

🗨 图形渲染系统用户手册

1. 系统简介

图形渲染系统是一个基于Java的应用程序，用于创建、管理和渲染基本图形元素。该系统支持多种渲染方式，包括SVG、控制台文本渲染以及通过代理的远程渲染。

2. 系统功能

系统提供以下主要功能：

- 创建基本图形元素（圆形、矩形、线段）
- 在画布上添加、移动和删除图形
- 撤销和重做操作
- 切换不同的渲染引擎
- 导出图形为JSON或XML格式
- 远程渲染服务

3. 安装说明

📦 系统要求

- Java 11或更高版本
- Maven 3.6或更高版本

📦 安装步骤

1. 克隆或下载项目代码
2. 在项目根目录执行：`mvn clean install`
3. 生成可执行JAR文件：`mvn package`

4. 使用指南

📦 启动系统

```
1 | java -jar target/graphic-rendering-1.0-SNAPSHOT-jar-with-dependencies.jar
```

📖 基本操作

系统启动后，将展示一个演示程序，依次展示各种设计模式的应用。

5. API参考

📖 主要类

- `Drawing` : 管理图形和渲染的主类
- `ShapeFactory` : 创建图形对象的工厂
- `RendererFactory` : 创建渲染器的抽象工厂
- `CommandManager` : 管理命令执行、撤销和重做

📖 示例代码

```
1 // 创建渲染器
2 RendererFactory factory = new SvgRendererFactory();
3 Renderer renderer = factory.createRenderer(800, 600);
4
5 // 创建绘图
6 Drawing drawing = new Drawing(renderer);
7
8 // 创建并添加图形
9 ShapeFactory shapeFactory = new ShapeFactory();
10 Shape circle = shapeFactory.createCircle(100, 100, 50);
11 drawing.addShape(circle);
12
13 // 渲染
14 drawing.render();
```

6. 设计模式应用

本系统应用了以下设计模式：

1. 创建型模式

- Factory: `ShapeFactory`创建具体图形
- Abstract Factory: `RendererFactory`创建不同渲染引擎
- Singleton: `RenderingConfig`提供全局配置

2. 结构型模式

- Adapter: `ThirdPartyRendererAdapter`适配第三方渲染库
- Bridge: `Renderer`接口与实现分离，支持多种渲染方式
- Proxy: `RemoteRendererProxy`提供远程渲染代理

3. 行为型模式

- Command: `AddShapeCommand`等命令类实现操作和撤销
- Visitor: `JsonExportVisitor`等访问器导出不同格式

○ Observer: ShapeObserver监听模型变化

7. 故障排除

常见问题

○ **问题** : 无法启动程序

解决方案 : 确认Java版本是否为11或更高

○ **问题** : 远程渲染连接失败

解决方案 : 检查网络连接和远程服务是否可用

8. 联系方式

mengxizhuo@gmail.com