

Nitro

Game Design Document

BlackGround

**Versão de requisitos tecnológicos
5.0**

Introdução à Computação Gráfica
Universidade de Brasília
11 de Maio - 2016
blackgroundicg@gmail.com
GitHub: BlackGround-ICG/Nitro

Sumário

Capa

I Conceito

- 1 Objetivo do Jogo
- 2 Principais características

II História do Jogo

- 3 Como e aonde começa o jogo
- 4 Cenários do jogo e suas inter-relações
- 5 Troca de cenários
- 6 Final do Jogo
- 7 Como chegar ao final

III Controles

- 8 Movimentos à disposição
 - 8.1 Teclado
 - 8.2 Mouse

IV Requisitos Tecnológicos

- 9 Ferramentas de Codificação
- 10 Ferramentas de Modelagem

Tabela 1: Tabela de Resisão

Data	Descrição	Versão	Autor
29/04/2016	Início do documento	1.0	Elmar
01/05/2016	Controle	2.0	Elmar
02/05/2016	Formatação	3.0	Elmar
04/05/2016	Revisão	4.0	Elmar
11/05/2016	Requisitos Tecnológicos	5.0	Elmar

Parte I

Conceito

1 Objetivo do Jogo

O jogo consiste em uma competição do gênero corrida entre jogadores da mesma máquina, eles terão que capturar nitros (pequenos combustíveis para a locomoção do carro) ao longo do percurso, que possibilitam o carro correr. É um jogo em linha que possui pequenas curvas, vários obstáculos ao longo das fases e vários objetos para o jogador coletar. Os usuários terão suas voltas atualizadas em rankings para então comparar com amigos e demais jogadores.

O objetivo principal é fazer o menor tempo correndo nos percursos, no caso do projeto serão duas fases, passando de fases e ganhando pontos com o objetos coletados. Um plano futuro é que possamos cadastrar o jogo no Facebook e os jogadores possam se inscrever no jogo através de seus cadastros na rede social.

Tabela 2: Listagem

Manter carrinho no percurso
Passar de fases
Bater rank de melhor tempo e de pontos
Coletar objetos para ganhar pontos
Desviar de obstáculos
Jogar em computadores
Coletar nitros para reabastecer o carro
Nitros reabastecem o carro mas o deixam mais pesado e assim mais devagar
Usuário acelera, freia e vira o carro
Cenários feitos com o auxílio do Blender e Unity
Uso da OpenGL , SDL e GLSL
Interação com o usuário será pelo teclado e mouse

2 Principais características

Parte II

História do Jogo

3 Como e aonde começa o jogo

Quando o usuário iniciar o jogo, aparecerá uma tela inicial com uma lista de opções [8.2]:

1. Jogador seleciona primeira opção.

O usuário será encaminhado para a tela de jogo. O jogador se deparará com o carrinho parado no início da pista com a paisagem em volta, uma contagem de 3 segundos regressivos se inicia, então o cronômetro de tempo de corrida também começa e o carro passa a perder combustível, o jogador então usa os botões do teclado para começar a correr.

2. Jogador seleciona segunda opção.

O usuário será encaminhado para um tela de rankings dos jogadores que jogaram na máquina.

3. Jogador seleciona terceira opção.

Será mostrado ao usuário uma tela com as funções dos botões do jogo, objetivo e relações de paisagens e cores.

4 Cenários do jogo e suas inter-relações

A ideia geral do jogo seria fazer infinitas fases tendo uma sequência de paisagens com determinada imagem conhecida mundialmente para cada nível e quando essa sequência acabar, seriam modificados apenas as cores, por exemplo, há uma sequência de 5 paisagens diferentes onde em cada paisagem contém uma referência visual mundialmente conhecida, quando chegasse ao nível 6, volta a paisagem 1, mas com cores diferentes.

Por conta das restrições de tempo, base e conhecimento, a ideia central é, com o auxílio do Blender, Unity e as linguagens, fazer apenas duas fases sem a necessidade do esquema de cores. A primeira é um cenário com montanhas e árvores na neve, onde terá o monte Everest em algum ponto da paisagem. A segunda é um cenário desertico onde terá uma pirâmide do Egito em algum ponto da paisagem.

O modelo de cenário será parecido com o modelo teste deste vídeo, [YouTube](#).

