	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF16014
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. Ele deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freeglut).

O aluno deverá implementar um jogo de ataque de barris. O jogo será composto por um jogador com uma arma para atirar nos barris que rolam em sua direção. O personagem do jogador será controlado pelo teclado e, poderá atirar nos barris com uma arma controlada pelo mouse. Cada barril terá uma quantidade de vezes que poderá ser atingido, sendo destruído quando essa quantidade extrapolar. Nesta versão, o jogo será 2D, portanto a arena será vista de cima (ver exemplo de arena a esquerda da Figura 1). Os barris iniciam em uma posição horizontal aleatória no topo da arena e rolam em linha reta para baixo. O jogador inicia no centro da base da arena e pode mover-se de um lado para o outro e para cima e para baixo sem sair da arena e sem ultrapassar o centro da arena na direção vertical (ilustrado com a linha pontilhada na imagem). Caso o barril encoste no jogador, ele morre. Cada barril poderá ter (decisão aleatória com 50% de chance) um inimigo em cima dele que poderá atirar no jogador. Caso o barril seja destruído, o inimigo em cima dele morre. Caso o tiro do inimigo acerte o jogador, o jogador morre e perde o jogo. O jogo acaba com vitória após o jogador destruir N barris. O número de barris destruídos deverá ser mostrado no topo da arena.

Barris destruídos: 0

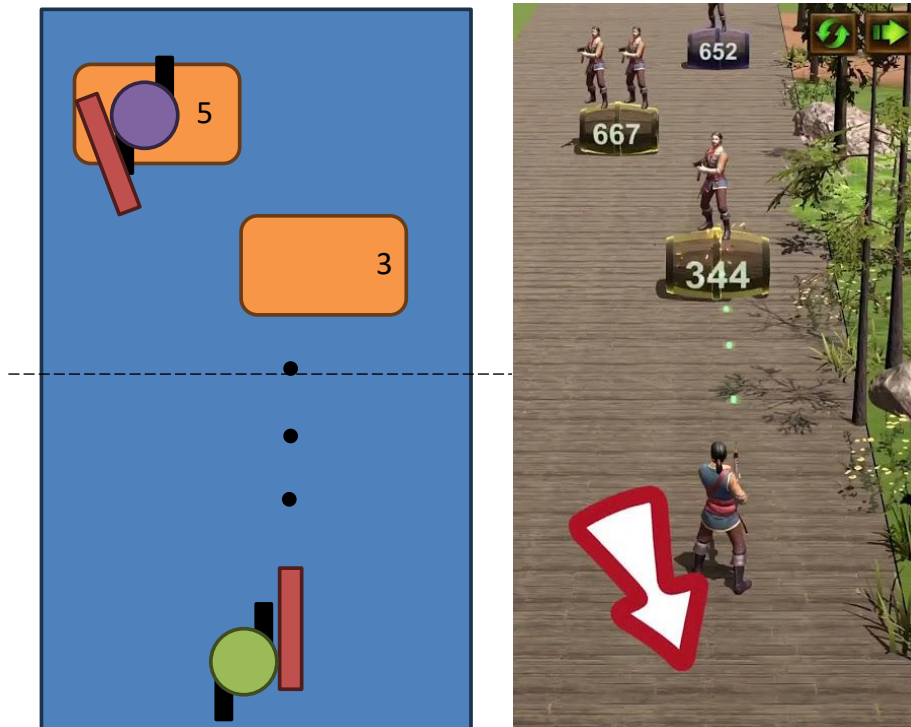



Figura 1: Na esquerda, tem-se um exemplo de arena (em azul) criada pelo jogo com o centro definido pela linha tracejada. O jogador está ilustrado com uma cabeça (em verde) os pés (em preto) e a arma (em vermelho), enquanto o inimigo tem a mesma configuração, porém com a cabeça em roxo. Os barris estão representados em laranja. A figura da direita mostra a inspiração do jogo.

	Centro Tecnológico Departamento de Informática
Disciplina: Computação Gráfica	Código: INF16014
Prof. Thiago Oliveira dos Santos	

2 Especificação Detalhada das Funcionalidades

O programa deverá ser configurável por meio de um arquivo de configurações colocado no mesmo diretório do executável. Deve-se utilizar um arquivo em formato XML (denominado `configuracoes.xml`). A leitura do XML poderá ser feita utilizando-se um parser para XML. A biblioteca Tinyxml é uma opção simples para lidar com XML e pode ser enviada juntamente com o código para ser compilada (lembre-se que não se pode demandar instalação de bibliotecas adicionais). O arquivo de configurações deverá permitir a alteração da largura e da altura da arena, do raio da cabeça dos jogadores e inimigos, da altura (diâmetro) e largura do barril, do número de vezes que um barril deve ser atingido para ser destruído, do número N (máximo de barris destruídos para ganhar o jogo), do número de tiros por segundo, da velocidade do tiro e da velocidade do barril, velocidade do jogador. Outros parâmetros podem ser usados pelo aluno para ajudar a demonstração do jogo.

Desenho

As configurações lidas do arquivo devem permitir desenhar uma arena com seus elementos com tamanhos apropriados. A arena deverá ocupar toda a janela, ou seja, o canto superior esquerdo e o canto inferior direito do retângulo da arena (visto na Figura 1) deverão respectivamente coincidir com o canto superior esquerdo e o canto inferior direito da janela do programa. A janela deverá ter a mesma dimensão (em pixels) do retângulo da arena e pode ser posicionada a gosto na tela. Não é necessário tratar o *resize*.

