Term Project Report (Computer Graphics, Spring, 2025)

Title: Tank Simulation

학번: 2020142149 이름: 김사윤

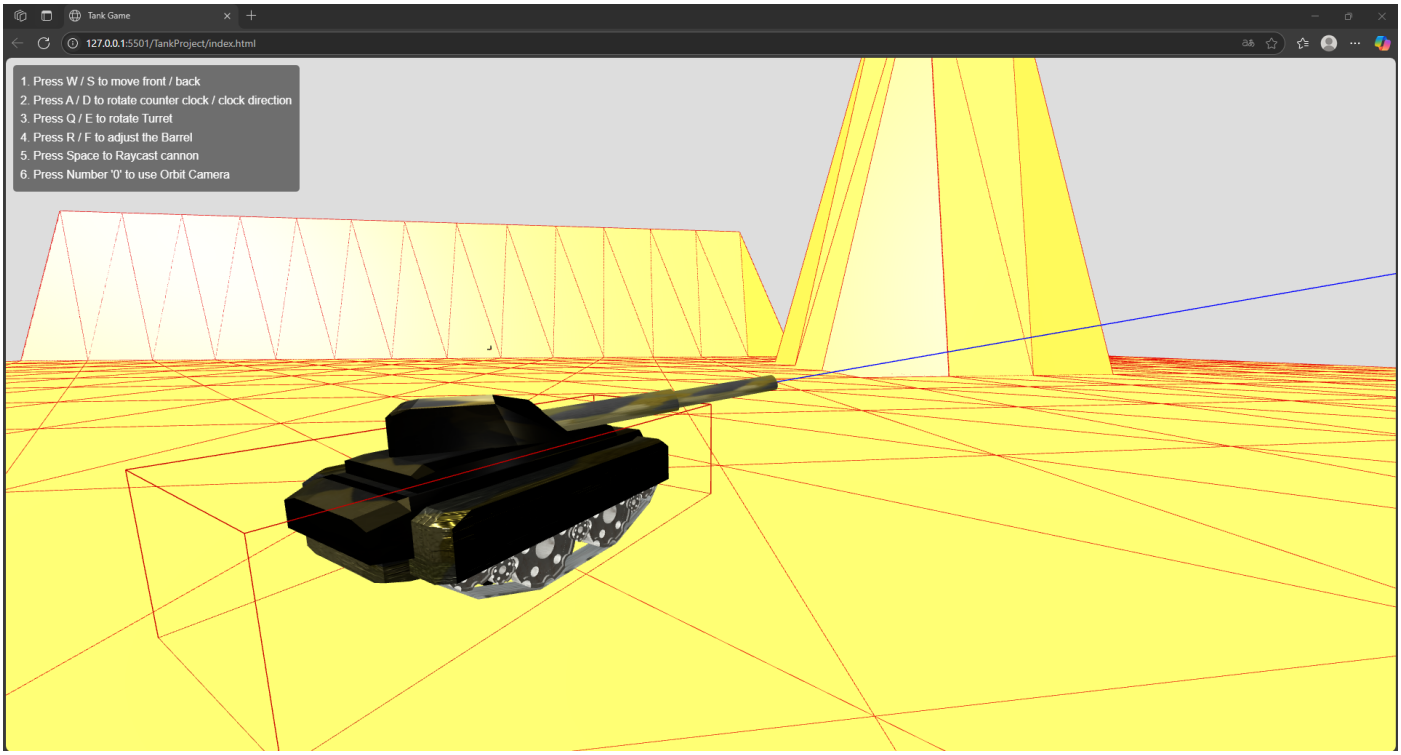
1. 요약:

<웹 기반 3D 탱크 게임>

Three.js와 Rapier.js를 활용하여 실제 물리 기반 탱크 조작, 포탄 발사 구현
탱크 조작: 전진/후진, 포탑·포신 회전, 바퀴 회전, 포신 위아래 조작, Spacebar 누를 시 raycast 실행
경사진 지형: 중력 및 충돌 대응
카메라 전환: 3인칭 ↔ Orbit Camera 시점 전환
* 18-scene-graph, 13-rapier-physics 참고
* Material을 잘 활용하여 탱크의 디테일을 보충

