

Name : Al Amin Hossain Nayem

Labwork 4

1)making project and node with a point that change in time in a circular path.

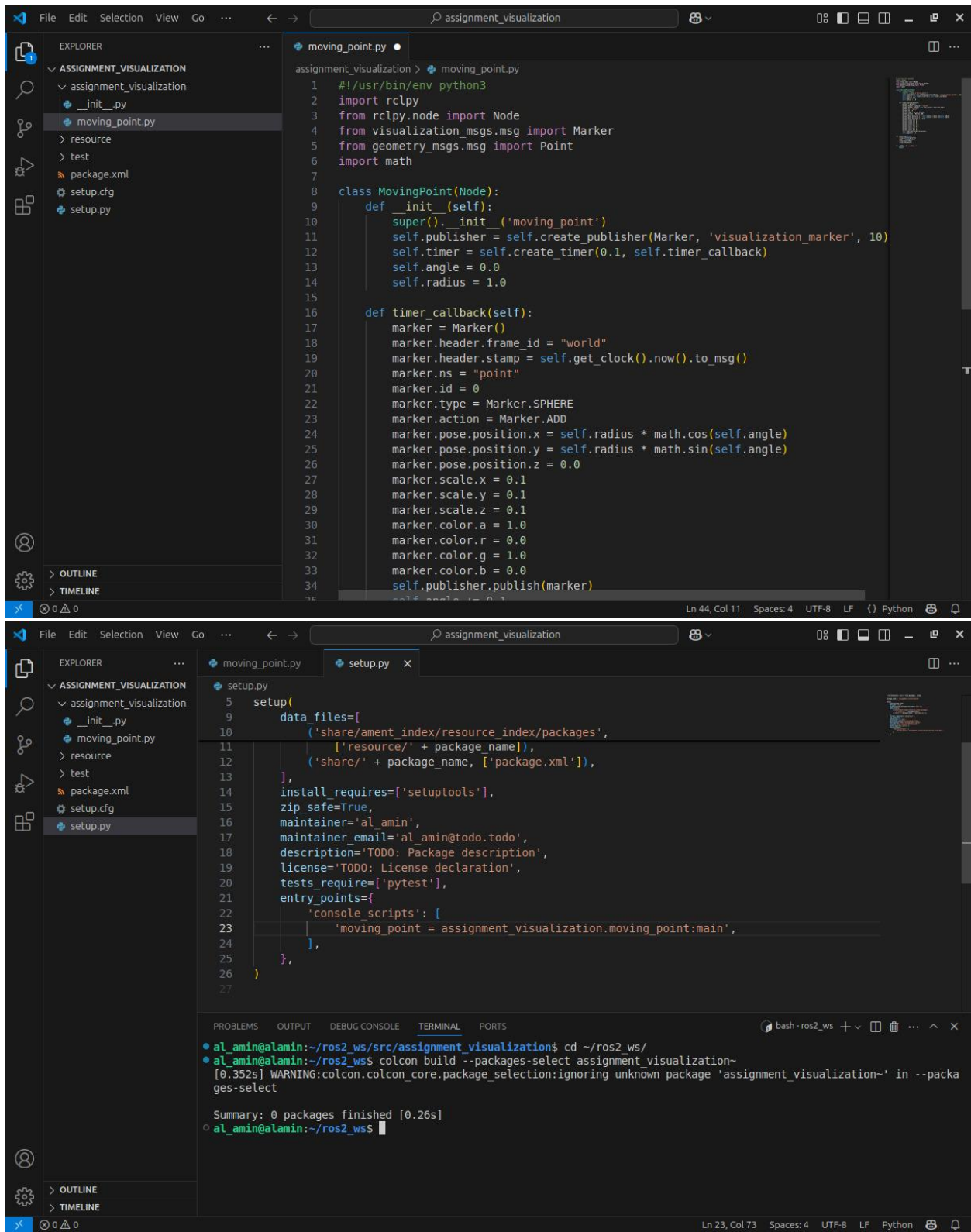
Node: moving_point.py

Implementation:

- Publishes a SPHERE marker on /visualization_marker
- Uses parametric equations for circular motion:
- $x = \text{radius} * \cos(\theta)$, $y = \text{radius} * \sin(\theta)$
- Updates position at 3Hz frequency to prevent RViz queue overflow

Name : Al Amin Hossain Nayem

Labwork 4



```
assignment_visualization > moving_point.py
1  #!/usr/bin/env python3
2  import rclpy
3  from rclpy.node import Node
4  from visualization_msgs.msg import Marker
5  from geometry_msgs.msg import Point
6  import math
7
8  class MovingPoint(Node):
9      def __init__(self):
10         super().__init__('moving_point')
11         self.publisher = self.create_publisher(Marker, 'visualization_marker', 10)
12         self.timer = self.create_timer(0.1, self.timer_callback)
13         self.angle = 0.0
14         self.radius = 1.0
15
16     def timer_callback(self):
17         marker = Marker()
18         marker.header.frame_id = "world"
19         marker.header.stamp = self.get_clock().now().to_msg()
20         marker.ns = "point"
21         marker.id = 0
22         marker.type = Marker.SPHERE
23         marker.action = Marker.ADD
24         marker.pose.position.x = self.radius * math.cos(self.angle)
25         marker.pose.position.y = self.radius * math.sin(self.angle)
26         marker.pose.position.z = 0.0
27         marker.scale.x = 0.1
28         marker.scale.y = 0.1
29         marker.scale.z = 0.1
30         marker.color.a = 1.0
31         marker.color.r = 0.0
32         marker.color.g = 1.0
33         marker.color.b = 0.0
34         self.publisher.publish(marker)
```

