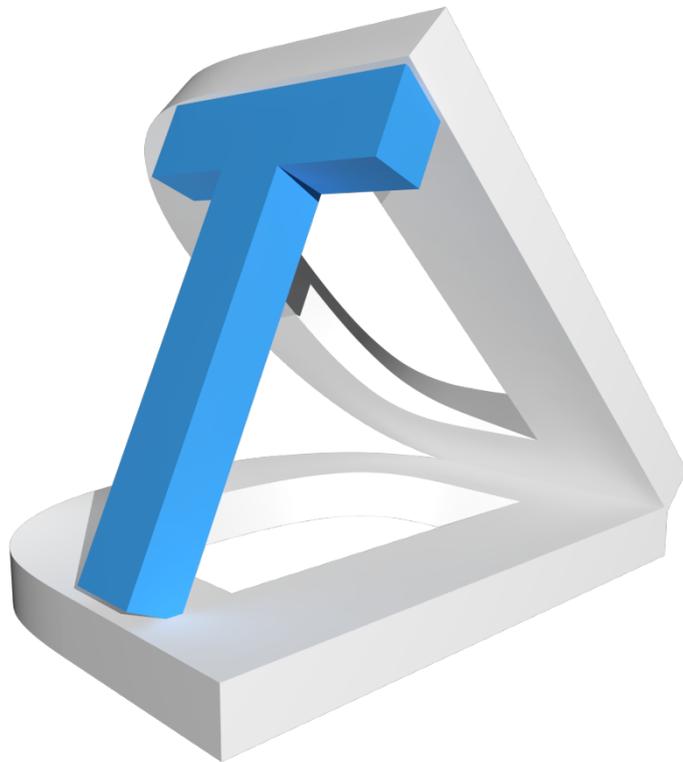


# MSVR 软件用户指南



# MSVR 软件用户指南

发布版本：2019/1/24

## 目录

1. 软件简介 .....	1 -
2. 安装说明 .....	2 -
2.1. 安装需求 .....	2 -
2.1.1. 操作系统 .....	2 -
2.1.2. 设备支持 .....	2 -
2.1.3. Unity .....	2 -
2.1.4. 立体支持 .....	3 -
2.2. 软件获取和安装 .....	3 -
2.2.1. 安装 .....	3 -
2.3. 授权 .....	3 -
3. 快速入门 .....	4 -
3.1. 范例环境 .....	5 -
3.2. 在 Unity 中导入 MSVRSDK .....	5 -
3.3. Unity 工程设置 .....	7 -
3.4. Unity 项目导出 .....	9 -
3.5. MSVR 设置 .....	9 -
3.6. 尽情尝试! .....	11 -
4. MSVR 详解 .....	11 -
4.1. 模块介绍 .....	11 -
4.2. 服务运行模块 .....	12 -
4.3. 显示环境模块 .....	13 -
4.3.1. CAD WALL 设置参数解释 .....	15 -
4.3.2. CAVE 设置参数解释 .....	16 -
4.3.3. 环幕设置参数解释 .....	17 -
4.4. 控制系统模块 .....	18 -
4.5. 集群投影模块 .....	20 -
5. 集群渲染 .....	23 -
5.1. 硬件 .....	23 -
5.2. 软件安装 .....	24 -
5.3. 应用程序安装 .....	24 -
5.4. 软件设置 .....	24 -
6. MSVR 开发包 For Unity .....	26 -
6.1. 导入开发包 .....	26 -
6.1.1. 导入 unitypackage .....	26 -
6.1.2. 为您的项目添加 MSVR .....	27 -

---

6.2. 在 Unity Editor 中运行 .....	28 -
6.2.1. 生成通道配置数据 .....	28 -
6.2.2. 在 Unity Editor 中运行您的场景 .....	28 -
6.3. 导出 Unity 项目 .....	29 -
6.3.1. 主动立体导出 .....	30 -
6.4. 运行您已导出的应用程序 .....	31 -
6.5. SynchronizeBehavior .....	31 -
6.6. 虚拟控制器 .....	33 -
6.7. 头部追踪器 .....	35 -
7. 常见问题 .....	36 -

## 1. 软件简介

MSVR 软件是一款专门用于沉浸式虚拟现实多通道的应用软件，由北京道航寰宇科技自主研发。

MSVR 软件可以在很短的时间内将 Unity3D 开发的内容发布至沉浸式虚拟环境，完成各种 3D 应用，支持多种的虚拟现实外设，多种立体显示方式，产品采用集群式设计，并且支持多画面高分辨率的立体展示环境。

MSVR 软件实现了以下几个主要的功能：

- ◆ 支持单机多显和高性能集群模式
- ◆ 拥有 CADWALL、CAVE、环幕、球幕等显示环境
- ◆ 可根据现实环境尺寸输入自动生成投影矩阵
- ◆ 集群设置时，无需输入 ip，可通过扫描自动生成集群
- ◆ 可拖拽屏幕安排投屏分配
- ◆ 支持非立体、主动立体、被动立体等显示方式
- ◆ 支持多种输入设备：VR 设备（HTC、Oculus、ART、游戏手柄、键盘鼠标等
- ◆ 提供输入设备扩展接口
- ◆ 统一虚拟手柄映射接口
- ◆ 提供“输入设备”可视化数据监控
- ◆ 提供集群计算机之间网络状态监控
- ◆ 提供“操作”录制，播放演示功能

MSVR 软件拥有快速而功能强大的配置工具，简单的几个步骤，就可

以建立属于您自己的 VR 系统，具有所见即所得的方式可以快速配置 VR 系统，为您更关注于您自己的 3D 应用赢得更多的时间。

MSVR 软件提供了成套的操作输入、操作同步、环境配置和环境自动生成解决方案，并且能够在不用修改工程的情况下更改通道配置、立体模式、输入设备。通过向您的工程导入 MSVRSDK 来制作、开发的应用程序，即可支持 MSVR 软件。

开发方面，MSVR 软件使用 C# 作为项目开发接口，并且提供 Unity3D 的 SDK 来支持在该平台上进行应用程序开发。提供 C++ 接口作为接入用户自定义设备。

