

ROS - Assignments 3

Practical Course for Mobile Systems Engineering

Viktor Seib

vseib@uni-koblenz.de

Institute for Computational Visualistics
University of Koblenz and Landau

November 11th, 2013



Task 1: Turtle Color

Task 2: Turtle TF Frames

Task 3: Turtle Markers

Task 4: Turtle URDF

Task 1: Turtle Color

Task 2: Turtle TF Frames

Task 3: Turtle Markers

Task 4: Turtle URDF

Turtle Color

Task:

- ▶ create a new package `turtle_color` and a node with the same name
- ▶ the node needs 3 parameters that should be passed on the command line: `[x_color]`, `[y_color]`, `[value]`
- ▶ `[x_color]`, `[y_color]` can be “r”, “g” or “b” (but must be different)
- ▶ `[value]` is a value between 0 and 255 for the color component not specified in the other two parameters, e.g. if `[x_color]` is “r” and `[y_color]` is “g” then `[value]` specifies the “b” color component
- ▶ the turtle’s position in the `turtle_sim` determines the background color

Turtle Color

Task:

- ▶ if the turtle is at the beginning of one axis, this axis' color component is 0, at the end of the axis it is 255
- ▶ the background color must change continuously as the turtle is moving
- ▶ create a launch file that starts your new node, `turtlesim` and `turtle_teleop_key`, set the parameters for your new node directly in the launch file

Hint:

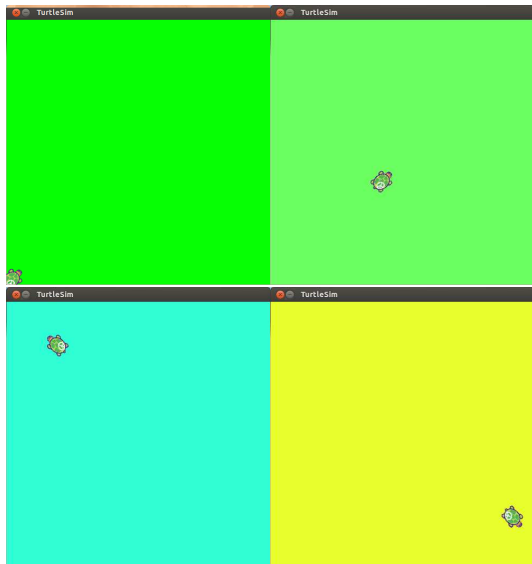
- ▶ take a look at <http://wiki.ros.org/roscpp/Overview> to get more information on how to access the parameter server
- ▶ you can use the provided file `turtle_color.cpp` and fill in all lines marked as "TODO"

Turtle Color

Goal:

Parameters:

```
[x_color]: r  
[y_color]: b  
[value]: 255  
(for green color  
component)
```



Task 1: Turtle Color

Task 2: Turtle TF Frames

Task 3: Turtle Markers

Task 4: Turtle URDF

Turtle TF Frames

Task:

- ▶ create a new package `turtle_tf_frames` and a node with the same name
- ▶ this node should broadcast three `tf`-frames:
 - ▶ a `turtle`-frame indicating the position of the turtle in world-coordinates
 - ▶ a `turtle_left_wheel`-frame: 0.2 m left of the `turtle`-frame
 - ▶ a `turtle_right_wheel`-frame: 0.2 m right of the `turtle`-frame
- ▶ whenever you move the turtle forwards (backwards) both wheel-frames should turn forwards (backwards) at equal amount
- ▶ to make it simple, this amount in radians should be the distance in meters travelled by the turtle

Turtle TF Frames

Task:

- ▶ whenever you turn the turtle left (right), the right wheel-frame turns forwards (backwards) and the left wheel-frame turns backwards (forwards)
- ▶ the turning amount in radians should be the angle in radians the turtle was turned by
- ▶ turning forwards (backwards) means into the positive (negative) turning direction around the y -axis
- ▶ create a launch file that starts your new node, `turtlesim` and `turtle_teleop_key` and also `rviz` with the provided config file for visualization

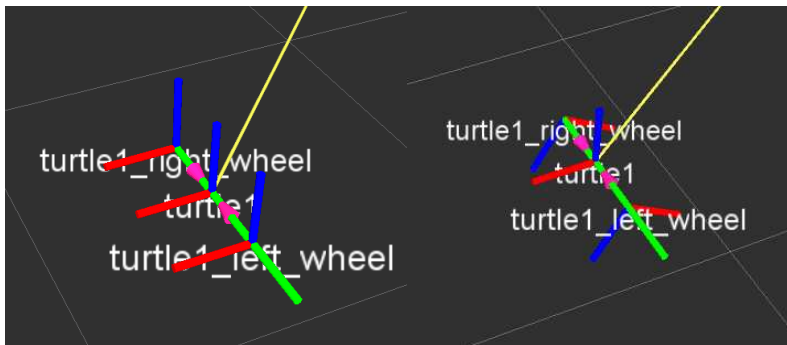
Hint:

- ▶ you can use the provided file `turtle_tf_frames.cpp` and fill in all lines marked as “TODO”



Turtle Color

Goal:



Task 1: Turtle Color

Task 2: Turtle TF Frames

Task 3: Turtle Markers

Task 4: Turtle URDF

Turtle Markers

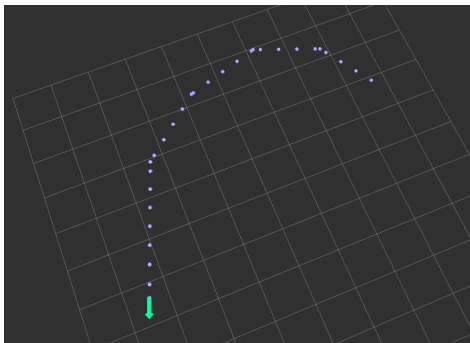
Task:

- ▶ create a new package `turtle_markers` and a node with the same name
- ▶ this node should subscribe to the pose of the turtle
- ▶ send `SPHERE` markers to show the path of the turtle
- ▶ only display the path of the last 10 seconds and don't send more than 4 markers per second
- ▶ also, send an `ARROW` marker to show the current position and orientation of the turtle
- ▶ create a launch file that starts
 - ▶ your new node `turtle_markers`
 - ▶ `turtlesim_node`
 - ▶ `turtle_teleop_key`
 - ▶ and `rviz` with all displays correctly set up for visualization



Turtle Markers

Goal:



Hint:

- ▶ you can use the provided file `turtle_markers.cpp` and fill in all lines marked as “TODO”

Task 1: Turtle Color

Task 2: Turtle TF Frames

Task 3: Turtle Markers

Task 4: Turtle URDF

Turtle URDF

Task:

- ▶ create an URDF description of a turtle
- ▶ decide for yourself how much time you want to put into this task
 - ▶ e.g. just use boxes or create a nicely looking turtle
- ▶ the turtle should have a body and a head
- ▶ the joint between body and head should be fix
- ▶ the turtle should have 4 legs
- ▶ the legs should be connected with a `revolute` joint
- ▶ create a launch file to display the turtle model

Turtle Markers

Goal:

