JOGOS 2D COM OPENGL

Curso: Ciência da Computação **Turmas:** CC7M
Professor: Célio Flores Siqueira Jr. **Ano:** 2008

Este trabalho prático da disciplina de Computação Gráfica, desenvolvido em **grupos de até 4 pessoas**, consiste na implementação de um jogo usando a linguagem C, a API OpenGL e o toolkit glut, aplicando os conceitos aprendidos na disciplina, como animação, transformações geométricas, uso de menus, etc.

1. OBJETIVO

Além de aprender a utilizar a biblioteca OpenGL em um projeto um pouco maior, fixar os principais conceitos de Computação Gráfica, que envolvem, basicamente, as transformações geométricas, a criação e manipulação de instâncias, animação, e a visualização de cenas 2D.

2. APRESENTAÇÕES DOS TRABALHOS

Os trabalhos serão apresentados no laboratório. Os alunos deverão mostrar o programa funcionando corretamente, explicar os algoritmos e estruturas de dados utilizados, demonstrar o correto funcionamento do sistema e **responder a questionamentos oralmente**. Após a apresentação cada grupo deve enviar uma cópia do programa fonte e dos executáveis, **compactados em um só arquivo zip**, para o email celiofsj@yahoo.com.br.

3. AVALIAÇÃO

Os trabalhos serão avaliados pelo correto funcionamento do programa, domínio dos conceitos empregados, técnicas de programação empregadas (estrutura do **CÓDIGO DOCUMENTADO**, lógica de programação, estruturas de dados) e organização da apresentação. Os trabalhos devem atender aos requisitos mínimos exigidos, mas nada impede que você adicione novos componentes que enriqueçam e melhorem o jogo (é claro, devem se encaixar na proposta).

4. VALOR DA NOTA MÁXIMA

4,0 pontos.

5. DATA DAS APRESENTAÇÕES

18 de junho (quarta-feira).

6. OBSERVAÇÕES IMPORTANTES

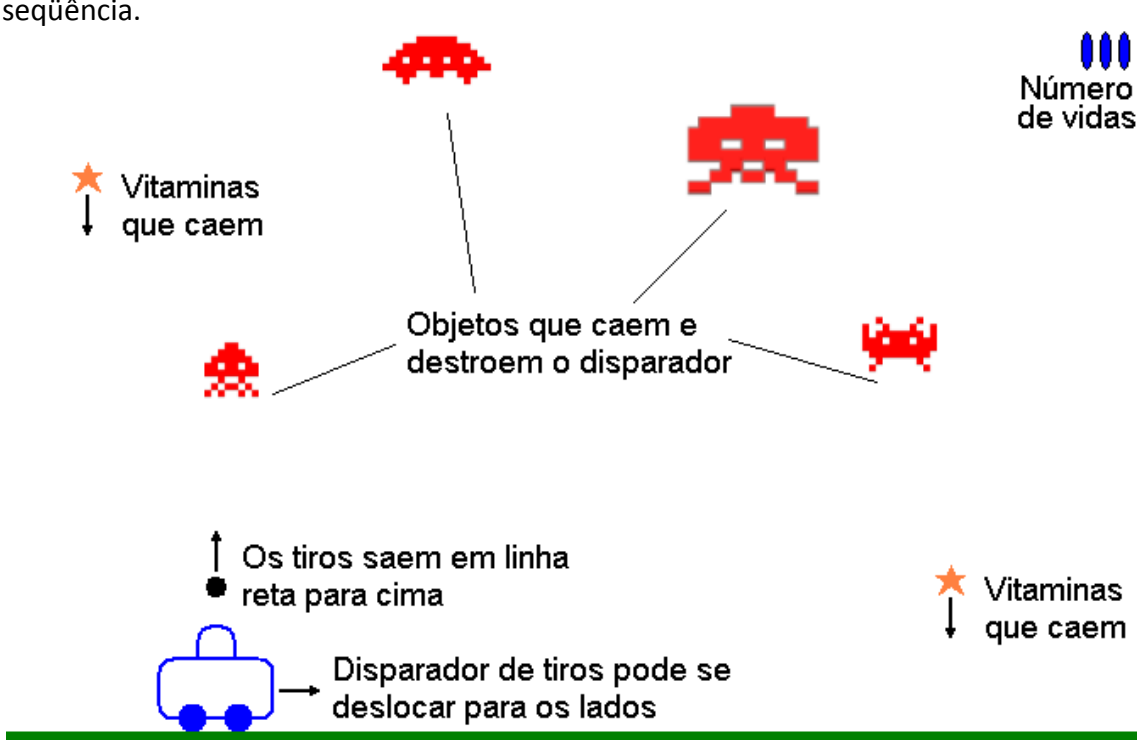
6.1. Todos os trabalhos deverão ser entregues e apresentados na data marcada, NÃO SENDO ACEITOS TRABALHOS ATRASADOS.
6.2. Os trabalhos que contiverem erros de compilação que impeçam sua execução receberão nota zero.
6.3. O trabalho deve ser OBRIGATORIAMENTE desenvolvido em C, usando a OpenGL e o toolkit glut, em plataforma Windows.
6.4. Trabalhos copiados, **de qualquer que seja a fonte** (internet, colegas de turma, **TRABALHOS DE PERÍODOS ANTERIORES**, etc.) resultarão em nota zero para todos os alunos envolvidos.
6.5. Receberá 1,0 ponto extra quem conseguir implementar seu jogo em 3D (verdadeiramente em 3D!).

