Easy Stats Kit_v1.0

感谢购买!

Easy Stats Kit 是一个功能强大且灵活的属性管理插件，旨在帮助游戏开发者轻松定义和管理各种属性值及其修正。无论是简单的塔防游戏，还是复杂的 RPG 属性系统，Easy Stats Kit 都能满足您的需求，减少从头编写复杂机制的麻烦。

Easy Stats Kit 提供了一个直观的编辑器窗口，使开发者能够通过简单的配置和 C# API 快速集成到现有项目中，实现高度可定制的属性管理。

特性

- **属性定义**：轻松定义任何对象的各种属性，如血量、攻击力、技能冷却时间等。
- **属性组**：通过创建属性组，实现属性的高重用性，支持多层次的继承和组合。
- **修饰符系统**：支持添加和移除修饰符，实现增益（BUFF）、减益（DEBUFF）等效果，支持多种计算类型，支持算法优先级排序。
- **初始化方式**：支持自动和手动初始化，灵活适应不同开发需求。
- **数据保存和加载**：支持将属性数据保存为 JSON 格式，并在不同场景中加载共享数据。
- **Stats 管理器**：内置一个强大的 Stats 管理器，帮助开发者管理某个对象的所有属性，提供全面的属性操作接口。
- **直观的编辑器属性界面**：提供方便的编辑器属性界面，可以管理和查看运行时的属性状态信息

无论您需要属性受到物品、法术、技能或其他因素的影响，Easy Stats Kit 都能有条理地管理并准确计算属性值。只需声明一个 MKStats 变量，设置其初始值，然后根据需要添加修饰符。最终的属性值可以从相关属性中获取。

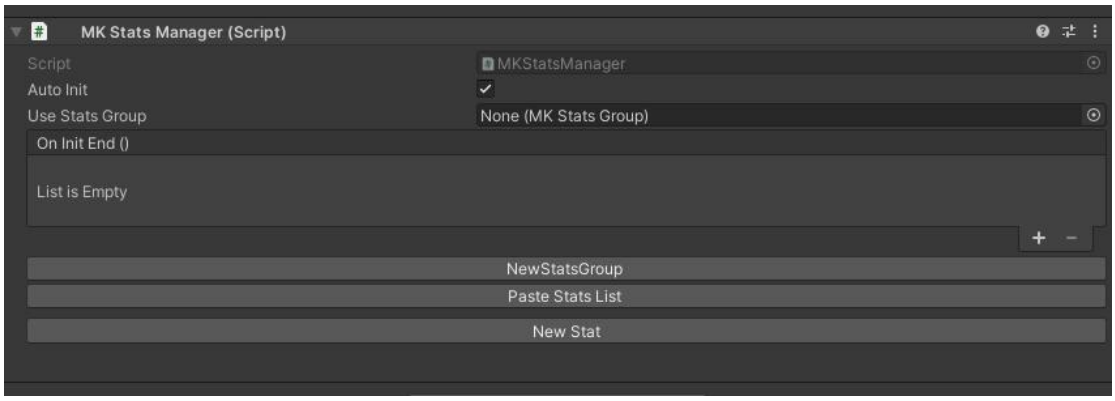
Easy Stats Kit 插件包含完整的源码，您可以根据需要随时自定义和扩展源码。

