

UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA
Computação Gráfica - Profª Dra. Regina Célia Coelho

Lara Croft World

Nome: Willian Dihanster Gomes de Oliveira

RA: 112269

SÃO JOSÉ DOS CAMPOS
11 de dezembro de 2018

Introdução

A Computação Gráfica (CG) é um objeto de estudo muito popular nos dias atuais. É utilizada em grandes filmes, jogos, *softwares*, *etc* e vem obtendo grandes resultados em suas aplicações.

Desde seu surgimento, representar visualmente seres humanos é uma tarefa comum e hoje em dia é possível representá-los com bastante realismo. No entanto, esta é uma tarefa difícil, tendo preciso um grande conhecimento na área, além de ferramentas especiais.

Para aplicações mais básicas, há pacotes de programação gráfica, como a OpenGL em que se é possível trabalhar com primitivas e transformações geométricas e ter uma experiência prática mais simples.

Sendo assim, neste trabalho a ideia é utilizar os conceitos de CG adquiridos e o pacote OpenGL na construção de uma humanóide que participe de uma cena e que tenha movimentos que lembrem um humano. Como forma de homenagem a primeira atriz digital de CG, o humanóide se baseia na personagem Lara Croft.

Objetivo

O objetivo da realização deste trabalho é unir todos os conhecimentos adquiridos da disciplina de Computação Gráfica na construção de um humanóide que interage com um ambiente e que se movimenta lembrando um humano.

Neste projeto, o humanóide construído é uma tentativa de homenagem a uma das personagens mais clássicas de computação gráfica, bem como a primeira atriz digital e personagem preferida (do mundo dos jogos) do autor, Lara Croft.

No cenário, a personagem se encontra em uma entrada para um templo, entre montanhas e árvores, um típico cenário para uma arqueóloga, como ela. E suas tarefas são, perseguir lanças que aparecem ao clique do mouse e então ela deve jogá-las para fora do lugar.

Além disso, há uma cena automática, onde a personagem anda pelo cenário com sua lupa e mapa, como se estivesse investigando o lugar, em busca de artefatos. Uma grande inspiração para toda a animação, como cenários e cores, foi o jogo Lara Croft GO, disponível para consoles e smartphones.

Metodologia:

Para o desenvolvimento deste projeto, foi utilizado a biblioteca OpenGL (freeglut) em linguagem C, com a IDE Codeblocks no sistema operacional Windows. Além disso, toda a cena foi construída por primitivas geométricas e transformações geométricas próprias da biblioteca.

- **Personagem**

Para a construção da personagem, foram utilizadas somente as primitivas geométricas e transformações geométricas disponíveis na OpenGL. Para a junção das partes foi utilizado o conceito da técnica de CSG (*Constructive Solid Geometry*), como visto em sala, considerando apenas a união de primitivas para montar outras formas, como pode ser observado na Figura 1, onde a cabeça e o cabelo da personagem são união de esferas.



Figura 1: Cabeça da personagem, construída por CSG.

Para a construção das tranças da heroína (marca registrada da personagem clássica), foi utilizado a técnica de *Sweep* translacional (ou extrusão), onde uma esfera foi translada 13 vezes, formando a trança da personagem, como pode ser visto na Figura 2.



Figura 2: Trança da personagem, construída por Sweep Translacional.

Mais detalhadamente, a cabeça é feita por duas esferas uma representando a cabeça e outra o cabelo. O olho, boca e óculos são esferas escaladas, o nariz é uma tetraedro e a trança foi obtida através do *Sweep* translacional de esferas.

Já para o tronco, há um cone que representa o tórax da personagem, mais acima uma esfera representando o busto. Atrás (nas costas) há um cubo e dois *torus*, que representam a mochila e as alças da mochila, respectivamente.

Para os braços, há o conceito de construção hierárquica (como visto em aula), com 3 “níveis”, sendo primeiro o braço, seguido de antebraço e a mão. Isto é, o braço mexe independentemente, mas influencia no antebraço e mão. Para o braço e antebraço foram utilizados cilindros, já para a mão foi utilizada uma esfera escalada. Além disso, há a presença de esferas entre as diferentes partes, para funcionarem como a articulação do modelo hierárquico, disfarçando o aspecto de descolamento entre as partes.

