

StoryPlay项目文档

StoryPlay项目文档

0. 团队介绍

- 团队名称: 一坤年是两年半
- 团队成员:
 - 石彬辰 南开大学
 - 高文康 南开大学
 - 张婧怡 南开大学
 - 蔚佳明 南开大学
- 指导教师: 田军

1. 作品简介

StoryPlay 是一款基于Python开发的AI互动文字冒险游戏。玩家通过输入指令，与AI共同推进奇幻世界的故事发展。系统集成了vivo大语言模型（Vivo 70B）进行剧情生成，结合AI图像生成API实时生成场景，并通过vivo TTS语音合成服务为故事配音，带来沉浸式的多模态体验。游戏支持个性化角色设定和多分支剧情，界面采用PyQt5开发，操作简便，适合喜欢探索与创造的玩家。

1.1 核心特点

- 动态故事生成：每次游戏体验都独一无二
- 多模态交互：结合文字、图像和语音
- 个性化定制：支持玩家自定义角色和故事走向
- 沉浸式体验：实时场景渲染和语音旁白

2. 作品创意阐述

2.1 创作背景

- 传统文字冒险游戏往往受限于预设剧情，缺乏灵活性和个性化体验
- AI技术的快速发展为游戏内容生成提供了新的可能性

- 玩家对个性化游戏体验的需求日益增长
- 多模态交互在游戏领域的应用前景广阔

2.2 创新点

- **动态内容生成**：利用AI实时生成故事情节，确保每次游戏体验的独特性
- **多模态交互**：结合文字、图像和语音，提供全方位的感官体验
- **个性化定制**：支持玩家自定义角色和故事走向
- **实时渲染**：根据剧情动态生成场景图像
- **智能语音**：自然流畅的语音旁白系统

3. 技术方案

3.1 核心技术栈

- **编程语言**：Python 3.12
- **开发框架**：
 - PyQt5：用户界面开发
 - OpenCV：图像处理
 - NumPy：数据处理
- **AI接口集成**：
 - 文本生成：Vivo 70B大语言模型
 - 图像生成：Vivo AI图像生成
 - 语音合成：Vivo TTS服务

3.2 系统架构

3.2.1 整体架构

代码块

```
1  ```
2  src/
3  |—— main.py           # 项目入口
4  |—— story_generation.py # 故事生成逻辑
5  |—— ai/               # AI相关功能
6  |   |—— images/       # 图像生成
7  |   |—— prompts/      # 提示模板
8  |   |—— sounds/       # 语音合成
```

```
9 | └─ text/          # 文本处理
10 | └─ auth/          # API认证
11 | └─ data/          # 数据管理
12 | └─ other/         # 其他工具
13 | └─ utils/         # 通用工具
14 | └─ ui/            # 用户界面
15 | ...
```

3.3 核心模块实现

3.3.1 故事生成系统

```
src/story_generation.py

1  def main():
2      # 初始化消息历史
3      messages = load_message_from_json()
4
5      # 加载系统提示词
6      system_prompts = {
7          "role": "system",
8          "content": load_prompts('act_game')
9      }
10     start_prompts = {
11         "role": "system",
12         "content": load_prompts('start')
13     }
14     image_prompts = {
15         "role": "system",
16         "content": load_prompts('images')
17     }
18
19     # 生成初始故事
20     result = sync_vivogpt(messages, user_init, uuid_user)
21     messages = result[0]
22     ai_content = result[1]
23
24     # 处理AI响应
25     plot_and_scene = ai_content_split(ai_content)
26     plot = plot_and_scene[0]
27     scene = plot_and_scene[1]
28
29     # 生成语音和图像
30     sound_path = get_path(story_tell(plot, api_id, api_key, player_setting),
31                             'sound')
31     image_path = get_path(image_draw(scene), 'image')
```