通过 Easy Stats Kit 插件，您可以高效地管理和动态调整游戏中的各种属性，为您的游戏开发提供坚实的支持和便利。

如果方便的话，请在商店页面写下你的评论，这对我来说很重要，是我继续更新开发更多新功能的动力！

快速入门

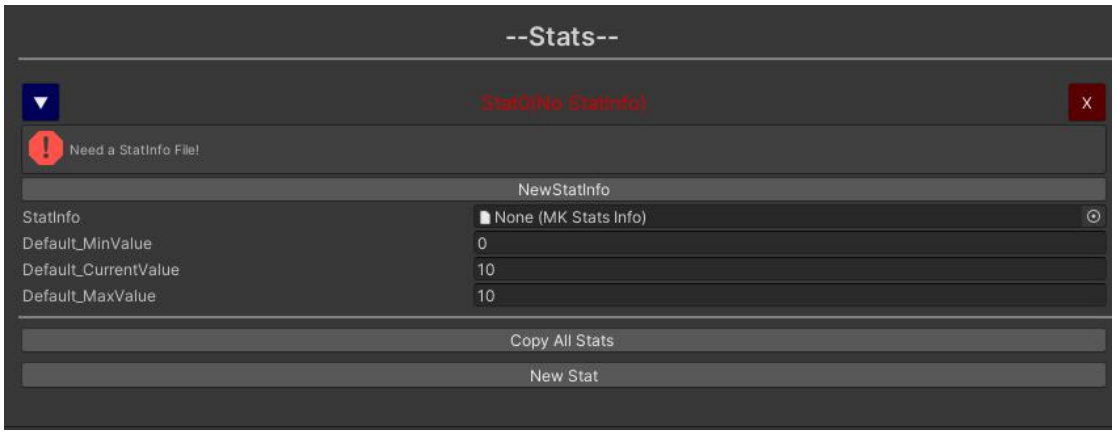
1. 设置 MKStatsManager



在任何游戏对象上添加 MKStatsManager 脚本（可能是你的玩家对象，或者怪物，或者任何其他需要动态变化属性的对象）

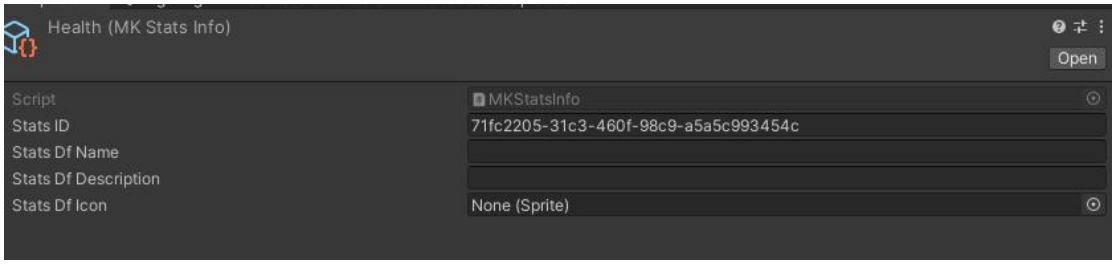
2. 创建 Stat

点击 MKStatsManager 组件上的 New Stat 按钮，新建一个属性

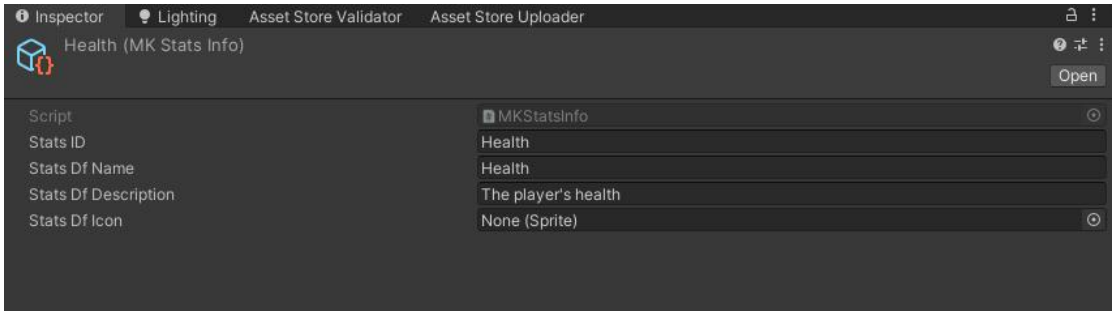


3. 创建 StatsInfo

此时会提示没有配置 Stat info，不用担心，我们点击 New StatInfo 按钮，创建一个新的属性信息配置文件，这里我们以生命值为例子

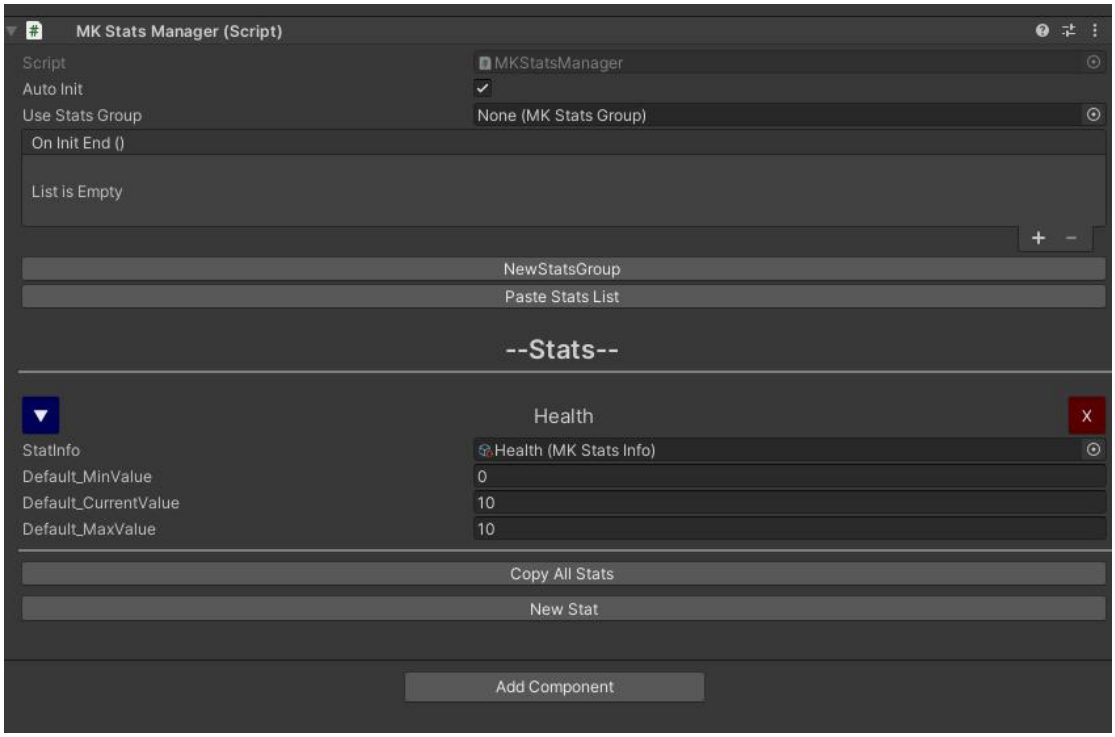


默认会为你生成一个唯一的 Stats ID,你可以自定义为其他 ID，如下图：



4. 配置 Stat

配置好你的 StatsTInfo 后，回去看刚才对象的 MKStatsManager 组件，他将自动赋值刚才创建的 Health 属性信息给这个新属性



你可以配置他们的默认初始值

InitDMinValue：最小值

InitDValue：当前值

InitDMaxValue：最大值

5. 完成！

一切准备就绪，你已经为一个对象配置了一个生命值的属性，你可以点击运行游戏，在 MKStatsManager 上，展开 Health 属性，点击 LOG 按钮，可以看到该属性当前的各种信息，包括当前数值和当前所有的修饰符

接下来要干什么？

接下来你可以在代码里获得和动态设置该属性的值（比如受到伤害或者治疗生命值），还可以为该属性添加修饰符（比如获得祝福 BUFF，最大生命值+50%）

```
public MKStatsManager mKStatsManager;//获得你的属性管理器

 mKStatsManager.SetInit();//手动或自动初始化属性管理器

 MKStats health = mKStatsManager.FindValidationStat("Health");//通过 ID 获取对应的属性对象

health .SetNowBaseValue(health .GetNowBaseValue() - damage);//造成伤害

//Another way: mKStatsManager.AddModToStats("Health", new MKStatModifier(0.5f, MKStatModType.PercentAdd, MKStatModTargetType.MaxValue, 0,"UpHeathBuff"));//添加修饰符

 Health .AddModifier(new MKStatModifier(0.5f, MKStatModType.PercentAdd, MKStatModTargetType.MaxValue,0, "UpHeathBuff"));//添加修饰符
```

