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

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 matar todos os adversários sem tomar um tiro. O jogo acaba quando você mata todos os inimigos, ou quando você morre.

O aluno deverá implementar um programa que incremente o trabalho curto 3 (TC3) para adicionar as funcionalidades descritas aqui. Nesse 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, o jogador não pode passar dos obstáculos baixos sem pular, não pode sair da arena, etc.). Além disso, os tiros deverão interagir com os inimigos (ou seja, ao atingi-los, deverão matá-los) e vice-versa. 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 (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 o obstáculo baixo de maneira geral, denominada <obstaculo>. 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 <obstaculo> terá um atributo “altura” (dado em porcentagem do alcance total do pulo, ex. 80%) que indicará o quanto o jogador deve pular para subir nos obstáculos.


Exemplo do arquivo config.xml

```
<aplicacao>
  <arquivoDaArena nome="arena" tipo="svg" caminho="caminho aqui para testar"></arquivoDaArena>
  <jogador velTiro="0.2" vel="0.1"></jogador>
  <inimigo freqTiro="0.2" vel="0.1" velTiro="0.2"></inimigo>
  <obstaculo altura="80"></obstaculo>
</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 vermelhos ao invés de círculos vermelhos, além dos outros elementos da arena como definidos nos trabalhos anteriores. Os personagens serão conforme descritos no TC3.

Pulo

Além das definições de pulo especificada nos trabalhos anteriores, o pulo do jogador agora deverá considerar a altura dos obstáculos baixos (vinda do arquivo de configurações). Por exemplo, um pulo que dure 2 segundos, demora 1 segundo para atingir a altura máxima. Caso esse pulo esteja considerando um arquivo de configurações com uma altura de 80% para os obstáculos baixos, ele somente conseguiria subir nos obstáculos entre o intervalo de 0.8 até 1.2 segundos. Fora desse intervalo ele não teria altura

	<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>	

suficiente para subir. Uma vez em cima do obstáculo, o jogador deverá ter o seu tamanho proporcional à altura do obstáculo. Ao sair do obstáculo andando, o círculo deverá retornar ao seu tamanho original. Pode considerar o momento em que o círculo não sobrepõe (a sobreposição deve considerar o raio original) mais o obstáculo como ponto de saída de cima do obstáculo. Ao pular de cima de um obstáculo, o círculo deverá levar em conta que está saindo dali de cima, ou seja, deverá crescer mais e demorar mais para atingir o chão do que se tivesse saindo do chão (calcule o tempo proporcional a altura do obstáculo).

Tiro

Todos os inimigos poderão atirar. O jogador terá o tiro controlado conforme o TC3, e os inimigos deverão atirar de tempos em tempos (utilizar o parâmetro “freqTiro”). Para saber se um tiro acertou um um dos personagens, basta verificar se o círculo envolvendo o personagem toca o círculo que define o tiro. Perceba que esse círculo é “virtual”, ele serve apenas para calcular a colisão e não deve ser mostrado na tela. Tiro amigo não mata! Ou seja, os inimigos não podem se matar. As regras de tiro definidas no TC3 continuam valendo e também se aplicam aos inimigos.

Mover Inimigos

O movimento dos inimigos deverá ficar restrito a região da pista, ou seja, eles não poderão sair da pista. Os inimigos devem colidir entre si, ou seja, dois inimigos não podem ocupar o mesmo espaço. Cada inimigo adversário 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. Exemplos de movimentos, seguir a direção da pista para dar voltas; andar em zig zag; de tempos em tempos escolher uma direção e ir, etc. O movimento aleatório deverá permitir o inimigo pular e subir nos obstáculos quando chegar perto deles.

Finalização do Jogo

O jogo deverá conter um placar no canto superior direito indicando o número de inimigos mortos até o momento. No final do jogo, uma mensagem deverá ser impressa na tela dizendo se você ganhou ou perdeu. O jogador ganha se matar todos os inimigos, 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: 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-2017-2] <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. Seja cuidadoso ao testar o seu

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

programa, isto é, não teste com o arquivo no diretório do programa, pois você pode esquecer de testa-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
Pulo	2
Tiro	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 freqüentar as aulas e se manter atualizado.