Os membros inferiores, também respeitam um modelo hierárquico de perna, joelho e pé. Para a região do quadril-perna foi utilizado dois cilindros marrons que representam o short. Há a presença de um cinto, feito por um *torus* e um cubo como fecho do cinto. A seguir, para a perna e joelho e bota, foram utilizados cilindros, com a presença de esferas para simular uma articulação.

O modelo final desenvolvido pode então ser conferido na Figura 3, a seguir.

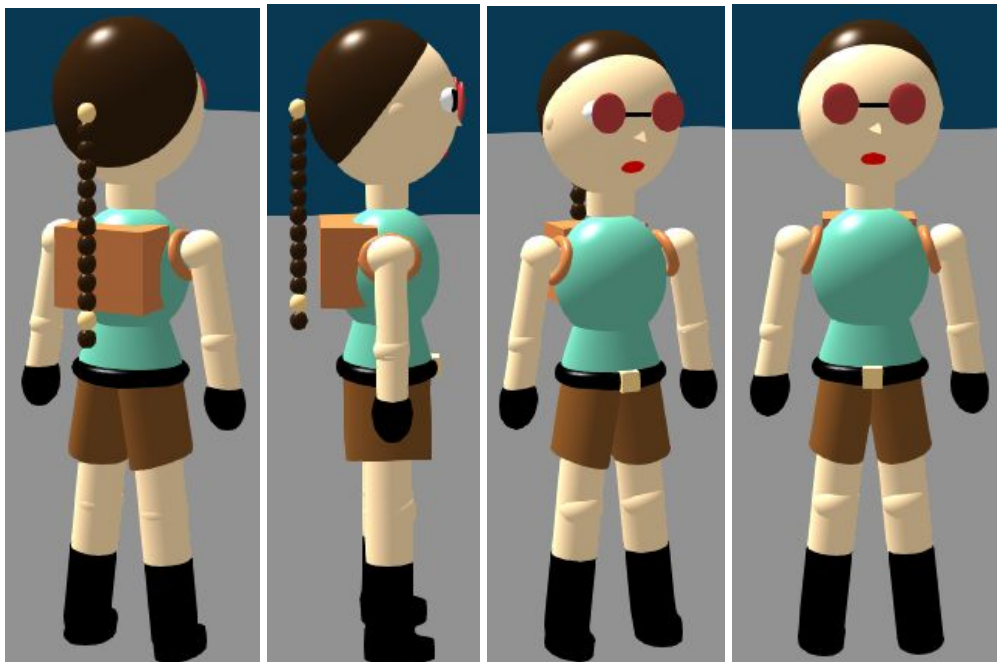


Figura 3: Vistas da personagem construída.

- **Objetos**

O objeto que aparece ao clique do mouse são lançadas como as da Figura 4 a seguir. Essa lança é construída pela união de um cilindro e um cone.



Figura 4: Lança que aparece ao clique do mouse.

Para a animação automática que a personagem deve realizar, caso seja pressionado um botão, dois novos objetos aparecem: uma lupa e uma tablet. Como mostra a Figura 5. A lupa foi construída a partir de um *torus* e um cilindro. Já o tablet, um cubo escalado.



Figura 5: Personagem segurando sua Lupa e Tablet.

- **Cenário**

Neste cenário, há uma certa quantidade de árvores que seguem o padrão visual da Figura 6. Essa árvore foi criada a partir de um dodecaedro (para a copa da árvore) e um cubo escalado para o tronco.

