

中望人工智能三维仿真软件 V2.5 使用说明书



目录

一、 软硬件环境要求	5
1. 处理器	5

2. 显卡	5
3. RAM	5
4. 虚拟内存	5
5. 硬盘	5
6. 分辨率	6
7. 定点设备	6
8. CD-ROM	6
二、 异常处理	6
1. 显卡检测	6
2. 动态库文件缺失检测	7
3. 卡顿检测	8
4. 模型破面检测	9
三、 左侧工具栏	9
1. 基本编辑	9
2. 全局属性设置	14
3. 物体属性设置	15
4. 边界（3D One AI V2.0 版本新增功能）	22
5. 受力与速度	26
6. 关节	28
7. 设置控制器	38
8. 设置电子件模型	39
9. 设置第一人称视角	76

10. 组 (3D One AI V2.0 版本新增功能)	77
11. 连线	78
12. 全局属性 (3D One AI V2.0 版本新增功能)	80
13. 匹配模型	81
四、库资源与编程	84
1. 模型库	84
2. 样例库	85
3. 案例库	86
五、编程与接口说明	87
1. 编程建模	87
2. 编程物理属性	89
3. 编程设置控制器总体设置	89
4. 编程设置控制器——虚拟传感器	100
5. 编程控制器——电子件	112
6. 编程控制器——图像识别	118
7. 编程控制器——语音技术	133
8. 编程控制器——机器学习	137
9. 编程控制器——机械臂	146
10. 编程控制器——控制	153
11. 编程控制器——物理	162
12. 编程控制器——关节	178

13. 编程控制器——人脸识别	181
14. 编程控制器——翻译	188
六、其他	189
1. minibar	189
2. 仿真过程控制	189
3. 仿真设置	190
4. 百度在线识别功能	197
5. Zspace 立体渲染	198
6. 比赛文件定制化处理	199
7. 库模型配置的关节与装配处理	202
8. 积分系统	206
9. 问题信息反馈模块	221
10. 文件锁定功能	223

一、软硬件环境要求

1. 处理器

英特尔奔腾® 4（2GHz 或更高主频）、英特尔至强®、英特尔酷睿™或等效的 AMD®处理器。

2. 显卡

支持 Microsoft DirectX® 9 及以上或 OpenGL3.3 及以上的显卡。

3. RAM

最低配置：1GB

推荐配置：4GB 及以上

4. 虚拟内存

最低配置：1GB

推荐配置：2GB 及以上。

5. 硬盘

最低配置：1GB 可用硬盘空间

推荐配置：8GB 可用硬盘空间及以上

6. 分辨率

最低配置：1024×768 32 位真彩色分辨率

推荐配置：1280×1024、1680×1050、1920×1080 32 位真彩色分辨率

7. 定点设备

3 键鼠标、轨迹球或其他设备

8. CD-ROM

任意速度（仅用于安装）

二、异常处理

1. 显卡检测

在打开软件闪屏阶段，对用户的显卡进行检测，当 OpenGL 低于 3.3 版本，弹出提示框，

提示升级：



点击更新指南即可转到 i3DOne 社区，按照步骤更新驱动即可。



【3D One AI 显卡兼容性问题】——3D One AI使用指南

【社区主编】大麦 2022-02-17 10:55

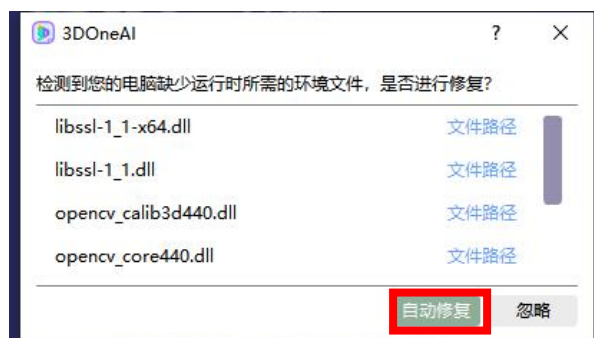
95

0

0

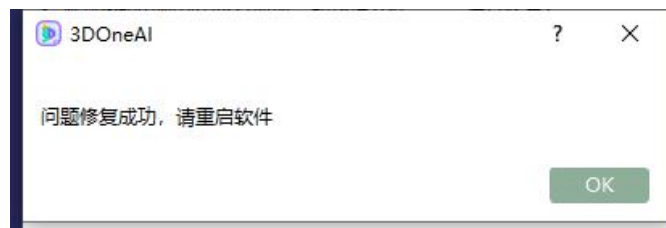
2. 动态库文件缺失检测

当动态库文件缺失时，对应的功能会出现无法使用的问题，因此在软件闪屏阶段会进行动态库文件是否缺失的检测，若出现动态库文件缺失会出现以下提示框：



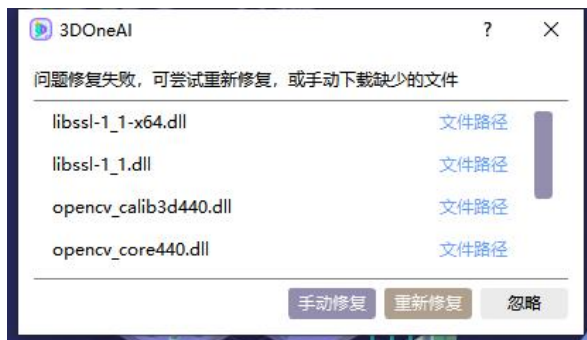
2.1 自动修复

点击自动修复，系统会自动下载缺失的动态文件，下载完成后重启软件即可：



2.2 手动修复

若自动修复失败，可以尝试重新修复或手动修复：



点击手动修复，可下载以下文件：



通过下载链接下载动态库，再放置于缺失文件相应的文件位置即可。

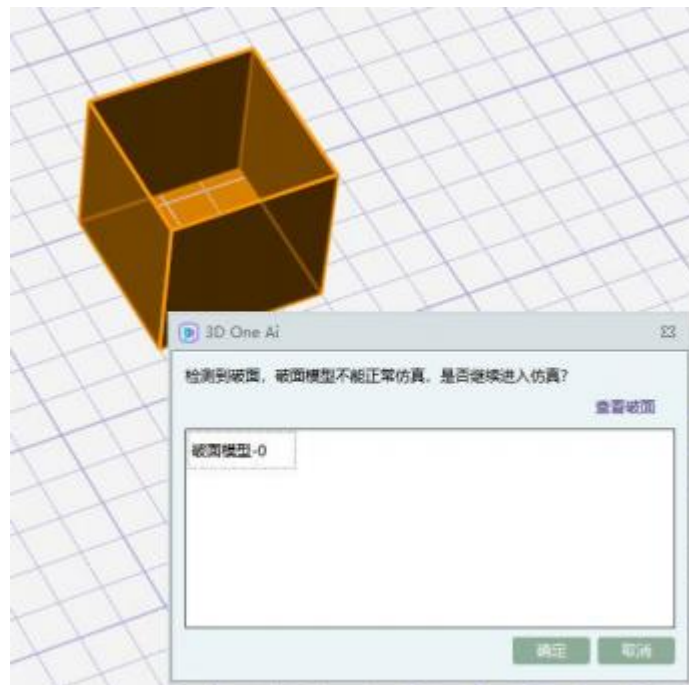
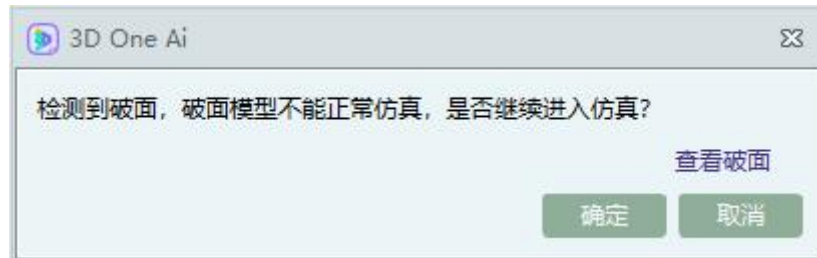
3. 卡顿检测

若仿真加载超过 1min，很有可能出现了加载卡住的异常问题，会出现“关闭软件”按钮，点击即可。



4. 模型破面检测

如果模型存在破面可能在仿真中运动出现问题,因此在仿真加载完成后如果模型存在破面则会弹出提醒,并会高亮显示破面位置。



(破面模型高亮显示)

三、左侧工具栏

1. 基本编辑

3D One AI 中并不强调三维设计部分,因此精简了三维设计部分的功能,推荐在 3D One

中设计好模型后再导入 3D One AI 中，3D One AI 源文件格式为 Z1AI，支持直接打开 Z1，可导入/导出 step、stl、obj、iges、3mf、amf 等通用三维格式。

左侧工具栏的第一项  就是物体的基本编辑功能项，其中设有 15 个基本功能编辑，包括圆角、倒角、抽壳、实体分割、移动、缩放、阵列、镜像、组合编辑、颜色、对其移动、自动吸附、距离测量、线框/着色模式以及隐藏/显示几何体功能。由于基本编辑的功能相对简单，这里只对其中的三个功能进行描述。

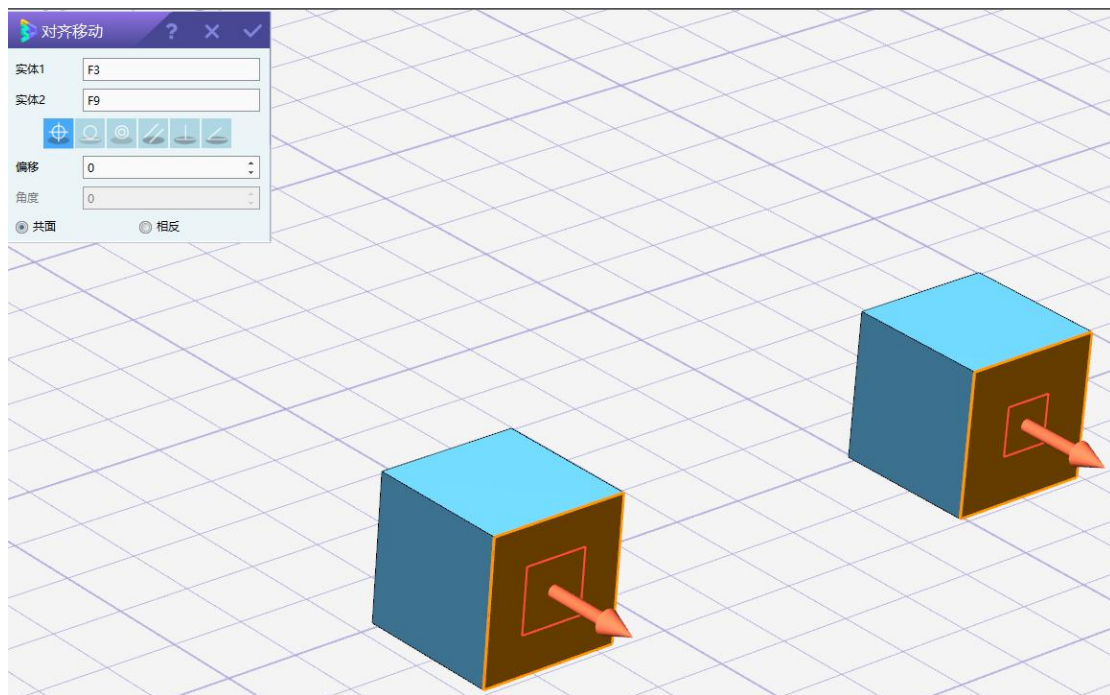
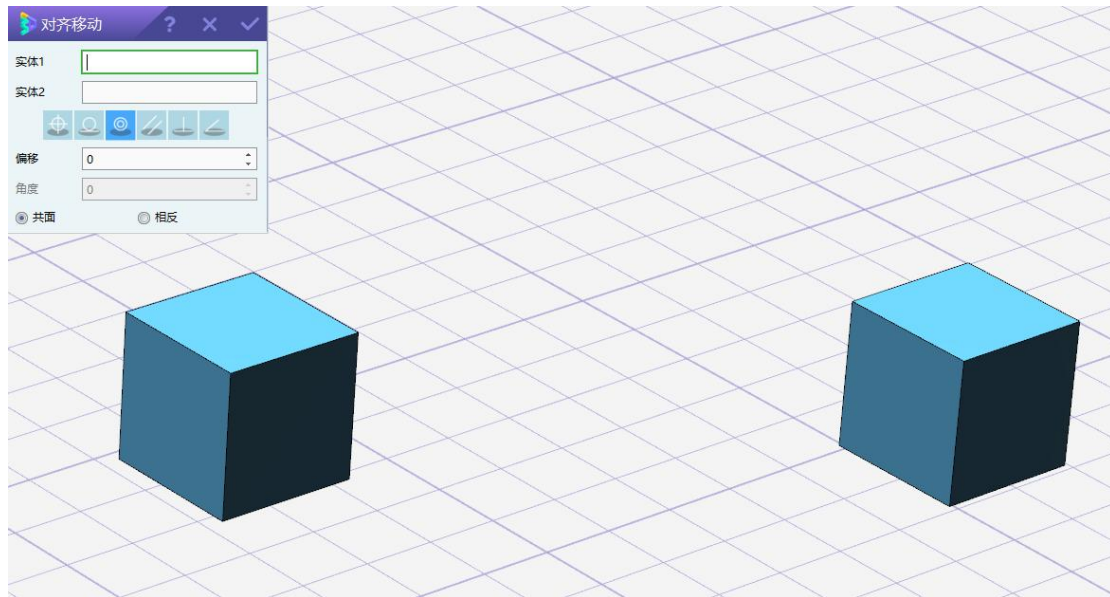
1.1 对齐功能

对齐两个实体

示意图：



简单示例：



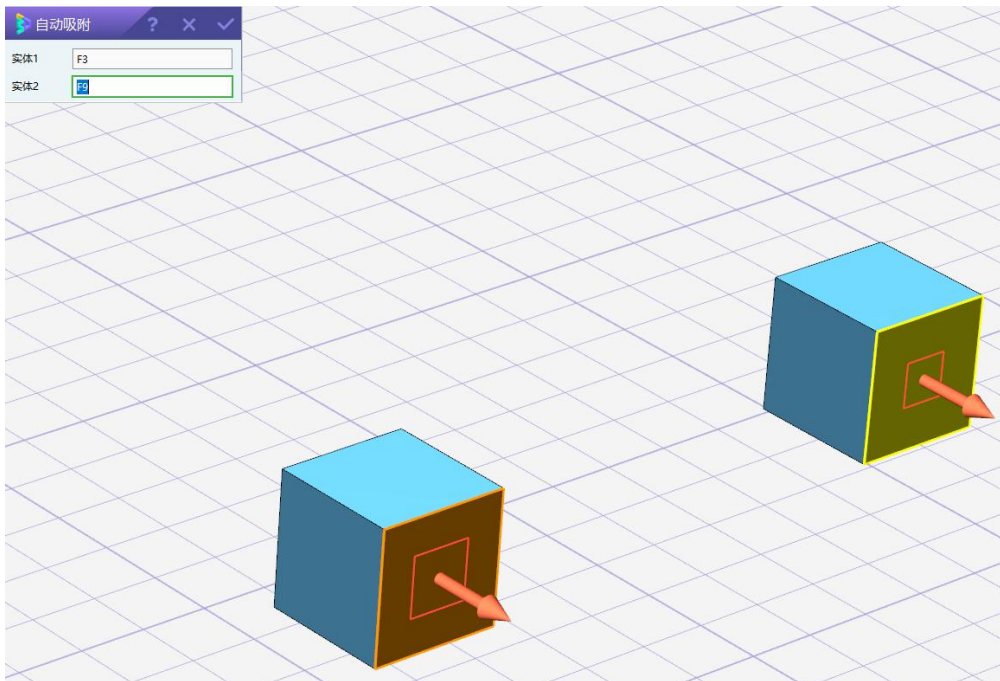
1.2 吸附功能

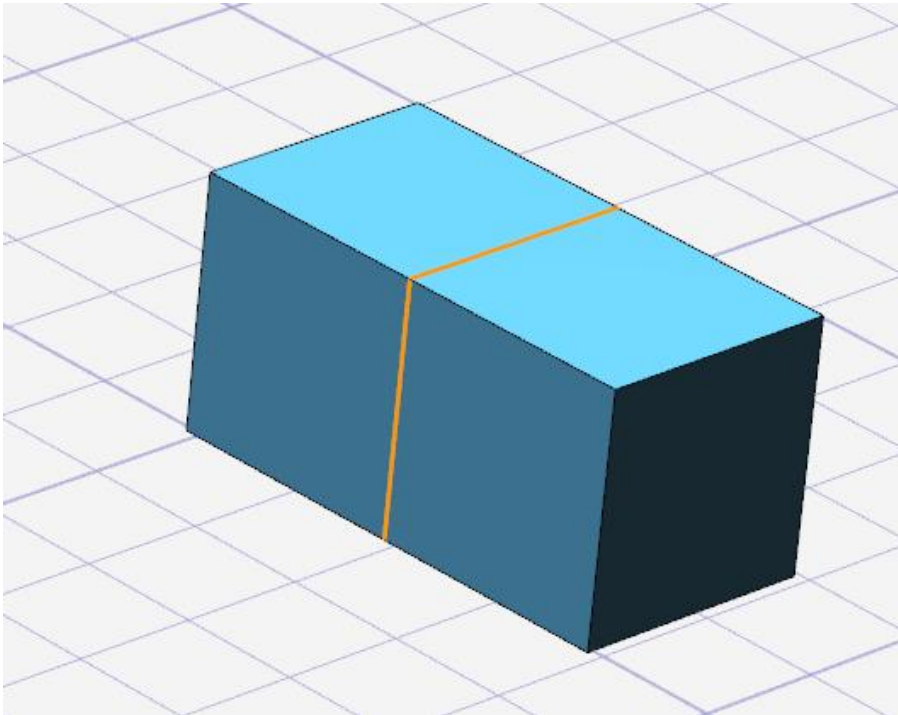
吸附两个物体

示意图:



简单示例：

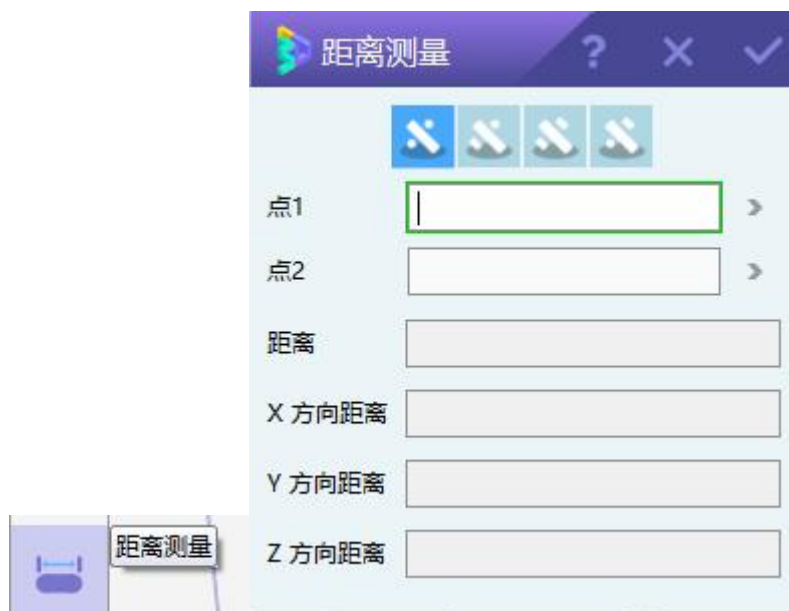




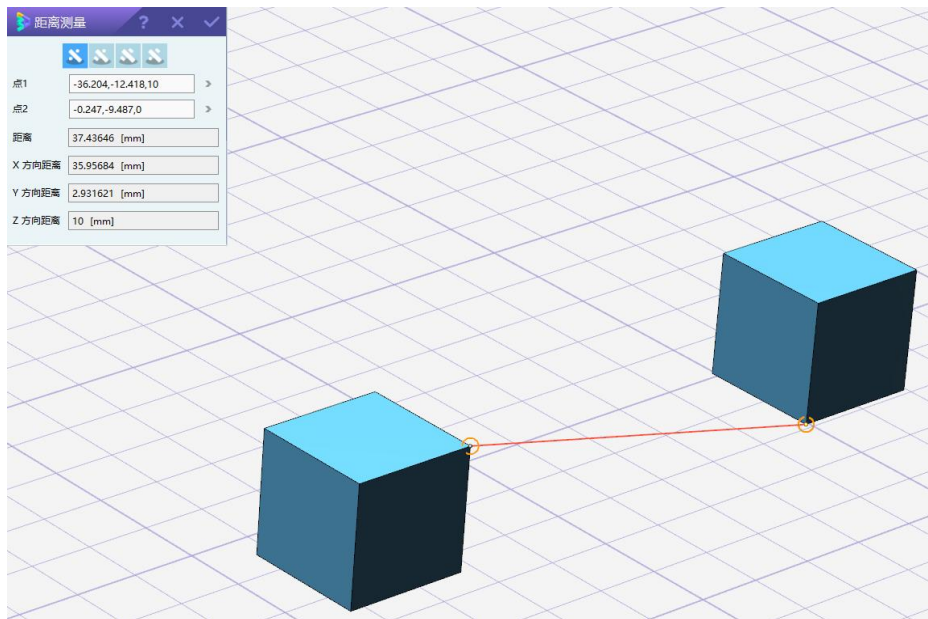
1.3 测量功能

测量两个点之间的距离

示意图:



简单示例: 给出两点见的 XYZ 轴的差值。



2. 全局属性设置

如图一，在全局属性设置中 ，世界属性模拟仿真的环境，匹配了该真实环境下的一些参数信息。世界属性改变，一些默认参数也相应地改变。如地球属性下，重力值默认为-9.8，范围从负无穷到正无穷，负号代表重力方向垂直地面方向向下，正号代表垂直地面方向向上。除了重力设置外，还有一系列高级参数设置，已配置好最佳默认数值，用户也可以根据需求自己修改，以下为高级设置的参数说明。



图一

3. 物体属性设置

物体属性即为单个物体设置物理属性，选择设置实体后，可设置物体的类型、名称以及材料类型，选择的材料不同，实体的一些默认参数也会相应的改变。

示意图：



3.1 高级参数

V2.2 中将高级参数设置默认收起，如果要调整物体的高级参数，如质量、摩擦系数等，点击，在下拉框内即可调整。示意图：



1) 质量

质量的取值范围为 0 到正无穷，默认数值为 0.001kg。通过调整质量也可以调整物体在仿真运动中的稳定性。

2) 摩擦系数

摩擦系数的取值为 $[0, \infty)$ ，默认取值为 0.35。通过修改摩擦系数可以改变物体在运动过程中受到的阻力。例如，给两个质量相同的六面体向同一方向施加一个相同的速度，摩擦系数大六面体运动的距离会更短。

3) 弹性系数

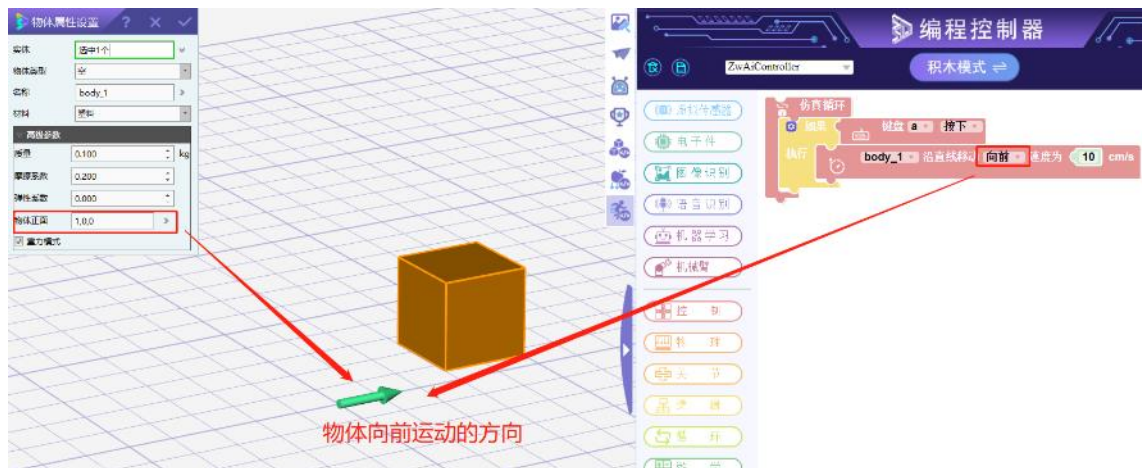
弹性系数取值范围为 $[0, 1]$ ，默认取值为 0.5。例如，两个质量相同的小球从同一高度做自由落体运动，弹性系数大的小球在落地后反弹的高度会比弹性系数小的小球高。

4) 物体正面

通过修改物体正面，可以修改物体向前运动时的方向。

示例:

当六面体的正面设置为 (1, 0, 0) 时, 编程设置使六面体向前运动, 则六面体运动的方向与物体的正面方向相同。



5) 重力模式

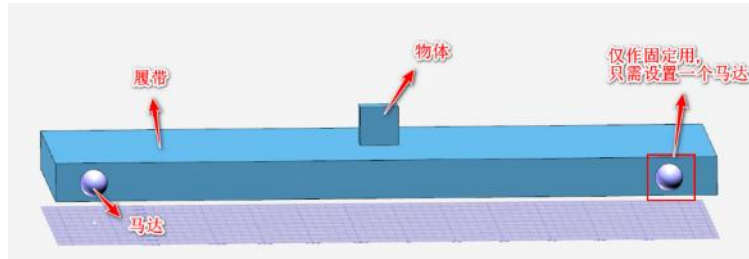
重力模式默认开启, 若选择取消, 则代表即使在重力场中, 该物体仍按照无重力的方式仿真。也可以设置物体的正面; 物体类型对物体进行一些特殊设置。物体所有参数均可按照物体的实际物理意义进行设置。

6) 添加履带

履带可作为一个物体的属性添加到物体上。如果用户需要设置履带, 可以按照如下流程进行设置:

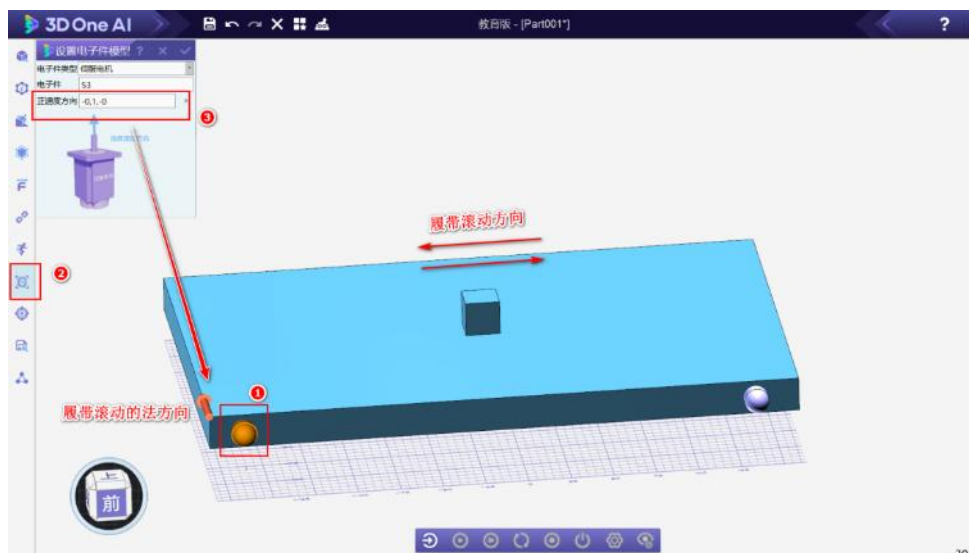
首先需要准备两个模型: 作为履带的物体和作为马达的物体。

马达和另一个固定用的物体可设置物体属性为地形, 主要用于固定履带。(让履带浮空, 否则履带下表面也会滚动从而让履带进行移动)

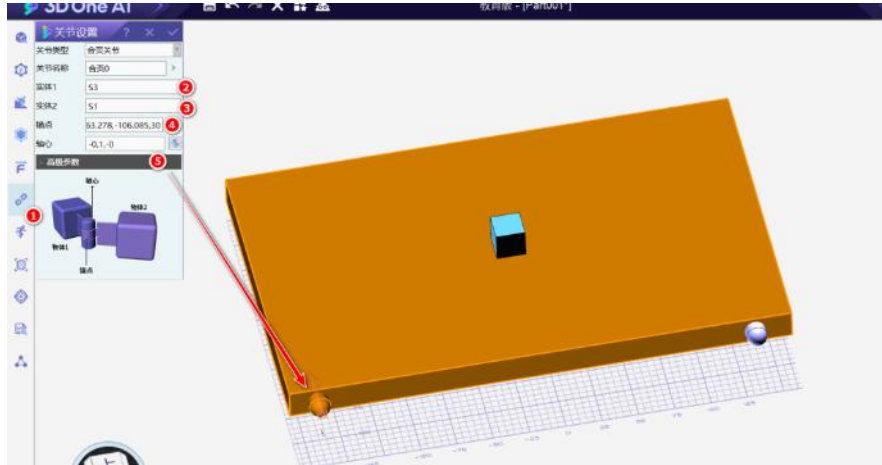


将作为履带的物体选中，在物体属性设置界面勾选上添加履带。

将作为马达的物体选中，设置电子件类型为马达、伺服电机、步进电机、舵机等类型的电子件，方向设置为履带滚动方向的法向方向，遵循右手螺旋定则。



将马达与履带物体之间添加合页关节，关节的方向为马达的方向，也就是履带滚动的法向方向。



此时控制马达转动，则履带就会进行滚动。

3.2 物体类型：仅仿真

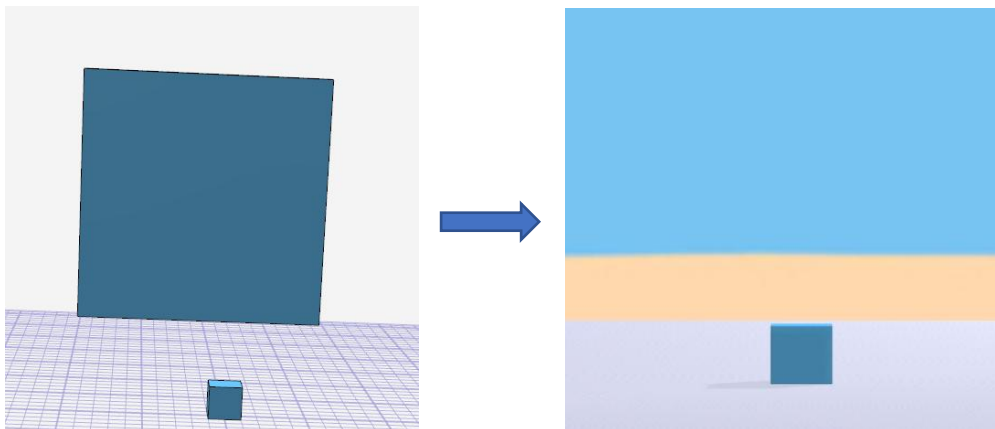
在选择物体类型时，有七种类型可供选择，分别为空、动画、仅显示、轨迹、仅仿真、地形以及跟随显示。

其中仅仿真类型，在仿真环境下表现出仿真特性，但是不显示仿真画面。

示意图：



仿真示意图：建模界面与进入仿真界面对比



3.3 物体类型：地形类型

地形类型，在仿真环境下表现出物体属性，但是拥有绝对的位置，不会受到改变。

示意图：

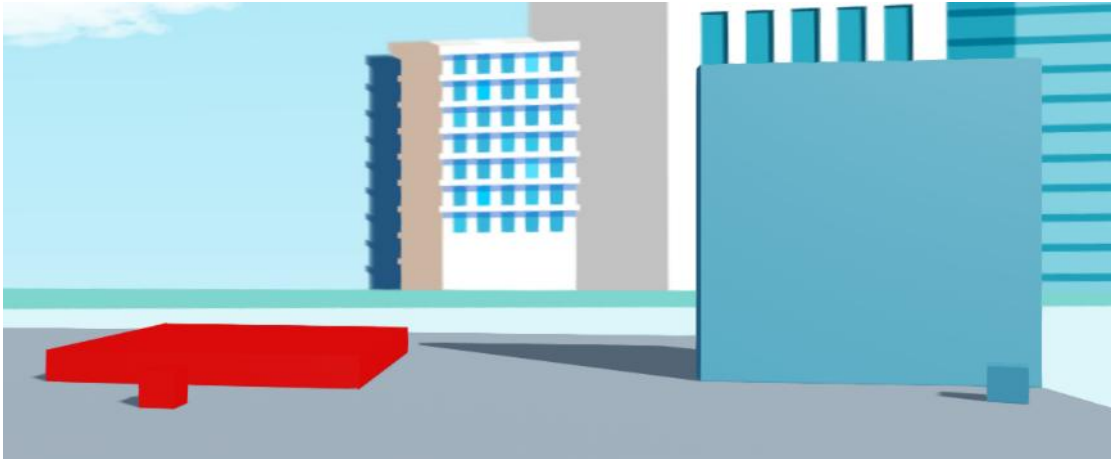


仿真示意图：地形类型与空类型物体在受到碰撞时的仿真对比。如下图所示。左边红墙为空类型，右边蓝墙为地形类型。它们同时受到一个物块的撞击，撞击后可以看到为空类型的红墙倒塌，而为地形类型的蓝墙没有：

仿真前：



仿真后：

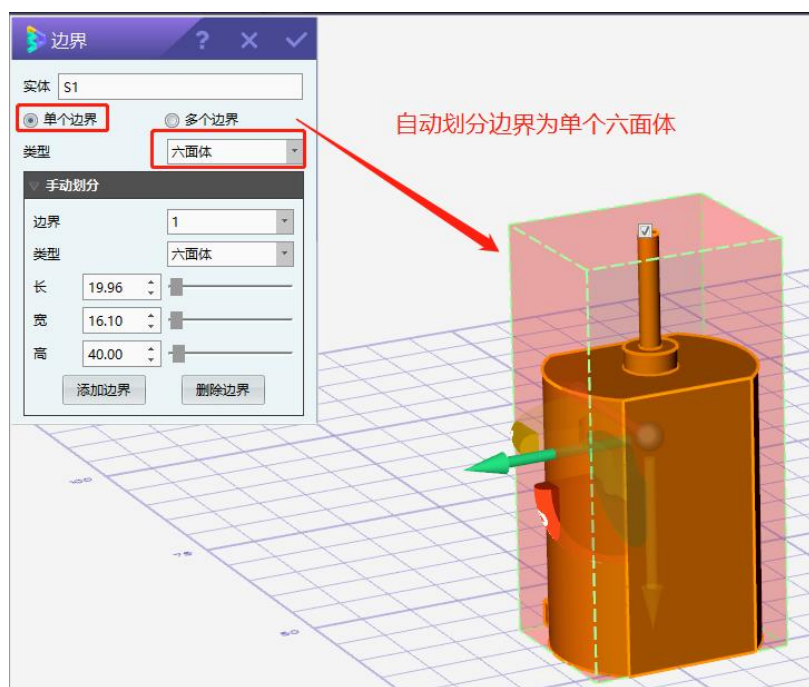


4. 边界 (3D One AI V2.0 版本新增功能)

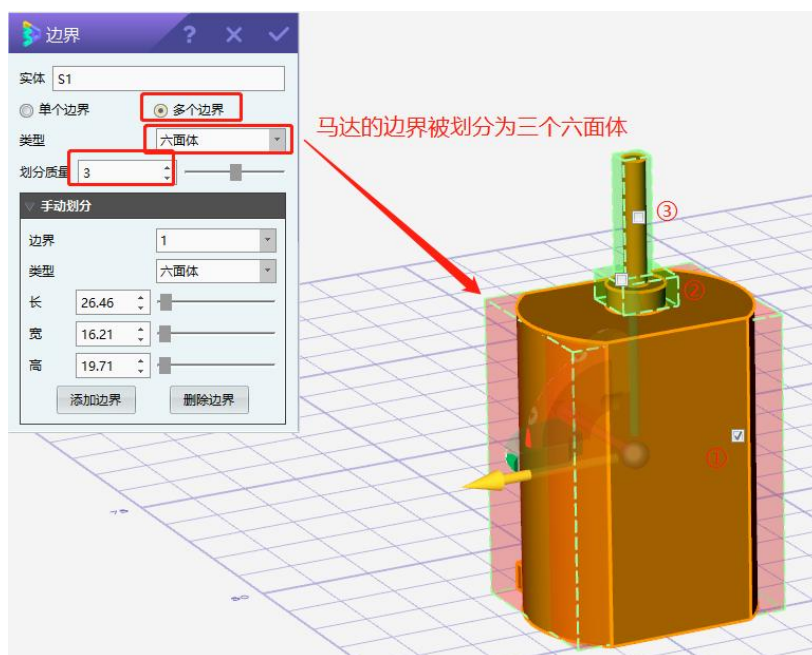
由于原生的物理引擎对三角面片的仿真检测碰撞支持并不好,导致了許多仿真异常现象如物体异常抖动、物理现象奇怪、多次仿真结果非常不一致等。几何划分功能是通过通过对原生物理引擎支持得较好的内置几何体对物体进行划分,生成一个几何体碰撞边界进行仿真,仿真效果更好。

点击左侧工具栏的第四个按钮  即可开始对物体进行边界划分。 选择了需要划分的物体后,默认先进入自动划分。可以选择是进行单个边界划分还是多个边界划分,也可以选择划分类型是六面体、圆柱体还是球等。当选择多个边界划分时,还要进一步选择划分质量,即将待划分物体具体划分为多少个边界。

以马达电子件为例,当选择单个边界、六面体类型划分划时,马达被简单划分为一个六面体:

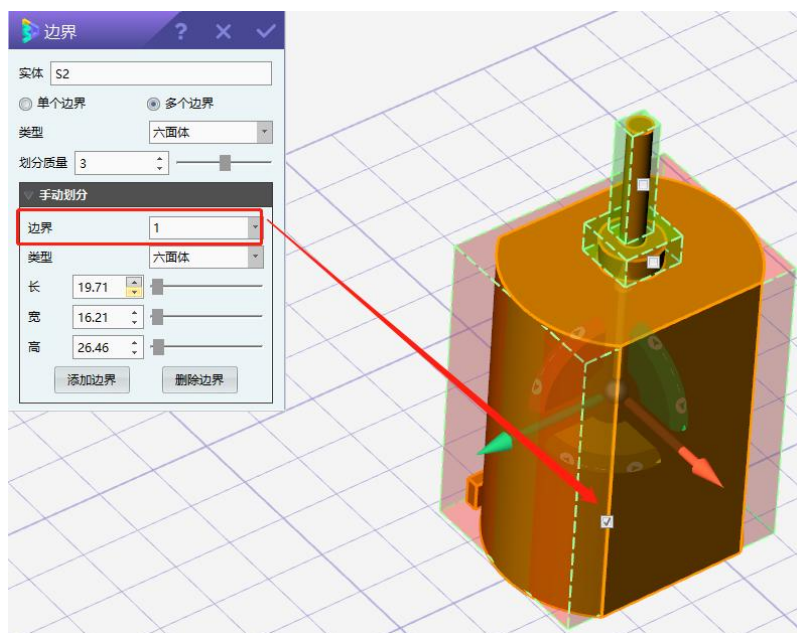


当选择多个边界、划分类型为六面体，划分质量为 3 时，马达被划分为三个六面体，
可以看到这样划分的马达边界更加精细：

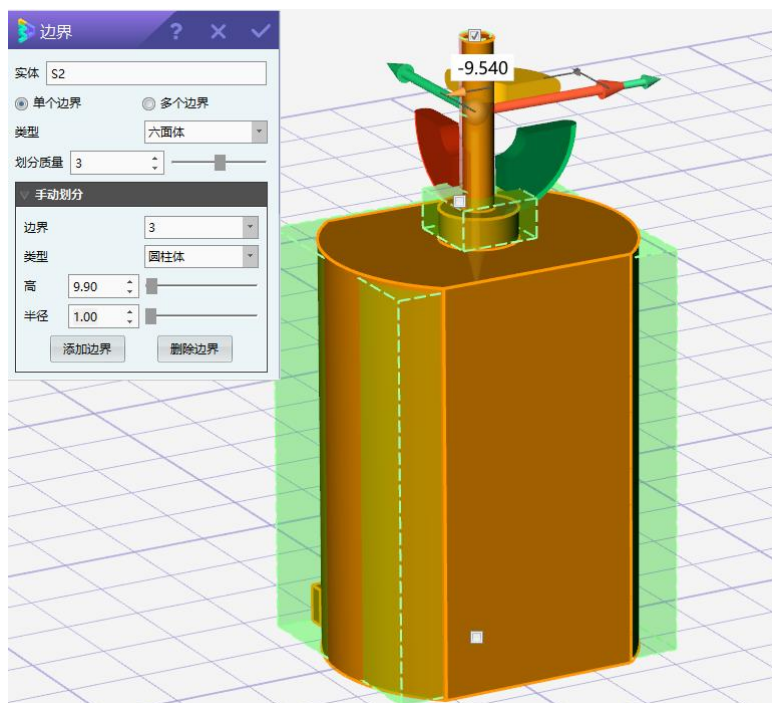


若此时对自动划分结果仍然不满意，可以在手动划分区域继续对物体的边界进行微调。
继续以上图马达边界为例，在多个边界划分的情况下，以通过【边界】下拉框选择具体的划分边界进行编辑，可以修改划分类型、尺寸、位置等，使得划分出的几何边界与实际物体接

近，使得碰撞边界比较准确，仿真效果也会比较好。



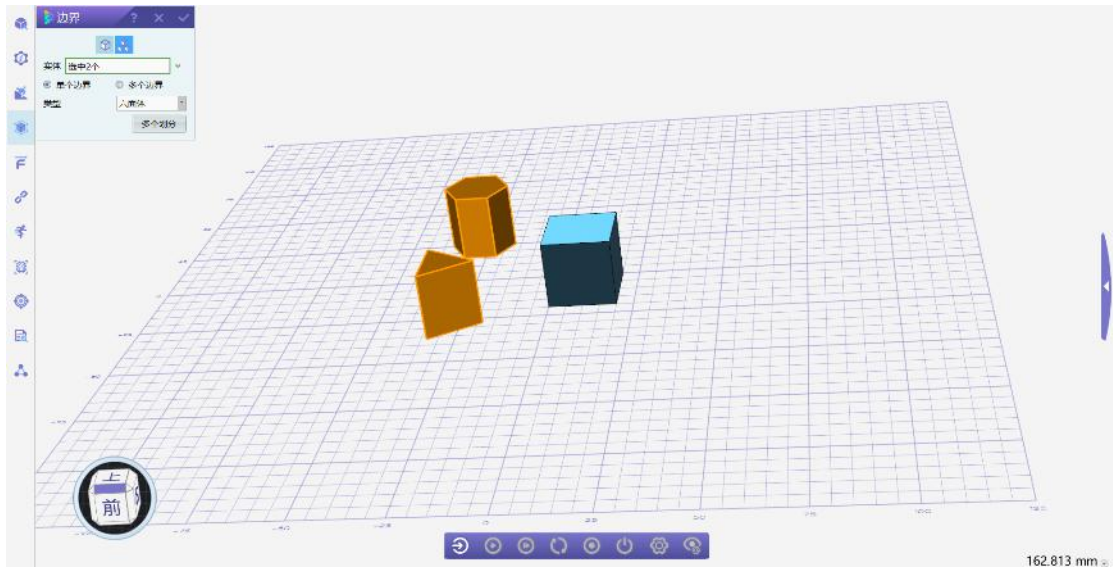
比如，第三个划分边界为六面体，与原有物体的几何形状不太接近，可以下拉【边界】框改为 3，将类型改为圆柱体，就可以对第三个边界进行大小和位置的调整。



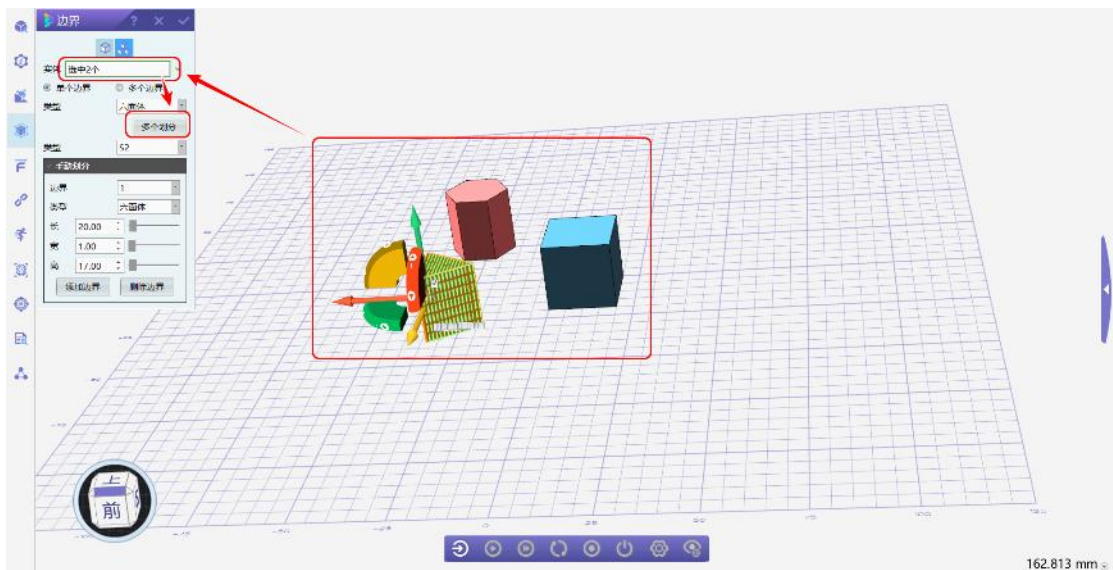
也可以同时选中多个物体进行划分的功能，减少用户的重复操作。

具体流程如下：

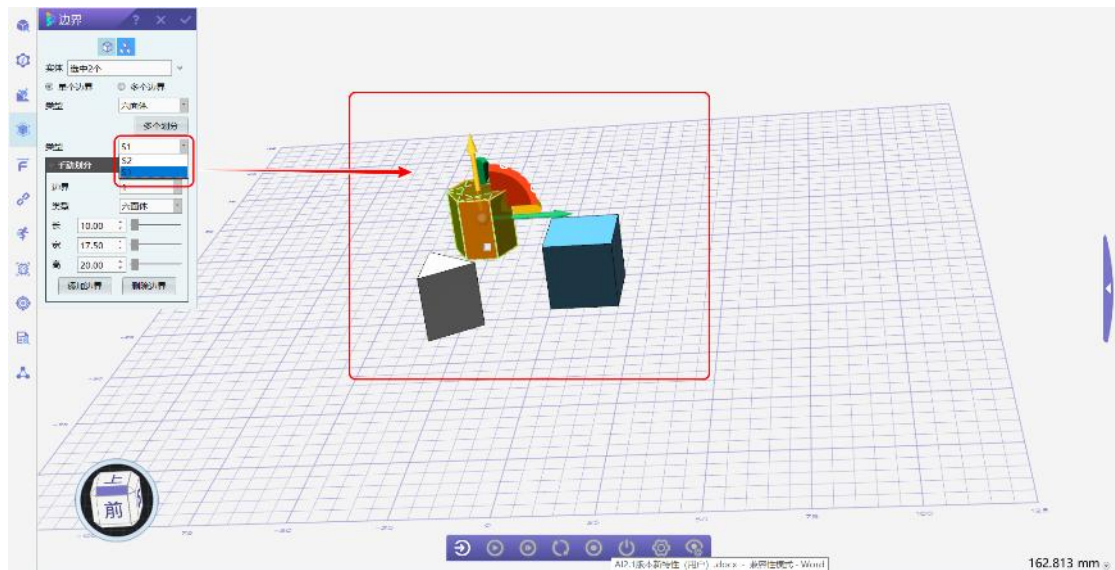
- 1、打开边界面板，将标签切换为多个（默认为单个）。



- 2、框选或者点击选择实体，点击多个划分，即可进入具体的划分界面。



- 3、类型栏处显示的是当前编辑的实体，可切换编辑不同的实体的边界。



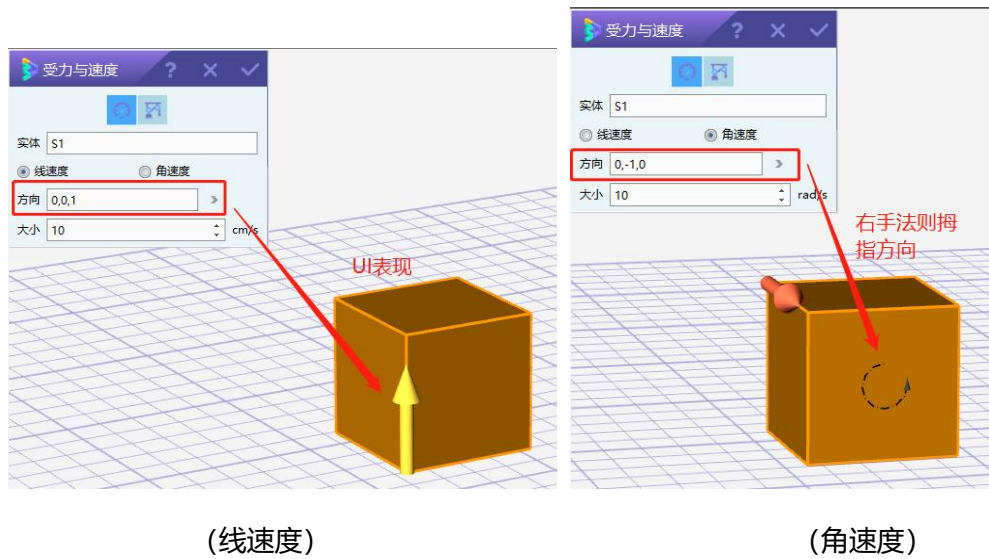
4、之后的划分操作就与单体划分相同，通过编辑边界的类型以及修改对应的参数来为物体添加边界。

5. 受力与速度

点击左侧工具栏的受力与速度按钮 ，功能分为【受力】与【速度】两个子功能，可以通过命令中的 tab 图标进行切换。受力与速度功能为目标物体施加受力或者速度，使物体在力或速度的作用下进行运动仿真。

5.1 速度

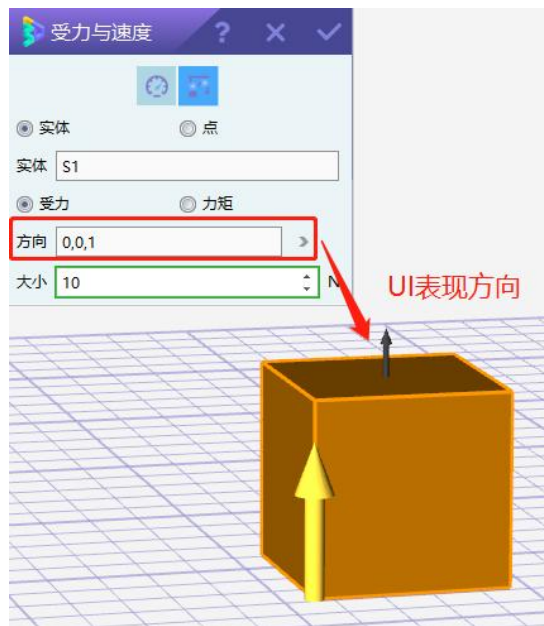
速度分为线速度与角速度，大小单位均为 m/s。角速度遵循右手法则，对话框中的方向代表右手的大拇指方向，实际物体运动方向则是右手法则的四指指向的方向。线速度与角速度的 UI 表现形式为虚线+箭头。线速度是指仿真时沿着指定方向直线运动，角速度则是沿着所选方向作为轴心旋转运动。



5.2 受力

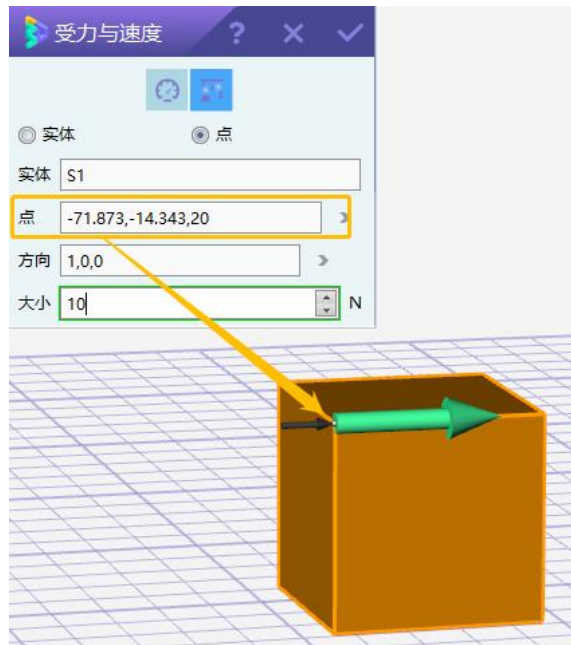
受力分为在实体上施加某一方向的平均力或实体某点上的力，大小单位为 N，UI 表现为实线+箭头。

当选择在实体整体上施加力时，软件会自动计算实体的质心，实际上是在物体质心上选择施加线性力或力矩，力矩的方向遵循右手法则，类似于角速度。



当选择在点上施加力时，即在某一点上施加力，此时不提供力矩的设置，仅可以为物体

上的某点施加一个线性力。



6. 关节

关节即是模型与模型之间的关联与约束，点击左侧工具栏的关节按钮 ，可以看到有两个子选项，分别为智能关节和关节设置。

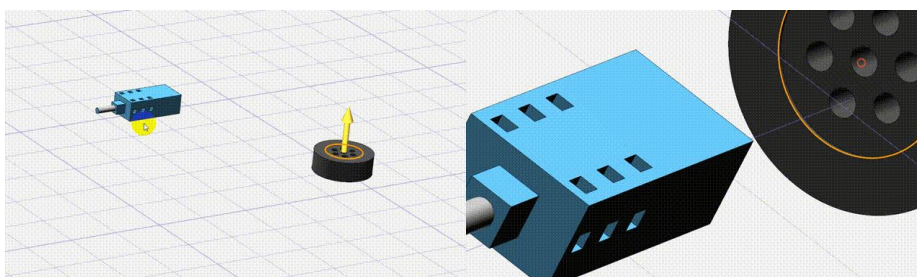
6.1 智能关节 (3D One AI V2.0 版本新增功能)

通过智能关节可以在对两个物体的轴向以及移动点的对齐过程中，同时配置关节。智能关节中关节的自动生成只针对简单常用的关节类型如合页关节、活塞关节、球窝关节、插销关节、成组，也可以选择不生成关节，仅移动模型。

示意图：

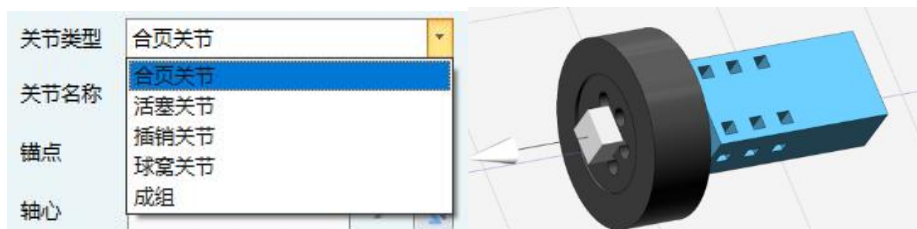


简单示例:



(通过面选择两个物体的对齐轴向)

(通过点选择两个物体的对齐点)



(选择生成的关节)

(确认命令后移动物体并生成关节)

6.2 关节设置

点击第二个子选项关节设置, 可以对物体与物体之间的关节属性进行设置, 在 3D One AI

2.0 中一共提供了 8 种关节类型, 不同的关节设置会导致不同的约束效果。

示意图:



6.2.1 关节类型设置

3D One AI 2.0 中提供了 8 种关节类型设置, 分别为合页关节、铰链关节、活塞关节、插销关节、球窝关节、万向轮关节、PR 关节和 PU 关节。该版本删除了 1.0 版本中的固定关节, 新增了组合命令替代 (详见下面的**小节 11 关于组的描述**)

6.2.1.1 合页关节

合页关节会约束两个实体的某一部分始终在同一位置并且它们始终以铰链轴联接在一起。给其中一个实体施加作用力, 在合页关节的约束下, 受力的实体会绕铰链轴做运动, 约束物体之间只有绕轴心转动的自由度, 典型的合页关节有门、风扇等。



关节名称: 为设置的关节命名;

实体 1: 需要约束的实体;

实体 2: 需要和实体 1 进行关节设置的另一个实体;

锚点: 实体与实体之间关节设置的连接点, 一般是沿着轴心的中心点;

轴心: 设置合页关节的轴心方向;

6.2.1.2 铰链关节

铰链关节可以看成是两个合页关节以不同的轴心串联连接在一起的关节。铰链关节包含一个锚点和两根轴（两根轴相互垂直），轴心 1 与实体 1 相关，轴心 2 跟实体 2 相关联。如小车的前车轮，如果实体 1 是车的底盘，轴 1 则是转动轴，实体 2 是车轮，轴 2 则是车轮轴。轴 1 控制轮子进行转向，使车子可以左右转向，轴 2 则使轮子旋转，使车子前进。



关节名称: 为设置的关节命名;

实体 1: 需要约束的实体, 即物体 1;

实体 2: 需要和实体 1 进行关节设置的另一个实体, 即物体 2;

锚点: 实体与实体之间关节设置的连接点, 一般是轴心 1 与轴心 2 的交点;

轴心 1: 设置实体 1 铰链关节的轴心方向 1;

轴心 2: 设置实体 2 铰链关节的轴心方向 2;

6.2.1.3 活塞关节

活塞关节类似于合页关节, 但它可以绕着平移轴移动, 也就是除了绕着轴心的旋转的自由度之外, 还有沿着轴心的移动的自由度。



关节名称: 为设置的关节命名;

实体 1: 需要约束的实体;

实体 2: 需要和实体 1 进行关节设置的另一个实体;

锚点: 实体与实体之间关节设置的连接点, 一般是沿着轴心的点;

轴心: 设置活塞关节的轴心方向;

6.2.1.4 插销关节

插销关节将实体 1 和实体 2 约束联接起来, 同时还约束两个实体运动的方向, 物体之间只有沿着轴心运动的自由度。



关节名称: 为设置的关节命名;

实体 1: 需要约束的实体;

实体 2: 需要和实体 1 进行关节设置的另一个实体;

轴心: 设置插销关节的轴心方向;

6.2.1.5 球窝关节

球窝关节可以使得一个实体 (球) 约束在另一个实体 (窝) 的同一位置处, 实体 (球) 可以在实体 (窝) 的约束下作旋转、环转以及屈伸运动等, 例如人体的肩关节就是典型的球窝关节。



关节名称：为设置的关节命名；

实体 1：需要约束的实体；

实体 2：需要和实体 1 进行关节设置的另一个实体；

锚点：实体与实体之间关节设置的连接点，一般是中心点。

6.2.1.6 万向轮关节

万向轮关节就像是一个球窝关节，但是多了一个额外约束自由旋转的角度。万两轮关节的两根轴必须是正交的，两个实体旋转方向的正交性和两根轴的正交性是相同的。在万向轮关节的约束下，实体 1 发生扭动，实体 2 也会同时扭动。



关节名称：为设置的关节命名；

实体 1：需要约束的实体；

实体 2：需要和实体 1 进行关节设置的另一个实体；

锚点：实体与实体之间关节设置的连接点，一般是轴心 1 与轴心 2 的交点；

轴心 1：设置实体 1 转动轴的轴心方向 1；

轴心 2：设置实体 2 转动轴的轴心方向 2；

6.2.1.7 PR 关节 (插销-合页关节)

PR 关节即插销关节和合页关节的结合体，即可以沿着轴心 P 进行转动（合页），也可以沿着轴心 R 进行移动（插销）。



关节名称：为设置的关节命名；

实体 1：需要约束的实体；

实体 2：需要和实体 1 进行关节设置的另一个实体；

锚点：实体与实体之间关节设置的连接点，一般是轴心 P 与轴心 R 的交点；

轴心 R：设置合页轴心方向 R；

轴心 P：设置插销轴心方向 p；

6.2.1.8 PU 关节（插销-万向轮关节）

PU 关节是插销关节和万向轮关节的结合，它提供一个自由度的平移（插销）和两个自由度的旋转（万向轮）。轴 1 为万向轮的 1 号轴，轴 2 为万向轮的 2 号轴，轴 3 是插销关节的轴。



关节名称：为设置的关节命名；

实体 1：需要约束的实体；

实体 2：需要和实体 1 进行关节设置的另一个实体；

锚点：实体与实体之间关节设置的连接点，一般是三个轴心的交点；

轴心 1：设置万向轮关节的轴心方向 1；

轴心 2：设置万向轮关节的轴心方向 2；

轴心 3：设置插销关节的轴心方向 3；

6.2.2 高级参数设置 (3D One AI V2.0 版本新增功能)

在该版本中，新增了对关节的高级参数进行设置的功能。如对合页关节可设置关节的松紧度、允许转动的关节角度范围、关节的初始值。

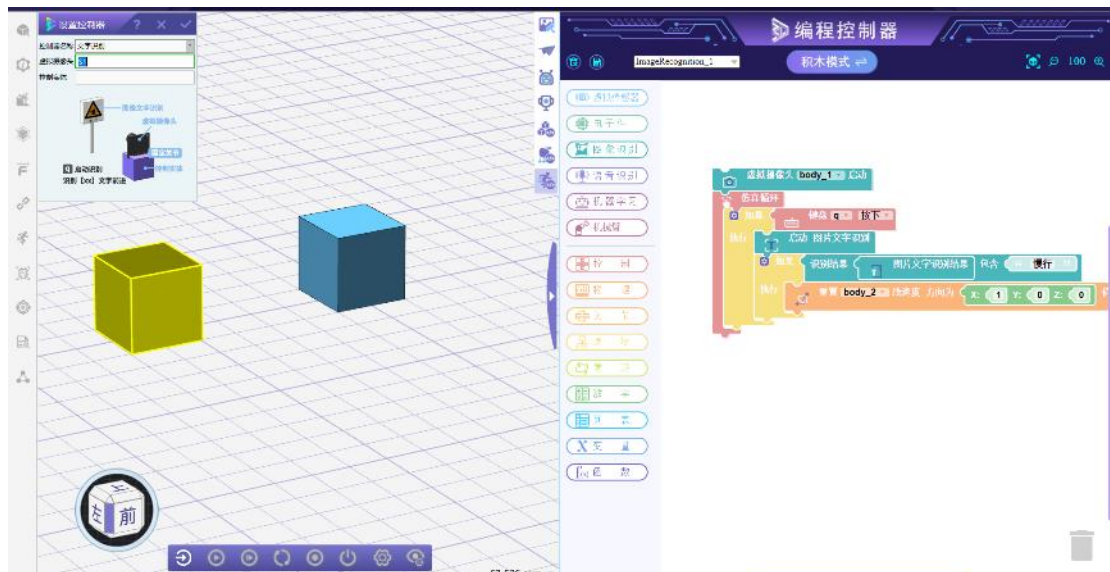
示意图：



7. 设置控制器

控制器设置可以为物体设置一个运动控制逻辑，即为物体添加一段控制程序，该控制程序通过封装为命令的形式展现给用户，用户只需通过界面操作及简单修改参数即可添加控制程序，从而实现在仿真环境中逻辑运动。

软件提供了 7 种控制器：自动循迹、自动避障、文字识别、移动控制、语音识别、马达控制、循迹避障控制器。以文字识别控制器为例：用户首先通过【设置电子件模型】将一个实体设为“虚拟摄像头”，再点击【设置控制器】选择虚拟摄像头和需要被控制的实体，设置相关参数后点击确定按钮，右边栏会自动打开【编程设置控制器】页面，用户可以直接修改文字识别控制器的源码程序进行参数调整，保存程序后进入仿真环境即可实现“当电脑摄像头识别到某种文字后，被控制实体做出特定动作”的控制逻辑。



8. 设置电子件模型

设置电子件模型主要为了控制器编程服务, 软件会检测当前文件是否存在电子件作为控制编程的对象, 当用户没有设置电子件或者没有从模型库中插入电子件时, 进入控制器编程积木块会提示无电子件。

软件提供的硬件设置类型有风扇、触碰传感器、开关、LED 灯、舵机、马达、循迹传感器、双向马达、电子显示屏、距离传感器、力传感器、光线传感器、颜色传感器、虚拟摄像头、转盘、位置传感器、灰度传感器、机械臂、插销夹爪。



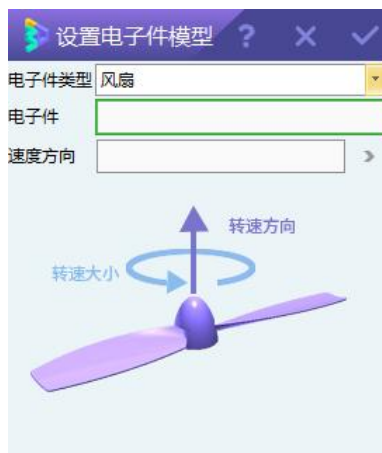
(当未设置或不存电子件时, 提示无电子件编程对象)

8.1 风扇

风扇主要与马达连接到一起工作使用, 马达使其具备转动的效果。

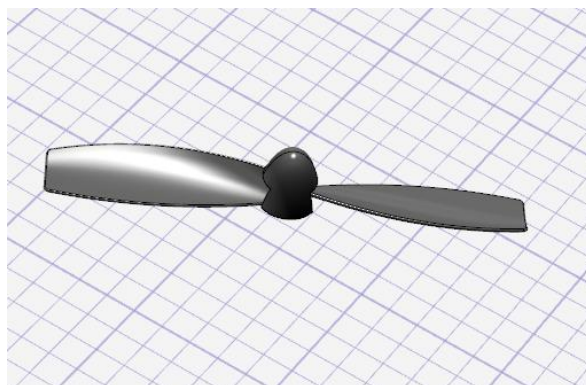
电子件: 选择需要被设置为风扇的实体;

速度方向: 遵循右手法则, 选择方向为大拇指方向, 物体运动方向为四指方向转动;



风扇有两种使用方式: 一种是自己独立的风扇不需要其他器件提供动力, 直接被控制器

代码控制。示例使用场景如下图所示:



对应的控制器代码如图所示:



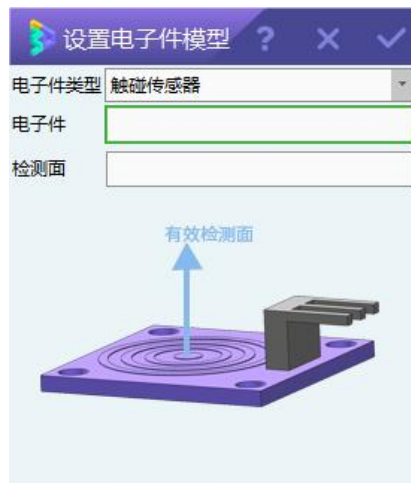
另外一种使用方式就是配合马达使用，见马达和双向马达。

8.2 触碰传感器

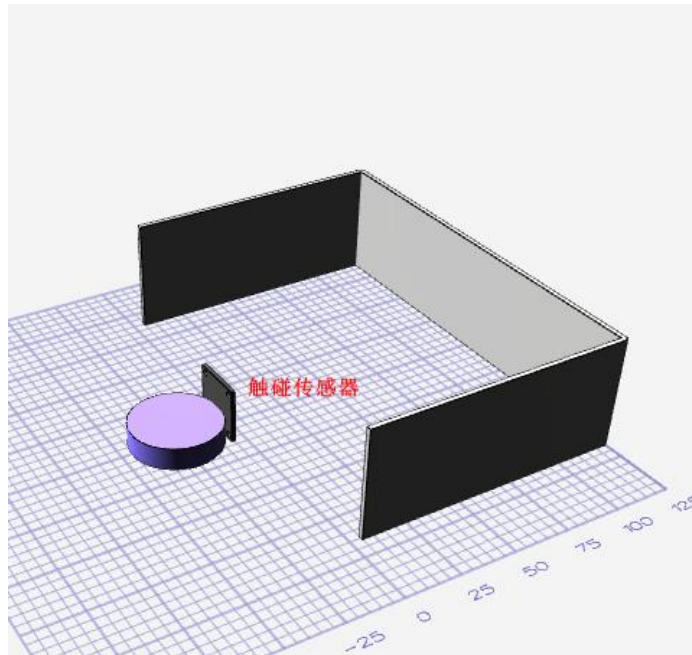
触碰传感器的作用是检测其他模型碰撞的情况，通过判断状态实现某些逻辑程序。

电子件：选择需要被设置为风扇的实体；

检测面：检测是否被触碰的面。



示例使用场景如下图所示：



程序设计：启用传感器，根据传感器是否与其他物体接触来决定物体运行状态，未接触则前行，已接触则后退并左转 10 度。对应的积木如下图所示：



8.3 开关

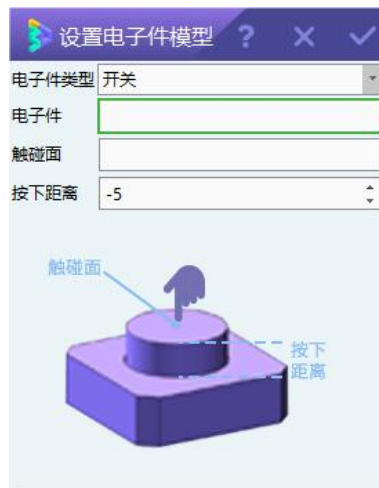
开关的作用是在编程时通过开关的状态判断事件，同时也可以通过鼠标点击或者编程

来改变开关的状态。给整个作品提供一个逻辑开关，整体运行逻辑根据开关的状态做出一些逻辑行为，例如开关开启则风扇转动，开关关闭则停止转动。

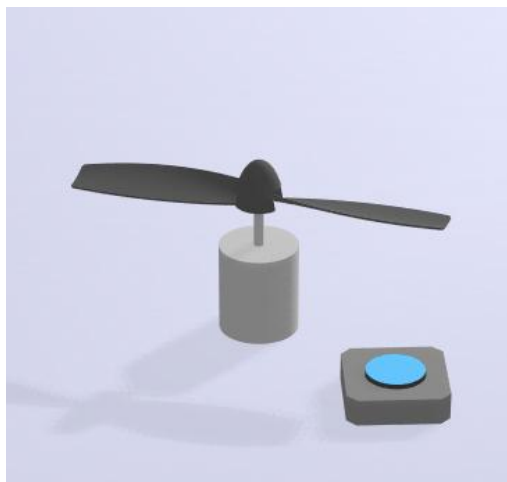
电子件：选择需要被设置为开关的实体；

触碰面：设置开关的触碰面

按下距离：设置触碰面被按下的距离



示例使用场景如下图所示：



程序设计：根据开关的闭合和断开状态来控制风扇的转动，运行仿真之后，鼠标点击开关可以改变开关的状态，从而控制风扇。对应的积木如下图所示：



8.4 LED 灯

可设置 LED 灯亮灯颜色与关闭颜色，可以通过编程来控制是否亮灯。

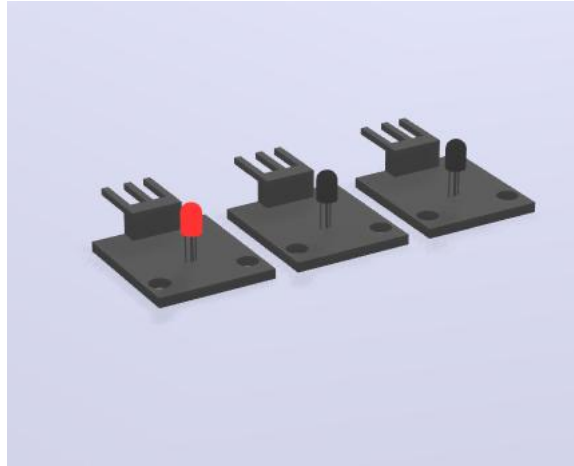
电子件：选择需要被设置为 LED 灯的实体；

打开颜色：亮灯颜色；

关闭颜色：关闭状态颜色；

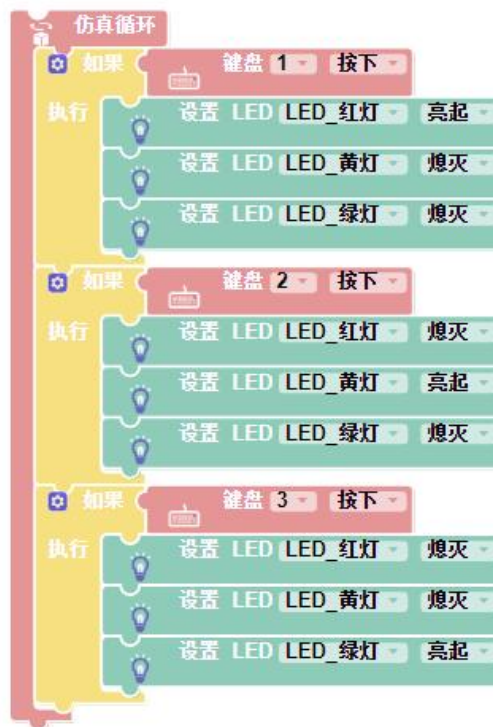


示例使用场景如下图所示：



程序设计：根据键盘上 1、2、3 按键来控制红、黄、绿灯的亮起和熄灭。对应的积木

如下图所示：



8.5 舵机

舵机的功能是提供角度输出，通过设置关节与编程，舵机能够将关联的物体转动指定角度。

电子件：选择需要被设置为舵机的实体；



示例使用场景如下图所示：



程序设计：舵机和实体通过合页关节连接起来，舵机为实体提供动力。根据键盘上 a 按键来控制舵机的转动角度每次转动 50 度。对应的积木如下图所示：



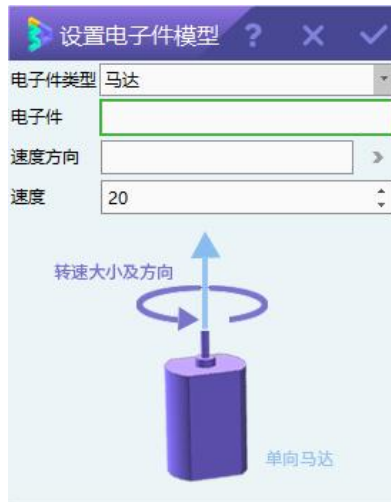
8.6 马达

这里的马达是单向马达, 马达的作用是给其他电子件提供动力的。通过设置关节与编程, 马达可以带动其他物体单向转动。

电子件：选择需要被设置为马达的实体；

速度方向：遵循右手法则，选择方向为大拇指方向，实际运动方向为四指指向方向旋转；

速度：单位为 rad/s。



示例使用场景如下图所示：



程序设计：马达和风扇通过合页关节连接起来，马达能够给风扇提供动力。根据键盘上 1、2、3 按键来控制马达的转速，通过 0 按键来控制马达停止转动。对应的积木如下图所示：



8.7 循迹传感器

循迹传感器可以自由选择左右两侧灰度检测点,来判断在指定方位检测到的物体是否为指定颜色。

电子件: 需要被设置的模型

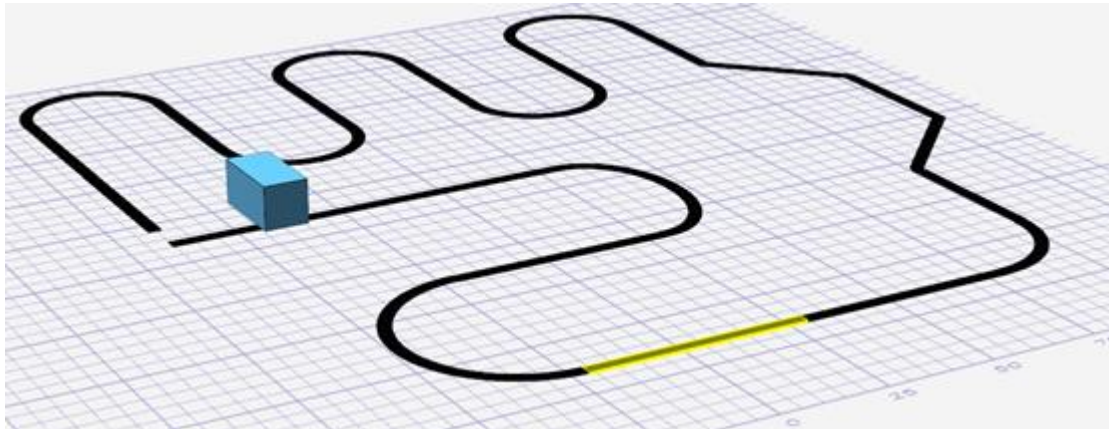
左检测点: 左边检测用射线的起始点位置(通常设置于模型表面)

右检测点: 右边检测用射线的起始点位置(通常设置于模型表面)

范围: 射线长度, 表示循迹传感器的检测范围



示例使用场景如下图所示：



程序设计：循迹传感器沿着轨迹上向前运行。根据循迹传感器检测到左边和右边的颜色，左边是黑色就左转 3 度，右边是黑色就右转 3 度，两边都是白色就直行。对应的积木如下图所示：



8.8 电子显示屏

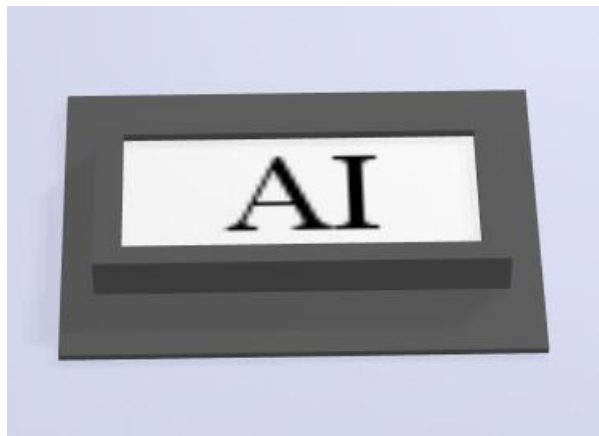
通过设置物体为电子显示屏，在仿真过程中可以实现一些文字的输出与显示。

电子件：选择需要被设置的实体；

检测面：用于作为显示屏的面。



示例使用场景如下图所示：



程序设计：启动电子显示屏，根据时间顺序显示“3D”、“One”、“AI”等内容。对应的积木如下图所示：



8.9 距离传感器

构建一条透明虚拟的射线, 通过射线计算与其他模型的碰撞, 探测前方物体并返回距离测量数值。

电子件: 需要被设置的模型

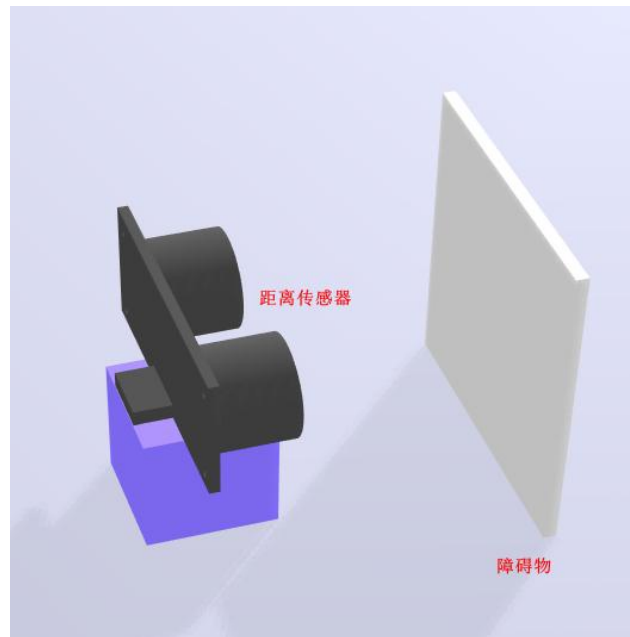
起始位置: 检测用射线的起始点位置(通常设置与模型表面)

方向: 检测用射线方向

范围: 射线长度, 表示距离传感器的检测范围(最大测量值)



示例使用场景如下图所示:



程序设计: 根据距离传感器检测到的与障碍物的距离来决定运行方式, 距离大于 50mm

就前行, 距离小于 50mm 就后退。对应的积木如下图所示:



8.10 力传感器

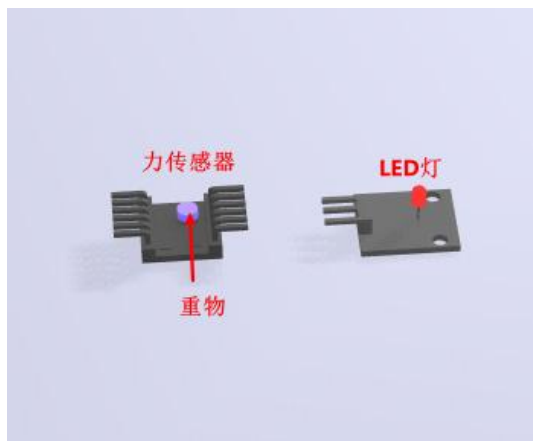
检测传感器模型指定方向与其他模型碰撞时, 所受力的大小。

电子件： 需要被设置的模型

检测面： 有效检测面



示例使用场景如下图所示：



程序设计：根据力传感器检测到的重物自由落下时产生力度来控制 LED 灯的亮起。对应的积木如下图所示：



8.11 光线传感器

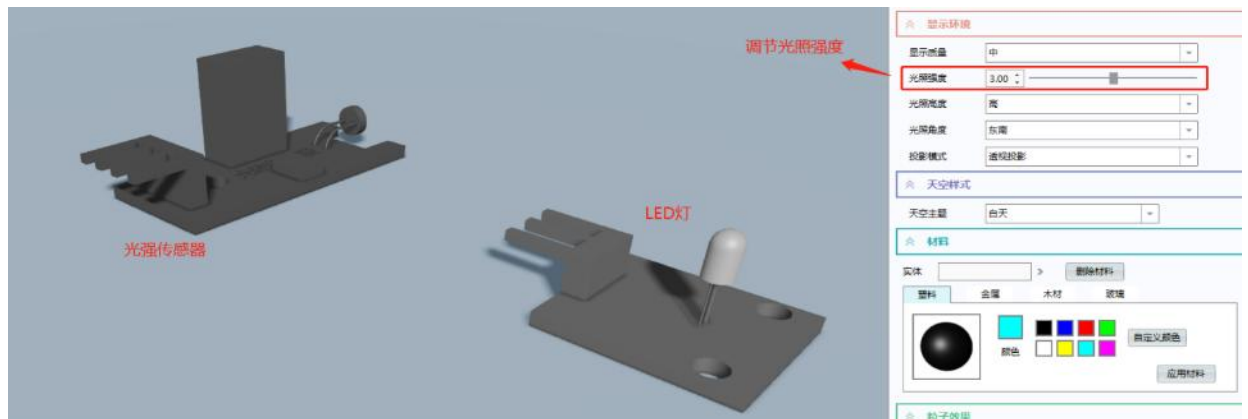
检测仿真环境光强度。仿真环境的光线强度可以通过右上角的按钮就行调节，光线传感器配合控制光线强度的调整的实现天亮灯灭，天黑灯亮的效果。

电子件：需要被设置的模型

检测面：有效检测面



示例使用场景如下图所示：



程序设计：根据光线传感器检测到的环境光线强度来控制 LED 灯的亮起和熄灭，光强小于等于 3 的时候灯亮，大于 3 时灯灭，仿真过程中可以通过调节环境的光线强度来模拟现实。对应的积木如下图所示：



8.12 颜色传感器

软件后台构建一条透明虚拟的射线，通过射线检测相交的模型碰撞点的颜色(RGB)。

电子件：需要被设置的模型

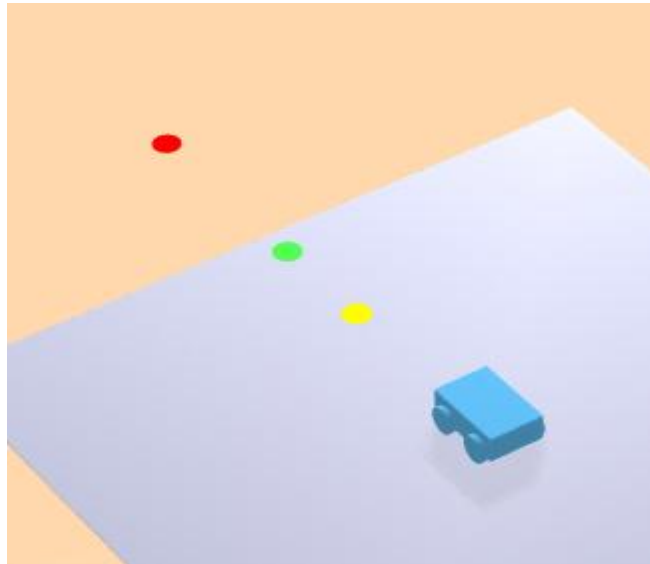
起始位置：检测用射线的起始点位置(通常设置与模型表面)

方向：检测用射线方向

范围：射线长度，表示距离传感器的检测范围(最大测量值)



示例使用场景如下图所示：



程序设计：根据颜色传感器检测到的实体颜色来控制小车的运行速度，检测结果为绿色则前行，为黄色则慢性，为红色则停。对应的积木如下图所示：



8.13 虚拟摄像头

提供一个类似摄像机功能，是一个虚拟的摄像头，设置后，在仿真时会将模型设置的视角展示在右下方窗口中，在仿真时可以获仿真环境中看到的图像。

电子件： 需要被设置的模型

起始位置： 虚拟摄像头视角中心

方向： 虚拟摄像机视角方向



示例使用场景如下图所示:



程序设计: 利用虚拟摄像头进行图像识别, 根据识别对象控制玩具球运动, 检测结果

为狗则玩具球开始前行。对应的积木如下图所示:



8.14 转盘

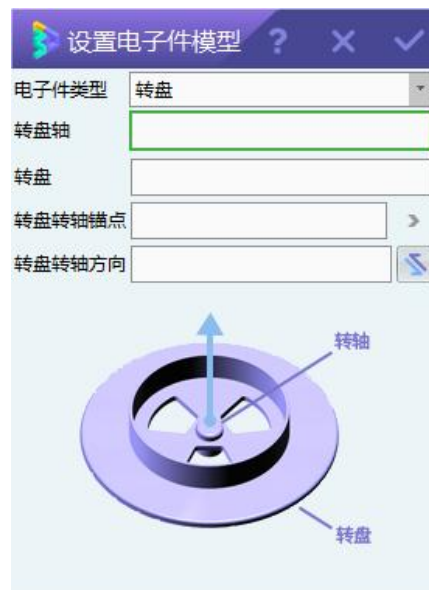
设置转盘的关节，让转盘成为电子件可以通过程序控制。

转盘轴： 转盘中间轴

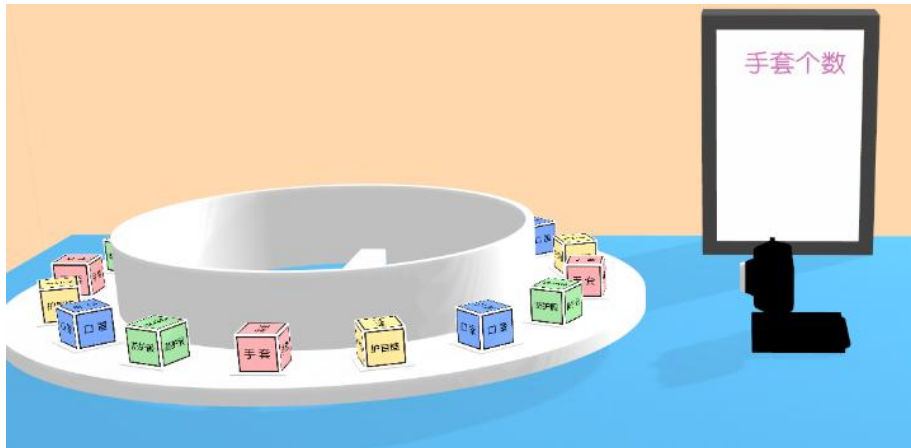
转盘： 转盘体

转盘转轴锚点： 转盘围绕着转的轴的中心点

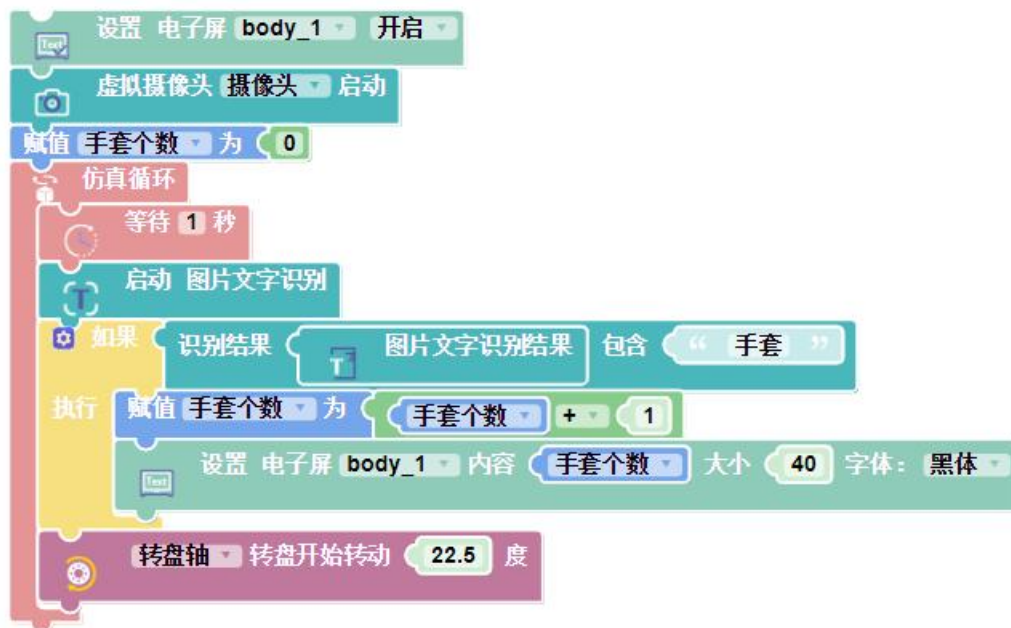
转盘转轴方向： 转盘轴延伸的方向



示例使用场景如下图所示:



程序设计: 利用虚拟摄像头进行图像识别, 如果识别到的结果为手套, 电子显示屏手套个数加一。

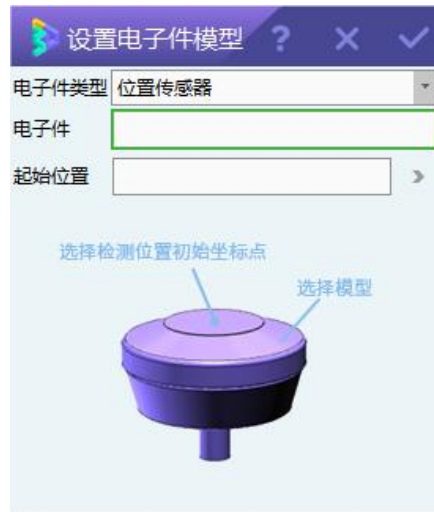


8.15 位置传感器

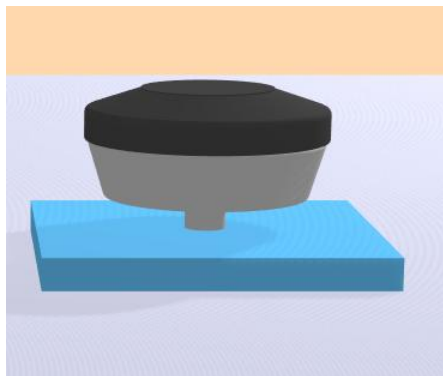
检测传感器模型的实时位置坐标, 编程中可以获取某一模型的在仿真过程中的实时坐标, 基于实时坐标做出特定的响应。

电子件： 需要被设置的模型。

起始位置： 传感器位置检测中心



示例使用场景如下图所示：



程序设计：根据位置传感器检测到的位置坐标控制实体的运动与停止，检测位置坐标 x

大于 50 则停止前行，否则前行。对应的积木如下图所示：



8.16 灰度传感器

灰度传感器可以自由选择左右两侧灰度检测点,来判断在指定方位检测到的物体是否为指定颜色。

电子件: 需要被设置的模型

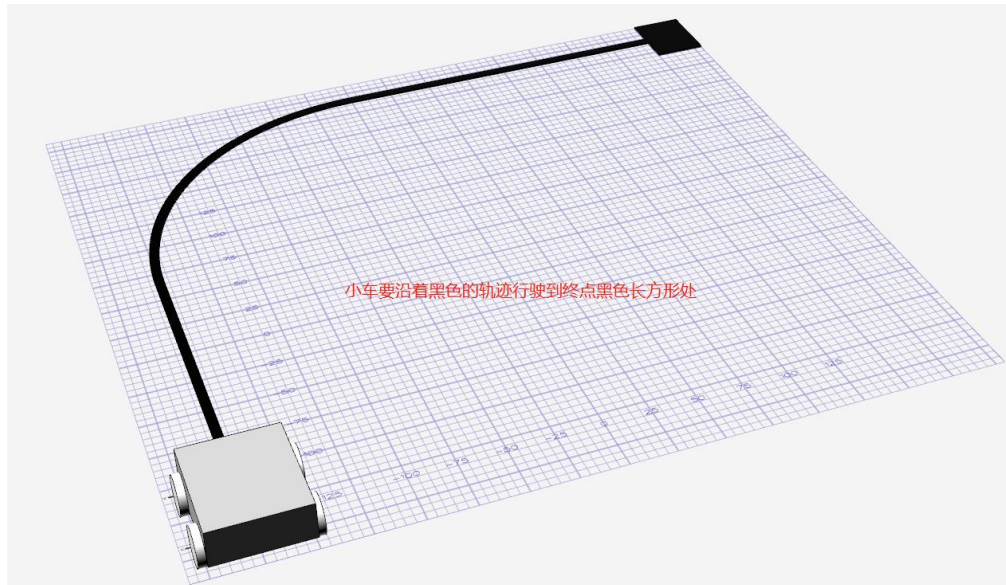
左检测点: 左边检测用射线的起始点位置(通常设置于模型表面)

右检测点: 右边检测用射线的起始点位置(通常设置于模型表面)

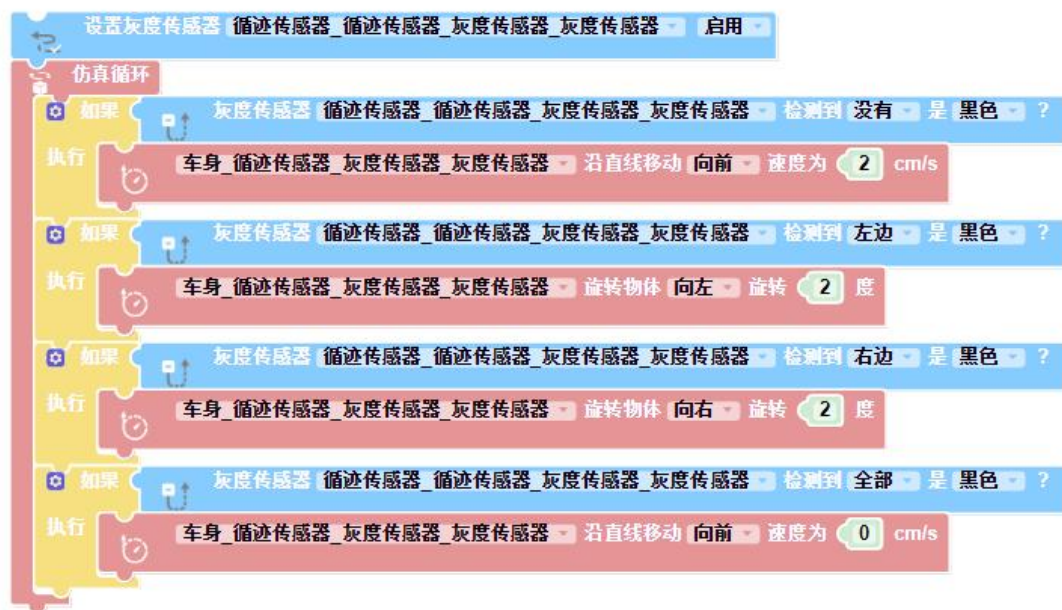
范围: 射线长度, 表示灰度传感器的检测范围



示例使用场景如下图所示:



程序设计：灰度传感器沿着轨迹上向前运行。根据灰度传感器检测到左边和右边的颜色，左边是黑色就左转 2 度，右边是黑色就右转 2 度，没有检测到黑色就直行，全部是黑色就停止前进。对应的积木如下图所示：



8.17 机械臂

设置机械臂关节，使机械臂成为电子件，能够通过程序控制。

机械臂：机械臂的竖直轴

z 轴臂：机械臂再 z 轴方向移动的臂

z 轴方向：z 轴臂移动的轴方向

大臂：二阶机械臂中带着小臂移动的臂

大臂锚点：大臂关节设置的点，大臂围绕 z 轴臂旋转的中心点

大臂轴方向：大臂围绕 z 轴旋转的轴心轴方向

小臂：二阶机械臂中带着夹爪的臂

小臂锚点：小臂关节设置的点，小臂围绕大臂旋转的中心点

小臂轴方向：小臂围绕大臂旋转的轴心轴方向

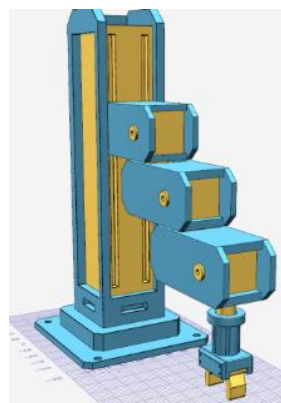
z 轴位置偏差：z 轴臂的初始位置偏差

大臂角度偏差：大臂的初始角度偏差

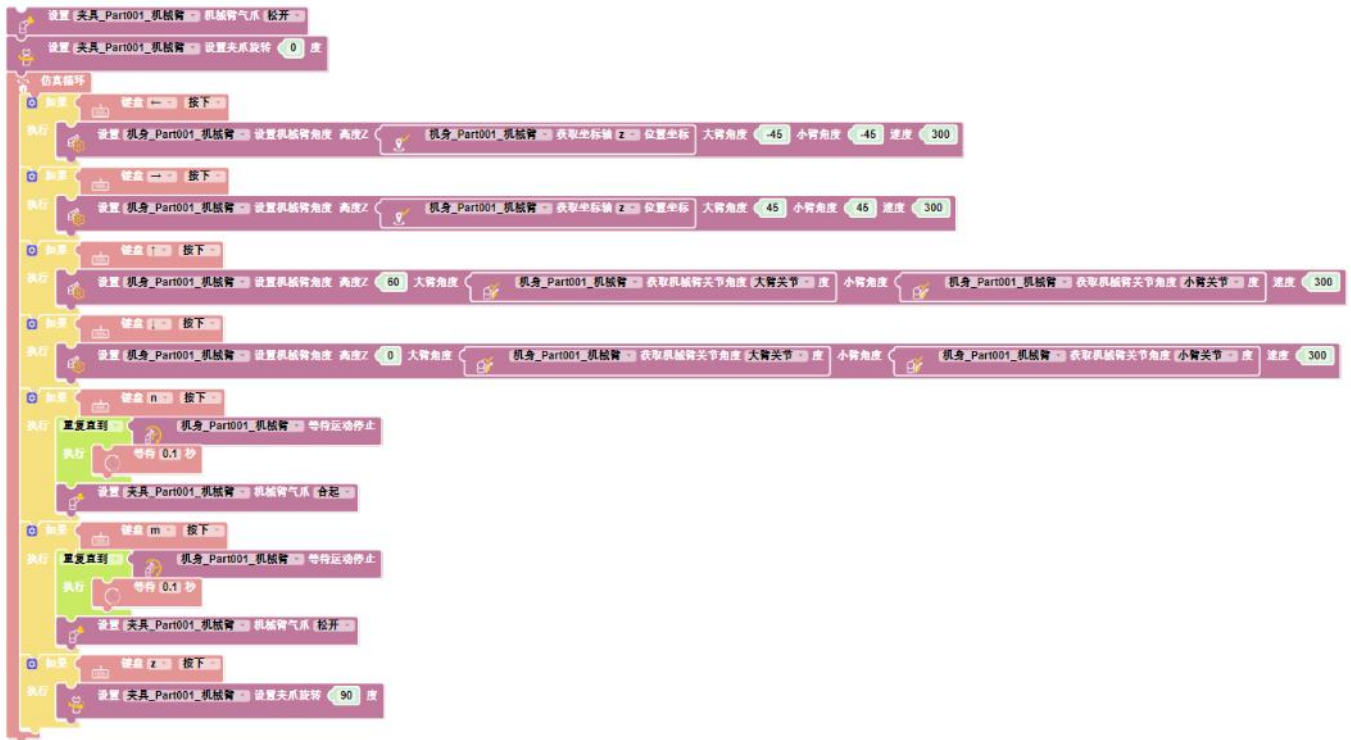
小臂角度偏差：小臂的初始角度偏差



示例使用场景如下图所示：



程序设计：先初始化夹爪、夹具电机还有机械臂，当←按下后，机械臂移动到大臂-45°、小臂-45°的位置，当→按下后，机械臂移动到大臂 45°、小臂 45°的位置，当↑按下后，机械臂向上移动，当↓按下后，机械臂向下移动。当按下 n 时，机械臂的夹爪合起；当按下吗、时，机械臂的夹爪松开，当按下 z 时，机械臂的夹爪旋转 90°。



8.18 插销夹爪

设置夹爪的关节点，使夹爪编程电子件，能够通过程序控制。

夹爪：夹爪的基体

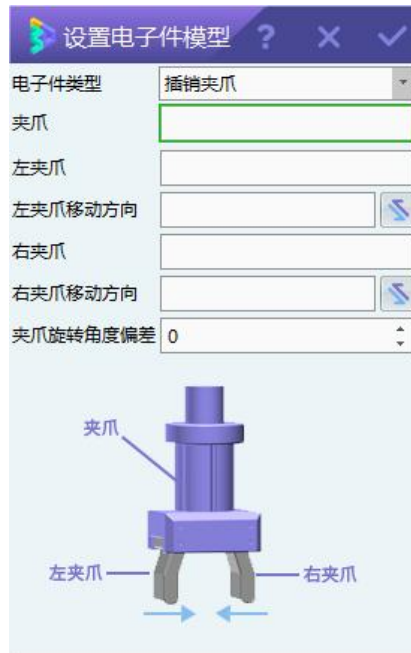
左夹爪：夹爪左边的夹子

左夹爪移动方向：爪子通过程序控制后移动的方向

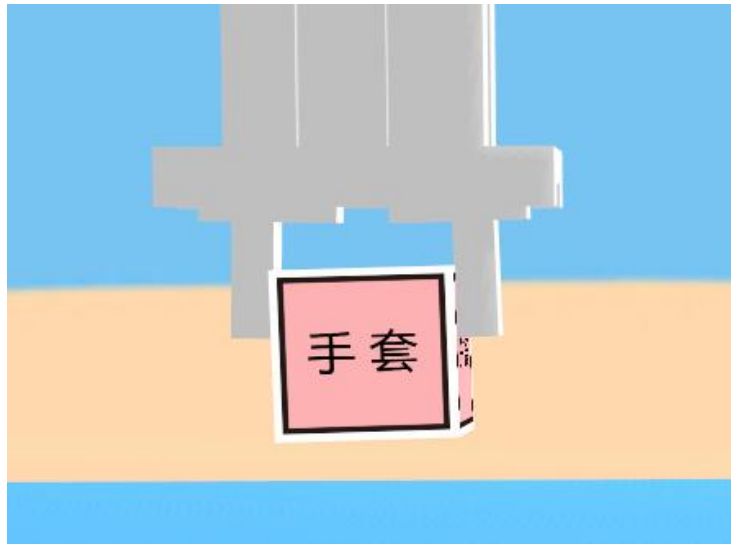
右夹爪：夹爪右边的夹子

右夹爪移动方向： 爪子通过程序控制后移动的方向

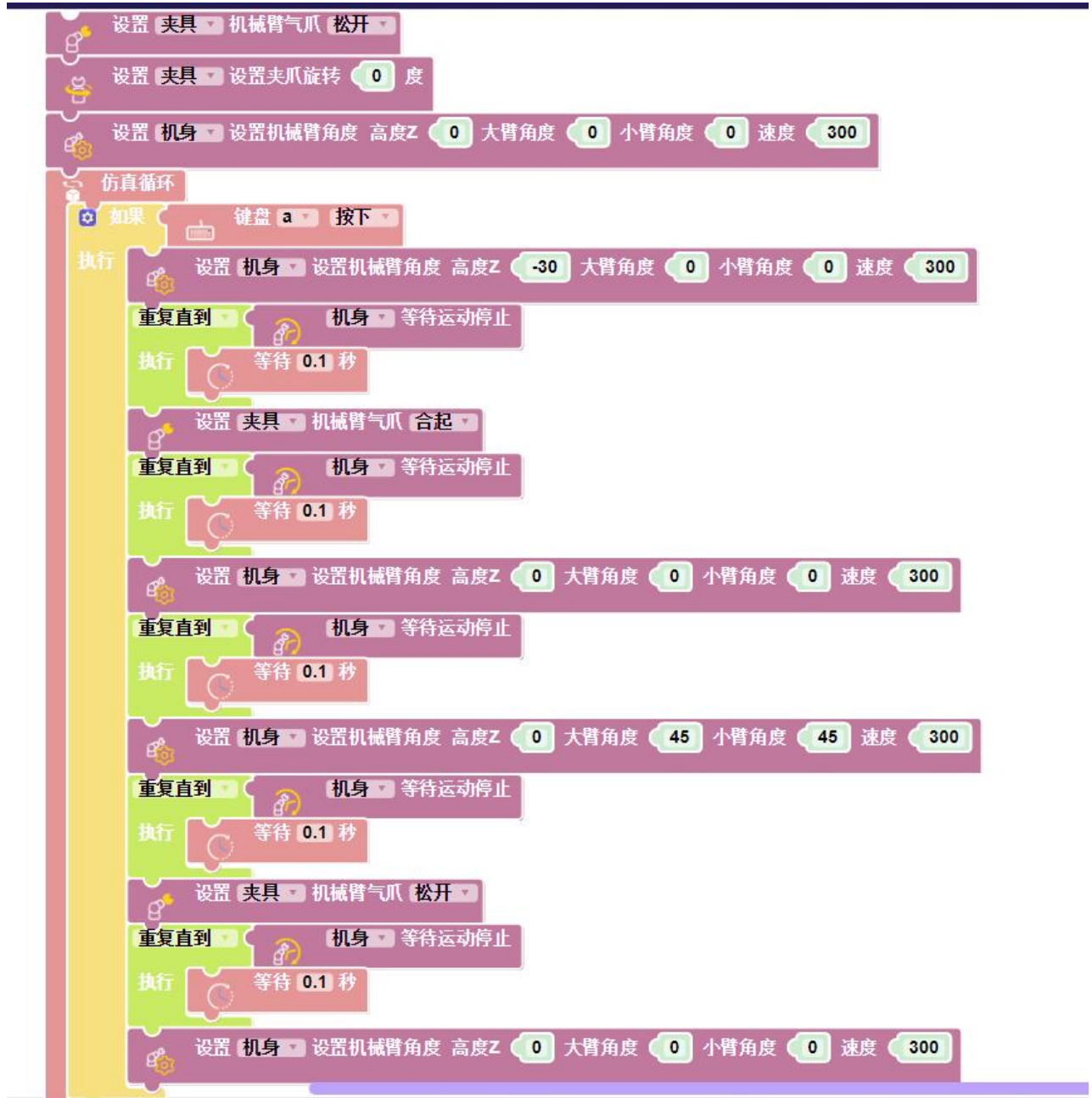
夹爪旋转角度偏差： 爪子的初始角度偏差



示例使用场景如下图所示：



程序设计：先初始化夹爪、夹具电机还有机械臂，当 a 按下后，机械臂向下移动，插销夹爪合起，再升起机械臂抓起物块，接着机械臂移动到指定位置，等待机械臂移动到位置



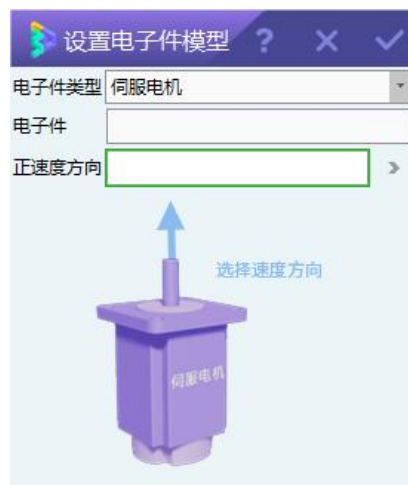
后插销夹爪松开，放下物块，机械臂回到原位。

8.19 伺服电机

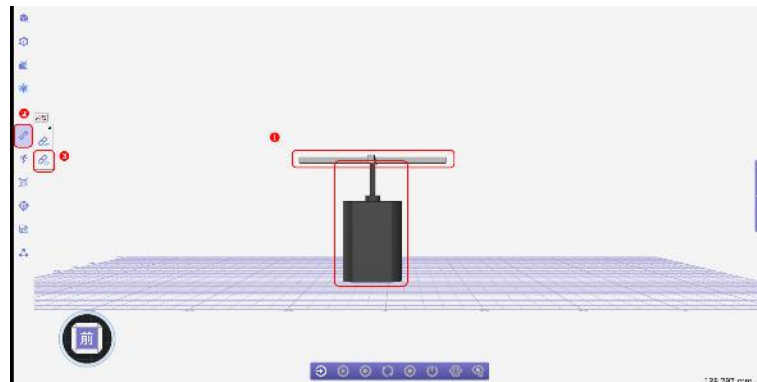
靠伺服控制电路（伺服控制指对物体运动的位置、速度及加速度等变化量的有效控制）以及传感器来闭环控制电机的转速和转动位置。

电子件：需要被设置的模型。

正速度方向：选择速度方向。

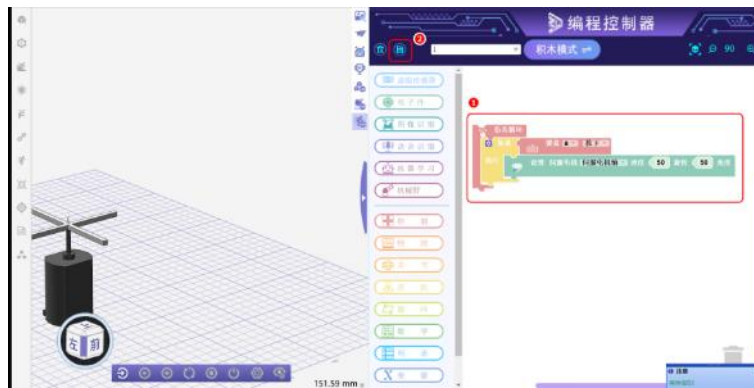


示例使用场景如下图所示：



程序设计：首先创建两个实体，一个为轴，一个为转盘。需要设置转盘围绕轴的轴心旋转，所以接着点击关节的关节设置，打开关节面板。选择关节类型为合页，并选中两个实体，设置锚点以及轴心的位置点击确定。打开电子件设置面板，选择电子件类型为伺服电机。选择作为伺服电机的电子件，设置速度方向。

打开编程设置控制器界面，设计如图程序通过键盘控制伺服电机。



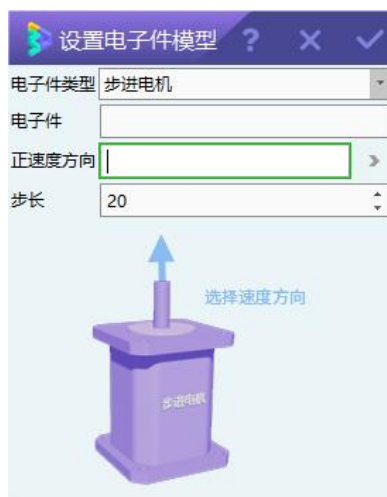
8.20 步进电机

是将电脉冲信号转变为角位移或线位移的开环控制元件。简单的说是靠电脉冲信号来控制角度与转动圈数。步长定义为每次脉冲转动的角度，转动角度也可实现 360° 旋转。

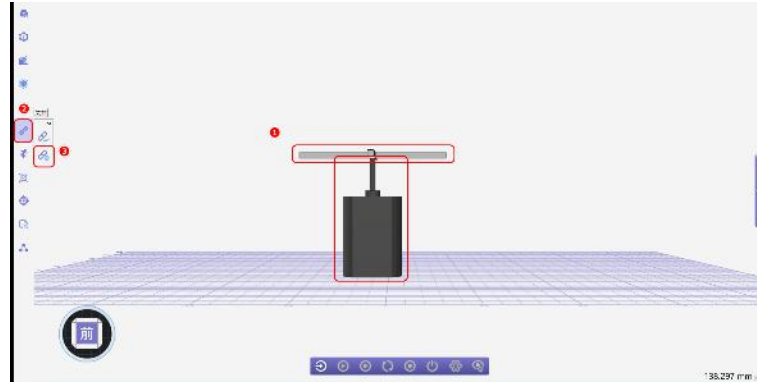
电子件： 需要被设置的模型。

正速度方向： 选择速度方向。

步长： 每次脉冲转动的角度。

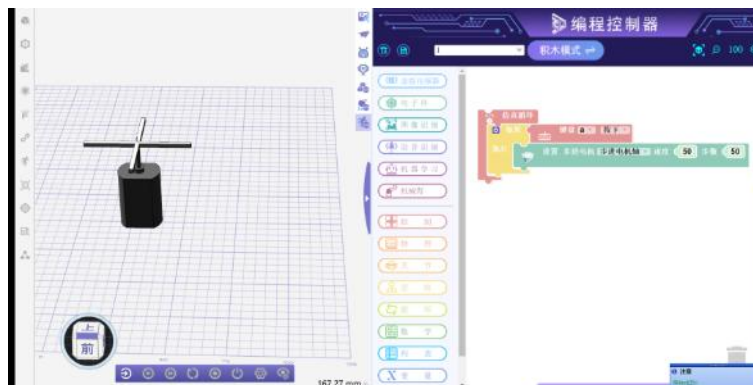


示例使用场景如下图所示：



程序设计：首先创建两个实体，一个为轴，一个为转盘。我们需要设置转盘围绕轴的轴心旋转，所以接着点击关节的关节设置，打开关节面板。选择关节类型为合页，并选中两个实体，设置锚点以及轴心的位置点击确定。打开电子件设置面板，选择电子件类型为步进电机。选择作为步进电机的电子件，设置正速度方向。

打开编程设置控制器界面，通过如图程序键盘去控制步进电机。



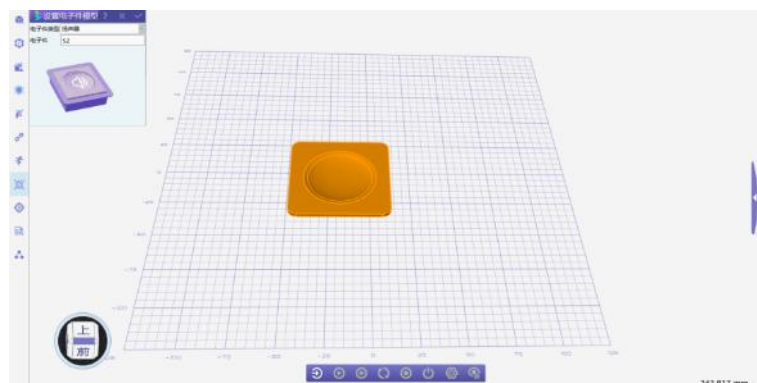


8.21 扬声器

可播放或暂停导入的音频文件。

电子件： 需要被设置的模型。

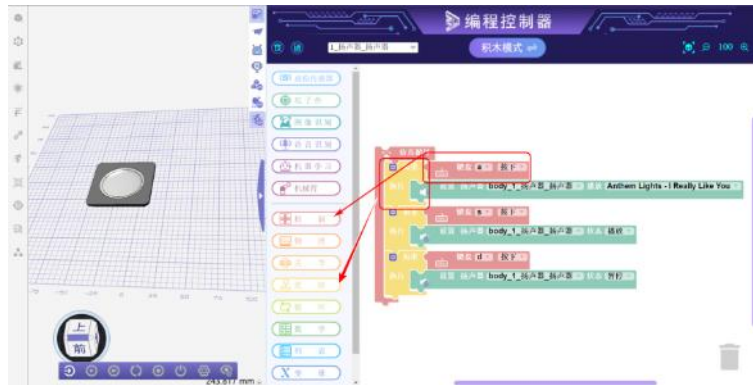
示例使用场景如下图所示：



程序设计：

从逻辑分类和控制分类中选择需要的积木块，按照下图的拼接顺序将积木块拼接，可实现以下功能：① 按下 a 键，将扬声器的音频切换为之前导入的音频，如果导入多个音频，可进行选择，并自动开始播放音频。② 按下 s 键，扬声器开始播放音频。③ 按下 d 键，扬

声器暂停播放音频。



8.22 陀螺仪传感器

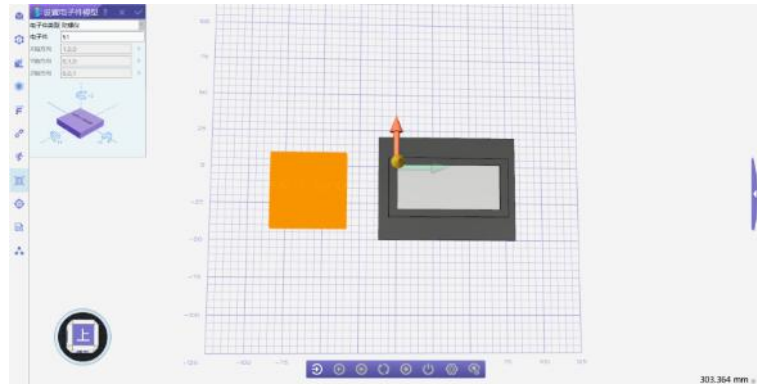
可获取绕 x、y、z 三个方向转动的角速度以及物体在 x、y、z 三个方向上的偏移角度。。

电子件： 需要被设置的模型。

软件设置参数： 无，陀螺仪传感器的检测方向默认与环境一致，且不可修改。x 轴 (1, 0, 0) , y (0, 1, 0) , z (0, 0, 1)



示例使用场景如下图所示：



程序设计:

设置陀螺仪, 打开电子件设置面板, 选择电子件类型为陀螺仪。选择作为陀螺仪的电子件。打开电子件设置面板, 选择电子件类型为电子显示屏。选择作为电子显示屏的电子件, 选择显示面。设计下图所示的代码, 实现的功能: ①点击键盘 a, 物体向左旋转 1 度。②点击键盘 s, 物体获得一个线速度, 大小为 10cm/s。③点击键盘 z, 电子显示屏显示物体在 y 轴方向的速度。④点击键盘 x, 电子显示屏显示物体绕 z 轴方向旋转的角度。电子显示屏显示角度数值为度数转换成弧度的数值, 范围为-3.14~3.14 之间。

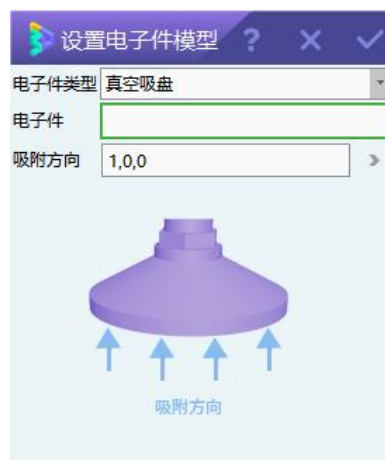


8.23 吸盘

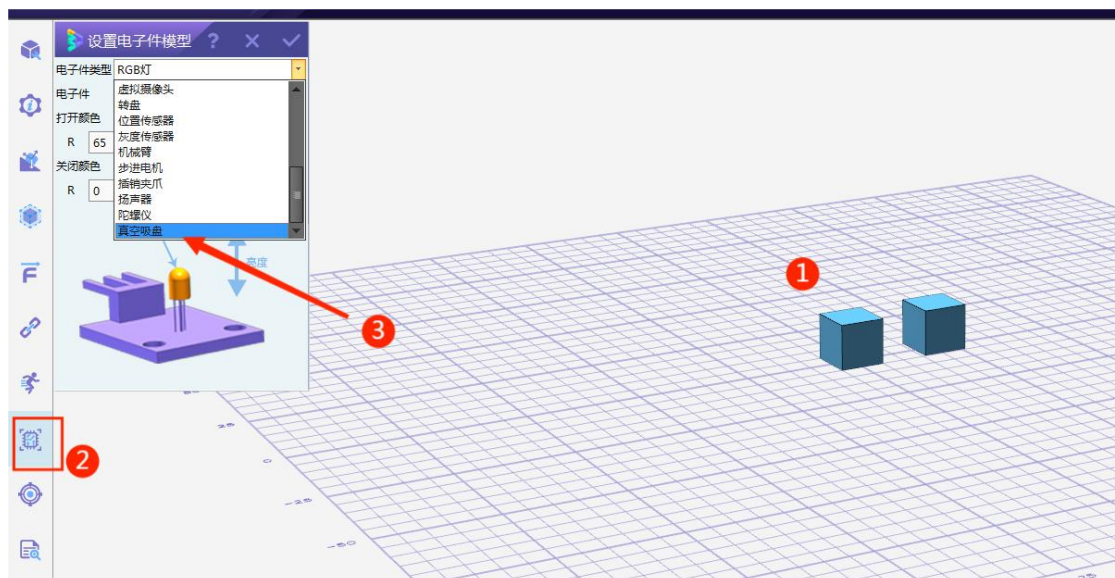
新增一个电子件在仿真中实现成组与解组的操作。吸盘打开后，当吸盘与物体相碰，物体与吸盘吸附固定。当吸盘关闭后，物体与吸盘取消固定。

电子件： 需要被设置的模型。

吸附方向： 吸附物体的方向。

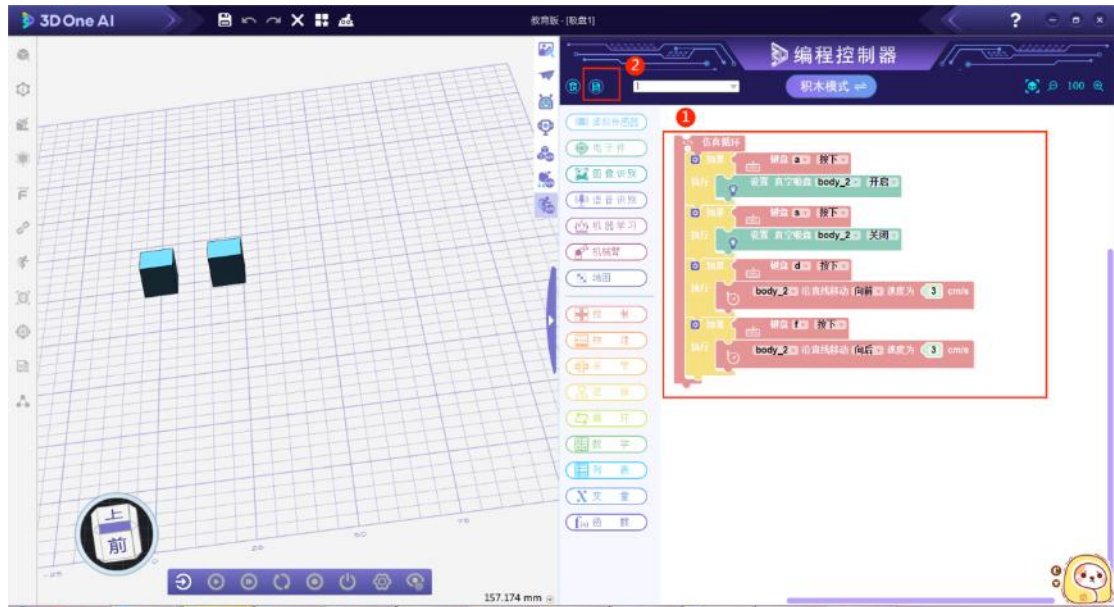


示例使用场景如下图所示：



程序设计：首先创建实体，点击设置电子件模型，选择电子件类型为真空吸盘。选择作为伺服电机的电子件（下图高亮显示），设置吸附方向，打开编程设置控制器界面，进入仿

真环境，点击启动仿真后，按键盘 a、s、d、f 查看吸盘吸附断开情况。



9. 设置第一人称视角

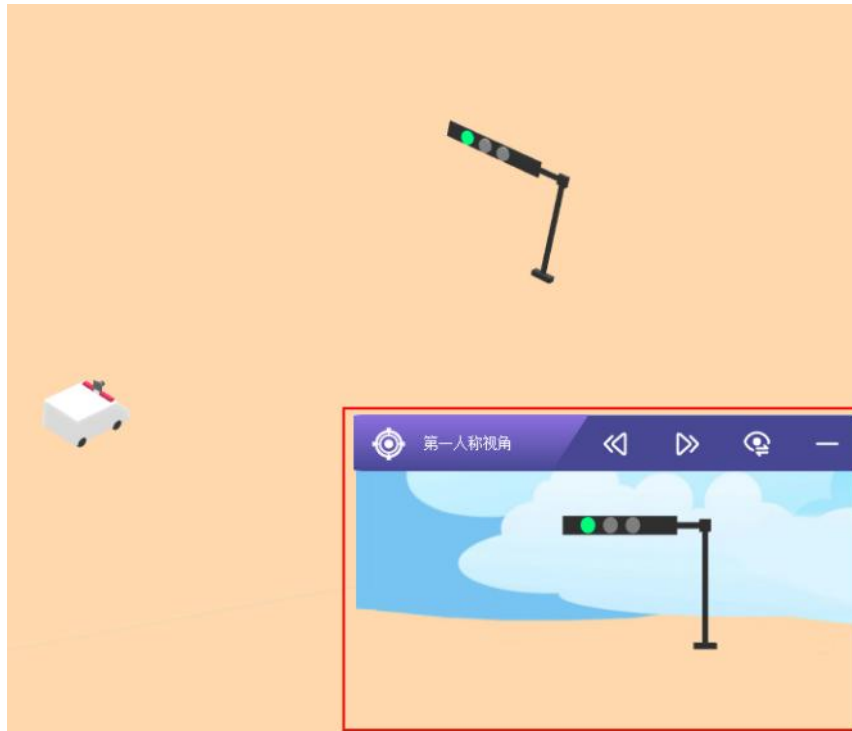
设置实体进入仿真后默认的第一人称视角, 此视角下看到的环境将显示在仿真环境中右下角的视角窗口中。

实体: 要被设置第一视角的物体

点: 第一视角的起始位置

方向: 第一视角的视线方向

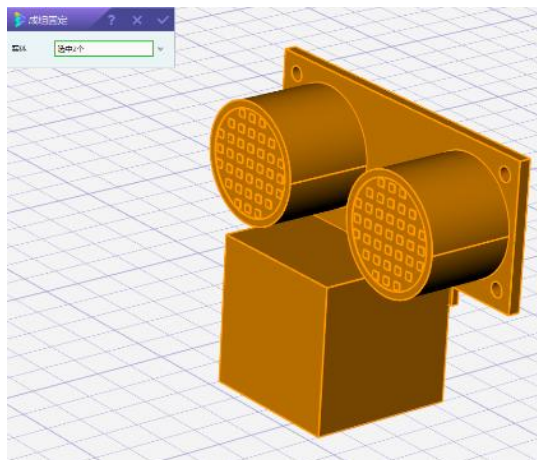


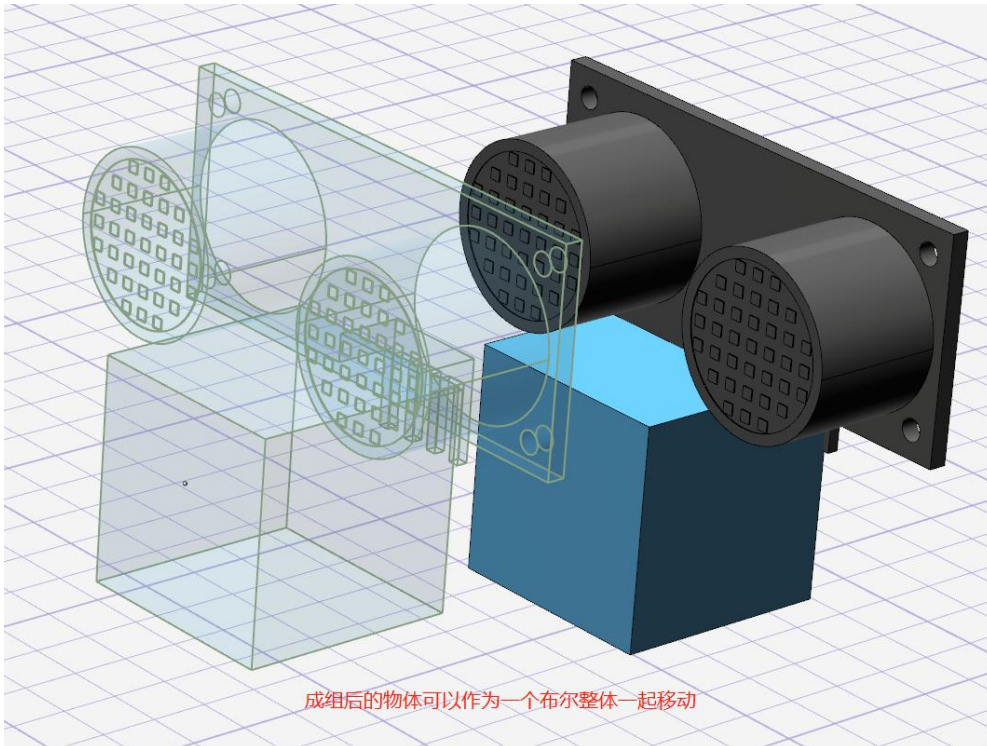


10. 组 (3D One AI V2.0 版本新增功能)

点击左侧工具栏中的最后一个选项，组命令 ，可看到三个子选项，分别为成组固定、炸开组、炸开所有组。

点击成组固定，可选中多个实体成组，组合实体相当于一个布尔加运算的整体，不会发生任何组合内模型挣脱的情况，示例图如下：



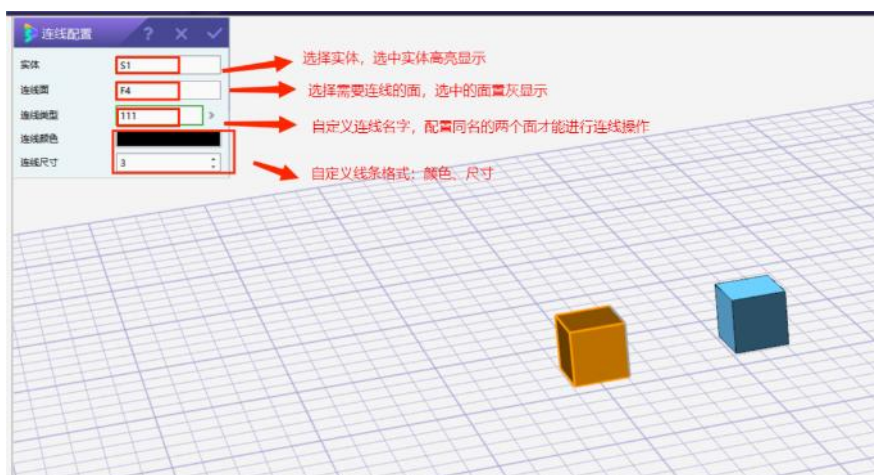


11. 连线

点击左侧  连线按钮，可以将已配置过连线的两个面在建模环境中相连。

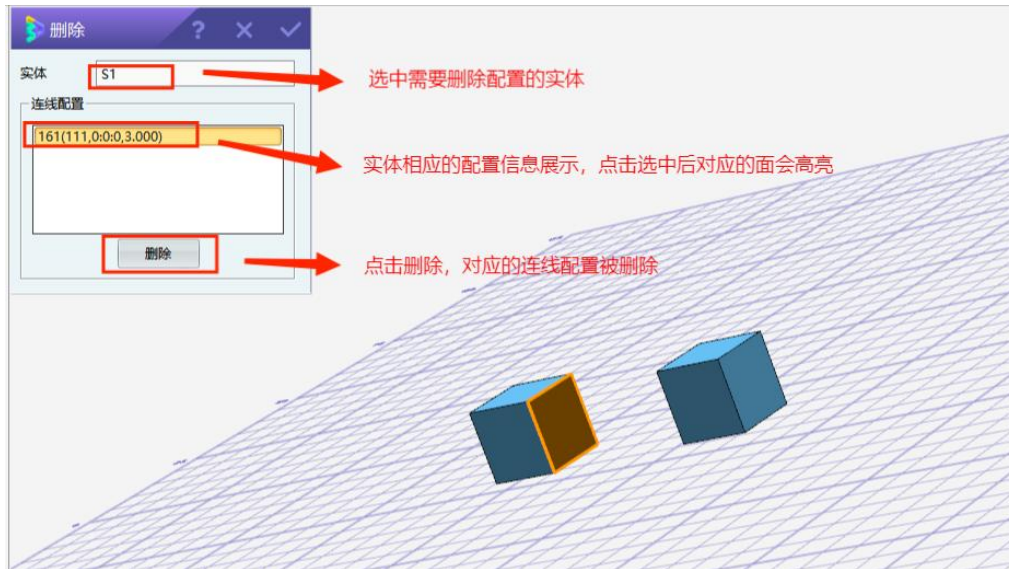
连线配置：

shift+alt+r 打开设置连线配置命令，选择对应连线的面，连线匹配的名称，连线的颜色，连线的尺寸。

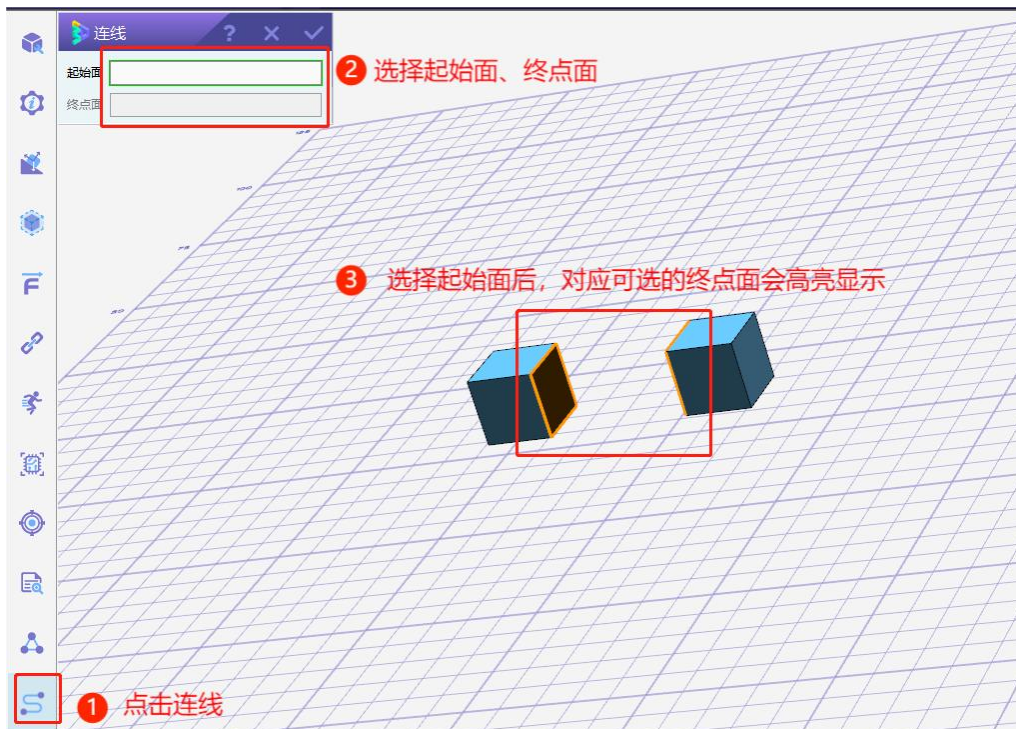


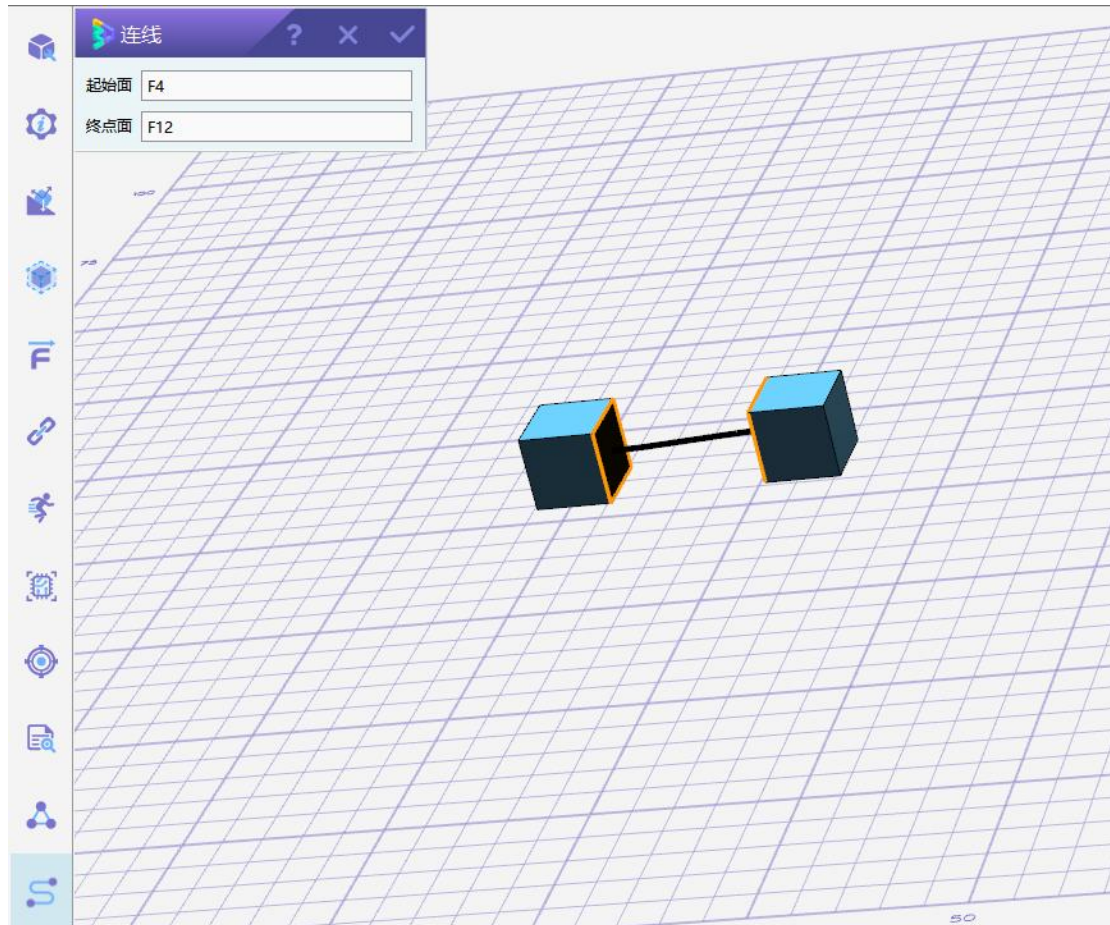
连线配置编辑:

shift+alt+t 打开编辑连线配置的命令, 可以查看已经设置的配置项 (点击后会高亮显示) 以及删除已经设置的配置。



连线:



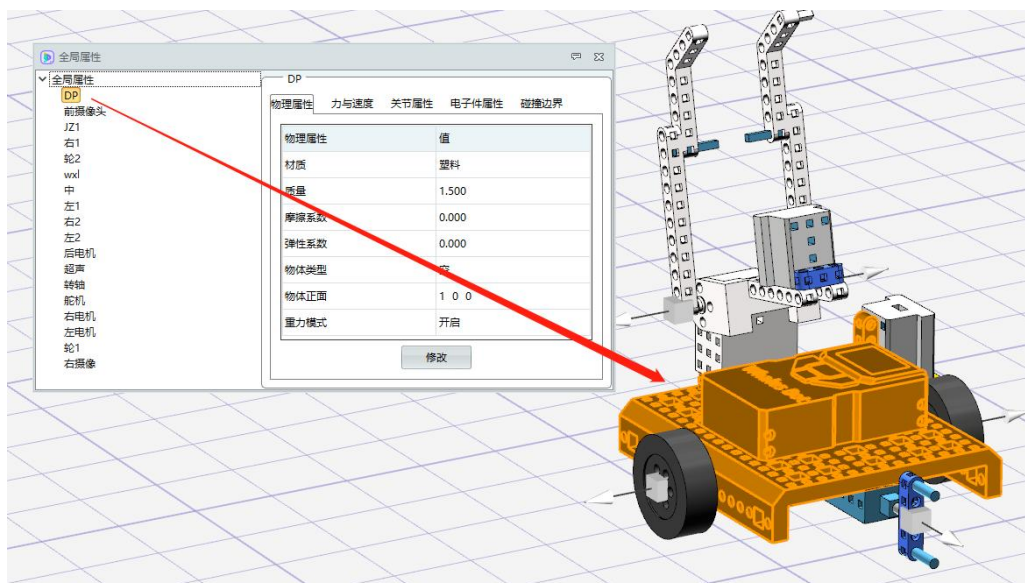


(建模环境下显示连线)

12. 全局属性 (3D One AI V2.0 版本新增功能)

点击左侧  全局属性按钮，可以查看并修改每个物体的物理属性、力与速度、关节属性、电子件属性，并查看碰撞边界。

同时，选中全局属性下的物体名称，还可以高亮所选中的物体。示例图如下：



13.匹配模型

击左侧  匹配模型，视图区高亮所有配置过的面，可以增加模型搭建的易用性。

使用方式可以分成两类：

一是自行配置模型，然后匹配模型。

二是使用官方模型，然后匹配模型。

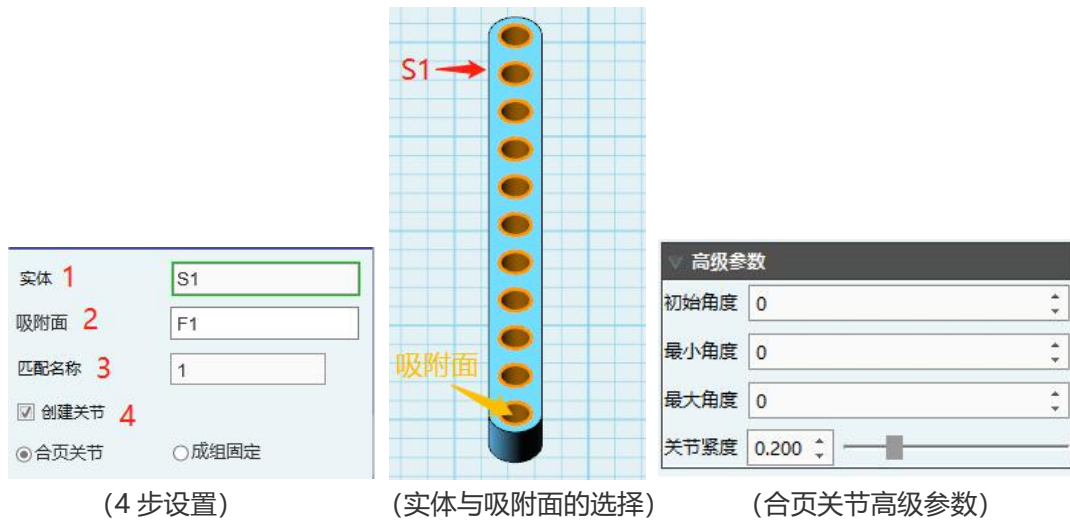
13.1 自行配置模型，然后匹配模型：

1) 调起命令

通过键盘组合键 shift+alt+y，调起搭建配置命令。

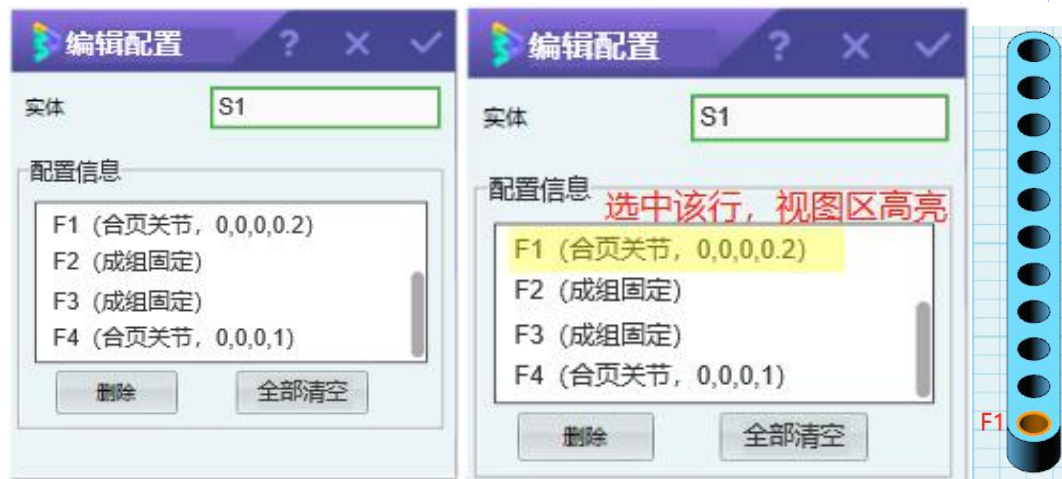
2) 配置模型

用户需要选择是否创建关节，如果创建，关节可以选择为合页或成组。当选择合页关节时，展示合页关节的设置高级参数；若选择成组固定，则不会出现高级参数。



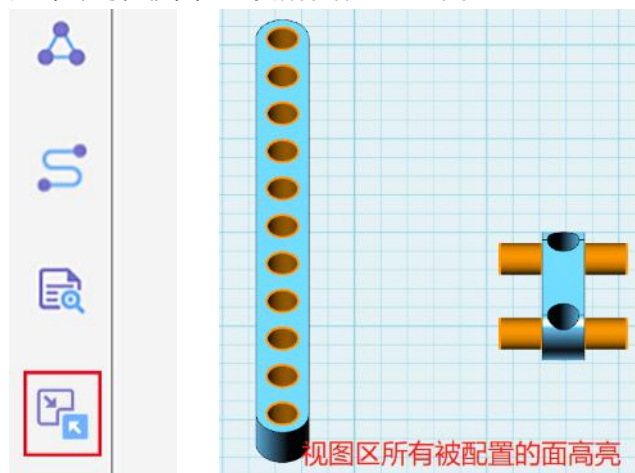
3) 编辑配置

通过键盘组合键 shift+alt+u, 调起编辑配置命令, 选择需要编辑的实体后, 对话框中显示该实体的搭建配置信息。单击选中某行, 该吸附面会高亮显示, 且可以对该行进行删除。

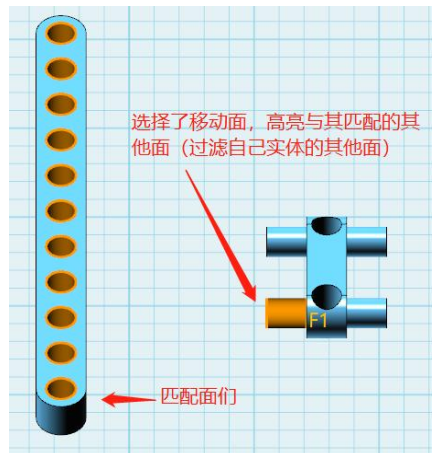


4) 匹配模型

左侧栏调起匹配模型命令时, 视图区高亮所有配置过的面。



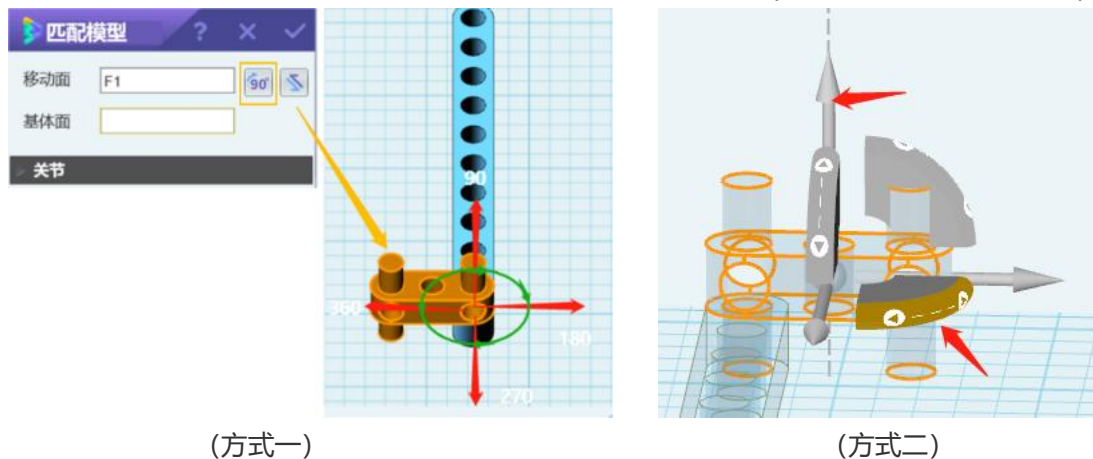
点击一个匹配过的面，此时根据所选的面的“匹配名称”查找同一匹配名称的面，高亮同一匹配名称的面。



此时可以通过两种方式调整模型的姿态与位置：

一是通过移动面旁边的旋转 90 度按钮 ，每次只能以 90 度作为旋转最小单位（顺时针）对单个法向旋转方向进行旋转。

二是通过界面上的可视化箭头进行吸附面法向的旋转与移动（其他 4 个方向不允许操作）。



(方式一)

(方式二)

13.2 使用官方模型，然后匹配模型：

1) 插入官方模型

插入模型时，带有搭建配置信息的模型自动调起匹配模型命令：



(两种调用命令方式)

2) 匹配模型：同 13.1 (4)

四、库资源与编程

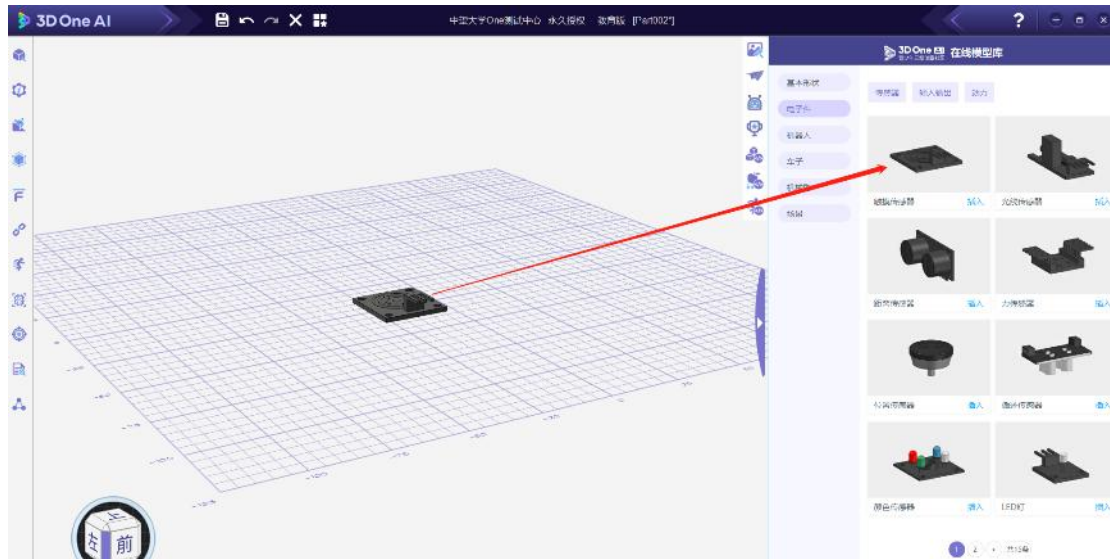
1. 模型库

模型库分为三类模型，一类是参数化模型（基本形状），一类是已配置信息模型（机器人、车、场景），一类是电子件模型。参数化模型支持重复修改模型参数，已配置信息模型已经配置好模型的物理信息、关节信息、受力运动，用户可以直接插入进入仿真查看效果，电子件模型为【编程控制器】提供虚拟电子件的编程对象，当插入电子件模型后，编程区会对应出现可编程的电子件模块。具体分类与数量如下表所示。（目前模型库的模型暂时不全，后续会继续更新）

本地模型	参数化模型	基本形状（13）
	已配置信息模型	机器人（0）
		车（8）
		场景（0）
电子件	传感器（7）	
	输入输出（4）	
	动力（4）	

● 电子件

插入电子件后，打开【编程控制器】页面，可以看到积木块有了对应的硬件编程对象：



2. 样例库

样例库现在包含本地案例以及社区在线案例，点击插入即可将模型导入。本地案例现在的数量分布如下：

名称	数量/个
基本形状	17

文字符号	33
机械建筑	3
文具	3
数字	6



3. 案例库

案例库资源包含传感器、电子件、图像识别、语音识别、机器学习几个模块案例的 z1AI 文件。点击【插入】即可插入整个案例。

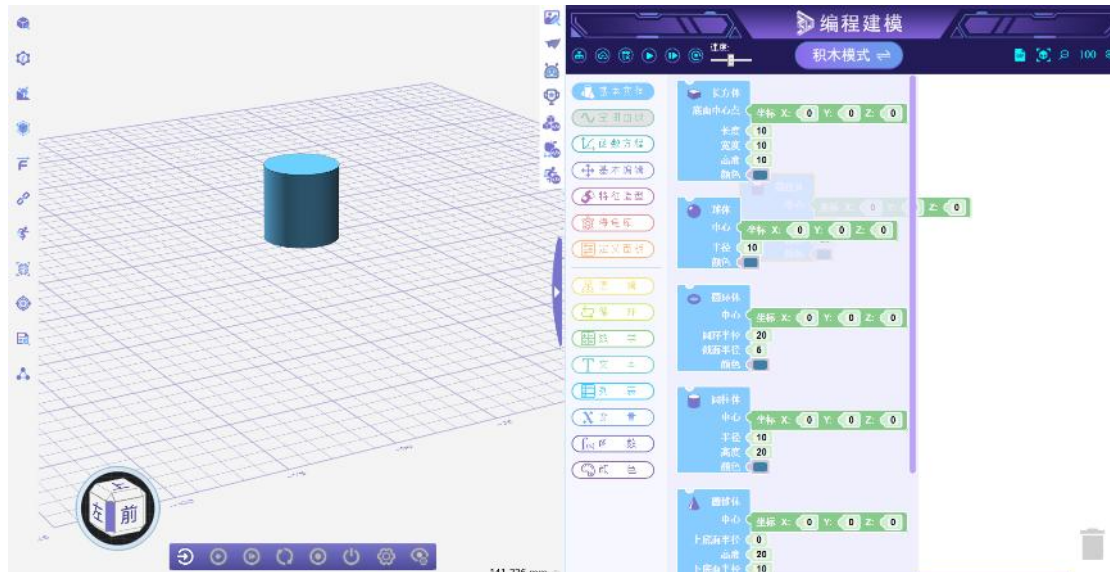


五、编程与接口说明

1. 编程建模

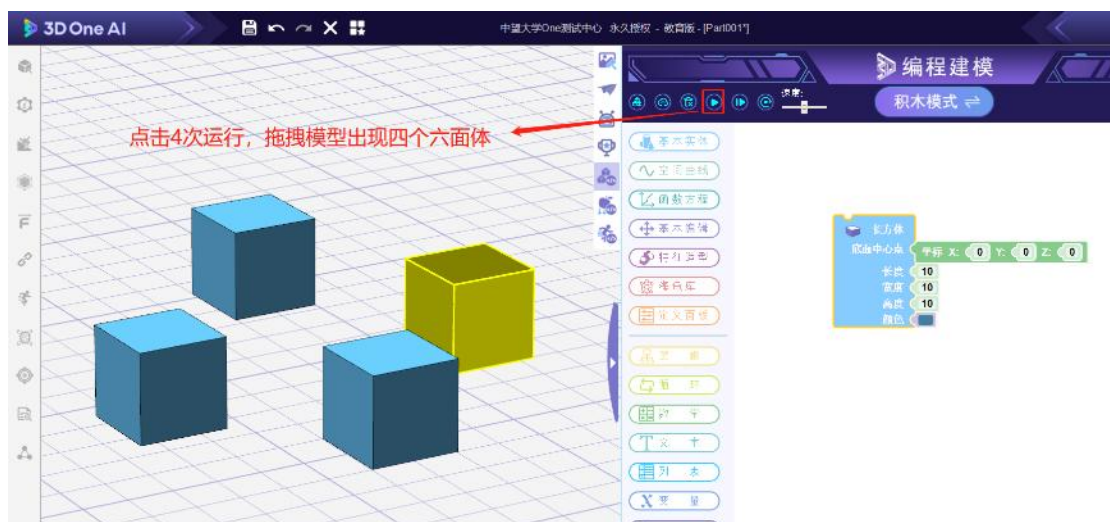
3D One AI V2.0 编程建模移植了 3D One V2.6 版本的编程建模模块, 每个命令的使用方式

与 Python 接口可以在 Python 模式下点击【语法库】查看:



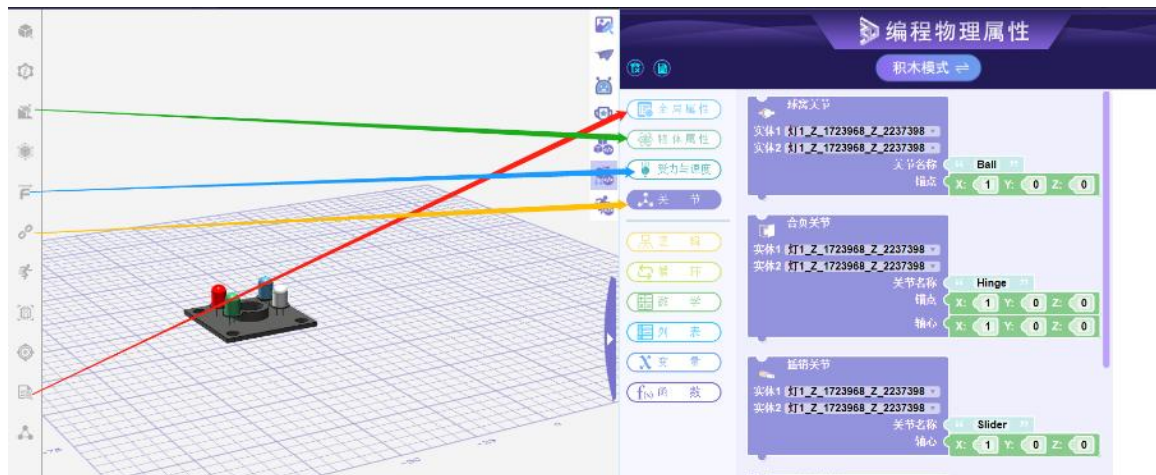
与 3D One 编程建模有所不同的地方是，模型代码生成控制方式发生了变化。3D One 编程建模采用的是一个积木代码对应视图区一个实体，每点击一次生成，就会清空左侧工具栏的建模操作（比如移动）。

3D One AI 编程建模生成模式是每点击一次生成，都会新增代码对应的模型，且不清空左侧工具栏的功能操作。例如六面体积木块点击 4 次生成，视图区会生成 4 个重叠的六面体，移动之后再次生成，不会清除移动操作。



2. 编程物理属性

编程物理属性主要有四大积木类，分别是【全局属性】、【物体属性】、【受力与速度】、【关节】，分别对应左侧工具栏的 4 个工具，右侧编程区可覆盖左侧的基本信息设置，用户可自由选择不同难易程度的设置形式进行基本信息的设置。



3. 编程设置控制器总体设置

(1) 积木类管理

编程控制器主要有十六个积木类，可通过拓展管理  进行取消添加/

添加积木类操作：



可通过排序

排序

按钮，对积木块的顺序进行设置：



(点击右侧上移按钮，积木类上移一个)

(2) 设备管理

①整体介绍

用户可通过设备管理添加主控板和扩展板。添加主控板后，软件会自动出现硬件模式，并可以添加相应的拓展版。

硬件模式下的控制页程序作用于专门的硬件模型，如主控板和扩展板。用户可将硬件程序在线上仿真，同时也可烧录到设备中进行线下仿真。

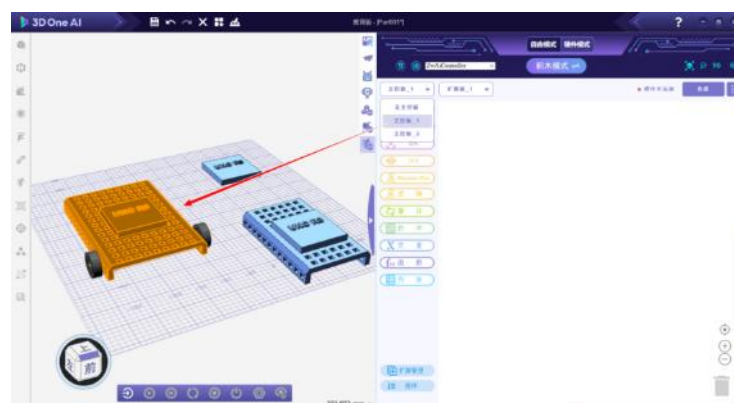
添加硬件模式后，仍然支持用户切换成自由模式，对虚拟场景进行编程。



②虚拟控制对象

硬件模式下提供主控板选项以及扩展板选项，用户可通过下拉选框的形式选择当前控制页

控制的主控板和扩展板模型。选框选中的模型对象会高亮显示。



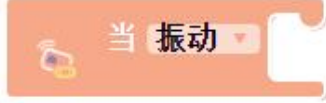
③硬件积木块

开发的积木块包含主控板和扩展板上的传感器积木块以及 micro:bit 常用的逻辑语句积木

块。Python 模式下，编程采用 micro:bit 的代码语句，仿真时转化成 AI 能识别的语句。

Micro: bit 积木块分类具体信息

分类	传感器	参数类型	积木块样式	功能描述
基本	点阵屏	数值		显示数字
		九宫格		显示自定义LED图案
		下拉选框		显示预设图案
		字符串		显示字符串
		/		清空屏幕
		下拉选框; int; int		点亮某个点
		下拉选框		播放预设声音

输入	其他	\		程序无限循环操作
		\		开机时就启动程序
		数值		等待一段时间再进行下一个操作
	可编程按钮	下拉选框		当按下某个按钮, 进行操作
		A/B		某个按钮被按下
	加速度计	下拉选框		当用户让板子呈现某种状态时, 进行操作
		下拉选框		板子呈现某种状态
		下拉选框		某个维度的加速度值
	光敏传感器	\		光敏传感器测到的亮度值
	指南针	\		指南针朝向
	温度传感器	\		温度值

	传感器			
	麦克风	下拉选框		当大声/安静时，进行操作
		\		大声/安静
		\		声音响度
无线通信	无线和蓝牙	数值		只有设置为同一组的 Micro:bit 才能进行无线通信
		下拉选框		打开/关闭无线通信
		字符串		通过无线电发送字符串
		\		当接收到无线数据时，进行操作
		\		无线接收的数据
Maquen	电机	下拉选框; 下拉选框; int		设置电机旋转方向和速度

	下拉选框		某侧电机停止
巡线传感器	下拉选框		巡线传感器某一侧
	下拉选框; 下拉选框		当某一侧的巡线传感器高（白色）/低（黑色）时，进行操作
LED 灯	下拉选框 下拉选框		打开某侧的 LED
舵机	下拉选框; int		设置舵机的角度
超声波传感器	\		超声波测到的距离
红外传感器	\		红外数据
	\		当接收到红外数据时，进行操作

- 由于【红外传感器】需要模拟线下遥控器发送红外数据。因此提供一个自定义遥控按键的方式，可让用户存储按键对应的红外数值。
- **入口：**在硬件模式下，点击 maqueen 分类，点击分类面板下的红外按键定义按钮

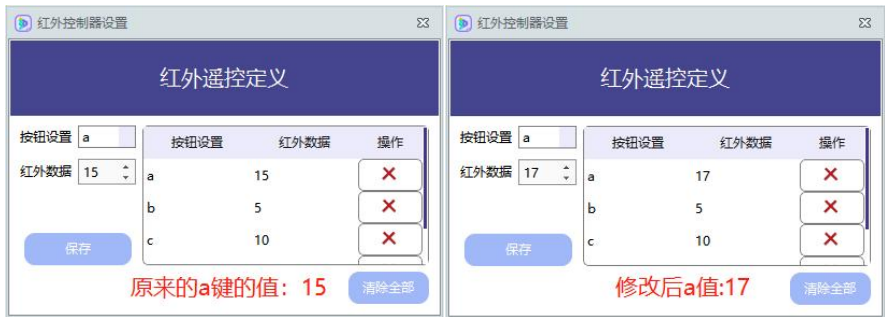


红外按键定义面板分为两部分，绑定按键和按键表。

● 绑定按键

功能名称	介绍
按键名称	下拉选框选择对应按键
红外数值	输入整数数值

一个按键只能绑定一个红外数值，如果重复绑定，在点击存储时，替换原有数值

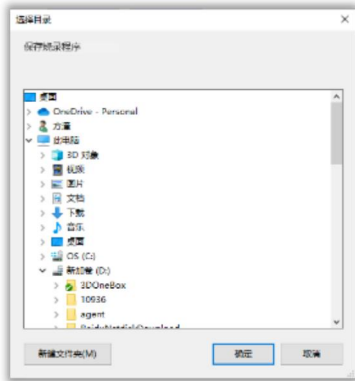


④烧录流程

硬件模式提供烧录功能，在该模式下编写的程序可烧录到硬件设备中进行运行。每个控制页可绑定一个烧录设备，切换控制页会自动切换到对应的烧录设备。

功能名称	介绍
硬件连接状态	● 硬件已连接 ● 硬件未连接

连接/烧录按钮	硬件未连接：点击连接按钮，弹出 设备连接面板  硬件已连接：点击烧录按钮，进行烧录 
更多按钮	点击弹出面板 硬件未连接：①下载文件  硬件已连接：①断开连接②下载文件

	
断开连接	点击后断开连接设备，硬件连接状态显示未连接 
下载文件	点击后弹出保存文件面板，保存.hex 文件 

(3) 其他



切换：点击切换按钮，可切换至本地识别/在线识别模式



清除：点击清除按钮，清除当前控制页所有积木块

保存：点击保存按钮，保存所有积木块程序

4. 编程设置控制器——虚拟传感器



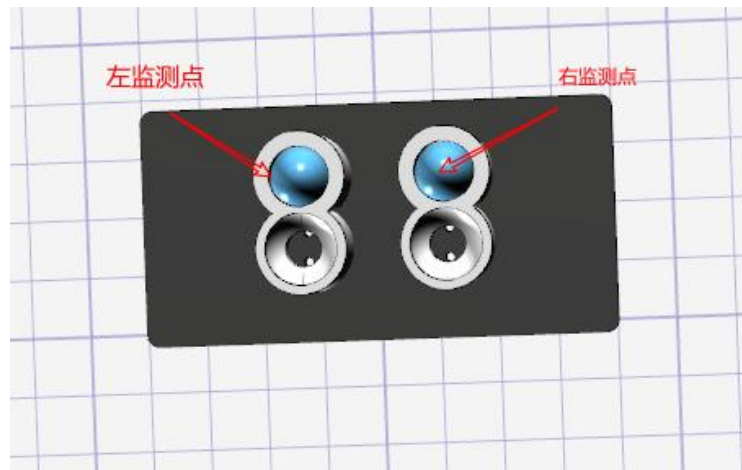
编程控制器主要的积木块分为六类，分别为虚拟传感器、电子件、图像识别、语音识别、机器学习和机械臂。虚拟传感器与电子件使用方法类似，以下为虚拟传感器相关积木接口的说明。

示例程序（循迹传感器）：

- 1.设计一段轨迹，将模型放置于轨迹中间位置；
- 2.将模型设为循迹传感器，设置时需要设置左右 2 个灰度检测点；

3.程序设计逻辑:

- ①如果循迹传感器左检测点检测到黑色，则将向前移动速度设为 0，向左转向 0.005 弧度；
- ②如果循迹传感器右检测点检测到黑色，则将向前移动速度设为 0，向右转向 0.005 弧度；
- ③如果循迹传感器左右检测点均检测为白色，则将向前移动速度设为 3。



4.1 设置距离传感器

此函数用于设置距离传感器启动与否。

示例：

```
zw_set_distance_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	距离传感器的名称（字符串）
State	距离传感器的状态，选项为“Enable” 或“Disable”分别表示启动或不启动。

4.2 获取距离传感器距离

此函数用于获取距离传感器的测量结果，返回传感器模型检测方向上传感器模型表面上的点到碰到的第一个模型的面上的点的距离。

示例：

```
zw_get_distance_sensor_value('body_1')
```

积木块：



参数说明：

SensorName	距离传感器的名称（字符串）
------------	---------------

4.3 设置触碰传感器

此函数用于设置触碰传感器启动与否。

示例：

```
zw_set_touch_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	触碰传感器的名称（字符串）
State	触碰传感器的状态，选项为“Enable”或“Disable”分别表示启动或不启动。

4.4 获取触碰传感器触碰

此函数用于获取触碰传感器的测量结果，返回传感器模型与其他模型触碰与否，若触碰到则返回 1，否则返回其他。

示例：

```
zw_get_touch_sensor_value('body_1')
```

积木块：



参数说明：

SensorName	触碰传感器的名称（字符串）
------------	---------------

4.5 设置力传感器

此函数用于设置力传感器启动与否。

示例：

```
zw_set_force_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	力传感器的名称（字符串）
State	力传感器的状态，选项为“Enable”或“Disable” 分别表示启动或不启动。

4.6 获取力传感器测量力度

此函数用于获取力传感器的测量结果，**返回传感器模型所受的力矢量。**

示例：

```
zw_get_force_sensor_value('body_1')
```

积木块：



参数说明：

SensorName	力传感器的名称（字符串）
------------	--------------

4.7 设置光强传感器

此函数用于设置光线传感器启动与否。

示例：

```
zw_set_light_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	光线传感器的名称（字符串）
State	光线传感器的状态，选项为"Enable" 或"Disable"分别表示启动或不启动。

4.8 获取光强传感器测量光强

此函数用于获取光线传感器的测量结果，**返回传感器模型检测到的光照强度。**

示例：

```
zw_get_light_sensor_value('body_1')
```

积木块：



参数说明：

SensorName	光线传感器的名称（字符串）
------------	---------------

4.9 设置位置传感器

此函数用于设置位置传感器启动与否。

示例：

```
zw_set_position_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	位置传感器的名称（字符串）
State	位置传感器的状态，选项为"Enable" 或"Disable"分别表示启动或不启动。

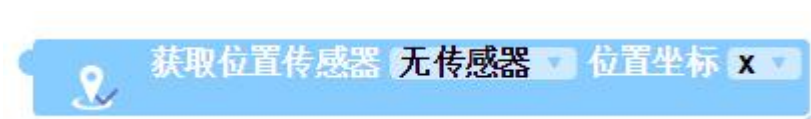
4.10 获取位置传感器坐标

此函数用于获取位置传感器的测量结果，返回传感器模型对应坐标轴上的位置坐标。

示例：

```
zw_get_position_sensor_value('','x')
```

积木块：



参数说明：

SensorName	位置传感器的名称（字符串）
------------	---------------

4.11 设置颜色传感器

此函数用于设置颜色传感器启动与否。

示例：

```
zw_set_color_sensor_enable('body_1', 'Enable')
```

积木块：



参数说明：

SensorName	颜色传感器的名称（字符串）
State	颜色传感器的状态，选项为"Enable" 或"Disable"分别表示启动或不启动。

4.12 获取颜色传感器结果

此函数用于获取颜色传感器的测量结果，返回传感器模型检测到的颜色在给定偏差内是否与给定颜色相符，若相符则返回 True，否则返回 False。

示例：

```
zw_get_color_sensor_value('body_1', zw_vec3(65,128,165), 10)
```

积木块：



参数说明：

SensorName	颜色传感器的名称（字符串）
Color	给定颜色的(R,G,B)向量
Deviation	给定偏差

4.13 启动灰度传感器

此函数用于设置灰度传感器

示例：

```
zw_set_tracking_sensor_enable(' ', 'Enable')
```

积木块：



参数说明：

SensorName	灰度传感器的名称（字符串）
State	灰度传感器的状态，选项为“Enable”或“Disable”分别表示启动或不启动。

4.14 检测灰度传感器左右灰度

灰度传感器是通过对地面的颜色灰白程度来判断，并控制实体运动的。

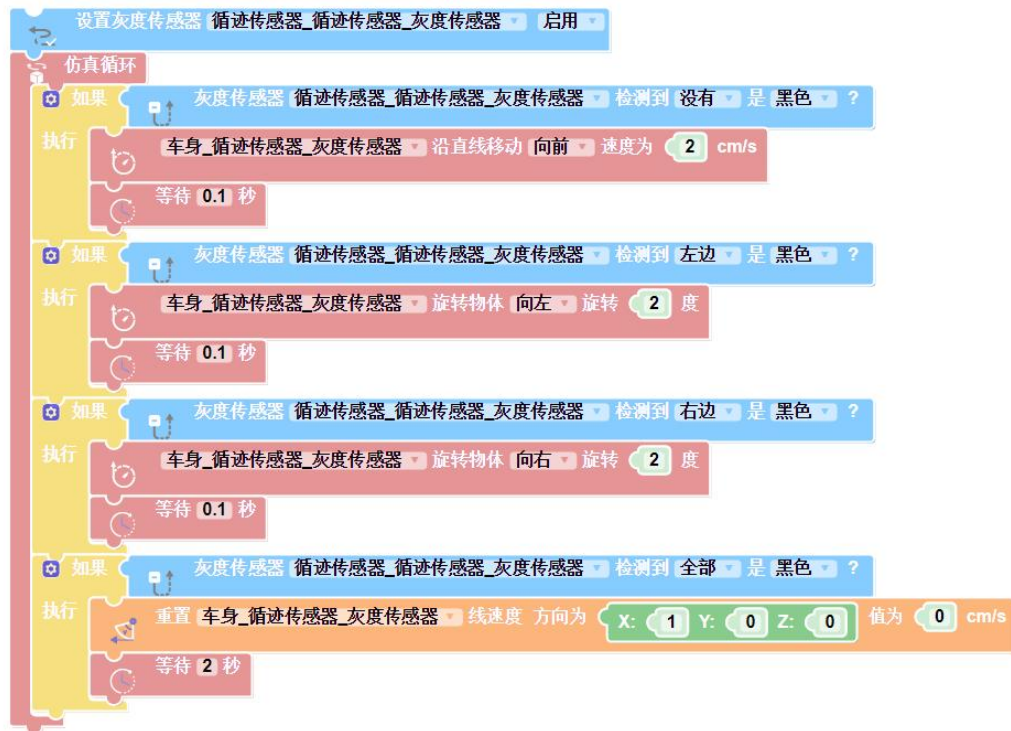
示例：

```
zw_get_tracking_sensor_detect(' ', 'left', 'black')
```

积木块：



案例演示：



根据传感器左右两侧检测点检测到的黑色程度来判定车身的移动方向。

参数说明：

SensorName	灰度传感器的名称（字符串）
left right all none	左边或右边
Color	黑色或白色

4.15 启动循迹传感器

此函数用于启动循迹传感器

示例：

```
zw_set_track_sensor_enable(' ', 'Enable')
```

积木块：



参数说明：

SensorName	循迹传感器的名称（字符串）
State	灰度传感器的状态，选项为"Enable" 或"Disable"分别表示启动或不启动。

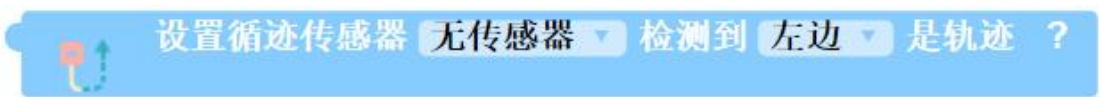
4.16 检测循迹传感器左右轨迹

传感器根据地面的轨迹分布情况来决策车身的运动方向。

示例：

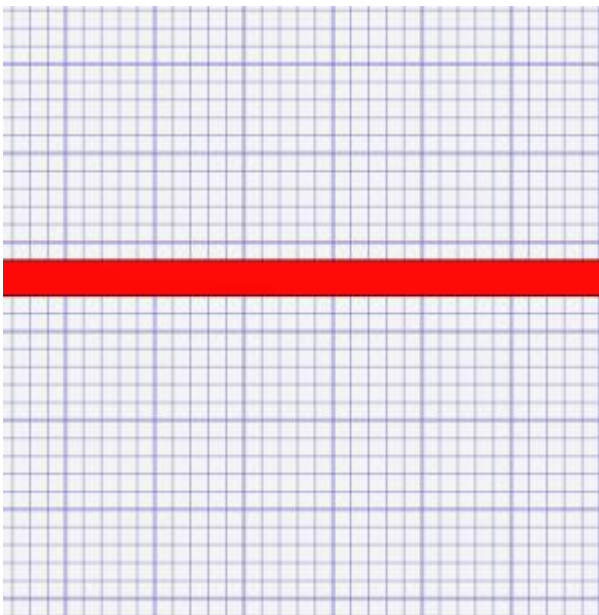
```
zw_get_track_sensor_detect(' ', 'left')
```

积木块：

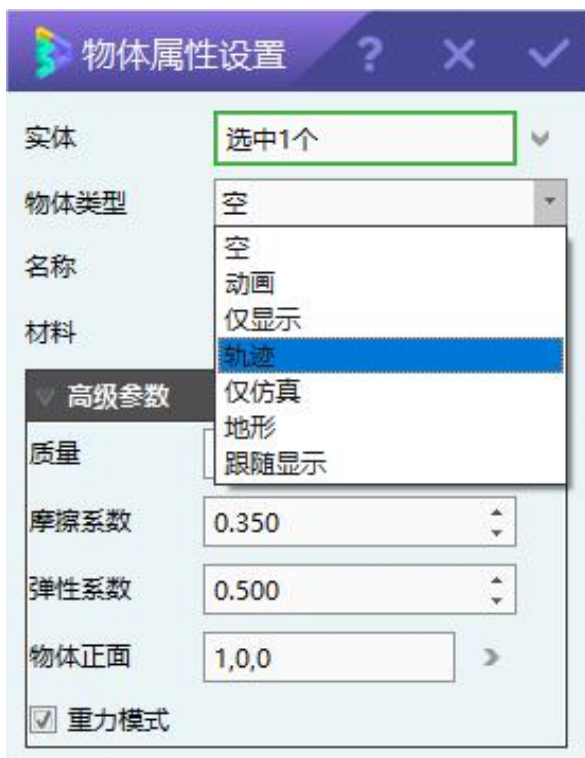


特殊说明：

循迹传感器的识别是通过识别轨迹来运转的，所以就算循迹轨迹不是黑色的也没有影响。



只需要给这条轨迹的物理属性设置为轨迹即可。



循路逻辑与灰度传感器一致。

参数说明：

SensorName	循迹传感器的名称（字符串）
------------	---------------

left right all none	左边、右边、全部或没有
---------------------	-------------

5.编程控制器——电子件

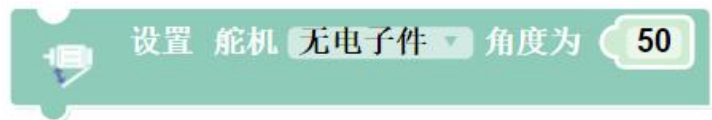
5.1 设置舵机角度

此函数用于设置伺服马达的角度。

示例：

```
zw_set_servomotor_angle('body_3', 50)
```

积木块：



参数说明：

Name	伺服马达的名称（字符串）
Angle	伺服马达的角度

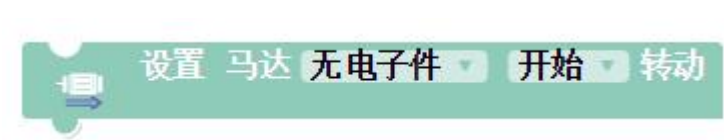
5.2 设置马达转动

此函数用于设置马达启动与否。

示例：

```
zw_set_onewaymotor('body_3', 'ON')
```

积木块：



参数说明：

Name	马达的名称（字符串）
State	马达的转动状态, 选项为"ON"或"OFF"分别表示开始或停止转动。

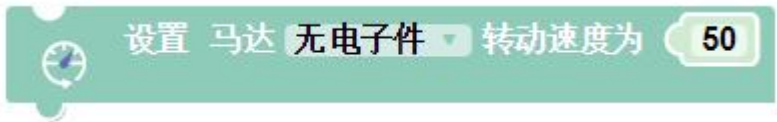
5.3 设置马达转动速度

此函数用于设置马达的转动速度。

示例：

`zw_set_onewaymotor_speed('body_3', 50)`

积木块：



参数说明：

Name	马达的名称（字符串）
Speed	马达的转动速度

5.4 设置马达转动速度百分比

此函数用于设置马达的转动速度百分比，对应于设置电子件时马达的标准速度

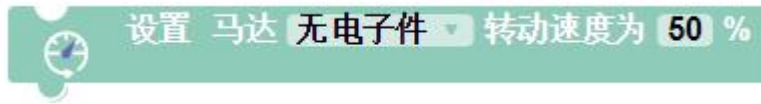


(与标准速度对应)

示例：

```
zw_set_onewaymotor_percentage('body_3',50)
```

积木块：



5.5 设置左、右马达转动速度百分比

此函数用于设置小车的左右马达速度百分比，对应于设置电子件时马达的标准速度

示例：

```
zw_set_onewaymotor_percentage2('body_3','body_4',50,50)
```

积木块：



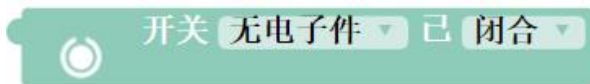
5.6 获取开关状态

此函数用于判断开关的开闭状态。

示例：

```
zw_check_switch('body_3','ON')
```

积木块：



参数说明：

SwitchName	开关的名称（字符串）
State	开关的状态，选项为“ON”或“OFF”分别表示打

	开或闭合。
--	-------

5.7 设置开关状态

此函数用于设置开关的状态。

示例：

```
zw_set_switch('body_3', 'ON')
```

积木块：



参数说明：

SwitchName	开关的名称（字符串）
State	开关的状态，选项为“ON”或“OFF”分别表示打开或闭合。

5.8 设置 LED 状态

此函数用于设置 LED 灯的开关状态。

示例：

```
zw_set_LED('body_3', 'ON')
```

积木块：



参数说明：

LEDName	LED 灯的名称（字符串）
---------	---------------

State	LED 灯的状态, 选项为"ON"或"OFF"分别表示 打开或关闭。
-------	---------------------------------------

5.9 设置电风扇

此函数用于设置电风扇的开关状态。

示例：

```
zw_set_electricfan('body_3', 'ON')
```

积木块：



参数说明：

FanName	电风扇的名称（字符串）
State	电风扇的状态, 选项为"ON"或"OFF"分别表示 打开或关闭。

5.10 设置电风扇转动速度

此函数用于设置电风扇的转速。

示例：

```
zw_set_electricfan_speed('body_3', 50)
```

积木块：



参数说明：

FanName	电风扇的名称（字符串）
Speed	电风扇的转速。

5.11 设置电子屏状态

此函数用于设置电子屏的状态。

示例：

```
zw_set_electricdisplay_state('body_1', 'ON')
```

积木块：



参数说明：

FanName	电子屏的名称（字符串）
State	电子屏的状态，选项为“ON”或“OFF”分别表示打开或关闭。

5.12 设置电子屏显示内容

此函数用于设置电子屏的显示内容。

示例：

```
zw_set_electricdisplay_content('body_1', '3D One AI', 10, '幼圆')
```

积木块：



参数说明：

FanName	电子屏的名称（字符串）
text	显示内容
size	文字大小的大小
font	文字的字体

5.13 设置扬声器状态

此函数用于设置扬声器状态

示例：

```
zw_set_speaker_state('body_1','Play')
```

积木块：



参数说明：

FanName	扬声器的名称（字符串）
State	扬声器的状态，选项为“play”、“pause”、“stop”分别表示播放、暂停、停止。

5.14 设置扬声器播放内容

此函数用于设置扬声器播放内容，用户可通过下拉菜单选择软件内置音频，也可以通过导入音频文件使用。

软件内置音频中按钮、打碎、掉落、鼓、滑动、完成来自于 100Audio.com，错误、击

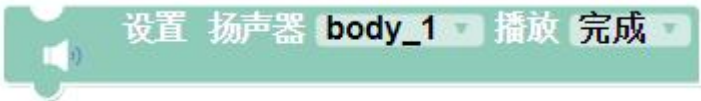
打 1、击打 2、清晨、小雨、循环、正确来自于 aigei.com

示例：

```
zw_set_speaker_play('body_1','D:/10.功能验收/3D One AI V2.5/3D One AI
```

```
V2.5/supp/AIblockly/media/Electronic/1-完成.mp3')
```

积木块：



参数说明：

FanName	扬声器的名称（字符串）
Play	播放的音频内容，用户可选择下拉菜单中的内置音频或自行上传音频文件

6. 编程控制器——图像识别

目前图像识别提供了百度在线识别接口以及本地离线识别。（具体可见第四章第四节百度在线识别功能）

示例 1（图像识别）：

- ④ 程序设计：启动电脑摄像头后，当按下键盘 a 键，启动图像识别，当图像识别结果包含人时，则红色方块向前移动，当图像识别结果包含狗时，则蓝色方块向前移动。

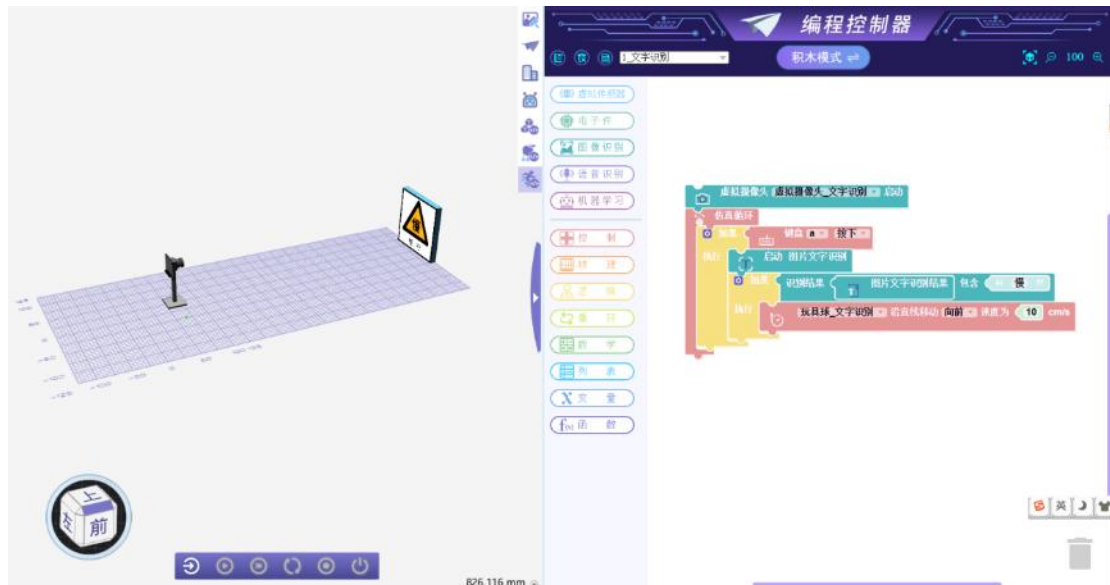


- ② 进入仿真环境，软件检测到程序包含图像模块则会自启动图像识别栏（右侧），将要识别的图像置于摄像头前，按下 a 键，软件开始进行图像的获取以及识别：

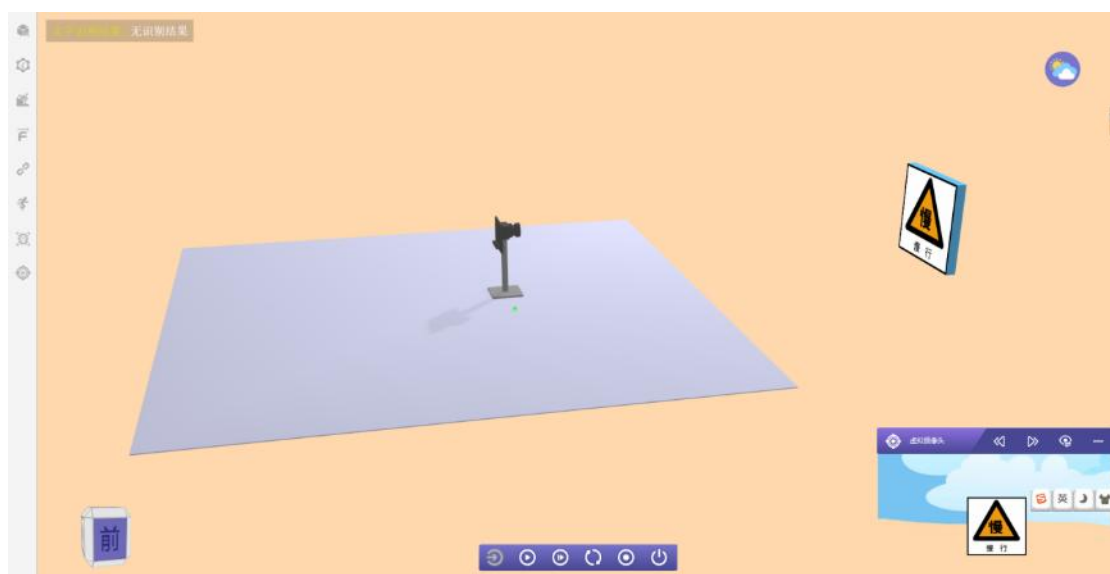


示例 2（文字识别）：

- ①需要设置物体 1 作为虚拟摄像头, 将要识别的文字放置在虚拟摄像头可以观察到的位置上。
- ②程序设计：启动虚拟摄像头后，按下键盘 A 键，启动文字识别，当文字识别结果包含“慢”字，物体玩具球则向前移动。



③进入仿真，按下键盘 A 键，启动文字识别，当识别结果含有“慢”时，绿色小球则向前移动：



6.1 启动电脑摄像头

此函数用于打开电脑设备的摄像头，在编程时需要放在程序的最开始位置，在仿真循环积木块之上。

```
zw_image_set_pc_camera()
```

积木块：



6.2 启动虚拟摄像头

此函数用于设置虚拟摄像头电子件启动，在编程时需要放在程序的最开始位置，在仿真循环积木块之上。

示例：

```
zw_image_set_virtual_camera('body_1')
```

积木块：



参数说明：

CameraName	虚拟摄像头的名称（字符串）
------------	---------------

6.3 启动图片对象检测

此函数用于开始识别图片。

```
zw_image_recognize_objectdetection()
```

积木块：



6.4 图片对象检测结果

此函数用于获取图像识别的结果，返回字符串类型的具体类别，如猫，狗，车等。

```
zw_image_recognize_objectdetection_result()
```

积木块:



6.5 启动图片文字识别

此函数用于开始识别图片中的文字。

`zw_image_recognize_text()`

积木块:



6.6 图片文字识别结果

此函数用于获取文字识别的结果, 返回字符串类型的具体类别, 支持简体中文、英文、电脑文字, 不支持繁体中文与手写文字, 要求识别图像背景较干净。

`zw_image_recognize_text_result()`

积木块:



6.7 启动图片交通信号灯识别

此函数用于启动图片交通信号灯识别。

`zw_image_recognize_trafficlight()`

积木块:



6.8 交通灯识别结果

此函数用于获取交通灯的识别结果，识别结果与传入参数一致，则返回 true，否则返回 false。

```
zw_image_recognize_trafficlight_result('Red')
```

积木块：



参数说明：

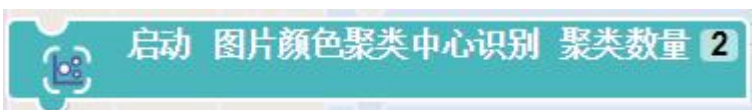
Color	交通的颜色（字符串）
-------	------------

6.9 启动图片颜色聚类中心识别

此函数用于启动颜色聚类中心识别，参数表示每种颜色将其分为的类数目。

```
zw_image_recognize_color_clustercoordinates(2)
```

积木块：



参数说明：

Num	每种颜色的聚类数目
-----	-----------

6.10 聚类中心位置中心的识别，

返回识别到的聚类中心的坐标 x 或 y 值

zw_image_recognize_color_clustercoordinates_result('x')

积木块:



参数说明:

Position	聚类中心的坐标 x 或 y 值
----------	-----------------

6.11 最近颜色聚类中心位置

距离最近的颜色的聚类中心位置 (左或右) 如果和参数相同则返回 true, 否则返回 false

zw_image_recognize_color_clustercoordinates_location('Left')

积木块:



参数说明:

Position	Left right 或者 middle
----------	----------------------

6.12 最近聚类中心颜色

距离最近的聚类中心的颜色如果和参数相同则返回 true, 否则返回 false;

zw_image_recognize_color_clustercoordinates_color('Red')

积木块:



参数说明:

Color	距离最近聚类中心的颜色 (字符串)
-------	-------------------

6.13 启动图片颜色识别

此函数用于开始识别图片中的文字。

zw_image_recognize_colordetect()

积木块：



6.14 图片颜色识别结果

此函数用于获取图片颜色识别的结果，结果取拍到的照片中颜色比例最多的颜色

zw_image_recognize_colordetect_result('Red')

积木块：



参数说明：

Color	图片识别的颜色（字符串）
-------	--------------

6.15 启动图像循路

此函数用于启动图像循路功能

zw_image_recognize_trackingline()

积木块：



6.16 判定循路结果

此函数用于与图像循路得到的结果对比，然后对物体进行相应操作。

`zw_image_recognize_trackingline_result('Left')`

积木块：



仿真示意图：启动循迹功能后，左上角会根据计算结果给出应该移动的方向。

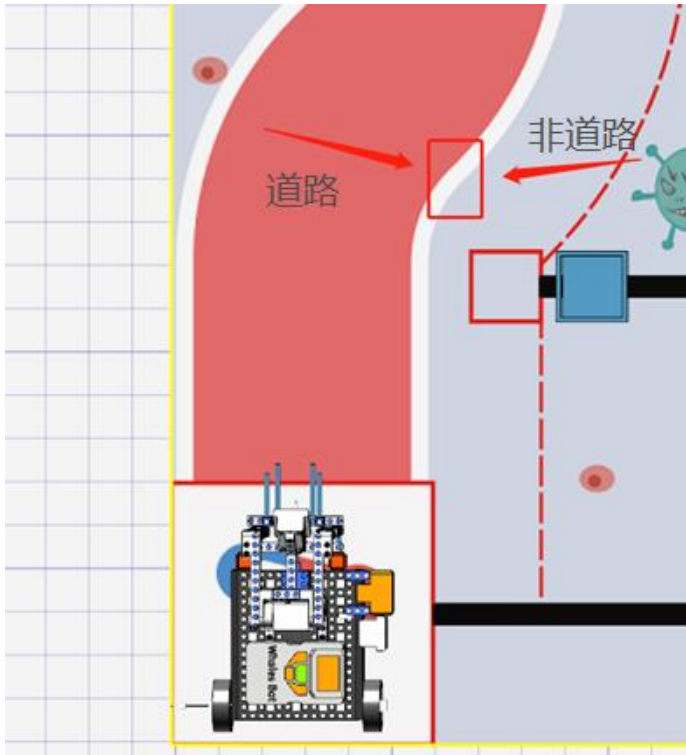


参数说明：

left right middle	左边、右边或中间
-------------------	----------

6.17 图像循路使用简介

道路与非道路需要有较为明显的差异，比如有颜色对比，具体颜色没有要求。道路与非道路之间不一定需要辅助线判断。注意：开始循路的时候需要在控制器里将主摄像头调为车前循路摄像头。



6.18 识别结果包含

此函数用获取图像识别结果并与用户输入的字符串进行比对,若结果包含用户输入内容,则接口返回真,否则返回假。

`zw_image_result_contains(zw_image_recognize_prediction_result(),text)`

积木块：



参数说明：

Text	给定的文本（字符串）
------	------------

6.19 条码模块

条码模块包含条码生成和识别两个功能，在编程界面生成条码，将条码通过贴图等形式放入建模场景中。进入仿真环境后，识别虚拟摄像头拍摄的条码，调用条码的内容用于运算、赋值或显示等操作。

(1) 条码生成

点击条码生成按钮，打开条码生成界面，界面样式如下。



可切换二维码和条形码两种不同类型的条码，默认显示二维码。在对应输入框输入文本、数值等内容后，点击生成按钮，自动生成二维码或条形码，并在下方预览框内显示。点击下载按钮即可进行下载。



(2) 条码识别

图像识别分类下新增两个条码积木块：启动条码识别、条码识别结果。

启动条码识别积木块是控制系统开始条码识别。



条码识别结果积木块是对条码识别结果进行调用，可选择调用的数据：文本和数值。



启动条码识别后，左上角将会显示识别状态和识别结果。



6.20 启动植物/动物识别

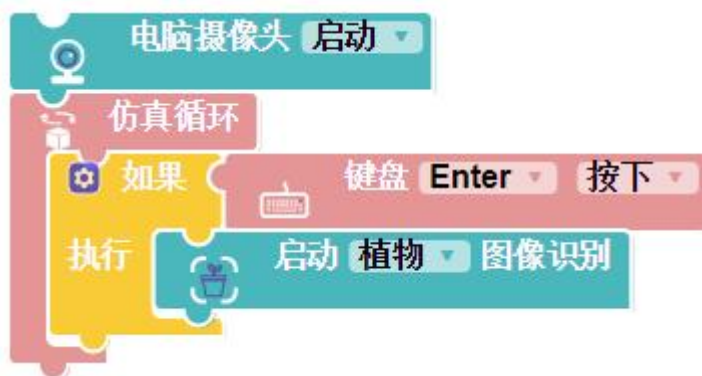
此函数用于开始进行植物/动物识别，可通过下拉菜单选择识别植物/动物

zw_image_recognize_xxx_recognize_plant()

积木块：

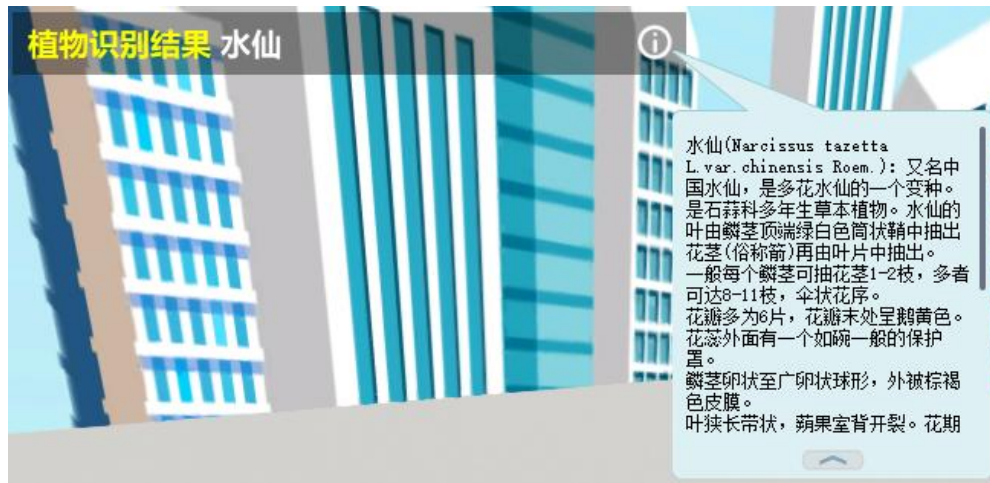


示例程序：



程序效果：

仿真环境左侧显示识别结果的名称，点击右侧  按钮，可查看该品种信息。



6.21 植物/动物识别结果

此函数用于存储识别的植物/动物的名称

zw_image_recognize_xxx_recognize_result_plant()

积木块：



6.22 启动手势识别

此函数用于开启手势识别功能。

zw_image_recognize_gesture()

积木块：



6.23 手势识别结果是否是指定内容

此函数用于对比手势识别结果是否为指定结果，结果可下拉选择，有：数字 1、剪刀、数字 3、数字 4、布、数字 6、数字 7、数字 8、数字 9、OK、石头、祈祷、作揖、作别、单手比心、点赞、我爱你、掌心向上、双手比心、Rock。

`zw_image_recognize_gesture_result_is("数字 1")`

积木块：



7. 编程控制器——语音技术

目前语音识别提供了百度在线识别接口以及本地离线识别。（具体可见第四章第四节百度在线识别功能）

示例程序：

当按下键盘 a 键时，启动语音识别 2 秒，当结果包含“前进”时，物体 1 则向前移动速度为 10mm/s



7.1 电脑麦克风启动

此函数用于启动电脑麦克风。

zw_speech_set_microphone()

积木块：



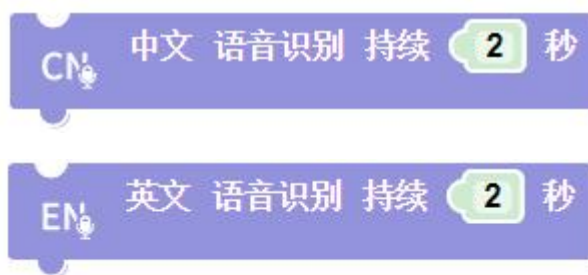
7.2 语音识别秒数

此函数用于启动语音录制和识别功能，参数为录音时长。

中文：zw_speech_recognize(int)

英文：zw_speech_english_recognize(int)

积木块：



参数说明：

int	识别时长
-----	------

7.3 语音识别结果包含

此函数用于判断检测到的语音中是否含有给定的文本。

zw_is_voice_contains_text(Text)

积木块：



参数说明：

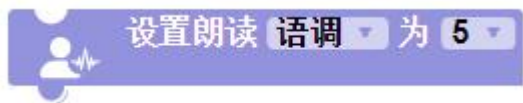
Text	给定的文本（字符串）
------	------------

7.4 设置朗读语调/语速

此函数用于设置朗读语调/语速，可选择 0-15 间的整数。

zw_speech_read_set_tone(5)

积木块：

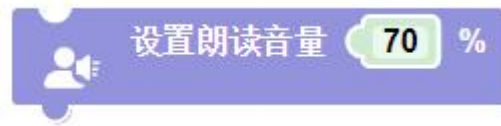


7.5 设置朗读音量

此函数用于设置朗读音量，可输入 ≥ 0 的数值

zw_speech_read_set_volume(70)

积木块：



7.6 设置朗读人声

此函数用于设置朗读人声，可选择磁性男声、成熟女声、成熟男声、可爱女童

```
zw_speech_read_voice(1)
```

积木块：



7.7 朗读

此函数用于启动朗读，参数为输入的文本

```
zw_speech_read('你好')
```

积木块：



7.8 语音识别结果

此函数该用于存储语音识别结果，可与朗读搭配使用

```
zw_speech_speech_recognize_result(1)
```

积木块：



8. 编程控制器——机器学习

8.1 训练模型

用户自行训练模型，训练模型过程：新建 3 个需要训练的分类，进行多个角度的拍摄，让机器认识这个物品，收集足够多的图片后，机器可以自行分辨物品是哪个分类，实际上是把图像识别中的机器训练过程简化为界面交互形式。用户可利用这个分类结果进行编程，实现一定的控制效果。

示例程序：

- ① 通过编程控制器机器学习分类下的“训练模型”启动机器学习页面：

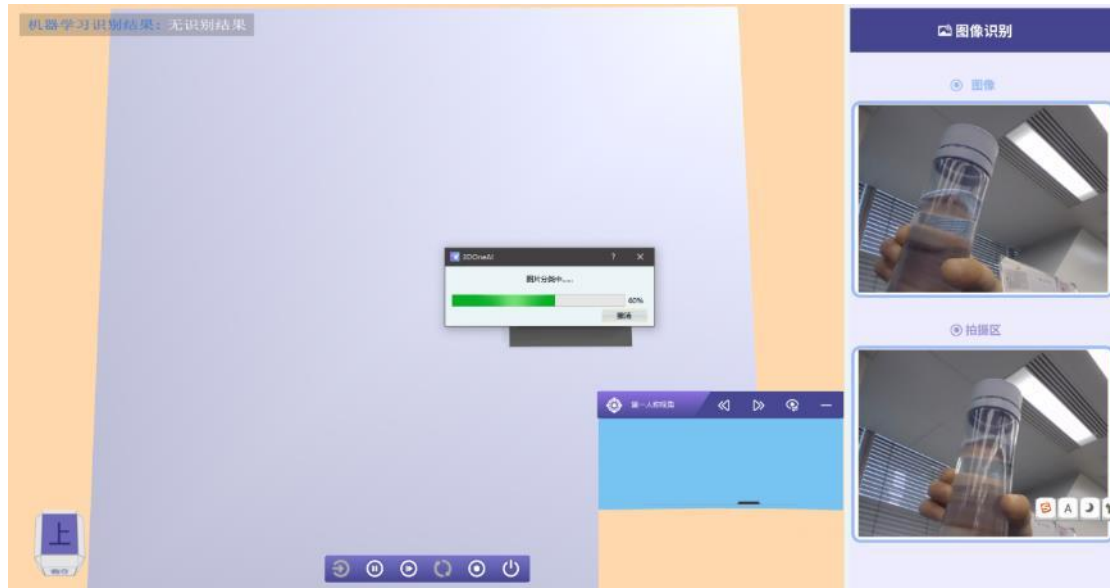


②按照页面提示进行模型的拍摄，拍摄足够多的照片后，软件自动进行特征点提取，让机器学习模型的特征，从而达到分类。流程可分为训练分类——编程——进入仿真使用三个阶段。

③程序设计：启动电脑摄像头后，当按下键盘 A 键，软件开始分类摄像头所拍摄到的图像，当拍摄图像认为是某分类时，物体 1 线速度方向改为 20mm/s。



- ③ 进入仿真，将要识别的图像放置摄像头前，按下 A 键，软件获取图像并进行分类对比：



8.1.1 训练分类阶段

- (1) 图片上传/拍照：可以点击上传/拍照/批量上传来进行素材的上传。

批量上传支持 zip 格式

- (2) 命名分类

8.1.2 编程阶段

- (1) 按需进行训练设置



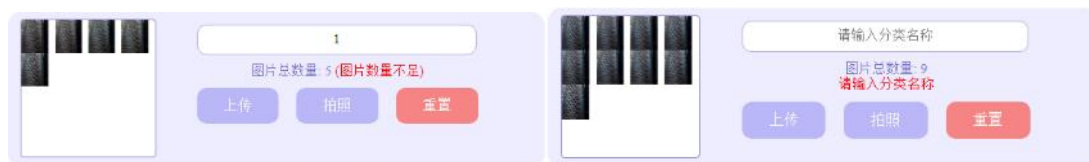
点击该按钮则展开训练设置:默认为 Mobilenet, 可以手动选择 opencv; 若选择了 opencv, 则出现多个选项 (可多选)。

(4) 训练模型



点击训练模型按钮后则开始训练, 若没达成训练要求会出现相应提示。在保存结果之前会出现学习的正确率。【输入的图片分为训练集和测试集(8:2), 正确率根据测试集进行计算, 测试集为每一分类下的后 20%的图片】

注意: 用户需要输入分类名称, 且每个分类至少上传八张照片, 否则无法进行模型训练





训练模型结果页

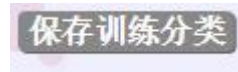
8.1.3 修改模型



在机器学习的选择模型处，点击修改图标可以重新进入页面进行修改。

8.2 保存训练分类

保存训练好的分类到本地磁盘;



8.3 导入训练分类

从本地导入训练好的分类;



8.4 电脑摄像头

此函数用于打开电脑设备的摄像头, 在编程时需要放在程序的最开始位置, 在仿真循环积木块之上。

```
zw_image_set_pc_camera(1)
```

积木块:



8.5 机器学习摄像头启动

此函数用于启动机器学习的摄像头。

```
zw_image_set_virtual_camera('')
```

积木块:



8.6 虚拟环境拍摄分类图片

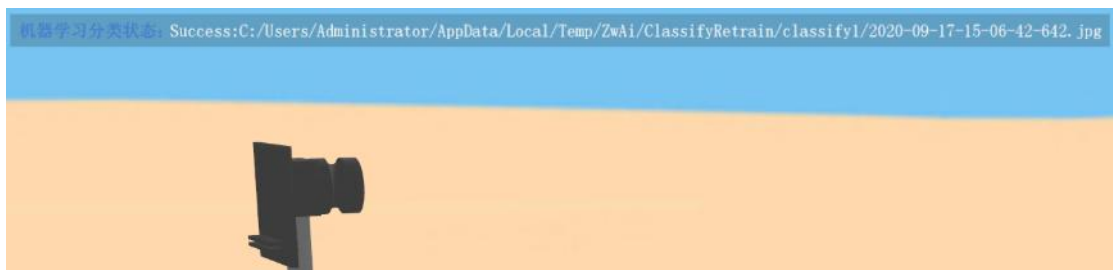
用户可以自定义分类名称，进入虚拟环境后，会以该分类名称进行虚拟环境中的图片拍摄。

`zw_machinelearning_capture("")`

积木块：



仿真示意图：将拍摄到的画面存到上方目录中



8.7 开始分类

此函数用于开始通过机器学习进行分类识别。

`zw_machinelearning_classify()`

积木块：



8.8 分类结果

此函数用于获得机器学习的分类识别的结果。

`zw_machinelearning_classify_result()`

积木块：



8.9 分类结果信心

此函数用于返回机器学习模型分类结果的可信度，即识别该类图像时的识别率。

`zw_machinelearning_classify_confidence()`

积木块：



8.10 分类结果

此函数用于将训练好的机器学习模型应用到预测中。

`zw_machinelearning_classify_judgement()`

积木块：



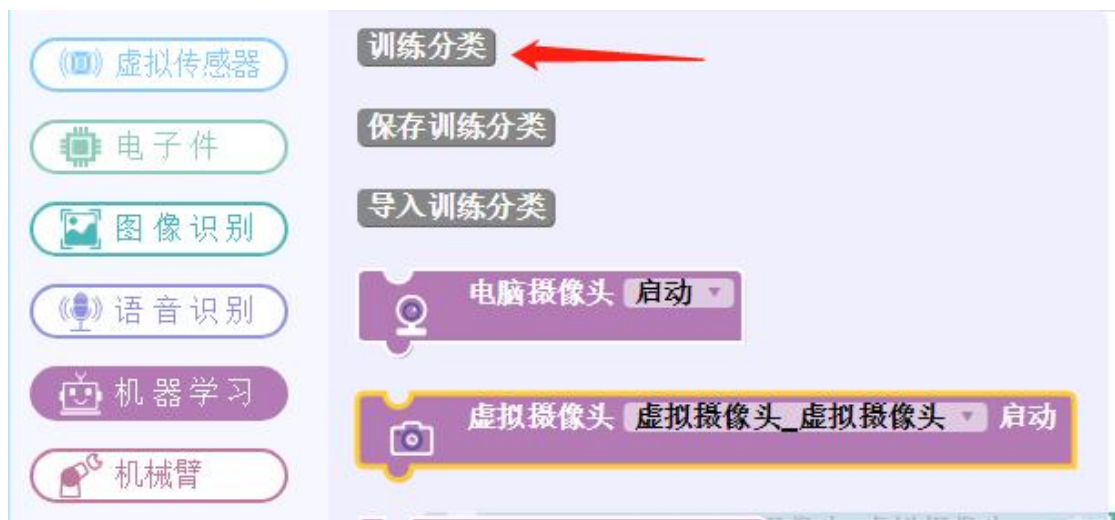
8.11 虚拟机器学习案例演示

进入仿真环境，将镜头对准分类 1，点击 a 开始拍摄。



每拍摄一次图片都会自动保存到本地

退出仿真打开机器学习模块



分类的图片和名称已经自动加载进机器学习模块中



点击开始训练模型并等待训练结束, 训练成功后可以点击保存训练模型到本地, 方便下次导入使用



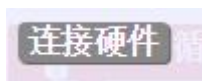
点击导入训练分类, 选中保存好的文件夹可以自动导入已经做好的分类。

9. 编程控制器——机械臂

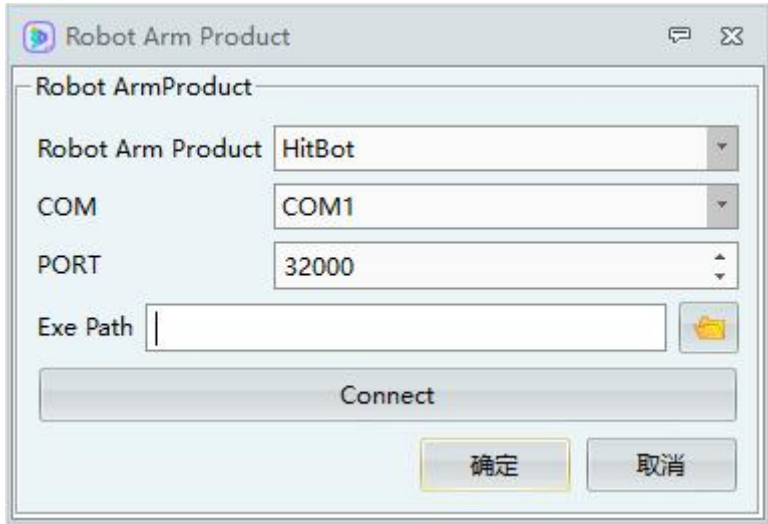
9.1 连接硬件

点击连接硬件可以控制实物机械臂。

积木块:



示例:

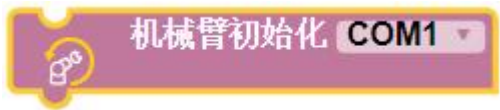


9.2 机械臂初始化

示例：

```
zw_robotarm_initial('1')
```

积木块：



参数说明：

参数	类型	默认值	说明
COM	str	无	需要初始化的 COM 口号

9.3 机械臂气爪合起

示例：

```
zw_robotarm_gripconfig('夹具', True)
```

积木块：



参数说明：

参数	类型	默认 值	说明
body_name	str	无	夹具的名称
state	boolean	False	True 为 Open，False 为 Close

9.4 设置夹爪旋转

例子：

```
zw_robotarm_setrotatedeg('夹具', 5)
```

积木块：



参数说明：

参数	类型	默认 值	说明
Body_name	Str	无	夹具的名称
degree	float	无	夹具转盘转动的角度

9.5 获取机械臂关节角度

示例：

```
zw_robotarm_getjointdeg('机身', 1)
```

积木块：



参数说明：

参数	类型	默认值	说明
Body_name	Str	无	机械臂的名称
joint_deg	float	无	选择机械臂的 joint

9.6 获取夹爪旋转电机的角度

示例：

```
zw_robotarm_getRmotordeg('夹具', )
```

积木块：



参数说明：

参数	类型	默认值	说明
Body_name	Str	无	夹具的名称

9.7 获取坐标轴角度

示例：

```
zw_robotarm_getcoordinatepos('机身', 'x')
```

积木块：



参数说明：

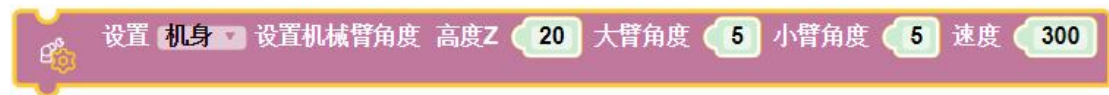
参数	类型	默认值	说明
Body_name	Str	无	机械臂的名称
xyz	tuple	无	选择获取坐标的轴

9.8 设置机械臂角度和高度

示例：

```
zw_robotarm_moveangle('机身', 20,5,5,300)
```

积木块：



参数说明：

参数	类型	默认值	说明
Body_name	Str	无	机械臂的名称
Z	float	无	夹具的高度

joint1_Angle	float	无	joint1 转动的角度
joint2_Angle	float	无	joint2 转动的角度
speed	float	无	机械臂转动的速度

9.9 等待机械臂运动停止

示例：

```
zw_robotarm_waitstop('机身', )
```

#标准用法 1：机械臂运动是否停止

```
if zw_robotarm_waitstop('机身', ):
```

```
    pass
```

#标准用法 2：当机械臂停止运动了以后做

```
while zw_robotarm_waitstop('机身', ):
```

```
    pass
```

#标准用法 3：当机械臂没有停止运动时，一直给予等待时间

```
while not zw_robotarm_waitstop('机身', ):
```

```
    zw_sleep(0.1)
```

积木块：



#积木块用法图示 1：



#积木块用法图示 2:



#积木块用法图示 3:



参数说明:

参数	类型	默认值	说明
Body_name	Str	无	夹具的名称

9.10 设置转盘轴转动角度

示例:

```
zw_robotarm_turntablemove('转盘轴', 22.5)
```

积木块:



参数说明：

参数	类型	默认值	说明
Body_name	Str	无	转盘轴的名称
move_degree	int	无	转盘转动的角度

10.编程控制器——控制

10.1 键盘按下

示例：

```
zw_get_keyboard('a', 'Press')
```

积木块：



参数说明：

Keyboard	键盘任一字母、方向键、空格键、回车键或任一按键
State	按下

10.2 鼠标按下

示例：

```
zw_get_mouse('Left', 'Down')
```

积木块：



参数说明：

Mouse	鼠标左键、鼠标中键或鼠标右键
State	按下、松开、点击

10.3 设置物体移动速度和方向

示例：

```
zw_move_linearmove(' ','forward',10)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Direction	向前或向后
Speed	物体前后移动速度

10.4 设置物体旋转角度与方向

示例：

zw_move_rotatemove(' ','Left',5)

积木块：



参数说明：

Name	物体的名称（字符串）
Direction	向左或向右
Speed	物体左右移动速度

10.5 设置当前时间

示例：

zw_set_current_time(0,0,0)

积木块：



参数说明：

Hour	设置当前小时
------	--------

Minute	设置当前分钟
Second	设置当前秒数

10.6 重置当前时间

示例：

```
zw_reset_current_time(0,0,0)
```

积木块：



参数说明：

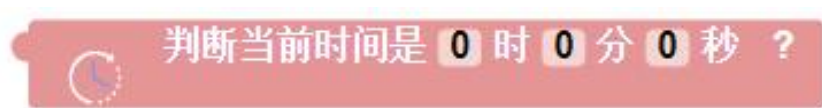
Hour	重置当前小时
Minute	重置当前分钟
Second	重置当前秒数

10.7 判断当前时间

示例：

```
zw_is_match_current_time(0,0,0)
```

积木块：



用于判断当前时间是否为某时间，作为条件。

参数说明：

Hour	获取当前小时
Minute	获取当前分钟
Second	获取当前秒数

10.8 设置等待时间

示例：

zw_sleep(0)

积木块：



参数说明：

Second	等待秒数
--------	------

10.9 当前计时

示例：

zw_elapsed_time()

积木块：



以时分秒的形式来显示目前累计的时间，比如从启动开始一直叠加时间，可以与电子显示屏搭配使用。



10.10 获取仿真时间

示例：

```
zw_elapsed_seconds('HOUR')
```

积木块：



计算从仿真开始到当前所用的所有时间。

10.11 重置仿真

在控制器中编程控制单个物体重置仿真。

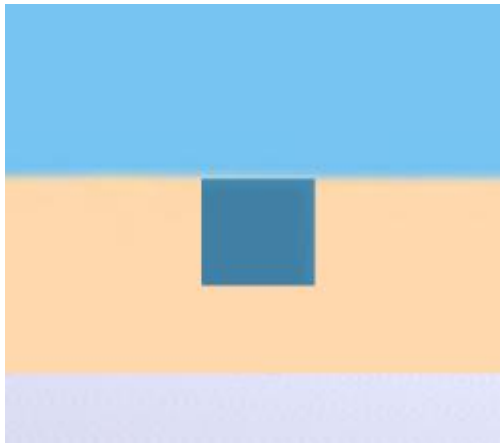
示例：

```
zw_reset_single_entity('')
```

积木块：



仿真示意图：受力漂浮起来的物体在通过重置仿真积木块后回到原位。



参数说明：

Name	需重置物体的名称（字符串）
------	---------------

10.12 退出控制器

在控制器中加入条件退出控制器积木块，可以实现单个控制器在一定条件下停止作用。该积木块可以实现退出改控制器仿真循环，减少仿真计算量，缓解电脑 cpu。

示例：

`zw_exit()`

积木块：



10.13 隐藏物体及其关联积木块

在仿真中可以通过控制器，动态的控制一个物体及其关联物体的隐藏。

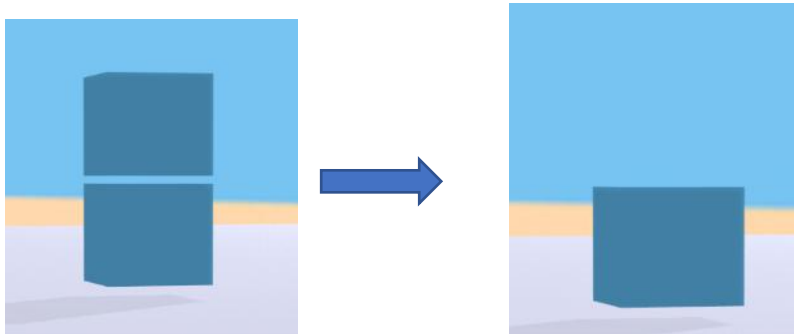
示例：

`zw_hide_entity(' ',True)`

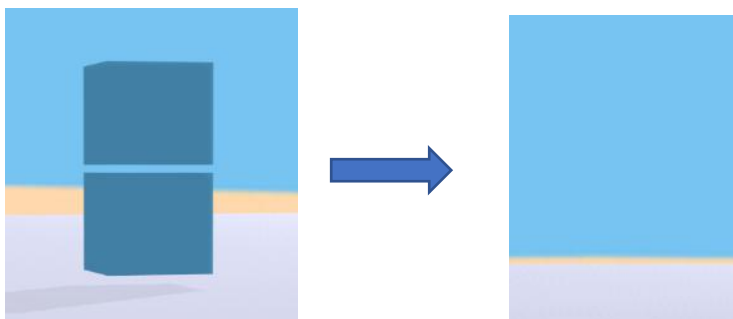
积木块：



仿真示意图：两个物体没有关节连接，隐藏上面的实体，下面的物体不会消失。



仿真示意图：两个物体通过固定关节连接，隐藏上面的实体，下面的物体会通过关节关联消失。



参数说明：

Name	物体的名称（字符串）
State	关联物体的状态，选项为“真”或“假”分别表示隐藏或不隐藏。

10.14 显示物体

在仿真中可以通过控制器，恢复被隐藏的物体。

示例：

```
zw_show_entity('')
```

积木块：



参数说明：

Name	需显示物体的名称（字符串）
------	---------------

10.15 在特定时间显示特定信息

示例：

```
zw_show_time_msg('Message',6,0,0,0)
```

积木块：



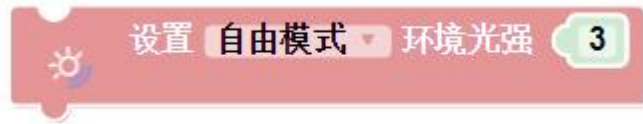
可以与显示屏搭配使用，在设定的时间显示想要的消息，并设定显示持续时间。

10.16 设置环境光强

示例：

```
zw_set_new_lightintensity(0,3)
```

积木块：



10.17 设置环境温度

示例：

```
zw_set_temperature(21)
```

积木块：



10.18 仿真循环

示例：

```
while zw_step():
```

```
    pass
```

积木块：



11. 编程控制器——物理

11.1 重置物体线速度方向与大小

示例：

zw_body_set_linear_vel(' ', zw_vec3(1, 0, 0), 20)

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis direction	线速度的 x 轴方向
Y axis direction	线速度的 y 轴方向
Z axis direction	线速度的 z 轴方向
Speed	线速度大小

11.2 重置物体角速度方向与大小

示例：

zw_body_set_angular_vel(' ', zw_vec3(0, 0, 1), 20)

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis direction	角速度的 x 轴方向
Y axis direction	角速度的 y 轴方向
Z axis direction	角速度的 z 轴方向

Speed	角速度大小
-------	-------

11.3 获取物体线速度

示例：

```
zw_body_get_linear_vel(' ')
```

积木块：



参数说明：

Name	需获取线速度的物体的名称（字符串）
------	-------------------

11.4 获取物体角速度

示例：

```
zw_body_get_angular_vel(' ')
```

积木块：



参数说明：

Name	需获取角速度的物体的名称（字符串）
------	-------------------

11.5 重置物体受力方向与大小

示例：

```
zw_body_set_force(' ', zw_vec3(1, 0, 0), 20)
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis direction	力的 x 轴方向
Y axis direction	力的 y 轴方向
Z axis direction	力的 z 轴方向
Force	力大小

11.6 重置物体受力矩方向与大小

示例：

```
zw_body_set_torque(' ', zw_vec3(1, 0, 0), 20)
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis direction	力矩的 x 轴方向
Y axis direction	力矩的 y 轴方向
Z axis direction	力矩的 z 轴方向
Torque	力矩大小

11.7 重置物体重力模式

示例：

```
zw_body_set_gravity_mode(' ', True)
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
State	物体的重力模式，选项为“真”或“假”分别表示重置或不重置。

11.8 获取物体重力模式

示例：

```
zw_body_get_gravity_mode(' ')
```

积木块：



参数说明：

Name	需获取物体的名称（字符串）
------	---------------

11.9 重置物体位置坐标

示例：

```
zw_body_set_position(' ', zw_vec3(1, 0, 0))
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis coordinate	x 轴坐标
Y axis coordinate	y 轴坐标
Z axis coordinate	z 轴坐标

11.10 获取物体位置坐标

示例：

```
zw_body_get_position(' ')
```

积木块：



参数说明：

Name	需获取物体的名称（字符串）
------	---------------

11.11 重置物体质量

示例：

zw_body_set_mass(' ', 50)

积木块：



参数说明：

Name	需重置的名称（字符串）
Quality	物体质量

11.12 获取物体质量

示例：

zw_body_get_mass(' ')

积木块：



参数说明：

Name	需获取物体的名称（字符串）
------	---------------

11.13 重置物体线速度阻尼大小

示例：

zw_body_set_linear_damping(' ', 0.5)

积木块:



参数说明:

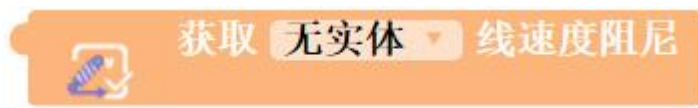
Name	需重置物体的名称 (字符串)
Linear velocity damping	线速度阻尼大小

11.14 获取物体线速度阻尼

示例:

```
zw_body_get_linear_damping(' ')
```

积木块:



参数说明:

Name	需获取物体的名称 (字符串)
------	----------------

11.15 重置物体角速度阻尼大小

示例:

```
zw_body_set_angular_damping(' ', 0.5)
```

积木块:



参数说明：

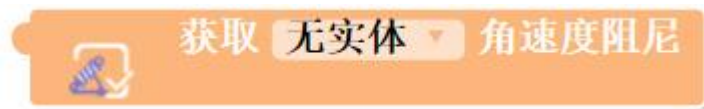
Name	需重置物体的名称（字符串）
Angular velocity damping	角速度阻尼大小

11.16 获取物体角速度阻尼

示例：

```
zw_body_get_angular_damping(' ')
```

积木块：



参数说明：

Name	需获取物体的名称（字符串）
------	---------------

11.17 重置物体摩擦系数大小

示例：

```
zw_body_set_friction_coefficient(' ', 0.5)
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
Coefficient of friction	摩擦系数大小

11.18 重置物体弹性系数大小

示例：

```
zw_body_set_elastic_coefficient(' ', 0.5)
```

积木块：



参数说明：

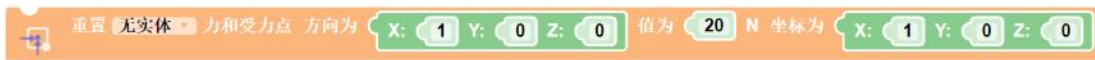
Name	需重置物体的名称（字符串）
Coefficient of elasticity	弹性系数大小

11.19 重置物体受力大小和方向以及受力点坐标

示例：

```
zw_body_set_force_at_pos(' ', zw_vec3(1, 0, 0), 20, zw_vec3(1, 0, 0))
```

积木块：



参数说明：

Name	需重置物体的名称（字符串）
X axis direction	力的 x 轴方向
Y axis direction	力的 y 轴方向
Z axis direction	力的 z 轴方向

X axis coordinate	受力点的 x 轴坐标
Y axis coordinate	受力点的 y 轴坐标
Z axis coordinate	受力点的 z 轴坐标

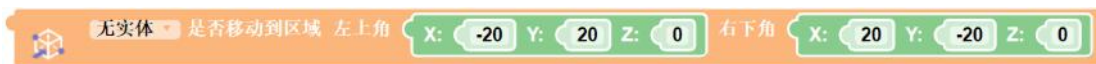
11.20 物体是否移动到指定区域

判定物体是否移动到了目标区域内,左上角与右下角的 xyz 坐标构成了一个立体的六面体空间区域。

示例：

`zw_is_move_to_area(' ', zw_vec3(-20, 20, 0), zw_vec3(20, -20, 0))`

积木块：



参数说明：

X axis coordinate	左上角的 x 轴坐标
Y axis coordinate	左上角的 y 轴坐标
Z axis coordinate	左上角的 z 轴坐标
X axis coordinate	右下角的 x 轴坐标
Y axis coordinate	右下角的 y 轴坐标
Z axis coordinate	右下角的 z 轴坐标

11.21 物体是否相交

判定两个物体是否相交。

示例：

```
zw_is_intersects(' ','')
```

积木块：



参数说明：

Name	物体的名称（字符串）
Name	物体的名称（字符串）

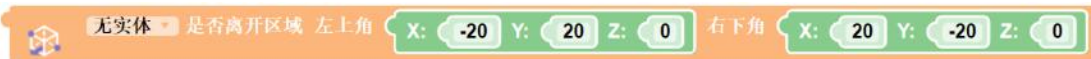
11.22 物体是否离开指定区域

判定一个物体是否离开了指定区域, 左上角与右下角的 xyz 坐标构成了一个立体的六面体空间区域。

示例：

```
zw_is_leave_area(' ',zw_vec3(-20, 20, 0),zw_vec3(20, -20, 0))
```

积木块：



参数说明：

Name	物体的名称（字符串）
X axis coordinate	左上角的 x 轴坐标
Y axis coordinate	左上角的 y 轴坐标
Z axis coordinate	左上角的 z 轴坐标

X axis coordinate	右下角的 x 轴坐标
Y axis coordinate	右下角的 y 轴坐标
Z axis coordinate	右下角的 z 轴坐标

11.23 物体是否离开半径为规定值的区域内

以初始位置作为圆心，区域半径半径画圆得出区域，判定一个物体是否离开了初始位置的指定半径范围。

示例：

```
zw_is_leave_init_pos(' ',5)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Radius	初始区域的半径

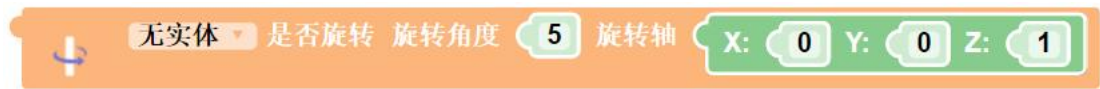
11.24 物体是否根据指定轴转动一定角度

判定物体是否围绕着某一轴旋转了指定角度。

示例：

```
zw_is_rotate_angle(' ',5,zw_vec3(0, 0, 1))
```

积木块：



参数说明：

Name	物体的名称（字符串）
Rotation angle	旋转角度
X axis coordinate	旋转轴的 x 轴坐标
Y axis coordinate	旋转轴的 y 轴坐标
Z axis coordinate	旋转轴的 z 轴坐标

11.25 获取物体的重置次数

获取一个物体被重置的次数

示例：

```
zw_reset_time(' ')
```

积木块：



参数说明：

Name	需获取物体的名称（字符串）
------	---------------

11.26 物体是否移动到了另一个物体指定误差区域内

判定一个物体是否进入到了另外一个物体的允许误差半径范围内。

示例：

zw_is_move_to_body_area(' ','',5)

积木块：



参数说明：

Name	需移动物体的名称（字符串）
Name	检测区域物体的名称（字符串）
Radius	误差半径

11.27 设置一个项目的得分以及其显示位置

设置一个得分项目，并设置它在左上角显示的顺序。

示例：

zw_set_item_score('Item1',0,1)

