

Universidade Federal de São Carlos
Campus Sorocaba

Projeto de Computação Gráfica
Fases 1 a 5

Grupo:

Felipe Furuse – 552186

Iago Piedade – 587567

João Victor Binatto – 552267

Sorocaba, 17 de Julho de 2017

Índice

1. INTRODUÇÃO.....	3
2. THREE.JS.....	3
3. ETAPA 1.....	3
3.1 IMPLEMENTAÇÃO.....	3
3.2 RESULTADOS.....	4
4. ETAPA 2.....	5
4.1 IMPLEMENTAÇÃO.....	5
4.2 RESULTADOS.....	5
5. ETAPA 3.....	5
5.1 IMPLEMENTAÇÃO.....	5
5.2 RESULTADOS.....	6
6. ETAPA 4.....	7
6.1 IMPLEMENTAÇÃO.....	7
6.2 RESULTADOS.....	7
7. ETAPA 5.....	8
7.1 IMPLEMENTAÇÃO.....	8
7.2 RESULTADOS.....	9

1. Introdução

Este relatório visa explicar a implementação e funcionamento de cada etapa do trabalho evoluindo passo a passo de forma incremental, até chegar à fase final.

O projeto, por sua vez, consiste na criação e manipulação de cenas contendo objetos 3D, câmeras, cenas, efeitos e luzes. Para isso foi utilizada a biblioteca Three.js, que é uma importante ferramenta para nós que estamos iniciando a caminhada nessa área de computação gráfica.

A cada etapa do projeto é aplicado um conceito novo complementando fases anteriores e caminhando em direção a animações e interações objeto-usuário mais próximas de aplicações reais.

2. Tree.js

Three.js é uma biblioteca/API JavaScript open-source utilizada para criar e apresentar gráficos animados 3D em um navegador. Dentre suas inúmeras funções temos algumas que se encaixam perfeitamente no que precisamos para obter as animações e resultados esperados em nossos projetos, como:

- renderizadores (Canvas, WebGL, and SVG);
- cenas – scenes (pode ser alterada no runtime, importada e exportada);
- câmeras (perspectiva e ortografia);
- luz (ponto, spot, direcional, e ambiente; objetos podem receber sombra);
- textura e materiais (Lambert, Phong, etc., inclui suporte a mapas);
- geometrias (incluindo linhas, cubos, esferas, polígonos, cilindros, partículas de sistemas e texto 3D);
- entre outros;

Para utilizá-la em qualquer projeto, basta apenas que seja importada sua biblioteca, de forma local ou remota e então todas as funções estão disponíveis para o programador.

3. Etapa 1

Na primeira etapa exibimos um objeto 3D e aplicamos zoom nele, fazendo com que se aproxime cada vez mais da câmera.

3.1 Implementação

Para que seja possível criar e exibir a animação na tela criamos nossos elementos essenciais, seguindo os seguintes passos:

- Primeiro instanciamos a câmera, passando os parâmetros de acordo com a visão que se deseja ter.

-Logo em seguida instanciamos a cena, que é o ambiente onde tudo está inserido.

-Jogamos então, a luz sobre a cena, para que seja possível visualizar os objetos e suas cores e texturas.

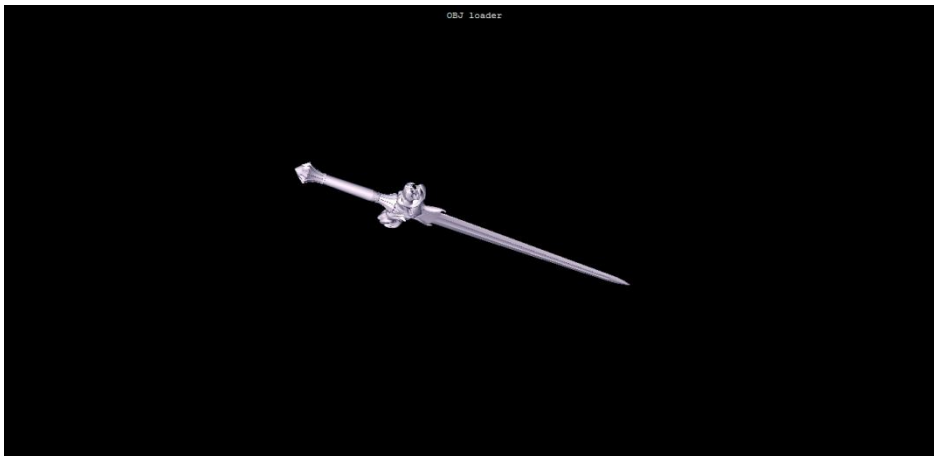
- Com tudo isso pronto, criamos o nosso objeto e o posicionamos dentro da cena.