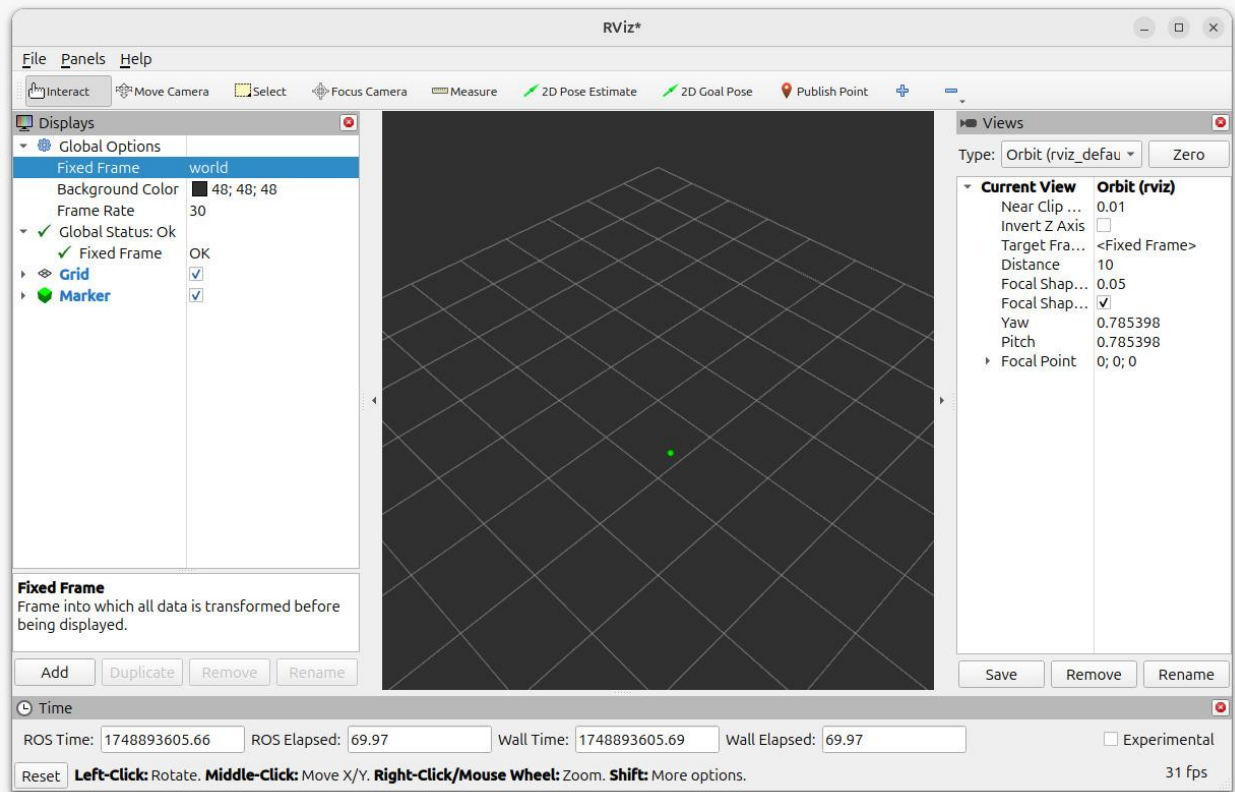
```
assignment_visualization > setup.py
5  setup(
6      data_files=[
7          ('share/ament_index/resource_index/packages',
8           ['resource/' + package_name]),
9          ('share/' + package_name, ['package.xml']),
10     ],
11     install_requires=['setuptools'],
12     zip_safe=True,
13     maintainer='al_amin',
14     maintainer_email='al_amin@todo.todo',
15     description='TODO: Package description',
16     license='TODO: License declaration',
17     tests_require=['pytest'],
18     entry_points={
19         'console_scripts': [
20             'moving_point = assignment_visualization.moving_point:main',
21         ],
22     },
23 )
```

```
al_amin@alamin:~/ros2_ws/src/assignment_visualization$ cd ~/ros2_ws/
al_amin@alamin:~/ros2_ws$ colcon build --packages-select assignment_visualization-
[0.352s] WARNING:colcon.colcon_core.package_selection:ignoring unknown package 'assignment_visualization~' in --packa
ges-select

Summary: 0 packages finished [0.26s]
al_amin@alamin:~/ros2_ws$
```

Name : Al Amin Hossain Nayem

Labwork 4



2) sending post and line list.

