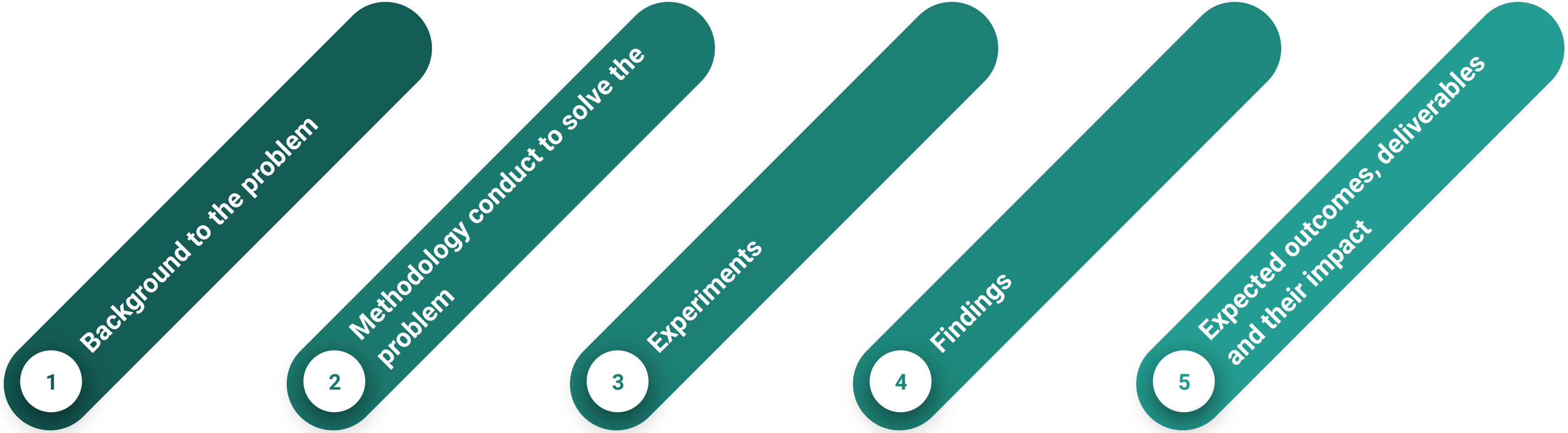
PeraSwarm: Simultaneous Localization and Mapping in Mixed Reality Environment

Group 19

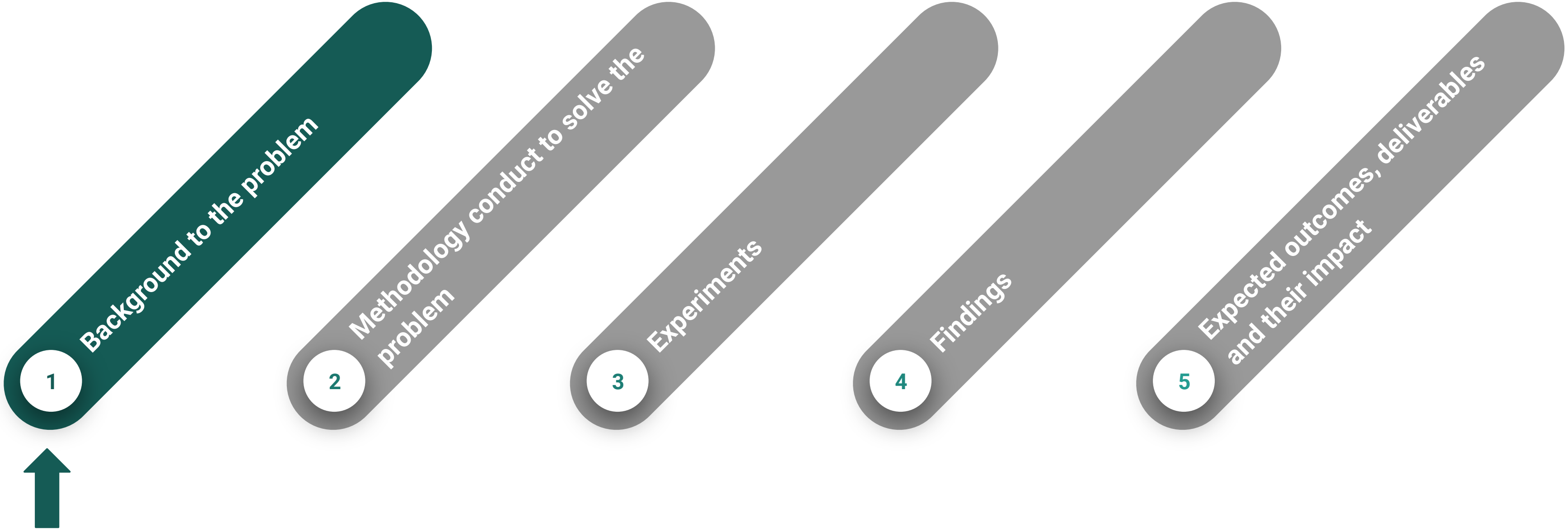
Supervisors:
Prof. Roshan Ragel
Dr. Isuru Nawinne
Mr. Nuwan Jaliyagoda

Team Members:
E/18/077 Nipun Dharmarathne
E/18/150 Yojith Jayarathna
E/18/227 Dinuka Mudalige

Content



1. Background to the problem
2. Methodology conduct to solve the problem
3. Experiments
4. Findings
5. Expected outcomes, deliverables and their impact



1 Background to the problem

2 Methodology conduct to solve the problem

3 Experiments

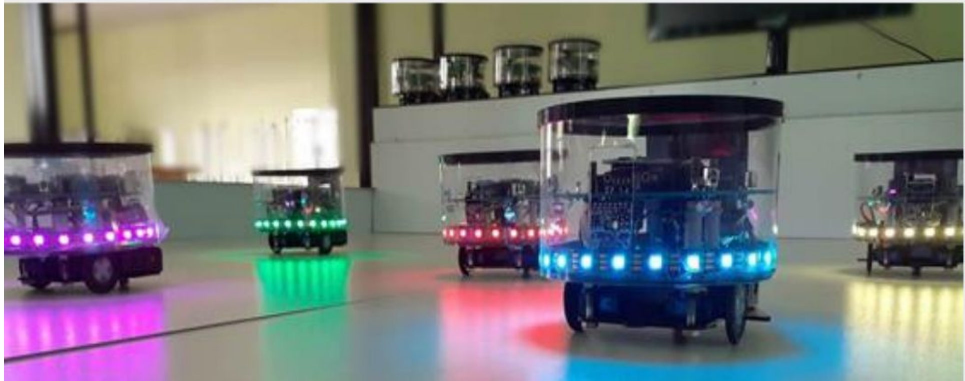
4 Findings

5 Expected outcomes, deliverables and their impact

Swarm Robotics

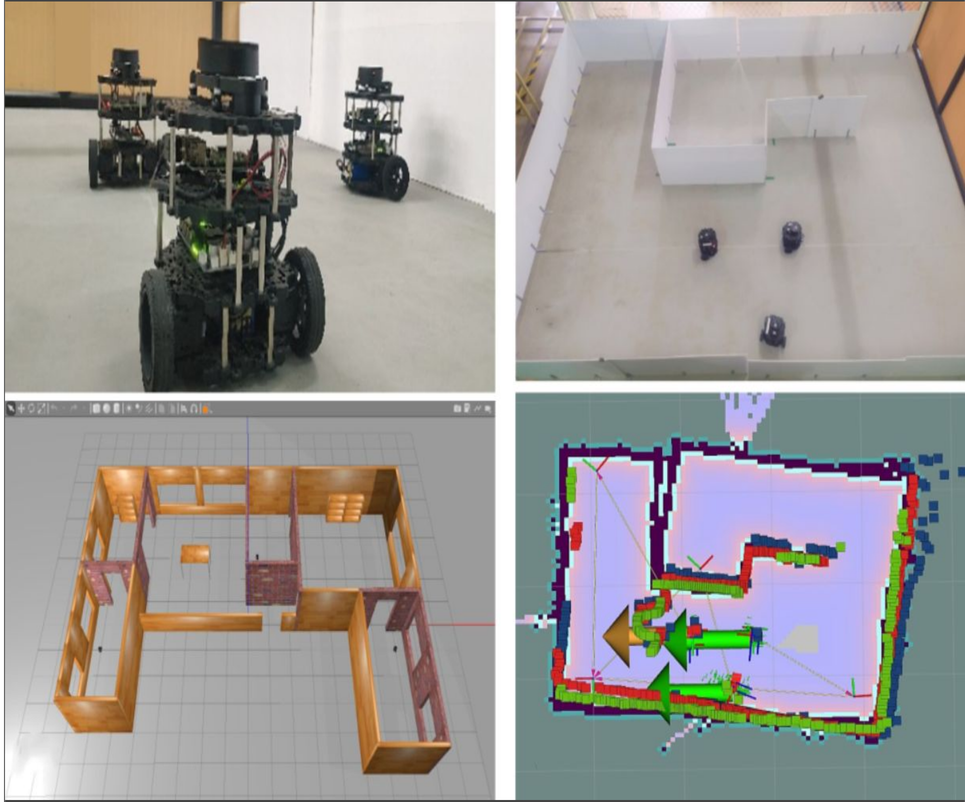
- Multiple robots collaboratively work together to accomplish tasks in a,
    - Distributed
    - Decentralized manner

