

设计模式实验

一、实验目的

1. 学会选择合适的设计模式解决在软件开发中遇到的实际问题,加深对常用设计模式的理解和掌握。
2. 掌握一些常用的设计模式联用技巧,学会同时运用多种设计模式解决实际问题。

二、实验要求

1. 根据实例场景正确选择设计模式,理解常用设计模式的动机和意图。
2. 结合实例,绘制对应的模式结构图。

三、实验内容

1. 练习 1

在某 FPS (First - Person Shooting , 第一人称射击) 游戏中提供了多个不同的游戏场景。在每一个游戏场景中,提供了对应的地图 (Map)、天气 (Weather) 和游戏背景音乐 (Sound) 等。

请选择一种合适的设计模式对游戏场景进行设计,使得当用户选择游戏场景时,该场景所对应的地图、天气和背景音乐能够同时出现;此外,还可以方便地在该游戏中增加新的游戏场景。要求给出该设计模式的名称并结合实例绘制对应的结构图(即类图,类名、方法名和属性名可自行定义)。

2. 练习 2

在某 FPS 游戏中，系统可以给所有游戏成员发送通知，例如提示任务执行完毕、发送新的任务提醒、发出敌人袭击警报等。

请选择一种合适的设计模式设计该系统通知模块，使得在系统中可以灵活地增加或删除游戏成员。要求给出该设计模式的名称并结合实例绘制对应的结构图（即类图，类名、方法名和属性名可自行定义）。

3. 练习 3

某 FPS 游戏中提供了一个游戏管理器（Game Manager），通过该管理器用户可以对音效（Sound Effect）、场景（Scene）、游戏角色（Role）等对象进行参数设置。为了节约系统资源并且保证对象状态的一致性，在游戏运行时，用户只能打开唯一的一个管理器界面。

根据以上描述，请选择两种合适的设计模式设计该游戏管理器，在实现对多个对象进行统一设置的同时保证游戏管理器的唯一性。

要求给出这两种设计模式的名称并结合实例绘制对应的结构图（即类图，类名、方法名和属性名可自行定义）。

4. 练习 4

为了让游戏场景呈现更加逼真的效果，在某 FPS 游戏中可以对场景（Scene）的光照效果等进行渲染（Rendering）。考虑到系统的可扩展性，开发人员可以实现表面渲染（Surface Rendering）和体渲染（Volume Rendering）等算法，也可以调用一些已有的渲染引擎（Render Engine）中的渲染算法。在设计时需要考虑渲染算法的可复用性，并能

够灵活地更换和增加新的渲染效果。

根据以上描述，请选择两种合适的设计模式设计该场景渲染模块，一方面证可以方便地调用已有的渲染算法，另一方面还可以灵活地嵌入新的算法。

要求给出这两种设计模式的名称并结合实例绘制对应的结构图（即类图，类名、方法名和属性名可自行定义）。

5. 练习 5

In a First - Person Shooting (FPS) Game , a building or a blindage (掩体) is a 3D structure that consists of many 3D Objects such as Cube (立方体), Cylinder (圆柱体), Pyramid (锥体) etc . When we fill a 3D block with color (such as Gray), the same color also gets applied to the Objects in the block . Here a 3D block is made up of different parts and they all have same operations . The parts of a 3D block can be small blocks .

Which design pattern can be used to implement the 3D structure ? Give the pattern's name and draw its structure diagram with this sample .

四、实验步骤

练习 1：结合实例，选择一种合适的设计模式并结合实例绘制对应的结构图。

练习 2：结合实例，选择一种合适的设计模式并结合实例绘制对应的结构图。

练习 3：结合实例，选择两种合适的设计模式并结合实例绘制对应的结构图。

练习 4：结合实例，选择两种合适的设计模式并结合实例绘制对应的结构图。

练习 5：结合实例，选择一种合适的设计模式并结合实例绘制对应的结构图。

五、实验结果

练习 1 ~ 5：需要分别给出所选设计模式的名称并结合实例绘制对应的结构图。

六、实验小结

请总结本次实验的体会，包括学会了什么、遇到哪些问题、如何解决这些问题以及存在哪些有待改进的地方。