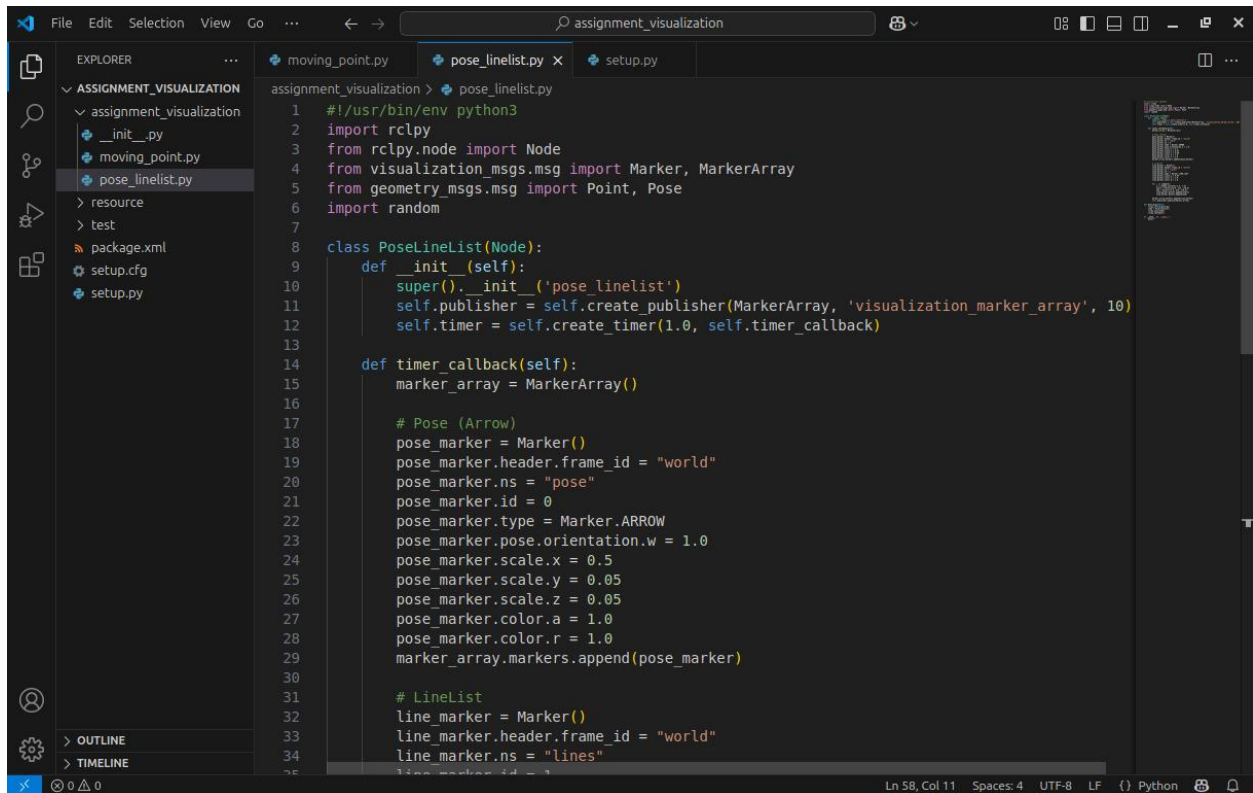
Node: pose_linelist.py

Implementation:

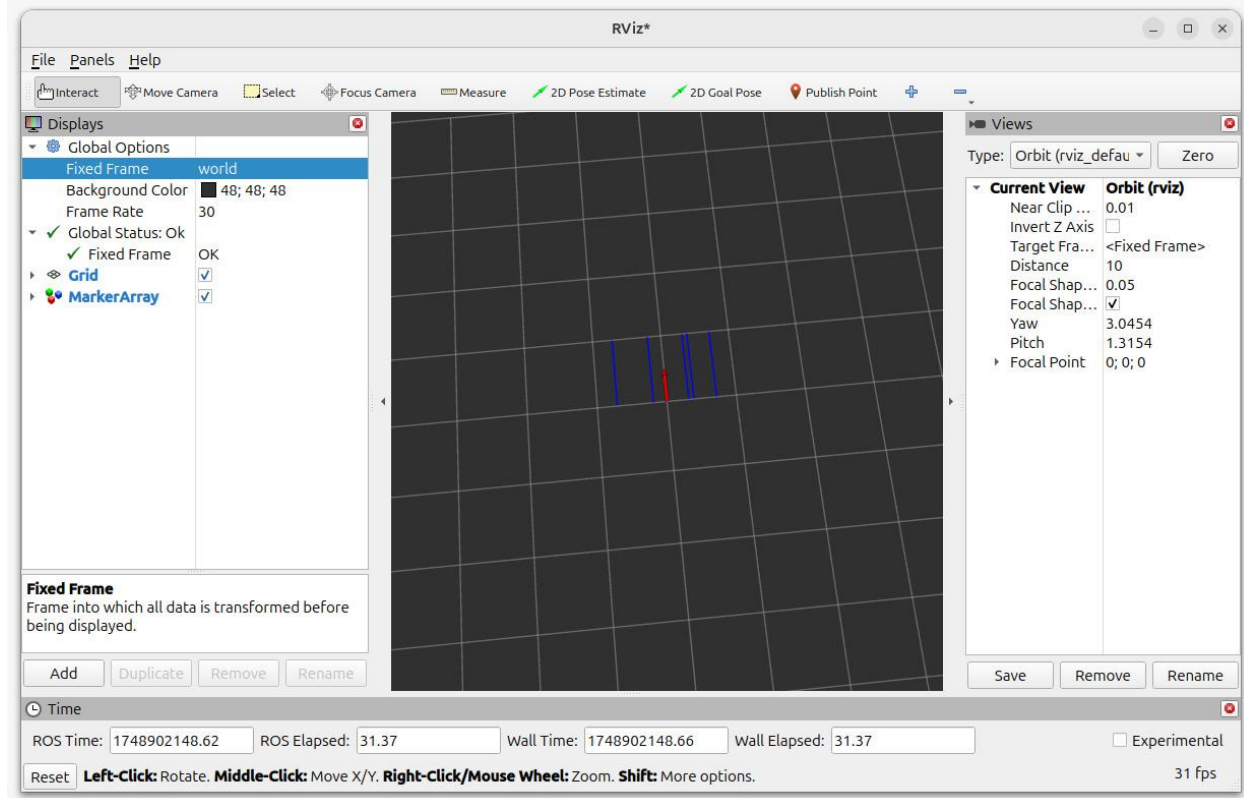
- Publishes a MarkerArray containing:
- ARROW for pose representation
- LINE_LIST with random X-axis lines
- Uses random generation for line positions

