ROTAVR: JOGO DE CONSCIENTIZAÇÃO NO TRÂNSITO UTILIZANDO A REALIDADE VIRTUAL IMERSIVA

Guilherme Fibrantz, 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

gfibrantz@furb.br, dalton@furb.br

**Resumo:** O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

**Palavras-chave**: Realidade virtual. Trânsito. Jogos educativos. Segurança viária.

# Introdução

O trânsito é um tema de grande importância na sociedade contemporânea. O movimento de veículos e pedestres nas vias públicas é um fator que impacta diversos aspectos da vida urbana, seja de maneira direta ou indireta. Como apontado por Miranda (2016), a educação para o trânsito desempenha um papel fundamental na redução dos conflitos envolvendo pedestres, passageiros e condutores de veículos. Afinal, segundo a Organização Mundial da Saúde (2021), os óbitos e os ferimentos resultantes de acidentes de trânsito representam uma das maiores ameaças ao progresso sustentável das nações. “Em todo o mundo, os sinistros de trânsito causam aproximadamente 1,3 milhão de mortes evitáveis e cerca de 50 milhões de feridos por ano – tornando-os a principal causa de mortes de crianças e jovens em todo o mundo.” (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2021, p. 6).

Uma das estratégias mais eficazes de modificar esta situação é a mudança do comportamento dos cidadãos, influindo no processo de segurança de todos. Conforme Miranda (2016), a maneira como os cidadãos se comportam no trânsito tem um impacto direto na segurança de todos os envolvidos. Nesse sentido, é fundamental que as pessoas possuam uma base ética e moral sólida, embora muitas vezes esses valores não sejam devidamente cultivados no ambiente familiar. Portanto, é no contexto escolar que se torna essencial a construção de princípios fundamentais para uma vida tanto individual quanto coletiva.

O caso do Brasil não é diferente. Como assinala Araujo Junior (2019), no cenário nacional lamentavelmente se evidencia a ideia de observar as normas de trânsito apenas para evitar multas. Por isto, é preciso implementar ações educativas positivas, sendo fundamental buscar um trânsito onde os condutores estejam plenamente conscientes da importância de obedecer às normas de trânsito, reconhecendo os benefícios positivos que essa atitude implica.

Diversas abordagens podem ser utilizadas para a incluir a educação para o trânsito em sala de aula, dentre as quais figuram em destaque os jogos digitais. Conforme indica Prensky (2001), os jogos não são o único caminho para melhorar o contexto educacional. Porém, os jogos eletrônicos são um dos elementos necessários para instigar as novas gerações a se interessarem pelos temas abordados em sala de aula. A possibilidade de interagir em um ambiente lúdico e controlar as situações vivenciadas, bem como a perspectiva de avançar para níveis mais altos conforme seu desempenho, é capaz de gerar motivação no aluno por meio dos jogos eletrônicos (DA SILVA *et al.*, 2014).

A fim de aprimorar a experiência do usuário e estimular sua interação, é possível empregar tecnologias que melhoram a interface entre o usuário e o sistema, como a Realidade Virtual (RV). Segundo Braga (2001), a realidade virtual oferece a possibilidade de experimentar situações perigosas e difíceis, muitas vezes não sendo possíveis de serem vivenciadas, além de permitir que sejam abordados temas que apresentam desafios de aprendizagem e que podem ser impossíveis de serem demonstrados na vida real. Conforme Tori, Hounsell e Kirner (2020), a sensação de imersão em ambientes virtuais e a capacidade de interagir com seus detalhes resultam em um aumento significativo do interesse dos alunos pelo conteúdo, além de promover uma melhoria em seu processo de aprendizagem.

Diante deste contexto, o presente trabalho desenvolveu um jogo para o ensino da educação no trânsito com o emprego da realidade virtual imersiva, na qual os indivíduos são incentivados a agir de maneira responsável e a adotar comportamentos conscientes ​​e seguros. O objetivo principal deste trabalho é o desenvolvimento de um jogo para dispositivos móveis que utiliza a realidade virtual imersiva para o ensino da educação no trânsito. Os objetivos específicos são: disponibilizar uma forma alternativa de mostrar conteúdos relacionados à educação no trânsito; realizar testes piloto para avaliar a experiência dos usuários com o jogo e disponibilizar diferentes cenários virtuais de trânsito, cada um apresentando desafios distintos.

# FUNDAMENTAÇÃO TEÓRICA

Esta seção é dedicada a expor bases e conceitos dos tópicos que são essenciais para apoiar o jogo desenvolvido. A subseção 2.1 se destina a comentar sobre a educação no trânsito. A subseção 2.2 se destina a fundamentar os jogos educativos. Na subseção 2.3 são abordados conceitos de Realidade Virtual. Por fim, a subseção 2.4 irá expor os trabalhos correlatos.

## EDUCAÇÃO NO TRÂNSITO

É evidente que o trânsito é um elemento constante em nossas vidas, contudo, a relevância atribuída a essa questão frequentemente não é destacada. Relatos sobre acidentes de trânsito são comuns na televisão, e há uma percepção de aumento no número de vítimas fatais. Apesar disso, pouco parece mudar em termos de ações ou políticas para abordar essa situação. (DA SILVA, 2017). “A insegurança no trânsito é um problema mundial crescente e alarmante. Ainda que muitos países se esforcem para reduzir a quantidade de acidentes, eles são hoje uma das maiores causas de óbitos no mundo, tirando a vida de mais de 1,3 milhão de pessoas por ano.” (AMBEV; FALCONI; ONSV, 2014, p. 11).

De acordo com Miranda (2016), é fundamental que o comportamento do cidadão no trânsito seja adequado, pois isso influencia diretamente na segurança de todos. Segundo Miranda (2016), para garantir a harmonia nesse contexto, é necessário que as pessoas tenham formação ética e moral, sendo o ambiente escolar um local significativo, onde surgem princípios importantes para a vida individual e coletiva. Como comenta da Silva (2017), um aspecto notável é a forte conexão entre as escolas e as comunidades onde estão situadas. O conhecimento obtido na escola frequentemente se estende além dos estudantes, alcançando suas famílias, vizinhos e, em muitos casos, se difundindo por toda a comunidade à qual pertencem.

