3DMGF – PRD

开发团队：江睿达 刘一苇 王子硕

目录

1. 产品背景 2

2. 用户需求 2

2.1. 用户目标 2

2.2. 用户痛点分析 2

2.3. 用户故事 3

3. 功能需求 4

3.1. 网页功能 4

3.2. 功能优先级 5

4. 技术方案与架构设计 6

4.1. 整体架构 6

4.2. 模块设计 7

5. 3D 模型生成 API 选择 8

5.1. API 对比： 8

5.2. 选取标准 8

5.3. 最终选取 9

6. 效果评估指标与系统设计 10

6.1. 评估指标 10

6.2. 评估系统设计 10

7. API 调用优化 11

7.1. 目标与问题界定 11

7.2. 可行方案 11

7.3. 最终方案 - 结果缓存 12

7.4. 其他优化 13

1. 产品背景

3DMGF，全称 Three Dimension Model Generation Forge。

本项目旨在开发一个基于 AI 的 3D 模型生成网页，用户通过输入文本或上传图片即可生成单个 3D 素材模型，以降低建模门槛，提升设计效率。与传统的 3D 建模软件相比，本产品强调“轻量化、低门槛、快速准确生成”的特点，更加贴合教育、游戏原型、设计展示等需求场景。

2. 用户需求

2.1. 用户目标

1) 独立设计师

- 对设计 3D 作品有快速获取的需求，但缺乏专业建模能力
- 希望降低建模门槛，提高创意转化为实际模型的速度

2) 游戏开发者/独立开发团队

- 在游戏开发的早期缺乏低成本素材来源
- 需要快速生成可用模型用于玩法验证或 Demo 演示

3) 教育工作者和培训机构

- 教学过程中需要直观的 3D 模型辅助讲解，但并不追求极高的渲染建模质量
- 缺乏快速生成、低成本的 3D 素材解决方案

4) 产品经理 / 创意工作者

- 需要在方案中展示 3D 概念，但不具备建模技能
- 希望降低对美术团队的依赖，提高沟通与交付效率

2.2. 用户痛点分析

针对 2.1 中提到的用户类型，通常具备以下痛点：

1) **效率**：传统 3D 建模软件（如 Blender）耗时长，无法满足快速原型和即时演示需求
2) **成本**：购买或外包 3D 素材价格高，且沟通周期长，不适合中小型项目或教学应用
3) **技能门槛**：缺乏专业建模能力的用户（如教师、产品经理）几乎没有可用的替代工具，且 3D 建模学习成本较高
4) **资源灵活性**：现有 3D 模型库受限，难以快速找到满足个性化需求的模型，创意实现受阻

2.3. 用户故事

用户角色	完成活动	实现目的
作为一个需要快速复现草图的设计师	我想要通过上传自己的手绘 2D 图像，快速获得用于设计演示的 3D 模型	以便于节省时间成本，更多的去关注如何进行整体的方案设计
作为一个没有专业建模技能的游戏开发者	我想要通过输入一段文字描述，生成一个符合描述的 3D 模型	以便于在游戏原型中测试开发的功能以及玩法的设计
作为一个语言表达能力一般的用户	我想要让自己模糊的文本输入也能精确的生成我想要的结果	以便于即使无法准确描述，也能得到符合预期的 3D 模型
作为一个绘画技术一般的用户	我想要在上传指定的图片之后，能够选择自己希望生成指定风格的 3D 模型	以便于避免风格偏差，减少模型与原始设想的不匹配
作为一个需要跨平台使用模型的游戏开发者	我想要把模型导出为能够在游戏引擎中使用的格式（如 OBJ 或 FBX）	以便于导入到 Unity 等游戏引擎中进行使用
作为一个需要展示 3D 模型的用户	我想要生成的模型具备对应的纹理贴图	以便于更好的展示模型的构造细节和实际效果
作为一个希望验证模型生成质量的用户	我想要在网页中直接旋转、缩放和观察模型	以便于判断生成结果是否符合预期
作为一个需要反复使用模型的用户	我想要拥有个人账号，并且能够把生成过的 3D 模型保存到个人模型库	以便于下次快速调用和复用
作为一个 3D 模型生成网站的新用户	我想要通过公共模型库浏览已有成果	以便于快速理解系统的生成效果，或使用合适的成品
作为一个已经生成了 3D 模型的用户	我想要选择是否公开生成的模型到公共模型库	以便于分享我的作品或保留我的隐私
作为一个要求较高，并且想要长期使用 3D 模型生成的用户	我想要对生成的 3D 模型进行简单的评价或标注，并且反馈到应用的开发者	以便于开发团队不断对程序进行优化，生成更高质量的 3D 模型
作为一个需要随时随地工作的用户	我想要在 PC 端和手机端都具备文字或图片生成 3D 模型的能力	以便于在不同场景下保持工作连贯性

