

2020년 한이음 ICT멘토링 프로젝트 중간점검 추진(안)

□ 추진개요

○ 목 적

- 프로젝트 진도점검을 통한 멘토링 내실화 및 프로젝트 품질 제고
- 프로젝트 개발계획 및 완료계획 수립을 지원하고, 수행과정 상의 애로사항 및 요청사항 등 문제해결을 통해 프로젝트의 성공적 수행을 지원

○ 대 상 : 2020년 한이음 ICT멘토링 프로젝트 수행 팀 (총 605개 팀)

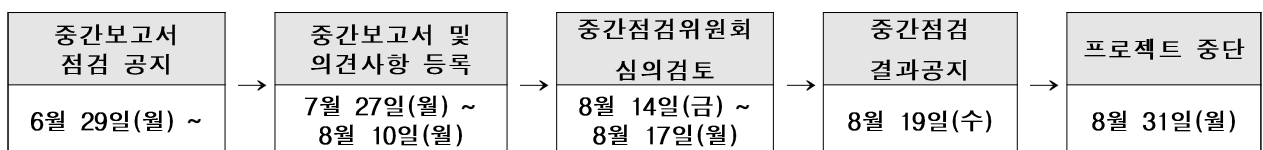
○ 일 정 : 2020년 7월 27일(월) ~ 8월 10일(월), 2주 간

□ 주요내용

○ 점검내용

- 프로젝트 중간보고서, 착수·분석 및 설계단계 산출물, 팀 별 멘토링 활동 여부 (회의록, Git test, 코딩테스트, 블렌디드 러닝) 애로사항 및 요청사항 등

○ 점검절차 및 내용



- (중간보고서 등록) 프로젝트 수행단계별 진행률, 주요 변경사항, 기대효과 및 개선사항 등 작성

* 한이음 사이트 → 로그인 → 마이페이지 → 팀 블로그 → 보고서 게시판 → 중간보고서 제출

- (의견사항) 프로젝트 문제점 및 애로사항, 요청사항 등

○ 점검결과

- 프로젝트별 최종점수 고득점 순으로 계속 진행 또는 중단 조치 결정

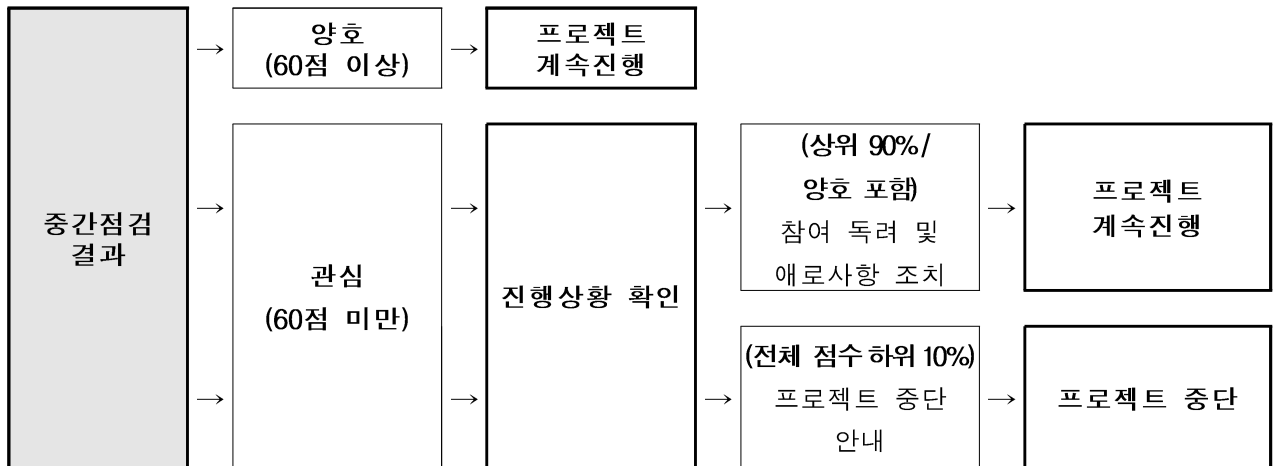
* 점검항목 및 배점은 공개하며, 결과 점수는 외부로 공개하지 않음

- 점수 60점 이상, 프로젝트 계속 진행
- 점수 60점 미만, 진행상황 확인 후 진행 가능 팀은 참여 독려 및 애로사항 조치, 프로젝트 계속 진행

