



# STEAM VR™

## Tracking 培训



STEAM® VR  
Tracking 培训

# JSON 文件

# JSON 文件

- 每个对象都包含一个描述自身的文件
  - 此文件为 JSON 格式
  - 由于使用极为频繁，因此我们直接叫它 “JSON 文件”
  - 之前我们已经讨论过传感器数据，但还有更多内容...
- 
- 在 100\_the\_json\_file 中打开 reference\_object.json

# 常见成员

"manufacturer" : "Valve"

"model\_number" : "REF-HMD"

"device\_class" : "controller"

"device\_vid" : 10462

"device\_pid" : 8960

"device\_serial\_number" : "LHR-X"

"render\_model" : "ref\_controller"

- 公司名称
- 产品型号
- “hmd” 或 “控制器”
- USB 供应商 ID
- USB 产品 ID
- 序列号
- 默认渲染模型

# 传感器位置

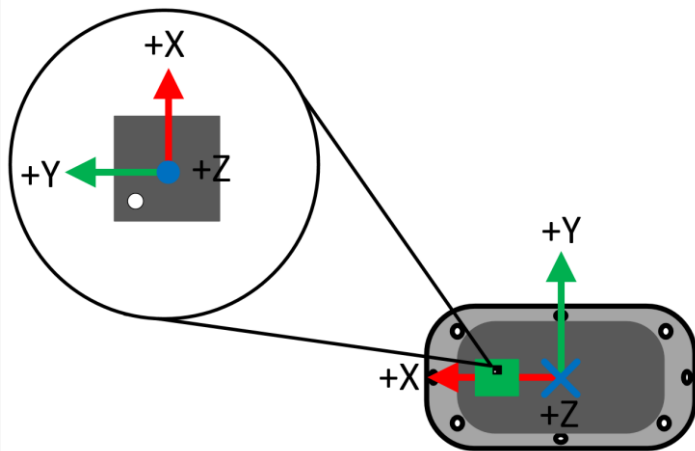
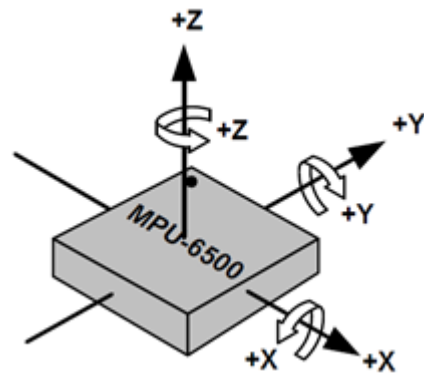
- 包含用于定义传感器的三个数组
  - 位置
  - 方向
  - 通道连接

```
"lighthouse_config" : {  
  "channelMap" : [...],  
  "modelNormals" : [...],  
  "modelPoints" : [...]  
}
```

# IMU 位置

- 存放有 IMU 校准数据
- 在对象的坐标系中定位 IMU

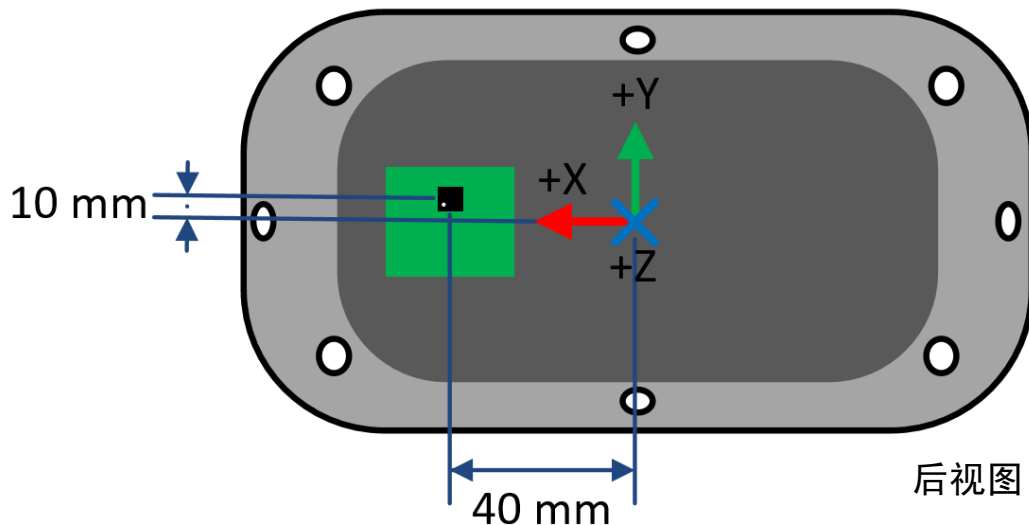
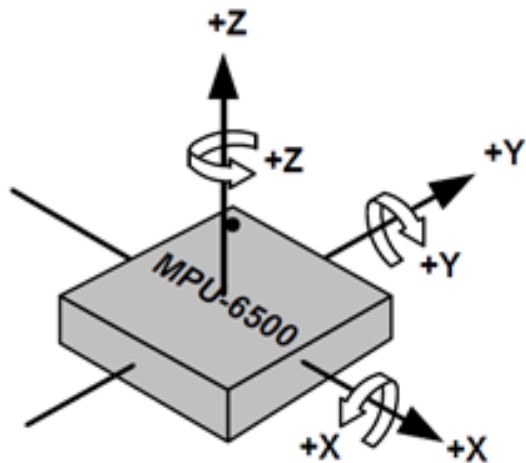
```
"imu" : {  
  "acc_scale" : [ 1, 1, 1 ],  
  "acc_bias" : [ 0, 0, 0 ],  
  "gyro_scale" : [ 1, 1, 1 ],  
  "gyro_bias" : [ 0, 0, 0 ],  
  "plus_x" : [ 1, 0, 0 ],  
  "plus_z" : [ 0, 0, 1 ],  
  "position" : [ 0.0, 0.0, 0.0 ]  
}
```



# IMU 练习

- 将“imu”添加到 add\_imu.json
- 可视化 json 以验证布置
  - 将 sensor\_shape.scad 用作形状

```
"imu" : {  
  "acc_scale" : [ 1, 1, 1 ],  
  "acc_bias" : [ 0, 0, 0 ],  
  "gyro_scale" : [ 1, 1, 1 ],  
  "gyro_bias" : [ 0, 0, 0 ],  
  "plus_x" : [ ?, ?, ? ],  
  "plus_z" : [ ?, ?, ? ],  
  "position" : [ ?, ?, ? ]  
}
```



## 映射到 SteamVR™ 坐标

- “Head” 变量使用对象的坐标描述 SteamVR 坐标系。
- 对于 HMD 和控制器来说各不相同

```
"head" : {  
  "plus_x" : [ 1, 0, 0 ],  
  "plus_z" : [ 0, 0, 1 ],  
  "position" : [ 0.0, 0.0, 0.0 ]  
}
```

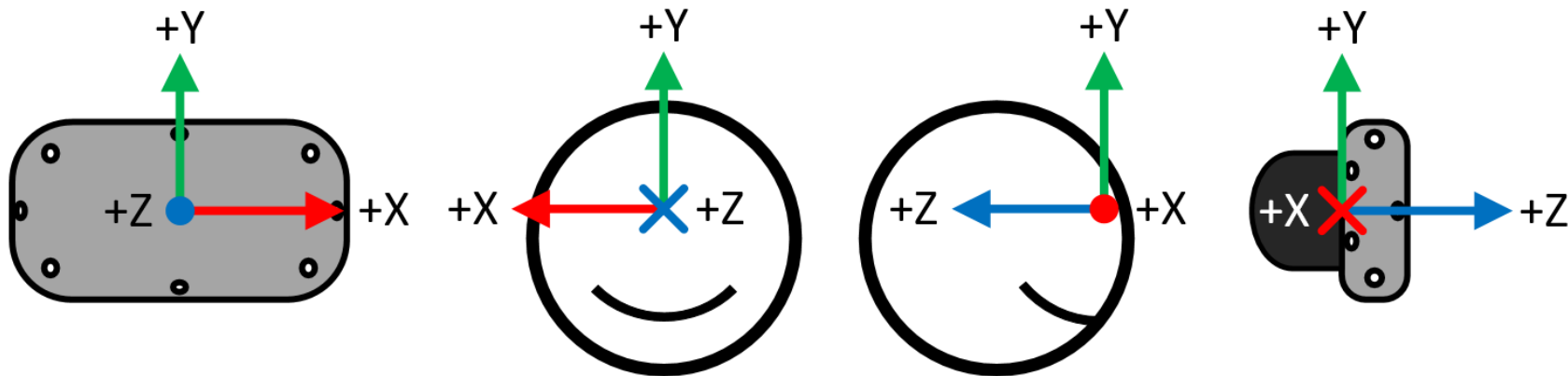


# HMD “head” 坐标

- SteamVR HMD 坐标

- 在两个瞳孔正中间
- +Y 在上方
- +X 在用户右侧
- +Z 在头部中

- 此例中，“plus\_x”、“plus\_z”和“position”是如何映射的？



# HMD 特定成员

```
"direct_mode_edid_vid" : xxxxx
```

```
"direct_mode_edid_pid" : xxxxx
```

- 显示 EDID 供应商 ID
- 显示 EDID 产品 ID
- HMD 需要存储在 JSON 文件中的光学校准数据
- 我们所关注的是定位
- 光学超出了本课的范围
- 使用 HMD? Valve 可以提供帮助!