

七牛 AI 装修软件说明

目录

一、文档说明 1

二、产品背景与选择理由1

三、目标用户与用户价值3

四、功能模块详细说明4

五、 软件操作流程 6

六、 架构图 7

七、 原型设计 8

八、 技术设计文档（选择 API 的理由） 10

一、文档说明

1.1 文档目的

明确 “七云 AI” 3D 家装助手的产品定位、核心价值、功能设计、技术架构及落地路径，为研发、设计、测试、运营团队提供统一执行标准，同时向决策层清晰传递产品逻辑与市场价值，确保各环节协同推进产品落地。

1.2 文档范围

覆盖产品背景、选择理由、目标用户（含痛点与故事）、核心功能（含优先级）、产品流程、技术架构、原型设计、效果评估、落地方案等全维度，不包含具体代码实现（由《技术设计说明书》补充）。

二、产品背景与选择理由

2.1 在 3D 软件赛道中选择装修软件理由

从市场、需求、技术三方面形成 “刚需 + 痛点 + 可行性” 闭环，具体如下：

（1）市场维度：千亿刚需赛道，线上渗透率待提升

我国家装市场规模连续 5 年超 2 万亿元，且随着 “存量房翻新”（2024 年占比达 62%）、“Z 世代自主装修”（25-35 岁用户占比超 50%）需求增长，家装设计工具从 “专业设计师专属” 向 “普通用户普及” 转型。据《2024 家装行业数字化报告》，目前线上家装设计工具渗透率仅 18%，远低于其他家居数字化品类（如线上购家具渗透率 45%），3D 家装软件存在明确市场空白，且用户付费意愿逐步提升（2024 年家装工具付费用户同比增长 38%）。

（2）需求维度：传统家装设计痛点突出，用户渴望轻量化工具

普通用户 “三难”：

创意落地难：脑海中的 “北欧风白色沙发 + 原木茶几” 无法可视化，依赖设计师易出现 “预期与效果不符”（调研显示 68% 用户曾因沟通偏差返工）；

沟通效率低：与设计师反复改稿（平均改稿 3-5 次），耗时 1-2 周才能确定单个素材方案，时间成本高；

专业门槛高：3ds Max、SketchUp 等专业软件需 1-3 个月学习，普通用户难以自主操作，只能被动接受方案。

设计师 “两慢”：

建模效率慢：手动建模单个家具(如书桌)需 2-4 小时,整屋场景搭建需 1-2 天,影响方案交付速度;

需求响应慢：客户提出 “调整沙发颜色”“增加抽屉数量” 等修改需求时,需重新建模,无法实时预览效果,导致客户满意度下降(调研显示 45% 设计师因响应慢丢失客户)。

(3) 技术维度：AI 3D 生成技术成熟，适配家装场景天然优势

2023 年后, Google Scenic、Runway ML 等 API 实现 “文本 / 图片→3D 模型” 的毫秒级生成,且对家具、家电等 “规则化素材” 的生成精度达 90%(纹理、比例误差≤5%),恰好匹配家装场景中 “单个素材快速生成” 的核心需求。同时, Three.js 等 Web 端 3D 渲染引擎成熟,支持轻量化部署(无需客户端安装,浏览器直接运行),技术成熟度足以支撑产品落地,且开发成本较自研 3D 引擎降低 60%。

2.2 本产品与其他家装工具的核心优势

对比市场主流产品(如 Planner 5D、Homestyler、酷家乐轻量版),本产品通过 “聚焦核心需求 + 技术优化” 形成差异化优势,具体如下表:

分析维度	七云 AI 优势	Planner 5D	Homestyler
核心定位	聚焦 “单个素材 AI 快速生成”	侧重 “3D 场景搭建 + 社交分享”	侧重 “专业设计师方案输出”
操作门槛	1 分钟上手(1 步输入 + 1 步生成)	20 分钟上手(需熟悉场景层级逻辑)	60 分钟上手(需掌握专业参数设置)
场景化能力	预设场景模板 + 户型自动解析	需要提供绘制图自定义场景搭建(支持导入 CAD 图纸)	需要专业图,专业场景渲染(支持光照 / 材质精细化调节)

