

# 人机交互技术 小组项目最终报告 2025.6.15

组员：闫皓翔、王艺衡、毕方舟

## 1. 项目内容

本项目是基于 Pygame 库开发的一款 STG（平面弹幕射击游戏），意图做到还原东方 Project 系列弹幕游戏。

## 2. 实现功能

游戏采用面向对象编程的思想设计，计划实现一个基础的人机交互的游戏系统，该游戏系统包括：

**基础游戏框架：**窗口管理、资源加载、音效播放

**核心玩法系统：**自机移动、弹幕生成、敌机 AI、Boss 战

**UI 交互系统：**菜单、HUD、暂停、结算界面

**游戏交互机制：**符卡系统、子机系统、资源收集、高低速切换

游戏的基础框架已经大体搭建完毕，游戏需要的相关贴图与音频文件已经导入，实现了场景堆栈管理。核心玩法相关的算法部分已经实现，有关自机与敌机的相关判定算法和行为逻辑已经编写完毕，有待后续对关卡设计上的内容运用。UI 系统设计了主菜单，游戏内菜单，游戏结算菜单，大体功能逻辑已经设计完毕。开发全过程采用了模块化开发，分工明确，可维护性较强，可开发空间很大。

## 3. 人机交互设计

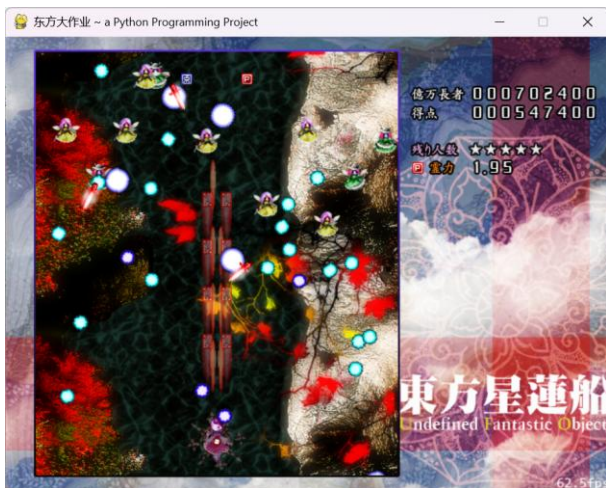
在开发过程中，我们希望能够运用了课程中所学习的人机交互技术的多项理论，包括认知心理学、情感化设计和交互设计原则等，结合课程材料中提供的相关内容，我们目前有意在游戏的这些方面引入人机交互技术的理论，具体实现内容如下：

### (1) 输入交互设计

**键盘控制：**

**方向控制：**玩家通过键盘上的上、下、左、右箭头键来操控自机（玩家操控的飞机角色）在游戏画面中的移动。这种经典的方向控制方式简单直观，符合大众玩家对于此类游戏的操作习惯。无论是躲避密集的敌方子弹，还是调整自机位置以更好地攻击敌机，玩家都能通过精准的键盘操作实现。例如，在面对敌方弹幕攻击时，玩家可以迅速按下相应方向键，让玩家可以灵活穿梭于弹幕之间，使用高超的技巧取得很高的分数。

**射击控制：**按下“Z”键，自机进行常规射击。射击频率根据游戏设定以及自机的当前状态有所不同。在游戏进程中，玩家需要根据敌人的出现时机和弹幕的密集程度，合理把握射击节奏，以达到消灭敌机、获取分数和资源的目的。例如，在敌方小怪成群出现时，玩家可以持续按下“Z”键，进行连续射击，快速清除小怪，为应对后续更强大的敌人做好准备。



**强力攻击 (Bomb)：**按下“X”键，自机释放强力攻击技能，即“Bomb”。此时，自机短暂进入无敌状态，使用风神录系统，消耗火力点数进行特殊攻击。Bomb 技能具有冷却时间和使用次数限制，玩家要谨慎选择释放时机。当玩家陷入敌方弹幕的绝境难以躲避时，使用 Bomb 技能可以摆脱僵局，清除周围的子弹和敌人，为自己争取喘息的机会。



**高速 / 低速模式切换：**通过按下“LShift”键，自机在高速和低速模式之间切换。在高速模式下，自机移动速度加快，便于快速躲避远距离的子弹和到达特定位置；在低速模式下，自机移动速度减慢，但攻击方式可能会发生变化，例如子弹的射速、射击角度等，并且会显示判定点，帮助玩家更精准地躲避子弹和攻击敌人。这种模式切换机制增加了游戏的策略性和操作的多样性。比如，在面对敌方复杂的弹幕阵型时，玩家可以先切换到低速模式，仔细观察弹幕规律，利用低速模式下更精准的攻击和判定点提示，安全地穿过弹幕，然后在合适的时机切换回高速模式，快速接近敌人进行攻击。

