

# Ball Sort Puzzle

Heuristic Search Methods for One Player Solitaire Games

Artificial Intelligence

Master in Informatics and Computing Engineering



Gonalo Alves – up201806451  
Gustavo Mendes – up201806078  
Pedro Seixas – up201806227

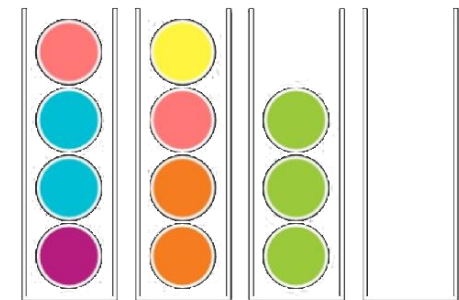
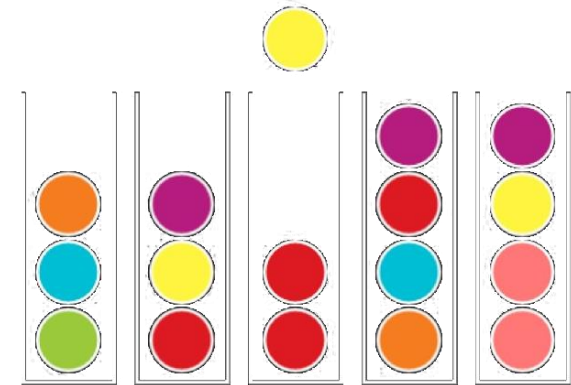
# Specification

The objective of the Ball Sort Puzzle is to sort the colored balls in the tubes until all balls with the same color stay in the same tube.

The player can only move a ball at the top of a tube to:

- an empty tube;
- another tube that has a ball with the same color on top and has enough space;

The player can also undo his moves if he finds himself without moves.



# Related Work

- [Link to the Google Play page](#);
- [Link of all existing levels](#);

We used the curricular unit's presentations to guide our implementation of the various algorithms.

# Formulation of the problem as a search problem

## State Representation:

- Ball:  $[1..\infty]$
- Tube: [Ball, Ball, Ball, Ball] or []
- Game: [Tube, Tube, ...]

## Initial State:

- Game: [Tube, Tube, ...], where every Tube doesn't contain 4 equal numbers

## Objective State:

- Game: [Tube, Tube, ...], where every Tube either contains 4 equal numbers or is empty

## Operators:

- Move(X,Y):
  - PreCond: Tube Y must be empty or have a Ball just like the one that is moved
  - Effect: move Ball from Tube X to Tube Y
  - Cost: 1 is the general case, it represents a move; C2, evaluation of the number of wrongly placed Balls, based on the Ball at the bottom of a Tube

# Implementation work already carried out

**Programming language:** [Python](#), with visualization using [pygame](#) package

**Development environment:** [VSCode](#)/[IntelliJ](#)

**Data Structures:**

- Lists, for representing the Tubes and the Game
- Nodes and Graphs

**File Structure:** The Project's Repository is available at [Github](#)

**Implemented Work:** All algorithms lectured; Graphical Interface, playable game with hints