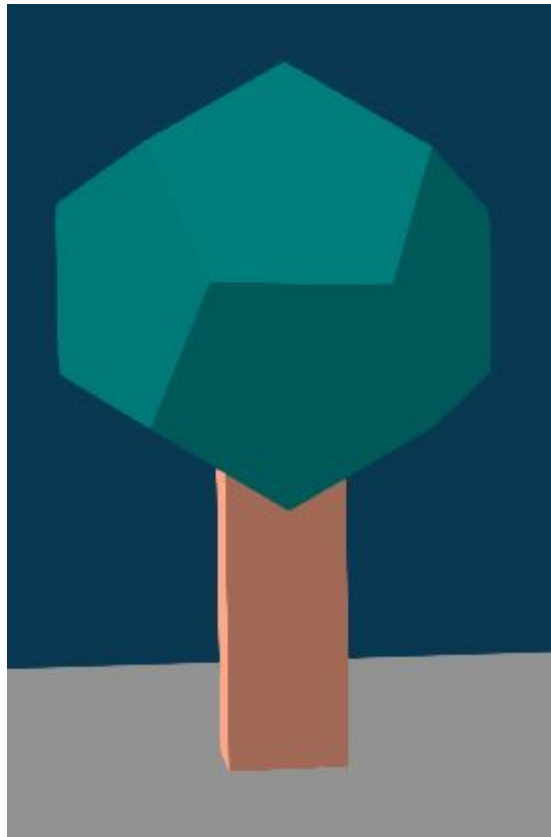


Figura 6: Árvores construída para a cena.

Uma outra parte do cenário são as ruínas, adicionadas na cena para gerar uma sensação de um cenário antigo, desgastado pelo tempo.

A pilastra foi construída por cubos escalados e transladados. E as rochas por esferas escaladas e transladadas, como mostra a Figura 7.

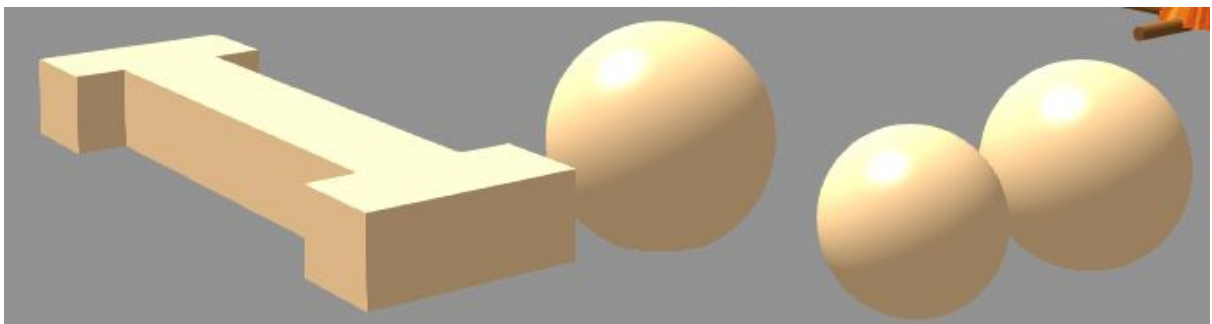


Figura 7: Ruínas da cena.

Há também uma fogueira (Figura 8) que foi construída por um cone e adição de uma textura com uma imagem de fogo, utilizando mapeamento esférico. A imagem foi baixada da internet e convertida para o formato *bmp*, com 24bpp. Além disso, a biblioteca para leitura e carregamento da imagem *bmp* foi feita pela utilização do código disponibilizada pela professora.

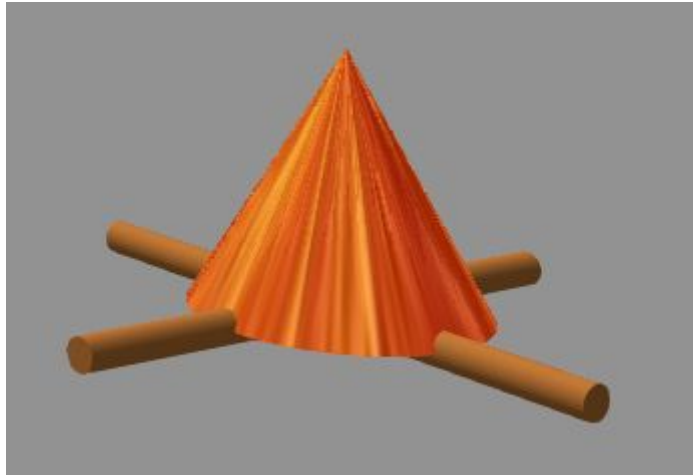


Figura 8: Fogueira utilizando textura de fogo.

Neste cenário há também uma montanha, gerada a partir de uma B-Spline (em 2D) como a vista em aula, com 6 pontos de controle e ordem 3. Após gerar a curva curva, os pontos formaram um polígono e esse polígono foi escalado e transladado para a posição final. No entanto, o polígono gerado é 2D e para tentar contornar a situação, foi feito um *Sweep* translacional, para que pelo menos, haja uma impressão de que a montanha seja 3D.

