

AQUÁRIO VIRTUAL: MULTIPLAYER E REALIDADE VIRTUAL

Matheus Waltrich Da Silva, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação

Departamento de Sistemas e Computação

Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

matheuswaltrich@gmail.com, dalton@furb.br

Resumo: Este artigo apresenta o desenvolvimento do projeto de extensão do projeto Aquário Virtual (LOSADA, 2019), adicionando as funcionalidades de multiplayer e realidade virtual. Este projeto foi desenvolvido na plataforma Unity3D, utilizando a biblioteca mirror, para o multiplayer. Os objetivos de projeto foram atingidos, sendo que o multiplayer e a realidade virtual funcionam corretamente em conjunto. Com um dispositivo executando o ambiente do aquário sendo o host e os demais dispositivos executando o movimento de realidade virtual como clientes. Neste projeto, também foi realizada a sincronização de uma câmera do dispositivo host para os dispositivos clientes, sendo necessário fazer conversão da imagem do host em um array de bytes para sincronização deste array para os clientes. Extensões também são existentes, uma vez que foi desenvolvido a funcionalidade multiplayer e realidade virtual, sendo possível incrementar novas funcionalidades e melhorias nestas duas funcionalidades.

Palavras-chave: Aquário virtual. Multiplayer. Realidade Virtual.

1 INTRODUÇÃO

Para Almeida (2004, p.4) a inserção das Tecnologias da Informação e Comunicação (TICs) na educação oportuniza romper com as paredes da sala de aula e da escola, integrando-a à comunidade que a cerca, à sociedade da informação e a outros espaços produtores de conhecimento. Tomé (2013) salienta que as tecnologias são imprescindíveis no aprendizado infantil, pois desenvolvem diferenciadas e ricas estratégias, fazendo com que os educandos aprendam de forma lúdica, dinâmica e prazerosa, respeitando suas limitações e individualidades. Um destes recursos da tecnologia da informação que vem se tornando cada vez mais presente na sala de aula é a realidade virtual.

A tecnologia de Realidade Virtual (RV) vem se tornando bastante popular e acessível sendo natural, dado o seu potencial como ferramenta didática, sua crescente introdução nas salas de aula tradicionais e virtuais. (QUEIROZ; TORI; NASCIMENTO, 2017, p. 1).

Uma forma de se utilizar a realidade virtual é através de simulações, que é bastante utilizada no âmbito educacional. Também é possível utilizar a realidade virtual com Interface de Usuário Tangível (IUT - do Inglês, TUI - Tangible User Interface), como pode-se observar no trabalho Aquário Virtual: Simulador de Ecossistema (LOSADA, 2019). Neste trabalho, o usuário pode utilizar sensores para realizar a alteração de parâmetros do ecossistema do aquário, como temperatura, luminosidade e a alimentação dos peixes. O comportamento dos peixes do simulador do aquário é realizado de forma autônoma respeitando algumas regras básicas da simulação do ecossistema.

A presente proposta constituiu em criar uma extensão do trabalho Aquário Virtual: Simulador de Ecossistema (LOSADA, 2019) utilizando o conceito de avatar, de forma que o usuário possa ter a visão de dentro do aquário. Será utilizada a tecnologia de cardboard, uma plataforma de realidade virtual desenvolvida pela Google com uma montagem de cabeça para um smartphone. Fazendo com que o usuário tenha uma percepção maior de como as ações no ambiente impactam na vida dos seres do aquário. Terá disponível em conjunto a tecnologia de cardboard a opção de multiplayer, possibilitando que vários usuários possam utilizar o mesmo aquário.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas as seguintes fundamentações: realidade virtual, multiplayer, bem como as bibliotecas utilizadas no desenvolvimento do trabalho.

2.1 REALIDADE VIRTUAL

A realidade virtual é uma tecnologia de interface que engana os sentidos do usuário, através de um ambiente virtual criado induzindo efeitos visuais e sonoros. Conforme Queiroz (2017) o potencial de aplicação da realidade virtual na educação vem sendo exaltado desde sua pré-história quando nem mesmo tinha essa denominação. A realidade virtual possibilita ao usuário o sentimento de estar presente no ambiente virtual. Para Baierle e Gluz (2017) a evolução que levou ao surgimento de tecnologias como a Realidade Virtual, leva a cada dia mais pessoas em diferentes áreas de estudo a experimentar simulações tridimensionais devido a sua semelhança física com o mundo real. Conforme Sgobbi et. al (2014), a utilização de tecnologias imersivas permite uma presença virtual na situação de ensino-aprendizagem, garantindo um maior aproveitamento nas atividades propostas.

Uma das formas de utilizar a realidade virtual é através do Head Mounted Display (HMD). O HMD é um dispositivo utilizado em um suporte na cabeça ou como parte de um capacete, que possui um pequeno display óptico em frente aos olhos do usuário (Monocular) ou de cada olho (Binocular). Uma forma de criar uma aplicação no Unity utilizando o HMD é com o Cardboard, conforme Google (2019).

Cardboard é um aplicativo de realidade virtual, desenvolvido pela Google, que pode ser utilizado com um suporte de cabeça para o smartphone, para que o usuário tenha a visão do ambiente virtual de forma imersiva. Nele o usuário pode montar o seu próprio visor com base nas instruções contidas no site da Google ou comprar um visor pronto.

2.2 MULTIPLAYER

A utilização multiplayer é permitir que vários usuários participem simultaneamente de uma mesma partida em um mesmo ambiente virtual simulado. Segundo Luis et al (2016) a utilização do recurso de multiplayer pode auxiliar a interação entre os alunos e assim facilitar a relação deles com as atividades. Possibilita assim uma forma diferente de uso onde todos os alunos possam participar simultaneamente.

