## 游戏简介

mini-Z是一款第三人称3D动作游戏，主要模仿绝区零的设计，以格挡为核心，鼓励进攻。玩家需要通过格挡，切人等操作始终掌握主动权，并通过不同的角色，技能，连招击败敌人。

## 具体设计

#### 角色

###### 概述：

上场两个角色,一个轻武器一个重武器，通常一个角色在前台行动一个角色在后台等待切人/格挡。动作有优先级，优先级高的动作可以打断优先级低的动作。攻击可以削减敌人的血量与韧性，韧性清空后敌人出现硬直并恢复韧性，血量清空后敌人死亡。

###### 详情：

基本动作分为

|  |  |  |
| --- | --- | --- |
| 行为 | 按键 | 优先级 |
| 待机 | 无 | 0 |
| 移动 | wasd | 1 |
| 普通攻击 | 左键 | 2 |
| 切人/格挡 | 右键 | 5 |
| 小技能 | E | 3 |
| 大技能 | Q | 4 |
| 闪避 | Shift | 5 |
| 受击 | 无 | 5 |

派生动作分为

|  |  |  |
| --- | --- | --- |
| 动作 | 触发条件 | 优先级 |
| 闪避攻击 | 闪避后攻击 | 5 |
| 格挡反击 | 格挡成功后攻击 | 5 |

动作拆分：

|  |  |
| --- | --- |
| 节点名 | 说明 |
| 动作开始帧 | 动作开始，开始播放动画 |
| 伤害判定帧 | 动作的这几帧进行攻击的判定 |
| 连招判定帧 | 在这帧后可以衔接下个连招 |
| 动作后摇帧 | 在这帧后可被任意动作打断 |
| 动作结束帧 | 动作结束，回到默认状态 |

软锁定：

当玩家朝向怪物，距离怪物一定范围内，进行攻击动作且没有主动偏离怪物方向的操作时，对角色的方向进行细微修正，使得玩家攻击能够持续的命中敌人。

格挡：

如果在敌人即将攻击时切换角色，新上场的角色将进行格挡。如果格挡成功，则能进行格挡反击。

多人在场：

在角色进行部分动作时切人，当前角色不会立即消失而是继续完成当前动作以达成多个角色同时在场的效果。

实时修改：

可在Inspector窗口实时修改角色参数并查看效果，便于参数的调整。

###### 具体实现：

移动：

通过character controller与root motion控制角色。角色的奔跑通过root motion及控制动画播放速度来控制，以便使角色的动作与位移匹配。攻击位移则结合root motion 与character controller的主动移动来实现，以便能自由调整攻击位移长度。

旋转：

通过smoothDampAngle函数来实现角色方向的平滑变化及自由修改角色方向修正的时间。

动作切换：

通过状态机控制角色的不同状态。

受击：

获取受击的方向，通过动画状态机的blendtree获得混合的受击动画并利用协程与character controller进行受击位移

格挡：

检测格挡范围内有无进行攻击的敌人，若检测到则计算对应的格挡位置与方向，切换角色并进入格挡状态。

多人在场：

切人时检测当前状态，通过延时标记在动作完成时再切换到后台状态。

打击感：

在攻击命中敌人时，通过控制动画播放速度实现打击停顿，通过cinemachine进行镜头震动。

在格挡成功时，通过修改全局timescale实现停顿，通过cinemachine实现镜头震动。

软锁定：

进行攻击时，通过当前wasd输入，角色与敌人距离，角色朝向与敌人方向的偏差判断是否修正方向。通过smoothDampAngle实现平滑的方向修正。

属性控制：

通过scriptable object实现玩家属性的实时修改，查看，保存，便于为角色动作配置合适的参数。

###### 代码结构：

类：

|  |  |
| --- | --- |
| 名称 | 说明 |
| Creature | 玩家角色，敌人的共同基类 |
| Player | 角色基类 |
| Doll / Big | 两个具体角色的控制类 |
| PlayerStateMachine | 状态机基类 |
| DollStateMachine/PlayerStateMachine | 具体角色的状态机类 |
| PlayerState | 角色状态的基类 |
| [Doll,Big]State\_attack | 具体角色攻击状态类 |
| [Doll,Big]State\_backEnd | 具体角色后台状态类 |
| [Doll,Big]State\_beAttacked | ~~~~受击~~~ |
| [Doll,Big]State\_dash | 冲刺 |
| [Doll,Big]State\_idle | 待机 |
| [Doll,Big]State\_parry | 格挡 |
| [Doll,Big]State\_run | 奔跑 |
| [Doll,Big]State\_skill\_E | 小技能 |
| [Doll,Big]State\_skill\_Q | 大技能 |
| PlayerWeapon | 控制角色的武器及攻击判定的类 |
| Team | 控制队伍里两个角色的初始化，角色切换，角色属性的实时同步 |
| [Doll,Big]AttrubuteData | 储存角色的动作参数并实现实时同步 |