* 60점은 프로젝트 중위 점수임

- 점수 하위 10%, 점수와 관계없이 프로젝트 중단

* 학점연계형 프로젝트의 경우, 학사일정을 반영하여 프로젝트 중단여부 결정



[붙임1] 한이음 ICT멘토링 프로젝트 정성 / 정량지표에 따른 항목별 배점 안내 1부

[붙임2] 한이음 ICT멘토링 프로젝트 중간보고서 서식 1부

[붙임3] SW/HW 제작설계서 양식 1부

□ 중간점검 FAQ

Q 중간점검 결과가 미흡하면 프로젝트가 중단(탈락)되나요?

A - 네 그렇습니다. 프로젝트별 최종점수 고득점 순으로 계속 진행 또는 중단 여부가 결정됩니다. (하위 10% 탈락)

Q 중간보고서를 제출하지 않게 되면 어떻게 되나요?

A - 중간보고서를 제출하지 않으면 프로젝트 개발 및 관리에 배점을 받을 수 없으므로 프로젝트에서 탈락(중단)할 수 있습니다.

Q 중간보고서 작성은 누가 하나요?

A - 팀원 멘티가 작성하여도 무방하나, 이후 취합하여 팀장 멘티가 제출해야 합니다. 또한, 멘토는 작성이 불가합니다.

Q 중간보고서의 모든 항목은 필수 입력인가요?

A - 기본적으로 모든 항목은 필수 입력이나 해당되는 부분이 아닐 경우에는 해당사항 없음이라고 기재해주시면 됩니다.

Q 중간보고서에 중간산출물을 등록하는 곳이 있는데요. 중간산출물이 아직 없을 경우에는 어떻게 하나요?

A - 중간산출물 양식을 보시면 시스템 구성도 등 기본적인 항목도 있습니다. 중간산출물이 아직 없더라도 기본적인 부분만 작성 후 등록하시면 됩니다. 중간산출물 내 모든 부분을 필수로 입력해야 하는 것은 아니며 해당되는 부분만 작성하여 등록하시면 됩니다.

Q 중간보고서 최종 제출 후 수정이 가능한가요?

A - 최종 제출 후에는 수정이 불가합니다. 제출기간까지 충분히 수정하신 후 최종 제출해 주시기 바랍니다.

[붙임1] 한이음 ICT멘토링 프로젝트 정성 / 정량지표에 따른 항목별 배점 안내

구분	점검항목	세부내용	배점
프로젝트 개발 (50)	가. 중간보고서 제출여부 * 중간산출물 포함	중간보고서 제출여부에 따른 배점 (설계포함) * 미제출(0점), 부분 제출(15점), 제출(20점)	20
	나. 일정 계획 적절성	수행계획대비 진행률에 따라 배점 * 매우 부적절(0점), 부적절(4점), 보통(6점), 적절(8점), 매우 적절(10점)	10
	다. 개발(구현)가능성	수행계획대비 개발 가능성 * 매우 부적절(0점), 부적절(4점), 보통(6점), 적절(8점), 매우 적절(10점)	10
	라. 문서 완성도	문서완성도 및 충실성 * 매우 불성실(0점), 불성실(4점), 보통(6점), 성실(8점), 매우 성실(10점)	10
프로젝트 관리 (50)	마. 의사소통 및 미팅	온/오프라인 미팅 회의록에 따른 배점 * 미진행(0점), 1~3회(6점), 4~6회(9점), 7~9회(12점), 10회 이상(15점)	15
	바. 멘토링 활동보고서	멘토링 활동보고서 등록건수에 따라 배점 * 미진행(0점), 1회(4점), 2회(8점), 3회(12점), 4회 이상(15점)	15
	사. 프로젝트 수행충실성	Git, 블렌디드러닝, 클라우드, 실습장비, 각종 교육 등 한이음 프로젝트 참여도 * 매우 불성실(0점), 불성실(5점), 보통(10점), 성실(15점), 매우 성실(20점)	20
의견사항	문제점 및 애로사항	팀 별 문제점 및 애로사항 파악	-
	요청사항	프로젝트 진행 간 필요 요청사항 파악	-
합 계			100