**暂停游戏：**按下“Esc”键，游戏进入暂停状态。此时，游戏画面暂停，弹出暂停菜单，玩家可以选择继续游戏、返回主菜单或调整游戏设置等操作。暂停功能在玩家需要暂时离开游戏但又不想丢失游戏进度时非常实用，确保游戏进程可以随时被灵活控制。



**游戏重启：**在游戏暂停状态下，玩家可以选择重新开始游戏，这一操作允许玩家在当前关卡表现不佳时，快速重新挑战，节省了返回主菜单再重新开始的时间，提高了游戏的流畅性和可玩性。

## (2)输出反馈设计

### 视觉反馈：

**游戏画面显示：**游戏场景包括动态背景，如随风飘动的云朵、流动的星空等，这些动态元素不仅增加了游戏的美感，还营造出沉浸式的游戏氛围。同时，自机、敌机、子弹、资源道具等游戏元素的绘制精细，具有鲜明的色彩和独特的造型设计，易于玩家区分和识别。例如，自机的造型设计独特，色彩鲜艳，在复杂的游戏场景中也能清晰地被玩家辨认，方便玩家进行操作。

**HUD 系统：**游戏界面右侧的 HUD 系统实时显示玩家的关键游戏信息。其中，分数计数器记录玩家在游戏过程中获得的分数，随着玩家消灭敌机、收集资源等操作，分数实时增加，让玩家直观感受到自己在游戏里的成就。生命值显示区域以直观的图形或数字形式展示自机的当前生命值，当受到敌方攻击时，生命值减少，提醒玩家注意自身状态。火力值显示则反映自机的攻击威力，随着游戏进程中玩家获取特定道具或提升等级，火力值会发生变化，影响自机的攻击效果和攻击的强度。此外，HUD 系统还可能包括其他信息，如游戏时间、剩余炸弹数量等，为玩家提供全面的游戏状态信息，帮助玩家针对游戏进程做出正确判断。

**特效反馈：**当玩家进行各种操作或游戏中发生重要事件时，会有丰富的特效反馈。例如，自机射击时，子弹发射会伴有光影特效，增加射击的视觉冲击力；释放 Bomb 技能时，屏幕会出现强烈的闪光和爆炸特效，同时敌方单位被瞬间清除，让玩家切实感受到技能的强大威力。当敌机被击中或摧毁时，会产生爆炸特效和碎片效果，并且根据敌机的不同类型，特效表现也有所差异，增强了游戏的打击感和趣味性。

### 听觉反馈：

**音效反馈：**各种游戏操作和事件都对应着独特的音效。自机射击时，会发出清脆的射击音效，让玩家感受到射击的节奏感；释放 Bomb 技能时，伴随着强烈的爆炸声效，增强技能释放的震撼感。敌机被击中或摧毁时，会产生不同的爆炸音效，根据敌机的大小和类型有所区别，增加游戏的真实感。当玩家获取资源道具时，会播放相应的提示音效，告知玩家成功获得了有用的物品。此外，游戏中的菜单操作也配有音效，如点击菜单选项时的确认音效，让玩家的操作得到及时反馈，提升游戏的交互体验。

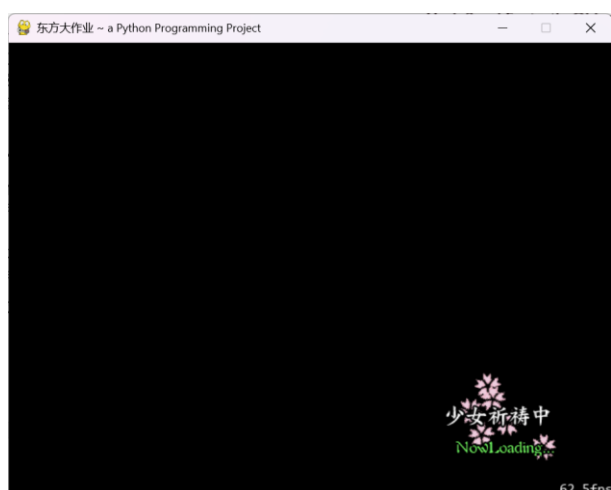
## (3)菜单与界面交互设计

## 主菜单：

**布局与选项：**主菜单界面设计简洁明了，以吸引人的游戏主题画面为背景，突出显示游戏的主要选项。包括“Game Start”（开始游戏）、“Extra Start”（额外模式开始，可能包含隐藏关卡或特殊挑战）、“Practice Start”（练习模式开始，玩家可以在此模式下熟悉游戏操作和关卡内容，无分数记录和游戏进度保存）、“Replay”（重播上次游戏，玩家可以回顾自己的游戏过程，分析自己的操作失误和优点）、“Result”（查看游戏结果，包括历史最高分、当前游戏的得分、游戏时长等详细信息）、“Music Room”（音乐室，玩家可以在此处选择播放游戏中的各种音乐，欣赏游戏配乐）、“Option”（游戏设置选项，包括音量调节、画面质量设置、控制键位自定义等）。