2. 사용법:

Key	기능	설명
W / S	앞/뒤로 이동	탱크를 앞으로 움직이거나 뒤로 움직입니다.
A / D	좌/우 회전	탱크를 시계 반대 방향 또는 시계 방향으로 회전시킵니다.
Q / E	포탑 회전	탱크의 포탑을 왼쪽 또는 오른쪽으로 회전시킵니다.
R / F	포신 상/하 조절	탱크의 포신(Barrel)을 위 또는 아래로 조절합니다.
Space	레이캐스트 (포 발사)	포신 방향으로 레이캐스트를 발사하여 충돌을 감지합니다.
숫자 '0'	시점 전환 (Orbit 카메라)	게임 카메라와 자유로운 Orbit 카메라 시점 간 전환합니다.



3. 기능 구현 표: (구현 기능 별로 구현 source code 위치를 표기, 아래 표는 example 임)

	기능	구현위치	비고
1	Three.js Scene, Camera, Renderer Setup	(line 6-27)	기본 씬, 카메라, 렌더러 초기화 및 설정
2	Three.js OrbitControls	(line 30-47)	카메라 제어 (Game Camera, Orbit Camera)
3	Three.js Lighting	(line 49-59)	Ambient, Directional, Point Light 설정
4	Rapier Physics World Initialization	(line 173-179)	물리 엔진 초기화 및 중력 설정
5	Rapier Debug Renderer	(line 181-187)	물리 콜라이더 시각화
6	GLB Model Loading (Map)	(line 195-248)	tank_map.glb 로드 및 콜라이더 생성
7	GLB Model Loading (Tank)	(line 254-320)	tankv1.glb (혹은 tankv3.glb으로 추정) 로드 및 콜라이더 생성
8	Keyboard Input Handling	(line 72-101)	키보드 입력 상태 관리 및 이벤트 리스너
9	Tank Movement Logic	(line 351-455)	W/S (전진/후진), A/D (회전) 및 바퀴 회전 애니메이션
10	Turret Rotation (Q/E)	(line 458-462)	포탑(Potap) 회전
11	Barrel Adjustment (R/F)	(line 465-478)	포신(PoshinRotateCore) 상/하 조절
12	Raycasting (Space)	(line 481-518)	포 발사 시 레이캐스트 및 시각화
13	Camera Follow Logic	(line 527-550)	탱크에 따른 카메라 위치 및 시점 조정
14	Camera Switching (0)	(line 331-346)	게임 카메라와 Orbit 카메라 전환
15	Window Resize Handling	(line 573-579)	화면 크기 변경 시 렌더러 및 카메라 업데이트

