|  |
| --- |
| SPRABBIT |
| SPENGINE 系统设计文档 Version 0.80 |
| System Design of SPENGINE Version 0.80 |

|  |
| --- |
| KenJ  6/22/2012 |

**修订历史 History**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 Date | 版本 Version | 说明 Description | 作者 Author |
| 2012/6/23 | 0.7 | 初步设计 | 邓楚健 |
| 2012/6/26 | 0.71 | 补充设计图 | 钟晓辉 |
| 2012/6/26 | 0.71 | 补充组件设计 | 解寒宇 |
| 2012/7/1 | 0.80 | 更新设计类图 | 叶泽飞 |

目录Contents

[1. 简介Introduction 2](#_Toc328314638)

[1.1. 目的Purpose 2](#_Toc328314639)

[1.2. 范围Scope 2](#_Toc328314640)

[1.3. 引用References 2](#_Toc328314641)

[1.4. 概述Overview 2](#_Toc328314642)

[2. 设计要素Design Considerations 3](#_Toc328314643)

[2.1. 假设Assumptions 3](#_Toc328314644)

[2.2. 约束Constraints 3](#_Toc328314645)

[2.3. 系统环境System Environment 3](#_Toc328314646)

[2.4. 设计方法Design Methodology 3](#_Toc328314647)

[3. 架构Architecture 3](#_Toc328314648)

[3.1. 系统设计System Design 4](#_Toc328314649)

[3.2. 系统分解System Decomposition 4](#_Toc328314650)

[4. 数据设计Data design 4](#_Toc328314651)

[4.1. 数据描述Data Description 4](#_Toc328314652)

[4.2. 数据字典Data Dictionary 4](#_Toc328314653)

[5. 组件设计Component Design 4](#_Toc328314654)

[6. 软件接口设计Software Interface Design 4](#_Toc328314655)

[6.1. 用户界面设计User Interface Design 4](#_Toc328314656)

[6.2. 模块接口设计Module interface design 4](#_Toc328314657)

[7. 类图Class Diagrams 5](#_Toc328314658)

[8. 附录Appendices 5](#_Toc328314659)

## 简介Introduction

#### 实施本项目是为了创建一个方便易用的游戏引擎。编程人员可以在其项目中引入本引擎的类库，简化游戏编程，一般用户可以使用本引擎自带的脚本，直接对引擎进行操作，从而方便地制作游戏。

#### 本系统设计说明书展示了本项目所使用或将应用的设计。所描述的设计遵循系统需求说明书（SRS）中所规定的需求。

### 目的Purpose

#### 本文档的目的在于描述关于SPENGINE的详细设计。

#### 本文档应根据SRS的改动而补充。

### 范围Scope

#### 本文档给出了SPENGINE的详细软件架构。

#### 本文档规定了SRS中提到的一些模块的结构和设计。

#### 本文档展示了SRS中的一些用例的活动图与时序图。

#### 类图展示了编程人员应如何实现相应的模块。

### 引用References

#### 本文档的用户应参考以下文档

#### IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

#### SPENGINE系统需求说明书 version 0.7。

### 概述Overview

#### 本文档根据“IEEE Recommended Practice for Software Requirements Specifications”中的标准而制定。

#### 第3-5部分包括关于设计方面的讨论与图片，第6部分展示了系统的用户界面，第7部分包括了相关类图。

#### 附录描述了部署方法，以实现的功能，将要实现的功能，以及项目所使用的外部运行环境以及工具。附录还将包括测试报告以及用例。

## 设计要素Design Considerations

### 假设Assumptions

#### 本系统的高级用户熟知C++软件编程，了解调用库函数的方法。

#### 本系统的入门用户了解一般文本编辑软件，并对游戏有一定的了解。

#### 用户理解用于引擎操作的基本术语。

### 约束Constraints

#### 本引擎使用C++编写，执行于DirectX9.0c运行环境。

#### 本引擎使用的脚本语言为KScript（暂定），其具体设计属于本文档的规定范围。

### 系统环境System Environment

#### 本引擎只能在微软的Windows XP以及以后的Windows平台下运行。

#### 运行本引擎需要安装DirectX 9.0c 运行环境。

### 设计方法Design Methodology

#### 本系统设计具备灵活性，以便适应变更的需求。

#### 系统被划分为可控制的多个过程，这些过程可以被组织为子模块以及抽象模块。

## 架构Architecture

### 系统设计System Design

#### 系统设计框图。

#### 

### 系统分解System Decomposition

#### 功能分解树。

#### 

## 数据设计Data design

### 数据描述Data Description

#### 利用XML实现基本配置数据的持久化，使用二进制文件对状态进行保存。

### 数据字典Data Dictionary

#### 具体数据因游戏的类型不同而不能统一。

## 组件设计Component Design

### 组件管理器Component Manager

#### 组件管理器用于管理现有的游戏组件，包括对组件进行初始化、更新、绘制、读取和释放。

### 屏幕管理器Screen Manager

#### 组件管理器用于管理现有的屏幕，包括对屏幕进行初始化、更新、绘制、读取和释放。

### 摄像机 Camera

#### 摄像机用于更新摄像机位置，调整投影参数。

### 光线管理器 Light Manager

#### 光线管理器用于储存光源并进行开关、更新光源位置以及设置相关系数。

### 材质管理器 Material Manager

#### 材质管理器用于生成材质。

### 精灵管理器 Sprite Manager

#### 精灵管理器用于储存并管理用Texture生成的精灵，对其进行创建、绘制、删除等操作。

### 音乐管理器 Sound Manager

#### 音乐管理器用于储存并分轨需要播放的音乐，对音乐播放状态进行实时更新以及控制。

### 视频管理器 Video Manager

#### 视频管理器继承于精灵管理器，用于更新管理控制视频信息，并对实时重绘到Texture上的视频进行绘制。

## 软件接口设计Software Interface Design

### 用户界面设计User Interface Design

#### 具体实现游戏的用户界面由用户自行决定编写。

#### 本脚本引擎并未打算实现图形用户界面。

#### 用户通过脚本文件控制引擎行为，交互语言为KScript（待定）。

### 模块接口设计Module interface design

#### 模块设计参考MVC架构，并针对视频游戏进行改进，即使用逻辑与显示层分离的设计方法。

## 类图Class Diagrams

### 总体设计

#### 

### SPGraphic

#### 

### SPInput

#### 

### SPSound

#### 

### SPVideo

#### 

### SPScript

#### 

### SPHelper

#### 

### SPFile

#### 

### SPGUI

#### 

### SPScreen

#### 

## 附录Appendices

#### 暂无