

## Plano de Ensino

### 1) Identificação

**Disciplina:** INE5420 - Computação Gráfica  
**Turma(s):** 05208  
**Carga horária:** 72 horas-aula      Teóricas: 36      Práticas: 36  
**Período:** 2º semestre de 2016

### 2) Cursos

- Ciências da Computação (208)

### 3) Requisitos

- Ciências da Computação (208)
  - INE5408 - Estruturas de Dados
  - MTM3120 - Cálculo 2
  - MTM3121 - Álgebra Linear

### 4) Professores

- Aldo Von Wangenheim (aldo.vw@ufsc.br)

### 5) Ementa

Computação Gráfica Básica. Sistema Gráfico Interativo. Transformações geométricas 2D e coordenadas homogêneas. Clipping. Curvas paramétricas em 2D e 3D. Estruturas de dados gráficas 3D. Navegação 3D. Projeções, perspectiva e clipping 3D. Superfícies paramétricas bicúbicas. Visualização em 3D contendo, Rayshading, Raycasting e Raytracing. Conversão por varredura e buffer de profundidade. Iluminação de objetos. Implementação de um rayshader. APIs Gráficas e OpenGL. Animação e utilização de modelos hierárquicos. Simulação de movimentação de animais e humanos. Realidade virtual e VRML.

### 6) Objetivos

**Geral:** Desenvolver os aspectos teórico-práticos da disciplina, com desenvolvimento em laboratório de sistemas gráficos composta de 4 módulos: Módulo I - Computação Gráfica Básica com Implementação de um Sistema Gráfico Interativo. Módulo II: Visualização Realística em 3D. Módulo III - APIs Gráficas Utilizadas Comercialmente. Módulo IV: Aplicações Avançadas e Assuntos Especiais. Objetiva passar ao aluno tanto conhecimentos matemáticos e técnicos fundados sobre as técnicas e preceitos teóricos da Computação Gráfica, quanto prover uma experiência prática no desenvolvimento de sistemas gráficos envolvendo estes conceitos.

#### Específicos:

- Compreender e Implementar Visualização Realística em 3D contendo: introdução a Rayshading, Raycasting e Raytracing; conversão por varredura; o buffer de profundidade; modelagem da iluminação de objetos; principais raytracers; implementação de um rayshader.
- Utilizar na prática de APIs Gráficas contendo: introdução a OpenGL; sólidos em OpenGL; operações geométricas em OpenGL; modelos de arame e facetas em 3D; normais e efeitos de Iluminação com OpenGL; Java3D e outras APIs.
- Desenvolver a Aplicações Avançadas contendo: animação e utilização de modelos hierárquicos; simulação de movimentação de animais e humanos e simulação de sistemas de partículas.

### 7) Conteúdo Programático

- 7.1) Introdução à CG, aplicações e consequências [2 horas-aula]
- 7.2) Conceitos Básicos de Computação Gráfica [2 horas-aula]
- 7.3) Coordenadas 2D [8 horas-aula]
  - Princípios de Transformações 2D e Coordenadas Homogêneas
  - Implementação de Transformações 2D e Coordenadas Homogêneas
  - Sistema de Coordenadas da Window
  - Clipping 2D

- 7.4) Curvas [6 horas-aula]
  - Métodos analíticos: Blending Functions
  - Métodos iterativos: Forward Differences
- 7.5) Computação Gráfica 3D [8 horas-aula]
  - Princípios de Projeções
  - Projeções Paralelas
  - Transformações 3D
- 7.6) Perspectiva [6 horas-aula]
  - Projeção em Perspectiva
  - Clipping 3D
- 7.7) Superfícies Curvas [4 horas-aula]
  - Superfícies Curvas Bicúbicas em 3D
  - Métodos analíticos para Superfícies Curvas Bicúbicas em 3D
  - Métodos iterativos para Superfícies Curvas Bicúbicas em 3D
- 7.8) Teoria da Iluminação [8 horas-aula]
  - Raytracing, Raycasting, Rayshading
  - Buffer de Profundidade
  - Conversão por Varredura
  - Modelagem de Iluminação de Ambientes e Objetos
- 7.9) Implementação de Iluminação [4 horas-aula]
  - Pixel Shading: fundamentos matemáticos e algoritmos
  - Pixel Shading: implementação em CPU e GPU
- 7.10) Ferramentas de Visualização Realística [4 horas-aula]
  - Raytracing com ferramentas open-source como POV-Ray
  - Modeladores 3D para Raytracing como Moray
  - Linguagens de descrição de cenas
- 7.11) Modelos de Interação de Fontes de Luz [4 horas-aula]
  - Radiância: conceitos e modelos matemáticos
  - Radiância: ferramentas e aplicações
- 7.12) APIs Gráficas [6 horas-aula]
  - OpenGL
  - Java3D e outras APIs
- 7.13) Modelos Hierárquicos [4 horas-aula]
  - Princípios Básicos
  - Modelando movimentos articulados
- 7.14) Carga horária reservada para o processo de avaliação [6 horas-aula]
  - Defesas dos Trabalhos
  - Auxílio para trabalhos de recuperação
  - Recuperação