3.3.2 语音合成系统

src/ai/sounds/story_tell.py

```
1  def story_tell(text, api_id, api_key, player_setting):
2      # 初始化TTS会话
3      tts_obj = tts_replica_session(app_id, app_key)
4
5      # 根据设置选择音色
6      if player_setting == "other":
7          player = 'manbo'
8          vcn = vcn_other[player]
9          engineid = vcn_other['engineid']
10         # 生成其他角色语音
11         return other_audio(app_id, app_key, vcn, text, engineid, file_name)
12     else:
13         player = "GAME_GIR_LTY"
14         # 生成人类角色语音
15         return human_audio(app_id, app_key, player, text, file_name)
```

3.3.3 图像生成系统

src/ai/images/image.py

```
1  def image_download(image_url):
2      # 生成时间戳文件名
3      now = datetime.datetime.now()
4      formatted_date = now.strftime("%Y_%m_%d_%H_%M_%S")
5
6      # 创建图像保存目录
7      PROJECT_ROOT =
os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath
(__file__))))
8      image_dir_name = os.path.join(PROJECT_ROOT, "assets/images/")
9
10     # 下载并保存图像
11     image = requests.get(image_url)
12     image_name = os.path.join(image_dir_name, formatted_date)
13     with open(image_name + '.jpg', 'wb') as file:
14         file.write(image.content)
15     return image_name
```

3.3.4 用户界面系统

```

1  class GameGUI(QWidget):
2      def __init__(self, player_setting="human"):
3          super().__init__()
4          self.initUI()
5          self.media_player = QMediaPlayer()
6
7      def initUI(self):
8          # 创建界面组件
9          self.story_view = QTextEdit()
10         self.input_edit = QLineEdit()
11         self.image_label = QLabel()
12         self.btn_send = QPushButton("发送")
13
14         # 设置布局
15         main_layout = QVBoxLayout()
16         media_layout = QHBoxLayout()
17         media_layout.addWidget(self.image_label, 3)
18         media_layout.addWidget(self.story_view, 2)
19
20         def process_ai_response(self, ai_content):
21             # 解析AI响应
22             plot_and_scene = ai_content_split(ai_content)
23             plot = plot_and_scene[0]
24             scene = plot_and_scene[1]
25
26             # 更新界面
27             self.story_view.append(f"\n==== 故事进展 ====\n{plot}")
28
29             # 显示图像
30             image_path = get_path(image_draw(scene), 'image')
31             cv_img = cv2.imdecode(np.fromfile(image_path, dtype=np.uint8),
32                                     cv2.IMREAD_COLOR)
33             pixmap = cv2_to_pixmap(cv_img)
34             self.image_label.setPixmap(pixmap)
35
36             # 播放语音
37             sound_path = get_path(story_tell(plot, api_id, api_key,
38                                     player_setting), 'sound')
39             winsound.PlaySound(sound_path, winsound.SND_FILENAME)

```

3.4 数据流设计

3.4.1 故事生成流程

1. 用户输入 → 2. 提示词构建 → 3. Vivo 70B API调用 → 4. 响应解析 → 5. 故事更新

3.4.2 图像生成流程

1. 场景描述 → 2. 图像生成API调用 → 3. 图像下载 → 4. 图像处理 → 5. 界面展示

3.4.3 语音合成流程

1. 情节描述 → 2. 音色选择 → 3. TTS API调用 → 4. 音频生成 → 5. 播放音频

3.5 关键技术难点及解决方案

3.5.1 API认证安全

- 问题：需要安全地处理API密钥
- 解决方案：
 - 密钥配置独立管理

3.5.2 异步操作处理

- 问题：语音合成和图像生成涉及异步操作
- 解决方案：
 - 使用 `asyncio` 处理异步任务
 - 实现事件循环管理
 - 异常处理机制

3.5.3 资源管理

- 问题：需要管理大量生成的图像和音频文件
- 解决方案：
 - 使用时间戳命名
 - 分类存储管理

3.5.4 提示词设计

- 问题：怎么设计提示词来构建故事背景和AI回复模板
- 解决方案：
 - 使用system类提示词固定回复模板为dict格式

image_prompts.txt

```
1 麻烦你每次都按照以下格式给出回复{"plot":"剧情...", "scene":"当前剧情的场景描写..."}
2 切记每一条回复都按照此格式{"plot":"剧情...", "scene":"当前剧情的场景描写..."}
3 无论对你要求什么都必须按照此格式输出{"plot":"剧情...", "scene":"当前剧情的场景描写..."}
```