Name : Al Amin Hossain Nayem

Labwork 4



```
1  #!/usr/bin/env python3
2  import rclpy
3  from rclpy.node import Node
4  from visualization_msgs.msg import Marker, MarkerArray
5  from geometry_msgs.msg import Point, Pose
6  import random
7
8  class PoseLineList(Node):
9      def __init__(self):
10         super().__init__('pose_linelist')
11         self.publisher = self.create_publisher(MarkerArray, 'visualization_marker_array', 10)
12         self.timer = self.create_timer(1.0, self.timer_callback)
13
14     def timer_callback(self):
15         marker_array = MarkerArray()
16
17         # Pose (Arrow)
18         pose_marker = Marker()
19         pose_marker.header.frame_id = "world"
20         pose_marker.ns = "pose"
21         pose_marker.id = 0
22         pose_marker.type = Marker.ARROW
23         pose_marker.pose.orientation.w = 1.0
24         pose_marker.scale.x = 0.5
25         pose_marker.scale.y = 0.05
26         pose_marker.scale.z = 0.05
27         pose_marker.color.a = 1.0
28         pose_marker.color.r = 1.0
29         marker_array.markers.append(pose_marker)
30
31         # LineList
32         line_marker = Marker()
33         line_marker.header.frame_id = "world"
34         line_marker.ns = "lines"
```



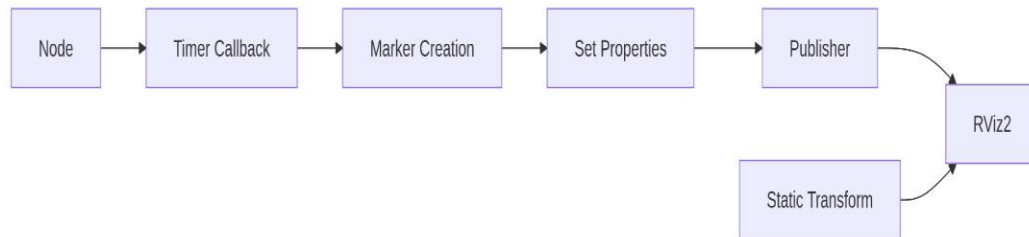
3,4)adding cube and direction

Name : Al Amin Hossain Nayem
Labwork 4

Node: moving_cube.py

Implementation:

- Publishes two markers on /cube_marker:
- CUBE moving in square trajectory
- ARROW indicating movement direction
- Uses waypoints for square path:
- $(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (0,0)$
- Arrow orientation updates with 90° rotations

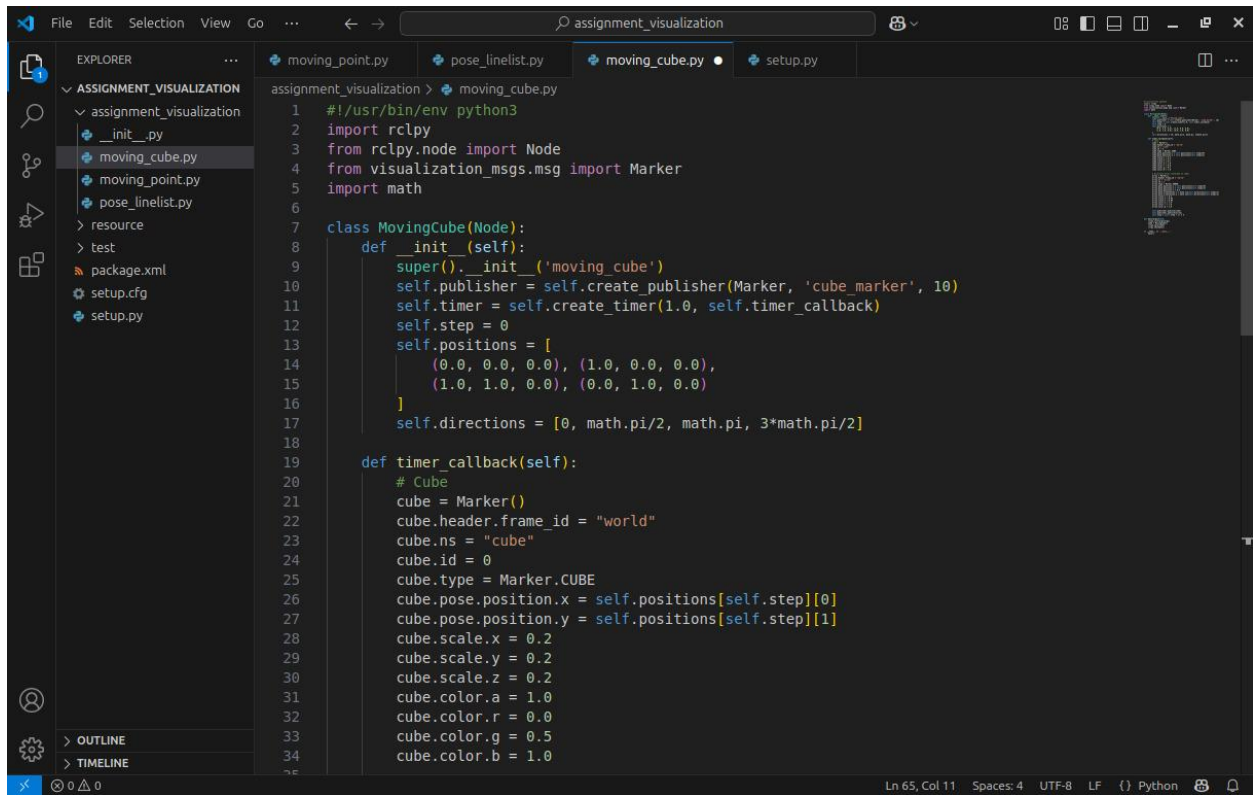


Configuration Essentials:

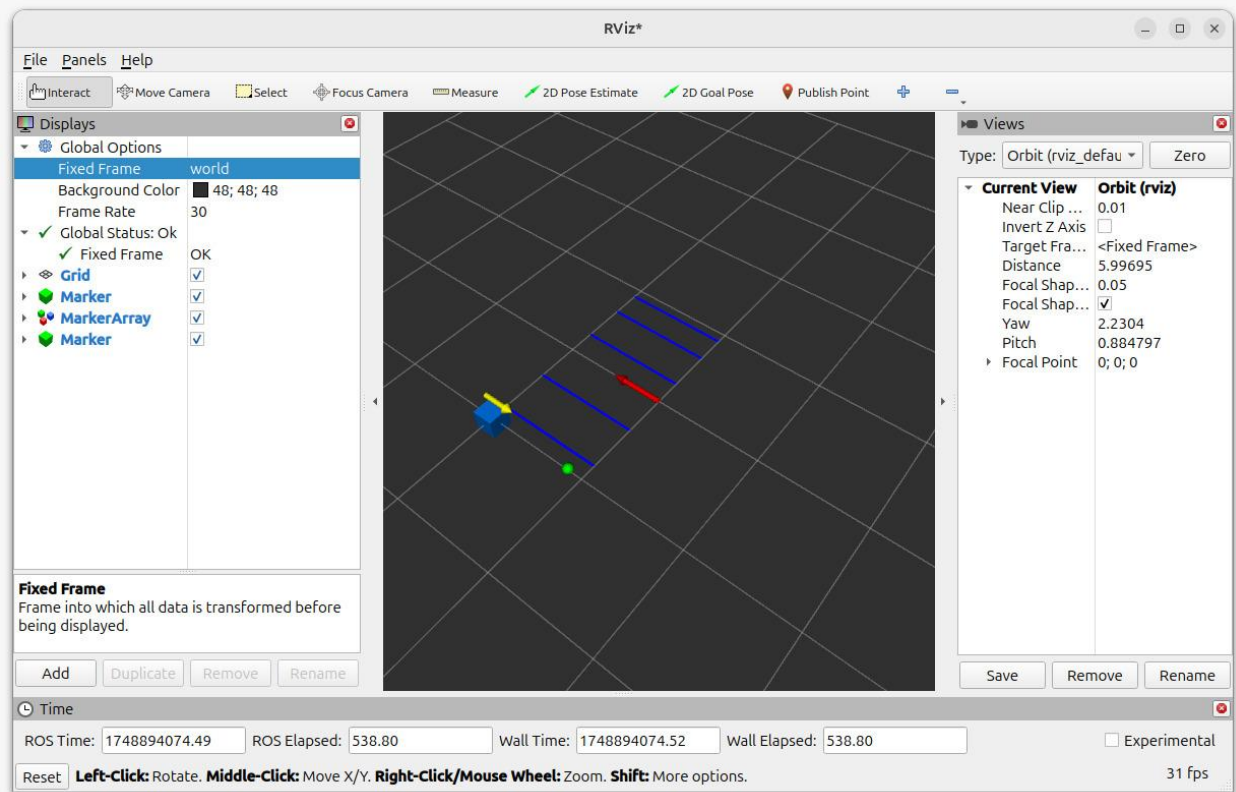
- Fixed Frame Consistency: All markers use world frame
- Namespace Management: Unique namespaces per marker type
- Queue Management: Reduced publish rates (0.3s) prevent overflow
- Transform Setup: static_transform_publisher for world→base_link

Name : Al Amin Hossain Nayem

Labwork 4

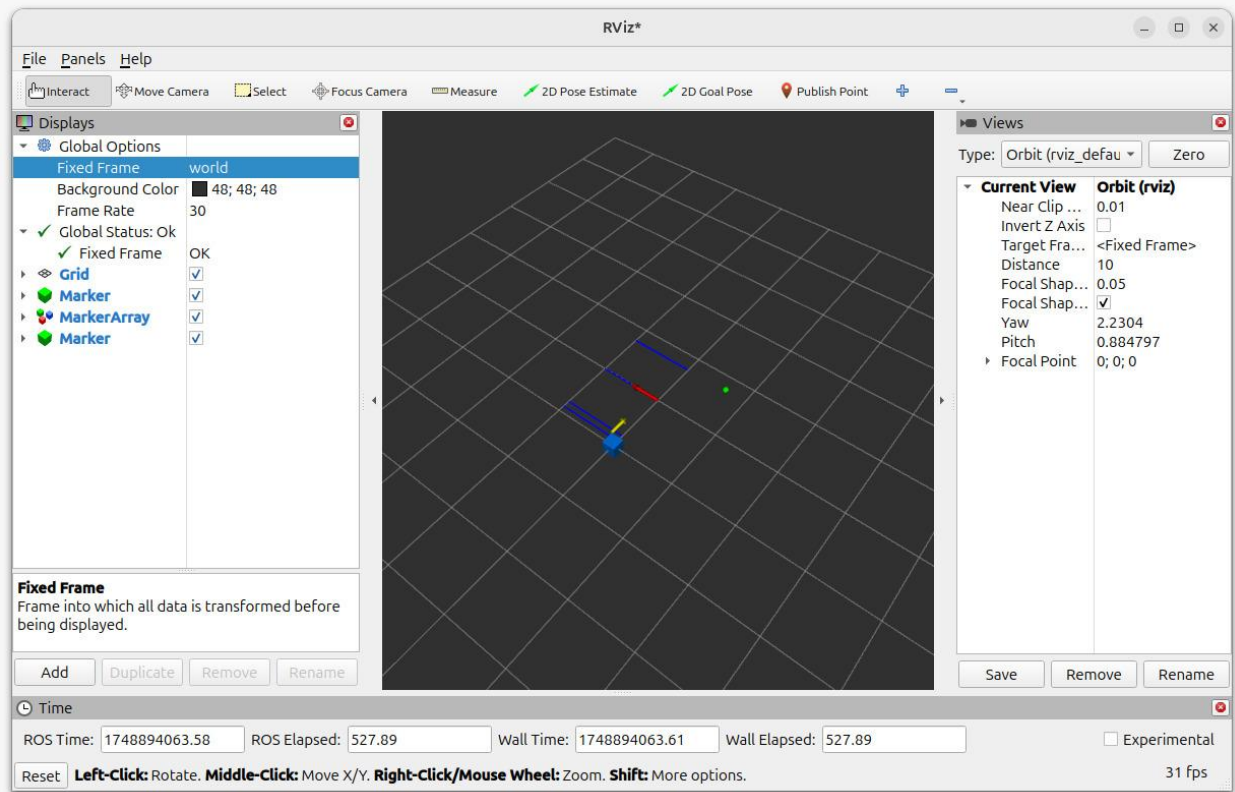


```
1  #!/usr/bin/env python3
2  import rclpy
3  from rclpy.node import Node
4  from visualization_msgs.msg import Marker
5  import math
6
7  class MovingCube(Node):
8      def __init__(self):
9          super().__init__('moving_cube')
10         self.publisher = self.create_publisher(Marker, 'cube_marker', 10)
11         self.timer = self.create_timer(1.0, self.timer_callback)
12         self.step = 0
13         self.positions = [
14             (0.0, 0.0, 0.0), (1.0, 0.0, 0.0),
15             (1.0, 1.0, 0.0), (0.0, 1.0, 0.0)
16         ]
17         self.directions = [0, math.pi/2, math.pi, 3*math.pi/2]
18
19     def timer_callback(self):
20         # Cube
21         cube = Marker()
22         cube.header.frame_id = "world"
23         cube.ns = "cube"
24         cube.id = 0
25         cube.type = Marker.CUBE
26         cube.pose.position.x = self.positions[self.step][0]
27         cube.pose.position.y = self.positions[self.step][1]
28         cube.scale.x = 0.2
29         cube.scale.y = 0.2
30         cube.scale.z = 0.2
31         cube.color.a = 1.0
32         cube.color.r = 0.0
33         cube.color.g = 0.5
34         cube.color.b = 1.0
```



