

Introdução

Este repositório é o projeto final da Disciplina de Computação Gráfica, do curso BCC-So 2017/1, da UFSCar Sorocaba. Desenvolvido pelos alunos Eduardo Rodrigues da Cruz, Gabriel Piovani e João Francisco de Paula.



Tema e Objetivos:

Como tema, pensamos em fazer algo como uma mistura da série de filmes Indiana Jones com a série de jogos Tomb Raider. Tivemos a ideia de criar uma cena como se fosse de um videogame com uma caverna ou tumba de pedra com um tesouro, algum inimigo, e uma câmera simulando o personagem jogável deste game.

Definimos como objetivos: a criação de uma salão quadrado, representando a caverna; disposição de quatro estátuas de ouro, para preencher a caverna; um diamante como tesouro, em cima de uma caixa ou baú; meios para que o usuário possa controlar o seu personagem; uma música de trilha sonora para o jogo; um inimigo sobrenatural, no caso um esqueleto, com um layout mesclando Minecraft e Lego.

Bônus: Na fase final pensamos em utilizar chamas para decorar o ambiente.

Funcionalidades:

A funcionalidade principal é a de controlar o seu personagem. Sendo um jogo em primeira pessoa, ao movimentar a câmera, o usuário movimenta o personagem. Por meio do teclado, o jogador pode realizar essa movimentação. Movendo o mouse, o jogador pode escolher pra onde olhar na caverna, e auxiliar sua movimentação, enquanto anda.

Uma segunda funcionalidade é a troca de câmeras, em que o jogador pode definir quais das três câmeras disponíveis utilizar (sendo uma para a sua movimentação, e as outras duas apenas para visualização da cena).

A terceira funcionalidade é a música ambiente, trilha sonora do jogo: a música tema principal dos filmes Indiana Jones.

Desenvolvimento do Projeto

Implementação geral e Bibliotecas:

No geral, o projeto foi desenvolvido utilizando WebGL, com o auxílio da biblioteca THREE, do site threejs.org, mas buscando utilizar sempre os conceitos aprendidos em sala de aula.

Também utilizamos:

Uma biblioteca especializada para as [chamas](#).

A ideia geral do [exemplo](#) de controle do Threejs, para a movimentação do personagem.

Divisão de Tarefas:

A teoria, o tema e os métodos a serem implementados em cada fase, foram discutidos informalmente entre os membros do grupo através de conversas pessoais e discussões em grupo de mensagens no Facebook criado para o projeto.

Durante as 5 fases iniciais, todos os membros desenvolveram igualmente as funcionalidades requisitadas. Na fase final, como o aluno Eduardo possuía uma maior facilidade com JavaScript, ele ficou a frente da implementação, com auxílio dos outros dois membros.

Dificuldades Encontradas:

As nossas maiores dificuldades foram em aprender o funcionamento e a utilização da biblioteca WebGL e no fim conseguir criar um projeto coeso, que fosse plausível. Uma segunda dificuldade esteve em abstrair os conceitos passados em aula e passá-los para a lógica especificada em cada fase.

Durante todo o semestre, a cada funcionalidade e conceito aprendido, mudamos as nossas ideias iniciais, adicionando novas ideias e excluindo ou aprimorando ideias mais complexas ou desnecessárias.

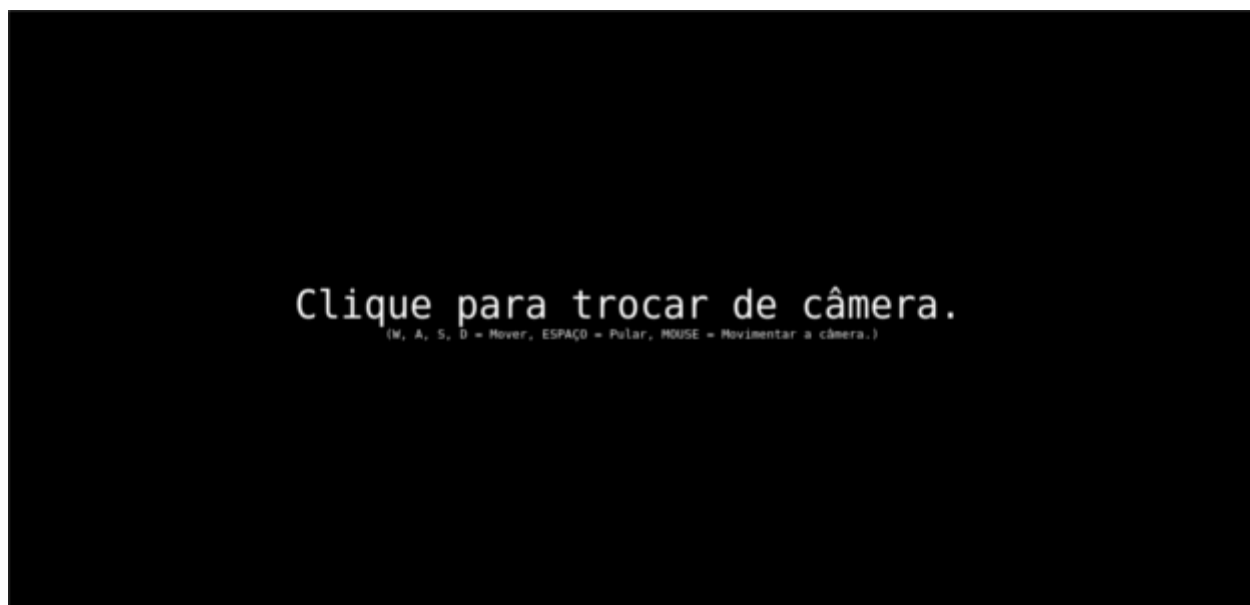
Manual de utilização

Arquivo:

O projeto foi desenvolvido para ser multiplataforma. Para executá-lo, basta ter um browser e abrir o arquivo [index.html](#), na pasta raiz do repositório.