## 8) Metodologia

A disciplina terá um enfoque eminentemente prático. A metodologia de ensino será baseada no contraponto entre Aulas Teóricas e Aulas Práticas. Para tanto todo novo assunto será introduzido em uma aula teórica que terá a duração de 2 horas, acompanhada ou não pelo Estagiário de Docência designado para a disciplina. Este conteúdo teórico será fixado através de uma aula de caráter prático, com exercícios e resolução de dúvidas, com duração de 2 horas, sempre com auxílio de um Estagiário de Docência. Nesta aula serão realizadas em laboratório atividades de modelagem e implementação com o objetivo de fixar o conteúdo, além da discussão em grupo de problemas de compreensão e implementação encontrados pela turma.

A ferramenta de EAD Moodle disponível em [moodle.ufsc.br](http://moodle.ufsc.br) será utilizada para guiar e organizar o ensino, sendo o repositório oficial de material de aula. A disciplina no Moodle também detalhará o cronograma deste plano de ensino, servindo para documentar alterações de cronograma advindas de necessidades identificadas no semestre. A lista de email da disciplina será utilizada para intermediar a comunicação entre professor, estagiário de docência e alunos. Para efeitos da avaliação da participação do aluno na disciplina, as suas estatísticas de utilização da ferramenta de EAD poderão ser levadas em consideração.

### Pressupostos da metodologia

A metodologia adotada nesta disciplina do Curso de CCO, que é um curso de período integral, pressupõe que os alunos do curso não se limitem a comparecer às aulas, mas que utilizem para as atividades extra-classe associadas a esta disciplina (leituras, resolução de exercícios teóricos e exercícios com o uso da ferramenta de EAD empregada) um número de horas igual ou superior ao número de horas-aula em sala de aula. Pressupõe-se que os alunos tenham estudado o conteúdo utilizando a bibliografia indicada e tenham resolvido, como atividade

extra-classe, todos os exercícios propostos pelo professor.

Parte do conteúdo (limitado a 20% da carga horária total) poderá ser ministrado na **modalidade semi-presencial**, na forma de **vídeo-aulas**, a serem assistidas fora de sala de aula, com base no art. 81 da Lei n. 9.394, de 20 de dezembro de 1.996 (Lei de Diretrizes e Bases da Educação), e no disposto na Portaria No 4.059, de 10 de dezembro de 2004. Isto ocorrerá quando o professor estiver impedido de ministrar a aula de forma presencial ou quando, em função do bom aproveitamento dos alunos, o professor considerar que essa forma de procedimento didático for adequada. Aulas semi-presenciais serão anunciadas com antecedência de pelo menos 24 horas, de forma a evitar o deslocamento desnecessário dos alunos.

## 9) Avaliação

A avaliação regular dos alunos se dará através de um **conjunto de até 5 notas**:

**1 prova teórica individual**, a ser realizada como e-Exam através do sistema do Moodle de Provas. Peso: 30% da nota semestral.

**4 trabalhos práticos realizados em equipes de 2 alunos**, valendo 70% da nota semestral. As equipes serão formadas no início do semestre e sua composição não se alterará mais. A avaliação dos trabalhos práticos realizar-se-á através de defesa. A média da nota dos trabalhos práticos será composta pela média ponderada do 1º trabalho (40% da nota de trabalhos) e pelas duas melhores notas dentre os três trabalhos restantes (30% da nota de trabalhos cada um):

