



Computação Gráfica

André Perrotta (avperrotta@dei.uc.pt)

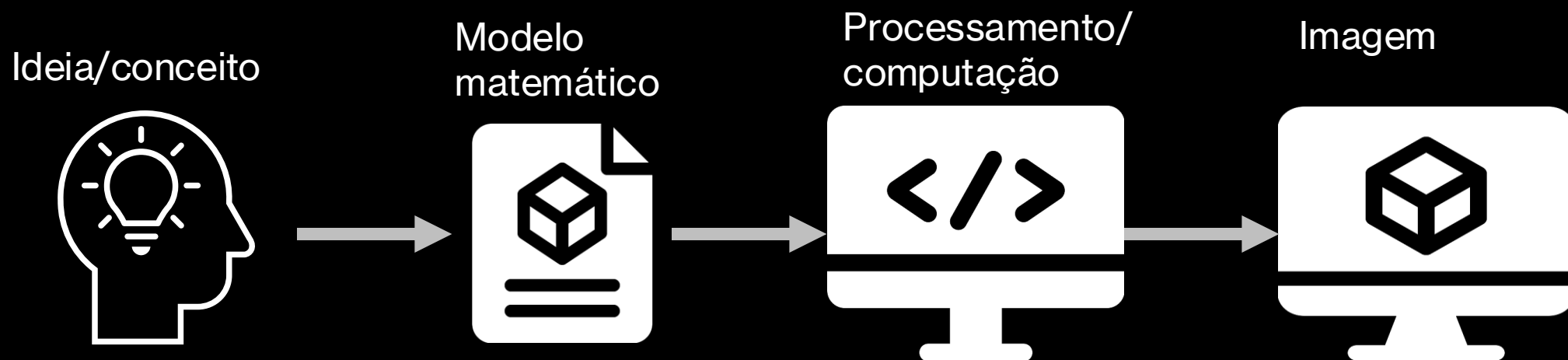
Hugo Amaro (hamaro@dei.uc.pt)

Introdução à disciplina

Docentes

- André Perrotta (T, TP, PL2,3,4,5,6)
 - Gab: D3.5
 - Horário de atendimento:
 - Seg: 16hs – 18hs, Qui: 14hs-18hs
 - (combinar por email)
 - avperrotta@dei.uc.pt || avperrotta@gmail.com
- Hugo Amaro (PL1)
 - hamaro@ipn.pt || hamaro@dei.uc.pt
 - GABINETE - D2.24
 - Seg. 10h30-13h00 e Qua. 15h-17h30

Computação Gráfica

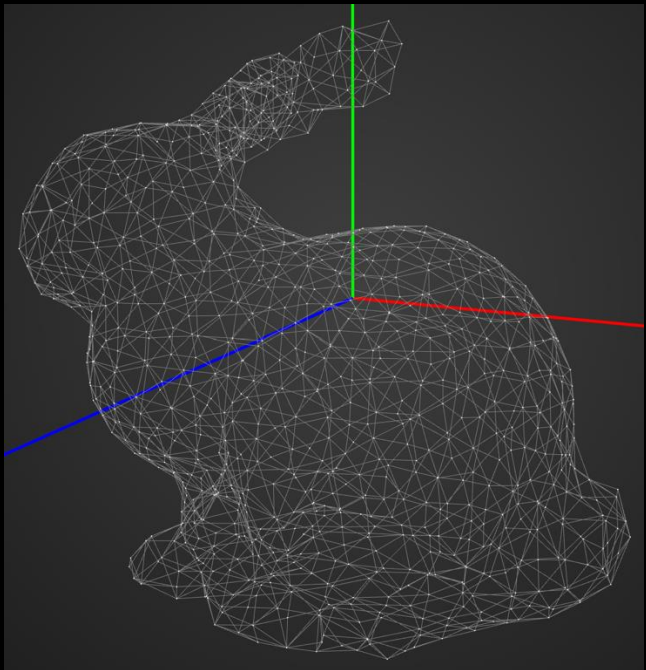


- Área da computação que estuda a criação/geração, processamento e exibição de imagens e sequências de imagens (vídeo).

