

Documentation:

The Program has 3 main Classes: Grid, State, Rule . It simulates Childrens' Puzzle Book.

Main points in the Code:

Main function: create a Grid, an array of words (from the Input) and an Initiate state and call flailWildy function.

flailWildy: (in the first version – wordSearchMaker.py) main while loop which checks each time if the game was over and if not, locates in the grid the next word from the "bank".

backtrack: (in the second version – wordSearchMaker2.py) this function consistently checks if the goal is achieved, or any possible fail that prevents us continue keep searching for a solution in a specific branch. Otherwise, continues do deeper at the search tree and update 3 global variables each iteration.

allCandidates: there are 8 possible directions for each word and MxN initiate places on the grid. So, it calls precondition function 8MN times and create a list with possible rules for apply.

precondition: A Boolean type function which checks:

1. if writing the word on the grid is "out of bounds"
2. if writing the word on the grid is "overlap another word"

applyRule: locates the word on the grid by the following steps:

1. Creating a new updates list of words
2. deep copy of the previous grid
3. placing the current word in the grid
4. Creating a new State and return it.

Outputs for wordSearchMaker.py with 12x12 Grid:

```

+---+
| | g | | | m | u | m | i | t | p | o | m |
+---+
| | r | | c | | l | | | | d | i |
+---+
| k | a | | i | l | o | y | | c | e | l | s |
+---+
| c | p | a | t | i | c | r | | a | | a | s |
+---+
| a | h | g | s | s | a | t | d | n | h | n | i |
+---+
| r | s | e | i | p | l | e | | n | c | o | o |
+---+
| t | e | n | r | | n | m | | i | r | i | n |
+---+
| k | a | t | u | d | | m | | b | a | t | a |
+---+
| c | r | | e | | | y | | a | e | a | r |
+---+
| a | c | | h | | | s | | l | s | r | y |
+---+
| b | h | a | d | m | i | s | s | i | b | l | e |
+---+
| | | | | | | g | l | o | b | a | l |
+---+

```

remaining words:

```
[ ]
```

```

+---+---+---+---+---+---+---+---+---+---+---+---+
| g |   |   | d | n | e | d | a | e | d |   |
+---+---+---+---+---+---+---+---+---+---+---+---+
| r |   | y | r | a | n | o | i | s | s | i | m |
+---+---+---+---+---+---+---+---+---+---+---+---+
| a | a | h | e | u | r | i | s | t | i | c |   |
+---+---+---+---+---+---+---+---+---+---+---+---+
| p |   | d | b | a | c | k | t | r | a | c | k |
+---+---+---+---+---+---+---+---+---+---+---+---+
| h | l | c | m | m | u | m | i | t | p | o | s |
+---+---+---+---+---+---+---+---+---+---+---+---+
| s | t | a |   | i |   | l |   |   |   | e |
+---+---+---+---+---+---+---+---+---+---+---+---+
| e |   | n | n |   | s |   | a |   |   | a |
+---+---+---+---+---+---+---+---+---+---+---+---+
| a |   | n | e | o |   | s |   | b |   | r |
+---+---+---+---+---+---+---+---+---+---+---+---+
| r |   | i |   | g | i | p | i |   | o |   | c |
+---+---+---+---+---+---+---+---+---+---+---+---+
| c |   | b |   |   | a | t | s | b |   | l | h |
+---+---+---+---+---+---+---+---+---+---+---+---+
| h |   | a |   |   |   |   | a | i | l |   | g |
+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   | l | l | a | c | o | l | r | l | e |   |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

remaining words:

```
['symmetry']
```

```

+---+---+---+---+---+---+---+---+---+---+
| | | | | | | a | | p | s | i | l |
+---+---+---+---+---+---+---+---+---+---+
| | | | | | | g | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| | | | g | | e | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| | | | l | c | a | n | n | i | b | a | l |
+---+---+---+---+---+---+---+---+---+---+
| | | | o | | t | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| l | | | b | | | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| o | | | a | d | n | e | d | a | e | d | |
+---+---+---+---+---+---+---+---+---+---+
| c | | | l | | | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| a | d | m | i | s | s | i | b | l | e | | |
+---+---+---+---+---+---+---+---+---+---+
| l | | b | a | c | k | t | r | a | c | k | |
+---+---+---+---+---+---+---+---+---+---+
| | | | c | i | t | s | i | r | u | e | h |
+---+---+---+---+---+---+---+---+---+---+
| g | r | a | p | h | s | e | a | r | c | h | |
+---+---+---+---+---+---+---+---+---+---+

remaining words:
['missionary', 'optimum', 'rational', 'search', 'symmetry']

```

```

+---+---+---+---+---+---+---+---+---+---+
| | | | d | g | | l | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| | e | k | e | r | h | i | | | | | |
+---+---+---+---+---+---+---+---+---+---+
| m | l | c | a | a | e | s | g | | | | |
+---+---+---+---+---+---+---+---+---+---+
| i | b | a | d | p | u | p | l | | | | |
+---+---+---+---+---+---+---+---+---+---+
| s | i | r | e | h | r | o | o | | | | a |
+---+---+---+---+---+---+---+---+---+---+
| s | s | t | n | s | i | p | b | | | g | |
+---+---+---+---+---+---+---+---+---+---+
| i | s | k | d | e | s | t | a | | e | | l |
+---+---+---+---+---+---+---+---+---+---+
| o | i | c | | a | t | i | l | n | | a | |
+---+---+---+---+---+---+---+---+---+---+
| n | m | a | | r | i | m | t | | c | | |
+---+---+---+---+---+---+---+---+---+---+
| a | d | b | | c | c | u | | o | | | |
+---+---+---+---+---+---+---+---+---+---+
| r | a | | | h | | m | l | | | | |
+---+---+---+---+---+---+---+---+---+---+
| y | l | a | b | i | n | n | a | c | | | |
+---+---+---+---+---+---+---+---+---+---+

remaining words:
['rational', 'search', 'symmetry']

```

Outputs for wordSearchMaker2.py with 12x12 Grid:

```
-----Rules-----
placed the word: "admissible" at: (0, 0) , direction: (0, 1)
placed the word: "agent" at: (0, 0) , direction: (1, 0)
placed the word: "backtrack" at: (0, 5) , direction: (0, 1)
placed the word: "cannibal" at: (0, 6) , direction: (0, 1)
placed the word: "deadend" at: (0, 7) , direction: (0, 1)
placed the word: "global" at: (0, 1) , direction: (0, 1)
placed the word: "graphsearch" at: (0, 8) , direction: (0, 1)
placed the word: "heuristic" at: (0, 9) , direction: (0, 1)
placed the word: "lisp" at: (0, 10) , direction: (0, 1)
placed the word: "local" at: (0, 11) , direction: (0, 1)
placed the word: "missionary" at: (1, 2) , direction: (0, 1)
placed the word: "optimum" at: (1, 3) , direction: (0, 1)
placed the word: "rational" at: (1, 4) , direction: (0, 1)
placed the word: "search" at: (4, 10) , direction: (0, 1)
placed the word: "symmetry" at: (11, 0) , direction: (1, 0)
-----States-----
```

```
Number of Failures:
0
Number of backtrack calls:
16
```

Outputs for wordSearchMaker2.py with 11x11 Grid:

```
-----Rules-----
placed the word: "admissible" at: (0, 0) , direction: (0, 1)
placed the word: "agent" at: (0, 0) , direction: (1, 0)
placed the word: "backtrack" at: (0, 5) , direction: (0, 1)
placed the word: "cannibal" at: (0, 6) , direction: (0, 1)
placed the word: "deadend" at: (0, 7) , direction: (0, 1)
placed the word: "global" at: (0, 1) , direction: (0, 1)
placed the word: "graphsearch" at: (0, 8) , direction: (0, 1)
placed the word: "heuristic" at: (0, 9) , direction: (0, 1)
placed the word: "lisp" at: (0, 10) , direction: (0, 1)
placed the word: "local" at: (1, 2) , direction: (0, 1)
placed the word: "missionary" at: (10, 3) , direction: (0, -1)
placed the word: "optimum" at: (4, 10) , direction: (0, 1)
placed the word: "rational" at: (1, 4) , direction: (0, 1)
placed the word: "search" at: (5, 8) , direction: (0, 1)
placed the word: "symmetry" at: (10, 0) , direction: (1, 0)
```

```
Number of Failures:
106
Number of backtrack calls:
122
```