

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Instituto de Informática

Trabalho final de Fundamentos de Computação Gráfica

Bruno Silva / Leonardo Korndorfer

01 de Julho de 2014

Sumário

1	Introdução	2
2	Metodologia	3
3	Código	4
3.1	Estrutura	4
3.2	Funcionamento	4
4	Dificuldades Encontradas	5
5	O Que Faltou	6
6	O Que Não Faltou	7

Capítulo 1

Introdução

Esse relatório fala sobre o desenvolvimento do jogo Pengo, proposto como trabalho final pela disciplina de Fundamentos de Computação Gráfica.

Capítulo 2

Metodologia

Como exigido pela definição, o trabalho foi desenvolvido em C++ usando openGl e Glut. Para integração do código desenvolvido pela dupla, controle da divisão de tarefas e checklist foi usado o repositório GitHub. As únicas partes do código não desenvolvidas pela dupla são as classes de carregamento de objetos 3d e de bitmaps disponibilizadas pelo professor.

Capítulo 3

Código

3.1 Estrutura

O código feito para o desenvolvimento dessa tarefa utiliza uma estrutura em classes. Foi criada uma classes abstrata chamada “Entity”, nela foram definidos todos os métodos e atributos comuns às entidades do jogo, como, por exemplo, posição na tela, colisões e movimentos. As seguintes classes herdam os métodos e atributos de Entity: Hero, que representa o personagem principal; Enemy, que representa os inimigos; Item, que representa os itens; Crate, que representa os blocos. Também foi criada uma classe chamada ‘Stage’, responsável por representar o estado do cenário. O cenário tem o tamanho padrão de 32 células por 32 células e é inicializado baseado em uma imagem com as mesmas dimensões. Nessa imagem, os pixels que tiverem a cor preta correspondem às células que serão iniciadas com blocos, os vermelhos correspondem às que serão iniciadas com inimigos e os brancos correspondem às que serão inicializadas vazias.

3.2 Funcionamento

Em linhas gerais, o programa funciona da seguinte forma: para cada objeto no jogo é feita uma instanciação da classe correspondente. Essas instancias são colocadas em um vetor que, a cada renderização da cena, será percorrido chamando o método Draw de cada objeto pertencente a ele. Alterações na cena são refletidas nos atributos dos objetos alterados que, por sua vez, irão refletir na renderização do objeto quando chamado o método Draw. Para eliminar algum elemento da cena, basta excluir do vetor seu objeto correspondente.

Capítulo 4

Dificuldades Encontradas

Entre as dificuldades encontradas, as que tomaram mais tempo durante o desenvolvimento foram: carregar, de maneira correta, a imagem que modela o cenário; achar modelos 3d livres para uso e que estivessem dentro das nossas necessidades.

Capítulo 5

O Que Faltou

Para que o jogo seja considerado completo, faltou a implementação de algumas características. São elas: Alguns tipos de colisão, minimapa do jogo, a inserção de itens e o comportamento dos inimigos quando próximos do personagem principal. Embora nem todas as características tenham sido implementadas, o código foi bem estruturado, permitindo com que várias partes já implementadas sejam reutilizadas para completar os comportamentos faltantes.

Capítulo 6

O Que Não Faltou

Foram implementados os seguintes comportamentos: movimentos interativos do personagem principal, criação do cenário baseado em um bitmap, movimentos dos inimigos baseado em um grid imaginário, comportamento aleatório dos inimigos, colisão entre os inimigos, colisão dos inimigos com blocos e os três tipos de visualização da cena.