Objetivo principal

- Entender a conceptualização e etapas de implementação do processo de **renderização**, desde a definição de um objeto 3D, através de sua descrição por polígonos, materiais e textura, passando pelas etapas de construção da cena (câmera, visualização, iluminação), até a etapa final de rasterização e exibição em formato 2D.
- Responder à pergunta:
 - Como é gerada uma imagem na tela do computador?

Renderização



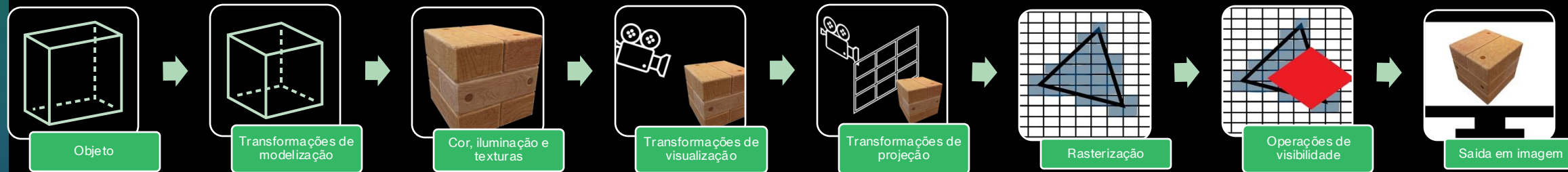
Objeto/modelo 3D
Vértices, arestas, faces, texturas, materiais
Cena
Câmera, luz, recorte

