



UNIVERSIDADE FEDERAL DE SÃO CARLOS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Daniel Leme Júnior

RA:725805

Igor Inácio de Carvalho Silva

RA:725804

Leonardo Zaccarias

RA: 620491

COMPUTAÇÃO GRÁFICA:
Relatório sobre animação

Sorocaba

2017

Tema, objetivos e funcionalidades

O objetivo desta animação é representar de forma simples o sistema solar, com o uso de formas e objetos 3D, assim como colocar em prática o conteúdo visto na disciplina de Computação Gráfica durante o semestre. Nela é possível ter diversas visões de câmera, que são alternadas com o clique do mouse. Há também outros componentes na animação que serão descritos ao decorrer do relatório.

Dificuldades, bibliotecas utilizadas e divisão de tarefas

Dentre as dificuldades encontradas, podemos destacar três. A utilização do Javascript e Html, já que nenhum dos criadores da animação está acostumado a utilizar tais linguagens, a implementação correta da trajetória do meteoro através de uma curva de bézier definida por quatro pontos, e também a criação de um shader com cálculo próprio de iluminação.

Em relação as bibliotecas, utilizamos apenas a three.js, que é usada para mostrar objetos 3D animados em um navegador.

Para otimizarmos o tempo e não sobrecarregarmos nenhum dos membros do grupo, dividimos o trabalho (animação + documentação) da seguinte forma:

- Daniel ficou responsável por implementar o plano de fundo estrelado, criar o shader de iluminação, utilizar shaders já existentes (fragment e vertex) e aplicar textura no Sol utilizando-os.
- Igor ficou responsável por colocar e aplicar textura nos diversos objetos contidos na animação, implementar a curva de bézier, controlar a troca de câmeras e fazer a documentação final.
- Leonardo ficou responsável por definir as posições, escalas, rotações e translações dos objetos contidos na animação, assim como a implementar o anel de saturno. Além disso, organizou todo o código e as pastas do GitHub.

Manual de execução

A animação conta com apenas uma interação direta com o usuário, que é a troca de câmeras. A troca é feita através de um simples clique no mouse.

Especificação e cumprimento dos requisitos

Foram especificados os seguintes requisitos para o desenvolvimento da animação:

- Dois objetos carregados de arquivos
- Utilização de uma forma simples (quadrado, círculo, esfera, etc.)
- Total de cinco objetos
- Dois movimentos distintos
- Movimentação de um dos objetos seguindo uma curva de bézier
- Aplicação de textura em um dos objetos carregados e também na forma simples
- Criação de um shader com cálculo de iluminação próprio
- Duas posições distintas de câmera
- Iteração do usuário
- Uso de movimento relativo
- Detecção de colisão, utilização de mais objetos, shaders, texturas, câmeras, entre outras funcionalidades (requisitos extras/opcionais)

Segue a indicação dos requisitos atendidos:

- O satélite e o meteoro foram carregados através de arquivo
- O Sol e os planetas foram representados por uma esfera e o anel de saturno, por um anel
- A animação contabiliza um total de dez objetos, sendo que os planetas têm movimentação distinta entre si
- A trajetória do meteoro segue a curva de bézier definida por quatro pontos

- Todos os objetos contêm textura, sendo que a textura do Sol foi modificada através de shaders utilizados na animação
- Shader de iluminação está contido no Material Shader de Netuno
- O satélite usufrui de movimento relativo em relação a Terra
- Há treze visões de câmera diferentes. Uma inicial, onde todos os planetas rotacionam através apenas de seu próprio eixo, outras oito que focam no Sol, nos planetas e no plano de fundo estrelado, além de mais três que dão uma visão aérea e lateral do sistema solar. É importante ressaltar que as dez primeiras câmeras só podem ser vistas no início da animação
- Música do Star Wars - Marcha Imperial como som de fundo do projeto

Considerações finais

Apesar de ter um conhecimento raso sobre computação gráfica, ficamos satisfeitos com o trabalho em questão, já que adquirimos conhecimento sobre javascript e animações. Durante o processo de desenvolvimento, foi possível entender melhor a parte teórica da disciplina, assim como sua importância para ter sucesso na implementação, já que em alguns momentos foi preciso recorrer aos slides das aulas teóricas para compreender o que estávamos fazendo ou compreendendo de forma errada. Em uma possível continuação desta animação, daríamos mais atenção aos detalhes e tentaríamos retratar os objetos mais próximos a realidade, além de aumentar a interação do usuário, fazendo com que a movimentação da câmera seja através do mouse, dando a sensação de liberdade a ele.

Links de apoio

www.threejs.org/docs

www.free3D.com

www.sharecg.com

www.youtube.com/watch?v=biZgx45MzqoLink5

stemkoski.github.io/Three.js/Shader-Halo.html

stemkoski.github.io/Three.js/Shader-Fireball.html

www.96methods.com/2012/01/three-js-simple-collision-detection/

<https://csantosbh.wordpress.com/2014/01/09/custom-shaders-with-three-js-uniforms-textures-and-lighting/>