* 정량지표는 상황에 따라 변경될 수 있음

□ 상세설명

점검항목	세부내용	비고
가. 중간보고서 제출여부 * 중간산출물 포함	중간보고서 제출여부에 따른 배점 (설계포함) * 미제출(0점), 부분 제출(15점), 제출(20점)	자동 산정

* 중간보고서 제출여부 : 프로젝트 중간보고서 산출여부에 따른 배점이며, 중간보고서 문서만 제출하는 경우 15점, ‘SW/HW_제작설계서’를 함께 첨부하는 경우 20점 배점

점검항목	세부내용	비고
나. 일정 계획 적절성	수행계획대비 진행률에 따라 배점 * 매우 부적절(0점), 부적절(4점), 보통(6점), 적절(8점), 매우 적절(10점)	심의 평가

예시) 수행계획서 일정 계획 (기제출)											
나. 프로젝트 추진일정 # 프로젝트 기간은 노란색 셀 색상으로 표시, 필요 시 줄 추가											
구분	추진내용	추진일정									
계획	아이디어 회의 및 세부계획 수립, 역할분담	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월
분석	플랫폼 선정, 제품 조사, 호환성 분석										
설계	모바일 로봇 플랫폼 설계										
	Object Detection 알고리즘 설계										
	광고 추천 및 송출 알고리즘 설계										
	모바일 로봇 경로이동 알고리즘 설계										
개발	모바일 로봇 외형 모델 개발										
	모바일 로봇 조립										
	Object Detection API를 활용한 고객 인식 알고리즘 개발										
	광고 추천 및 송출 알고리즘 개발										
테스트	모듈 테스트 및 호환성 검사										
	최종 보고서 작성 및 한이음 공모전 참가										
요약/마무리	주 1회 이상 팀원들과 미팅										



예시) 중간보고서 일정 계획											
프로젝트 기간											
구분	추진내용	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월
계획	아이디어 회의 및 세부계획 수립, 역할분담										
분석	플랫폼 선정, 제품 조사, 호환성 분석										
설계	모바일 로봇 플랫폼 설계										
	Object Detection 알고리즘 설계										
	광고 추천 및 송출 알고리즘 설계										
	모바일 로봇 경로이동 알고리즘 설계										
개발	모바일 로봇 외형 모델 개발										
	모바일 로봇 조립										
	Object Detection API를 활용한 고객 인식 알고리즘 개발										
	광고 추천 및 송출 알고리즘 개발										
테스트	모바일 로봇 경로설정 및 이동 알고리즘 개발										
	모듈 테스트 및 호환성 검사										
종료	최종 보고서 작성 및 한이음 공모전 참가										

* 일정 계획 적절성 : 수행계획서 대비 중간보고서 상의 일정 진행(변경) 정도를 비교하여 배점 (중간보고서 일정 계획만 작성하여 제출하면 되며, 심사위원이 수행계획서 일정과 비교하여 배점)

점검항목	세부내용	비고
다. 개발(구현)가능성	수행계획대비 개발 가능성 * 매우 부적절(0점), 부적절(4점), 보통(6점), 적절(8점), 매우 적절(10점)	심의 평가

예시) 수행계획서 개발(구현) 가능성 (기제출)		
가. 주요 기능		
구분	기능	설명
S/W	고객이미지캡처	카메라 센서를 이용하여 복합 쇼핑몰 내의 고객들의 이미지를 캡처한다.
	고객정보추출	캡처한 이미지 속 고객들을 Face API를 이용해 Detection하고, 고객들의 정보(성별, 나이, 감정 등)를 추출한다.
	광고 추천	추출된 고객 정보를 기반으로 고객들에게 알맞은 광고를 추천한다.
	지도작성	Depth Sensor에서 Depth정보를 받아서 ROS & SLAM기술을 이용해 로봇이 이동하기 위해 쇼핑몰 내의 지도를 작성한다.
H/W	모바일 로봇	아두이노와 ROS & SLAM 기술, ROS Serial 통신을 이용하여 쇼핑몰 안에서 로봇을 자유롭게 이동하게 한다.
	스크린	맞춤 광고 알고리즘에 의해 선택된 광고를 로봇 스크린과 매장 스크린에 Display한다.