Segundo Bonfim *et al.* (2017), é fundamental investir na educação para o trânsito desde a infância, pois isso proporciona ao indivíduo o conhecimento necessário para agir com autonomia, contribuindo assim para a melhoria da qualidade do trânsito na sociedade em que está inserido. Segundo do Nascimento (2021, p. 73), “[...] a sala de aula é uma ferramenta importante e, essencial, para a formação de cidadãos aptos a conviver em uma época de descontrole populacional relacionado a um trânsito de carros e pessoas cada vez mais denso, principalmente, nas grandes metrópoles”.

## JOGOS EDUCATIVOS

A utilização de jogos digitais na aprendizagem possui um grande potencial, como citado por Carvalho (2018), que comenta que jogos educacionais têm o potencial de ampliar as oportunidades de aprendizagem, ao mesmo tempo em que contribuem para a construção da autoconfiança e estimulam a motivação no contexto educacional. Segundo Falkembach (2006), os jogos educacionais adotam uma abordagem pedagógica que valoriza a exploração livre e o aspecto lúdico, o que estimula o aprendizado do jogador. Falkembach (2006) comenta que além disso, os jogos são capazes de contribuir para a construção da autoconfiança e aumentar a motivação no contexto da aprendizagem.

De acordo com Carvalho (2018), é importante considerar dois aspectos essenciais ao criar um jogo educacional: a capacidade de transmitir a mensagem educativa de maneira eficiente e satisfatória, ou seja, ensinar de fato, e a capacidade de tornar a atividade atraente e prazerosa, o que é conhecido como fator entretenimento. Carvalho (2018) aponta que quando esses dois fatores são alcançados, o jogo pode ser uma ferramenta eficaz para a aprendizagem e cumprir o seu papel com sucesso. Conforme da Silva *et al.* (2014), na abordagem de jogos digitais educativos, é importante adaptar os conceitos que tornam os jogos populares bem-sucedidos, integrando-os ao processo educativo para aproveitar seu potencial de engajamento e aprendizado.

## REALIDADE VIRTUAL

De acordo com Kirner e Kirner (2011, p. 14), a “[...] realidade virtual é uma interface computacional que permite ao usuário interagir em tempo real, em um espaço tridimensional gerado por computador, usando seus sentidos, através de dispositivos especiais”. Além disso, como indica Roussou (2004), a RV em geral, é amplamente utilizada nas áreas de educação e treinamento devido às suas potencialidades estimulando a interação e motivação. Bowman e McMahan (2007, p. 36, tradução nossa) definem as tecnologias de RV como “[...] tecnologias complexas que [substituem] informações sensoriais do mundo real por estímulos sintéticos, como imagens visuais 3D, sons espaciais ou força ou feedback tátil”.

Tori, Hounsell e Kirner (2020) comentam que é notável que a RV possui um vasto potencial de aplicações, pois permite a imersão em uma ampla variedade de experiências do mundo real, além de possibilitar a criação de cenários imaginários, tudo isso a um custo acessível e sem riscos significativos. Como apresentado por Tori, Hounsell e Kirner (2020), a expansão do uso de aplicativos em smartphones tem desencadeado uma revolução na maneira como interagimos com a tecnologia, especialmente no campo da visualização 3D. Com o advento de dispositivos como os Cardboards, tornou-se possível experimentar realidades imersivas e visualizações tridimensionais de maneira acessível e conveniente, usando apenas o celular.

Equipamentos de realidade virtual mais avançados do que o Cardboard, como o Gear VR, ambos exemplificados na Figura 1, incorporam funcionalidades adicionais que enriquecem a experiência do usuário. Estes dispositivos são equipados com botões específicos que facilitam a navegação e interação com o ambiente virtual. Além disso, alguns modelos vêm acompanhados de um controle remoto, proporcionando uma experiência de interação mais confortável e intuitiva para os usuários. (TORI; HOUNSELL; KIRNER, 2020).

Figura 1 – Equipamentos de realidade virtual

Tela de um aparelho celular

Descrição gerada automaticamente com confiança baixa

Fonte: Tori, Hounsell e Kirner (2020, p. 101).

## trabalhos correlatos

A seguir, são expostos estudos relacionados que tratam de temas de RV no âmbito da educação para o trânsito. O primeiro, mostrado no Quadro 1, é um jogo educacional denominado Induca, que busca promover e facilitar o aprendizado sobre segurança no trânsito para crianças nos primeiros anos do ensino fundamental (Santana; Tronto; Sousa, 2018). O segundo trabalho, desenvolvido por Zeuwts *et al.* (2023), e detalhado no Quadro 2, tem como principal objetivo testar a eficácia de uma nova ferramenta de realidade virtual para promover a segurança no ciclismo para crianças. Por último, o Quadro 3 apresenta o trabalho de Santos *et al.* (2019), um jogo 2D que aborda conceitos referentes ao trânsito, visando a atuação do jogador no trânsito incluindo várias regras e condições que eles experimentarão na prática.

O trabalho de Santana, Tronto e Sousa (2018) foca no desenvolvimento de um jogo educativo para ensinar sobre o trânsito. Projetado para ser interativo e envolvente, o jogo utiliza a Engine Unity3D, e busca melhorar a consciência sobre segurança no trânsito e incentivar comportamentos responsáveis em motoristas e pedestres. Este projeto atende à crescente demanda por maior foco na educação sobre trânsito no contexto educacional escolar.