重要方法：

|  |  |
| --- | --- |
|  |  |
| 以Animator\_开头 | 附在动画关键帧上的方法,一般用于控制角色在动画播放到关键帧时进行相应的行动 |
| 以Animation\_开头 | 一般由角色状态类调用,控制状态切换时动画的播放,也可用于连续攻击的触发. |
| 以Beg开头 | 一般由InputBuffer传回的命令类调用,是某个命令的具体实现 |
| 含有Beg/End/Prepared/free | 与动画相应状态有关,分别为状态开始,结束,连招检测开始,可被任意动作打断 |
| Appear | 控制角色登场的位置,朝向,状态 |

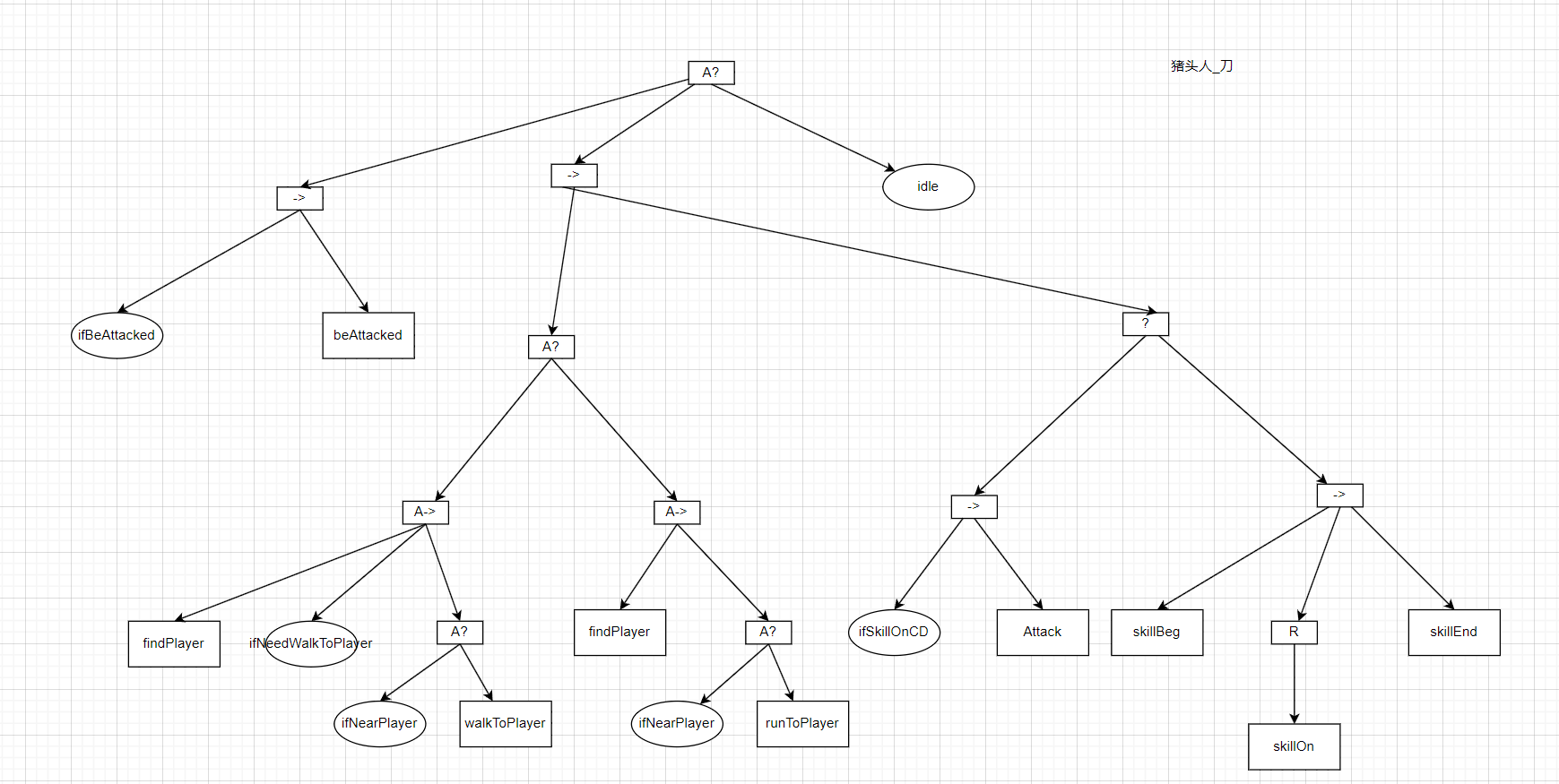
#### 怪物

一共两种怪物,一个近战猪头兵一个远程猪头兵。敌人未发现角色时待机，发现角色且距离较远时缓慢移向角色，距离到特定值时快速跑向敌人并在角色进入攻击范围后攻击。近战兵有普通攻击与连击两种攻击方法，都可以触发角色的格挡。远程兵有一种攻击方式，无法触发格挡。怪物有血量与韧性，韧性归零后进入受击硬直并恢复韧性，血量归零后死亡。当场景中全部怪物死亡后游戏胜利。