A Curva Spline transformada em polígono e a mesma Curva Spline com adição da técnica de *Sweep* translacional, pode ser visualizada na Figura 9.

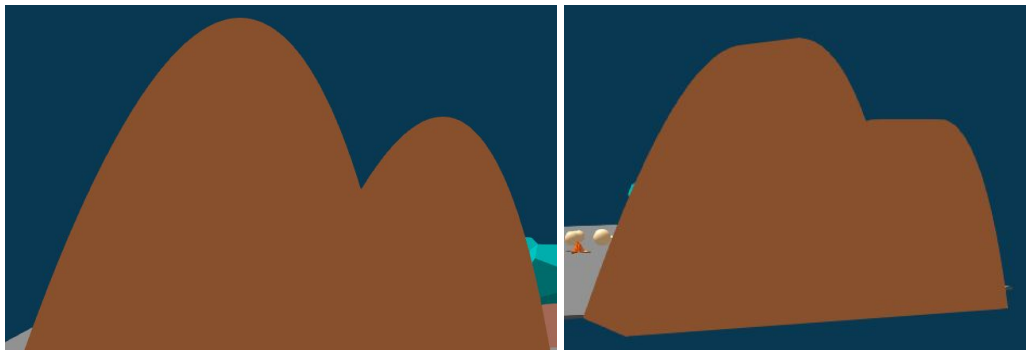


Figura 9: Montanha gerada por Curva Splines (Polígono) e Montanha gerada por Curva Splines (Polígono) com adição da técnica de *Sweep* translacional..

Por fim, há também um templo, gerado a partir de cubos escalados e duas estátuas na frente do templo, construídas por um cubo e esferas escaladas. O resultado pode ser conferido na Figura 10.

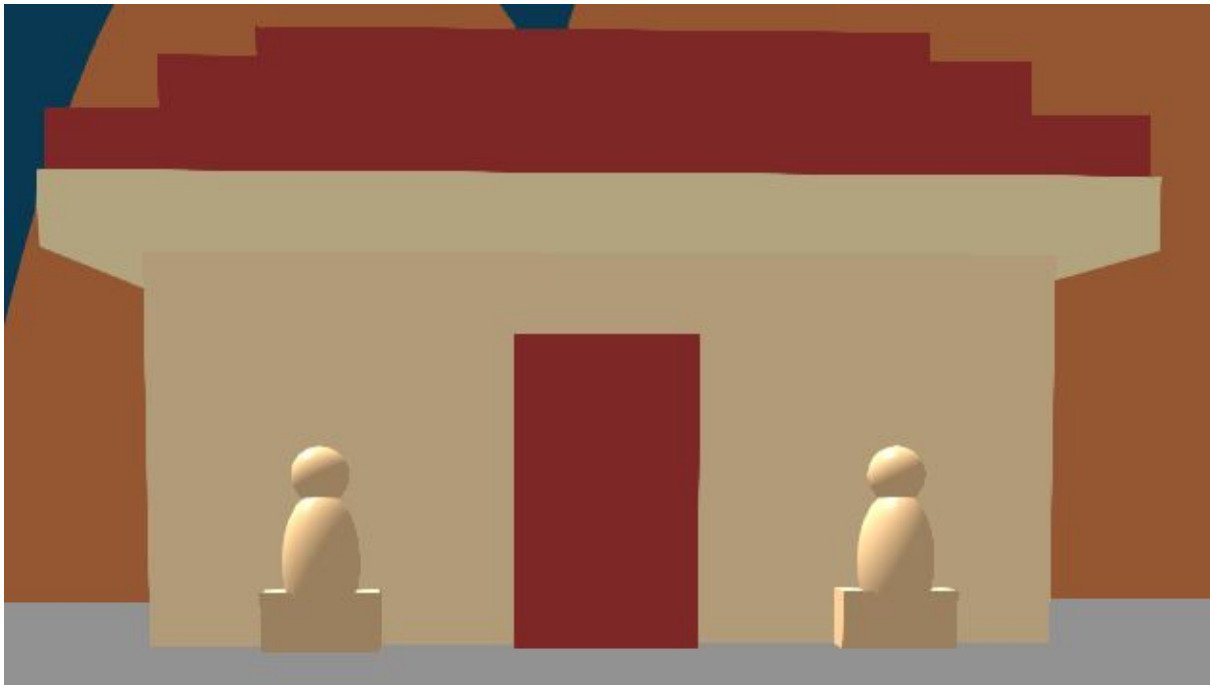


Figura 10: Templo da cena.

