	<p>Centro Tecnológico Departamento de Informática</p>
<p>Disciplina: Computação Gráfica</p>	<p>Código: INF09282 e INF09284</p>
<p>Prof. Thiago Oliveira dos Santos</p>	

Trabalho Curto 2

1 Introdução

Esse trabalho tem como objetivo aprimorar o conhecimento dos alunos em relação ao tópico de formato de arquivos gráficos usando dispositivos gráficos e representação da informação visual.

Para isso, o aluno deverá implementar um programa que lerá informações descrevendo uma arena de um arquivo do tipo Scalable Vector Graphics (SVG), e desenhará a arena com seus respectivos elementos na tela. A arena deverá ser interativa, ou seja, quando o usuário clicar em seus elementos, o nome do elemento deverá ser impresso no terminal. O trabalho deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freelut).

2 Especificação das Funcionalidades

Ao rodar, o programa deverá ler, de um arquivo de configurações (denominado “config.xml”), as configurações necessárias para suas tarefas. O arquivo de configurações deverá estar no formato xml e será fornecido juntamente com a especificação do trabalho. A localização do arquivo “config.xml” será fornecida pela linha de comando ao chamar o programa. Por exemplo, se o arquivo estiver dentro de uma pasta chamada “Test1” localizada na raiz, basta chamar o programa com “/Test1/” como argumento. As informações contidas nesse arquivo servirão para ler o arquivo SVG contendo as informações da arena.

O arquivo de configurações deverá conter uma tag xml global <aplicacao> com uma sub-tag específica para descrever a arquivo de entrada, denominada <arquivosDeEntrada>. A tag <arquivosDeEntrada> por sua vez terá uma sub-tag <arquivoDaArena> com atributos para descrever o nome, a extensão, e o caminho do arquivo descrevendo a arena (“nome”, “tipo” e “caminho” respectivamente). Perceba que o caminho do arquivo SVG descrevendo será especificado por esses 3 atributos combinados.

Exemplo do arquivo config.xml com um arquivo SVG dado por: ~/caminho/do/arquivo/arena.svg

```
<aplicacao>
  <arquivosDeEntrada>
    <arquivoDaArena nome="arena" tipo="svg" caminho="~/caminho/do/arquivo/"></arquivoDaArena>
  </arquivosDeEntrada>
</aplicacao>
```

Após ler as informações do arquivo de configurações, o programa deverá carregar os elementos da arena do arquivo do tipo SVG respectivo e desenhá-las na tela. A arena será composta por uma série de elementos (ver Figura 1): um retângulo branco representando a arena em si; um retângulo cinza representando um posto de abastecimento, um círculo verde representando o helicóptero do jogador, pelo menos um círculo vermelho representando helicópteros dos inimigos; e pelo menos um círculo azul representando objetos a serem resgatados. Todos esses elementos estarão contidos no arquivo SVG descrevendo a arena. Um arquivo SVG será fornecido como exemplo juntamente com a descrição do trabalho.

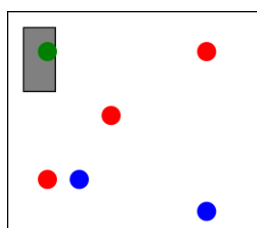



Figura 1: Exemplo de arena com seus elementos.

	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos		

Cada um dos elementos da arena, contidos no SVG, tem informação suficiente para desenhá-los na tela (ex. posição, tamanho, cor, etc.). A arena deverá ser desenhada em uma tela tamanho do retângulo branco que descreve a arena. Os outros elementos da arena deverão ser desenhados em cima da arena considerando as posições, tamanhos e cores lidos do arquivo SVG. Cada elemento da arena no arquivo SVG virá acompanhado de um “id” que deverá ser impresso no terminal, quando o usuário clicar em cima daquele elemento na arena (ex. se o usuário clicar no retângulo cinza, seu id=PostoAbastecimento deverá ser impresso no terminal).

3 Regras Gerais

O trabalho deverá ser feito individualmente. Trabalhos identificados como fraudulentos serão punidos com nota zero. Casos típicos de fraude incluem, mas não se restringem à cópias de trabalhos, ou parte dele, assim como trabalhos feitos por terceiros. Caso seja necessário confirmar o conhecimento do aluno a respeito do código entregue, o professor poderá pedir ao aluno para apresentar o trabalho oralmente em um momento posterior. A nota da apresentação servirá para ponderar a nota obtida no trabalho.

3.1 Entrega do Trabalho

O código deverá ser entregue por email (para: todosantos@inf.ufes.br) dentro do prazo definido no portal do aluno. Trabalhos entregues após a data estabelecida não serão corrigidos.

A entrega do trabalho deverá seguir estritamente as regras a seguir. O não cumprimento acarretará na **não correção do trabalho** e respectivamente na atribuição da nota zero.

- Assunto da mensagem: [CG-2015-2] <tipo do trabalho>. Onde, <tipo do trabalho> pode ser TC1, TC2, TC3 e representa respectivamente trabalho curto 1, 2, 3, etc, ou TF para o trabalho final.
- Anexo da mensagem: arquivo zippado (com o nome do autor, ex. FulanoDaSilva.zip) contendo todos os arquivos necessários para a compilação do trabalho;
- Não enviar arquivos já compilados, inclusive bibliotecas!
- O diretório deverá necessariamente conter um makefile que implemente as seguintes diretivas "make clean" para limpar arquivos já compilados, "make all" para compilar e gerar o executável. O executável deverá ser chamado *trabalhocg*.

Lembre-se que a localização do arquivo config.xml será passada via linha de comando e portanto não se deve assumir que haverá um arquivo desses na pasta do executável.

3.2 Pontuação

O trabalho será pontuado conforme a tabela abaixo. Bugs serão descontados caso a caso.

Funcionalidade	Peso
Ler e desenhar a arena e seus elementos corretamente	5
Responder corretamente aos cliques nos objetos	5

4 Erratas

Qualquer alteração nas regras do trabalho será comunicada em sala e no portal do aluno. É de responsabilidade do aluno frequentar as aulas e se manter atualizado.