



Centro Tecnológico
Departamento de Informática

Disciplina: Computação Gráfica

Código: INF09282 e INF09284

Prof. Thiago Oliveira dos Santos

Trabalho 2D

1 Introdução

Esse trabalho tem como objetivo avaliar o conhecimento dos alunos em relação computação gráfica interativa 2D.

Para isso, o aluno deverá implementar um jogo de arena de combate. O jogo será composto por dois jogadores e uma arena com alguns elementos (obstáculos) que interagirão com o jogador. Os personagens dos jogadores serão controlados pelo teclado, sendo que o primeiro jogador poderá usar também o mouse, e poderá atirar. O objetivo do jogador é matar o jogador inimigo. 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 e interpretar os elementos da arena do arquivo do tipo SVG informado pela linha de comando. A arena será composta por uma série de elementos (ver Figura 1): uma arena representada sempre por um círculo azul; um círculo verde representando sempre o personagem do jogador 1 (controlado pelo mouse e teclado); um círculo vermelho representando sempre o personagem do jogador 2 (controlado teclado apenas) e círculos pretos representando obstáculos. Um arquivo SVG será fornecido como exemplo juntamente com a descrição do trabalho, porém é responsabilidade do aluno testar outros arquivos com configurações (posições e tamanhos) diferentes para os elementos. A leitura do SVG poderá ser feita utilizando-se um parser para XML. Sugiro utilizar a Tinyxml que é simples e pode ser enviada juntamente com o código para ser compilada (não é permitido usar bibliotecas pré-compiladas). Use o Inkscape para visualizar a imagem da arena.

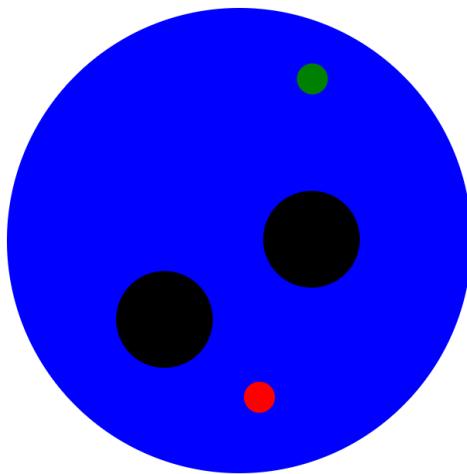


Figura 1: Arquivo SVG representando a arena em azul, obstáculos em preto, jogador 1 em verde e jogador 2 em vermelho.

Desenho

Os elementos lidos do arquivo SVG vão servir somente para desenhar a configuração inicial do jogo e não para desenhar os personagens completos, portanto ele conterá somente os elementos e cores descritos anteriormente e seguindo aquela definição. A arena será circular e vista por uma janela de visualização que engloba toda ela (com altura e largura iguais ao diâmetro da arena), sendo exibida em uma janela de 500x500 pixel do sistema operacional. Não é necessário tratar o *resize*.

Os círculos verde e vermelho servirão somente para saber onde e com qual tamanho desenhar respectivamente os personagens dos jogadores (utilizar as cores do círculo para diferenciar os personagens ao desenhá-los). Mais especificamente, os círculos representarão a posição inicial dos



personagens (olhando um para outro) e estarão sempre dentro da arena. O diâmetro do círculo definirá a cabeça dos personagens. Cada personagem será formado por uma cabeça, um tronco, um braço e duas pernas. Ver um exemplo do jogador na Figura 2.

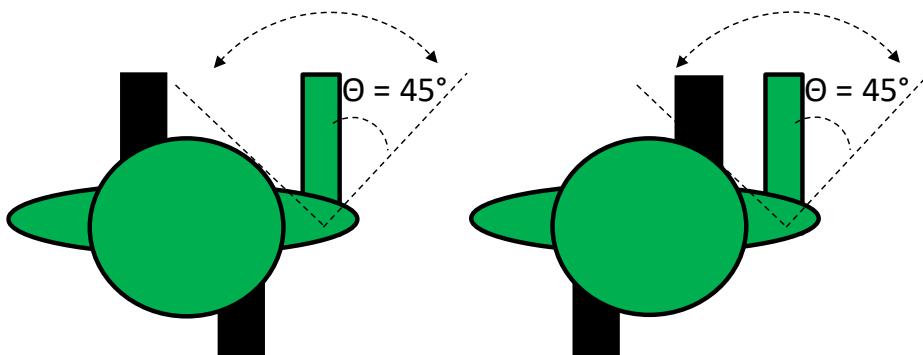


Figura 2: Jogador com o pé a esquerda a frente e o pé direito atrás (esquerda) e jogador com o pé direito à frente e pé esquerdo atrás (direita).

Andar

Os jogadores deverão se mover para frente e para trás ao pressionar respectivamente as teclas “w” e “s” para o jogador 1 e “o” e “l” para o jogador 2. Os pés dos jogadores deverão se alternar indo para frente e para trás enquanto ele se movimenta. O jogador deverá girar em torno de seu centro ao se pressionar as teclas direcionais “a” e “d” para o jogador 1 e “k” e “ç” para o jogador 2. Os direcionais “a” e “k” controlarão o giro no sentido anti-horário e as teclas “d” e “ç” controlarão o giro no sentido horário em relação a cada jogador. O programa deverá permitir combinações de teclas para andar e girar ao mesmo tempo, sendo que a componente de andar deverá ser mais influente do que a componente de girar (isto é, fazendo com que o jogador ande fazendo uma curva aberta e não fique girando em torno do próprio eixo).

Colisão

Os personagens não deverão: sair da arena, invadir obstáculos ou ocupar um mesmo espaço do outro jogador. A colisão deverá ser calculada considerando sempre o círculo da cabeça.

Atirar

Cada jogador poderá atirar, sendo um tiro disparado a cada clique do botão esquerdo do mouse, para o jogador 1 ou ao clicar na tecla “5” para o jogador 2. O braço do jogador será controlado pelo movimento horizontal do mouse para o jogador 1 e com as teclas “4” e “6” para o jogador 2. Ao mover para esquerda ou pressionar “4”, o braço girará no sentido anti-horário, sendo análogo quando mover para direita ou pressionar “6”. O movimento de giro do braço deverá obedecer aos limites definidos na Figura 2. A sensibilidade para ir de um limite ao outro pode ser ajustada para um movimento do mouse com a mão. Isto é, calibre o movimento para evitar ter que mover o mouse várias vezes para baixo até chegar ao limite ou não conseguir parar com o braço nas posições intermediárias. O tiro deverá sumir ao atingir um obstáculo ou sair da arena e deverá tirar uma vida ao atingir um personagem. A velocidade do tiro deverá ser duas vezes a do jogador.

Fim de jogo

O jogo acaba quando um jogador é atingido 3 vezes (neste caso, sumindo e dando a vitória ao outro). Uma mensagem deverá ser exibida no centro da tela indicando qual jogador ganhou ou perdeu. Ao teclar “r” o jogo deve recomeçar. Um placar no topo da janela deve mostrar o número de vidas de cada jogador e atualizar sempre que alguém for atingido.

Geral

O aluno deverá tratar os movimentos para funcionar de forma equivalente em máquinas diferentes (ou seja, tratar o tempo decorrido entre chamadas *idle*), conforme visto em sala. Ele deverá também 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 do trabalho seguinte.



Centro Tecnológico
Departamento de Informática

Disciplina: Computação Gráfica	Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos	

3 Regras Gerais

O trabalho deverá ser feito individualmente. Trabalhos identificados como fraudulentos serão punidos com nota zero e os envolvidos poderão ter que responder a instâncias superiores da universidade. Casos típicos de fraude incluem, mas não se restringem à cópia de trabalhos, ou parte deles, assim como trabalhos feitos, no todo ou em partes, por terceiros (sendo estes pessoas ou sistemas de inteligência artificial).

3.1 Entrega do Trabalho

O código deverá ser entregue pelo Google Classroom dentro do prazo definido. Trabalhos entregues após a data estabelecida não serão avaliados.

A entrega do trabalho deverá seguir estritamente as regras a seguir. O não cumprimento **inviabilizará a correção do trabalho**.

- 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 arquivo zip 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 da arena 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.

3.2 Apresentação do Trabalho

O trabalho terá 15 minutos para ser apresentado para a turma. A apresentação será feita para a turma no laboratório. As apresentações ocorrerão no horário da aula e em uma data posterior à de entrega, conforme cronograma das aulas. Durante o tempo de apresentação, o aluno deverá mostrar e testar todas as funcionalidades requeridas do trabalho. O trabalho (arquivos) a ser utilizado na apresentação deverá ser o mesmo enviado para o professor, e será fornecido pelo professor na hora da apresentação. A ordem de apresentação será sorteada durante a aula, portanto, todos os alunos devem estar preparados para apresentar o trabalho durante o período de apresentações. Os alunos devem estar preparados para responder possíveis perguntas sobre o trabalho. Prepare-se para fazer a apresentação dentro do seu tempo. **Pontos só serão dados para funcionalidades apresentadas**, isto é, a audiência deverá ser capaz de ver e perceber o resultado produzido pela funcionalidade implementada no jogo. Cabe aos alunos, portanto, criarem atalhos (para habilitar e desabilitar funcionalidades, por exemplo, movimento do oponente) no trabalho para facilitar a apresentação das funcionalidades.



Centro Tecnológico
Departamento de Informática

Disciplina: Computação Gráfica	Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos	

3.3 Pontuação

A avaliação do trabalho seguirá a tabela abaixo, com bug sendo descontados caso a caso.

Itens	Sub-Itens	Feito	Observações	Pontos	Nota
Desenho	Arena e personagens			1,0	
Andar	Andar corretamente			2,0	
Colisão	Não sair da arena ou invadir obstáculos			1,0	
	Não invadir o oponente			1,0	
Atirar	Tiro certo			1,0	
	Movimento do braço			1,0	
Fim de Jogo	Mensagens, placar e reinício			1,0	
Movimento do oponente	Andar corretamente			1,0	
	Atirar corretamente			1,0	

4 Erratas

Qualquer alteração nas regras do trabalho será comunicada durante a aula e no portal. É de responsabilidade do aluno frequentar as aulas e manter-se atualizado.