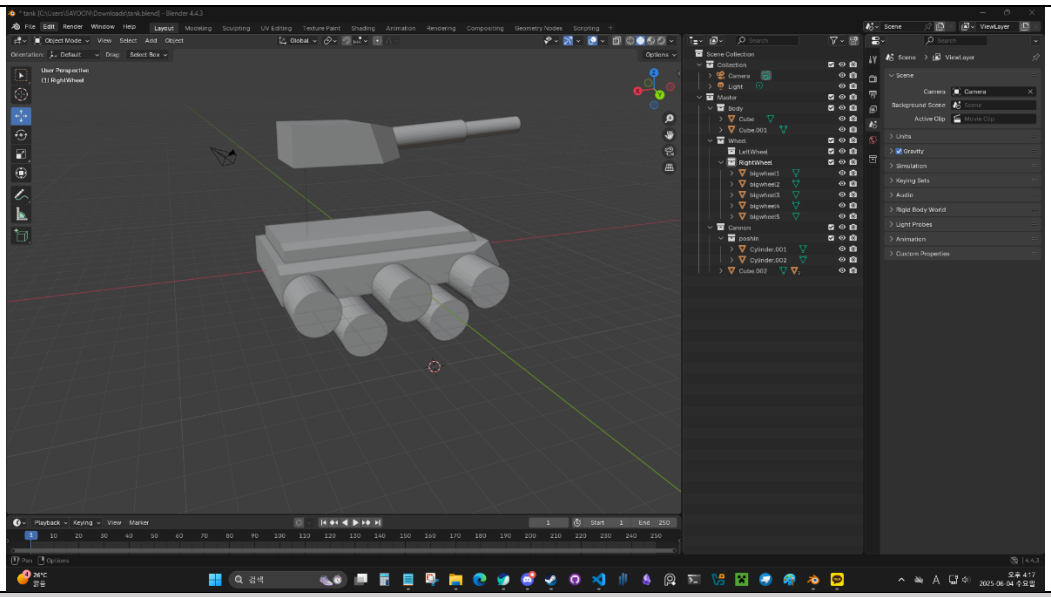
4. 기타 사용 기능

기술 스택 레퍼런스 링크

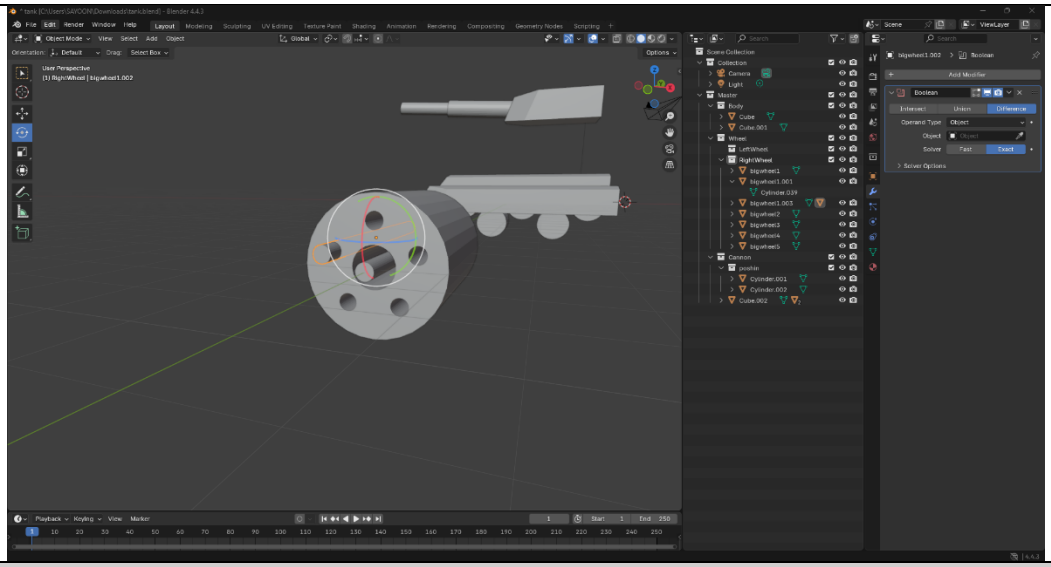
- Three.js 공식 문서: Three.js - JavaScript 3D Library
- Rapier.js 공식 문서: Rapier Physics Engine
- Three.js GLTFLoader 문서: GLTFLoader - three.js docs
- Three.js OrbitControls 문서: OrbitControls - three.js docs