TEMAS

(Lembre-se que é permitido e altamente recomendável que você incremente e crie novas funcionalidades para os sistemas propostos)

1. Space Invaders

Breve Descrição: Neste jogo o usuário deverá manipular um disparador de tiros com o objetivo de eliminar objetos que caem do céu e podem destruí-lo. Além disso, também caem "vitaminas" que podem lhe dar uma vida extra. A idéia é que jogo comece com três "vidas", de maneira que se o disparador for atingido por um objeto o jogo ainda pode continuar. Um exemplo do cenário do jogo é apresentado na figura abaixo. Os modelos dos objetos e da vitamina são de livre escolha, contanto que sejam identificáveis. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.



Algumas "regras" para o desenvolvimento do jogo:

- O disparador de tiros deve ser manipulado pelo usuário com as setas para esquerda e para direita, mas ele não pode passar dos limites laterais da janela.
- Além de movimentar o disparador, deve ser possível atirar cada vez que o usuário pressionar a tecla 'a'.
- É necessário ter, no mínimo, quatro modelos de objetos que irão destruir o disparador, e um modelo de vitamina. Os modelos devem ser armazenados apenas uma vez, e instanciados várias vezes.
- As vitaminas e os objetos que forem destruídos ou capturados (no caso da vitamina) não devem mais aparecer. Se eles chegarem até o chão sem serem destruídos, eles deverão começar a cair novamente.
- Devem existir, pelo menos, 15 objetos e 3 vitaminas em cada jogada.
- O jogador possui três "vidas", isto é, se um objeto colidir com o disparador, ele perde uma vida.
- O jogo termina quando todos os objetos forem destruídos ou quando as vidas acabarem, ou seja, o que acontecer primeiro.

- Para calcular a intersecção do tiro com os objetos e dos objetos e vitaminas com o disparador, pode ser usado o x e y mínimo e máximo.
- Se o jogador conseguir destruir todos os objetos, uma mensagem avisando que ele ganhou o jogo é exibida e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

Resumindo: O programa deverá permitir o deslocamento de um disparador sempre que as setas para a esquerda e para a direita forem pressionadas, e o disparo de tiros sempre que a tecla 'a' for pressionada. Se um objeto tocar o disparador, o jogador perde uma vida. Depois da terceira vez, o programa é encerrado. Se uma vitamina tocar o disparador, o jogador ganha um vida. Se todos os objetos forem destruídos, o jogador ganha e o jogo é encerrado.

