

DRIVER VR

Aluno: Vítor Gabriel Eduardo

Orientador: Dalton Solano dos Reis

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos / Especificação / Implementação
- Resultados / Conclusões

Introdução

- Trânsito
- Ambiente Virtuais
- Dispositivos de Realidade Virtual
- Educação sobre o Trânsito

Objetivos

- Criar um simulador que possibilite usuários a aprenderem de maneira lúdica sobre a direção de um veículo:
 - Criar um cenário dinâmico que simule alguém dirigindo em um carro
 - Utilizar o Óculos Quest 2 com volante, pedal e marcha para aumentar a imersão
 - Testar a eficácia do simulador com grupos de usuários

Fundamentação Teórica

Trânsito

- Vítimas do trânsito aumentam
- Países com maiores índices
- O papel das infrações e educação

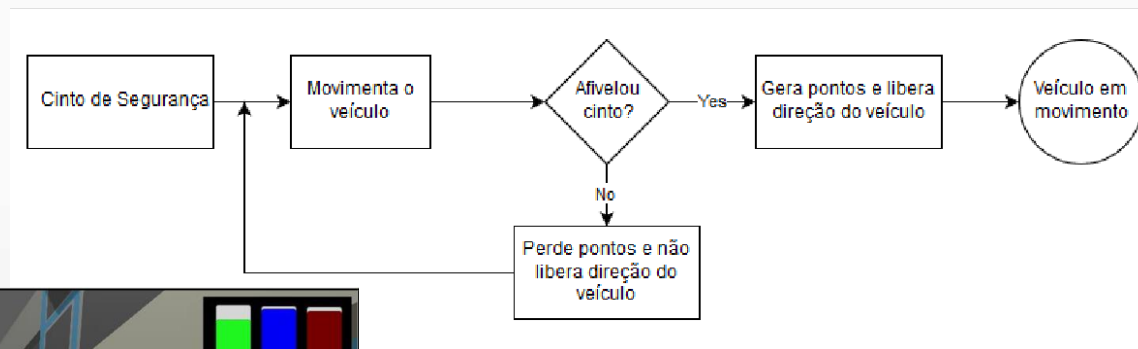
Realidade Virtual Imersiva

- Ampliam oportunidades de aprendizado
- Constroem autoconfiança e motivação no ambiente educacional
- A influência da imersão
- Ensino e diversão

Trabalhos Correlatos

TransitAR – Jogo de Conscientização Sobre o Trânsito (Buzzi, 2018)

- Gamificação dos aspectos do trânsito
- Diversificação de eventos
- Complexidade



Trabalhos Correlatos

Análise da Validade Comportamental de Um Simulador de Direção Imersivo (Andiola, 2021)

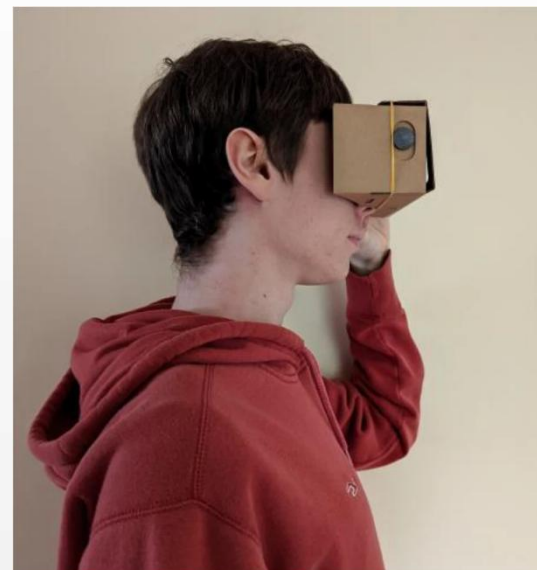
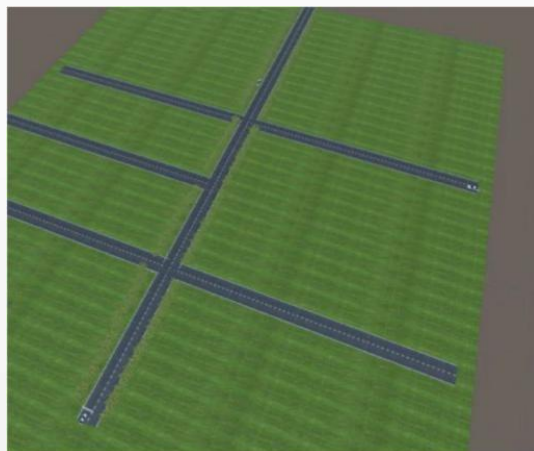
- Ambiente virtual x real
- Realidade virtual de alto custo
- Testes e validações



Trabalhos Correlatos

A Case Study of a Virtual Reality-Based Drink Driving Educational Tool (Mastertonl e Wilson, 2023)

- Realidade Virtual em dispositivos móveis
- Mundo virtual, padrões reais?
- Resultados



Especificação

Requisitos Funcionais

- RF01 - Possuir um menu inicial para a escolha de diferentes cenários
- RF02 - Mimetizar as mãos reais para mão virtuais dentro do ambiente virtual
- RF03 - Mostrar o movimento do volante real dentro do ambiente virtual
- RF04 - Possuir um sistema de feedback no volante para pista molhada
- RF05 - Possuir um sistema de feedback no volante para pista de terra
- RF06 - Colocar o usuário em primeira pessoa dentro de um kart
- RF07 - Colocar um sistema de sinto de segurança
- RF08 - Colocar um sistema para o motor morrer
- RF09 - Ter um sistema de troca de marcha

