

数据保存与恢复

目录

- [编写目的](#)
- [结构设计](#)
 - [对话数据结构](#)
 - [背包道具和成就数据结构](#)
 - [地图数据结构](#)
- [运用设计](#)
 - [对话运用](#)
 - [机关运用](#)
 - [背包运用](#)
 - [地图运用](#)

编写目的

为了让参与本项目的设计人员、开发人员、测试人员、管理人员等了解本系统的数据储存与恢复的设计思路与整体构架，特编写此文档。

结构设计

用Dictionary<String, Object>序列化保存数据：主角状态、NPC状态等等，其中String是key，Object是任意类型的，比如状态枚举，道具列表、成就列表。

对话数据结构

与NPC对话最多有四个状态，分别是beforeMission(与NPC第一次对话时的内容，也就是准备进入章节剧情)，onMission（剧情进行中但没有完成剧情时的内容），afterMission（剧情完成后的对话），other(当主角靠近NPC一定距离时，NPC会自言自语地讲话)。

```
[
  {
    "name" : "村民A",
    "beforeMission" : ["", ""],
    "onMission" : ["", ""],
    "afterMission" : ["", ""],
    "other" : ["", ""]
  },
  {
    "name" : "村民B",
    "beforeMission" : ["", ""],
    "afterMission" : ["", ""],
    "other" : ["", ""]
  }
]
```

背包道具和成就数据结构

itemName和achieveName分别表示道具和成就，用JSON存储名称和详细描述

```

{} item.json > {} achieveName
1  {
2    "itemName" : {"name" : "", "decription" : ""},
3    "achieveName" : {"name" : "", "decription" : ""}
4  }

```

地图数据结构

用JSON存储区域地图的名字，比如会在界面上显示地图名字，然后渐变消失。

```

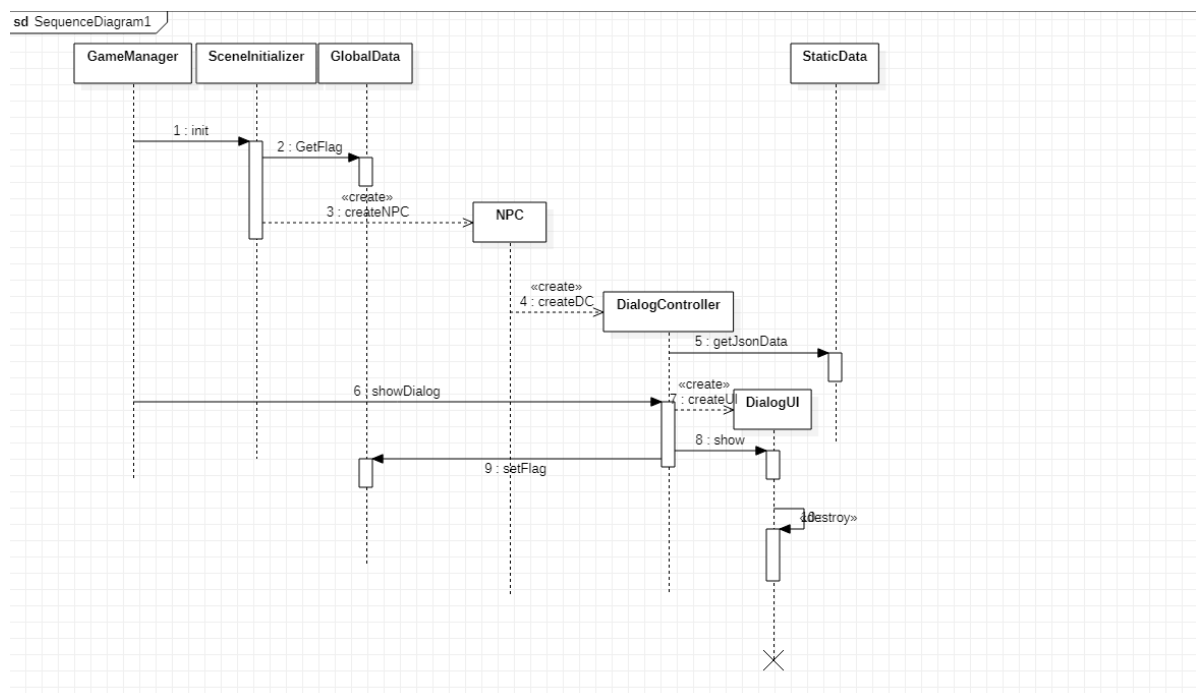
{} map.json > ...
1  {
2    "region" : "name"
3  }