- **Movimento da Personagem**

Para que a personagem se movimente lembrando um humano, toda a personagem foi criada utilizando o conceito de construção hierárquica. Sendo assim, com o movimento das pernas, o joelho e pé também mexem. Portanto, ao clique do mouse, a flag de andando permanece em 1, até que termine sua tarefa. Enquanto a personagem não terminou, as variáveis de rotação sofrem um incremento utilizando o cálculo do seno e a posição da personagem é sempre incrementada. Primeiramente, checka-se se o ponto está à esquerda ou à direita e rotaciona a personagem. Em seguida, anda-se, transladando a personagem até o ponto (somente em Z) e depois em X, até chegar o ponto.

Para o movimento das partes do corpo pelo menu, primeiro, checka-se que parte do corpo deve-se mexer, setando uma flag correspondente. Em seguida, checka-se que seta está pressionada e então, a variável correspondente a parte do corpo escolhida, é incrementada ou decrementada.

- **Como rodar o trabalho**

Para rodar o trabalho, basta ter a IDE Codeblocks com a biblioteca OpenGL. (Caso a biblioteca instalada no seu computador seja a glut, será necessário trocar na seção do código de includes para `"#include <GL/glut.h>"`), pois o trabalho foi realizado a versão freeglut da biblioteca. Visto que essa foi a versão que funcionou no Windows e além disso, possui mais primitivas básicas, como a que desenha cilindros.

Além disso, pode ser necessário setar o diretório de trabalho, caso seja executado em um outro computador. Para que seja possível carregar o arquivo de textura.

- **Funções**

Neste trabalho entre as funções disponíveis estão a da animação automática, em que a personagem anda pelo cenário como se estivesse investigando o local, com uma lupa e um tablet. Esta função pode ser ativada pela tecla 'A' e finalizada pela tecla 'a'.

Uma outra funcionalidade é a de perseguir lanças, que aparecem ao clique do mouse. Ao clicar com o mouse, uma lança é gerada em uma posição aleatória da cena e então a personagem vai até a lança, pega ela e joga-a para fora da cena. Caso haja vários cliques, as lanças são desenhadas, mas aguardam na fila.

Além disso, há opções em um menu que pode ser acessado pressionando o botão direito do mouse. Neste menu há diversas opções, como a de finalizar o programa, rotacionar partes do corpo, iniciar a animação automática e carregar a cena toda ou apenas a personagem.

Também, há opções nas teclas do mouse. Com estas teclas é possível rotacionar a cena em X, Y ou Z, a personagem (em Y), finalizar o programa, dar *zoom in* e *zoom out* na cena, alterar o ângulo da abertura da câmera e aplicar a técnica de *panning* em X, Y ou Z.

Resultados e discussões

Finalizado a construção de todas as partes da cena, temos os resultado final da Figura 11.

