# About the Algorithm Package

## **Requirements**

This project was created using:

* Python version >= 3.6.14
* OpenCV version >= 4.5.2.54
* Pygame version >= 2.0.1
* Scipy version >= 1.7.1

**Installation and Running**

* To install the application, use:

*git clone* [*https://github.com/alice-st/DARP-Python.git*](https://github.com/alice-st/DARP-Python.git) *cd DARP-Python  
 ./Dependencies.sh DARP  
 source DARP/bin/activate*

* To run the application, use:

*python3 darpinPoly.py*

**Usage**

* To modify the Grid Dimensions, use:

*python3 darpinPoly.py -grid x y*

where x, y are the desired rows and columns of the Grid respectively (default: 10, 10).

* To modify the number of Robots and their Initial Positions, use:

*python3 darpinPoly.py -in\_pos a b c*

where a, b, c, are the cells' numbers in the Grid (default: 1, 3, 9)   
(row=0,column=0 --> cell=0, row=0,column=1 --> cell=1 etc.)

* To assign different portions to each Robot (not Equal), use:

*python3 darpinPoly.py -nep -portions d e f*

where d e f are the portions assigned to Robots a, b and c respectively. Their sum should be equal to 1. (default: 0.2, 0.3, 0.5)

* To use different positions for the obstacles in the Grid, use:

*python3 darpinPoly.py -obs\_pos o1 o2 o3*

where o1 o2 and o3 are the positions of the obstacles in the Grid. Obstacle positions should not overlap with Robots' initial positions. (default: 5, 6, 7) (row=0,column=0 --> cell=0, row=0,column=1 --> cell=1 etc.)

* To visualize the results, use:

*python3 darpinPoly.py -vis*

**Features:**

* Obstacle resolution: 1x1
* Cell resolution: 0.5x0.5
* Path resolution : 1x1 (since the path obtained is a closed loop! to ensure no backtracking)
* Resolution can be refined if needed. Higher computation power is needed for the path optimization

**STC paths for each robot, ensuring complete area coverage**

### **Mode = 0** (connection on bottom between the side of the branches of the MST)

### **Mode = 1** (connection on top between the side of the branches of the MST)

### 

### **Mode = 2** (connection on right between the side of the branches of the MST)

### **Mode = 3** (connection on left between the side of the branches of the MST)**Extra Material**

Paper: [Zenodo](https://zenodo.org/record/2591050#.YTCvBVtRVH6)

Medium: [Medium](https://medium.com/@athanasios.kapoutsis/darp-divide-areas-algorithm-for-optimal-multi-robot-coverage-path-planning-2fed77b990a3)

GitHub repositories: [Java](https://github.com/athakapo/DARP)

GUI demo: [YouTube](https://www.youtube.com/watch?v=LrGfvma41Ak)

