**游戏主题：交互式末日逃生**

**各项人员负责情况**  
展星宇--游戏程序编写（python）

陈嘉辉--游戏剧情编写  
赵一帆--美术插图绘画

高麒栋--ppt展示和示范视频  
周帅--pygame运行游戏调试并帮助组员完成任务

**项目名称：末日余晖：抉择与生存**

项目愿景： 开发一款轻量级的2D末日生存交互游戏，核心玩法融合探索、资源管理、NPC对话与抉择，通过视觉小说的形式展现剧情和角色。

**技术栈：**

**编程语言： Python**

**核心游戏引擎/库： Pygame** *(一个非常流行的Python 2D游戏库，功能强大，社区活跃。）*

**美术工具： Aseprite / Photoshop / Krita** *(像素画或普通2D美术)，***或使用免费资源库***（如 itch.io, OpenGameArt.org）*

音频工具： Bosca Ceoil *(免费音乐制作)* **或 bfxr** *(音效生成)*，**或使用免费资源库**

# 项目计划（按四周一个阶段划分）

**第一阶段：需求分析与设计搭建 (第1-4周) - “奠基阶段”**

*目标： 完成所有前期设计工作，搭建稳定的开发环境，并创建可运行的最小化demo。*

需求细化 (第1周):

撰写详细的游戏设计文档（GDD），包括：

故事大纲： 世界观、主角背景、主要剧情节点。

核心玩法： 移动、交互（拾取、对话）、简单的生存指标（如饥饿值）。

角色设定： 主角和至少2个关键NPC的设定、对话树草图。

美术风格确认： 确定是像素风格还是其他2D风格，统一分辨率（如 1280x720）。

**技术选型与环境搭建 (第1周):**

安装Python (3.8+版本)。

安装Pygame库 (pip install pygame)。

配置代码编辑器（如VSCode、PyCharm）。

运行Pygame官方示例，确保环境无误。

**引擎框架搭建 (第2-4周):**

创建游戏窗口和主循环。

实现状态管理： 设计游戏状态机（如：开始菜单、游戏进行中、对话中、暂停菜单）。

**实现基础系统：**

渲染系统： 能够加载和绘制背景、角色精灵（Sprites）。

输入系统： 处理键盘（WASD/方向键移动）和鼠标事件。

音频系统： 能够播放背景音乐和音效。

交付成果： 一个可以显示静态背景、一个可移动的玩家角色（使用占位符方块或简单图片）、按下ESC可弹出菜单的可执行程序。

**第二阶段：核心玩法实现 (第5-8周) - “攻坚阶段”**

*目标： 实现游戏的所有核心交互功能。*

**完善移动与碰撞 (第5周):**

实现玩家角色的八向或四向平滑移动。

创建简单的碰撞系统（使用矩形碰撞即可），让玩家不能穿过墙壁（地图边界）。

加载一个简单的测试地图（例如，一个房间和几条走廊）。

**对话系统开发 (第6周) - “Galgame核心”:**

设计一个简单的对话数据格式（如JSON）。

json

{

"npc\_id": "old\_man",

"dialogues": [

{"text": "你好，幸存者。外面的情况怎么样？", "next": 1},

{"text": "很糟糕？我就知道...我这里有些干净的水，你需要吗？", "next": 2, "gives\_item": "water"}

]

}

实现对话界面UI：显示NPC头像、姓名、对话文本。

实现对话逻辑：按空格键继续下一句，对话结束后返回游戏。

**交互与物品系统 (第7周):**

实现“可交互物”系统：当玩家靠近NPC或物品时，显示提示（如“按E交谈”）。

实现一个简单的物品库存系统（UI界面），可以收集和使用物品（如使用食物恢复饥饿值）。

**整合与测试 (第8周):**

将移动、碰撞、对话、交互系统整合在一起。

创建一个简单的可玩场景：玩家找到一把钥匙，与一个NPC对话获得信息，然后用钥匙打开一扇门。

交付成果： 一个包含完整核心玩法循环的Demo。玩家可以移动、与物体碰撞、与NPC对话并获取物品。

**第三阶段：内容填充与优化 (第9-12周) - “丰富阶段”**

*目标： 让游戏变得丰富、有趣、美观。*

**美术资源导入 (第9周):**

用最终制作或下载的美术资源替换所有占位符（玩家角色、NPC、地图瓦片、物品图标、UI元素）。

确保资源尺寸统一，风格一致。

**地图与关卡设计 (第10周):**

使用Tiled地图编辑器或其他方法制作更大的游戏地图。

设计多个小场景，并通过路径连接起来（如避难所、废墟街道、商店）。在场景中放置更多的交互点和NPC。

**剧情与对话填充 (第11周):**

撰写所有NPC的完整对话树，确保分支和选择能影响剧情（即使是微小的影响）。

将对话内容填入JSON文件，并在游戏中测试。

**生存系统与玩法深化 (第12周):**

实现简单的生存指标（如饥饿值、生命值），并使其随时间下降。

增加寻找食物、水的需求，强化生存压力。

加入简单的昼夜循环或天气系统（可选，如果时间允许）。

交付成果： 游戏内容基本完成，拥有完整的第一张地图、所有NPC对话、初步的生存玩法。游戏看起来已经像最终产品。

**第四阶段：打磨与发布 (第13-16周) - “抛光阶段”**

*目标： 修复错误，提升体验，完成最终发布。*

**全面测试与Debug (第13-14周):**

进行内部测试，邀请同学朋友试玩。

收集反馈，修复所有发现的Bug（碰撞错误、对话卡死、物品无法使用等）。

优化代码性能，确保运行流畅。

**音频与UI抛光 (第15周):**

添加背景音乐（BGM）和所有音效（脚步声、交互音、对话声效）。

完善UI动画和反馈（如按钮高亮、菜单滑动效果）。

制作开始菜单、结束菜单和游戏暂停界面。

**最终整合与打包 (第16周):**

录制游戏宣传视频、制作和完善游戏宣传ppt。

使用pyinstaller等工具将Python脚本打包成可执行文件（.exe），方便他人在没有Python环境的电脑上运行。

整理项目源代码、设计文档和最终报告。

交付成果： 一个稳定、完整、可独立运行的游戏最终版本，以及所有项目文档。

**风险与应对策略**

风险1： 代码结构混乱，后期难以维护。

对策： 前期就注重代码模块化，将不同系统（渲染、输入、游戏逻辑）分开编写。

风险2： 美术资源制作耗时过长。

对策： 优先使用免费/付费的现成资源包，或在前期使用简单的占位符，核心玩法实现后再替换。

风险3： 功能范围蔓延，无法按期完成。

对策： 严格遵守“简单”的原则。先完成核心功能（移动、对话），额外功能（复杂的生存系统）在核心完成后作为加分项添加。