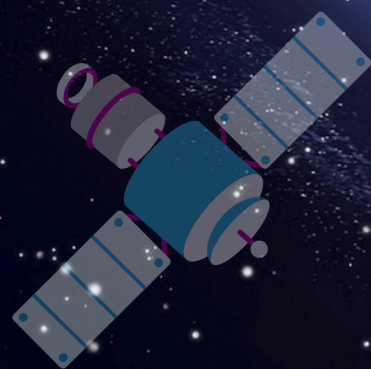


“程上启下 创想专家”



全国青少年科技教育成果展示大赛
启程-近地轨道回收挑战赛赛项
Scratch-vegeta智能硬件项目教程





赛事主题

竹苑科技
启程家

启程-近地轨道卫星回收挑战赛

科技与教育本质互通，
而考察青少年对近地轨道卫星、结构设计、人工智能算法、
硬件编程等有关知识的综合运用情况，
大赛特别设计「启程-近地轨道卫星回收挑战赛」，
希望通过比赛让青少年创客领会到“使命情怀，
团结协作、顽强拼搏、戮力创新、攻坚克难”的探空精神，
并以高水平的赛事设置来培养学生的逻辑思维能力、
创新意识和实际操作能力，提高应用编程水平，
激发学生的科技创新潜能。

初识 Scratch-贝吉塔机器人.....	8
第一课 进入 Vegeta 的世界	4
一、认识 Vegeta.....	5
二、Vegeta 软件界面.....	6
三、如何编写脚本和运行脚本.....	6
四、碰到边缘就反弹.....	11
第二课 飞舞的蝴蝶.....	15
一、导入新背景.....	16
二、添加删除角色.....	18
三、造型切换实现动画.....	21
四、多个角色和脚本的复制.....	23
五、工程的新建、保存和打开.....	26
第三课 神奇的魔法师.....	28
一、角色会说话.....	30
二、角色的特效.....	32
三、角色的大小、显示和隐藏.....	35
四、神奇魔法师的脚本编写.....	36
五、练习总结:	42
第四课 迷宫游戏.....	45
一、坐标和方向的理解.....	47
二、事件和键盘控制.....	50
三、条件判断和侦测.....	51
四、编写迷宫游戏.....	52
第五课 计算的乐趣.....	60
一、创建一个变量.....	61
二、为变量赋值.....	64
三、我来问你来答.....	66
四、统计得分.....	68
第六课 画梅花.....	72
一、导入新角色和背景.....	73
二、海龟作图法.....	75
三、绘制梅花树干.....	77
四、盛开的梅花.....	79
第七课 贪吃的小老鼠.....	83

一、找甜甜圈的路线.....	84
二、如何检测颜色.....	87
三、听话的小老鼠.....	90
创享 Robot	91
第一课 进入传感器的世界.....	93
一、切换到上传模式.....	93
二、上传模式主界面.....	94
三、Arduino 程序的基本框架.....	95
第二课 控制 LED 灯.....	97
一、硬件连接.....	99
二、驱动安装.....	100
三、串口连接.....	101
四、闪烁的 LED 灯.....	102
五、按键控制 LED 灯.....	103
第三课 交通灯.....	107
一、硬件连接.....	108
二、点亮 RGB 全彩灯	108
三、五彩斑斓.....	110
四、交通灯的实现.....	110
第四课 红灯停、绿灯行.....	114
一、RGB 全彩灯.....	115
二、添加背景角色.....	116
三、交通灯下的行人.....	117
第五课 灵活小河马.....	120
一、认识超声波.....	121
二、添加背景角色造型.....	122
三、给小蜜蜂添加脚本.....	123
四、给小球添加脚本并运行脚本.....	123
第六课 打气球.....	125
一、声音传感器.....	126
二、气球造型与爆炸.....	126
三、爆炸气球之打气的过程.....	127
实用 Robot	128
第一课 超声波台灯.....	129
一、搭建台灯造型.....	130

二、简易感应台灯.....	132
三、实用感应台灯.....	132
第二课 手持测距仪.....	134
一、搭建结构造型.....	136
二、编程基础程序.....	136
三、总结与反思.....	137
第三课 动力基础.....	139
一、搅拌器.....	140
二、后轮驱动车.....	149
chǐhuò	
三、机械 <u>尺 蠖</u>	155
chōng	
四、 <u>春</u> 米机	162
五、旋转杯子.....	168
六、稻草人.....	174
七、缝纫机.....	179
八、六足步行机器人.....	184
九、手动风扇.....	189
第四课 幸运大转盘.....	194
一、搭建抽奖机造型.....	195
二、幸运大转盘.....	196
三、随机的结果.....	197
掌控 机器人交互.....	200
第一课 模拟呼吸灯.....	201
一、视觉残留.....	202
二、LED 灯亮度的改变	204
三、呼吸灯的实现.....	205
第二课 避障机器人.....	210
一、组装贝吉塔小车.....	211
二、贝吉塔动起来.....	211
三、避障机器人.....	213
第三课 循迹机器人.....	215
一、红外循迹传感器.....	216
二、自定义程序块.....	218
三、循迹机器人.....	220

第三课 遥控警车.....	222
一、使用计时器实现闪烁的灯光.....	224
二、遥控警车.....	226
三、计时器的妙用.....	228
第四课 机器人足球挑战赛.....	231
一、机器人工程控制赛.....	231
二、点球机器人大赛（含守门员）.....	237
竞技-赛事专业提升课.....	242
第一课 赛场布置与道具.....	243
一、小学组卫星搭建.....	244
二、小学组卫星支架.....	253
三、小学组矿石支架.....	256
四、小学组回收舱.....	263
五、小学组火箭发射器.....	280
六、高中组桥梁.....	303
七、高中组旗帜.....	317
八、高中组收集框.....	323
九、高中组通讯卫星.....	334
第二课 赛事车体推荐搭建指南.....	354
一、车的基础载体.....	354
二、车身及框架.....	380
三、机械臂与机械爪.....	389
第三课 积木结构件基础.....	407
一、齿轮组结构件特点.....	407
二、结构件-轴的特点.....	407
三、结构件-梁的特点.....	408

第一章 编程新探索

// — 初识 贝吉塔机器人 — //



“第一章”

编程新探索



章节序言

编程是现代社会中最重要的技能之一。它已经不再是一个独立的领域，而是贯穿于各行各业中。从游戏开发到金融分析，从医学研究到天气预报，编程已经成为了许多领域的核心工具。

随着技术的发展和社会的变化，编程也在不断地进化和创新。新的编程语言、工具和框架不断涌现，为开发者提供了更加高效和优雅的方式来实现自己的想法。同时，新的应用场景和技术挑战也不断涌现，为编程带来了更加广阔和有趣的发展空间。智慧是教不来的，只有在体验中去探索发现。

本章就是遵循了从孩子的视角看世界，从实用性出发、以联系生活为主旨，将科学、艺术、拓展、理论、实操等操作一团，使孩子们从小就能实现自己变成科学家的梦想，更让他们具备科学的态度与钻研精神。在对课程的思考过程中，将图形化编程知识、数学学科知识、科学类知识、AI知识按难度与接受程度整合排布，分成了多个阶段。

愿景是：孩子们可以从信息获取、任务分解、模式生成、运算实施的步骤来探索编程学习的原理，同步培养学生解决问题的思维方法及习惯的生成；让学生们深入实践探究和开放式探索，体现沟通力，以动手操作和以探究为本，关于解决方案的决策也是由学生沟通生成。

通过分享想法，对设计方案、解决办法可精进、迭代，产生不同的结果；让学生参与团队合作，体现协作力，用相同的目标、期望，一起协作从而成为富有成效的团队；不受限于一个标准的答案，以适当的方式使用技术和工具设计、实现自己的作品。

——编者

补充：STEAM 是科学（Science）、技术（Technology）、工程（Engineering）、艺术（Art）和数学（Mathematics）的缩写。

STEAM 是一种重实践的超学科教育理念，强调任何事情的成功都不仅仅依靠某一种能力，而是需要综合应用多种能力。STEAM 理念旨在培养人的综合才能。

准备开始：

- 初识 Vegeta 的编程环境
- 学习不同类型的编程积木模块
- 创建你的图形化编程游戏

计算机的程序是由一系列的机器指令组成的，而指令就是要计算机执行某种操作的命令。

程序=算法+数据结构+程序设计方法+语言工具和环境

从计算机组成的层次结构来说，计算机的指令分：

- 微指令：是微程序级的命令，它属于硬件。
- 宏指令：是由若干条机器指令组成的软件指令，它属于软件。
- 机器指令：介于微指令和宏指令之间，通常简称为指令，每一条指令可以完成一个独立的算术运算或逻辑运算操作。

计算机程序就是指电脑可以识别运行的指令集合。大部分的编程语言都是基于文本为主的，这就意味着你要敲击输入很多英文字母。（编写程序最初的是利用阿拉伯数学来完成的，字符主要是使用 0 与 1,而且 0 才是开头，是第一个符号，这方面我们普遍存在误解）

为什么程序编写不建议使用中文呢？

<pre>1 #include<stdio.h> 2 int main() 3 { 4 int a; 5 scanf("%d",&a); 6 if(a==0) 7 { 8 printf("你好"); 9 } 10 else 11 { 12 printf("你不好"); 13 } 14 return 0; 15 }</pre>	<pre>1 #引用<头文件1.h> 2 3 加载 主函数 () 4 { 5 定义整数变量 a; 6 常规格式输入 ("%d",&a); 7 如果 (a==1) 8 { 9 常规打印输出 ("你好\n"); 10 } 11 否则执行 12 { 13 常规打印 ("你不好\n"); 14 } 15 返回值 0; 16 }</pre>
--	--

中文与英文程序编辑的对比

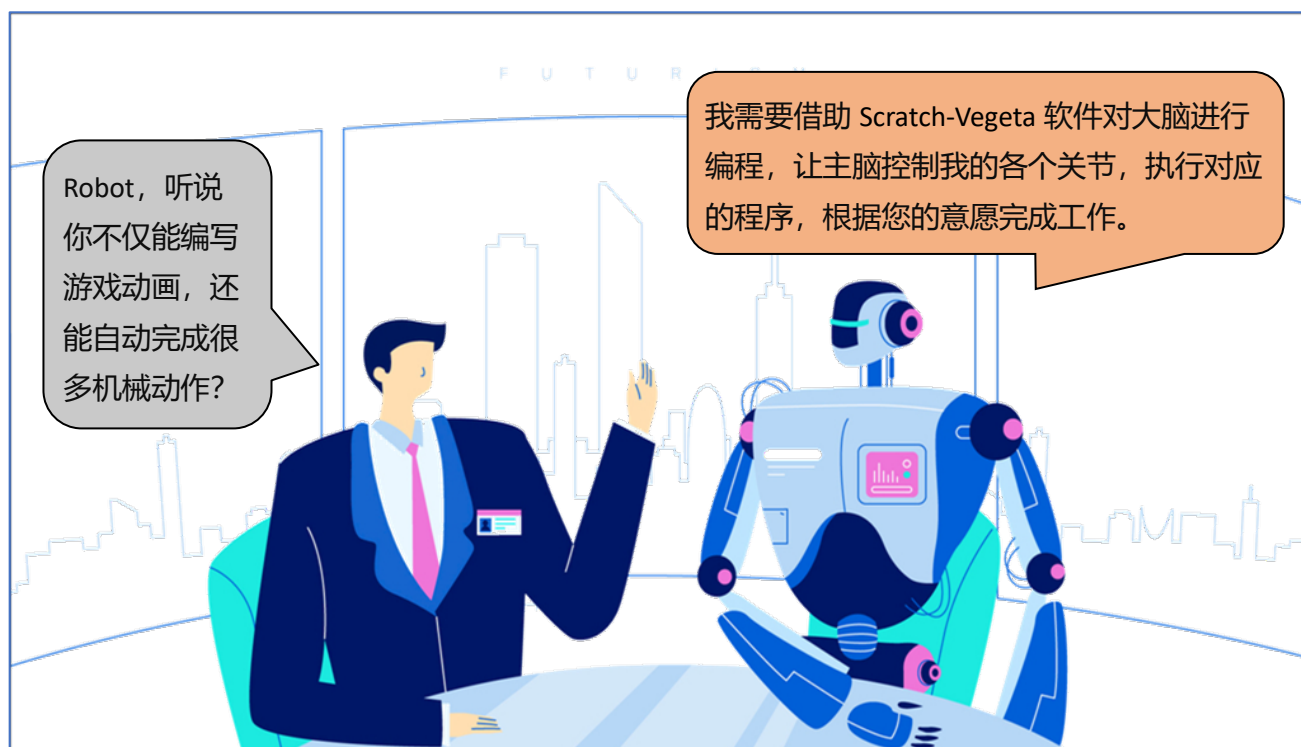
- 中文打字效率太低，任何一个字最少也要两次击键，效率上不如英文。
- 中文符号不规范，存在全角和半角的问题。
- 大部分编程语言在一开始就是以英语作为母语的（交流的需要）
- 英语能够被所有的操作系统和编译器支持，而操作系统与编译器要想支持中文，则需要付出额外的努力。
- 很多亚洲语言都属于模糊的语言，即语法中存在太多不确定因素，所以天生不适合作为编程语言使用。（例如冬天“有多少穿多少”和“夏天有多少穿多少”）
- 最后，其实中文程序语言并不比英文的好学，而且可读性也比不上英文程序。

对于绝大多数初学者来说，学习编程语言并了解对应的语法规则是非常困难的，因此图形化编程应运而生，设计初衷就是更容易让孩子们去了解并学习编程，在“瞎搞”的过程中让学习的过程更加丰富有趣。

第一课 进入 Vegeta 的世界



1. 认识 Scratch-vegeta 的主界面
2. 掌握为角色编写脚本的操作方法
3. 会使用  控制脚本的运行
4. 实现角色在舞台上来回移动



Scratch 程序由一些“角色”组成(原文为 Sprites)。你可以通过改变角色的造型来改变它的外观。可以让它变成一个“人”，或者变成一辆“火车”，或是一只“蝴蝶”，亦或者是其它的任何东西。你可以使用任何图片作为造型：可以在绘图编辑器中编辑、可以导，也可以从网上下载。尝试给角色下一些命令同其它角色进行交互。

一、认识 Vegeta

大家好，欢迎来到 Vegeta 的世界。

Scratch 是由美国麻省理工学院终生幼儿园团队设计开发的一种图形化编程工具，利用它可以轻松完成舞蹈、音乐、交互式故事、游戏、艺术创作、模拟实验等作品。Scratch 的特点是积木式编程，它能让编程像搭建积木一样简单，只需要通过拖拽和组合的方式操作 Scratch 中的积木块，就能完成各种各样的程序，程序运行的结果在电脑屏幕上会立即显示出来。

Vegeta 是在 Scratch 的基础上，增加了电子积木的支持。电子积木是指将各种电子元器件组合在一起，并且制作成积木的形式，结构上和塑料积木件兼容，便于结合塑料积木进行作品搭建。Vegeta 软件在继承了 Scratch 软件特点的基础上，将电子积木的操作转化为编程积木块的方式，从而将现实作品和屏幕中的动画效果等进行结合。

如图 1-1 所示为利用 Vegeta 软件制作的各种各样的作品，你喜欢吗？学习完基础教程，你就可以尝试设计作品从而实现自己的创意了！



图 1-1 用 Vegeta 完成的作品

下载方式：使用百度网盘下载链接（产品说明书中扫码下载）。

二、Vegeta 软件界面

Vegeta 软件界面主要分为：舞台区、角色区、模块区、脚本区、角色信息区等，

如图 1-2 所示：

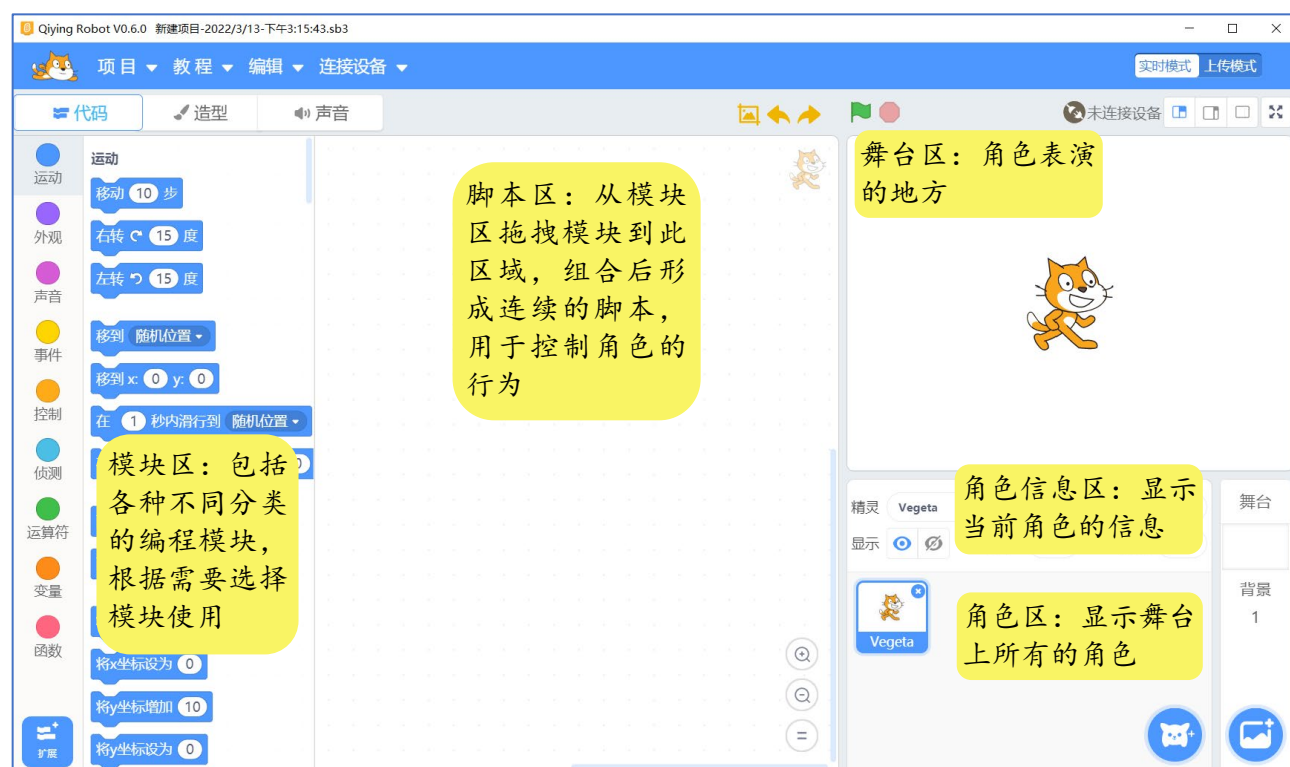


图 1-2 Vegeta 软件主界面

三、如何编写脚本和运行脚本

在打开 Vegeta 软件后，舞台区和角色区默认会显示一只小猴子，能不能让这个
小猴子动起来呢？接下来，我们学习脚本的编写，以及如何运行脚本。



搭建小猴子“vegeta”在舞台上运动的脚本。

1. 实现“小猴子移动脚步”的脚本，具体操作如图 1-3 所示：

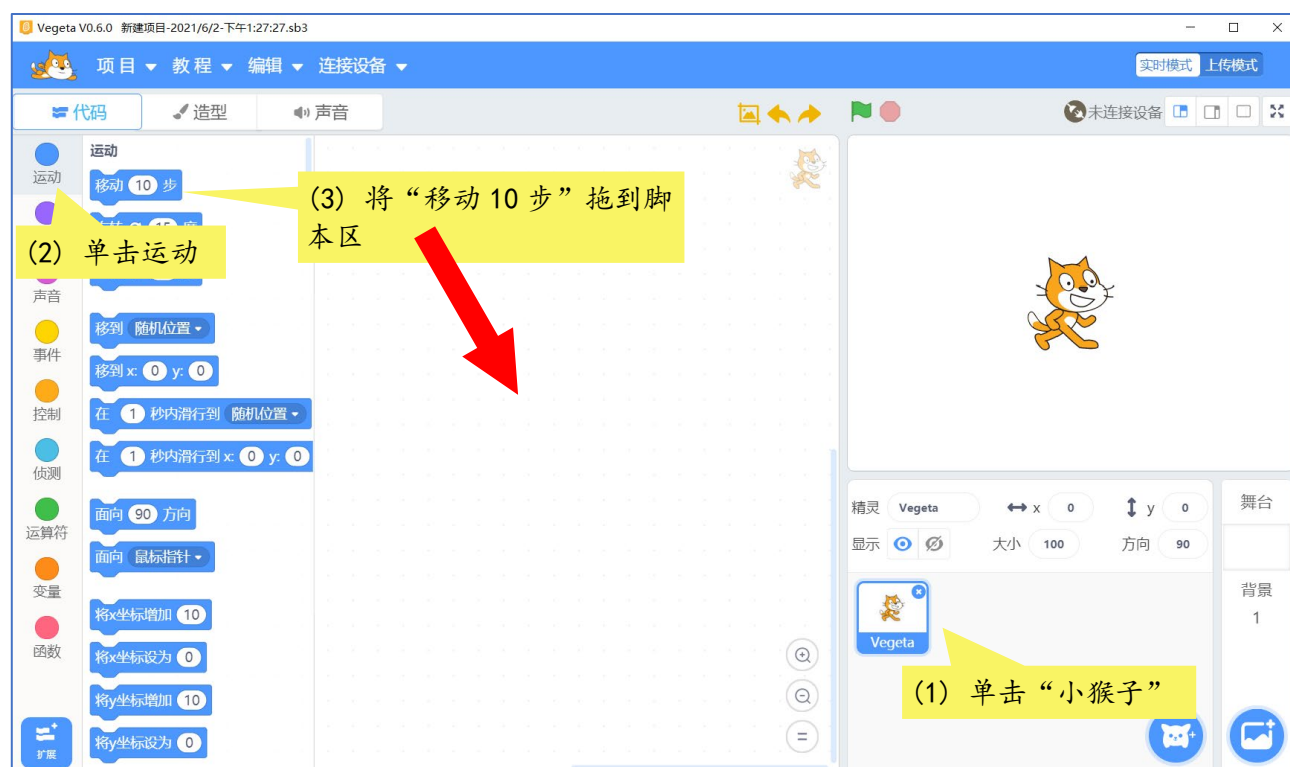


图 1-3 为角色编写脚本

通过上面的操作，你应该掌握了如何从模块区拖拽脚本到脚本区。用鼠标点击脚本区的 **移动 10 步** 模块，你会发现每点击一次，小猴子都会移动一定的距离。



移动 10 步 模块中的数字 10 是一个输入框，可以进行修改，我们称之为“模块的参数”。尝试修改 **移动 10 步** 的参数，再用鼠标点击，看看小猴子移动的效果有什么变化。