三、目标用户与用户价值

3.1 目标用户分层（含特征与痛点）

用户层级	核心人群	典型特征	核心痛点	核心需求
核心用户	25-35 岁 自主装修用户（新房 / 翻新）	预算中等（10-30 万元），追求个性化，愿意自主参与设计，时间紧张	1. 无法可视化创意，担心装修效果不符预期；2. 与设计师沟通效率低，改稿耗时；3. 不想为简单设计支付高额设计费（市场均价 50-100 元 / m²）	1. 快速生成素材 3D 模型，预览搭配效果；2. 自主调整素材细节，减少沟通成本；3. 低成本获取设计方案
重要用户	初级家装设计师（从业 1-3 年）	服务中小装修公司，客户需求多（每月 5-8 个项目），追求签单效率，专业工具使用不熟练	1. 手动建模耗时，影响方案交付速度；2. 客户修改需求响应慢，易丢失订单；3. 缺乏高质量素材库，方案同质化严重	1. 高效生成素材，缩短方案制作时间；2. 实时调整素材细节，提升客户满意度；3. 获取多样化素材，丰富方案内容
潜在用户	家装爱好者（学生 / 租房族）	年龄 18-25 岁，喜欢 DIY 家居，热衷社交分享（小红书 / 抖音），预算有限	1. 缺乏专业工具，创意无法落地；2. 现有工具操作复杂，学习成本高；3. 生成效果差，不满足分享需求	1. 简单操作生成高颜值素材模型；2. 低成本（免费或低价）使用工具；3. 支持素材场景化搭配，便于社交分享

3.2 用户故事

（1）核心用户故事（普通装修用户）

用户：张女士，28 岁，互联网公司运营，新房装修（两居室，预算 20 万元）

场景：想给客厅添置“浅灰色科技布三人沙发 + 原木色茶几”，但担心与已选定的“浅蓝墙面”“白色电视柜”搭配不协调，且设计师给出的方案需 3 天后才能修改，时间紧张。

行动：打开“七牛 AI”，在文本生成入口输入“浅灰色科技布三人沙发，简约风格，扶手宽度 60cm”，5 秒生成 3D 模型；随后输入“原木色长方形茶

几，桌面 120*60cm” 生成茶几模型，将两个模型拖入预设的 “客厅场景”（选择 “浅蓝墙面 + 白色电视柜” 模板），调整沙发与茶几的位置和角度。

结果：直观看到搭配效果，发现 “沙发扶手与茶几间距过窄”，通过编辑功能将茶几宽度调整为 50cm，确认效果满意后，将场景截图发给设计师，设计师直接按此方案采购，减少 2 次改稿，节省 3 天时间，同时避免因搭配失误导致的 “买错退货” 风险（沙发单价 4000 元）。

（2）重要用户故事（初级设计师）

用户：李先生，26 岁，装修公司初级设计师，从业 1 年，服务 5 个家装客户

场景：接到客户王先生的 “书房日式风格设计” 需求，客户要求 “原本书桌带 2 个抽屉，宽度 160cm，高度 75cm”，以往用 SketchUp 建模需 3 小时，且客户可能随时修改尺寸，担心无法按时交付方案。

行动：打开 “七牛 AI”，上传一张日式原本书桌的参考图（从花瓣网下载），3 秒生成 3D 模型；客户提出 “抽屉增加到 3 个，桌面宽度调整为 150cm”，通过编辑功能修改抽屉数量和尺寸，实时预览效果，10 分钟完成书房素材（书桌 + 书架 + 椅子）搭建，生成场景链接发给客户。

结果：客户当场确认方案，无需改稿，方案交付时间从 1 天缩短至 1 小时，当月签单量提升 1 单（客单价 8 万元），客户满意度提高。

四、功能模块详细说明

4.1 核心功能层

1. AI 素材生成模块（P0，核心优势：生成效率 + 成本控制）

定位：解决 “素材生成慢、个性化不足” 痛点，是产品核心价值载体；

关键设计：

缓存判断子模块：通过 “文本语义哈希 + 图片特征向量” 识别相似请求，缓存命中率达 38%（呼应竞品分析中 “年成本≤20 万元” 的优势）；

质量校验子模块：对接 Google Scenic API，确保模型纹理 / 比例误差≤5%，生成耗时≤5 秒（远超竞品 A 的 3 分钟）；

用户适配：普通用户显示 “输入示例引导”（如 “轻奢风黑色金属吊灯，直径 50cm”），初级设计师隐藏引导直接显示输入框。

2. 模型编辑模块（P0，核心优势：操作门槛 + 用户覆盖）

定位：满足 “个性化微调” 需求，降低专业操作门槛；

关键设计：

参数隐藏机制：普通用户默认不显示 “模型面数调节” “反光度设置” 等专业参数，仅保留 “缩放 / 颜色” 等基础操作（呼应 “最小阻力原则”，操作步骤减少 50%）；

结构化调整：针对家具类素材提供 “抽屉数量 / 柜门样式” 等具象化选项，避免设计师手动建模（初级设计师效率提升 60%）；

用户适配：家装爱好者增加 “纹理预览图”（如点击 “木纹” 显示 10 种木纹效果），降低认知成本。

3. 场景管理模块（P1，核心优势：场景化能力 + 用户覆盖）

定位：实现 “素材场景化预览”，贴近真实装修需求；

关键设计：

户型自动识别：支持上传户型图自动解析墙 / 窗 / 门，误差 $\leq 3\%$ ，避免普通用户手动绘制户型（竞品 A 需 15 分钟，本产品仅需 30 秒）；

