham sido requisitadas e retornadas em imagens.

Esta API trabalha com um tipo de *shader* diferente chamado *skybox/6sided*, que define que um determinado *material* deve conter seis texturas diferentes, uma para cada face de um cubo. A **figura 44** mostrará o resultado final do uso do shader após uma requisição do streetview, utilizando para cada lado do skybox, as imagens retornadas do local. Por formar um skybox, a sensação do usuário é de uma visão panorâmica do local ao movimentar-se para todas as direções.

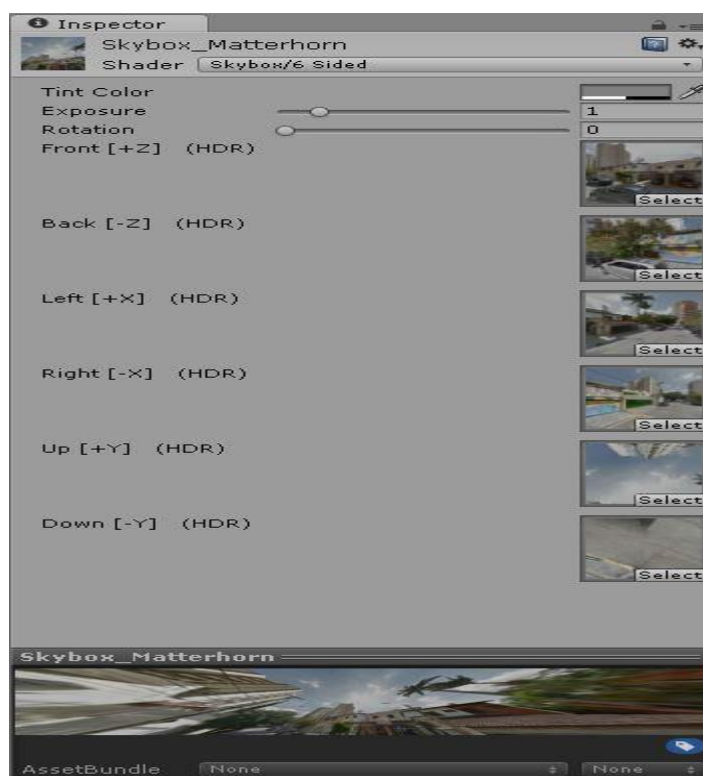


Figura. 44 – Material contendo um *shader* do tipo SkyBox



Figura 45 - Representação das texturas encaixadas no material neste tipo de shader representando um quadrado

É possível observar pela **figura 45**, as linhas representando as texturas encaixadas no material neste tipo de *shader*. A **figura 46** demonstra um usuário realizando gestos de movimentos para ter acesso ao modo street view e ao mapa panorâmico.