No motor de jogos Unity é possível desenvolver jogos multiplayer, com recursos que auxiliam o desenvolvimento do jogo. O Unity permite desenvolver jogos com baixa latência, prevenção de trapaceiras e criação de partidas multiplayer, conforme UNITY (2019).

O Unity também fornece exemplos de jogos desenvolvidos para utilização multiplayer, para auxiliar o desenvolvedor no processo de implementação do jogo. Possui recursos para hospedagem de servidores para jogos online, recurso para comunicação de bate-papo tanto por voz quanto por texto e criação de partidas em grupo. Auxiliando o desenvolvimento de jogos multiplayer na ferramenta.

2.3 VERSÃO ANTERIOR DO SISTEMA/FERRAMENTA/BIBLIOTECA/Framework

Na versão anterior do sistema tem-se o aquário virtual desenvolvido na plataforma Unity3D utilizando a linguagem de programação C# para implementação de comportamentos (LOSADA, 2019). O simulador foi desenvolvido utilizando o asset AIFishes (UNITY, 2019) como base, obtendo assim os peixes em 3D com alguns comportamentos padrões. Também utiliza o recurso de Interface de Usuário Tangível (IUT), para interação com o aquário virtual.

Na Figura 5 é apresentado o resultado do trabalho Losada (2019) com os componentes da interface IUT. Na Figura 7 consta um botão para realizar a alimentação dos peixes, um sensor para verificar a luz do ambiente externo, leds de indicação e um potenciômetro para controle da temperatura.

Figura 7 - Aquário Virtual

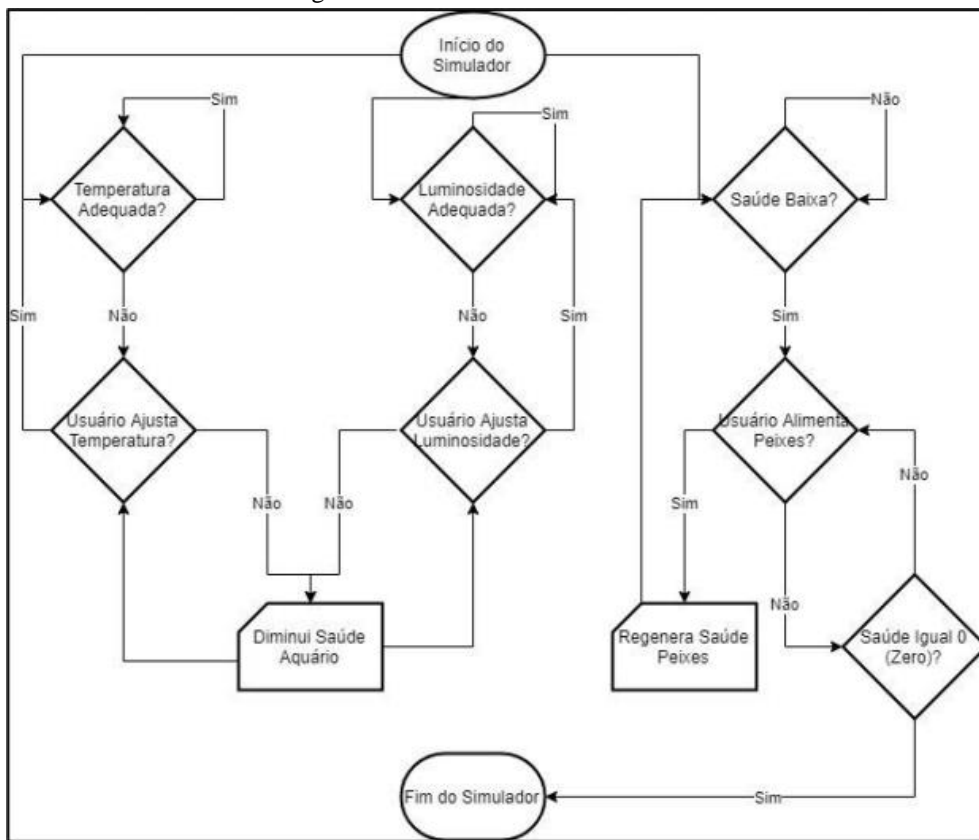


Fonte: Losada (2019).

No simulador consta a saúde dos peixes, aquecedor/resfriador, clima externo, hora, luminosidade, comidas e termômetro. Esses indicadores são utilizados para verificar a saúde dos peixes e o ambiente do aquário virtual.

Na Figura 8 tem-se o ciclo de vida do ecossistema do aquário virtual. A simulação inicia em Início do simulador, verificando a temperatura, luminosidade e saúde dos peixes.

Figura 8 - Ciclo de vida do ecossistema



Fonte: Losada (2019).

2.4 TRABALHOS CORRELATOS

Serão apresentados três trabalhos que foram criadas aplicações em realidade virtual com o foco na educação. O primeiro trabalho é Metáforas para o Ensino de Ciências em Ambientes de Realidade Virtual (SILVA; PIO, 2017), que consiste em utilizar a realidade virtual no ensino de ciências. O segundo trabalho Watt: Imersão 3D Compartilhada e Acessível na Realidade Virtual do Surgimento da Revolução Industrial (BAIERLE; GLUZ, 2017) que utiliza a realidade virtual para apresentar o ambiente da revolução industrial. O terceiro trabalho é VRCircuit: Realidade Virtual aplicada ao Ensino de Circuitos Elétricos (SANCHES et al., 2017) que utiliza a realidade virtual para ensinar o funcionamento de circuitos elétricos.

