

Estruturas de Dados I

Trabalho 1

Prof. Bruno Azevedo

Instituto Federal de São Paulo



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Catanduva

Avaliações

- Prova 1 (P1) + Trabalho 1 (T1).
- Prova 2 (P2) + Trabalho 2 (T2).
- Prova: 7 pontos, Trabalho: 3 pontos.
- Nota final = $\frac{P1 + T1 + P2 + T2}{2}$
- Quem não obteve a nota mínima para passar na disciplina poderá fazer o Instrumento Final de Avaliação (IFA) se: $4,0 \leq \text{Nota final} < 6,0$.
- Data a ser definida.

Avaliações – Datas

- Podem mudar futuramente. Se for necessário, vocês serão avisados com antecedência.
- P1: 25/04 | T1: 25/04 até as 23:59.
- P2: 20/06 | T2: 20/06 até as 23:59.

Trabalho 1

- Crie um jogo de RPG em C/C++ de acordo com as seguintes instruções:
- O RPG simulará um combate em uma arena entre personagens. Terá (no mínimo) 6 arquétipos distintos. Por exemplo, **Guerreiro(a)**, **Bruxo(a)**, **Ladrão**, **Arqueiro(a)**, etc.
- O **Personagem** deve ter no mínimo os seguintes atributos:
 - Nome do personagem.
 - Pontos de vida (VID).
 - Pontos de ataque (ATQ).
 - Pontos de defesa (DEF).
 - Lista de personagens derrotados.
- Cada arquétipo terá um **ataque especial**, que será exclusivo. Este ataque poderá ocorrer aleatoriamente durante um *round* de combate. Ou seja, em vez do ataque normal, ocorrerá o ataque especial. Este ataque terá um nome específico e causará mais dano do que um ataque normal.
- O ataque especial também poderá causar **sangramento**, onde o outro personagem sangrará por um número de *rounds* calculado aleatoriamente (mínimo de 2 rounds, máximo de 6 rounds). O sangramento causará uma perda (pequena) de vida a cada round, independente do dano recebido no round.

O sangramento deve ser calculado no início de cada *round*, portanto, um personagem pode ser derrotado sem receber dano de ataque no *round*.

Caso ambos os personagens fiquem sem vida devido a sangrarem no início do *round*, um dos dois personagens será sorteado para avançar e o outro será considerado derrotado.
- Implemente uma função para inicializar o nome, VID, ATQ e DEF do personagem. Implemente uma função para exibir as estatísticas do personagem.
- Cada personagem irá armazenar a lista de personagens que derrotou. Use um vetor de ponteiro para estrutura, de modo a armazenar cada personagem que foi derrotado. Ou seja, os personagens derrotados serão acessados diretamente, não será uma cópia.
- Cada combate poderá durar no máximo 10 rounds. Se nenhum dos dois personagens forem derrotados ao fim de 10 rounds, o personagem que causou mais dano será considerado o vencedor.
- Ao fim de cada combate seu jogo deve exibir uma tabela com as estatísticas do combate, considerando dano efetuado por cada personagem, vida restante de cada personagem, número de rounds ocorridos, e quantos ataques especiais ocorreram.

Trabalho T1

- Use estruturas para armazenar os personagens. Implemente uma função que permita criar um número específico de personagens. Implemente uma função para listar todos os personagens no jogo. Implemente uma função que permita aos personagens lutarem entre si.
- As estruturas representando os personagens devem ser **alocados dinamicamente**. Efetue a liberação de memória corretamente, quando necessário.
- A luta ocorre entre dois personagens, reduzindo os pontos de vida do alvo considerando o ATQ de quem ataca, a DEF de quem defende, e um número sorteado (RND) que influenciará neste cálculo.

Cada personagem ataca por vez, e será sorteado quem será o primeiro a atacar.

Cada personagem atacará uma vez por *round*. Ou seja, ambos os personagens atacam em um *round*, a não ser que um deles seja derrotado.

Se um personagem ficar com VID zero ou menos, ele é derrotado e removido do jogo. Exiba uma mensagem informando o resultado da luta.

Você deve criar pares de combatentes, e os combates devem ocorrer até que exista apenas um personagem sobrando. Crie uma organização estilo torneio.

A vida do personagem vencedor recuperará ao fim de cada combate.

- Implemente uma função **main()** que executará o seu jogo.
- Esta deve criar no mínimo 16 personagens de diferentes arquétipos, que irão se enfrentar no torneio.
- Assim que sobrar apenas um vencedor, o jogo deve exibir as seguintes informações referentes ao personagem campeão do torneio: VID, ATQ, DEF, arquétipos dos personagens que derrotou no torneio (**acessando o personagem derrotado e não apenas o seu nome**), dano total causado, dano total sofrido, maior dano causado em um round, maior dano recebido em um round.

Trabalho T1

- Trabalho em grupos de até 3 integrantes.
- Seu código obrigatoriamente deve ser feito em C/C++ e utilizando o conhecimento obtido até o momento.
- O trabalho será avaliado com o cumprimento da lista destes requisitos mínimos, mas alunos que adicionarem elementos extras (que o professor avaliar relevantes) podem obter pontuação extra. Entretanto, a pontuação nunca passará de 3 pontos.
- Entreguem um documento junto com o código do trabalho, detalhando tudo que foi implementado em seu jogo.
- Data de entrega: deverá ser entregue na mesma data da prova até as 23:59. Trabalhos entregues até o dia seguinte poderão obter no máximo 66% da nota total do trabalho. Trabalhos entregues após este dia não serão considerados (zero).