小贴士

请自行尝试 **移到 x: 0 y: 0**（将参数全部设置为 0）的功能，如果多次点击

移动 10 步 模块，小猴子可能会“走出舞台”区域，使用该模块可以使得小猴子重新回到舞台中心。



2. 实现“小猴子走过来”的脚本

要实现“小猴子走过来”，需要多个模块进行组合，在 Vegeta 中，脚本的编写主要就是选择合适的功能模块，并将这些模块进行组合。按照图 1-4 的操作，实现“等待 1 秒后小猴子移动 10 步”：

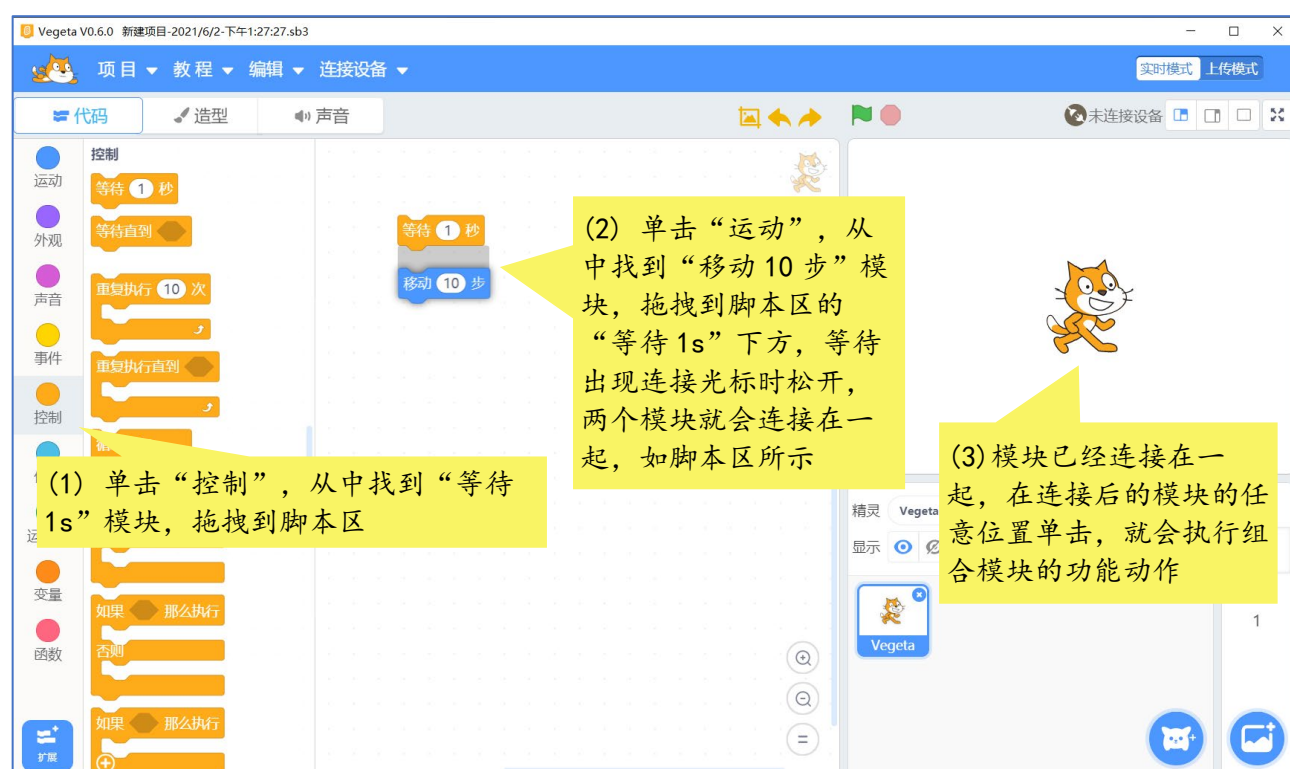


图 1-4 多个模块的组合

完成后，在组合的模块上点击，脚本块就会执行，点击之后等待 1 秒，小猴子才会移动。

编写“小猴子动起来”的完整脚本：

脚本功能	脚本实现	操作说明
让小猴子动起来		从“运动”分类中选择“移动 10 步”拖拽到脚本区
重复执行		从“控制”分类中选择“循环执行”拖拽到“移动 10 步”的上方，“重复执行”就会自动张开，将“移动 10 步”模块包含进去 提示：也可以将“重复执行”先拖到脚本区空白区域放下，将“移动 10 步”模块拖拽到“重复执行”中。
设置小猴子初始的位置为舞台中心		从“运动”分类中选择“移到 x: 0 y: 0”拖到脚本区，放在循环执行的上面（注意：设置参数全部为 0）。
控制脚本的运行		从“事件”分类中选择“当 被点击”拖到脚本的最上面。

3. 控制脚本程序的开始、结束和舞台大小。如图 1-5 所示，单击舞台上方的按钮可以运行脚本，单击按钮可以停止脚本的运行，单击按钮可以将舞台全屏显示，单击按钮可以切换舞台的大小。



图 1-5 舞台控制按钮

不断练习模块的拖拽和组合操作，熟练掌握脚本的编写，为后面的学习打下坚实的基础。在脚本区，用鼠标尝试在已经组合的模块上的不同区域按下并拖拽，观察有什么效果，你能学会如何拆散已经组合的模块吗？

四、碰到边缘就反弹

当我们点击  运行脚本时，会发现小猴子走了一段时间后会走出舞台，那么，如果希望小猴子能在舞台上来回走动，如何实现呢？



1. 在“运动”分类中，找到模块 **碰到边缘就反弹**，利用它编写如下脚本：

脚本功能	脚本实现	操作说明
小猴子在舞台来回运动		将“碰到边缘就反弹”拖拽到“移动 10 步”的下方

2. 在角色信息区，点击“只能水平旋转”设置角色的旋转模式：



图 1-6 设置角色旋转类型

点击小绿旗运行脚本，小猴子在舞台上来回走动了。

小贴士

模块分类区会根据模块的功能，对所有的模块进行分类。根据脚本中模块的颜色，能快速的从模块分类区中找到对应的模块，如下图所示，“等待 1 秒”这个模块是黄色的，那么它应该在“控制”中。

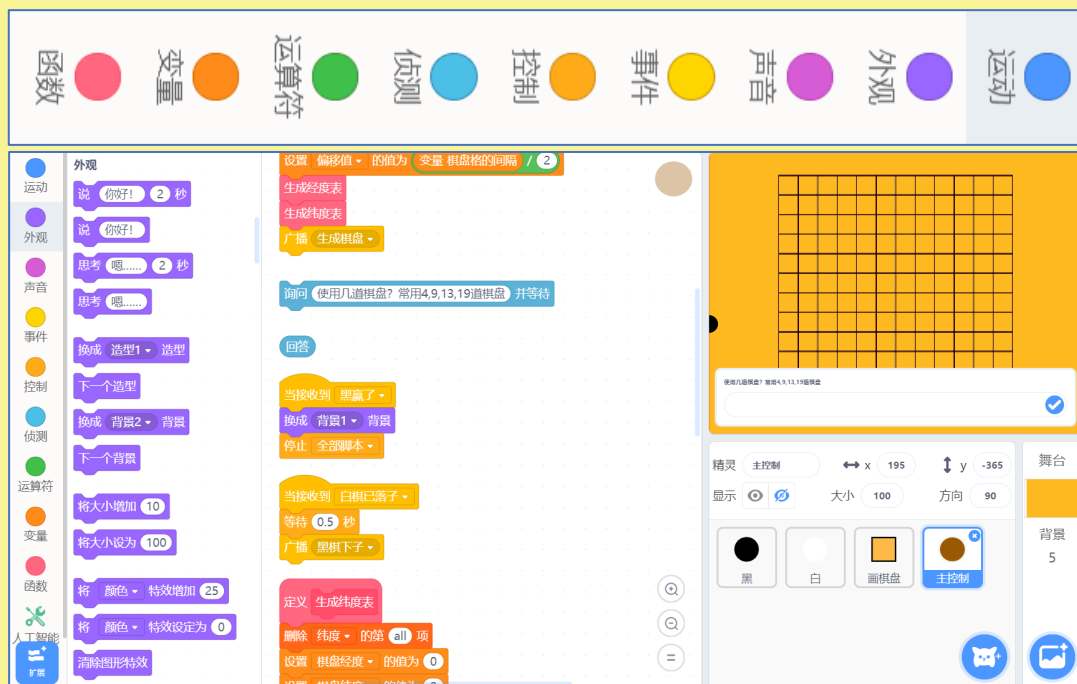


图 1-7 模块颜色和模块类型的颜色对应关系



1. 回顾前面的内容，小组讨论 Vegeta 编写脚本的过程。
2. 在“控制”中找到等待 1 秒模块，将其插入到碰到边缘就反弹的后面，修改参数的值分别为 0.1, 0.5, 1, 2 等，观察脚本运行的效果，小组讨论交流对该模块的理解。



在我们学习的新知识领域中，如何将知识点穿插理解后，成为自己的东西，这点是非常重要的。每个人都好学，喜欢模仿，从出生就不断接受知识，然后根据自己的想法去拼凑，连成一个整体。将繁杂的知识点贯穿成互相连接的线，这点不仅需要经验，更需要条理性的思维方式。当接触到积木时，就想着如何去拼凑好，这个新概念在你头脑中呈现的状态如下图。搭积木式组合模块化，能让我们将头脑中的信息组块，按照有意义的方式形成一个逻辑性、概念性的视野，帮助小孩更好地应用知识进行创作，让其呈现出另一种样子。



思考：如果是你第一次接触、你会怎么做？

善于钻研的你一定可以做到这样。

当这个新知识经过理解、练习后，它就会形成一个新组块，存入我们的长期记忆，参与到下一次的组块化中。

Scratch 中的积木语言模块共计有四种形状：代表不同种类的积木类型：命令积木、功能积木、触发积木、控制积木。

命令积木：主要执行某一特定指令，特点“上凹下凸”+可以对其进行编辑并单机

可执行；

等待 1 秒

功能积木：只要执行某一功能的指令命令，特点得出布尔值,即“真和假”（是和不是）；true 也就真，代表着满足了条件内容。false 也就是假，代表着条件没有满

足。

碰到 鼠标指针 ▾ ?