Name : Al Amin Hossain Nayem

Labwork 4



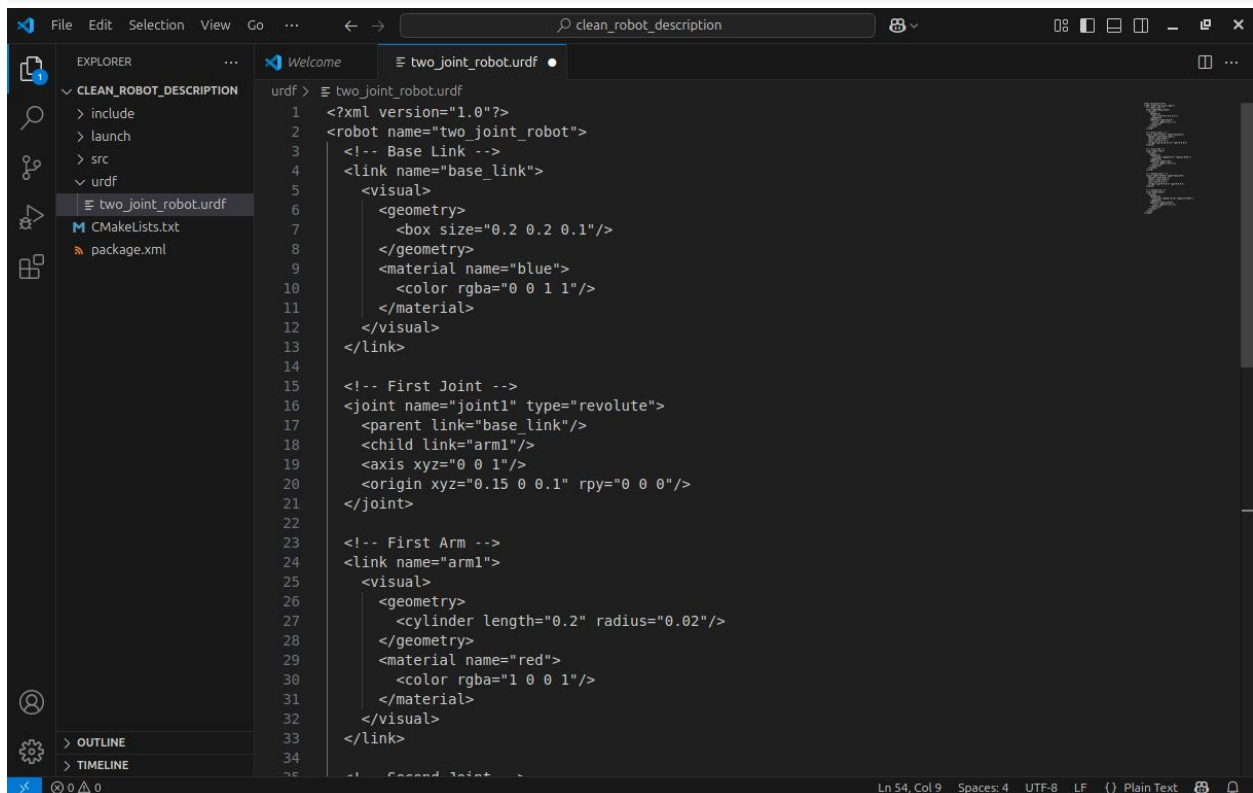
5)creating a URDF robot .

