## 3.5.模块五

### 3.5.1.模块编号

人物控制逻辑设计

编号3.5

### 3.5.2.功能描述

本模块主要负责玩家互动模式设计，使玩家可以操控场景内的3D模型进行移动，攻击等动作。此模块所需要的数据部分由游戏主体系统即时生成，部分由后端服务系统提供。

### 3.5.3.输入信息

给出对每一个输入参数的特性，包括名称、标识、数据的类型和格式、数据值的有效范围、输入的方式。 数量和频度、输入介质、输入数据的来源和安全保密条件, 输入时代码表与基本表的情况,使用的特殊输入设备情况等等

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 特性 | 名称 | 标识 | 数据类型 | 格式 |
| 短字符 | W | 前进键 | 字符型 | 字符型 |
| 短字符 | S | 后退键 | 字符型 | 字符型 |
| 短字符 | A | 左移键 | 字符型 | 字符型 |
| 短字符 | D | 右移键 | 字符型 | 字符型 |
| 下跳沿 | LClick | 攻击键 | 八位二进制数 | 八位二进制数 |
| 下跳沿 | RClick | 导弹键 | 八位二进制数 | 八位二进制数 |

### 3.5.4.输出信息

给出对每一个输出参数的特性，包括名称、标识、数据的类型和格式，数据值的有效范围，输出的形式、 数量和频度，输出介质、对输出图形及符号的说明、安全保密条件、输出时代码表与基本表的情况等等

|  |  |  |  |
| --- | --- | --- | --- |
| 特性 | 名称 | 数据类型 | 格式 |
| 短字符 | 导弹计时 | 布尔型 | 布尔型 |
| 短字符 | 人物死亡 | 布尔型 | 布尔型 |
| 短字符 | 人物状态 | 字符串 | 字符串 |

### 3.5.5.算法

本模块使用C#中UnityEngine Namespace中的Input类实现对玩家键盘，鼠标操作的读取工作，其中具体应用到的方法有：

**（1）键盘的输入**

**Input•GetKey，Input.GetKeyDown，Input.GetKeyUp三个方法分别获取用户键盘按键的输入**

**GetKey:用户长按按键有效；**

**GetKeyDown：用户按下按键时有效；**

**GetKeyUp：用户抬起按键时有效；**

**Input.anyKey，Input.anyKeyDown**

**anyKey用于监听用户的任意按键输入或鼠标点击，事件触发后返回true。anyKeyDown：当任意按键松开时返回true。**

**Input.GetButton(string buttonName)**

**GetButton用于监听Button的输入，参数为自己指定的button的名字，在Unity中可以使用管理器添加或修改Input参数。**

**Input.GetButtonDown（ 当某个按钮被按下的那一帧返回true）**

**Input.GetButtonUp（ 当某个按钮弹起时返回true）**

**（2）鼠标的输入**

**Input.•mousePosition**

**Input.mousePosition返回当前鼠标的位置，这里指的是距离原点的像素位置，说明一下，Unity中的原点（0，0）位置为左下角，上位y轴正方向，右为x轴正方向。返回值类型为Vector3。**

**Input.•GetMouseButton**

**GetMouseButton 对应的键处于按下状态时返回true**

**GetMouseButtonDown 对应的键被按下时返回true**

**GetMouseButtonUp 对应的键弹起时返回true**

### 3.5.6.处理流程

本模块包含6个内置类，分别为PlayerSkill，PlayerShooting，PlayerMovement，PlayerInput，PlayerHealth，OtherPlayerControl，都继承UnityEngine namespace内置脚本MonoBehavior，其中PlayerInput类关联PlayerMovement，PlayerMovement基于PlayerInput类检测的玩家输入改变玩家的状态，给予玩家攻击动作，Playerhealth类控制玩家是否结束游戏，不同状态下的动作。Playshooting基于PlayerInput的读取，控制玩家的攻击动作，OtherPlayerControl主要控制玩家的其他操作，如陷阱，导弹冷却等。

### 3.5.7.类设计

给出本模块的类设计，包括类图和类说明

对于J2EE应用，可以分控制类（例如用到的Servlet）、实体类（例如DAO）、业务类（例如处理业务的Handler）、视图类（例如JSP）、接口类（例如供别的模块调用的API）、工具类（例如对字符串进行处理的StringUtil）进行描述。JSP可以放在视图类中进行描述，描述包括使用到的重要的JavaScript。

#### 3.1.7.1.类图

