

창업연계공학설계입문 AD Project

3분반 3조

INDEX

문제 정의



요구 사항

구현해야 할 기능
기능적 요구사항
비기능적 요구사항

시스템 구현

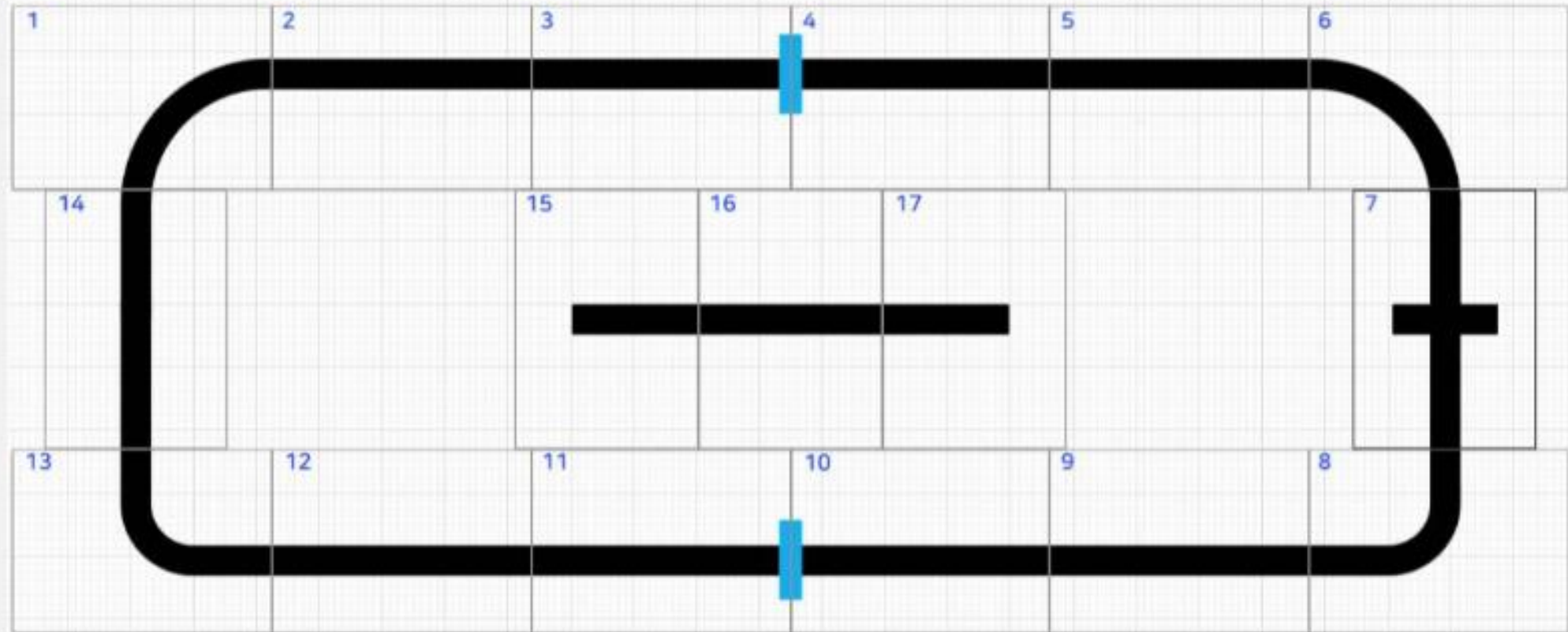


시험 및 평가



1. 문제 정의

주어진 트랙을 기본으로,
각 조의 창의성을 발휘하여 자율 미션을 수행!!!!



