

# Lista de Atividades

## Malhas e Iluminação

Universidade Estadual Vale do Acaraú

Professor: Gilzamir Ferreira Gomes

Esta atividade deve ser realizada individualmente. Questões possuem pontuações associadas do seguinte modo:

- Se o aluno conseguir realizar as três primeiras questões, conseguirá pontuar 1 ponto.
- As questões 4 e 5 valerão conjuntamente 1.5 pontos.
- A questão 6 valerá 2.5 pontos.

**Data de entrega do trabalho: dia 12 de maio de 2017.**

1) Abre o arquivo extractjson.blend no Blender. Veja que a tela ficará parecido com o que mostra a Figura 1. Nessa figura, temos um piso xadrez e um cubo distorcido.

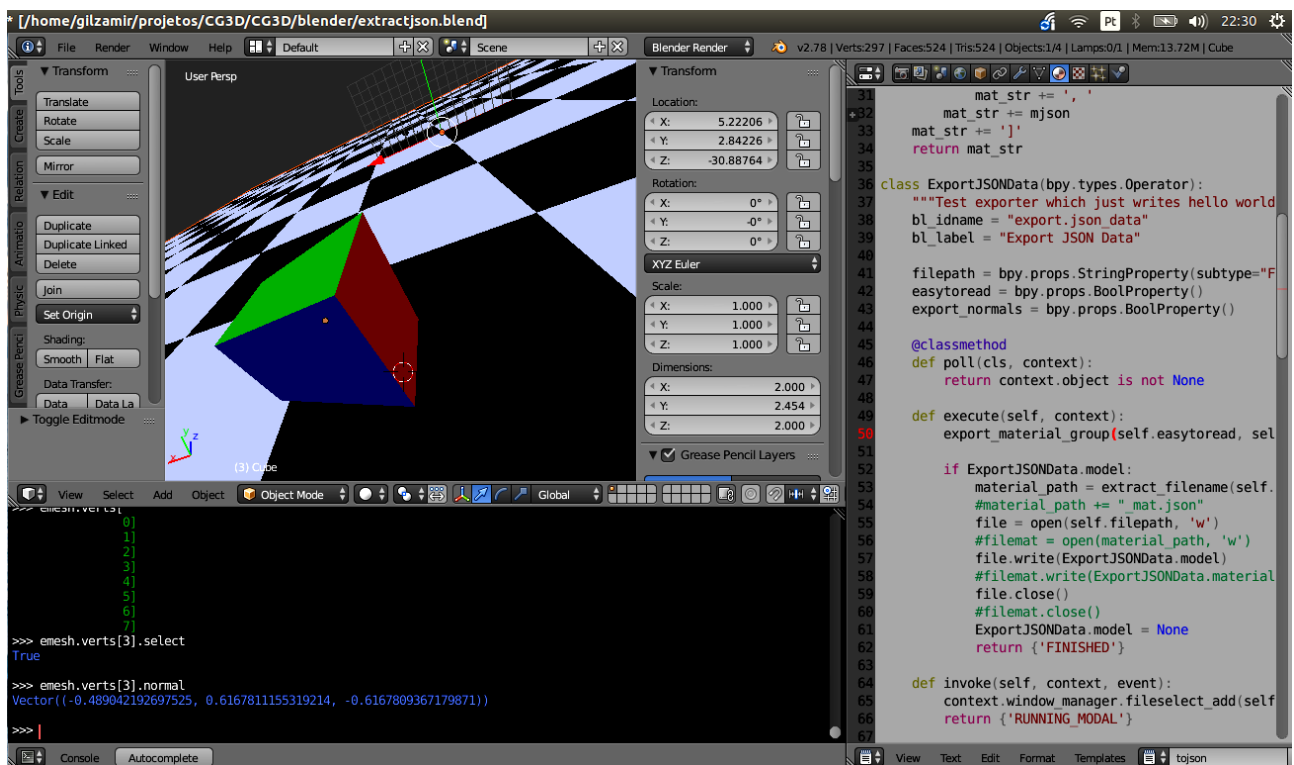


Figura 1: script exporttojson em ação.

A janela do Blender é subdividida em várias outras subjanelas embutidas. No geral, a interface é altamente personalizável. O usuário pode adaptá-la a muitas situações. Para este script, dividi a janela em três partes: no canto superior esquerdo, tomando a maior parte da janela, temos a 3D View, que mostra um cubo distorcido sobre um piso xadrez; no canto inferior esquerdo, na mesma largura da 3D View, temos o Python Console, um terminal onde podemos digitar comandos em Python para várias operações do Blender ou no modelo mostrado na 3D View; na lateral direita,

tomando toda a altura, temos um editor de texto, onde podemos escrever scripts em Python e armazená-los para reutilização futura.

