Kiah Johnson

CS 510

Assignment 2

For this assignment, I was assigned to design a python program that could be able to solve the sliding brick puzzle game. I was not allowed to use any external libraries or packages, except the built in standard libraries and NumPy. My program for this assignment creates functions and variables to represent the game state, determine moves, validate moves, executing moves, and determining whether the puzzle has been solved. In addition to this, I have a shell script that can pass an argument to my main file, so the code can be run within a consistent interface.

As my final product, I turned in a project that has the functions of load_state, print_state, clone_state, goal_state, individual_moves, available_moves, apply_move, compare_states, normalize_state, swap_idx, and random_walk, all with different uses to solve the sliding brick puzzle. Each function is described below:

1. Load_state: loads game state
2. Print_state: prints the game state
3. Clone_state: clones a game state
4. Goal_state: checks if game state matches the goal state (checks if puzzle is solved)
5. Individual_moves: define all the possible moves for any piece, once given a piece will find all the possible moves for that piece
6. Available_moves: finds all the available moves for each state, so prints any possible moves that can be made
7. Apply_move: applies the move
8. Compare_states: compares two state to see if they are the same
9. Normalize_state: normalizes the state so that it looks like the sliding puzzle in real life but in matrix form
10. Swap_idx: swaps the indexes for normalize_state
11. Random_walk: performs random moves a certain amount of times for given state

Result using **print** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh print SBP-level1.txt
5,5,
1,1,1,1,1
1,3,2,2,1
1,0,4,5,1
-1,0,6,7,1
1,1,1,1,1
```

Result using **done** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh done SBP-level0.txt
False
kj645@tux2:~/CS510/A2$ sh run.sh done SBP-level0-solved.txt
True
```

Result using **availableMoves** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh availableMoves SBP-level1.txt
(3, down)
(4, left)
(6, left)
```

Result using **applyMove** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh applyMove SBP-level1.txt "(3,down)"
5,5,
1,1,1,1,1
1,0,2,2,1
1,3,4,5,1
-1,0,6,7,1
1,1,1,1,1
```

Result using **compare** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh compare SBP-level0.txt SBP-level1.txt
False
kj645@tux2:~/CS510/A2$ sh run.sh compare SBP-level0.txt SBP-level0.txt
True
```

Result using **norm** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh norm SBP-test-not-normalized.txt
6,8,
1,1,1,1,1,1
1,3,2,2,4,1
1,5,2,2,6,1
1,7,7,8,8,1
1,9,9,10,10,1
1,0,0,0,0,1
1,0,0,0,0,1
1,1,-1,-1,1,1
```

Result using **random** command:

```
kj645@tux2:~/CS510/A2$ sh run.sh random SBP-level1.txt 3
5,5,
1,1,1,1,1
1,3,2,2,1
1,0,4,5,1
-1,0,6,7,1
1,1,1,1,1

(4, left)
5,5,
1,1,1,1,1
1,3,2,2,1
1,4,0,5,1
-1,0,6,7,1
1,1,1,1,1

(6, left)
5,5,
1,1,1,1,1
1,3,2,2,1
1,0,4,5,1
-1,6,0,7,1
1,1,1,1,1

(3, down)
5,5,
1,1,1,1,1
1,0,2,2,1
1,3,4,5,1
-1,0,6,7,1
1,1,1,1,1
```