

米哈游·智能体拟真交互设计 + 活动质量体系建设 — STAR法则深度拆解

岗位：米哈游 在研项目AI功能及玩法测试组·产品分析与策略优化实习生

时间：2025.06 - 2025.12

项目：旗舰开放世界项目 AI NPC交互系统 + 活动质量管理

Part A：智能体拟真交互设计

一、项目全景

Why — 为什么要做这件事？

旗舰开放世界项目中有一个核心社交场景——场景内的“老板娘”NPC。她是玩家在游戏世界中的情感锚点：玩家探索归来会去她那里歇脚、聊天、获取信息。

但传统方案下，老板娘NPC的对话体验有三个致命问题：

问题	表现	玩家感受
千篇一律	对话走固定对话树，所有玩家听到的是同一套台词	"这个NPC就是个任务发放机"
没有记忆	玩家昨天告诉她"我要去打Boss"，今天她完全不记得	"她根本不认识我"
情感断裂	玩家兴奋地分享战斗经历，她用平淡语气回复"好的，我知道了"	"在和一台机器说话"

核心矛盾：开放世界项目投入大量资源构建了沉浸式的视觉和玩法体验，但NPC交互的拉胯直接打破沉浸感——玩家在一个精心构建的世界里，却和一个“失忆的复读机”对话。

How — 怎么做的？

我设计了以老板娘NPC为核心的多智能体协作交互方案，分三层构建：

第一层：端到端情感语音——让NPC“有感情地说话”

问题：传统TTS生成的语音平淡机械，缺乏情感起伏。

方案：引入端到端情感语音能力，让老板娘NPC的语音输出具备真实情绪表达：

- 玩家兴奋地讲述战斗经历 → 老板娘用惊喜的语气回应：“真的吗！太厉害了！”
- 玩家疲惫地说“今天好累” → 老板娘用温柔的语气说：“辛苦了，坐下喝杯茶吧”
- 玩家沉默不语只是坐下 → 老板娘用关切的语气主动问：“怎么了，看你心情不太好？”

关键设计点：不是简单的“识别情绪关键词→切换语音模板”，而是让模型直接从玩家的语音中感知情感状态，并在生成回复时保持情绪对称——这和我在腾讯推的GVoice E2E方向在技术理念上一脉相承。

第二层：RAG记忆机制——让NPC"记住你"

问题：传统NPC每次对话重新开始，没有上下文延续。

方案：设计双层记忆架构：



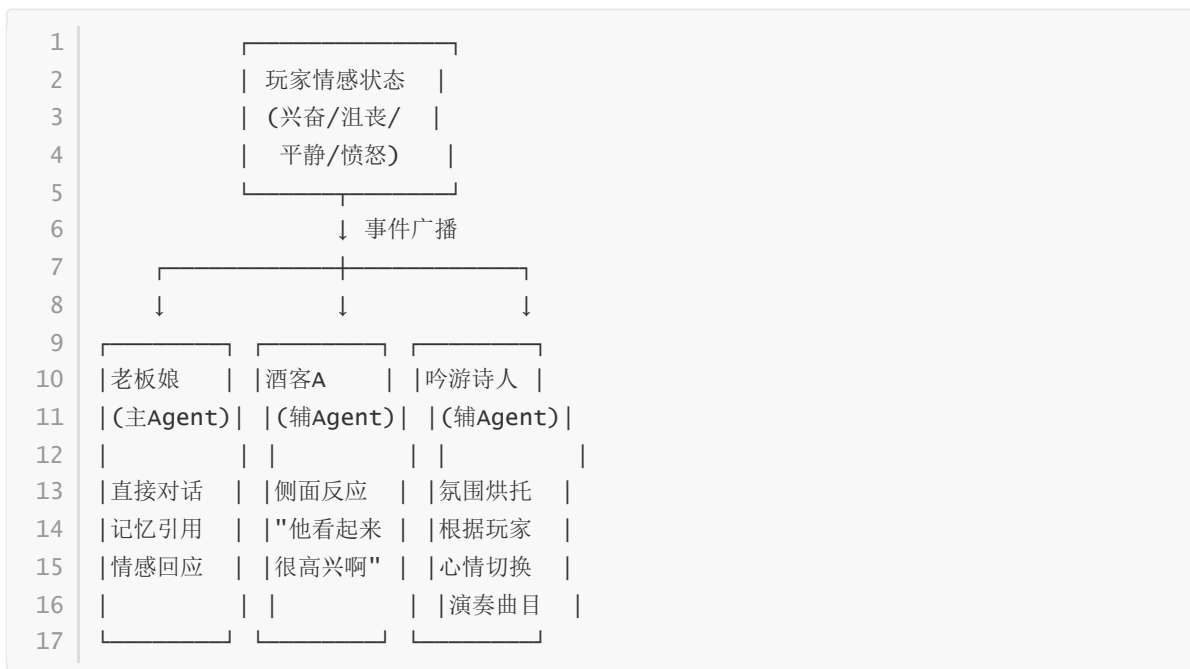
实际效果：

- 玩家3天前说"我要去打冰霜巨龙" → 今天回来，老板娘主动问"冰霜巨龙打赢了吗？"
- 玩家上周送了花 → 这周老板娘说"上次你送的花还摆在柜台上呢"
- 玩家连续一周没来 → 再次登录时老板娘说"好久不见你了，我还以为你不回来了"