- **Sistema Gráfico Interativo Básico:** Implementação, usando a linguagem C++ e sem utilização de APIs gráficas, de um sistema gráfico interativo (SGI) que deverá possuir a capacidade da representação de modelos de arame em 3D e realizar todas as transformações (trabalho obrigatório). Peso: 40% da nota de trabalhos.
- **Pixel Shading:** Um Projeto de Implementação de um Pixel Shader com base no Sistema Gráfico Interativo anterior, estendendo-o, ou então, a critério do professor, com base em um ambiente como Shader Maker, nesse caso empregando-o mais tarde no trabalho de Modelos Hierárquicos. Peso: 30% da nota de trabalhos.
- **Renderização:** Realização da modelagem realística de uma cena utilizando:
  - a linguagem de descrição de cenas de um Raytracer de domínio público ou
  - a API de uma game engine previamente indicada pelo professor,

com base em fotos de uma locação disponibilizadas pelo professor. Peso: 30% da nota de trabalhos.

· **Animação:** Um Projeto de Implementação envolvendo a implementação de uma animação através da técnica de Modelos Hierárquicos de um animal animado seguindo princípios de movimentação de animais dados em aula e, caso se tenha utilizado um ambiente de desenvolvimento de shaders, exportando empregando a iluminação ali programada. Poderá ser utilizada:

- a API OpenGL, com implementação em C++ ou, alternativamente, a API WebGL, com implementação em HTML5 e JavaScript ou
- a API de uma game engine previamente indicada pelo professor,

Peso: 30% da nota de trabalhos.

Os trabalhos de implementação do Sistema Gráfico Interativo Básico deverão obrigatoriamente satisfazer os seguintes requisitos:

- terem sido entregues no VPL
- terem sido desenvolvidos de forma a executarem no VPL
- terem sido desenvolvidos utilizando-se as linguagens de programação "C" ou C++
- serem baseados em código devidamente documentado e escrito de acordo com boas práticas de programação.
- o trabalho do Sistema Gráfico Interativo Básico, para fins de uniformidade de código, caso o aluno deseje utilizar um framework para GUI, deverá ser desenvolvido utilizando-se exclusivamente GTK++ ou QT como framework para desenvolvimento de interfaces, sendo aceita a modelagem de interface em Glade.

Havendo bom aproveitamento dos alunos (parâmetro: média das notas dos trabalhos da turma > 7,0), a Prova poderá não ser realizada, caso em que a nota dos trabalhos, respeitados os seus pesos individuais, perfazerá a nota semestral.

Para avaliação do SGI será analisada a existência das 10 funcionalidades exigidas como requisito durante o semestre e cada uma delas valerá 1 ponto.

A avaliação dos trabalhos práticos ocorrerá através de defesa, na qual participará obrigatoriamente todaa equipe envolvida no desenvolvimento dos trabalhos.

Para a avaliação do trabalho de modelagem realística de uma cena utilizando ou (a) a linguagem de descrição de cenas de um Raytracer de domínio público ou (b) uma game engine, serão utilizados os seguintes critérios, valendo até dois pontos cada um, dependendo do grau com que houverem sido atingidos:

- Uso da técnica de Radiância/Radiosidade

- Emprego de mais de uma fonte de luz
- Modelagem realista de pelo menos um objeto composto utilizando geometria construtiva ou uma técnica de modelagem hierárquica
- Uso de textura sobre objetos
- Emprego de transparência em superfícies envidraçadas e reflexos/espelhos

Para avaliação dos demais trabalhos práticos, os critérios de avaliação são os abaixo:

- **Compreensão do Problema:** entendeu o que era para fazer ? (2 pontos)
- **Solução:** soube encontrar uma solução ? (2 pontos)
- **Conhecimento teórico:** compreendeu as implicações teóricas da solução escolhida ? (2 pontos)
- **Algoritmos:** possui compreensão dos algoritmos empregados e sabe descrevê-los ? (2 pontos)
- **Código:** compreende a implementação, sabendo detalhar aspectos de seu funcionamento ou de falhas que estão ocorrendo ? (2 pontos)

A defesa dos trabalhos práticos, é considerada Prova Oral nos termos do Regimento da UFSC e o não comparecimento injustificado implica em conceito nulo nestes trabalhos.

Para efeitos de defesa, os trabalhos práticos dos alunos deverão estar disponíveis no Moodle da disciplina. Todas as defesas serão realizadas na presença de uma Testemunha e as perguntas e suas respostas protocoladas por esta testemunha. Esta testemunha será preferencialmente o estagiário de docência mas poderá ser algum aluno de pós-graduação ou professor do INE.