2. PacMan

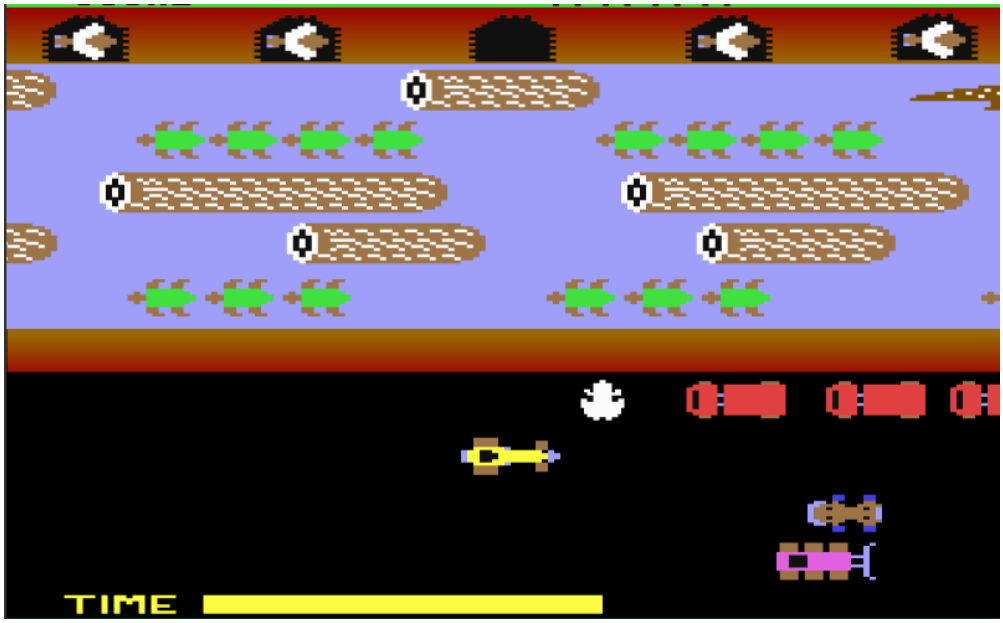
Breve Descrição: Neste jogo o usuário deverá manipular o personagem principal através de um labirinto, coletando objetos enquanto é perseguido por certo número de fantasmas. A idéia é que jogo comece com três "vidas", de maneira que se o personagem for atingido por um fantasma o jogo ainda pode continuar. O cenário do jogo deverá ser **BASEADO** no da figura ao lado. Os modelos dos objetos e personagens são de livre escolha, contanto que sejam identificáveis. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.

Algumas "regras" para o desenvolvimento do jogo:

- O PacMan deve ser manipulado pelo usuário com as setas direcionais.
- É opcional a implementação das “passagens secretas” (sai por um lado da tela e entra por outro).
- O jogador possui três "vidas", isto é, se um fantasma colidir com o PacMan, ele perde uma vida.
- O jogo termina quando todos os objetos forem coletados ou quando as vidas acabarem, ou seja, o que acontecer primeiro.
- Se o jogador conseguir coletar todos os objetos, uma mensagem avisando que ele ganhou o jogo é exibida e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

3. Frogger

Breve Descrição: O jogador controlará um sapo que tenta atravessar um movimentado cenário, repleto de obstáculos com carros, jacarés, água, caminhões, etc. A idéia é que jogo comece com três "vidas", de maneira que se o sapinho for atingido por um objeto o jogo ainda pode continuar. O cenário do jogo deverá ser **BASEADO** no exemplo da figura abaixo. Os modelos dos objetos são de livre escolha, contanto que sejam identificáveis. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.

