



HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI



DESIGN AND DEVELOPMENT OF ● HOMOGENEIOUS SWARM OF COOPERATIVE ROBOTS IN VIRTUAL ENVIRONMENT

- INTRODUCTION



● INTRODUCTION

Inspiration



MOTIVATION



- **Improved performance:** if tasks can be decomposable then by using parallelism, groups can make tasks to be performed more efficiently.
- **Task enablement:** groups of robots can do certain tasks that are impossible for a single robot.
- **Distributed sensing:** the range of sensing of a group of robots is wider than the range of a single robot.

OBJECTIVES



CHALLENGES

- **Interference:** robots in a group can interfere between them, due to collisions, occlusions, and so forth.
- **Uncertainty concerning other robots' intentions:** coordination requires to know what other robots are doing.

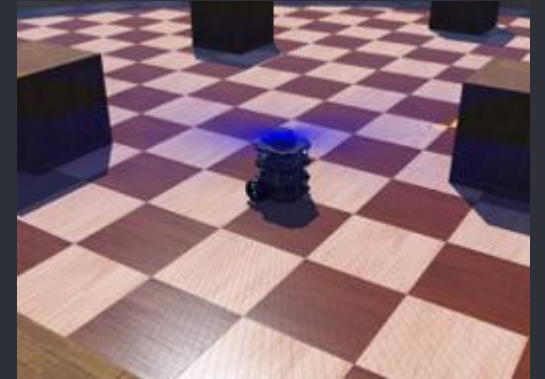


WEBOTS

Webots is a free and open-source 3D robot simulator used in industry, education and research.

TurtleBot

TurtleBot is a low-cost, personal robot kit with open-source software. The TurtleBot's core technology is SLAM and Navigation, making it suitable for service robots.



Review of literature

TABLE 1

Title	Author	Joural/ publication	Inference
Leader Follower formation for UAV Robot Swarm based on Fuzzy Logic Theory	Wilson O. Quesada, Jonathan I. Rodriguez, Juan C. Murillo, Gustavo A. Cardona, David Yanguas-Rojas, Luis G. Jaimes, and Juan M. Calder´ on	Springer International Publishing AG, part of Springer Nature 2018 L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10842, pp. 740–751, 2018	Results depicted a robot swarm showing some bio-inspired behaviors, such as swarm agents surrounding the leader when it is in a static position or when it is traveling from one place to another place.
LINK: https://repository.usta.edu.co/bitstream/handle/11634/18031/127.pdf?sequence=1			

● **TABLE 2**

Title	Author	Journal/ publication	Inference
WebotsTM: Professional Mobile Robot Simulation	Olivier Michel	International Journal of Advanced Robotic Systems, Volume 1 Number 1 (2004)	can equip each robot with a large number of available sensors and actuators. can program these robots using your favorite development environment, simulate them and optionally transfer the resulting programs onto your real robots.
LINK: https://journals.sagepub.com/doi/pdf/10.5772/5618			

● **TABLE 3**

Title	Author	Journal/ publication	Inference
Turtlebot 3 as a robotics education platform	Robin Amsters, Peter Slaets	October 25, 2019	Designing and functioning of turtlebot 3
LINK: https://www.researchgate.net/publication/335023861 Turtlebot 3 as a Robotics Education Platform			

TABLE 4

Title	Author	Journal/ publication	Inference
An Introduction to Swarm Robotics	Iñaki Navarro and Fernando Matía	Hindawi Publishing Corporation ISRN Robotics Volume 2013, Article ID 608164, 10 pages	The main tasks and experimental results in swarm robotics has been explained with future research platforms and robots mentioned clearly. The major drawbacks and benefits helps in understanding the real world implications.
LINK: http://dx.doi.org/10.5402/2013/608164			

TABLE 5

Title	Author	Journal/ publication	Inference
<p>ADVANCED COMMUNICATION PROTOCOLS FOR SWARM ROBOTICS: A SURVEY</p>	<p>Emaad Mohamed H. Zahugi, S.V.A.V. Prasad and T.V. Prasad</p>	<p>2012</p>	<p>Understood the various modes of transmission of data and under which best possible conditions they are used.</p>
<p>LINK: https://www.researchgate.net/publication/280112433_Advanced_Communication_Protocols_For_Swarm_Robotics_A_Survey </p>			