Quadro 1 – Trabalho Correlato 1

|  |  |
| --- | --- |
| Referência | Santana, Tronto e Sousa (2018). |
| Objetivos | Criar um jogo que objetiva entender as funções e sinais no trânsito, aprender a se comportar no carro e seguir as leis de trânsito, e perceber a importância de ser responsável e gentil ao dirigir. Para isso, o jogador precisa superar todos os desafios e obstáculos presentes no cenário. |
| Principais funcionalidades | O jogador encontra diferentes missões espalhadas pela cidade em um ambiente tridimensional, que envolvem elementos como faixas de pedestres, motoristas, semáforos e leis de trânsito. O jogo utiliza o teclado e mouse para movimentação, oferecendo uma jogabilidade interativa e envolvente, além de um sistema de pontuação que penaliza o jogador por infrações de trânsito e recompensas por comportamentos corretos. Além disso, é dividido em diferentes fases, cada uma abordando um aspecto específico da educação no trânsito. |
| Ferramentas de desenvolvimento | Foi utilizada a Engine Unity3D e a ferramenta CorelDraw para a criação das imagens bidimensionais. |
| Resultados e conclusões | Os autores comentam que a educação sobre trânsito, embora tenha uma relevância significativa, continua sendo um tópico que recebe pouca atenção no contexto escolar. Os autores observaram que processo de criação de um jogo se difere da criação de um software convencional, visto que antes de começar o desenvolvimento de um jogo, é essencial ter uma documentação clara e ideias bem estruturadas sobre os objetivos e métodos a serem empregados. Apesar de não ter sido testado na rede de escolas, os autores apontaram que o jogo pode usado como um recurso valioso no processo de aprendizagem. |

Fonte: elaborado pelo autor.

O estudo de Zeuwts *et al.* (2023) apresenta um teste de RV para jovens ciclistas, usando a Engine Unity3D e rastreamento ocular. Com 130 estudantes participantes, o projeto une realidade virtual e um simulador de ciclismo, oferecendo uma abordagem inovadora para a educação em segurança no ciclismo infantil. A experiência de RV é enriquecida com o óculos HTC Vive e rastreamento ocular da Pupil Labs, além do uso de uma bicicleta em quadro estacionário que permite avaliações detalhadas do comportamento dos ciclistas. O ambiente de execução dos testes pode ser visto na Figura 2.

Figura 2 – Ambiente de teste composto pela bicicleta e óculos com complemento ocular utilizado

Uma imagem contendo no interior, mesa, mulher, homem

Descrição gerada automaticamente

Fonte: Zeuwts *et al.* (2023, p. 4).

Quadro 2 – Using an immersive virtual reality bicycle simulator to evaluate hazard detection and anticipation of overt and covert traffic situations in young bicyclists

|  |  |
| --- | --- |
| Referência | Zeuwts et al. (2023) |
| Objetivos | Desenvolver um teste de antecipação de perigos em VR interativo e imersivo, realizado em um simulador de ciclismo, incluindo uma variedade de perigos típicos enfrentados por jovens ciclistas. O teste permite a avaliação de movimentos oculares e comportamento dos ciclistas. |
| Principais funcionalidades | Utiliza tecnologia de rastreamento ocular para entender para onde os ciclistas olhavam e como detectavam perigos, combinado a realidade virtual para ciclistas, oferecendo um ambiente tridimensional realista. O estudou também contou com uma bicicleta montada em quadro estacionário, com a roda traseira ligada a um volante que permite medir a velocidade de pedalada, a cadência, a frenagem e o ângulo de direção dos ciclistas. |
| Ferramentas de desenvolvimento | Foi utilizada a Engine Unity3D |
| Resultados e conclusões | O estudo contou com a participação de 130 estudantes com média de 11 anos de idade, que já sabiam andar de bicicleta. Nos resultados, os cálculos de correlação do estudo revelaram que os participantes que pedalaram mais rápido pelo ambiente virtual pontuaram mais alto (pior) para erros e violações, além de mostrar um comportamento menos protetor em comparação com as crianças que pedalaram mais lentamente. Por fim, concluem que simuladores de RV são ferramentas úteis e realistas para documentar habilidades de antecipação de perigos em ciclistas jovens. Além disso, oferece uma base para o desenvolvimento de programas de treinamento interativos e envolventes para promover a segurança no ciclismo infantil. |

Fonte: elaborado pelo autor.

O estudo de Santos *et al.* (2019) introduz o jogo Educação na Faixa, uma ferramenta educativa em 2D para ensinar a educação no trânsito de maneira lúdica e interativa. Desenvolvido com a plataforma Construct 3, o jogo visa facilitar o ensino do trânsito nas escolas e foi validado por estudantes de informática, destacando a importância de alinhar usabilidade e conteúdo pedagógico em ferramentas educativas digitais.

Quadro 3 – Educação na Faixa: um Jogo 2D para o Ensino da Educação Para o Trânsito

|  |  |
| --- | --- |
| Referência | Santos *et al.* (2019) |
| Objetivos | Criar um jogo 2D destinado a ensinar conceitos de educação para o trânsito de forma lúdica e interativa. Os autores buscam superar a dificuldade que os professores têm em abordar a temática do trânsito nas escolas, oferecendo uma ferramenta que facilita este processo educativo, além da proposta de validação com um grupo de estudantes de informática, avaliando aspectos de usabilidade e eficácia pedagógica, para identificar áreas de melhoria. |
| Principais funcionalidades | O jogo apresenta um cenário 2D, onde os jogadores devem guiar um carro por ruas e enfrentam diversos desafios relacionados ao trânsito, como semáforos, placas e faixas de pedestres. O jogo possui a capacidade de oferecer feedback imediato: quando as regras do trânsito são desrespeitadas, o jogo reinicia com orientações adicionais, reforçando os conceitos aprendidos. Possui fases que evoluem em complexidade e conteúdo pedagógico integrado, além de possuir elementos como moedas e tempo, animações e efeitos sonoros, presentes na maioria dos jogos atuais. |
| Ferramentas de desenvolvimento | Construct 3, uma plataforma de desenvolvimento que facilita a criação de jogos específicos, eliminando a necessidade de programação complexa. |
| Resultados e conclusões | A validação com estudantes de um curso de informática revelou que o jogo é de fácil compreensão e confirmou seu alinhamento com as diretrizes pedagógicas, incluindo a Base Nacional Comum Curricular (BNCC). No entanto, identificou-se a necessidade de melhorias, especialmente em termos de motivação e engajamento dos estudantes. Alguns aspectos, como a quantidade de texto e a mecânica de repetição de fases, foram apontados como áreas a serem aprimoradas. Além disso, a plataforma Construct 3 mostrou-se eficiente para o desenvolvimento de jogos educacionais. |

Fonte: elaborado pelo autor.

