ProtoType2 개발 내용

1) 좌표계/축 차이 동기화 (오른손/왼손, Z 부호)
- ReSpace 결과의 좌표계와 Unity 좌표계가 완전히 동일하지 않아서,
- Unity 배치 시 pos.z 부호 반전, Euler의 y/z 부호 반전 등 변환 규칙을 적용해야 했음(안 하면 배치가 거울처럼 뒤집히거나 방향이 틀어짐).

2) 레이아웃 배치 AI 모델의 의존성/환경 문제 해결 (Windows + GPU)
- ReSpace는 사실상 CUDA(GPU) + torch + transformers 가 정상이어야 추론이 가능하고, 이게 Windows에서 세팅/검증이 까다로웠음.
- 그래서 서버 파이썬 환경과 분리된 전용 venv(.respace_venv) 를 두고, 그 python으로 서브프로세스를 실행하는 방식으로 고립/안정화가 필요했음.
- Health 체크에서 “RESPACE_SG_LLM 존재 여부 / CUDA 가능 여부 / missing_dataset_env” 같은 상태 노출이 필요했음.

3) JSON 생성 로직 구체화
- LLM 출력에는 설명 텍스트/불필요 문자열이 섞일 수 있어서, 필요한 정보의 블록만 찾아 JSON으로 뽑는 로직이 구현.
- safe_parse_scene 가 실패할 수 있어 json.loads 로 fallback하는 등 파싱 안정화하여 구현.

4) 회전 값 정규화
- ReSpace/scenegraph의 rot이 [x,y,z,w] quaternion으로 나오는 케이스가 있음.
- Unity 툴에서는 Euler(3개)가 훨씬 다루기 쉬워서 Quaternion→Euler(deg) 변환 을 backend에서 구현.

ProtoType 개발 간 발생한 문제점

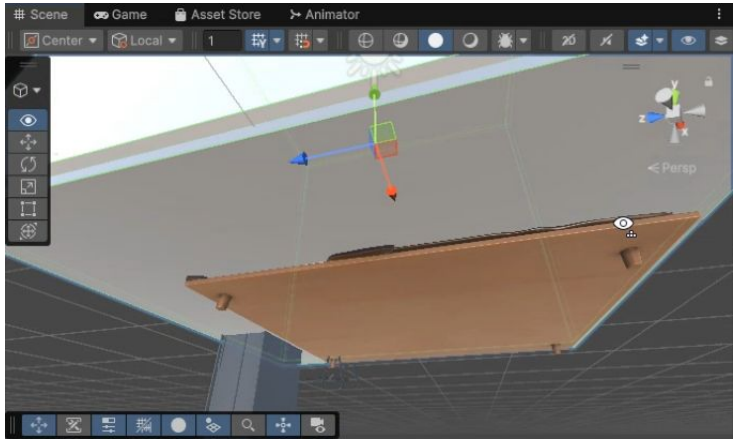
1. 가구 자체의 orientation의 후보정 로직 구현 필요
   - 가구 카테고리별 특징을 통한 방향 조정

2. 바닥면 접촉을 위한 배치 로직 수정 필요
   - scale의 1/2값 만큼 y축값 수정

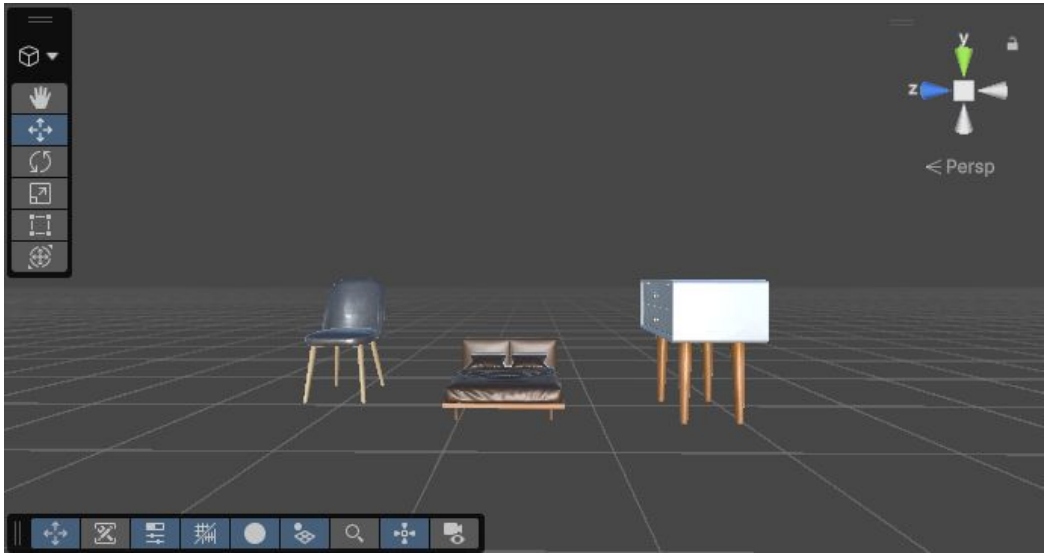
3. 3D 오브젝트를 생성할때의 분위기 키워드 반영 필요
   - Respace 모델의 LLM의 교체/추가학습 필요