TABLE 6

Title	Author	Journal/ publication	Inference
Tracking Algorithm Using Leader Follower Approach for Multi Robots	B Madhevan, M Sreekumar	2013	The task flow, path flow and the overall controller process is clear and the algorithm process is successfully explained.
LINK: https://www.sciencedirect.com/science/article/pii/S1877705813017384			

TABLE 7

Title	Author	Journal/ publication	Inference
<p>Robot Swarm Communication Networks:: Architecture, Protocols and Application</p>	<p>Ming Li, Kejie Lu, Hua Zhu, Min Chen, Shiwen Mao, B.Prabhakaran</p>	<p>(Not mentioned)</p>	<p>In this paper, wireless intercommunication between swarm robots was established using ROBOTRAK equipped with wifi as well as GPS.</p>
<p>LINK: https://www.eng.auburn.edu/~szm0001/papers/LiCHINACOM08.pdf</p>			

● **TABLE 8**

Title	Author	Journal/ publication	Inference
Collaborative multi-robot exploration	Wolfram Burgard, Marks Moors, Dieter Fox, Reid Simmons, Sebastian Thrun	IEEE International Conference on Robotics and Automation (ICRA), 2000	This technique for target allocation to multiple robots and determining the cost of reaching the target point in minimum time was successfully implemented.
LINK: https://www.researchgate.net/publication/3847261_Collaborative_multi-robot_exploration			

● **TABLE 9**

Title	Author	Journal/ publication	Inference
Homogeneous Swarm Robot Exploration	W.A.F.W Othman, M.A Rosli, A.A.A. Wahab, S.S.N Alhady	International Journal of Trend in Scientific Research and Development (IJTSRD) ISSN: 2456-6470 volume 2 Sept-Oct 2018	This study has succeeded in investigating the exploration technique using homogeneous multi agent system.
LINK: https://www.researchgate.net/publication/333690029_Homogeneous_Swarm_Robots_Exploration			

● **TABLE 10**

Title	Author	Joural/ publication	Inference
Swarm Robots	Erol S,ahin , Sertan Girgin , Levent Bayındır and Ali Emre Turgut	January , 2008	Brief review of swarm robotics as a new approach to the control and coordination of multi-robot systems is studied,
LINK: https://www.researchgate.net/publication/226173060_Swarm_Robotics			

● **TABLE 11**

Title	Author	Journal/ publication	Inference
<p>Swarm Robotics: From Sources of Inspiration to Domains of Application</p>	<p>Erol S,ahin</p>	<p>January 2005</p>	<p>Define the newly emerging field of swarm robotics as a new approach to the control and coordination of multi-robot systems.</p> <p>Got the knowledge of Biomimicry involved in Swarm Robotics</p>
<p>LINK: https://www.researchgate.net/publication/225703852_Swarm_Robotics_From_Sources_of_Inspiration_to_Domains_of_Application/link/546f5abb0cf24af340c08747/download</p>			