라. 예상결과물

예상 결과물 이미지	설명
	<p>모바일 로봇:</p> <ul style="list-style-type: none"> - 쇼핑몰 안을 돌아다니며, 모바일 로봇 안에 있는 카메라 센서로 고객들을 Detection하고, 각 고객들의 정보를 추출한다. - Face API를 이용해 Detection 및 정보를 분류하고, 분류된 정보를 분석하여, 적절한 광고를 선택하여 로봇의 스크린에 Display한다.
	<p>쇼핑몰 스크린:</p> <ul style="list-style-type: none"> - 모바일 로봇에서 고객 맞춤형 광고 알고리즘에 의해 추출된 광고를 매장 스크린에도 같이 송출한다.

예시) 중간보고서 개발(구현) 가능성

가. 주요 기능

기능	설명	작동실패사진
로봇 속도 제어	ROS 토픽을 통해 로봇의 제어 명령과 x, y 좌표, 회전 속도를 제어한다.	<pre>void Byu_Ch(const geometry_msgs::Transform& byu_val) // 경멸 회 빙에서 byu robot 제어 { motor_forward_speed = byu_val.translation.x; if (motor_forward_speed > forward_speed_max) motor_forward_speed = forward_speed_max; motor_forward_speed = -motor_forward_speed; motor_moveToSideL_speed = byu_val.translation.y; if (motor_moveToSideL_speed > side_speed_max) motor_moveToSideL_speed = side_speed_max; motor_moveToSideL_speed = -motor_moveToSideL_speed; motor_turnR_speed = byu_val.rotation.z; if (motor_turnR_speed > turn_speed_max) motor_turnR_speed = turn_speed_max; motor_turnR_speed = -motor_turnR_speed; start_robot = true; robot_mode = (int)byu_val.translation.z; } }</pre>
장애물 회피 및 낙상 방지	IR 센서 데이터를 이용해 낙상을 방지하고, 로컬화 센서 6개를 이용해 전방 180도 반경 안의 장애물을 감지하여 회피하는 로봇을 제어한다.	<pre>void Collision_Avoidance() { if (IR_sensor_check == true) { drive_line_body_frame(0, 0, 0, 100); drive_line_body_frame(motor_forward_speed, 0, 0, 2000); //포진 drive_line_body_frame(0, 0, motor_turnR_speed, turn_delay * 6); //회전 } else { if (distance_F <= front_detection_distance) { //전방에 장애물이 감지되면 drive_line_body_frame(motor_forward_speed * f_decay_rate , 0, 0, 0); //속도감소 if (distance_F <= front_detection_distance2) { //정면이 막혔을 때 drive_line_body_frame(0, 0, 0, front_delay); //회전 front_check += 1; if (front_check >= 50) { // front_check +50 -> +1s drive_line_body_frame(0, 0, motor_turnR_speed, turn_delay * 6); front_check = 0; } Avoidance_Check(); } } else { //전방의 장애물 감지 관료연속 된, 측면이 안 막혔을 때 Avoidance_Check(); } } drive_line_body_frame(motor_forward_speed, 0, 0, 0); // 정지상태 일수록 } } void Avoidance_Check() { if (distance_L <= left_detection_distance && distance_R <= right_detection_distance) { //양측이 부딪기 후로고된, 앞 뒤로는 0인 drive_line_body_frame(0, 0, motor_turnR_speed, turn_delay); front_check = 0; } else if (distance_L <= left_detection_distance) { //왼쪽 방향은 주파수로 회전 drive_line_body_frame(0, 0, motor_turnR_speed, turn_delay); front_check = 0; } else if (distance_R <= right_detection_distance) { //오른쪽 방향은 주파수로 회전 drive_line_body_frame(0, 0, motor_turnR_speed, turn_delay); front_check = 0; } if (distance_SideL <= sideL_detection_distance) { //위 리터에 회전 한 후에 측면에 장애물이 있으면 감속 주파수로 이동하거나 정지함 drive_line_body_frame(0, 0, motor_moveToSideL_speed , 0, side_delay); front_check = 0; } if (distance_SideR <= sideR_detection_distance) { //옆쪽으로 움직이는 것보다 교차로에 가까워지면 감속 주파수나 다시 좌우 움직임대응 drive_line_body_frame(0, 0, motor_moveToSideL_speed * side_decay_rate, 0, side_delay); } else if (distance_SideL <= sideL_detection_distance) { //앞 뒤 리터에 회전 한 후에 주파수 감속으로 이동하거나 정지함 drive_line_body_frame(0, 0, motor_moveToSideL_speed , 0, side_delay); front_check = 0; } if (distance_SideR <= sideR_detection_distance) { //옆쪽으로 움직이는 것보다 교차로에 가까워지면 감속 주파수나 다시 좌우 움직임대응 drive_line_body_frame(0, 0, motor_moveToSideL_speed * side_decay_rate, 0, side_delay); } } }</pre>

