

一、设计思路

我选择孟浩然的《春晓》作为主题，通过Python的海龟绘图库创作了一幅表现诗歌意境的绘画作品。《春晓》描述了春天早晨的情景，诗句简短而意境深远。

整体设计思路如下：

1. **整体布局**：将画布划分为四个区域，对应诗中的四句诗，每个区域表现一句诗的意境
2. **场景设计**：
 - 春眠不觉晓：表现春日清晨，人还在屋里睡觉，但太阳已经升起
 - 处处闻啼鸟：展示充满鸟鸣的春天树林
 - 夜来风雨声：描绘夜晚风雨交加的景象
 - 花落知多少：表现花瓣飘落的意境和思考

二、关键代码解释

整体框架设计

```
1 # 设置屏幕和画笔
2 screen = turtle.Screen()
3 screen.setup(1400, 1400) # 设置画布尺寸
4 screen.bgcolor("sky blue") # 背景颜色
5 screen.title("春晓 - 孟浩然") # 标题
6
7 # 创建画笔
8 t = turtle.Turtle()
9 t.speed(0) # 设置最快速度
10 t.hideturtle() # 隐藏画笔
11
12 # 字体设置
13 default_font = ("SimHei", 20, "normal") # 默认字体
14 title_font = ("SimHei", 30, "bold") # 标题字体
15
16 # 绘制网格线，用于分隔四个场景
17 def draw_grid():
18     # 代码略...
```

设置画布和绘图环境，创建了海龟对象，并定义了绘制网格线的函数，将画布分为四个区域

基础图形元素设计

```
1 # 绘制地面
2 def draw_ground(x_offset, y_offset, scale=1.0):
3     # 代码略...
4
5 # 绘制山
6 def draw_mountain(x, y, size, color):
7     # 代码略...
8
9 # 绘制树
```

```

10 def draw_tree(x, y, scale=1.0):
11     # 代码略...
12
13 # 绘制花瓣
14 def draw_petal(x, y, color, scale=1.0):
15     # 代码略...
16
17 # 绘制鸟
18 def draw_bird(x, y, scale=1.0):
19     # 代码略...

```

设计了一系列基础图形元素的绘制函数，这些函数可以在不同场景中复用，通过传入不同的参数实现灵活绘制

场景一：春眠不觉晓

```

1 def scene1():
2     """绘制左上角场景：春天的早晨"""
3     # 定义左上区域的边界
4     left_edge = -500
5     right_edge = 0
6     bottom_edge = 0
7     top_edge = 500
8
9     # 计算区域中心点
10    x_center = (left_edge + right_edge) / 2 # -250
11    y_center = (bottom_edge + top_edge) / 2 # 250
12
13    # 绘制背景、地面、山脉、太阳
14    # 绘制小屋、窗户、花朵和早起的小鸟
15    # 代码略...

```

这个场景描绘了春天早晨的宁静景象，包括简单的山峦、一座小屋、窗户和门、升起的太阳，周围点缀了各种花朵和几只早起的小鸟，寓意"春眠不觉晓"的意境

场景二：处处闻啼鸟

```

1 def scene2():
2     """绘制右上角场景：鸟鸣的春天"""
3     # 定义右上区域的边界
4     left_edge = 0
5     right_edge = 500
6     bottom_edge = 0
7     top_edge = 500
8
9     # 绘制背景、地面
10    # 随机分布树木，避免重叠
11    # 绘制花朵、草丛和飞翔的鸟
12    # 代码略...

```

这个场景展现了充满鸟鸣的春日树林。场景中有花鸟，象征着充满活力的春天，呼应"处处闻啼鸟"的诗句

场景三：夜来风雨声

```
1 def scene3():
2     """绘制左下角场景：夜晚的风雨"""
3     # 定义左下区域的边界
4     left_edge = -500
5     right_edge = 0
6     bottom_edge = -500
7     top_edge = 0
8
9     # 绘制深色背景、湿润的地面
10    # 绘制房子、明亮的窗户、月亮、云层
11    # 绘制闪电、被风吹弯的树枝、雨滴
12    # 代码略...
```

这个场景描绘了夜晚风雨交加的情景，通过深蓝色背景营造夜晚氛围，使用雨滴、闪电、弯曲的树枝表现风雨。小屋窗户透出的灯光与窗上流动的雨痕形成对比，加深了“夜来风雨声”的意境

场景四：花落知多少

```
1 def scene4():
2     """绘制右下角场景：落花的春天"""
3     # 定义右下区域的边界
4     left_edge = 0
5     right_edge = 500
6     bottom_edge = -500
7     top_edge = 0
8
9     # 绘制淡蓝色背景、绿色地面
10    # 绘制河流、石头、樱花树
11    # 绘制飘落的花瓣、石凳、小鸟和蝴蝶
12    # 代码略...
```

这个场景展现了花瓣飘落的春天，中心是一棵盛开的樱花树，周围和河流上漂浮着花瓣。石凳上的小鸟和散落的花瓣暗示有人刚刚离开，表现了“花落知多少”中对时光流逝和自然变化的思考

调用各个组件完成整体作品

```
1 # 绘制整体画面
2 write_poem_title() # 绘制标题
3 draw_grid()        # 绘制分隔线
4 scene1()            # 绘制场景1
5 scene2()            # 绘制场景2
6 scene3()            # 绘制场景3
7 scene4()            # 绘制场景4
8
9 # 保持画面显示
10 turtle.mainloop()  # 保持窗口打开
```

最后调用各个函数完成整体作品的绘制

三、效果截图



四、心得体会

通过这次《春晓》海龟绘图实验，我深入体会到了编程与艺术结合的魅力

- 1. **函数模块化设计：**通过定义基础图形元素函数（如绘制树、花、鸟等），实现了代码的高度复用，使得整体结构清晰，便于修改和扩展
- 2. **参数化设计的灵活性：**基础函数中加入scale等参数，实现了元素的灵活缩放，同一函数可以绘制不同大小的同类元素，增强了代码的灵活性

五、AI辅助说明

在本次实验中，我适度使用了AI辅助工具帮助完成部分编程工作，具体如下：

1. 使用的AI工具

- **GitHub Copilot + Claude3.7**: 用于代码补全和函数实现

2. AI辅助的主要环节

- **整体框架设计**: AI帮助构思了四个场景对应四句诗的布局方案
- **基础图形函数**: 通过Copilot辅助实现了 `draw_ground()`、`draw_mountain()` 等基础函数
- **算法实现**: 场景二中树木的无重叠分布算法、场景三的雨滴效果和场景四的花瓣飘落效果

3. 我的独立贡献

- 整体意境的设计和场景细节的构思
- 修改和优化AI生成的代码，提高运行效率
- 调整视觉效果，使其更符合诗歌意境
- 纠正代码中的逻辑错误，如树冠与树干对齐问题
- 测试调试和整体协调工作