A 3D View é onde podemos criar nossos modelos, geralmente tridimensionais. Aqui, visualizaremos uma cena por vez. O terminal para Python nos permite automatizar tarefas ou acrescentarmos ao Blender novas funcionalidades por meio de *plugins* e/ou extensões. O editor de texto nos permite criarmos arquivos de texto e possui suporte para programação, como coloração de sintaxe e execução de *script*. Para executarmos um scripts direto, por meio da interface do editor de textos, basta clicarmos no botão “Run Script”. Então, execute o *script* Python já carregado do arquivo exporttojson.blender quando este arquivo foi aberto com o Blender. Explore as opções de exportação e veja os arquivos exportados. Para que as opções de exportação servem? Que diferenças de resultados foram obtidas com cada opção de exportação em diferentes valores? (0.5 pontos)

2) A 3D View tem um modo de edição que nos permite alterar a malha do objeto selecionado. No modo de edição, podemos selecionar um vértice, uma aresta uma face, dependendo do modo de seleção utilizado. Para entrarmos no modo de edição, com o objeto selecionado e com o ponteiro do mouse na 3D View, pressione a tecla TAB do teclado alfanumérico. A Figura 2 mostra o resultado dessa ação quando realizada na tela mostrada na Figura 1.

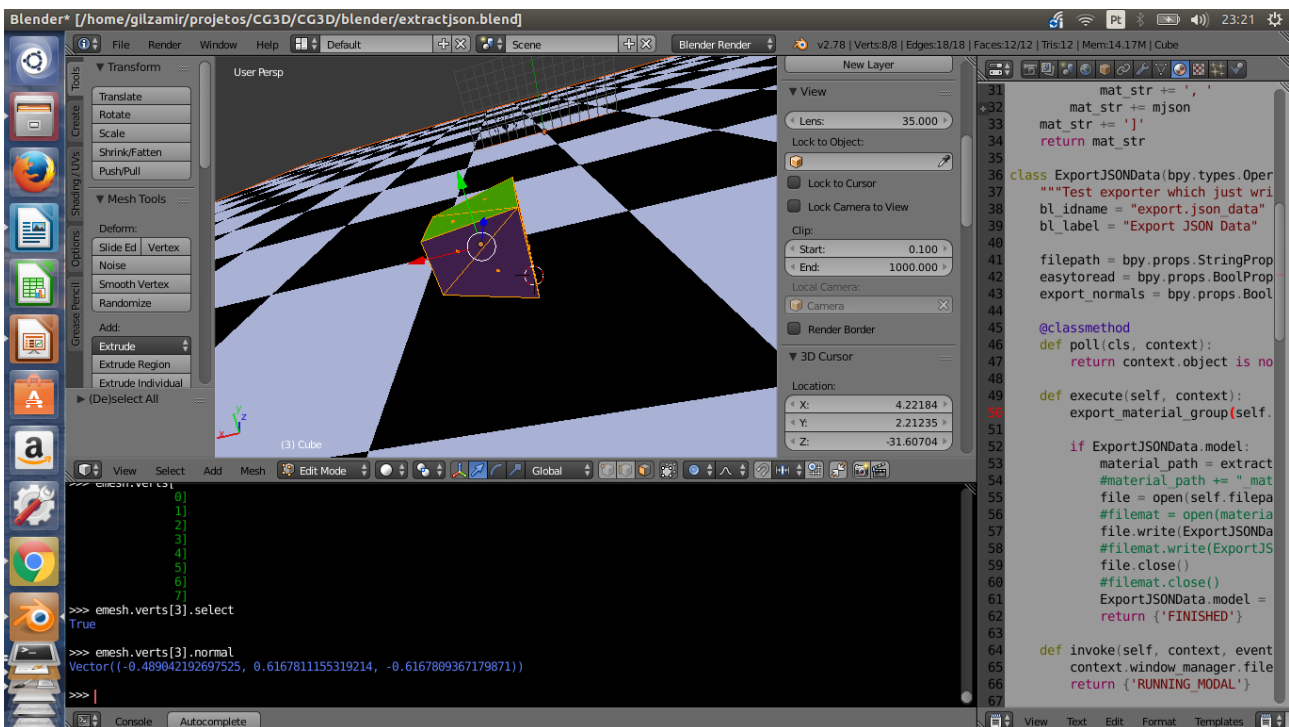


Figura 2: Entrando no modo de edição de malhas do Blender.

No modo edição, já podemos fazer a estrutura da malha do objeto e podemos selecionar os elementos da malha, alterá-los, manipulá-los de forma a obtermos algum resultado de interesse. No modo de edição, há diferentes modos de seleção. O modo de seleção pode ser escolhido por meio de 3 botões na barra de tarefas na 3D View, como mostrado na Figura 3.

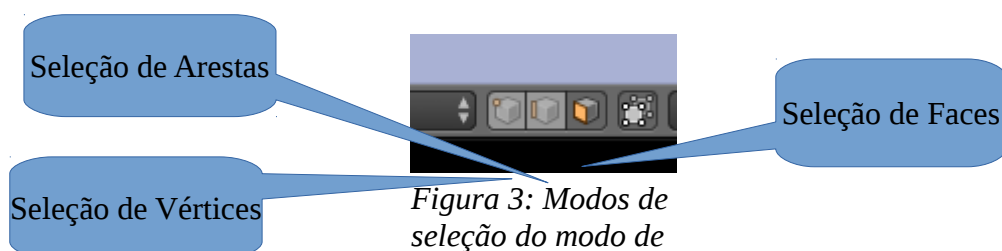


Figura 3: Modos de seleção do modo de edição do Blender.

