

Relatório do projeto de Computação Gráfica

Antônio Adelino

Armstrong Lohãns

Carlos Antônio

2019

Introdução

Como projeto de disciplina da cadeira de Computação Gráfica do período de 2019.2 da Universidade Federal Rural de Pernambuco - Unidade Acadêmica de Garanhuns, ministrada pelo professor Doutor Ícaro Lins Leitão da Cunha, foi solicitado a nós alunos a implementação de um projeto que envolvesse teorias expostas na disciplina, projeto o qual deveria ser codificado usando a biblioteca OpenGL na linguagem C ou Python.

Partindo dessas especificações iniciais codificamos a representação 3D de um boneco de neve, o qual tem em sua implementação conceitos vistos em aula como a criação da estrutura de armazenamento dos elementos visuais, aplicação de luz e sombra, além de suportar também a interação com o usuário.

Com isso, o presente trabalho busca relatar e ilustrar o que foi implementado por nós, tendo como principal finalidade descrever o processo de codificação e mostrar detalhadamente partes do código.

1 Código

1.1 Estrutura de armazenamento

Dado que escolhemos modelar a imagem de um boneco de neve, escrevemos a estrutura mostrada no Listing 1 para a criação da estrutura inicial de dados responsável pela construção da estrutura do boneco de neve.

Listing 1 – Estrutura de armazenamento

```
# Metodo para definir a estrutura do boneco
def desenharBonceDeNeve():
    glColor3f(1.0, 1.0, 1.0)

    # Desenha o corpo
    glTranslatef(0, 0.75, 0)
    glutSolidSphere(0.78, 20, 20)

    # Desenha a cabeca
    glTranslatef(0, 1, 0)
    glutSolidSphere(0.25, 20, 20)

    # Desenha os olhos
```

```

glPushMatrix()
glColor3f(0, 0, 0)
glTranslatef(0.05, 0.10, 0.18)
glutSolidSphere(0.05, 10, 10)
glTranslatef(-0.1, 0, 0)
glutSolidSphere(0.05, 10, 10)
glPopMatrix()

# Desenha o nariz
glColor3f(1, 0.5, 0.5)
glutSolidCone(0.08, 0.5, 10, 2)

```

De acordo com o trecho de código descrito no Listing 1, é criado inicialmente o corpo do boneco com um objeto esférico de raio igual a 0.78, em seguida é modelada a cabeça do objeto com uma forma também esférica e de raio igual a 0.25.

Após essa modelagem estrutural básica, é construído no boneco mais duas esferas de diâmetro igual a 0.025, ambas referentes aos olhos do boneco. E por fim, é criado um sólido em forma de cone com diâmetro igual a 0.08 para que esse seja o nariz do boneco de neve.

Assim, a estrutura básica de armazenamento do boneco de neve é criada e o objeto é inicialmente modelado. Veja a Figura 1.

1.2 Luz e sombra

Partindo da imagem modelada do boneco de neve, devemos nesse subtópico vamos detalhar a aplicação de sombra e luz, efeitos usados a partir dos métodos mostrados na Listing 2.

Listing 2 – Criação e posicionamento da luz

```

# Criacao do sombreamento (suave)
glShadeModel(GL_SMOOTH)

# Habilita o foco de luz
glEnable(GL_LIGHTING)

# Posicionamento do foco da Luz
lightZeroPosition = [1.0, 4.0, 15.0, 0.]

# Cor da luz
lightZeroColor = [0.8, 1.0, 0.8, 1.0]

```

De acordo com o código descrito, temos a aplicação de uma sombra suave na imagem, depois disso criamos uma fonte de luz, definimos a sua cor e a posicionamos no cenário de forma a favorecer a forma do modelo escolhido. Todo esse processo resulta na imagem ilustrada na Figura 2.

1.3 Interação

Buscando gerar uma interação com o usuário a função mostrada no código do Listing 3 é implementada. Esta função tem por finalidade configurar as setas do teclado para que cada uma delas seja responsável por uma ação.