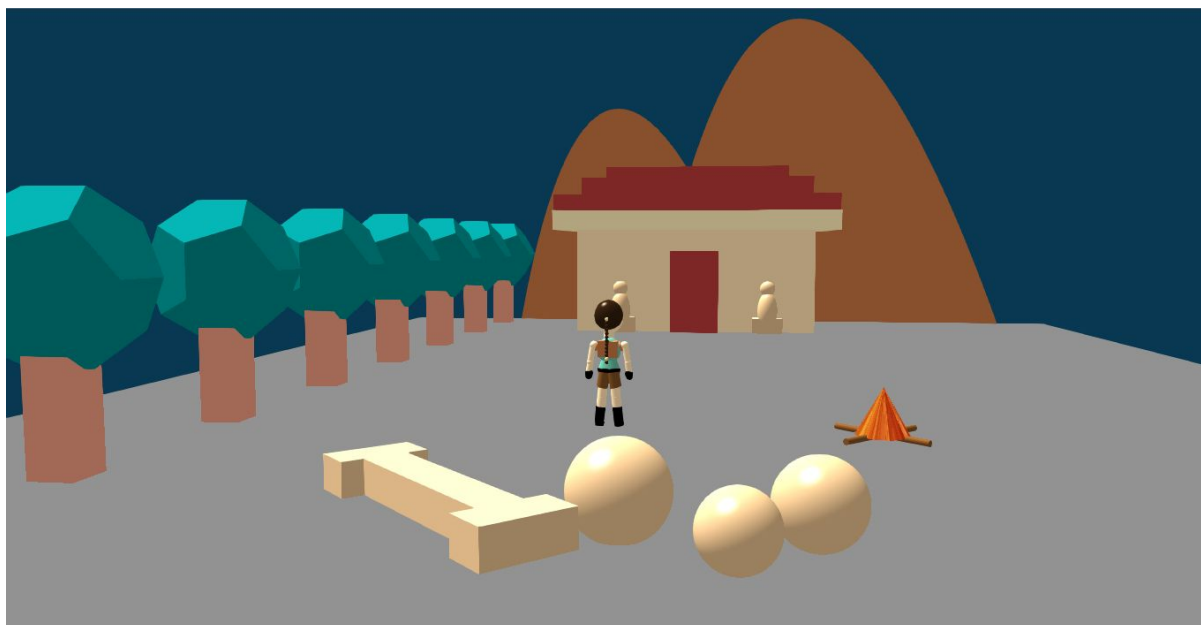


Figura 11: Cena final construída.

- **Objetos no clique do Mouse**

Quando o usuário clica com o mouse sobre a tela, uma lança aparece aleatoriamente em uma posição e a personagem vai ao encontro do objeto, pegando-o e lançando-o para fora da cena. Além disso, se houver mais cliques e a personagem ainda não tiver terminado sua tarefa, as lanças são desenhadas, mas aguardam na fila, até que a personagem a pegue. O objeto pode ser visualizado na Figura 12, em que é possível observar a personagem com a lança na mão, jogando-a para fora da cena, enquanto as outras lanças aguardam. Na Figura 13 é possível ver a lança sendo jogada para fora da cena.

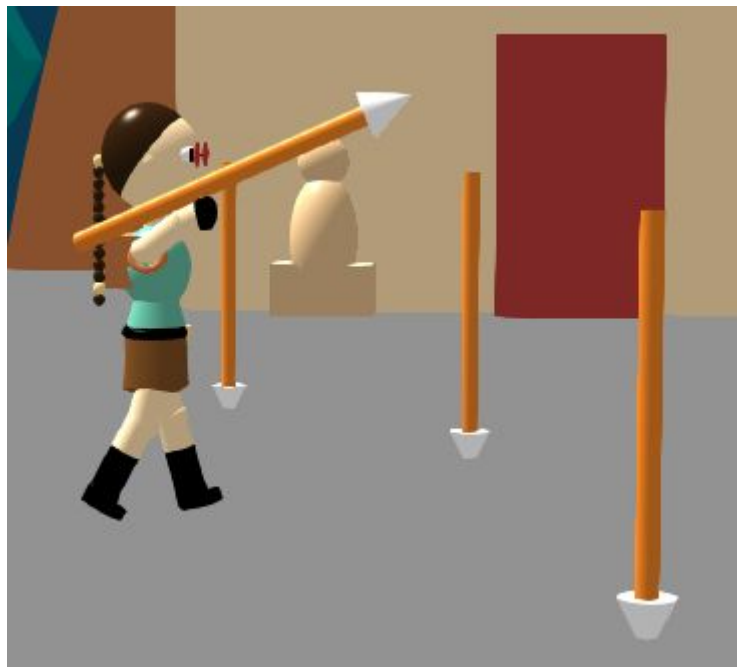


Figura 12: Lara pegando a lança, enquanto outras lanças aguardam na fila.



Figura 13: Lança sendo jogada para fora da cena.

