

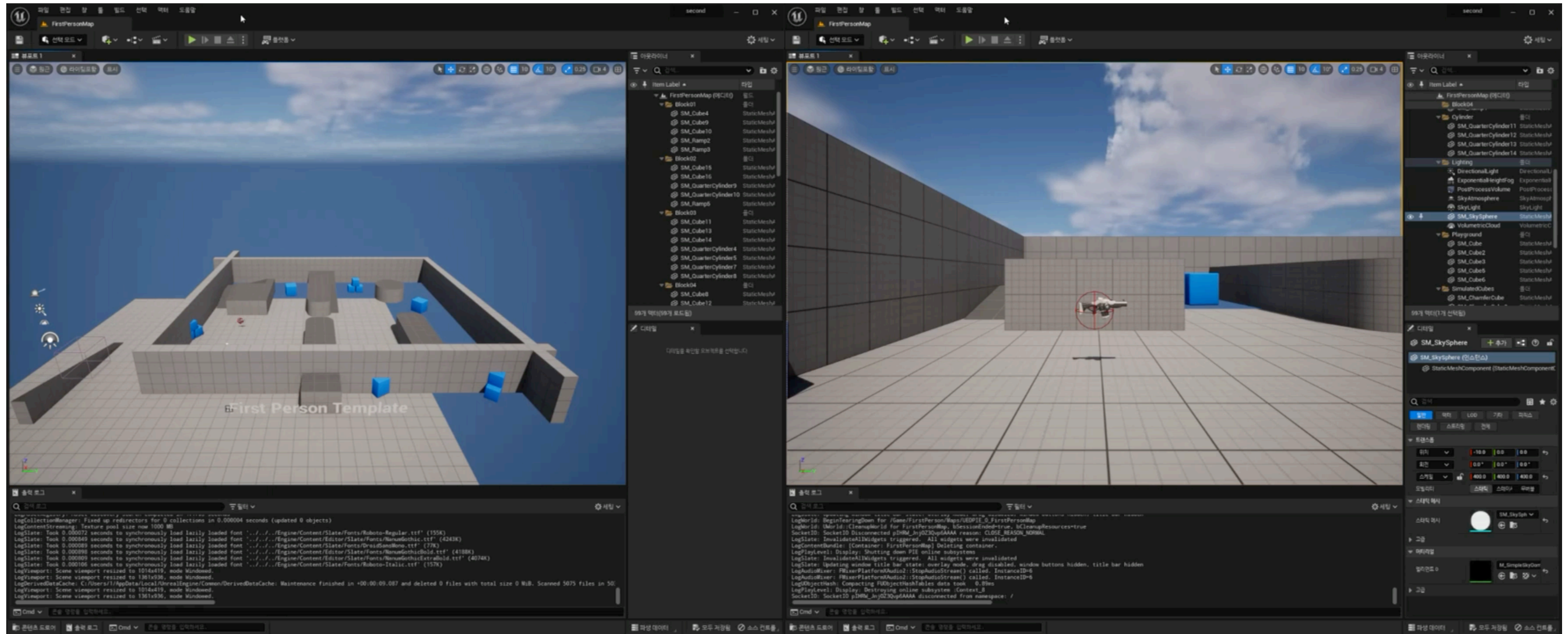
# Redirected Walking

19101188 고은수

# Todo

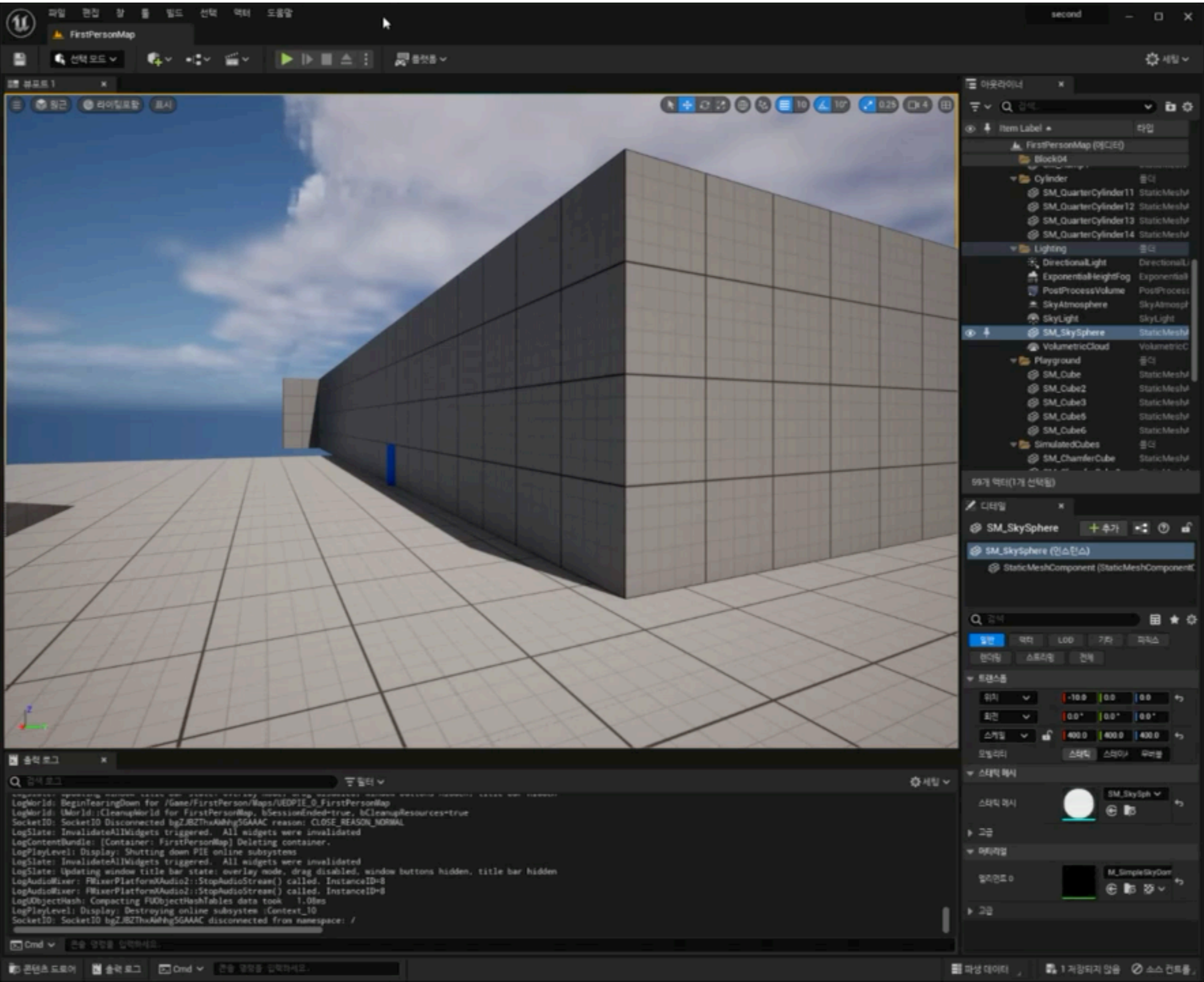
1. BoT-SORT 카메라 연동 테스트
2. 서버에서 위치 뿐만이 아니라 방향도 전송할 수 있도록
3. 여러 사람의 평균 위치를 구해서 사용하도록
4. 위의 코드를 어떻게 프로젝트에 적용할 수 있을지

# 서버에서 사람이 바라보는 방향으로 방향 전환





# 서버에서 사람이 바라보는 방향으로 방향 전환



# 평균위치 전송

```
def plot_tracking_bottom(image, tlwhs, obj_ids, scores=None, frame_id=0, fps=0., ids2=None):
    im = np.ascontiguousarray(np.copy(image))
    im_h, im_w = im.shape[:2]

    top_view = np.zeros([im_w, im_w, 3], dtype=np.uint8) + 255

    #text_scale = max(1, image.shape[1] / 1600.)
    #text_thickness = 2
    #line_thickness = max(1, int(image.shape[1] / 500.))
    text_scale = 2
    text_thickness = 2
    line_thickness = 3

    radius = max(5, int(im_w/140.))
    cv2.putText(im, 'frame: %d fps: %.2f num: %d' % (frame_id, fps, len(tlwhs)),
                (0, int(15 * text_scale)), cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 255), thickness=2)
    coordinates = []
    text = ''
    for i, tlwh in enumerate(tlwhs):
        x1, y1, w, h = tlwh
        intbox = tuple(map(float, (x1, y1, x1 + w, y1 + h))) # x_start, y_start, x_end, y_end
        bbox = cv2.perspectiveTransform(