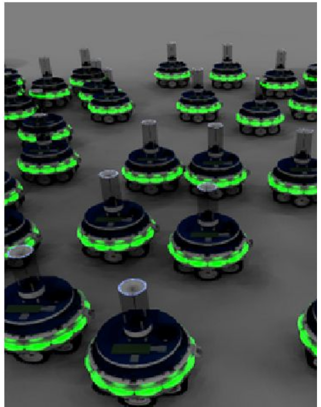
- Inspired by the collective behaviour observed in natural swarms such as,
    - Ants
    - Bees



SLAM with Swarm Robots

- Involves a team of robots working together to map an unknown environment while localizing
- Advantages:
  - Increased efficiency
  - Rapid coverage
  - Distributed exploration



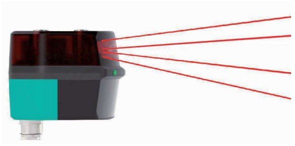


- Features:
    - Decentralized control
    - Self-organizing behavior
    - Robustness
    - Scalability
    - Efficiency

- Applications:
    - Warehouse management
    - Exploration
    - Surveillance
    - Search and rescue
    - Environmental monitoring

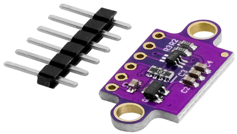
Problems and Proposed Solutions

- Expensive sensors → Cost effective sensors



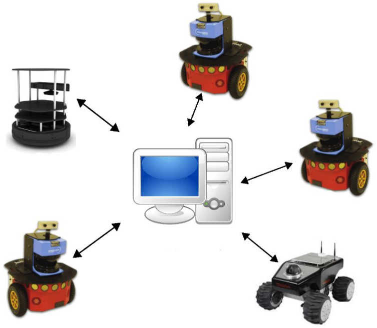
Lidar

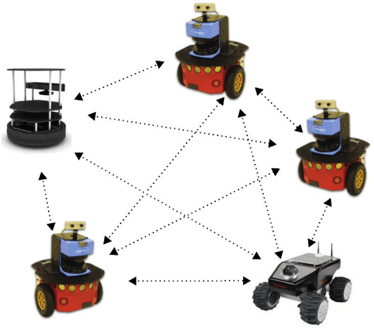
Camera



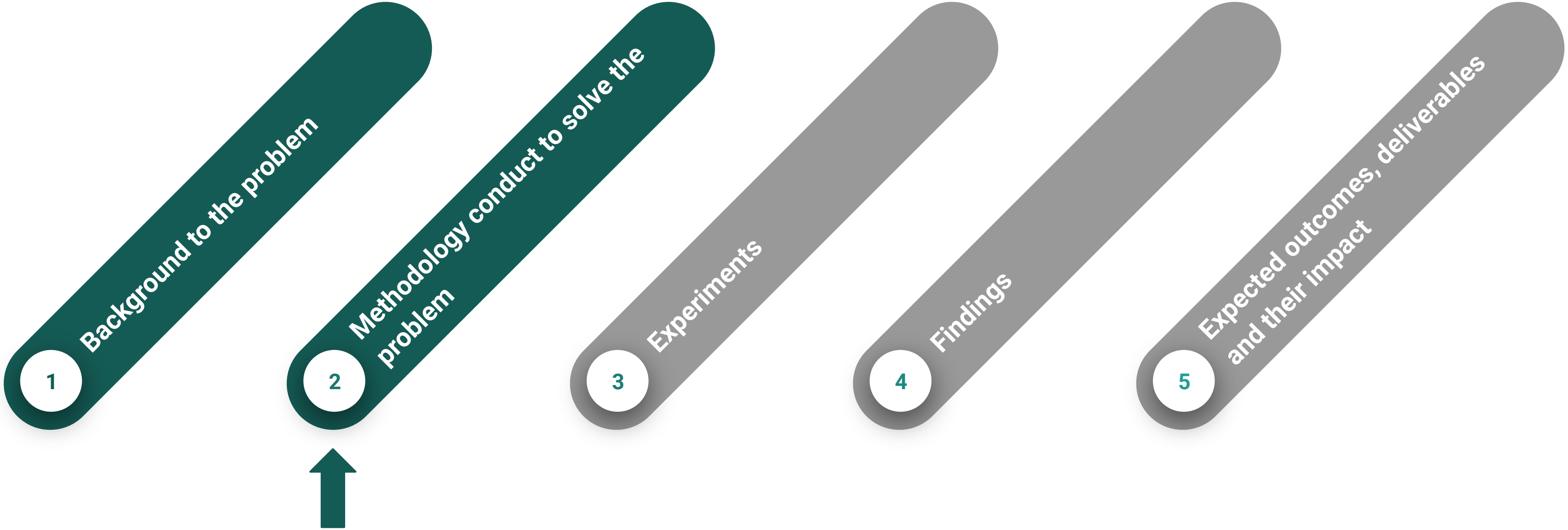
ToF Sensor

- Centralized communication → Decentralized multi-robot communication





- Novel algorithms with more performance enhancements



1 Background to the problem

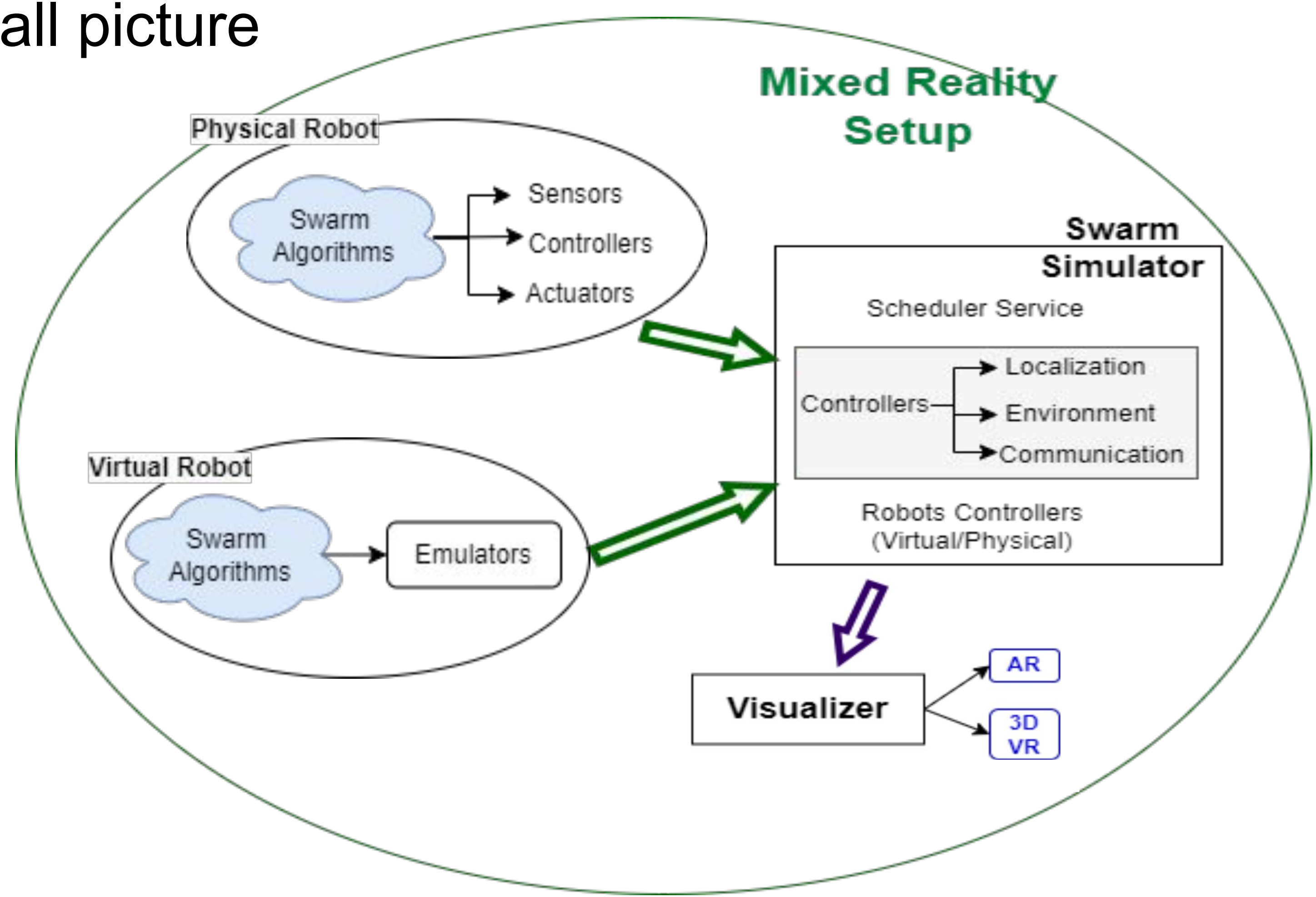
2 Methodology conduct to solve the problem