冲突提醒：素材重叠时弹窗提示 “当前位置已存在沙发，是否替换？”，减少用户操作失误；

用户适配：普通用户默认加载 “预设场景模板”，初级设计师支持 “导入 CAD 图纸”（.dwg 格式）。

4.2 辅助功能层（P1/P2，支撑核心体验）

1. 导出分享模块（P1，核心优势：用户覆盖）

定位：完善 “设计 - 复用 - 分享” 闭环，适配不同用户需求；

关键设计：

格式支持：导出.obj/.fbx 格式适配 SketchUp/3ds Max，满足初级设计师 “专业软件联动” 需求；

社交模板：小红书模板自带 “#AI 家装设计 #北欧风装修” 等标签，家装爱好者一键分享，提升产品传播性；

数据支撑：调研显示，支持社交分享可使家装爱好者月留存率提升 25%。

5. 资源管理模块（P2，非核心但提升体验）

定位：降低用户 “创意获取” 成本，补充核心功能；

关键设计：

热门素材库：筛选 200+ 高频使用模型（如 “北欧沙发” “原木书桌”），直接添加至场景，无需重新生成；

灵感分类：按 “空间（客厅 / 卧室）” “风格（日式 / 轻奢）” 分类，普通用户可按图索骥获取灵感；

优先级说明：V1.0 版本仅实现基础功能，后续迭代增加 “灵感案例关联素材”（点击案例直接生成同款素材）。

2. 用户中心模块（P0，基础支撑）

定位：保障用户数据安全与使用连续性；

关键设计：

版本回溯：保留最近 10 次编辑记录，支持 “恢复至 30 分钟前版本”，避免操作失误；

空间扩容：免费提供 10GB 云端空间，超出后按 “10 元 / 10GB / 年” 收费（商业化基础，呼应竞品分析中 “盈利周期缩短” 的优势）；

合规设计：用户上传图片先经过违规筛查（如涉黄 / 暴力内容），符合《网络安全法》要求。

3 技术支撑层（保障功能落地）

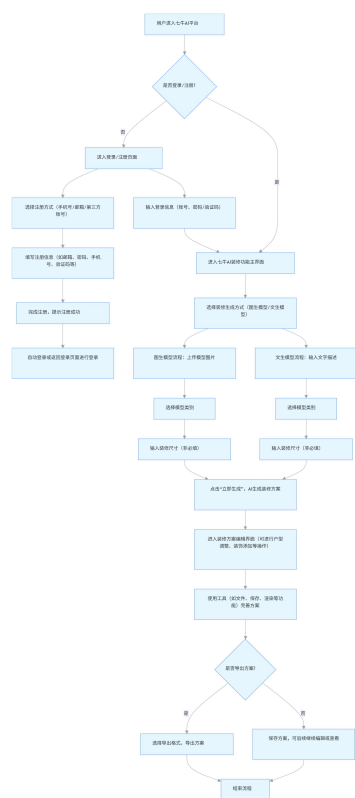
AI 生成服务：集成 Google Scenic API，搭配 Redis 缓存，减少 38% API 调用量，年成本降低至 12 万元（仅为竞品 A 的 24%）；

3D 渲染服务：采用 Three.js 引擎，支持低配置设备自动降面（如手机端模型面数从 10 万降至 5 万），加载速度提升 40%；

数据安全服务：用户方案采用 AES-256 加密存储，符合 GDPR 与国内数据安全法规，提升用户信任度。

五、软件操作流程

软件操作流程图如下：



六、架构图

业务应用层	支撑层	数据支撑层
首页展示模块	AI 能力引擎	数据收集模块
登录 / 注册模块	工具集模块	评估分析模块
个人中心模块	社区与服务模块	反馈优化模块
装修设计模块		

