

Trabalho 2.1

Acadêmico (a): _____

Data: 26/09/2022 Conceito: _____

Protótipo de RPG – Ponteiros e Alocação Dinâmica

Esse trabalho vai se basear em um código inicialmente feito, onde existem algumas funções e structs previamente feitas e faz um “jogo” que executa sozinho, se vocês executarem o código a saída vai ser mais ou menos assim:

```
Entrando em Green Hill Zone
0 jogador atacou goblin e ele ficou com -6 de vida
0 goblin atacou ao mesmo tempo e o jogador ficou com 88 de vida
goblin foi morto!

0 jogador atacou outro goblin e ele ficou com -8 de vida
0 outro goblin atacou ao mesmo tempo e o jogador ficou com 78 de vida
outro goblin foi morto!

0 jogador atacou gobluis e ele ficou com -5 de vida
0 gobluis atacou ao mesmo tempo e o jogador ficou com 66 de vida
gobluis foi morto!

0 jogador atacou goberto e ele ficou com -6 de vida
0 goberto atacou ao mesmo tempo e o jogador ficou com 55 de vida
goberto foi morto!

0 jogador atacou goblin chefe e ele ficou com 31 de vida
0 goblin chefe atacou ao mesmo tempo e o jogador ficou com 43 de vida
0 jogador atacou goblin chefe e ele ficou com 20 de vida
0 goblin chefe atacou ao mesmo tempo e o jogador ficou com 35 de vida
0 jogador atacou goblin chefe e ele ficou com 10 de vida
0 goblin chefe atacou ao mesmo tempo e o jogador ficou com 22 de vida
0 jogador atacou goblin chefe e ele ficou com -6 de vida
0 goblin chefe atacou ao mesmo tempo e o jogador ficou com 14 de vida
goblin chefe foi morto!
```

O código simula um rpg onde o jogador encontra 5 inimigos em sequência e os enfrenta em combate, ele vence se levar a vida de todos os inimigos para 0 e perde se sua própria vida chegar a zero. A parte chata é que não tem gameplay, neste trabalho vamos colocar uma pequena interação do usuário com o jogo, e para isso vamos começar criando um mapa:

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.

(O código disponibilizado deve ser utilizado como base, mas podem ser adicionadas funcionalidades e variáveis caso achem necessário, por exemplo, alguma forma de curar vida ou armas diferentes)

Structs

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

- As structs do código disponibilizado, alterando Jogador para armazenar a sua posição no mapa
 - 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 $A \times L$ (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 código disponibilizado 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).

Inimigos	Goblinson	Goblerto	Gobo	Goblínio	Pedro
----------	-----------	----------	------	----------	-------

Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não *Goblerto	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não
Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não *Pedro	Pedra? Não Inimigo? Não
Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não
Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Sim *Goblinson	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Sim *Goblínio	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Sim *Gobo	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não
Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não
Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Sim Inimigo? Não	Pedra? Não Inimigo? Não	Pedra? Não Inimigo? Não