

Computação Gráfica e Processamento de Imagens

CGPI: Animação

Unidade 3 - Seção 3

Esta webaula apresenta uma biblioteca de processamento gráfico e seus principais conceitos.

As animações gráficas são talvez os produtos mais atraentes da computação gráfica. Para produzir uma animação, todos os conceitos de síntese de imagens e modelagem tridimensional são necessários.

Além da criação do modelo 3D, é preciso associar a esse modelo uma sequência de transformações ao longo do tempo e, para cada instante, executar o pipeline de visualização para sintetizar um quadro da animação, que exige uma grande quantidade de linhas de código. O processo de criação do modelo 3D exige também ferramentas computacionais, de captura de imagens com múltiplas câmeras ou de interação homem computador para o desenho manual de modelos por artistas.

Jogos digitais, vinhetas de TV ou filmes de animação na qualidade dos de hoje só podem ser criados com o auxílio de aplicativos de criação e bibliotecas. O profissional de computação deve compreender os fundamentos da computação gráfica e também deve ter capacidade de aprender a usar as bibliotecas do mercado.

Bibliotecas

Há centenas de bibliotecas de computação gráfica tridimensional, gratuitas ou comerciais. Grande parte dessas bibliotecas são de alto nível, construídas utilizando outras bibliotecas, as bibliotecas de baixo nível.

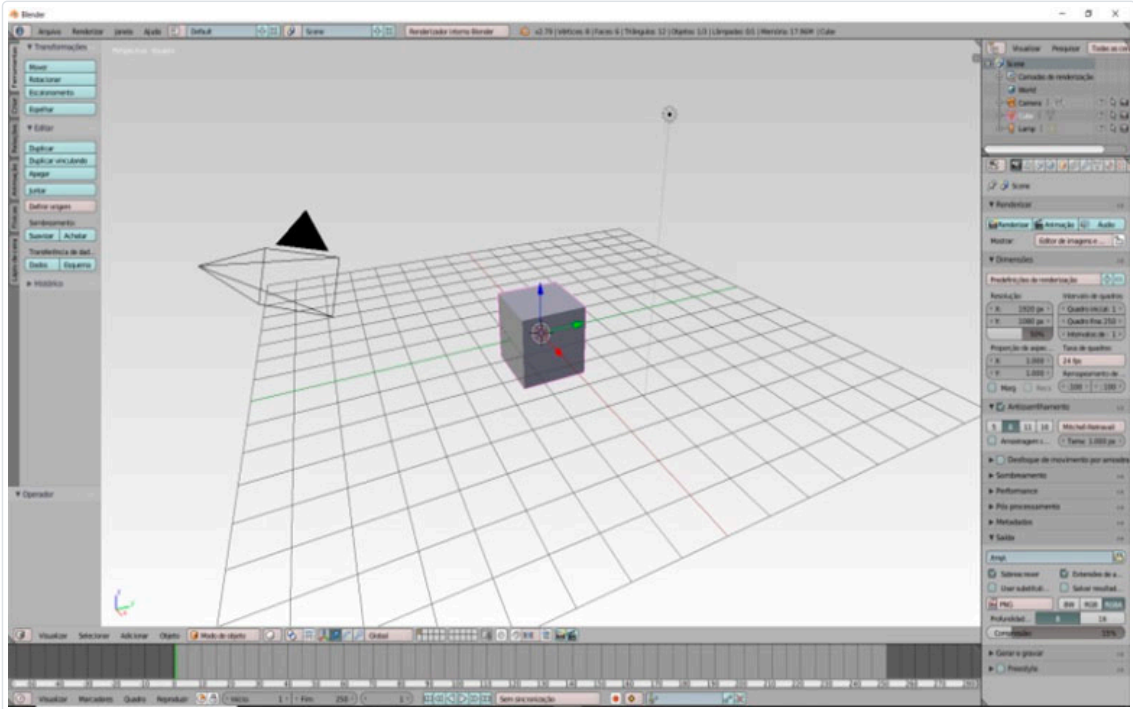
- [Bibliotecas de baixo nível](#)
- [Bibliotecas de alto nível](#)

As bibliotecas de baixo nível implementam os algoritmos mais fundamentais, como as transformações geométricas, geração de malhas, modelos de iluminação e o pipeline de visualização. Dentre elas podemos citar a OpenGL (ANGEL E SHREINER, 2012) e a Direct3D (STENNING, 2014). São bibliotecas utilizadas pelo mercado e bastante populares. Ambas (e suas concorrentes) implementam algoritmos otimizados na unidade de processamento gráfico (Graphics Processing Unit – GPU).

Processamento de modelos 3D

Para realizar o processamento de modelos 3D em Python com o **Blender** é essencial que o desenvolvedor tenha acesso ao manual da *Application Programming Interface* (API) Python do Blender (BLENDER, 2016).

Ao inicializar o Blender, o aplicativo cria um modelo inicial contendo um cubo, uma câmera e uma fonte de luz. A interface tem menus verticais à esquerda e à direita da tela, e duas áreas. A área maior em posição superior é a visualização 3D, e a área menor em posição inferior, é uma linha de tempo. O arquivo pode é salvo com extensão `.py`



Fonte: O Autor, adaptado do software Blender 2.79b.

No Blender:

- bpy é a API Python do Blender.