## 2. 安装说明

### 2.1. 安装需求

#### 2.1.1. 操作系统

MSVR 软件支持 Windows 7 / 10，兼容 32 位和 64 位的系统。系统需要安装 DirectX 环境支持包。

#### 2.1.2. 设备支持

支持 HTC Vive、oculus，但是需要您的操作系统安装 SteamVR。

支持 ART，但是需要您的操作系统安装 DTrack。

#### 2.1.3. Unity

若要在 Unity 上进行应用程序的开发，您需要 Unity5.6.4 及以上版本；如果需要支持主动立体功能，您需要 Unity3D Pro 支持。

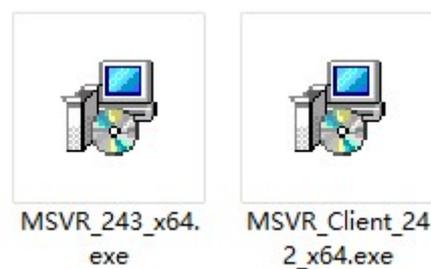
## 2.1.4. 立体支持

为了让您的应用程序能够正确地在主动立体模式下运行，您需要一块 nVidia Quadro 型号的显卡，并且安装了最新的显卡驱动。

## 2.2. 软件获取和安装

### 2.2.1. 安装

MSVR 软件包含两个软件包：MSVR 和 MSVR\_Client。



您可以通过双击以打开安装向导，作为开始，您只需安装 MSVR 即可。

关于客户端和配置器的具体安装请参阅 [5.2.软件安装](#)。

注意：请确保安装路径没有中文。

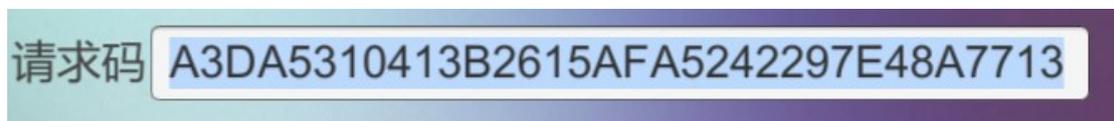
## 2.3. 授权

完成 MSVR 的安装后，通过打开桌面快捷方式进入软件，您会看到如下界面：



看到此界面意味着您还没有软件的授权或您的授权码已过期。

为了获取激活码，您首先需要向厂商发送请求码：首先单击请求码，然后按下 CTRL+A，以确保选中所有的激活码内容，您会看到请求码呈如下状态：



此时按下 CTRL+C 将请求码拷贝至粘贴板，并且粘粘至文本文件或其他地方，发送给厂商即可。

当您收到激活码后，只需拷贝至粘贴板后点击注册码的输入框，按下 CTRL+V 粘贴进注册码的输入框中，单击注册按钮即可。

### 3. 快速入门

在这一小节您将了解：

- ◆ 如何在 Unity3D 上正确导入、设置和生成应用程序；
- ◆ 如何使用配置器来启动该应用程序。

### 3.1. 范例环境

以下教程是在该环境下进行的：

- ◆ Windows 10 操作系统
- ◆ Unity5.6.4 Pro
- ◆ 有一个通道授权的 MSVR 软件
- ◆ USB 接入的标准键盘、鼠标

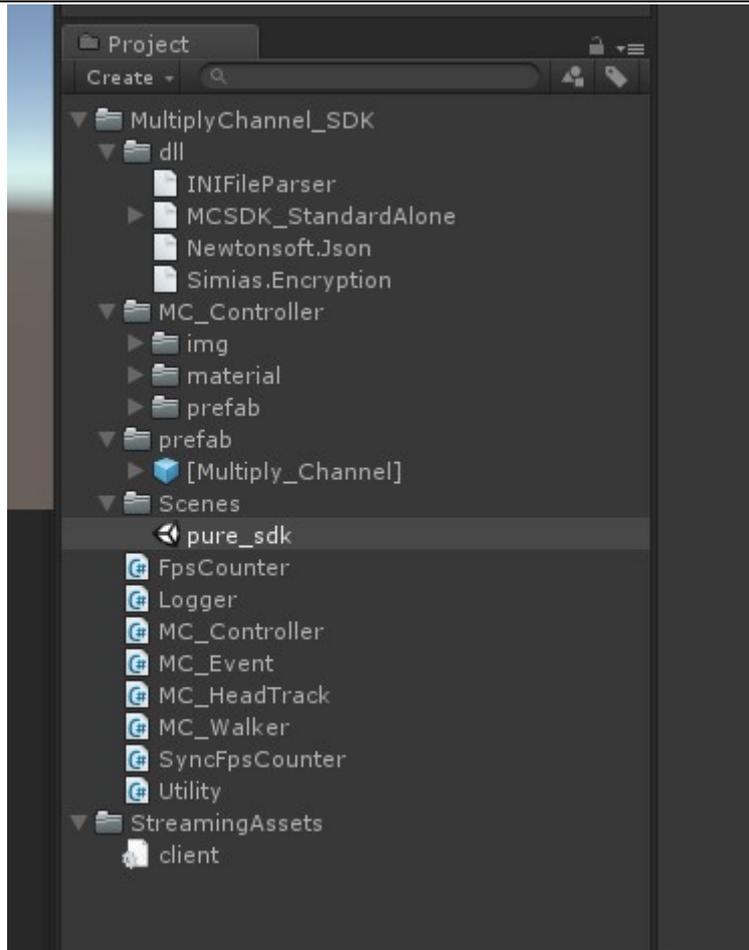
### 3.2. 在 Unity 中导入 MSVRSDK

一般情况下，MSVR 的 Unity 开发包的路径是 “C:\Program Files (x86)\VRbeyond\mc\MCSdk.unitypackage”。如果您自定义了软件的安装位置，请在您自定义的目录下寻找该文件。

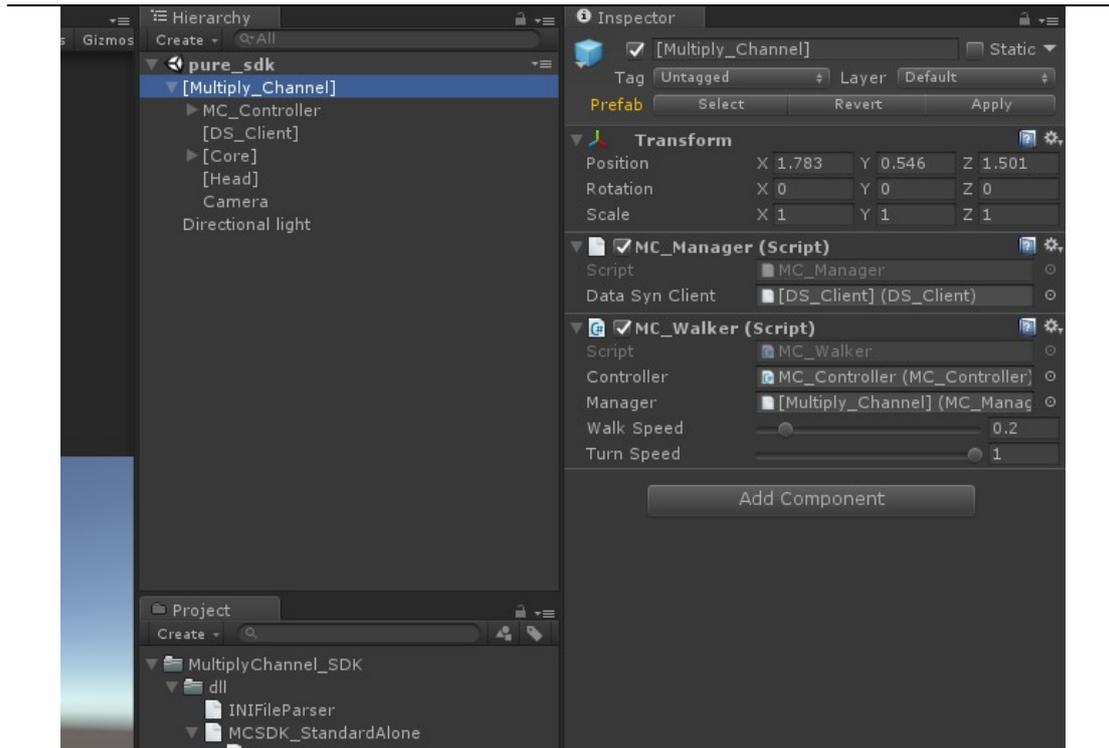
首先创建一个创建或打开您自己的场景，您可以通过以下方式中任何一种导入 MSVR 开发包：

- ◆ 依次打开菜单上的 Assets -> Import Package -> Custom Package...，在弹出的文件选择对话框中选择开发包；
- ◆ 在 Unity Editor 启动并打开场景后双击开发包
- ◆ 将开发包拖入 Unity Editor 的 Project 窗口

导入成功后您将看到如下的目录结构：

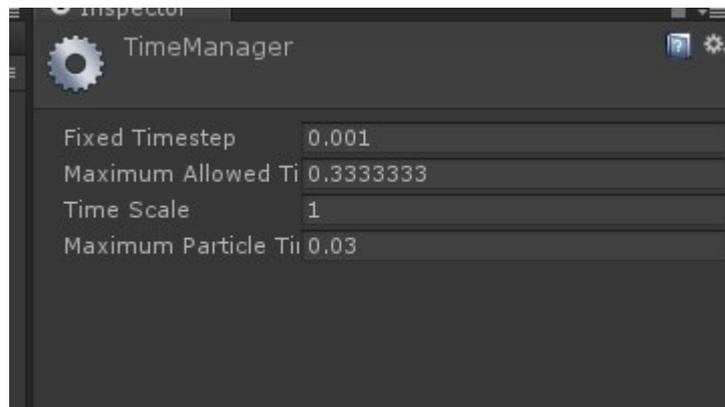


下一步您可以打开 Scenes 文件夹中的 pure\_sdk 场景或将 prefab 文件夹中的 “[Multiply\_Channel]” 预制体拖入您的场景中。预制体的结构如下图所示：

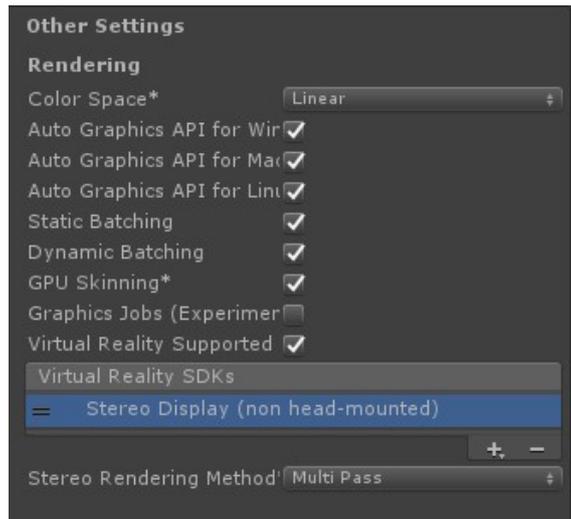


### 3.3. Unity 工程设置

打开 Edit -> Project Settings -> Time, 将 Fixed Timestep 设置为 0.001;

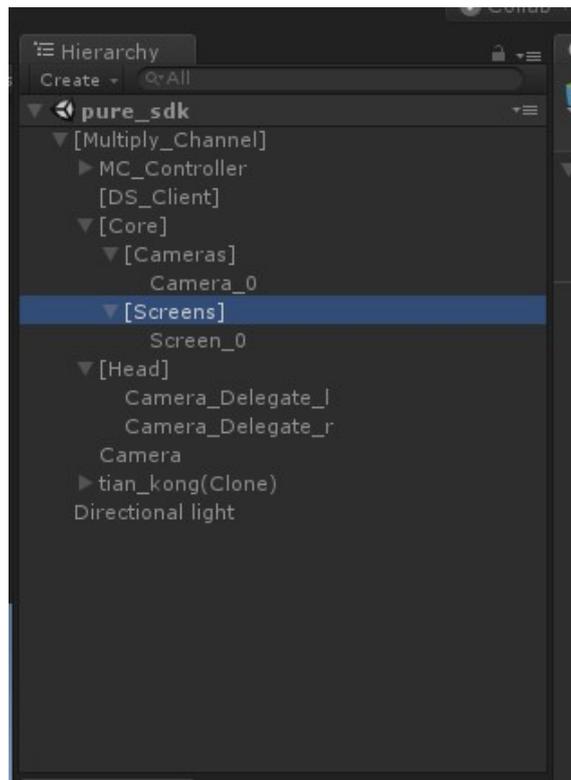


如果需要支持主动立体, 打开 Edit -> Project Settings -> Player, 在 Other Settings 选项中勾选 Virtual Reality Supported 选项, 并为 Virtual Reality SDKs 选项卡添加 Stereo Display(non head-mounted) 选项。