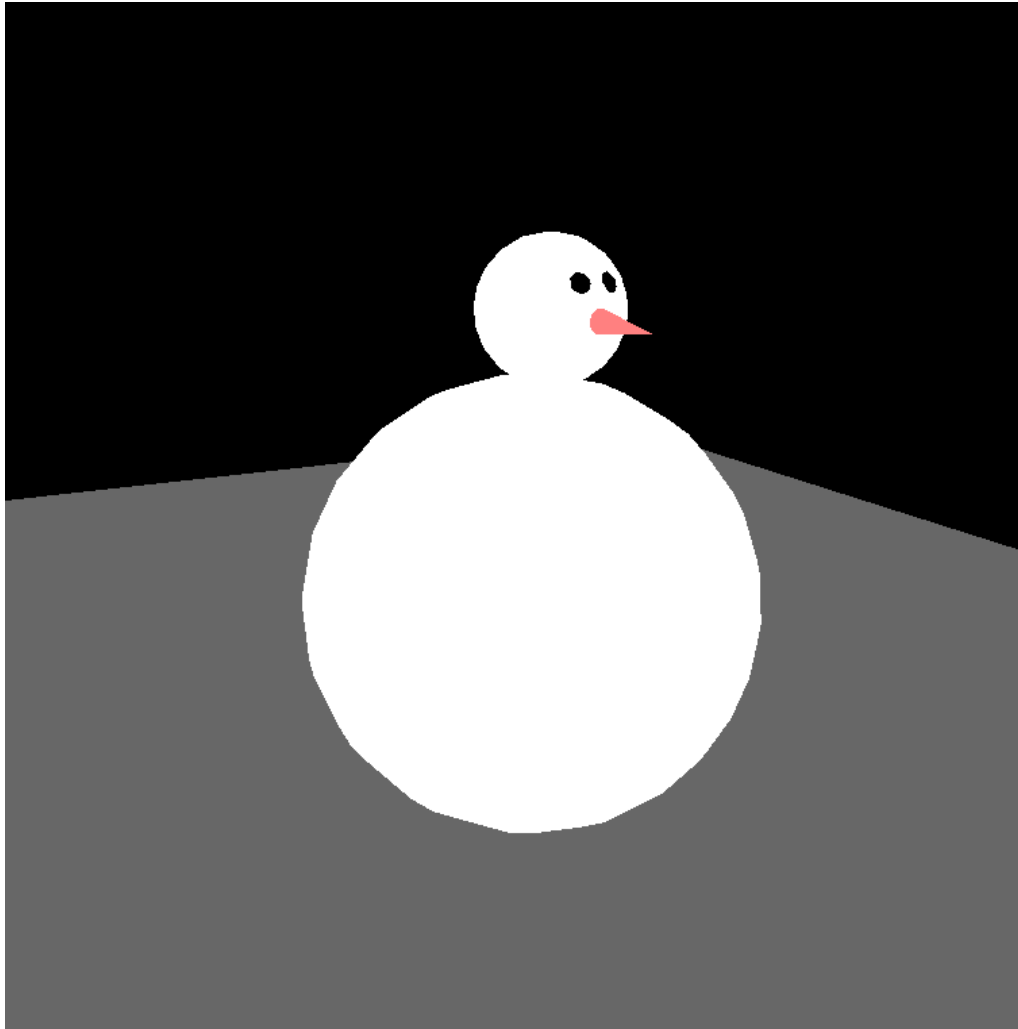


Figura 1 – Modelagem inicial do boneco de neve

Listing 3 – Interação com o usuário

Metodo que permite utilizar as setas do teclado para rotacionar o boneco
def controleSetas(key, x, y):

```
# Para o controle de rotacao vertical
if(key == GLUT_KEY_UP):
    glRotatef(-1.0, 1.0, 0.0, 0.0)
elif(key == GLUT_KEY_DOWN):
    glRotatef(1.0, 1.0, 0.0, 0.0)
# Para o controle de rotacao horizontal
elif(key == GLUT_KEY_LEFT):
    glRotatef(-1.0, 0.0, 1.0, 0.0)
elif(key == GLUT_KEY_RIGHT):
    glRotatef(1.0, 0.0, 1.0, 0.0)

glutPostRedisplay()
```