RENDERIZAÇÃO



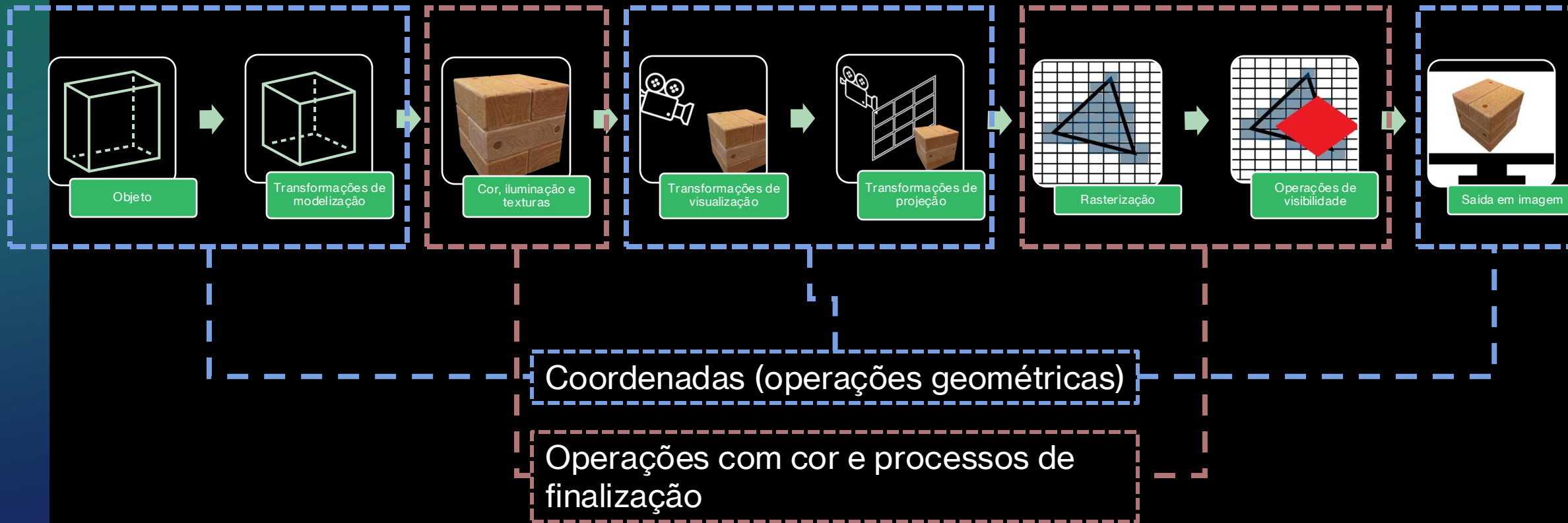
Tela 2D

Render pipeline

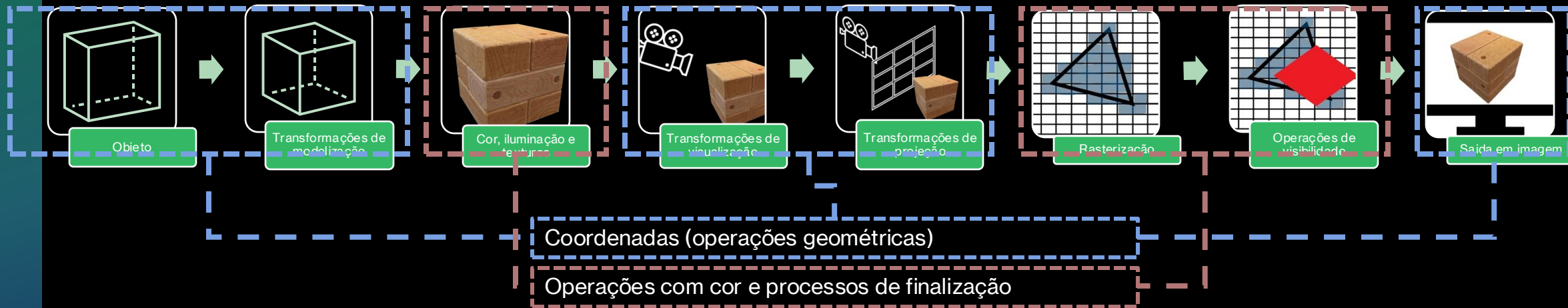


Pipeline de renderização **POLIGONAL**

Render pipeline

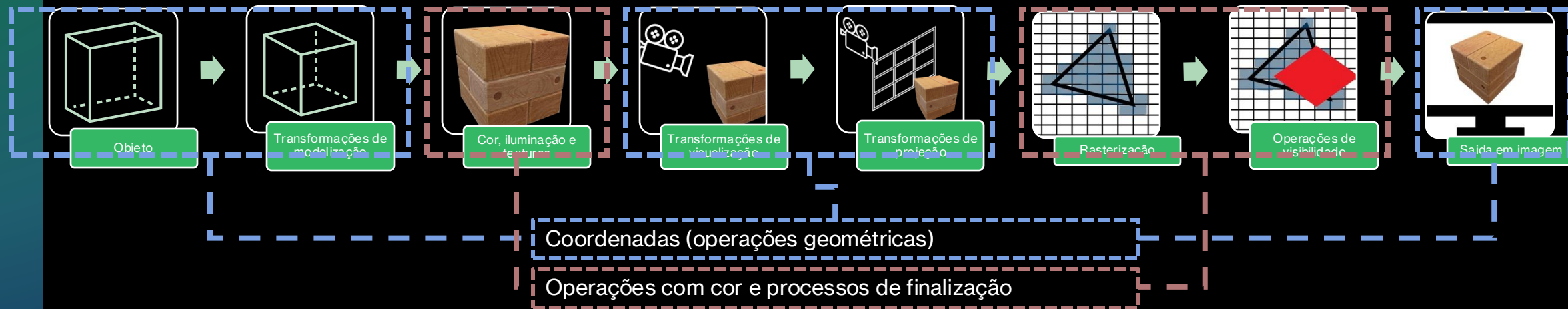


Render pipeline



Estas operações todas seriam muito complicadas de implementar “from scratch”

Render pipeline



Por isso nós (e praticamente todos os developers de CG do mundo) vamos utilizar o **OpenGL = Open Graphics Library**

Nosso foco

- Criação de objetos de forma procedural, através da definição de vértices, arestas, planos, normais e das transformações geométricas necessárias.
- Animação/movimentação de objetos de forma procedural, por modelos de simulação física (simplista) ou *ad hoc*.
- Determinação da estética dos objetos através da definição de cores, materiais e utilização de texturas estáticas (imagens pré-produzidas) e dinâmicas, mapas de elevação e “rugosidade” e interação com as componentes de luz pela definição de materiais.
- Determinação composição final da cena através do posicionamento da câmera, luzes e recorte da tela.