            np.array([[(intbox[0]+intbox[2])//2, intbox[3]]]) # (x_center, y_end) projection by h_matrix
            , h_matrix).reshape((2)) # x는 중심으로 y는 하단 값으로 넣고 projection 시킴
        text += f'{{float(bbox[0])}} {{float(bbox[1])}},'
        intbox = tuple(map(int, (x1, y1, x1 + w, y1 + h)))
        obj_id = int(obj_ids[i])
        # id_text = '{}'.format(int(obj_id))
        id_text = f'{{obj_id}}, xy: {{round(bbox[0], 2)}} {{round(bbox[1], 2)}}'
        if ids2 is not None:
            id_text = id_text + ', {}'.format(int(ids2[i]))
        color = get_color(abs(obj_id))
        cv2.circle(im, ((intbox[0]+intbox[2])//2, intbox[3]), radius=line_thickness*3, color=color, thickness=-1)
        # cv2.rectangle(im, intbox[0:2], intbox[2:4], color=color, thickness=line_thickness)
        cv2.putText(im, id_text, ((intbox[0]+intbox[2])//2, intbox[3]), cv2.FONT_HERSHEY_PLAIN, text_scale, (0, 0, 255),
                    thickness=text_thickness)
        coordinates.append((float(bbox[0]), float(bbox[1])))
    f.write(f'{{text}}\n')
    x_mean = sum(coord[0] for coord in coordinates) / len(coordinates)
    y_mean = sum(coord[1] for coord in coordinates) / len(coordinates)
    return im, float(x_mean), float(y_mean)
```

# Todo

1. RDW 알고리즘 분석 및 적용
2. 카메라의 영상을 언리얼에서 시뮬레이션

감사합니다