





# 3D One AI V2.0 使用说明书



广州中望龙腾软件股份有限公司



H 22	

一、左侧工具栏	12
1. 基本编辑	
1.1 对齐功能	
1.2 吸附功能	
1.3 测量功能	
2. 全局属性设置	
3. 物体属性设置	
3.1 高级参数	
3.2 物体类型:仅仿真	
3.3 物体类型: 地形类型	20
4. 边界(3D One Al V2.0 版本新增功能)	22
5. 受力与速度	
5.1 速度	25
5.2 受力	25
6. 关节	
6.1 智能关节(3D One Al V2.0 版本新增功能)	27
6.2 关节设置	28



6.2.1 关节类型设置28
6.2.1.1 合页关节28
6.2.1.2 铰链关节29
6.2.1.3 活塞关节30
6.2.1.4 插销关节31
6.2.1.5 球窝关节32
6.2.1.6 万向轮关节33
6.2.1.7 PR 关节(插销-合页关节)
6.2.1.8 PU 关节(插销-万向轮关节)
6.2.2 高级参数设置(3D One Al V2.0 版本新增功能)
7. 设置电子件模型37
7.1 风扇
7.2 触碰传感器
7.3 开关40
7.4 LED 灯
7.5 舵机43
7.6 马达44
7.7 循迹传感器46
7.8 双向马达
7.9 电子显示屏



7.10 距离传感器	51
7.11 力传感器	53
7.12 光线传感器	54
7.13 颜色传感器	
7.14 虚拟摄像头	57
7.15 转盘	59
7.16 位置传感器	60
7.17 灰度传感器	61
7.18 机械臂	63
7.19 插销夹爪	65
8. 设置控制器	67
9. 设置第一人称视角	68
10. 全局属性(3D One Al V2.0 版本新增功能)	69
11. 组(3D One Al V2.0 版本新增功能)	
二、库资源与编程	
1. 模型库	
2. 样例库	
3. 案例库	73
三、编程与接口说明	74



1.	编程建	模	74
2.	编程物	理属性	75
3.	编程控制	制器——虚拟传感器	77
	3.1	设置距离传感器	78
	3.2	获取距离传感器距离	79
	3.3	设置触碰传感器	79
	3.4	获取触碰传感器触碰	80
	3.5	设置力传感器	80
	3.6	获取力传感器测量力度	81
	3.7	设置光强传感器	81
	3.8	获取光强传感器测量光强	82
	3.9	设置位置传感器	82
	3.10	获取位置传感器坐标	83
	3.11	设置颜色传感器	83
	3.12	获取颜色传感器结果	84
	3.13	启动灰度传感器	84
	3.14	检测灰度传感器左右灰度	85
	3.15	启动循迹传感器	86
	3.16	检测循迹传感器左右轨迹	86



4.	编程控制器——电子件	. 88
	4.1 设置舵机角度	. 88
	4.2 设置单向马达转动	. 88
	4.3 设置单向马达转动速度	. 89
	4.4 设置双向马达正向/反向转动	. 89
	4.5 设置双向马达转动速度	. 90
	4.6 获取开关状态	. 90
	4.7 设置开关状态	. 91
	4.8 设置 LED 状态	. 91
	4.9 设置电风扇	. 92
	4.10 设置电风扇转动速度	. 92
	4.11 设置电子屏状态	. 93
	4.12 设置电子屏显示内容	. 93
5.	编程控制器——图像识别	. 94
	5.1 启动电脑摄像头	. 96
	5.2 启动虚拟摄像头	. 96
	5.3 启动图片对象检测	. 97
	5.4 图片对象检测结果	. 97
	5.5 启动图片文字识别	97



	5.6 图片文字识别结果	98
	5.7 启动图片交通信号灯识别	98
	5.8 交通灯识别结果	98
	5.9 启动图片颜色聚类中心识别	99
	5.10 聚类中心位置中心的识别,	99
	5.11 最近颜色聚类中心位置	99
	5.12 最近聚类中心颜色	. 100
	5.13 启动图片颜色识别	. 100
	5.14 图片颜色识别结果	. 100
	5.15 启动图像循路	. 101
	5.16 判定循路结果	. 101
	5.17 图像循路使用简介	. 101
	5.18 识别结果包含	. 102
6.	编程控制器——语音识别	. 103
	6.1 电脑麦克风启动	. 103
	6.2 语音识别秒数	. 103
	6.3 语音识别结果包含	. 104
7.	编程控制器——机器学习	. 104
	7.1 保存训练分类	. 106