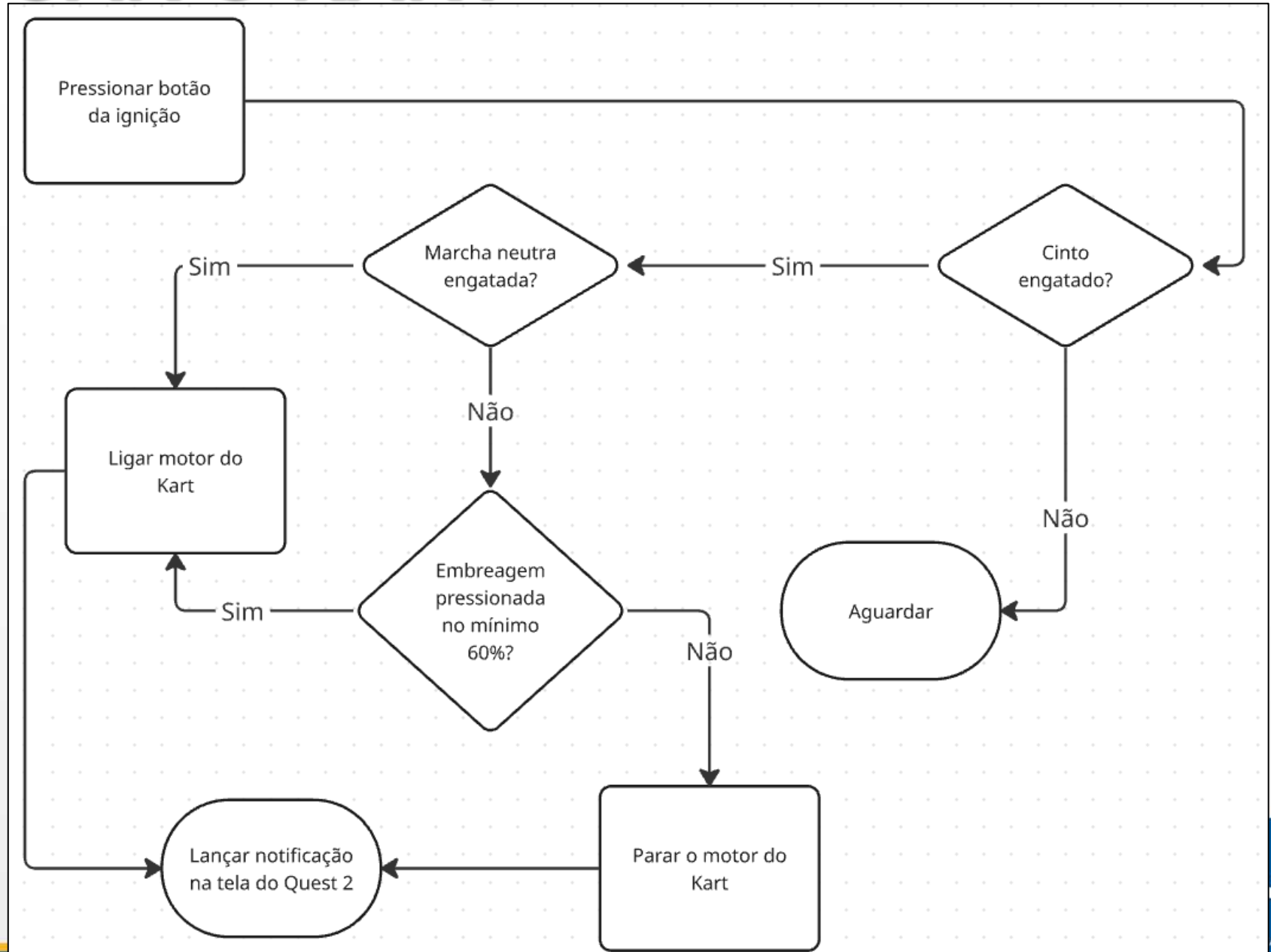
Requisitos Funcionais

- RF10 - Considerar pedais de embreagem, aceleração e freio independentemente;
- RF11 - Aplicar força G aplicada ao kart;
- RF12 - Ter um sistema de injeção eletrônica;
- RF13 - Ter um sistema de rotação do motor e rotação de roda independente;
- RF14 - Aplicar força de torque conforme a rotação interna do motor e marcha atual;
- RF15 - Aplicar freio a motor conforme a rotação interna do motor.

Requisitos Não Funcionais

- RNF01 - Ser desenvolvido no Unity
- RNF02 - Rodar no computador
- RNF03 - Utilizar C# como linguagem de programação
- RNF04 - Utilizar o Quest 2 como exibição de vídeo
- RNF05 - Ler dados do ambiente real do Quest 2
- RNF06 - Ler dados dos controles Logitech G29 (pedais e volante) e a Câmbio DE9 Logitech

LIGAR O KART



CarroManager

- inputsLogi : LogitechGSDK.DIJOYSTATE2ENGINES
- _INDEX_LOGI : int
- _MAXIMO_INPUT_VOLANTE : float
- _DOWN_FORCE : int
- controllerManager : ControllerManager
- volante : VolanteManager
- rodas : RodasManager
- motor : MotorManager
- transmissao : TransmissaoManager
- offRoad : bool
- pistaMolhada : bool
- pivotCameraVR : Transform
- carroRigidBody : Rigidbody
- centroDeMassa : GameObject
- motorLigado : bool
- botoesPressionados : bool[]
- embreagem : float
- freio : float
- acelerador : float
- rotacaoVolanteAbsoluta : float
- rotacaoVolanteAbsolutaNormalizado : float
- kph : float
- marcha : MarchaEnum
- marchaNova : MarchaEnum