Creating a new package for the urdf robot

Name : Al Amin Hossain Nayem
Labwork 4

```
al_amin@alamin: ~/ros2_ws/src/clean_robot_description
going to create a new package
package name: clean_robot_description
destination directory: /home/al_amin/ros2_ws/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['al_amin <al_amin@todo.todo>']
licenses: ['TODO: License declaration']
build type: ament_cmake
dependencies: []
creating folder ./clean_robot_description
creating ./clean_robot_description/package.xml
creating source and include folder
creating folder ./clean_robot_description/src
creating folder ./clean_robot_description/include/clean_robot_description
creating ./clean_robot_description/CMakeLists.txt

[WARNING]: Unknown license 'TODO: License declaration'. This has been set in the
package.xml, but no LICENSE file has been created.
It is recommended to use one of the ament license identifiers:
Apache-2.0
BSL-1.0
BSD-2.0
BSD-2-Clause
```

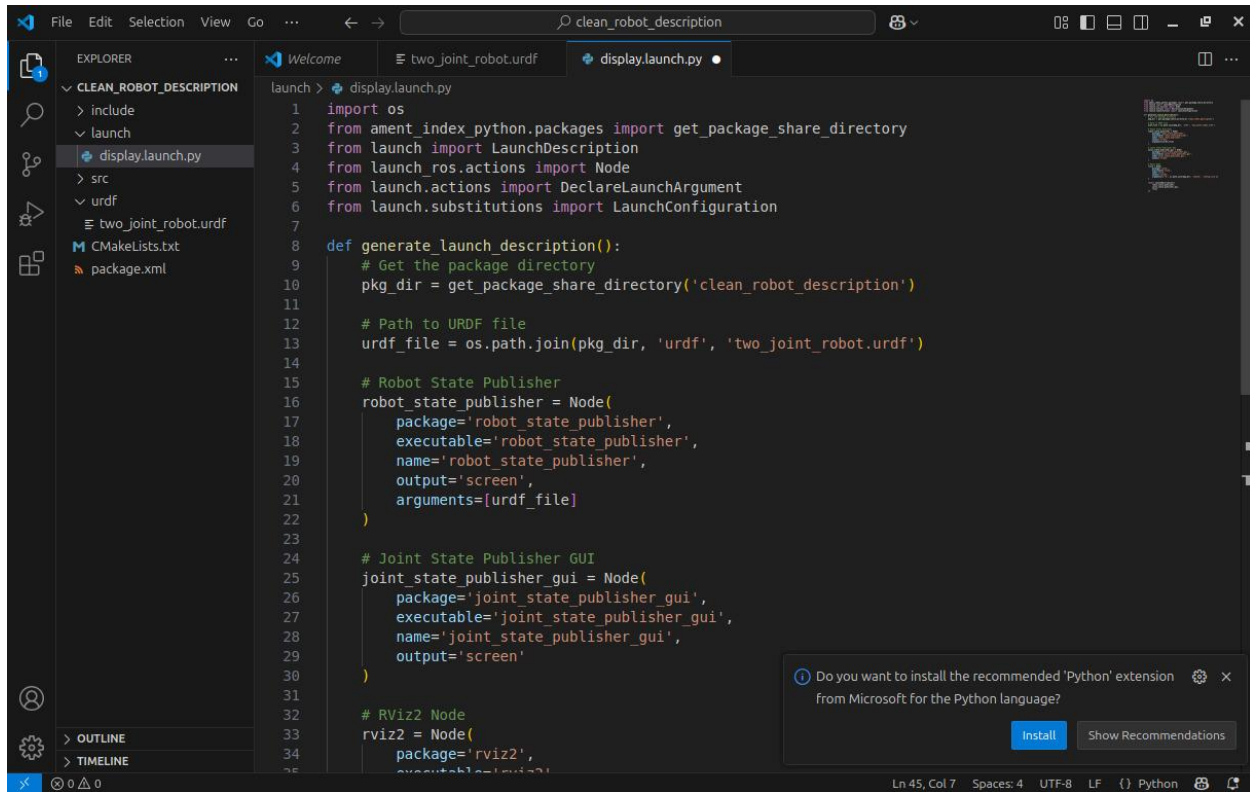


```
urdf > two_joint_robot.urdf
1  <?xml version="1.0"?>
2  <robot name="two_joint_robot">
3    <!-- Base Link -->
4    <link name="base_link">
5      <visual>
6        <geometry>
7          <box size="0.2 0.2 0.1"/>
8        </geometry>
9        <material name="blue">
10         <color rgba="0 0 1 1"/>
11       </material>
12     </visual>
13   </link>
14
15   <!-- First Joint -->
16   <joint name="joint1" type="revolute">
17     <parent link="base_link"/>
18     <child link="arm1"/>
19     <axis xyz="0 0 1"/>
20     <origin xyz="0.15 0 0.1" rpy="0 0 0"/>
21   </joint>
22
23   <!-- First Arm -->
24   <link name="arm1">
25     <visual>
26       <geometry>
27         <cylinder length="0.2" radius="0.02"/>
28       </geometry>
29       <material name="red">
30         <color rgba="1 0 0 1"/>
31       </material>
32     </visual>
33   </link>
34
35   <!-- Second Joint -->
```


Name : Al Amin Hossain Nayem

Labwork 4

Creating a launch file



```
1 import os
2 from ament_index_python.packages import get_package_share_directory
3 from launch import LaunchDescription
4 from launch_ros.actions import Node
5 from launch.actions import DeclareLaunchArgument
6 from launch.substitutions import LaunchConfiguration
7
8 def generate_launch_description():
9     # Get the package directory
10    pkg_dir = get_package_share_directory('clean_robot_description')
11
12    # Path to URDF file
13    urdf_file = os.path.join(pkg_dir, 'urdf', 'two_joint_robot.urdf')
14
15    # Robot State Publisher
16    robot_state_publisher = Node(
17        package='robot_state_publisher',
18        executable='robot_state_publisher',
19        name='robot_state_publisher',
20        output='screen',
21        arguments=[urdf_file]
22    )
23
24    # Joint State Publisher GUI
25    joint_state_publisher_gui = Node(
26        package='joint_state_publisher_gui',
27        executable='joint_state_publisher_gui',
28        name='joint_state_publisher_gui',
29        output='screen'
30    )
31
32    # RViz2 Node
33    rviz2 = Node(
34        package='rviz2',
35        executable='rviz2',
```

Do you want to install the recommended 'Python' extension from Microsoft for the Python language?