5. 기타

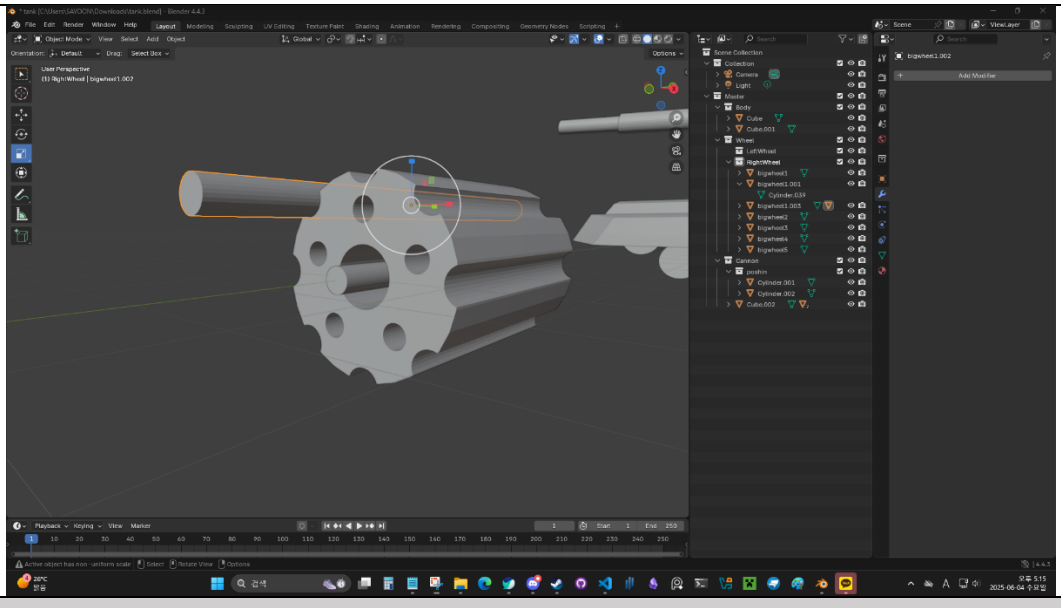
[모델링 과정 사진입니다.]



