Dark Allies 技术总结

Benhou Li

September 2024

1 物理系统

1.1 移动

对于拥有控制权的玩家角色，从输入设备读取移动指令，并将移动参数（如垂直和水平移动量、移动程度等）发送到服务器。

对于非拥有控制权的角色，则从服务器接收该角色的移动参数。

移动方向通过相机的正向向量乘以前进输入，再加上相机的右向向量乘以侧向输入来计算。

跳跃由独立的自由落体运动方程控制，以实现跳跃时的移动效果。

AI 角色的移动由导航网格（NavMesh）控制。

1.2 相机

相机的水平和垂直方向移动由两个父对象分别控制。

当未锁定目标时，相机根据玩家的输入自由旋转。

水平和垂直旋转分别根据操纵杆或鼠标的左右移动输入，通过改变角度来实现。

碰撞避免：

从相机的支点(旋转轴心)沿当前方向发射一个球形射线(Physics.SphereCast)，检测相机与支点之间是否有障碍物。

如果检测到碰撞，计算支点与碰撞点之间的距离。

调整相机的位置，使其目标位置设定在离碰撞点一定距离处，确保相机不会穿透障碍物。

设置最小距离限制，如果调整后的距离小于相机的碰撞半径，则将相机设置为距离支点的最小安全距离。

通过插值（Mathf.Lerp），将相机的当前位置逐渐移动到目标位置，确保调整过程的平滑性。

2 锁定系统

在扇形范围内搜索所有可锁定的对象。

锁定距离最近的目标。

存储左侧和右侧最近的目标，以便在锁定状态下切换目标。

使用四元数插值的方法，将相机的当前旋转逐渐调整到面向目标的方向，实现平滑的旋转效果。

创建协程，控制相机在垂直方向上平滑移动到比目标稍高的固定高度。

最后，保存当前的左右和上下角度，确保玩家在解除锁定时相机不会突然跳动。

3 存档与读档

3.1 存档

游戏的存档数据以 JSON 文件的形式保存在设备的持久化存储路径，确保在不同设备上都可以正确地保存和加载数据。

当玩家选择保存游戏时，系统根据当前使用的存档槽位选择相应的文件名。

游戏数据（如玩家的属性、状态、物品等）被提取并保存在当前的角色数据中。

然后，使用文件写入器将角色数据写入到 JSON 文件中，完成存档。

3.2 读档

当玩家选择加载游戏时，系统根据当前的存档槽位选择相应的存档文件名，并从 JSON 文件中读取数据，填充到当前的角色数据中。

然后，玩家的状态根据角色数据重新加载到游戏中。

在加载或新建游戏时，系统调用协程异步加载游戏世界场景，同时恢复玩家的数据。

4 伤害与攻击系统

伤害与攻击系统相互独立。当触发伤害时，同时播放音效和视觉特效，并根据受击角度决定受击者的动画表现。

攻击通过在动画系统中启用或禁用碰撞体（Collider）来实现，判断攻击是否命中目标。

5 武器系统

5.1 武器属性定义

系统首先定义了一个武器类，描述武器的各种属性，包括外观模型、攻击力、使用要求、攻击效果、耐力消耗等。这部分负责定义武器的基本特征，是武器系统的核心数据源。

该类还包含与武器动作相关的信息，如攻击动画、攻击音效等，用于在实际使用武器时提供相应的反馈。

5.2 玩家武器管理

系统包含一个玩家装备管理模块，用于管理玩家当前装备的武器。玩家可以在左右手之间切换不同的武器，并管理武器的快捷切换槽位。

这个模块连接玩家和武器，负责维护当前装备的武器状态，并提供接口供其他系统查询玩家当前装备的武器。

5.3 武器伤害管理

系统还包含一个武器管理模块，用于在战斗中处理武器伤害的应用。当玩家使用武器攻击时，该模块从武器定义中提取伤害信息，并将其传递给攻击的碰撞体（负责检测攻击命中的对象）。

通过与武器属性的关联，该模块将武器的伤害信息、攻击力和效果应用到战斗系统中，确保攻击的伤害计算与武器的属性一致。

总结

- **数据层**：武器属性类提供了武器的核心数据定义，包含所有与武器相关的属性信息。
- **管理层**：玩家武器管理模块负责维护和操作与玩家装备相关的武器状态，提供切换、装备、查询等功能。
- **应用层**：武器伤害管理模块负责在战斗中实际应用武器的属性，将玩家的攻击与武器的伤害属性相结合，实现战斗效果。

整体上，武器系统将武器的定义、装备管理和伤害处理分离成不同的模块，确保代码组织清晰，易于维护和扩展。

6 怪物 AI

状态机驱动：使用有限状态机（FSM）管理 AI 行为，包括待机、追击、战斗等，根据不同情况切换状态。

导航和移动：利用 Unity 的导航网格（NavMesh），实时检测和更新移动状态。

战斗管理：根据目标的位置和状态调整战斗行为，智能地进行追击和攻击。

动画与同步：动画管理确保移动和战斗动作与实际位置同步，网络同步则保证在多玩家环境下动作的一致性。

7 联机功能

主机：进入游戏时自动生成初始角色，如果读取存档则销毁初始角色，根据存档数据重新生成。

客户端：加入游戏时读取初始化的角色数据，未来将加入读取自定义角色的功能。