

DOCUMENTAÇÃO ROBOX FOREVER

Nas páginas seguintes realizamos um detalhamento sobre o funcionamento do jogo, informando orientações, divisões realizadas no trabalho e destrinchando as etapas feitas para a sua conclusão:

1. Descrição

O Robox Forever foi inspirado no jogo *Sokoban* (倉庫番, Literalmente: "Armazém guarda"), um tipo de jogo de transporte e movimentação de cubos em um armazém. O objetivo do jogo é pegar e estocar determinados objetos em posições específicas chamadas de espaços-alvo.

Ele foi criado em 1981 por Hiroyuki Imabayashi, e publicado em 1982 por Thinking Rabbit, uma empresa de software localizada em Takarazuka.

É um jogo bem desafiador e interessante, pois envolve lógica no sentido de que há movimentos específicos que podem fazer com que as caixas sejam empurradas para as posições especificadas. Assim, deve haver um raciocínio a priori para que se saiba exatamente onde movimentar os objetos.

2. Estudo do jogo

O jogo *Sokoban* pode ser estudado do ponto de vista da teoria da complexidade. Podemos dizer que resolver os níveis de *Sokoban* demonstrou-se ser um problema de complexidade NP-difícil, no sentido de que o jogo faz parte da classe mais geral de problemas de planejamento de viagens, onde o jogador pode empurrar ou puxar um ou mais objetos por vez.

Assim, o jogo também é interessante para pesquisadores de inteligência artificial, pois resolver níveis apresenta problemas difíceis, para os quais não existem algoritmos de resolução rápida até o momento. Fato é que o *Sokoban* está incluso no espaço de problemas do PSPAGE¹ completo.

A dificuldade do *Sokoban* deriva do seu fator de ramificação, mas também da vasta profundidade de sua árvore de busca. Desse modo, pode-se dizer que alguns níveis requerem vários milhares de movimentos e muitos empurrões.

Jogadores experientes confiam principalmente em métodos heurísticos. Eles geralmente são capazes de eliminar rapidamente táticas de jogo desnecessárias ou redundantes e identificam layouts de jogo e sub-objetivos, permitindo-lhes restringir significativamente o escopo das pesquisas.

¹ Na ciência da computação teórica, especificamente na teoria da complexidade, PSPAGE é a classe de complexidade dos problemas de decisão decididos por uma máquina de Turing determinística com polinômio espacial.

3. Funcionamento

a. Regras do jogo

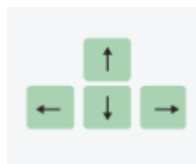
As regras do jogo são simples. O guardião do armazém – no nosso caso, o robô – deve guardas as caixas nos espaços-alvos. Nesse sentido, ele pode mover as caixas nas quatro direções e empurrar uma caixa de cada vez. A única restrição é que ele não pode puxar uma caixa.

Assim, uma vez que todas as caixas foram colocadas, o nível é passado e o jogador vai para o próximo nível, que é mais difícil.

O objetivo é resolver o quebra-cabeça, mas o ideal é resolvê-lo com o menor número possível de movimentos e empurrões.

b. Controles

Como dito, o robô empurra as caixas em torno de um labirinto e tenta colocá-las em locais designados. Para isso, pressione as teclas de navegação para mover o robô para as direções equivalentes.



Além disso, as teclas ‘W’, ‘A’, ‘S’ e ‘D’ também movem as caixas nas mesmas direções equivalentes, o que é comum na maioria dos jogos que envolvem movimentação. Por isso, mantivemos essa boa prática.

c. Gameplay

O jogo inicia-se com um menu inicial.

Ao apertar ‘espaço’ o usuário acessa um menu com os níveis disponíveis.

Feito isso, basta clicar com o mouse sobre algum nível e jogar.

Assim, a fase será iniciar e basta jogar!

4. Etapas do projeto

As Etapas do projeto consistiram-se inicialmente num Brainstorm onde discutimos quais jogos poderíamos fazer e a complexidade de cada um. No final, decidimos fazer o *Sokoban*.

A próxima etapa consistiu em realizar os diagramas do projeto, sendo eles o diagrama de classes, o diagrama de uso e o diagrama de atividades.

Com a finalização desta etapa, realizamos a criação do repositório e o desenvolvimento do projeto, criamos as classes, as funções e por fim os níveis.

Feito isso, realizamos as etapas finais de ajuste de erros, tratamento de exceções e realizamos também o processo de gamificação, inserindo sons, ajustando imagens e organizando.