第三层：环境事件联动——让周围NPC"有反应"

问题：即使老板娘NPC变"聪明"了，周围的NPC还是木头人，打破整体沉浸感。

方案：以老板娘为核心Agent，周围NPC为辅助Agent，通过事件总线共享玩家状态：



设计原则：

- 每个辅助Agent有自己的**性格设定**——酒客粗犷直接、吟游诗人文艺含蓄，对同一个玩家状态的反应方式不同
- 辅助Agent的反应是**低频、非侵入式**的——不是每个NPC都跑过来和你搭话，而是偶尔的一句侧面评论或行为变化
- 通过这种**"众人皆有感知"**的设计，让玩家觉得自己真的身处一个"活"的世界

What — 做出了什么结果？

维度	成果
交互体验	老板娘NPC具备记忆连贯性（能引用历史互动）+ 情感反馈（情绪对称回应）+ 主动关怀（基于长期记忆主动发起对话）
多Agent协同	核心NPC + 周围辅助NPC的事件联动，营造"活"的社交场景
落地验证	成功上线一测，实现从设计到体验验证的完整闭环

二、面试官深挖问题

Q1: RAG记忆机制的长期记忆怎么做关键事件提取？

答：不是把每句话都存，而是在对话结束后用一个**摘要提取模块**筛选"值得记住的事"。判断标准：

1. **玩家明确表达了意图或计划**——"我要去打Boss""我明天再来"
2. **发生了情感强度较高的互动**——玩家送礼物、玩家表达不满
3. **时间间隔异常**——玩家很久没来或突然频繁来

提取后用embedding向量化存储，检索时按语义相似度匹配当前对话上下文，把最相关的历史事件注入prompt。

Q2: 多Agent的性能开销怎么控制？

答：两个关键优化：

1. **情性触发**：辅助Agent不是实时推理，而是当玩家情感状态发生**显著变化**时才触发反应（比如从"平静"变为"兴奋"）
2. **预设模板+动态填充**：辅助Agent的反应不是每次都走完整LLM推理，而是预设一组反应模板（按性格×情绪类型组合），动态填充具体内容——大幅降低推理成本

Q3: 上线一测的效果怎么样？

答：一测的核心目的是**验证方案可行性和用户感受**，不是追求完美效果。主要观察：

1. **玩家与老板娘NPC的平均对话轮次**——对比传统对话树方案是否显著增加（说明玩家愿意多聊）
2. **玩家主动回访频率**——是否有更多玩家"没事就去找老板娘聊天"
3. **定性反馈**——测试玩家的体验问卷中关于NPC交互的评分和文字反馈

一测验了方案的**基本可行性**——玩家确实能感知到NPC"记得自己"和"有情感回应", 但也暴露了一些问题, 比如长期记忆偶尔检索到不相关的历史事件导致NPC"说错话", 这是后续迭代的重点。

Q4: 和腾讯做的AI语音助手有什么本质区别?

答:

维度	腾讯AI语音助手	米哈游NPC智能体
定位	功能型助手 ——帮玩家执行指令 (买装备、指挥BOT)	情感型伙伴 ——成为玩家的情感锚点
核心指标	指令执行成功率、响应时延	情感沉浸感、对话黏性、回访频率
技术侧重	ASR精度、指令解析、TTS自然度	记忆连贯性、情感感知、多Agent协同
商业逻辑	to B技术组件售卖	to C用户体验驱动留存

Part B: 活动质量体系建设

一、项目全景

Why — 为什么要做这件事?

旗舰开放世界项目的活动系统极其复杂——每个版本都包含多个限时活动、赛季内容、节日主题等, 而且版本迭代频率高 (约每2-3周一个小版本)。测试团队面临的核心压力:

痛点	具体表现
测试覆盖不足	活动玩法多样 (解谜/战斗/收集/社交), 每种玩法的边界条件和异常分支都不一样, 手动编写测试用例容易遗漏
缺陷追踪断裂	测试发现的Bug从提交到修复到验证, 中间可能跨多个版本周期, 缺少全链路的状态追踪
被动响应	经常是活动上线后玩家反馈Bug才发现问题, 而不是在Beta阶段主动拦截

一句话总结: 高频迭代 + 复杂活动系统 + 有限测试资源 = 必须从"人肉测"升级为"体系化测"。

How — 怎么做的?