触发积木：特点某一事件发生了，这一积木开始运行+形状特征是上有圆弧，下

凸起连接其他积木；

当作为克隆体启动时

控制积木（堆栈积木）：特点控制某一特定指令，控制运行流程+控制执行中间

指令。

循环执行

控制积木和命令积木两者上方均有一个缺口，下方通常有凸起，这些缺口和凸起可卡扣在一起形成更长的脚本；触发积木（又名帽子）上方是圆形的且无缺口，代表着它总是处于一段脚本的起始位置。一旦事件触发则执行下方脚本。功能积木没有缺口，仅仅得到一个数值。还有一些功能模块的前面有复选框，舞台内会出现值的显示器，会持续地显示当前功能模块的值。

第二课 飞舞的蝴蝶



1. 学会舞台背景、角色、造型的导入
2. 掌握造型切换和动画制作的基本原理
3. 了解舞台、角色和造型的关系
4. 学会多个角色的操作和脚本的复制



读完本书的第 1 课，你一定对一些新鲜的术语和名词应接不暇了。什么是“角色”？什么又是“造型”？那只可爱的“小猫”是干什么用的？本节将一一解开你的这些疑惑。别着急，我们一个一个来介绍吧！

制作动画是 Vegeta 中最基本的功能，在本节课，我们将结合上节课的知识，学习如何制作一个更加逼真的动画，体验 Vegeta 的乐趣。

一、导入新背景

电影中的人物角色出现的时候，往往会有不同的场景。在实时模式中，背景就像是电影中场景。当角色在舞台上出现的时候，背景是衬托在最底层的图像式场景。我们可以给舞台分配一个或多个背景，从而在应用程序的执行过程中改变舞台的外观。



1. 打开背景库

选中舞台，点击  进入造型面板，然后点击  “背景库”按钮，会弹出背景

库窗口，操作如图 2-1 所示：



图 2-1 打开背景库

2. 从背景库中选择背景

在主题中找到“奇幻”主题，从里面找到“castle2”这个背景，选中后点击“确认”按钮，操作如图 2-2 所示：



图 2-2 从背景库中选择背景

默认情况下，所有的 Scratch3.0 应用程序都会分配到一个空白的背景。可以点击位于编辑器右下方的“舞台”工作区的“背景”动态图标，从弹出的 4 个图标中选取，来添加一个新的背景。

导入成功后，观察舞台区的变化。

二、添加删除角色

在电影或者电视剧里面总是有很多的角色。在 Scratch3.0 中，角色（Sprite）就好比电影或电视剧里的演员所扮演的角色。我们所编写的程序，总是要通过角色来做出动作，发出声音，或者完成一项任务。

这就好像再好的剧本也要由演员来表演一样。Scratch3.0 中有一个默认的角色，就是一只可爱的“小猫”。我们当然可让这只小猫做很多的事情，例如，移动，发出“喵喵”的声音，执行动画，等等。

但是，很多时候，在我们编写的程序中，都需要加入自己的角色来做特定的事情，这时候就要忍痛删除“小猫”这个角色了。



1. 删除默认角色

在角色“贝吉塔”上，单击鼠标右键，点击“删除”，就可以将小猴子角色从舞台上删除掉，操作如图 2-3 所示：



图 2-3 角色的删除

2. 从角色库导入角色

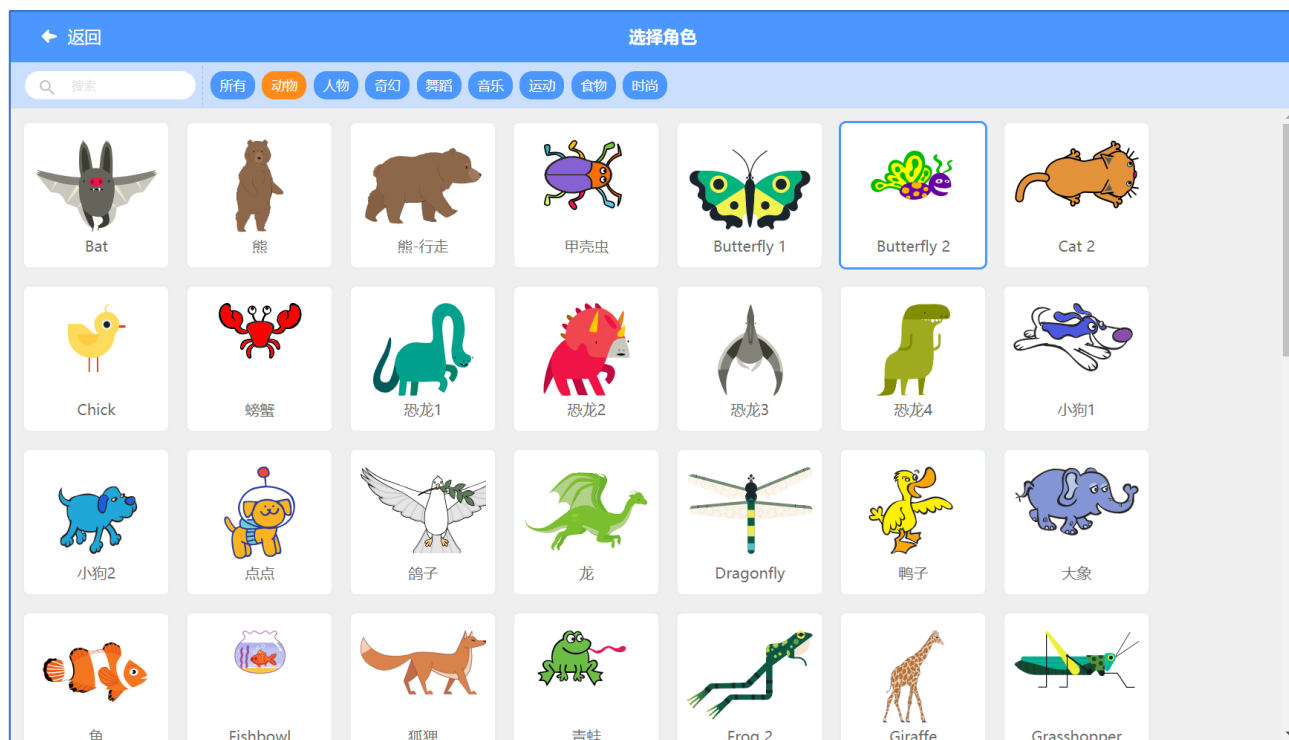
点击角色区的  按钮，在弹出的窗口中单击“ 角色库 

出角色库窗口，具体操作如图 2-4 所示：



图 2-4 导入角色

和背景库中选择背景的操作一样，找到“动物”分类下的“Butterfly2”角色后确认并导入，如图 2-5 所示：



在造型区域中，出现两张图片，我们让蝴蝶角色在两个造型之间切换，从而实现飞舞的动作和动画。

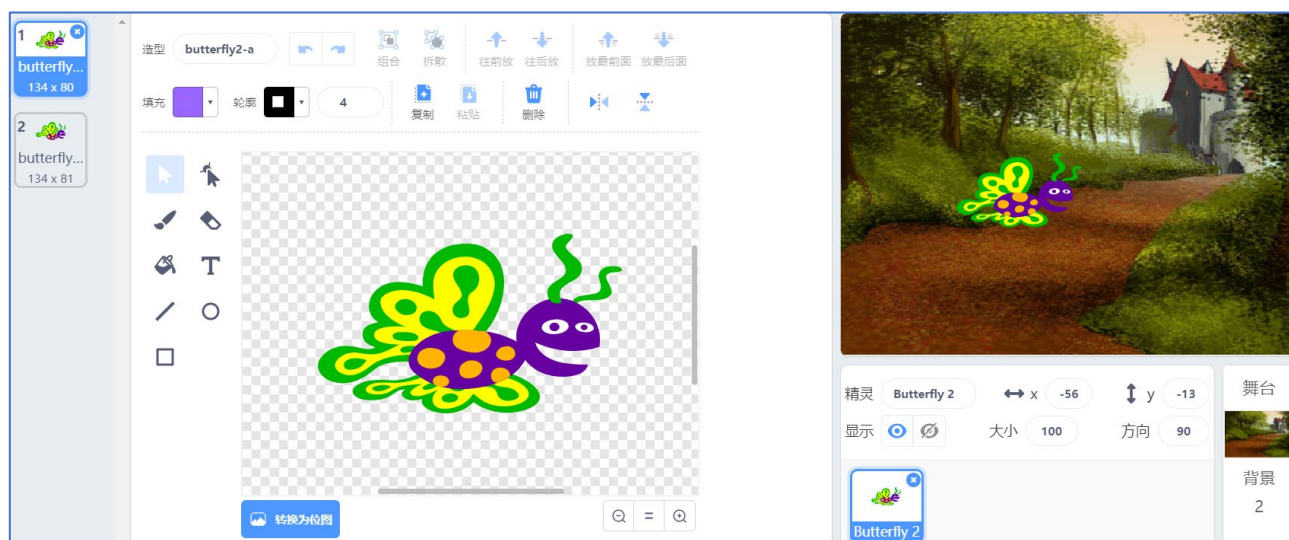


图 2-5 导入 Butterfly2 角色

三、造型切换实现动画



1. 造型的切换

切换到代码区  **代码**，将“外观”分类模块中的 **下一个造型** 拖拽到脚本区，不停的点击该模块，观察舞台上的变化，小蝴蝶是不是开始舞动翅膀了？

小贴士

动画的制作就是通过切换角色的造型来完成的，只需要重复不停的切换造型，就能完成连续的动画效果了。

2. 编写脚本实现飞舞的蝴蝶

按照下面的过程，结合其他模块，完成蝴蝶在舞台上来回飞舞的效果。

脚本功能	脚本实现	操作说明
让蝴蝶在舞台上来回运动		从“控制”分类中选择“当  被点击”和“重复执行”，从“运动”分类中选择“将旋转方式设为左右翻转”“移动 10 步”和“碰到边缘就反弹”组合成脚本。

切换造型		从“外观”分类中选择“下一个造型”拖拽到“碰到边缘就反弹”的下方。
每次切换造型后等待一段时间，避免动画过快		从“控制”分类中找到“等待 1 秒”，拖拽到“下一个造型”的下方，将参数设置为 0.1

点击  运行脚本，蝴蝶将会在舞台来回飞舞，一个逼真的动画就完成了。

四、多个角色和脚本的复制

在舞台上，只有蝴蝶飞舞是不是很孤单呢？能不能在舞台上添加更多动画。接下来，我们添加一个“小狗”角色，并且让小狗也在舞台上来回运动。



1. 导入小狗角色。按照导入蝴蝶角色一样的操作，在角色库中的“动物”分类中找到“点点”角色，导入后，在角色区会出现两个角色，如图 2-6 所示：

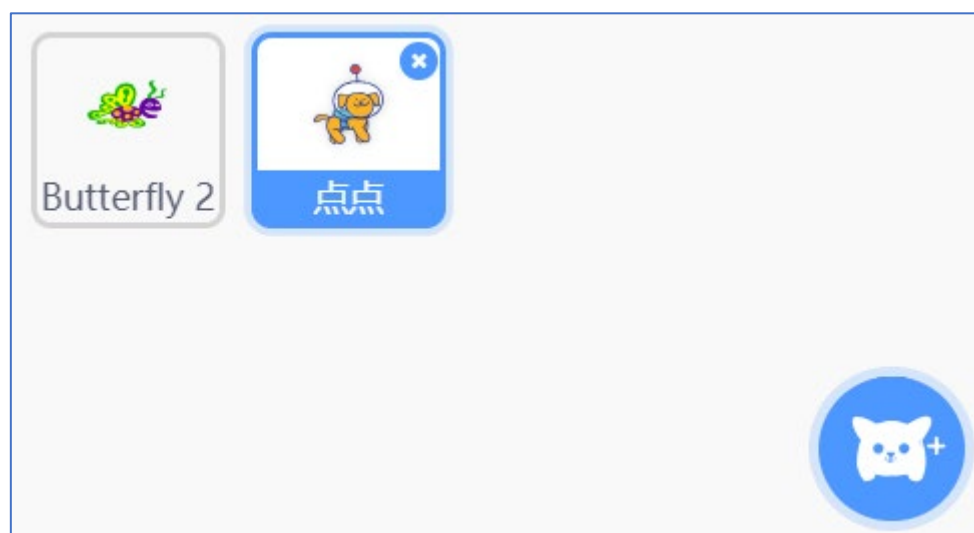


图 2-6 角色区出现 Butterfly2 和点点两个角色

2. 脚本的复制

在 Vegeta 中，每一个角色都拥有控制自己的脚本程序。但是，如果想让不同的角色实现相同或相似的效果，我们可以通过复制脚本程序来简化操作过程。将蝴蝶的脚本复制给小狗的操作如图 2-7 所示：



图 2-7 脚本在角色间的复制

脚本复制完成后，点击  观察运行结果，蝴蝶在舞台上飞舞，小狗在舞台上奔跑。



思考一下，为什么小狗能奔跑呢？切换到小猫角色的脚本和造型面板，看看有什么发现吧。

小贴士

在 Scratch 中，舞台、角色、脚本和造型的关系非常重要。它们的关系如同一场舞台剧，一场舞台剧只有一个舞台，舞台上会有多个不同的演员（相当于“角色”）进行表演，每个角色有自己的剧本动作（相当于“脚本”），表演过程中，角色会根据需要，切换它的服饰（相当于“造型”）。一句话总结为：一个舞台有多个角色，每个角色有自己的多个脚本块和多个造型。

五、工程的新建、保存和打开

在 Vegeta 顶部的“项目”菜单中，有“新建”、“打开”“最近编辑”和“保存”、“另存”五个选项。在每次完成作品后，一定要记得保存项目，以便于能够与他人分享，以及在下一次打开时修改程序。如果需要重新开始编写作品，那么可以点击“新建”菜单。

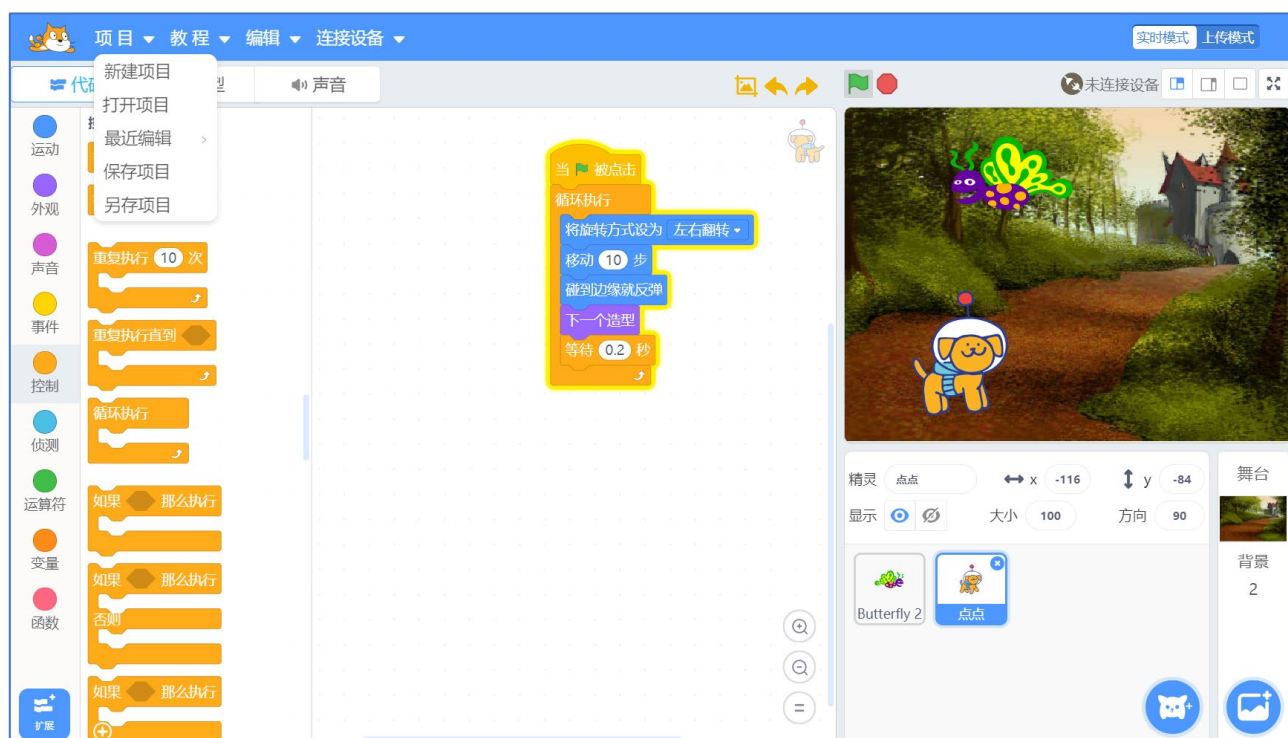


图 2-8 工程相关菜单



1. 在角色的造型面板中，能否找到“上传角色”的按钮，尝试打开造型库，掌握造型库的操作。
2. 在造型面板中，通过选择“绘制”选项可以打开画图板，来创造自己的动画造型吧！

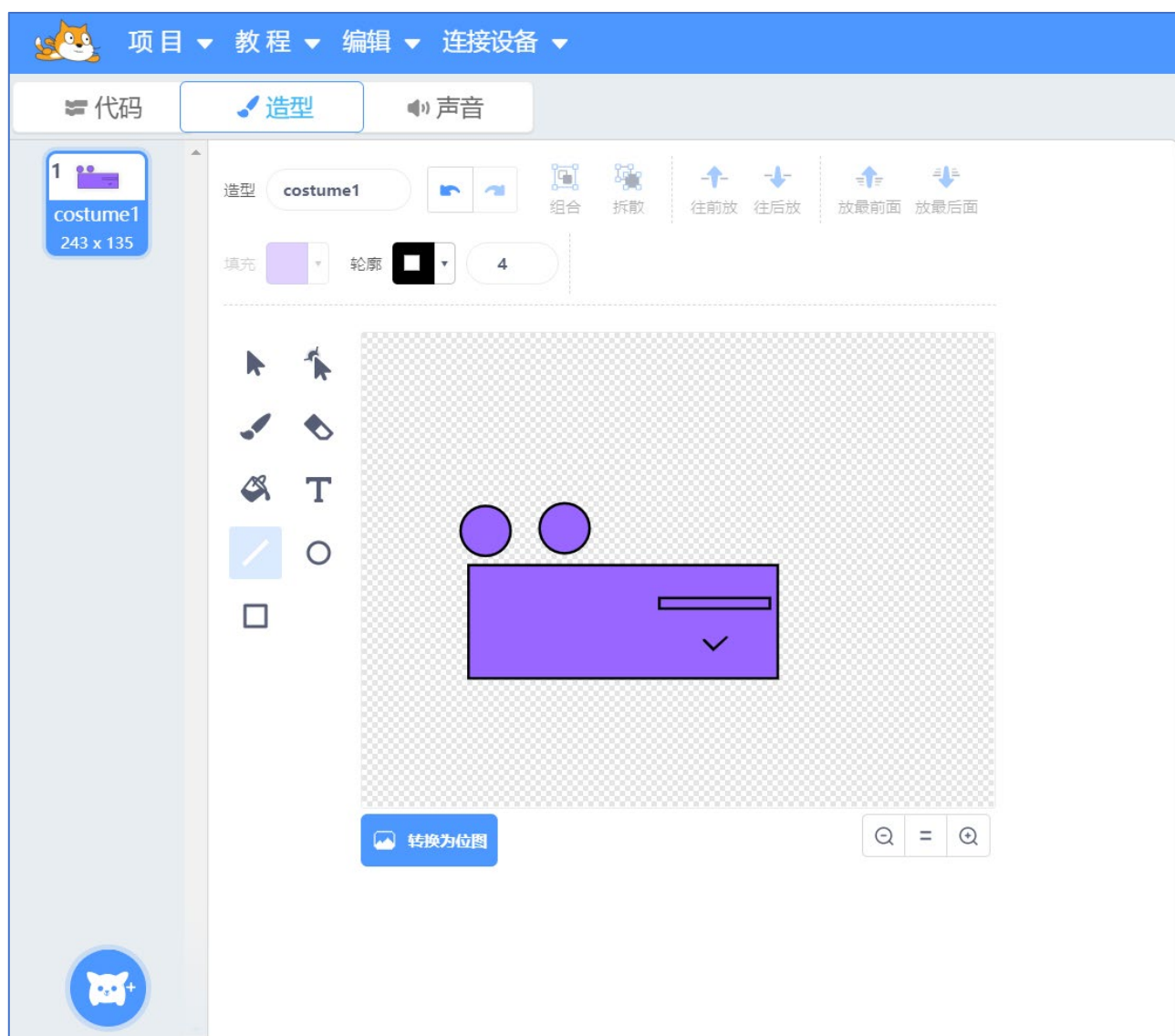


图 2-9 画图板界面

3. 在造型面板中，通过选择“上传造型”选项，可以从本地上传素材图片作为角色的造型，赶紧试试吧！

第三课 神奇的魔法师



本节课我们将详细学习外观和声音模块，目标是：

- 创建动画和图形特效
- 学习图形化编程中较难的内容
- 学习声音文件并使用音乐蜂鸣器演奏音乐
- 制作完整的动画场景

外观模块可以创建动画效果，还能给角色造型、背景增添各类特效，如鬼影、亮度、漫画效果等；声音模块可以给程序脚本增添更有趣的互动形式，让我们从创建动画开始吧！

1. 巩固导入背景角色等操作，能灵活运用
2. 学会并理解“说”和“思考”模块
3. 学会使用“外观”中的模块控制角色的特效和大小
4. 学会使用 **等待 1 秒** 模块控制两个角色之间的脚本的同步工作