其中可以设置 MKStatModifier 的 Order 属性，Order 越大，排序越后，越后参与计算，默认情况下，MKStatModType 类型从小到大的 Order 为，
Add=100,PercentAdd=200,PercentMult = 300

你可以在任何地方直接使用一个 MKStats 变量，并且用自己的代码去管理和使用他，不一定需要 MKStatsManager ， MKStats 已经包含了完整的 API。

案例场景

在插件包中，我们提供了一个示例场景来演示 MKStats 插件的功能。

您可以按照以下步骤运行和查看案例场景：

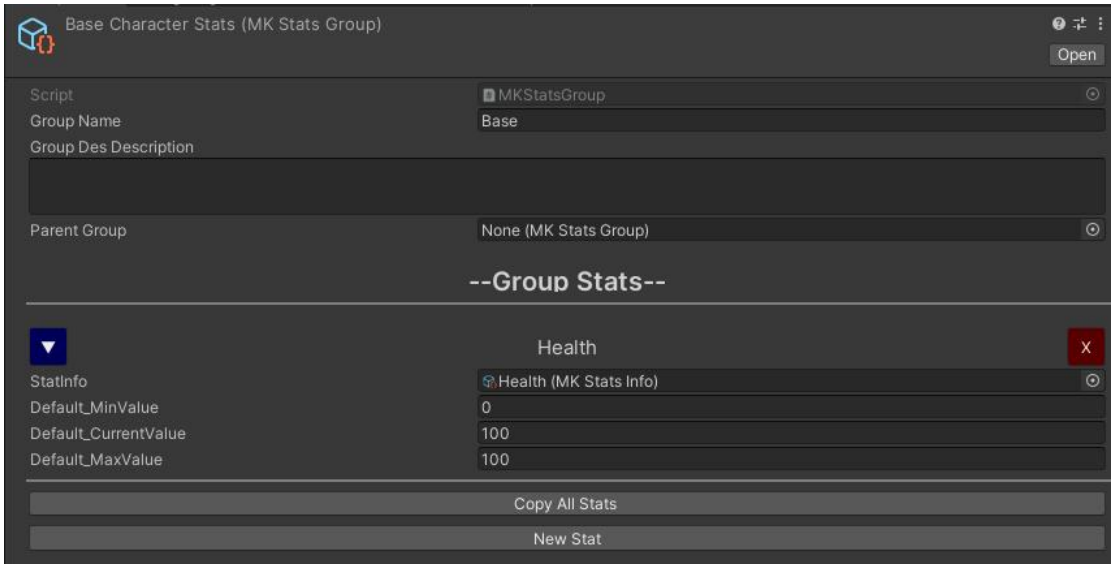
1. 导入插件包后，在项目视图中找到 **MKStats/Example** 文件夹。
2. 打开 **BaseMKStatsDemo** 场景文件。
3. 在场景中，您可以看到预配置的 **MKStatsManager** 以及相关的 UI 元素。
4. 运行场景，体验如何通过插件管理和操作属性值。

场景中的 TestPlayerStatsController 物体上的 TestMKStatsSet 脚本里，实现了大部分常用的属性接口使用方式，可以打开脚本参考使用

