

Ao importar verificar :

* Nenhum campo deve ser nulo
* Se o cod missao é nulo lança excecao
* Se o cod missao já existe, adicionar o cenário a essa missao senão criar uma nova
* Se a versão já existe lança excecao
* Se existe pelo menos duas divisão senão lança excecao
* Se existe pelo menos um inimigo senão lanca excecao
* Se existe pelo menos uma ligação senão lança excecao
* Nas ligações a divisão tem de existir no edifício!!O grafo já faz isso
* Se o inimigo já existe no mesmo cenário lança excecao
* Divisao associada à entrada tem de existir nas divisões do edifício senão lança excecao
* O poder do inimigo tem de ser maior que 0.

Json Import dividir por funções

WeighteAdjMatrixGraph a ser utilizado para o edifico. Utilizar lista ordenada por pontos de vida para a lista de cenários, dentro da classe missao.

Importar missões de diferentes versões

Classe Missoes:

relatorioMissao(String codMissao)

Classe Cenario:

simulacaoManual()

* Pedir a entrada ao tó cruz. Verificar se a entrada existe!
* Iterativamente pedir a divisão a entrar. Verificar se existe a divisão e se é um vértice adjacente. Retirar o poder do inimigo se existir.
* Missao com sucesso se passar pelo alvo, sair do edifício por alguma entrada e a vida>0. Ir registando o caminho, se a vida <=0. Sucesso=false.

simulacaoAutomatica()

* Encontrar o caminho que permita sair com o maior numero de vida possível.
* Considerar todas as entradas e saídas. Utilizar minHeap para encontrar o shortest path menor dano.Todos os Shortest path entre entrada-Alvo-entrada possíveis e adicionar à minheap. Quando terminar removemos o mínimo e temos o caminho mais curto. Se o caminho=0, sucesso = true senão sucesso = false;

--------------------------------------------**???Visualizar mapa na consola???**------------------------------------