3 Experiments

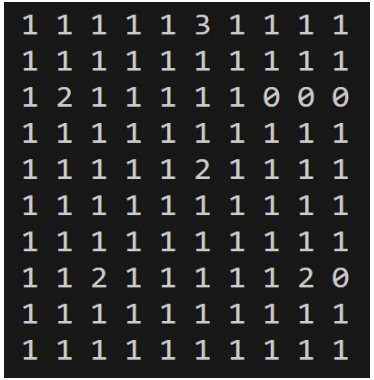
4 Findings

5 Expected outcomes, deliverables and their impact

Overall picture



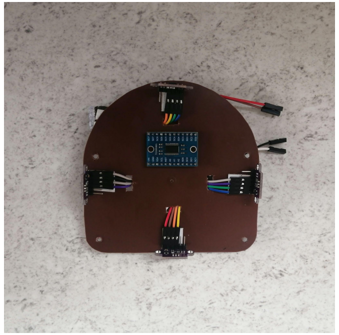
Our Approach

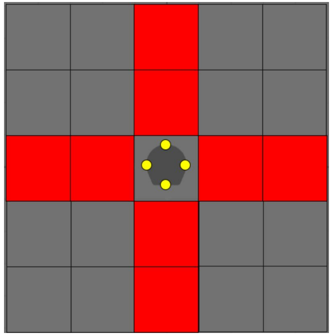


```
1 1 1 1 1 3 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 2 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 2 1 1 1 1 1 2 0
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

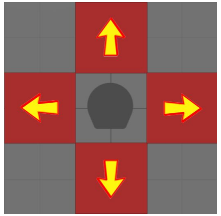
- Occupancy grid representation
- 2D integer array was used to keep track of the map

- Robot has 4 distance sensors faced to
  - North, South, East, West
  - With a range of 2 cells

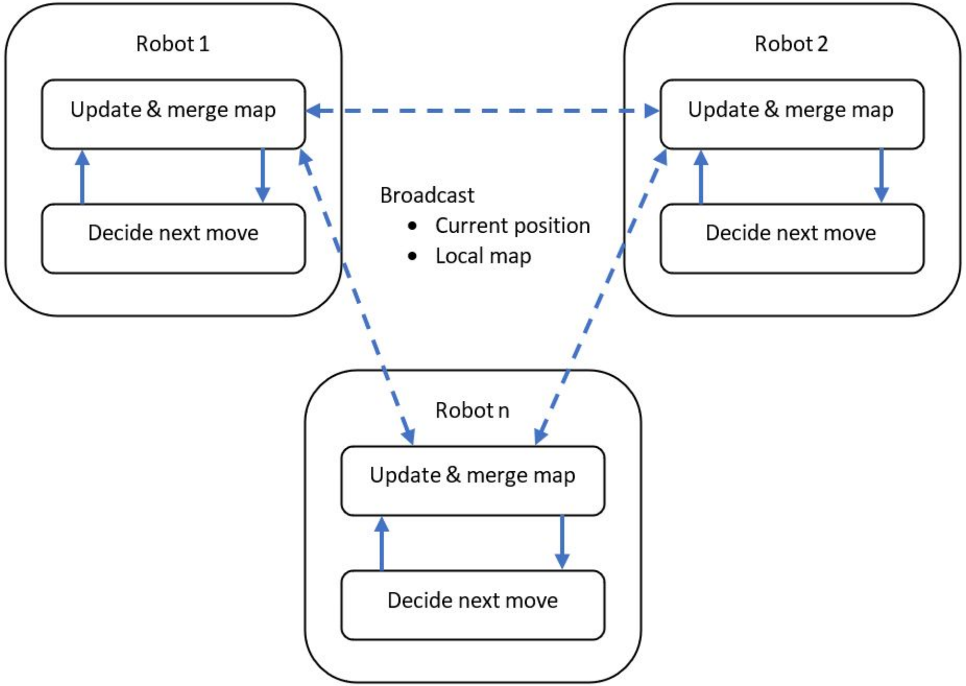




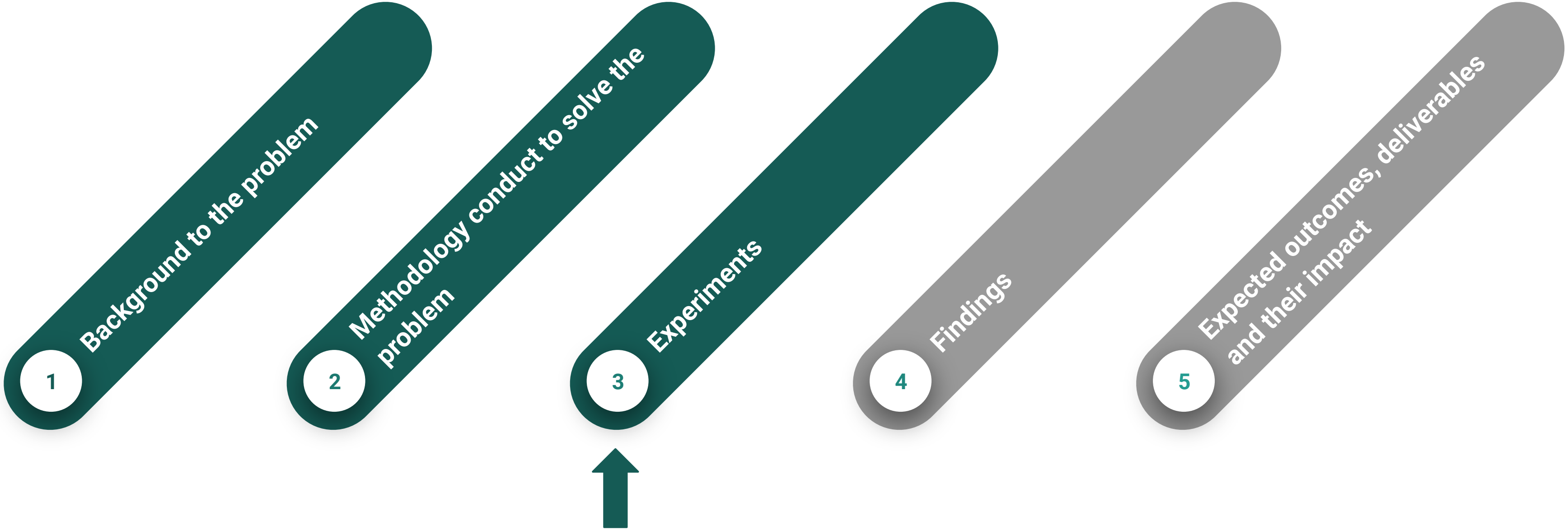
- Assumptions
  - Robots can only move one step at a time to
    - North, South, East, West
  - Initial position and heading directions of robots are known



- For each step robots broadcast,
  - Current position
  - Local map



- Algorithms tested
  a. Random Movement Algorithm
  b. Heuristic Based on Wavefront Algorithm (HWA)
  c. Heuristic Based on Least Cost Estimate (HLCE) - Novel Algorithm
  d. Voronoi Coverage



1 Background to the problem

2 Methodology conduct to solve the problem

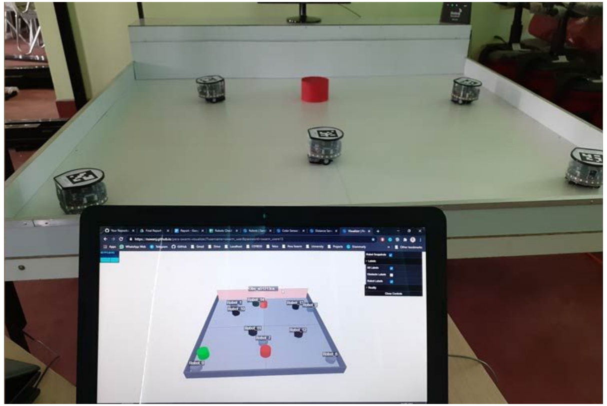
3 Experiments

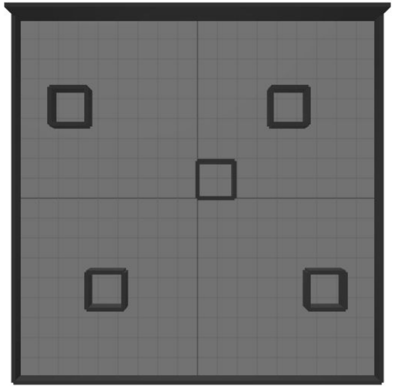
4 Findings