	7.2	导入训练分类	. 106
	7.3	电脑摄像头	. 107
	7.4	机器学习摄像头启动	. 107
	7.5	虚拟环境拍摄分类图片	. 107
	7.6	开始分类	. 108
	7.7	分类结果	. 108
	7.8	分类结果信心	. 108
	7.9	分类结果	. 109
	7.10	虚拟机器学习案例演示	. 109
8.	编程控制	制器——机械臂	. 111
	8.1 连接	硬件	. 111
	8.2 机械	戏臂初始化	. 111
	8.3	机械臂气爪合起	. 112
	8.4	设置夹爪旋转	. 112
	8.5	获取机械臂关节角度	. 113
	8.6	获取夹爪旋转电机的角度	. 113
	8.7	获取坐标轴角度	. 114
	8.8	设置机械臂角度和高度	. 114
	8.9	等待机械臂运动停止	. 115



8.10	设置转盘轴转动角度	116
9.	编程控制器——控制	117
9.1 键盘	盘按下	117
9.2 鼠核	示按下	117
9.3 设置	置物体移动速度和方向	118
9.4 设置	置物体旋转角度与方向	118
9.5 设置	置当前时间	119
9.5 重置	置当前时间	119
9.6	判断当前时间	120
9.7 设置	置等待时间	121
9.8 当育	前计时	121
9.9 获耳	双仿真时间	121
9.10 重	置仿真	122
9.10 退	出控制器	123
9.11 隐	藏物体及其关联积木块	123
9.12 显	示物体	124
9.12 在	特定时间显示特定信息	125
9.13 仿	真循环	125
10.	编程控制器——物理	126



10.1 重置物体线速度方向与大小	126
10.2 重置物体角速度方向与大小	126
10.3 获取物体线速度	127
10.4 获取物体角速度	127
10.5 重置物体受力方向与大小	128
10.6 重置物体受力矩方向与大小	128
10.7 重置物体重力模式	129
10.8 获取物体重力模式	129
10.9 重置物体位置坐标	130
10.10 获取物体位置坐标	130
10.11 重置物体质量	131
10.12 获取物体质量	131
10.13 重置物体线速度阻尼大小	131
10.14 获取物体线速度阻尼	132
10.15 重置物体角速度阻尼大小	132
10.16 获取物体角速度阻尼	133
10.17 重置物体摩擦系数大小	133
10.18 重置物体弹性系数大小	134
10.19 重置物体受力大小和方向以及受力点坐标	134

	10.20	物体是否移动到指定区域	135
	10.21	物体是否相交1	L35
	10.22	物体是否离开指定区域	136
	10.23	物体是否离开半径为规定值的区域内	L37
	10.24	物体是否根据指定轴转动一定角度	L37
	10.25	获取物体的重置次数1	138
	10.26	物体是否移动到了另一个物体指定误差区域内	138
	10.27	设置一个项目的得分以及其显示位置1	139
	10.28	设置左上角是否显示重置次数 1	139
	10.29	设置仿真时间是否显示1	140
	10.30	积分系统全局解释1	141
11	1.	编程控制器——关节1	141
	11.1 i	设置合页关节和旋转角度	141
	11.2 i	设置合页关节速度	142
	11.3 i	设置插销关节移动距离	142
	11.4 i	设置插销关节速度	143
	11.5 i	设置铰链关节力矩1	143
	11.6 i	设置铰链关节旋转角度	144
四、	其他		144



1.minibar
2.仿真过程控制145
3.仿真设置(3D One Al V2.0 版本新增功能)
3.1 显示环境设置146
3.2 天空样式设置148
3.3 材质设置149
3.4 粒子效果设置150
4. 百度在线识别功能(3D One Al V2.0 版本新增功能)
5. Zspace 立体渲染(3D One Al V2.0 版本新增功能)
6. 比赛文件定制化处理152
6.1 比赛文件定制化处理152
7. 库模型配置的关节与装配处理155
7.1 关节装配处理155
7.2 编辑关节装配处理158

# 一、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。

1.1 对齐功能

对齐两个实体

示意图:

	👂 对齐移动		?	×	~	
	实体1					ŀ
	实体2					
	. €	Q	4 ⊥	6		·
	偏移	0			÷ v	ł
2.2 对齐移动	角度	0			÷	ŀ
	◉ 共面		◎ 相反			-

简单示例:





# 1.2 吸附功能

吸附两个物体

示意图:



	훩 自动吸附		?	×	~
自动吸附	实体1				
	实体2				

# 简单示例:







# 1.3 测量功能

### 测量两个点之间的距离

示意图:

	▶ 距离测量 ? X ✓
	× × × ×
	点1 》
	点2 》
	距离
	X 方向距离
	Y 方向距离
距离测量	Z方向距离

# 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<sup>(2)</sup>,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。

🏓 全	局属	性设置	?	×	~
世界	地球				*
重力	-9.80	D		÷	N/kg
▽ 高级属	酣				
碰撞修正	E系数	0.2			* *
约束柔转	欠系数	0.0			*
线速度限	眤	0.0			* *
角速度限	眤	0.0			* *
			_		

图—

#### 3. 物体属性设置

物体属性 赵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:

👂 物体属性设置 🛛 ? 🛛 🗙 🗸 🗸						
实体	选中1个	~				
物体类型	호	Ψ.				
名称	body_2	>				
材料	塑料	*				
▶ 高级参数						



3.1 高级参数

V2.0 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
点击
高级参数
,在下拉框内即可调整。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。
示意图:

👂 物体属性	註设置 <b>?</b> 〉	×	~
实体	选中1个		¥
物体类型	窒		*
名称	body_2		>
材料	塑料		Ŧ
▽ 高级参数			
质量	0.001	* *	kg
摩擦系数	0.350	* *	
弹性系数	0.500	* *	
物体正面	1,0,0	>	
🗹 重力模式			

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,∞),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。



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:



👂 物体属	性设置	×	~
实体	选中1个		¥
物体类型	仅仿真		*
名称	body_2		>
材料	塑料		Ŧ
▽ 高级参数			
质量	0.001	÷	kg
摩擦系数	0.350	÷	
弹性系数	0.500	÷	
物体正面	1,0,0	>	
☑ 重力模式			

# 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:



物体属	性设置	×
实体	选中1个	~
物体类型	地形	-
名称	body_2	>
材料	塑料	*
∨ 高级参数		
质量	0.001	‡ kg
摩擦系数	0.350	* *
弹性系数	0.500	÷
物体正面	1,0,0	>
🔽 重力模式		

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,导致了许多仿真异常现象 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。



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。



(线速度)

(角速度)

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。

<b>&gt;</b>	受力与速度	? X	~
	0	7	
() 实	体 💿	点	
实体	S1		
点	-71.873,-14.343,20		2
方向	1,0,0		>
大小	10		‡ N
	++++		
	+++		
		$\overline{-}$	
À	$\pm\pm\pm$	7	
1	+++		
	+++	77	

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:

		👂 智能	送节	?	×	~
		移动体轴				2
		基体轴				2
		移动体点				>
		基体点				>
	-	🔽 创建关	Ť			
C	智能关节	关节类型	合页关节			*
		关节名称	合页			>
3	S.	锚点				>
1	V@	轴心			>	2

简单示例:



(通过面选择两个物体的对齐轴向)

关节类型	合页关节
关节名称	合页关节
++	活 奉天卫 插销关节
通品	球窝关节
轴心	

(选择生成的关节)

(通过点选择两个物体的对齐点)



(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。



🎐 关节设置	? X ✓
关节类型	合页关节 🔹
关节名称	合页 >
实体1	
实体2	
锚点	>
轴心	0,0,1
▶ 高级参数	
轴心 物体1 描点	物体2

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。



关节设置	? X 🗸
关节类型	铰链关节 ▼
关节名称	铰链 >
实体1	
实体2	
锚点	>
轴心1	0,0,1
轴心2	-1,0,0
+抽心1 物体2 物体1 +抽心2 - 描点	

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。



关节设置	? × ✓
关节类型	活塞关节
关节名称	活塞 >
实体1	
实体2	
锚点	>
轴心	1,0,0
▶ 高级参数	
抽心 前心 物体1 物体2	

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。



👂 关节设置	? X 🗸
关节类型	插销关节 🔹
关节名称	插销 >
实体1	
实体2	
轴心	1,0,0
> 高级参数	
<b>初</b> 体1	1000000000000000000000000000000000000

**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。



👂 关节设置	? X V
关节类型	球窝关节 🔹
关节名称	球窝
实体1	
实体2	
锚点	>
▶ 高级参数	
物体1	物体2

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。



👂 关节设置	? X 🗸
关节类型	万向轮关节
关节名称	万向轮 >
实体1	
实体2	
锚点	>
轴心1	0,0,1
轴心2	0,-1,0
轴心1	
物体1	物体2
轴心2	

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)。



🔰 关节设置	? X ✓
关节类型	PR关节 🔹
关节名称	PR >
实体1	
实体2	
锚点	>
轴心R	0,0,1
轴心P	1,0,0
<del>轴</del> 心P 物体2 物体1	

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;

错点: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。



👂 关节设置	? × ✓	
关节类型	PU关节 *	
关节名称	PU >	
实体1		
实体2		
锚点	>	
轴心1	0,1,0	
轴心2	0,0,1	
轴心3	1,0,0	
抽心2 物体2 物体2 物体2 物体1		

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:


<b>》</b> 关f	市设置	?	× ✓
关节类型		合页关节	*
关节名称		合页	>
实体1			
实体2			
锚点			>
轴心		0,0,1	Z
▽ 高级参	数		
初始角度	0		* *
最小角度	0		* *
最大角度	0		÷
关节紧度	0.200 🗘	_	
轴心 物体2 物体1			

7. 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。



(当未设置或不存在电子件时,提示无电子件编程对象)



7.1 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;

👂 设置	跑子件模型 <b>? × 、</b>	/
电子件类型	风扇	*
电子件		
速度方向		>
转速人	转速方向	

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。

### 7.2 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:



## 7.3 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



👂 设置	油子件模型 <b>? ×</b>	~
电子件类型	开关	*
电子件		
触碰面		
按下距离	-5	÷
触碰面	投下	



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:





7.4 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:





7.5 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:



### 7.6 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。



훩 设置	油子件模型 <b>? × ×</b>
电子件类型	马达 🔹
电子件	
速度方向	>
速度	20 ‡
转速大	小及方向



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:





### 7.7 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



👂 设置	电子件模型 <b>? ×</b>	~
电子件类型	循迹传感器	Ψ.
电子件		
左检测点		>
右检测点		>
范围	50	* *
右 Manana 注:	检測点 た た た や 測点 た た 池測点 た た 池測点 の の の の の の の の の の の の の	

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:





## 7.8 双向马达

双向马达的用法与单向马达类似,比单向马达多了反向转动的自由度,可以控制物体在 平面内 360 度转动。

电子件:选择需要被设置为马达的实体;

**速度方向**:正向转动方程,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:正向速度,单位为 rad/s。

**反向速度方向**:反向转动方程,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;

反向速度:反向速度,单位为 rad/s。



👂 设置印	B子件模型 ? X 、	~
电子件类型	双向马达	*
电子件		
速度方向		>
速度	20	+
反向速度方向		>
反向速度	20	+
正向转速及大小 页向转速及大小 双向马达		



程序设计:根据键盘的方向键来控制双向马达的正向和反向转动。对应的积木如下图

所示:





#### 7.9 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:

◎ 设置 电子屏 电子显示屏 □ 开启 □
· 仿真循环
读置 电子屏 (电子显示屏 ≤ 內容 ( 3D ) 大小 10 字体: Times New Roman ≤     □□     □□□□□□□□□□□□□□□□□□□□□□□□□
等待 0.5 秒
读置 电子屏 电子显示屏 ≤ 內容 ( One  >> 大小 10 字体: Times New Roman ≤     □□
等待 0.5 秒
後置 电子屏 (电子显示屏 ≤ 内容 ( · Al >> 大小 (10) 字体: Times New Roman ≤
等待 🛯 秒

### 7.10 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)



👂 设置	油子件模型 <b>? × ×</b>
电子件类型	距离传感器 🔹
电子件	
起始位置	>
方向	>
范围	200 ‡
● ● 射线方	I线长度 Wing Name



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:





## 7.11 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:



## 7.12 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



👂 设置	跑子件模型 <b>? ×</b>	~
电子件类型	光线传感器	٣
电子件		
检测面		
Ċł	模型 选择有效检测面	

		☆ 显示环境	
	调节光照强度	显示质量	<b>#</b>
		光照强度	3.00 🛟
		光照高度	<b>商</b>
		光照角度	东南 🗸
		投影模式	透视投影
		≪ 天空样式	
	LEDXJ	天空主题	白天
光强传感器		☆ 材料	
		实体	> 割除材料 全层 ±材 研存
		☆ 粒子效果	

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:



## 7.13 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:

### 7.14 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



电子件类型	虚拟摄像头	
电子件		
起始位 <mark>置</mark>		>
方向		>
视角方		



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:





## 7.15 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

👂 设置电	B子件模型 ? X 🗸
电子件类型	转盘    ▼
转盘轴	
转盘	
转盘转轴锚点	>
转盘转轴方向	2
	特抽

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。



### 7.16 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



👂 设置	电子件模型 ? X 、	1
电子件类型	位置传感器	*
电子件		
起始位置		>
选择检	潮位置初始坐标点 选择模型	



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 ×

大于 50 则停止前行,否则前行。对应的积木如下图所示:



## 7.17 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

👂 设置	电子件模型 <b>? ×</b>	~
电子件类型	灰度传感器	*
电子件		
左检测点		>
右检测点		>
范围	50	* *
ź.	检測点 左检測点 を放測点 な初点 大校測点 た 位別点 大校測点	

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:



#### 7.18 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

设置印	a子件模型 ? X 🗸	
电子件类型	机械臂	
机械臂		
Z轴臂		
Z轴方向	2	
大臂		
大臂锚点	>	
大臂轴方向	2	
小臂		
小臂锚点	>	
小臂轴方向	2	
Z轴位置偏差	0 ‡	
大臂角度偏差	0 ‡	
小臂角度偏差	0 ‡	
Z轴方向         小臂		

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°。

设置 (夹具_Part001_机械管一机械管气机 松开一
後重 <del>実具_</del> Part001_机械育 ■ 设置夹系族转 <b>● 0</b> 度
● ● ● ● ● ● ● ● ● ●
347 ピ 後夏 机身 Part001 机械管 = 改夏机械管角度 高度Z (
NT 2 現身_Part001_机放育 == 改互机结常角度 高度Z 2 机身_Part001_机放育 == 衣取坐标論 Z == 位置坐标 大背角度 (45) 水背角度 (300)
教行 👔 説量 机身_Part001_机械管 = 设置机械管角度 高度Z 🔞 大管角度 (11.9_Part001_机械管 = 表取机械管关节角度 大管关节 = 度) 小管角度 (11.9_Part001_机械管 = 表取机械管关节角度 小管关节 = 度) 注度 (300
外行 🖉 设置 机身_Part001_机械管 回 设置机械管角度 高度Z 🕕 大管角度 ( 通) 大管角度 ( 通) 大管角度 ( 通) 水管角度 ( 通) 水 ( Ⅲ) ( Ⅲ) ( Ⅲ) ( Ⅲ) ( Ⅲ) ( Ⅲ) ( Ⅲ) (
秋行 <b>直复直到 【 </b>
後望 英具_Part001_机械管 机械管气质 合起 ■
注意 来見 Part001 机械管 - 机械管气机 松开 -
*** 2回 来具 Part001_机放管 □ 设置先术旋转 ●90 度

## 7.19 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:

	يستغير وكتر
	手套
Service second point	



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。

□ 设置 [夹具 ] 机械臂气爪 [松开 ]
─────────────────────────────────────
"     "
· · · · · · · · · · · · · · · · · · ·
◎ 如果 〈 」 键盘 a   按下
执行 送置 机身 · 设置机械臂角度 高度Z (30) 大臂角度 (0) 小臂角度 (0) 速度 (300)
■ 「重复直到 ■ く おう ● 新会动停止
执行 等待 [0.1] 秒
☆ 設置 実具 ・ 机械臂气爪 合起 ・
重复直到 🔟 🤇 👔 初身 🔹 等待运动停止
执行 等待 [0.1] 秒
300 00 大臂角度 ● 小臂角度 ● 水子 ● 水子 ● 水子 ● ひょう ● ひ
■ 「重复直到 」 ( 水身 」 等待运动停止
执行 等待 [0.1] 秒
登置 机身 ■ 设置机械臂角度 高度Z ● 大臂角度 ● ● 大臂角度 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
重复直到 1 ( 一) 新日本 等待运动停止
执行 等待 [0.1] 秒
2 设置 実具 → 机械臂气爪 松开 →
【重复直到 ▼ (
执行 等待 [0.1] 秒
2 设置 机身 □ 设置机械臂角度 高度Z ● 大臂角度 ● 小臂角度 ● 速度 300

## 8. 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。



### 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



	👂 设置第一人称视角 <b>? × ×</b>
	实体
	点
	物体正面
~*	
	🎯 第一人称视角 🛛 🔿 😳 —

# 10. 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:





## 11. 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:







## 二、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)

	参数化模型	基本形状 (13)
大地拉巴	已配置信息模型	机器人 (0)
<b>平地快空</b>		车 (8)
		场景 (0)
	传感器 (7)	
电子件	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。



	机器学习	识别 语音识别	子件图像	传感器
改置力传感器 力传感器 2 月	-	() ()	林感器・「島用・	2. 设置位置传感器 位置付
		调整环境分		
力传感器 插	插入	光强传感器	插入	位置传感器
· 秋期餘時作成器 <b>餘盛传感器 二</b> 1		(		1
	▲·/文代48-28 □ 人1月1 □	Contraction of the second seco	前述作感苦	Co attended
		P P		

# 三、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。





# 3.1 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')



	设置距离传感器	无传感器	•	启用	Ŧ
$ \longrightarrow $					

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。

#### 3.2 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:

获取距离传感器 **无传感器 🔻** 测量距离 ----

参数说明:

# 3.3 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')



6	设置触碰传感器	无传感器 🔻	[ 启用 ▼
B			

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。

# 3.4 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:

获取触碰传感器 【无传感器 ▼】触碰 P

参数说明:

SensorName	触碰传感器的名称 (字符串)

# 3.5 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')



	设置力传感器	无传感器	启用 ▼
ë			

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。

# 3.6 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:

获取力传感器 **无传感器 🔻** 测量力度 =

参数说明:

# 3.7 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')





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。

# 3.8 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)

# 3.9 **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。	

# 3.10 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:

获取位置传感器 无传感器 🔽 位置坐标 🗴 🗸 L

参数说明:

SensorName	位置传感器的名称(字符串)

# 3.11 **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)
--	------------	----------------



St	tate	颜色传感器的状态,选项为"Enable"	
		或"Disable"分别表示启动或不启动。	

# 3.12 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:

● 无传感器 检测颜色为	RGB值偏差(10)
--------------	------------

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	

# 3.13**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')





```
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。

# 3.14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:

	灰度传感器	无传感器	▼ 检测到	左边 🔻	是 黑色	?
- U						

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

# 3.15**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。

# 3.16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')



# 设置循迹传感器 无传感器 ◇ 检测到 左边 ◇ 是轨迹 ?

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。

훩 物体属性	生设置 ?	× ✓
实体	选中1个	~
物体类型	空	*
名称	空 动画	
材料	仅显示	
▽ 高级参数	仅仿真	
质量	【 地形 【 跟随显示	
摩擦系数	0.350	÷
弹性系数	0.500	÷
物体正面	1,0,0	>
☑ 重力模式		

循路逻辑与灰度传感器一致。



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

# 4.编程控制器——电子件

# 4.1 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

# 4.2 设置单向马达转动

此函数用于设置单向马达启动与否。

示例:

zw\_set\_onewaymotor('body\_3', 'ON')

积木块:



参数说明:



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

# 4.3 设置单向马达转动速度

此函数用于设置单向马达的转动速度。

示例:

zw\_set\_onewaymotor\_speed('body\_3', 50)

积木块:



参数说明:

Name	单向马达的名称 (字符串)
Speed	单向马达的转动速度

# 4.4 设置双向马达正向/反向转动

此函数用于设置双向马达启动与否以及转动方向。

示例:

zw\_set\_twowaymotor('body\_3', 'POSITIVE', 'ON')

积木块:



参数说明:



Name	双向马达的名称(字符串)
Direction	双向马达的转动方向,选项为''Positive"
	或"Negative"分别表示正向或反向转动。
State	单向马达的转动状态,选项为"ON"或"OFF"
	分别表示开始或停止转动。

# 4.5 设置双向马达转动速度

此函数用于设置双向马达的转动速度。

示例:

zw\_set\_twowaymotor\_speed('body\_3', 50)

积木块:



参数说明:

Name	双向马达的名称(字符串)
Speed	双向马达的转动速度

# 4.6 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')





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。

# 4.7 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。

# 4.8 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')





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。

# 4.9 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。

# 4.10 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)





FanName	电风扇的名称 (字符串)
Speed	电风扇的转速。

#### 4.11 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。

#### 4.12 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')



	设置	电子屏	body_1 🔹 内	溶 🥌	3D One Al	>>	大小 🤇	10	字体:	幼圆	7
(212)											

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. 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):

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

🛪 3D One Al 中望大学测试中心 永久授权 - 教育版 - [Part001\*] ? ŵ, 🖾 图像识别 ŵ () 開像 麗 F 8 st. 0 ◎ 拍摄区 💿 R-ARRA 🔇 D 💽 9 0 0 C 0 U

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:



# 5.1 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:



5.2 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')



Ø	虚拟摄像头	无传感器	V	启动

CameraName	虚拟摄像头的名称 (字符串)

#### 5.3 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:



# 5.4 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:



5.5 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()



# 5.6 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:



# 5.7 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:



# 5.8 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)
-------	-------------



# 5.9 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

# 5.10 聚类中心位置中心的识别,

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
----------	-----------------

# 5.11 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')





Position	Left right 或者 middle

## 5.12 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:

# 5.13 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:



5.14 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

 $zw\_image\_recognize\_colordetect\_result('Red')$ 





Color	图片识别的颜色 (字符串)

# 5.15 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:



# 5.16 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。

图像循路结果: 中

参数说明:

left right m	iddle

左边、右边或中间

5.17 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。



# 5.18 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:

识别结果	<b>a</b>	图片对象检测结果	包含(	66	"	
------	----------	----------	-----	----	---	--

参数说明:

Text	给定的文本 (字符串)

# 6. 编程控制器——语音识别

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 20mm/s

● 电脑麦克风 启动
□ 仿真循环
◎ 如果 〈 ↓ 键盘 a ▼ 按下 ▼
执行 语音识别 持续 (2) 秒
○ 如果 < 语音识别结果 包含 (前进)
执行 body_1 3 沿直线移动 向前 3 速度为 (10) cm/s

# 6.1 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:



# 6.2 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。

zw\_speech\_recognize(int)



<u>(</u> گ)	语音识别	持续	2	秒
参数说明	:			

int	识别时长

# 6.3 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:

#### 7. 编程控制器——机器学习

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"训练分类"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:

	🦲 模型训练	
<ul><li>20 关闭摄像头</li></ul>		水杯 (1) 拍照 (2) 重置
		卷紙 ⑥ 拍照 ⑥ 重置
	● 新建分类	🛛 训练模型

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。

◎ 电脑摄像头 启动 ▼
☆ 仿真循环
◎ 如果 ~ · · · · · · · · · · · · · · · · · ·
执行 88 开始 分类 选择训练分类结果,如cat
◎ 如果 ( 〕 分类结果 是 无分类 2 ?
执行 重置 body_1 线速度 方向为 X: 1 Y: 0 Z: 0 值为 20

④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:



# 7.1 保存训练分类

保存训练好的分类到本地磁盘;

保存训练分类

# 7.2 **导入训练分类**

从本地导入训练好的分类;

导入训练分类



# 7.3 **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:



7.4 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:



7.5 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





# 7.6 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:



# 7.7 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:



# 7.8 **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()




### 7.9 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:



#### 7.10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。



机器学习分类状态: Success:C:/Users/Administrator/AppData/Local/Temp/ZwAi/ClassifyRetrain/classify1/2020-09-18-09-01-13-113.jpg

