

Universidade Federal de São Carlos – Campus Sorocaba  
Ciência da Computação

Data de entrega: 21/07/2017  
Prof. Dr. Mario Augusto de Souza Liziér

Daniel Liberato de Jesus - 552127  
Vitor Fernando Souza Silva - 552488

## Índice

1.Introdução	3
2.Desenvolvimento do projeto	3
3.Compilação do jogo	4
4.Manual do jogo	4
5.Requisitos atendidos	5
6.Funcionalidades futuras	5
7.Considerações Finais	5

## 1. Introdução

Este jogo desenvolvido utilizando a Modern OpenGL inspirado nos clássicos de corrida, mais especificamente os de Fórmula 1, onde o jogador deve percorrer um traçado pré determinado com um carro no menor tempo possível.

O veículo pode andar para frente, para trás, virar para os lados. Ao mesmo tempo, desde que não sejam movimentos opostos.

## 2. Desenvolvimento do Projeto

O projeto foi desenvolvido em linguagem C++, utilizando as seguintes bibliotecas:

- Glew: biblioteca C / C ++ multiplataforma que ajuda na consulta e carregamento de extensões OpenGL;
- Glm: renderização de software (*raytracing* / rasterização), processamento de imagem, simulações físicas e qualquer contexto de desenvolvimento que exija uma biblioteca de matemática simples e conveniente;
- SOIL2: É uma pequena biblioteca C usada principalmente para carregar texturas no OpenGL. Baseia-se em stb\_image, o código de domínio público;
- Glfw 3: API simples e independente de plataforma para criar janelas, contextos e superfícies, leitura de entrada, gerenciamento de eventos, etc;
- Assimp: biblioteca portátil de código aberto para importar vários formatos de modelos 3D de forma uniforme.

O desenvolvimento foi baseado nos seguintes tutoriais:

- <https://open.gl/>
- <http://learnopengl.com/>
- <https://github.com/SonarSystems/Modern-OpenGL-Tutorials>

Foram utilizados modelos 3D de terceiros encontrados nos seguintes links:

Ferrari F1 : <https://clara.io/view/dcb0b529-b3a9-421e-a1ad-dabfd6717bbc>

Road: <https://free3d.com/3d-model/street-system-v10-48448.html> (editada à mão no blender).

Durante a execução do projeto encontramos muitas dificuldades. Principalmente ao que tange o uso de modelos de terceiros. Muitas vezes estavam incompletos, ou não eram compatíveis com a versão do OpenGL.

Houve muito impasse também na questão da implementação dos movimentos laterais do carro, da implementação de funções matemáticas para iluminação e curva de Bézier. Algumas texturas também tomaram algum tempo para serem corrigidas.

Ambos os membros se encarregaram de funções parecidas. Enquanto um produzia as linhas de código, o outro verificava erros ou manipulava objetos para se enquadrarem ao projeto.

### 3. Compilação e Execução do Jogo

Como mencionado anteriormente, existe a necessidade de instalar determinadas bibliotecas para que o jogo possa ser executado.

Todo o desenvolvimento foi executado na seguinte plataforma:

**Linux Ubuntu - 16.04 LTS - Xenial Xerus**

- A instalação pode ser feita de várias formas, mas para facilitar, pode-se executar a seguinte linha no terminal:

```
apt-get install cmake make g++ libglew-dev libglfw3 libglfw3-dev libglm-dev libx11-dev libxi-dev libgl1-mesa-dev libglu1-mesa-dev libxrandr-dev libxext-dev libxi-dev libsoil-dev assimp-utils libassimp-dev
```

- Feito isso deve-se entrar na pasta do jogo e compilar com a seguinte linha:

```
g++ -o exec main.cpp Model.h Shader.h Camera.h -IGLEW -lglfw -IGL -IX11 -lpthread -IXrandr -IXi -lassimp -ISOIL
```

- A execução é feita normalmente

```
./exec
```

### 4. Como jogar

Como mencionado anteriormente, o veículo se dispõe de determinados movimentos. Aqui descrevemos como as ações são representadas:

- “W” - Anda para frente;
- “S” - Anda para trás;
- “A” - Anda para a esquerda;
- “D” - Anda para a direita;
- “1” - Câmera do piloto (*first person view*);
- “2” - Câmera aérea (*third person view*);
- “Movimento do mouse” - move a câmera;

## 5. Requisitos atendidos

- Dois objetos carregados de arquivos;
- Uma forma simples;
- Cinco objetos no total;
- Dois movimentos distintos;
- ~~Uma curva de bézier;~~
- Textura em algum modelo obj;
- Textura em algum objeto simples;
- ~~Shader próprio com cálculo de iluminação próprio (em shader próprio);~~
- Duas posições distintas de câmeras;
- Alguma interação do usuário (teclado ou mouse);
- ~~Um objeto articulado (uso movimento relativo);~~

## 6. Funcionalidades futuras

Funcionalidades extras implementadas:

- Movimento de câmera com o mouse para facilitar a visualização do ambiente;
- Objetos e texturas de montanhas modeladas à mão;

Funcionalidades futuras:

- Skybox para construção construção do céu no cenário;
- Melhora na movimentação do objeto carro;
- Animação do movimento das rodas;
- Adição de sons do carro;
- Iluminação dinâmica;
- Suporte multiplayer;
- Iluminação;
- Detecção de colisão.

## 7. Considerações finais

Este projeto terá continuidade até que esteja satisfatoriamente jogável. A ideia é que este jogo tenha a mesma proposta de jogos como GRID - Autosport, Asseto Corsa, Project Cars.