```
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

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)


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

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)


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
```

```
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)


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
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)
```