Conhecimentos fundamentais

- Vetores e fundamentos de álgebra linear (operações com matrizes)
 - Para construir objetos com polígonos
 - Para definir as normais de um polígono
 - Para definir o posicionamento de câmera e sua projeção na tela de visualização
 - Para calcular a interação dos vértices com as fontes de luz na cena
 - Para definir o mapeamento de coordenadas de textura em coordenadas de objetos
- Modelos de cor e iluminação
 - Para entender como nós, seres humanos, percebemos a cor dos objetos
 - Para definir a cor dos objetos da cena
 - Para definir materiais para os objetos

O que não vamos ver (não há tempo para tudo!)

- Pipeline de renderização moderno, com recurso a GLSL (OpenGL shader language)
 - Permite executar computação em vértices e fragmentos diretamente na placa gráfica.
 - Super poderoso, mas também super complexo de entender e aprender.
 - Pressupõe que o desenvolvedor domine completamente as operações do pipeline poligonal.

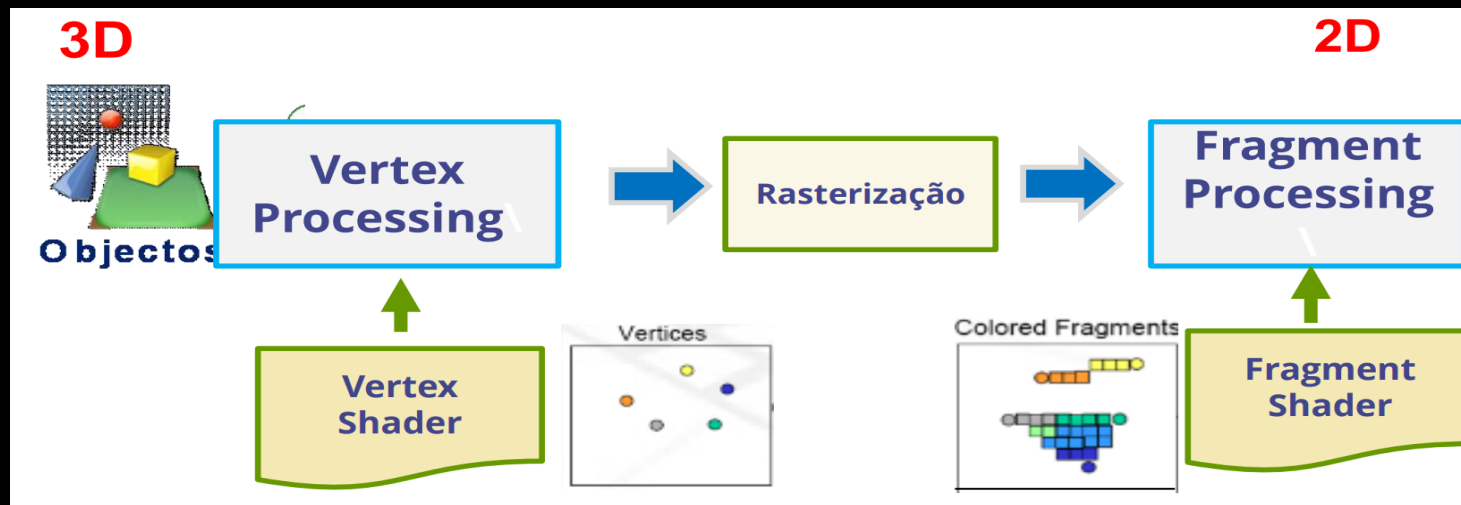


Figura “emprestada” de aulas
do prof. Jorge Henriques

O que não vamos ver (não há tempo para tudo!)

