

Trabalho Final

Valor: 6.0 pontos

Equipe: máximo de 5 componentes

Descrição:

Um simulador de jogo de RPG simples: a ideia é criar um jogo básico onde o jogador controla um personagem com atributos como força e velocidade. Ele poderá explorar um mapa simples, lutar contra inimigos e coletar itens.

Seu programa deve permitir ao usuário definir um personagem, o qual iniciará sua jornada contra X inimigos que estarão posicionados aleatoriamente no mapa (uma matriz), cujo tamanho será definido pelo usuário. Seu personagem caminhará por este mapa e poderá encontrar até X poderes/ itens que aumentarão suas habilidades (os poderes também serão alocados de forma aleatória no mapa). Usem a função randômica do C para definir as posições destes elementos.

Uma posição não pode alocar dois objetos ao mesmo tempo.

Para calcular o valor de X, use a seguinte fórmula: $X = (\text{ladoDaMatrix} * \text{alturaDaMatrix})/4$

Funcionalidades:

- **Criação do Personagem:**
 - Utilizar uma struct para armazenar os atributos do personagem.
 - Permitir que o jogador escolha o nome do personagem.
 - Atribuir valor inicial da força.
 - Valor inicial da velocidade = 1
- **Exploração do Mapa:**
 - Representar o mapa como um vetor de vetores.
 - Utilizar estruturas de repetição para percorrer o mapa e exibir a posição do jogador.
 - Crie uma função que nos permita visualizar as posições dos poderes e dos inimigos
 - Implementar a movimentação do jogador pelas diferentes direções (cima, baixo, esquerda, direita).
 - O atributo velocidade do jogador vai definir quantas casas o personagem anda. Se a velocidade for 2, o personagem deve andar de 2 em 2 casas.
- **Combate:**
 - Quando o jogador encontra um inimigo, iniciar uma sequência de combate.
 - Considera-se o encontro de um inimigo, quando o personagem alcança quaisquer uma das casas do inimigo.
 - Calcular o dano causado com base nos atributos do jogador e do inimigo.
 - Quem tiver com menor força perde. Quem tiver maior força, ganha o total de força do perdedor.
- **Status do jogo**
 - Deve ser possível, a qualquer momento do jogo, salvar o status do jogo para que o jogador possa continuar quando desejar.

- Deve ser dada a opção de iniciar novo jogo ou retomar de onde parou quando ele retornar o usuário programa

Avaliação:

- A equipe deverá exibir todas as funcionalidades de seu programa em um vídeo que deve ser carregado no youtube, seguindo a seguinte sequência:
 - mostrar o mapa vazio;
 - mostrar a criação de um personagem;
 - mostrar o mapa com o personagem, inimigos e poderes
 - caminhar com o personagem pelo mapa
 - mostrar dados do personagem
 - encontrar uma força
 - mostrar os dados do personagem
 - encontrar um inimigo mais fraco
 - mostrar os dados do personagem pós vitória
 - encontrar uma velocidade
 - mostrar os dados do personagem
 - mostrar mapa
 - salvar status do jogo
 - fechar o programa
 - abrir o programa
 - retornar o jogo de onde terminou
 - encontrar um inimigo mais forte
 - mostrar dados do personagem pós vitória
- Um arquivo zip deve ser postado no AVA até o dia **12/09** contendo:
 - Nomes dos membros da equipe;
 - Link para acesso ao vídeo demonstrativo;
 - O código fonte do programa em .C
- No dia 08/04, os estudantes farão uma alteração no código, a ser revelada no mesmo dia pelo docente. A alteração será realizada individualmente, utilizando os computadores disponíveis no laboratório de aulas.
- Serão avaliados:
 - atendimento aos requisitos
 - qualidade do código (legibilidade, manutenibilidade, clareza etc)
 - uso das estruturas apresentadas nas aulas de forma adequada
 - dentre outros elementos.

OBS: Quaisquer intercorrências e dificuldades na execução do programa será de inteira responsabilidade da equipe.

OBS2: o uso de inteligência artificial para geração, adequação ou melhoria de código será considerado plágio e a nota zero será atribuída a todos os componentes da equipe.

	Barema	
Item		Pontuação
Função	0,2	
Parâmetro por valor	0,2	
Parâmetro por referência	0,2	
Struct	0,5	
Ponteiro	0,5	
Modularização	0,5	
Alocação dinâmica / gerenciamento de memória	0,9	
Arquivo	0,5	
Atendimento aos requisitos	0,5	
Qualidade do código (comentários, nomes significativos, organização, estruturação)	0,5	
Alteração do código	1,5	