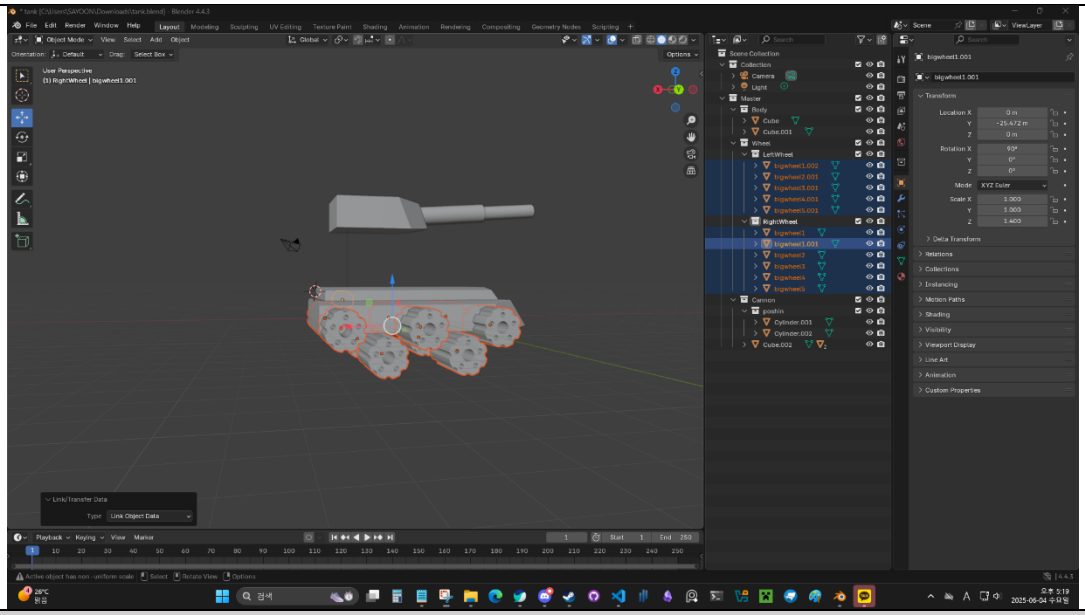
기본 골조



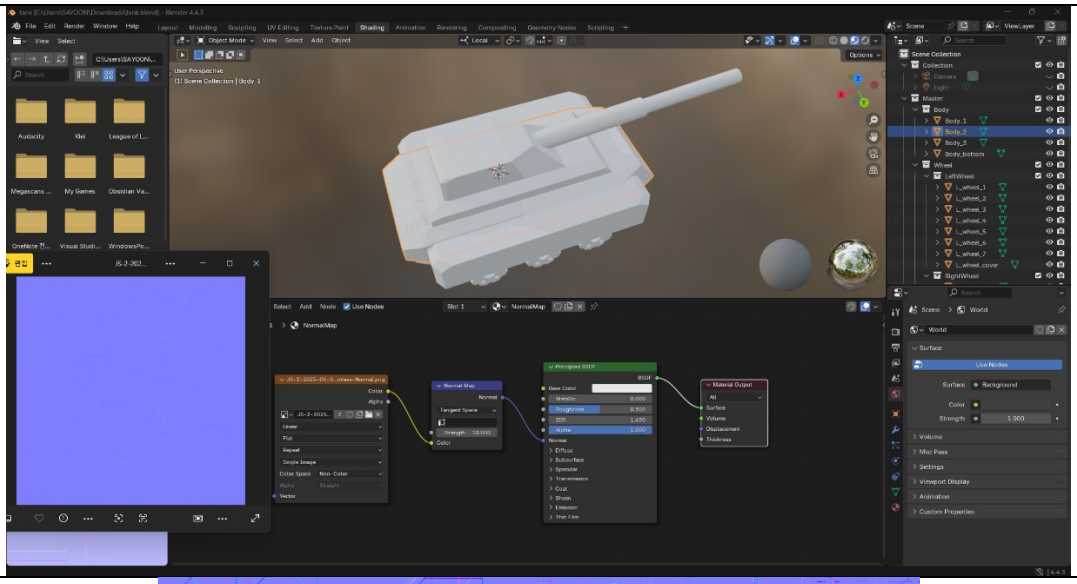
바퀴 구멍 뚫기

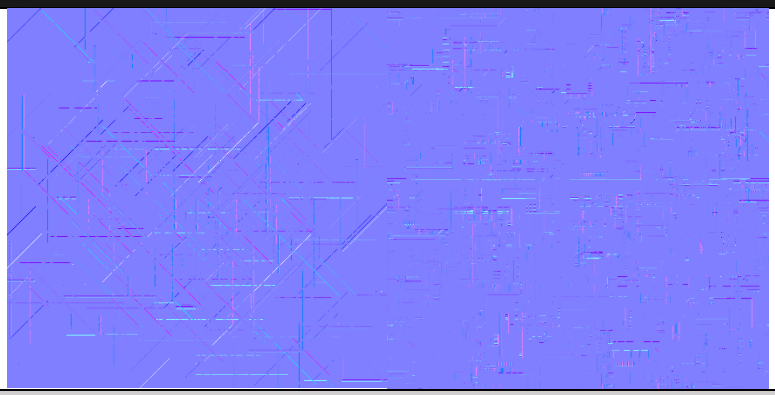


바퀴 톱니 모양 깎기 (boolean difference)

