

Criação de um simulador para o treinamento de combate e prevenção a incêndios através da realidade virtual

Bruno Moreira Carnaúba
Sistemas de informação
Centro Universitário CESMAC
Maceió, Brasil
brunomc.contato@gmail.com

Rouger Brian Rousan de Lucena
Alexandre
Sistemas de informação
Centro Universitário CESMAC
Maceió, Brasil
rougert.brian@gmail.com

Carlos Alberto Correia Lessa Filho
Centro de Inovação Tecnológica
Centro Universitário CESMAC
Maceió, Brasil
carloswgama@gmail.com

Resumo—A realidade virtual é uma tecnologia que tem ganhado cada vez mais espaço no treinamento prático de pessoas, tendo como algumas das vantagens reduzir custos e riscos. Levando isso em consideração, esta pesquisa teve como alvo o desenvolvimento de um simulador para o treinamento de combate e prevenção a incêndios com extintores através da realidade virtual por meio do motor de jogo *Unity* e da ferramenta para modelagem 3D *Blender*. Essas e outras ferramentas são apresentadas no decorrer do trabalho, assim como alguns equipamentos de realidade virtual disponíveis no mercado como o *Oculus Rift S* que foi o equipamento utilizado no desenvolvimento da aplicação. No decorrer da pesquisa também é abordado qual metodologia foi definida para dar início à criação, trechos do desenvolvimento e por fim como o simulador, que conta com um treino teórico e prático, funciona.

Palavras-chave—Realidade virtual, Simulador, Incêndio, Treinamento

I. INTRODUÇÃO

Os incêndios são situações trágicas que podem trazer não apenas danos materiais, mas também pôr a vida de pessoas em risco. Em 2013, um incêndio na cidade de Santa Maria causou 242 mortes e mais de 600 feridos [1]. Diante disso, é notável o quão é importante saber como se prevenir para evitar essas situações e como agir em caso de incêndio.

Muitas empresas, mesmo que isentas da obrigação de ter um setor voltado para a brigada de incêndio optam por treinar seus funcionários para criar e manter um ambiente seguro, para isso passam por um treino teórico e prático, onde no treino prático é realizado simulações que incluem os passos que serão executados em uma situação real [2].

Nos dias de hoje tem sido comum vermos o uso de tecnologias na realização de variados treinamentos, uma delas é a realidade virtual que tem a capacidade de enganar os sentidos de um usuário fazendo com que ele tenha a sensação de estar naquele ambiente virtual [3].

Esses equipamentos estão presentes em áreas como a de jogos e entretenimento, turismo, concepção de projetos para *design* na área de construção civil e apresenta aplicação em diferentes simuladores, como os de medicina, exploração espacial e treinamentos militares [4].

O uso de tecnologia em treinamentos trazem benefícios para os treinos, tanto para a empresa como também para o participante, pois permite reduzir o desgaste de

equipamentos, diminuir o tempo e custo do treinamento, melhorar a retenção de conhecimentos, neutralizar os riscos de acidentes, entre outras vantagens que são concedidas [5].

Assim, este trabalho tem como objetivo a criação de um simulador que faça uso da realidade virtual para o treinamento de prevenção e combate a incêndios com extintores com intuito de colaborar diretamente para a prevenção de incidentes e possíveis acidentes, garantindo uma postura mais segura em caso de fogo, podendo ser utilizado por exemplo no treinamento de funcionários de uma empresa.

II. REALIDADE VIRTUAL

A realidade virtual permite que o usuário sinta como se estivesse em um determinado ambiente sem de fato estar. Dessa forma um usuário pode ser imerso em um ambiente de realidade virtual e experimentar sensações nunca antes vividas [6].

O principal truque da realidade virtual que cria a ilusão de volume e imersão, está no uso da técnica de estereoscopia, que consiste da exibição de duas imagens semelhantes, mas com o ângulo ligeiramente diferente para cada olho, como pode ser visualizado na [7, Fig. 1], que quando o usuário visualiza pelo *headset* de realidade virtual, tudo fica alinhado [7].

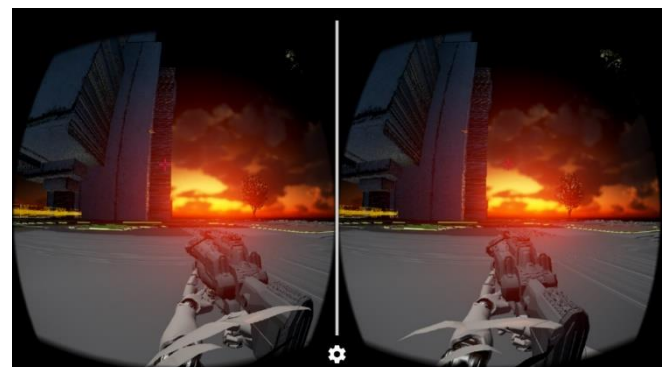


Fig 1. Exemplo de exibição estereoscópica de um jogo para o Google Cardboard.