进阶使用

1.使用 MKStatsGroup 预配置属性组

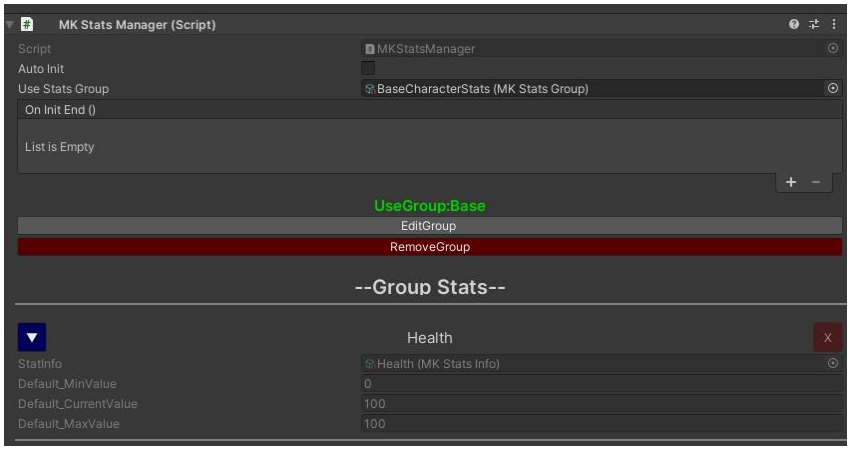
创建属性组: 在项目视图中右键点击并选择 Create -> MKGame -> StatsGroup，创建新的属性组文件。为属性组命名并保存。



配置属性组: 在属性组中添加属性信息文件，并配置每个属性的初始默认值、最小值和最大值。

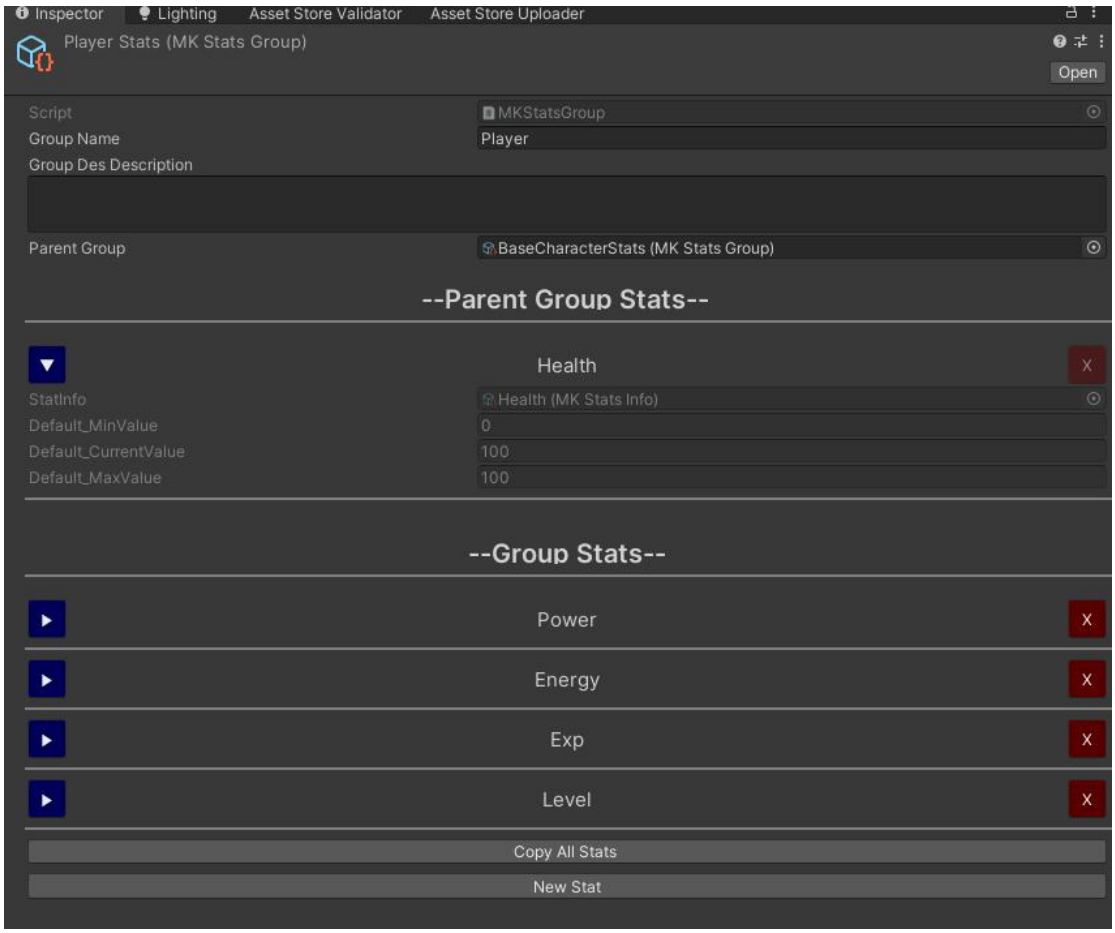
在 MKStatsManager 的 Inspector 中设置使用的属性组（UseStatsGroup）。

