

Introdução e objetivo

Estratégia em tempo real (RTS: *Real-time Strategy*) é um subgênero de jogos eletrônicos de estratégia, em que o andamento do jogo ocorre em tempo real. Em um RTS, os jogadores controlam unidades e edifícios com o objetivo de eliminar as forças de seus oponentes. Outras tarefas características do gênero são: produção de novas unidades, construção de edifícios, coleta de recursos e pesquisa de tecnologias.

Em jogos comerciais, a Inteligência Artificial (IA) acaba não sendo tão bem desenvolvida por diversos motivos, entre eles [1]:

- Alta complexidade de um RTS, já que são diversas ações sendo executadas simultaneamente pelos jogadores, com novas unidades e edifícios sendo criados no decorrer da partida;
- Dificuldade em se conduzir experimentos de IA, pois o suporte oferecido pelos jogos é muito baixo.

Uma alternativa para a realização de experimentos de IA de um RTS é a criação de um simulador próprio. Esta solução provê uma maior autonomia, pois a IA não ficaria amarrada às particularidades do jogo, sendo possível modificá-lo de acordo com as necessidades.

O objetivo deste trabalho é o desenvolvimento de um simulador de RTS para o estudo e aplicações práticas de técnicas de IA.

Simulador

O simulador implementado possui as seguintes características:

- Representação do mapa em grade bidimensional quadrada, visto de cima para baixo;
- Coleta de recursos distribuídos pelo mapa;
- Treinamento de unidades;
- Construção de edifícios;
- Movimentação e combate de unidades;
- Detecção e tratamento de colisão entre unidades.

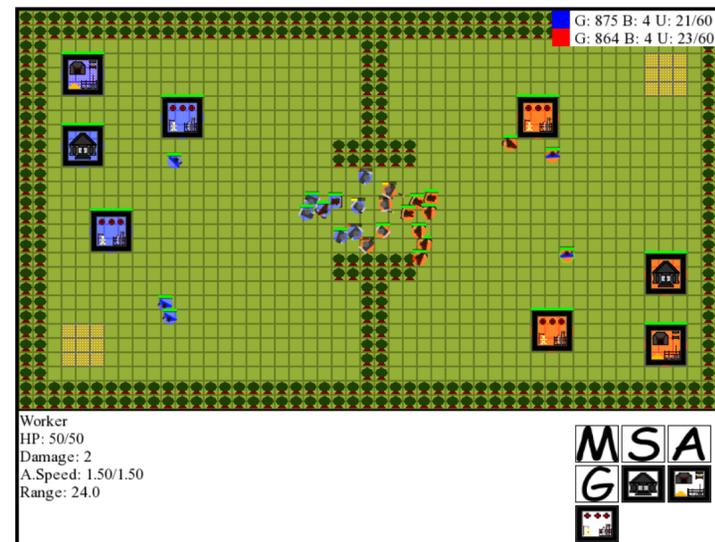


Figura 1: Tela do simulador.

Conceitos e técnicas

Os principais problemas encontrados na implementação do simulador de RTS estão relacionados a movimentação e colisão de unidades. A seguir são apresentados dois problemas e suas respectivas soluções implementadas.

Movimentação

As unidades podem se mover livremente pelo mapa, porém existem obstáculos distribuídos pelo mapa, que podem ser naturais (árvores) ou controlados pelos jogadores (edifícios). Para que a movimentação das unidades ocorra de forma eficiente, foi utilizado o algoritmo A* para a busca do caminho.



Figura 2: Busca de caminho utilizando o algoritmo A*. O caminho encontrado está destacado em vermelho. Os nós visitados pelo algoritmo estão em branco.

Detecção de colisão

A detecção de colisão é uma tarefa que demanda bastante processamento, pois ela é realizada a todo instante. A maneira mais simples de detectar todas as colisões que estão ocorrendo é realizando comparações dois a dois entre as unidades do jogo. Para reduzirmos o número de comparações, foi utilizada uma técnica de particionamento do mapa em áreas menores, chamadas de quadrantes. Um quadrante pode ser subdividido em quadrantes menores, assim que a quantidade de unidades presentes nele ultrapasse um limite pré-determinado.

Na tarefa de detecção de colisão realizamos as comparações somente entre unidades que estejam no mesmo quadrante, reduzindo consideravelmente o número de comparações. Para a implementação desta técnica foi utilizada uma estrutura de dados chamada *Quadtree*, que é uma árvore a qual cada nó possui exatamente quatro filhos. Em cada nó da árvore é mantida uma lista de unidades que estão estritamente no quadrante que ele representa.

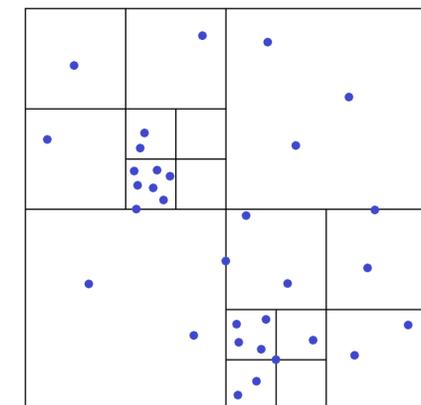


Figura 3: Particionamento do espaço em quadrantes. Nas áreas de alta concentração de unidades, ocorrem mais subdivisões do espaço.

Referências

- [1] Michael Buro. Call for ai research in rts games. In *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, pages 139–142, 2004.