您可以在开发机上安装 MSVR 客户端来接受多通道配置数据，并在 Unity Editor 中按下 Play 按钮来查看效果。

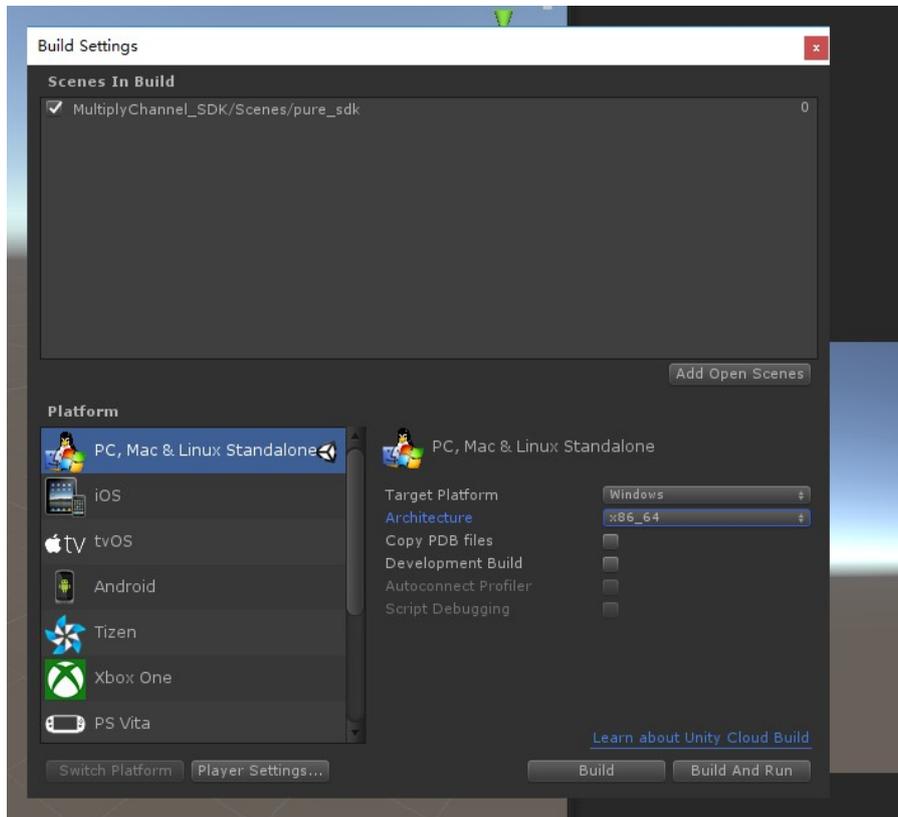
关于 MSVR 客户端与远端集群，请参阅[第 5 节](#)。



上图展示了读取配置数据后生成的通道在 Hierarchy 视图的结构。

### 3.4. Unity 项目导出

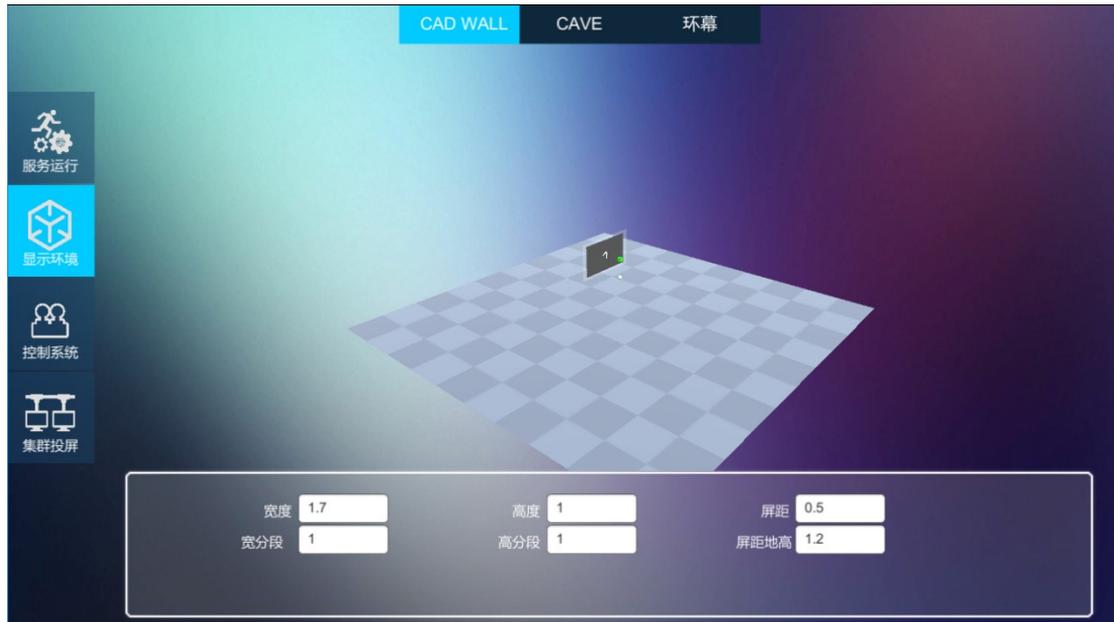
MSVR 开发包没有特殊的导出设置需求，只需按正常步骤导出场景至可执行文件即可。



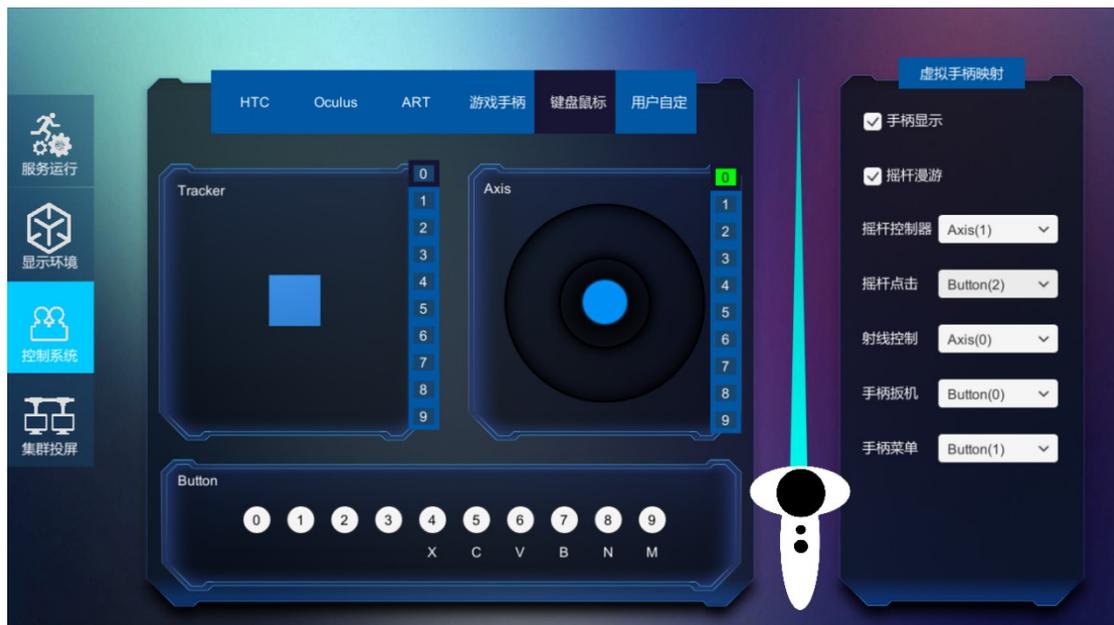
### 3.5. MSVR 设置

接下来，打开 MSVR，首先进行屏幕设置。

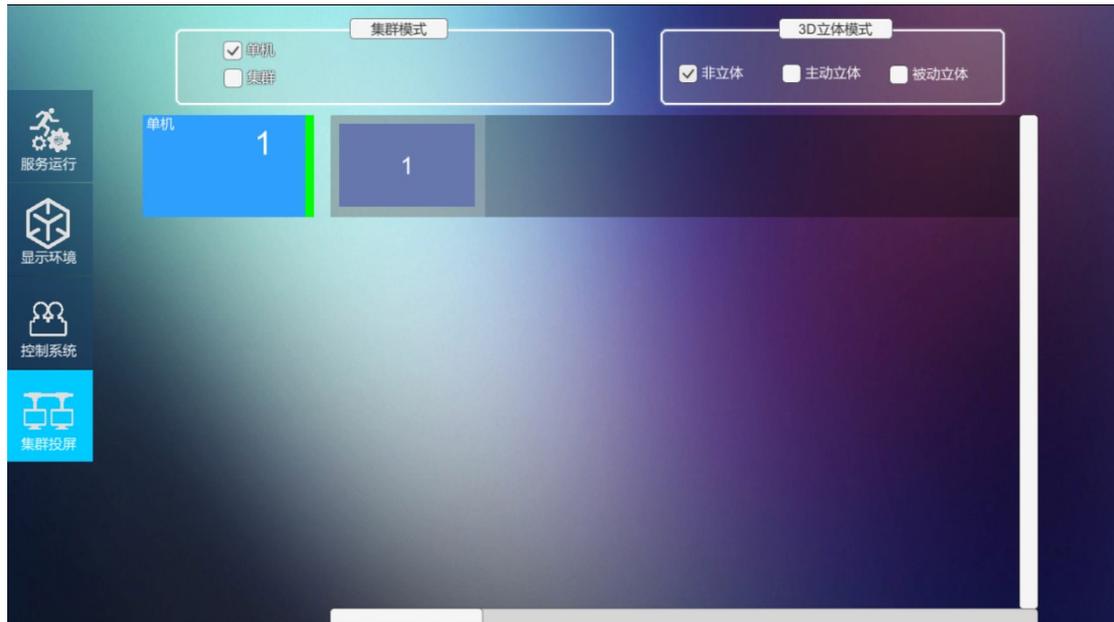
设置一个宽 1.7、高 1、屏距 0.5、宽分段 1、高分段 1、屏距地高 1 的 CAD WALL：



然后在控制系统中选择键盘鼠标，并对虚拟手柄映射做如下图设置：



接着在集群投屏中的集群模式勾选单机，并在 3D 立体模式中选择非立体：



最后打开服务运行选项，单击选择文件按钮选择导出的 Unity 可执行文件，单击 Server 即可开始运行。

### 3.6. 尽情尝试！

您已学会如何导入 MSVR 开发包到 Unity 中，正确设置项目并导出，在 MSVR 中配置通道以及设置虚拟手柄映射。

接下来您可以根据自己的需要，自由选择输入设备，更改屏幕尺寸或者类型，应用程序会根据您的配置自动调整内容。

## 4. MSVR 详解

在这一节，您将了解 MSVR 各个功能区的作用和配置方法，包括如何启动服务，如何配置环境，如何设置输入设备、映射虚拟手柄以及集群设置。

### 4.1. 模块介绍

MSVR 分为四个主要模块，它们分别是服务运行模块、显示环境模块、

控制系统模块和集群投屏模块。



服务运行模块管理开启/关闭配置器的同步服务（软件同步和操作同步），更改需要启动的目标应用程序，以及配置器日志的输出。

显示环境用于设置通道配置，例如通道配置、通道的实际尺寸以及输出的通道数量。

控制系统配置不同的输入设备，将它们映射至统一的虚拟输入设备中。在该界面，您可以选择输入设备并配置其至虚拟输入设备的映射位置。该界面也带有调试功能，您可以通过图形化界面查看该输入设备是否工作正常及每个输入被映射至的位置。

