

# 生产要素管理游戏说明文档

3230102783 石文怡

## 一、游戏设计思想

### 1. 游戏背景

本游戏模拟了一个简化版的经济系统，玩家需要管理三种生产要素（劳动力、资金和土地）在三个不同产业（农业、工业和科技）之间的分配。游戏通过多轮生产周期，让玩家体验资源分配决策对生产的影响。

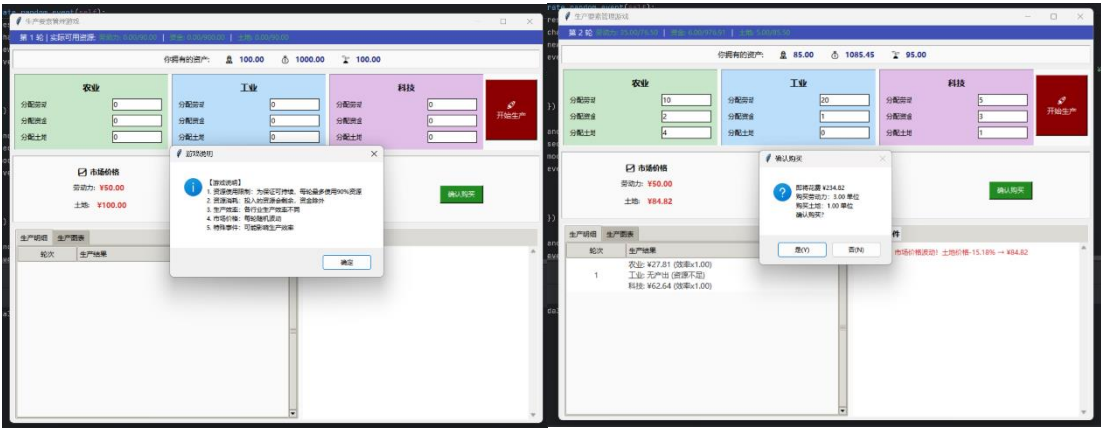
### 2. 经济学原理应用

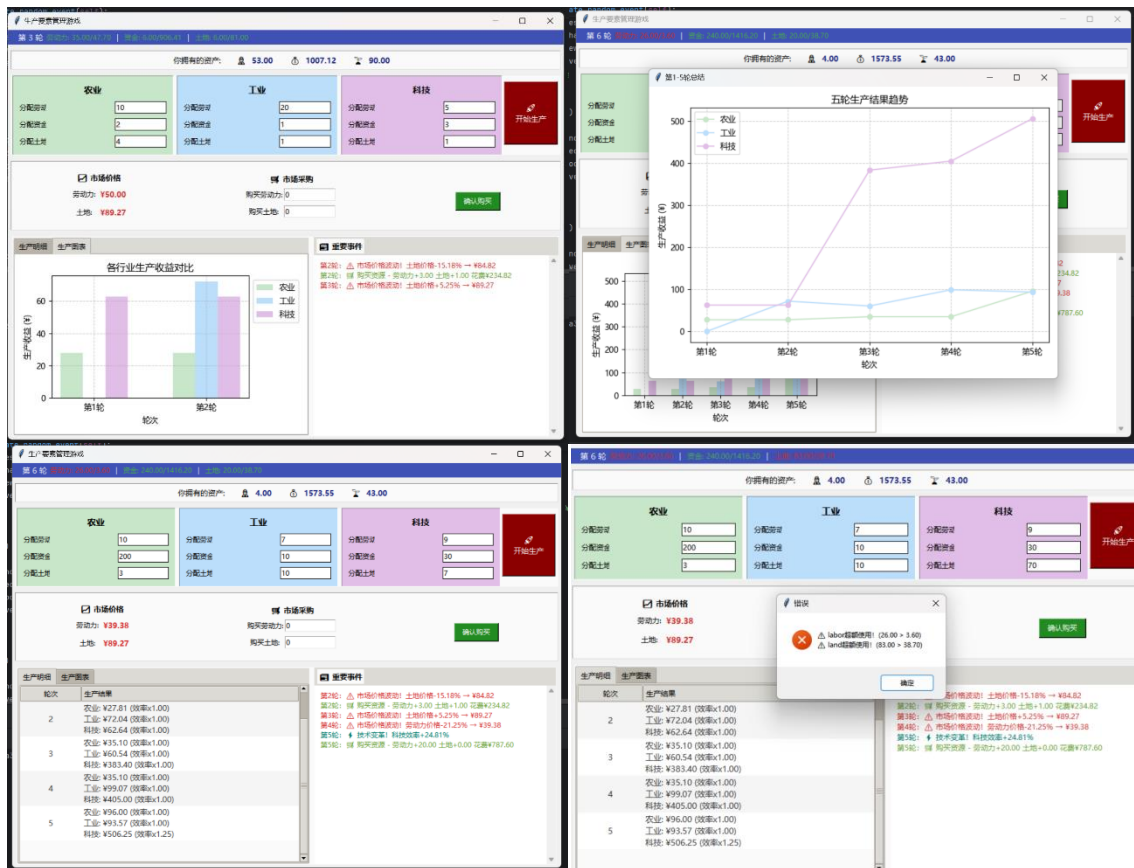
- ①**生产要素分配**：体现了经济学中资源有限性与选择的基本原理。
- ②**边际效益递减**：各产业的生产函数设计体现了投入与产出的非线性关系。
- ③**市场价格波动**：模拟了市场供需变化对资源价格的影响。
- ④**技术变革**：通过效率系数变化体现技术进步对生产力的影响。
- ⑤**政府干预**：随机事件中的政府补贴反映了宏观经济政策的影响。