Definimos então os botões de setas no teclado para que cada um deles execute uma

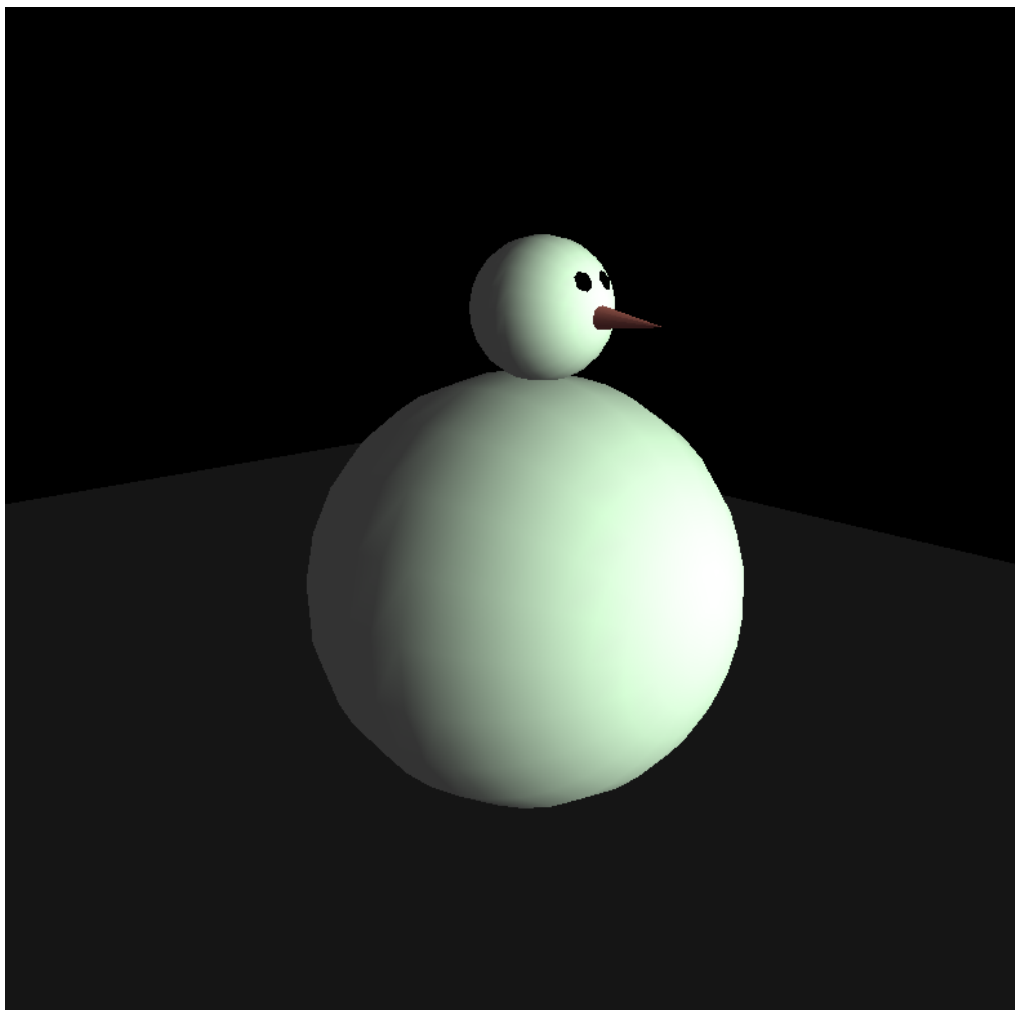


Figura 2 – Modelagem sombreada do boneco de neve

função correspondente ao movimento direcionado pelo seu descritor. Estes movimentos se dão por funções de rotação.

2 Conclusão

Tendo em vista os aspectos descritos e observados nos tópicos e subtópicos desse trabalho podemos concluir que a construção de uma representação em 3D exige do programador um grande conhecimento em técnicas visuais, uma vez que esta representação tem em si um aglomerado de conceitos que devem ser usados visando uma representação mais próxima possível do real.

Quanto ao algoritmo descrito ao longo deste relatório, podemos destacar o uso do efeito de luz e sombra como elemento que mais aproximou a representação do boneco de neve em 3D de uma imagem real, uma vez que a partir dessa aplicação foi notória a mudança visual dada ao objeto.

Além disso, a implementação da interação com o usuário tornou a aplicação mais dinâmica, o que nos provocou uma maior curiosidade quanto a construção do ambiente.

Dado que, a partir dessa interação, ao executarmos o *script* conseguimos ter a visualização do objeto em todos os pontos.

Por fim, vale destacar a fixação dos conceitos vistos em sala a partir da construção desse projeto, uma vez que ele nos trouxe a aplicação prática destes conceitos. Ao fim, podemos compreender que ao aplicar um único conceito ao objeto pode-se causar uma grande modificação na representação visual do mesmo.