

# sliding\_puzzle

## 设计思路

主要程序被拆分为三个部分

- main.py 主要负责游戏显示和用户交互
- engine.py 主要负责后端的状态更新以及约束验证
- solver.py 主要负责解法的运算

设计思路主要使用两个单例(GameEngine和Solver)，分别作为模拟器和计算器

游戏的GUI使用pygame实现 \

<https://www.pygame.org/docs/>

其中为了保证搜索的效率，solver采用双向BFS实现 \

<https://www.geeksforgeeks.org/bidirectional-search/>

其中涉及的状态压缩(见Solver.getHash)的算法为康拓展开 \

<https://zh.wikipedia.org/wiki/%E5%BA%B7%E6%89%98%E5%B1%95%E5%BC%80>

## 使用说明

在运行之前先安装必备python库

```
pip install -r requirements.txt
```

可以直接使用python调用main.py运行

```
python main.py \  
--board_size=3 \  
--random_step=20 \  
--input_puzzle=input_puzzle.txt
```

将命令行在文件夹根目录运行

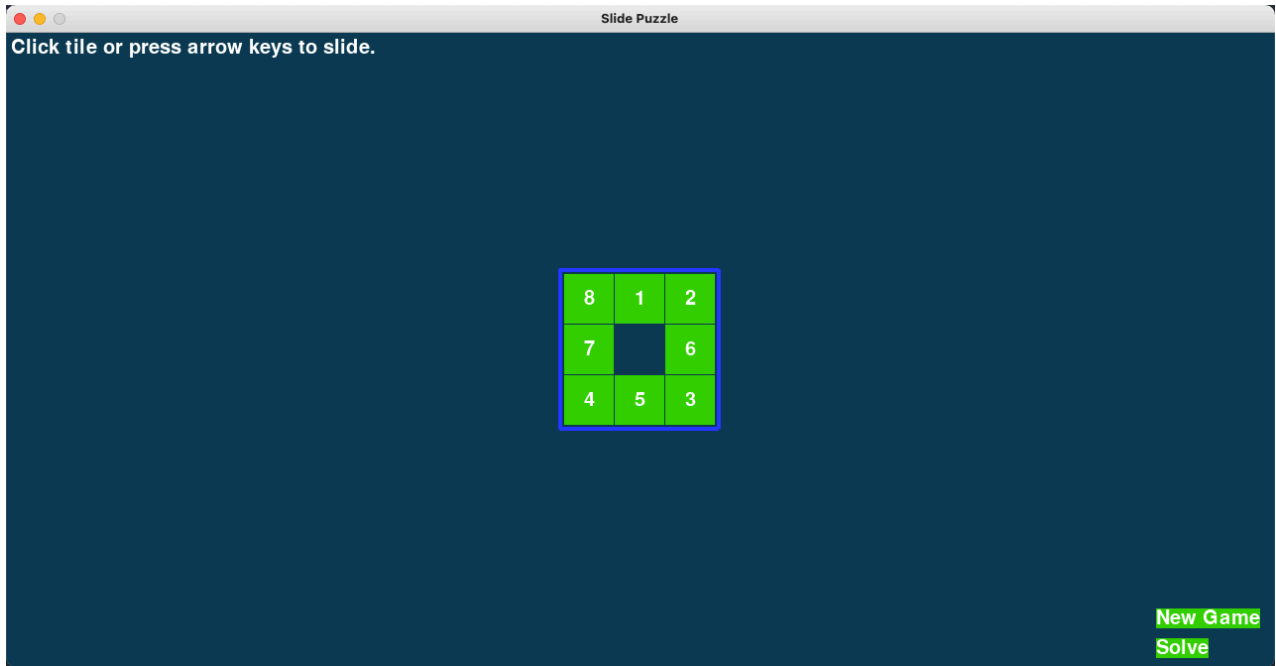
```
bash play_game.sh
```

也可以直接使用python

可以调整的参数包括

- board\_size 如果是自己玩，可以任意设置3-10任意尺寸，如果需要调用solve，建议设置为<=5的数，否则运算量超出极限
- random\_step 打乱的步数，默认为20，可以不用动这个参数

- input\_puzzle 如需输入给定的初始局面，在对应txt文件中按照input\_puzzle.txt格式输入即可



#### 游戏操作

- 游戏中任意时刻，可点击New Game，打乱拼图开始新的游戏
- 游戏中任意时刻，可点击Solve，可以让程序自动开始解决剩余步数，界面左上角显示"Automatically Solving..."
- 当胜利时，界面左上角显示"You WIN!"