	<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 4

1 Introdução

Esse trabalho tem como objetivo integrar os conceitos dos trabalhos anteriores em um pequeno jogo.

O jogo terá como objetivo destruir todas as torres de comando nas ilhas sem tomar um tiro dos inimigos. O jogo acaba quando você destrói todas as torres de comando (vencendo), ou quando você morre (perdendo).

O aluno deverá implementar um programa que incremente o trabalho curto 3 (TC3) para adicionar as funcionalidades descritas aqui. Neste trabalho, os componentes da arena serão dinâmicos, ou seja, os inimigos deverão se mover “aleatoriamente” (o tipo de movimento é de livre escolha do aluno, mas seguindo os movimentos básicos definidos para o jogador no TC3, isto é, os inimigos não podem sair da arena, entrar nas ilhas, etc.). Além disso, os tiros deverão interagir com os inimigos (ou seja, ao atingi-los, deverão matá-los) e vice-versa (fogo amigo não mata). 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 (outros exemplos de caminhos possíveis “../Test1/”, “../../Test1/”, etc.). 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 e 3, o arquivo de configurações deverá conter uma sub-tag específica para descrever parâmetros dos inimigos, denominada <inimigo>, e uma para descrever as torres de comando de maneira geral, denominada <torre>. A tag <inimigo> terá, além de um atributo de velocidade do inimigo e velocidade do tiro, similares às tags do jogador (vide TC3), um atributo “freqTiro” para determinar a frequência dos tiros dos adversários, isto é, quantos tiros os adversários darão por instante de tempo. Ele será dado em número de tiros por segundos. Percebam que os valores dados como exemplo foram escolhidos aleatoriamente e, portanto, podem não representar valores ótimos para teste. A tag <torre> terá um atributo “tamanho” (dado em porcentagem do raio da ilha, ex. 20% no arquivo abaixo) que indicará o raio da torre de comando que deverá estar centrada em cada uma das ilhas.

Exemplo do arquivo config.xml

```
<aplicacao>
  <arquivoDaArena>
    <nome>arena</nome>
    <tipo>svg</tipo>
    <caminho>coloque o seu caminho aqui para testar</caminho>
  </arquivoDaArena>
  <jogador velTiro="0.2" vel="0.1"></jogador>
  <inimigo freqTiro="0.2" vel="0.1" velTiro="0.2"></inimigo>
  <torre tamanho="20"></torre>
</aplicacao>
```

Após ler as informações do arquivo de configurações, o programa deverá carregar os elementos da arena do arquivo do tipo SVG respectivo e colocar um jogador verde ao invés de um círculo verde, jogadores inimigos vermelhos ao invés de círculos vermelhos, além das torres de comandos e dos outros elementos da arena já definidos nos trabalhos anteriores. Os personagens serão conforme descritos no TC3.

	<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>	

Torpedo de Frente

Todos os inimigos poderão atirar torpedos de frente e não poderão atirar torpedos de cima. Os tiros do inimigo poderão matar o jogador quando ele estiver submerso e houver colisão com o tiro. Da mesma forma, o tiro do jogador poderá matar o inimigo. O jogador terá o torpedo de frente controlado conforme o TC3, e os inimigos deverão atirar de tempos em tempos (utilizar o parâmetro “freqTiro” do arquivo de configurações). Tiro amigo não mata! Ou seja, os inimigos não podem se matar.

Torres de Comandos

Cada ilha deverá ter uma torre de comando representada por um círculo amarelo de raio com X% (em que X vem do arquivo de configurações) do raio da ilha e localizado no centro da ilha. A torre de comando só poderá ser destruída por um torpedo de cima. A destruição ocorrerá quando o houver sobreposição do círculo pontilhado que define a posição de queda do torpedo e o círculo da torre de comando. A torre de comando deverá sumir ao ser destruída. Os torpedos de cima só destruirão as torres de comando, não tendo efeito sobre os inimigos. Nesta versão do trabalho, o torpedo de cima só poderá ser atirado com o submarino submerso e seguirá as mesmas funcionalidades já descritas no TC3.

Mover Inimigos

O movimento dos inimigos deverá ficar restrito a região da água, ou seja, eles não poderão sair do círculo azul da arena. Eles também estarão sempre submersos. Os inimigos devem colidir entre si, ou seja, dois inimigos não podem ocupar o mesmo espaço. Cada inimigo deverá ficar se movendo aleatoriamente (algoritmo de livre escolha do aluno, mas respeitando as propriedades de movimentos definidas no TC3) e de tempos em tempos atirar (considerar a frequência dada no arquivo de configurações). Exemplos de movimentos, dar voltas em torno do centro da arena; andar em zig zag; de tempos em tempos escolher uma direção e ir, etc. O movimento aleatório deverá permitir o inimigo colidir corretamente com os obstáculos e não ficar preso (ou travado) em uma só posição. Isso vale também para o caso dos inimigos se encontrarem e se prenderem. O jogador deverá se colidir normalmente com os inimigos quando ambos estiverem submersos, porém não haverá colisão se o jogador estiver emerso e o inimigo submerso.

Finalização do Jogo

O jogo deverá conter um placar no canto superior direito indicando o número de torres destruídas até o momento e o número que faltam. No final do jogo, uma mensagem deverá ser impressa na tela dizendo se você ganhou ou perdeu. O jogador ganha se ele destruir todas as torres, e perde se levar um tiro.

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 cópia de trabalho, 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: todsantos@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-2018-2] <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!

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

- 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. Seja cuidadoso ao testar o seu programa, isto é, não teste com o arquivo no diretório do programa, pois você pode esquecer de testá-lo em outro lugar posteriormente.

4 Pontuação

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

Funcionalidade	Peso
Funcionalidade do TC3	2
Torpedo de frente	2
Torres de comando	2
Mover inimigos	2
Finalização do Jogo	2

5 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.