The Three Stones Game

Table of Contents

1- Technical features

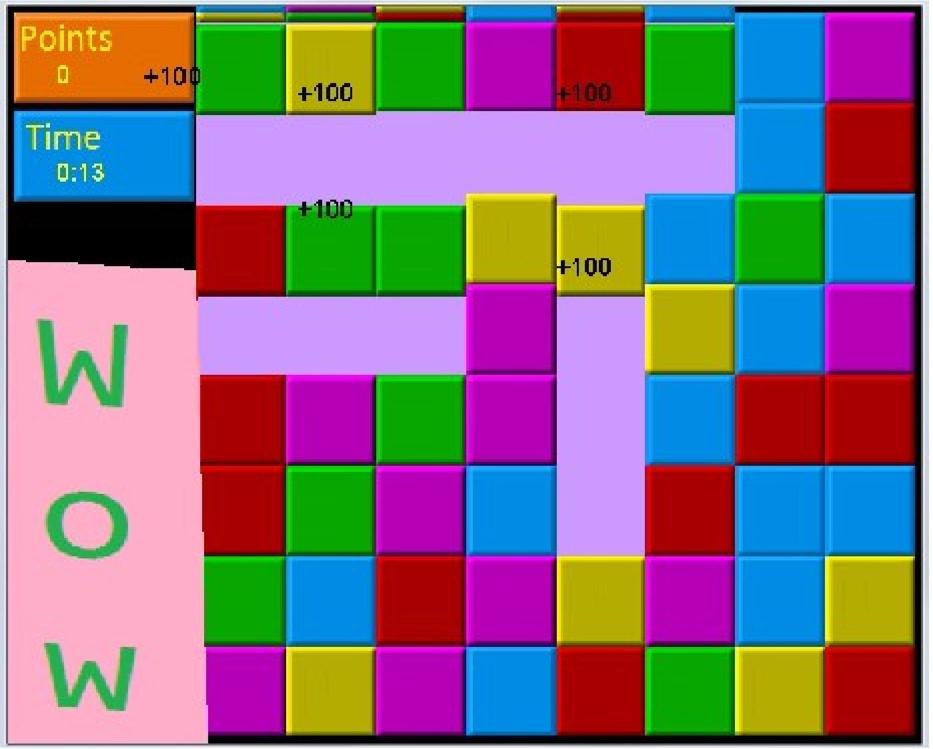
The main technical features implemented in the game are:

1-Memory management:
- Overload of new and delete operators
- Overload std::allocator to be used by the STL containers
- Implementation of a memory tracker to detect possible memory leaks
- Use of Boost::smart_ptr library

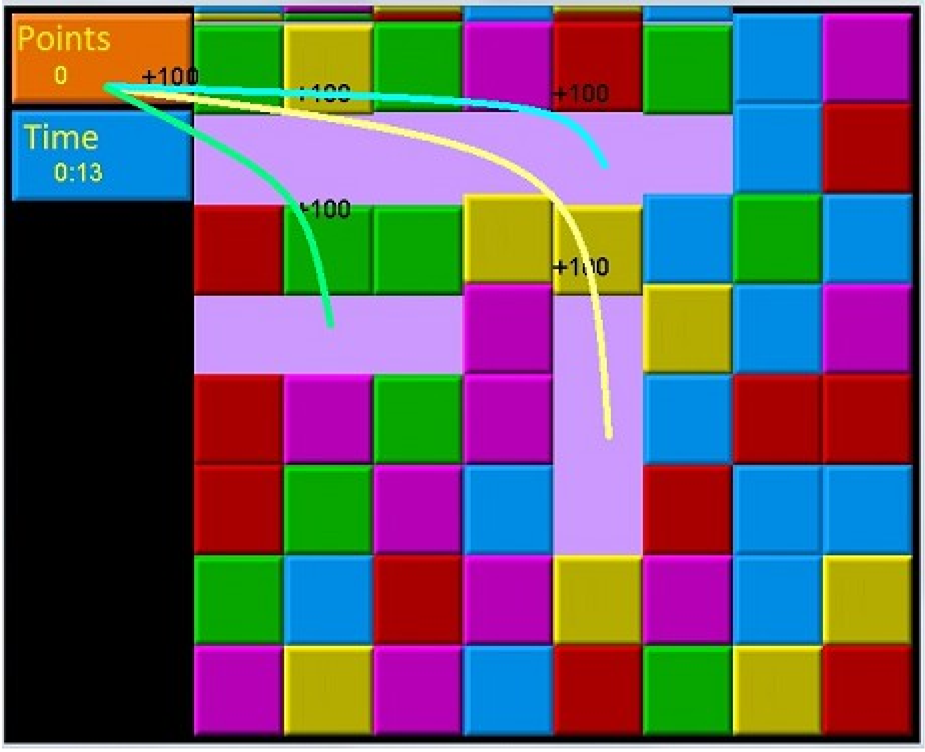
2-Delayed functions execution:
Some functions should not be executed when they are called. They are registered for later execution. This is implemented through the structure DelayedFunctionInfo.