```

运用设计

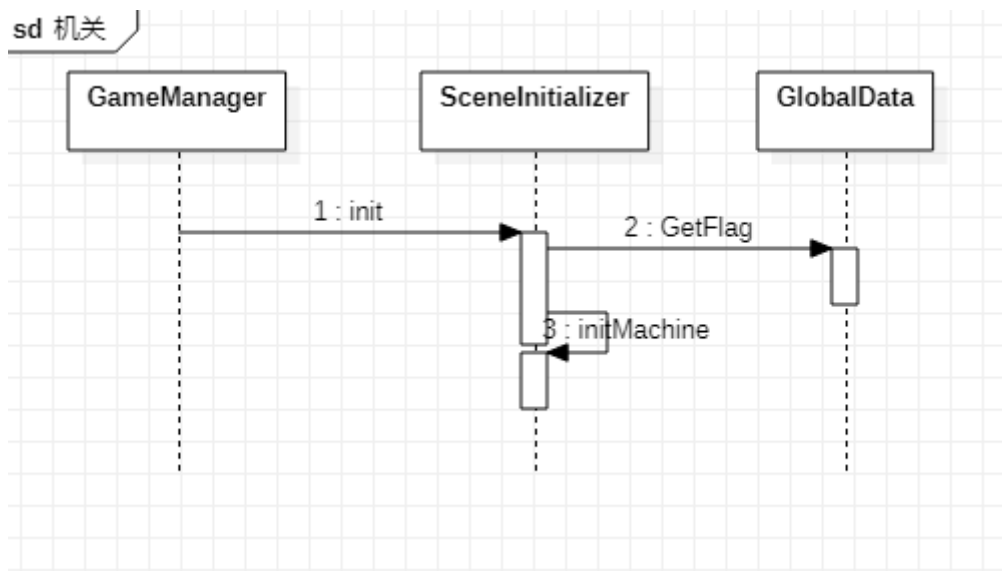
对话运用

场景初始化器从全局数据获取NPC的状态，初始化并传递给NPC，NPC根据状态初始化对话控制器，对话控制器从静态数据获取相应的对话数据，需要对话时，游戏管理器调用对话控制器的showDialog，由对话控制器创建并显示对话，最后保存状态。



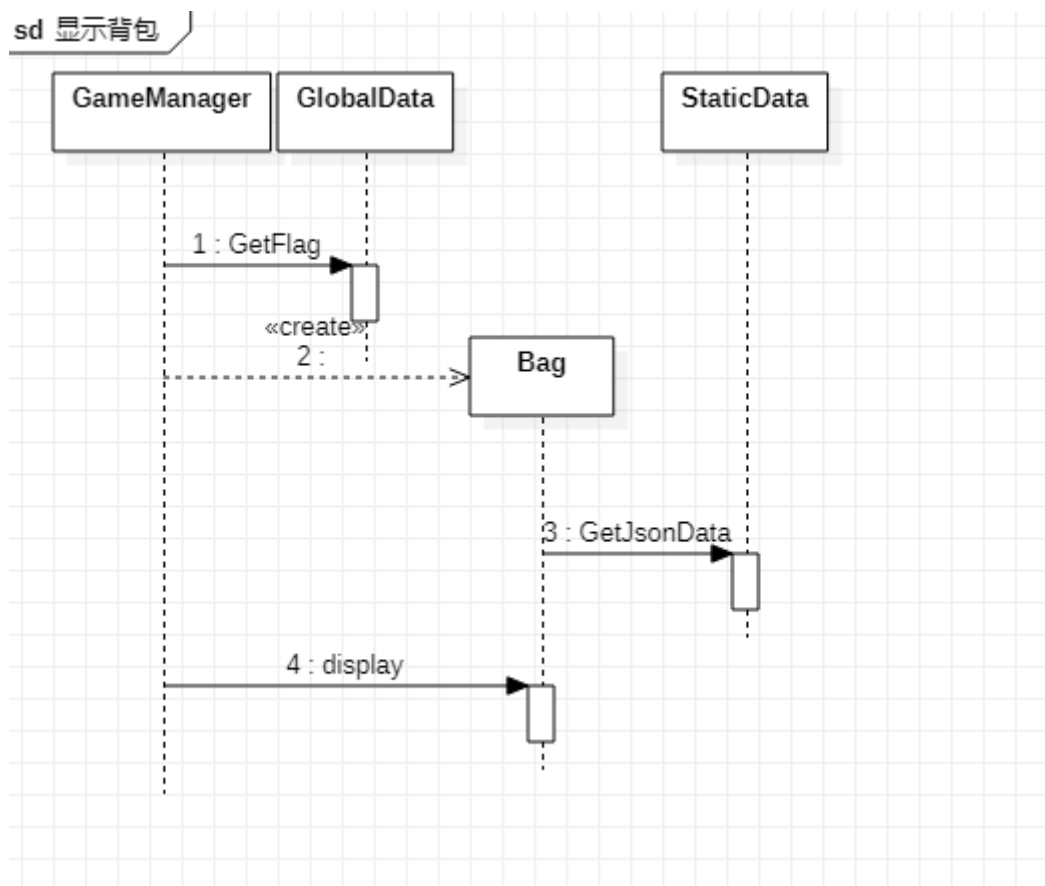
机关运用

场景初始化器从全局数据获取机关的状态，设置机关。



背包运用

场景初始化器从全局数据获取主角拥有的道具列表，创建背包。背包从静态数据获取道具说明，在需要时显示背包。



地图运用

地图的状态不同于地图的查看，地图的设置与机关类似，同样是从全局数据获取地图的状态，设置隐藏的解谜地图、需要道具解锁的地图。