- A ultima etapa é renderizar tudo, fazendo com que nossa animação seja mostrada na tela.

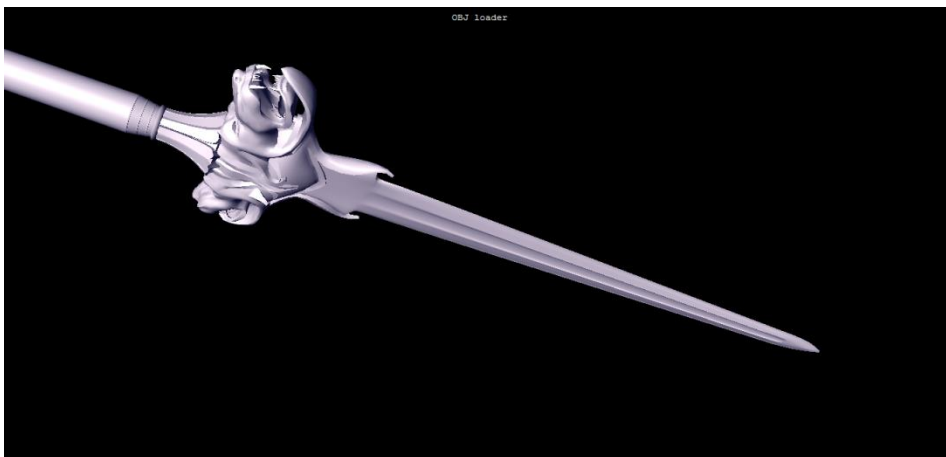
Passando os passos essenciais, temos a função de zoom, que pega as coordenadas do objeto e aproxima das coordenadas da câmera, fazendo com que ele fique cada vez mais próximo dos nossos olhos e de a impressão de aumento no tamanho.

3.2 Resultados

Visão inicial da cena:



Visão após aplicação do zoom:



4. Etapa 2

A etapa 2 consiste no carregamento de um objeto e a aplicação de uma textura.

