

Trabalho Prático Grau A
Algoritmos e Estruturas de Dados em C++
Robot Kombat

OBJETIVO

O objetivo desta atividade é integrar os conhecimentos avançados de programação e orientação a objetos. É uma atividade que requer dedicação e estudo, mas certamente é um desafio que irá incrementar suas habilidades de programador!

DESCRIÇÃO GERAL

Criar um jogo de ação que simula batalhas de Robôs. Para cada batalha o robô deve ser configurado de acordo com a arena e com o inimigo. O objetivo é você modelar as classes do problema com heranças e polimorfismos, salvamento e carregamento em arquivo.

TELAS

O jogo possui as seguintes telas:

- Tela Principal: na tela principal é possível escolher novo jogo, carregar jogo, exibir ranking, créditos e sair. Ao escolher a opção sair encerra o aplicativo;
- Tela de Ranking: exibe as cinco melhores pontuações. Lista o nome do jogador e a pontuação. Esta listagem é ordenada da maior para menor pontuação. Da tela de ranking retorna para tela principal;
- Tela de Créditos: exibe os créditos do programa, nomes dos desenvolvedores. Da tela de créditos retorna para tela principal;
- Tela para Carregar jogo: exibe até 3 slot para carregar um jogo salvo anteriormente. O nome dos slots é o número da batalha que foi salvo junto com dia e horário. Quando o jogo é carregado vai para tela de jogo;
- Tela de Jogo:
 - A tela principal onde o jogador verifica o seu andamento no jogo, numero de batalhas disputadas e quantas faltam para o desafio contra o campeão da Concelho Mundial de Robôs Guerreiros, próximo adversário e possibilita ao jogador escolher o que deseja fazer:
 - Inventário: tela onde é possível realizar a configuração do robô. Ela deve permitir o usuário possuir até 10 itens variados e escolher qual arma, armadura e pernas/locomção.
 - Loja: permite o jogador comprar e vender itens para melhorar seu robô.
 - Arena: onde deve ser realizada a batalha em si.
 - A qualquer momento o jogador poderá cancelar a partida e será exibida uma tela de encerramento. O jogador será questionado se deseja encerrar a partida. Caso confirme o encerramento retorna para o a Tela principal. Se não quiser encerrar retomar para a tela do jogo;
 - No término de cada batalha em caso de sucesso pede-se o nome do jogador para salvar o

ranking. O ranking deve ser salvo somente com os cinco melhores jogadores e no formato sequencial. Em caso de fracasso exibe-se uma tela de game over e permite retomar para a Tela Principal.

DETALHAMENTO

Você é um engenheiro de Robôs famoso e foi convidado para disputar o campeonato organizado pelo Concelho Mundial de Robôs Guerreiros. O campeão invicto há 10 anos e 1º do ranking é o ***Electro Robot***. Para poder desafiar o campeão o seu robô deve vencer 8 batalhas consecutivas contra outros robôs. No caso de uma derrota seu robô deve recomençar os desafios.

Os robôs podem ser modificados em 3 características:

- **Rodas/Pernas**: permitem o jogador andar mais uma distancia maior/menor por turno (deve ser implementadas no mínimo 3 possibilidades – um quadro, dois quadros ou 3 quadros por turno);
- **Armadura**: Permite ao usuário modificar sua Defesa (devem ser implementadas no mínimo 3 armaduras: como modificadores de 1 a 3);
- **Arma**: Permite modificar o ataque do personagem. Deve ser implementado no mínimo 3 tipos de armas (uma que modifica o poder de ataque em +2 e possui o alcance de 3 quadros, uma que permite ao usuário dar 3 disparos na mesma rodada com alcance de 2 quadros e a terceira permite ao usuário disparar 2 vezes com o modificador + 2 a uma distancia de quatro quadros);

Cada robô tem um poder de ataque (PA), uma defesa (DEF) e uma de capacidade de dano (DAM). Quando são criados estes valores são inicializados em 1. Além desses, cada robô tem 30 pontos de vida (HP). A mecânica da simulação de combate é a seguinte: é criado o robô do jogador com as estatísticas-padrão para combater. No inicio do jogo deve ser criada uma estrutura que armazene os 9 robôs que o jogador deve vencer, esta estrutura deve pegar os robôs de um arquivo sequencial que contem um numero mínimo de 20 robôs pré-definidos e selecionar 8 e colocar junto com ultimo adversário que é o *Electro Robot*.

As batalhas encerram quando os pontos de vida de um dos robôs chegar a zero. A batalha é dividida em turnos alternando os ataques dos robôs (para cada turno é possível movimentar o numero de quadros permitidos pelas Rodas/Pernas) e realizar o ataque. Cada golpe tem um nível de ataque máximo (PA) determinado aleatoriamente (valor de 1 à PA). O adversário pode fazer uma jogada aleatória de defesa (valor de 1 à DEF). Caso PA seja maior que DEF o adversário foi atingido e deve-se calcular o dano (valor de 1 à DAM) que será subtraído dos seus pontos de vida. Em outro caso o adversário consegue defender. Quando um novo item é adicionado ao robô, aplica-se os bônus, sejam estes positivos ou negativos na estatística correspondente, por exemplo, se uma melhoria dá +1 de bônus de ataque, soma-se um no bônus de ataque do robô. Em cada de bônus negativo nenhuma estatística poderá ficar com valor menor do que zero. A qualquer momento a simulação poderá ser encerrada e um arquivo binário com o estado do jogo deve ser salvo com o nome “save.bin”. Deverá armazenar todos os atributos dos dois robôs, além do número de vitórias, número de partidas. O usuário poderá reiniciar a simulação usando este arquivo.

O robô derrotado perde um dos seus itens que pode ser pego pelo vencedor. Sendo assim deve sempre ser feito um sorteio nos itens do inventário para selecionar qual item o robô ira perder.

Os Robôs devem ser armazenados da seguinte maneira em arquivo:

<Nome> < arma> <bônus de ataque> <alcance > <numero de disparos> < rodas/pernas> <numero de quadros> <armadura> <bônus de defesa>

Exemplo :

Electro Robot Master_Shock 4 3 2 Extra_Speed 3 Electric_Armor 2

REQUISITOS

- Deve ser implementado em C++;
- Não podem ser usadas bibliotecas de uso exclusivo do Windows (p.e. glaux/windows);
 - Usar a libEGL ou SFML como biblioteca gráfica;
 - Uma solução orientada a objetos e organizada modularmente em diversos arquivos;
 - Uso intensivo de ponteiros!

CRITÉRIOS DE AVALIAÇÃO

- **Deve ser funcional! Programa que não executa não é programa. Deve ser compilável e executável. Aplicativos que não executarem serão desconsiderados e será atribuída nota zero para os aprendizes.**
- Você poderá adaptar a temática de robôs para outro tema observando as seguintes condições:
 - (a) deve manter a mesma estrutura do problema; e
 - (b) não pode ser um tema pornográfico, racista ou qualquer tipo de discriminação ou tema ilegal.
- Projeto adequado orientado a objetos com herança e polimorfismo;
- Código estruturado, indentado e comentado. Não esqueça de colocar seu nome no código;
- Respeitar **TODOS** os requisitos descritos anteriormente.
- Fidelidade as regras do jogo e descrição da aplicação. É para ser implementado exatamente o que está sendo pedido. Não tente simplificar o que está sendo solicitado.

MUITO IMPORTANTE!

Trabalhos copiados da Internet ou dos colegas terão nota zero.

INSTRUÇÕES PARA ENTREGA

Entregar via moodle todo código-fonte incluindo arquivos de imagens, sons, *.h, *.cpp, *.txt até as 19:30 do 10/10/2013, além do executável e manual de instruções. A entrega também poderá ser feita no início da aula por pen drive. No dia da entrega TODOS irão apresentar seus trabalhos para o educador. Um membro da dupla será sorteado para apresentar. O outro não poderá ajudar. Quem não estiver presente no dia da apresentação irá receber nota zero. Trabalhos entregues após as 19:30 não serão mais aceitos.

Data de Entrega: 10/10/2013