3. 功能需求

3.1. 网页功能

为了满足用户需求，本产品网页具备以下功能（无功能优先级排序）：

1) 文本生成 3D 模型

功能描述：用户通过输入简单的文本描述来快速生成一个基础 3D 模型

目标：提升创意实现的效率，尤其适用于没有专业建模能力的用户

2) 模糊文本转换

功能描述：系统能够自动理解并提炼用户的文本输入，并且生成统一样式、高效且适用于 3D 模型生成 API 的提示词

目标：避免由用户表达能力不足导致的句意理解偏差

3) 图片生成 3D 模型

功能描述：用户可以上传草图或参考图片，系统将根据图片内容生成对应的 3D 模型

目标：通过图片输入快速生成 3D 模型，节省传统建模的时间

4) 模型纹理输出

功能描述：支持自动输出对应的基础纹理贴图，用户可以选择是否下载带纹理的版本

目标：提升模型的可用性和视觉效果，减少用户工作量，进一步降低使用门槛

5) 模型浏览和交互

功能描述：用户在网页中可以通过鼠标或触摸屏操作旋转、缩放和查看生成的 3D 模型

目标：提供直观、互动的模型评估功能，方便用户判断是否需要下载

6) 模型下载和导出

功能描述：用户可以将生成的 3D 模型导出为 3D 文件格式（GLB）进行其他使用

目标：支持用户对生成的模型进行使用

7) 模型保存和管理

功能描述：用户能够保存生成的模型，并在个人账户下进行模型管理和复用

目标：减少重复生成，提升用户的效率

8) 模型公开或隐私

功能描述：用户能够选择是否公开自己生成的模型到公共模型库

目标：提供更多分享渠道，并且允许用户保护自己的作品隐私

9) 模型风格选择（针对图生 3D 用户）

功能描述：使用图片生成 3D 模型的用户能够选择想要的模型风格

目标：确保生成结果与用户的设计意图保持一致

10) 模型质量选择

功能描述：用户可以选择希望生成高质量/普通质量的模型

目标：确保生成的模型文件大小或渲染难度在用户预期之内

11) 效果评估和反馈

功能描述：用户在浏览模型后，可以对生成结果进行评价或简单的反馈

目标：为后续优化提供数据支持，帮助系统生成更符合用户需求的模型

12) 多平台访问

功能描述：用户在 PC 端和手机端使用同一账户生成、浏览和管理模型，支持设备间无缝切换

目标：提升用户的跨平台使用体验

3.2. 功能优先级

基于功能的业务价值和实现难度，定义以下优先级：

P1（MVP 功能)

核心基础功能，必须实现，保证产品可用性和最小可行性：

- 文本生成 3D 模型
- 图片生成 3D 模型
- 模型浏览和交互
- 模型下载和导出

P2（加强功能）

增强型功能，用于提升用户体验与结果质量，会影响用户的满意度和复用度：

- 模型保存和管理
- 模型纹理输出
- 模型风格选择（针对图生 3D 用户）
- 模型质量选择

P3（未来功能-优化方向）

战略型或探索型功能，用于长期优化和扩展产品能力，需要在迭代中实现：

- 效果评估和反馈
- 模糊文本转换
- 多平台访问
- API 调用优化
- 模型公开或隐私
- 用户留言和评论（*未完成*）
- 模型自带物理属性（*未完成*）
- 插件化支持（*未完成*）

4. 技术方案与架构设计

4.1. 整体架构

本产品的整体架构采用前后端分离模式，核心由前端网页交互层、后端服务层、第三方 API 调用层、数据存储层构成。

1) 前端（Web 客户端）

技术选型：React

功能模块：

- 文本输入框与图片上传控件
- 模型浏览与交互（基于 Three.js）
- 模型下载按钮
- 用户账户管理与个人模型库展示
- 用户模型评分选项

2) 后端（业务逻辑层）

技术选型：SpringBoot+LangChain4j

功能模块：

- 调用大模型增强用户输入
- 调用第三方 3D 模型生成 API
- 处理返回的模型文件与纹理数据
- 提供模型存储与下载接口
- 接收用户反馈与评分，写入数据库