O modo de seleção determina quais elementos primitivos (arestas, faces ou vértices) podemos selecionar e, portanto, manipular em relação à malha do objeto selecionado. Experimente os diferentes modos de seleção e experimente manipular de diversas formas. Por que é interessante podermos selecionar cada elemento primitivo individualmente?

3) Para obtermos informações programaticamente de um objeto selecionado na *3D View*, temos que:

**1º passo:** obter a malha do objeto

```
obj = bpy.context.object #obj é o objeto selecionado  
me = obj.data #me = mesh = malha do objeto selecionado
```

**2º passo:** obter os elementos selecionados

```
#acessar as informações do modo de edição  
import bmesh  
  
#Obter a malha do objeto selecionado  
edit_mesh = bmesh.from_edit_mesh(me)  
selected = [] #lista vazia  
  
if edit_mesh.select_mode == {'FACE'}: #se selecao de face  
    for f in edit_mesh.faces: #para cada face na malha  
        if f.select: #se a face esta selecionada  
            selected.append(f) #adiciona na lista  
  
elif edit_mesh.select_mode == {'VERT'}:  
    for v in edit_mesh.verts:  
        if v.select:  
            selected.append(v)  
  
elif edit_mesh.select_mode == {'EDGE'}:  
    for e in edit_mesh.edges:  
        if e.select:  
            selected.append(e)
```

**3º passo:** alterar ou obter a propriedade desejada do objeto ou elemento selecionado. Por exemplo, se quisermos obter a normal de todos os itens selecionados:

```
for item in selected:  
    print(item.normal)
```

Escreva os comandos dessa questão no terminal do Python no Blender e veja o que acontece quando, antes de executar os comandos, você:

- a) selecionou vértices
- b) selecionou as faces
- c) selecionou as arestas

Descreva o resultado obtido.

**4)** Execute os *scripts* da questão 3 no terminal de forma a obter a normal de um vértice por meio das normais da face. Para isso, faça:

i) selecionar as faces adjacentes a um determinado vértice e somar as normais das faces selecionadas. Normalize o resultado.

ii) verificar se a normal calculada pelo *Blender* para um vértice é igual ao resultado obtido em i. Se não, explique o motivo.

Dica 1: leia a documentação oficial com exemplos:  
<https://wiki.blender.org/index.php/Doc:PT/2.4/Manual/Modeling/Scripts>

Dica 2: veja o script para exportar do Blender para o formato json que implementei e disponibilizei no repositório <http://www.github.com/professorgilzamir/CG3D>.

**5)** Manipule o exemplo [http://localhost:8000/webgl\\_aula5/ex4/](http://localhost:8000/webgl_aula5/ex4/) para visualizar o objeto da forma mostrada na Figura 4. Explique o resultado obtido com base na reflexão especular.

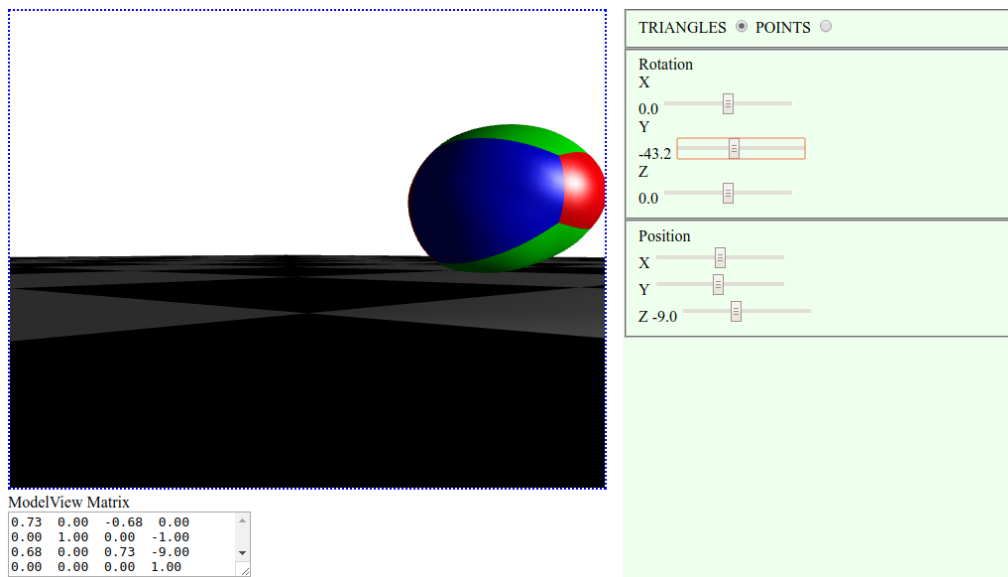


Figura 4: Manipule para obter este resultado.

6) Implemente o modelo de iluminação de Phong com interpolação de Phong. Faça isso a partir do exemplo [http://localhost:8000/webgl\\_aula5/ex4/](http://localhost:8000/webgl_aula5/ex4/).