



Trabalho Prático 1º bimestre

Ultimate Fighting Championship

Valor: 15 pontos

Data de entrega: 31/03/2019

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos da Orientação a Objetos e com a programação utilizando a linguagem Java. Para realização do trabalho o aluno deverá implementar um jogo descrito a seguir. Os princípios da Orientação a Objetos devem ser utilizados.

1. Regras do jogo

1. O jogo é composto de um lutador (jogador) que deve lutar em uma liga com outros lutadores (máquina). A cada luta caso o jogador vença ele ganha uma premiação que permite a compra de novas habilidades.
2. O objetivo do jogo é ganhar de todos os lutadores da liga.
3. Os jogadores da liga devem ser previamente selecionados;
4. As novas habilidades adquiridas são incorporadas às possibilidade de ataque (poder) do lutador.
5. A liga de lutadores deve possuir todos os jogadores em uma sequência a ser jogada;
6. Cada luta tem duas lista de ações, que corresponde as ações dos dois jogadores no jogo. Esta lista deve ser armazenada em memória em um vetor de 1000 posições, sendo que cada posição corresponde uma ação, ou seja tem um objeto do tipo Ação. O código que gera as ações está anexado.
7. É a lista quem determina qual jogador ganha a luta, sendo que cada luta tem duas novas listas (uma para o jogador e a outra para a máquina).
8. A primeira posição da lista do jogador dever ser comparada à primeira posição da lista da máquina segundo a regra descrita no item 6, assim, o jogador deve ganhar, manter ou perder pontos, e perder ou manter sangue.
Da mesma forma, a segunda posição das listas devem ser comparadas seguindo a mesma regra, e assim por diante.
O fim das comparações (fim da luta) deve terminar se o sangue de algum jogador terminar ou se a lista de ações terminar as posições.

9. A regra de combate está descrita abaixo. O objeto ação deve estar setado com alguma dessas possibilidades ou é um ataque ou uma defesa. Caso seja uma ataque, o ataque poderá ser soco, chute ou poder. O 1 representa a ação. Na primeira linha por exemplo, foi um ataque sendo este ataque um soco.

Ataque	Soco	Chute	Poder	Defesa
1	1	0	0	0
0	0	0	0	1
1	0	0	1	0

1. Soco e defesa: nenhum jogador perde ponto
 2. Chute e defesa: nenhum jogador perde ponto
 3. Defesa e defesa: nenhum jogador perde ponto
 4. Soco e soco: ganha o jogador com mais habilidade (modalidades de lutas)
 5. Chute e chute: ganha o jogador com mais habilidade (modalidades de lutas)
 6. Poder contra qualquer outro golpe: ganha o jogador que deu o poder
10. Para cada posição da lista de ações, o jogador que ganhar, adquiri pontos (você escolhe a quantidade de pontos), o que perder, diminui uma porcentagem de sangue (você escolhe a porcentagem).
11. As modalidades de lutas que o jogador podem adquirir seguem abaixo. Cada modalidade de luta deve ser representada por uma classe. Quando um jogador tiver uma ação “poder” ele poderá escolher uma das modalidades que ele possuir, para o poder da máquina a modalidade deve ser aleatória.
- a) **Boxe**: todo jogador inicia com essa modalidade
 - b) **Jiu-jitsu**: ganha essa habilidade ao atingir 1000 pontos
 - c) **Muay thai**: ganha essa habilidade ao atingir 2000 pontos
 - d) **Wrestling**: ganha essa habilidade ao atingir 3000 pontos
 - e) **Karatê**: ganha essa habilidade ao atingir 4000 pontos
12. Você poderá inferir alguma especificação que não esteja detalhada neste documento, mas para isso você deverá informar na documentação do jogo.
13. Jogos que não sigam a especificação acima não serão avaliados, então, faça seu jogo, não se preocupe em procurar um igual.

2. Elementos da orientação a objetos que devem ser utilizados

1. Classes objetos
2. Encapsulamento
3. Herança e composição
4. Interfaces
5. Polimorfismo

Documentação

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado.
4. Testes: descrição dos testes realizados e listagem da saída (não edite os resultados). Você pode propor outros testes além dos fornecidos com o enunciado.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

O que deve ser entregue:

Envie um arquivo ZIP com o nome no formato 'tp1-primeironome1.zip', contendo os seguintes arquivos:

- Todo o projeto utilizado no Eclipse (apenas Eclipse);
- A documentação do trabalho.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, indentação e comentários no programa também serão avaliados.
- O trabalho deverá ser feito individualmente.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.