集群投屏界面可以设置立体模式，集群模式，控制集群渲染通道。

## 4.2. 服务运行模块

服务运行模块的界面如下图所示：



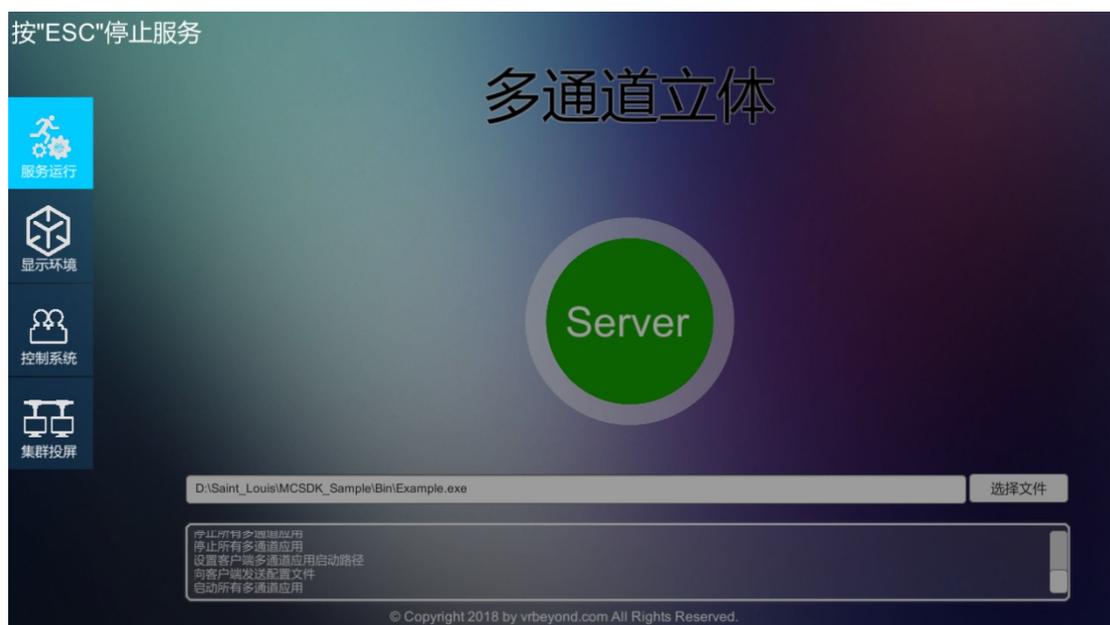
①. 运行服务按钮，该按钮的功能是启动所有连接上配置器的客户机

上指定位置的可执行程序，并启动配置器的同步服务。当服务运行时，界面将不能再作出改动，但依然可以查看。可以通过按下键盘的 ESC 键结束服务并再次编辑配置，该操作会让所有客户端关闭可执行文件；

②. 可执行程序选择栏，该部件提供了用于输入及显示的应用程序选择框，以及打开文件对话框来方便地选择可执行文件的“选择文件”按钮；

③. 配置器日志输出，您可以在这里查看该配置器的网络 IO 记录，及服务启动记录。

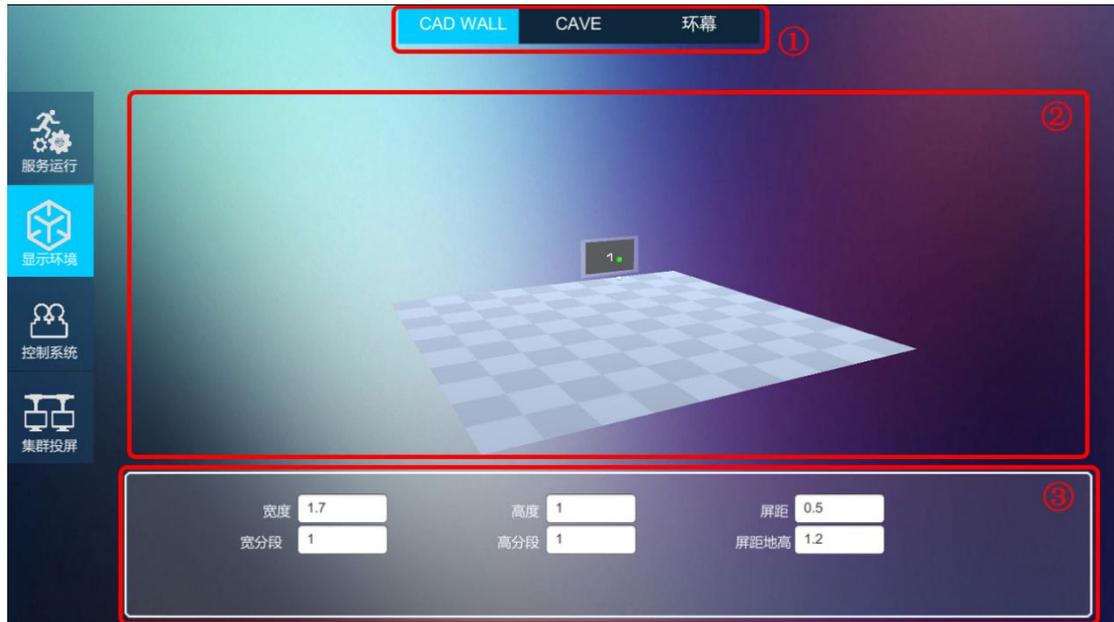
下图展示了当服务启动时的 UI：



此时您可以在各个模块中浏览已设置的信息但不能更改它们。仅当您按下键盘的 ESC 按键，使得服务停止后才能再次编辑。

### 4.3. 显示环境模块

显示环境模块的界面如下图所示：

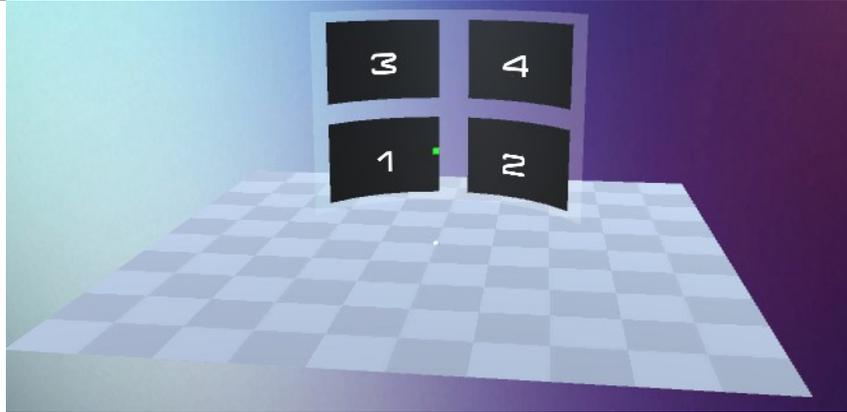


①. 显示环境选项，目前支持的屏幕有 CAD WALL, CAVE 和环幕三种。CAD WALL 是使用多个相同尺寸（多数情况下是相同品牌和相同信号的显示器）的屏幕拼接为一个大屏幕的模式；CAVE 则是通过多个屏幕或 CAD WALL 组成一个长方体的空间；环幕则是由多个屏幕拼接成一个弧形的屏；

②. 交互式环境预览，通过在该区域按住鼠标左键并拖动鼠标，可以 360° 预览当前配置的投影环境；

③. 环境配置参数，该区域提供几个重要的环境参数输入接口，您在此输入真实环境的参数后，交互式环境预览界面会立即更新。

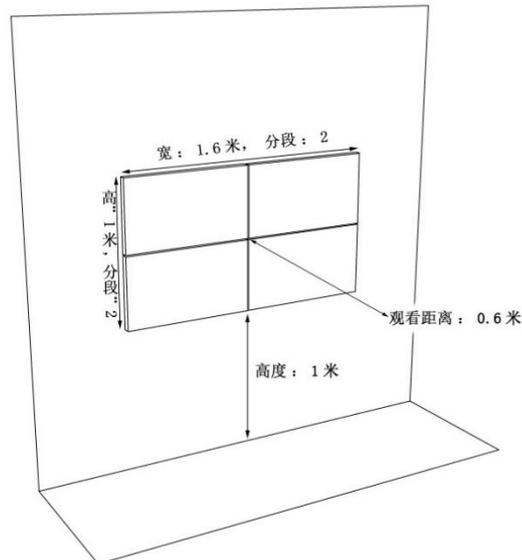
当宽分段或者高分段大于 1 时，您会注意到预览界面生成的环境有多个灰色矩形，并且每个矩形都被标上了一个数字：



其中，每个灰色矩形代表了一个渲染窗口，其所被标记的数字代表了该通道的编号。其具体含义将在 [4.4.集群投影模块](#) 中介绍。

### 4.3.1. CAD WALL 设置参数解释

如之前所描述的，CAD WALL 模式是指多块相同的矩形屏幕组合成一块大型的矩形屏幕。以下图中的环境为范例：



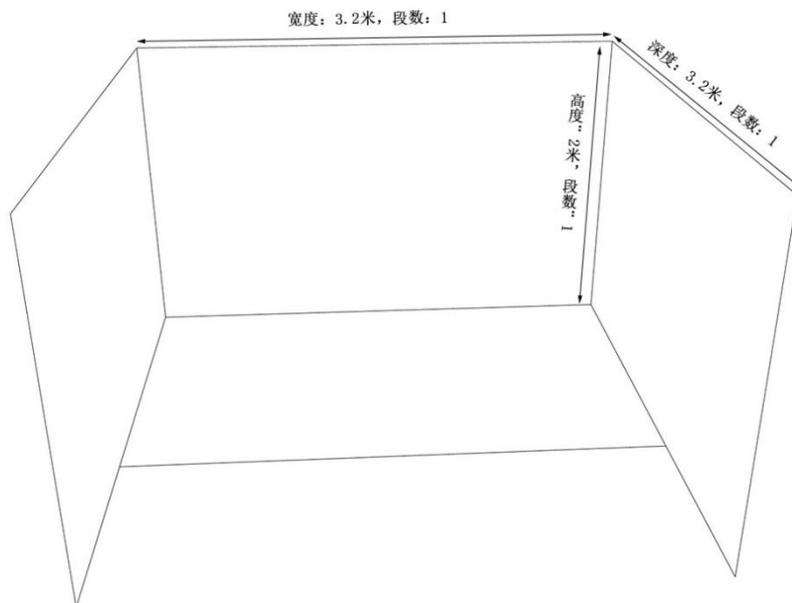
该环境是由四块屏幕一起组成的一面宽度 1.6 米，高度 1 米的 CAD WALL, 则其参数如下图所示：