# descrição DO APLICATIVO

Esta seção visa apresentar detalhes sobre o desenvolvimento do aplicativo, incluindo as técnicas e ferramentas empregadas. Ela se divide em duas partes: a subseção 3.1, que expõe a especificação do aplicativo, e a subseção 3.2, que enfoca nos aspectos principais da implementação e tecnologias utilizadas.

## ESPECIFICAÇÃO

Para descrever as funções do aplicativo, são expostos os Requisitos Funcionais (RF) no Quadro 4 e os Requisitos Não Funcionais (RNF) no Quadro 5.

Quadro 4 – Requisitos Funcionais

|  |
| --- |
| RF01: permitir que o usuário possa se movimentar pelo ambiente virtual. |
| RF02: exibir na tela o feedback sobre o desempenho dos jogadores, informando sobre erros cometidos |
| RF03: incluir cenários variados de trânsito para aumentar a exposição do usuário a diferentes situações de risco. |
| RF04: permitir que o jogador interaja com elementos do ambiente |
| RF05: possuir um sistema de tráfego onde veículos e outros pedestres se movem de forma autônoma |
| RF06: ser projetado para que o jogador vivencie a experiência através de uma visão em primeira pessoa |

Fonte: elaborado pelo autor.

Quadro 5 – Requisitos Não Funcionais

|  |
| --- |
| RNF01: ser desenvolvido para plataforma Android |
| RNF02: utilizar o motor de jogos Unity |
| RNF03: utilizar controle Bluetooth para movimentação do jogador no cenário |
| RNF04: utilizar a linguagem C# para o desenvolvimento do aplicativo |

Fonte: elaborado pelo autor.

O jogo desenvolvido posiciona o jogador no papel de um pedestre em um ambiente tridimensional, composto por um total de três fases distintas. Ele inclui um menu de seleção de fases que facilita a transição entre elas, permitindo ao jogador escolher e navegar pelas diferentes fases do jogo. Inicialmente é feita uma verificação de quais fases estão desbloqueadas e torna os respectivos botões interativos. Durante a cena, toques no botão dos óculos de realidade virtual são detectados para manipular a seleção de fases, monitorando a duração do pressionamento. Se um botão está interativo, indicando que a fase está desbloqueada, e for mantido pressionado pelo tempo necessário, a fase correspondente será carregada. Caso o botão for solto antes do tempo necessário ou receber apenas um toque, a fase selecionada é alterada. Além disso, são exibidas setas indicativas na tela, indicando qual a fase selecionada. Se o usuário permanecer inativo por um certo período, uma mensagem de ajuda é exibida na tela.

O processo de inicialização de cada fase ocorre quando o usuário está na seleção de fases e toca no botão do óculos RV para escolher a fase desejada e após, pressiona o botão até que o indicador, exposto na Figura 3, esteja completo para carregar a fase desejada.

Figura 3 – Tela de seleção de fases

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Cada fase do jogo inicia com uma *cutscene* que proporciona uma visão panorâmica e oferece uma breve contextualização do cenário, preparando o jogador para a ambientação e os desafios que virão a seguir. Após a *cutscene*, o jogador poderá se movimentar e observar o cenário ao redor, já que ele está em um ambiente de RV.

Para o desenvolvimento do projeto foram escolhidos três cenários: o cenário no qual o jogador deve se atentar a atravessar a rua em duas faixas de pedestres distintas, denominada FaixaPedestre; o cenário denominado Patinete, onde o jogador deve se atentar a procedimentos seguros relacionados a este meio de locomoção; e o cenário Acidente, onde o jogador é colocado em uma situação que deve seguir os procedimentos seguros em uma situação de acidente de trânsito.

No cenário FaixaPedestre, o jogador assume a perspectiva de um pedestre e tem como objetivo alcançar um local determinado para completar a fase. Ao longo do trajeto, é essencial manter a atenção em práticas de segurança e consciência no trânsito. Isso inclui ficar alerta a veículos que possam invadir a calçada e adotar medidas adequadas para atravessar a rua com segurança, utilizando corretamente a faixa de pedestres.

Nesta fase, o usuário tem liberdade de movimento até alcançar um ponto onde um carro cruza a calçada. Neste momento, o movimento do jogador é bloqueado e um alerta é exibido na tela, destacando a importância de estar atento a veículos que podem invadir a calçada, enfatizando a necessidade de vigilância constante no trânsito. Depois que o veículo se afasta, o movimento do jogador é novamente liberado, levando-o até um cruzamento que possui uma faixa de pedestres e um semáforo específico para pedestres. Neste ponto, o jogador deve observar o botão do semáforo e pressioná-lo utilizando o botão correspondente no controle Bluetooth, conforme é indicado na tela. Após isso, é necessário aguardar até que o semáforo fique verde para, então, atravessar a rua com segurança.

Quando o jogador executa uma ação inadequada, o jogo apresenta uma mensagem explicativa na tela, como pode ser visto na Figura 4. Isso ajuda os jogadores a entenderem as repercussões de suas escolhas e promove um aprendizado mais eficaz sobre segurança no trânsito.

Figura 4 – Mensagem explicativa

Tela de jogo de vídeo game

Descrição gerada automaticamente com confiança baixa

Fonte: elaborado pelo autor.

Após atravessar o primeiro cruzamento, o jogador encontra outra faixa de pedestres. Neste ponto, é essencial que ele olhe para ambos os lados da rua antes de atravessar, garantindo assim a própria segurança e reforçando a importância de estar atento ao trânsito, mesmo em áreas designadas para pedestres. Para orientar o usuário de forma clara, textos explicativos são exibidos na tela, como pode ser visto na Figura 5, detalhando as ações necessárias e guiando o usuário a cada passo do processo.