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。



# 8. 编程控制器——机械臂

## 8.1 连接硬件

点击连接硬件可以控制实物机械臂。

积木块:



示例:

▶ Robot Arm Product						
-Robot ArmProduct-	- Robot ArmProduct					
Robot Arm Product	HitBot		Ŧ			
сом	COM1		Ŧ			
PORT	32000		÷			
Exe Path						
Connect						
	确定取	消				

#### 8.2 机械臂初始化

示例:

zw\_robotarm\_initial('1')

积木块:



参数	类型	默认值	说明
СОМ	str	无	需要初始化的 COM 口号



#### 8.3 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	说明
		值	
body_name	str	无	夹具的名称
state	boolean	False	True 为 Open,False 为 Close

### 8.4 设置夹爪旋转

例子:

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

积木块:



参数	类型	默	说明
		认值	
Body_name	Str	无	夹具的名称



degree float 无	夹具转盘转动的角度
----------------	-----------

#### 8.5 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

### 8.6 获取夹爪旋转电机的角度

示例:

zw\_robotarm\_getRmotordeg('夹具',)

积木块:



参数	类型	默认值	说明
Body_name	Str	无	夹具的名称



#### 8.7 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
хуz	tuple	无	选择获取坐标的轴

# 8.8 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	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

#### 8.9 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

### 8.10 设置转盘轴转动角度

示例:

zw\_robotarm\_turntablemove('转盘轴', 22.5)

积木块:





参数	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

# 9. 编程控制器——控制

9.1 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

# 9.2 鼠标按下

示例:

zw\_get\_mouse('Left', 'Down')

积木块:



4	2	鼠标	左键,	<b>按下▼</b>
	4			

### 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

# 9.3 设置物体移动速度和方向

示例:

zw\_move\_linearmove(' ','forward',10)

积木块:

ta	无实体	沿直线移动	向前 🔹	] 速度为 🌘	10	cm/s

# 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

# 9.4 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

# 9.5 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

# 9.5 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

#### 9.6 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。

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



# 9.7 设置等待时间

示例:

zw\_sleep(0)

积木块:



参数说明:

Second	等待秒数

# 9.8 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。

设置电子屏 电子显示屏 内容(	当前计时
-----------------	------

9.9 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。

9.10 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。





#### 9.10 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:



#### 9.11 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。

## 9.12 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)

# 9.12 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:

C	在时间:	0时0分0秒	显示消息: (	< Message	"	显示持续时间: 🙆 秒
-0						

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。

#### 9.13 仿真循环

示例:

while zw\_step():

pass

积木块:





#### 10. 编程控制器——物理

## 10.1 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

### 10.2 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:



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

### 10.3 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	)
-----------------------	---

# 10.4 获取物体角速度

示例:

zw\_body\_get\_angular\_vel(' ')

积木块:



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

# 10.5 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 × 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

# 10.6 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:



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

# 10.7 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。

### 10.8 获取物体重力模式

示例:

zw\_body\_get\_gravity\_mode(' ')

积木块:





Name

需获取物体的名称 (字符串)

#### 10.9 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

# 10.10 获取物体位置坐标

示例:

zw\_body\_get\_position(' ')

积木块:





### 10.11 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

# 10.12 获取物体质量

示例:

zw\_body\_get\_mass(' ')

积木块:



参数说明:

#### 10.13 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

# 10.14 获取物体线速度阻尼

示例:

zw\_body\_get\_linear\_damping(' ')

积木块:

获取 无实体 • 线速度阻尼 anter

参数说明:

Name	需获取物体的名称 (字符串)

# 10.15 重置物体角速度阻尼大小

示例:

zw\_body\_set\_angular\_damping(' ', 0.5)

积木块:





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

### 10.16 获取物体角速度阻尼

示例:

zw\_body\_get\_angular\_damping(' ')

积木块:



参数说明:

# 10.17 重置物体摩擦系数大小

示例:

zw\_body\_set\_friction\_coefficient(' ', 0.5)

积木块:



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



### 10.18 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

### 10.19 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:

重置 无实体 2 力和受力点 方向为 (X: 1 Y: 0 Z: 0 值为 20 N 坐标为 (X: 1 Y: 0 Z: 0

Name	需重置物体的名称(字符串)
X axis direction	力的 × 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

#### 10.20 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:

★ 无实体 是否移动到区域 左上角 X: -20 Y: 20 Z: 0 右下角 X: 20 Y: -20 Z: 0

参数说明:

X axis coordinate	左上角的 × 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 × 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

## 10.21 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)

### 10.22 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:

⑦ 无实体 是否离开区域 左上角 (X: ●20 Y: ② Z: ◎) 右下角 (X: ② Y: ●20 Z: ◎)

Name	物体的名称(字符串)
X axis coordinate	左上角的 × 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

# 10.23 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

# 10.24 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:



Name	物体的名称(字符串)
Rotation angle	旋转角度



X axis coordinate	旋转轴的 × 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

### 10.25 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)

#### 10.26 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:





Name	检测区域物体的名称(字符串)
Radius	误差半径

### 10.27 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('ltem1',0,1)
```