3) 第三方 API 调用层

技术选型：Tripo 3D API

- 对接外部 3D 模型生成服务
- 提供 3D 模型对应的纹理生成服务

4) 数据存储层

技术选型：腾讯云对象存储 COS

- 数据库：存储用户账户、生成的模型记录、用户反馈
- 文件存储：保存生成的 3D 模型与纹理文件

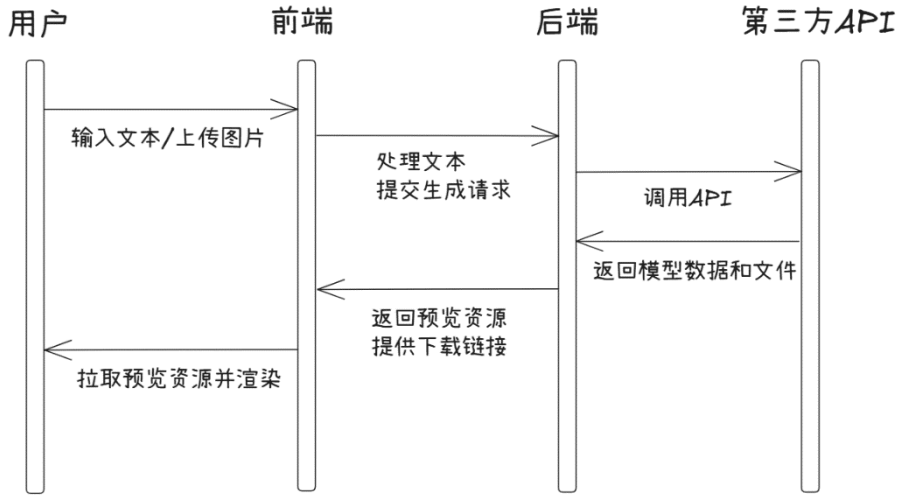
5) 应用部署

技术选型：腾讯云轻量云服务器 & 宝塔 Linux 面板

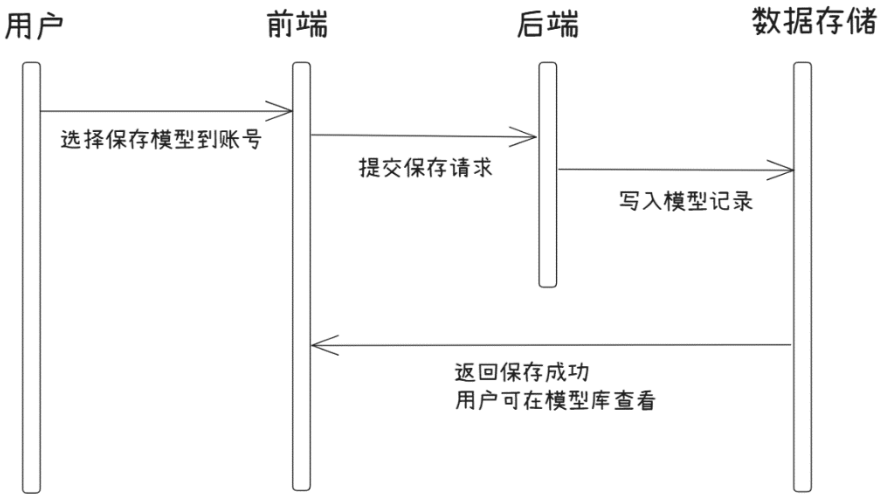
通过以上架构设计，本产品能够实现从用户输入到模型生成与预览的完整闭环，**前端**负责用户交互与模型展示，**后端**承担业务逻辑处理与数据存储，**第三方 API** 提供模型与纹理生成能力，各层之间通过标准化接口解耦，保证了系统的可扩展性与可维护性

4.2. 模块设计

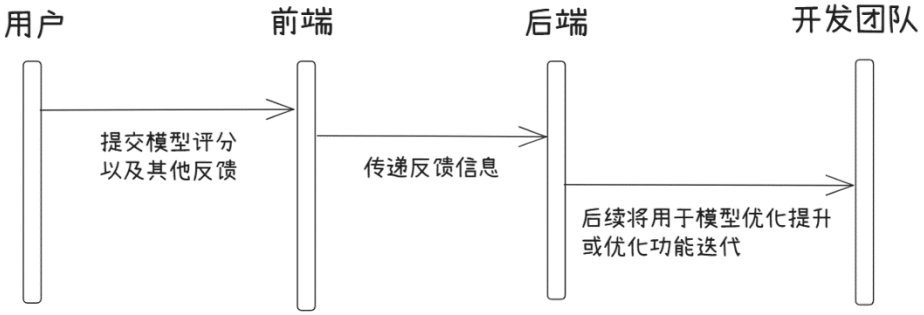
1) 输入与生成模块



2) 模型保存与管理模块



3) 效果评估和反馈模块



5. 3D 模型生成 API 选择

5.1. API 对比:

内容 / API	输入输出能力	付费情况	参考文档
腾讯混元生 3D	可输入：文本或图片，支持多视角； 可输出：OBJ/GLB/STL/FBX	首次开通赠送一次性免费资源包，用尽后可走预付费资源包或后付费；并发可购买叠加包	腾讯混元生 3D 简介 腾讯云
Tripo API	可输入：文本或图片，支持多视角； 可选后处理做格式转换/风格化等	按量预付 API credits；	Platform of Tripo AI
Meshy API	Text-to-3D 两阶段流程：Preview（出网格）-> Refine（上贴图）；提供 Image/Multi-image 等能力	按 Credits 计费；API 仅对 Pro 及以上套餐开放	Introduction - Meshy Docs
Kaedim API	2D 图像-> 3D，Web API + Webhooks 回调下载链接； 可输出：OBJ/FBX/GLB/GLTF	偏企业向商用， 需要注册企业商务账号	Web API | Kaedim

5.2. 选取标准

为了更好的选择适合本项目的 3D 模型生成 API，以下标准将会被考虑为最终的选择标准:

5.2.1 硬性要求

能力：必须支持 Text-to-3D 的云端 REST API，异步任务+查询（提交任务、轮询任务状态、获取下载链接）

可达性：国内网络可直接访问（无需本地常驻进程/自建服务）

输出：产出 GLB 等网页可用格式，贴图与 UV 完整，能直接在 Three.js 预览

文档与集成：官方文档清晰（鉴权、端点、示例、错误码）

合规：允许项目内演示与测试使用，条款清晰，支持内容安全/审查要求

5.2.2 评分维度

生成质量：几何完整度（无明显破洞/非流形）、细节与比例、贴图与 UV 质量

接入效率：端点简单（提交/查询/下载）、SDK/示例齐全、错误定位友好。

成本与配额：按量计费可控，小额充值即可覆盖 >10 次测试；价格/扣费规则清晰

网络可达性与稳定性：国内延迟、超时/失败率、限流策略是否易于应对

输出兼容性：是否直接产出 GLB/OBJ+贴图、是否提供后处理（格式转换/简模）

合规与支持：商用条款、隐私与数据留存说明、支持渠道响应度

5.3. 最终选取

本项目最终选用 **Tripo API** 作为第三方 3D 模型生成 API

5.3.1 选择理由（对照上面的标准）

能力契合：原生支持 **Text-to-3D**，并提供标准的**异步任务模型**（提交生成 -> 任务查询 -> 结果下载），满足我们“不自建、云端直连”的要求

接入效率高：接口设计简单清晰，能够在开发周期（5 天）内完成“后端封装+前端预览”的闭环

成本可控：采用按量预付 API credits 的计费模式，具备免费积分额度，足以覆盖我们的测试与回归；费用边界清晰，便于在文档中说明成本假设

输出友好：默认产出 **GLB**，并提供**后处理端点**（如转 OBJ/FBX、简模/贴图强化），方便网页与后续资产流程

国内可用性：国内网络通常可直连，结合我们的小流量测试规模，可满足 5 天内演示与评审的稳定性需要

5.3.2 已知风险与应对

API 额度不足：预留小额余量；后端开启“输入去重+结果缓存”，降低重复调用，或者创建新账号

生成质量波动：预设稳定的 Prompt 模板（结构化提示），必要时启用后处理（格式转换/简模）以保证预览体验

6. 效果评估指标与系统设计

6.1. 评估指标

为了保证 3D 模型生成效果符合用户预期，同时为后续优化提供数据支撑，本项目定义以下关键指标：

1) 生成速度

指标：单次任务平均耗时（指用户提交 到 生成模型结果可下载的用时）

目标：≤ 90 秒

2) 几何完整度

指标：模型无明显破洞、非流形结构，基础形状可识别

目标：> 90% 模型通过自动几何检查

3) 贴图质量

指标：纹理清晰度、UV 展开合理性

目标：模型具备可用的基础纹理并且渲染到用户页面

4) 文件性能指标

指标：polycount（多边形数）、文件大小

目标：提供“低模/高模”两种选项，文件可快速下载

5) 用户满意度

指标：应用操作评分（1–5）、模型质量（1-5）、符合期望程度（1-5）

目标：平均评分 ≥ 3.5

6) 使用数据

指标：保存次数、再次生成次数、上传次数

目标：作为衡量模型复用度与产品价值的指标

6.2. 评估系统设计

1) 数据收集

自动收集：任务 ID、生成耗时、文件大小、多边形数

用户输入：评分与反馈标签

系统日志：API 调用状态、失败率、重试次数

2) 数据处理

后端定期导出数据，生成 report.csv（含任务 ID、耗时、polycount、用户评分）

数据库存储用户反馈与使用行为，便于后续分析

3) 效果分析

定量分析：生成速度、几何完整度、文件大小等硬指标

定性分析：基于用户反馈和评分的满意度

4) 优化机制

用户反馈将作为 Prompt 工程的输入，通过 LLM API 优化提示词

后端根据 polycount、文件大小自动选择“普通质量/高质量”生成策略

定期汇总分析结果，指导后续迭代版本改进

7. API 调用优化

7.1. 目标与问题界定

在不牺牲核心体验（生成质量与时延）的前提下，降低第三方 3D 生成 API 的调用次数与单次成本。

关键问题:

- 重复/相似输入导致的冗余调用；
- 用户仅为试探效果而触发的高成本生成；
- 失败/波动引起的重试放大；
- 缺乏缓存与配额守护的不可控成本。

7.2. 可行方案

1) 请求去重与输入规范化

将用户输入进行标准化处理（文本去除多余空格、图片进行哈希计算），对于完全相同的请求直接返回已有结果，而不是重复调用。

优点：实现简单，能直接消除一部分重复请求。

风险：对“相似但不完全相同”的输入不一定能识别。

2) 结果缓存（Result Cache）

对已经生成过的结果进行存储，用户再次提交相同请求时直接返回缓存链接。

优点：显著降低调用次数，提升响应速度。

风险：缓存过多可能导致存储成本增加，需要设定清理策略。

3) 低精度预览 + 高清生成

在用户初次请求时，默认返回低多边形、简化贴图的“预览模型”；用户确认后，再调用 API 生成高清版本。

优点：减少用户“试探性操作”的高成本调用。

风险：用户需要多一步确认，流程稍微变长。

4) 调用限流与重试机制

针对高并发情况设置合理的限流阈值，避免系统过载；在调用失败时采用延时重试策略，而不是立即重新生成。

优点：提高整体系统稳定性，避免 API 额度过快耗尽。

风险：用户可能在限流时感到等待时间增加。

5) 相似度检索复用

对文本和图片输入生成特征向量，检索相似度较高的历史结果，并推荐用户复用或基于历史模型进行二次修改。

优点：进一步减少“表达方式不同但目标相似”的重复调用。

风险：需要额外的相似度计算模块，开发复杂度较高。

7.3. 最终方案 - 结果缓存

第三方生成式接口（如 3D 模型生成）调用成本高、时延大且配额受限。

目标：当用户输入完全一致（含模型参数一致）时，优先返回历史已生成结果，避免重复调用第三方 API；同时通过幂等性机制，避免并发/重试导致的重复任务与重复计费。

实现思路：

- 请求签名：对用户完整的请求内容直接计算 sha256Hex 值，作为唯一标识。只有在请求内容一模一样时，标识才会相同。
- 结果缓存：将生成结果与对应的 sha256Hex 标识一同保存到数据库，并建立索引。
- 幂等性保证：相同标识的请求在系统中只会触发一次生成任务；如果结果已经存在，直接返回缓存内容，而不再调用 API。
- 用户感知：

首次请求 -> 调用 API 生成结果，耗时较长。

相同请求再次提交 -> 立即返回已有结果，几乎无延迟。

使用场景

- 用户多次尝试相同的生成任务。
- 不同用户提交相同输入时，可以共享结果（视权限设定）。
- 避免因网络重试、并发请求导致重复生成。

效果

- 命中缓存时：返回速度提升，用户体验显著改善。
- 系统层面：API 调用次数减少，降低运营成本。
- 数据层面：可统计缓存命中率，衡量优化效果。

7.4. 其他优化

系统原本采用串行方式处理任务，当大规模用户同时发起请求时，容易出现排队等待，导致响应延迟和整体性能下降。

目标：通过引入自定义线性池机制，将原本的串行处理方式改为并发处理，提升系统的吞吐能力和响应速度，保障在高并发场景下依然能稳定运行。

实现思路：

- 任务池化：将用户请求统一放入自定义的线性池中进行管理。
- 并发执行：任务在池中被动态分配到可用的执行通道，支持同时处理多个用户请求。
- 资源调度：根据系统资源状况自动调节执行并发度，保证性能与稳定性的平衡。
- 用户感知：

高并发场景 → 不再出现长时间排队等待，响应更加及时；

单个用户请求不会因其他用户的高负载而被明显拖慢。

使用场景：

大规模用户同时提交生成任务；

高峰期请求量集中，避免任务积压；

需要保证系统对多用户的公平处理和稳定响应。

效果：

性能提升：支持并发运行，显著提高整体处理能力；

体验优化：用户等待时间缩短，交互更流畅；

系统层面：资源利用率更高，避免串行执行造成的瓶颈。