Figura 5 – Indicadores na tela

Texto

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Na fase Patinete, o jogador tem a oportunidade de conduzir um patinete em um cenário virtual, enfatizando a importância de práticas seguras e conscientes no trânsito, tal como na fase anterior. É fundamental que o jogador atente para aspectos cruciais como a utilização do capacete e a manutenção de uma velocidade adequada durante o percurso. Para começar, o jogador precisa subir no patinete, mas seu movimento no mesmo só é liberado após colocar o capacete e desligar a música, enfatizando a importância de estar atento ao ambiente. O jogador deve utilizar o controle bluetooth para aceleração e movimentação do patinete. Ao longo da rota, a velocidade do patinete não deve exceder 20 km/h na ciclovia, e o jogo fornecerá alertas caso o jogador ultrapasse esse limite, reforçando a necessidade de cautela e observância das regras de trânsito. Além da movimentação, em todas as ações e interações na fase, o mesmo controle Bluetooth é utilizado, como pode ser visto na Figura 6, visando oferecer uma experiência de jogo mais intuitiva.

Figura 6 – Indicador de ação

Imagem de vídeo game

Descrição gerada automaticamente com confiança média

Fonte: elaborado pelo autor.

Na fase Acidente, o jogador é colocado em uma simulação de emergência onde ocorre um acidente. O objetivo é gerenciar a situação com responsabilidade e presteza. Inicialmente, o jogador deve instruir os pedestres próximos a se afastarem para um local seguro, garantindo a segurança de todos. Posteriormente, é essencial sinalizar a área do acidente, como por exemplo, usando um galho, para evitar outros incidentes. O passo crucial é ligar para o serviço de emergência, discando 193. Com a chegada da ambulância, a fase se encerra, destacando a importância de ações rápidas e eficazes em emergências.

Ao concluir cada fase do jogo, ao atingir o objetivo estabelecido, a tela de vitória, como ilustrada na Figura 7, é exibida. Neste momento, o jogador deve pressionar o botão do óculos de RV para prosseguir. Após isto, o jogador é levado para a tela de seleção de fases, onde a próxima fase, imediatamente após a que foi completada, é desbloqueada e se torna acessível para ser jogada. Com isto, é possível ter uma progressão das fases, proporcionando um sentido de realização e motivação para avançar para os próximos desafios.

Figura 7 – Tela de vitória

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

## implementação

O desenvolvimento do jogo foi realizado utilizando o motor gráfico Unity3D, especificamente a versão 2022.3.5f1. Além disso, foi utilizado o ambiente de desenvolvimento Visual Studio Community 2022 na versão 17.6.5, que é disponibilizado de maneira gratuita. Para a utilização da realidade virtual imersiva foi utilizado o plug-in XR do Google Cardboard para Unity. Ele suporta recursos essenciais de RV, tais como: rastreamento de movimento; renderização estereoscópica e interação do usuário através do botão do visualizador. (GOOGLE DEVELOPERS, 2023).

Os cenários do jogo foram criados com uma variedade de modelos, incluindo construções, personagens e plantas. Todos esses modelos foram adquiridos da *asset store* oferecido pela Unity, proporcionando uma ampla gama de recursos para enriquecer o ambiente do jogo. As *cutscenes* foram criadas utilizando duas ferramentas fornecidas pela Unity: Timeline e Cinemachine.

O Timeline é uma ferramenta fornecida pelo Unity para a elaboração de sequências de eventos (UNITY TECHNOLOGIES, 2020a). Com ele, é possível mover clipes no tempo, ajustar quando começam e determinar como se combinam e interagem com outros clipes na sequência. Conforme Unity Technologies (2020b) a Cinemachine torna simples e eficiente configurar e animar câmeras, sem a necessidade de escrever código. Utilizando câmeras virtuais geridas pelo Cinemachine Brain, que se integra à câmera principal no Unity, os usuários podem animar e mesclar estas câmeras virtuais de diversas formas. Quando usados em conjunto, o Timeline e a Cinemachine permitem a sincronização de animações de câmera com outros elementos da cena, como animações de personagens e efeitos sonoros e visuais. Como exemplificado na Timeline da Figura 8, é possível observar que o estado de dois GameObjects com o nome FurbotPersonagemImg e Picape foram manipulados, ao mesmo tempo em que foi feito a animação da câmera através da Cinemachine CM Intro Shot.

Figura 8 – *Cutscene* utilizando Timeline e Cinemachine

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Para a movimentação do personagem, foi utilizado o novo sistema de entradas da Unity, disponível a partir da instalação de um pacote disponibilizado de maneira gratuita pela própria Unity, chamado de Input System. Inicialmente, deve ser adicionado o componente PlayerInput a um GameObject desejado. Para que o componente possa processar entradas (como comandos de teclado, mouse ou *gamepad*), ele precisa estar vinculado a um conjunto de Input Actions. A partir do PlayerInput é possível criar este Asset com um conjunto pré-preenchido com ações padrão. A partir dele é possível definir como as entradas dos dispositivos são interpretadas e transformadas em ações dentro do jogo. Com isso, é foi possível editar todos os métodos de entrada e botões necessários para o jogo, como visto na Figura 9, onde é possível visualizar as ações para a movimentação e interações do patinete.

Figura 9 – Definição das entradas necessárias no componente Input Actions

Tela de computador

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Uma vez que o componente tenha suas ações definidas, é necessário configurar uma resposta para cada ação. O PlayerInput permite estabelecer respostas de diversas maneiras, utilizando a propriedade Behaviour na janela do Inspector. Conforme representado na Figura 10, a opção utilizada foi a Invoke Unity Events, que, para cada ação ligada ao componente, faz com que o Unity exiba um evento correspondente. Essa funcionalidade permite a associação direta de um método específico a cada um desses eventos. Cada método aceita um argumento do tipo InputAction.CallbackContext, que permite acessar o controle que responsável por iniciar a ação e o valor associado a essa ação.