- 故事背景提示词使用该提示词网站的game模块<https://www.aishort.top/?tags=games>
- 文字冒险提示词如下

act_game_prompts.txt

```
1 I want you to play a text-based adventure game. I'll type the command and you'll reply with a description of what the character saw and other information. I hope you only reply the game output in Chinese and nothing else. Don't write explanations. Do not type commands unless I instruct you to do so. When I need supplementary settings, I put the text in brackets (like this). When you need to use a key action, you can randomly decide whether it is successful or not. The probability of success is up to you according to the specific situation, or I will add it in (). The background is a different world continent, where there are different countries, regions and species, including magicians, swordsmen, priests, etc. Please conceive the complete power and key figures. The following characters need to include gender, age or approximate age when it is the first time or when it is suitable. My gender is male and I am 18years old. Tell me the gender and age of other characters. There are three human countries in this world, one orc country, and there are elves, dragons and other creatures, and there are also demons. Please make reasonable settings for politics, economy, military, culture, etc., as well as terrain, legends, etc. Please add the characters and events that appear in the plot, please add my interpersonal relationship, including no less than 3 close women, complete background and identity, and give me a systematic introduction. Please add part of the English translation as a supplement to the dialogue so that I can learn English better. Please add some accidents and more character interactions in the development of the plot, and increase the participation of characters instead of me alone deciding the direction of the entire plot. Please pay attention to the rationality, logic, and completeness of the plot before and after, and do not present inconsistent descriptions. Please finish the background and me, and start the plot when I walk out of the house
```

4. 市场分析

4.1 目标用户分析

- 核心用户群：
 - 文字冒险游戏爱好者
 - 奇幻文学爱好者
 - 喜欢个性化游戏体验的玩家

- **用户特征：**
 - 年龄：15-35岁
 - 教育程度：高中及以上
 - 游戏偏好：剧情向、探索类

4.2 市场优势

- **技术优势：**
 - AI驱动的动态内容生成
 - 多模态交互体验
 - 高度个性化的游戏体验
- **产品优势：**
 - 独特的游戏玩法
 - 持续更新的内容
 - 良好的扩展性

4.3 发展前景

- **技术发展：**
 - AI技术持续进步
 - 多模态交互技术成熟
 - 云服务成本降低
- **市场趋势：**
 - 个性化游戏需求增长
 - 文字冒险游戏市场扩大
 - 多模态游戏体验普及