Além disso, o uso da tecnologia de áudio 3D, a interação com o cenário por meio de controles sem fio e outras técnicas usadas na criação do ambiente, estão envolvidas na criação da experiência da realidade virtual [7].

A. Óculos de realidade virtual

Na atualidade, os equipamentos de realidade virtual mais comuns de se encontrar no mercado são os que usam da tecnologia *Head mounted display*, os comumente chamados óculos de realidade virtual [8].

Um dispositivo que tem sido sucesso de vendas, chegando a ser o equipamento mais vendido no último trimestre de 2019, é o *Playstation VR*, assim como mostrado na Tabela I. Um dos principais motivos de seu sucesso é o custo, já que exige de um console *Playstation 4*, que possui *hardware* mais modesto quando comparado com os requisitos de computador exigidos em alguns de seus concorrentes [9].

TABELA I. TABELA DE VENDAS DE HEADSETS DE REALIDADE VIRTUAL NO ÚLTIMO TRIMESTRE DE 2019.

Headset	Número de vendas (unidades)
Playstation VR	338.000
Oculus Quest	317.000
Valve Index	103.000
Oculus Go	84.000
Oculus Rift S	71.000

^a Elaborado por SuperData [10]

Entretanto, dispositivos que são sem fio e independentes, ou seja, que não precisam do computador têm ganhado grande espaço no mercado, que é o caso do *Oculus Quest* e do *Oculus Go* onde os aplicativos são instalados e processados no próprio óculos, o que traz como uma das vantagens a liberdade de movimento do usuário no espaço onde está sendo usado [11].

B. Ferramentas de desenvolvimento de uma realidade virtual

Uma aplicação de realidade virtual necessita de um ambiente virtual interativo, que esteja integrado com dispositivos que façam uso dessa tecnologia, como óculos, luvas entre outros, por onde são reproduzidos estímulos visuais, auditivos ou táteis [12].

Uma das ferramentas para o desenvolvimento são as denominadas de motor de jogo (*game engine*). Tori, Kiner e Siscoutto [13, pp. 205] adotam a seguinte definição para *game engine*: “um conjunto de componentes de *software* integráveis e reutilizáveis, que fornecem serviços utilizados em um jogo eletrônico”. É nela onde são programadas as ações, reações e interações do cenário e objetos do ambiente virtual [14].

Alguns dos motores de jogos que suportam o desenvolvimento de realidade virtual são o *Unity*, *Unreal Engine* e a *CryEngine*. Sendo o *Unity* o motor de jogo mais usado, tendo como uma das razões o fato de ser fácil de iniciar, por ter uma ampla documentação e uma comunidade sólida com muito conteúdo disponível na *internet* [15].

O *Unity* tem suporte nativo de desenvolvimento de realidade virtual para os equipamentos: *Oculus Rift*, *Steam VR / HTC Vive*, *Playstation VR*, *Microsoft HoloLens* e o *Daydream View*, e também é possível aumentar o número de aparelhos suportados com a adição do SDK (*Kit* de desenvolvimento de *software*) disponibilizado pela fabricante do equipamento [16].

O principal concorrente da *Unity* é a *Unreal Engine*, que se destaca no quesito de gráficos, porém tem uma

documentação menos explicativa e uma comunidade menor, o que pode afetar a facilidade de aprendizagem [17].

A *Unreal* suporta o desenvolvimento para *Oculus Rift*, *Steam VR / HTC Vive*, *Playstation VR*, *Samsung Gear VR*, *Google VR* e também tem suporte para outros equipamentos por meio da conexão do *kit* de desenvolvimento do equipamento [18].

Softwares de modelagem 3D complementam os motores de jogo com os recursos de criação de objetos tridimensionais, que são usados no ambiente virtual, nos quais posteriormente são aplicadas texturas, que podem ser ditas como a “pele” do objeto 3D criado e dar movimentos através da aplicação de animações [13].

O *Blender* é um desses *softwares*, tratando-se de um programa gratuito e de código aberto. Permite fazer modelagem, estruturação de esqueleto digital, animação, simulação, renderização, composição e rastreamento de movimento inteiramente 3D, além de também disponibilizar funções para edição de vídeo e criação de jogos. Normalmente essa é a ferramenta que está presente em pequenos estúdios [19].

O *Maya* é um outro *software* de modelagem 3D que tem como um de seus destaques suas opções de texturização, que permitem o desenvolvimento de objetos muito detalhados e realistas. Apesar de ser pago, essa é a ferramenta padrão de grandes corporações [19].

III. TRABALHOS CORRELATOS

Dois trabalhos que tem relação com o objetivo desta pesquisa de desenvolver uma aplicação para treinar ou educar pessoas são o *KinectE.R.* e o *IntEducaTrânsito*, ambos publicados na *SBGames*.