- **Animação Automática**

Também, há uma animação automática que pode ser disparada pelas teclas 'A' e 'a' (ou pela opção do menu), que disparam e terminam a animação, respectivamente. Nesta animação a personagem faz um caminho até a entrada do templo, utilizando uma lupa e um tablet, como se estivesse investigando algo na cena. Nesta animação a personagem vai até a entrada do templo e volta para a posição infinitamente até que o usuário opte por terminar a animação pela tecla 'a' ou pela opção do menu. Uma visualização desta animação pode ser observada na Figura 14.



Figura 14: Lara analisando o cenário com sua lupa e tablet.

- **Menu de Opções**

Ainda, há um menu, que possui 17 opções como mostra a Figura 15, que podem ser acessados pelo clique do mouse com o botão direito. Em seguida, clicar com o botão esquerdo na opção desejada. Caso seja uma opção de rotação, as setas esquerdas e direita, controlam a direção da rotação.

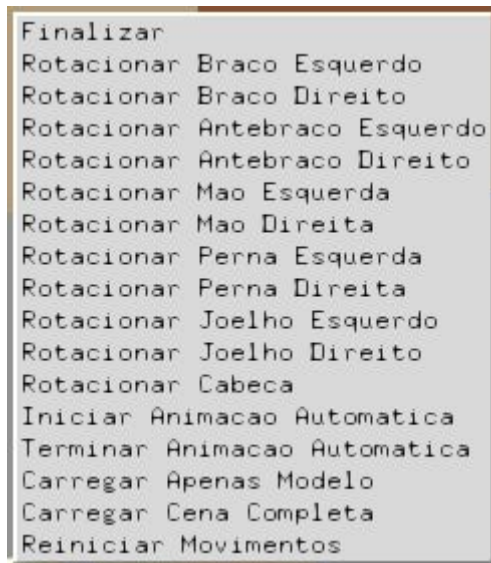


Figura 15: Menu de opções.

A 1ª opção “Finalizar” serve para finalizar a execução do programa. Já as opções 2-12 servem para rotacionar as partes do corpo da personagem. A opção 13 e 14 servem para disparar a animação automática. 15 e 16 são opções para carregar apenas a personagem ou toda a cena (para facilitar na visualização apenas da personagem, por exemplo). A opção 17 “Reiniciar Movimentos” é responsável por setar todas as variáveis de rotação/movimento da personagem para o padrão original, caso desejado reiniciar a rotação das partes do corpo.

Por exemplo, na Figura 16, a personagem teve seu braço esquerdo e cabeça rotacionadas.

