	<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 uma submarino visto de cima na arena implementada no trabalho curto 2. Nesse trabalho, a arena será estática, ou seja, não haverá movimentações ou interações de objetos, exceto do jogador. O jogador será controlado pelo teclado e pelo mouse, e poderá atirar para frente e para cima, submergir, emergir e navegar pela arena, considerando as colisões do TC2. O funcionamento definido do trabalho 2 continuam valendo, exceto quando redefinidos especificamente aqui. 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, o arquivo de configurações deverá conter uma sub-tag específica para descrever parâmetros do jogador, denominada <jogador>. A tag <jogador> terá atributos para descrever a velocidade do tiro, e do jogador (“velTiro” e “vel” respectivamente). Elas definirão respectivamente o quanto um tiro e o jogador se movem a cada frame e deverão estar em unidades por milissegundos (como mostrado no laboratório). As velocidades do jogador e do tiro deverão ser invariantes a velocidade da máquina (i.e., o jogador com uma mesma informação no arquivo de configurações de se movimentar com a mesma velocidade em máquinas mais lentas ou mais rápidas)¹. 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>
  <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>
</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-los na tela (assim como feito no trabalho curto 2), exceto o círculo do personagem do jogador que terá o submarino do jogador da Figura 1, ao invés de um círculo.

¹ Para testar o correto funcionamento em velocidades de máquina diferentes, pode-se utilizar alguma função que coloque o *idle* em *sleep* por um curto tempo.

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

Jogador

O jogador deverá ser capaz de se movimentar pela arena e detectar as colisões, realizando as ações apropriadas para tratá-las que já foram implementadas no trabalho anterior. Ele iniciará na posição definida pelo centro do círculo verde na arena, apontando para cima e terá o tamanho aproximado do círculo que o representa, ou seja, o jogador deverá estar contido no círculo definido no arquivo de configurações. O jogador será composto por uma elipse verde, um leme para guiar sua direção, um círculo preto que indica a saída dos torpedos de frente e um círculo vermelho (a direita do submarino) que indica a saída dos torpedos de cima, e triângulos amarelos representando as hélices, ver Figura 1 como exemplo.

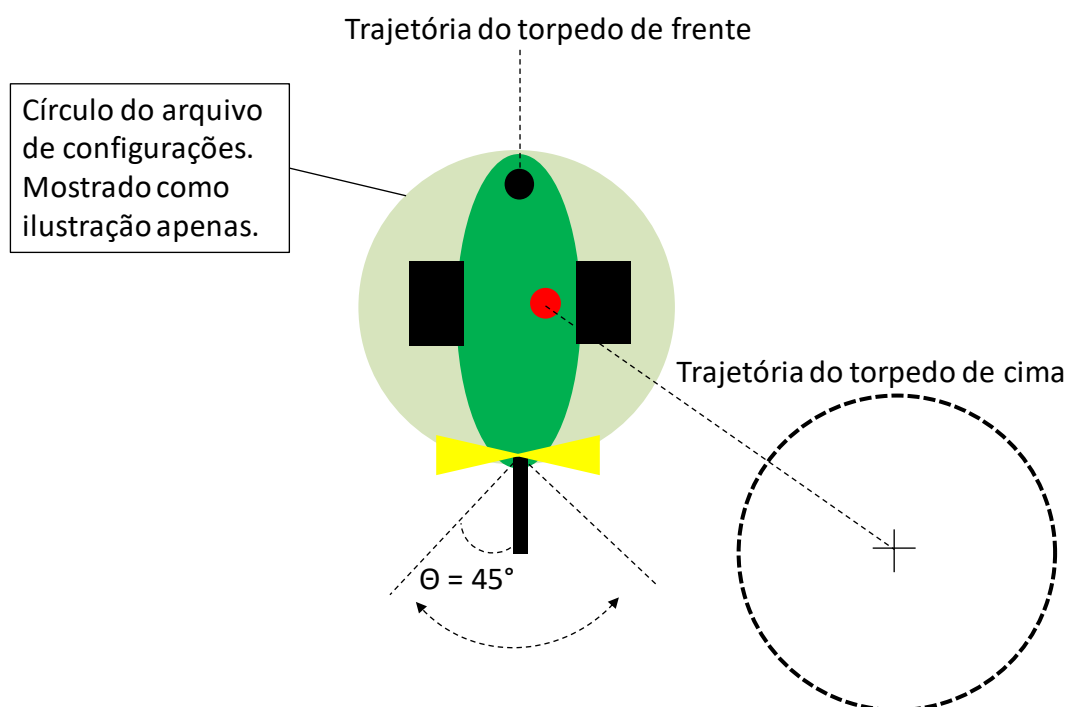



Figura 1: Submarino com leme (barra preta) que deve se movimentar de um lado para o outro, sendo no máximo a 45 graus para ambos os lados. As hélices são representadas pelos triângulos amarelos que girão com o movimento do submarino. O círculo preto indica as saídas dos torpedos que atiram para frente em linha reta. O círculo vermelho (localizado no lado direito do submarino) indica a saídas dos torpedos que atiram torpedos para cima para caírem no + marcado dentro do círculo lateral. O + será controlado pelo mouse.