5 Expected outcomes, deliverables and their impact

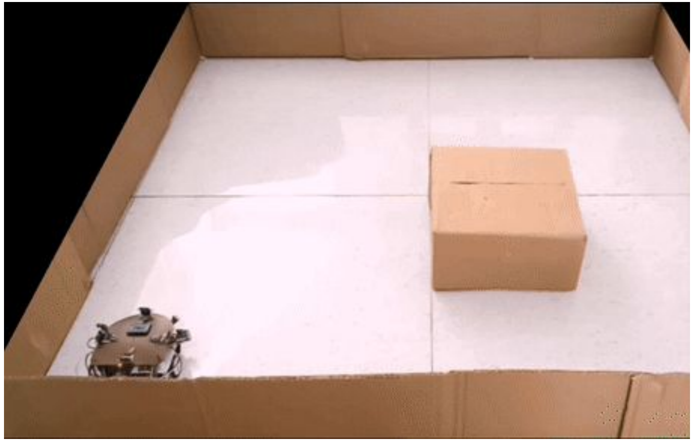
Experiments

- Different arena sizes
  - ***Default Arenas*** (18x18 cell grid)





  - ***Small Test Arenas*** ⇒ mainly for physical robots testing

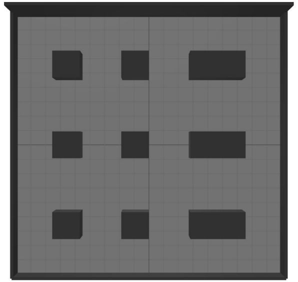


  - ***Large Arenas*** (60x60 cell grid) ⇒ for performance tests with large number of robots

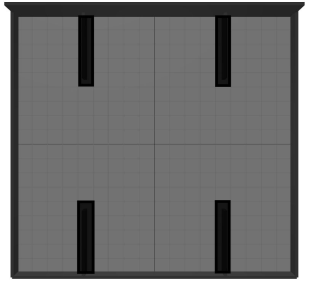
Experiments

- Different arena types

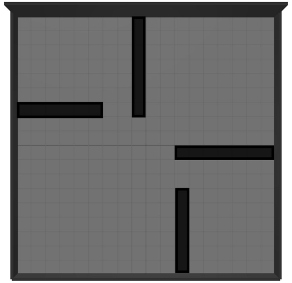
arena_warehouse

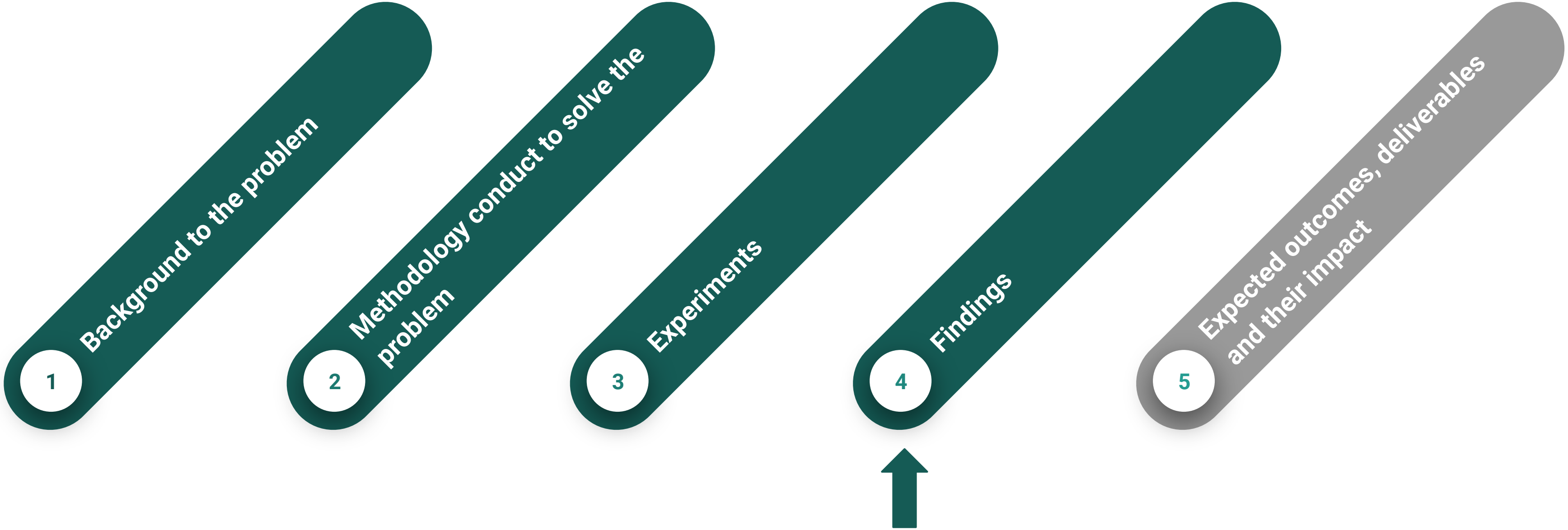


arena_parking_lot



arena_cubicles





1 Background to the problem

2 Methodology conduct to solve the problem

3 Experiments

4 Findings

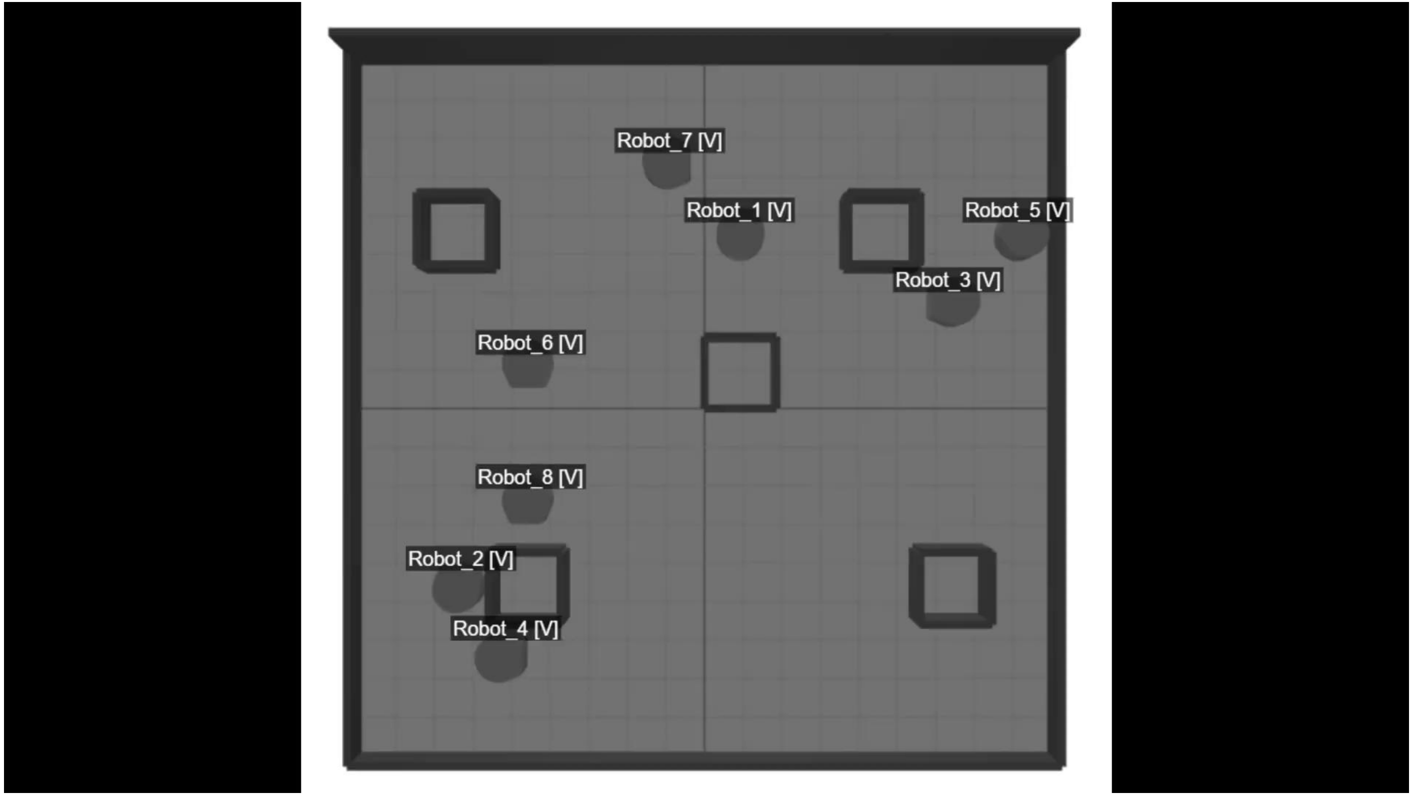
5 Expected outcomes, deliverables and their impact

Findings

- Performance Measurements
  - Full Coverage Time
  - Correctly Explored Probability
  - Stability Comparison

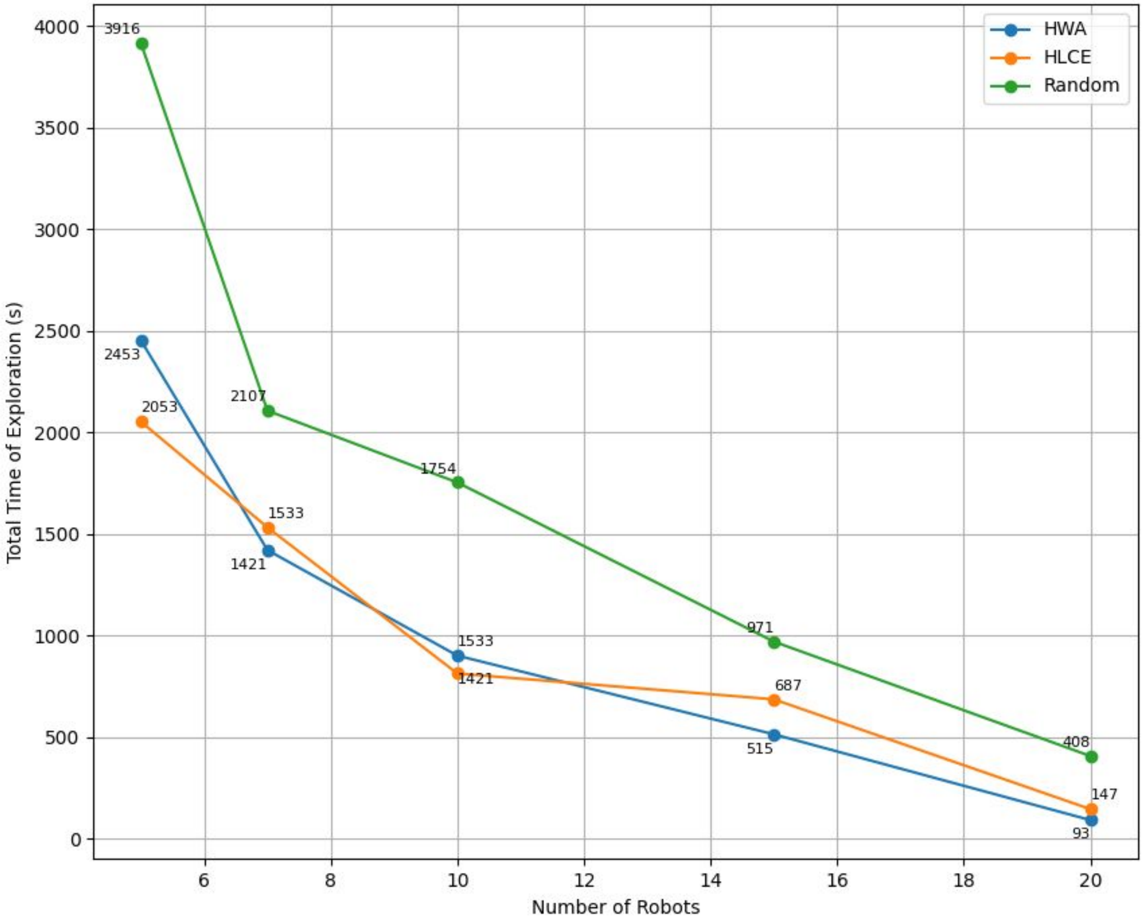
Full Coverage Time

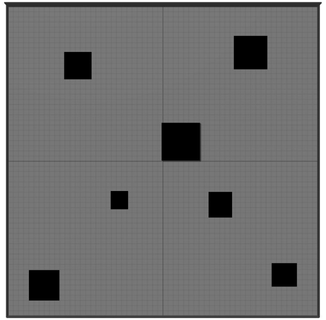
- Time taken for the swarm to cover the entire environment.



Full Coverage Time

HWA → Heuristic Based on Wavefront Algorithm
HLCE → Heuristic Based on Least Cost Estimate



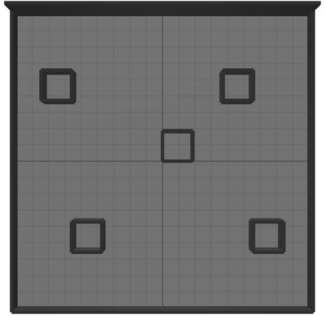


arena_obstacles_large
(60x60 cells)

Probability $_{\textbf{correctly explored}}$ = (correctly explored cells / total cells)%