2.4.1 METÁFORAS PARA O ENSINO DE CIÊNCIAS EM AMBIENTES DE REALIDADE VIRTUAL

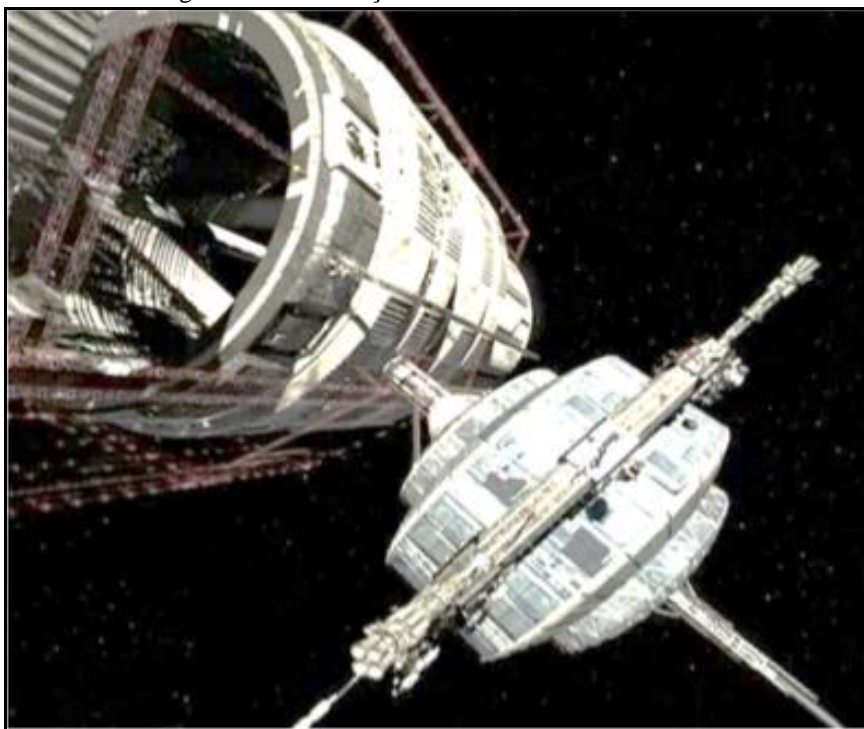
Neste trabalho Silva e Pio (2017) utilizam metáforas no ambiente de realidade virtual para o ensino de ciências. De acordo com Silva e Pio (2017), metáfora é a técnica de colocar um símbolo no lugar de outro com a finalidade de promover a comunicação efetiva e gerar interação com domínios distintos. O trabalho utilizou o jogo The Expanse VR, cujo enredo é um suspense de ficção científica em que os seres humanos colonizaram Marte e os cintos de asteroides externos. Utilizou o dispositivo de realidade virtual Gear Reality Virtual Headset da Samsung disponível na plataforma Android para o Smartphone Samsung modelo S7, para aumentar a imersão do usuário no mundo virtual. Na Figura 1 tem-se a imagem que apresenta uma aluna imersa no ambiente virtual e na Figura 2 o ambiente virtual utilizado. O The Expanse VR é um jogo que o jogador utiliza a realidade virtual para explorar Marte muitos anos no futuro, em meio a uma guerra entre Marte e a Terra.

Figura 1 - Aluna imersa no ambiente Gear Reality Virtual durante a aplicação experimental



Fonte: Silva e Pio (2017).

Figura 2 - Visualização do ambiente virtual utilizado



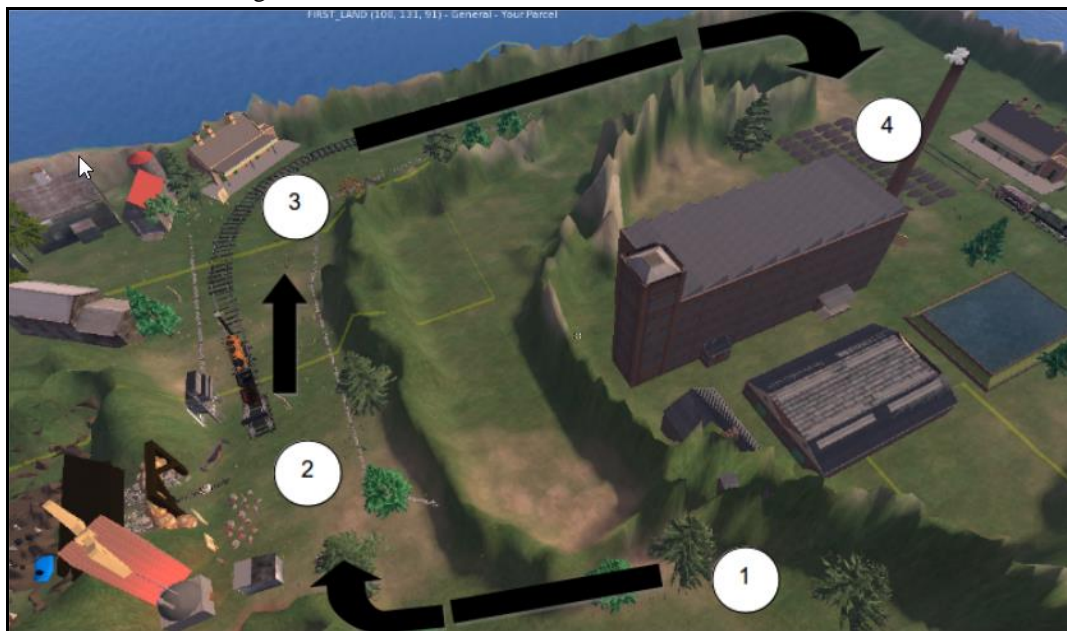
Fonte: Silva e Pio (2017).

Conforme Silva e Pio (2017), foi realizada a aplicação do trabalho em uma turma com 36 alunos para a imersão na realidade virtual. Segundo Silva e Pio (2017), para a construção das metáforas foi utilizado um conjunto de 4 procedimentos: observação do ambiente real, identificação das características do ambiente real que possam ser traduzidas no ambiente virtual, tradução das convenções do contexto real para o ambiente virtual e montagem das metáforas. No trabalho foram utilizadas três metáforas: a metáfora do ambiente, que representa objetos, cenários e características do ambiente; as metáforas de ensino, que traduzem como as informações serão apresentadas aos alunos; a metáfora de aprendizagem, que representa as relações que o ambiente real possui com o ambiente virtual.