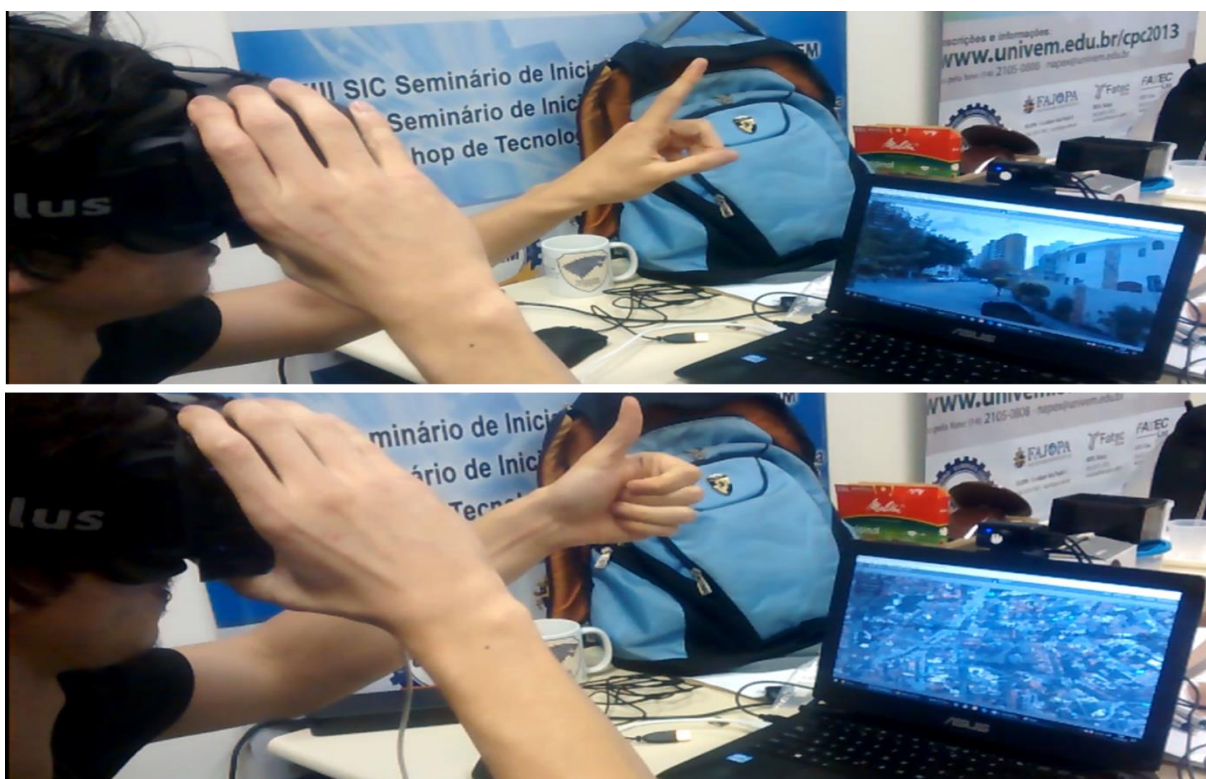


Figura 46 – Usuário utilizando gestos para navegar ao sistema

4.2.1 – Instanciando Modelos Tridimensionais na Cena do Crime

A última etapa do projeto foi instanciar modelos tridimensionais referentes a cada instância da situação, como: criminoso, vítima, objeto, local, estes modelos foram estrategicamente instanciados e posicionados no street view no exato local em que ocorreu a situação. Para isso foi preciso primeiramente buscar modelos 3D que seriam utilizados no projeto. Inicialmente foram utilizados sites que disponibilizam estes modelos gratuitamente, neste caso o site 3dwarehouse. sketchup.com.

Existem vários formatos para modelos 3D, como: .blender, .sketchup, kmz, .dae, .3ds, .obj, entre outros. Neste projeto, por ter sido utilizado o site do sketchup para realizar o download dos modelos 3D, o formato destes modelos por padrão são em .skp, que foram exportados para o formato .dae para o Unity3D. A **figura 47** mostra os modelos 3D utilizados no projeto.



Figura 47 - Modelos tridimensionais utilizados no projeto instanciados em uma cena Vazia.

Ao importar os modelos 3D para o unity, estes ficaram em uma escala muito grande para a cena do projeto, então foi preciso modificar a escala dos eixos de cada modelo para a cena do projeto. Foi preciso importar também as texturas de cada modelo 3D, feitas de várias imagens que compõem cada parte do modelo 3D, para que o mesmo não fique um modelo 3D cinza.

O próximo passo foi buscar os dados referentes à localização da situação. Para isso foi utilizado como base os dados retornados do pacote JSON. A classe que contém os dados JSON é responsável por enviar estes dados à classe associada ao mapa panorâmico, informações estas acessíveis a partir de um método implementado na classe referente à entidade do local, onde por fim retorna o valor referente ao atributo “gps”, que contém a latitude e longitude do local da cena do crime.

Agora com acesso e visão deste local através do modo *street view*, em seguida foi realizado a instanciação dos modelos 3D ao *street view*, para o quê foi preciso criar na classe

associada ao mapa panorâmico variáveis do tipo Objeto (Object) e associá-las para cada modelo.

Os modelos tridimensionais criminoso e vitima objeto são posicionados de forma automática após o usuário acessar o modo *street view* do local da situação. Foi preciso modificar os valores referentes aos eixos de cada modelo 3D para assim posicioná-los de forma estratégica ao ambiente virtual, aumentando o nível de imersão do usuário ao visualizá-los ao seu redor utilizando o dispositivo *Óculus Rift*.

O último passo do projeto foi associar a cada modelo 3D suas devidas informações, referentes aos dados retornados do pacote JSON do sistema ESAS, do mesmo modo que foi feito para cada objeto denúncia instanciado no mapa panorâmico. Foi criada uma classe C# para cada modelo 3D, onde foi implementado um método responsável por receber estas informações e atribuir a um campo de texto um componente associado a cada modelo 3D. A **figura 48** mostra o fluxograma destas etapas.