Andar

O jogador deverá se mover para frente e para trás ao pressionar as teclas do “w” e “s” respectivamente. O leme do submarino deverá girar conforme a Figura 1 ao se pressionar as teclas direcionais “a” e “d” e servirá para guiar a direção do submarino. Os direcionais “a” e “d” controlarão respectivamente o giro no sentido horário e anti-horário em relação ao jogador. O submarino deverá fazer curva como um submarino real, ou seja, se os direcionais de curva forem pressionados com o submarino parado, somente o leme deverá virar; quando o submarino se mover, ele deverá curvar na direção definida pelo leme. Note que o submarino não deverá girar entre o próprio eixo e sim fazer uma curva de acordo com o leme. Essa proporção deverá levar em conta a informação de velocidade do jogador vinda do arquivo de configurações, ou seja, alterações nas velocidades do arquivo de configurações não deverão influenciar a quantidade de giro do submarino.

Hélices

As hélices serão usadas para representar o movimento do submarino. Elas deverão girar continuamente, porém toda vez que o submarino estiver se movendo, elas deverão girar mais rápido.

	<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>	

Submergir e Emergir

O jogador deverá implementar as mesmas funcionalidades de submersão e emersão do segundo trabalho, ou seja, ao apertar a tecla “u” em qualquer lugar da arena, ele deverá submergir/emergir conforme definido no trabalho 2. Ao manter a tecla “u” pressionada, o jogador deverá ficar emergindo e submergindo continuamente, mas respeitando do tempo de troca de estado definido no trabalho 2.

Atirar Torpedo de Frente

O jogador deverá atirar na direção para a qual o submarino estiver apontando sempre que o botão esquerdo do mouse for clicado, e uma vez atirado, o torpedo se torna independente do submarino, ou seja, alterações na rota do submarino não devem influenciar a rota do torpedo que deve continuar em linha reta. O torpedo deverá sair do círculo preto do submarino, ver Figura 1. Perceba que tiros que saírem da arena podem ser descartados. O tiro deverá ser representado por um retângulo, imitando um torpedo. A velocidade do tiro deverá levar em conta a informação vinda do arquivo de configurações. Tiros não devem ultrapassar os obstáculos, ou seja, eles devem sumir ao atingi-los. O jogador não deve atirar torpedo de frente enquanto estiver emerso. A colisão com o torpedo pode ser feita usando o centro do retângulo, ou seja, se o centro do retângulo do torpedo estiver dentro do círculo de interesse, então houve colisão.

Atirar Torpedo de Cima

Sempre que o botão direito do mouse estiver pressionado, o jogo deverá mostrar um + (com um círculo tracejado de raio igual ao do círculo jogador em seu entorno) na posição do cursor do mouse, ver Figura 1. O jogador deverá atirar torpedos para cima indo em direção ao + sempre que o botão direito do mouse for solto. O torpedo deverá sair do círculo vermelho do submarino e cair na posição onde o + estava quando o botão direito foi solto, ver Figura 1. Uma vez atirado o torpedo se torna independente do submarino, como o tiro para frente. O tiro deverá ser representado por um círculo pequeno, ou um ponto, imitando um torpedo visto de cima. Para simular o torpedo subindo e descendo, o tamanho do círculo deverá crescer e diminuir o seu tamanho durante a trajetória. O tiro começará com tamanho normal, atingirá o dobro do seu tamanho no meio da trajetória, e diminuirá até o tamanho normal novamente no final da trajetória. O tiro deverá sumir no final da trajetória.

O aluno deverá utilizar os conceitos de *double buffer* e variável de estado das teclas para interação com teclado (como visto em aula e feito no TC2). 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 jogador com funções moverParaFrente, girar, etc. Evite usar variáveis globais dentro das funções, de preferência a passagem por parâmetros, mesmo que os argumentos venham de variáveis globais. Restrinja o uso de variáveis globais às chamadas mais externas.

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 às 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-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;

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

- 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 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
Jogador	2
Andar	2
Atirar Torpedos para Frente	2
Atirar Torpedos para Cima	2
Hélices	1
Submergir e Emergir	1

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.