

Trabalho Prático 1

Neste primeiro trabalho prático vamos experimentar programação orientada a eventos usando OpenGL. Para isso, vamos implementar um video game simples. Um dos primeiros videogames que surgiu se chamava Galaxian. Nele, o jogador pilota uma nave que fica na parte de baixo da tela, e com ela se defende de um ataque alienígena. Os alienígenas realizam o seu ataque como uma grande esquadra que se movimenta lateralmente na parte superior da tela. Os diversos alienígenas podem soltar bombas contra a heróica nave e, de tempos em tempos, saem da formação e mergulham individualmente ou em pequenas esquadrilhas num ataque mais agressivo contra o defensor solitário que, por sua vez, também se movimenta lateralmente esquivando-se das bombas e disparando mísseis contras os invasores alienígenas.

A figura ao lado é um screenshot do Galaxian original. Na configuração básica do trabalho, vamos modificar e resumir um pouco o jogo.

Primeiramente, o jogo terá um fim. Se o jogador elimina todos os alienígenas, ele é declarado vencedor. Porém, se algum alienígena conseguir chegar ao solo, então ele ativa sua arma secreta mutante e será o vencedor. Além disso, a interface original do jogo era feita com 3 botões, sendo um para lançar os mísseis (apenas 1 míssil por vez na tela), e os outros dois para movimentar a nave defensora para a esquerda e direita. Vamos mudar essa interface de forma que a movimentação da nave seja controlada pelo mouse. A posição dele na tela indica a velocidade da nave defensora, ou seja, o mouse na lateral esquerda da tela significa a nave mover para a esquerda com velocidade máxima, enquanto que o mouse na lateral direita indica velocidade máxima para a direita. O mouse centralizado (e pode se criar um “dead zone” para o mouse, uma região que seria considerada região neutra) indica a nave parada na tela. A medida que o mouse se desloca para a direita e para a esquerda a nave defensora acelera na mesma direção. O disparo dos mísseis pode ser realizado pressionando o botão esquerdo do mouse.



Além disso, ainda na funcionalidade básica, o jogo deve interpretar o pressionamento da tecla R como comando para reiniciar o jogo, e a tecla ESC para terminar o programa. O botão direito do mouse deve colocar o jogo em pausa, deixando o mesmo congelado até que o botão direito seja novamente pressionado. E finalmente, uma opção de depuração deve ser reconhecida, ou pressionando-se a tecla D, ou pelo botão do meio do mouse. Nesse modo, o jogo também entra em modo pausa, porém imprime os parâmetros de todos os objetos ativos na tela (alienígenas, nave, míssil e bombas). Desse modo de depuração o jogador pode voltar ao modo de jogo normal pressionando o botão de direita (que tira o jogo do modo pausa). Pode também andar apenas uma interação no ciclo do jogo, atualizando todos os objetos na tela e novamente imprimido os

parâmetros de cada um. Isso é feito pressionando novamente ou a tecla D ou o botão do meio do mouse. Esse conjunto de funcionalidades básicas vale 70% da nota desse trabalho.

A avaliação do trabalho será baseada no efeito global obtido pelo programador, num critério empírico-adaptativo. Os gráficos não precisam ser absurdamente trabalhados, mas é preciso que fiquem “bonitinhos” de forma que o jogo tenha uma boa apresentação. Pode, no entanto, ser feito desenhando pequeno polígonos coloridos usando as primitivas clássicas do OpenGL.

Para se obter o restante dos pontos do trabalho (ou até mesmo mais pontos extras) funcionalidades adicionais podem/devem ser implementadas no jogo. Essas funcionalidades serão avaliadas conforme a dificuldade da implementação, o efeito obtido com ela no jogo, e a qualidade da implementação. Exemplos de funcionalidades extras com suas respectivas pontuações:

- 1) Implementação de múltiplas fases no jogo, com aumento progressivo na dificuldade. O jogo pode até ficar infinito. (5-10%)
- 2) Implementação de múltiplas chances (ou vidas) para o defensor. (5%)
- 3) Os gráficos podem ser feitos com texturas, inclusive texturas do jogo original (5%)
- 4) Os objetos gráficos podem ser animados, ou seja, a apresentação deles varia com o tempo (5-10%)
- 5) O alienígena pode, ao descer, ao invés de destruir e vencer o jogo, voltar para a esquadra (pode ser por rotação, ou seja, na medida que ele some na parte inferior da tela, ele reaparece na parte superior (10%)
- 6) Os alienígenas podem, ao invés de lançarem apenas bombas, deixar cair segredos industriais alienígenas que podem ser usados para melhorar características da nave, por exemplo: vida extra, escudo defletor, míssil de proximidade explosivo, entre outros (use sua criatividade). (10-15%)
- 7) Colocar um mecanismo de pontuação do jogador e colocar um placar na tela. Cada nave que a defensora destrói pode valer mais ou menos pontos dependendo da nave e das circunstâncias. (5%)
- 8) Qualquer outra idéia que torne o jogo melhor ou mais bonito, como por exemplo, colocar sons, fazer explosões mais bem elaboradas ou colocar estrelas coloridas no fundo da tela, como era feito no jogo original. (5-10%)

O trabalho deve ser individual. Podem discutir idéias e etc mas cada um deve ter a sua implementação independente dos colegas. Deve ser entregue um arquivo .tar.gz ou .zip contendo todo o programa fonte, com os Makefiles e bibliotecas necessárias para a compilação do programa. Nesse pacote deve haver também um arquivo README com uma descrição da sua implementação, e as instruções para a compilação e a execução do seu jogo. E finalmente deve haver um filme em formato AVI de aproximadamente 1 minuto com o jogo em ação. Esse vídeo pode ser produzido com um capturador de tela, com alguém jogando o jogo. O objetivo de se ter esses vídeos é produzir uma página final da disciplina com um vídeo curto de cada jogo produzido.

Qualquer dúvida, entre em contato comigo. Ou acrescente a sua interpretação no arquivo README, e mãos a obra.

Bom trabalho,
Renato.