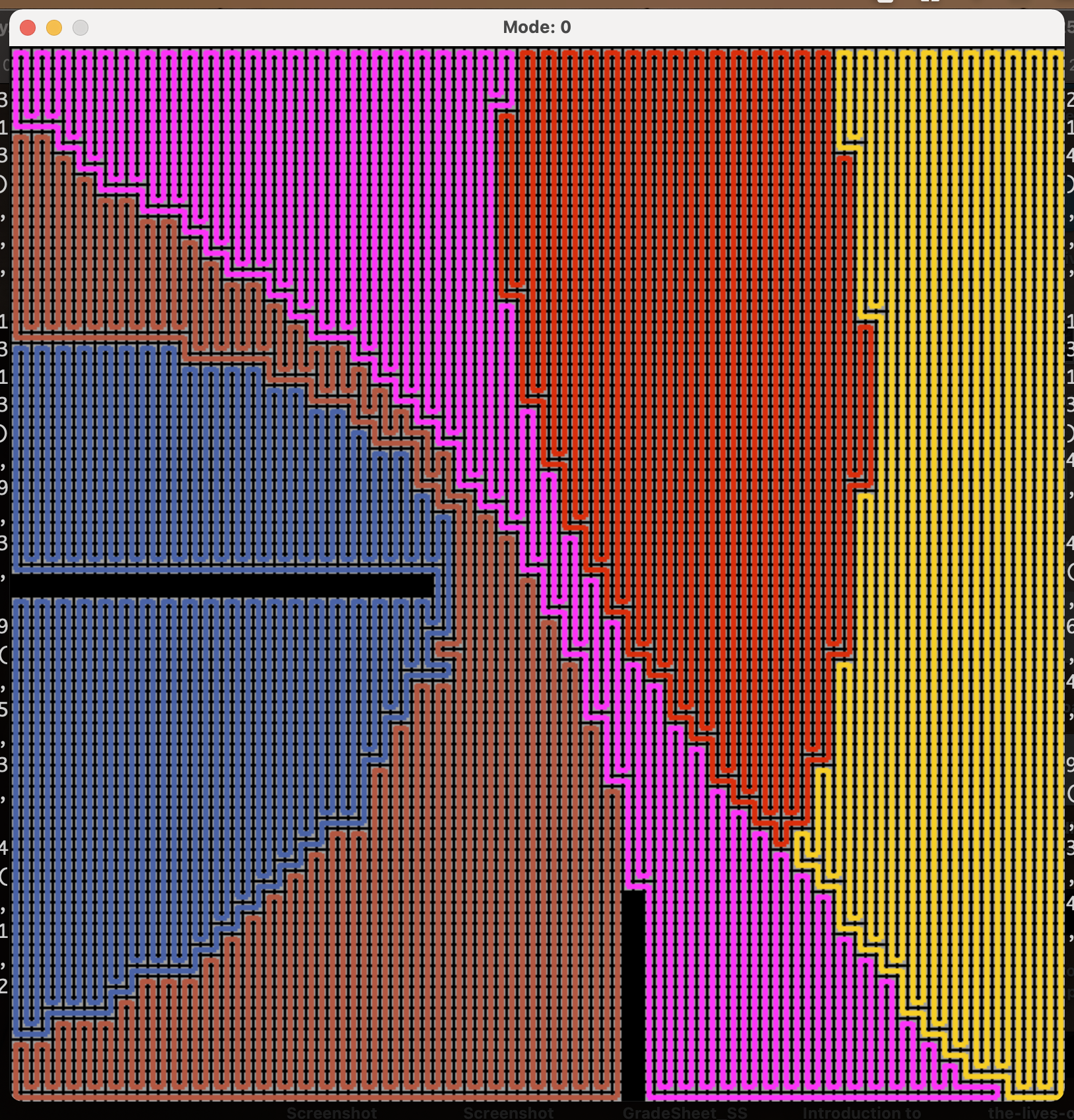
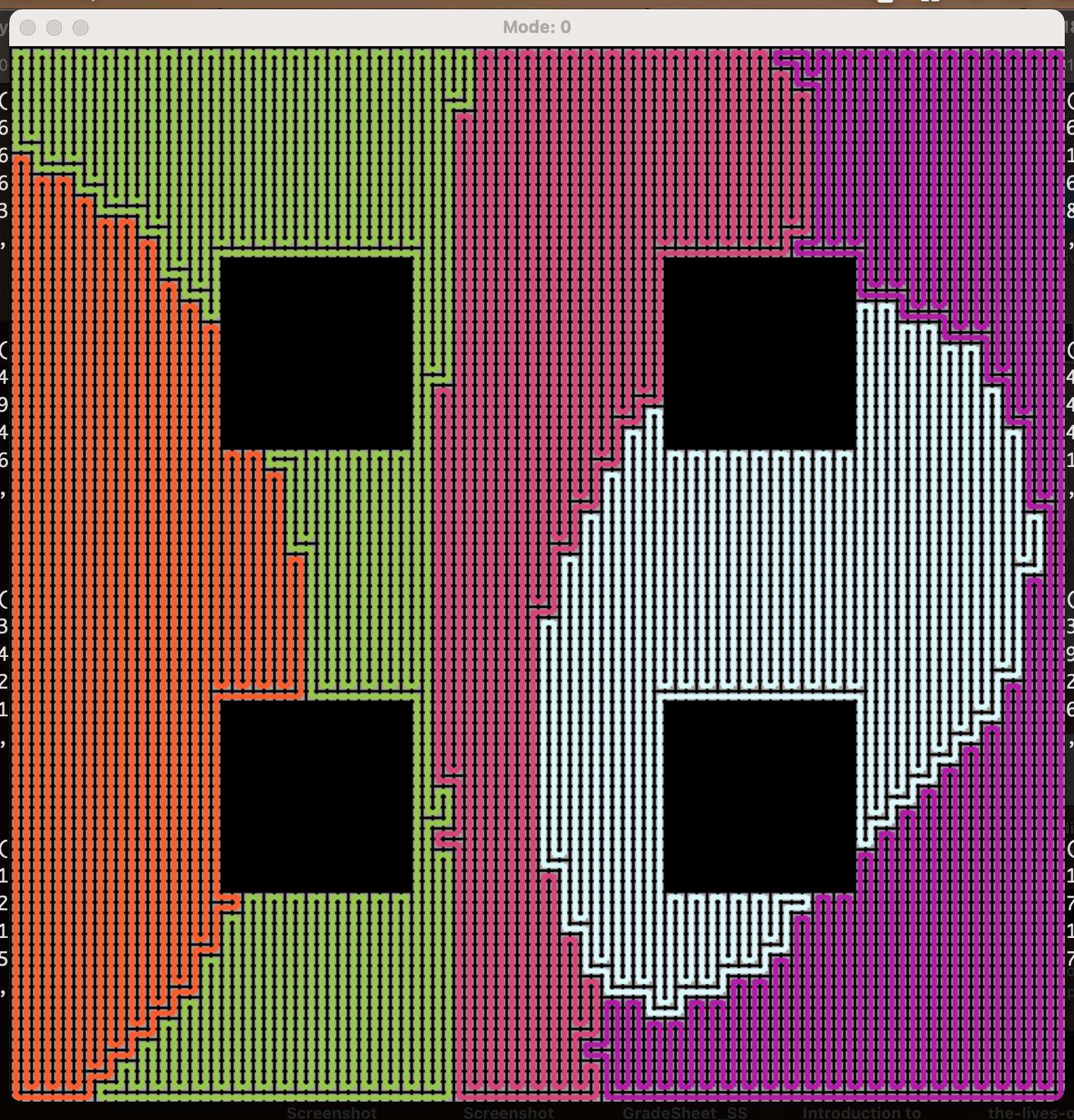
ROS integration: [Wiki](http://wiki.ros.org/area_division)

**Cite as**

@article{kapoutsisdarp, title={DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning},  
 author={Kapoutsis, Athanasios Ch and Chatzichristofis, Savvas A and Kosmatopoulos, Elias B},   
 journal={Journal of Intelligent \& Robotic Systems},  
 pages={1--18},  
 publisher={Springer}}

# Dry Run

# 



# 

# 

# Process

Take 10k random positions such that mutual distance between bots is atleast 10m

Vary parameters so that the simulation does not take too long to run

Summarize simulation results