O projeto *KinectE.R.*, segundo Lanza, Lacerda e Souza [20], foi desenvolvido para treinar profissionais do setor médico em procedimentos cirúrgicos dos mais simples aos mais complexos com o uso do *Kinect*, de forma que consiga educar e divertir ao mesmo tempo. Onde o jogador deve escolher as ferramentas cirúrgicas corretas para fazer determinados procedimentos levando em consideração a profundidade, velocidade e firmeza dos movimentos, e se necessário aplicar medicamentos para estabilizar a condição do paciente.

O segundo projeto apresenta o jogo denominado *IntEducaTrânsito*, que tem como objetivo educar o jogador sobre as normas de trânsito, por meio da condução de um carro pelas ruas de um ambiente virtual, respeitando os veículos parados ou em movimento, as placas, sinalizações, sinaleiras e os gestos de autoridade. No jogo desenvolvido é possível inclusive controlar o veículo através de um volante, *joystick*, *tablet* ou *smartphone* [21].

Em um estudo de caso sobre um simulador de guindastes Petrobras elaborado por Alves, Haydu e Souza [22], foi apresentado a possibilidade de qualificar os operadores sem interferir na produção e sem causar desgaste indevido nos equipamentos. Onde também é possível reproduzir condições que não são viáveis nos treinamentos tradicionais, como falhas estruturais, acidentes, maus funcionamentos do equipamento, falhas de procedimentos, condições climáticas adversas, entre outros.

Uma das possíveis situações nesse simulador é o rompimento de um cabo de aço de um guindaste com carga sobre uma planta de processo de gás com intuito de ensinar ao aluno como deve proceder nesse caso.

Como pode ser observado, os jogos citados acima apresentam características que trazem praticidade para a realização de treinos práticos, que fora do virtual apresentam algumas dificuldades para a sua execução, seja pelo tempo gasto na preparação do treino, pelos custos, por precisar de condições especiais que não são controláveis, tanto do ambiente onde acontece o treino quanto do objeto a ser manipulado, ou outros motivos.

Além disso, dependendo do tema abordado e da forma como foi elaborado o jogo, pode atrair pessoas que jogam por achar divertido e que acabam por adquirir conhecimentos que podem ajudá-lo em alguma situação do mundo real. O *KinectE.R.* por exemplo, tem um modo “casual”, com intuito de simplificar alguns aspectos e deixar mais apresentável para quem não é da área médica [20] e o *IntEducaTrânsito* tem um público alvo que também inclui crianças e adolescentes, mesmo não tendo a idade mínima para dirigir e tirar carteira de habilitação [21].

Assim como nos trabalhos citados anteriormente, o trabalho desta pesquisa visa desenvolver um estudo para a criação de um simulador que possa contribuir no treinamento de pessoas, mesmo para quem não tenha conhecimento prévio. Pode ser adotado por exemplo para o treinamento de funcionários de uma empresa, para que estejam mais seguros sabendo como se prevenir para evitar situações de incêndio e como combatê-lo caso aconteça em seu ambiente de trabalho.

IV. METODOLOGIA

O simulador para treinamento de combate e prevenção a incêndios foi desenvolvido com objetivo de disponibilizar uma nova forma de aprendizado sobre como se prevenir e evitar um incêndio e como se comportar caso um incêndio aconteça. Para o seu desenvolvimento foi pensada em uma metodologia com intuito de definir como buscaria as informações acerca dos treinamentos para combate a incêndios feitos por profissionais dessa área, o que o simulador precisaria ter, qual equipamento de realidade virtual seria usado e quais ferramentas e linguagem de programação seriam utilizadas para a criação do simulador.

A. Treinamento para prevenção e combate a incêndios

O treino reproduzido no simulador foi baseado no treinamento de brigada de incêndio, que é usado para o treinamento de funcionários de uma empresa para que possam atuar na prevenção de ocorrências de incêndios e saibam como combater-las caso ocorra, para isso, o funcionário precisa passar por um treino teórico e prático [2].

Após o estudo por materiais escritos e vídeos, foi dividido o simulador da seguinte forma:

1) Treino teórico abordando:

a) Classes de incêndio A, B, C, D, e K, que são referentes ao tipo de material que está em combustão;

b) Agentes extintores para cada classe, como água, espuma, pó químico, entre outros, que podem atender a uma ou mais classes;

c) Dicas de combate a incêndio;

d) Dicas de prevenção para evitar que incêndios aconteçam.

2) Treino prático abordando:

a) Uso correto do extintor, onde o participante precisa escolher o extintor correto para combater o fogo de alguns objetos presentes no cenário;

b) Ordem das ações para combater a um incêndio, como no caso do combate ao incêndio em equipamentos elétricos onde contém vítimas, no qual o participante precisa desligar primeiro a eletricidade, para depois socorrer a vítima e por último apagar o fogo.

B. Equipamento e ferramentas a serem usadas

Como o cenário requer alguns modelos 3D, alguns foram adquiridos de forma gratuita em alguns sites e outros foram criados por meio da ferramenta *Blender*, que foi escolhida principalmente pelo motivo de ser gratuita, mas também por ter bastante conteúdo disponível na internet que ensinam como usá-lo. Já para os modelos de personagens 3D foi utilizada a ferramenta *Adobe Fuse*, que permite a criação de personagens humanos sem a necessidade de ter conhecimento com modelagem 3D e a ferramenta online *Mixamo*, que permite animar os personagens que foram criados.

