**虚拟组装编程规范**

Issue Date – 2017/09/14

|  |  |
| --- | --- |
| **DOCUMENT ID:** | |
| **PROJECT: 虚拟装配** | **RELEASE: 1.0.0** |
| **FEATURE:** | **SUBSYSTEM:** |
| **DISTRIBUTE TO:** | |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Author** | **Reviewed By** | **A. Description Of Change**  **B. Summary of Review** | **Issued by / Date** |
| v1.0.0 | 张琦 |  | 创建文档 | 2017/09/14 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 4](#_Toc493254254)

[1.1 概述 4](#_Toc493254255)

[1.2 编写目的 4](#_Toc493254256)

[1.3 关键术语 4](#_Toc493254257)

[2 需求 4](#_Toc493254258)

[2.1 虚拟组装开发环境 4](#_Toc493254259)

[3 虚拟组装编码规范 4](#_Toc493254260)

[3.1 注释 4](#_Toc493254261)

[3.2 命名规则 4](#_Toc493254262)

[3.3 文件格式 5](#_Toc493254263)

[3.4 初始化 5](#_Toc493254264)

[3.5 UI布局 5](#_Toc493254265)

[3.6 循环 5](#_Toc493254266)

[3.7 单例 5](#_Toc493254267)

[3.8 命名空间 6](#_Toc493254268)

[4 相关文档 6](#_Toc493254269)

[5 相关文献 6](#_Toc493254270)

[6 附件 6](#_Toc493254271)

1. **概述**
   1. **概述**

虚拟组装的编程规范。

* 1. **编写目的**

本文档规定开发的编码规范，为开发提供参考，作为团队开发的标准，提高代码的可阅读性和可维护性。

**预期读者**

项目开发团队成员

* 1. **关键术语**

|  |  |
| --- | --- |
| **缩写** | **描述** |
| Holocaust | MR混合现实智能眼镜 |
| 编程规范 | 项目代码规范 |

1. **需求**
   1. **虚拟组装开发环境**

通过本文档，能够统一编码规范，作为团队开发标准。

1. **虚拟组装编码规范**
   1. **注释**
      1. 单行注释：“//”，注释单行代码
      2. 多行注释：“/\* \*/”
      3. 文档注释：“///” 注释方法或者类
   2. **命名规则**
      1. 类命名：见名知意，单词首字母大写
      2. 方法命名：见名知意，单词首字母大写
      3. 属性命名：见名知意，以“\_”开头，或者字母开头，单词首字母大小
      4. 局部变量命名：字母开头，首字母小写，后面的单词首字母大写
   3. **文件格式**
      1. “.cs”文件：程序脚本文件，是Hololens程序的逻辑代码部分
      2. “.txt”文件，临时数据存储文件
      3. “.unity”文件，unity场景文件
      4. “.anim”文件，unity动画文件
      5. “.FBX”文件，模型文件
      6. “.dll”文件，底层库文件
      7. “.prefab”文件，unity预制物体文件
      8. “.mat”文件，材质文件
      9. “.shader”文件，着色器文件
   4. **初始化**
      1. 所有变量必须先初始化在使用，尽量避免在unity中拖拽物体赋值，保持代码的可维护性
      2. 脚本的执行顺序都使用代码控制，尽量不要使用Script Execution Order，保持代码可维护性和阅读性
      3. 初始化过程，避免过多计算，保持代码性能
   5. **UI布局**
      1. 采用自动布局，增加代码的可维护性
      2. UI部分减少计算，减少耦合性
   6. **循环**
      1. 尽量避免循环new变量
      2. 尽量避免3层以上的循环
   7. **单例**
      1. 避免C#的单例模式，因为继承了MonoBehaviour类，无法new出对象，尽量继承Hololens单例类，实现单例模式
      2. 注意单例类的初始化，防止空引用异常
   8. **命名空间**
      1. unity并不具备命名空间的概念，所以为了代码的可扩展性，自己手动添加命名空间
      2. 注意命名空间的范围，防止不同命名空间，相同名字的类的调用造成的错误
2. **相关文档**

无

1. **相关文献**

《C#从入门到精通》

《C#实战经典》

1. **附件**

无