Computação Gráfica

Ciência da Computação | Engenharia da Computação

Prof.: Rafael Peiter | rpeiter@unisc.br

UNISC
UNIVERSIDADE DE SANTA CRUZ DO SUL

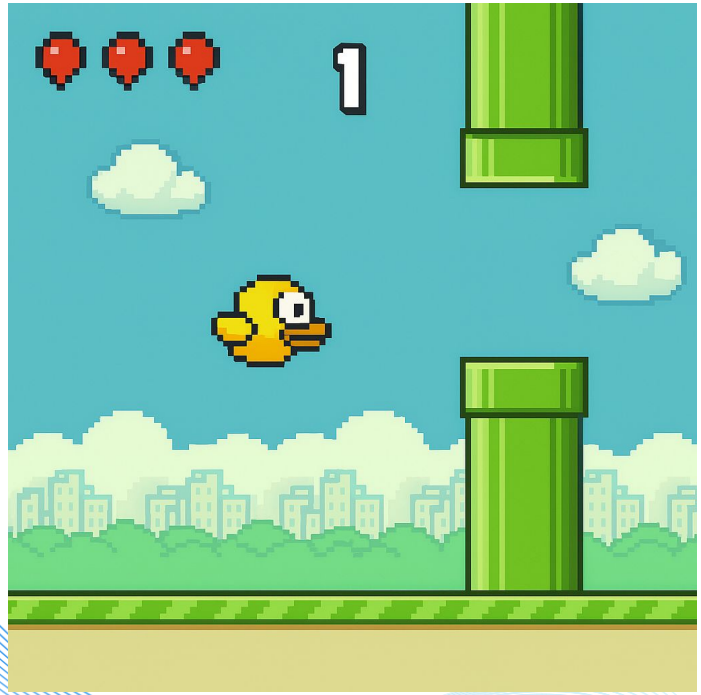
UNISC
UNIVERSIDADE DE SANTA CRUZ DO SUL

Computação Gráfica
Aula 04 - Definição do trabalho

Departamento de Informática - UNISC
Professor: Rafael Peiter
rpeiter@unisc.br

Definição do trabalho

- Criar um programa no formato do jogo **Flappy Bird** utilizando os conceitos de computação gráfica estudados na disciplina



Requisitos do programa

- Deve ser feito na linguagem **Python**

- Será permitido utilizar **somente** as seguintes bibliotecas:
  - PyOpenGL_accelerate
  - glfw
  - Pillow (PIL)
  - time
  - numpy
  - random

Formato do trabalho

- O trabalho poderá ser feito em grupos de até **5 pessoas**
- A apresentação de cada trabalho será feita para o professor
- Cada grupo terá o tempo máximo de **10 minutos** com tolerância de **5 minutos** para apresentar

Critérios de avaliação

- A avaliação ocorrerá de duas formas
  1. Avaliação do **código fonte** do programa conforme os critérios atingidos
  2. Avaliação da **apresentação** de **cada participante** do grupo

Critérios de avaliação

- Fazer o personagem voar apertando a tecla ESPAÇO: **1.0 pontos**
- Fazer os obstáculos e personagem se movimentarem: **1.0 pontos**
- Realizar o tratamento de colisão: **1.0 pontos**
- Contador de quantos obstáculos já passou: **1.0 pontos**
- Exibir um contador de vidas, o jogo só finaliza quando termina o número de vidas: **1.0 pontos**

Critérios de avaliação

- Criar objetos que aparecem aleatoriamente no jogo, pode ser vidas, alternador de velocidade, etc: **1.0 ponto**

- A entrega do trabalho deverá ser feita enviando os arquivos do projeto no **ambiente virtual** e compartilhando um link do **github** que conste todo o código fonte, uma explicação do projeto e algumas telas do jogo: **1.0 ponto**

- Apresentação do trabalho: **3.0 pontos**

Critérios de avaliação

- Na apresentação todos devem falar, será feito perguntas específicas para cada integrante do grupo. Será descontado pontos da nota do trabalho caso algum integrante não responda adequadamente às perguntas

- Devem ser utilizados os comandos vistos em aula, podem ser utilizados comandos adicionais. **Atenção! Se a estrutura do programa for diferente do que foi visto em aula, será descontado pontos**

Dicas

- Podem personalizar os personagens e obstáculos, usem a criatividade!
- Tanto o personagem quanto os obstáculos podem ser representados por qualquer primitiva geométrica
- Fiquem livre para criar conceitos de vida, condição para terminar o jogo, pontuação, ranking, etc
- Pode ser feito tanto no ambiente 2D quanto no 3D

Dicas

- Utilizem a variável de tempo para controlar os movimentos das primitivas
- Fiquem livres para criar múltiplas classes ou arquivos para o projeto
- Utilizem vetores para armazenar as posições
- Definam variáveis públicas para determinar o tamanho dos sprites, range de colisão, velocidade, etc