MAC0216 - Fase 3

Gabriel Kazuyuki Isomura
N° USP: 9793673

Guilherme Costa Vieira
N° USP: 9790930

Victor Chiaradia Gramuglia Araujo
N° USP: 9793756

20 de novembro de 2017

1 Introdução

Nesta fase do EP foi introduzido um controlador gráfico para exibir o jogo em modo janela. A adição do controlador gráfico facilitou a detecção de bugs das fases anteriores que foram devidamente corrigidos.

2 Correção de bugs

2.1 Células vizinhas

A função numToPos() do arquivo arena.c não calculava as células vizinhas corretamente para algumas posições. O cálculo anterior não levava em consideração a coluna em que a célula se encontra para calcular os vizinhos, isto é errado uma vez que existem dois cálculos diferentes de vizinhos conforme a coluna em que a célula se encontra.

2.2 Robôs clones

A struct de cada célula da matriz hexagonal possui os campos robô e armyID, que guardam o robô que ocupa a posição e a identificação do exército desse robô, respectivamente. Porém, na chamada de sistema MOV, quando os robôs eram movidos para uma nova posição, os campos citados acima da posição antiga não eram resetados, logo os robôs eram clonados toda vez que se moviam.

3 Adição do arquivo APRES

A implementação original do arquivo apres desenha hexágonos no leiaute horizontal odd-r (pointy topped), porém nas fases anteriores foram considerados hexágonos com leiaute vertical odd r (flat topped). Logo, foi necessário alterar o cálculo das posições dos vértices do hexágono no método draw() e a função convert() para converter coordernadas da matriz em pixels corretamente para o novo leiaute de hexágonos.

Foi também adicionada uma pasta chamada "assets"que contém todas as imagens usadas no jogo. Todas as imagens foram retiradas do website OpenGameArt e têm licença livre.

4 Protocolos

4.1 Mudança de posição

O protocolo de mudança de posição é utilizado na função InsereExercito() da arena.c para imprimir a posição inicial do robô assim que ele é criado. Também é utilizado na chamada de sistema MOV para mover um robô de posição.

4.2 Protocolo "rob"

Foi adicionado no maq.h um campo ID que é o mesmo número de registro usado no apres. O registro dos robôs por meio do protocolo rob acontece na função InsereExercito() do arena.c. A imagem carregada para o robô está na pasta assets e tem o formato: nomeDoExercito.png.

4.3 Protocolo "cristais"

No apres foi criada a classe Cristal para facilitar a manipulação dos cristais. O método draw() do cristal desenha também o número de cristais n presentes na posição i, j. Os cristais são adicionados em uma lista para serem redesenhados quando forem sobrescritos, como por exemplo, quando um robô sai de uma posição onde haviam cristais, então os cristais são sobrescritos quando a célula é redesenhada, logo é necessário uma estrutura para manter os cristais.

O protocolo cristais é utilizado na InicializaArena() do arena.c. que distribui cristais aleatoriamente pela matriz hexagonal depositando até 3 cristais em uma célula.

4.4 Protocolo "rmcristais"

Usada quando um robô deposita todos os seus cristais em uma posição vizinha na chamada de sistema DEPO. O protocolo rmcristais está definido assim:

rmcristais i j – remove o cristal que está na posição (i, j)

4.5 Protocolo "base"

No apres foi criada uma classe base para facilitar a impressão das bases, uma vez que cada base recebe uma imagem diferente. O protocolo é utilizado na InsereExercito() da arena.c que carrega a imagem da base contida na pasta assets no seguinte formato HQnomeDoExercito.png.

4.6 Protocolo "morre"

Assim que a vida de um robô vai para 0 a apres é chamada a partir da chamada de sistema ATK (ataque) para desenhar uma célula no lugar em que o robô morreu. O protocolo está definido assim:

morre i j – remove o robô da posição (i ,j)

4.7 Protocolo "cel"

Tem como objetivo identificar graficamente o tipo de terreno da célula. A matriz é desenhada normalmente como anteriormente com a cor cinza que representa a estrada. Porém o protocolo cel é chamado na InicializaArena() do arena.c para desenhar outros tipos de terreno. Neste EP estão implementados os tipos estrada e pântano. O tipo pântano é representado pela cor verde musgo na apresentação gráfica. O protocolo cel está definido assim:

cel i j terreno – redesenha a célula (i, j) com a cor que representa o terreno.

5 Saída dos robôs de um terreno

Na fase anterior, os robôs precisavam gastar cristais para sair do terreno pântano. Nesta fase, foi implementado um contador na execmaquina(). Toda vez que o contador mod 6 == 0, uma instrução de chamada de sistema pode ser executada. Se o contador não for 0, será incrementado em uma unidade a cada ciclo até que a chamada de sistema possa ser executada, então o contador é zerado. Quando uma chamada de sistema é executada ele é incrementado em 3 unidades, mas quando o terreno for pântano é incrementado em 2 unidades e logo levará um ciclo a mais para ser executada.

6 Teste

No teste fornecido existem dois exércitos com dois robôs cada um. Um robô se move em um círculo para demonstrar a MOV, 2 robôs (um de cada exército) atacam todas as posições ao seu redor

duas vezes e um dos robôs eventualmente morre. Por fim, um robô irá pegar todos os cristais que estão na sua vizinhança, irá se mover uma casa para cima e depositar seus cristais na diagonal direita inferior. Para executar o teste basta rodar o make e executar o ./teste.