积木块：



参数说明：

Name	项目的名称（字符串）
Score	分数
Order	显示顺序

11.28 设置左上角是否显示重置次数

设置左上角是否显示重置次数。

示例：

```
zw_set_show_reset_time(True)
```

积木块：



参数说明：

State	重置次数的显示状态，选项为“真”或“假”分别表示显示或不显示。
-------	---------------------------------

11.29 设置仿真时间是否显示

设置左上角是否显示仿真时间

示例：

```
zw_set_show_sim_time(True)
```

积木块：

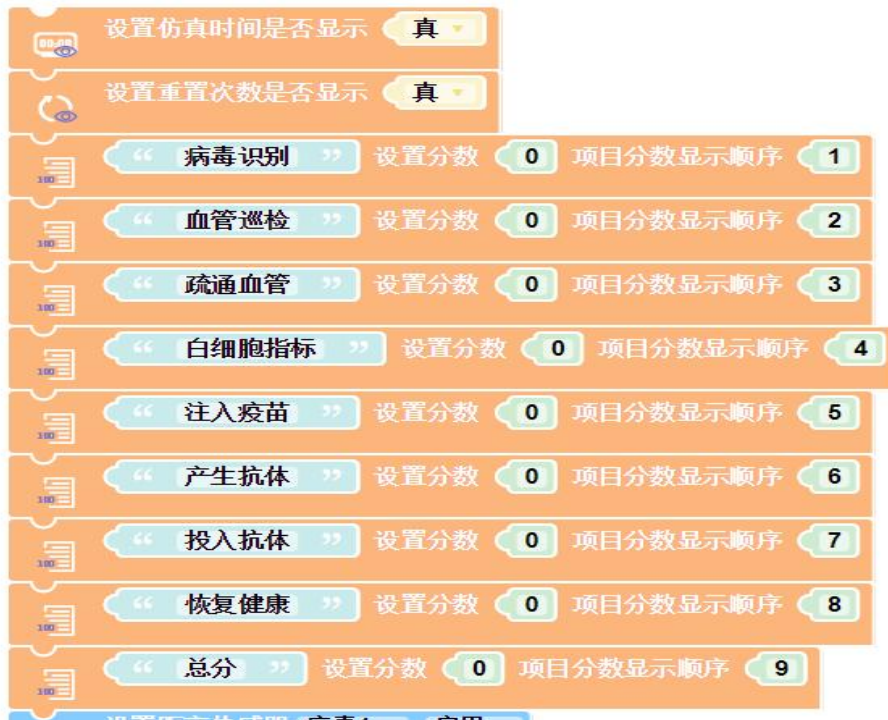


参数说明：

State	仿真时间的显示状态，选项为“真”或“假”分
-------	-----------------------

别表示显示或不显示。

11.30 积分系统全局解释



PS:在使用积分系统时，有部分的条件限制。

限制一：设置仿真时间和重置次数的显示，必须放在设置分数积木块的上方。

限制二：所有初始化积分系统的积木块都需要放置在同一个控制器中，其他控制器对积分进行修改不受影响。

限制三：积分系统的功能只有在将文件设置为比赛文件过后才会生效--[设置详见](#)

12 编程控制器——关节

12.1 设置合页关节和旋转角度

示例：

```
zw_controller_rotate_hingejoint(' ',20)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Angle	旋转角度

12.2 设置合页关节速度

示例：

```
zw_controller_velocity_hingejoint(' ',1)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Speed	速度

12.3 设置插销关节移动距离

示例：

```
zw_controller_move_sliderjoint(' ',20)
```

积木块：



参数说明：

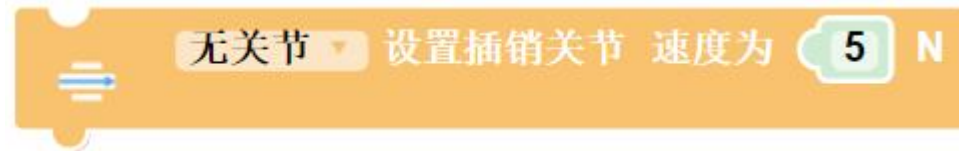
Name	物体的名称（字符串）
Distance	移动距离

12.4 设置插销关节速度

示例：

```
zw_controller_velocity_sliderjoint(' ',5)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Speed	速度

12.5 设置铰链关节力矩

示例：

```
zw_move_linear(' ',1)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Torque	力矩

12.6 设置铰链关节旋转角度

示例：

```
zw_move_turn(' ',15)
```

积木块：



参数说明：

Name	物体的名称（字符串）
Angle	旋转角度

13. 编程控制器——人脸识别

13.1 启动人脸检测

此函数用于调用人脸检测功能，通过摄像头去检测是否存在人脸，以及存在人脸数量。

```
zw_image_recognize_face_detect()
```

积木块：



13.2 人脸检测结果

此函数用于获取人脸检测反馈的结果，获取结果分为两种情况：是或否。

`zw_image_recognize_face_detect_result('True')`

积木块：



13.3 人脸检测数量

此函数用于获取人脸检测数量的反馈结果，获取结果为数值

`zw_image_recognize_face_detect_number_result()`

积木块：



13.4 启动人脸搜索

人脸搜索包含录入人脸信息库和人脸信息搜索两部分。

(1) 人脸信息库

在图像识别分类下增加人脸信息库功能按钮。



人脸信息库主要包含四个功能：单个信息录入、批量信息录入、导入信息库文件、导出信息库文件。人脸信息库界面左侧为固定区域，右侧上方操作栏为固定区域，右侧下方为滚动区域。

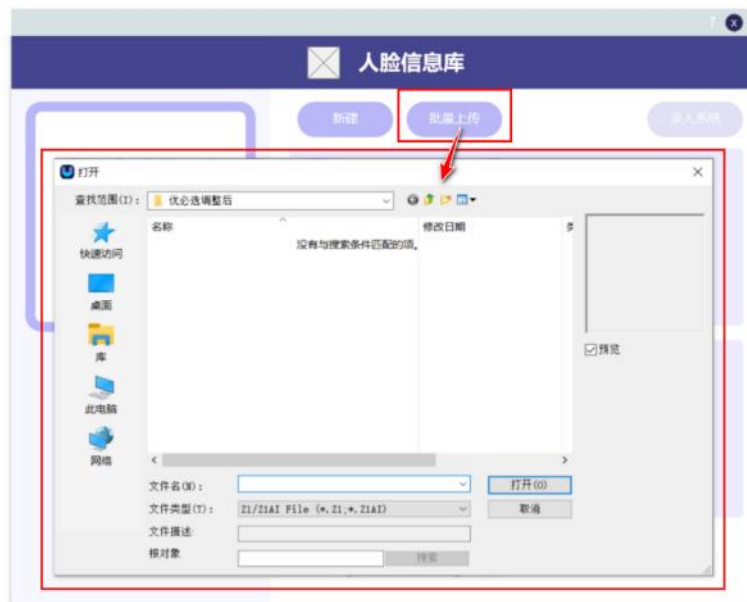


摄像头打开和关闭

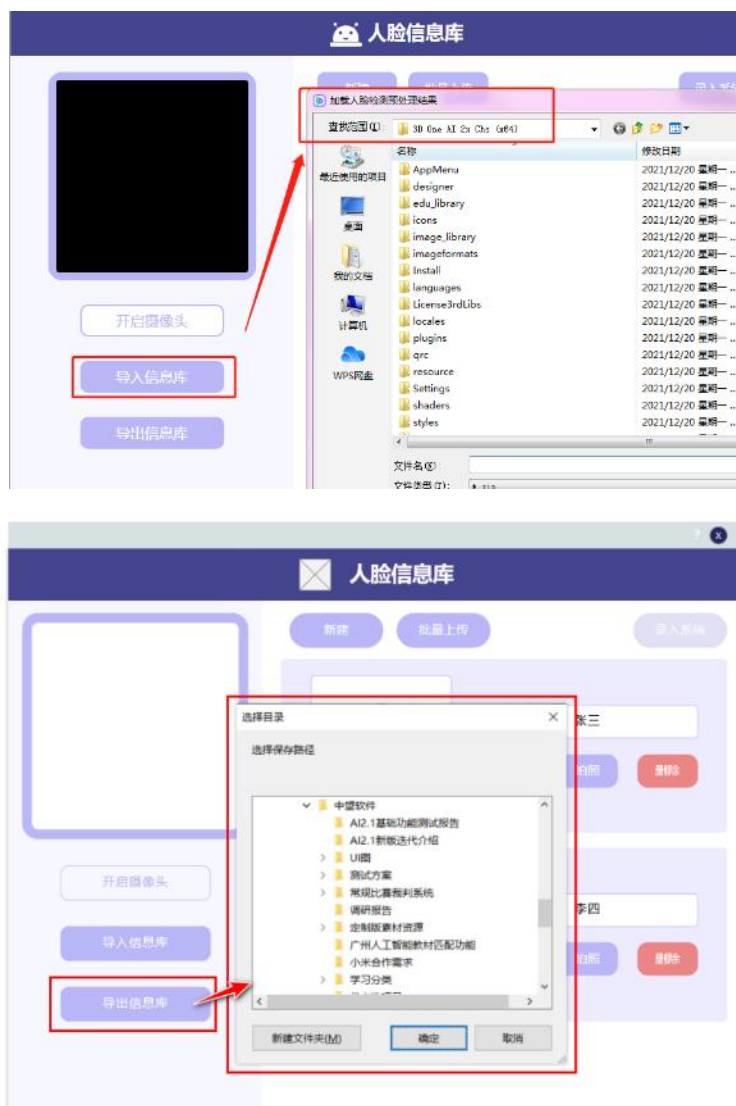
新建用户信息：点击新增用户按钮会在右侧区域增加一个空的用户信息，并按先后顺序从上到下排序。在编辑完用户信息后需要点击录入系统按钮才能将信息保存成功。



批量上传：点击打开系统文件框，选择需要批量上传的照片（在同一个文件中）；自动根据照片数批量创建用户信息，用户名称默认为照片的名称



导入/导出信息库：导入导出信息库功能主要用于整个人脸信息库文件的导入导出



(2) 人脸信息搜索

此函数用于启动在人脸信息库中进行搜索

zw_image_recognize_face_search()

积木块：



启动人脸搜索后，人脸照片将会与库中的照片进行比对，并反馈回相似度最高的用户名和相似度。

13.5 人脸信息搜索结果

此函数用于储存人脸信息搜索的结果，其中可选择全部（即名称和相似度）、名称、相似度。

`zw_image_recognize_face_search_result_all()`

积木块：

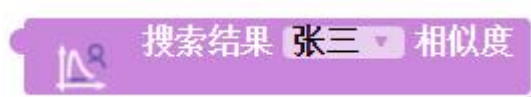


13.6 搜索结果相似度

此函数用于储存人脸信息与指定用户对比的相似度

`zw_image_recognize_face_search_similarity('张三')`

积木块：



13.7 判断搜索结果是否为指定用户

此函数用于储存搜索结果是否为指定用户

`zw_image_recognize_face_search_judgement('张三')`

积木块：

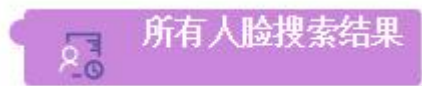


13.8 所有人脸搜索结果

此函数用于储存程序运行过程中所有人脸搜索结果

`zw_image_recognize_face_search_result_list()`

积木块：



13.9 清除所有人脸搜索结果

此函数用于清除程序运行过程中的所有人脸搜索结果

`zw_image_recognize_face_search_result_list_clear()`

积木块：



13.10 启动人脸情绪识别

此函数用于识别并显示情绪及置信度,其中置信度以百分比形式体现,保留到小数点后两位。

`zw_image_recognize_face_emotion()`

积木块：

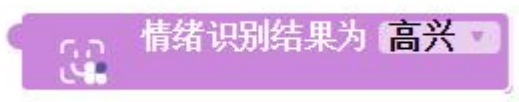


13.11 情绪识别结果是否为指定情绪

此函数用于储存搜索结果是否为指定情绪。

`zw_image_recognize_face_emotion_result(int)`

积木块：



14. 编程控制器——翻译

14.1 将指定内容翻译为各种语言

此函数用于指定内容（包括语音识别结果、图片文字识别结果、图片对象检测结果等）翻译为各种语言。

zw_translate ('',int)

积木块：

Name	翻译内容（字符串）
Int	语言类型

14.2 翻译结果

此函数用于储存翻译结果。

zw_translate_result ()

积木块：



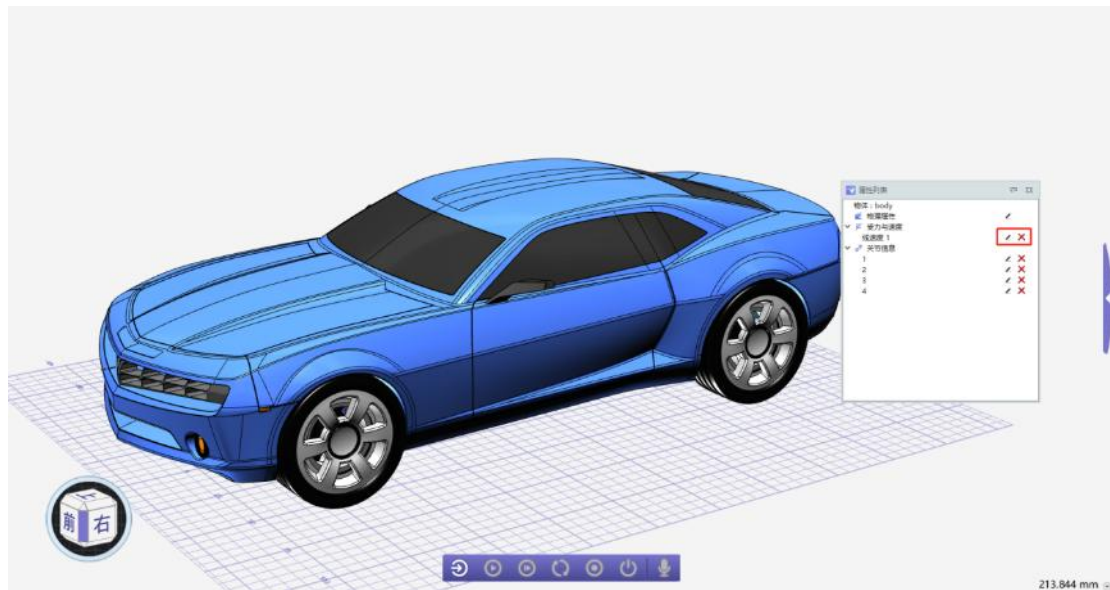
六、其他

1. minibar

单击视图区中实体，弹出的 minibar 中包含的功能有：移动、对其到网格、缩放、阵列、镜像、抽壳、属性列表、单个模型导出。其中相对于 3D One 新增的功能为【属性列表】。



对于配置了信息的实体，单击 minibar 后弹出的【属性列表】按钮，可查看该实体配置的信息，信息项包含物体属性、关节、受力与速度和电子件，并且可以通过列表进行删除、重编辑参数等操作。



2. 仿真过程控制



从左往右依次为:

进入仿真环境: 从建模环境进入仿真环境;

启动仿真: 开始进行仿真;

单步仿真: 以默认步长, 进行单步仿真, 每点一次运行一个步长;

重置仿真: 将仿真状态退回初始状态, 点击启动仿真即可再次运行仿真;

录制视频: 可以在仿真过程中录制视频、音频等;

退出仿真: 退出仿真环境, 回到建模环境;

仿真设置: 可以调整仿真的显示环境、天空样式、物体的材料以及粒子效果

设置第一人称视角: 点击可以开启第一人称视角, 并切换第一人称视角

3. 仿真设置

在仿真过程控制任务栏中进入仿真设置, 可以调整仿真的显示环境、天空样式、物体的材料以及粒子效果。

示例图:



3.1 显示环境设置

显示环境设置包括显示质量、光照强度、光照高度、光照角度、投影模式。(流畅版仅提供投影模式设置)

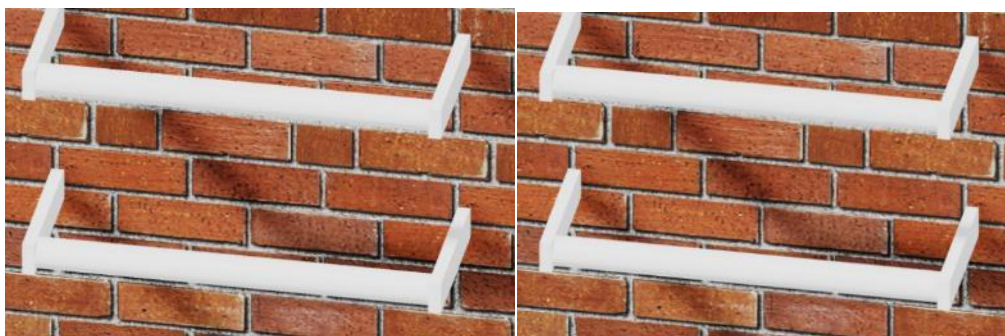
示例图:



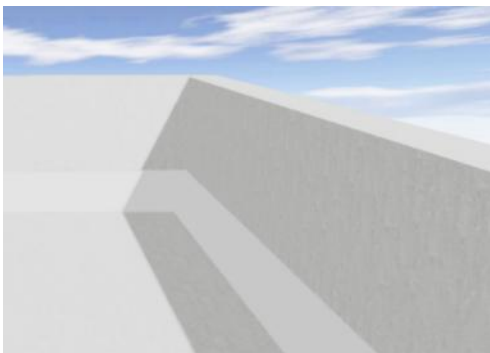
其中显示质量分为流畅、超清、自定义三个选项，流畅版使用高性价比的显示渲染技术，具备更流畅的运行效果，超清版使用现代的显示渲染技术，具备更高质量的渲染效果。以下为两者显示特性的对比：

显示特性	流畅版	超清版
光照模型	√	√
抗锯齿	√	√
天空盒	√	√
粒子效果	√	√
投影模式	√	√
透明渲染	√	√
材质设置		√
调整光照强度与角度		√

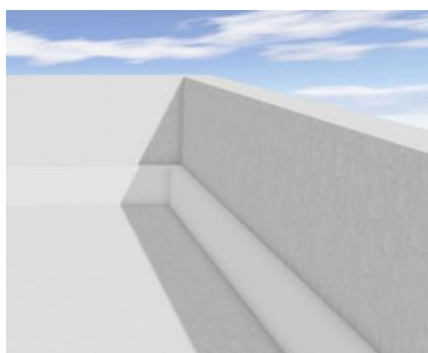
自定义显示环境下相较于超清版，增加三种显示参数的修改：抗锯齿、环境遮罩、阴影质量。



(抗锯齿关)



(抗锯齿开)



(环境遮罩关)



(环境遮罩开)



(阴影质量关)



(低阴影质量)

(高阴影质量)



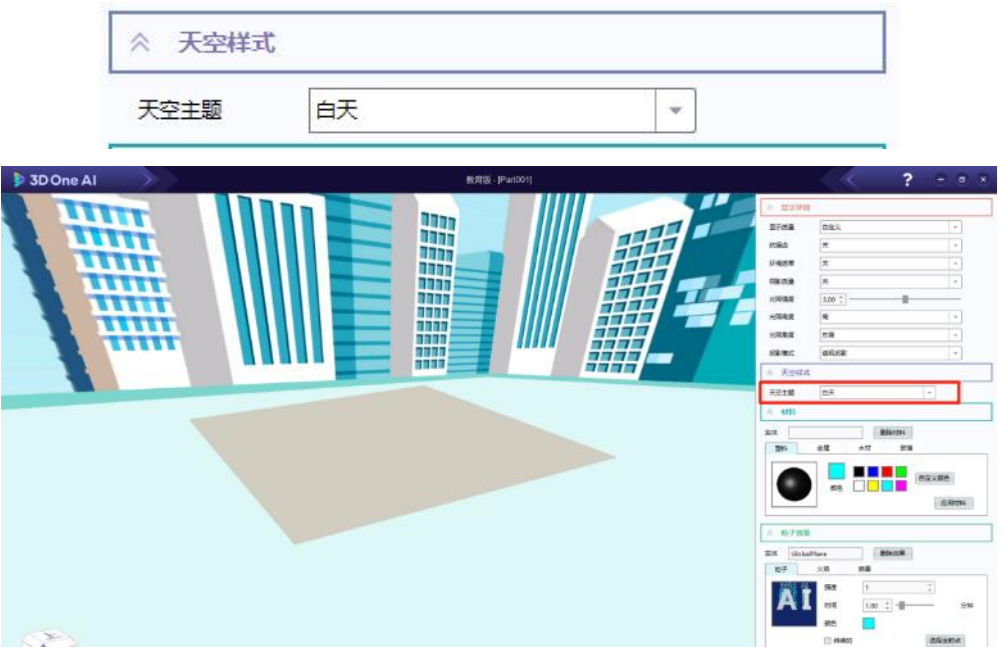
名称	介绍	参数范围	默认参数	对应传感器
全局温度	可控制整个环境的温度	-5~50℃	21	温度传感器
光强标准	通过下拉选框切换显示的光强标准,	自由模式	自由模式	

	可改变光照强度的区间和单位	Microbit		
光照强度	可控制整个环境的光强	0~255	128	光敏传感器

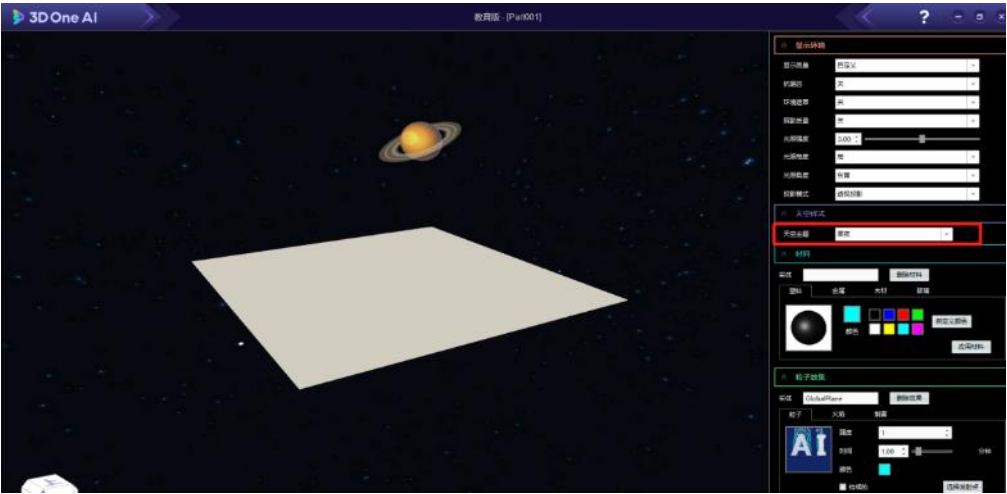
3.2 天空样式设置

天空样式分为白天、黑夜、颜色三种主题。

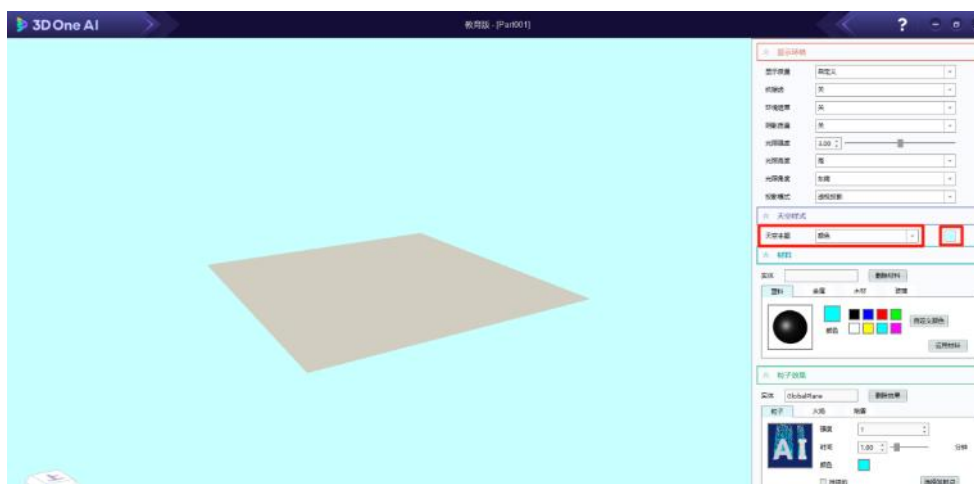
示例：



(白天主题)



(黑夜主题)



(颜色主题)

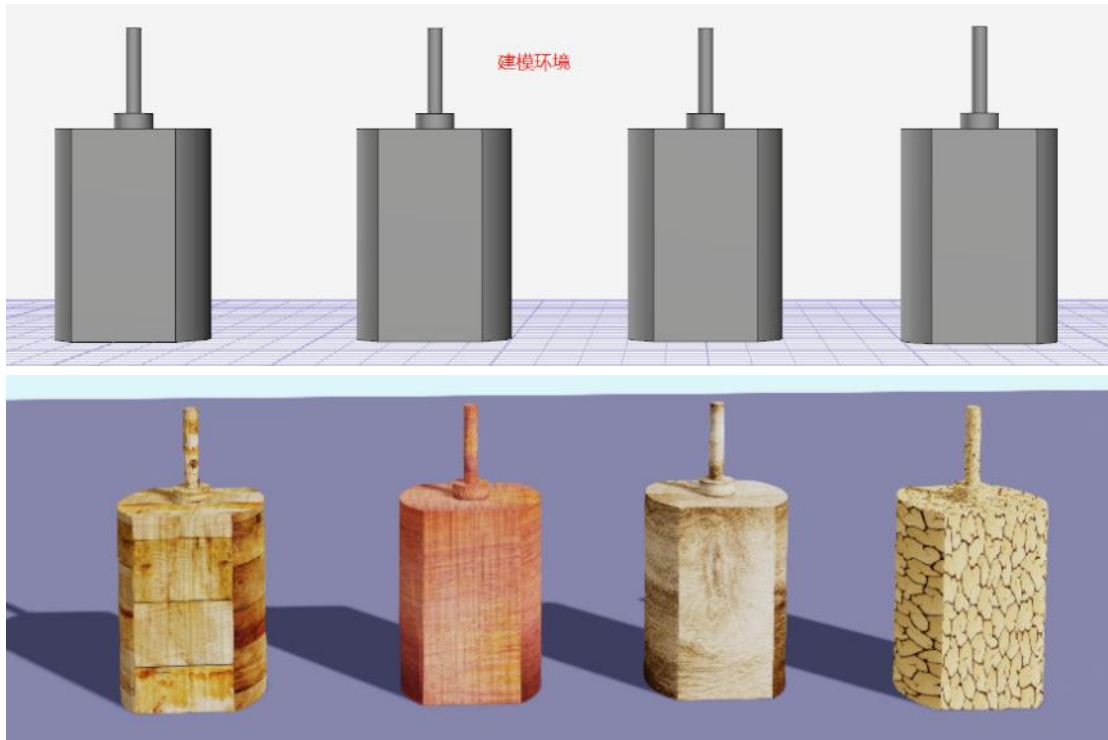
3.3 材质设置

材质设置可以通过点选仿真中的物体进行设置，包括塑料、玻璃、金属、木材四种材质。

其中塑料与玻璃支持自定义颜色的设置，金属与木材仅支持 4 种颜色的设置。

以木材为例进行设置：





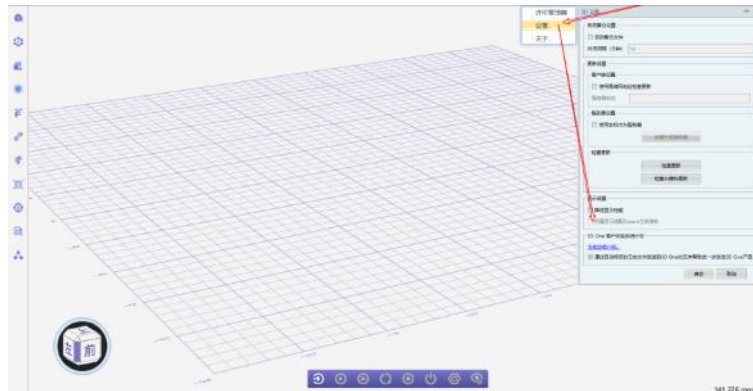
3.4 粒子效果设置

粒子效果设置可以为物体添加动态的效果，包括粒子、火焰、烟雾。

示例



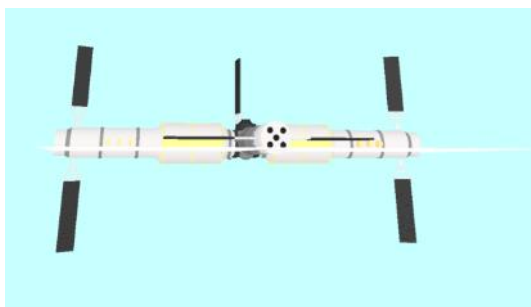
3.5 仿真环境支持 zSpace 眼镜追踪



启动 3D One AI 软件时，系统会自动检测设备是否支持 zSpace 立体渲染，如果该设备不支持，则 zSpace 立体渲染选项是灰色不可勾选；如果支持，则该选项可勾选且默认为选中状态。

在 zSpace 设备上,如果勾选 zSpace 立体渲染选项,则进入仿真环境后会自动检测 zSpace 眼镜并追踪。

初始时，主界面会呈现默认参数的立体视图并开始检测 zSpace 眼镜，当检测到眼镜时，软件仿真界面的主视图会持续呈现立体的效果并跟踪眼镜；检测不到眼镜时，则会自动恢复普通视图状态。下图为两种情况下，裸眼观察效果图（因为截屏软件技术限制，立体视图效果无法截屏，（图二）为相机拍摄）；



图一（未检测到眼镜时）普通视图效果



图二（检测到眼镜时）立体视图效果

4. 百度在线识别功能

此版本 64 位及 32 位的植物/动物识别、手势识别、朗读仅提供百度在线识别接口, 64 位的图像识别、文字识别、语音识别提供了百度在线识别接口以及本地离线识别, 32 位的图像识别、文字识别、语音识别只提供百度在线识别接口。64 及 32 位中其它的人工智能识别与机器学习均为离线使用。当使用在线识别接口时, 进入仿真前会提示需要登录。



当在线识别次数用完后会在仿真环境中提示, 如下图所示, 选择使用本地识别后会转为本地离线识别。



5. Zspace 立体渲染

该功能开启后，点击进入仿真，戴上 3D 眼镜即可以看到仿真结果的 3D 效果。

点击软件右上角的 ，再点击设置，在显示设置处可勾选 Zspace 立体渲染，该功能默认是不开启的。

示例：

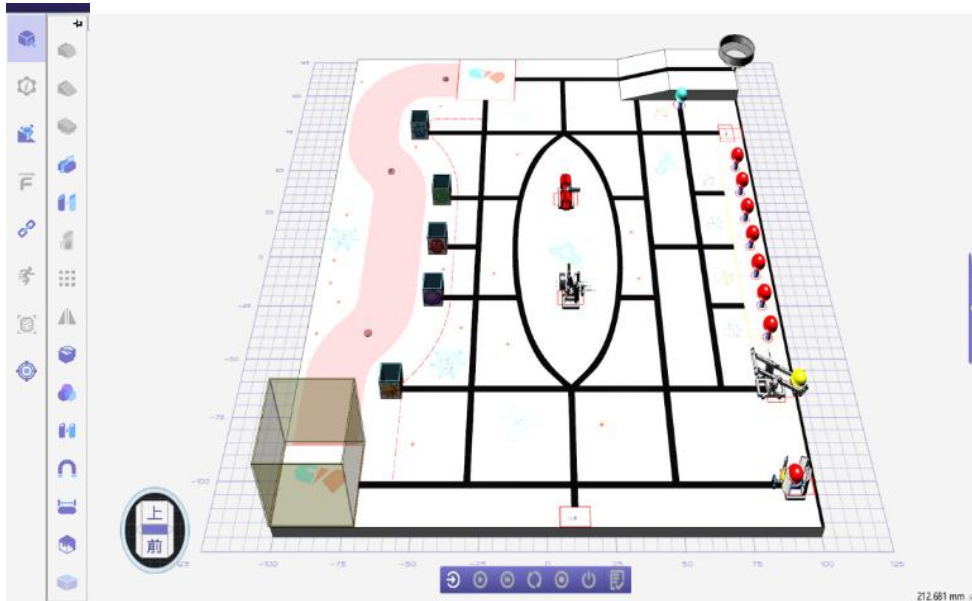


目前，Zspace 立体渲染仅支持 NVIDIA 的 Quadro K 或 Quadro M 系列，AMD FirePro V 或 W 系列，AMD Radeon R7 * 及以上，更多详情请查看 https://support.zspace.com/s/article/zSpace-Supported-Graphics-Cards-GPUs?language=en_US。如果电脑不配置以上显卡，Zspace 立体渲染效果将无法开启。

6. 比赛文件定制化处理

6.1 比赛文件定制化处理

- 1) 锁定提供的比赛场景，让学生无法进行再编辑操作，不能修改比赛方提供的场景的位置，物体的属性，以及全局设置，无法选中比赛场景中的所有实体。



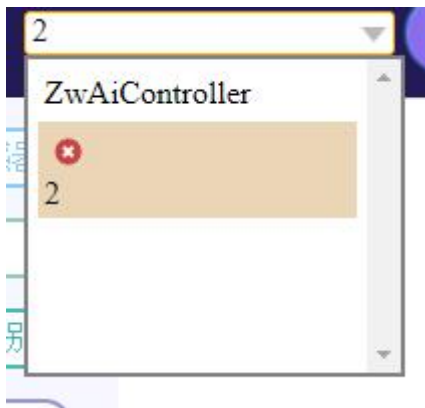
2) 右侧功能栏缩减

只保留贴图、鲸鱼专区以及控制器编程模块

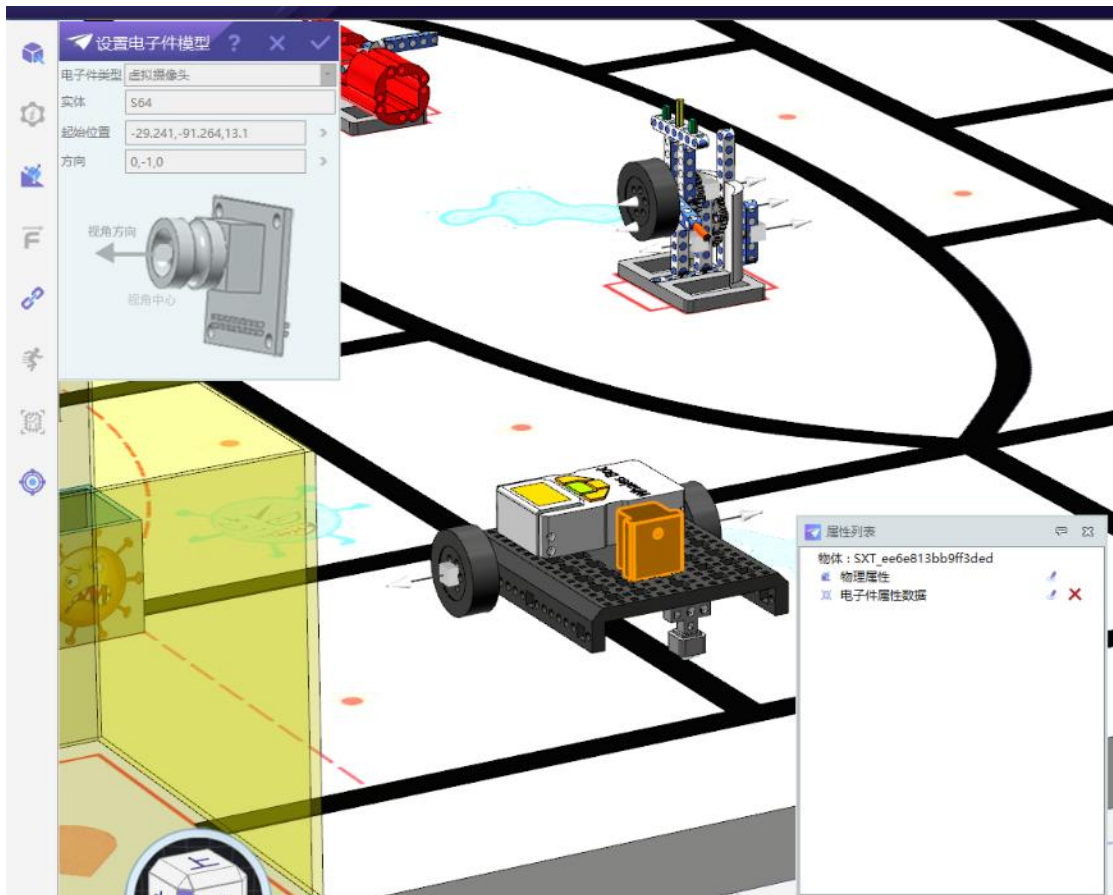


3) 隐藏控制器功能

控制器中只保留了学生自己写好的程序, 所有关于赛方设定好的积分规则控制器全部被隐藏, 学生无法修改赛方规则操作。



4) 从鲸鱼专区下载模型或者电子件, 可操作性被限制



5) 可以根据自己的得分情况提交成绩至社区之中

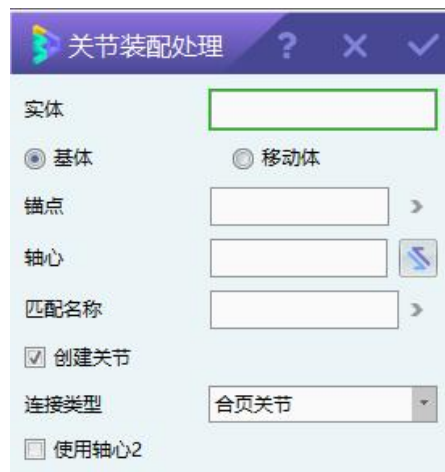


7. 库模型配置的关节与装配处理

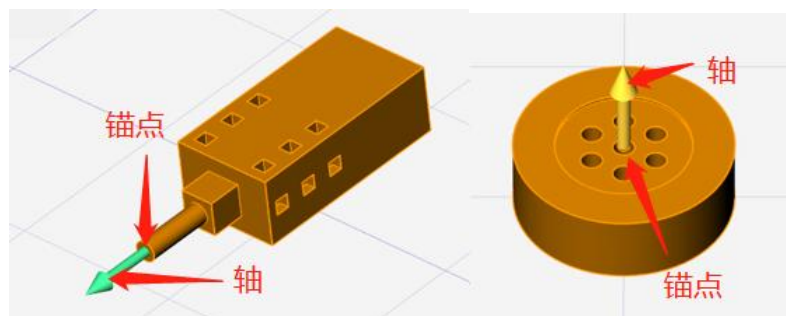
7.1 关节装配处理

关节装配处理命令需要通过键盘快捷键：**shift+Alt+w** 调出命令。对物体进行此功能的设置之后，在库中插入模型即可动态显示可吸附位置，并自动生成关节。是快速搭建模型的方法，且不需要配置任何信息，进行编程与仿真即可。

示例：



- 1) 选择实体
- 2) 选择该实体作为基体/移动体：基体即插入时作为不动的实体，移动体是插入时移动的实体，比如马达作为基体，那么轮子就应该设为移动体；
- 3) 轴与点：即物体的对齐轴与吸附点（与智能关节类似）；



(基体)

