

Projeto

Nada melhor que um bom projeto para fixar e colocar à prova o que estamos aprendendo, e nada melhor que unir o útil ao agradável na hora de escolher um projeto.

Neste semestre, o projeto que desenvolverão é um clone de um clássico dos jogos de computadores, **Bomberman**! Bom, não exatamente um clone, mas os rudimentos de um clone. Ainda assim, o projeto não é trivial e por isso vamos dividi-lo em etapas para gerenciar a complexidade. De forma geral, podemos dividir o projeto em duas etapas:

- Etapa 1
 - Estruturas de dados para representação dos elementos do jogo usando **listas**
 - Uso de tipos primitivos para representação do estado e dos elementos do jogo.
 - Uso de `type`
 - Funções de manipulação das estruturas
- Etapa 2
 - Uso de tipos definidos pelo usuário para representação do estado e dos elementos do jogo.
 - Uso de data
 - Tipos algébricos
 - Funções de manipulação das estruturas
 - Visualização do tabuleiro
 - Movimentação de (pelo menos) um bomberman pelo tabuleiro usando o teclado

Etapa 1

O jogo bomberman acontece em um "tabuleiro" onde os vários jogadores, bombermen de diversas cores, tentam explodir os seus competidores até serem os únicos restantes no jogo.



No processo, os competidores se movimentam pelo tabuleiro explodindo barreiras e coletando presentes que lhes conferem habilidades, como aumentar a quantidade de bombas que cada um pode colocar no tabuleiro concorrentemente, aumentar a velocidade de deslocamento, arremessar bombas em posições adjacentes no mapa, ou aumentar o alcance das explosões. Uma explosão segue em linha reta nas quatro direções (norte, sul, leste e oeste) até que encontre algum obstáculo, que pode ou não ser destruído pela explosão.

Tabuleiro

Um tabuleiro de bomberman pode ser visto como uma matriz em que cada célula é uma pilha de elementos. Para representar a matriz do tabuleiro, usaremos uma tupla de tuplas. Assim, defina as seguintes estruturas de dados.

- Tabuleiro é uma tupla com 8 Linha.
- Linha é uma tupla com 8 Célula.
- Célula é uma pilha com 4 Item

Célula e Items

Cada posição do tabuleiro é uma célula e contém uma pilha de itens. Use uma lista para representar a pilha. Em nosso protótipo, os seguintes itens podem estar presentes na célula:

- grama
- presente_patins
- presente_arremesso
- bomba

- jogador_X

Algumas regras devem ser respeitadas pela pilha, onde "sobre" quer dizer imediatamente subsequente acima na pilha.

- uma pilha vazia é um buraco no tabuleiro
- grama só pode estar na base da pilha
- presente só pode estar sobre grama
- parede só pode estar na base da pilha, sobre grama ou sobre presente
- pedra só pode estar na base da pilha
- bomba só pode estar sobre grama
- jogador só pode estar sobre grama

Jogador

Para cada jogador, você precisa manter algumas informações extra como:

- identificador (o X que aparece no Item jogador_X)
- localização - é uma tupla com coordenadas X e Y do tipo Int que representam a linha e coluna em o item jogador_X correspondente está.
- direção - é um caractere que indica para onde o jogador está olhando.
 - 'N', 'S', 'L' e 'O'
- capacidades - é uma tupla com 3 elementos com um dos seguintes valores, onde a, b e c são Int
 - (Patins,a)
 - (Bomba,b)
 - (Arremesso,c)

Funções

O jogo só tem graça se tiver alguma coisa acontecendo nele. Nesta primeira etapa, você desenvolverá as funções que permitirão criar e manipular os elementos do jogo, de acordo com algumas regras, para as seguintes funcionalidades:

- Criação de um tabuleiro
 - Função receba uma lista de listas de items e constrói um tabuleiro válido.
- Movimentação de um jogador em qualquer dos sentidos
 - Função que receba um tabuleiro e uma instrução de movimentação de um jogador e retorne um novo tabuleiro, com o jogador na nova posição.

- Jogador só pode se deslocar para célula adjacente que não tenha pedra ou bomba
- Pode ser impossível ao jogador se deslocar
- Ao se deslocar para uma célula vazia, cai no buraco
- Ao se deslocar para uma célula com um presente, o coleta
- Coleta de presente
 - Ao coletar um presente, incrementa a posição correspondente ao presente nas suas capacidades
- Arremesso
 - Se estiver adjacente, olhando para uma bomba, e tiver a capacidade "arremesso", o jogador arremessa a bomba uma distância proporcional à capacidade.
 - Teste de direção
- Explosão
 - Direção
 - Capacidade
 - Eliminação de presentes
 - Eliminação de pedra
 - Eliminação de jogador
- Detecção de fim de jogo

Vocês são livres para preencher as lacunas da especificação de forma razoável. Na dúvida, me pergunte.