宽度	1.6	高度	1	屏距	0.6
宽分段	2	高分段	2	屏距地高	1

宽度和高度为 CAD WALL 的总宽度、总高度。宽分段和高分段分别是组成该 CAD WALL 的水平方向屏幕数量和垂直方向屏幕数量,在该例子中,其水平方向和垂直方向的屏幕数量都是 2;屏距地高是屏幕的最底端距离地面的高度;屏距指拟定的用户观看位置和屏幕的距离。

### 4.3.2. CAVE 设置参数解释

CAVE 是由几面相互拼接为长方体的屏幕组成的,每块屏幕都朝向中心,用户在屏幕组成的长方体内进行操作,一般配合主动立体及跟踪器使用。以下图作为环境范例:



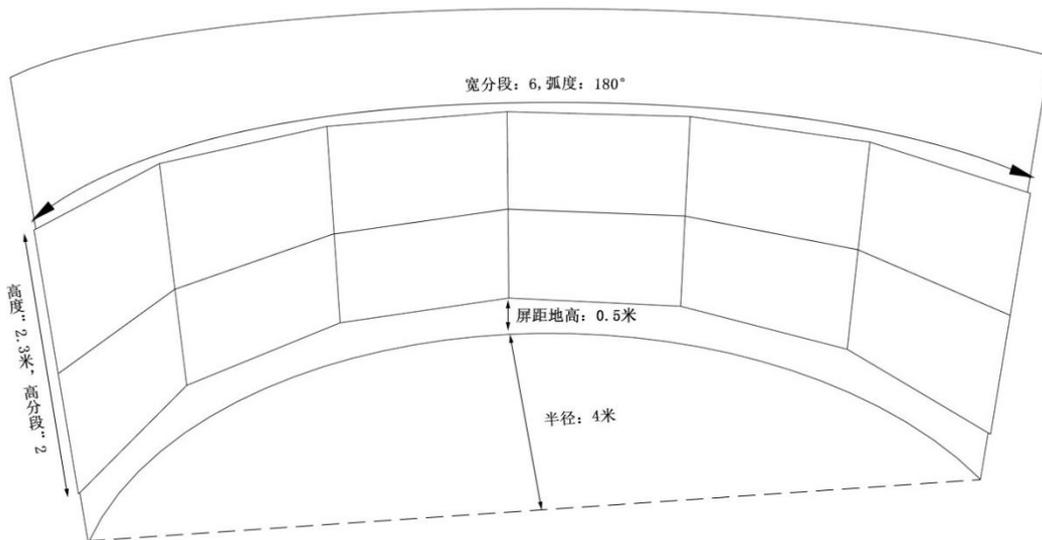
则其配置参数如下:

宽度	3.2	高度	2	深度	3.2	<input type="checkbox"/> 上
宽分段	1	高分段	1	深分段	1	<input checked="" type="checkbox"/> 下
屏距地高	0	<input type="checkbox"/> 全地面		<input checked="" type="checkbox"/> 前		<input checked="" type="checkbox"/> 左
				<input type="checkbox"/> 后		<input checked="" type="checkbox"/> 右

该环境是由四个面组成的 CAVE 系统（即四通道 CAVE），分别是左面、右面、前面和底面。在参数中，通过取消勾选“上”和“后”来配出正确的通道。“全地面”是指底面和顶面的高度（CAVE 的深度方向）是否与 CAVE 的深度一致，该选项会使底面在垂直方向变形，但是能够让底面与左右面的接缝完全拼合。

### 4.3.3. 环幕设置参数解释

环幕通过几面屏幕拼接出带有弧度的屏幕效果，以下图作为环境范例：

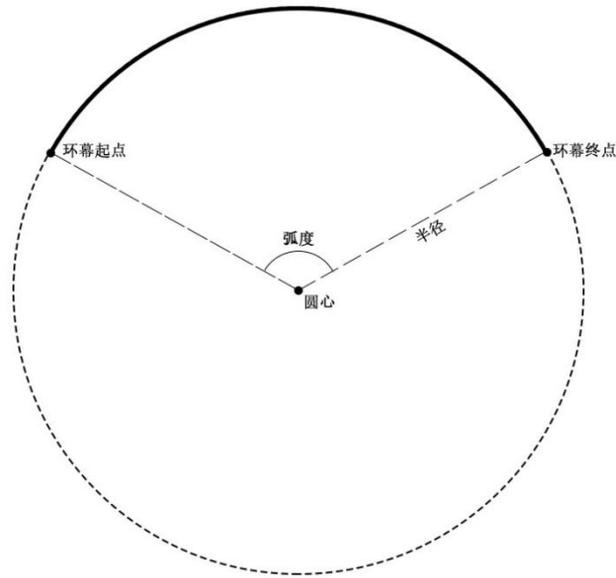


其配置参数如下：

弧度	180	高度	2.3	半径	4
弧分段	6	高分段	2	屏距地高	0.5

弧分段是环幕在弧长方向上屏幕的数量，高分段是环幕在高度方向上的屏幕数量，屏距地高是整个环幕的最底端距离地面的高度。

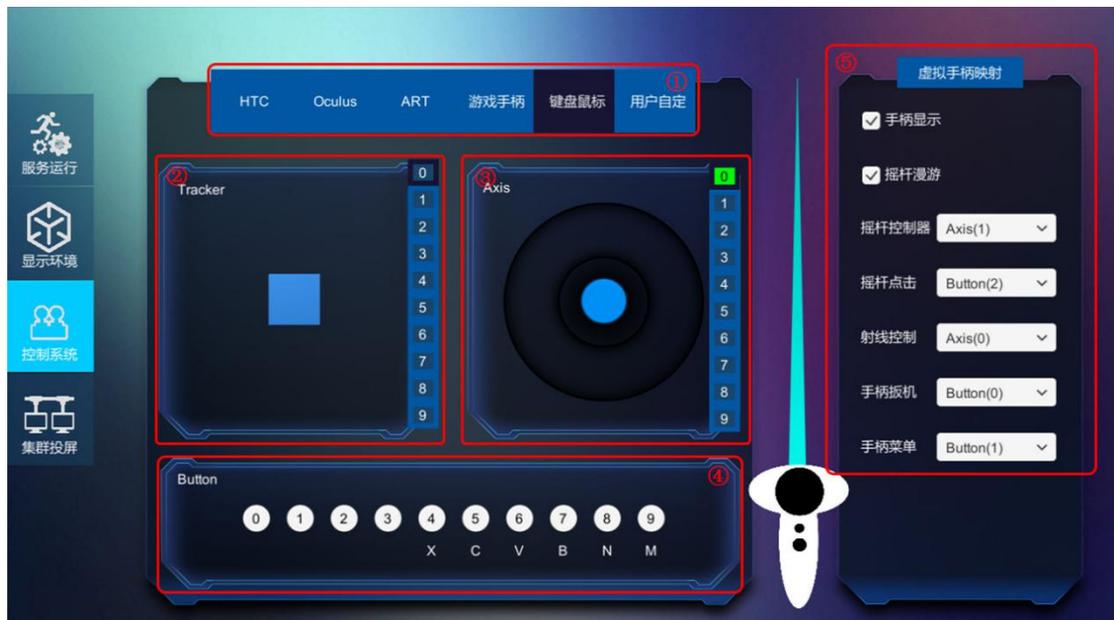
如果鸟瞰一个环幕，那么这个环幕是一个圆的一部分，即一段弧：



如上图所示，弧度为该弧圆心到环幕起点所作直线和该弧圆心到环幕终点所作直线的夹角。半径是圆心到环幕起点或者环幕终点的距离。

#### 4.4. 控制系统模块

控制系统模块的界面如下：



①. 输入设备选择。您可以在这里选择一个输入设备，目前支持的输入设备有：HTC Vive、Oculus、ART、游戏手柄；

②. 输入设备的追踪器监视界面。MSVR 设备的操作模块共支持 10 个 6 自由度跟踪器。该界面会将当前选择的输入设备的追踪器通过图形化的方式反馈。您可以通过点击右侧的按钮来查看不同编号的追踪器的图形化输出，同时，在某个追踪器有输入时，其对应编号的按钮会亮起绿灯，以提示您该编号的追踪器正在接受有效输入；

③. 输入设备的摇杆监视界面。MSVR 设备的操作模块共支持 10 个二维摇杆或触摸板输入。该界面会将当前选择的输入设备的摇杆通过图形化的方式反馈。您可以通过点击右侧的按钮来查看不同编号的摇杆的图形化输出，同时，在某个摇杆有输入时，其对应编号的按钮会亮起绿灯，以提示您该编号的摇杆正在接受有效输入；

④. 输入设备的按钮监视界面。MSVR 设备的操作模块共支持 10 个带有类比输入的按钮。该界面按编号顺序排列并同时显示 10 个按钮的按键状态，如果某个按钮正在被按下，则这个编号的按钮图标便会亮起绿灯；

⑤. 虚拟手柄映射界面。该界面用于将您所选择的输入设备映射至虚拟手柄中，虚拟手柄用于在应用程序开发中使用，使得您可以开发一次后修改输入设备而无需修改应用程序。其可能会列出的选项及其含义如下：