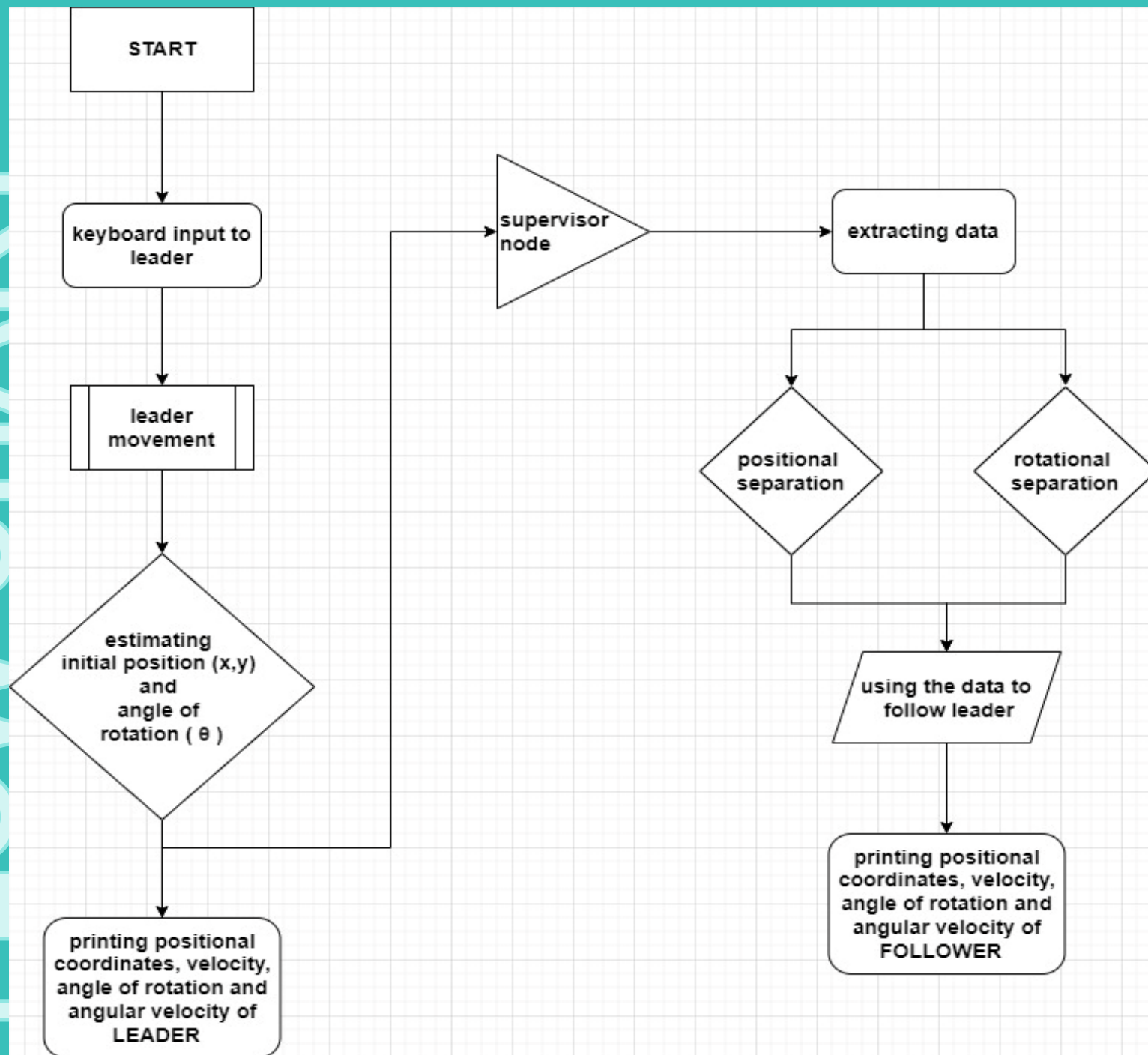
TABLE 12

Title	Author	Journal/ publication	Inference
Swarm robots' communication and cooperation in motion planning	Khiem N.Doan, An T. Le, Than D. Le and Nauth Peter	January 2017	<p>Overlooked Kinematic models and obstacle avoiding algorithms.</p> <p>Overlooked experiments and results on setup, connecting (synchronize),path planning of robots.</p>
LINK: https://www.researchgate.net/publication/306386315 Swarm Robots' Communication and Cooperation in Motion Plannin g			

● Important points extracted from literature review

- Brief review of swarm robotics as a new approach to the control and coordination of multi-robot systems is studied. - PAPER 10
- Got the knowledge of Biomimicry involved in Swarm Robotics - PAPER 11
- Knowledge on desirable properties and defining characteristics of Swarm robots is gained. PAPER 11
- The task flow, path flow and the overall controller process is clear and the algorithm process is studied. PAPER -6
- Wireless intercommunication between swarm robots was established using ROBOTRAK equipped with wi-fi as well as GPS has been studied. - PAPER 7
- Understood the various modes of transmission of data and under which best possible conditions they are used. - PAPER 5
- Details and specs of turtlebot 3 is gone through. - PAPER 3
- Overlooked Kinematic models and obstacle avoiding algorithms. PAPER - 12
- investigating the exploration technique using homogeneous multi agent system. - PAPER 9

FLOW CHART



CODES

- `turtle_node = robot.getFromDef('master')`
`turtle1_node = robot.getFromDef('slave')`
: defining master and slave turtle bot respectively
- `translation_field=turtle_node.getField('translation')`
`slave_field=turtle1_node.getField('translation')`
: defining translational attribute of master and slave

CODES

- `robot.getName()=='master'`
: *extracting the name of the robot from respective controller*
- `(key==Keyboard.CONTROL+ord('W'))`
: *key input for the movement of the bot*

CODES

- `position=translation_field.getSFVec3f()`
`position2 = turtle1_node.getPosition()`
: extracting positional details of master as well as slave bot
- `slave_field.setSFVec3f(position2)`
: setting position for slave bot

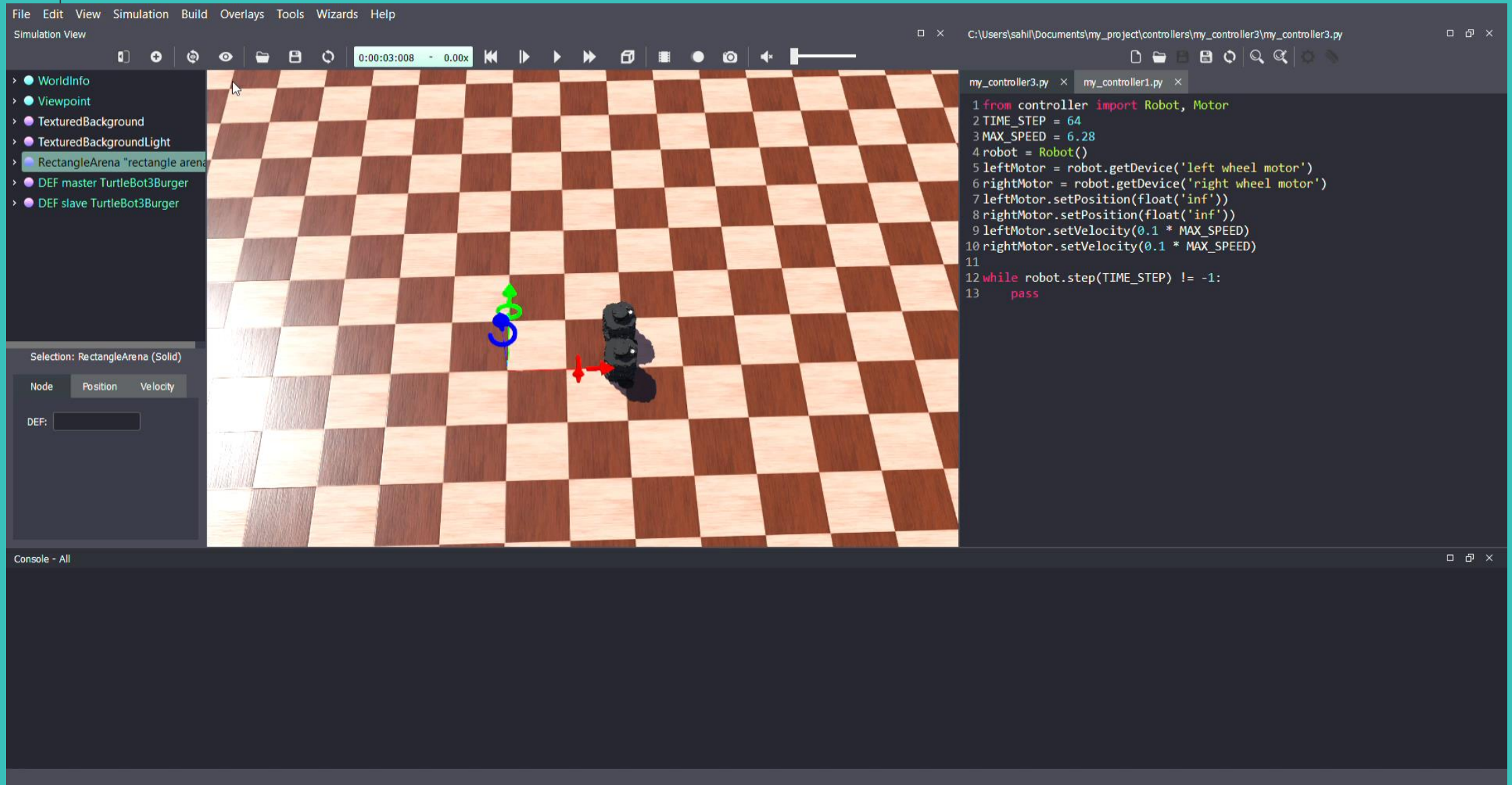
CODES

- `rotation = turtle_node.getField('rotation').getSFRotation()`
: extracting rotational details of master as well as slave bot
- `turtle1_node.getField('rotation').setSFRotation(rotation1)`
: setting rotational details for slave bot

CODES

- `k=turtle_node.getVelocity()`
: *extracting velocity details of master as well as slave bot*
- `turtle1_node.setVelocity(s)`
: *setting velocity details for slave bot*

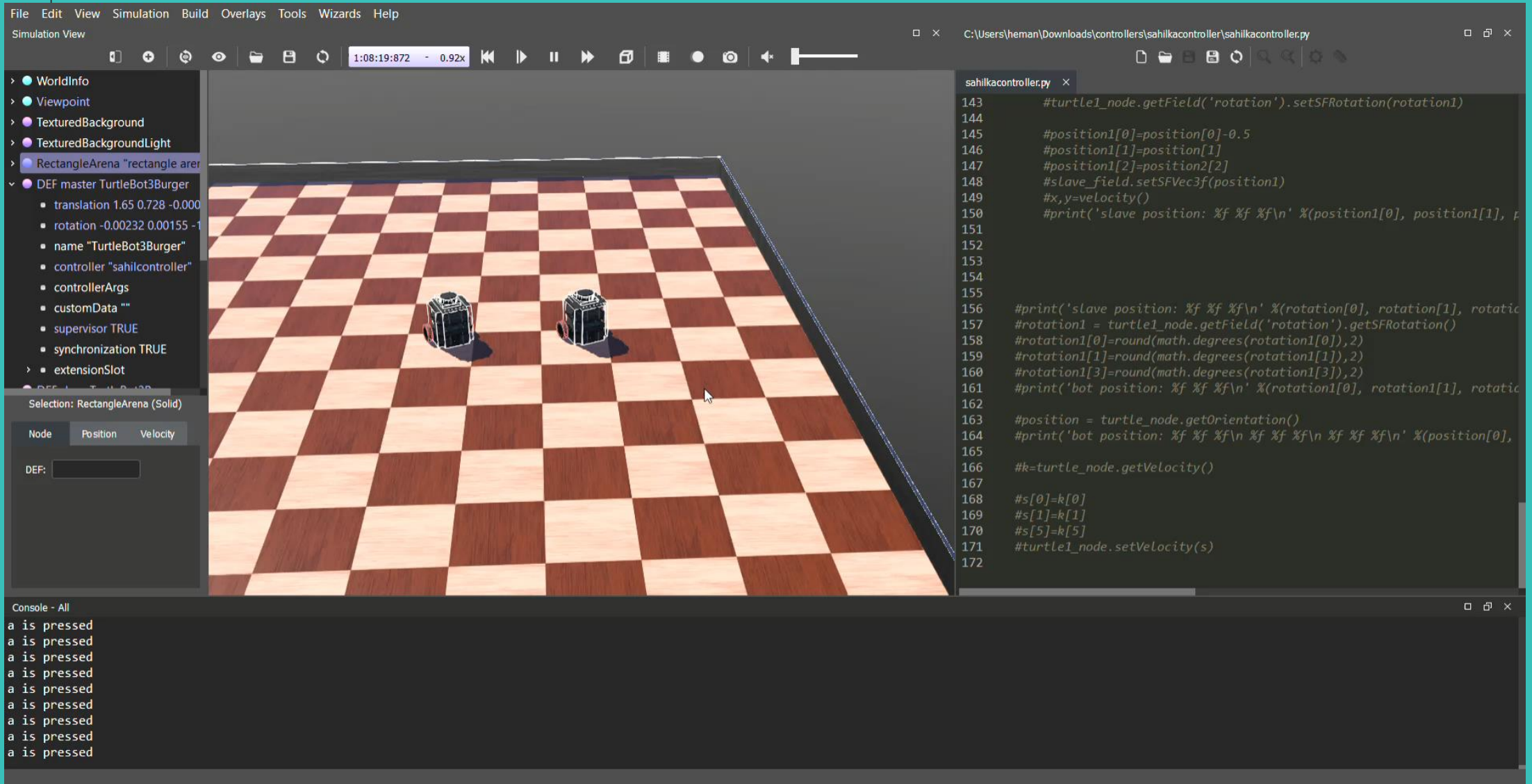
Printing positional details



The screenshot displays a ROS simulation environment. The main window shows a 3D view of a robot (a black two-wheeled robot) on a checkered floor. The robot is positioned in the center of the frame. To the left of the robot, there are three coordinate axes: a blue vertical axis, a green horizontal axis, and a red horizontal axis. The simulation interface includes a menu bar (File, Edit, View, Simulation, Build, Overlays, Tools, Wizards, Help) and a toolbar with various icons. The status bar at the bottom indicates the simulation time as 0:00:03:008 and the zoom level as 0.00x. On the left side, there is a 'Simulation View' panel with a tree structure showing the simulation components: WorldInfo, Viewpoint, TexturedBackground, TexturedBackgroundLight, RectangleArena "rectangle arena", DEF master TurtleBot3Burger, and DEF slave TurtleBot3Burger. Below this, there is a 'Selection: RectangleArena (Solid)' panel with tabs for Node, Position, and Velocity. The 'Position' tab is selected, and the 'DEF:' field is empty. On the right side, there is a code editor window showing the contents of 'my_controller3.py'. The code is as follows:

```
1 from controller import Robot, Motor
2 TIME_STEP = 64
3 MAX_SPEED = 6.28
4 robot = Robot()
5 leftMotor = robot.getDevice('left wheel motor')
6 rightMotor = robot.getDevice('right wheel motor')
7 leftMotor.setPosition(float('inf'))
8 rightMotor.setPosition(float('inf'))
9 leftMotor.setVelocity(0.1 * MAX_SPEED)
10 rightMotor.setVelocity(0.1 * MAX_SPEED)
11
12 while robot.step(TIME_STEP) != -1:
13     pass
```

Operating turtle bot using keyboard



The screenshot displays a ROS simulation environment with a 3D view of a rectangular arena with a red and white checkered floor. Two small, box-like turtle bots are positioned in the center of the arena. The interface includes a top menu bar (File, Edit, View, Simulation, Build, Overlays, Tools, Wizards, Help), a toolbar with various simulation controls, and a status bar showing the simulation time as 1:08:19:872 at 0.92x speed.

On the left, a panel shows the simulation's object hierarchy. The selected object is the 'DEF master TurtleBot3Burger', which is a 'RectangleArena' with a 'rectangle arer' property. The 'DEF master TurtleBot3Burger' object has several properties: translation (1.65 0.728 -0.000), rotation (-0.00232 0.00155 -1), name 'TurtleBot3Burger', controller 'sahilcontroller', controllerArgs, customData '', supervisor TRUE, synchronization TRUE, and an extensionSlot.

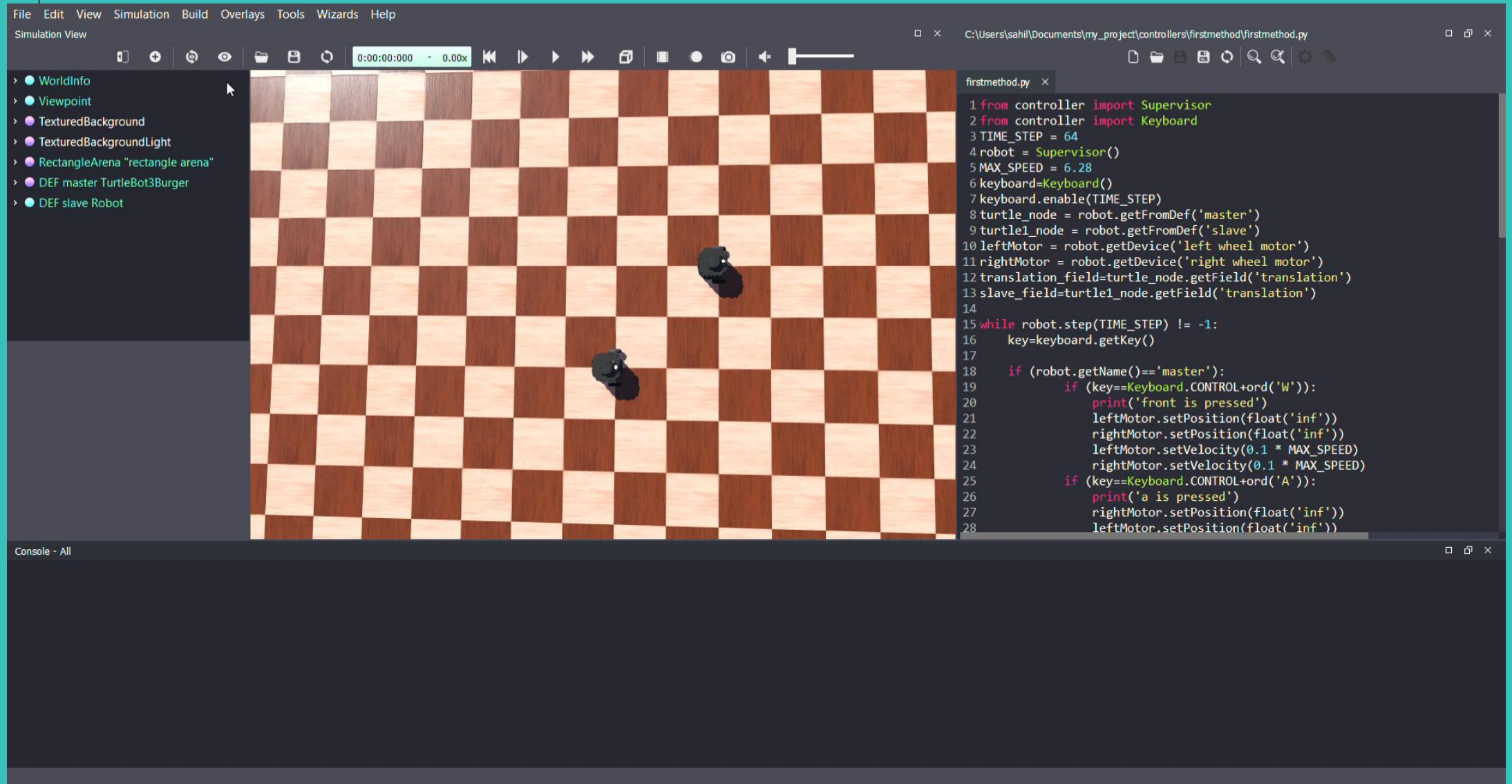
Below the hierarchy panel, a 'Selection: RectangleArena (Solid)' panel shows the 'Node' tab selected, with a 'DEF:' field.

On the right, a code editor window displays the 'sahilkacontroller.py' file. The code includes comments and function calls for setting rotation, position, and velocity, as well as printing statements for debugging.

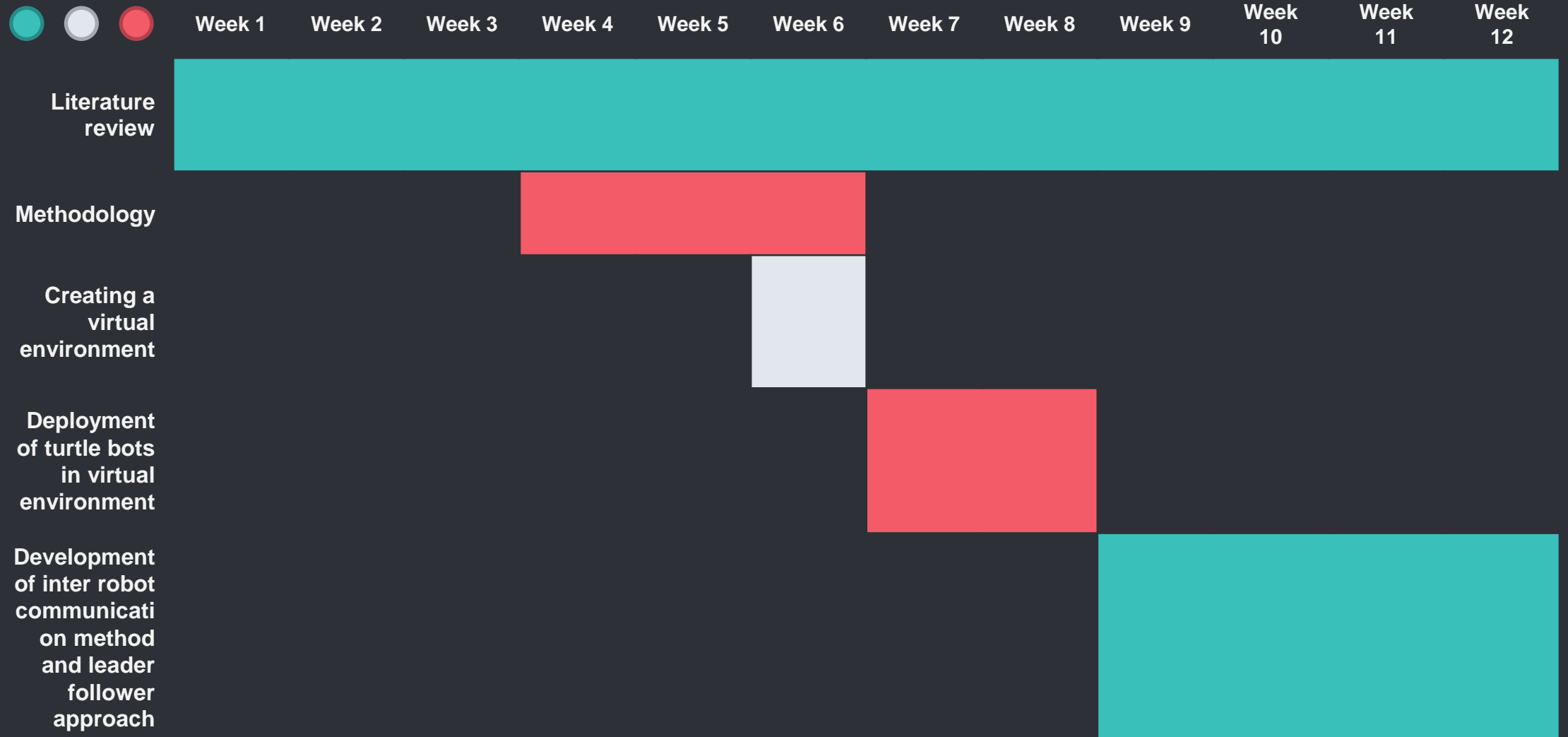
```
143 #turtle1_node.getField('rotation').setSFRotation(rotation1)
144
145 #position1[0]=position[0]-0.5
146 #position1[1]=position[1]
147 #position1[2]=position2[2]
148 #slave_field.setSFVec3f(position1)
149 #x,y=velocity()
150 #print('slave position: %f %f %f\n' %(position1[0], position1[1], p
151
152
153
154
155
156 #print('slave position: %f %f %f\n' %(rotation[0], rotation[1], rotati
157 #rotation1 = turtle1_node.getField('rotation').getSFRotation()
158 #rotation1[0]=round(math.degrees(rotation1[0]),2)
159 #rotation1[1]=round(math.degrees(rotation1[1]),2)
160 #rotation1[3]=round(math.degrees(rotation1[3]),2)
161 #print('bot position: %f %f %f\n' %(rotation1[0], rotation1[1], rotati
162
163 #position = turtle_node.getOrientation()
164 #print('bot position: %f %f %f\n %f %f %f\n %f %f %f\n' %(position[0],
165
166 #k=turtle_node.getVelocity()
167
168 #s[0]=k[0]
169 #s[1]=k[1]
170 #s[5]=k[5]
171 #turtle1_node.setVelocity(s)
172
```

At the bottom, a console window shows the output of the simulation, displaying a series of 'a is pressed' messages.

Final test run



Gantt chart



DESIGN AND DEVELOPMENT OF HOMOGENEOUS SWARM OF COOPERATIVE ROBOTS IN VIRTUAL ENVIRONMENT

MHB4244- DESIGN PROJECT-II

Team:

Devashish Sharma

Hemang Parab

Sahil Shaik

Saran Reddy

Guides:

Dr. RM. Kuppan Chetty

Associate Professor

Ms.Seenu N

Assistant Professor

THANKS!

Team: ANDRO WIZARDS

