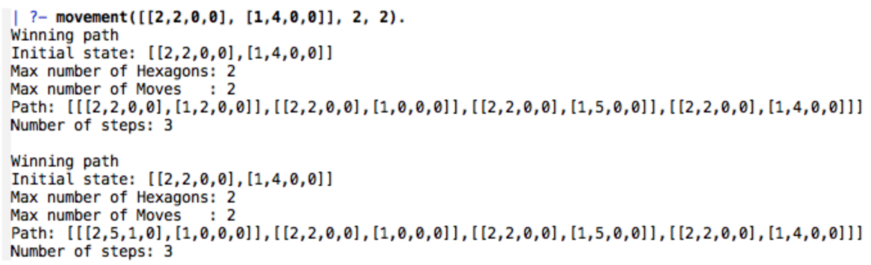

```
| ?- movement([[2,2,0,0], [1,4,0,0]], 2, 2).
Winning path
Initial state: [[2,2,0,0],[1,4,0,0]]
Max number of Hexagons: 2
Max number of Moves   : 2
Path: [[[2,2,0,0],[1,2,0,0]],[[2,2,0,0],[1,0,0,0]],[[2,2,0,0],[1,5,0,0]],[[2,2,0,0],[1,4,0,0]]]
Number of steps: 3

Winning path
Initial state: [[2,2,0,0],[1,4,0,0]]
Max number of Hexagons: 2
Max number of Moves   : 2
Path: [[[2,5,1,0],[1,0,0,0]],[[2,2,0,0],[1,0,0,0]],[[2,2,0,0],[1,5,0,0]],[[2,2,0,0],[1,4,0,0]]]
Number of steps: 3
```

Figure 8: *FightHPV*-Ep.1: Output example of our constraint model.

amount of playing time allocated to each Ep. According to Tab.1, 'Zero' winning paths simply means that the Ep. cannot be solved and hence is intractable. In the other cases, the guidelines estimate the difficulty based on how hard it was for a computational approach to solve an Ep. This information can then be used to guide the designer in his choices to order the various Ep.

This approach however does not guarantee *playability* or *pleasure*. Both these factors should ideally be objectively evaluated through user testing, but gameplay design is still a craftman's activity and large-scale experiments are difficult and costly to set up. Using the number of winning paths and the average time required to find a winning state with an automated procedure, to evaluate the difficulty of an episode is discussable. On the one hand, human reasoning is not comparable and reductible to these numbers, but on the other hand, adopting this methodology while the game is not yet deployed at the larger scale (i.e., worldwide) accelerates its development and analysis. For *FightHPV*, using the proposed constraint model, it is possible to evaluate objectively the computational difficulty and thus to use the proposed guidlines to improve the gaming experience in further releases of the game.

Nb. of winning paths	Av. Time to find a win. path	Ep. Difficulty
Zero	N/A	Intractable
Low	Low	Easy/Medium
Low	High	Hard
High	Low	Easy
High	High	Medium/Hard

Table 1: Guideline for determining the difficulty of an Ep.

Our experimental analysis aimed at evaluating the level of difficulty of each Ep. and compare them.

Evaluating Each Episode

For each episode, the constraint model was exhaustively explored up to a certain boundary to evaluate its difficulty. The results for Ep.3 are given in Tab.2. Given a maximum number of actions (1^{st} col.), i.e., rotation and translations, we computed the number of paths leading to one winning condition of the game (2^{nd} col.). We also measured the total runtime required to compute all these paths (3^{rd} col.) and computed the average runtime to find one such path (4^{th} col.). We computed these tables for each episode up to Ep.10. In fact, computing these tables for Ep. higher than 7 already required more than 4 hours.

Max. num. of act.	Num. of win. pat.	Runtime	Average Runtime
1	0	0min 0sec 02	–
2	5	0min 0sec 01	2.00ms
3	27	0min 0sec 04	1.48ms
4	265	0min 0sec 29	1.09ms
5	1654	0min 1sec 92	1.16ms
6	12810	0min 14sec 66	1.14ms
7	84805	1min 44sec 98	1.24ms
8	615719	12min 4sec 96	1.18ms

Table 2: *FightHPV*-Ep.3: Number of winning paths, runtime to find all solutions, average runtime to find one solution.

Comparing the Difficulty of Episodes

We compared the tables computed for each episode when the maximum number of actions was set up to 7. This arbitrary number was selected because it corresponds to a reasonable effort of the player. By reasonable effort, we meant an appropriate level of effort to find one solution path.

Tab.3 shows the comparison for all episodes of the game up to Ep.10.

Level	Number of win. paths	Runtime	Average Runtime to find a win. path (ms)
1	123276	2min 55sec	1.02ms
2	96510	52min 56sec	32.91ms
3	84805	1min 44sec	1.24ms
4	61602	28min 59sec	28.24ms
5	828	245min 49sec	17,813.97ms
6	68	46min 4sec	40,945.44ms
7	1	75min 28sec	452,8440.00ms
8	0	157min 37sec	–
9	48	26min 33sec	33,191.25ms
10	0	158min 19sec	–

Table 3: Number of winning paths and runtime to find solutions for all levels ($MaxMove = 7$).