- + Start() : void
- + Update() : void
- addEffects() : void
- AtualizarVariaveis() : void
- CapturarValoresBotoesVolante() : void
- ObterValorDPad() : void
- VerificarBotao(indiceBotao: int, acao: Action) : void
- + VelocidadeTotal() : float
- + RpmMotor() : float
- ObterMarchaAtual() : MarchaEnum
- OnApplicationQuit() : void
- NormalizarDadoPedal(dadoBruto: float) : float
- ValidarBugInicio(rec: DIJOYSTATE2ENGINES) : bool
- addDownForc

MOTOR DO KART

MOTOR DO KART

RodasManager

- pneus : WheelCollider[]Attributes
- meshPneus : GameObject[]
- tracao : TracaoEnum
- poderDeFreioPedal : int
- poderDeFreioMao : int

- + RodasRPM() : float
- + Mover(forcaTotalMotor: float) : void
- + FreiarMao(freioNormalizado: float) : void
- + FreiarPedal(freioNormalizado: float) : void
- + GirarRodas(anguloAbsoluto: float) : void
- AnimarPneu

MotorManager

- controllerManager : ControllerManagerAttributes
- eficienciaMotor : AnimationCurve
- curvaTorque : AnimationCurve
- curvaFreioMotor : AnimationCurve
- curvaLimiteRPMMotor : AnimationCurve
- marchaAtual : MarchaEnum
- rotacaoLivre : float
- rpmMotor : float
- rpmMotorAnterior : float
- diferencial : float
- velocidadeSuavizada : float
- freioMot

- + CalcularFreioMotor() : float
- + RelacaoMarcha() : float
- + RpmAlvoMotorLivre(pedalAceleracao: float) : float
- + EhMarchaNeutra() : bool
- + CalcularPotenciaMotor(pedalAceleracao: float) : float
- + CalcularRpmMotor(rpmMotorAlvo: float, impactoEmbreagem: float) : float
- + RpmMotor() : float
- + ValidarSeRpmMuitoBaixo(rpmMotorAlvo: float, pedalEmbreagem: float) : void
- + ValidarVariacaoRpmMuitoAlta(rpmMotorAlvo: float, pedalEmbreagem: float) : void
- + TrocarMarcha(marcha: MarchaEnum, pedalEmbreagem: float, notificar: bool) : void
- Start() : void
- EmbreagemPressionada(pedalEmbreagem: flo