By comparing generated map with the ground truth

Actual Map



convert

Ground Truth

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Generated Map by Robots

0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	2	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	2	1	0	0	1	1	2	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(after ‘t’ seconds time)

0’s → Unexplored cells

- Free cells: 1
- Obstacles: 2

Example:-

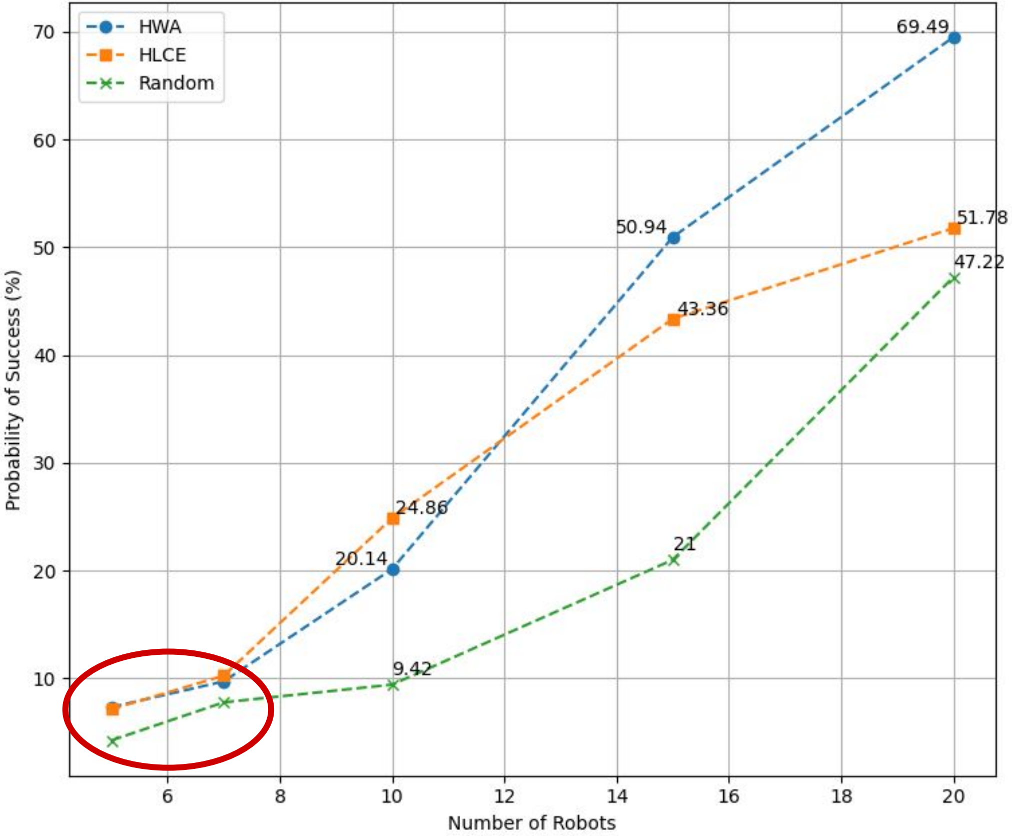
Correctly explored cells = 82
Total Cells = 10x10 = 100

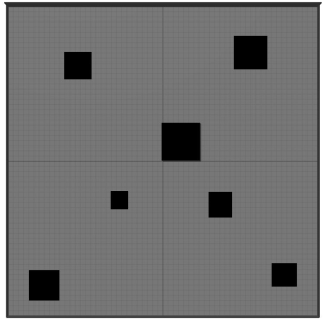
P $_{\textbf{correctly explored}}$ = (72/100)x100% = 82%

Correctly Explored Probability

HWA → Heuristic Based on Wavefront Algorithm
HLCE → Heuristic Based on Least Cost Estimate

For equally biased testing ⇒ applied a time bound of 20% of ***Full Coverage Time***





arena_obstacle_large
(60x60 cells)

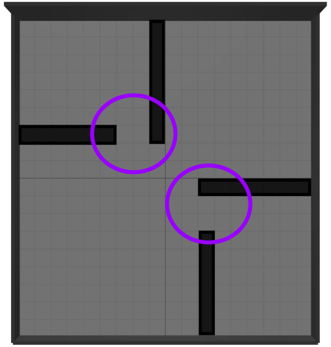
Stability Comparison

Done by calculating the Sample Standard Deviation ($\mathbf{SD}_{\mathbf{sample}}$) of 3 test instances (**n**=3) for the two performance metrics (*Full Coverage Time* and *Correctly Explored Probability*)

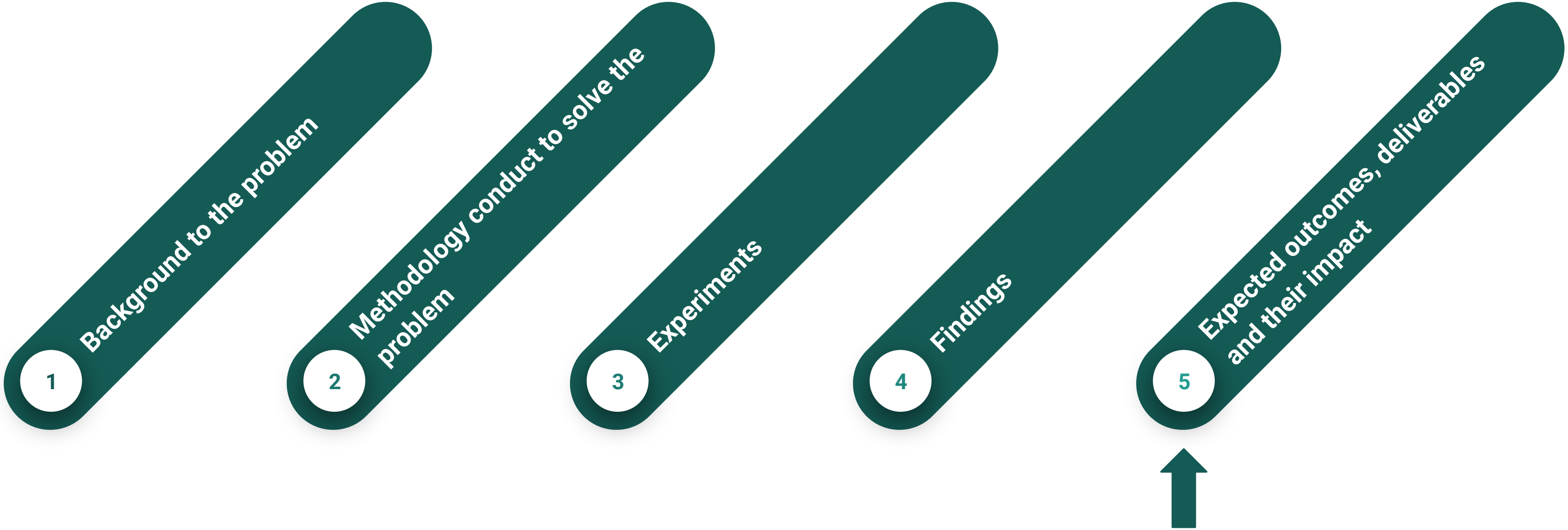
$$\mathrm{SD}_{\text{sample}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2}$$

Standard Deviation ↓ → Variability ↓ → Can guarantee higher accuracies

Example:-



