基于文本与图像的 3D 模型生成网页系统

## 一、作品说明

1. 你计划将这个网页面向什么类型的用户？这些类型的用户他们面临什么样的痛点，你设想的用户故事是什么样呢？
2. 你认为这个网页需要哪些功能？这些功能各自的优先级是什么？你计划本次开发哪些功能？
3. 你计划采纳哪家公司提供的 3D 模型生成 API？你对比了哪些公司的 API，你为什么选择用这家公司的 API？
4. AI 应用和传统应用不同，它需要持续进行效果迭代。你会设置什么样的关键指标来作为验证 3D 模型生成效果的评估？你会设计一个什么样的系统来进行评估和改善，请描述评估系统的设计。
5. 有什么方式可以让我们减少对第三方 3D 模型生成 API的调用次数？请描述你认为各种可能的思路，并选择一种进行落地。

本项目旨在构建一个基于网页的 3D 模型生成系统，用户可以通过输入文本描述或上传图片来生成单个 3D 模型。不同于传统建模软件的复杂流程，该系统致力于降低使用门槛，为开发者、设计师以及科研教育人员提供高效便捷的建模工具。由于目标仅限于素材级别的模型生成，例如生成“一匹马”或“一把椅子”，而非完整的场景建模，因此系统能够更好地服务于快速原型设计、教学演示和创意探索等应用场景。

项目的目标用户主要包括独立游戏开发者、AR/VR 开发者、创意设计师以及教育科研人员。这些用户的共同痛点在于 3D 建模门槛高、外包或购买模型成本昂贵、传统建模流程耗时耗力以及素材库模型难以满足个性化需求。为了解决这些问题，本系统允许用户通过简单直观的操作快速获取模型。例如，游戏开发者输入“一匹战马”，系统便能生成对应的 3D 模型；教师上传一张恐龙化石图片，即可获得骨架模型；设计师输入“一把未来风格的椅子”，则能得到一个初步的 3D 椅子原型作为灵感起点。

在功能设计方面，本系统聚焦于两个核心功能：一是基于文本或图片生成 3D 模型，二是通过图像检索减少重复调用第三方 API 的需求。前者的实现方式包括使用 GPT-5 或开源的 InstructPix2Pix 对图片进行 prompt 微调，然后调用 3D 模型生成引擎（如 TripoSR 或 Luma AI）完成建模。GPT-5 在效果上表现优异，但调用成本较高且速度稍慢；InstructPix2Pix 虽然生成质量不及 GPT-5，但作为开源工具易于集成，适合用作成本可控的替代方案。后者功能主要通过感知哈希（Perceptual Hash, pHash）或 CLIP embedding 实现相似图像检索。当用户上传的图片与数据库中已有的图像相似度超过阈值（如 95%），系统将直接返回已有的模型，而无需再次调用第三方 API，从而降低成本并提升响应速度。在开发过程中，团队最开始尝试了phash算法，因为phash算法极其轻量且速度快，十分适合应对我们现在低算力，数据集小的情况。但多次测试后我们发现phash 只能处理低层次的相似度（比如整体轮廓、颜色），无法理解“语义上的相似”，更适合只想快速找出几乎一模一样的图的情景。如果数据库图片差异较大，效果则不如人意。于是我们团队想在有限算力的基础上也许可以试着尝试一些极其轻量的深度学习方法以获取语义信息。经过尝试，我们发现CLIP 或 DINO 等轻量级模型生成的语义 embedding 更适合语义检索，因此最终决定采用 CLIP embedding 存储与检索机制，以保证搜索的准确性与效率。

在第三方 API 的选择上，项目团队对比了 Luma AI、Kaedim3D、OpenAI Shape-E 以及 Stability AI 的 TripoSR 等服务。Luma AI 提供了高质量的文本与图像到 3D 的生成能力，并且已有成熟商用案例，因此被选为首选 API。然而，考虑到其成本问题，TripoSR 作为开源替代方案则用于本地推理，适合高频调用场景。通过将 Luma AI 与 TripoSR 结合，系统能够在保证效果的同时兼顾成本优化。

为了评估生成结果，本系统设计了一套效果评估机制，结合了自动指标与用户反馈两方面。自动评估包括几何合理性（例如面片数量、法向一致性）和视觉相似度（通过 CLIP embedding 计算输入与生成结果的匹配程度）。用户评估则通过网页端的打分系统收集主观反馈。所有评估数据将统一存储在数据库中，用于分析不同 API 的表现，指导后续的 prompt 优化与算法替换。

在系统架构方面，前端基于 React 构建，并使用 Three.js 提供 3D 模型的交互式展示，支持旋转、缩放与平移操作。后端采用 FastAPI 作为主要框架，负责处理输入请求、模型生成、缓存检索与评估数据存储。数据库方面，初期采用 SQLite 以支持快速开发与测试，未来可扩展至 PostgreSQL，以支持更大规模的检索与存储需求。为了便于演示与部署，系统利用 Gradio 搭建交互界面，并通过服务器反向代理方式实现公网访问，甚至可以直接在本地 PC 上作为服务端运行。

在项目开发过程中，团队规划了三个核心演示视频，分别对应三大功能展示。第一个演示展示了上传图片生成 3D 模型的流程，以“水壶”为例，从空白网页上传图片到生成模型，再通过分辨率调整和模型旋转进行展示，并在最后进行用户评分。第二个演示展示了图像预处理与 prompt 微调的流程，以“椅子”为例，用户上传图片并输入提示词“crystal chair”，系统通过 InstructPix2Pix对图像进行优化后生成模型，用户可在界面中预览和评分。第三个演示展示了减少 API 调用次数的优化机制，以“汉堡”为例，用户上传图片后系统检测到已有高相似度模型，直接从数据库返回结果，从而避免重复生成，用户仍可进行浏览与评分。

