

Trabalho 2.1

Acadêmico (a): _____ Data: 25/04/2022 Conceito: _____

Protótipo de RPG – Ponteiros e Alocação Dinâmica

Continuando as ideias do trabalho passado, vamos fazer agora um mapa para o nosso RPG e pensar em outras funções, nele vamos precisar de ponteiros e alocação dinâmica, mas não vamos abandonar os conceitos abordados anteriormente. Para o projeto de hoje temos alguns elementos obrigatórios:

Vamos trabalhar com um mapa 2D de blocos, que representará o cenário do nosso mundo:

Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco
Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco
Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco
Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco
Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco
Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco	Bloco

Esse mapa deve ser dinâmico, armazenado em uma matriz $A \times L$ de variáveis do tipo bloco, que vão conter algumas informações sobre o jogo, baseados em uma struct. Também deverá ter um pequeno gameplay. As funcionalidades mínimas estão descritas a seguir.

(Podem utilizar o trabalho anterior como base para esse, ou o projeto disponibilizado no material didático)

Structs

O trabalho deverá conter pelo menos as seguintes **structs**:

-
- Arma;
 - Jogador: Essa struct deve conter vida, e as posições x e y no mapa e uma arma;
 - Inimigo: Essa struct deve conter um nome e vida e uma arma;
 - Bloco: Essa struct representa um bloco no mapa, deve conter a informação se é bloqueado ou não (caminho ou pedra), se tem um inimigo ou não, e um **ponteiro** para uma variável do tipo Inimigo (que pode ficar vazio se não tiver nenhum inimigo no bloco)
 - Mapa: Essa struct tem que ter uma altura **A**, uma largura **L**, e uma matriz de variáveis do tipo Bloco **AxL** (essa matriz deve ser alocada dinamicamente, os valores **não** devem vir de template)
 - Fase: Deve conter um nome, uma variável do tipo Mapa, um número de inimigos **N** e um vetor de tamanho **N** de variáveis do tipo Inimigo (também alocado dinamicamente)

Lógica

O trabalho deverá conter pelo menos as seguintes funções:

- **Mapa CriarMapa(int altura, int largura)**: Essa função deve receber a altura e largura do mapa e deve retornar uma nova variável desse tipo alocada dinamicamente com o tamanho passado por parâmetro (essa alocação deve ser pelo método utilizando memória contínua). Cada bloco desse mapa deve ter 20% de chance de ser bloqueado (pedra) e 80% de não ser bloqueado (caminho) e começar sem inimigos.
 - **Fase CriarFase(int numInimigos, Inimigos* inimigos, int alturaMapa, int larguraMapa)**: Essa função deve receber por parâmetro o número de inimigos e um vetor de inimigos e também a altura e largura do mapa. Ela deve chamar a função CriarMapa para gerar o mapa. Depois ela deve, para cada inimigo, adicionar esse inimigo como membro de um bloco aleatório do mapa, desde que ele não esteja bloqueado e ainda não tenha nenhum inimigo. No final todos os inimigos devem estar alocados em posições no mapa, então se a posição sorteada for inválida, deve ser sorteada uma nova.
 - **void Movimentar(...)**: Essa função deve movimentar o jogador pelo mapa (o usuário deve utilizar o teclado para isso), ele não deve conseguir andar para fora do mapa ou para uma posição que seja bloqueada. Se ele andar
-

para uma posição com um inimigo, deve ser iniciado um combate automático (utilizar função do trabalho anterior para um inimigo). O jogo acaba se o jogador ou o último inimigo morrer.

Segue abaixo um exemplo para um mapa com 5 de altura e 10 de largura e 5 inimigos (lembrando que as posições são aleatórias).

[illegible]