## 二、实现步骤

### 1. 核心玩法

- ①游戏开始前，会跳出一个弹窗，提醒玩家**游戏规则**。
- ②每轮开始时，玩家将拥有的**生产要素**分配到不同产业（当某行业资源分配中某项数值为 0 时，则无法进行生产获得收益）。
- ③开始生产后，系统根据分配情况和产业特性**计算产出**（各行业生产效率不同）。
- ④产出转化为**资金收入**，可用于下一轮的生产或**资源购买**（每轮分配的资金会全部消耗完，但土地和劳动力会有适当剩余）。
- ⑤每轮可能触发**随机事件**（技术变革/市场价格波动/政府补贴），影响生产效率或资源价格。
- ⑥每轮结束后，玩家可查看**生产明细**和**生产图表**（为节省空间，这两部分放在同一显示框内）。
- ⑦每 5 轮系统会生成**生产趋势总结**。





## 2. 游戏机制说明

- ①资源限制：为体现可持续化发展理念，且防止下一轮玩家无法进行生产，每轮最多只能使用 90%的现有资源。
- ②产业特性：
  - 农业：主要依赖劳动力和土地，资金投入效益较低。
  - 工业：主要依赖劳动力和资金，受科技产业投入影响。
  - 科技：主要依赖资金，高回报但需要持续投入。
- ③随机事件：包括价格波动、技术变革和政府补贴三类，发生概率不同，也可能无事发生。
- ④图表展示：提供生产明细和趋势图表可视化。

## 3. 运行与操作

- ①运行环境：需要 Python 3.x 及 tkinter、matplotlib 库。
- ②启动方式：直接运行脚本生产要素管理游戏.py
- ③操作界面：顶部显示当前轮次和资源使用情况；中间面板用于分配资源和市场交易；底部面板显示生产明细、图表和事件日志。
- ③基本操作流程：
  - 在各产业输入框分配资源——(可选)在市场中购买更多资源——点击"开始生产"按钮——查看结果并进入下一轮

## 三、代码结构说明

### 1. 模块划分

- ①GUI 界面：使用 tkinter 构建主窗口和各种控件。
- ②游戏逻辑：包含在 **EnhancedEconomicGame** 类中。
- ③数据可视化：使用 matplotlib 生成生产趋势图表。
- ④事件系统：处理随机事件生成和效果应用。

