

---

# **slidingpuzzle**

***Release 0.0***

**J. Marcus Hughes**

**Jan 30, 2022**



# CONTENTS

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Usage . . . . .	3
1.2	API . . . . .	3



**slidingpuzzle** is a Python library for the *sliding puzzle game* <[https://en.wikipedia.org/wiki/Sliding\\_puzzle](https://en.wikipedia.org/wiki/Sliding_puzzle)>. Check out the *Usage* section for further information, including how to *Installation* the project.

---

**Note:** This project is under active development.

---



## CONTENTS

## 1.1 Usage

### 1.1.1 Installation

To use slidingpuzzle, first clone the repo and then install it using pip:

```
(.venv) $ git clone https://github.com/jmbhughes/slidingpuzzle.git  
(.venv) $ pip install .
```

Some day it might be available on PyPI.

### 1.1.2 Creating a puzzle

To create a predefined NPuzzle, you use a list defining the size, the configuration, and the solution. For example:

```
>>> import slidingpuzzle as sp  
>>> my_puzzle = sp.NPuzzle(3, [1, 2, 6, 3, 5, 0, 4, 7, 8], solution=[1, 2, 3, 4, 5, 6, 7, 8, 0])
```

Alternatively, you can generate a random puzzle.

To solve, you can use breadth-first search (BFS) or A\*. The syntax is the same. For A\*:

```
>>> import slidingpuzzle as sp  
>>> my_puzzle = sp.NPuzzle.random_puzzle(3)  
>>> solver_bfs = sp.AStarNPuzzleSolver(my_puzzle, sp.ManhattanHeuristic())  
>>> solution_bfs = solver_bfs.solve()  
>>> print(solution_bfs)
```

## 1.2 API

test