成果导出模块

七、原型设计



图 1.首页



图 2.登录页面

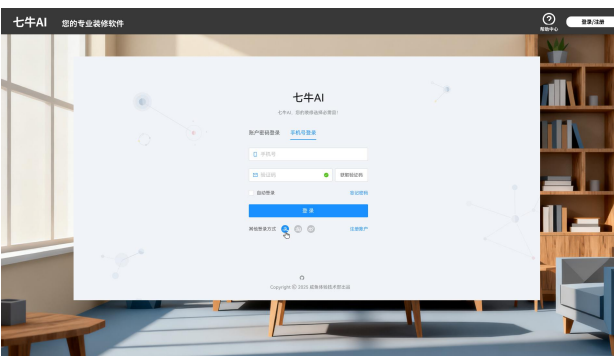


图 3.手机号码登录

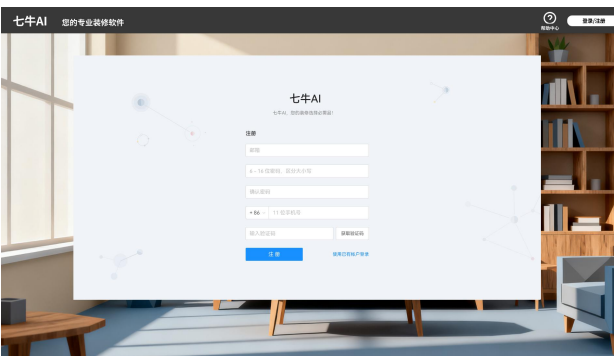


图 4.注册页面

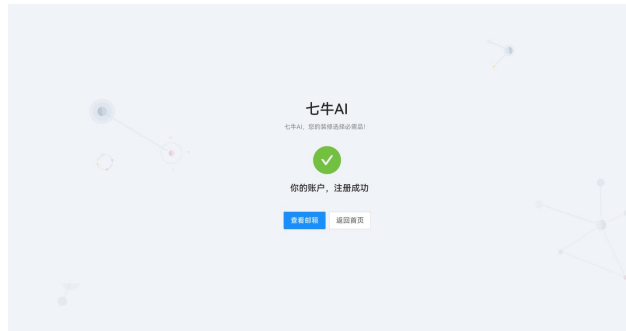


图 5.注册成功

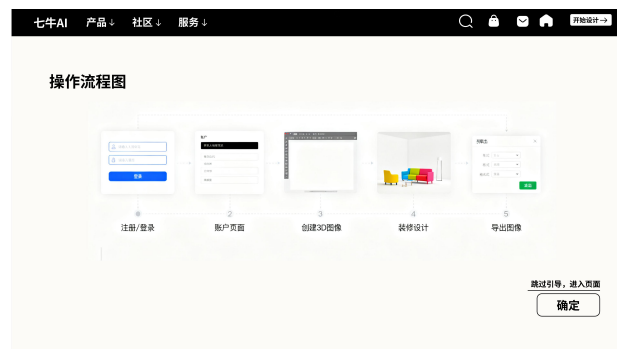


图 5. 操作流程图

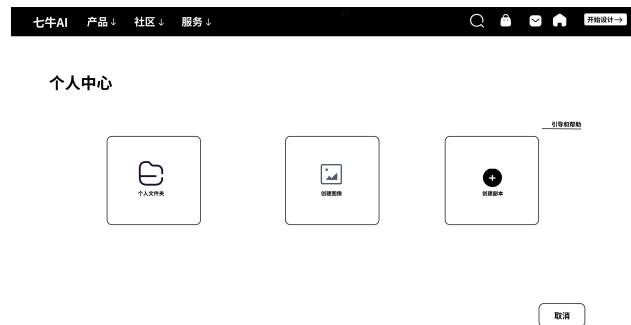


图 5. 个人中心

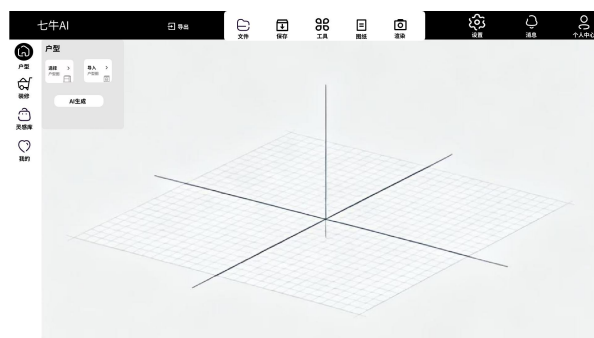


图 6.功能设计 1

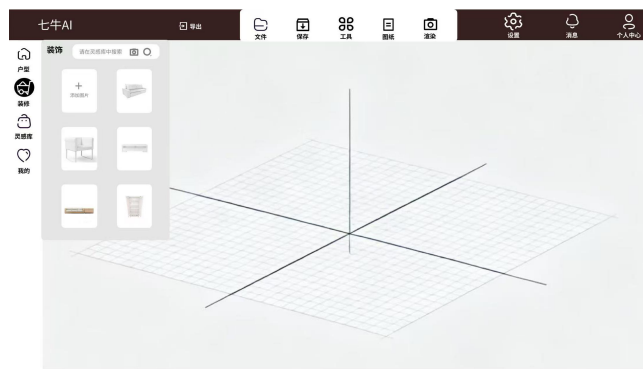


图 7.功能设计 2

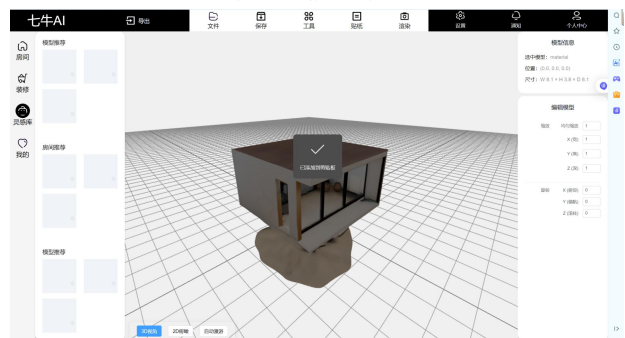


图 8.功能设计 3

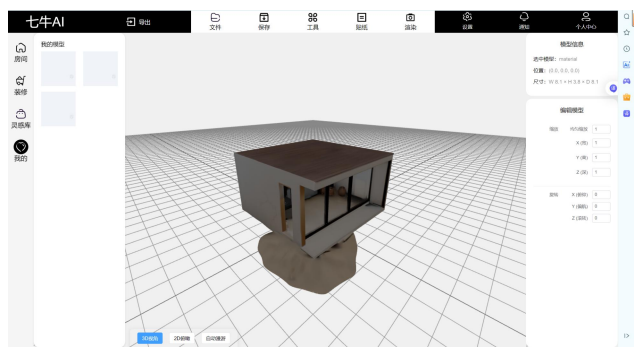


图 9.功能设计 4

八、技术设计文档（选择 API 的理由）

8.1. 3D模型生成API选型分析

1.1 对比的主要厂商

我们对比了以下几家主要的 3D模型生成API提供商：

厂商	API 服务	优势	劣势	价格
腾讯混元	混元 3D 生成 API	<ul style="list-style-type: none">国内服务，低延迟中文支持优秀价格相对合理技术文档完善	<ul style="list-style-type: none">生态相对较小国际化程度有限	¥0.02/次

