“市场失灵：污染企业的外部性挑战”说明文档

雷钰琪 3230101767

1. 设计思想
2. 游戏背景：

游戏背景设定在一个**市场经济环境**中，玩家扮演一家**工厂的决策者**。工厂的生产活动会产生污染，对周边环境造成负面影响（负外部性）。游戏通过**多轮决策模拟**，让玩家体验个人利润最大化与社会福利最大化之间的矛盾。

（2）经济学原理：

**外部性**：外部性又称为溢出效应、外部影响、外差效应或外部效应、外部经济，指一个人或一群人的行动和决策使另一个人或一群人受损或受益的情况。小游戏主要体现了经济外部性，其是指经济主体（包括厂商或个人）的经济活动对他人和社会造成的非市场化的影响。本游戏聚焦负外部性（污染）。

**帕累托最优**：帕累托最优，也称为帕累托效率，是指资源分配的一种理想状态，假定固有的一群人和可分配的资源，从一种分配状态到另一种状态的变化中，在没有使任何人境况变坏的前提下，使得至少一个人变得更好，这就是帕累托改进或帕累托最优化。

**社会最优生产**：由于本游戏中存在外部成本，社会最优产量低于私人利润最大化产量。

**市场失灵**：市场失灵是指在不完全竞争市场下，由于各种原因而无法实现资源的最优配置。一般认为，导致市场失灵的原因包括垄断、外部性、公共物品和不完全信息等因素。游戏中的市场失灵主要来源于外部性，当个体仅追求私人利润最大化时，会导致过度生产和污染，降低社会福利。

（3）核心玩法与机制说明：

玩家将扮演**工厂决策者**，在**5个生产周期**中通过**产量决策**，亲身体验市场失灵现象的核心原理。

游戏采用回合制进行，共分为5个生产周期。每个回合中，玩家需要通过滑块设定本期的生产量（0-100单位）。这一关键决策将同时影响**两个核心指标：企业利润和环境污染**。

利润计算采用简单线性公式：**利润=生产量×2**，体现生产带来的直接经济效益；

污染计算采用非线性公式：**污染量=生产量×(1+产量/100)**，模拟现实世界中污染随生产规模加速增长的特性。

游戏引入了独特的"污染累积机制"：每期产生的污染会加入环境总污染量，这意味着随着游戏进程，相同的生产量会造成越来越高的社会成本，准确模拟了环境污染的持续性和累积效应。其中，**社会成本=累计污染总量×1.2**。

玩家的终极目标是实现"社会福利最大化"，其计算公式为：**社会福利=总利润—社会成本**。

系统预设的"社会最优产量"为每期60单位左右，这一数值平衡了企业利润与环境成本。若玩家追求短期利润最大化（每期100单位），将导致污染剧增和社会成本飙升；反之过度保守的产量则无法实现合理的企业效益。

游戏结束后，系统会**根据最终社会福利水平给出科学评价**：超过理论最优值80%为"优秀"，60%-80%为"良好"，40%-60%为"及格"，低于40%则为"不及格"。

通过5轮决策的实践，玩家将直观理解"负外部性"导致的市场失灵现象，认识个人最优选择（利润最大化）与社会最优选择（福利最大化）的根本冲突，并由此领悟政府通过环境税等干预手段矫正市场失灵的经济学原理。

二、实现步骤

（1）游戏运行

A.**初始化**：设置游戏参数（5轮），创建游戏状态对象。

B.**玩家决策**：每轮玩家通过滑块选择生产量（0-100单位）。

C.**计算影响**：根据核心玩法中的计算公式和玩家给出的生产量计算——私人利润、污染、 社会成本、社会福利。

D.**记录历史**：游戏系统会存储每轮数据供分析。

E.**游戏结束**：5轮后展示最终结果和评价。

（2）操作游戏

A.**滑块控制**：鼠标拖动滑块选择产量或点击滑块条任意位置快速定位

B.**按钮**：

游戏进行中：点击“提交决策”确认本轮产量

游戏结束后：点击“重新开始”重置游戏

C.**数据查看**：

进行中：实时指标卡片显示预估影响

结束后：左右面板分别展示历史曲线和详细数据

三、代码结构

（1）模块划分

A.**初始化与主循环**：`main()`函数，处理事件循环。

B.**游戏状态管理**：`GameState`类，存储游戏数据。

C.**UI组件**：按钮：`Button`类 ; 生产滑块：`draw\_production\_slider()` ; 指标卡片：`draw\_metric\_card()` ; 历史图表：`draw\_history\_chart()` ; 历史表格：`draw\_history\_table()` ;进度环：`draw\_round\_progress()` ; 结束画面：`draw\_game\_over()`。