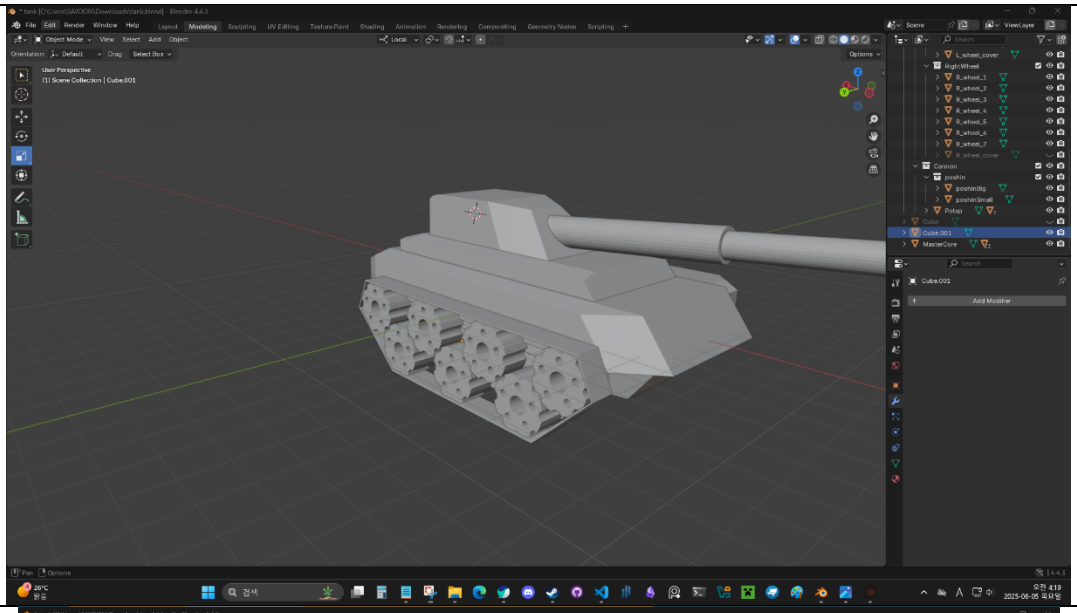


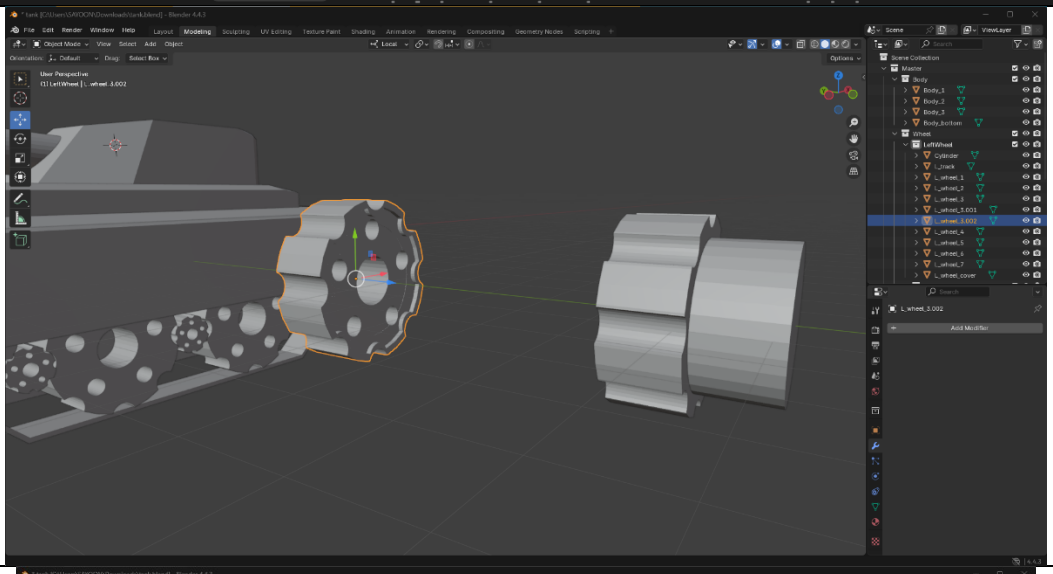
바퀴 배치

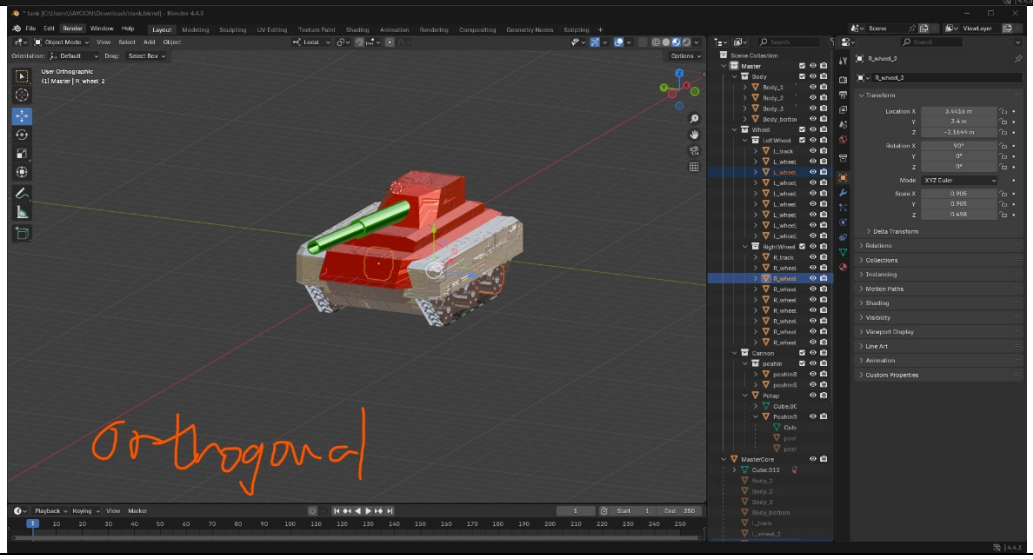




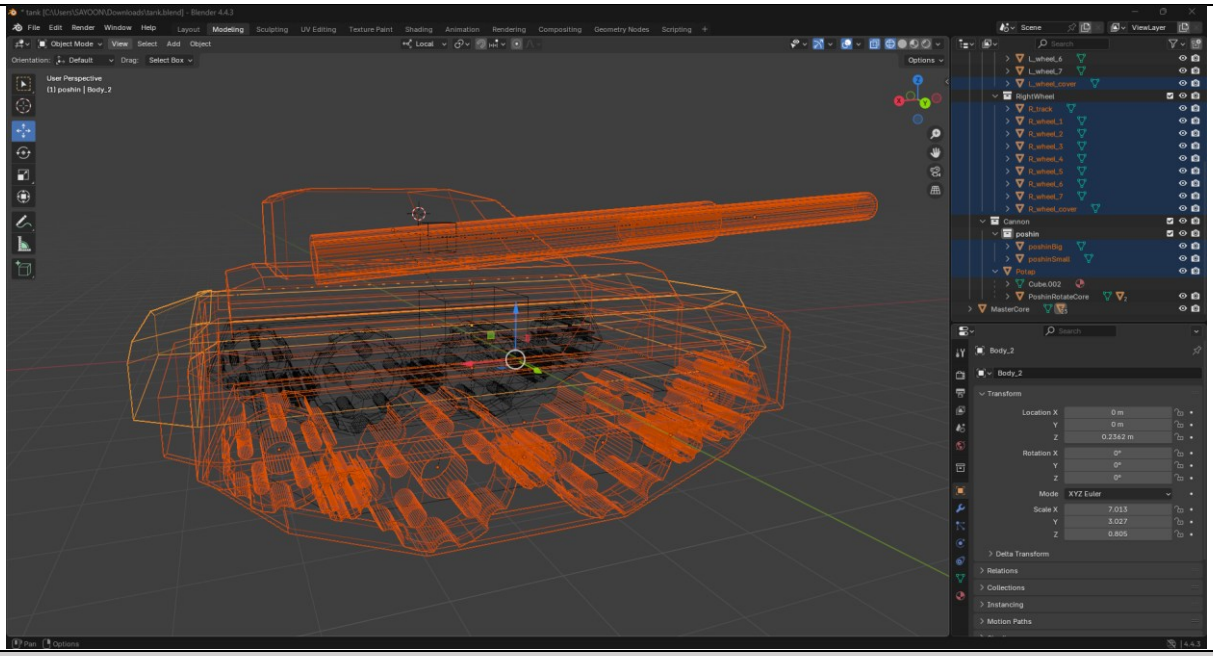
위와 같은 Normal Map 적용



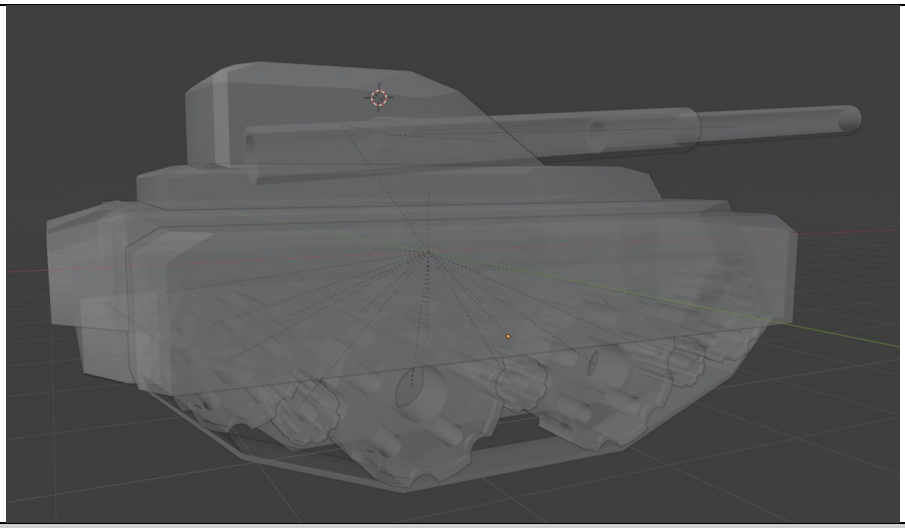



Orthogonal

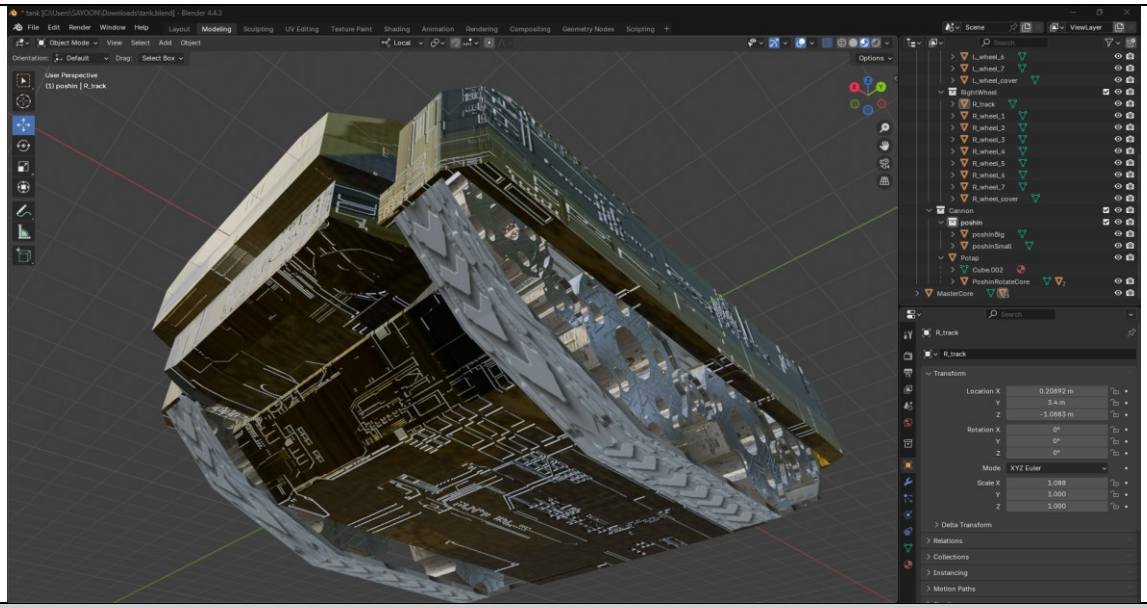
[모델링 완성본 사진]



와이어 프레임으로 바라본 내부 구조



전체 구조 (No Material)



바퀴, 몸체, Rail에다가 텍스쳐와 Normal Map을 적용한 완성본

완성본

[추가 설명]

- Tank의 제자리 회전에 영감을 얻어서 Parent - Child 구조를 극한으로 사용해보고자 기획함.
- Tank의 제자리 회전 시 좌우 바퀴 회전이 반대로 돌아가는 구조를 구현
- 스페이스바를 누를 시 포신 방향으로 raycast가 될 수 있게끔 하기 위해 2개의 파츠로 이루어진 포신의 Origin끼리 선을 그어서 구현하였음
- 전체적인 모습을 관찰할 수 있게 추가 카메라로 Orbit에서 구경하게끔 하였음
- Rapier를 토대로 큰 맵에 탱크를 올려두어 조작을 해보니 게임 엔진에서는 쉽게 구현가능한 물리 구현이 굉장히 부자연스럽게 되며, 고려해야 하는 상황이 많았음을 깨달았음.
- 수업 시간에 배운 Transform과 Scene graph 개념을 알차게 사용해볼 수 있어서 마무리 프로젝트로 만족할 수 있는 결과물이 나왔다고 느끼게 되었음.