라. 예상결과물

- MS Azure의 Face API를 이용한 얼굴 검출 및 특징 분류



MS Azure의 Face API(Python)를 이용하여 얼굴을 인식하여 성별, 나이 Feature Data를 얻는다. 나이, 성별 Feature Data는 딕셔너리로 저장된다. 실제 카메라로 받은 이미지 속 얼굴을 검출하고, 특징을 추출 후 각각 Feature Data는 [성별, 연령대별]로 분류한다.

- MongoDB

이미지로부터 얻은 시간, 장소, 고객들의 얼굴의 나이, 성별 Feature Data를 DB에 저장한다. 현재 장소는 임의로 정하였고, 로봇을 구동할 장소로 지정한다.

- 모바일 로봇

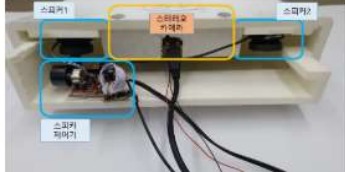
- 하체부



- 허리부



- 머리부



* 개발(구현) 가능성 : 수행계획서 대비 주요기능, 예상결과물 등의 비교를 통한 개발(구현) 가능성 (중간보고서에만 작성하여 제출하면 되며, 심사위원이 수행계획서와 비교하여 배점)

점검 항목	세부내용	비고
라. 문서 완성도	문서완성도 및 충실성 * 매우 불성실(0점), 불성실(4점), 보통(6점), 성실(8점), 매우 성실(10점)	심의 평가

* 문서완성도 : 중간보고서 문서 전체 작성의 완성도 및 충실성을 평가하여 배점

점검 항목	세부내용	비고
마. 의사소통 및 미팅	온/오프라인 미팅 회의록에 따른 배점 * 미진행(0점), 1~3회(6점), 4~6회(9점), 7~9회(12점), 10회 이상(15점)	자동 산정

* 의사소통 및 미팅 : 한이음 사이트에 등록된 온/오프라인 회의록 수를 산정하여 배점

점검항목	세부내용	비고
바. 멘토링 활동보고서	멘토링 활동보고서 등록건수에 따라 배점 * 미진행(0점), 1회(4점), 2회(8점), 3회(12점), 4회 이상(15점)	자동 산정

* 멘토링 활동보고서 : 한이음 사이트에 등록된 멘토링 활동보고서 수를 산정하여 배점

점검항목	세부내용	비고
사. 프로젝트 수행충실성	Git, 블렌디드러닝, 클라우드, 실습장비, 각종 교육 등 한이음 프로젝트 참여도 * 매우 불성실(0점), 불성실(5점), 보통(10점), 성실(15점), 매우 성실(20점)	심의 평가

<프로젝트 관리 - 프로젝트 수행충실성 (표)>

구 분	세부내역	월별 이용내역				합 계	비고
		4월	5월	6월	7월		
Git	Commit						
	Branch						
	Push						
	Merge						
	Issues						
	Star						
	Fork						
팀별 지원내역	클라우드 비용						
	실습장비 비용						
	회의실 이용						
	교통비 이용						
에뮬레이션	팀 참여 수						
온/오프라인 교육	팀 참여 수						
블렌디드 러닝	팀 배지 수						
Git 테스트 참여	팀 참여 수						
코딩 테스트 참여	팀 참여 수						

* 프로젝트 수행충실성 : 프로젝트 수행충실성(표)를 참고하여 심사위원이 배점
(Git, 팀별 지원금 신청내역 등은 프로젝트 수행의 참고자료이며, 해당 자료를 바탕으로 심의위원이
정성적 종합 평가함. 단순 이용횟수나 금액이 많다고 좋은 점수를 배점하지 않음)