4. 3D 오브젝트 생성 시 예상된 시간 초과
   - 오브젝트 1개당 약 2분 전후의 시간 소요됨.
   - API 사용 병렬화처리를 통한 최적화 필요

5. UI 구현



가구 자체의 orientation 후보정 필요성



가구 생성 시 일정한 방향으로 가구가 생성되지 않는 문제점 발생
예시로 의자의 rotation값의 (0,0,0) 과 침대의 rotation 값의 (0,0,0).

해결방안: 3D 모델의 검사를 통해 카테고리 별 일반적인 특징을 활용하여 방향성 결정하여 수정하는 후보정 로직 필요.

STP, SRS 수정

FR-1.4 LLM 출력값 Json 형식으로 전처리
유효한 출력값에서 가구 종류 및 수 등의 변수를 정수(Integer) 형태로 추출하여 Json 형식으로 정리해 후속 레이어로 전달해야 한다.

FR-1.5 API 통신 및 응답 처리
생성된 프롬프트를 통해 API를 호출하고 네트워크 타임아웃이나 API 할당량 초과(Quota Error) 발생에 대응하는 예외 처리 및 사용자 알림 기능을 수행해야 한다.

FR-1.6 API 반환값 정합성 검증
LLM이 출력한 결과물에서 단순히 JSON을 추출하는 것뿐만 아니라 정수(Integer)형태인지, 방 크기가 양수의 유효한 값인지 등 데이터 타입 유효성 검사(Type Check)를 수행해야 한다.

FR-1.7 .Json 파일 저장
상기 생성된 .Json 파일을 로컬에 저장한다.

FR-1.8 UI로 진행상황 전송
Input Module에서 .Json 파일을 저장하는 것으로 진행이 모두 끝났다면 자신의 진행상황이 끝났음을 알리는 시그널을 UI로 보낸다.

3.2.2 FR-2 Positioning Module

FR-2.1 Json 형식 Data 기반 배치 Model 사용
입력된 가구 카테고리 정보를 바탕으로 각 가구의 위치, 회전값 생성해야 한다.

FR-2.2 산출된 좌표 정보 검증 및 보정
생성된 좌표가 바닥이나 벽을 관통하지 않는지 검증 후 모든 가구의 한쪽 면이 방의 벽 적어도 한곳에 정확히 밀착 및 정착되어 있도록 보정을 한다.

FR-2.3 Collision 판단
두 가구 간의 Bounding Box가 겹칠 경우, 단순한 면 접촉(Edge Contact)은 허용하되 실제 충돌 시에는 반복(Iteration)을 통해 위치를 재산출해야 한다 .

FR-2.4 가구 수 판단
사용자가 입력한 가구의 수에 맞게 가구가 배치되어 있는가 판정해야 한다.

FR-2.5 반복 제한 및 예외 처리
가구 배치가 불가능한 한정된 공간에 수용가능한 개수 이상의 가구가 배치되는 경우 무한 루프에 빠지지 않도록 최대 반복 횟수를 제한하고, 초과 시 사용자에게 알림을 주거나 가능한 부분만 배치해야 한다

FR-2.6 방과 가구 크기의 정합성
입력된 방의 크기 대비 과도하게 큰 가구 배치가 요청될 경우 이를 인식하고 예외 처리해야 한다

FR-2.7.Json 파일 저장
3.2.1.5에 의해 생성된 .Json 파일에 새로운 Data를 추가하여 덮어쓰기 형식으로 써서 저장한다.

Contents