- O módulo `bpy.ops.mesh` é o módulo de operações sobre malhas.
- A função `bpy.ops.mesh.primitive_cube_add` possui diversos parâmetros e permite criar esferas com uma malha com diferentes números de polígonos.
- A função `bpy.ops.render.render(use_viewport=True)` é utilizada para obter uma renderização da cena na câmera pelo Python com que realiza o pipeline de visualização, incluindo transformações geométricas, efeitos de iluminação, *ray tracing*, para gerar um arquivo de imagem 2D no diretório temporário.

Para saber mais verifique a documentação.

Aplicação de transformações geométricas sequenciais

Como já estudado, a aplicação sequencial de transformações geométricas gera acúmulo de erros. Por este motivo a API do Blender sempre realiza transformações geométricas sobre os objetos a partir da origem do sistema de coordenadas do mundo.

Um objeto (malha, câmera ou fonte de luz) é representado pelo tipo `bpy.types.Object`, que possui atributos como `bpy.types.Object.location`, que define a translação do objeto a partir da origem, `bpy.types.Object.rotation_euler` que define a rotação do objeto em torno da origem, antes de se realizar a translação.

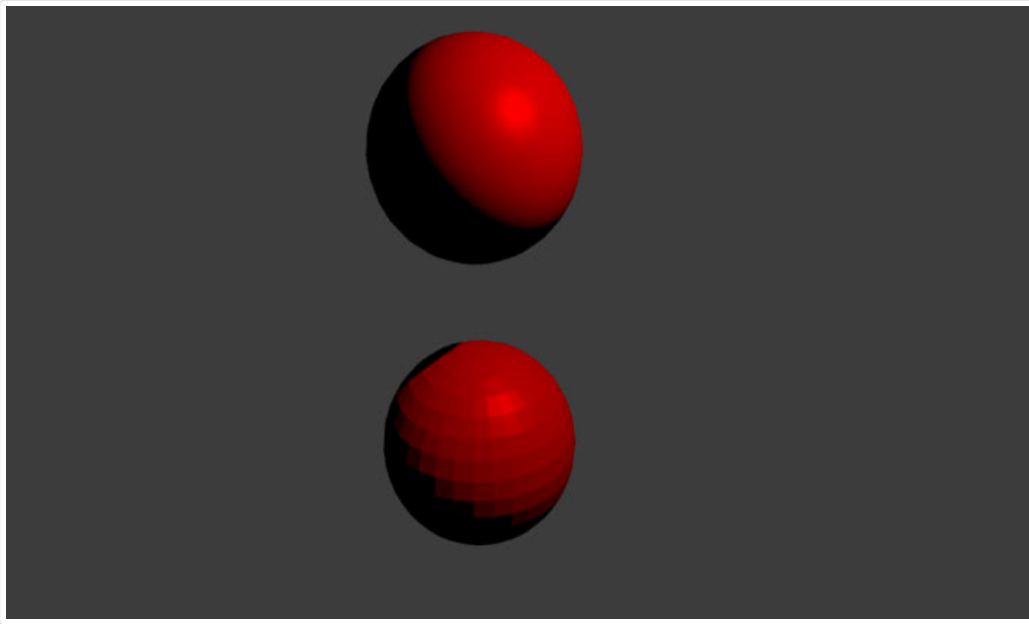
Para transformações geométricas mais complexas é possível alterar o sistema de coordenadas do mundo para o objeto, por meio do atributo `bpy.types.Object.matrix_world`, que é uma matriz 4×4 de transformação afim.

Para se realizar uma sequência de transformações geométricas, com a finalidade de representar um movimento ou deformação do objeto, é preciso calcular previamente todas as localizações e rotações sequenciais aplicadas ao objeto. Basta criar uma lista de localizações ou rotações e alterar os atributos do objeto em um laço de repetição.

Aplicação de efeitos de iluminação sequenciais

O modelo de tonalização pode ser alterado para utilizar superfícies curvas, com o uso da função `bpy.ops.object.shade_smooth`. Esta função não altera a malha do objeto, apenas o modelo de tonalização.

Renderização com tonalização constante e interpolada



Fonte: O Autor, adaptado do software Blender 2.79b.

Também é possível atribuir a um objeto um material, definindo para o material as características de cor e reflexão com base no modelo de iluminação de Phong (1975). O material possui as características de cor e intensidade das componentes de reflexão especular e difusa. A cor e a intensidade da componente de reflexão ambiente são atributos da cena.

Criação de animação gráfica

A combinação da aplicação de transformações geométricas e efeitos de iluminação sequenciais permitem a criação de uma [Link Interativo](#).

Para criar uma animação, basta definir um número total de quadros e associar a cada objeto em movimento um conjunto de quadros de referência. O movimento entre dois quadros de referência é construído pela própria API. Os mesmos movimentos das malhas 3D podem ser aplicados à fonte de luz e à câmera.

Animações podem ser executadas por scripts Python, utilizando o pacote Blender Game Engine (*bge*).

Todas as APIs de processamento gráfico de alto nível oferecem ferramentas similares à API Python do Blender.

Esta webaula apresentou os conceitos de biblioteca de processamento gráfico. Explore a documentação do Blender e também de qualquer outra API existente no mercado.

Para visualizar o vídeo, acesse seu material digital.