我作为测试服导盘的角色, 统筹Beta阶段的质量管理。核心做了三件事:

第一：重构标准化测试用例体系

之前的问题：测试用例散落在各个测试同事的本地文档里，没有统一的模板和分类标准。新版本来了，大家各写各的，覆盖面靠经验和手感。

我的做法：

动作	具体内容
统一模板	设计活动测试用例标准模板：前置条件 → 操作步骤 → 预期结果 → 实际结果 → 关联配置项
分类标准	按活动类型（解谜/战斗/收集/社交）× 测试维度（功能/边界/性能/兼容性）建立矩阵
复用机制	同类型活动的测试用例可以继承和复用——新版本的"收集类活动"可以直接基于上版本的用例模板扩展
优先级标记	P0（必测，阻塞上线）/ P1（应测，影响体验）/ P2（选测，优化项），在测试资源紧张时按优先级裁剪

第二：建立缺陷全链路追踪流程

1	[测试发现Bug]
2	↓
3	录入缺陷管理系统（标准化字段：复现步骤/严重度/影响范围/关联活动/截图录屏）
4	↓
5	[状态：待修复] → 分配给对应开发
6	↓
7	[状态：修复中] → 跟踪排期，超时预警
8	↓
9	[状态：待验证] → 回归测试验证
10	↓
11	[状态：已关闭] 或 [状态：重开]（验证不通过）
12	↓
13	[版本复盘] → 按活动类型统计缺陷分布，识别高风险模块

关键改进：

- 每个Bug必须关联到具体活动和配置项——方便后续同类活动排查
- 超时预警机制——P0 Bug超过24h未修复自动升级
- 版本维度的缺陷统计——每个版本闭环时汇总"发现/修复/遗留"数据

第三：统筹Beta阶段质量管理

Beta阶段我的角色：作为"导盘"角色，我负责统筹测试计划、协调测试资源、跟进关键缺陷、并在Beta结束时给出质量评估结论。

阶段	我的工作
Beta前	根据版本内容制定测试计划——哪些活动需要测、用多少测试资源、P0用例有哪些
Beta中	每日同步测试进度和新发现缺陷，P0 Bug即时升级，协调开发修复排期

阶段	我的工作
Beta后	汇总缺陷数据，评估版本质量是否达到上线标准——P0全闭环、P1闭环率>90%、无遗留阻塞项

What — 做出了什么结果？

维度	成果
测试效率	标准化用例模板 + 复用机制，新版本测试用例编写效率提升
缺陷管控	全链路追踪流程覆盖Bug从发现到闭环的完整生命周期
质量前移	从"上线后被动修Bug"转向"Beta阶段主动拦截"，减少线上事故
可持续性	缺陷数据按活动类型沉淀，后续同类活动可提前识别高风险模块

二、面试官深挖问题

Q5: 活动质量体系和你做AI产品有什么关系？

答：表面看一个是测试质量管理，一个是AI产品设计，但底层能力是相通的：

- 结构化思维**——把"测一个复杂活动"拆解为"按类型×维度的矩阵"，和我后来把"甲方的模糊需求"拆解为"5个产品机会"是同一种思维
- 全链路闭环**——缺陷追踪的"发现→修复→验证→闭环"和AI商业化的"需求→设计→交付→复盘"是同构的
- 质量意识**——理解了"什么是好的产品交付标准"，这让我在做AI产品时会主动思考"这个AI方案的质量红线在哪里"

Q6: 怎么在测试资源紧张时做取舍？

答：核心是**风险优先**：

- 先保P0——阻塞上线的功能性Bug必须全覆盖
- P1按影响面裁剪——影响全体玩家的P1优先于影响特定条件玩家的P1
- P2在时间充裕时补充
- 同类型活动的历史缺陷数据可以帮助判断——上个版本收集类活动在"边界条件"维度出过3个P0 Bug，那这次重点加强这个维度

三、面试叙事模板

Part A 叙事（智能体拟真交互，可直接背诵）

在米哈游，我的核心项目是**旗舰开放世界项目的AI NPC交互设计**。

为什么做：项目里有一个核心社交NPC——场景中的"老板娘"，她是玩家在游戏世界的情感锚点。但传统方案下她走固定对话树、没有记忆、情感表达机械——玩家感觉在和"失忆的复读机"对话，直接打破了开放世界精心构建的沉浸感。

怎么做：我设计了三层方案。**第一层，端到端情感语音**——让老板娘能“听懂”玩家的情绪并用对称的情感回应，不是识别关键词，而是直接从语音感知情感状态。**第二层，RAG记忆机制**——短期记忆保证单次对话连贯，长期记忆向量化存储历史关键事件，让她能主动引用“你上次说要去打Boss，打赢了吗？”。**第三层，环境事件联动**——老板娘作为主Agent，周围的酒客、吟游诗人等辅助Agent通过事件总线共享玩家状态，各自根据性格设定做出低频、非侵入式的侧面反应。

做出什么：成功上线—测验证方案可行性——玩家确实能感知到NPC“记得自己”和“有情感回应”，验证了整个技术路线的落地可行性。

Part B 叙事（活动质量体系，简洁版）

同时我还负责**活动质量体系建设**。项目的活动系统复杂且迭代频繁，测试覆盖压力大。我做了三件事：重构标准化测试用例模板（按活动类型×测试维度矩阵化）、建立缺陷全链路追踪流程（发现→修复→验证→闭环+超时预警）、统筹Beta阶段质量管理（测试计划→每日同步→质量评估）。核心价值是把验证流程从“上线后被动修Bug”转向“Beta阶段主动拦截”。

最后更新：2026-04-09 15:32 | 基于简历描述 + 合理推演