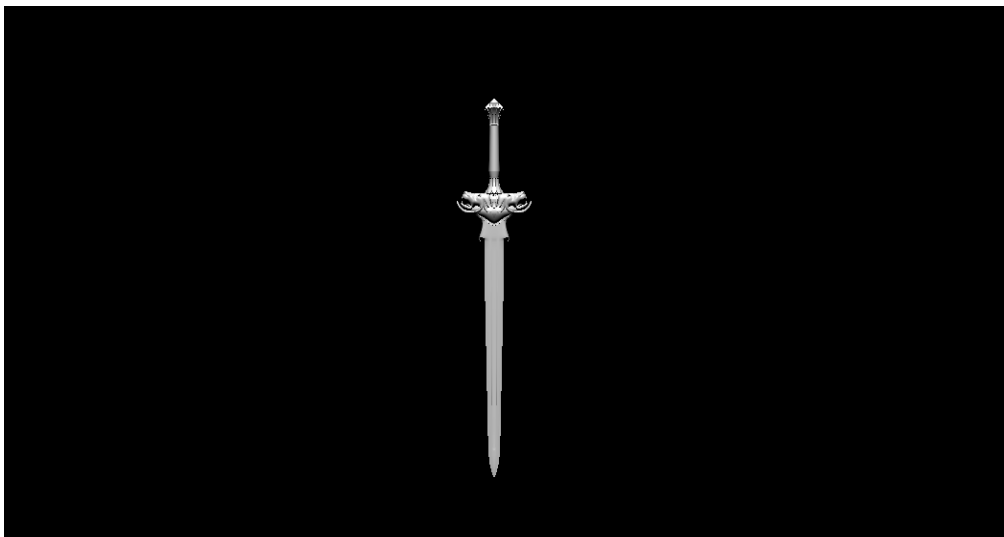
4.1 Implementação

Para realizar esta etapa tomamos como partida a primeira, então criamos a cena, a câmera e a luz da mesma forma, bem como carregamos o objeto 3D que fui usado anteriormente.

A diferença entre as duas etapas é que retiramos a função de zoom que fazia o objeto se aproximar da câmera, fazendo com que o mesmo permaneça parado, mudamos sua textura e posicionamento na tela, onde agora ela fica em pé.

4.2 Resultados

Visualização do objeto:



5. Etapa 3

Nesta etapa temos a criação de diversos objetos 3D, animação, movimento e textura dentro de uma única cena.

5.1. Implementação

Diferente das etapas anteriores, nesta temos vários objetos dentro da mesma cena, 3D, 2D, parados, animados, com textura e sem textura.

Os passos básicos de cena, câmera e luz permanecem os mesmos, porém agora temos que carregar todos nossos objetos nos locais corretos, imputando-lhes coordenadas específicas dentro da cena:

- sol e nuvens acima;
- bonecos abaixo a esquerda;
- cascos abaixo a direita;

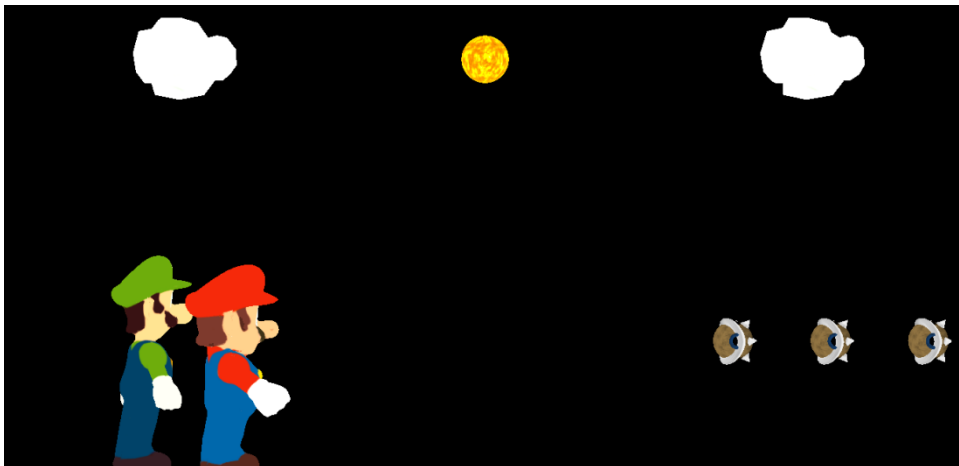
Com cada elemento posicionado corretamente, são aplicadas as animações desejadas objeto por objeto individualmente:

- nuvens e bonecos não recebem nenhuma animação, permanecendo parados;
- o sol e os cascos realizam uma rotação de 360°;