Environment	Algorithm	SD of Full Coverage Time	SD of Correctly Explored Probability
arena_cubicles	Random	25.47	2.84
	HLCE	6.74	1.35
	HWA	7.18	3.36



1 Background to the problem

2 Methodology conduct to solve the problem

3 Experiments

4 Findings

5 Expected outcomes, deliverables and their impact

Expected outcomes, deliverables and their impact

Expected outcomes and deliverables:

- A functional multi-agent swarm robotic system capable of performing SLAM
- Algorithms for decentralized decision-making and task allocation
- Communication infrastructure for swarm coordination

Impact:

- Cost reduction in swarm SLAM research:
    - Lower experimental and development costs
    - Preservation of swarm intelligence experiment scale
- Optimization of warehouse management:
    - Improved mapping and navigation
    - Enhanced inventory logistics
- Advancement in search and rescue operations:
    - Improved mapping of disaster areas
    - Enhanced victim location capabilities
    - Better navigation in hazardous terrains

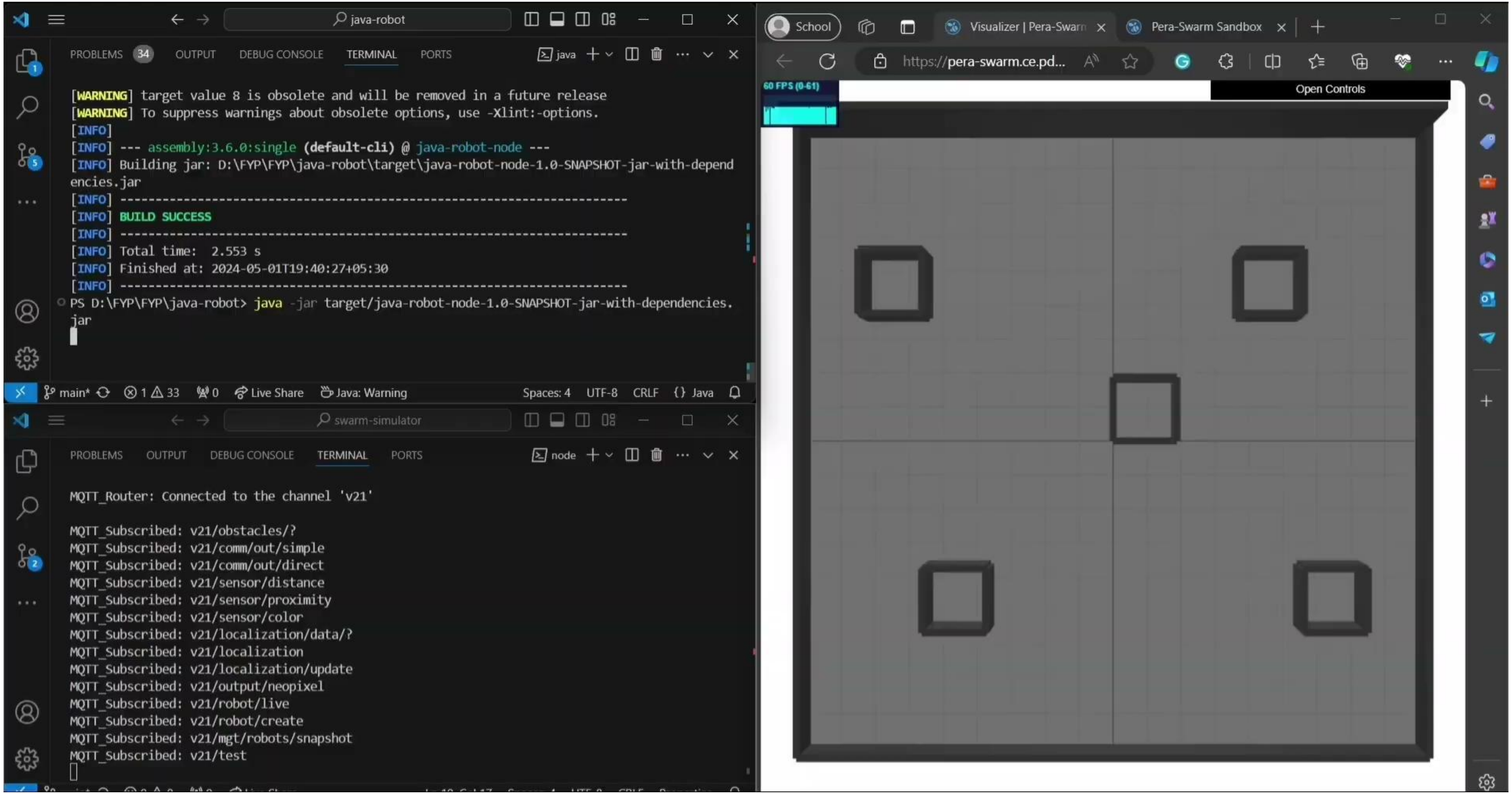
Thank You !!!

Q & A

Demonstration

1. Virtual Robots
   a. Random Movement Algorithm
   b. HLCE
   c. HWA
   d. Voronoi Coverage
2. Physical Robots
   a. HLCE
   b. MQTT Communication Modifications

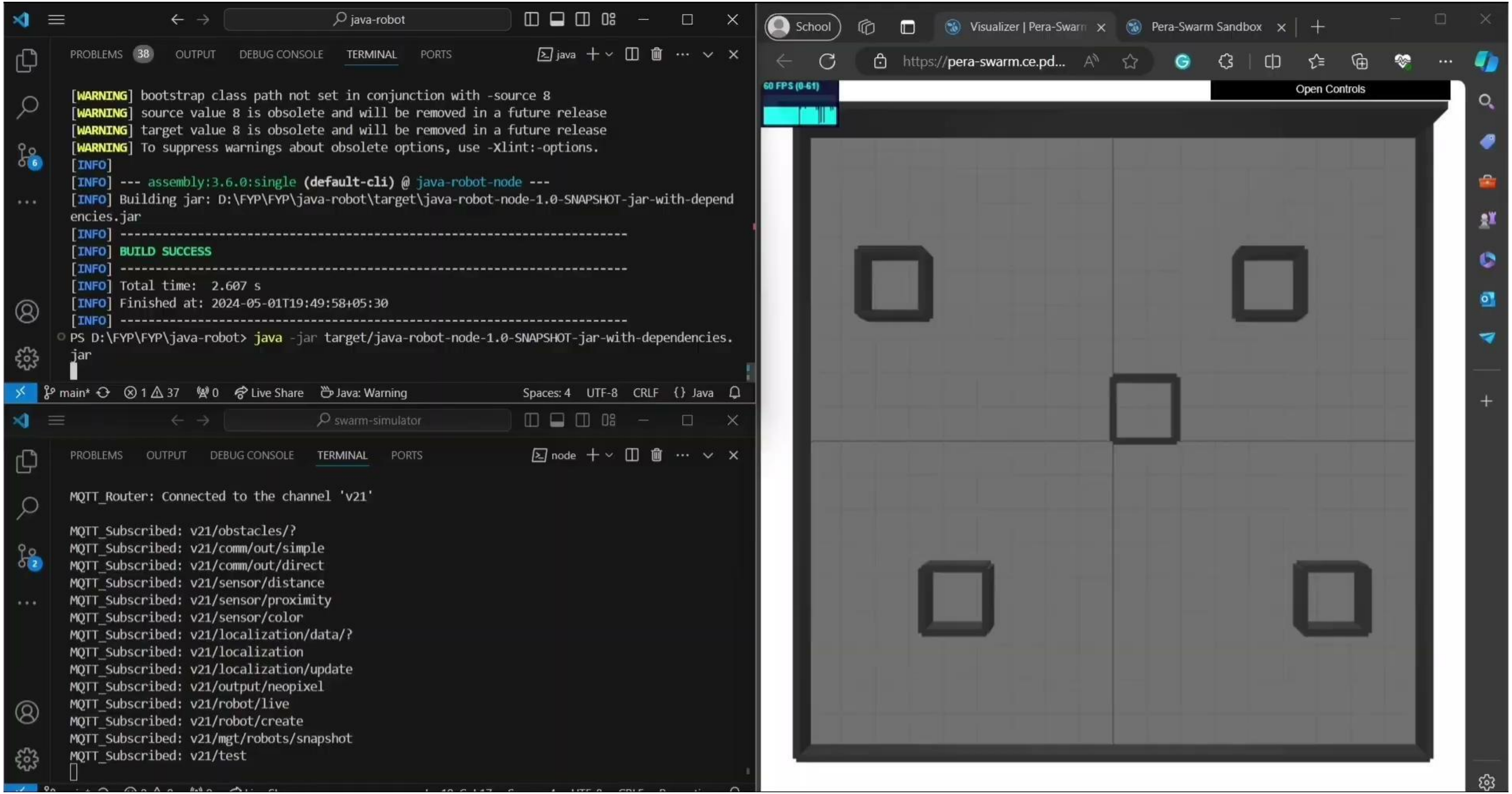
- Demonstration: Random movement algorithm