配置了 MKStatsGroup 的 MKStatsManager ，也可以定义自己的其他 Stat, 运行时 MKStatsManager 会自动合并，克隆实例化所有属性。



2. 属性组的继承和组合

属性组继承：您可以创建一个属性组并设置其父属性组。子属性组会自动继承父属性组中的所有属性，并可以在此基础上添加属性。



属性组和直接配置属性的组合：MKStatsManager 支持同时使用预配置的属性组和直接配置的属性。在运行时，MKStatsManager 会合并这些属性，并进行统一管理。

3.自动和手动初始化

默认情况下，MKStatsManager 会在 Awake 方法中自动初始化属性。

如果需要手动初始化，可以关闭 AutoInit 选项。

在需要手动初始化属性时，可以调用 MKStatsManager.SetInit() 方法来完成初始化。

API 接口

MKStatsManager 类

void SetInit() 初始化所有属性。

MKStats AddOneNewStat(MKStatsInfo mKStatsInfo) 添加一个新的属性。

参数: MKStatsInfo mKStatsInfo - 要添加的属性信息。

void RemoveOneStat(int id) 根据索引移除属性。

参数: int id - 要移除的属性索引。

void RemoveOneStat(MKStats stats) 根据属性实例移除属性。

参数: MKStats stats - 要移除的属性实例。

MKStats FindValidationStat(MKStatsInfo statsinfo) 根据属性信息查找属性。

参数: MKStatsInfo statsinfo - 属性信息。

MKStats FindValidationStat(string statId) 根据属性 ID 查找属性。

参数: string statId - 属性 ID。

void GetNowAllModifier(List<MKStatModifier> OutAllModifierList) 获取当前所有修饰符。

参数: List<MKStatModifier> OutAllModifierList - 用于存储当前所有修饰符的列表。

void AddModToStats(string statId, MKStatModifier mKStatModifier) 向属性添加修饰符。

参数: string statId - 属性 ID。
参数: MKStatModifier mKStatModifier - 修饰符实例。

void AddModToStats(MKStatsInfo statsinfo, MKStatModifier mKStatModifier) 向属性添加修饰符。

参数: MKStatsInfo statsinfo - 属性信息。
参数: MKStatModifier mKStatModifier - 修饰符实例。

void RemoveModToStats(string statId, MKStatModifier mKStatModifier) 从属性中移除修饰符。

参数: string statId - 属性 ID。
参数: MKStatModifier mKStatModifier - 修饰符实例。

void RemoveModToStats(MKStatsInfo statsinfo, MKStatModifier mKStatModifier) 从属性中移除修饰符。

参数: MKStatsInfo statsinfo - 属性信息。
参数: MKStatModifier mKStatModifier - 修饰符实例。

void RemoveModToStatsFromSource(MKStatsInfo statsinfo, string sourceKey) 从属性中移除特定来源的所有修饰符。

参数: MKStatsInfo statsinfo - 属性信息。
参数: string sourceKey - 修饰符来源键。

void RemoveModToStatsFromSource(string statsid, string sourceKey) 从属性中移除特定来源的所有修饰符。