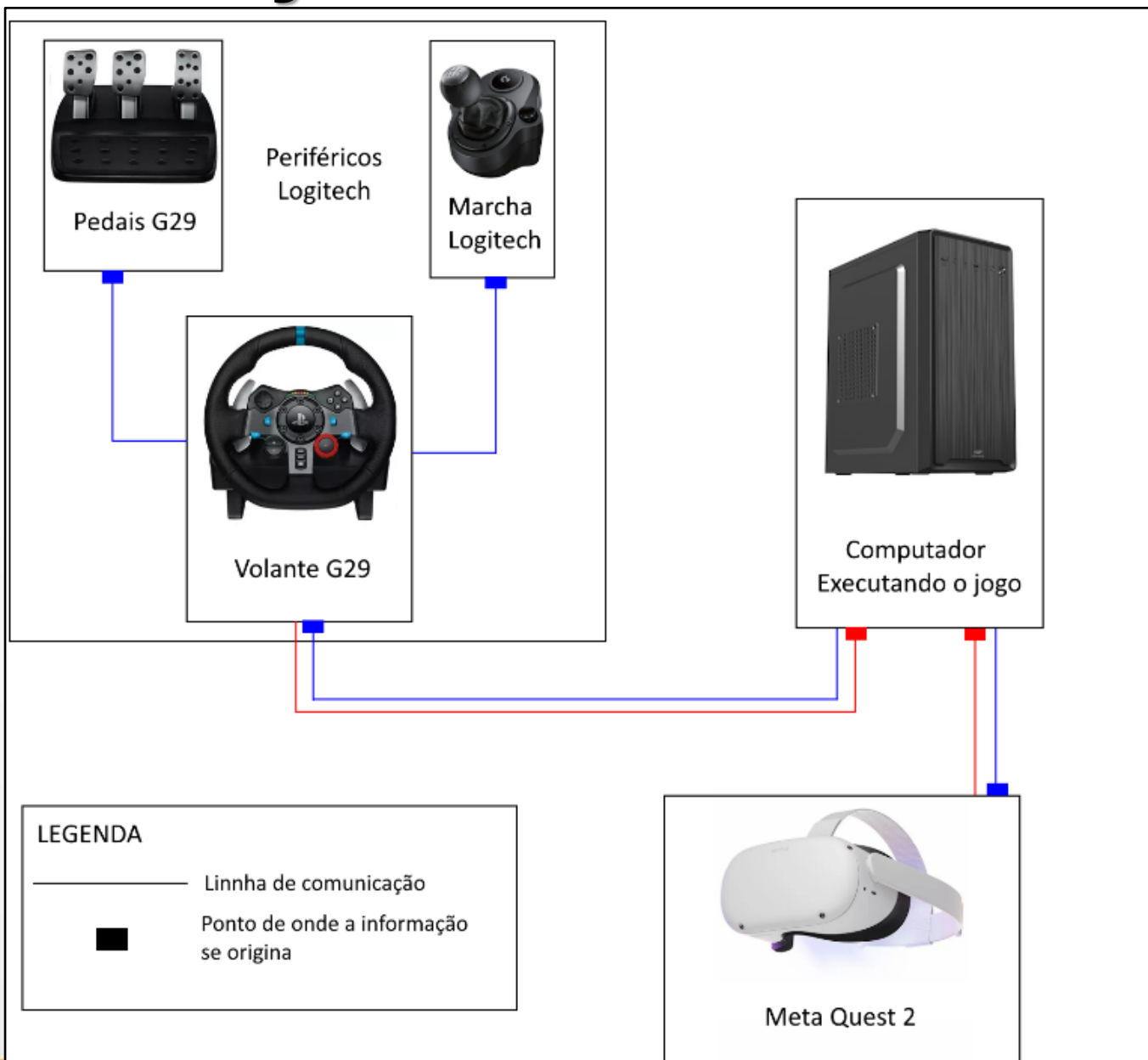
TransmissaoManager

- motor : MotorManagerAttributes
- pedalEmbreagem : float
- pedalEmbreagemInv : float
- pedalAceleracao : float
- rpmRod

- + Acelerar(rodas: RodasManager, motor: MotorManager, pedalEmbreagem: float, pedalAceleracao: float, kph: float, deveAcelerar: bool) : void
- CalcularForcaFreioMotor(pedalEmbreagem: float, pedalAceleracao: float, motor: MotorManager, kph: float) : float
- CalcularForcaAceleracaoMotor(deveAcelerar: bool) : float
- FaseMotorNeutro() : float

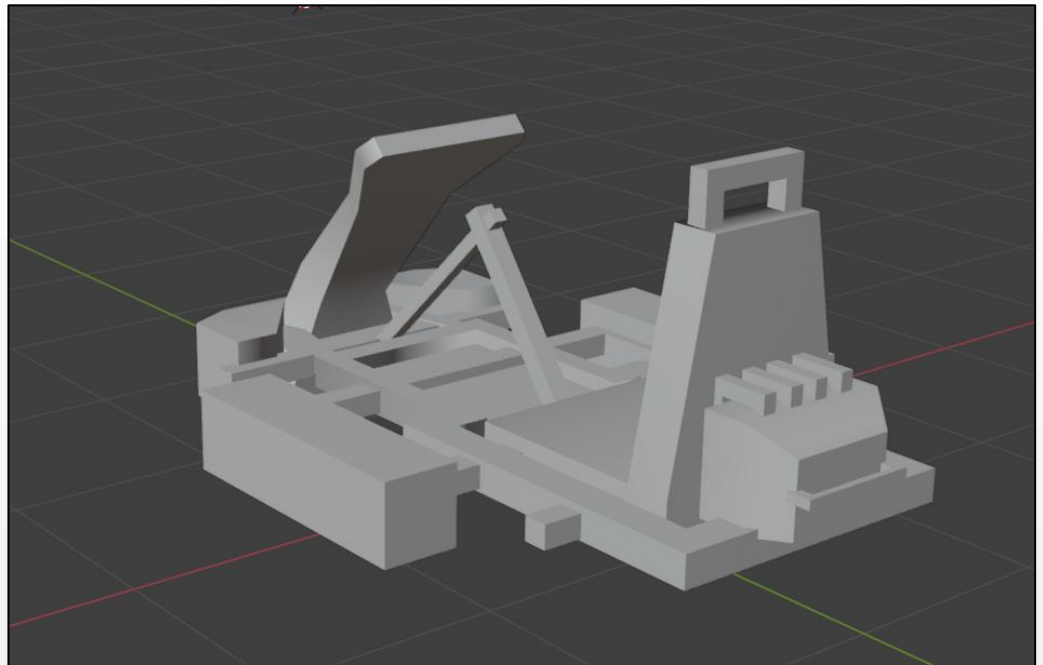
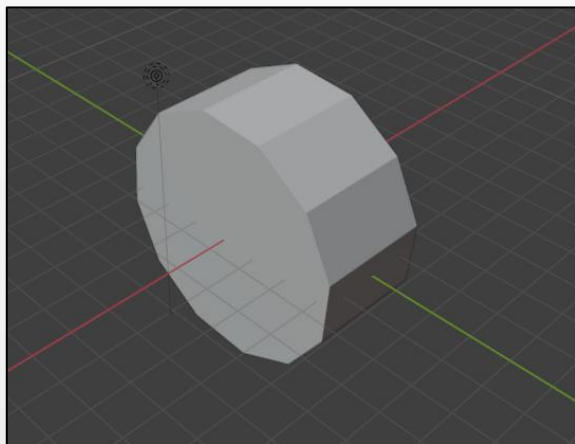
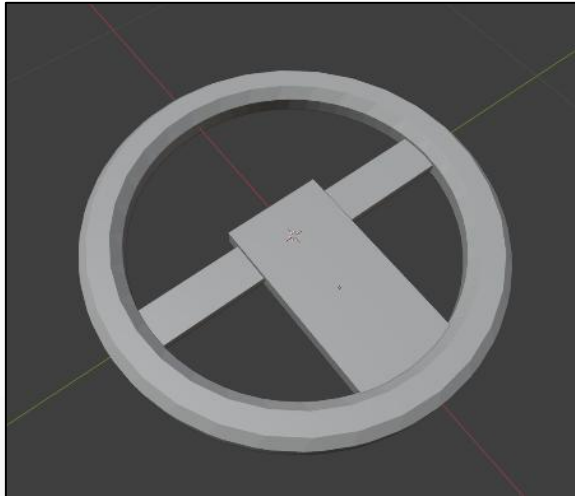
Implementação