## 2. 关键代码段功能

### ①游戏初始化与数据结构

```
def __init__(self, master):  
    self.sectors = ['农业', '工业', '科技'] # 三大产业  
    self.resources = {'labor': 100.00, 'capital': 1000.00, 'land': 100.00} # 初始资源  
    self.prices = {'labor': 50.00, 'land': 100.00} # 资源市场价格  
    self.efficiency = {sector: 1.0 for sector in self.sectors} # 各产业初始效率  
    self.history = [] # 存储历史轮次数据
```

功能：初始化游戏核心数据，包括产业类型、初始资源量、市场价格、生产效率和历史记录。

### ②生产逻辑（核心算法）

```
def start_production(self):  
    # 计算各产业产出  
    if sector == '农业':  
        results[sector] = (labor * 0.6 + land * 1.2) * (1 + capital * 0.015) * 2.5 * efficiency  
    elif sector == '工业':  
        results[sector] = (labor * 0.8 + capital * 0.8) * (1.25 + tech_capital * 0.03) * 3.2 *  
efficiency  
    elif sector == '科技':  
        results[sector] = (labor * 0.5 + capital * 1.1) * 1.8 * 6 * efficiency
```

功能：定义三大产业的生产函数：**农业**（劳动力和土地主导，资金增益较低）、**工业**（劳动力和资金主导，受科技产业资金投入影响）、**科技**（资金主导，高回报乘数）。体现边际收益递减（非线性公式）和产业关联性（工业依赖科技投入）。

### ③随机事件系统

```
def generate_random_event(self):  
    # 价格波动事件 (60%概率)  
    if random.random() < 0.6:  
        res = random.choice(['labor', 'land'])  
        change = random.uniform(-0.25, 0.35) # 价格波动范围  
        self.prices[res] *= (1 + change)  
  
    # 技术变革事件 (40%概率)  
    if random.random() < 0.4:  
        sector = random.choice(self.sectors)  
        modifier = random.uniform(0.85, 1.25) # 效率变化范围  
        self.efficiency[sector] *= modifier  
  
    # 政府补贴事件 (25%概率)  
    if random.random() < 0.25:
```

```
for sector in self.sectors:
    self.encyclopedia[sector] *= 1.15
```

**功能：**随机触发三类事件（**价格波动：**影响劳动力和土地的市场价格；**技术变革：**提升或降低特定产业效率；**政府补贴：**全产业效率提升 15%）。增加游戏不可预测性，模拟真实经济环境中的外部冲击。

#### ④资源分配验证

```
def update_usage_display(self):
    max_use = {
        'labor': self.resources['labor'] * 0.9,
        'capital': self.resources['capital'] * 0.9,
        'land': self.resources['land'] * 0.9
    }
    if total_used[res] > max_use[res]:

        error_msgs.append(f"⚠ {res}超额使用！ ")
```

**功能：**实时检查资源分配是否超过每轮 90%的限制，并显示警告。体现资源稀缺性和可持续性管理。

#### ⑤数据可视化

```
def update_chart(self):
    # 绘制最近 5 轮各产业产出柱状图
    ax.bar([xi + i*width for xi in x], results_data[i], width,
           label=sector, color=self.colors[sector])
    ax.set_title('各行业生产收益对比')
```

**功能：**使用 Matplotlib 生成柱状图，直观展示多轮生产结果对比。颜色区分产业（农业绿、工业蓝、科技紫），支持动态更新。

#### ⑥市场交易逻辑

```
def buy_resources(self):
    total_cost = purchased['labor'] * self.prices['labor'] + purchased['land'] *
self.prices['land']
    if total_cost > self.resources['capital']:
        raise ValueError("资金不足！ ")
    self.resources['capital'] -= total_cost
    self.resources['labor'] += purchased['labor']
    self.resources['land'] += purchased['land']
```

**功能：**允许玩家用资金购买劳动力和土地，需确保资金充足。模拟市场交易行为，提供资源补充途径。

### 3. 使用的外部库

①tkinter：构建图形用户界面

②matplotlib：生成生产数据和趋势图表

③random：生成随机事件和价格波动

④platform：检测操作系统以适配中文字体

游戏通过模拟真实经济系统中的关键要素，让玩家在资源分配决策中理解基本经济学原理，同时提供了直观的数据反馈帮助玩家分析决策效果。