头部跟踪：启用/禁用头部跟踪，指定一个跟踪器作为头部跟踪的输入；

手柄显示：启用/禁用手柄的功能及模型显示；

摇杆漫游：启用/禁用自带的摇杆漫游功能，该功能可以使您的应用程序带有简单的漫游功能而无需编写代码；

摇杆控制器：指定一个摇杆输入作为虚拟手柄的摇杆输入；

摇杆点击：指定一个按钮输入作为虚拟手柄的摇杆按键按钮；

射线控制：指定一个摇杆输入操作虚拟手柄自身的旋转，该选项出现在没有跟踪器作为控制器输入的情况下；

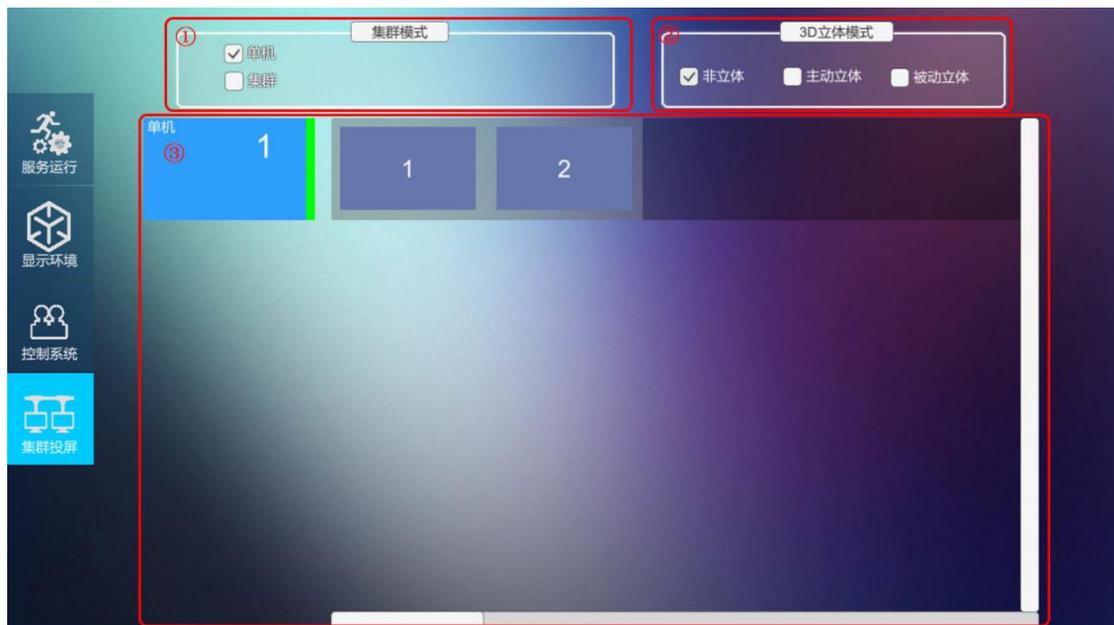
手柄跟踪：指定一个追踪器作为手柄的位置和旋转信息输入；

手柄扳机：指定一个按钮输入作为虚拟手柄的 Trigger 按钮；

手柄菜单：指定一个按钮输入作为虚拟手柄的 Menu 按钮。

## 4.5. 集群投影模块

集群投影的界面如下：



①. 集群功能面板。您可以选择单机和集群两种模式，单机模式下您的应用程序将在当前机器上启动，集群模式则会开启集群功能。

当集群功能开启时，将会进行一次客户端扫描，如下图所示：



扫描结果将以列表的方式在扫描结果中列出，您可以通过点击其 IP 号前面的勾选框将该客户端加入集群渲染中。要注意的是，当前配置器中的客户端并不会出现在扫描结果中，因为其总是被连接至服务。另外，如果已选择的客户端数量到达授权所允许的上限时，剩余的客户端将无法被选择，除非您取消一个已选择的客户端。关于集群，具体请参阅 [5.集群渲染](#)。

当您切换至集群模式时，界面类似如下：



其中集群功能面板增加了扫描功能，以允许您重新扫描并指定客户端。“服务器渲染”勾选框则指定当前 MSVR 是否需要参与渲染，若开启该选项，则屏幕分配器也会为其分配渲染窗口，并且在服务开启时本机也会启动您指定的应用程序。

通道面板则会显示所有您选择的客户端，其具体含义在③中阐述。

②. 3D 立体模式面板。您可以在该面板中选择渲染的模式：非立体意味着所有应用程序以正常的方式渲染画面；主动立体模式需要您的显卡及显示器支持，应用程序会尝试以主动立体的方式启动。关于主动立体的硬件及软件支持，请参阅 [2.1.2.设备支持](#)；被动立体模式则应用程序将以左右格式的被动立体输出画面。

③. 通道配置面板。您可以在该面板查看您所选择的客户端，它们的 IP 地址，编号及所负责的渲染窗口。该面板的第一位永远是服务器。客户端信息卡片右侧的竖状灯条代表该客户端目前的状态，绿色意味着该客户端正常且已连接至 MSVR，红色则意味着该客户端无法连接至服务器。

信息卡片的右侧则是渲染窗口配置，渲染窗口则是您在显示环境中配

置的每一个渲染窗口。您可以使用鼠标拖拽窗口到其他位置来更改渲染窗口顺序，您也可以将渲染窗口从一个客户端转移至另一个客户端。

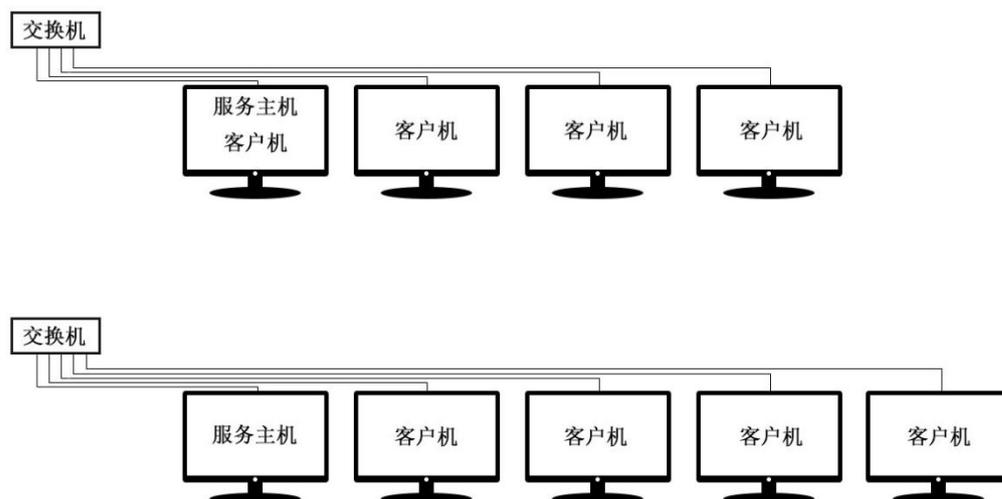
渲染窗口配置将在 MSVR 服务开启时自动通知每一个客户端，客户端将会以您的配置方式启动应用程序并设置渲染的画面至您所调整的渲染窗口配置。

## 5. 集群渲染

在这一节，您将了解如何在由多台计算机组成的渲染集群上配置 MSVR。

### 5.1. 硬件

在集群渲染中，您需要指定其中一台计算机作为服务主机使用，您可以单独使用一台机器作为服务主机，也可以使用一台渲染机同时作为客户端和服务主机。例如同样的四通道集群，可以由如下两种方式实现：



您只需要把输入设备接入服务主机，操作同步服务会自动将您的操作

同步至每一台客户机。

## 5.2. 软件安装

MSVR 软件共有两个软件包：MSVR 和 MSVR\_Client。您只需按照以下两个规则安装软件包即可：

1. 对于服务主机，无论其是否作为渲染客户机，都只安装 MSVR；
2. 对于客户机，只要不是服务主机，都只安装 MSVR\_Client。

## 5.3. 应用程序安装

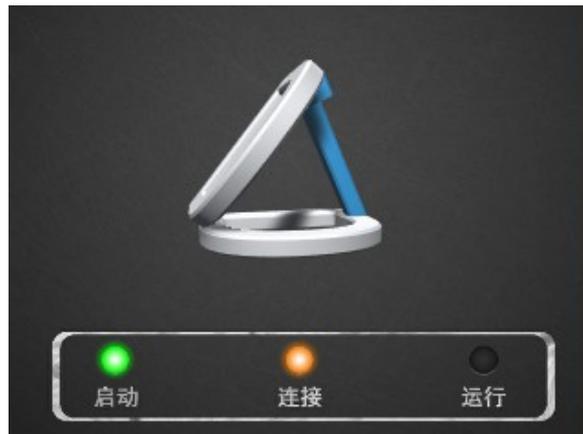
MSVR 软件要求每一台客户端的机器都安装您的应用程序，并且它们都在相同的路径下。服务主机若是也作为客户机使用，则服务主机所在的机器也许安装应用程序，并且安装路径与其他客户机保持一致。

## 5.4. 软件设置

当您的软件与应用程序都安装完毕，并拥有合适的授权时，即可按照以下步骤开始使用集群功能：

- 1.启动 MSVR，正确配置环境、输入映射及启动位置。您也可以先打开客户机上的 MSVR\_Client 软件（例如将客户端设置为开机自启动），在这里没有强制要求其打开的先后顺序，您只需确保在配置器上扫描客户端时客户端已正常启动即可。

- 2.打开所有客户机上的 MSVR\_Client 软件（如果您有一台客户机也作为服务主机使用，则不用做任何操作），其界面如下：



界面中的三个指示灯其不同颜色下的含义如下：

启动：MSVR\_Client 在开启时会进行简单的环境检查，若启动指示灯不是绿色，则说明软件在该环境下无法正常启动；

连接：该指示灯指示当前客户端与 MSVR 的连接状态，若指示灯为黄色，则客户端在等待连接状态。若指示灯为绿色，则客户端成功与 MSVR 建立连接；

运行：该指示灯只有在 MSVR 启动服务时会亮起。如果配置器指示的应用程序路径在当前机器上无法找到或无法启动，则该指示灯变为红色。如果客户端顺利找应用程序并启动则会亮起绿灯。