Para efeitos de defesa, os trabalhos práticos dos alunos deverão constar no VPL correspondente ao trabalho. Deverão compilar e executar neste ambiente.

Dado que a disciplina apresenta pelo menos 50% da carga horária consistindo de aulas práticas, conforme deliberação do Colegiado do Curso de Ciências da Computação de 18 de março de 2008, ela não prevê a realização de avaliação no final do semestre (recuperação) de que trata o parágrafo 2º do artigo 70 da Resolução 17/CUn/97.

## 10) Cronograma

- Aula 1 Apresentação da disciplina, explicitação das estratégias e métodos de ensino e avaliação.
- Aula 2 Conceitos Básicos de Computação Gráfica
- Aula 3 Transformações 2D e Coordenadas Homogêneas
- Aula 4 Transformações 2D e Coordenadas Homogêneas
- Aula 5 Sistema de Coordenadas da Window
- Aula 6 Clipping 2D
- Aula 7 Curvas Paramétricas 2D
- Aula 8 Laboratório: Curvas Paramétricas 2D
- Aula 9 Curvas Paramétricas 2D: Forward Differences
- Aula 10 Laboratório: Curvas Paramétricas 2D: Forward Differences
- Aula 11 Computação Gráfica 3D – Projeções Paralelas
- Aula 12 Laboratório: Computação Gráfica 3D – Projeções Paralelas
- Aula 13 Computação Gráfica 3D – Transformações 3D
- Aula 14 Laboratório: Computação Gráfica 3D – Transformações 3D
- Aula 15 Computação Gráfica 3D – Projeção em Perspectiva e Clipping 3D
- Aula 16 Laboratório: Computação Gráfica 3D – Projeção em Perspectiva e Clipping 3D
- Aula 17 Superfícies Curvas Bicúbicas em 3D
- Aula 18 Superfícies Curvas Bicúbicas em 3D
- Aula 19 Laboratório: Superfícies Curvas Bicúbicas em 3D
- Aula 20 Raytracing, Raycasting, Rayshading
- Aula 21 Buffer de Profundidade e Conversão por Varredura
- Aula 22 Modelagem de Iluminação de Ambientes e Objetos
- Aula 23 Laboratório: Modelagem de Iluminação de Ambientes e Objetos
- Aula 24 Pixel Shading
- Aula 25 Laboratório: Pixel Shading
- Aula 26 Raytracing com POV-Ray e Moray
- Aula 27 Laboratório: Modeladores 3D para Raytracing como Moray
- Aula 28 Radiância
- Aula 29 Visão Humana
- Aula 30 OpenGL em C e Outras APIS como Java3D
- Aula 31 Reservado para Prova
- Aula 32 Modelos Hierárquicos
- Aula 33 Laboratório: Modelos Hierárquicos
- Aula 34 Defesas dos Trabalhos e divulgação do tema do Trabalho de recuperação

Aula 35 Encontro em laboratório para auxílio à implementação do trabalho de recuperação.

Aula 36 Defesa dos trabalhos de recuperação.

### **11) Bibliografia Básica**

- Fundamentals of Interactive Computer Graphics. ANGEL, EDWARD, DAM, A., VAN, FEINER, S., FOLEY, JAMES D. ADDISON WESLEY (PEARSON), 1995, 2a. Edição.
- Computação Gráfica - Teoria e Prática. Conci, Aura; Azevedo, Eduardo. Editora CAMPUS, 2003.
- Computação Gráfica - Teoria e Prática Vol. 2 Conci, Aura; Leta, Fabiana; Azevedo, Eduardo / CAMPUS, 2007.
- Material online disponibilizado no site da disciplina.

### **12) Bibliografia Complementar**

- Principles of Interactive Computer Graphics. Williem Newman & Robert Sproull. McGraw-Hill/Kogakusha
- Interactive Computer Graphics. McGraw-Hill (Livro verde)
- Computer Graphics, C Version, Second Edition by Donald Hearn and M. Pauline Baker, Prentice-Hall, ISBN: 0135309247.
- The OpenGL Super Bible. 2nd. Edition.
- "OpenGL 1.2 Programming Guide, Third Edition: The Official Guide to Learning OpenGL, Version 1.2" by Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, OpenGL Architecture Review Board, Addison-Wesley Pub Co; ISBN: 0201604582
- "OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.2" by Dave Shreiner (Editor), Opengl Architecture Review Board, Addison-Wesley Pub Co; ISBN: 0201657651