



UFRR

**PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

COMPUTAÇÃO GRÁFICA

RELATÓRIO DO TRABALHO FINAL: DESENVOLVIMENTO DE JOGO DE CORRIDA RACING PRO

ALUNOS:

Luis Henrique Matos Sales - 2017020657

Pedro Aleph Gomes de Souza Vasconcelos – 2016.007150

**Maio de 2021
Boa Vista/Roraima**

Introdução

Este jogo foi produzido com a finalidade de observar as técnicas de computação gráficas sendo usadas na prática pela engine unity. É um jogo simples mas já é possível observar como muita teoria foi utilizada para torná-lo possível, o que passa despercebido, mas nós tentamos identificá-las mesmo que no mínimo.

Roteiro do Jogo

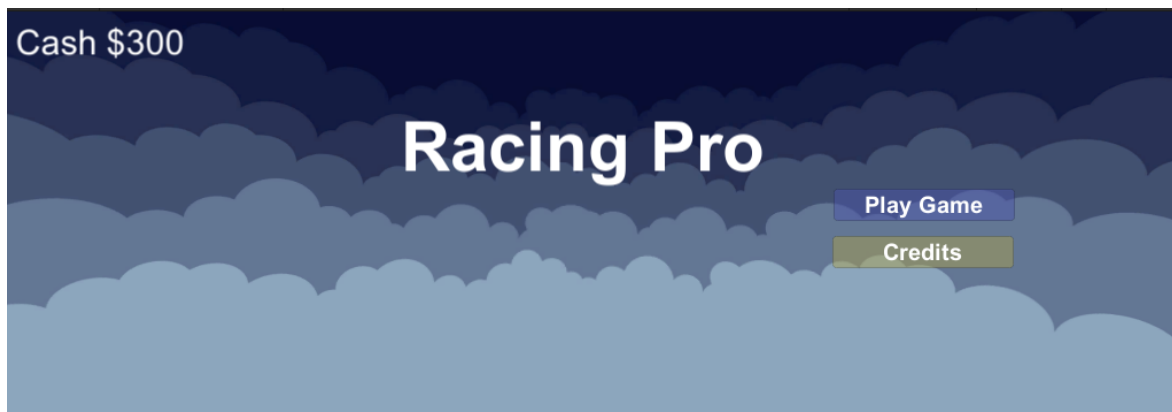
Experimente este novo modelo de carro na prática, você pode escolher qualquer cor desde que ela seja uma das três disponíveis, e imediatamente faça o seu teste drive em três modalidades com os melhores gráficos disponíveis. Jogue agora mesmo!

Detalhes do Jogo

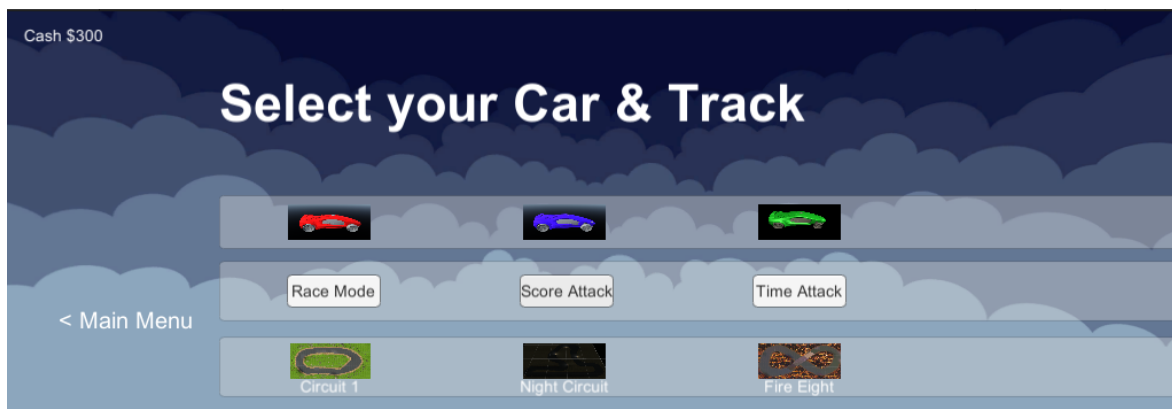
Tela inicial do jogo:



Ao clicar seguimos para o menu principal:



Ao clicar em Play Game seguiremos para a seleção de carros, logo depois a seleção de tipo de corrida e enfim o circuito, há três seleções para cada:



No race mode, o jogador compete contra uma IA, para quem atravessa primeiro a linha de chegada em 2 voltas, imagem da largada no primeiro circuito:



No Score Attack, o objetivo é coletar os cubos no caminho para conseguir aumentar seu score. Nesta imagem no circuito 2, o jogador deve ter mais cuidado na pista, observando a área iluminada pelo farol do carro na pista escurecida:



No Time Attack, o jogador deve medir seu tempo até atingir a linha de chegada, imagem no terceiro circuito:



Detalhes Técnicos

Câmera do carro

Para a criação da câmera que acompanha o carro foi inserido primeiramente um cubo dentro do objeto carro e dentro do cubo foi inserido a câmera para que a mesma se mantivesse estável pois o cubo apenas acompanha o eixo y do carro, a troca de câmera se deve ao fato de haver três câmeras inseridas como filhas do cubo sendo uma para visão normal, uma para visão de longe e uma para primeira pessoa sendo a troca feita usando a tecla C no teclado.

Os Carros

Ambos os carros presentes no game sendo o do jogador e o da máquina são provenientes do pacote de assets “vehicles” onde já existem prefabs com script tanto para o carro baseado pontos de guia, tanto para o carro do jogador. Para o carro da máquina percorrer o circuito foi criado um objeto Tracker em que foi configurado por padrão para o carro fosse em direção do mesmo, quando o carro da máquina atinge o ponto Tracker a posição deste ponto se altera para outro marcador presente no circuito e assim é feito sucessivamente.

Mini Mapa

O mini mapa se trata da visão de uma câmera posicionada acima do carro do jogador apontada para baixo, para exibir a imagem no canvas foi criado um render texture com o nome miniMap e então esse render texture foi posto no campo Target texture da câmera, após isso foi posto um rawImage no canvas e então bastou arrastar o render texture miniMap para dentro do campo texture no rawImage criado.

Canvas

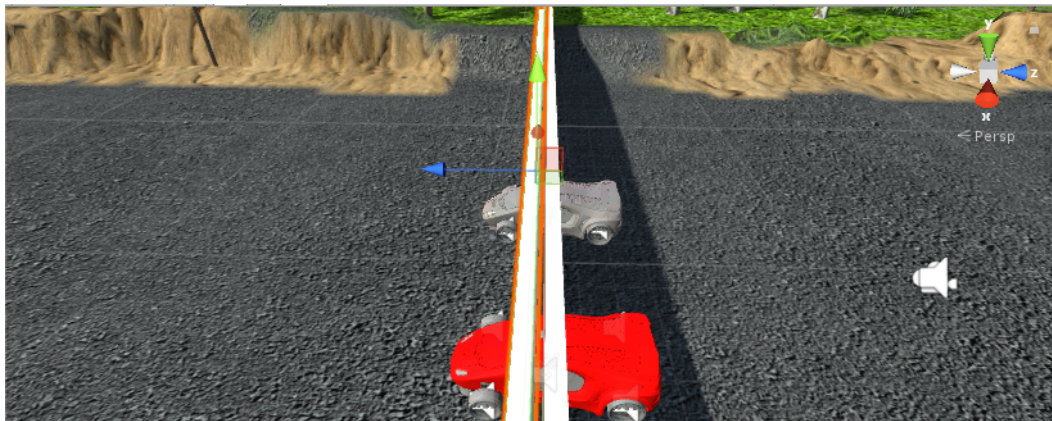
O Canvas presente no jogo exibe os status da corrida como o número de voltas, a posição do jogador, o tempo da volta e o tempo da melhor volta ou o score caso o jogador esteja no modo score attack.

Gatilhos de meia volta e volta completa

Ao passar pelo gatilho de meia volta o mesmo é desabilitado e o gatilho de volta completa é ativado para que ao passar pela linha de chegada novamente o gatilho de volta completa contabilize a volta, checar se o tempo de volta é menor do que a anterior para mostrar no painel canvas e checa se o jogador já está na última volta para poder ativar o gatilho de final de corrida. O gatilho de final de corrida zera a velocidade de ambos os carros, desabilita o controle do carro e adiciona 100 de cash para o jogador.