3.打开多通道配置软件，在集群投影模块中选择集群模式的“集群”选择框。

4.在弹出的扫描结果窗口中选择您需要的客户机（您可以通过列表中的客户机 IP 来判断是哪个客户端，并勾选需要参与集群的客户端），按下确认键选择需要的客户端。

5.此时被连接的客户端的连接指示灯会亮起绿灯，在通道配置面板中也能查到该客户端的状态指示灯为绿色。

6. 切换至服务运行模块点击 “Server ”按钮开始集群服务。

恭喜，现在您已经知道如何在集群渲染系统中安装和设置 MSVR 软了。

## 6. MSVR 开发包 For Unity

在这一节，您将了解：MSVR 开发包的结构，以及如何使用 MSVR 开发包进行应用程序的开发。

### 6.1. 导入开发包

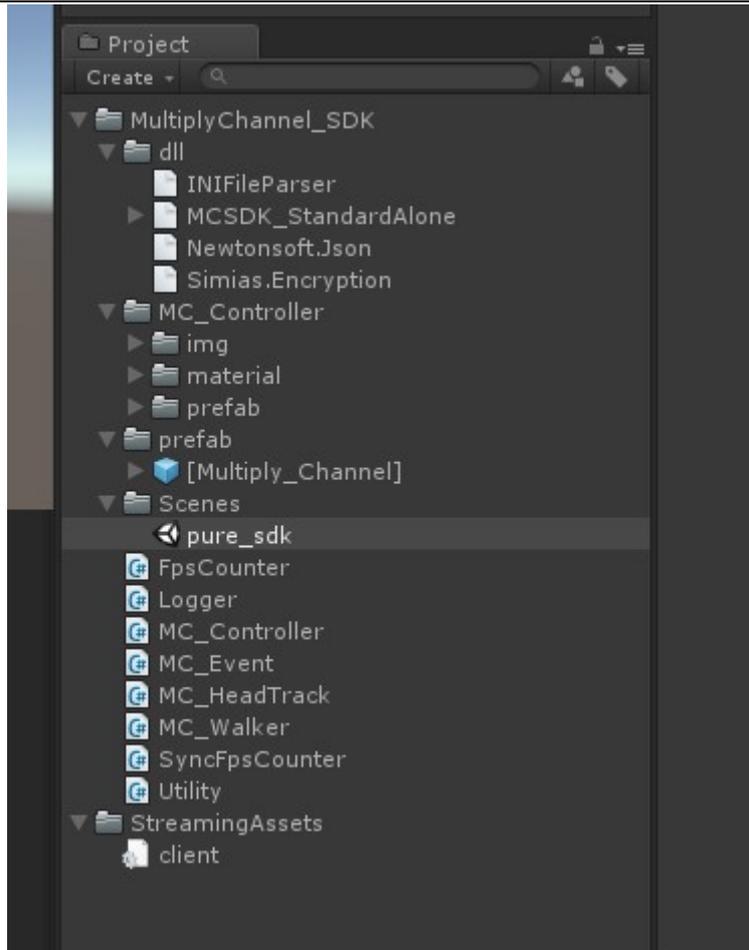
#### 6.1.1. 导入 unitypackage

一般情况下，MSVR 的 Unity 开发包的路径是 “C:\Program Files (x86)\VRbeyond\mc\MCSDK.unitypackage”。如果您自定义了软件的安装位置，请在您自定义的目录下寻找该文件。

在创建一个创建或打开您自己的场景后，您可以通过以下方式中任何一种导入 MSVR 开发包：

- ◆ 依次打开菜单上的 Assets -> Import Package -> Custom Package...，在弹出的文件选择对话框中选择开发包；
- ◆ 在 Unity Editor 启动并打开场景后双击开发包
- ◆ 将开发包拖入 Unity Editor 的 Project 窗口

导入成功后您将看到如下的目录结构：



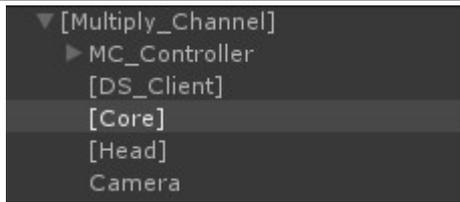
至此，您已经完成了 MSVRUnity 开发包的导入。

### 6.1.2. 为您的项目添加 MSVR

在将 MSVRUnity 开发包导入之后，您可以在项目目录“Assets / MultiplyChannel\_SDK / prefab /”中找到名为 “[Multiply\_Channel]” 的预制体，将该预制体拖至您的场景即可导入 MSVR。

如果您是从头开发，您也可以在“Assets / MultiplyChannel\_SDK / Scenes /”目录中找到名为“pure\_sdk”的场景，该场景已导入预制体，您可以直接开始制作自己的内容。

预制体 “[Multiply\_Channel]” 的结构如下图所示：



## 6.2. 在 Unity Editor 中运行

当您把预制体导入到场景中后，您的应用程序就能够支持 MSVR 的功能了。

### 6.2.1. 生成通道配置数据

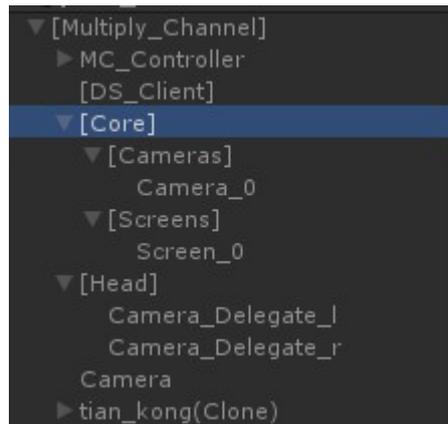
为了在 Unity Editor 中运行该应用程序，您需要在开发机上安装 MSVR 客户端或 MSVR 以生成通道配置数据。

在配置器中配置好您的环境后，将运行程序位置清空并开始服务，这将在开发机上生成配置数据，之后您可以不用再次配置环境，该配置数据会一直被保留直至被覆盖。

关于如何使用 MSVR\_Client 接受配置数据请参阅 [5.集群渲染](#)，如何在单机上生成配置数据您可以参阅 [3.快速入门](#)。

### 6.2.2. 在 Unity Editor 中运行您的场景

在您的开发机上生成通道数据后，您可以按下 Play 按钮来预览场景。在开始运行后，预制体中会根据您的配置增加几个游戏对象：

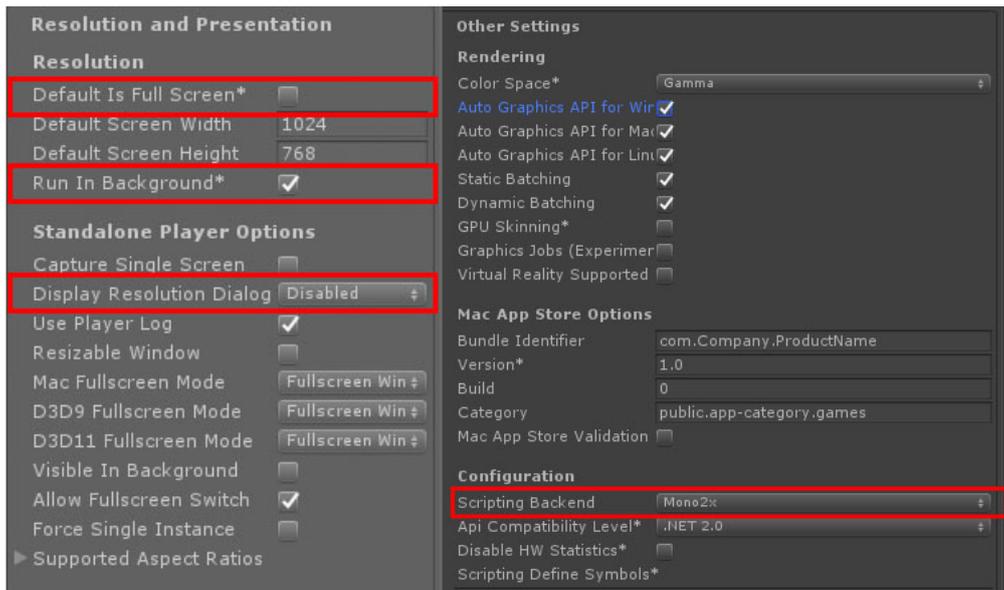


在 “[Core]” 中，增加了您所配置的相机群和屏幕群； “[Head]” 中根据您的立体模式会添加 “Camera\_Delegate\_l” 和 “Camera\_Delegate\_r” 或者 “Camera\_Delegate” ，这些节点用来模拟观察位置；在 “Multiply\_Channel” 中增加了 “tian\_kong(Clone)” 游戏对象，这个游戏对象是由多个带有碰撞盒的平面拼成的盒子，用来限制虚拟控制器的射线范围。

### 6.3. 导出 Unity 项目

在将您的应用程序导出为 StandAlone 版本之前，您需要先设置 Unity 配置参数。

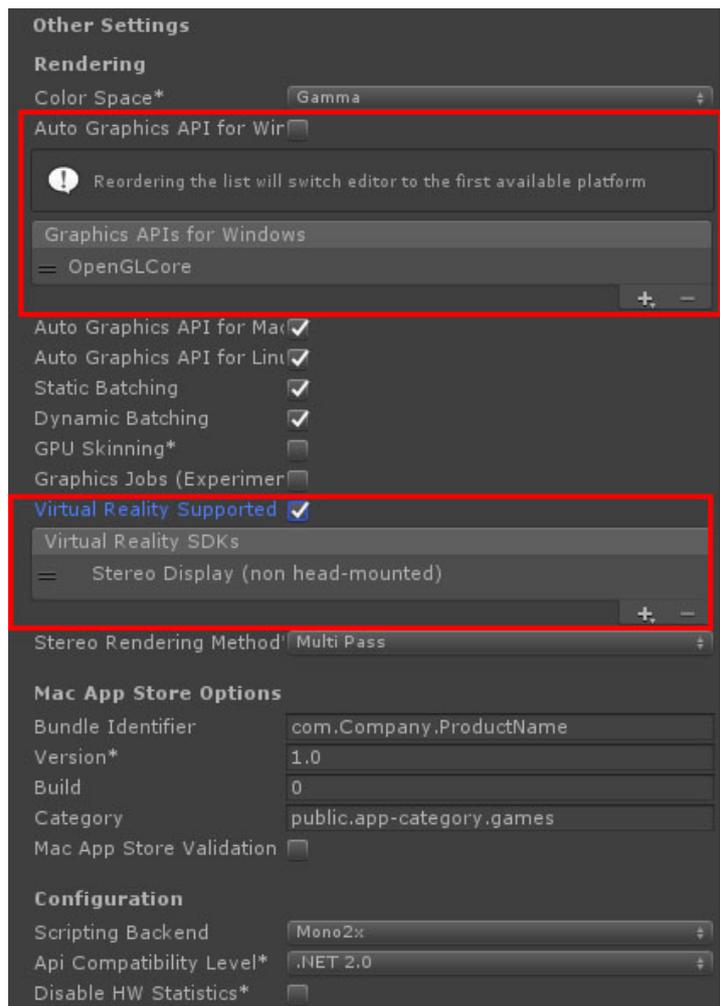
打开 Edit -> Project Settings -> Player，确保下图中红色框中的选项与下图保持一致：



