```
marbel(other) {
@ for (let v of this.vertices) [fj
let c = other.center;
let r = other.r;
let p = v.copy();
p.sub(c);
let m = p.mag();
let root = sqrt(1 + (r * r) / (mx m));
p-mult( root);
p.add(c);
v.set(p);

function createGroundMesh() {
// adding object: ground
const groundGeo = new THREE.BoxGeometry(2@, 20, 0.5);
const groundMat = new THREE.MeshStandardMaterial({
color: '#031D02',
side: THREE.DoubleSide,
wireframe: false
1);
const groundMesh = new THREE.Mesh(groundGeo, groundMat) ;
groundMesh.name = 'ground';
groundMesh. receiveShadow = true;
scene.add(groundMesh) ;
return groundMesh;
from langchain_openai import ChatOpenAI
```

```
from langchain_community.1lms import Ollama

from langchain_core.output_parsers import StrOutputParser
from langchain_community.document_loaders import PyPDFLoader
from fpdf import FPDF

import os
from PIL import Image
import pytesseracy
def generate_pdf():
pdf = FPDF()
pdf.add_page()
pdf.set_font("helvetica", size = 12)
with open('generated.txt', 'r') as file:
for line in file:
pdf.cell(200, 10, txt = line, In = True, align = 'L')
# Save the PDF with name .pdf
pdf_output = 'generated.pdf'
pdf. output (pdf_output)
def extract_text_from_images(folder_path):

isExist = os.path.exists('generated.txt')

if (isExist == True):
```

```
os. removel(|' generated. txt ')]

# Check if the folder exists

if not os.path.exists(folder_path) :
print(f"The folder {folder_path} does not exist.")
return

# Iterate over all files in the folder
for filename in os.listdir(folder_path):
file_path = os.path.join(folder_path, filename)

# Check if the file is an image
if filename. lower().endswith(('.png', '.jpg', '-jpeg', ?.tiff', '.bmp', '.gif')):
try:
# Open the image file
with Image.open(file_path) as img:
# Use pytesseract to extract text
text = pytesseract.image_to_string(img)
text2 = text.encode('latin-1', 'replace').decode('latin-1')
print(f"Text from {filename}:")
with open('generated.txt', 'a') as file:
file.write(text2)
except Exception as e:
print(f"Failed to process image {filename}: {e}")
pdf = FPDF(()]
```

```
pdf.add_page()

pdf.set_font("helvetica", size = 12)

with open('sample.txt', ?'r') as file:

for line in file:


pdf.cell(200, 10, txt = line, In = True, align = 'L')

# Save the PDF with name .pdf

pdf_output = ?sample.pdf'

pdf. output (pdf_output)
```