O equipamento de realidade virtual para o qual o simulador foi desenvolvido é o *Oculus Rift S* e para a criação do simulador foi escolhido o motor de jogo *Unity*, pela mesma razão que foi escolhido o *Blender*, por ser gratuito e ter muito conteúdo disponível na internet. Essa escolha levou ao uso da linguagem de programação *C#* durante o desenvolvimento, que foi utilizada para fazer o código do simulador por meio do editor de código *Visual Studio Code*.

Com intuito de simplificar o desenvolvimento do treino teórico, as suas instruções são passadas por meio de um vídeo reproduzido no simulador que foi criado através da ferramenta *Vegas Pro*. Também foi necessária fazer edições de imagens como das texturas dos modelos 3D, para isso foi utilizado o programa *Adobe Photoshop*.

1) Oculus Rift S

O headset de realidade virtual *Oculus Rift S* [23, Fig. 2], é uma versão aprimorada do seu antecessor, o *Oculus Rift*. O equipamento conecta-se ao computador por meio de uma porta *USB 3.0* e uma porta *DisplayPort*. Além do headset, também é usado em conjunto um par de controladores sem fio que contam com um botão analógico e outros dois botões “X” e “Y” para o polegar, um botão de gatilho para o dedo indicador e um botão localizado na alça do controle, nos quais esses dois últimos tem sua área de toque sensíveis, o que permite imitar o descanso dos dedos [24].



Fig. 2. Oculus Rift S.

O rastreamento da posição do Usuário e dos controles no ambiente real é feito pelas câmeras que estão integradas ao *headset*, sendo duas câmeras virada para frente, duas viradas para o lado e uma virada para cima, o que permite um melhor rastreamento e uma instalação mais fácil [23].

O aparelho é posicionado na cabeça com o auxílio de uma cinta de velcro que segura o *headset* por cima e de um molde de plástico acolchoado e ajustável, que fica na testa e nas laterais da cabeça. É nesse molde onde ficam os alto-falantes direcionais do aparelho, mas se preferir também é possível conectar um outro fone de ouvido pela porta localizada na lateral do equipamento. O dispositivo também conta com um único LCD com resolução de 2560x1440, no qual tem uma taxa de atualização dos quadros de 80Hz [23].

As especificações recomendadas do aparelho pedem um computador com processador i5-4590/AMD Ryzen 5 1500X ou superior, placa de vídeo NVIDIA GTX 1060/AMD Radeon RX 480 ou superior, 8 GB de memória RAM ou mais, uma porta USB 3.0 e uma saída de vídeo de porta *DisplayPort* [25].

2) Adobe Fuse e Mixamo

Como citado anteriormente, o *software Adobe Fuse* não exige experiência com modelagem 3D para usá-lo. Ele disponibiliza diferentes padrões de modelos de personagens que podem ser personalizados com diferentes cores, texturas, ajustes no corpo e mais de 280 atributos, como roupas, chapéus e sapatos que também podem ter suas cores alteradas. Além disso, enquanto se encontra na versão *beta*, o programa está gratuito [26].

Já a ferramenta *online Mixamo*, que também faz parte da família Adobe, além de disponibilizar personagens prontos, também permite animar personagens 3D que foram criados em uma plataforma externa, como no *Adobe Fuse* ou pelo *Blender*, isso porque a ferramenta implementa um esqueleto humano completo ao modelo, como ilustrado na [27, Fig. 3], no qual cada osso do esqueleto poderá ser movido. Essa técnica de animação esquelética também é chamada de *Rigging*. São milhares de animações capturadas por movimento por atores profissionais disponíveis de forma gratuita [28].

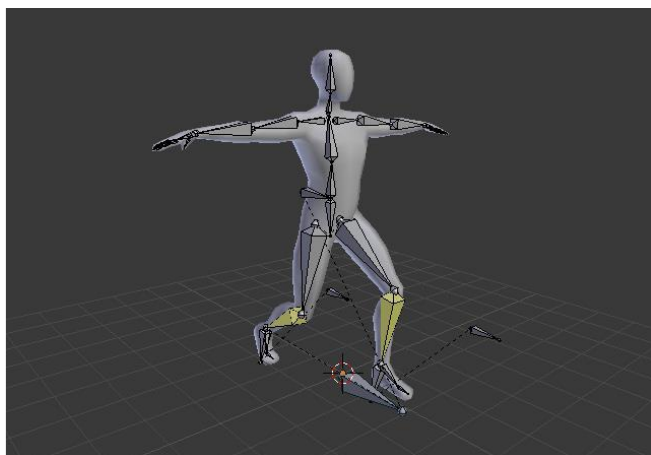


Fig. 3. Ilustração de um modelo 3D humano com esqueleto.