参数: string statsid - 属性 ID。
参数: string sourceKey - 修饰符来源键。

void RemoveAllModToStatsFromSource(string sourceKey) 从所有属性中移除特定来源的所有修饰符。

参数: string sourceKey - 修饰符来源键。

string GetSaveJsonData() 获取当前属性数据的 JSON 表示。

void LoadSaveDataFromJson(string jsondata) 从 JSON 数据加载属性数据。

参数: string jsondata - JSON 数据字符串。

MKStatsInfo 类

string GetShowName() 获取显示名称。

string GetShowDescription() 获取显示描述。

Sprite GetShowIcon() 获取显示图标。

MKStatModifier 类

alue: float - 修饰符的值。
Type: MKStatModType - 修饰符的类型（如加法、百分比加法、百分比乘法）。
TargetType: MKStatModTargetType - 修饰符的目标类型（当前值、最大值、最小值）。
Order: int - 应用该修饰符的顺序。越大，排序越后，越后参与计算
SourceKey: string - 修饰符的来源标识符。
LastAppendMKStats: MKStats (只读) - 最近附加的 MKStats 实例。

全参数构造函数
使用所有参数创建一个新的 MKStatModifier 实例。
public MKStatModifier(float value, MKStatModType type, MKStatModTargetType ttype, int order, string sourcekey, MKStats TargetStat)

复制构造函数
通过复制另一个 MKStatModifier 创建一个新的 MKStatModifier 实例。
public MKStatModifier(MKStatModifier other)

部分参数构造函数
使用部分默认参数创建一个新的 MKStatModifier 实例。
public MKStatModifier(float value, MKStatModType type, MKStatModTargetType ttype)
public MKStatModifier(float value, MKStatModType type, MKStatModTargetType ttype, int order)
public MKStatModifier(float value, MKStatModType type, MKStatModTargetType ttype, string sourcekey)

从另一个 MKStatModifier 复制数据并可选地更新 LastAppendMKStats。
public void CloneFromOtherData(MKStatModifier mKStatModifier, MKStats NewMKStats)

使用一个新的 MKStats 实例克隆当前的 MKStatModifier。
public MKStatModifier CloneNew(MKStats NewMKStats)

string GetSaveJsonData() 获取修饰符数据的 JSON 表示。

static MKStatModifier LoadFromJsonData(string jsondata, MKStats targetstat) 从 JSON 数据加载修饰符。

参数: string jsondata - JSON 数据字符串。
参数: MKStats targetstat - 目标属性实例。

MKStats 类

可用的监听事件
UnityEvent<float,float> onBaseMinValueChanged
UnityEvent<float, float> onBaseValueChanged
UnityEvent<float, float> onBaseMaxValueChanged
UnityEvent<float, float> onMinValueChanged
UnityEvent<float, float> onValueChanged
UnityEvent<float, float> onMaxValueChanged
UnityEvent<MKStatModifier> OnAddModifie
UnityEvent<MKStatModifier> OnRemoveModifier

void SetNowBaseMaximum(float value) 设置当前最大值。

参数: float value - 最大值。

void SetNowBaseMinimum(float value) 设置当前最小值。

参数: float value - 最小值。

void SetNowBaseValue(float value) 设置当前值。

参数: float value - 当前值。

void AddModifier(MKStatModifier mod) 添加修饰符。

参数: MKStatModifier mod - 修饰符实例。

bool RemoveModifier(MKStatModifier mod) 移除修饰符。

参数: MKStatModifier mod - 修饰符实例。

bool RemoveAllModifiersFromSource(string sourceKey) 移除特定来源的所有修饰符。

参数: string sourceKey - 修饰符来源键。

string GetSaveJsonData() 获取属性数据的 JSON 表示。

static MKStats LoadFromJsonData(string jsondata, MKStatsInfo mKStatsInfo) 从 JSON 数据加载属性。

参数: string jsondata - JSON 数据字符串。
参数: MKStatsInfo mKStatsInfo - 属性信息。

支持和联系方式

如果您在使用过程中遇到任何问题或有任何建议，欢迎通过以下方式联系我们：

邮箱: 1174283584@qq.com

感谢您的支持，我们将不断改进和完善插件，为您提供更好的使用体验。

请一定给个好评！