O raio da cabeça servirá somente para saber com qual tamanho desenhar respectivamente os personagens do jogador e dos inimigos (utilizar as cores do círculo para diferenciar os personagens ao desenhá-los). Os círculos representarão a cabeça dos jogadores e estarão sempre dentro da arena. Os outros elementos para compor os jogadores devem ser escolhidos e desenhados pelo aluno usando uma escala controlada pelo tamanho da cabeça. Por exemplo, se o personagem tiver uma cabeça de raio X e um braço de comprimento Y , um personagem com a cabeça de raio $X/2$ deverá ter um braço de comprimento $Y/2$. Os personagens serão compostos pelos elementos descritos na Figura 2. O barril deverá ser desenhado com um retângulo respeitando as dimensões dadas nas configurações e mostrando a sua resistência a tiros.

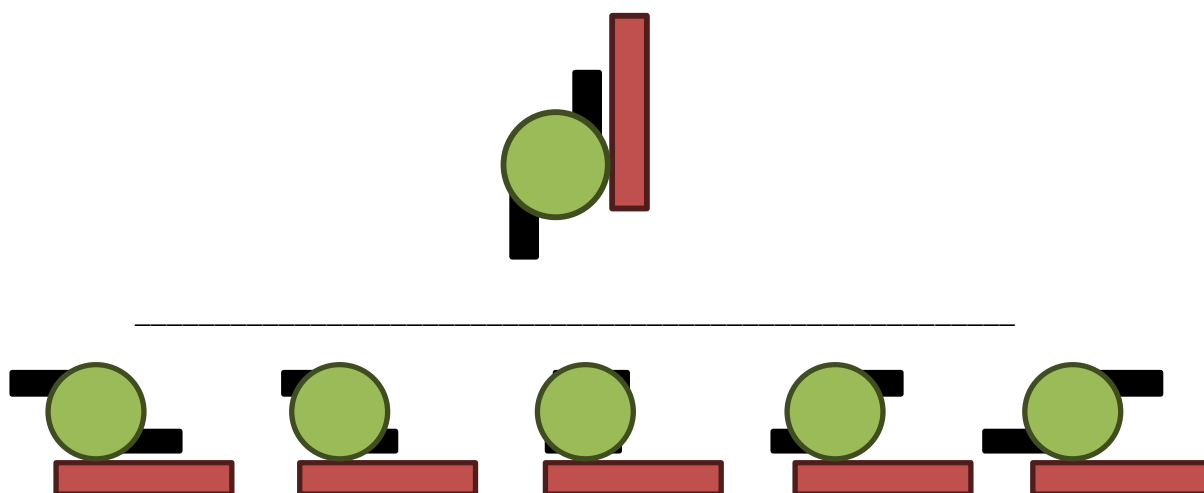
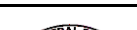


Figura 2: Topo: Ilustração do personagem composto por uma cabeça (em verde), uma arma (em vermelho) e duas pernas (em preto). O lado maior da arma indica a frente do personagem. Base: movimento do jogador andando (ilustrado da esquerda para direita, mas representado de baixo para cima)

Andar

O controle do movimento de andar será feito com teclas *a*, *s*, *d* e *w* do teclado. As teclas *a* e *d* servirão para mover o jogador para a esquerda e para a direita respectivamente. As teclas *w* e *s* servirão para

	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF16014
Prof. Thiago Oliveira dos Santos		

mover o jogador para frente e para trás respectivamente. A combinação de duas teclas, uma de movimento lateral e outra frontal, deverão causar um na diagonal. Quando o jogador mover para frente ou para trás, suas pernas deverão simular um efeito de andar (Figura 2). Esse efeito não precisa ser feito ao mover para o lado.

Colisão

O jogador não deverá sair da arena (a cabeça não deve sair). Portanto, a colisão deverá ser calculada considerando a cabeça do jogador. A colisão com o barril deverá considerar o retângulo representado cada barril e ao colidir, o jogador deve morrer finalizando o jogo.

Atirar

O jogador poderá atirar com o botão esquerdo do mouse, cada clique será um tiro que continuará sua trajetória independente do jogador. A arma do jogador deverá girar no sentido anti-horário ao mover o mouse para a esquerda e horário ao mover para a direita (ver Figura 3).



Figura 3: Ilustração do movimento de giro da arma.

Pontuar

Toda vez que um tiro acertar um barril, uma “vida” deve ser retirada dele sendo o número atualizado no barril. O jogo o jogador destruir N barris sendo este número atualizado e mostrado no topo da arena (canto superior direito). Ao finalizar o jogo, deve-se mostrar uma mensagem na tela indicando se perdeu ou ganhou.

Oponente

O oponente será uma cópia do jogador e deverá mover-se juntamente com o barril. Suas pernas deverão sempre fazer o movimento de andar. Ele deve atirar na direção do jogador com a frequência e velocidades informadas nas configurações. A arma dele deverá, sempre que possível (não extrapolando o limite de ângulo permitido), apontar para o jogador.

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.

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 trabalhos, ou parte deles, assim como trabalhos feitos por terceiros.

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.


	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF16014
Prof. Thiago Oliveira dos Santos		

A entrega do trabalho deverá seguir estritamente as regras a seguir. O não cumprimento **inviabilizará a correção do trabalho** que, por sua vez, receberá nota zero.

- 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*.

3.2 Apresentação do Trabalho

O trabalho terá 30 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: INF16014
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	Elementos			1,0	
Andar	Andar corretamente			2,0	
	Manter as partes do corpo				
Colisão	Não sair da arena			2,0	
	Morrer				
Atirar	Movimento			2,0	
	Tiro				
Pontuar	Atingir barril			2,0	
	Destruir barril				
Oponente	Anda			1,0	
	Atira				

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 se manter atualizado.