BaseInput类：

工具类，简单的封装Input模块函数。供输入模块查询鼠标或者触摸或者轴事件的一些基本信息。可以重写自定义输入。通过封装Input类的方法和属性将不同输入设备的输入统一抽象为输入模块提供统一的接口。目的是能够支持自定义输入源覆盖Input类提供的默认输入。

下面是BaseInput提供的查询输入信息的方法及作用：

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

BaseInputModule类：

定义输入模块的生命周期和基础方法。输入模块的抽象基类，依赖EventSystem组件，给子类输入模块提供通用的字段、属性和方法（例如Process等流程控制等方法、事件冒泡相关方法）。

提供参与处理输入流程的方法（在EventSystem的Update中被调用的方法），即所谓的“生命周期”：



文本

AI 生成的内容可能不正确。

提供的统一访问输入数据的入口：

使BaseInputModule能够以统一的方式访问输入设备，而无需关心具体实现细节。允许在自定义输入逻辑的情况下复用输入模块的输入事件处理机制。

文本

AI 生成的内容可能不正确。

提供的其他的成员方法：

输入事件数据构建方法：

Ps：没有PointerEventData对象相关的方法是因为图上这两种是通用的输入模块数据，几乎所有输入设备都能触发这两种事件

文本

AI 生成的内容可能不正确。

提供的其他一些方法：

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

处理指针的离开和进入事件：

1.处理指针移出原物体的逻辑

文本

AI 生成的内容可能不正确。

2.处理指针没有离开原物体时的逻辑

文本

AI 生成的内容可能不正确。

3.处理指针移入新的可接收指针事件物体的逻辑

文本

AI 生成的内容可能不正确。

PointerInputModule：

继承BaseInputModule，定义指针输入的基本信息和基本方法，封装了处理所有指针类设备的公共逻辑。在基类的基础上扩展了与设备无关的通用处理逻辑。（如通用的拖拽、悬停等方法）

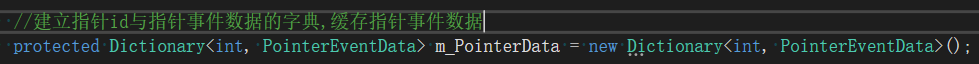
提供的重要的属性与方法

文本

AI 生成的内容可能不正确。

Ps：对于鼠标指针事件的数据，将鼠标左键、右键和中键分别建立独立的PointerEventData，原因：

1. 每个鼠标按键需要独立记录其按下、拖拽、释放等状态
2. 支持多按键并行操作



文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

建立触摸输入的指针事件数据：

文本

AI 生成的内容可能不正确。

Ps：触摸状态的枚举

文本

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。



PointerInputModule提供的一些虚方法及作用：

图片包含 文本

AI 生成的内容可能不正确。

StandaloneInputModule：

继承PointerInputModule，在基类的基础上扩展了鼠标、键盘、触摸、控制器这些处理具体设备输入的处理逻辑。

参与输入处理流程控制的方法的具体实现：

文本

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

处理具体设备输入的方法

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

总结

图形用户界面, 应用程序

AI 生成的内容可能不正确。