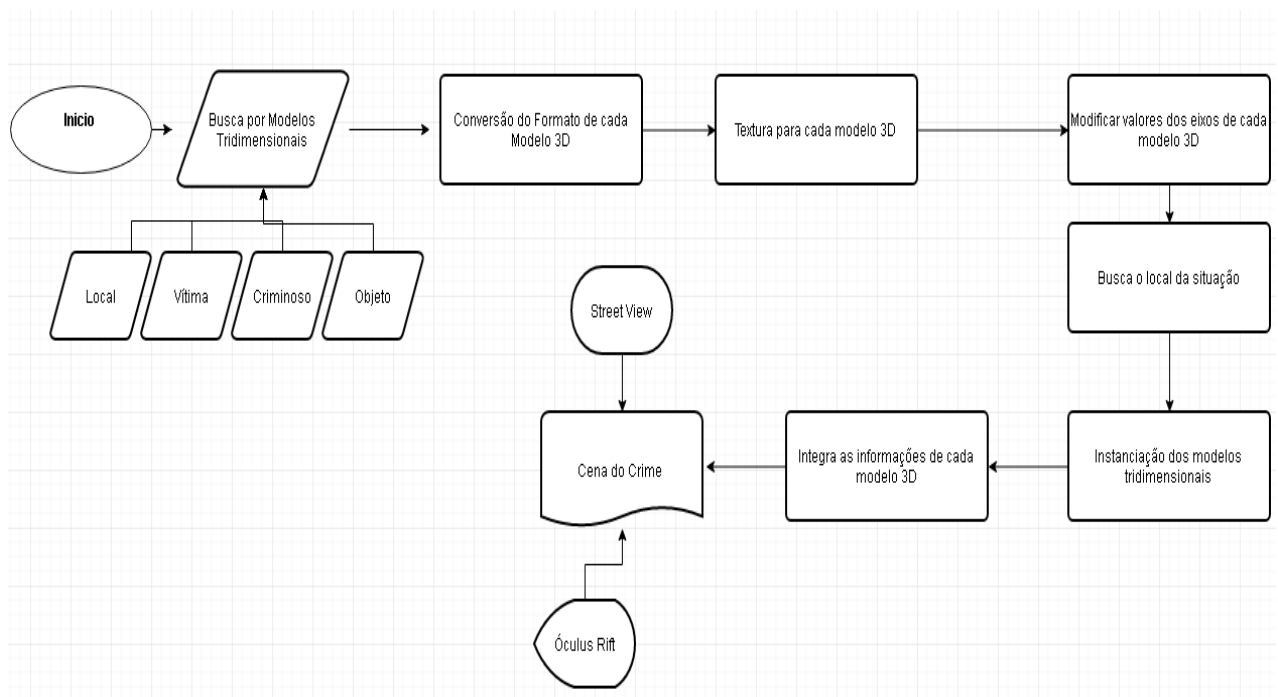


Figura 48 – Fluxograma instanciação modelos 3D a cena do crime

A classe responsável por enviar estas informações é a mesma classe que possui todos os dados retornados em JSON, convertidos em strings e armazenados para suas respectivas variáveis. Esta classe envia os dados referentes a cada modelo 3D para um método implementado e associado à classe do mapa panorâmico. Por possuir todos os modelos 3D, e

cada um associado a uma variável do tipo Objeto (Object), é possível chamar a função de cada um destes objetos enviando as informações necessárias de cada modelo 3D.

Para que as informações fiquem posicionadas ao lado de cada Modelo 3D foi preciso modificar os valores referentes aos eixos e da posição do componente de texto associado a cada modelo3D, foi preciso modificar também os valores referente às propriedades de: alinhamento, anchor, o espaçamento da linha no texto, como também o tamanho da fonte, para que o usuário tenha uma melhor visualização destas informações ao utilizar o dispositivo de imersão (Óculus Rift).

Foi também associado a cada modelo 3D um componente do Unity responsável por identificar colisões sobre objetos, o “Box Collider”, que circula o objeto de tal forma que qualquer *input* que seja realizado sobre esta área receba um determinado comportamento. As informações referentes a cada modelo são demonstradas na tela, ativando e desativando o componente de texto de cada modelo apenas pelo usuário estar olhando na direção do modelo 3D por um segundo com o uso do *Óculus Rift*, direcionando um ícone parecido com um alvo para o objeto do qual ele deseja visualizar as informações. As **figura 49** e **50** demonstram os modelos tridimensionais já posicionados na cena do crime e as informações referentes ao local do crime.

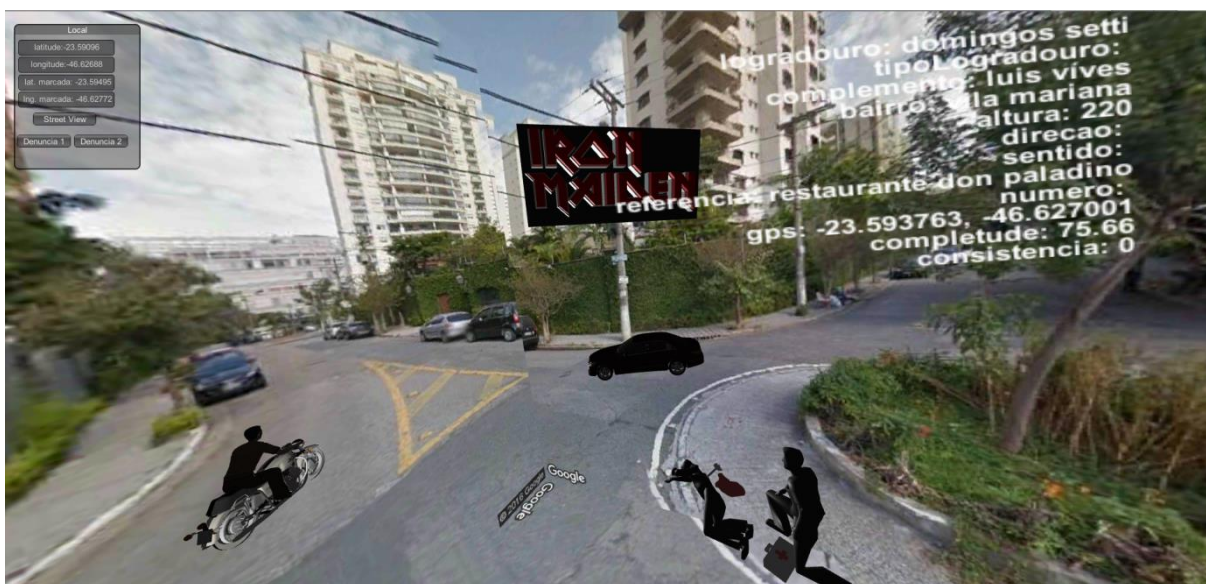


Figura 49 – Modelos 3D instanciados na cena do Crime e informações referentes ao Local do crime



Figura 50 – Alvo direcionado pelo Óculus Rift selecionando um modelo 3D

Todas estas etapas descritas são realizadas após o usuário clicar sobre o botão denominado “Denúncia 1” localizado no canto superior da tela, como na **figura 50**. O botão de “Denúncia 2” não possui nenhuma função, pois para este projeto só foi implementado um tipo de situação. De outra forma, a função deste botão seria projetar toda a cena de uma nova situação, com um novo local, outros modelos tridimensionais posicionados na cena e consequentemente com suas devidas informações.

4.3 Realidade Virtual e Gestos de Movimento

Ao final do projeto algumas alterações foram realizadas para acoplar o uso dos gestos de movimento junto à tecnologia de realidade virtual.

- Adequamos o mapa na RV para ter um modelo 3D curvo, em vez de um plano, o que deu uma sensação melhor na RV, como mostra a **figura 51**.
- Usamos o gesto de fechar a mão para controlar a panorâmica do mapa, e polegar direito pra cima com o punho cerrado para entrar no street view, e com a mão esquerda para sair.
- Usamos o ponto de visão do jogador para carregar as informações sobre as denúncias e objetos 3D. Em vez do clique do mouse, o jogador olha para a entidade por 1 para ter carregadas as informações.

Realizamos testes com o óculus rift e o leap motion, para verificar o desempenho do sistema ao acoplar a tecnologia de RV juntamente com seus dispositivos, como também o comportamento do sistema com esta integração, como mostra na **figura 52**.