D.**经济模型**：`GameState.calculate\_effects()`

（2）使用的外部库

A. Pygame：游戏开发框架，用于创建图形界面、处理用户输入和渲染

B. sys：系统功能，用于程序退出

C. math：数学计算，主要用于角度和弧度转换

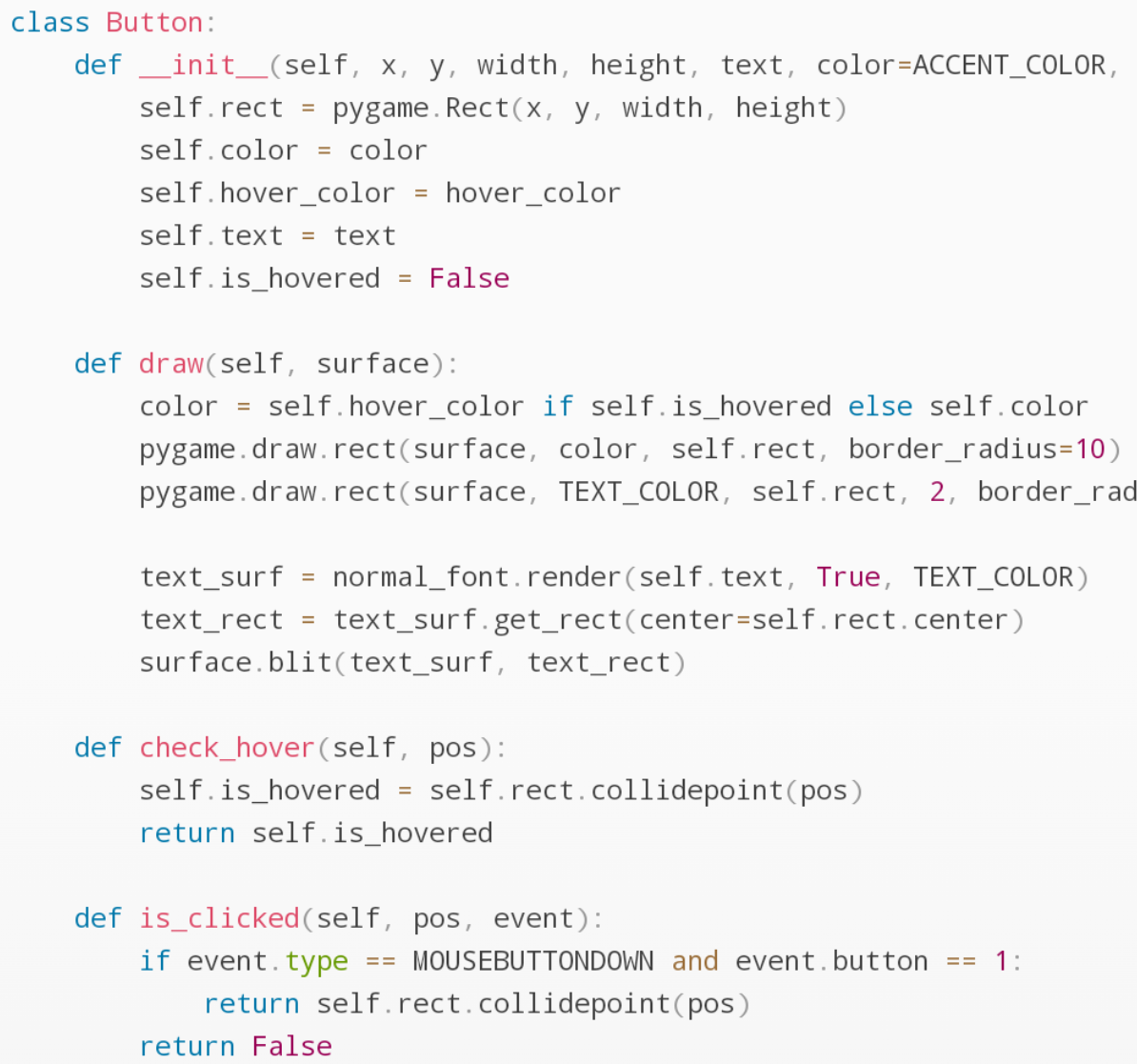
D. numpy：数值计算

（3）关键代码的功能解释

A.**初始化游戏状态**，包括当前回合、总回合数、生产量、利润、累积污染、社会成本等。



B.**初始化按钮的位置、大小、文本和颜色**



C.**绘制生产滑块**,用于用户输入生产决策



D.**绘制经济指标卡片**,显示利润、社会成本和社会福利



E.**游戏核心循环**，处理所有交互和渲染





F.**游戏结束画面**,展示最终游戏结果

