

# 税收与公共支出经济模拟游戏说明文档

## 一、游戏背景与经济学原理应用

### (一) 游戏背景

本游戏基于宏观经济学理论，模拟政府通过调整税收政策和公共支出，实现经济增长、控制财政赤字并平衡社会福利的过程。玩家扮演政策制定者，在 10 个回合内通过调节税率和支出项目，推动 GDP 从 1000 亿增长至 2000 亿，同时确保财政赤字不超过 500 亿，体验政策制定的复杂性与权衡取舍。

### (二) 核心经济学原理

#### 1. 凯恩斯乘数理论：

**政府支出乘数：**增加公共支出会通过需求拉动经济增长，乘数效应为  $\frac{1}{1-MPC}$ （MPC 为边际消费倾向，游戏中设为 0.7）。

**税收乘数：**增税会抑制消费和投资，乘数效应为  $\frac{-MPC}{1-MPC}$ ，体现税收对经济的逆向调节作用。

**2. 奥肯定律：**模拟 GDP 增长与就业率的关系：GDP 实际增长率超过潜在增长率（3%）时，就业率上升；反之则下降，公式为：

$$\text{就业变化率} = 0.5 \times (\text{GDP增长率} - \text{潜在增长率})$$

**3. 财政赤字与通胀关联：**过度支出可能导致需求拉动型通胀，公式为：

$$\text{通胀率} = 2\% + \max(0, \text{GDP增长率} - \text{潜在增长率} - 2\%) \times 0.5$$

**4. 社会福利指数模型：**综合教育、医疗、福利支出与税负影响，支出增加提升福利，税负加重降低福利。

## 二、游戏核心玩法与机制说明

### (一) 核心玩法

- 1.政策调整：**通过左侧滑块调节个人所得税（5%-80%）、公司税（5%-80%）、消费税（5%-50%）。通过右侧滑块调节教育支出（5-150 亿）、基建支出（5-150 亿）、医疗支出（5-150 亿）、社会福利（5-150 亿）。
- 2.执行政策：**点击“执行政策”按钮应用设置，系统自动计算政策对 GDP、赤字、就业率、福利指数和通胀的影响，并推进回合。
- 3.达成目标：**在 10 回合内使  $GDP \geq 2000$  亿且赤字  $\leq 500$  亿，或避免赤字超过 500 亿导致破产。

## （二）关键机制

### 1.税收收入计算

个人所得税：占 GDP 的 60% 部分  $\times$  税率

公司税：占 GDP 的 30% 部分  $\times$  税率

消费税：占 GDP 的 70% 消费部分  $\times$  税率  $\times 0.5$ （模拟实际征收效率）

### 2.财政赤字

赤字 = 总支出 - 税收收入

### 3.GDP 增长模型

新GDP = 原GDP  $\times (1 + \text{潜在增长率}) + \text{税收政策影响} + \text{支出政策影响}$

其中，政策影响通过乘数效应计算。

### 4.游戏结束条件

成功：回合结束后 GDP 达标且赤字可控。

失败：赤字超过 500 亿或达到最大回合数未达标。

## 三、游戏操作说明

**1.滑块调节：**拖动圆形手柄调整数值，实时显示当前设置。

### 2.按钮交互：

“执行政策”：应用当前设置，进入下一回合。

“重新开始”：重置游戏数据。

### 3.可视化提示：

指标颜色：绿色（健康）、红色（警告）。

趋势图：蓝色曲线为 GDP 实际值，黄色虚线为目标值。

## 四、代码结构与关键模块说明

### (一) 模块划分

模块	功能描述
Button	按钮交互组件，处理点击事件和视觉反馈。
Slider	滑块组件，支持数值调节和实时显示。
EconomySimulator	核心经济模拟逻辑，包含参数初始化、政策执行、状态计算和界面渲染。
主循环	处理游戏事件、更新界面状态、控制帧率。

### (二) 关键代码段解析

#### 1.经济模型计算 (execute\_policy 方法)

tax\_multiplier = -self.base\_mpc / (1 - self.base\_mpc)

spending\_multiplier = 1 / (1 - self.base\_mpc)

tax\_impact = sum(tax\_changes.values()) \* tax\_multiplier \* self.GDP

spending\_impact = sum(spending\_changes.values()) \* spending\_multiplier

乘数效应：通过税收乘数和支出乘数计算政策对 GDP 的影响。

奥肯定律与通胀：根据 GDP 增长调整就业率和通胀率。

#### 2.界面渲染 (draw 方法)

# 绘制 GDP 趋势点

for i, gdp in enumerate(self.history['GDP']):

```
x_pos = x + 20 + i * (width-40)/(len(...)-1)  
y_pos = y + height-40 - (gdp-min_gdp)/(max_gdp-min_gdp)*(height-60)  
pygame.draw.circle(surface, POSITIVE, (x_pos, y_pos), 4)
```

使用 Pygame 绘制矩形、线条和文本，实现数据可视化（如 GDP 趋势图、指标面板）。

### 3.事件处理（主循环）

```
for slider in game.tax_sliders + game.spending_sliders:  
    if slider.handle_event(event):  
        slider_updated = True
```

监听鼠标移动和点击事件，更新滑块和按钮状态。

## （三）外部库依赖

**pygame**：用于创建图形界面、处理用户输入、渲染 2D 图形和文本。

## 五、扩展与建议

- 1.新手策略**：初期可适当降低税率（如个人所得税 15%-20%）并增加基建支出（80-100 亿），利用乘数效应拉动 GDP 增长，同时监控赤字不超过 300 亿。
- 2.进阶玩法**：后期平衡福利支出与税收，避免高通胀（通胀率 > 5% 时需减少支出或增税）。
- 3.代码扩展**：可添加货币政策模块（如利率调节）、国际贸易影响或随机事件（如经济危机）以增强真实感。