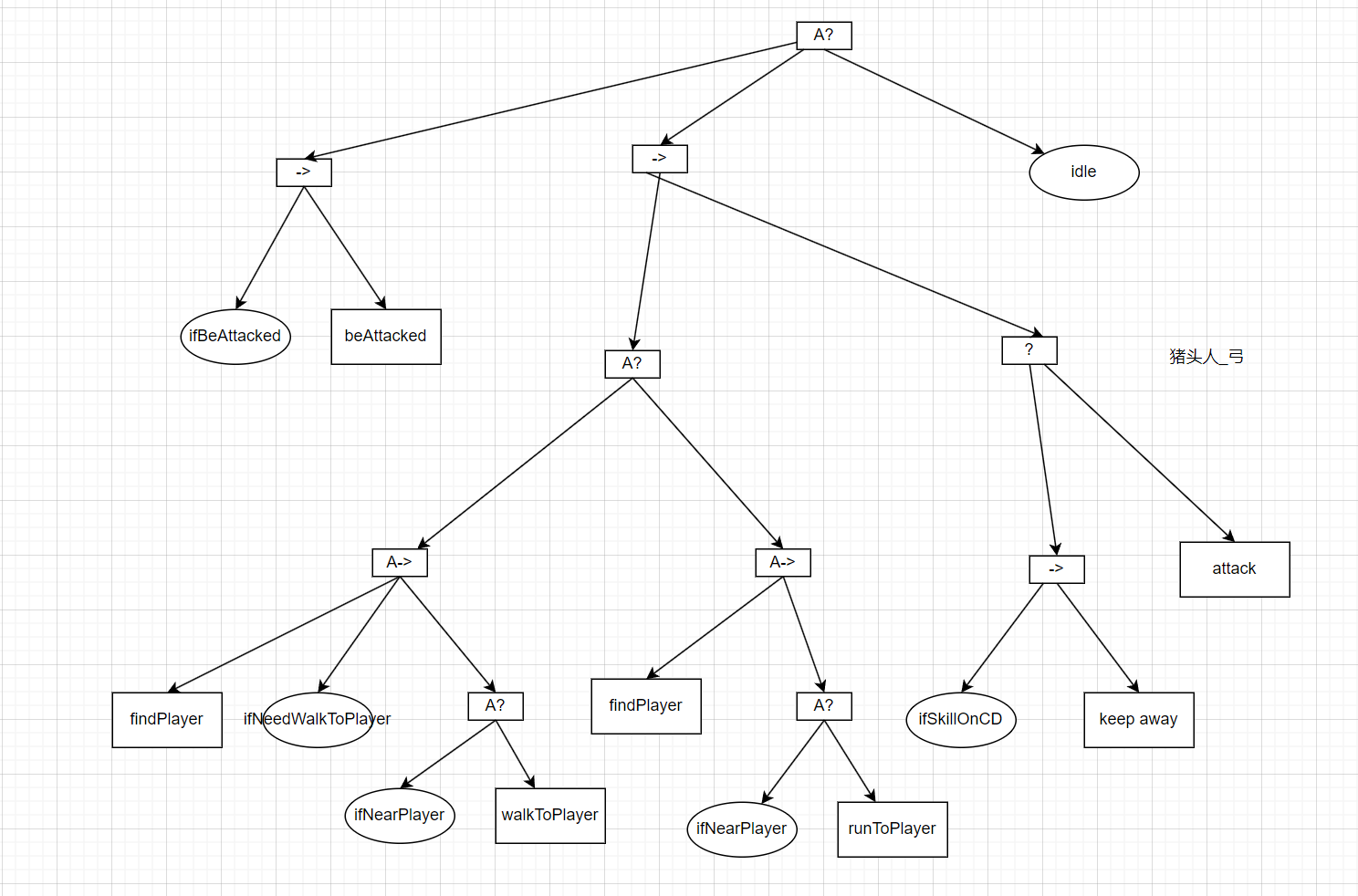
###### 具体实现：

通过行为树实现怪物的行为选择。

近战兵行为树：



远程兵行为树：



为每个节点定义一个节点类，为每个怪物编写单独的叶节点类，每帧调用行为树通过节点的执行，切换来实现怪物控制，通过叶节点类连接到怪物的类上控制相应的动画播放与逻辑判断。

#### 系统

通过GameManager类以状态机来控制整个游戏过程，每个状态对应一个场景。一共有4个场景：Root ，Start ，LevelSelection , Level\_forest 及3个状态：开始 ， 选关 ，关卡。

|  |  |  |
| --- | --- | --- |
| 场景 | 对应状态 | 详细说明 |
| Root | 无 | 用来导入不随场景销毁的game object，导入后立即进入Start场景且不会再返回该场景 |
| Start | 开始 | 游戏开始界面，Root场景导入game object后立刻切换到该场景。 |
| LevelSelection | 选关 | 玩家可以自定义关卡内敌人的数量 |
| Level\_forest | 关卡 | 根据上个场景的设置在几个固定范围内生成敌人 |