3-Vertex animation in Encouragement class.

In the figure below, the Envouragement class is showing the animated “wow” text with the player scoring a high score.



4-Score class uses interpolation along a quadratic-polynomial-curvy path (inspired from the Uniform B-Spline curve) to move the score from the squares that created it to the score rectangle. The following figure shows the paths crossed by three scores.



5-Finite state machine:
Implemented in Game, Board and Tile classes.

2- Overview

Overview of the main elements in the game:

1-
Game class contains the different phases of the game, implemented using finite state machine (see chart below)
It contains and manages a board, a score, a time counter and an encouragement object.

2-
The board class contains 64 tiles. Each tile can be one of 5 types (colors)

3-
ObjectsManager is a class that calls Update and Draw functions of the main classes of the game, that are derived from Object class.

Objects contained between activeobjects.next and activeobjectstail.prev are updated and drawn.
Those contained between activeobjectstail.next and activeobjects.prev are drawn only. This allows the board, tiles, score and time counter to be drawn without being updated when the intro, the count down and the time out objects are in the front.

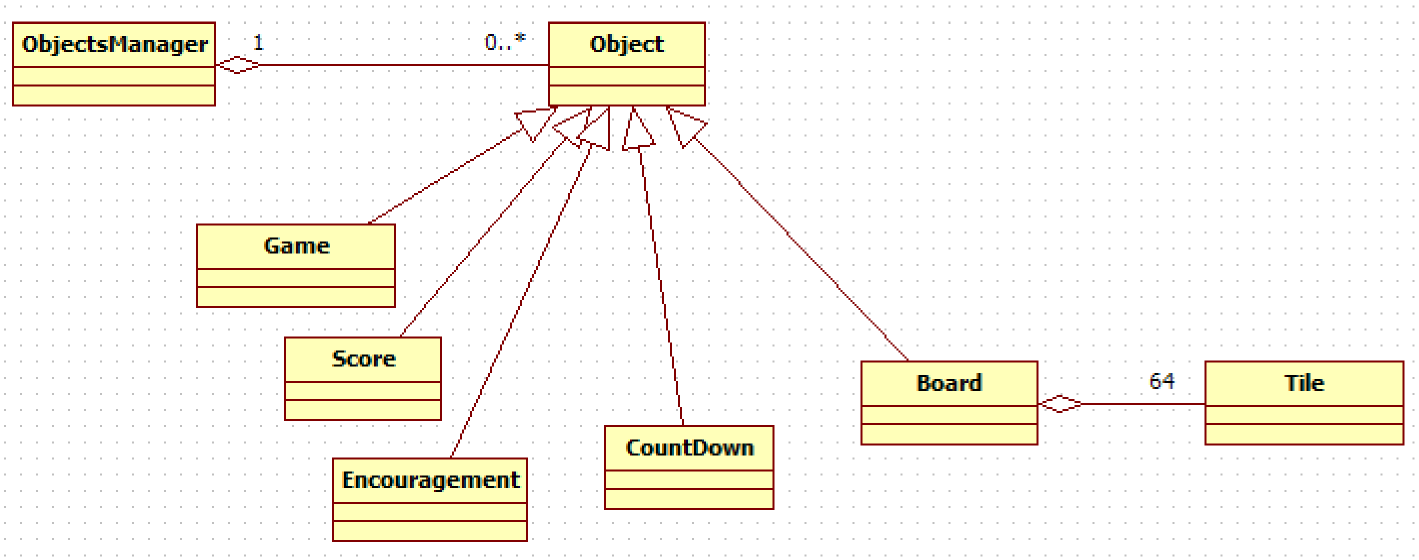
3- Compiling the game

- Boost library is needed to compile the game.
- The game was compiled and tested in Microsoft VC 2010 Express

4- Class diagrams

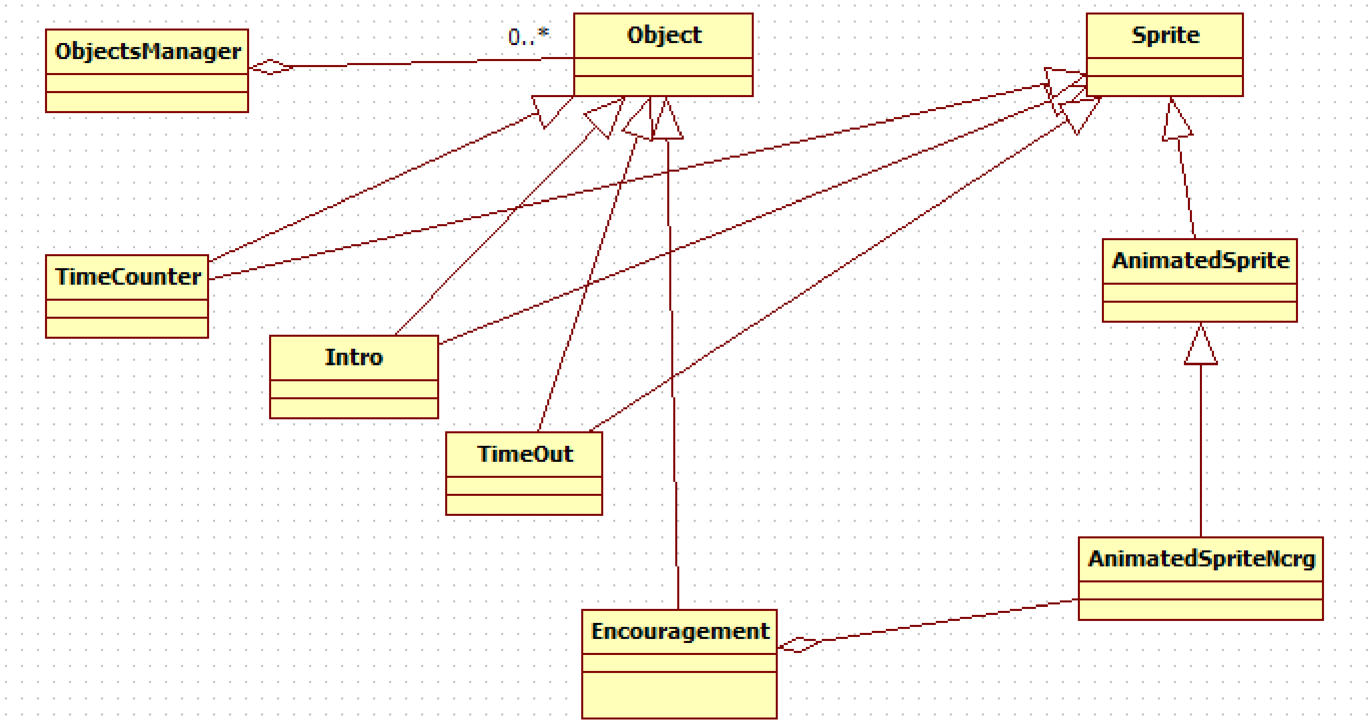
4.1- Object class

The following is the class diagram showing the relation between Object class and some main classes in the game :

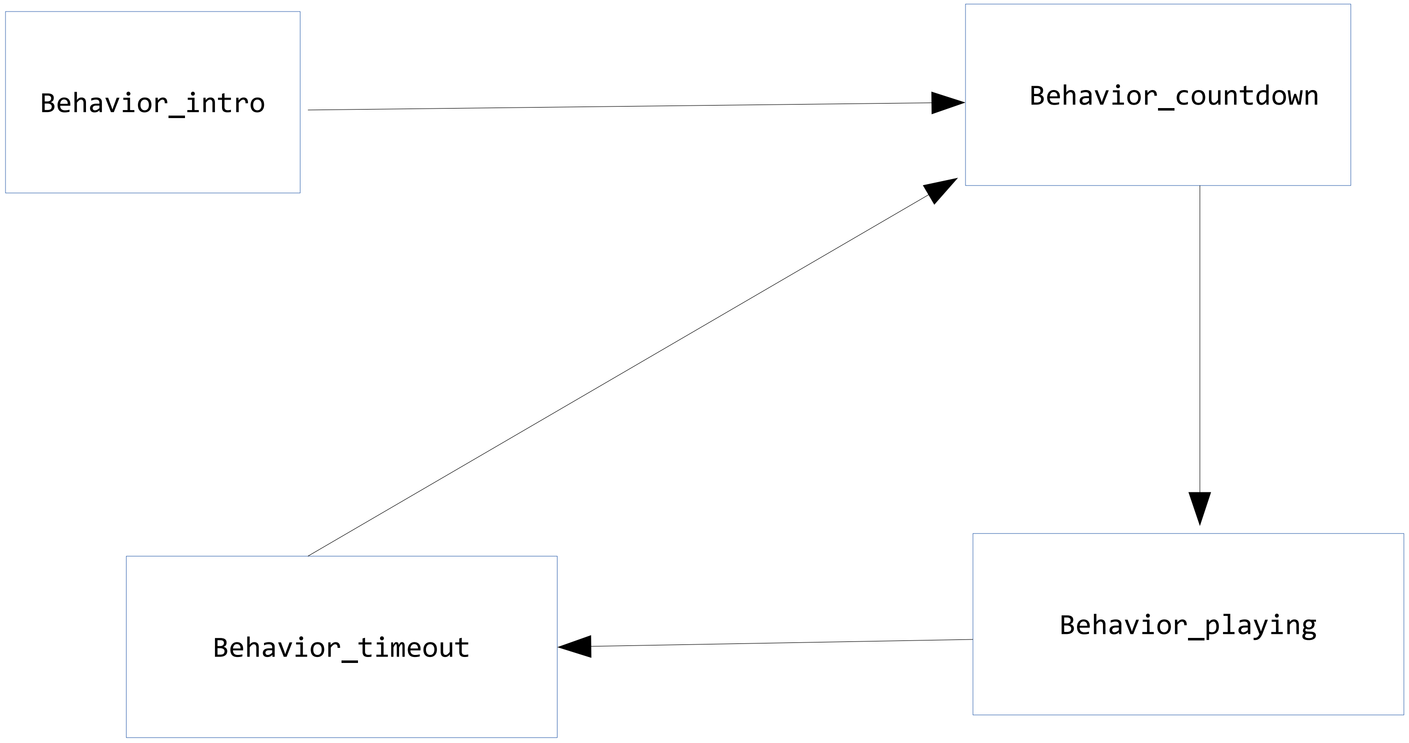


4.2- Sprite

The following is the class diagram showing the relation between Object and Sprite classes and some main classes in the game :

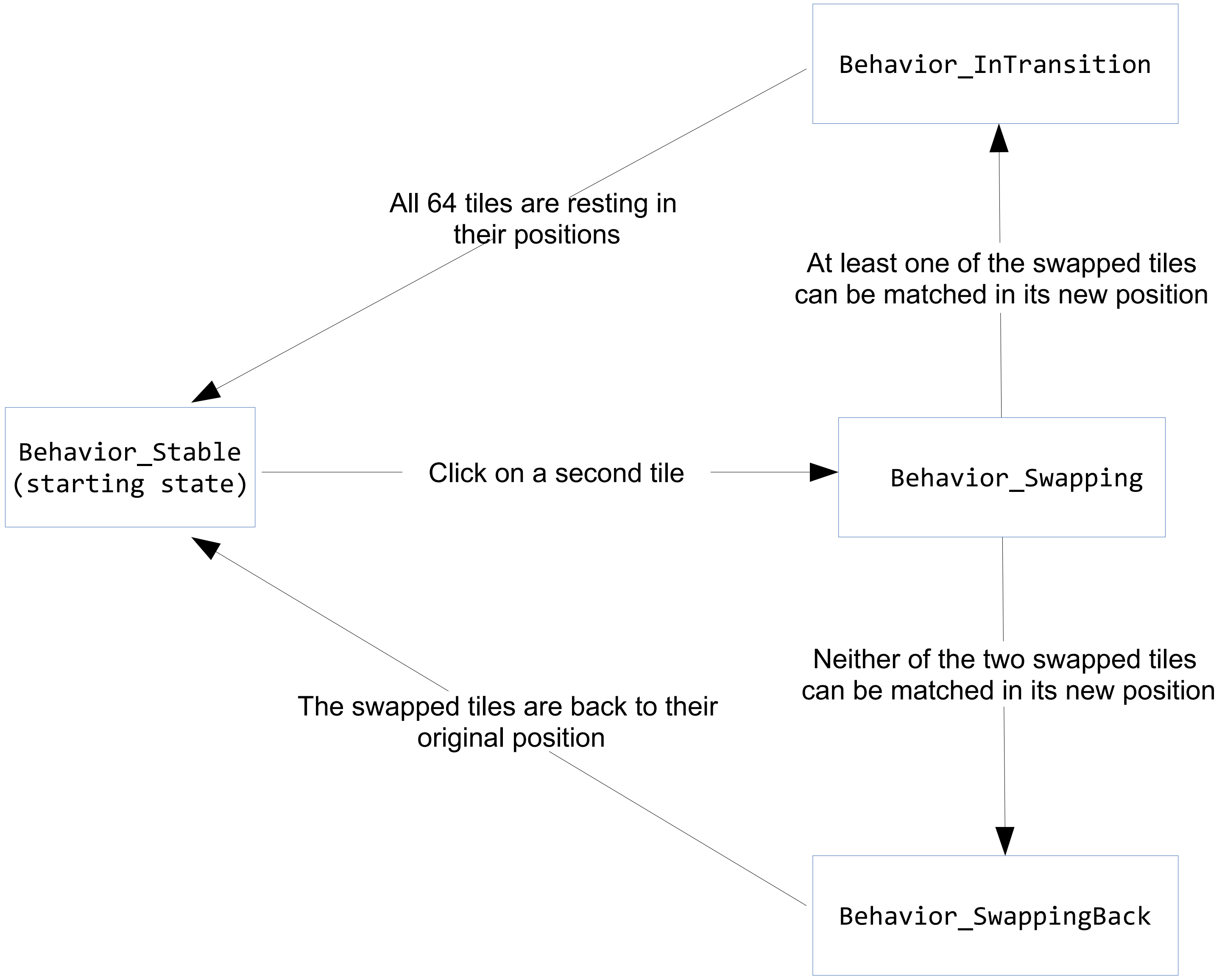


5- The Game's finite state machine



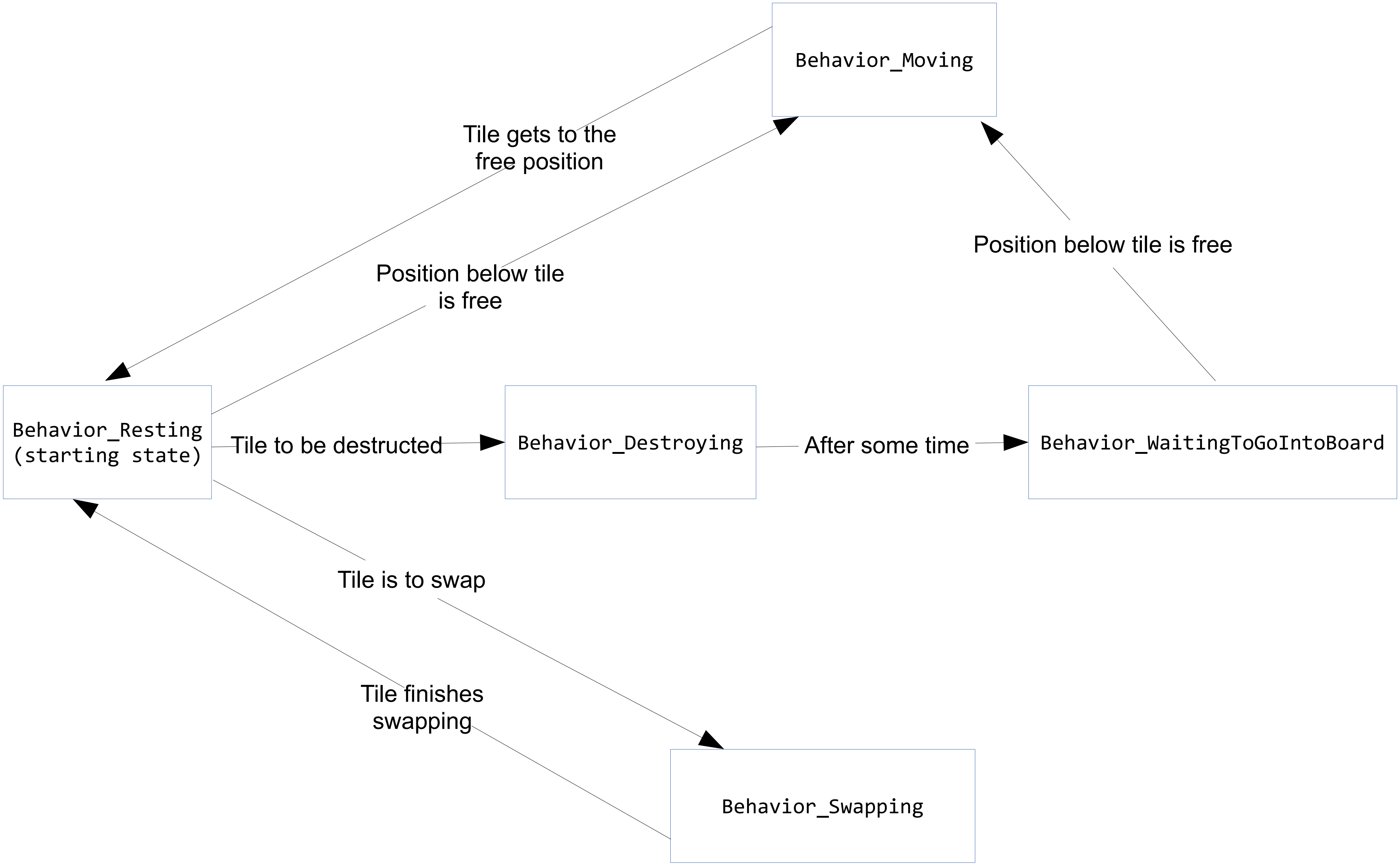
6- The Board's finite state machine

The board's states are determined by the behavior of the tiles.
Here are the states of the board:


Behavior_InTransition
All 64 tiles are resting in their positions
At least one of the swapped tiles can be matched in its new position
Behavior_Stable (starting state)
Click on a second tile
Behavior_Swapping
Neither of the two swapped tiles can be matched in its new position
The swapped tiles are back to their original position
Behavior_SwappingBack

7- The Tile's finite state machine

Here are the states of the tile :


Behavior_Moving
Tile gets to the free position
Position below tile is free
Position below tile is free
Behavior_Resting (starting state)
Tile to be destructed
Behavior_Destroying
After some time
Behavior_WaitingToGoIntoBoard
Tile is to swap
Tile finishes swapping
Behavior_Swapping