Posições

Para a mudança de posição foi criado no carro do jogador um cubo com o nome de CarPosition e com a tag CarPos, em seguida foram criados dois cubos anexados ao carro da máquina upPos e downPos que foram expandidos ficando praticamente em forma retangular para agirem como gatilhos de subir a posição e descer a posição respectivamente. Abaixo há uma imagem dos gatilhos no carro da máquina e os códigos que são responsáveis pela mudança de posição.



```
public class PosUp : MonoBehaviour {
    public GameObject positionDisplay;
    void OnTriggerExit(Collider other) {
        if (other.tag == "CarPos") {
            positionDisplay.GetComponent<Text> ().text = "1º
lugar";
        }
    }
}
```



```

public class PosDown : MonoBehaviour {
    public GameObject positionDisplay;
    void OnTriggerExit(Collider other) {
        if (other.tag == "CarPos") {
            positionDisplay.GetComponent<Text> ().text = "2º
lugar";
        }
    }
}

```

Ferramentas usadas

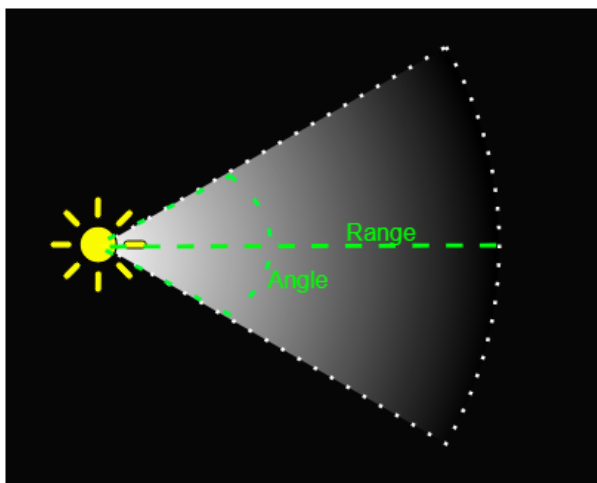
o jogo foi desenvolvido pelo unity, usando assets na store do mesmo



Principais técnicas de Computação Gráfica aplicadas

Iluminação e Sombreamento

O Unity possui modelos detalhados de como a luz trabalha para um resultado mais realístico, ou modelos simplificados para estilização. Em específico, para a iluminação dos faróis de cada carro na segunda pista, destacamos o tipo de luz spot light, em que dado um ponto específico, ilumina um intervalo limitado pelo ângulo resultando em iluminação em forma de cone.



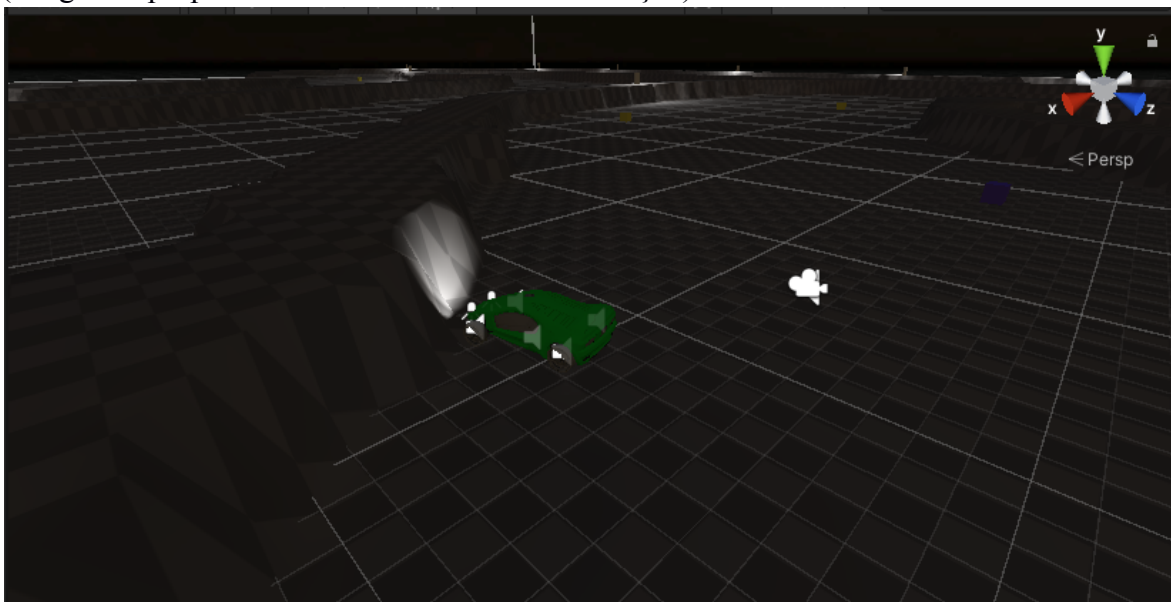
(imagem usada na descrição do tipo de luz na documentação do unity)

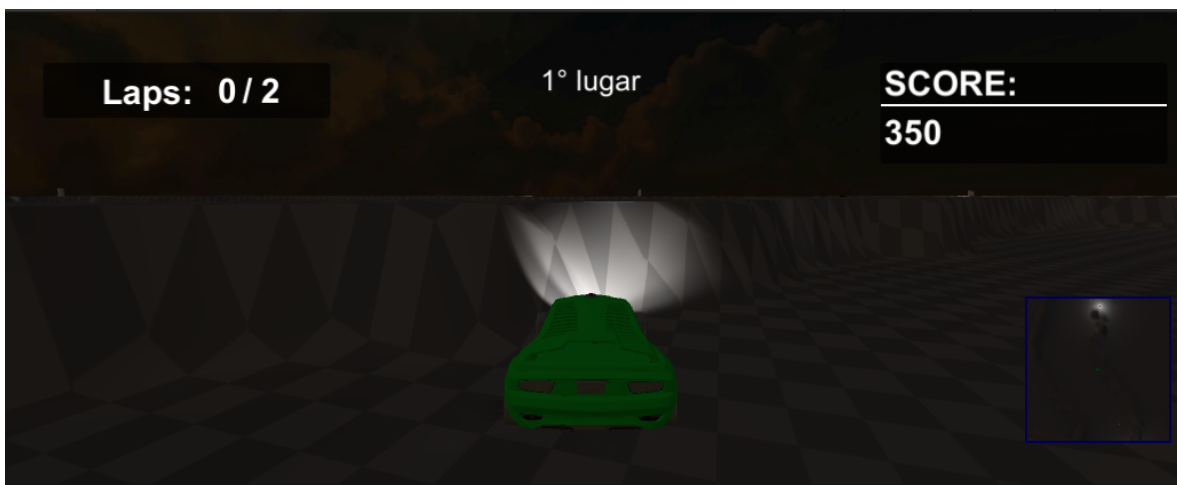
Projeção em Perspectiva/Paralelo

O Unity permite nas configurações da câmera, setar o tipo de projeção na cena.

Projection	Toggles the camera's capability to simulate perspective.
<i>Perspective</i>	Camera will render objects with perspective intact.
<i>Orthographic</i>	Camera will render objects uniformly, with no sense of perspective. NOTE: Deferred rendering is not supported in Orthographic mode. <u>Forward rendering</u> is always used.

(image das propriedades da câmera na documentação)





Anti aliasing

O unity por padrão já tem anti aliasing ativado , mas é possível desativar nas configurações de qualidade:

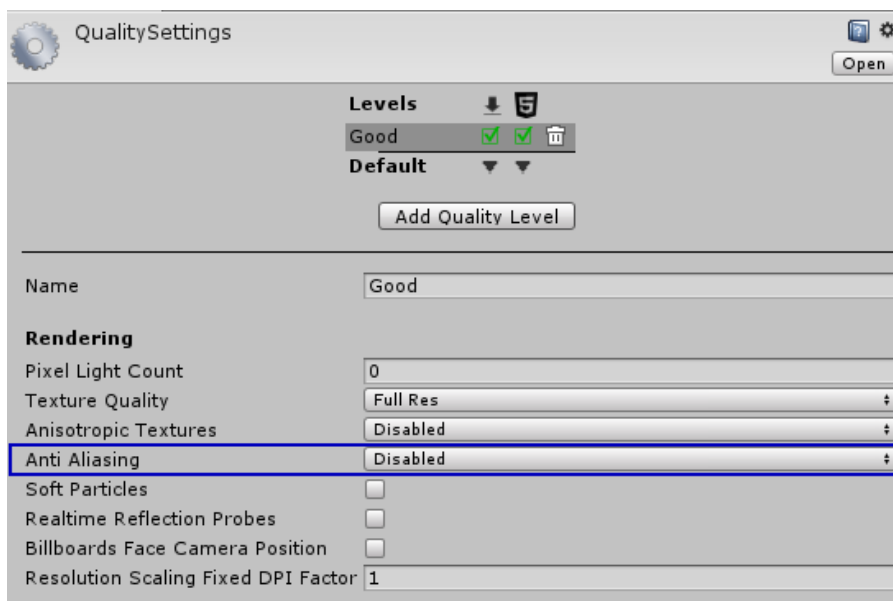


Imagem do jogo na escala 3.0



Anti Aliasing em 8x multi sampling

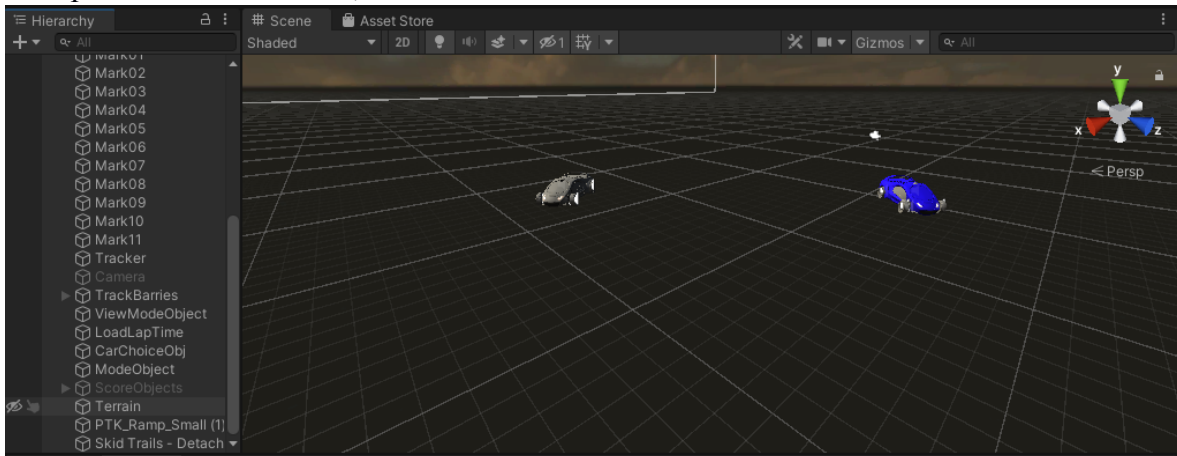


Anti Aliasing disabled

Tratamento de Superfícies escondidas

Não é possível demonstrar propriamente como ocorre tratamento, mas é possível mostrar que ele está lá, isso se dá por que o unity nos permite mostrar e esconder os gameObject na cena sem alterar sua visibilidade em jogo.

exemplo: ocultando terreno,



Conclusões

É impressionante a engine utilizada das técnicas de computação gráfica por baixo dos panos, dando ao usuário somente o dever montar a peças do seu jogo, sendo a forma que tais peças foram constituídas, a parte mais importante de todo o processo, o resto é apenas customização. Mesmo sendo um jogo simples, já é feito com técnicas que revolucionaram a computação gráfica ao longo dos anos.

Links

<https://docs.unity3d.com/Manual/Lighting.html>

<https://docs.unity3d.com/Manual/class-Camera.html>

[https://tiled2unity.readthedocs.io/en/latest/fixing-seams/#:~:text=Disable%20Anti%20Daliasing,-The%20largest%20contributor&text=Go%20into%20your%20Unity%20project's,Quality%20\)%20and%20turn%20it%20off.](https://tiled2unity.readthedocs.io/en/latest/fixing-seams/#:~:text=Disable%20Anti%20Daliasing,-The%20largest%20contributor&text=Go%20into%20your%20Unity%20project's,Quality%20)%20and%20turn%20it%20off.)