I choose the game: Unroll Me (http://turbochilli.com/games/unroll-me/). The following screenshots (from the same website above) show a possible and valid start state (left) and a possible goal state (right).

**States:** Given a $w$ by $h$ board with 1 initial square with a white ball in it, 1 goal square, $m$ movable squares, $n$ static squares, $ss$ spaces, and $2 + m + n + ss = wh$ ($w, h > ss$), the initial and goal square each has 4 placements (up, down, left, right; all are passes), each movable and static square has 8 placements (4 types of passes, 4 types of turns), the number of possible states is:

$$\frac{(wh)!}{(2+ss)!} * 2^4 * (wh-(2+ss))^8$$

Some of the states are not valid. The slot in the initial or the goal square cannot face the edge or each other. Note that the slot of a static square does not have to face inwards because the goal path may not consist of a static square that has its slot facing the edge. Spaces cannot be
completely surrounded by static squares. In addition, there must exist a goal path as a result of a player’s square shuffling to pass the goal test.

**Initial State:** It can be any valid state excluding the invalid cases discussed above. Thus, the game has multiple initial states.

**Actions:** A movable square that has an adjacent space neighbor can be moved towards the space in four directions (i.e., left, right, up, and down). Thus, in a state \( s \), for any vertical and horizontal movable square neighbor \( n \) of a space \( ss \), and the direction \( d \) of \( n \) it is facing to \( ss \),

\[
\text{Actions}(n, s) = \{ \text{Move}(n, d) \}
\]

Note that because of the two parameters passed into Actions, the is only one possible move for a specific space and one of the space’s neighbors. For Actions(\( s \)), there exist at most \( 4o \) possible moves when the \( o \) spaces are neither on the edges nor next to each other, there exist at least 2 moves when there is only 1 space on one of the edges. Thus,

\[
\text{Actions}(s) = \{ \text{Move}(n_1, d_1), \text{Move}(n_2, d_2), \ldots \}
\]

In this game, Actions is a function of \( s \) because the environment is deterministic and the next possible action depends on the current state.

**Transition Model:** Given an action and a state where a space \( s \) is positioned at \( p_1 \) and one its movable neighbor \( n \) is positioned at \( p_2 \), the movement makes the positions of \( s \) and \( n \) switched. Mathematically, it can be expressed as

\[
\text{Result}(s, a) = (\text{Position}(s, p_2), \text{Position}(n, p_1))
\]

The transition from the current state to the next state is deterministic because the next state is completely determined by the current state (i.e., we can predict all possible moves leading to the next state).

**Goal Test:** In the limited time, a path is created so that the white ball can travel through the path and reach the red goal square without crashing. Mathematically, the goal state can be specified as a singleton set: \( \{ \text{In(ball, } p\text{goal}) \} \). If a given state is in the set, the test passes (i.e., won the game).

**Path Cost:** Each action is a step and has equal step cost: 1, so the path cost is the number of steps until the goal test passes. Given \( n \) steps to reach the goal state, each \( i_{th} \) step cost for an action \( a \) in a state \( s_i \) to state \( s_j \) is \( sc(s_i, a, s_j) \), the path cost can be expressed as:

\[
\sum_{i=1}^{n} sc(s_i, a, s_j)
\]

Zhao Han