2.4.2 WATT: IMERSÃO 3D COMPARTILHADA E ACESSÍVEL NA REALIDADE VIRTUAL DO SURGIMENTO DA REVOLUÇÃO INDUSTRIAL

Neste trabalho, Baierle e Gluz (2017) realizaram o desenvolvimento de um mundo virtual chamado WATT que se apresenta como uma alternativa de ensinar História, ajudando no ensino a respeito dos processos sociais, econômicos, científicos e tecnológicos durante o surgimento da Revolução Industrial, os jogadores podem imergir em um mundo virtual da revolução industrial de forma multijogador. Como se pode observar na Figura 3 apresenta-se o mundo virtual WATT desenvolvido no trabalho. Na Figura 3 o número 1 representa a cena de entrada logo após o login no sistema WATT. O número 2 seria a segunda cena do ambiente virtual do WATT, apresentando um protótipo de motor criado na revolução industrial. O número 3 é a terceira cena que apresenta um pequeno vilarejo com uma estação ferroviária. A quarta e última cena apresenta as tecelagens da indústria têxtil com a mecanização.

Figura 3 - Vista aérea do cenário do mundo virtual WATT

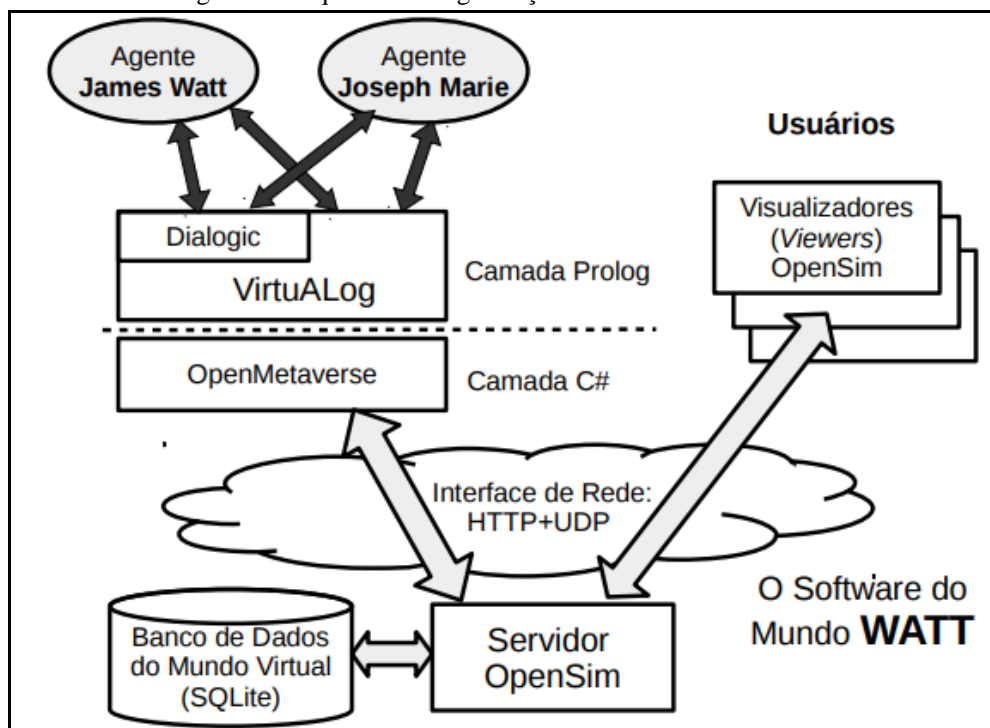


Fonte: Baierle e Gluz (2017).

Conforme Baierle e Gluz (2017), foram realizados experimentos em laboratório e com voluntários. Nestes experimentos evidenciou-se que o emprego do WATT pode incrementar a qualidade do ensino de história. Segundo Baierle e Gluz (2017) para a realização dos testes foi aplicado um pré-teste, que a média de acertos antes da simulação foi de 37,85% e após a simulação foi de 67,14%.

Segundo Baierle e Gluz (2017), para o desenvolvimento do WATT foi utilizado o simulador de mundos virtuais 3D OpenSim e o ambiente de programação de agentes inteligentes VirtuALog. O OpenSim possui integração com vários bancos de dados; para o desenvolvimento do WATT foi utilizado o banco de dados SQLite. A interface de comunicação do servidor OpenSim com a rede é realizada via HTTP e UDP. O gerenciamento dos personagens não jogáveis (do Inglês, NPC - Non-player characters) no mundo virtual WATT fica a cargo do VirtuALog, em um ambiente de programação Prolog que usa a biblioteca OpenMetaverse. Na Figura 4 pode-se observar a arquitetura do mundo virtual WATT. Nela temos os agentes que interagem com o ambiente virtual pelo VirtuALog, que envia estas informações para o servidor do OpenSim que grava no banco de dados.

Figura 4 - Arquitetura e organização do mundo virtual WATT



Fonte: Baierle e Gluz (2017).