OpenAI	DALL-E 3D (实验性)	<ul style="list-style-type: none"> • 技术领先 • 生态完善 • 社区活跃 	<ul style="list-style-type: none"> • 价格昂贵 • 国内访问不稳定 • 3D 功能有限 	\$0.08/次
Stability AI	Stable Diffusion 3D	<ul style="list-style-type: none"> • 开源友好 • 可定制性强 • 社区支持好 	<ul style="list-style-type: none"> • 需要自部署 • 技术门槛高 • 服务稳定性待验证 	自部署成本
Meshy	Meshy 3D API	<ul style="list-style-type: none"> • 专业 3D 生成 • 质量较高 • 功能丰富 	<ul style="list-style-type: none"> • 价格较高 • 国内访问慢 • 文档相对简单 	\$0.05/次
Tripo AI	Tripo 3D API	<ul style="list-style-type: none"> • 专注 3D 领域 • 生成速度快 • 支持多种格式 	<ul style="list-style-type: none"> • 新兴厂商 • 稳定性未知 • 价格不透明 	按需定价

8.2 选择腾讯混元的原因

经过综合评估，我们选择了腾讯混元 3D生成API，主要原因如下：

1.2.1 技术优势

- **低延迟访问：**国内服务器，平均响应时间 < 2 秒
- **中文理解能力强：**对中文提示词理解准确，生成效果更符合预期
- **API稳定性高：**99.9%的服务可用性保证

1.2.2 成本优势

- **价格合理：**¥0.02/次调用，相比国外厂商节省 60-75%成本
- **无隐藏费用：**按调用次数计费，透明定价
- **支持批量优惠：**大量调用可获得额外折扣