2. 요구 사항 분석

구현해야 할 기능

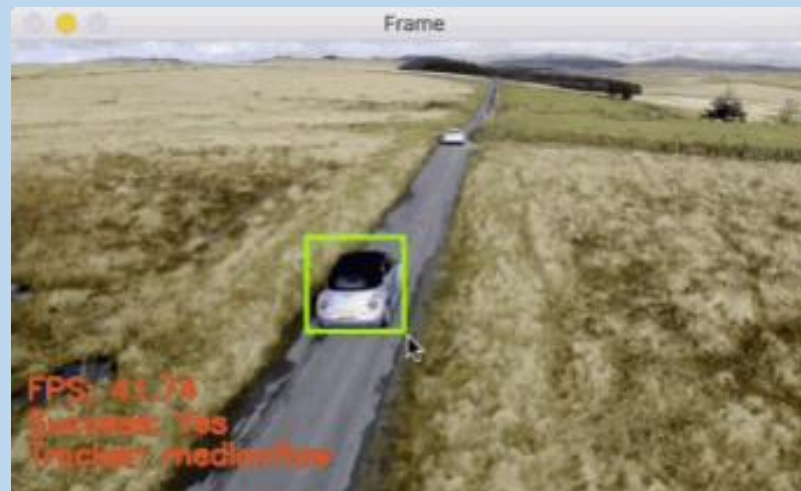
- **오브젝트 트래킹** : 자동차가 특정한 물체를 추적

기능적 요구 사항

- 자동차가 특정 물체를 인식해야 함
- 물체가 움직이는 방향을 따라 자동차가 주행해야 함

비기능적 요구 사항

- 파이썬을 이용함
- 물체가 카메라에 잘 인식되게 하기 위한 환경 조성



3. 시스템 구현

```
def conv_image(self, data):
    if self.ros_node:
        self.cam_img = self.bridge.imgmsg_to_cv2(data, 'bgr8')
        self.recorder.write(self.cam_img)
    else:
        self.cam_img = data
    gray = cv2.cvtColor(self.cam_img, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    edge = cv2.Canny(blur, 100, 150)
    mask = np.zeros_like(edge)
    if len(edge.shape) > 2:
        self.color = self.color3
    else:
        self.color = self.color1

    roi = self.cam_img[self.roi_vertical_pos:self.roi_vertical_pos + self.scan_height, :]
    self.cam_img = cv2.rectangle(self.cam_img, (0, self.roi_vertical_pos),
                                (self.image_width - 1, self.roi_vertical_pos + self.scan_height),
                                (255, 0, 0), 3)

    v = self.roi_vertical_pos
    roi = self.cam_img[v:v + self.scan_height, :]
    hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)

    avg_value = np.average(hsv[:, :, 2])
    value_threshold = avg_value + 1.0

    lbound = np.array([0, 0, 200], dtype=np.uint8)
    ubound = np.array([100, 255, 255], dtype=np.uint8)
    self.edge = cv2.inRange(hsv, lbound, ubound)
```

```
self.edge = cv2.inRange(hsv, lbound, ubound)

def detect_lines(self):
    self.left, self.right = -1, -1

    bin = self.edge

    self.view = cv2.cvtColor(bin, cv2.COLOR_GRAY2BGR)

    for l in range(self.area_width + 30, self.lmid):
        area = bin[self.row_begin:self.row_end, l - self.area_width:l]
        if cv2.countNonZero(area) > self.pxl_cnt_threshold:
            self.left = l
            break

    for r in range(self.image_width - 30 - self.area_width, self.rmid, -1):
        area = bin[self.row_begin:self.row_end, r:r + self.area_width]
        if cv2.countNonZero(area) > self.pxl_cnt_threshold:
            self.right = r
            break

    if self.left != -1:
        lsquare = cv2.rectangle(self.view, (self.left - self.area_width, self.row_begin), (self.left, self.row_end),
                                (0, 255, 0), 3)

    if self.right != -1:
        rsquare = cv2.rectangle(self.view, (self.right, self.row_begin),
                                (self.right + self.area_width, self.row_end), (0, 255, 0), 3)
```

4. 시험 및 평가

구현의 편리를 위해 물체의 색을 흰색으로 결정

**>> 자율 주행 스튜디오 벽이 대부분 흰색이라
물체 인식이 잘 안 되는 경우 발생**

>> 벽이 인식되지 않도록 ROI 범위를 수정

차량 교체로 인해 go4 파일의 부재로 시연에 어려움 발생

감사합니다