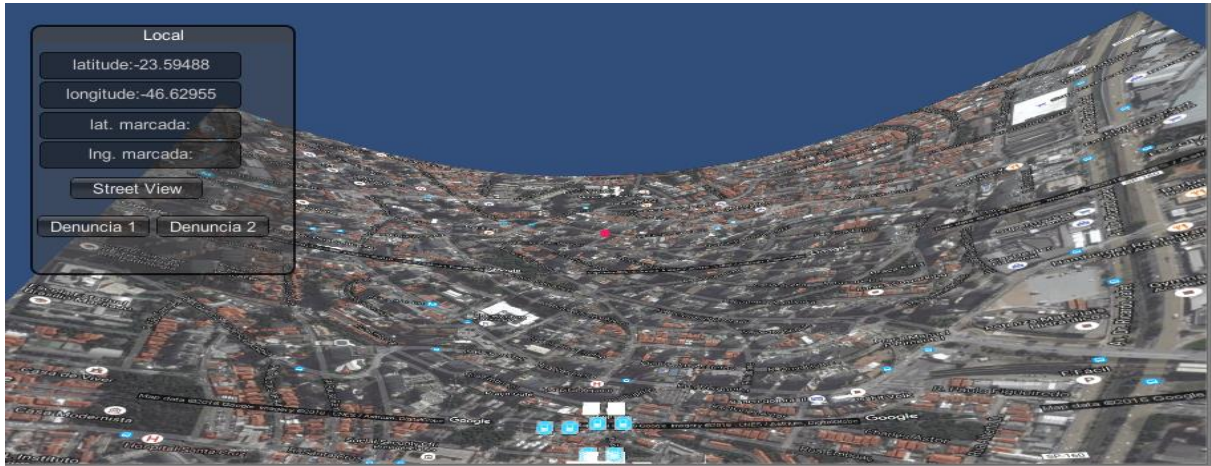


Figura 51 – Mapa Panorâmico com um tipo de modelo3D curvo



Figura 52 – Testes com óculus rift e leap motion sendo realizados

RESULTADOS OBTIDOS

Ao final do projeto obteve-se um novo módulo acoplado a um framework de visualização de informações estabelecido pelo grupo de IHC (Interação-Humano-Computador) do laboratório COMPSI – UNIVEM, juntamente com o grupo de pesquisas de IHC da universidade de São Carlos - UFScar. Um novo serviço que poderá ser invocado sob demanda.

Tal framework possuía apenas técnicas de visualização de informação utilizando overlays sobre mapas georreferenciados. Com este novo módulo, o operador do sistema será

capaz de optar por qual tipo de visualização deseja consultar a situação que acontece em tempo real: bidimensional ou imersiva com uso da realidade virtual.

CONCLUSÃO

Como resultado, obtivemos um sistema capaz de oferecer ao usuário uma forma de reconstruir um cenário de emergência a partir de dados reais de uma situação, cenário este composto por elementos tridimensionais estrategicamente posicionados ao ambiente associados a informações referentes às denúncias das vítimas para cada entidade envolvida na situação. Deste modo, o especialista teve acesso a este cenário utilizando um dispositivo estereoscópico e um dispositivo de sensor de movimento, aumentando assim seu nível de imersão ao ambiente, como também seu nível de SAW, através do acesso a estas informações.

Dessa forma, conclui-se que a visualização de informações tridimensionais em ambientes virtuais utilizando dispositivos de imersão e a interação do usuário ao sistema utilizando dispositivos de sensores de movimento, oferece um suporte diferencial para sistemas que visam otimizar o nível de SAW aos especialistas que necessitam de uma tomada de decisão rápida e eficaz em uma situação de emergência.

