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*

Reason for here I made this flowchart to better explain code development stage. I decided to explain this flowchart in two steps to better understand. In the first step there is only explanation of two important functions to control two types of tokens and define the operation of any conditions used in code and second step deals with part after while, it's mean how functions come into play and what conditions are defined for each probability of encountering silver token and golden token and all of this second step procedures I define in flowchart form.

First Step

So I defined two different types, first one for find silver token and second for find golden token.

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
```

*Inside function is defined a default distance to check silver token that equals 100, then inside a for loop defined R.see () method to say the token you are seeing it must have this 3 conditions. First condition for distance, second condition to find only silver colored ones and third condition to define the base you are looking for. if these 3 conditions are met then I return distance and token. Instead if silver token very far do not return distance and token, otherwise return.

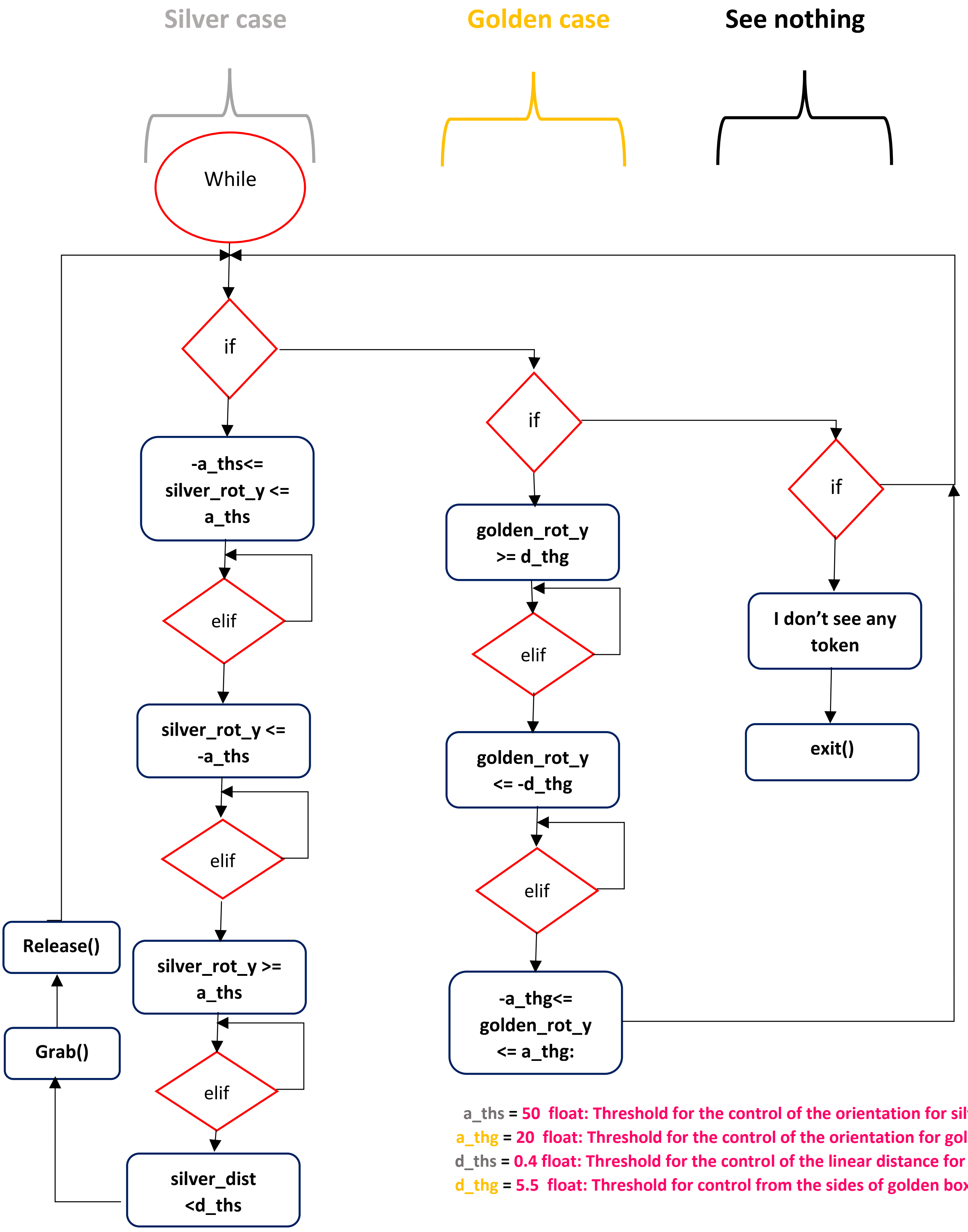
Function to find golden tokens:

```
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
```

*Inside function is defined a default distance to check golden token that equals 100, then inside a for loop defined R.see () method to say the token you are seeing it must have this 2 conditions. First condition for distance, second condition to find only golden colored ones and. if these 2 conditions are met then I return distance and token. Instead if golden token very far do not return distance and token, otherwise return.

Second Step



a_ths = 50 float: Threshold for the control of the orientation for silver box

a_thg = 20 float: Threshold for the control of the orientation for golden box

d_ths = 0.4 float: Threshold for the control of the linear distance for silver box

d_thg = 5.5 float: Threshold for control from the sides of golden box