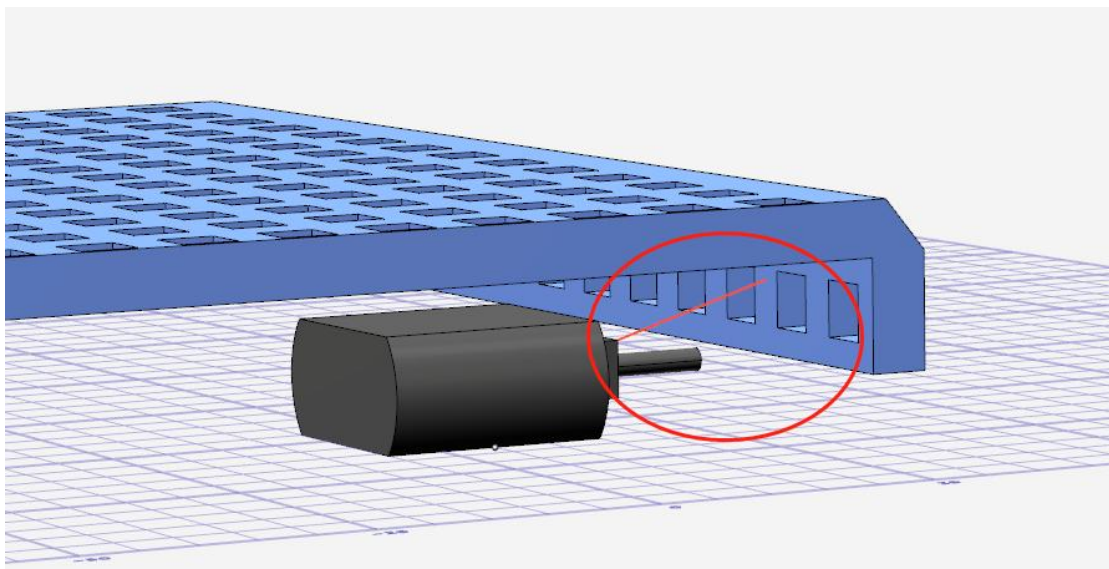
(移动体)

- 4) 匹配名称：即基体与移动体的匹配名，比如马达在与轮子的装配中是基体，但在与车底盘的装配下是移动体，那么马达与轮子的装配应该是一个匹配名称，马达与车底盘是另一个匹配名称，从而实现一个物体可以作为基体也可以作为移动体进行插入。
- 5) 生成关节：根据选择的对齐轴与点获取生成关节的数据，在插入时才会自动生成关节（类似智能关节）
- 6) 固定轴：两个物体插入时可能存在装配约束不够导致插入的对齐效果未达到预设效果，此时再增加一个固定轴即可完全约束物体的对齐状态。

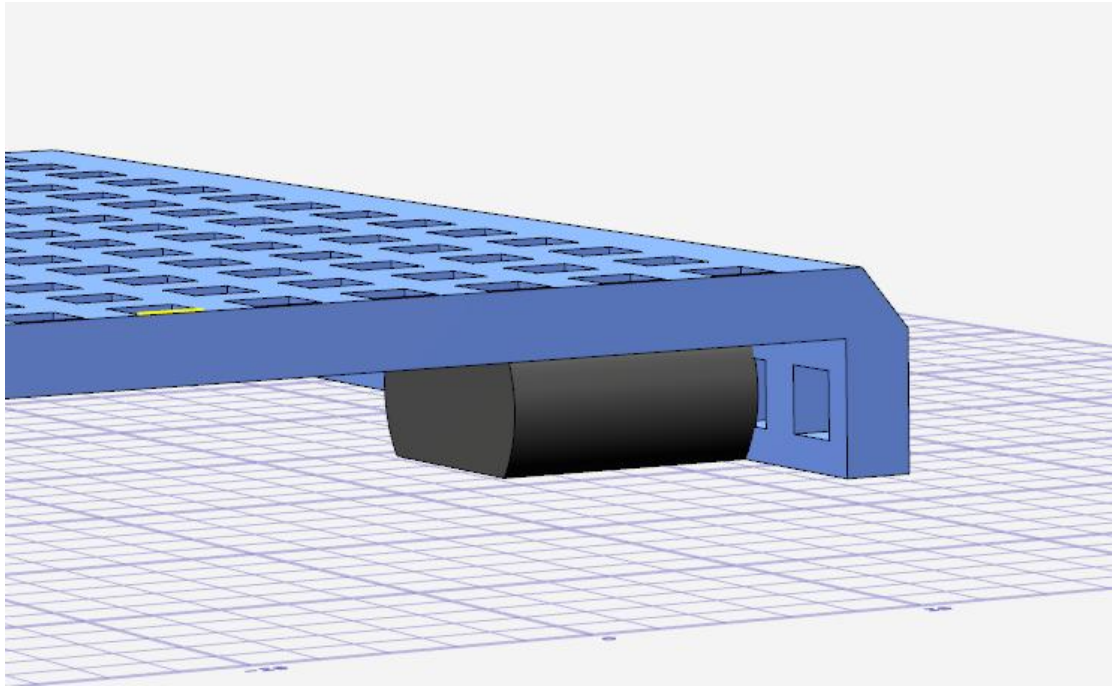
模型库与场景专区的模型已经进行过设置，可以直接用里面的模型进行装配。

示例 1：

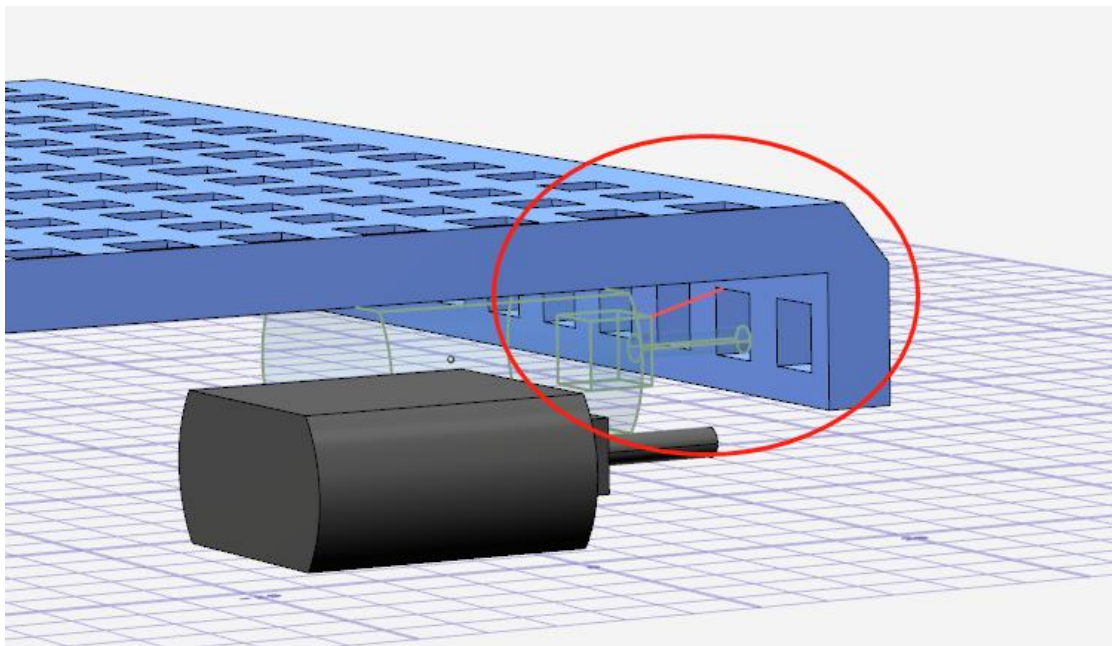
在模型库中选择车子的模型，先插入大底盘，然后插入马达，在大底盘附近晃动，可以看到出现了一条红色的装配线：



此时单击鼠标左键，马达自动装配到底盘上，无需再进行其他操作：

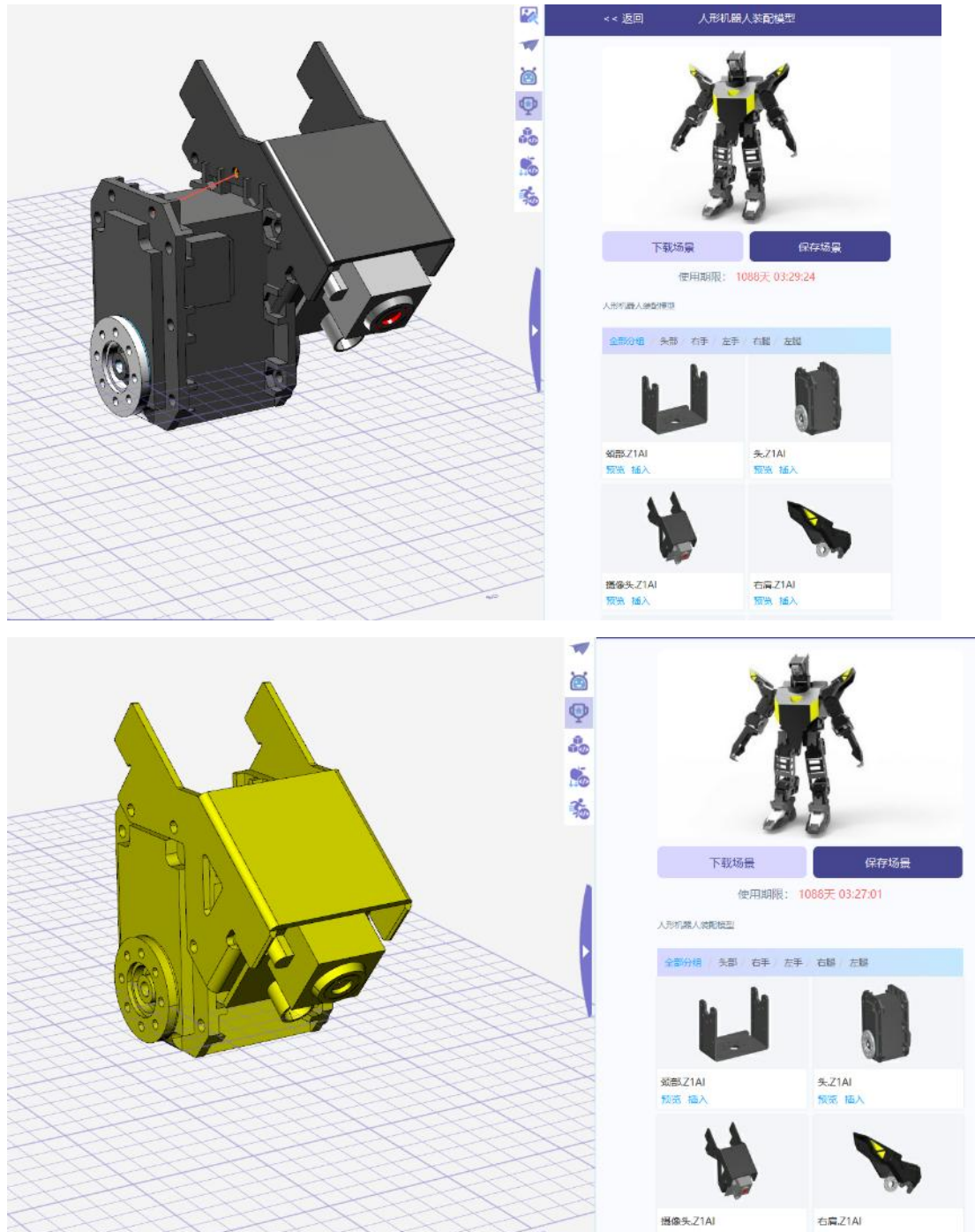


当将该马达拖离底盘，重新进行装配时，红色的装配线还是出现，马达和底盘还是能继续进行自动装配



示例 2:

场景专区的模型同样能够进行装配处理。进入场景专区的人形机器人装配模型，将头部拖入，再插入摄像头，两个部件可进行自动装配：

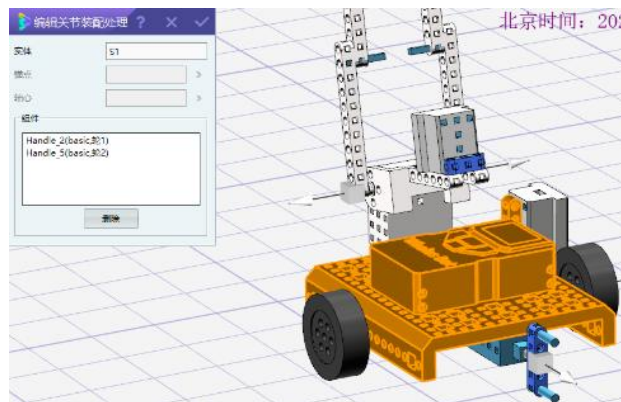


7.2 编辑关节装配处理

关节装配处理需要通过键盘快捷键：shift+Alt+e 调出命令。选中实体后，鼠标点击信息即可通过箭头显示已配置的关节装配信息，需要重新配置该信息时可以通过此命令删除配置，

再重新配置实体。

示例：



8. 积分系统

为了让不了解积分编写规则的用户也能快速上手，设计一套简单的比赛规则，设置了积分系统。



该界面包含两个标签：成绩界面和规则设置。成绩界面用于查看每次比赛的成绩，规则界面用于设置比赛的规则。

8.1 规则设置



1) 规则页标签

规则页含有基础规则和积分规则，每个规则页与编程界面的控制页一一对应。

可新建和删除规则页，不同规则页之间的积分项目没有联系。

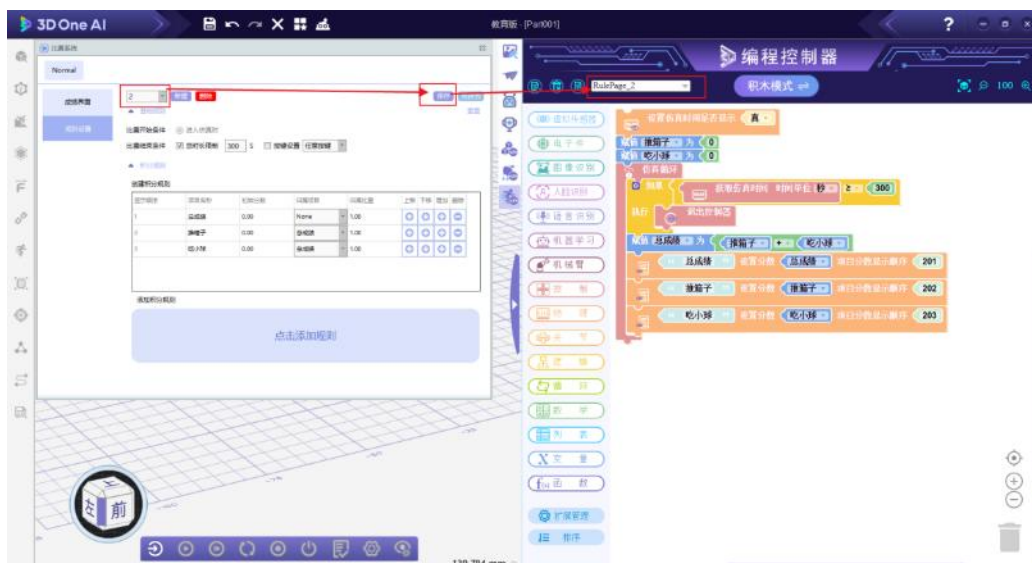
2) 保存规则页

保存：规则页有改动或者新建的规则页需要点击保存才能保存修改后的内容。

另存为：将当前规则页修改后的内容保存到新的规则页中，原规则页不变。

点击保存和另存为都会弹出积木块编程界面，每一页规则页都会对应生成一页积木块控制页。

用户对规则页修改的内容保存后都会同步更新到积木块控制页。



3) 基础规则



比赛开始条件：默认为进入仿真时。

比赛结束条件：默认为勾选总时长限制【300】s，即比赛时长到达 300s 比赛结束。

此外还提供通过按键结束比赛：①选择具体按键，可以手动结束比赛②选择“任意按键”，可以作为防作弊功能，检测到用户用键盘控制即自动结束比赛。



4) 创建积分项目

用户根据需求创建多个积分项目，并为积分项目设置参数。

显示顺序	项目名称	初始分数	归属项目	归属比重	上移	下移	增加	删除
1	总成绩	0.00	None	1.00	⬆	⬇	⬆	⬇
2	推箱子	0.00	总成绩	1.00	⬆	⬇	⬆	⬇
3	吃小球	0.00	总成绩	1.00	⬆	⬇	⬆	⬇

显示顺序：自动从上往下排序。表示仿真环境下左上角成绩的显示顺序以及成绩界面中成绩的显示顺序。



项目	分数
总成绩	0.00
推箱子	0.00
吃小球	0.00

仿真环境下左上角成绩的显示顺序

成绩界面中成绩的显示顺序

项目名称：可手动修改，积分项目的名称。

初始积分：可手动修改，默认为 0。积分项目初始的积分数值，在该数值上增减积分。

归属项目：可选项，默认为无。如果 A 项目选择归属项目为 B 时，则 B 项目在计算积分时会加上 A 项目的积分，即 A 项目作为 B 项目的子项目进行计算。**存在子项目的项目不可直接获得积分。**

归属比重：选择归属项目后可修改归属比重，默认为 1。如果 A 项目选择归属项目为 B 时，**B 总积分=B 初始积分+ A 总积分*归属比重**

显示顺序	项目名称	初始分数	归属项目	归属比重	上移	下移	增加	删除
1	总成绩	0.00	None	None	⬆	⬇	⬆	⬇
2	推箱子	0.00	None	None	⬆	⬇	⬆	⬇
3	吃小球	0.00	总成绩	1.00	⬆	⬇	⬆	⬇

(一个项目只能有一个归属项目)

上移/下移：移动项目的顺序，从而改变显示顺序

增加：在当前项目下方插入一个新增项目

删除：删除当前项目

5) 添加积分规则

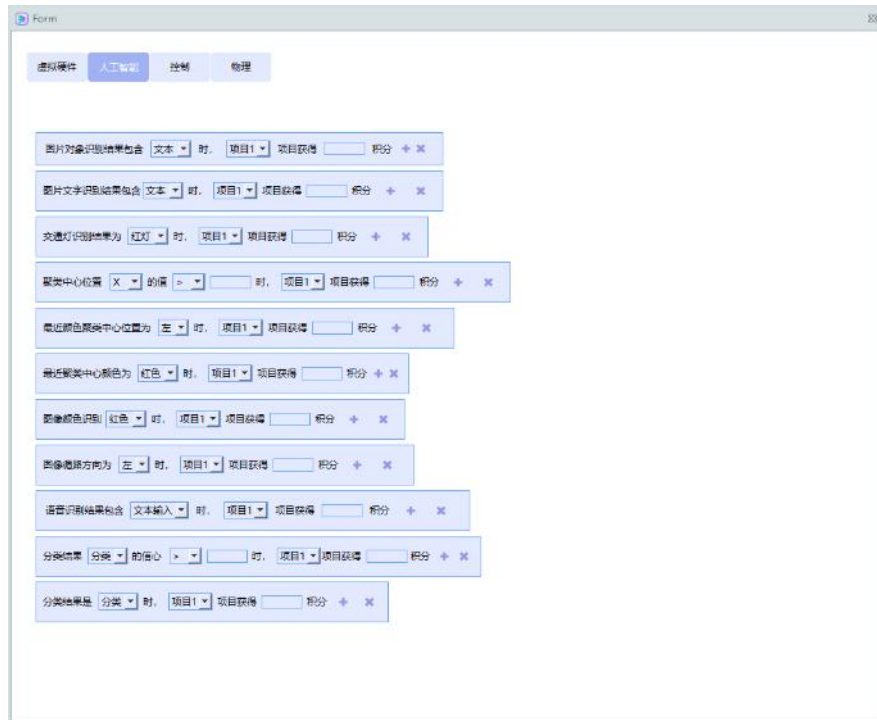
规则添加流程如下：



规则面板：

规则面板提供了丰富的预设规则，里面包含软件所有可获得的传感器、功能、操作的反馈结果，并对反馈结果进行条件判断，如果满足条件，即可获得对应项目的积分。

(虚拟硬件分类需要用户使用了电子件或传感器后，才会出现对应的规则)



规则池：

用于存放具体的规则，执行规则时，按顺序从上往下执行。每条规则只能触发一次，之后就不再生效。（可自行改为触发多次，详见规则扩展章节）



用户在添加规则后可对规则进行编辑。点击 + 号，弹出规则面板，在当前规则下方插入新规则，点击 - 号删除当前规则。

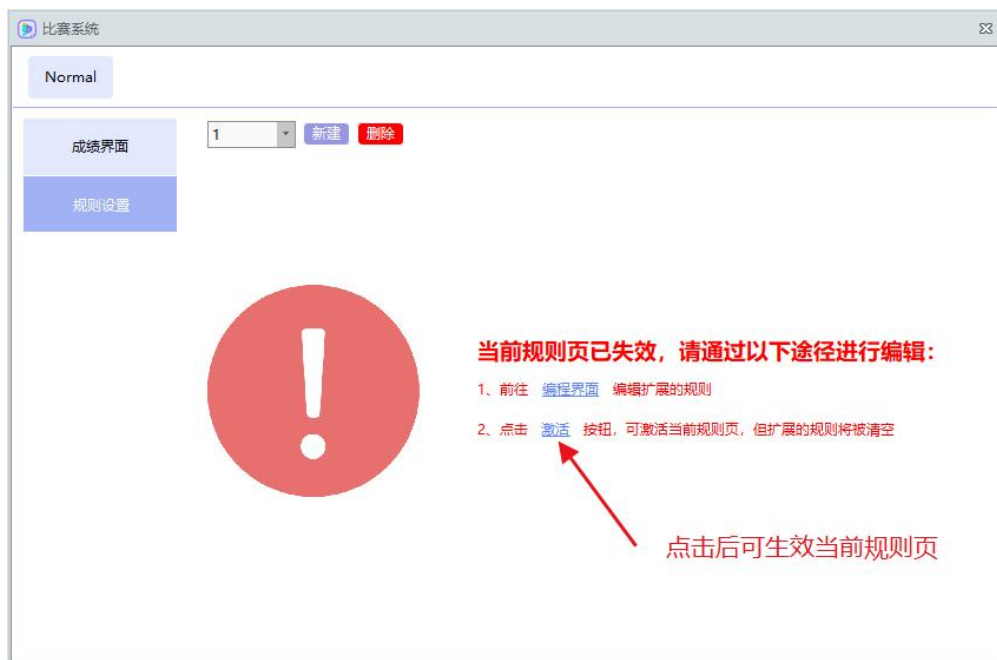
场景变化时，规则的下拉选框（作用对象）无法实时更新，可关闭界面再打开来刷新数据。

5) 规则扩展

用户通过保存和另存为生成的规则页，系统会同步在积木块编程界面生成一页积木块控

制页，用户可修改积木块控制页的内容去扩展规则。

积木块控制页一旦被修改会导致对应的规则页失效,即无法继续通过规则页去编辑当前修改后的规则。



点击激活按钮可激活当前规则页，对应的积木块控制页也会被重置为修改前的内容，即自行扩展的内容会被重置。

➤ 规则扩展案例

① 使当前规则可执行多次

每条规则会附带有个 Rule 值, 该值可控制规则的执行次数, 如图所示, 默认 Rule1 为真, 满足规则的触发条件后获得积分, 同时 Rule1 变为假, 则该条规则无法被再次触发。



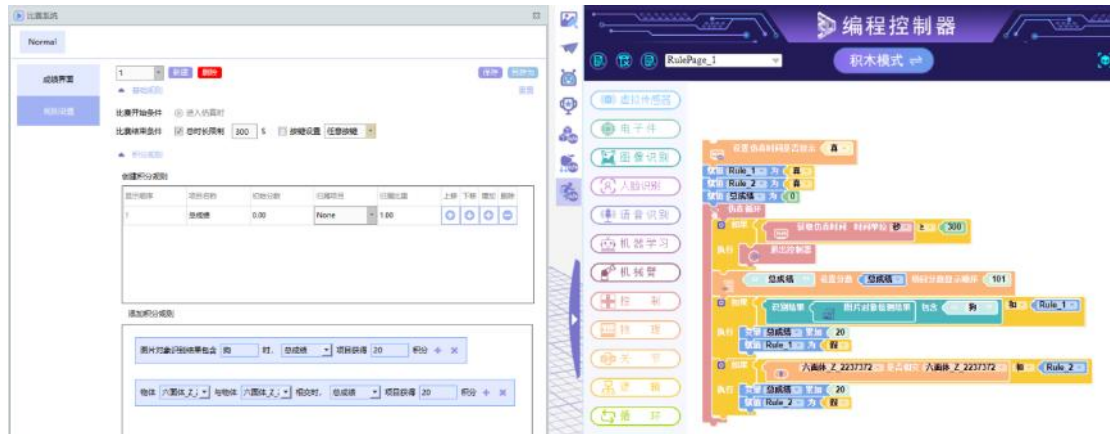
修改方式:

把 Rule1 赋值为 0, 触发条件改为 Rule 值<3, 每次触发后 Rule 值+1, 则该规则可触发 3 次。(需加个等待时间积木块否则会不间断的触发三次)

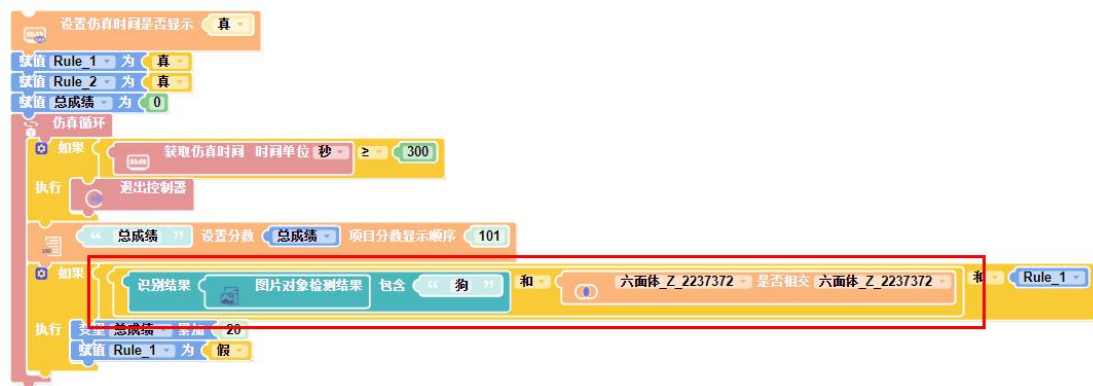


② 同时触发多条规则才能获得积分 (规则触发条件扩展)

如果用户想要设计同时满足两条积分规则才能获得积分的程序时, 可以先添加两条规则, 保存后生成积木块控制页。



将两条规则按照如下图的形式进行合并。

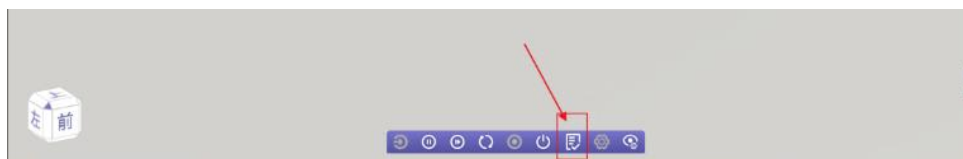


更多的扩展形式等待用户自行探索

8.2 成绩界面

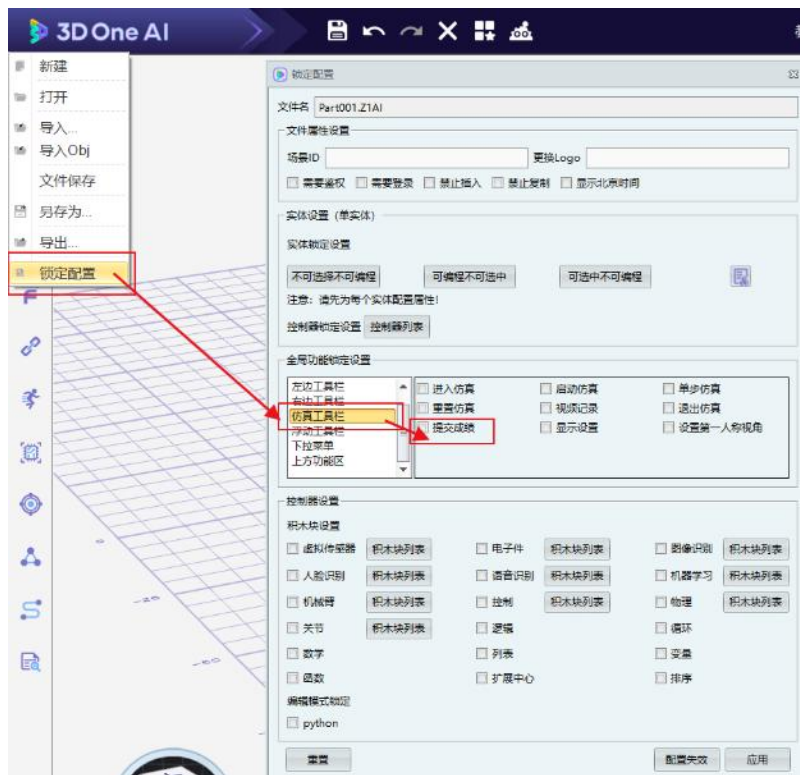
成绩界面主要显示两部分内容:比赛时间和积分项目得分情况。在仿真界面提交成绩后,可退出仿真在该界面查看成绩。

针对官方比赛的积分规则，可以点击查看详情跳转到对应的官网成绩地址，查看更多信息





提交成绩按钮默认为隐藏，需要手动到锁定配置中启用：点击左上角，在下拉选项中点击锁定配置，进入锁定配置界面，选择仿真工具栏，取消勾选提交成绩，点击应用。



8.3 仿真信息

仿真信息模块，页面主要包含：硬件参数、物理参数、系统信息、预览等，设置完以后可

以在仿真页面实时显示设置参数的数据。



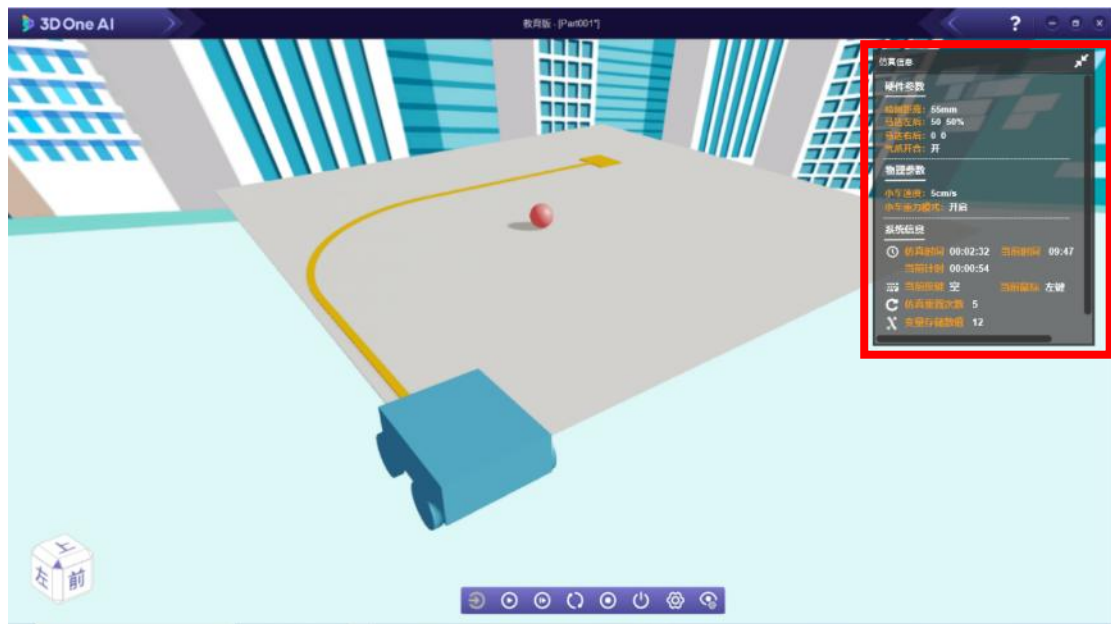
比赛系统入口



仿真信息入口



仿真信息设置页面



仿真页面

序号	名称	简介
(1)	硬件参数	电子件、传感器、机械臂等硬件在仿真中实时反馈的参数。
(2)	物理参数	坐标等物理信息、质量等物理属性信息，在仿真中反馈的参数。
(3)	系统信息	显示整体仿真情况的信息以及交互的反馈。
(4)	预览	查看设置后的预览效果。

(1) 硬件参数

硬件参数 全选

顺序	实体	类型	参数信息	名称	排序	选择

● 硬件参数类型

分类	硬件类型	参数信息	信息详情
虚拟传感器	距离传感器	测量距离	mm
	触摸传感器	触摸状态	（触碰、未触碰）
	力传感器	压力大小	N
	光照传感器	光照强度	0-5
	位置传感器	检测坐标	x, y, z
	颜色传感器	检测色值	R, G, B
	灰度传感器	检测位置及灰度值	左, 右（0-255）
	循迹传感器	检测轨迹	左, 右
	陀螺仪	速度	速度（x, y, z）
	陀螺仪	角度	角度（x, y, z）
电子件	舵机	绝对角度	°
	马达	转速及百分比	速度（百分比%）
	开关	状态	（开或关）
	RGB 灯	状态及颜色	（开启状态，当前 RGB 值）
	步进电机	转动速度及剩余步数	速度，步数
	伺服电机	转动速度及剩余转动角度	速度，角度
	扬声器	音量	%
	真空吸盘	状态	开启；关闭
机械臂	机械臂	机械臂高度、大臂角度、小臂角度	数值
	气爪	气爪状态	（开或关）

(2)物理参数

物理参数 田 添加

顺序	名称	实体	类型	参数信息	排序	删除
1	系统参数1	None	选择	None	↑	↓

● 物理参数类型

分类	硬件类型	参数信息	信息详情	是否在下拉列表显示
虚拟传感器	距离传感器	测量距离	mm	仅显示文件中包含的参数信息
	触摸传感器	触摸状态	(触碰、未触碰)	
	力传感器	压力大小	N	
	光照传感器	光照强度	0-5	
	位置传感器	检测坐标	x, y, z	
	颜色传感器	检测色值	R, G, B	
	灰度传感器	检测位置及灰度值	左, 右 (0-255)	
	循迹传感器	检测轨迹	左, 右	
	陀螺仪	速度	速度 (x, y, z)	
	陀螺仪	角度	角度 (x, y, z)	
电子件	舵机	绝对角度	°	
	马达	转速及百分比	速度 (百分比%)	
	开关	状态	(开或关)	
	RGB 灯	状态及颜色	(开启状态, 当前 RGB 值)	
	步进电机	转动速度及剩余步数	速度, 步数	
	伺服电机	转动速度及剩余转动角度	速度, 角度	
	扬声器	音量	%	
	真空吸盘	状态	开启; 关闭	
机械臂	机械臂	机械臂高度、大臂角度、小臂角度	数值	
	气爪	气爪状态	(开或关)	

物理参数的实体通过在显示区中拾取的形式选中：



(3) 系统参数

系统信息

☐ 显示按键信息	☐ 显示鼠标信息	☐ 显示仿真时间
☐ 显示变量储存数值	选择变量	

● 系统信息类型

参数信息具体分类情况如下：

分类	系统信息
控制	当前按键信息
控制	当前鼠标信息
控制	仿真时间
变量	变量存储数值

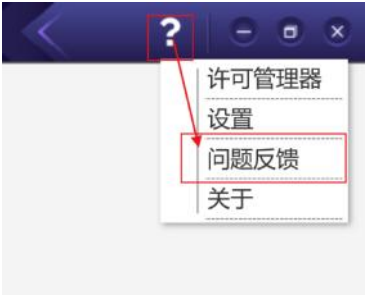
● 添加系统信息

由于系统信息较少，通过直接勾选即添加完成。（由于案例中可能存在多个变量，因此在下拉框进行选择，支持多选）。



9. 问题信息反馈模块

问题快速反馈功能主要用于用户主动反馈非崩溃问题,强调用户通过录制视频去重现问题, 通过上传 z1ai 文件, 方便我们对问题进行复现。



因此界面主要分为三部分, 重现问题模块, 上传附件模块, 同意授权发送模块。



重现问题模块：

录制视频：根据不同的录制情况，录制按钮会有功能和名称上的变化。

问题描述、联系方式：用户录制视频后可对问题进行简单的描述以及留下联系方式等信息。

上传附件模块：

允许用户自行上传截图、数据包等其他附件，方便研发复现问题。点击上传其他附件按钮，打开系统文件框，选择上传的文件，即可上传文件，可上传多个文件。

附件列表会根据上传文件数量有展示不同效果。未上传附件：显示暂未上传；上传一个附件：显示当前附件；上传附件>1：显示最新上传附件，出现展开按钮，文件按照先后顺序从上往下排序。可点击展开按钮，展开附件列表。展开后显示三条附件其余以滚动形式呈现，点击折叠按钮可折叠。

文件上传后显示在附件列表中，上传成功后，显示删除按钮。点击删除可删除对应附件。

上传附件模块：

用户发送反馈信息之前需要勾选同意软件收集数据，默认为勾选，不勾选则发送按钮变灰无法点击发送。



取消勾选按钮变灰无法点击

用户可点击查看按钮查看收集的数据内容，如有些内容比较敏感可进行删除操作。如果重新录制视频，收集的数据进行覆盖。

10. 文件锁定功能



如上图所示，新版本为用户提供了可视化的文件锁定配置功能,软件 90%的功能可通过此命令实现锁定配置，所有可配置功能均罗列在命令界面；

通过【更换 Logo】功能可更换软件左上角 Logo，只需输入 Logo 链接即可；打开文件左上角 Logo 即可自动更换，此功能需要联网；

需要注意 logo 文件的格式需为 SVG 格式。

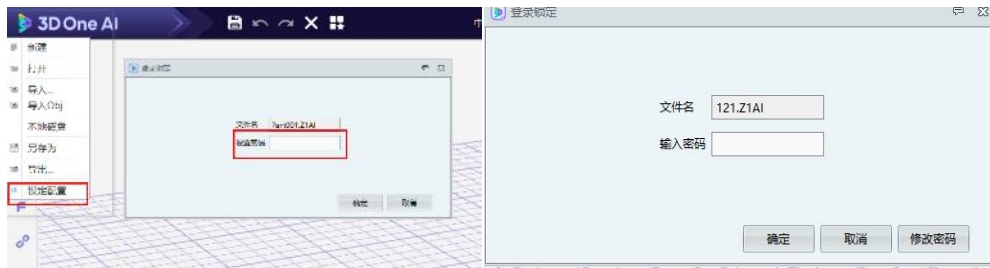
设置【显示北京时间】可以在打开文件时，界面上方显示北京时间,此功能需要联网；

点击【配置生效】即可进行配置锁定，同时按钮变为【锁定失效】；

点击【锁定失效】可以取消锁定配置，并灰置界面进入不可配置状态，同时按钮变为【锁定生效】。

配置完毕点击【应用】即可完成文件的配置锁定。

点击【重置】可一键取消所有勾选的配置项进行重新设置；



该功能提供文件锁定配置密码功能，密码可以为英文字母、数字、汉字以及常用符号，当文件进行了配置锁定，若设置了密码则只有知道密码的用户才能使用此功能进行文件锁定配置编辑，使用时需要输入密码才可登录，文件锁定密码只在首次打开文件进行锁定配置时需要；若不设置密码，则所有用户均可重新对此文件进行锁定配置编辑，使用时忽略密码直接点击登录即可；