积木块:

	< (Item1) >>	) 设置分数 🤇	0	项目分数显示顺序(	1
100					

参数说明:

Name	项目的名称(字符串)
Score	分数
Order	显示顺序

## 10.28 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。

# 10.29 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:



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。



#### 10.30 积分系统全局解释

设置重置次数是否显示(真一
·····································
·····································
·····································
·····································
·····································
·····································
·····································
·····································
·····································

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

#### 11. 编程控制器——关节

## 11.1 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

# 11.2 设置合页关节速度

示例:

zw\_controller\_velocity\_hingejoint(' ',1)

积木块:

œ	无关节 🔻	设置合页关节	速度为	1	N·mm

参数说明:

Name	物体的名称(字符串)
Speed	速度

# 11.3 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

# 11.4 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

# 11.5 设置铰链关节力矩

示例:

zw\_move\_linear(' ',1)

积木块:



Name	物体的名称(字符串)
------	------------



Torque

力矩

### 11.6 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

# 四、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 (3D One AI V2.0 版本新增功能)

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。

示例图:

☆ 显示环境	
显示质量	<b>中</b>
光照强度	3.00 🛟
光照高度	高 <b>▼</b>
光照角度	东南
投影模式	透视投影

其中显示质量分为低、中、高、自定义四个选项,控制着抗锯齿、环境遮罩、阴影质量 三个显示参数,这三个显示参数对仿真帧率影响较大。其中低显示质量会将这三个参数都调 到最低或无,中显示质量与高显示质量仅在阴影质量这个参数上有区分。





(抗锯齿关)

(抗锯齿开)



(环境遮罩关)

(环境遮罩开)



(阴影质量关)



(高阴影质量)

(低阴影质量)



其他的显示环境设置基本不影响帧率,只是丰富环境渲染的设置。

# 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:

☆ 天空样式	
天空主题	白天



(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

☆ 粒子效果				
实体	1	删除效果		
粒子	火焰	烟雾		
	强度	1	* *	
	时间	1.00 🛟 -		分钟
	颜色			
	🗌 持续的		选择发射	时点
	□ 持续的		选择发射	村点

# 4. 百度在线识别功能 (3D One AI V2.0 版本新增功能)

此版本 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,每个账户的每个接口有 50 次识别体验次数,暂时不提供购买次数服务。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。当使用图像识别、文字识别、语音识别这三个在线识别接口时,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,选择使用低精度识别后会转

为本地离线识别。

3DOne <sup>A1</sup>	() 3D One Ai	₽ %
	本版本部分识别功能有次数限制,现文字识别 用完	刘使用次数已
	退出仿真使用低精度识	別
		4

5. Zspace 立体渲染 (3D One AI V2.0 版本新增功能)

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<sup>?</sup>,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:



设置	Ş		
自动备份设置			
□ 自动备份文件			
町间间隔 (会納)	10		
	10	_	
更新设置			
客户端设置			
🗌 使用局域网地	址检查更新		
服务器地址			
	·,		
─检查更新───	创建升级服务器		
	检查更新		
	检查AI模块更新		
-显示设置			
🔲 降低显示性能			
□ 仿真显示适配Z	□ 仿真显示适配Zspace立体渲染		
- 3D One 客户体验			
为我详细介绍。			
☑ 通过自动将您的日志文件发送到3D One社区来帮助进一步改进3D One产品			
		_	
	WHAL RX/FI		

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 <u>https://support.zspace.com/s/article/zSpace-Supported-Graphics-Cards-</u> <u>GPUs?language=en\_US</u>。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

 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。

	2	-
	ZwAiController	^
ie ie	0	
	2	
叧		_
-		Ť

4) 从鲸鱼专区下载的模型或者电子件,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:

> 关节装配处	œ ? X ✓
实体	
◉ 基体	◎ 移动体
锚点	>
轴心	2
匹配名称	>
🔽 创建关节	
连接类型	合页关节 🔹
🔲 使用轴心2	

- 1) 选择实体
- 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);



- 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:

