



Ball Sort Puzzle Artificial Intelligence

André Malheiro up201706280@fe.up.pt

Diogo Gomes up201806572@fe.up.pt

Rúben Almeida up201704618@fe.up.pt

FEUP MIEIC IART Grupo43 2020/2021

Turma 3MIEIC07

Especificação do Problema e Referências Bibliográficas

Especificação do Problema:

Ball Sort Puzzle é um jogo do tipo puzzle cujo objetivo é alcançar uma ordenação homogênea das cores das bolas em todos o tubos que compõem o problema.

Ball Sort Puzzle originalmente é um jogo sem condição de vitória definida. Existe uma geração de níveis cada vez mais difíceis aos jogadores. Para efeitos de simplificação este pressuposto foi alterado. Passando a existir a consideração de vitória ao fim de um número finito de níveis ultrapassados

Referências Bibliográficas:

Existem várias produções deste jogo comerciais ou não. A produção sugerida como ponto de referência está disponível para Android na [PlayStore](#).

Para além desta produção comercial que não disponibiliza o código fonte publicamente, para tópicos diferentes da GUI, relacionados com controlo e IA foram usado como referência múltiplos repositórios de Github não comerciais dos quais se destaca o [seguinte](#), por fazer uso de heurísticas relevantes e do algoritmo A*

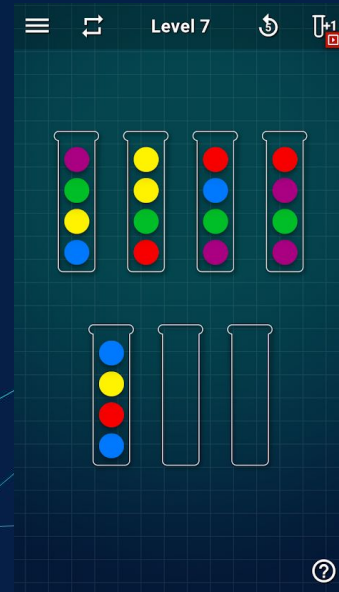


Fig.1:Layout da implementação em Android

Formulação do problema

- **Representação do estado**

Uma lista de tubos de capacidade N. Cada tubo por sua vez possui a lista das bolas que é portador.

Cada bola é um objeto complexo que para além das “sprites” necessárias para as funções de rendering, mantém um *Pair* ternário representativo da sua cor em RGB.

Implicitamente o estado pode ser abstraído como uma matriz de bolas.

- **Estado inicial**

A lista de tubos é preenchida de acordo com os valores definidos para representar o nível 1. Um preenchimento com bolas de cor e quantidade aleatória.

Poderão existir tubos que inicialmente não têm preenchimento

- **Estado objetivo**

Cada um dos tubos estará vazio ou terá de possuir uma lista homogênea de precisamente N bolas da mesma cor

- **Função de Avaliação / Heurísticas**

No caso do Algoritmo A*, a formulação heurística é feita de acordo com a fórmula $f(n)=g(n)+h(n)$, em que o custo da jogada $[f(n)]$ é obtido pelo somatório de $g(n)$, o custo real da jogada, e $h(n)$ o valor da heurística.

$g(n)$ é avaliado com a simples quantificação do número de movimentos já executados. Os diferentes $h(n)$ considerados são os seguintes:

h1 - Quantificação da Entropia - Somatório do número de bolas que estão no local errado, ou seja, que perturbam a homogeneidade da cor da sequência de bolas de cada tubo.

h2 - Quantificação da Distância à Homogeneidade - Somatório do número de movimentos que eram necessários para permitir a cada bola atingir o tubo cuja bola na base é da mesma cor sem contemplar regras.

h3 - Inspirada em Taboo Search - Número de tubos vazios * 10 + Somatório das sequências homogêneas de bolas a partir da base de cada um dos tubos * 5 (Não pode ser usada no Algoritmo A*)

Operadores

move(Tube1, Tube2) - Transfere a bola de um tubo para o outro

Tube1 - Tubo de onde a bola é extraída

Tube2 - Tubo onde a bola é inserida

Pré-condições:

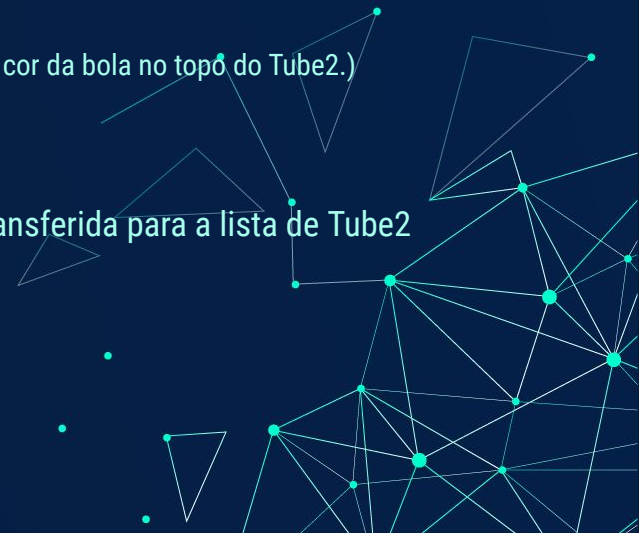
- Tube1 não está vazio && - Tube1 não está resolvido && - Tube2 não está cheio
 &&
 (- O Tube2 está vazio || - A cor da bola no topo de Tube1 é igual à cor da bola no topo do Tube2.)

Efeitos:

- A bola no topo do Tube1 é removida da lista de bolas deste tubo e é transferida para a lista de Tube2

Custo:

- Todas as operações têm custo unitário de 1



Detalhes de Implementação

- Linguagem de programação - Python 3.8
- Arquitetura adaptada de MVC usando paradigma OOP
- Bibliotecas usadas - GUI: Pygame 2
- Ambiente de Desenvolvimento: Pycharm
- Estruturas de Dados:

- Listas de Tubos

-Listas de Bolas

-Queue para visitar os nós

-Set para guardar nós visitados

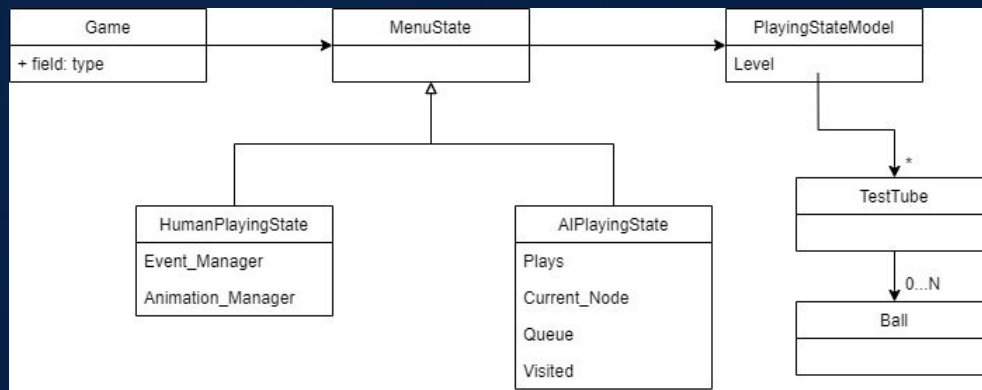


Fig.2:Esquema UML da estrutura central de Controlo do programa

Estado do Desenvolvimento

Concluído:

- É possível executar movimentos no modo humano com resposta da GUI.
- O Algoritmo BFS produz um resultado satisfatório, mas sem produção de animação na GUI.
- Metodologia de posicionamento dos tubos na GUI inspirado em flex row

Por Desenvolver:

- O modo de AI produzir animação nos seus movimentos.
- Introdução dos método de pesquisa orientada
- Desenho dos restantes menus acessórios da GUI
- Capacidade de produzir *hints* no modo de jogador humano

Incertezas:

- Desacoplar o estado do jogo dos elementos gráficos. A replicação de TestTubes e Balls torna as cópias de estado impraticáveis pela dimensão acrescida da inclusão de elementos Visuais como *sprites* nas cópias.
- Forma de Tratar o critério de fim de jogo. Visto Ball Sort Puzzle originalmente ser um jogo com um número infinito de níveis- Hipótese de criar metodologia de geração de níveis programaticamente

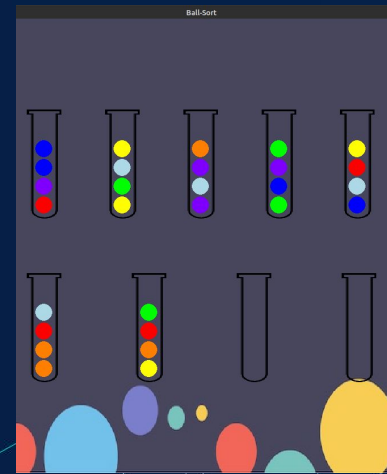


Fig.3: Imagem do layout atual da solução