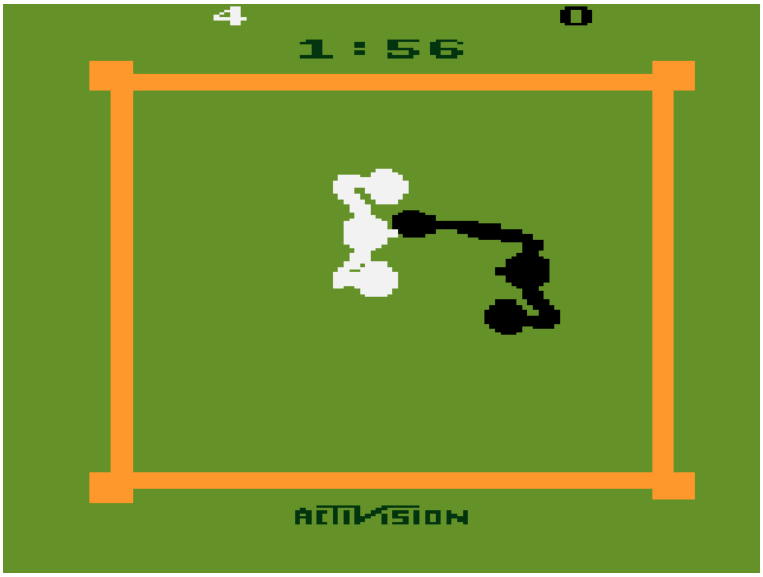


Algumas "regras" para o desenvolvimento do jogo:

- O sapo será manipulado com as setas direcionais, podendo se movimentar em todas as direções (cima, baixo, esquerda e direita), mas ele não pode passar dos limites laterais da janela.
- O sapo conseguirá atravessar todo o cenário com 10 passos (a frente).
- Produza pelo menos três cenários diferentes, para que sejam implementadas três fases do jogo (se o sapo conseguir chegar ao outro lado, automaticamente se passa de fase).
- É necessário ter, no mínimo, quatro modelos de objetos que irão colidir com o sapo. Os modelos devem ser armazenados apenas uma vez, e instanciados várias vezes.
- Devem existir, pelo menos, 20 objetos em cada jogada.
- O jogador possui três "vidas", isto é, se um obstáculo colidir com o sapo, ele perde uma vida.
- O jogo termina quando todas as fases forem concluídas ou quando as vidas acabarem, ou seja, o que acontecer primeiro.
- Se o jogador chegar ao fim do jogo, uma mensagem avisando que ele ganhou o jogo é exibida e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

4. Boxing

Breve Descrição: Aqui dois jogadores irão manipular lutadores de boxe virtuais. Vence um *round* quem conseguir atingir o rosto do oponente mais vezes dentro de um limite de tempo. Quando um jogador é atingido, é jogado para trás. A luta consiste de três *rounds* com tempo fixo. É declarado vencedor da luta quem vencer mais *rounds*. Um exemplo do cenário do jogo é apresentado na figura abaixo. Os modelos dos lutadores e o cenário são de livre escolha, contanto que sejam identificáveis. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.



Algumas "regras" para o desenvolvimento do jogo:

- Os lutadores serão controlados por conjuntos de teclas diferentes, podendo se movimentar em todas as direções (cima, baixo, esquerda e direita), contanto que não ultrapassem os limites do rinque.
- É considerado um ponto para quem acertar o rosto do adversário (centro da cabeça).
- O jogo termina quando os três *rounds* forem disputados. Se houver empate e não for possível determinar o vencedor, mais um *round* deve ser disputado para decidir o campeão.
- Ao fim do jogo, uma mensagem avisa quem ganhou e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

5. Frostbite

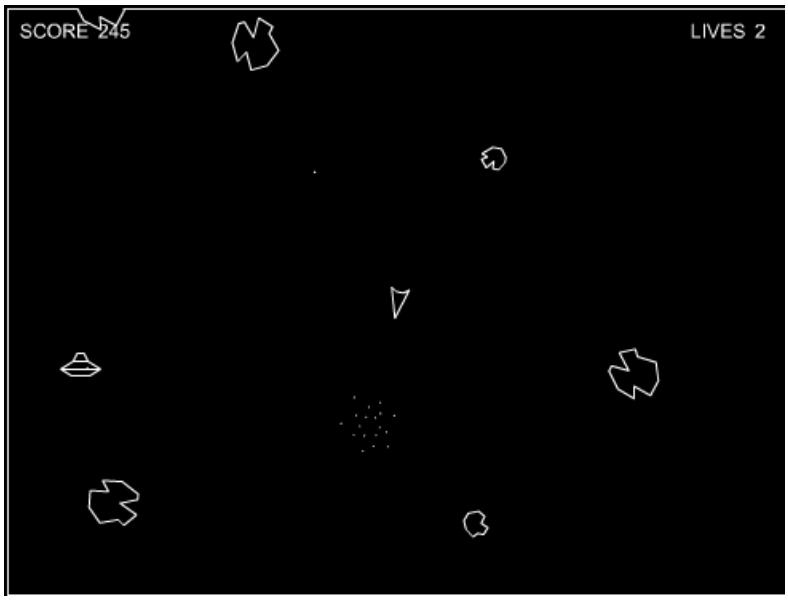
Breve Descrição: Semelhante ao jogo *Frogger*. Neste jogo existe um rio na metade inferior da tela com quatro fileiras de icebergs que se movem em sentidos alternados. Controlando um esquimó, o jogador deve pular sobre os icebergs enquanto evita tocar nos peixes e gaivotas. O objetivo é alcançar a entrada do iglu e passar para a próxima fase. Na terceira e última fase, ainda será necessário escapar de um urso polar que ronda a entrada do iglu (fica caminhando de um lado para outro). A idéia é que jogo comece com três "vidas", de maneira que se o esquimó for atingido por um objeto, o jogo ainda pode continuar. O cenário do jogo pode ser **baseado** na figura abaixo. Os modelos dos objetos são de livre escolha, contanto que sejam identificáveis. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.