**交互效果：**当玩家将鼠标指针悬停在菜单选项上时，选项会有明显的视觉变化，如颜色加深、边框亮起或文字放大等，以提示玩家当前选中的选项。点击选项时，会播放相应的音效，并且界面会根据玩家的选择进行相应的切换。例如，点击“Game Start”选项后，游戏会进入加载界面，然后开始正式游戏。



## 暂停菜单：

**功能与设计：**在游戏过程中按下“Esc”键，会弹出暂停菜单。暂停菜单覆盖在游戏画面之上，以半透明的背景显示，既不影响玩家查看游戏画面的大致情况，又能突出菜单选项。菜单选项包括“Resume”（继续游戏）、“Restart”（重新开始当前关卡）、“Main Menu”（返回主菜单）、“Options”（再次进入游戏设置界面）。玩家通过键盘上的方向键选择菜单选项，按下“Enter”键确认选择，这



种操作方式与主菜单的交互方式保持一致，方便玩家快速上手。



### 游戏结束界面：

**信息展示：**当玩家在游戏中生命值降为零或完成游戏目标时，会进入游戏结束界面。界面会显示玩家的最终得分、历史最高分对比、游戏时长等信息。如果玩家在游戏过程中有特殊成就或完成了特定任务，也会在此界面进行展示和提示。例如，玩家在游戏中达成了不使用 Bomb 技能通关的成就，游戏结束界面会特别显示该成就，并给予玩家相应的奖励提示。

**操作选项：**游戏结束界面提供“Retry”（重试当前关卡）和“Main Menu”（返回主菜单）两个选项。玩家可以根据自己的意愿选择重新挑战当前关卡，或者返回主菜单选择其他游戏模式或进行其他操作。同样，通过键盘方向键和“Enter”键进行选项选择和确认操作，确保玩家能够方便快捷地进行后续操作。

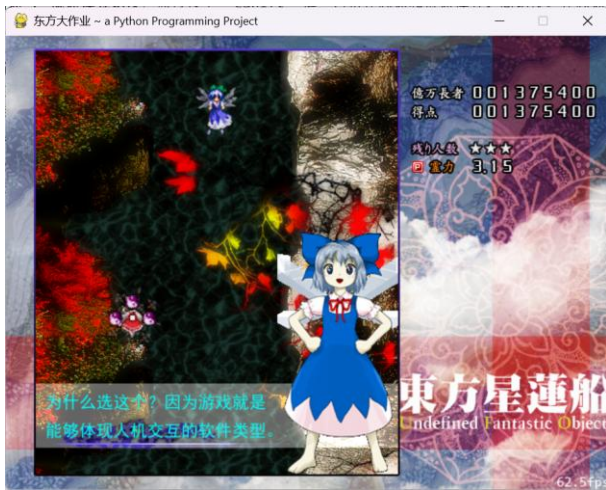


### 对话界面：

**触发与呈现：**在游戏进行到特定阶段，如 BOSS 战前，会自动触发对话界面。对话界面以对话框的形式出现在屏幕底部或特定位置，对话框内显示角色的对话内容。对话内容可以根据游戏剧情和角色设定精心编写。同时，对话框旁边可能会显示参与对话角色的立绘，立绘根据角色的情绪和对话内容进行相应的表情和姿态变化，增强对话的生动性和代入感。

**交互操作：**玩家通过按下“Z”键来推进对话内容。在对话过程中，如果玩家想要暂停对话或返回

游戏，也可以按下“Esc”键，此时游戏会暂停，玩家可以选择继续对话或返回游戏界面。这种简单的交互方式确保玩家能够轻松控制对话节奏，充分体验游戏剧情。



#### 4. 开发实现

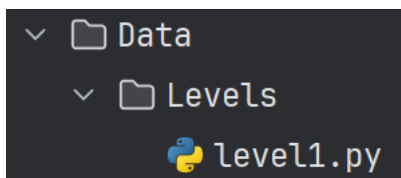
项目基于 Pygame 框架完成开发，采用面向对象设计模式：

##### 核心系统实现

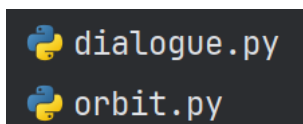
**游戏框架：**场景堆栈管理、资源动态加载（`resource.py`）、多通道音效控制（`music.py`）。

