Renato Sanabria (ECE 4752)
Filipe Pereira (MAE 5750)
Jim Yu (CS 4752)

HW1
06 Feb 2015

Answers

1.
```
cd ~/catkin_ws/src
catkin_create_pkg group_hw1 std_msgs rospy roscpp
cd ~/catkin_ws
mkdir launch src msg srv
```

2. launch/hw1.launch contents :

```
<launch>
      <node
       pkg = "group_hw1"
       type= "sim_master"
       name = "group_hw1"
       output = "screen"
      />

      <node
       pkg = "group_hw1"
       type= "controller"
       name = "controller"
       output = "screen"
      />

        <param name="num_blocks" value="11"/>
        <param name="configuration" value="scattered"/>

</launch>
```

3. The message and service contents are the following:

```
srv/moveRobot.srv
      int64 command
      int64 target
      ---
      bool valid_action

msg/State.msg
      string blocks
      bool gripper_open
```

The representation of the **world state** is the following:

- Blocks are represented by numbers ranging from 1 to n (where n is the `num_blocks` parameter value)
- Block 0 represents the table
- 'g' represents the gripper location
- Blocks are separated by the character '|'
- The world state is stored in a string ending with '0|'
  e.g. stacked_ascending : 0|1|2|3|4|5|6|7|8|9|10|11|g0|

4. Please see code

5. The following commands were used to debug :
   a. **`rosnode list`** – to check nodes running , including message topics
   b. **`rostopic pub /group_hw1/command std_msgs/String 'scattered'`** – to check correct message reception and verify the reported change in the world state according to the selected mode
   c. **`rosservice info move_robot`** – to check that the move_robot service is well implemented. The command reports the node providing service requests, and service arguments.

6. The controller implements three configuration modes : **scattered , stacked_ascending, stacked_descending** . It will act only if the requested configuration is different from the current one

7. No time to do this one ☹

   The controller will get a state update, as soon as a command is issued to change configuration, and before the robot starts moving. Once the robot starts moving , we assume there are no external actions and therefore it is **not** necessary to wait for state updates to issue robot actions. The state will be frozen until task completion

8. Two instances of **sim_master** or **controller** will not run since we are using global names and it is not allowed that 2 nodes run with the same global name. The following error message will appear [new node registered with same name]