Algumas "regras" para o desenvolvimento do jogo:

- O esquimó será manipulado com as setas direcionais, podendo se movimentar para cima e para baixo.
- O esquimó conseguirá chegar ao iglu após ultrapassar 4 níveis de icebergs.
- Produza pelo menos três cenários diferentes, para que sejam implementadas três fases do jogo (se o esquimó conseguir chegar ao iglu, automaticamente se passa de fase).
- É necessário ter, no mínimo, dois modelos de objetos que irão colidir com o sapo. Os modelos devem ser armazenados apenas uma vez, e instanciados várias vezes.

- O jogador possui três "vidas", isto é, se um obstáculo colidir com o esquimó, ele perde uma vida.
- O jogo termina quando todas as fases forem concluídas ou quando as vidas acabarem, ou seja, o que acontecer primeiro.
- Se o jogador chegar ao fim do jogo, uma mensagem avisando que ele ganhou o jogo é exibida e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

6. Asteroids



Breve Descrição: Neste jogo o usuário deverá manipular um disparador de tiros localizado no centro do cenário, com o objetivo de eliminar asteróides de todos os tamanhos que vagam pelo universo, vindos de toda a parte e podem destruí-lo. Além disso, existem asteróides especiais (identificados pela cor vermelha) que, caso sejam destruídos, podem lhe dar uma vida extra. A idéia é que jogo comece com três "vidas", de maneira que se o disparador for atingido por um asteróide, o jogo ainda pode continuar. Um exemplo do cenário do jogo é apresentado na figura abaixo. Os modelos dos asteróides são de livre escolha, contanto que sejam identificáveis e de variados tamanhos. Entretanto, algumas regras devem ser seguidas, conforme especificado na seqüência.

Algumas "regras" para o desenvolvimento do jogo:

- O disparador de tiros deve ser manipulado pelo usuário com as setas direcionais (cima, baixo, esquerda, direita).
- Além de movimentar o disparador, deve ser possível atirar cada vez que o usuário pressionar a tecla 'a'.
- É necessário ter, no mínimo, quatro modelos (e três tamanhos diferentes) de asteróides que irão destruir o disparador. Os modelos devem ser armazenados apenas uma vez, e instanciados várias vezes.
- Os asteróides que forem destruídos ou capturados não devem mais aparecer.
- Devem existir, pelo menos, 20 asteróides em cada jogada.
- O jogador possui três "vidas", isto é, se um asteróide colidir com o disparador, ele perde uma vida.
- O jogo termina quando todos os asteróides forem destruídos ou quando as vidas acabarem, ou seja, o que acontecer primeiro.
- Se o jogador conseguir destruir todos os asteróides, uma mensagem avisando que ele ganhou o jogo é exibida e a execução do programa é encerrada.
- Deve ser possível sair do jogo a qualquer momento pressionando a tecla ESC.

DIVISÃO DOS GRUPOS

GRUPO	COMPONENTES	TEMA
1	•	
2	•	
3	•	
4	•	
5	•	
6	•	
7	•	
8	•	