Figura 10 – Componente Player Input

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Cada ação pode passar por várias fases distintas em resposta a uma entrada, incluindo started, performed e canceled. No Quadro 6, observa-se o método de movimentação do personagem no script personagem, chamado de SetMovimento, que emprega as condições started e canceled. Started indica que uma interação com a ação foi iniciada, como, o botão de movimentação sendo pressionado, enquanto canceled sinaliza que a interação foi encerrada, ou seja, o usuário soltou o botão.

|  |
| --- |
| public void SetMovimento(InputAction.CallbackContext value)  {  if (!podeMover)  {  return;  }  if (value.started)  {  movimentouPersonagem = true;  }  else if (value.canceled)  {  movimentouPersonagem = false;  }  movimentoControle = value.ReadValue<Vector2>();  } |

Quadro 6 – Método que recebe a entrada de movimentação do personagem

Fonte: elaborado pelo autor.

A movimentação do personagem é efetuada a partir do método Update no script Personagem. Ele começa lendo as entradas do usuário (movimento horizontal e vertical do controle) armazenando nas variáveis horInput e verInput e utiliza a orientação da câmera principal para determinar as direções de movimento relativas. A rotação do personagem é ajustada para coincidir com a da câmera, sendo bloqueada no eixo X, já que o personagem não deve rotacionar no eixo vertical. O movimento é calculado combinando as direções para frente e lateral, com um ajuste na velocidade lateral. A direção de movimento de um personagem é calculada combinando dois vetores: camFrenteRelativa e camDireitaRelativa. Estes vetores representam, respectivamente, as direções para frente e para a direita com base na orientação atual da câmera, e são ponderados pelas entradas do usuário. Ao somá-los, o vetor resultante, moveDir, indica a direção diagonal em que o personagem deve se mover, efetuando o movimento em relação à orientação da câmera, e não apenas em relação ao mundo do jogo.

Isso significa que, independentemente de para onde o personagem está virado inicialmente, ao mover-se para frente, ele se deslocará na direção para onde a câmera está apontando. Por fim, o vetor de movimento é aplicado ao personagem usando o método Move do NavMeshAgent, multiplicado por Time.deltaTime, garantindo fluidez independente da taxa de quadros e pela variável velocidade, que define a velocidade do personagem. Permitindo assim um deslocamento responsivo do personagem no ambiente de jogo baseado nos comandos do usuário. O Quadro 7 demonstra o trecho de código responsável pela movimentação do personagem.

Quadro 7 –Trecho de código responsável movimentação do personagem

|  |
| --- |
| horInput = movimentoControle.x;  verInput = movimentoControle.y;  Vector3 camFrente = cameraPrincipal.forward;  Vector3 camDireita = cameraPrincipal.right;  Quaternion cameraRotacao = cameraPrincipal.rotation;  // bloqueia rotacao no eixo x  Quaternion novaRotacao = Quaternion.Euler(0f, cameraRotacao.eulerAngles.y, cameraRotacao.eulerAngles.z);  this.transform.rotation = novaRotacao;  camFrente.y = 0;  camDireita.y = 0;  Vector3 camFrenteRelativa = verInput \* camFrente.normalized;  Vector3 camDireitaRelativa = horInput \* camDireita.normalized;  float multiplicaMovimentacaoLateral = 2.0f;  camDireitaRelativa \*= multiplicaMovimentacaoLateral;  Vector3 moveDir = (camFrenteRelativa + camDireitaRelativa);  movimentoPersonagem = new Vector3(moveDir.x, velocidadeY, moveDir.z);  navMeshAgent.Move(movimentoPersonagem \* Time.deltaTime \* velocidade); |

Fonte: elaborado pelo autor.

Para gerenciar as áreas de movimentação dos agentes do jogo, foi empregado o componente NavMesh da Unity. Foram utilizados os componentes NavMeshSurface, Nav Mesh Agent e NavMeshModifier do Unity. O NavMeshSurface atua como o componente responsável por delinear as áreas de circulação disponíveis para diferentes tipos de agentes. Na cena FaixaPedestre, por exemplo, foram empregados três NavMeshSurfaces distintos: um para a movimentação do personagem, outro para a circulação dos veículos e um terceiro destinado aos pedestres. Para associar os agentes a cada NavMeshSurface específico, adiciona-se a esses objetos um componente do tipo Nav Mesh Agent. Isso permite, ao gerar o NavMesh, delimitar as áreas de circulação permitidas para cada tipo de agente. A Figura 11 ilustra a área azul que representa o NavMesh utilizado para os pedestres, restringindo suas áreas de movimentação somente às calçadas do cenário. TIRAR NUVENS

Figura 11 – Navmesh dos pedestres

Tela de computador com jogo

Descrição gerada automaticamente com confiança média

Fonte: elaborado pelo autor.

Além disso, empregou-se o componente NavMeshModifier, que possibilita a personalização do comportamento de um GameObject durante a geração do NavMesh. Com ele, é possível, por exemplo, selecionar e excluir certos obstáculos para agentes específicos ou remover GameObjects específicos do processo de construção do NavMesh. Isto foi empregado na lógica da faixa de pedestres, na qual o GameObject faixa\_pedestre recebeu o componente NavMeshModifier, e através da propriedade Override Area, a área foi definida como Not Walkable, que permite excluir a faixa do processo de construção do NavMesh para os agentes Personagem e NPC.

Nesse contexto, o componente é ativado quando o semáforo para pedestres está vermelho e desativado quando está verde. Isso significa que o NavMesh é atualizado em tempo real, permitindo a travessia na faixa de pedestres apenas quando o semáforo para pedestres está verde. O Quadro 8 exemplifica o código utilizado no script ControlaSemaforo para atualização para o sinal verde e reconstrução do NavMesh.

Quadro 8 –Trecho de código responsável pela alteração do semáforo

|  |
| --- |
| IEnumerator SinalVerde()  {  //espera antes de ficar verde  yield return new WaitForSeconds(8f);  sinalVerde.GetComponent<MeshRenderer>().SetMaterials(materialsVerde);  sinalVermelho.GetComponent<MeshRenderer>().SetMaterials(materialsPreto);  faixaPedestre.GetComponent<NavMeshModifier>().enabled = false;  ReconstruirNavmesh();  //fica verde depois vermelho novamente  yield return new WaitForSeconds(duracaoVerde);  SinalVermelho();  }  private void ReconstruirNavmesh()  {  navMeshPersonagem.UpdateNavMesh(navMeshPersonagem.navMeshData);  navMeshCarro.UpdateNavMesh(navMeshCarro.navMeshData);  } |

Fonte: elaborado pelo autor.

O movimento dos Non-Player Characters (NPCs), incluindo pedestres e veículos, foi implementado através de um *script* denominado MovimentaNPC. Para isso, foi estabelecido uma série de pontos de referência na cena, os quais foram integrados ao código do *script*. Esses pontos de referência foram dispostos no ambiente do jogo utilizando GameObjects nomeados como Waypoint, seguidos de um índice numérico para sua identificação.

No *script* MovimentaNPC, o método Update é utilizado para controlar a movimentação dos NPCs no jogo. A cada *frame*, ele calcula a distância do NPC até seu ponto de referência atual. Se o NPC estiver suficientemente próximo do ponto de referência, o *script* atualiza para o próximo ponto de referência ou retorna ao primeiro se todos os pontos de referência já foram visitados. Para alguns agentes, como Van ou Carro, o script desativa o NavMeshAgent e o próprio GameObject após atingir o último próximo ponto de referência. Além disso, para NPCs identificados como carros, realiza-se uma verificação adicional de colisão à frente do veículo. O *script* também ajusta as animações do NPC com base em sua velocidade de movimentação, e, finalmente, se o NPC estiver habilitado para se mover (podeAndar), seu destino é atualizado para a posição do ponto de referência seguinte, guiando-o automaticamente pelo cenário do jogo a partir do método SetDestination do NavMeshAgent.

Devido a necessidade da interação ao observar GameObjects específicos, como o botão do semáforo, foi utilizado o *script* CardboardReticlePointer, fornecido pelo plug-in Google Cardboard XR. No método Update deste *script*, um raio de colisão é emitido a partir da câmera para detectar se algum GameObject está sendo focalizado. Quando um GameObject é identificado, o *script* utiliza o método SendMessage para interagir com ele. SendMessage é uma função que permite a comunicação entre diferentes *scripts* e componentes de um GameObject. Utilizando esse método, a função OnPointerEnter é invocada. Portanto, qualquer *script* associado ao GameObject focado que contenha o *script* que contenha o método OnPointerEnter é acionado, permitindo uma resposta adequada a essa interação. O método OnPointeEnter foi implementado em um *script* chamado de ObjetoGenerico, por conta de ser necessário esta mesma lógica em diferentes GameObjects. O script deve ser associado ao GameObject desejado, e assim que o GameObject é focalizado pela câmera na cena, o método OnPointerEnter é chamado devido ao método SendMessage descrito acima. O Quadro 9 demonstra como o método OnPointerEnter é utilizado para diferentes GameObjects.

Quadro 9 – Método OnPointerEnter que efetua interações para diferentes GameObjects

|  |
| --- |
| public void OnPointerEnter()  {  string nomeObjeto = this.gameObject.name;  switch (nomeObjeto)  {  case "LadoEsquerdo" when alerta.mostrouAlertaOlharDoisLados:  alerta.MostrarAlerta("Esquerda");  personagem.olhouEsquerda = true;  break;  case "LadoDireito" when alerta.mostrouAlertaOlharDoisLados:  alerta.MostrarAlerta("Direita");  personagem.olhouDireita = true;  break;  case "Botao":  alerta.MostrarAlerta("Botao");  break;  case "Capacete" when patinete.subiuPatinete:  patinete.olhouCapacete = true;  alerta.MostrarAlerta("ColocarCapacete");  break;  default:  break;  }  } |

Fonte: elaborado pelo autor.

Para orientar os jogadores sobre como realizar ações no jogo e prover feedback sobre acertos e erros através de informações na tela, foi implementado o *script* Alerta. O método MostrarAlerta no *script* Alerta é uma função responsável por exibir mensagens de alerta no jogo, de acordo com diferentes situações e eventos. O método recebe uma string alerta como parâmetro, que especifica o tipo de alerta a ser mostrado. Dentro do método, há uma estrutura switch que verifica o valor de alerta e executa diferentes blocos de código baseados nesse valor. Cada caso dentro da estrutura switch configura uma mensagem específica a ser exibida para o jogador. Por exemplo, se o alerta for Botao e o jogador estiver em frente ao botão do semáforo (estaFrenteBotao), uma mensagem específica será definida. O Quadro 10 demonstra um trecho do método MostrarAlerta. Por fim, o GameObject mensagemAlerta, que contém o texto a ser exibido, é ativado. Este GameObject é filho do Canvas e assim que ativado, exibe a mensagem definida na tela.

Quadro 10 – Método MostrarAlerta

|  |
| --- |
| public void MostrarAlerta(string alerta)  {  texto.fontSize = 40;  switch (alerta)  {  case "Botao" when estaFrenteBotao:  textoBotaoSemaforo.SetActive(true);  podeAcionarBotao = true;  break;  case "Faixa":  alertaVisivel = true;  texto.text = "Nunca atravesse a faixa sem olhar para os dois lados!";  mostrouAlertaOlharDoisLados = true;  mensagemAlerta.SetActive(true);  break;  case "Capacete":  texto.text = "Sempre coloque o capacete ao andar de patinete!";  mensagemAlerta.SetActive(true);  break; |

Fonte: elaborado pelo autor.

O Canvas na Unity é essencial para a User Interface (UI), atuando como a área onde todos os elementos de UI, como imagens e botões, devem ser posicionados. Esses elementos devem ser filhos de um Canvas para serem renderizados e interativos. Se não houver um Canvas na cena ao criar um elemento de UI, a Unity cria um automaticamente. Há dois tipos principais de Canvas: Screen Space, que cobre a tela inteira, e World Space, que situa a UI no espaço tridimensional do jogo (UNITY TECHNOLOGIES, 2023). Para este projeto, foi escolhido o modo Screen Space – Camera, onde o Canvas é posicionado a uma distância específica de uma câmera selecionada. Isso permite que os elementos da UI sejam projetados na tela conforme a posição e orientação da câmera. O Canvas ajusta seu tamanho e escala automaticamente com mudanças na resolução da tela ou no campo de visão da câmera. Assim, é adequado para ambientes que utilizam a RV, já que a UI adapta-se dinamicamente ao campo de visão da câmera RV, mantendo proporção e legibilidade, garantindo que os alertas estejam sempre visíveis no campo de visão do usuário, no caso deste jogo, centralizados na tela.

# resultados

# CONCLUSÕES

Referências

AMBEV; FALCONI Consultores de Resultados; ONSV – Observatório Nacional de Segurança Viária. **Retrato da Segurança Viária 2014**, Brasília: ONSV, 2014.

ARAUJO JUNIOR, Delcides. **Educação de trânsito:** a necessidade premente de um trânsito mais altruísta. 2019. 14 f. Trabalho de Conclusão de Curso (Especialização em Gestão de Trânsito) - Universidade do Sul de Santa Catarina, Santa Catarina.

BONFIM, Lilian M. L. V. *et al.*, Educando Para o Trânsito. **Revista Científica Semana Acadêmica**. Fortaleza, v.1, n. 000112, p. 1-12, set. 2017.

BOWMAN, Doug A.; MCMAHAN, Ryan P. Virtual Reality: how much immersion is enough? **Computer**, [S.L.], v. 40, n. 7, p. 36-43, jul. 2007. Institute of Electrical and Electronics Engineers (IEEE).

BRAGA, Mariluci. Realidade Virtual e Educação. **Revista de Biologia e Ciências da Terra**, v. 1, n. 1, p. 1-8, 2001.

CARVALHO, Gabriel R. de. **A importância dos jogos digitais na educação**. 2018. 41 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Minas Gerais, Belo Horizonte.

DA SILVA, John W. *et al*. Educa Trânsito – Um jogo de apoio à educação no trânsito. **RENOTE**, Porto Alegre, v. 12, n. 2, 2014.

DA SILVA, Maksuel L. N. **Educação para o trânsito: Uma parceria em prol da vida**. 2017. 30f. Monografia (Especialista) - Curso de Pós-Graduação em Segurança Viária Urbana, Universidade Federal do Tocantins, Araguaína.

DO NASCIMENTO, Beatriz R. Impactos da educação de trânsito na saúde pública. **Revista UniCET**, v. 3, n. 1, 2021.

FALKEMBACH, Gilse A. M. O lúdico e os jogos educacionais. Porto Alegre**: CINTED-Centro Interdisciplinar de Novas Tecnologias na Educação**, UFRGS, 2006.

GOOGLE DEVELOPERS**. Quickstart for Google Cardboard for Unity**, **[**2023]. Disponível em: https://developers.google.com/cardboard/develop/unity/quickstart?hl=en. Acesso em: 16 nov. 2023.

KIRNER, Claudio; KIRNER, Tereza G. Evolução e tendências da Realidade Virtual e da Realidade Aumentada. **Realidade Virtual e Aumentada: Aplicações e Tendências**. São Paulo: Ed. Blucher, v. 1, p. 10-25, 2011.

MIRANDA, Marli. F. S. Trânsito dos alunos no espaço escolar, sua disciplina e reflexo no contexto social. In: FESTIVAL ESTUDANTIL TEMÁTICO DE TRÂNSITO. 2016, Pouso Alegre. **Anais** [...]. Pouso Alegre, 2016, p. 01-10.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. **Resolução 74/299**. Plano Global - Década de Ação pela segurança no trânsito 2021-2030. Genebra: Organização Pan-Americana da Saúde (OPAS/OMS), 2021, p. 01-36.

PRENSKY, Marc. The games generations: How learners have changed**. Digital game-based learning**, v. 1, n. 1, p. 1-26, 2001.

ROUSSOU, Maria. Learning by doing and learning through play. **Computers In Entertainment**, [S.L.], v. 2, n. 1, p. 10-10, jan. 2004. Association for Computing Machinery (ACM).

SANTANA, Alessandro D.; TRONTO, Iris F. de B.; SOUSA, Pedro M. de. Jogo Educativo para Auxílio na Educação no Trânsito. **Revista Brasileira de Educação e Cultura (RBEC)**, v. 17, p. 25-45, 2018.

SANTOS, Jarles G. *et al.* Educação na Faixa: um Jogo 2D para o Ensino da Educação Para o Trânsito. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 25., 2019, Brasília. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 763-772.

TORI, Romero; HOUNSELL, Marcelo da Silva; KIRNER, Claudio. Realidade Virtual; in: TORI, Romero; HOUNSELL, Marcelo da Silva (org.). **Introdução a Realidade Virtual e Aumentada**. 3. ed. Porto Alegre: Editora SBC, 2020.

UNITY TECHNOLOGIES. **Canvas**, [2023]. Disponível em: https://docs.unity3d.com/2022.3/Documentation/Manual/UICanvas.html. Acesso em: 21 nov. 2023.

UNITY TECHNOLOGIES. **Introduction to Timeline - 2019**, [2020a]. Disponível em: https://learn.unity.com/tutorial/introduction-to-timeline-2019. Acesso em: 16 nov. 2023.

UNITY TECHNOLOGIES. **Overview of Cinemachine**, [2020b]. Disponível em: https://learn.unity.com/tutorial/overview-of-cinemachine#. Acesso em: 16 nov. 2023.

ZEUWTS, Linus H. R. H. *et al*. Using an immersive virtual reality bicycle simulator to evaluate hazard detection and anticipation of overt and covert traffic situations in young bicyclists. **Virtual Reality**, v. 27, p. 1-21, 2023.