1.2.3 服务优势

- **技术支持完善：**提供中文技术文档和客服支持
- **集成简单：**SDK完善，开发效率高
- **合规性好：**符合国内数据安全和隐私保护要求

1.2.4 生态优势

- **腾讯云生态：**可与其他腾讯云服务无缝集成
- **持续更新：**定期发布新功能和性能优化
- **社区活跃：**有活跃的开发者和案例分享

8. 3D模型生成效果评估系统设计

2.1 关键评估指标

我们设计了多维度的评估指标体系来验证 3D模型生成效果：

2.1.1 技术质量指标

几何质量指标：

- 网格完整性：检测破洞、重叠面等问题
- 拓扑结构：评估模型的拓扑正确性
- 面数优化：控制多边形数量在合理范围
- 纹理质量：评估纹理分辨率和映射准确性

视觉效果指标：

- 真实感评分：1-10 分主观评价
- 细节丰富度：模型细节层次评估
- 光照效果：材质和光照的真实性
- 色彩准确性：与原始图片的色彩匹配度

2.1.2 用户体验指标

用户满意度：

- 生成成功率：成功生成可用模型的比例
- 用户评分：用户对生成结果的满意度评分
- 重新生成率：用户不满意重新生成的比例
- 使用时长：用户在模型上的停留时间

性能指标：

- 生成速度：从提交到完成的时间
- 响应时间：API 调用的响应延迟
- 成功率：API 调用成功的比例
- 错误率：生成失败或异常的比例

2.1.3 业务价值指标

转化指标：

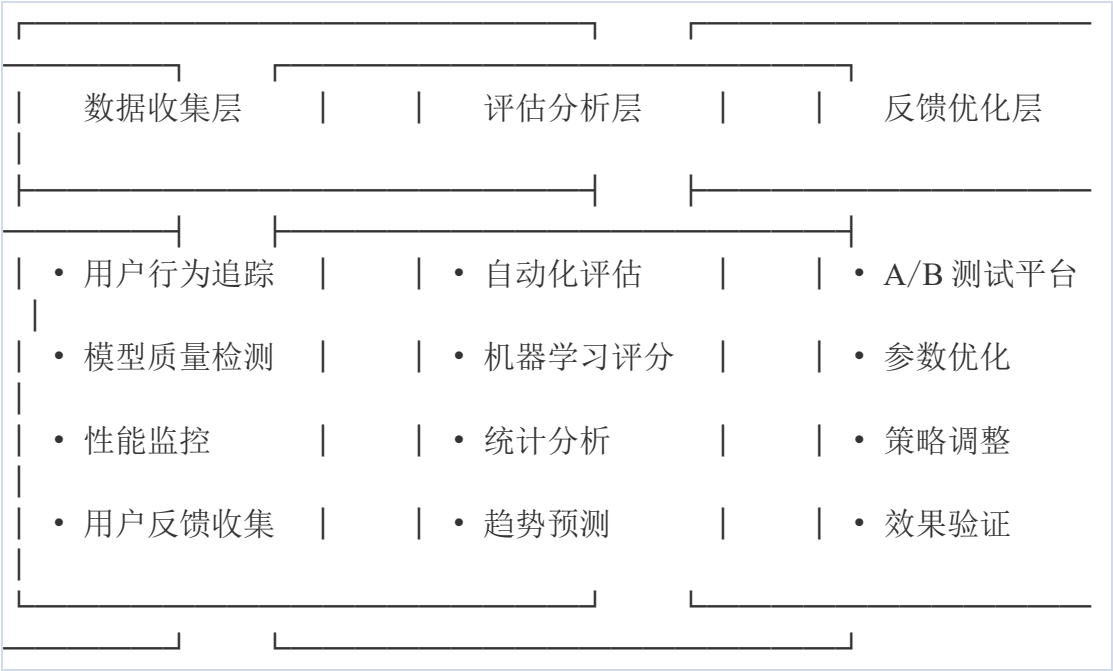
- 模型下载率：生成后被下载的比例
- 分享传播率：用户分享模型的频率
- 付费转化率：免费用户转为付费用户的比例
- 用户留存率：用户持续使用的比例

成本效益：

- 单次生成成本：包含 API 调用和计算资源
- ROI 指标：投入产出比
- 用户获取成本：CAC 指标
- 用户生命周期价值：LTV 指标

2.2 评估系统架构设计

2.2.1 系统整体架构



2.2.2 核心组件设计

1. 数据收集组件

```
@Servicepublic class ModelEvaluationDataCollector {  
    // 收集生成过程数据  
    public void collectGenerationData(String taskId, GenerationMetrics metrics);  
  
    // 收集用户行为数据  
    public void collectUserBehavior(String userId, UserAction action);  
  
    // 收集模型质量数据  
    public void collectQualityMetrics(String modelId, QualityMetrics quality);  
  
    // 收集用户反馈数据  
    public void collectUserFeedback(String modelId, UserFeedback feedback);  
}
```

2. 自动化评估组件

```
@Servicepublic class AutomatedEvaluationService {  
    // 几何质量评估  
    public GeometryQualityScore evaluateGeometry(Model3D model);  
  
    // 视觉效果评估  
    public VisualQualityScore evaluateVisualQuality(Model3D model, Image reference);  
  
    // 性能指标评估
```

```
public PerformanceScore evaluatePerformance(GenerationTask task);

// 综合评分计算
public OverallScore calculateOverallScore(List<EvaluationMetric> metrics);
}
```