- Como modelar objetos 3D utilizando softwares de alto nível (Blender, CAD, etc.)
 - Não há aqui desafio numa perspectiva de desenvolvimento.
 - Quem se interessar pode aprender o básico por tutoriais online e funções e modelos mais complexos com cursos específicos.
 - Por exemplo: modelar avatares humanos em nível profissional requer um amplo conhecimento de anatomia e representação anatômica técnica e artística.
 - O que queremos é que vocês entendam o que está por trás dos botões do Blender e consigam (no futuro) conceptualizar e implementar novos botões e quem sabe até novos paradigmas de modelação 3D.

Metodologia

- Aulas teóricas (2horas)
 - Fundamentos matemáticos e conceitos teóricos de CG.
- Aulas TP (1 hora)
 - Fundamentos computacionais e implementação de algoritmos (programação C++).
 - Exercícios “em papel”
 - Definição dos objetivos das PL
- Aula PL (2 horas)
 - Exercícios computacionais.
 - Acompanhamento e suporte de projeto.
 - Defesa de projeto.

Teóricas

- Introdução
- Geometria/transformações 2D
- Geometria/transformações 3D
- Hierarquia de transformações (OpenGL matrix stack)
- Projeções e visualização
- Lógica de videojogos (máquina de estados, colisões, movimento/animação)
- Cor, iluminação e materiais
- Texturas estáticas (imgs) e dinâmicas (vídeos)
- Partículas
- Visualização estereoscópica (VR)

Avaliação

- Exame teórico
 - 10 valores
 - Sem consulta
 - +- 2hs
- Mini teste (frequências)
 - 4 Valores
 - Sem consulta
 - 1 hora
 - 28/10
 - questões sobre construção e visualização de objetos 3D
- Projeto
 - Individual
 - 6 valores
 - 2 metas
 - Meta 1 (24/11): geometria/animação/gameplay
 - Meta 2 (05/01/2024): estética/interação (luz, materiais, texturas)

Regras

- Mínimos (para ir à recurso)
 - 30% exame
- Melhorias em época normal
 - Não é possível realizar(melhorar) os “mini-testes” na época de exame normal
- Melhorias em época de recurso
 - Em época de recurso não é possível realizar a componente de Projeto.
 - Como estratégia de melhoria da nota da época normal, poderá realizar um teste para substituir o mini-teste realizado em época normal (caso obtenha uma melhor avaliação). Este teste acontecerá no mesmo dia do exame de recurso, com início logo após o exame.
 - A nota final em época de recurso será uma entre as 3 opções:
 - A - $\text{ExameRecurso} \cdot 0.5 + \text{MiniTesteNormal} \cdot 0.2 + \text{Projeto} \cdot 0.3$
 - B - $\text{ExameRecurso} \cdot 0.5 + \text{MiniTesteRecurso} \cdot 0.2 + \text{Projeto} \cdot 0.3$
 - C - $\text{ExameNormal} \cdot 0.5 + \text{MiniTesteRecurso} \cdot 0.2 + \text{Projeto} \cdot 0.3$
- Nota final = $\max(\text{época normal}, \text{época recurso})$

Funcionamento das aulas

- Não há marcação de presenças nas aulas T, TP ou PL
 - Usualmente, os alunos assíduos têm melhores notas finais na cadeira.
- Nas aulas T e TP, espera-se que os alunos se comportem de maneira adequada: concentrados e em silêncio.
 - Aulas com interrupções por motivos de barulho e conversas paralelas, serão interrompidas e a aula (e conteúdo da aula) será considerada “aula dada”.
- Nas aulas PL, espera-se que os alunos tragam seu computador de trabalho, com as ferramentas de desenvolvimento necessárias já instaladas e configuradas e que de fato tentem realizar os exercícios propostos.



Dúvidas?