University of Genoa

ROBOTIC ENGINERING

Research Track 1

Object: flowchart for python robot simulator

Produce by: *Mohammad Reza Haji Hosseini*

STUDENT ID: 4394567

Lesson in charge by: *Prof. Carmine Tommaso Recchiuto*

The aim of the following flowchart is to better explain the code development stage. The clarification is based on a two-step process. In the first place, there are introduced two important functions that control two types of tokens and define the operation of any conditions used in the code. In the second place, it is elucidated how functions come into play and what conditions are defined for each probability of encountering silver token and golden token.

First Step

Two different functions are defined so as the robot may detect the silver and the golden tokens.

Function to find silver tokens:

```
def find_token_silver():
    silver_dist=100
for token in R.see():
        if (token.dist < silver_dist ) &  (token.info.marker_type==MARKER_TOKEN_SILVER ) & (-120 < token.rot_y < 105):
            silver_dist=token.dist
            silver_rot_y=token.rot_y
    if silver_dist==100:
        return -1, -1
else:
        return silver_dist, silver_rot_y
```

*In the function it is defined a default distance to check whether or not the silver token's distance equals 100. It is used a for loop to check if the silver token meets the three following conditions. The first condition is related to the distance, the second is associated with finding only silver colored tokens and the third condition is used to define the interval you are looking for. If these 3 conditions are met, then distance and token is returned in output. Instead, if silver token is very far, the function does not return distance and token, otherwise it does.

Function to find golden tokens:

```
def find_token_golden():
    golden_dist=100
 for token in R.see():
     if (token.dist < golden_dist ) &  (token.info.marker_type==MARKER_TOKEN_GOLD):
            golden_dist=token.dist
            golden_rot_y=token.rot_y
    if golden_dist==100:
```

```
    return -1, -1
else:
    return golden_dist, golden_rot_y
```

*In the function it is defined a default distance to check wether or not the golden token's distance equals 100. It is used a for loop to check if the golden token meets the three following conditions. The first condition is related to the distance, the second is associated with finding only golden colored tokens and the third condition is used to define the interval you are looking for. If these 3 conditions are met, then distance and token is returned in output. Instead, if silver token is very far, the function does not return distance and token, otherwise it does.

Second Step