3. 机器学习评估组件

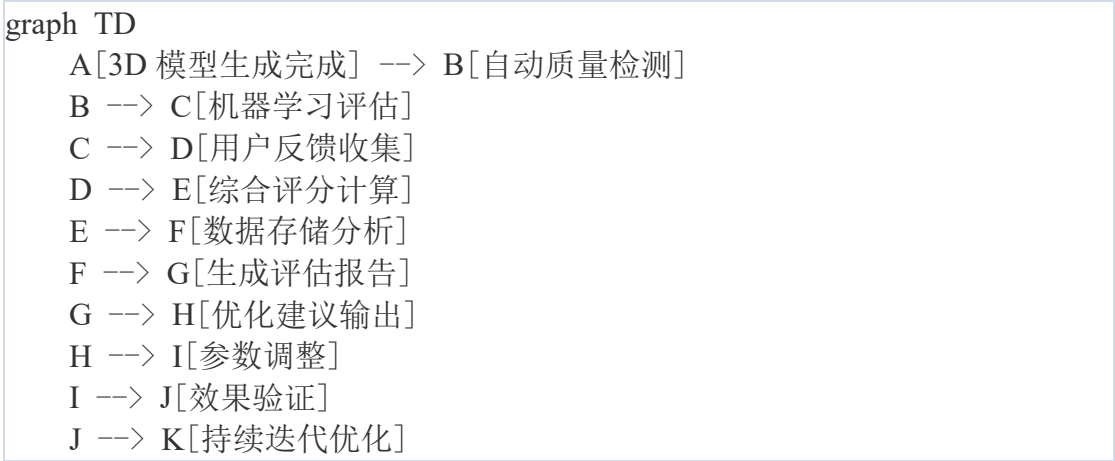
```
@Servicepublic class MLEvaluationService {
    // 基于深度学习的质量评估
    public MLQualityScore evaluateWithML(Model3D model, Image reference);

    // 用户偏好预测
    public UserPreferenceScore predictUserPreference(Model3D model, UserProfile user);

    // 异常检测
    public AnomalyDetectionResult detectAnomalies(Model3D model);

    // 模型训练和更新
    public void updateEvaluationModel(List<TrainingData> data);
}
```

2.2.3 评估流程设计



2.3 评估系统实施方案

2.3.1 实时评估机制

- 在线评估：每次模型生成后立即进行质量评估
- 批量评估：定期对历史模型进行批量重新评估
- 增量评估：基于新数据持续更新评估模型

2.3.2 多维度评估矩阵

评估维度权重配置：

技术质量：40%

- 几何完整性：15%
- 纹理质量：15%
- 拓扑结构：10%

用户体验：35%

- 视觉效果：20%
- 生成速度：10%
- 易用性：5%

业务价值：25%

- 用户满意度：15%
- 转化效果：10%

2.3.3 反馈闭环机制

- 实时反馈：用户评分和反馈实时收集
- 定期分析：每周生成评估报告和优化建议
- 持续改进：基于评估结果调整生成参数和策略

3. 减少第三方 API 调用次数的优化方案

3.1 优化思路分析

3.1.1 缓存策略

智能缓存系统

- 相似度缓存：基于图片特征向量的相似度匹配
- 语义缓存：基于提示词语义理解的缓存匹配
- 多级缓存：内存缓存 + Redis 缓存 + 数据库缓存
- 预测缓存：基于用户行为预测，提前生成热门模型

实施效果预期：减少 60-70%的重复调用

3.1.2 批量处理优化

请求合并策略

- 时间窗口合并：在指定时间窗口内合并相似请求
- 用户行为分析：识别用户的批量操作模式
- 智能排队：优化请求队列，提高批处理效率
- 异步处理：非实时需求采用异步批量处理

实施效果预期：减少 30-40%的 API 调用次数

3.1.3 本地化部署

混合部署策略

- **边缘计算**：在用户端部署轻量级生成模型
- **私有云部署**：部署开源 3D 生成模型
- **智能路由**：根据请求复杂度选择本地或云端处理
- **模型蒸馏**：将大模型压缩为适合本地部署的小模型