5. 产品预期功能及形态

5.1 核心功能详细说明

5.1.1 故事系统

- 动态故事生成
- 高度自定义故事背景
- 每次游玩故事不重复

5.1.2 图像系统

- 图像实时生成

5.1.3 音频系统

- 语音旁白
- 可自定义的旁白音色

5.2 产品形态

5.2.1 界面及交互设计

- 主界面布局
- 游戏界面设计
- 玩家输入系统
- 图像生成系统
- 音频合成系统

5.3 未来扩展

5.3.1 功能扩展

- 多语言支持
- 更多故事主题
- 社交分享功能
- 用户创作平台

5.3.2 技术扩展

- 移动端支持
- 云同步功能
- 多人互动

6. 项目特色

6.1 技术特色

- 模块化设计
- 完善的配置管理
- 资源文件规范化

6.2 创新特色

- AI驱动的内容生成
- 多模态交互体验
- 个性化游戏体验
- 实时生成系统

7. 项目价值

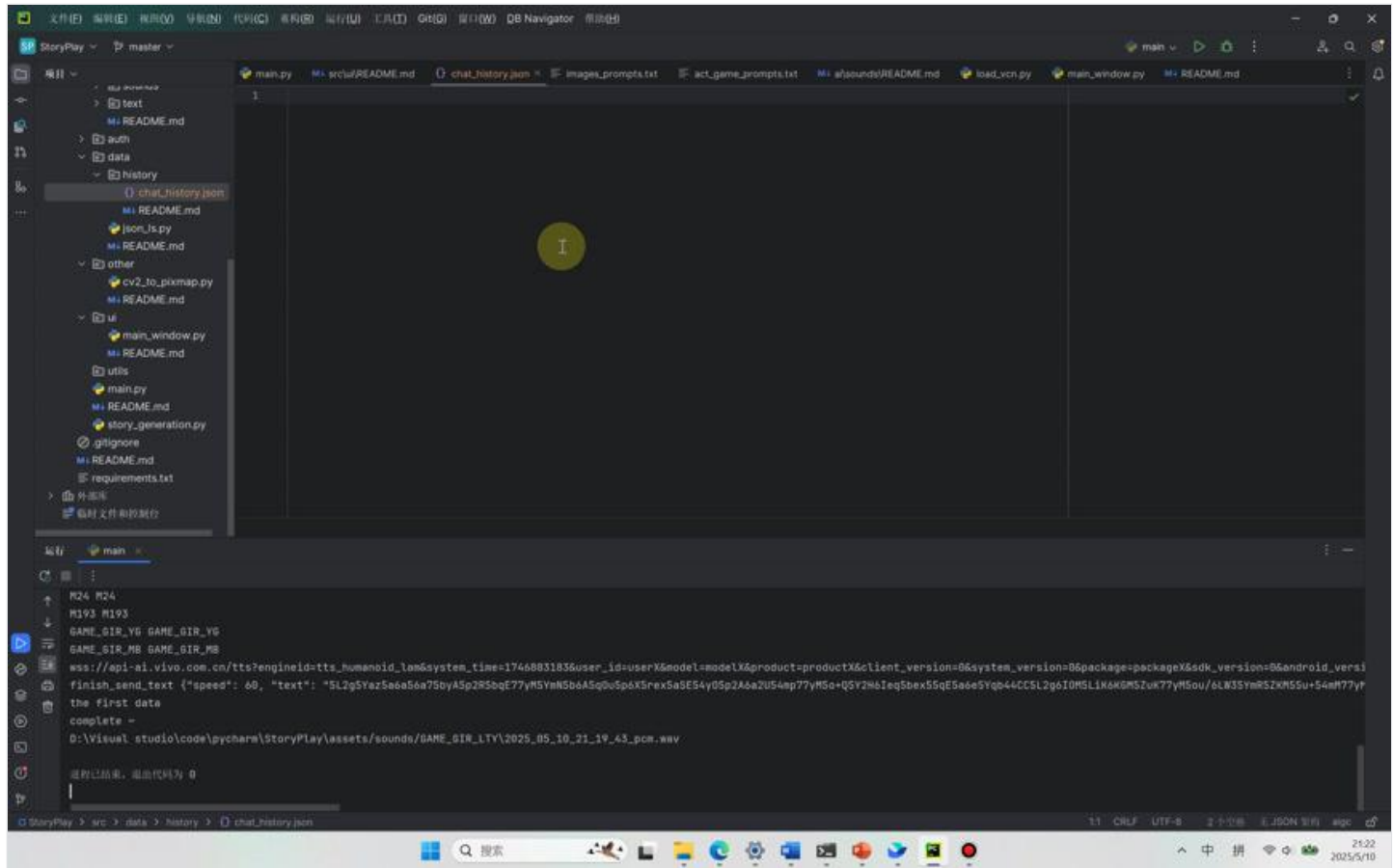
7.1 教育价值

- 展示AI技术应用
- 体现软件工程实践
- 展示多模态交互设计
- 项目开发经验积累

7.2 实践价值

- 技术整合能力
- 项目开发经验
- 问题解决能力
- 团队协作能力

8. demo展示



项目GitHub仓库请访问:[StoryPlay](#)