



实验1 创建型模式2

1 实验目的

学会用UML设计类图，熟练掌握基于原型模式、单例模式的设计思维。

2 实验环境

开发环境：JDK 8.0（或更高版本，高版本要下载独立JavaFX）

开发工具：Eclipse

设计工具：StarUML（或PlantUML）

3 基础知识

3.1 原型模式

原型模式通过复制快速创建对象，具体知识参考讲义。

3.2 单例模式

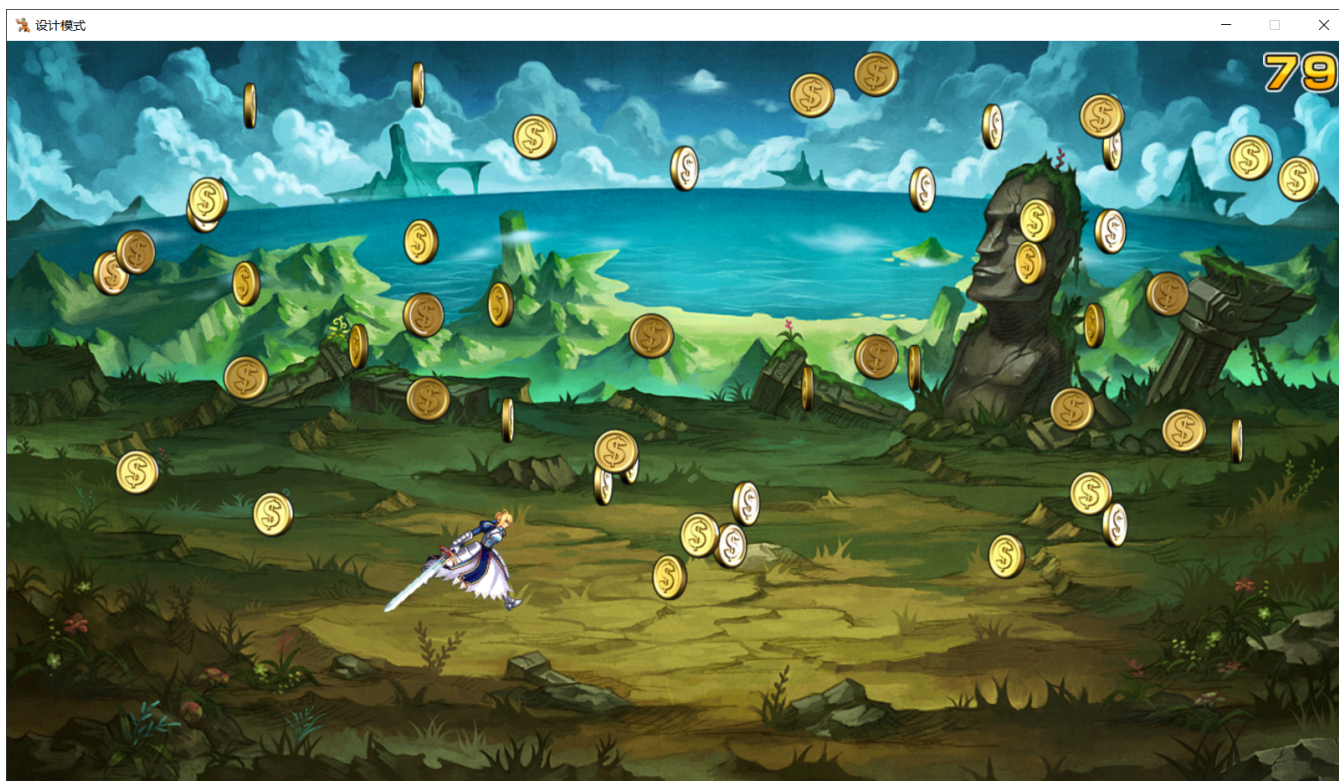
单例模式使系统运行时目标类只被实例化一次，并提供全局访问，具体知识参考讲义。

4 实验内容

实验1 接金币游戏

问题描述

学习设计实现一个接金币的小游戏，效果如下图。功能要求：（1）通过左右键（或自定义方向键）控制角色左右移动，移动范围不超出地图边界。（2）随机刷新金币，金币向地面掉落，接触地面时消失。（3）金币与角色发生碰撞时消失，金币积分加1，并触发金币音效。（4）计分板实时更新分数。（5）游戏进行时实时播放背景音乐。



提示与解析:

- 1、分析得到程序涉及的对象：玩家角色（`Player`）、金币（`Coin`）、数字（`Number`）、地图（`Map`）。这些实体类共同的特征包括当前外观（`currentLook`）、位置（`position`）和空间大小（`width` 和 `height`），将公共特征提取出来形成实体类（`Entity`）。实体类定义一个相交判断函数（`isOverlap()`）为碰撞检测提供支持。
- 2、玩家和金币两类对象在移动过程中会变化外观，这里设计一个动态图类（`DynamicImage`）来承载动态变化的外观所需要的图片。
- 3、程序主进程执行一个游戏循环（`GameLoopTimer`），每一帧执行流程分为输入处理（`inputProcess()`）、逻辑更新（`logicUpdate()`）和显示更新（`displayUpdate()`）三个步骤。
- 4、输入处理部分的代码主要封装在游戏控制器（`GameController`）中，包括输入信息预处理和调用游戏实体运动命令；游戏资源管理以及大部分逻辑运算封装在一个游戏管理器（`GameManager`）中，包括数据初始化（`initialize()`）、刷新金币（`refresh()`）、实体状态更新（`move()`）、得分判断（`judge()`）等，游戏界面更新代码封装在一个渲染器（`Render`）中，主要为渲染功能（`rendering()`）。

具体要求:

- 1、调试样例程序，使其正常运行；

- 2、哪些类采用原型模式设计，具体是如何实现的，深浅复制是如何处理的？结合程序功能解析设计意图和作用。（给出类图、实现代码截图以及必要的说明）
- 3、哪些类采用单例模式设计，采用什么方式实现？结合程序功能解析设计意图和作用。（给出类图、实现代码截图以及必要的说明）
- 4、分析程序，角色和金币的碰撞检测是如何做到的？
- 5、尝试修改金币的生产规则，观察效果。
- 6、程序还存在哪些可以改进的？给出一些思路。

5 实验要求

5.2 实验评价

- 1、完成实验内容（60%）
- 2、对实验思路进行阐述（20%）
- 3、对实验过程进行总结（20%）

5.1 实验报告

- 1、根据要求完成实验内容、思路阐述和总结。
- 2、截取清晰的核心代码、设计图和效果到报告中；