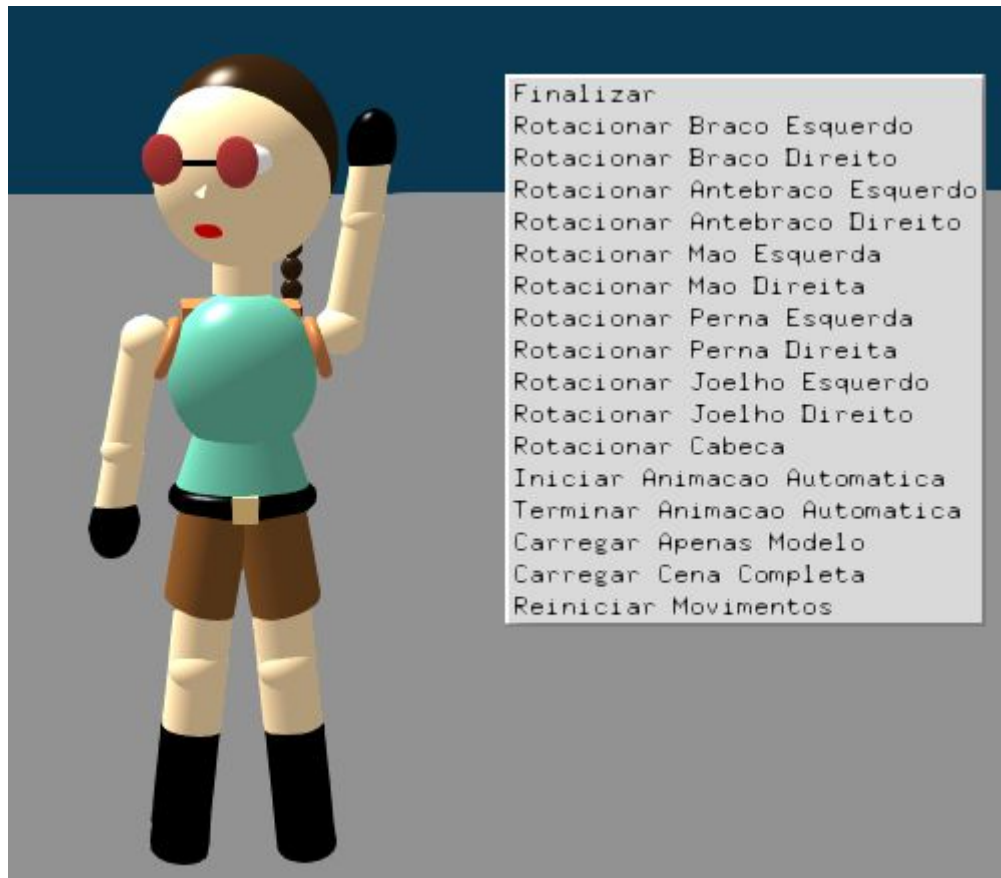


Figura 16: Menu de opções e Lara com cabeça e seu braço esquerdo rotacionados.

• Funções do Teclado

No teclado também há algumas opções, como as de rotação da cena em X, Y e Z, controlado pelas teclas 'x', 'y', 'z', 'X', 'Y' e 'Z'.

Com as teclas 'R' e 'r' é possível rotacionar a personagem.

Para finalizar o programa, pode-se pressionar as teclas 'q' ou 'ESC'.

Para zoom, há as teclas 's' e 'S', que diminuem e aumentam o zoom, respectivamente.

As teclas 'g' e 'G' são responsáveis por aumentar e diminuir o ângulo da abertura da câmera, respectivamente.

As teclas 'a' e 'A', terminam e disparam a animação automática da personagem.

Por último, as teclas 'p' e 'P' aumentam e diminuem, respectivamente o *panning* da câmara em X. E o análogo para as teclas 'o' e 'O' em Y e as teclas 'l' e 'L' para em Z.

• Informações Relevantes

- Devido às dificuldades encontradas em pegar o ângulo entre dois vetores que estão opostos (Personagem de costas para onde deveria ir, ao clique do mouse) e depois de várias tentativas, foi optado para que a personagem ande em linha retas (primeiramente em Z e depois em X) e ao final, retorne à posição inicial ($X = 0$ e $Z = 0$), como na distância de Manhattan.
- Com o uso da projeção perspectiva, pegar a posição do clique do mouse, é um pouco mais complexo. Por isso, optou-se por apenas gerar objetos em posições aleatórias, quando há um clique do mouse.

- Modelar Superfícies Splines também é mais complexo que gerar Curvas Splines. Após tentar modelar uma montanha por Superfícies Splines, sem muito sucesso, optou-se por usar apenas Curvas Splines para modelar a montanha. Embora seja 2D, é interessante trabalhar com Splines e para contornar a situação de ser 2D, utilizou-se a técnica de *Sweep* translacional, fazendo com que essa curva pareça de fato, 3D.

- **Extra**

- Para gerar objetos um pouco mais complexos, foram utilizados *Sweeps* translacionais e B-Splines.
- Também foi feita uma opção para carregar apenas o modelo da personagem, para ter uma melhor visualização do modelo feito.
- Movimento de *panning* da câmera em X, Y e Z.
- Alterar o ângulo da abertura da câmera.
- Rotação da personagem em Y.
- Rotação do tronco da personagem.

Conclusões

Neste projeto foi desenvolvido um humanóide baseado na personagem do mundo dos games, Lara Croft. O objetivo era modelar um humanóide que interagisse numa cena, com determinadas tarefas ao clique do mouse ou apertando alguma tecla, além das simulações das partes individuais do corpo.

Com o desenvolvimento deste trabalho, foi possível unir boa parte do conhecimento adquirido durante o semestre da disciplina de Computação Gráfica, seja ele teórico ou prático. Assim, confirmando a importância de sua realização, no aprendizado da disciplina.

Além disso, com a efetuação deste trabalho pode-se ter novas noções de trabalhos gráficos, seja em questões de projeto que envolve várias simulações, testes, técnicas e também noções básicas de como construir um personagem/animação (relativamente simples), com programação gráfica.