Como a aplicação carrega arquivos externos (os objetos, texturas, etc), dependendo das restrições de segurança do navegador, ele pode impedir que a aplicação seja executada. A melhor solução é rodar um servidor local na pasta do projeto, e acessar o arquivo pelo servidor. Uma outra solução (menos segura) é alterar as configurações de segurança do seu navegador.

Gameplay:



- Use as teclas W-A-S-D para movimentar-se pela caverna.
- Pressione a tecla ESPAÇO para pular.
- Mova o mouse para movimentar a visão do personagem e orientar o seu deslocamento pela caverna.
- Clique com o botão (esquerdo ou direito) do mouse para trocar de câmera.

Requisitos atendidos:

Básicos:

Objetos carregados de arquivos:

Utilizamos dois objetos carregados de arquivos. Um objeto na forma de um homem:



E outro na forma de uma mulher:



Ambos para serem utilizados como representantes das estátuas de ouro, após a aplicação de texturas.

Formas simples:

Utilizamos diversas formas simples. Cubo (a caixa embaixo do diamante), octaedro (o diamante), paralelepípedos (as paredes, o teto e o chão. Todos com texturas.



Cinco objetos no total:

No total utilizamos mais que cinco objetos, sendo eles, no contexto da cena:

- Duas estátuas femininas.
- Duas estátuas masculinas.
- Um diamante.
- Uma caixa.
- As 4 paredes, o teto e o chão.

Movimentos distintos:

O diamante faz:

- Um movimento de rotação em torno do seu eixo y, girando no sentido horário.
- Um movimento de translação também no eixo y, subindo e descendo, sutilmente.

A caveira faz:

- Um movimento de rotação em torno do seu eixo y, dependendo de sua posição na Curva de Bézier, para dar a impressão de que anda sempre para frente.
- Um movimento de translação, ao longo da Curva de Bézier.
- Seus braços e pernas também rotacionam, enquanto ele anda.

Duas das câmeras se movimentam automaticamente, sempre olhando para paredes específicas, causando um movimento agradável da tela, indo até certo ponto e voltando.

Curva de Bézier:

Como dito acima, a caveira anda seguindo a Curva de Bézier.

Fizemos uma curva invisível que começa fora da caverna, atravessa as paredes, passa a

frente de uma das estátuas de ouro e termina também fora da caverna, tudo á altura do chão da caverna, para que a caveira não saia do chão.

Texturas:

Tanto nos objetos simples, quanto nos carregados de arquivos, todos os objetos em cena têm texturas carregadas de imagens nos formatos PNG ou JPG.

Posições de câmera:

Como dito acima, temos três posições de câmera distintas na cena.

- Uma para que o jogador movimente seu personagem pela caverna, andando, pulando e olhando.
- Duas fixas, olhando para paredes, uma movimentação automática e com ângulos e posições diferentes.

Interação do usuário:

Obs: Essa seção é uma cópia da "Gameplay".

O usuário pode:

- Usar as teclas W-A-S-D para movimentar-se pela caverna.
- Pressionar a tecla ESPAÇO para pular.
- Mover o mouse para movimentar a visão do personagem e orientar o seu deslocamento pela caverna.
- Clicar com o botão (esquerdo ou direito) do mouse para trocar de câmera.

Objeto articulado:



Criamos um inimigo caveira, numa mistura de Lego com Minecraft, articulado, utilizando várias formas simples (cubos, cilindros e paralelepípedos). Unidas por meio da classe "group" da Threejs, que une vários objetos, nos permitindo configurar o boneco (como um

grupo) a realizar os movimentos de translação e rotação, enquanto seus braços e pernas podem movem-se de forma relativa e independente.

Utilizamos uma textura branca com aspecto envelhecido e rachaduras em todas as partes do boneco, com exceção da cabeça, onde aplicamos a imagem do rosto da caveira do Lego.

Este boneco pode mexer as pernas e os braços na cena, enquanto anda automaticamente pela Curva de Bézier.

Extras:

- Como requisitos extras, temos primeiramente, o fato de termos utilizado mais texturas, objetos, e câmeras do que o mínimo pedido.
- A música, que inserimos para melhorar a experiência do usuário enquanto joga e para aumentar a imersão no tema (Indiana Jones e Tomb Raider).
- O fogo, no qual utilizamos uma biblioteca a parte para inseri-lo na cena, melhorando o visual e a ambientação.
- Detecção de colisão para a caveira:
Colocamos a curva de bézier começando e terminando fora da caverna para criar o problema de a caveira atravessar a parede e sumir da cena, ao chegar ao começo ou ao final da curva.
Para resolver isso, programamos a caveira, para ao chegar á uma posição limite (a parede), dar meia-volta, inverter o seu movimento e voltar seguindo a Curva de Bézier. Assim, ao bater na parede, voltar, ao invés de atravessá-la.

Como possíveis continuações futuras, pensamos na implementação de detecção de colisão para os outros objetos e para o personagem jogável.