2.4.3 VRCIRCUIT: REALIDADE VIRTUAL APLICADA AO ENSINO DE CIRCUITOS ELÉTRICOS

Neste trabalho, Sanches et al. (2017) realizaram o desenvolvimento de ambiente virtual para o ensino do funcionamento de circuitos elétricos. Para o desenvolvimento do ambiente virtual foi utilizada a ferramenta Unity 3D na versão 5.5.2. Sanches et al. (2017) realizaram o desenvolvimento deste jogo com o objetivo de criar uma abordagem mais lúdica e diferente do ensino tradicional de circuitos eletrônicos. O jogador tem uma visão em primeira pessoa do ambiente Virtual. O jogo inicia com o usuário, que pode mudar de acordo com a dificuldade definida pelo jogador. Cada cenário tem uma proposta de circuito elétrico para estudo. No início da fase são apresentados os problemas que devem ser corrigidos pelo jogador, sendo utilizados os conceitos de física que foram estudados em sala. O personagem pode ser controlado pelo mouse ou por um controle, podendo se movimentar e rotacionar pelo cenário, consultar a todo instante materiais com conceitos de física. No cenário há ferramentas que o jogador pode interagir para solucionar os problemas de circuitos eletrônicos, conforme pode-se observar na Figura 5.

Figura 5 – Ferramenta do jogo



Fonte: Sanches et al. (2017).

Conforme pode-se observar na Figura 5 todos os instrumentos que possuem interação com o usuário, contém a cor azul, para facilitar a sua identificação por parte do jogador. Estes objetos quando selecionados indicam uma instrução. Segundo Sanches et al. (2017), para realizar a correção, o jogador deve observar qual é o tipo do circuito elétrico, a tensão da bateria e quais problemas estão afetando o seu funcionamento. Ao final da correção o jogador deve ligar a bateria para verificar se os ajustes foram corretos ou não. Caso as escolhas realizadas não forem corretas serão apresentadas faíscas e mostrará a mensagem de tente novamente (Figura 6).

Figura 6 – Circuito concluído com erros



Fonte: Sanches et al. (2017).

Segundo Sanches et al. (2017) foram aplicados testes em 10 indivíduos com idade entre 18 e 24 anos que estão cursando a disciplina de Física Elétrica do terceiro período de bacharelado em Ciência da Computação. Conforme Sanches et al. (2017), após a realização dos testes foi aplicado um questionário individual para cada aluno. Neste questionário foi apontado que a maioria dos alunos aprovou a utilização da realidade virtual para a aplicação do que foi aprendido em sala de aula de forma prática.

Quadro 1 – Comparativo entre os trabalhos correlatos

Características	Silva, Pio (2017)	Baierle, Gluz (2017)	Sanches et al (2017)
Plataforma	virtual The Expanse VR	OpenSim	Unity 3D
Multiplayer	Não	Sim	Não
Aplicado em sala de aula	Sim	Voluntários	Sim
Contexto Educacional	Ciências	História	Circuitos Elétricos

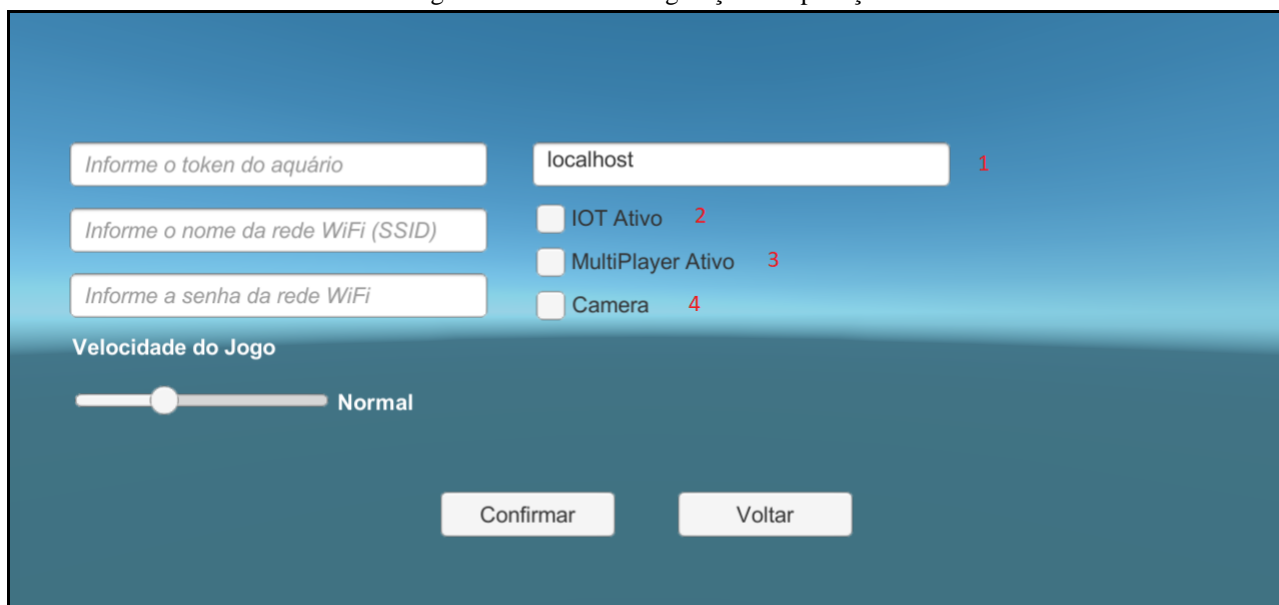
Fonte: elaborado pelo autor.

3 DESCRIÇÃO

Neste capítulo serão apresentadas algumas informações sobre o desenvolvimento do trabalho. Este projeto foi dividido em duas partes, primeiro a realidade virtual e depois o multiplayer.

Para poder compatibilizar o projeto anterior, com as novas funcionalidades, foram adicionadas algumas novas configurações. Na figura 7 abaixo, podemos observar as novas configurações que foram adicionadas ao projeto. O item 1 da figura é um campo que foi adicionado para configurar o ip do host, dispositivo que vai estar rodando o aquário. O item 2 da figura é uma configuração para ativar ou desativar a integração com o IOT, possibilitando ao usuário utilizar a aplicação, mesmo sem o dispositivo IOT. O item 3 da imagem é uma configuração para ativar ou desativar o multiplayer, possibilitando ao usuário utilizar a aplicação, sem o multiplayer. O item 4 da imagem, é uma configuração para ativar a câmera de desenvolvimento, esta configuração inverte o lado do aquário que a câmera é apresentada. Esta configuração foi adicionada, para facilitar os testes durante o desenvolvimento.

Figura 7 – Tela de configuração da aplicação

A tela de configuração da aplicação possui um fundo azul com uma transição de cor. No topo, há três campos de entrada brancos com bordas arredondadas. O primeiro campo contém o texto "Informe o token do aquário". O segundo campo contém "Informe o nome da rede WiFi (SSID)". O terceiro campo contém "Informe a senha da rede WiFi". À direita desses campos, há um campo de entrada branco com o texto "localhost" e um número "1" em vermelho ao seu lado. Abaixo disso, há três opções de configuração: "IOT Ativo" com um checkbox e o número "2" em vermelho; "MultiPlayer Ativo" com um checkbox e o número "3" em vermelho; e "Camera" com um checkbox e o número "4" em vermelho. Abaixo das opções, há um slider para "Velocidade do Jogo" com um botão deslizante e o texto "Normal". No rodapé, há dois botões brancos com bordas arredondadas: "Confirmar" e "Voltar".

Fonte: Elaborado pelo autor.

Na tela inicial da aplicação, foi adicionado um novo botão chamado Jogar RV e alterado o nome do botão original, para jogar aquário. Como podemos ver na figura 8 abaixo. O botão jogar aquário quando desmarcada a opção multiplayer, inicia a mesma aplicação do projeto original, sem o multiplayer. O botão jogar RV, com a opção multiplayer desmarcada, inicia a aplicação no modo de realidade virtual, sem o multiplayer. Com a opção multiplayer ativa, o botão jogar aquário inicia um host, já com um peixe iniciado. O botão jogar RV, inicia um cliente no ip que está configurado no campo server, este cliente utilizando a realidade virtual.

Figura 8 – Tela inicial da Aplicação

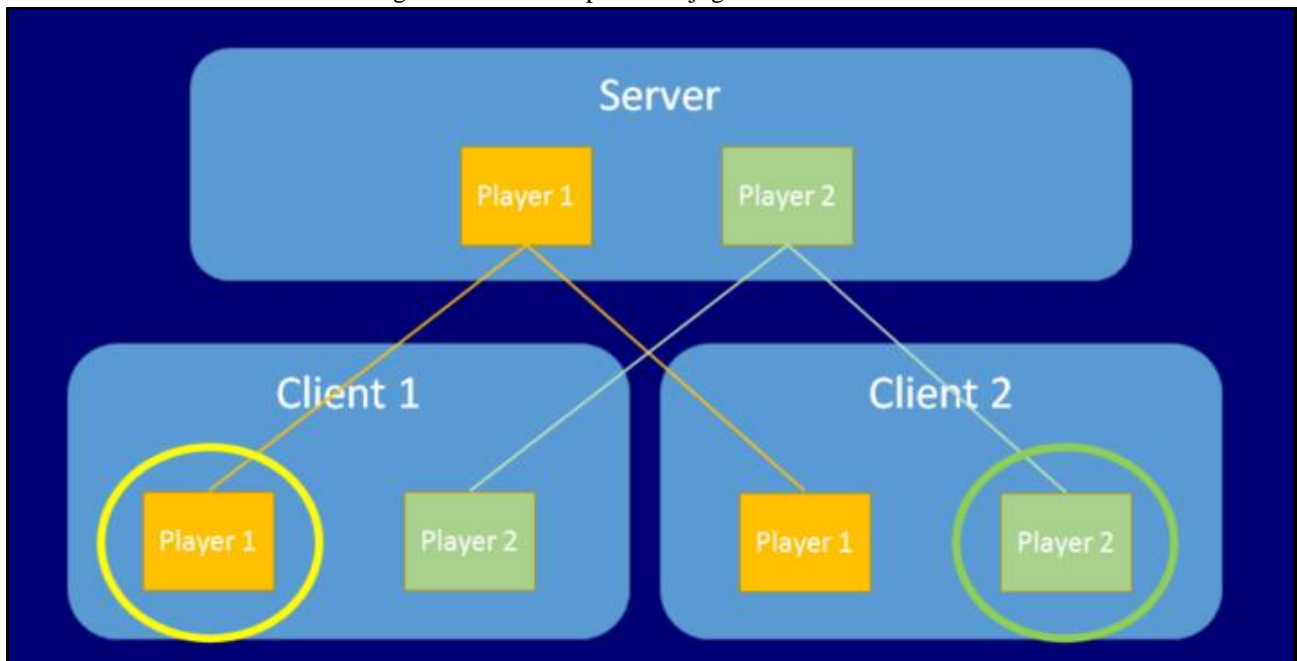


Fonte: Elaborado pelo autor.

Para o desenvolvimento da funcionalidade multiplayer deste trabalho, foi utilizado uma biblioteca gratuita do Unity, chamada Mirror. Esta biblioteca utiliza os próprios recursos multiplayer da Unity, mas torna a utilização destes recursos mais amigável para o desenvolvedor. Mirror é uma API de alto nível para vários jogadores multiplayer. É um sistema para criar recursos para vários jogadores dos jogos Unity.

Esta biblioteca realiza o espelhamento dos players entre os vários clientes. Sendo controlado pelo servidor host. Na figura 9, podemos observar um pouco, como esta arquitetura funciona, com o server, espelhando os dois players nos clientes.

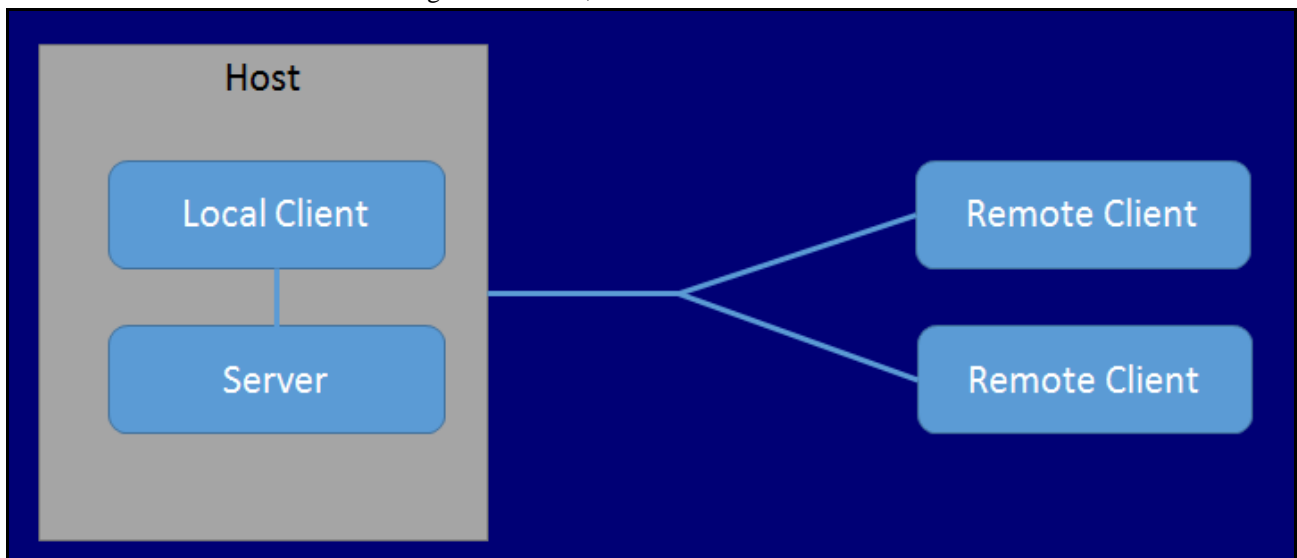
Figura 9 – Server espelhando jogadores nos clientes



Fonte: Mirror Networking. (2020).

Com o mirror também é possível que o server seja um server com um cliente, chamado de host. O host é um server com um cliente local iniciado, juntamente com o server, conforme podemos observar na Figura 10 abaixo. Neste caso a aplicação não possui um servidor dedicado, um dos clientes faz o papel de server também. Neste projeto foi utilizada esta arquitetura, pois o aquário é um host, possui um server e um cliente local.

Figura 10 – Host, server com um cliente local



Fonte: Mirror Networking. (2020).

No desenvolvimento do trabalho, foi necessário realizar o compartilhamento da câmera no dispositivo host para os clientes, mas para isto foi encontrado duas limitações da biblioteca mirror. Primeiro o mirror somente realiza a sincronização da posição do objeto, não da sua textura, e de variáveis primitivas como por exemplo int. Então para poder realizar o compartilhamento da câmera do dispositivo host para os dispositivos cliente, foi necessário a cada frame do host realizar uma captura dos pixels da câmera naquele momento, converter estes em uma imagem e converter a imagem em um array de bytes. Mas o mirror também não sincroniza array, mas permite, criar uma classe para sincronizar dados personalizados. Desta forma foi criada uma classe, para sincronizar o array de bytes da imagem.

Agora com o array de bytes da imagem, temos a segunda limitação da biblioteca mirror, que possui um limite de bytes que pode realizar a sincronização entre o host e os clientes. Para poder atender esta demanda, foi reduzida a resolução da imagem em 75%, para poder ter um array de bytes pequeno o suficiente para se tornar possível a sua sincronização do host para os clientes.

Nos clientes agora com o array de bytes sendo sincronizado é convertido este array novamente em imagem e aplicada esta imagem na textura do objeto. Com isso, fazendo com que a cada frame uma imagem seja sincronizada entre o host e os clientes, dando a impressão para o usuário que é a câmera do host.

4 RESULTADOS

O projeto desenvolvido neste artigo teve os resultados esperados alcançados, sendo realizada a integração entre o multiplayer e a realidade virtual, possibilitando utilizar os dois recursos, juntamente com o projeto já existente do Losada. Os testes do projeto foram realizados no decorrer do desenvolvimento do projeto. Os resultados foram divididos em quatro seções, sendo a seção 4.1 para o módulo de Realidade Virtual, a seção 4.2 para módulo de Multiplayer.

4.1 REALIDADE VIRTUAL

Para o desenvolvimento da realidade virtual no projeto foi adicionado uma câmera no peixe que o segue e foi convertida esta câmera para que seja apresentada utilizando a funcionalidade de realidade virtual. Foi realizada a criação de um script para que a câmera acompanhasse o peixe durante o seu percurso no aquário virtual. Podemos observar no quadro 2 abaixo o código desenvolvido para este fim.

Quadro 2 – Script para que a câmera acompanhe o peixe

```
public GameObject alvo;
public GameObject cameraposicao;

// Update is called once per frame
void Update()
{
    transform.LookAt(alvo.transform);
    transform.position = cameraposicao.transform.position;
}
```

Fonte: Elaborado pelo autor.

Para ativar a câmera em modo de realidade virtual foi utilizada a propriedade XRSettings, da Unity que controla a ativação de desativação da realidade virtual.

4.2 MULTIPLAYER

Conforme já mencionado na descrição do arquivo foi utilizada a biblioteca Mirror. No desenvolvimento do multiplayer foram encontradas dificuldades para realizar a sincronização da câmera do dispositivo host para os dispositivos clientes. Para realizar esta sincronização foi necessário a cada frame converter a imagem da câmera em pixel e salvar em uma imagem separada e após isso converter esta imagem em um array de bytes. Como podemos observar no script do quadro 3 abaixo, são obtidos os pixels da câmera e salvos na variável “photo”. Após isso é reduzida a resolução da imagem em 25% do tamanho original, para poder ter um array de bytes pequeno o suficiente para ser sincronizado. Após isso é convertida a imagem para o array de bytes e este é salvo na variável inventory que é sincronizada entre os clientes do host.

Quadro 3 – Script para capturar a imagem da câmera e converter em array de bytes

```
Texture2D photo = new Texture2D (width, height, TextureFormat.RGB24, false);
photo.SetPixels (backCam.GetPixels ());
photo.Apply ();
TextureScale.Bilinear (photo, width / 4, height / 4);
byte[] t = photo.EncodeToJPG ();
item.back = t;
if (t.LongCount () < 10000) {
    inventory.Clear ();
    inventory.Add (item);
}
```

Fonte: Elaborado pelo autor.

Nos clientes do host foi desenvolvido um script, para converter o array de bytes novamente em imagem. Podendo assim aplicar esta imagem na textura do objeto de parede do aquário gerando o efeito de vidro, possibilitando ver fora do aquário. Conforme podemos observar o script no quadro 4 abaixo, é criada uma variável local chamada também de photo e uma variável local chamada “t”, na variável local “t” é atribuído o array de bytes que foi sincronizado e utilizado este array, para converter novamente a imagem na variável photo. Após convertida a imagem é atribuída a textura do objeto, que está com este script atribuído.

Quadro 4 – Script para converter o array de bytes novamente em imagem.

```
Texture2D photo = new Texture2D (width / 4, height / 4, TextureFormat.RGB24, false);  
byte[] t = inventory[0].back;  
photo.LoadImage (t);  
GetComponent<Renderer> ().material.mainTexture = photo;
```

Fonte: Elaborado pelo autor.

5 CONCLUSÕES

Os resultados esperados foram alcançados no desenvolvimento deste projeto. Houve o desenvolvimento do Aquário Virtual como um simulador de ecossistema com a funcionalidade do multiplayer e da realidade virtual, demonstrando funcionar bem em conjunto.

A biblioteca Mirror se mostrou eficiente para o desenvolvimento de aplicações multiplayer e fácil utilização e aprendizado. Esta biblioteca se mostrou muito poderosa para o desenvolvimento de aplicações multiplayer, pois permite manter sincronizados n clientes com o servidor, contanto que o servidor contenha desempenho para tal. A biblioteca permite sincronizar a animação e posição do player entre os vários clientes, de forma transparente para o desenvolvedor facilitando o desenvolvimento de aplicações multiplayer.

Mesmo que os objetivos do projeto tenham sido alcançados com sucesso, ainda há a possibilidade de extensão. Não foram utilizadas todas as funcionalidades da realidade virtual e multiplayer, pois hoje o usuário apenas acompanha o peixe enquanto nada pelo aquário. Mas, utilizando todas as funcionalidades da realidade virtual é possível fazer o peixe nadar para a direção em que o usuário está olhando, proporcionando uma imersão ainda maior no ambiente virtual do aquário.

REFERÊNCIAS

- ALMEIDA, Maria Elizabeth Bianconcini. Tecnologia de Informação e Comunicação na Escola: novos horizontes na produção escrita (2004). Disponível em: http://www.eadconsultoria.com.br/matapoio/biblioteca/textos_pdf/texto24.pdf . Acesso em: 12 de set. 2019.
- BAIERLE, Ivan Luis Feix, Lopes et al.; GLUZ, João. Watt: Imersão 3D Compartilhada e Acessível na Realidade Virtual do Surgimento da Revolução Industrial Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 585, out. 2017. ISSN 2316-6533. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/7587/5383>. Acesso em: 12 set. 2019.
- CARDBOARD, Google. 2019. Disponível em: https://arvr.google.com/intl/pt-BR_pt/cardboard/. Acesso em: 18 out. 2019.
- LOSADA, Flávio Omar. Aquário Virtual: Simulador De Ecossistema. 2019. 19 pg. Universidade Regional de Blumenau, Blumenau, 2019.
- LUIS, Edwyn et al. Desenvolvimento e avaliação de um jogo multiplayer voltado à prática de atividades em sala de aula. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 647, nov. 2016. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/6746/4633>. Acesso em: 12 set. 2019.
- MIRROR, Mirror. 2020. Disponível em: <https://mirror-networking.com/docs/General/index.html>. Acesso em: 20 jan. 2020.
- QUEIROZ, Anna Carolina; TORI, Romero; NASCIMENTO, Alexandre. Realidade Virtual na Educação: Panorama das Pesquisas no Brasil. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 203, out. 2017. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/7549/5345>. Acesso em: 12 set. 2019.
- SANCHES, Pablo; FAÊDA, Leonardo; MACHADO, Alex. VRCircuit: Realidade virtual aplicada ao ensino de circuitos elétricos. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 887, out. 2017. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/7617/5413>. Acesso em: 18 out. 2019.
- SGOBBI, Fabiana Santiago et al. Interação com artefatos e personagens artificiais em mundos virtuais. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 642, nov. 2014. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/2994/2505>. Acesso em: 12 set. 2019.
- SILVA, Simon Jeferson Silva e; PIO, José Luiz. Metáforas para o Ensino de Ciências em Ambientes de Realidade Virtual. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 725, out. 2017. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/7601/5397>. Acesso em: 12 set. 2019.
- TOMÉ, Thalita. A importância da Tecnologia no aprendizado infantil. DINO - Divulgador de notícias. 2013. Disponível em: <https://www.dino.com.br/releases/a-importancia-da-tecnologia-no-aprendizado-infantil-dino8902670131>. Acesso em: 12 set. 2019.
- UNITY. Unity AssetStore. 2019. Disponível em: <https://assetstore.unity.com/packages/3d/characters/animals/ai-fishes-readysolutions-for-your-project-124645>. Acesso em: 12 set. 2019.