Como trabalho futuro, é possível implementar um modo de zoom ao mapa panorâmico, para que o usuário possa visualizar um ponto do mapa com uma determinada proximidade, assim como a implementação de navegação do usuário no modo *Street View* por meio de gestos captados pelo sensor de movimento, de modo que seja possível navegar ao redor dos elementos que compõem o cenário de uma situação e visualizá-los a partir de vários ângulos. Outra ideia a ser adicionada ao projeto é armazenar um grande número de modelos tridimensionais, para que quando uma nova situação de emergência for adicionada ao projeto, uma lógica implementada busque os modelos tridimensionais mais similares com as entidades referidas em casos particulares.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., & Pavel, M. (2005, March). A typology for visualizing uncertainty. In *Electronic Imaging 2005* (pp. 146-157). International Society for Optics and Photonics.
- [2] Pang, A. T., Wittenbrink, C. M., & Lodha, S. K. (1997). Approaches to uncertainty visualization. *The Visual Computer*, 13(8), 370-390.
- [3] WRV'97 1º Workshop de Realidade Virtual São Carlos, SP, 9-12 de Novembro de 1997 – Profº Dr. Claudio Kirner – “Introdução à Realidade Virtual” – Disponível em <http://docplayer.com.br/8775647-Wrv-97-1-o-workshop-de-realidade-virtual-sao-carlos-sp-9-12-de-novembro-de-1997-mini-curso-introducao-a-realidade-virtual.html>
- [4] Endsley, M. “What is Situation Awareness?”, 2011 pg 13-30.
- [5] Gouin, D., e Evdokiou, P. (2004). A showcase of visualization approaches for military decision makers. *Defence Research and Development Canada Valcartier (QUEBEC)*.
- [6] BOIN, A. 2009. The new world of crises and crisis management: Implications for policymaking and research. *Review of Policy Research* 26, 4, 367–377.
- [7] DUGDALE J., BELLAMINE-BEN SAOUD N., P. B., AND N., P. 2010. Simulation and emergency management. *Information Systems for Emergency Management*, 229–253.
- [8] Paravati, G; Lambert, F; Sanna, A; Ramirez, E. H; Demartini, C. “An immersive visualization framework for monitoring, simulating and controlling smart street lighting networks” – *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques – Pages 17-26 ACM Belgium 2012*
- [9] van de BROEK, A.C.; Neef, R.M. ; Hanckmann, P. ; van Gosliga, S. P. ; van Halsema, D. - “Improving maritime situational awareness by fusing sensor information and intelligence” - *Information Fusion (FUSION), Proceedings of the 14th International Conference on- pages 1-8 IEEE Chicago, IL 2011.*
- [10] Bui, A.A.T.; Aberle, D.R.; Kangarloo, H. “Timeline : Visualization integrated Patient Records” - *Information Technology in Biomedicine, IEEE Transaction on - Pages 462-473 - 2007.*
- [11] CHITTARO, L. ; BUTTUSI, F. ; ZANGRANDO, N. ; “Desktop virtual reality for emergency preparedness: user evaluation of an aircraft ditching experience under different fear arousal conditions” - *VRST '14 Proceeding of the 20th ACM Symposium on Virtual Reality Software and Technology - Pages 141-150 ACM New York, NY, USA 2014.*
- [12] St Julien, Tazama U., and Chris D. Shaw. "Firefighter command training virtual environment." *Proceedings of the 2003 conference on Diversity in computing*. ACM, 2003.

- [13] Broughton, Michael. "Virtual planning rooms (ViPR): a 3D visualisation environment for hierarchical information." *Proceedings of the 7th Australasian User interface conference-Volume 50*. Australian Computer Society, Inc., 2006.
- [14] Pintore, Giovanni, et al. "3DNSITE: A networked interactive 3D visualization system to simplify location awareness in crisis management." *Proceedings of the 17th International Conference on 3D Web Technology*. ACM, 2012.
- [15] BOTEGA, L. "Modelo de Fusão Dirigido por Humanos e Ciente de Qualidade de Informação" – Universidade Federal de São Carlos – São Carlos - SP - 2015
- [16] Freitas, C. M. D. S., et al. Introdução à Visualização de Informação. *Revista de Informática Teórica e Aplicada*, v.2, p.143-158. 2001.
- [17] Zorzal, Ezequiel Roberto, et al. "Visualização de informação com Realidade Virtual e Aumentada." *IRealidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*: 256-275.
- [18] Kirner C., Pinho S. M. "Uma introdução à realidade virtual" Departamento de Computação UFScar- São Carlos – 2010 – fonte : <http://grv.inf.pucrs.br/tutorials/rv/index.htm>
- [19] Kirner C., Tori R., Siscouto R. – "Fundamentos e tecnologia de realidade virtual e aumentada" – Livro de Pré-Simpósio VIII Symposium on virtual Reality – Belém – 2006.
- [20] Machado, L. S. Conceitos básicos da realidade virtual, Monografia, INPE-5975-PUD/025, Instituto Nacional de Pesquisas Espaciais, São José dos Campos/SP, novembro. Disponível on-line em: <http://www.lsi.usp.br/~liliane/conceitosrv.html>, 1995.