综上所述，本系统在技术上实现了基于文本与图像的 3D 模型快速生成，并通过 CLIP embedding 检索机制有效减少了对第三方 API 的依赖。在功能设计上，项目平衡了效果与成本，既利用了商用 API 的高质量生成能力，也结合了开源方案的灵活性和低成本优势。同时，通过效果评估与用户反馈，系统能够不断优化生成质量，提升用户体验。本次开发的成果为一个低门槛、可扩展的 3D 模型生成平台，未来可以在游戏开发、教育科研与创意设计等多个领域发挥价值。

## 二、UI展示

1. 初始界面：
   * **顶部标题与说明**  
     页面上方标题为“3D模型生成演示”，下面有简短的说明文字，提示用户上传图片或输入文字，即可生成 3D 模型，并且可以对生成结果进行交互查看和评分。
   * **输入区（左侧）**
     1. **文本输入框**：在页面左上角，用户可以输入提示词（例如“一把椅子”），作为生成 3D 模型的条件。
     2. **图片上传区域**：在文本框下方，有一个 “Drop Image Here” 的上传框，支持拖拽或点击上传本地图片，旁边提供了基本的操作按钮（清空、放大等）。
     3. **扩展参数设置**：在上传区下方，用户可以输入“扩展参数”数值，调节生成时的细节控制。
     4. **3D模型分辨率滑动条**：允许用户在 100 到 256 的范围内调整生成模型的分辨率，以平衡速度和质量。
     5. **开始生成按钮**：大号的橙色按钮，位于输入区最下方，用于启动 3D 模型生成。
   * **生成结果区（中间）**
     1. 在页面中央，有一个展示框用于显示生成的 3D 模型或其缩略图（目前标注为 “OBJ/模型生成结果预览”）。
     2. 下方有一个评分模块，用户可以对生成效果进行选择打分，选项包括“差、一般、好、非常好”。
     3. 评分完成后，可以点击旁边的“提交评分”按钮，将结果反馈存储。
   * **历史模型区（右侧）**
     1. 右侧区域标注为“你可能喜欢的相似模型”，展示了多个历史生成的 3D 模型缩略图。
     2. 每个模型都以小方框的形式呈现，用户可以滚动浏览，快速复用已有的模型结果，从而减少重复调用 API。

社交网站的手机截图

AI 生成的内容可能不正确。

2. 无提示词，生成茶壶

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

图形用户界面, 应用程序, 网站

AI 生成的内容可能不正确。

1. 无提示词，生成汉堡

图形用户界面, 应用程序

AI 生成的内容可能不正确。

1. 有提示词，生成椅子

社交网站的手机截图

AI 生成的内容可能不正确。

社交网络的手机截图

AI 生成的内容可能不正确。

## 三、运行程序说明

本项目名称为 3D\_generator，主要功能是基于文本或图像生成单个 3D 模型，并在网页端进行可视化展示与效果评估。以下为程序运行的环境配置与使用说明。

首先，需要在本地创建运行环境。在命令行中执行以下命令，即可根据 environment.yml 文件创建新的 Conda 环境：

conda env create -f environment.yml

随后需要更新 setuptools 以避免依赖冲突：

pip install --upgrade setuptools

其余依赖可以通过 requirements.txt 文件批量安装：

pip install -r requirements.txt

部分依赖需要手动安装。例如 dlib 库需要先安装 CMake，然后根据 Python 版本选择对应的预编译 .whl 文件进行安装。以 Python 3.10 为例，安装命令如下：

pip install cmake

python -m pip install dlib-19.22.99-cp310-cp310-win\_amd64.whl

此外，项目依赖的 k-diffusion 模块需通过源码安装，可运行以下命令：

pip install git+https://github.com/crowsonkb/k-diffusion.git

完成上述依赖安装后，即可运行演示网页。进入项目根目录，执行以下命令：

python ./demo\_app.py

运行后，系统将在本地启动一个 Web 服务（默认地址为 http://127.0.0.1:7860）。在浏览器中输入该地址即可打开 3D 模型生成演示界面。

在网页界面中，用户可以通过两种方式输入生成条件：其一是在文本框中输入提示词（如 “a crystal chair”）；其二是上传一张本地图片作为参考。用户还可以在界面左下方调整生成参数和 3D 模型分辨率，以平衡生成效果与速度。点击 开始生成 按钮后，系统会调用 3D 模型生成引擎，并在中央区域展示生成结果。用户可以通过鼠标交互对模型进行旋转、缩放和观察。

在生成结果下方，用户可以对模型效果进行主观评价（差、一般、好、非常好），并通过点击 提交评分 按钮反馈给系统。右侧区域则展示了相似模型的历史生成结果，当输入与数据库中已有图片高度相似时，系统会直接返回已有模型，从而减少重复调用第三方 API。

综上所述，该程序的运行流程包括 环境配置 → 依赖安装 → 启动服务 → 网页交互 四个阶段，最终为用户提供了一个基于文本或图像的 3D 模型生成与展示平台。

## 四、分工

模块1：数据库+CLIP embedding检索。负责人：唐依林

模块2：前端UI设计和实现。 负责人：庆刘鑫

模块3：在UI交互界面中集成 TripoSR 三维重建模块。负责人：唐依林

模块4：在图像预处理阶段集成 InstructPix2Pix 模块以实现基于指令的图像微调，并植入UI。负责人：庆刘鑫