3) C# e Visual Studio Code

A linguagem de programação C# é a linguagem utilizada pelo moto de jogo *Unity*. Segundo Dionysio, Tavares e

Santos Júnior [29], a linguagem foi criada pela *Microsoft* e teve grandes influências de linguagens como C++, *Java* e *Object Pascal*. Tem sua estrutura baseada na linguagem C/CC++, e possui várias semelhanças com elementos do C++ e *Java*.

O próprio *Unity* possui tutoriais em seu site que introduz um usuário iniciante a codificação com C# em sua plataforma, que começa abordando conceitos básicos e programação, como variáveis, funções e classes. A comunidade também produz muito conteúdo gratuito que ensinam como programar aplicações com C# junto com o *Unity*.

O *Visual Studio Code*, escolhido para editar os códigos C#, é um editor de códigos multiplataforma que atende a mais de 30 linguagens de programação, e ainda pode ser estendido com novas funções através de extensões, que são disponibilizadas na loja do programa [30].

V. SIMULADOR

Com os requisitos registrados e ferramentas escolhidas, foi dado início ao desenvolvimento do simulador, de forma que ele cumpra o objetivo de instruir o participante sobre como se prevenir, o que usar e como se comportar durante uma ocorrência de incêndio no local em que se encontra.

A. Desenvolvimento

O desenvolvimento foi iniciado pela elaboração do cenário, que é composto de uma casa com algumas salas, onde em cada sala possui materiais em fogo referentes à uma classe de incêndio, e uma área externa onde são passadas as instruções teóricas por meio de um telão.

Para a reprodução do vídeo que passa as instruções teóricas, foi usado o componente *Video Player* da própria *engine* que permite reproduzir um vídeo através da substituição da textura do objeto em tempo de execução. O objeto no qual o vídeo é reproduzido foi criado no próprio *Unity*. O *GameObject* no formato “Plano” foi escolhido dentre os disponibilizados (Fig. 4), porque por padrão renderiza apenas um lado, que é o lado no qual o vídeo é reproduzido.

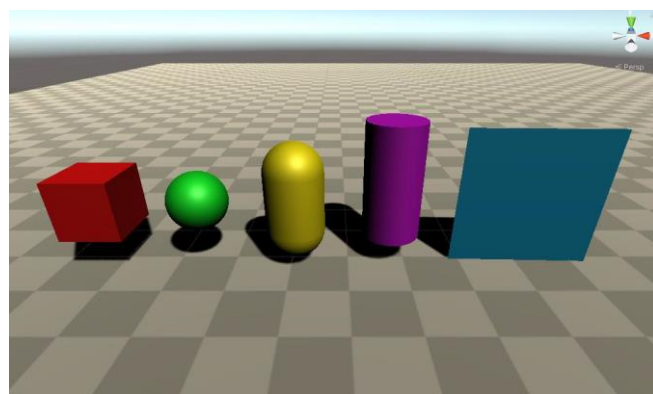


Fig. 4. Alguns formatos de objetos presentes no Unity.

O primeiro objeto buscado na *internet* foi o extintor, que teve sua textura referente ao rótulo editada através do *photoshop*, para poder criar rótulos diferentes e dessa forma diferenciar o agente extintor de cada um.

A casa onde é feito o treino prático foi modelada através do *Blender* (Fig. 5), no qual partiu de um objeto “Plano” que foi dividido em faces, para poder serem movidas e então

levantar as paredes e criar o teto. Posteriormente, as faces referentes as paredes foram unidas em apenas uma, assim como a do teto e a do chão, com intuito de facilitar a texturização com o uso de materiais para cada face unida e melhorar a performance do modelo 3D. As faces nas quais o participante não tem ângulo para enxerga-las durante a execução do simulador, também foram deletadas para otimizar a performance.

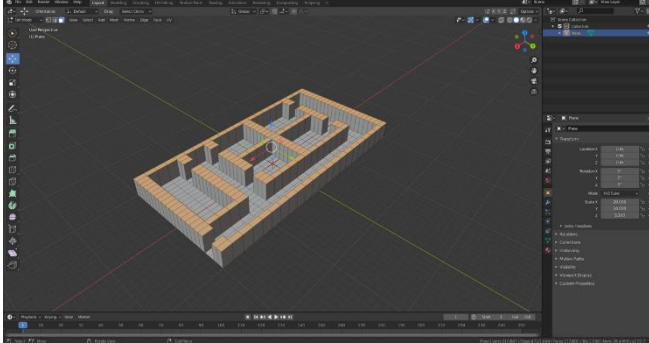


Fig. 5. Modelagem da casa pelo Blender.

Com a casa criada, implantada e texturizada no Unity, foi feito a decoração das salas com objetos encontrados na internet como paletes de madeira, baterias de lítio, filtro de linha, painéis e outros, que em fogo representam uma classe de incêndio. Após a elaboração, a casa ficou como mostrado na Fig. 6.



Fig. 6. Visão de cima da casa onde acontece o treino prático.