魔法师是在民间传说、幻想文学、桌面和电子游戏中经常被提及的角色，指可以驱使某种神秘力量的人群。魔法师就是那些能够变魔法的人，不论这种魔法是邓布利多教授在魔法学校教授的那些，还是乡村巫师念的咒语那样的小把戏，或者是舞台逃脱魔术表演大师表演的那种魔法。在世界的各种文化中都存在这样的神奇魔法师。



本节课则带领孩子们走进 Vegeta 的魔法世界，在这个世界里有不同的角色和千变万化的造型。角色除了造型之外，还可以使用更多的“外观”模块，比如控制角色的大小等，利用这些模块，能制作更复杂的动画效果。在这节课，我们将学习这些神奇的模块，一起来看看神奇的魔法师有什么本领吧！



图 3-1 变魔术



一、角色会说话

1. 从“外观”分类中找到 **说 你好!** **2 秒**，将其拖拽到脚本区后，点击运行，看看舞台上有什么效果吧。小猴子是不是开始说话了，两秒后，文字消失。



图 3-2 角色说话

2. 从“外观”分类中找到**说 你好!**模块拖拽到脚本区，点击运行，小猴子就会一直说“你好”了。如果要让文字消失，只需要将**说 你好!**的参数删除后变为空白就可以了。

你能不能用**说 你好!**和**等待 1 秒**模块组合后，实现**说 你好! 2 秒**的功能呢？

从“外观”分类中拖拽**思考 嗯... 2 秒**模块到脚本区，点击运行，看看舞台上的效果，分析一下，“思考”和“说”模块的区别。

小贴士

如果将角色比喻成舞台剧的演员，那么“说”和“思考”模块相当于角色的嘴巴，后面的参数相当于台词。所以，“说”和“思考”的文字框同一时间只会在舞台上出现一个。在执行下面的脚本时，舞台上会先显示“你好!”，等待一秒后，再显示“嗯...”。



图 3-3 说和思考

二、角色的特效

接下来，我们来了解角色的特效，在开始之前，先将前面的工程保存后，新建一个空白工程进行学习。

1. 将舞台绘制为红色的纯色背景

为了更好的理解特效，我们给舞台添加一个纯色的背景，方便进行对比。使用画板绘制舞台背景造型的操作如图 3-4 所示：

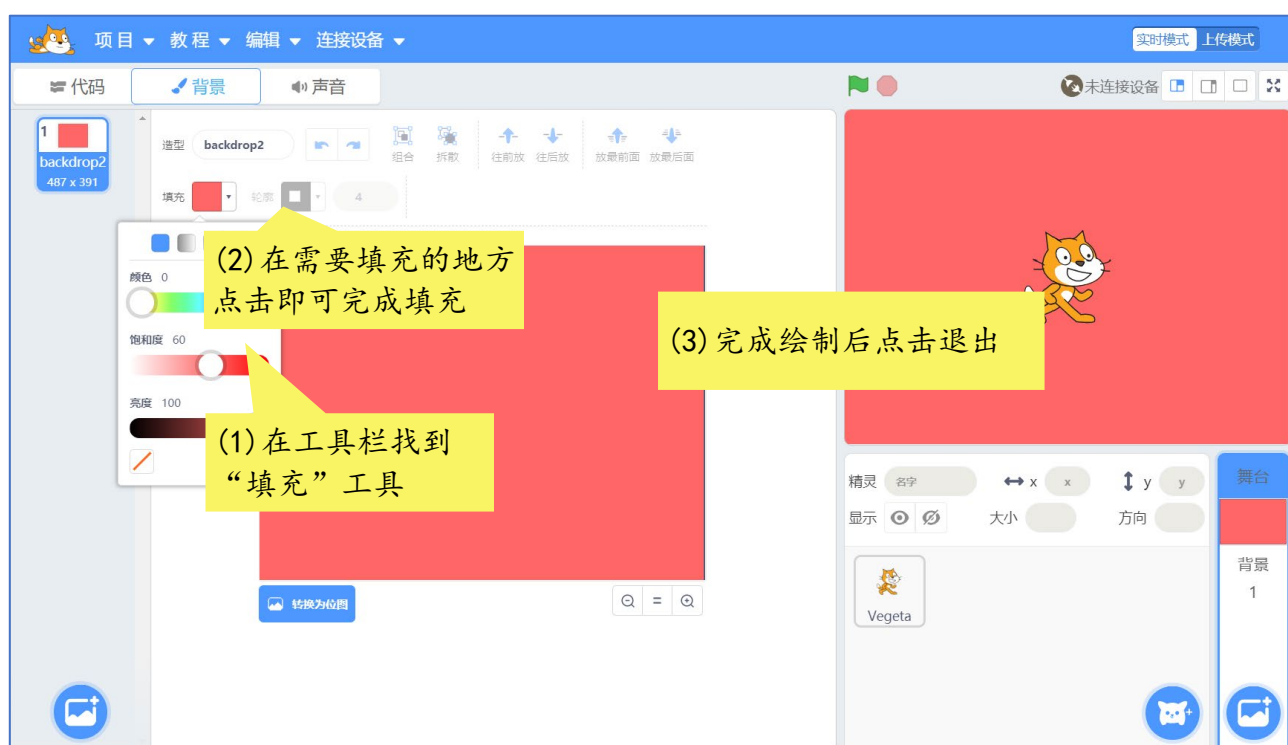


图 3-4 使用画板填充舞台背景

2. 为小猴子角色编写特效脚本

脚本功能	脚本实现	操作说明
设置图形的特效增加一定的值		从“外观”分类中找到“将虚像特效增加 25”拖拽到脚本区
每次设置后等待 1s 来观察效果		将“等待 1 秒”放到特效增加模块的下面
重复四次，观察每一次增加的效果		将“重复执行 10 次”嵌套在写好的脚本上，设置参数为 4 次
每次开始时清除特效		将“当 被点击”和“清除图形特效”拼接在脚本块上方

3.脚本编写完成后，点击小绿旗，测试“虚像”特效的功能，舞台上角色的变化如

图 1-5 所示：

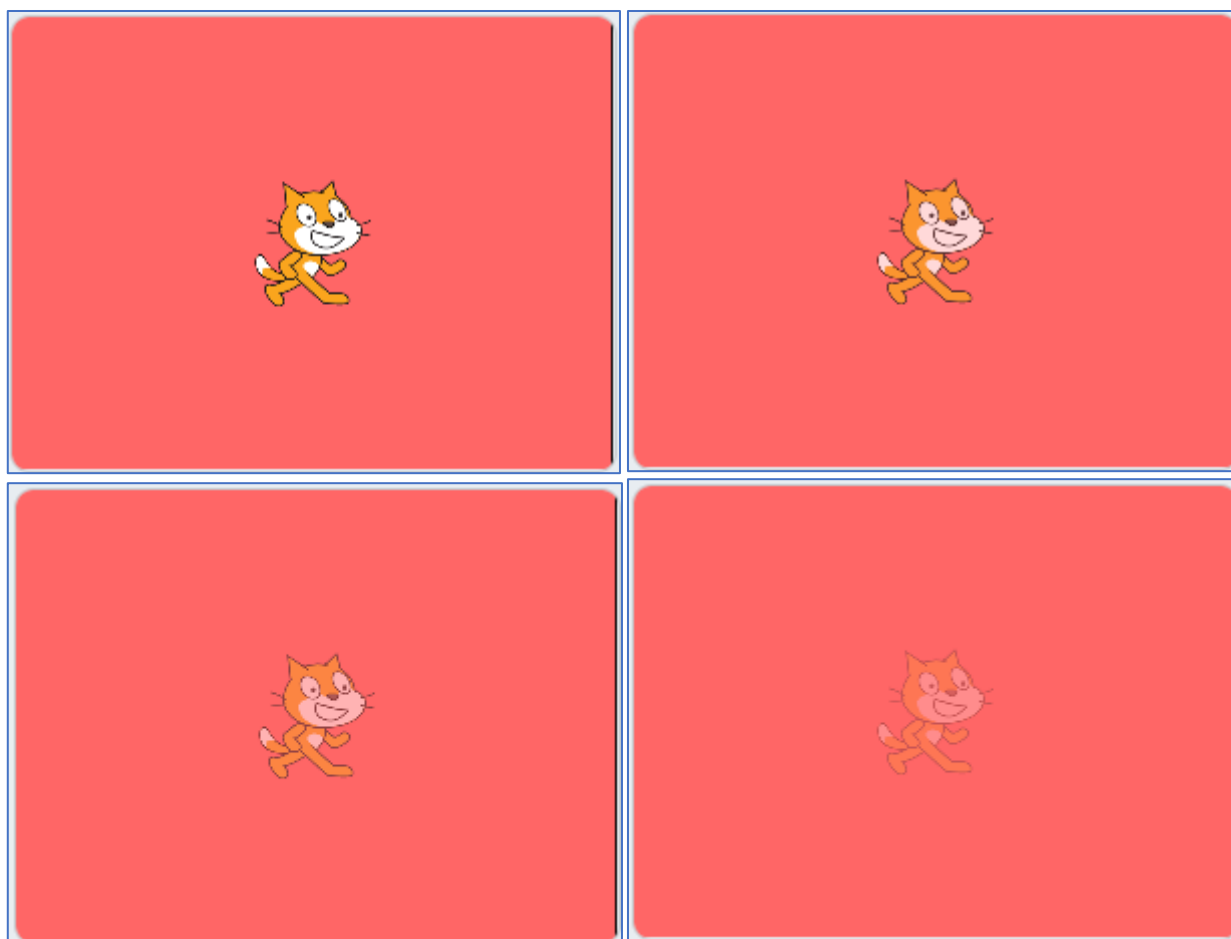


图 3-5 鬼影特效的变化过程

4. 尝试修改“将虚像特效增加 25”的第一个参数，分别尝试颜色、鱼眼、漩涡、像素化、马赛克、亮度等多种特效的效果。如图：



图 3-6 不同的图形特效

三、角色的大小、显示和隐藏

接下来，我们来了解角色的大小、显示和隐藏，在开始之前，先将前面的工程保存后，新建一个空白工程进行学习。下面，编写一个脚本，实现小猴子闪烁并变大的效果。搭建脚本如图 3-7 所示：



