

**DISIM, University of L’Aquila**

**Robot World**

Intelligent Systems and Robotics Laboratory, prof. Giovanni De Gasperis

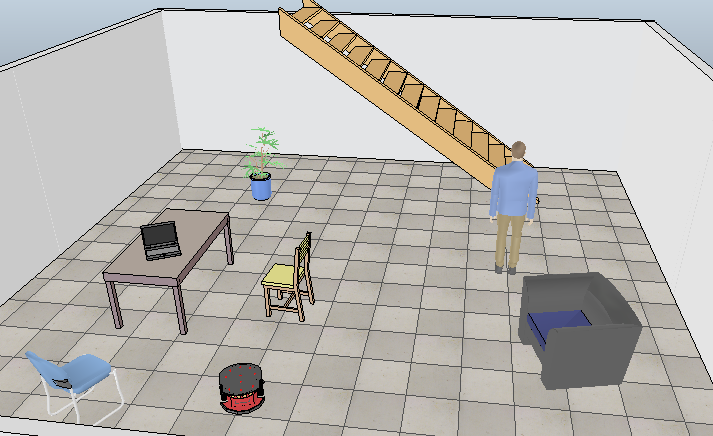
**By Yuna Frolov**

**Homework Specifications:**

1. Write a RobotWorld Python 3.5.x wrapper library with classes that encapsulates the V-REP Remote API
2. Write a RobotBrain class which instances controls V-REP simulated robots that:
   1. have proximity sensors in the front (decide how many)
   2. have left and right motors
   3. wanders the virtual environment avoids physical obstacles
3. Produce a screencast video of the running simulation (.ogv format), with Python console output and V-REP world both visible
4. Zip the V-REP scene and Python Project in a single ZIP file
5. Write an accompanying documentation file

**Preconditions:**

* 1. Have the files: *remoteApi.dll, vrep.py, vrepConst.py* in the same folder as project
  2. Have **vrep** and **numpy** libraries installed on the machine



**V-Rep file:**

|  |  |
| --- | --- |
| File name | RobotAvoidingObstacles.ttt |

Description: In the scene there is a robot of the type **Pioneer p3dx**, with 16 proximity sensors all around its body. Inside its LUA code there is an initialization of the sensors and motors, and an actuation for detection of the sensors. I left the original code of the robot as it is, and created a small **cuboid** object, and in its code I created the connection to the remote API.

All around the scene there are obstacles, resembling a room in a house; a table, chairs, a walking man and walls all around. The goal of the robot is to avoid obstacles while wandering around the environment.

**Python files:**

|  |  |
| --- | --- |
| File name | robotControllMain.py |
| Purpose | This file created the objects of the World and the Robot brain, attempts the connection to the simulation and actuates the code by making the robot “think” |
| Functions | Main function: creating objects, calling the robots’ “think” function |

|  |  |
| --- | --- |
| File name | \_\_init\_\_.py (in RobotWorld folder) |
| Purpose | This file contains the World and the RobotBrain classes. All of its functions allow it to communicate with the simulation and lead the robot to move around the room while avoiding obstacles. |
| Classes | Class World  Class RobotBrain |

|  |  |
| --- | --- |
| Class | Class World |
| Functions | * \_\_init\_\_ : this is the initialization function, and it starts the simulation through connection to the given ip and port. If the connection was successful – the initialization is complete. If not – a message appears and the application is stopped. * Sense : This function collects the data from the robot’s sensors * Act: This function detects the motors of the robot, as well as interacts with the sensors that were retrieved in sense function to perform a ‘read’ on them.   There is calculation on which sensor get the closest reading of an obstacle, and according to him – if the obstacle is closer than 0.2 meters – the robot avoids it by setting a steering value and implying it in the velocity speed. After the velocity is set – the motors get the new value.  At the end of the loop we get to stop the robot.   * Close : this function closes the connection and ends the simulation. |

|  |  |
| --- | --- |
| Class | Class RobotBrain |
| Functions | * \_\_init\_\_: initialization of self. * Perception : The robot senses the world then makes a decision * Decision : The robot decides to act * Think: The robot thinks by perceiving his surroundings. |