O efeito de fogo também foi encontrado pronto na internet, disponibilizado de forma gratuita na loja do próprio Unity. Já o efeito de fumaça que representa os agentes extintores, foi criado através do Unity. Eles são criados por meio do sistema de partículas do motor de jogo, que permite a criação do efeito de nevoeiro, explosão, fumaça, entre outros.

Após a elaboração do cenário, o próximo passo foi implementar a realidade virtual. Para isso, a primeira coisa que foi feita foi ir nas configurações do projeto no Unity, e habilitar o suporte à realidade virtual na seção de jogador. Isso já faz com que o usuário possa olhar ao seu redor quando a aplicação for iniciada. Mas, para ter mais recursos para o Oculus Rift, foi baixado e inserido no projeto o pacote de integração do Oculus, encontrado na loja do Unity de forma gratuita com o nome “Oculus Integration”.

O pacote disponibiliza o objeto “OVRPlayerController”, que permitiu que o usuário pudesse se movimentar pelo cenário. Além disso, dentro dele foi inserida as mãos pela adição dos objetos “CustomHandLeft” e “CustomHandRight”, que também fazem parte do kit, o que possibilitou que o usuário pudesse pegar e mover qualquer

objeto do cenário que tenha o componente *collider*, a física *RigidBody* do próprio Unity e o script “OVR Grabbable” disponibilizado nesse mesmo pacote.

Após a realidade virtual ser integrada, foram criados os demais mecanismos do simulador como o do extintor. Para ele foi criado um script que ao atender uma condição, ele reproduz a partícula referente ao agente extintor. Essa condição checa se o usuário está pressionando o botão “Primary Index Trigger”, referente ao gatilho do controle touch do Oculus Rift, e acessa a variável “isGrabbed” do script “OVR Grabbable” comentado anteriormente, para checar se o usuário está segurando o equipamento. Se estiver atendido a essa condição, é iniciado a reprodução da partícula.

O agente extintor só deve efetuar a extinção do fogo se ele for o correto para aquela classe de incêndio. Para isso, foi utilizada a função “OnParticleCollision” que é do próprio Unity, no script de cada agente extintor, em que é acionada quando há a colisão da partícula em questão com o colisor de algum outro objeto, no caso, no “Box Collider” do fogo. Quando essa função é chamada, é checada a tag do objeto fogo para saber a qual classe de incêndio ele está representando, para que sejam aplicadas as ações correspondentes a aquele contato como parar a reprodução da partícula de fogo se estiver usando o agente extintor correto para aquela classe de incêndio.

Além do mecanismo de extinção do fogo, foi criada uma mecânica na sala referente à classe C (Fig. 7), em que o usuário precisa seguir uma ordem para combater o incêndio, no qual primeiro precisa primeiro desligar os disjuntores, para depois salvar uma vítima e por último apagar o fogo. Para isso, foi criado um objeto que possui um script que serve como controle, possuindo variáveis booleanas, onde cada uma representa um item dessa ordem, e são alteradas quando a mão do usuário colide com um desses itens, como na colisão da mão com a vítima, onde a variável referente a ele é alterada para verdadeiro se a variável referente ao disjuntor já estiver verdadeira. Dessa forma, é possível saber se o usuário está pulando algum passo.



Fig. 7. Sala referente à classe de incêndio C.

B. Funcionamento

No início da execução do simulador, o participante pode escolher se deseja começar pela etapa teórica, ou se quer pular direto para a etapa prática, pois ele já pode ter passado pelas instruções teóricas anteriormente.

Como mencionado anteriormente, a etapa teórica consiste de apresentar ao participante o que são as classes de incêndio e quais são, informar quais são os agentes extintores para cada classe e dar algumas dicas de combate e prevenção a incêndios, através de um vídeo que é reproduzido em um

telão presente no cenário. Enquanto isso, o participante pode interagir com os extintores que ali estão. A área referente a essa etapa é demonstrada na Fig. 8.



Fig. 8. Área do cenário referente à etapa teórica do treinamento.

Ao concluir a etapa teórica, é indicado no final do vídeo o início das atividades práticas, informando o que o participante deve fazer para concluí-las com sucesso. As atividades consistem de utilizar os extintores corretos para apagar os incêndios de diferentes classes, seguindo a ordem correta de combate de acordo com a situação do ambiente em que se encontra. O combate do incêndio no simulador, usando um equipamento de realidade virtual, pode ser visualizado na Fig. 9.



Fig. 9. Usuário fazendo uso do extintor no simulador.

Se o participante usar o extintor errado no combate ou seguir a ordem errada de como proceder, é apontado com uma mensagem na tela o erro do participante, como demonstrado na Fig. 10.



Fig. 10. Mensagem de erro por estar usando o extintor incorreto.

No fim do combate de incêndio de algumas salas, algumas dicas de prevenção são dadas por meio de áudio para dizer como aquele fogo poderia ter sido evitado. O treino é concluído quando o participante consegue combater o incêndio de todas as salas de forma correta.

VI. CONSIDERAÇÕES FINAIS

A proposta do simulador desenvolvido é contribuir para o aprendizado acerca da prevenção a incêndios e o combate a eles com extintores, com o uso da tecnologia de realidade virtual, que permite tornar a experiência mais próxima do real, mas de forma livre de riscos, com diminuição de custos, afinal os materiais queimados e os extintores usados são virtuais, economia de tempo, pois não precisa preparar o cenário para cada novo treinamento e demais vantagens que com a continuação do desenvolvimento podem ser exploradas, como a implementação do registro do progresso do usuário de forma automática, adição de mais situações de incêndios específicas, de outras formas de combate, como usando mangueira, entre outras.

No estado atual do simulador, acredita-se que seja possível adquirir bastante conhecimentos que podem ser levados para situações reais, mas continua sendo importante o acompanhamento de um profissional da área para que haja instruções mais específicas durante o treino, como as referentes ao posicionamento do combatente. Em projetos futuros, para medir a qualidade do simulador e até mesmo realizar novos aprimoramentos, planeja-se realizar testes com alguns estudantes ou funcionários da instituição ao qual os autores fazem parte.

O projeto desenvolvido tem seu código e arquivos disponibilizados em um repositório público no *GitHub*¹, para que outros desenvolvedores possam reutilizar modelos 3D, trechos de códigos, usá-lo como estudo ou aprimorar o que já foi desenvolvido, e tem sua versão executável disponibilizada no *Google Drive*², com objetivo de realizar testes e para que possibilite *feedbacks* sobre possíveis correções de erros e demais aprimoramentos técnicos. Vale frisar que a versão disponibilizada na data de publicação do artigo não está completa.

REFERÊNCIAS

- [1] M. Luiz, “Dois anos depois, veja 24 erros que contribuíram para tragédia na Kiss,” *GI*, 2015. <http://g1.globo.com/rs/rio-grande-do->

¹ Disponível em: <https://github.com/guminsha/TCC-Simulador-Incendio>

² Disponível em:

https://drive.google.com/drive/folders/1455eJbbjOkyNrl_BqngGqvU4LSoBIPpq

- sul/noticia/2015/01/dois-anos-depois-veja-24-erros-que-contribuiram-para-tragedia-na-kiss.html (accessed Dec. 03, 2019).
- [2] Previnsa, “Guia completo sobre Brigada de Incêndio: saiba mais!,” *Previnsa*, 2018. <https://blog.previnsa.com.br/guia-completo-sobre-brigada-de-incendio-saiba-mais/> (accessed Jan. 15, 2020).
- [3] D. Coutinho, “O que é Realidade Virtual? Entenda melhor como funciona a tecnologia,” *Techtudo*, 2015. <https://www.techtudo.com.br/noticias/noticia/2015/09/o-que-e-realidade-virtual-entenda-melhor-como-funciona-a-tecnologia.html> (accessed Dec. 04, 2019).
- [4] P. Zambarda, “CINCO PROFISSÕES QUE JÁ ESTÃO UTILIZANDO VR NO SEU COTIDIANO,” *RevistaMundo360*, 2016. <https://revistamundo360.com/2016/11/25/cinco-profissoes-que-ja-estao-utilizando-vr-no-seu-cotidiano/> (accessed Dec. 05, 2019).
- [5] Oniria, “5 MOTIVOS PARA UTILIZAR SIMULADORES NOS TREINAMENTOS DE EQUIPES,” *Oniria*, 2018. <https://oniria.com.br/4-erros-sobre-a-gamificacao-no-trabalho/> (accessed Jul. 22, 2020).
- [6] A. Cardoso, E. Lamounier Júnior, C. Kirner, and J. Kelner, “Tecnologias e Ferramentas para o Desenvolvimento de Sistemas de Realidade Virtual e Aumentada,” in *Tecnologia para o Desenvolvimento de Sistemas de Realidade Virtual e Aumentada*, Pernambuco, 2007.
- [7] Codificar, “O que é e como funciona a Realidade Virtual,” *Codificar*, 2018. <https://codificar.com.br/o-que-e-e-como-funciona-realidade-virtual/> (accessed Jul. 22, 2020).
- [8] R. da S. Barboza, M. P. Barbosa, and J. M. da Silva Junior, “Desenvolvimento rápido de ambientes para realidade virtual em Unity utilizando PhotoSphere e CubeMap,” in *XVI SBGames*, 2017, pp. 1312–1326, [Online]. Available: <https://www.sbgames.org/sbgames2017/papers/Tutoriais/176371.pdf>.
- [9] N. Pino, “PlayStation VR Review,” *Techradar*, 2019. <https://www.techradar.com/reviews/gaming/playstation-vr-1235379/review> (accessed Feb. 16, 2020).
- [10] SuperData, “SuperData XR Q4 2019 Update,” *News Break*, 2020. <https://www.newsbreak.com/news/1498396977270/superdata-xr-q4-2019-update> (accessed Jul. 21, 2020).
- [11] B. Caddy, “Oculus Quest review,” *Techradar*, 2019. <https://www.techradar.com/reviews/oculus-quest-review> (accessed Feb. 16, 2020).
- [12] J. W. V. De Faria, E. G. Figueiredo, and M. J. Teixeira, “Histórico da realidade virtual e seu uso em medicina,” *Rev Med*, São Paulo, pp. 106–114, 2014.
- [13] R. Tori, C. Kirner, and R. Siscouto, *Fundamentos e tecnologia de realidade virtual e aumentada*, 1st ed. Belém: Editora SBC, 2006.
- [14] A. Brito, “Desenvolvimento de jogos com Blender,” in *Blender 3D Jogos e Animações Interativas*, 1st ed., São Paulo: Novatec Editora, 2011, pp. 14–24.
- [15] T. T. Ogusko, “Ferramentas para desenvolver conteúdo imersivo VR,” *Medium*, 2018. <https://medium.com/@ogusko/ferramentas-para-desenvolvedor-de-conteudo-xr-e265c2ef72a2> (accessed Feb. 17, 2020).
- [16] Unity, “Unity para RV e RA,” *Unity*, 2020. <https://unity3d.com/pt/unity/features/multiplatform/vr-ar> (accessed Feb. 19, 2020).
- [17] W. Soares, “Qual a melhor: Unity ou Unreal?,” *Crie Seus Jogos*, 2019. <https://www.crieseusjogos.com.br/qual-a-melhor-unity-ou-unreal/> (accessed Feb. 19, 2020).
- [18] Unreal, “Unreal Engine for AR, VR & MR,” *Unreal*, 2020. <https://www.unrealengine.com/en-US/vr> (accessed Feb. 19, 2020).
- [19] K. Plewa, “Battle of software 2020: Blender vs Maya,” *Sculpteo*, 2018. <https://www.sculpteo.com/blog/2018/12/17/battle-of-software-blender-vs-maya/> (accessed Feb. 19, 2020).
- [20] D. F. Lanza, A. J. Lacerda, and A. A. de Souza, “KinectE.R. Desenvolvendo um game educacional com o uso do Kinect,” in *XII SBGames*, 2013, pp. 541–546, [Online]. Available: http://www.sbgames.org/sbgames2013/proceedings/artedesign/AeD_full_Kinect.pdf.
- [21] D. V. Macedo, Y. R. Serpa, Y. R. Serpa, A. P. Abreu, and M. A. F. Rodrigues, “IntEducaTrânsito: um jogo 3D interativo e educativo sobre as normas de trânsito controlado por dispositivos não tradicionais,” in *XII SBGames*, 2013, pp. 39–47, [Online]. Available: <http://www.sbgames.org/sbgames2013/proceedings/artedesign/06-dt-paper.pdf>.
- [22] J. B. Alves, N. B. Haydu, and R. M. de Souza, “O uso de simuladores para treinamento em áreas de alta periculosidade – Case Simulador de Guindastes Petrobras.,” in *IX SBGames*, 2010, pp. 161–169, [Online]. Available: <http://www.sbgames.org/papers/sbgames10/culture/full/full118.pdf>.
- [23] R. L. Percival, “Oculus Rift S: Stare down the competition,” *Design Listicle*, 2019. <https://www.designlisticle.com/oculus-rift-s-stare-down-the-competition/> (accessed Jun. 08, 2020).
- [24] G. Lynch, “Oculus Rift S review,” *Techradar*, 2019. <https://www.techradar.com/reviews/oculus-rift-s-review> (accessed Jun. 08, 2020).
- [25] Facebook, “Especificações de sistema e requisitos mínimos do Oculus Rift e do Rift S,” 2020. https://support.oculus.com/248749509016567/?locale=pt_PT (accessed Jun. 08, 2020).
- [26] Adobe, “Fuse,” *Adobe*, 2020. <https://www.adobe.com/br/products/fuse.html> (accessed May. 08, 2020).
- [27] K. Shah, “Building A Basic Low Poly Character Rig In Blender,” *Envato Tuts+*, 2012. <https://cgi.tutsplus.com/tutorials/building-a-basic-low-poly-character-rig-in-blender--cg-16955> (accessed May. 13, 2020).
- [28] Adobe, “Home,” *Mixamo*, 2020. <https://www.mixamo.com/#/> (accessed May. 08, 2020).
- [29] R. C. C. Dionysio, N. S. Tavares, and C. I. dos Santos Júnior, “A Linguagem C#,” in *Introdução a Linguagem C# - Conceitos Básicos*, 1st ed., Taquaritinga: AgBook, 2013.
- [30] J. C. Macoratti, “Visual Studio Code – Apresentando o editor multiplataforma da Microsoft,” *iMasters*, 2016. <https://imasters.com.br/desenvolvimento/visual-studio-code-apresentando-o-editor-multiplataforma-da-microsoft> (accessed Apr. 29, 2020).