5 Troca de cenários

Como o jogo em questão terá apenas duas fases, a troca de cenário será feita no mesmo mundo, onde a partir da distância percorrida pelo carro, o jogo saberá que haverá um troca de cenário e assim a dificuldade aumentará junto com as características da paisagem.

6 Final do Jogo

Ao final da segunda fase terá uma bandeira de linha de chegada. Quando o jogador chegar a linha, o jogo encerra mostrando ao usuário uma tabela com as informações de tempo de corrida, sua colocação no rank principal e uma opção de voltar a tela inicial e outra de reiniciar a corrida.

7 Como chegar ao final

O jogador terá que passar por todo o percurso das duas fases, desviando dos obstáculos, coletando os combustíveis e os objetos de pontos. Se o competidor não coletar os nitros e a gasolina acabar ou esbarrar em algum obstáculo, ele perde e aparece uma tabela de informações do tempo de corrida, informações de pontuação e duas opções, reiniciar corrida ou voltar a tela inicial . Se o competidor seguir desviando dos obstáculos, coletando gasolina e atravessar as duas fases por completo, ele termina o jogo.

Parte III

Controles

8 Movimentos à disposição

As imagens a seguir mostram as relação dos comandos no teclado e os botões que o usuário devera clicar com mouse.

8.1 Teclado

Nas telas, movimentação, vitória e derrota, o usuário terá apenas opções de clique no teclado nos botões citados nas imagens.

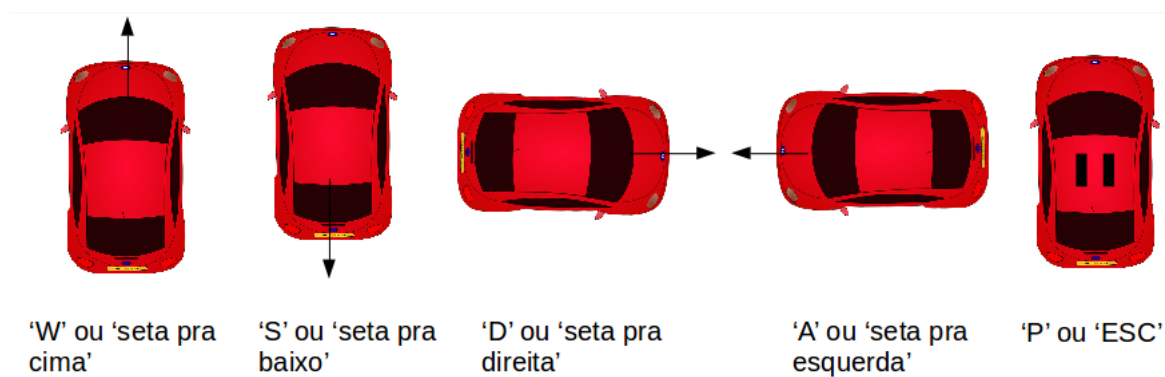


Figura 1: Imagem ilustrativa das relações das teclas com o carro

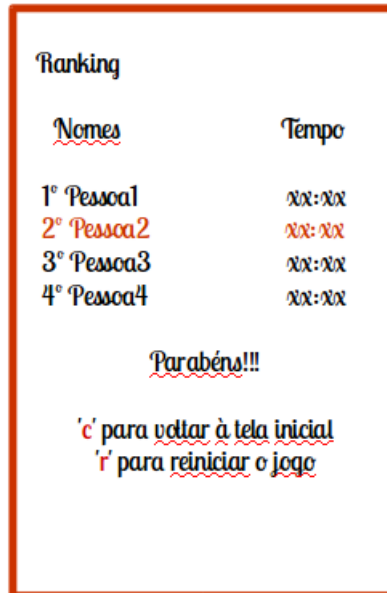


Figura 2: Imagem ilustrativa da tela de vitória

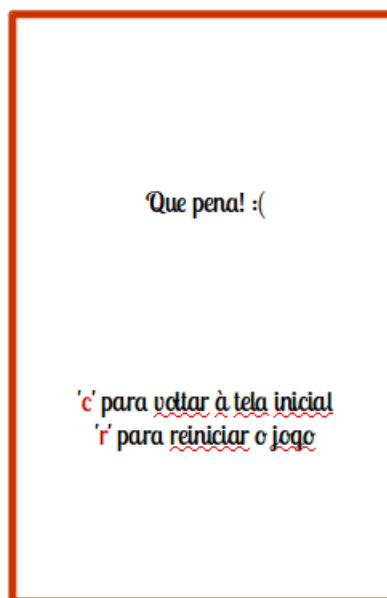


Figura 3: Imagem ilustrativa da tela de derrota

8.2 Mouse

Nas telas, inicial e de ranks, o usuário terá apenas opções de clique com o mouse.



Figura 4: Imagem ilustrativa da tela inicial



Figura 5: Imagem ilustrativa da tela de rankings

Parte IV

Requisitos Tecnológicos

9 Ferramentas de Codificação

Para realização do projeto, uma das ferramentas mais importantes disponível para uso é a sofisticada API OpenGL, pois ela permite a criação de aplicações gráficas 2D e 3D em tempo real. A OpenGL dá suporte à configuração de várias características dentro da criação de cena de jogos, como exemplo, a iluminação, animação, texturização, entre mais diversos efeitos. Serão utilizadas várias bibliotecas relacionadas a API em questão, como a GLU, GLUT etc.

Uma das bibliotecas usadas para complementar a OpenGL estabelecendo saída gráfica e capturando ações do usuário com mouse e teclado é a Simple DirectMedia Layer (SDL), onde além de todo esse suporte, ela também ajuda nas operações de pixel 2D, manipulação de eventos, som etc.

A linguagem que dará suporte para programação direta no hardware, ou seja, disponibilizará aos desenvolvedores um controle direto a pipeline de gráficos, é a OpenGL Shading Language (GLSL) conhecida também como uma linguagem de shading da OpenGL. De alguma forma tem que se encontrar os erros cometidos nos códigos, por isso para a realização da depuração da implementação será usado o glslDebugger que é uma ferramenta de debugging para programas OpenGL, onde passa linha por linha no código.

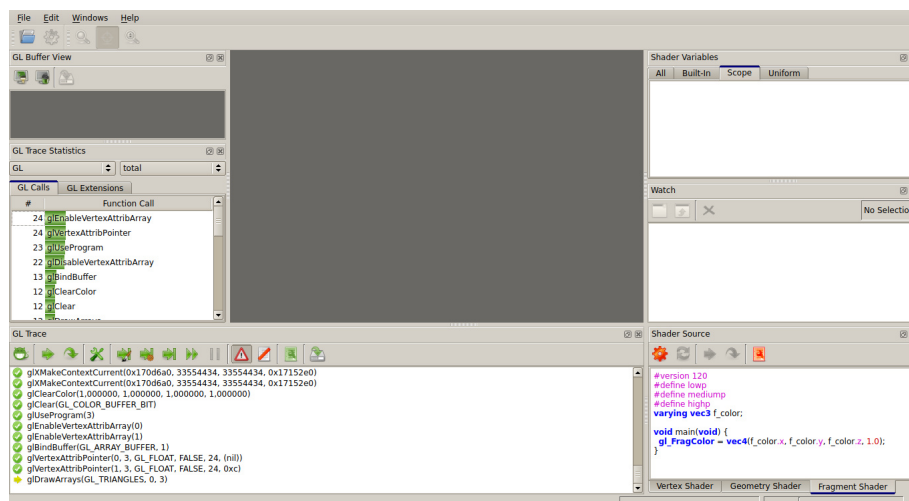


Figura 6: Layout do glslDebugger

A game engine será a Unity3D que possuem suporte para multiplataforma, altamente otimizado, realiza renderizações e uma das coisas mais chamativas é e que o Unity aceita três linguagens de programação, Boo, JavaScript e C#, onde o C será usado como a linguagem de script para o projeto pode ser bem parecido com o c e C++, facilitando assim o andamento da criação do jogo, pois os desenvolvedores já possuem conhecimentos prévios sobre as linguagens citadas. Os códigos serão compilados diretamente no terminal do computador.

10 Ferramentas de Modelagem

A ferramenta para construir as cenas fazendo toda a modelagem 3D, será o Blender. É um programa de fácil manipulação, devolve resultados extramente satisfatórios, sistema de animação avanado, exporta seus arquivos para uma integração com o Unity e por ser um software aberto, pode-se programar puglins livremente para ele, além de várias outras vantagens.