Implementação



Implementação

- Ajustando assets



Implementação

- Código principal do motor

```
64 private float FaseMotor()  
65 {  
66     // Pegar Valor inverso do pedal da embreagem  
67     // Enquanto mais toca no pedal, menos deve considerar a rotação da roda  
68     float rpmMotorAlvoMotor = pedalEmbreagem * motor.RpmAlvoMotorLivre(pedalAceleracao);  
69     float rpmMotorAlvoRodas = pedalEmbreagemInv * rpmRodas * motor.RelacaoMarcha();  
70     float rpmMotorAlvo = rpmMotorAlvoMotor + rpmMotorAlvoRodas;  
71  
72     rpmMotorAlvo -= 30_000 * Time.deltaTime;  
73  
74     // Se a embreagem estiver muito apertada a velocidade com que o motor aumenta as rotações aumenta também  
75     float impactoEmbreagem = (Mathf.Abs(pedalEmbreagem - 1) * 0.4f);  
76     float rpmMotorAtual = motor.CalcularRpmMotor(rpmMotorAlvo, 0.05f + impactoEmbreagem);  
77  
78     // Se a rotação alvo for menos que a livre deve matar o carro!!!  
79     motor.ValidarSeRpmMuitoBaixo(rpmMotorAtual, pedalEmbreagem);  
80  
81     float torqueTotal = motor.CalcularPotenciaMotor(Mathf.Max(0.1f, pedalAceleracao)) * pedalEmbreagemInv;  
82  
83     return torqueTotal;  
84 }
```

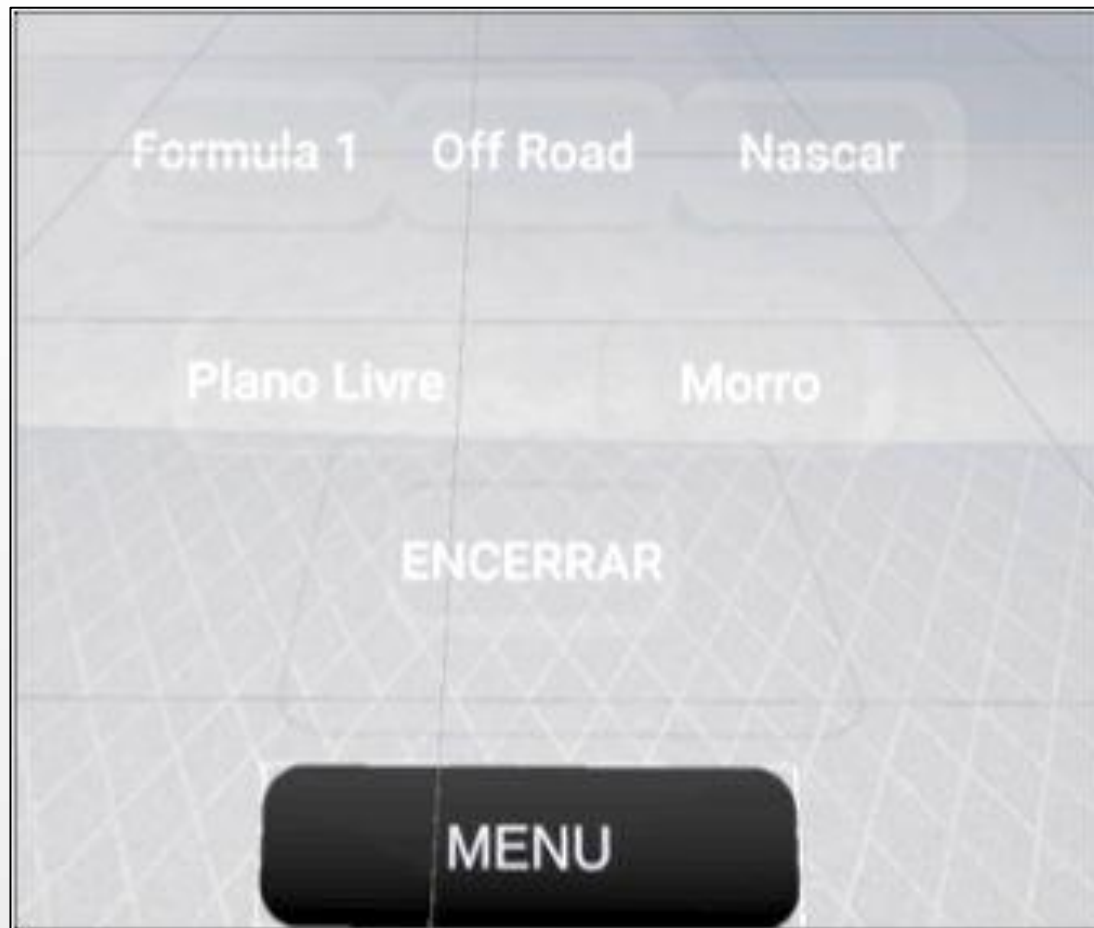
Implementação

- Código que atualiza o volante

```
8      [SerializeField] private float velocidadeGiroVolante = 1.34f;
9      private float rotacaoVolanteAnt = 0;
10     1 referência
11     public void RotacionarVolante(float rotacaoVoltanteAbsoluto)
12     {
13         float qtdRotacaoRelativa = rotacaoVoltanteAbsoluto - rotacaoVolanteAnt;
14         if (qtdRotacaoRelativa != 0)
15         {
16             float valorRelativo = qtdRotacaoRelativa * velocidadeGiroVolante;
17             Vector3 rotacao = transform.localEulerAngles;
18             rotacao.y += (valorRelativo / 100);
19             transform.localEulerAngles = rotacao;
20             rotacaoVolanteAnt = rotacaoVoltanteAbsoluto;
21         }
22     }
```

