

 배정	
 상태	자율주행
 일자	

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/cbda3706-e0eb-4a64-b442-6347c29a7c48/depthcamera.cpp>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/08423f22-3396-4e3a-8203-cfd73b3c0a38/main.py>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/66cec17c-f20c-4f89-afd8-9cbfb20badd0/ob_avoid.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0dbee59b-ce18-4c30-ad98-94f8110e6aef/chase_the_ball.py

파일

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/17ebb34a-a5cf-4ebd-bfda-fa19ddf0cda9/main.py>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/74972ed5-30ec-4d0c-8f6b-201718215de3/ob_avoid.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/3d4b449c-d25d-4f34-b61f-8915310ad061/chase_the_ball.py

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fc85260a-3b56-456e-87c1-26149bb64e4b/depthcamera.cpp>

ob_avoid → 현재 anlge값을 받아 2에서 틀었던 방향으로 30도 트는 것으로 설정해놓음
→ 그러나 방향이 맞는지 확인할 시간이 별로 없었기에 확인 필요 + anlge값은 계속 조정해서 우선적으로 장애물을 피하도록 구성할 필요가 있다고 생각됨

→ 다른 아이디어가 있다면 구현해도 됩니다

> 벽이 있어서 장애물 회피 시퀀스인, 회전 > 직진 > 재회전 중, 직진 상태에서 근접하는 물체가 있을 때(벽이라 판단될 때), stop_flag 를 주어 30도로 틀게 하였고

이외의 경우는 재회전까지 진행하여 진행 방향을 유지하도록 하였습니다

chase_the_ball → 한바퀴 회전에서 오류가 발생함 → 시간을 받아서 해결하는 등 다른 해결 방법 혹은 문제를 잡으면 좋겠음

> 한바퀴 회전으로 끝나지 않는 이유가, 서브스크라이버로 받아오는 값이 linear 하지 않고 discrete 하기에 if 문 조건 속의 == 연산자가 동작하지 않았기 때문입니다.

임의의 0.1 라디안 범위를 지정하여 한바퀴를 회전할 수 있도록 하였습니다

이론상 ob_avoid로 회피 후 공이 find 되지 않는다면 바로 한바퀴를 돌것임 → 이를 해결해 주면 좋겠음

> 진행 중

오늘 카메라 테스트에서 애가 이상했던 이유는 프레임상의 x값에 비해 steer_action값이 너무 커서 그런것 같습니다. 비슷하게 조정후에 다시 테스트 부탁드립니다

> steer_action 값을 삼각함수를 이용하여 sinusoidal 하게 조정하였습니다. x 값이 중간 범위에 가까워질수록 천천히 회전합니다.

벽이든 장애물이든 만났을때 잘피하고 공만 잘찾아가면 일단 완성으로 두고, 나머지 움직이는 방향은 제가 나중에 와서 해볼게요 ππ

1차완성 부탁드립니다용

1. ob_avoid.py 의 set_avoid_obstacle_flag, get_avoid_obstacle_flag, run 메소드 수정

set_avoid_obstacle_flag , run 메소드 삭제

get_avoid_obstacle_flag 메소드의 output 은 동작이 필요할 때 True, 필요 없을 때 False 를 반환

2. main.py. 조건문 수정

while 내부에서 get_avoid_obstacle_flag 값을 받는 if 문 내부에서 chase_ball.run() 구문 삭제

3. chase_the_ball.py 의 Initial_start_angle 메소드 수정

Initial_start_angle 메소드 삭제

해당 메소드가 동작하는 main.py. 의 구문 삭제

초기 wait_for_message 기능을 __init__ 에 이관

4. chase_the_ball.py 의 odomCallback 메소드 수정

from math import pi 구문 추가

입력된 yaw 의 값을 라디안 값으로 self.angle 에 반영되게 수정

연계된 Findball 메소드 input 및 end_angle 구문 수정

5. chase_the_ball.py 의 global 변수들을 class 내부 변수로 수정

6. chase_the_ball.py 의 get_control_action 메소드 수정

2개의 rospy.sleep 구문 삭제

중복 if 문 수정(self.blob_x 가 -1 일 때가 중복)

steer_action 값을 cos 을 이용하여 ' 부드럽게 ' 회전하도록 수정

throttle_action, y 로 분리되어 관리되는 값을 통합

forward_flag 를 최상위 if 문으로 수정

7. chase_the_ball.py 의 count 변수명을 no_detected_ball 로 변경, 자료형을 bool 형으로 변경

8. ob.avoid.py 의 avoid_obstacle_2 메소드 수정

stop_flag 를 받는 if 문의 rospy.sleep 구문 삭제

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/73441768-fc07-470b-81db-40dad2b9d0c9/chase_the_ball.py

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5949edeb-963d-473f-86a9-e1d44b0667ec/main.py>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bb4e77e0-8025-4dbd-8b71-8ec68e5e9dbd/ob_avoid.py

1. ob_avoid.py 의 avoid_obstacle_2 메소드 수정
turn_time 과 return_time 을 관리하는 조건문을 메소드의 상단으로 이동
finish_return_time 은 time.time() 으로 대체
2. chase_the_ball.py 의 FindBall 메소드 수정
end_angle 를 self.angle 로 대체
self.start_angle 과 self.angle 을 == 의 비교가 아닌, 범위를 이용한 비교로 수정
(angle 의 정확한 값 비교가 불가능하여 임의의 범위로 비교)
forward_flag 의 기능을 FindBall 메소드에 이관 후 삭제
3. ob_avoid.py 의 avoid_obstacle_2 메소드 수정
감지 범위(임의의 0.2 값 등)를 self.warning_zone 으로 수정
위험 범위를 self.dangerous_zone 으로 수정
stop_flag 를 받은 뒤 60도만큼 회전시키도록 전체적인 수정 (입사각을 90도에 가깝다고 가정, 벽으로부터 임의로 반사되어 나오는 각을 30도로 생각한다면 $180 - 90 - 30 = 60$ 도 만큼 회전을 주어야 한다고 가정)
stop_flag 가 True 로 바뀌는 조건문에 start_angle 값과 end_angle 을 설정하도록 수정
straight_process_done 조건문 안에서 done 플래그 두개를 바꾸어 두단계를 뛰어넘고 stop_flag 조건문을 실행하도록 수정
4. 현재까지 임의의 각도로 정의된 각도 값을 - 3.1415 ~ 3.1415 범위로 normalize 하는 코드를 삽입 (odometry 의 콜백에서 받아오는 angle 값이 해당 범위 안으로 normalize 되기 때문입니다)
5. chase_the_ball.py 의 변수 추가

findBall_forward_flag, start_time 추가

findBall_forward_flag 를 chase_ball 의 findBall 메소드의 회전과 전진을 반복하는 조건으로 활용

start_time 을 chase_ball 의 findBall 메소드의 전진을 위한 조건으로 활용

5초간 전진 (1m 전진을 목표로 함, 0.2 m/s), 한 바퀴 회전의 반복

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bb7b91ea-7d0a-4e0b-b054-0092de1f70f8/chase_the_ball.py

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/554b1f84-7996-4bb0-9729-cbd7ba14e277/main.py>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/04dbea3a-63bf-4509-949a-c2e838a3cb1d/ob_avoid.py

장애물 회피 시퀀스에서 벽과 장애물의 구분을 다른 방식으로 구현해야

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/07908452-eb44-4780-accb-9e06d44764ca/ob_avoid.py