```
[WARNING] target value 8 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- assembly:3.6.0:single (default-cli) @ java-robot-node ---
[INFO] Building jar: D:\FYP\FYP\java-robot\target\java-robot-node-1.0-SNAPSHOT-jar-with-dependencies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.553 s
[INFO] Finished at: 2024-05-01T19:40:27+05:30
[INFO] ------------------------------------------------------------------------
PS D:\FYP\FYP\java-robot> java -jar target/java-robot-node-1.0-SNAPSHOT-jar-with-dependencies.jar
```

```
MQTT_Router: Connected to the channel 'v21'

MQTT_Subscribed: v21/obstacles/?
MQTT_Subscribed: v21/comm/out/simple
MQTT_Subscribed: v21/comm/out/direct
MQTT_Subscribed: v21/sensor/distance
MQTT_Subscribed: v21/sensor/proximity
MQTT_Subscribed: v21/sensor/color
MQTT_Subscribed: v21/localization/data/?
MQTT_Subscribed: v21/localization
MQTT_Subscribed: v21/localization/update
MQTT_Subscribed: v21/output/neopixel
MQTT_Subscribed: v21/robot/live
MQTT_Subscribed: v21/robot/create
MQTT_Subscribed: v21/mgt/robots/snapshot
MQTT_Subscribed: v21/test
```

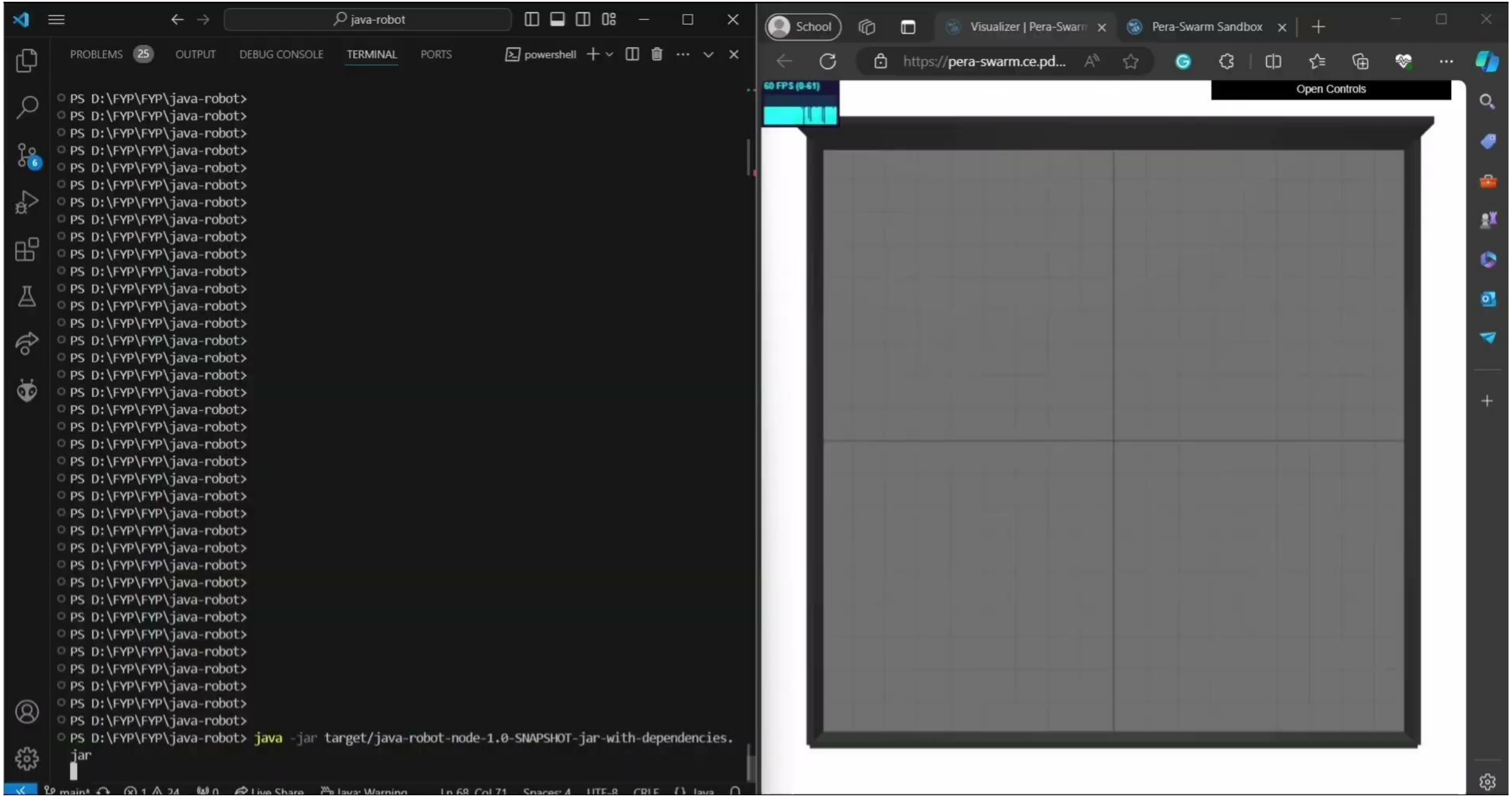
- Demonstration: HLCE



```
[WARNING] bootstrap class path not set in conjunction with -source 8
[WARNING] source value 8 is obsolete and will be removed in a future release
[WARNING] target value 8 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- assembly:3.6.0:single (default-cli) @ java-robot-node ---
[INFO] Building jar: D:\FYP\FYP\java-robot\target\java-robot-node-1.0-SNAPSHOT-jar-with-depend
encies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.607 s
[INFO] Finished at: 2024-05-01T19:49:58+05:30
[INFO] ------------------------------------------------------------------------
PS D:\FYP\FYP\java-robot> java -jar target/java-robot-node-1.0-SNAPSHOT-jar-with-dependencies.
jar
```

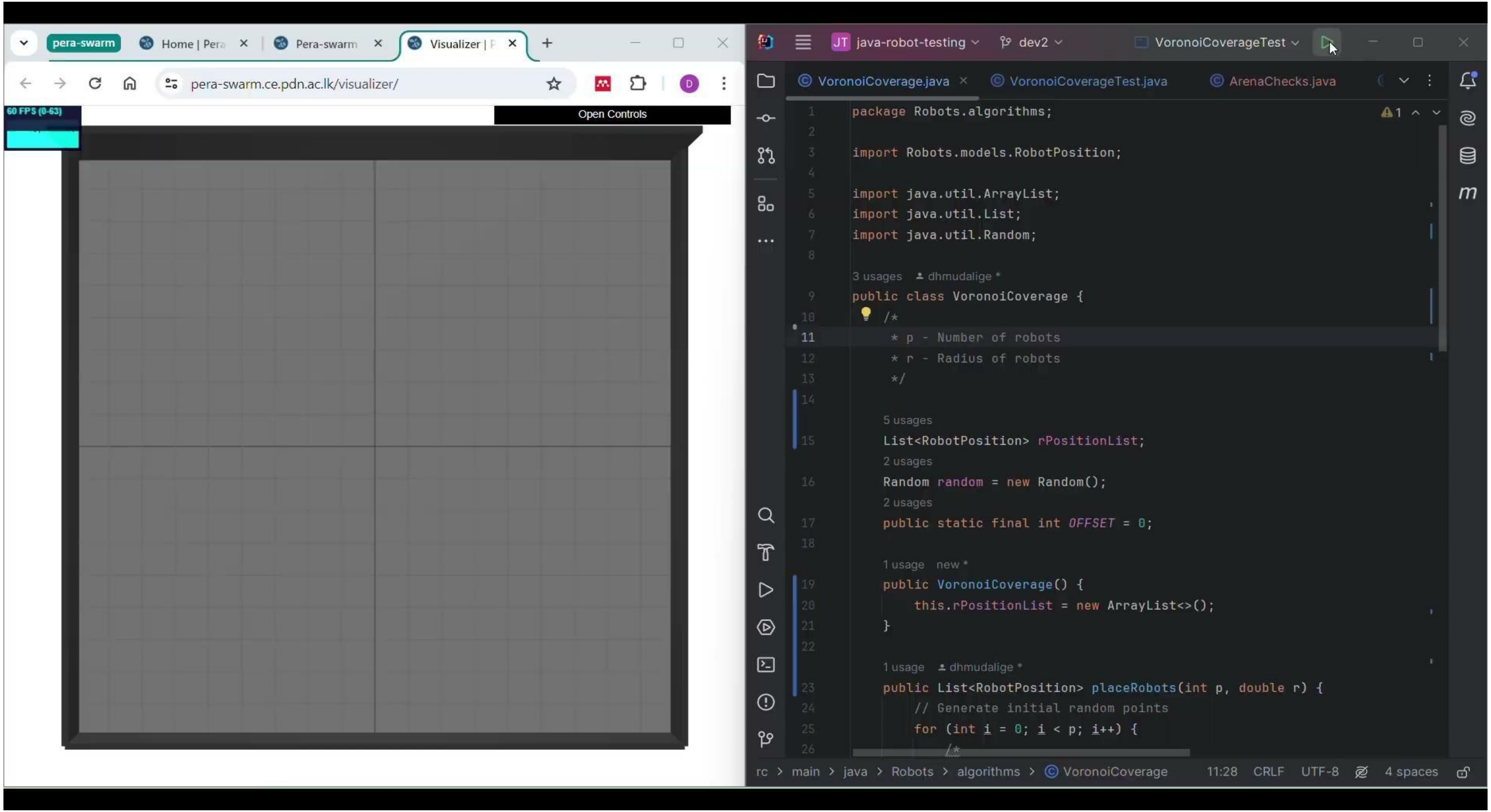
```
MQTT_Router: Connected to the channel 'v21'

MQTT_Subscribed: v21/obstacles/?
MQTT_Subscribed: v21/comm/out/simple
MQTT_Subscribed: v21/comm/out/direct
MQTT_Subscribed: v21/sensor/distance
MQTT_Subscribed: v21/sensor/proximity
MQTT_Subscribed: v21/sensor/color
MQTT_Subscribed: v21/localization/data/?
MQTT_Subscribed: v21/localization
MQTT_Subscribed: v21/localization/update
MQTT_Subscribed: v21/output/neopixel
MQTT_Subscribed: v21/robot/live
MQTT_Subscribed: v21/robot/create
MQTT_Subscribed: v21/mgt/robots/snapshot
MQTT_Subscribed: v21/test
```

- Demonstration: HWA



```
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot>
PS D:\FYP\FYP\java-robot> java -jar target/java-robot-node-1.0-SNAPSHOT-jar-with-dependencies.
jar
```

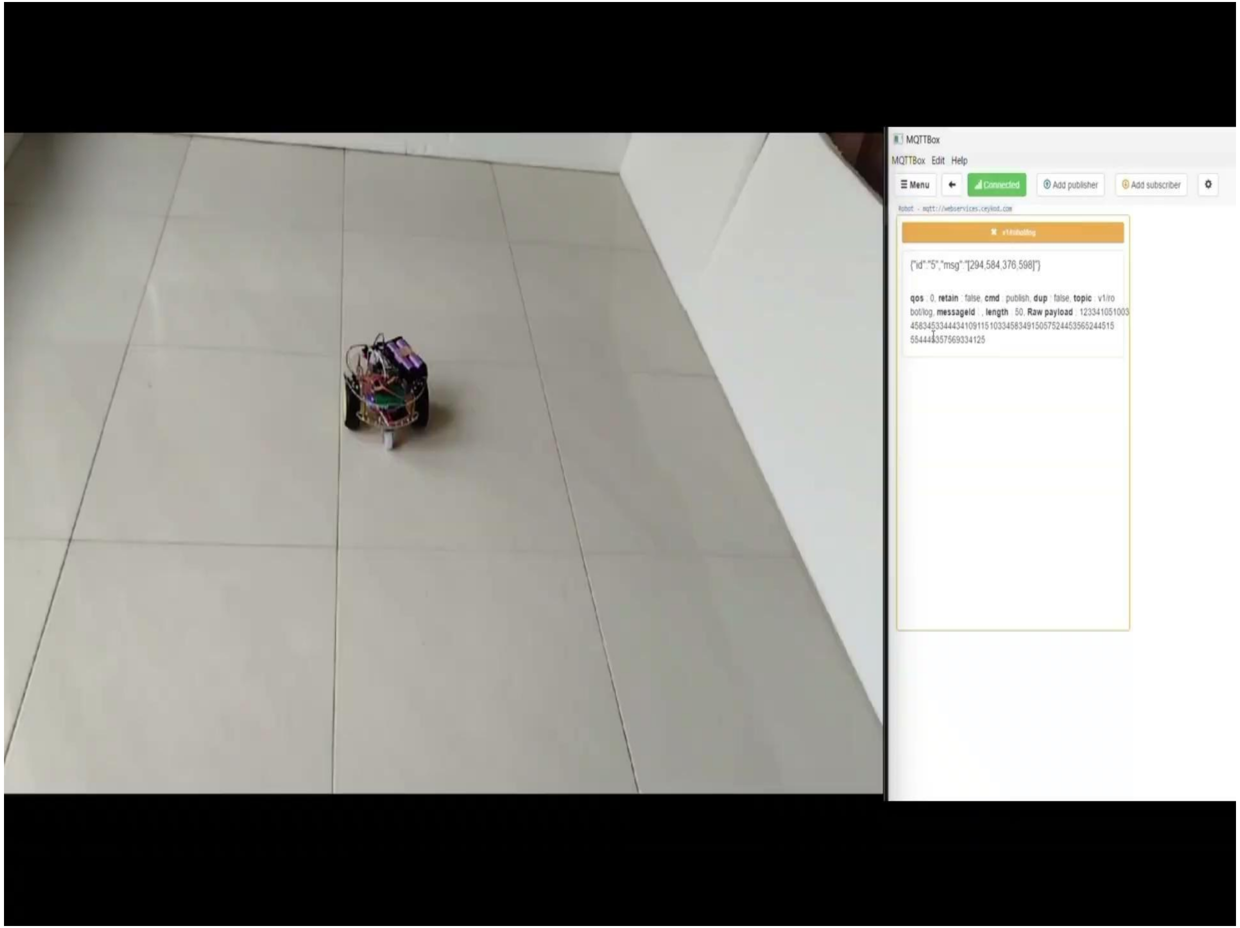
- Demonstration: Voronoi coverage



```java
package Robots.algorithms;

import Robots.models.RobotPosition;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class VoronoiCoverage {
    /*
     * p - Number of robots
     * r - Radius of robots
     */

    List<RobotPosition> rPositionList;
    Random random = new Random();
    public static final int OFFSET = 0;

    public VoronoiCoverage() {
        this.rPositionList = new ArrayList<>();
    }

    public List<RobotPosition> placeRobots(int p, double r) {
        // Generate initial random points
        for (int i = 0; i < p; i++) {
            /*
```

- Demonstration: HLCE with physical robot

- Demonstration: MQTT connectivity of the Test Robot



{"id":"5","msg":"[294,584,376,598]"}

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : v1/robot/log, **messageId** : , **length** : 50, **Raw payload** : 12334105100345834533444341091151033458349150575244535652445155444535756933412 5