Implementação

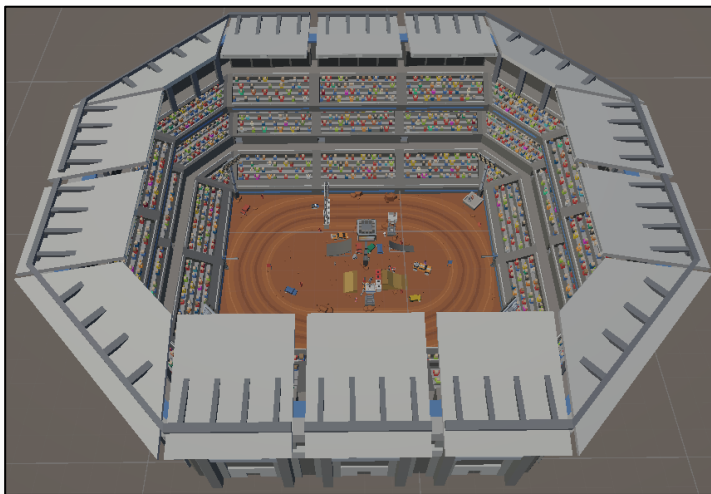
- Tela inicial da aplicação



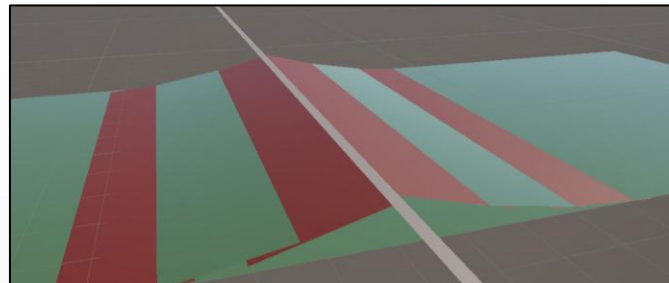
Implementação

- Cenários

1



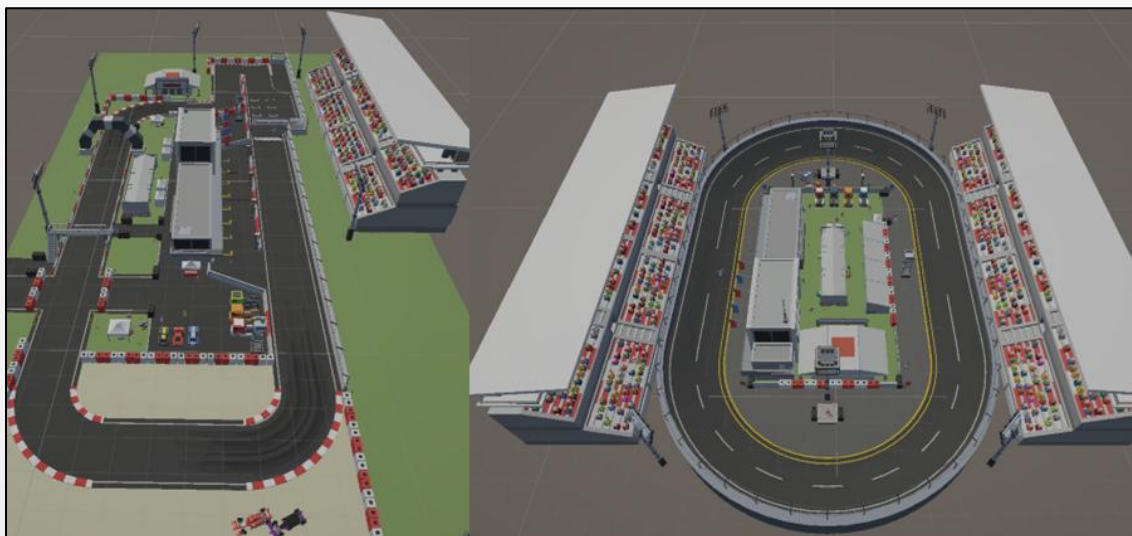
2



3



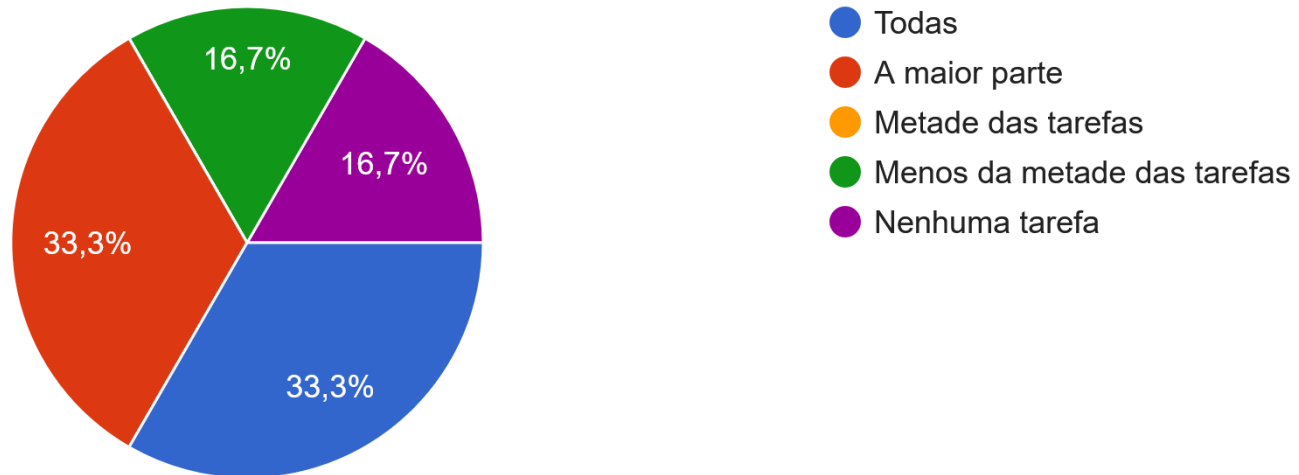
4



Análise dos Resultados

Das atividades solicitadas, quantas atividades você conseguiu executar sem auxilio?