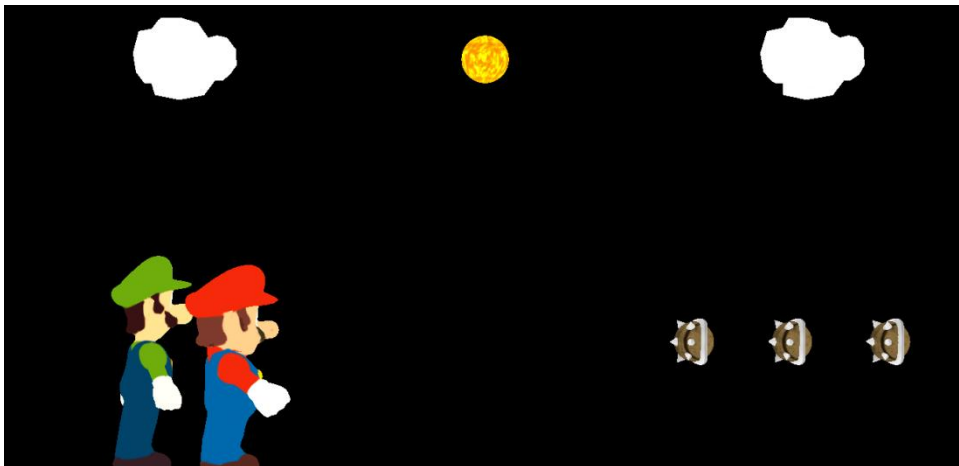
Também temos como movimentar a câmera, através das setas direcionais do teclado, permitindo ver a imagem 360°.

5.2 Resultados

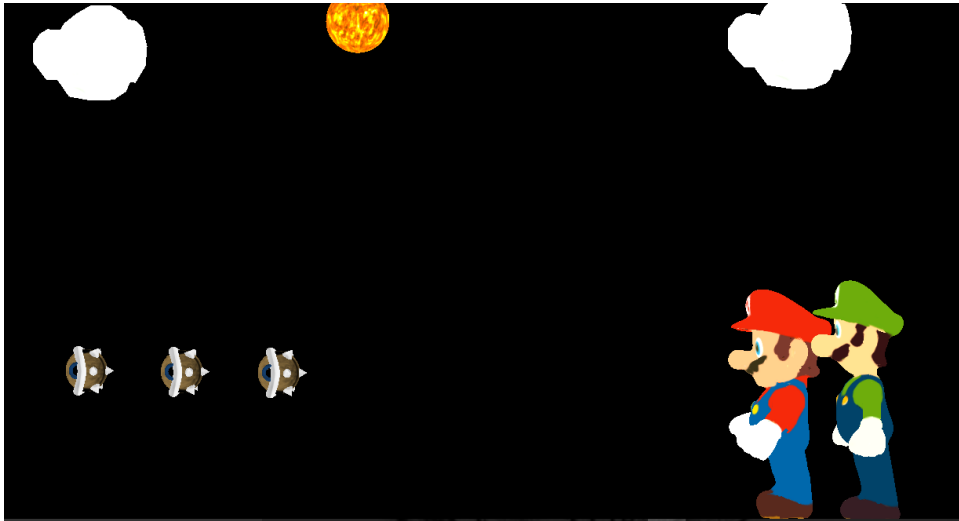
Objetos devidamente posicionados, cascos virados para a direita e sol com textura frontal:



Os objetos estáticos permanecem nos mesmo lugares, porem os cascos rotacionam, virando os espinhos para os bonecos e a textura do sol que antes estava na parte de trás vem para frente:



Rotação do ângulo da câmera em 180°, onde podemos notar que o boneco verde agora se encontra a frente do vermelho, pois a câmera se posicionou do outro lado de cena:



6. Etapa 4

Na etapa 4 foi incluído o movimento do boneco.

6.1 Implementação

Foram carregados os elementos básicos de cena, câmera e iluminação e carregamos o objeto do Mario e o casco, os objetos do sol e das nuvens não foi incluído pois não há diferença da fase anterior, dando mais ênfase na função nova.

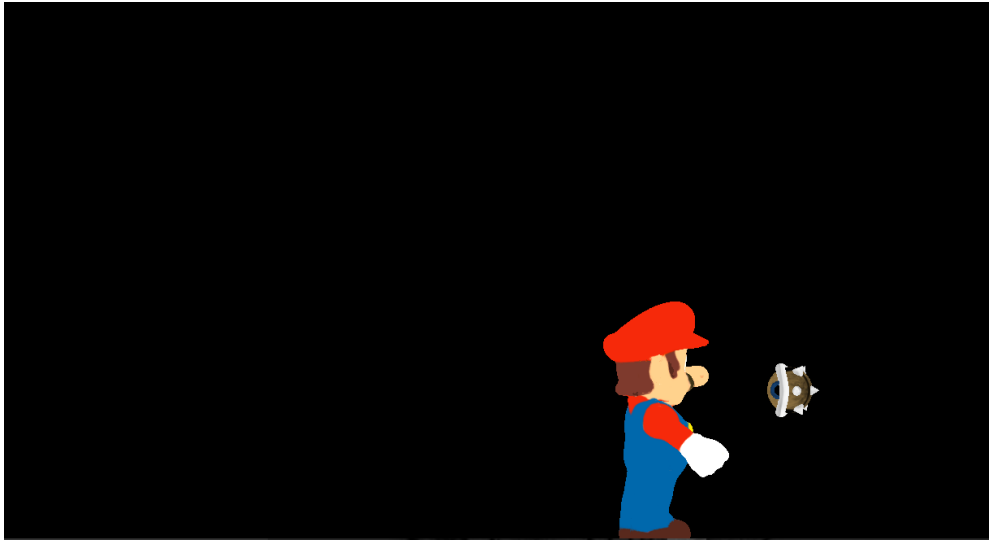
Com a cena criada é colocada então a função que confere movimento ao boneco, recebendo a entrada do teclado (seta a direita) e jogando o boneco posição por posição à direita a cada clique. O outro objeto não possui tal função, permanecendo assim parado no seu lugar, apenas rotacionando.

6.2 Resultados

Posição inicial:



Posição final após 10 cliques:



7. Etapa 5

Nesta etapa é colocado movimento nos objetos e a cena completa.

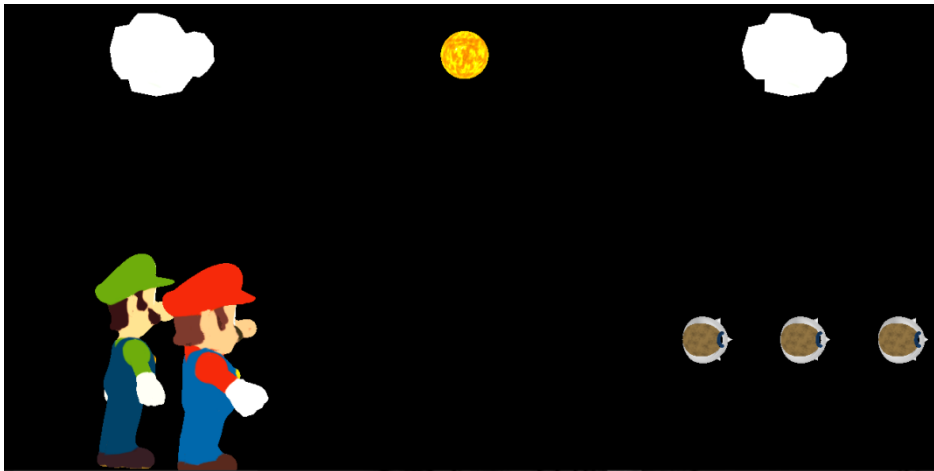
7.1 Implementação

Primeiramente criamos a mesma cena, câmera e luz da etapa 3 e carregamos os objetos do boneco do Mario, boneco do Luigi, o sol e os cascos mantendo as mesmas posições, animação e texturas.

A novidade nesta etapa é a inclusão do movimento de ambos os bonecos. Após carregá-los na imagem temos a função que identifica o clique na seta direcional da direita que faz o boneco do Mario se movimentar igual na etapa anterior e também temos o movimento do boneco do Luigi que permite que ele pule e volte ao chão, se mantendo no mesmo lugar após o movimento. Ambos possuem movimentos independentes.

7.2 Resultados

Posição inicial dos bonecos:



Posição após 10 cliques na seta para a direita e um clique na seta para cima:

