	<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 3

1 Introdução

Esse trabalho tem como objetivo aprimorar o conhecimento dos alunos em relação ao tópico de transformações afins.

Para isso, o aluno deverá implementar um programa que coloque um helicóptero na arena implementada no trabalho curto 2. Nesse trabalho, a arena será estática, ou seja, não haverá movimentações ou interações de objetos de objetos, exceto o helicóptero do jogador. O helicóptero do jogador será controlado pelo teclado e pelo mouse, e poderá decolar, voar, pousar e atirar. O trabalho deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freeglut).

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.

Além das tags já especificadas no trabalho curto 2, o arquivo de configurações deverá conter uma sub-tag específica para descrever parâmetros do helicóptero, denominada <helicoptero>. A tag <helicoptero> terá atributos para descrever a velocidade do tiro, do helicóptero e da hélice (“velTiro”, “velHelicoptero” e “velHelice” respectivamente). Elas definirão o quanto um tiro e o helicóptero se movem e também o quanto a hélice gira a cada frame e deverão estar em unidades por milissegundos (como mostrado no laboratório). Percebam que os valores dados como exemplo foram escolhidos aleatoriamente e portanto podem não representar valores ótimos para teste.


Exemplo do arquivo config.xml

```
<aplicacao>
  <arquivosDeEntrada>
    <arquivoDaArena nome="arena" tipo="svg" caminho="~/caminho/do/arquivo/"></arquivoDaArena>
  </arquivosDeEntrada>
  <helicoptero velTiro="0.2" velHelicoptero="0.1"></helicoptero>
</aplicacao>
```

Após ler as informações do arquivo de configurações, o programa deverá carregar os elementos da arena do arquivo do tipo SVG respectivo e desenha-las na tela (assim como feito no trabalho curto 2), exceto o círculo do helicóptero que terá o helicóptero da Figura 1, ao invés de um círculo.

Helicóptero

O helicóptero deverá ser capaz de voar sobre a arena. Ele iniciará pousado na posição definida pelo centro do círculo verde na arena e ter o tamanho aproximado do círculo que o representa, ou seja, o helicóptero deverá estar contido no círculo. O helicóptero será composto por uma hélice com 4 paletas que deverá estar sempre girando, um corpo com uma cauda, e um canhão, ver Figura 1 como exemplo. Nesse trabalho, não é necessário se preocupar com colisões entre objetos. A velocidade de giro da hélice deverá ser controlada pelas teclas + e - que deverão aumentar e diminuir a velocidade respectivamente.

	<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>	

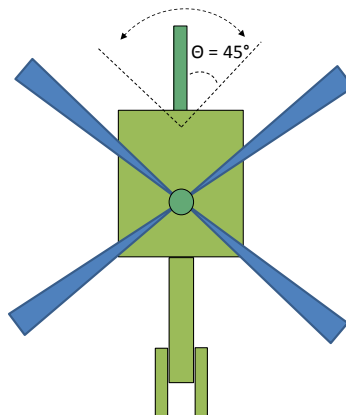


Figura 1: Helicóptero.

Decolar e Pousar

Para decolar e pousar o helicóptero o usuário deverá clicar no botão direito do mouse. Cada clique alternará o estado do helicóptero (pousado ou voando). Quando o helicóptero estiver voando, ele será 1.5 vezes maior do que quando estiver pousado. O efeito visual de pouso e decolagem não precisa ser implementado, ou seja, ao clicar o botão o helicóptero pode passar imediatamente de um estado para o outro.

Voar

Para voar, o helicóptero terá que ter decolado primeiro. O controle do voo será feito pelas teclas *a*, *s*, *d* e *w* do teclado. As teclas *a* e *d* servirão para girar o corpo do helicóptero no sentido anti-horário e horário respectivamente. As teclas *w* e *s* servirão para mover o helicóptero para frente e para trás respectivamente. A combinação de duas teclas, uma de giro e uma de movimento, deverão causar um movimento curvilíneo, como se estivesse fazendo uma curva.

Atirar


O helicóptero deverá atirar sempre que o botão esquerdo do mouse for clicado. O canhão do helicóptero será controlado pelo deslocamento lateral do mouse, ou seja, arrastar o mouse para a direita move o canhão para a direita e arrastar para a esquerda move para a esquerda. O canhão não deverá mover mais do que 45 graus do centro, ver Figura 1. Perceba que tiros que saírem da arena podem ser descartados. O tiro poderá ser representado por um círculo pequeno.

O aluno deverá utilizar os conceitos de *double buffer* e variável de estado das teclas para interação com teclado (usando o evento *KeyUp* como visto em aula). A utilização de conceitos de modularização (e.g. usando classes para representar os objetos da cena) facilitará a implementação dos trabalhos seguintes, como por exemplo uma classe helicóptero com funções *moverParaFrente*, *girarHelicoptero*, *desenharHelice*, etc. Evite usar variáveis globais dentro das funções, de preferência a passagem por parâmetros. Restrinja o uso de variáveis globais às chamadas mais externas.

OBS: É altamente recomendável que os alunos utilizem a ferramenta *valgrind* para detectar problemas de acesso à memória no código.

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.

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

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.

4 Pontuação

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

Funcionalidade	Peso
Pousar e decolar	2
Movimentos de voar	2
Girar hélice	2
Atirar	2
Mover canhão	2

5 Erratas

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