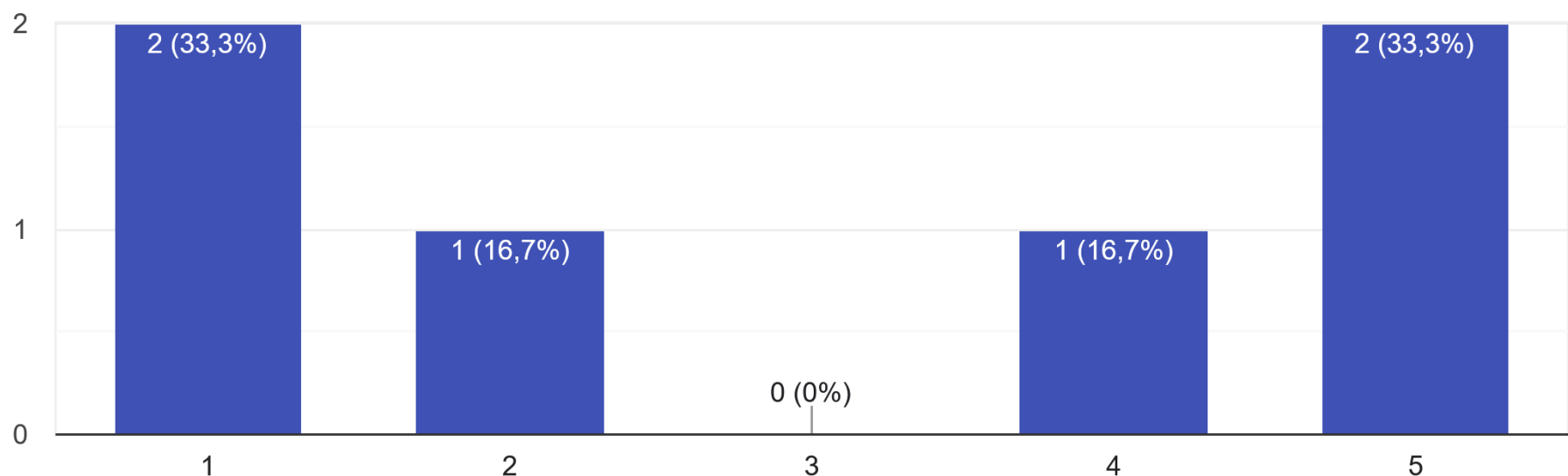
6 respostas



Análise dos Resultados

Em uma escala de 1 a 5 onde 5 é muito e 1 é pouco, qual o nível de enjoo sentido ao decorrer da utilização?

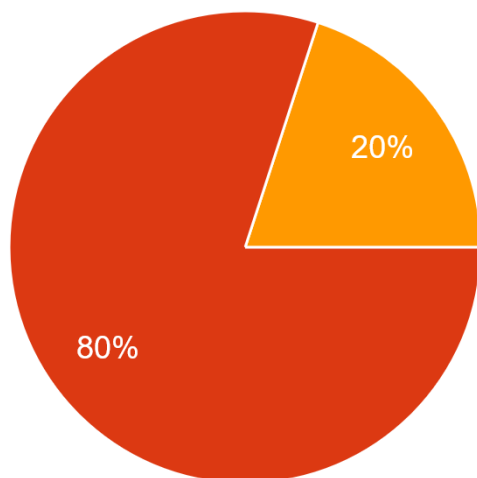
6 respostas



Análise dos Resultados

Se você considera que sabe dirigir com alguma proficiência, o quão parecido com a experiencia de dirigir um carro real você considera esse jogo?

5 respostas

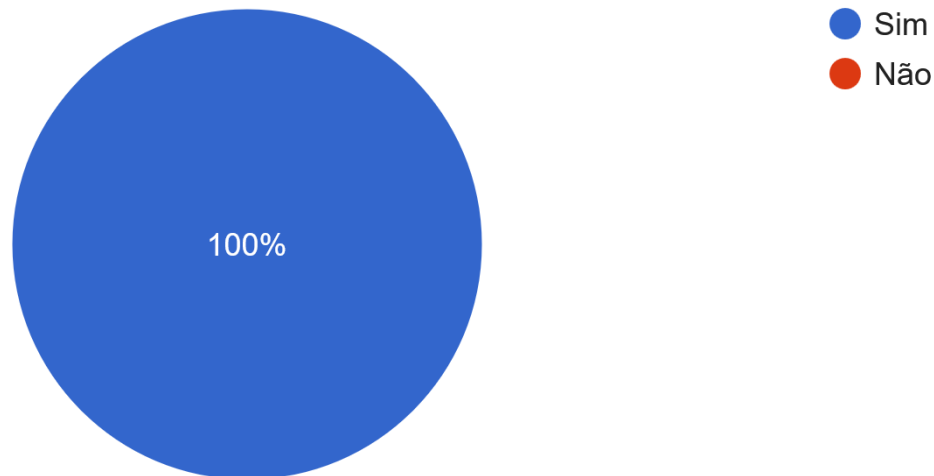


- Completamente igual
- Muito parecida
- Parecida
- Pouco parecida
- Completamente diferente

Análise dos Resultados

Se você considera que não sabe dirigir com alguma proficiência, esse jogo te ajudou a apresentar os conceitos de troca de marcha?