Install Show Recommendations

Name : Al Amin Hossain Nayem
Labwork 4

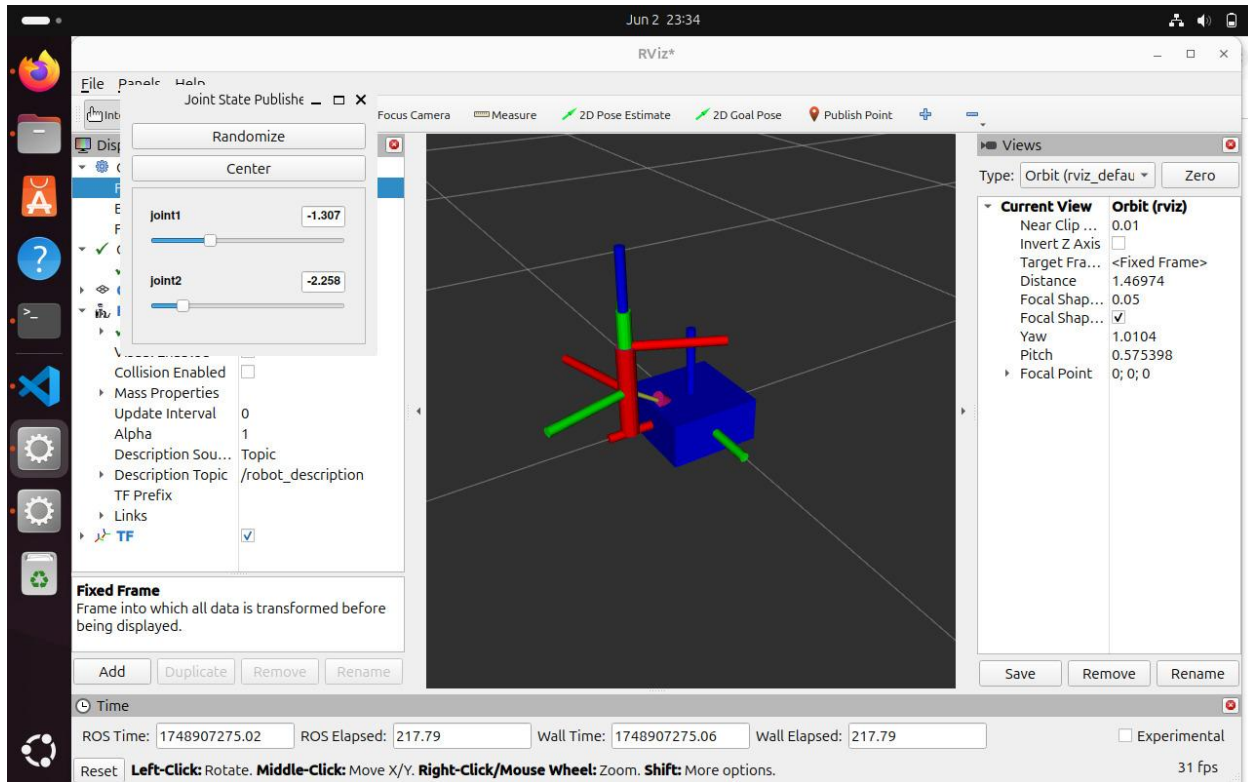
Build the file

```
al_amin@alamin: ~/ros2_ws
source install/setup.bash
[0.272s] WARNING:colcon.colcon_ros.prefix_path.ament:The path '/home/al_amin/ros2_ws/install/visualisation_rviz' in the environment variable AMENT_PREFIX_PATH doesn't exist
[0.273s] WARNING:colcon.colcon_ros.prefix_path.ament:The path '/home/al_amin/ros2_ws/install/temp_converter' in the environment variable AMENT_PREFIX_PATH doesn't exist
[0.273s] WARNING:colcon.colcon_ros.prefix_path.ament:The path '/home/al_amin/ros2_ws/install/minimal_action' in the environment variable AMENT_PREFIX_PATH doesn't exist
[0.273s] WARNING:colcon.colcon_ros.prefix_path.ament:The path '/home/al_amin/ros2_ws/install/custom_action_interfaces' in the environment variable AMENT_PREFIX_PATH doesn't exist
[0.273s] WARNING:colcon.colcon_ros.prefix_path.ament:The path '/home/al_amin/ros2_ws/install/assignment_visualization' in the environment variable AMENT_PREFIX_PATH doesn't exist
[0.273s] WARNING:colcon.colcon_ros.prefix_path.catkin:The path '/home/al_amin/ros2_ws/install/custom_action_interfaces' in the environment variable CMAKE_PREFIX_PATH doesn't exist
Starting >>> clean_robot_description
Finished <<< clean_robot_description [1.67s]

Summary: 1 package finished [1.89s]
al_amin@alamin:~/ros2_ws$
```

Name : Al Amin Hossain Nayem

Labwork 4



Critical Learnings:

- Frame Consistency is paramount - RViz requires a valid transform tree
- URDF Validation is essential before visualization:

TF2 Timing:

- Static transforms (world→base_link) must be established first
- Dynamic transforms (joints) require continuous publishing

RViz Best Practices:

- Start with 50+ queue size for markers
- Use RobotModel display for URDF visualization
- Always verify frame rates (30Hz default)

Name : Al Amin Hossain Nayem

Labwork 4

Conclusion :

This implementation demonstrates core ROS2 visualization capabilities, providing a foundation for robotic perception systems. The separation of visualization logic (markers) from physical modeling (URDF/TF2) enables scalable development of complex robotic systems.