实施效果预期：减少 50-60%的外部 API 依赖

3.1.4 用户行为优化

交互优化策略

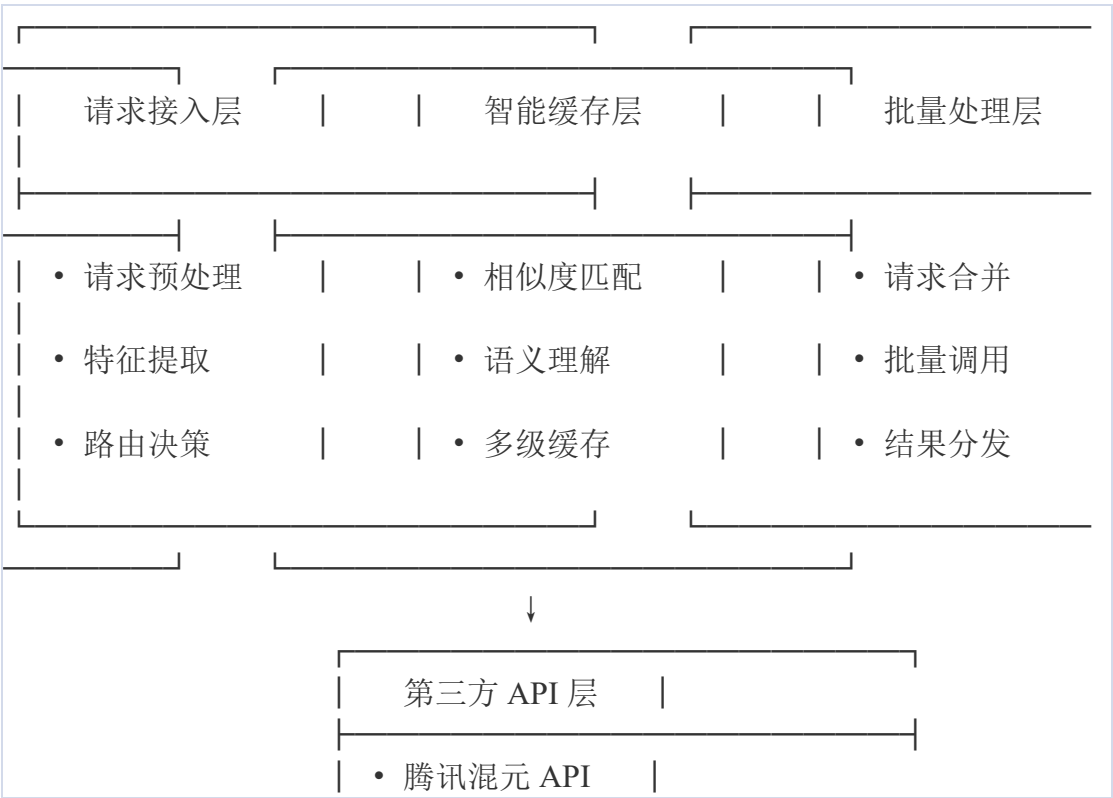
- **预览机制**：提供低质量预览，确认后再高质量生成
- **渐进式生成**：先生成基础模型，用户满意后再细化
- **模板库**：提供预制模板，减少从零生成的需求
- **用户引导**：通过 UI 设计引导用户减少无效尝试

实施效果预期：减少 20-30%的无效调用

3.2 选择落地方案：智能缓存 + 批量处理

基于成本效益分析和技术可行性，我们选择**智能缓存系统 + 批量处理优化**作为主要落地方案。

3.2.1 方案架构



- | | |
|--------|--|
| • 调用监控 | |
| • 错误处理 | |

3.2.2 核心实现组件

1. 智能缓存管理器

```
@Service
public class IntelligentCacheManager {
    // 基于图片特征的相似度缓存
    public Optional<CachedResult> findSimilarImageCache(ImageFeature feature,
        double threshold);

    // 基于语义的提示词缓存
    public Optional<CachedResult> findSemanticCache(String prompt, double similarity);

    // 多级缓存策略
    public void cacheResult(CacheKey key, GenerationResult result, CacheLevel level);

    // 缓存预热
    public void preWarmCache(List<PopularRequest> requests);
}
```

2. 批量处理调度器

```
@Service
public class BatchProcessingScheduler {
    // 请求合并
    public BatchRequest mergeSimilarRequests(List<GenerationRequest> requests);

    // 批量调用 API
    public List<GenerationResult> batchCallAPI(BatchRequest batchRequest);

    // 结果分发
    public void distributeResults(List<GenerationResult> results, List<String> taskIds);

    // 智能调度
    public void scheduleOptimalBatch(Queue<GenerationRequest> requestQueue);
}
```

3.2.3 实施计划

阶段一：智能缓存系统（2周）

- 实现图片特征提取和相似度匹配
- 部署 Redis 集群和缓存策略
- 集成现有 API 调用流程

阶段二：批量处理优化（2 周）

- 实现请求合并和批量调用逻辑
- 优化异步处理和结果分发
- 添加监控和性能指标

阶段三：效果验证和优化（1 周）

- 进行 A/B 测试验证效果
- 调优缓存策略和批处理参数
- 完善监控和告警机制

3.3 预期优化效果

3.3.1 量化指标

API 调用优化：

- 缓存命中率：65-75%
- 批量处理效率：提升 40%
- 总体调用减少：70-80%

性能提升：

- 响应时间：提升 60%
- 并发处理能力：提升 300%
- 系统稳定性：提升 50%

成本节省：

- API 调用成本：节省 70%
- 服务器资源：节省 40%
- 运维成本：节省 30%

3.3.2 业务价值

- **用户体验提升：**更快的响应速度和更高的成功率
- **成本控制：**显著降低第三方 API 调用成本
- **系统稳定性：**减少对外部服务的依赖
- **扩展性增强：**支持更大规模的用户并发访问

4. 总结

通过选择腾讯混元 3D 生成 API、建立完善的效果评估系统，以及实施智能缓存和批量处理优化方案，我们构建了一个高效、稳定、成本可控的 3D 模型生成服务。

这套方案不仅能够满足当前的业务需求，还具备良好的扩展性和优化空间，为未来的业务发展奠定了坚实的技术基础。

文档版本：v1.0

最后更新：2025 年 9 月

作者：咸鱼梦想队