**玩法系统：**自机移动与碰撞检测（`player.py`）、弹幕生成算法（`tama.py`）、敌机状态机 AI（`enemy.py`）。

**交互机制：**符卡技能冷却系统、高低速切换操作优化、子机协同攻击逻辑。

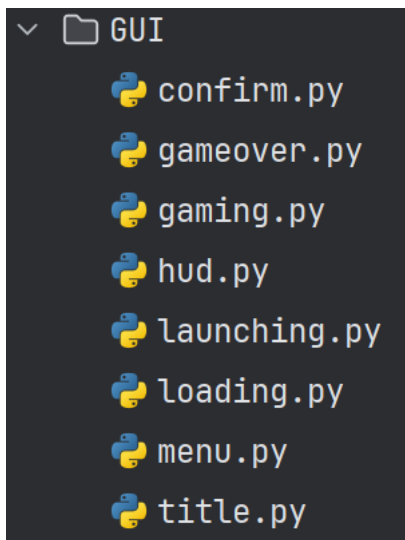


Level 单独存放关卡安排

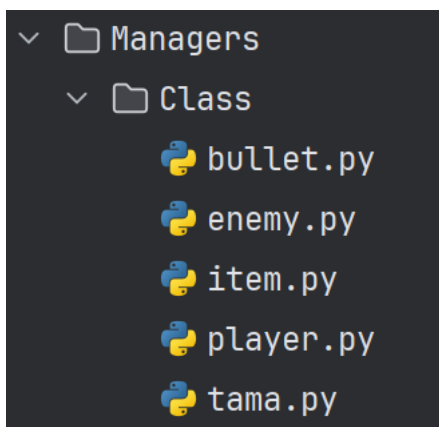


Dialogue 存放剧情对话

Orbit 存放游戏中产生的不同射弹类型



GUI 目录下的文件单独存放每一种菜单 GUI 的页面内容与交互逻辑



Managers 存放游戏中实现的不同系统内容

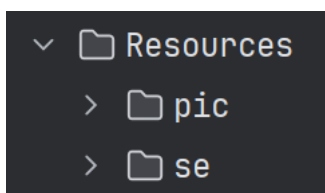
Bullet 具体实现自机与敌机的射弹判定

Enemy 具体实现游戏中的敌机

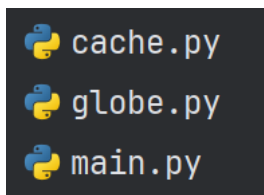
Item 具体实现游戏中产生的分数与灵力资源

Player 具体实现游戏中的自机

Tama 具体实现游戏中自机身旁根据火力点提供的僚机



Resources 具体存放游戏运行需要的贴图与音效文件



Cache 与 Globe 主要存放项目运行中需要存放于内存的资源读取与处理的全局函数

Main 为游戏的主要逻辑启动器

### 人机交互技术应用

**认知层面设计：**通过色彩对比来体现出注意力管理（敌弹与自机弹之间色彩差别很大），引入一些动态效果引导玩家注意力

**多任务处理：**优化 HUD 布局，确保关键信息生命值/分数/技能数量始终在视野范围内，关键信息（生命值/分数/技能）固定于屏幕顶部，避免视觉焦点切换

**情感反馈：**击中敌机时的爆炸特效+音效组合，被命中时的相关反馈

**外观设计：**STG 游戏中的弹幕可以提供华丽的视觉效果，给玩家提供一些兴奋感

**社会化交互：**存在积分榜系统可以促进玩家竞争

### GUI 系统

实现场景化 UI 堆栈：主菜单（menu.py）、实时战斗 HUD（hud.py）、暂停界面（confirm.py）与结算界面（gameover.py），支持键盘/鼠标双向交互。

## 5. 实验评估

经过小组成员的**可用性测试**，能够正常完成程序提供的完整游玩过程，没有产生游戏漏洞，基础游戏功能实现完毕，部分原作中的细节没有做到，游戏设置等内容未能实现。

因为本实验整体聚焦于人机交互部分，所以在开发周期内主要实现了游戏功能，小组成员基于以下方面对该项目进行了**启发式评估**：

**系统状态可见性：**系统运行稳定，游戏过程中不会产生恶性漏洞

**系统与现实世界匹配：**系统大体还原了东方 project 游戏的主要功能

**用户控制与自由：**玩家可以正常控制游戏中角色

**一致性与标准：**游戏内容每次执行时流程保持一致

**错误预防：**游戏代码运行时有数据溢出探测，运行逼近系统极限时会进行资源释放来进行错误预防

**灵活性与效率：**本项目性能需求低，本地部署后均可以理想情况下执行

**美观与简洁：**本项目的 GUI 根据东方原作设计相关内容，没有冗余信息，基于审美需求保证了一定的美观度

**错误识别、诊断与修复：**当游戏运行出错时会对产生错误的模块进行报错，可以精确定位到模块与函数

**帮助与文档：**游戏项目配备简易操作文档