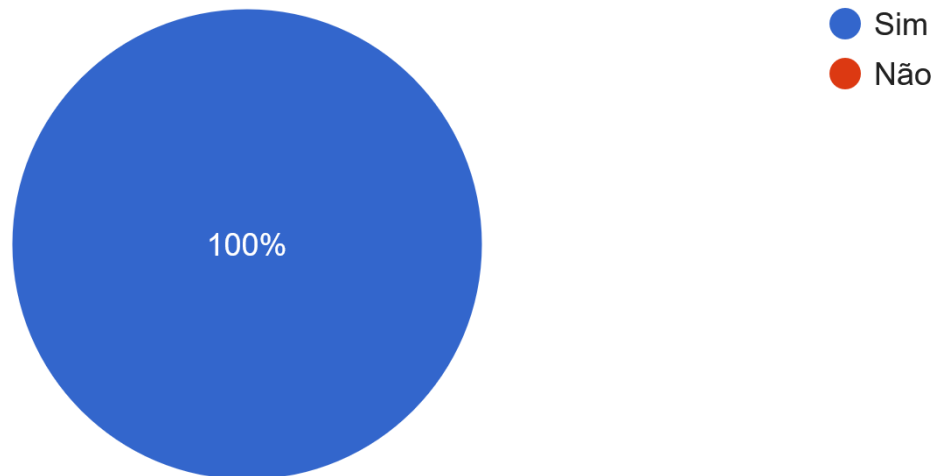
2 respostas



Análise dos Resultados

Você acha que o modo de pista molhada te ajudou a perceber os riscos de dirigir em rápido em momentos de chuva?

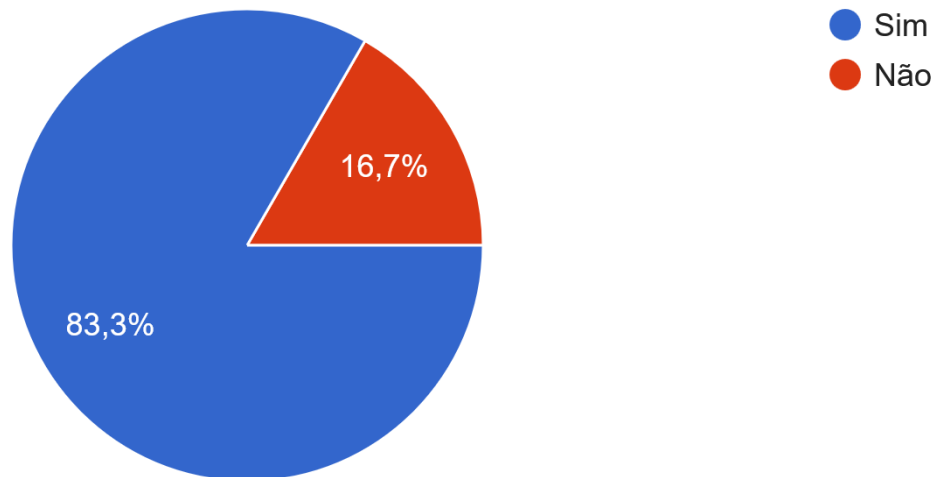
6 respostas



Análise dos Resultados

Você acha que o modo de pista de terra te ajudou a perceber a dificuldade de dirigir rápido em pistas com esse tipo de terreno?

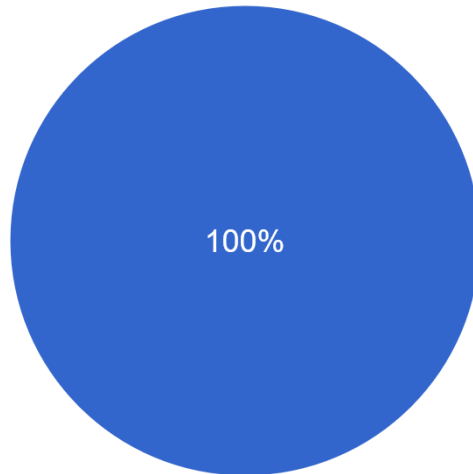
6 respostas



Análise dos Resultados

Qual é a sua avaliação da aplicação?

6 respostas



- Muito bom
- Bom
- Regular
- Insatisfatório

Conclusões

- Simulação do cenário real
- Alta imersão percebida
- Diversão e aprendizado garantido

Sugestões

- a) implementar um sistema de pontos para beneficiar boas práticas e negativar mais práticas na direção
- b) implementar gasto de combustível associado a maneira como o usuário dirige, melhorando ou piorando a economia do veículo
- c) implementar freio de mão para que o usuário possa perceber e sentir a diferença entre freio do motor, freio do pneu e freio de mão
- d) adicionar cenários com possibilidade de estacionar o veículo para que o usuário possa treinar baliza
- e) implementar a sinalização de setas para auxiliar no aprendizado da direção do veículo
- f) adicionar cenários noturnos com a função de faróis para que o usuário percebe seus usos no dia a dia

Demonstração

- Vídeo
- Prática