### 6.3.1. 主动立体导出

如果您需要您的应用程序支持主动立体，则需要设置额外的参数。

打开 Edit -> Project Settings -> Player，确保下图中红色框中的选项与下图保持一致：



## 6.4. 运行您已导出的应用程序

当您将自己的应用程序导出后，即可以在 MSVR 中设置该可执行文件为目标路径运行程序。关于如何正确启动导出的应用程序，请参考 [3.5MSVR 设置](#) 的设置服务运行部分。

## 6.5. SynchronizeBehavior

在 Unity 中，当我们为场景创建新脚本时，Unity 会通过一个模板生成一个新的脚本，该脚本会自动继承自 MonoBehaviour。

而在进行 MSVR 开发时，您需要将脚本更改为继承自

SynchronizeBehavior, 并且使用 Init 函数、Release 函数、SyncUpdate 函数和 LateSyncUpdate 函数代替 MonoBehaviour 中的 Start 函数、OnDestroy 函数、Update 函数和 LateUpdate 函数:

```
public class SyncExample : SynchronizeBehavior {
    //当同步系统初始化时
    protected override void Init() {
        base.Init();
    }

    //当同步系统退出时
    protected override void Release() {
        base.Release();
    }

    //与Update函数功能一致
    protected override void SyncUpdate() {
        base.SyncUpdate();
    }

    //与LateUpdate函数功能一致, 在所有SyncUpdate之后
    protected override void LateSyncUpdate() {
        base.LateSyncUpdate();
    }
}
```

注意, 您不能使用 Time 类中的任何值和方法, 因为它们都是基于机器的本地时间计算的, 这可能会导致不同步的情况发生。

如果您需要游戏对象的移动保持平滑, 如我们经常做的:

```
void Update(){
    transform.Translate(Vector3.forward * speed * Time.deltaTime, Space.Self);
}
```

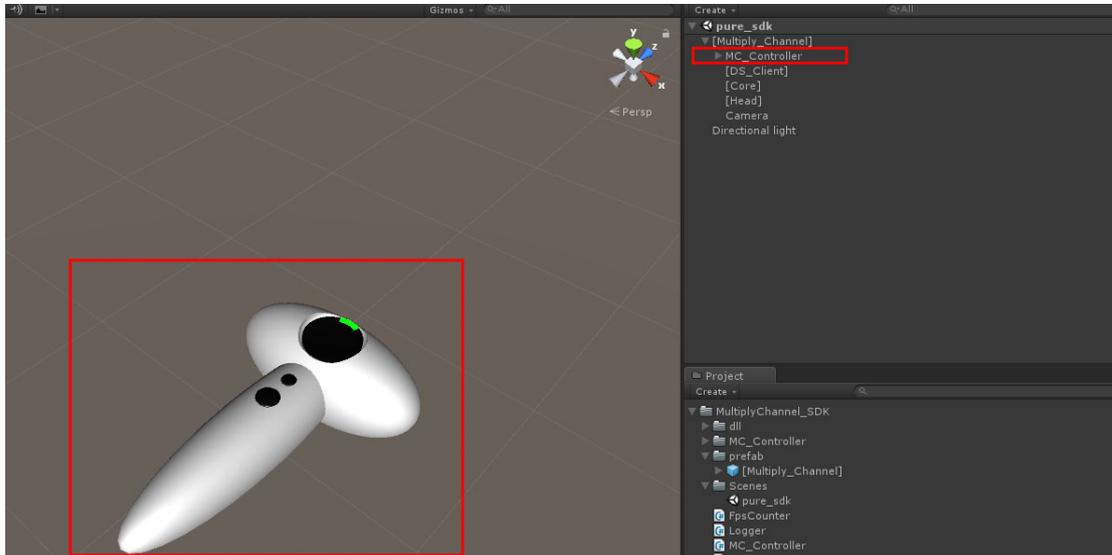
在 MSVR 开发中, 我们用如下代替:

```
//与Update函数功能一致
protected override void SyncUpdate() {
    transform.Translate(Vector3.forward * speed * deltaTime, Space.Self);
}
```

SynchronizeBehavior 类已经有一个值 “deltaTime”, 该浮点数储存了同步的更新时间差, 以确保所有客户端保持同步。

## 6.6. 虚拟控制器

MSVRSDK 包含了一个通用的虚拟控制器，并且包含在了预制体中：

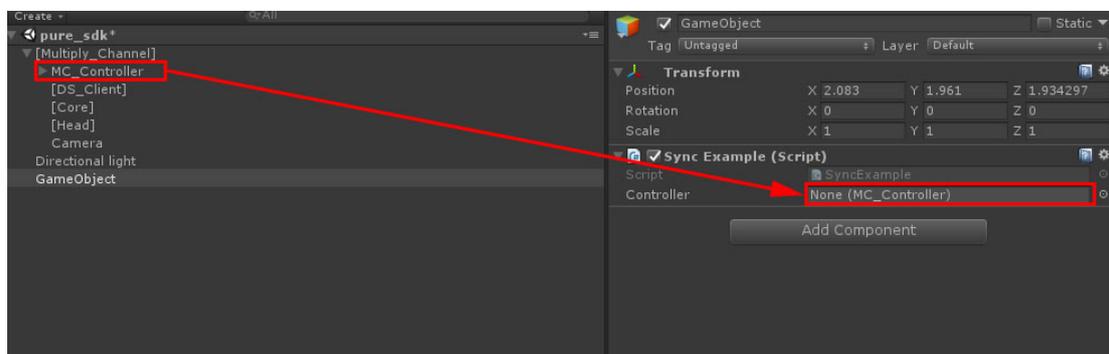


该虚拟控制器包含三个按钮，一个摇杆和 6 自由度信息，其输入是通过 MSVR 的控制系统模块中虚拟手柄映射设置的。

为了在您自己创建的脚本中使用虚拟控制器，首先您要在脚本中申明该类：

```
using MultiplyChannel;  
  
public class SyncExample : SynchronizeBehavior {  
    public MC_Controller controller;  
}
```

然后将 MC\_Controller 脚本连接到您的脚本中：



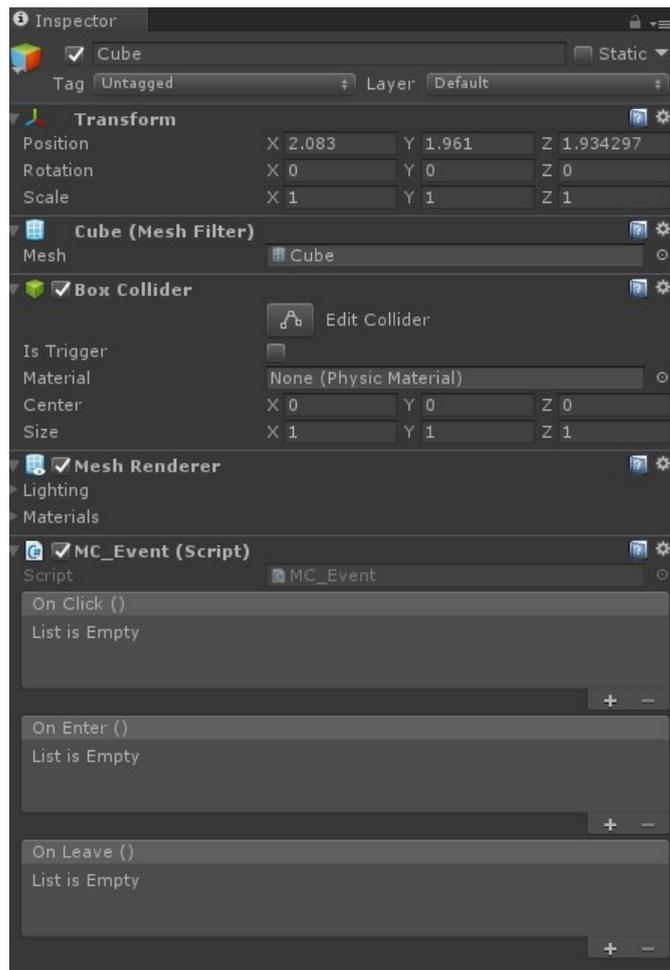
之后，您可以通过编码来获取虚拟控制器的数据：

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using MultiplyChannel;

public class SyncExample : SynchronizeBehavior {
    public MC_Controller controller;
    protected override void SyncUpdate() {
        //获取控制器摇杆
        controller.GetAxis();
        //查询扳机按钮是否被按下
        controller.TriggerButtonDown();
        //查询菜单按钮是否被按下
        controller.MenuButtonDown();
        //获取控制器位置
        Vector3 pos = controller.transform.localPosition;
        //获取控制器旋转
        Vector3 rot = controller.transform.localRotation.eulerAngles;

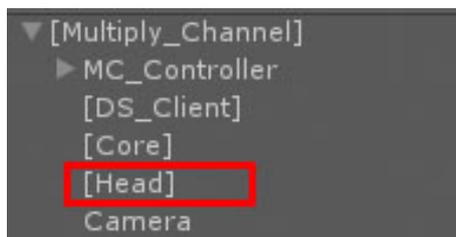
        //注册点击物体事件
        controller.onClickObj += (GameObject target) => { };
        //注册射线进入物体事件
        controller.onEnterObj += (GameObject target) => { };
        //注册射线移出物体事件
        controller.onLeaveObj += (GameObject target) => { };
    }
}
```

您也可以通过向物体挂载碰撞盒和 MC\_Event 脚本来接受虚拟控制器的射线事件：



## 6.7. 头部追踪器

MSVRSDK 包含了一个通用的头部追踪器，并且包含在了预制体中：



您可以直接获取该游戏对象的位置和旋转来获取头部追踪器的位置和旋转。

## 7. 常见问题

Q1: MSVR\_Client 运行指示灯正常, 但是 MSVR 无法扫描到客户端怎么办?

A1: 请检查您的网络防火墙设置, 使用 “Ping” 命令查看局域网连接是否有问题, 并在设置后重启软件。

Q2: 屏幕间有接缝无法完全拼接过渡?

A2: 我们的软件属于全屏运行模式, 如果屏幕间出现缝隙, 请重新校准所有投影仪的投射屏幕。