图 3-7 角色闪烁并变大

四、神奇魔法师的脚本编写

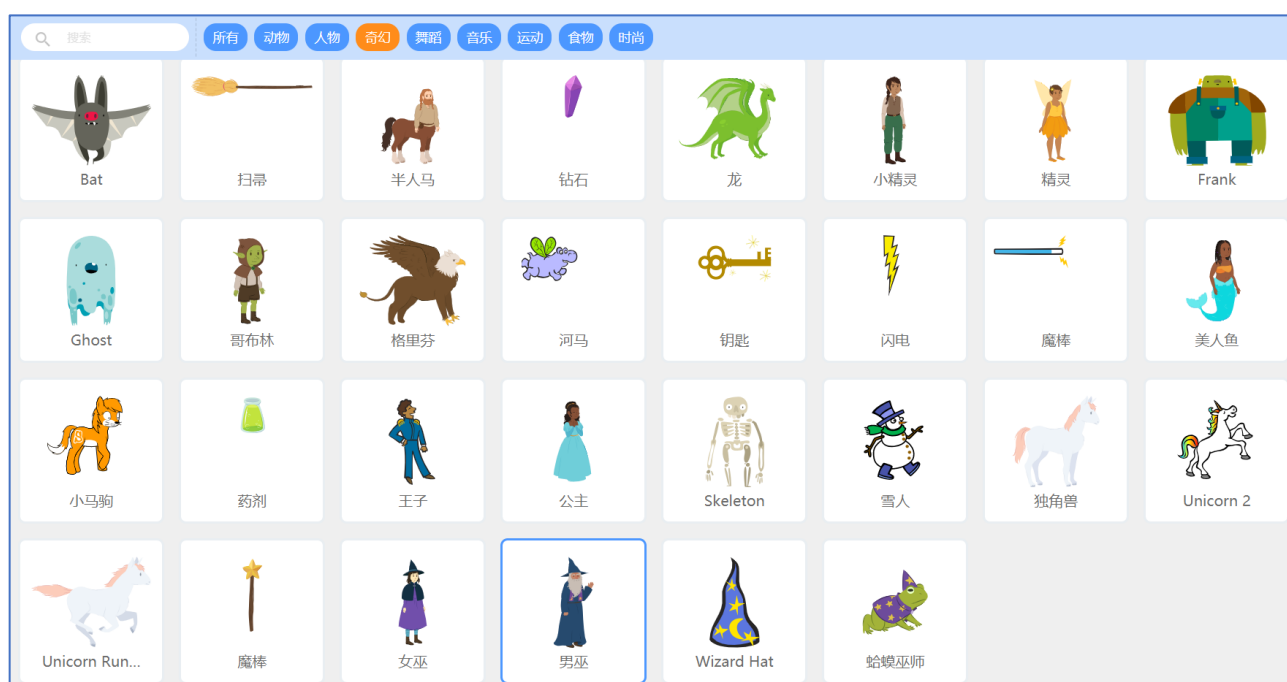
在掌握了外观中的模块的功能后，我们将结合前面学到的内容，编写一个完整的程序，制作一个“神奇的魔法师”，魔法师会念出一些咒语来控制一条龙完成各种各样的动作；当魔法师说“变大”的时候，龙就会变大，当魔法师说“变小”的时候，龙就会变小；当魔法师说“变色”的时候，龙就会变得五彩缤纷；当魔法师说“喷火”的时候，龙就会喷火，魔术表演结束后，龙就会隐藏，魔法师说“谢谢大家！”整理一下，整个过程如下表所示：

魔法师咒语	持续时间	龙的动作
大家好！	2 秒	等待 4 秒
接下来，看看我神奇的魔法吧！	2 秒	
变大！	2 秒	变大，等待 2 秒
变小！	2 秒	变小，等待 2 秒
变色！	2 秒	设置五彩纸屑特效，等待 2 秒，恢复特效
接下来，睁大眼睛看！	2 秒	等待 2 秒
喷火！	3 秒	切换造型实现喷火动画，喷火 3 秒
谢谢大家！	2 秒	隐藏

1. 创建一个新的工程，并删除小猴子默认角色
2. 从舞台背景库的“音乐”主题中找到“舞台聚光灯”背景，导入



3. 从角色库的“奇幻”分类中找到“男巫”角色，导入该角色；从角色库的“奇幻”分类中找到“龙”角色，导入该角色



4. 在舞台上拖动魔法师和龙角色到合适的位置，并使用“将大小设为 100”改变为合适的大小，舞台的参考效果如图 3-8 所示：



图 3-8 舞台的效果

5 编写角色脚本

(1) 男巫角色的脚本

角色：（男巫）		
脚本功能	脚本实现	操作说明

设置说话的内容和时间	 <pre>当绿旗被点击 移到 x: -138 y: -5 将大小设为 70 换成 男巫-a 造型 说 大家好! 2 秒 说 接下来请欣赏我的神奇魔法! 2 秒 换成 男巫-b 造型 说 变大! 2 秒 换成 男巫-a 造型 等待 0.1 秒 换成 男巫-b 造型 说 变小! 2 秒 换成 男巫-a 造型 等待 0.1 秒 换成 男巫-b 造型 说 变色! 2 秒 换成 男巫-a 造型 说 眼睛睁大仔细看! 2 秒 换成 男巫-c 造型 说 喷火术! 2 秒 换成 男巫-a 造型 说 感谢观看! 2 秒</pre>	从“外观”中选择  模块，编写相应的脚本，将脚本组合到小绿旗下方
-------------------	---	---

(2)Dragon 角色的脚本

角色:



(龙)

脚本功能	脚本实现	操作说明
启动脚本，等待咒语		拖拽相关模块，修改参数
变大并等待		拖拽角色大小增加的模块，设置参数
变小并等待		拖拽角色大小增加的模块，设置参数-80，表示变小
变色 2 秒后恢复，等待咒语		拖拽特效增加的模块，设置参数为“颜色”，等待 2.1 秒后使用“清除特效”模块清除
喷火动画，造型切换 3 秒后隐藏		使用 下一个造型 、 重复执行 10 和 等待 1 秒 实现一段时间的喷火动画，总时间的计算为 6 乘以 0.5 秒，完成后隐藏角色
最终的脚本		按照前面的操作组合后，最终脚本完成



自己设计故事情节，设计自己的魔术交互故事。



教育是具有决定性的力量，不但会改变一个人的人生轨迹，也会改变世界的面貌。西方能成为世界的中心，绝对不是因为他们的军事力量，而是因为他们头脑和智慧的力量。事实上，过去的 200 年里，很多技术发明、创新和变革都是由西方发明并且走向全世界的。

一个革命性的教育变革，就是从线性的教育体制，转变为一个定制化的教育模式。如果新的教育系统鼓励创造性的天才找到自己的生产方式，这种教育系统对政府的需求，对行政命令的要求和依赖就会越来越少。这种教育系统应该鼓励人们自己找到问题的解决方案。

五、练习总结：

以上内容是后续所有课程的基础，我们不仅了解学习到编程软件 Vegeta 的简单操作与用法，也了解了界面中的各种模块，还学习到了算数的用法。下面来总结一下本章节的收获！

练习 1：算出乘法模块的值，并寻找相应的规律：

$$\begin{array}{ccccccc} 1 \times 1 & \underline{\hspace{1cm}} & 11 \times 11 & \underline{\hspace{1cm}} & 111 \times 111 & \underline{\hspace{1cm}} & \\ 1111 \times 1111 & \underline{\hspace{1cm}} & 11111 \times 11111 & \underline{\hspace{1cm}} & 111111 \times 111111 & \underline{\hspace{1cm}} & \end{array}$$

经过观察它们之间的规律是：_____

（乘数和被乘数有 n 个 1，乘积就以 1 为增量由 1 开始依次递增到 n ，再依次递减到 1，如 $1111111 \times 1111111 = 12345654321$ ）

练习题 2：将练习题中的 1 全部换成 9 得到的结果是多少？你能否根据结果找出规律呢？

（有几个 9 相乘，结果就有（几-1）个 9、一个 8、（几-1）个 0 和一个 1 按顺序构成）

现在你已经有了创建精密作品的的能力，在接下来不同章节的课程学习中将会更深入的学习相关知识，并开始学习电子积木传感器相关的知识，请拭目以待吧。

练习题 3：你能否运用绘画板功能，画出一个矩形+三角形？

练习题 4：这里有三句话。下面哪个模板可以正确地进行抽象，使这三个模板都与模板匹配相同的模式？

运用到了“光”，有些运用到了“乐”，你能不能查询资料了解到光速和声音的速度分别是多少呢？如何表达呢？

光的速度为：[光的传播速度约为每秒钟 300000 千米](#)

声音的速度为：[声音根据在不同的物质中的传播速度不同。声音的传播也与温度和阻力及密度不同都有关。会影响声音的传播速度。声音在空气中的传播速度是：空气（15℃）340m/s，空气（25℃）346m/s。](#)

练习题 8：计算 50、47、98 的平均数，结果用“说……积木”展示：

说 $(50 + 47 + 98) / 3$

练习题 9：已知 $x=3$ ， $y=5$ 。请尝试计算下列表达式的值：

a) $(131 + x) =$ _____ b) $(6 \times x + 7 \times y) =$ _____

c) $(y \times x) =$ _____ d) $(y + (12 / x \times y)) =$ _____

e) $(y / x \times y + (12 / x - 3)) =$ _____

第四课 迷宫游戏



1. 掌握 Scratch-vegeta 中坐标和方向的概念
2. 了解事件模块、命令模块、侦测模块的概念和区别
3. 学会条件判断语句的基本使用
4. 学会如何使用键盘控制角色
5. 学会结合更多模块自己编写小游戏



晚上写功课写累了，我就会玩玩小游戏。



玩游戏多没意思！自己写一个小游戏才厉害呢！试试闯迷宫吧~



走一走迷宫，看看会是怎样的结果呢？这里会聚了小朋友们喜爱的、非常独特的地图迷宫。沿着迷宫里的道路，兴致勃勃地寻找一下出口，这样就能使小朋友变得更加聪明。

走迷宫的简要方法：

第一条，进入迷宫后，可以任选一条道路往前走；

第二条，如果遇到走不通的死胡同，就马上返回，并在该路口做个记号；

第三条，如果遇到了叉路口，观察一下是否还有没有走过的通道。有，就任选一条通道往前走；没有，就顺着原路返回原来的叉路口，并做个记号。然后就重复第二条和第三条所说的走法，直到找到出口为止。如果要把迷宫所有地方都搜查到，还要加上一条，就是凡是没有做记号的通道都要走一遍。”本节课就一起动手制作一款迷宫游戏吧！

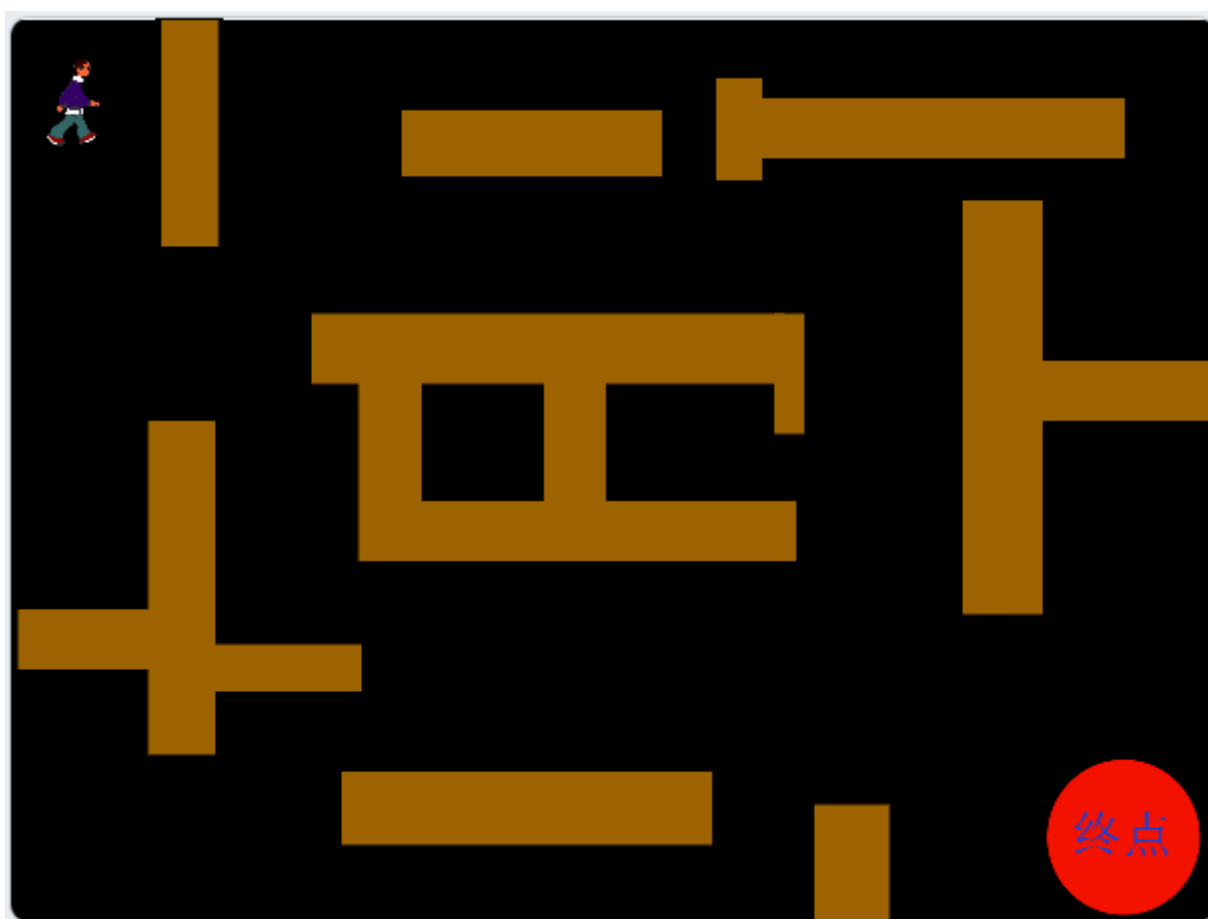


图 4-1 迷宫游戏



一、坐标和方向的理解

1. 角色的坐标

在“运动”分类中，有 **移到 x: 0 y: 0** 这个模块，前面的课程已经讲到，利用这个模块，可以将角色恢复到舞台中央。在这节课，我们将介绍舞台坐标系。

在 Vegeta 中，通过坐标来确定角色的位置，坐标包括 x 和 y 两个参数，x 表示的是角色在水平方向的位置，往屏幕右边为+的方向，往屏幕左边为-的方向；y 表示的是角色在竖直方向的位置，往屏幕上边为+的方向，往屏幕下边为-的方向；角色的位置一般用 (x, y) 来表示，舞台中央的坐标为 (0, 0)。**移到 x: 0 y: 0** 用于设置角色在舞台上的位置。

舞台坐标中 x 的范围是-240 到 240 之间，y 的范围是-180 到 180 之间。从舞台背景库的“其他”分类中找到“Xy-grid”背景，导入后将在舞台上显示坐标的范围，如图 4-2 所示：

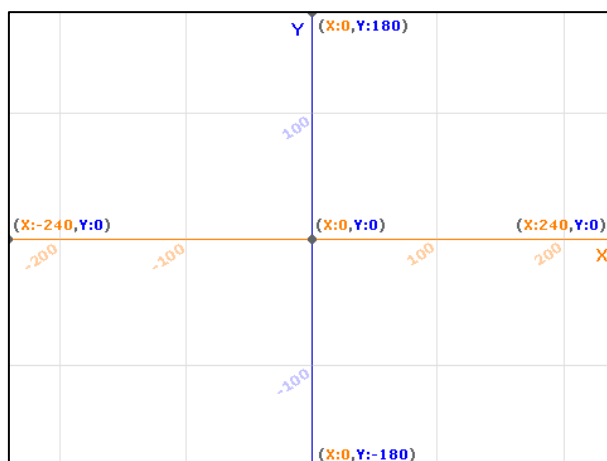


图 4-2 舞台坐标系

坐标在 Vegeta 编程中非常重要，需要仔细理解并熟练操作。通过在舞台拖动角色，**移到 x: 0 y: 0** 中的参数会自动调整为当前角色的当前位置，另外，拖动过程中，角色信息区也会显示对应的坐标，通过这些信息，能帮助我们很轻松的将角色移动到指定位置，如图 4-3 所示：



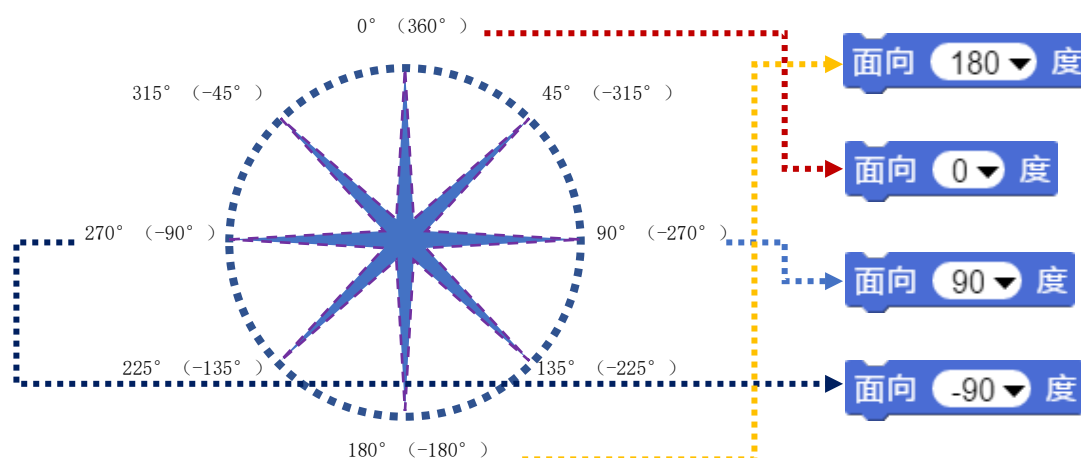
图 4-3 坐标随着角色位置变化而变化

在舞台上尝试拖动角色，并观察坐标变化。除了 **移到 x: 0 y: 0**，在“运动”分类中，找找看还有哪些模块和坐标有关系，尝试并理解它们的含义。

2. 角色的方向

在角色信息区，除了坐标之外，还显示了方向的信息，在 Vegeta 中，方向决定了 **移动 10 步** 模块控制角色时所移动的方向，在动作分类中，**面向 90 方向** 用于设置角色的方向，90 表示面向屏幕右边，-90 表示面向屏幕左边，0 表示屏幕上边，180 表示

屏幕下边。也可以使用 **右转 15 度** 和 **左转 15 度** 控制角色顺时针旋转和逆时针旋转，旋转之后，角色的方向也会发生相应的改变。



给小猴子角色编写如下脚本，点击 ，观察舞台上的角色变化和角色信息区方向值的变化。

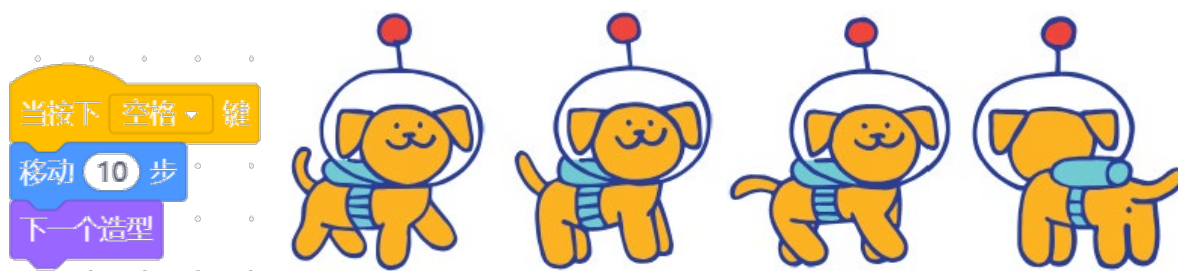


二、事件和键盘控制

在“事件”分类中， 可用于监控键盘的按键事件，通过下拉框设

置参数，可以监控键盘上所有的按键被按下的事件。

1. 新建项目，删除默认的小猴子角色
2. 从角色库中导入“点点”角色
3. 在舞台上将点点拖拽到舞台最左边
4. 为点点编写如下脚本后，连续多次按下鼠标空格键，点点就会在舞台上动起来了，每次按下键盘空格键，点点会移动一次，同时切换造型。



小贴士

在 Scratch-vegeta 编程中，模块有不同的形状， 这种上边像“帽子”一样的模块，它们作为“事件模块”只能放在程序的最顶层。如果发生了某一个事件，就会自动触发执行它们下面的脚本程序。这也是为什么脚本块上面有 的时候，点击小绿旗能触发它执行的原因。像 和 这种上面是凹槽的模块，一般为命令模块，用于执行一种操作，如果模块下方有凸槽，表示可以继续连接其他模块。

积木语言模块共计有四种形状：代表不同种类的积木类型：命令积木、功能积木、触发积木、控制积木。

命令积木：主要执行某一特定指令，特点“上凹下凸”+可以对其进行编辑并单机可执行；

功能积木：只要执行某一功能的指令命令，特点得出布尔值,即“真和假”（是和不是）；true 也就真，代表着满足了条件内容。false 也就是假，代表着条件没有满足。

触发积木：特点某一事件发生了，这一积木开始运行+形状特征是上有圆弧，下凸起连接其他积木；

控制积木（堆栈积木）：特点控制某一特定指令，控制运行流程+控制执行中间指令。

三、条件判断和侦测

在“控制”分类中，“如果……”用于进行条件判断，当满足某一个条件的时候，就执行它的凹槽包含的部分的内容，否则直接跳过执行它后面的内容，另一个条件判断模块是“如果……否则……”，当满足条件的时候，它会先执行“如果”包含的内容，当不满足条件的时候，就先执行“否则”所包含的内容，然后继续执行后面的内容。

条件判断模块的参数和前面的都不太一样，不是一个输入框，而是一个箭头状的参数。在侦测分类中，有很多箭头状的模块，这类模块用于进行判断，一般用于作为条件判断模块的参数，比如侦测中的 **按下 空格 键?** 模块，用于判断当前时刻空格键是否被按下。

将点点移动到舞台最左边，重新为点点编写脚本，实现不断检测按键状态，只要空格键按下，点点就往前走的脚本。



小贴士

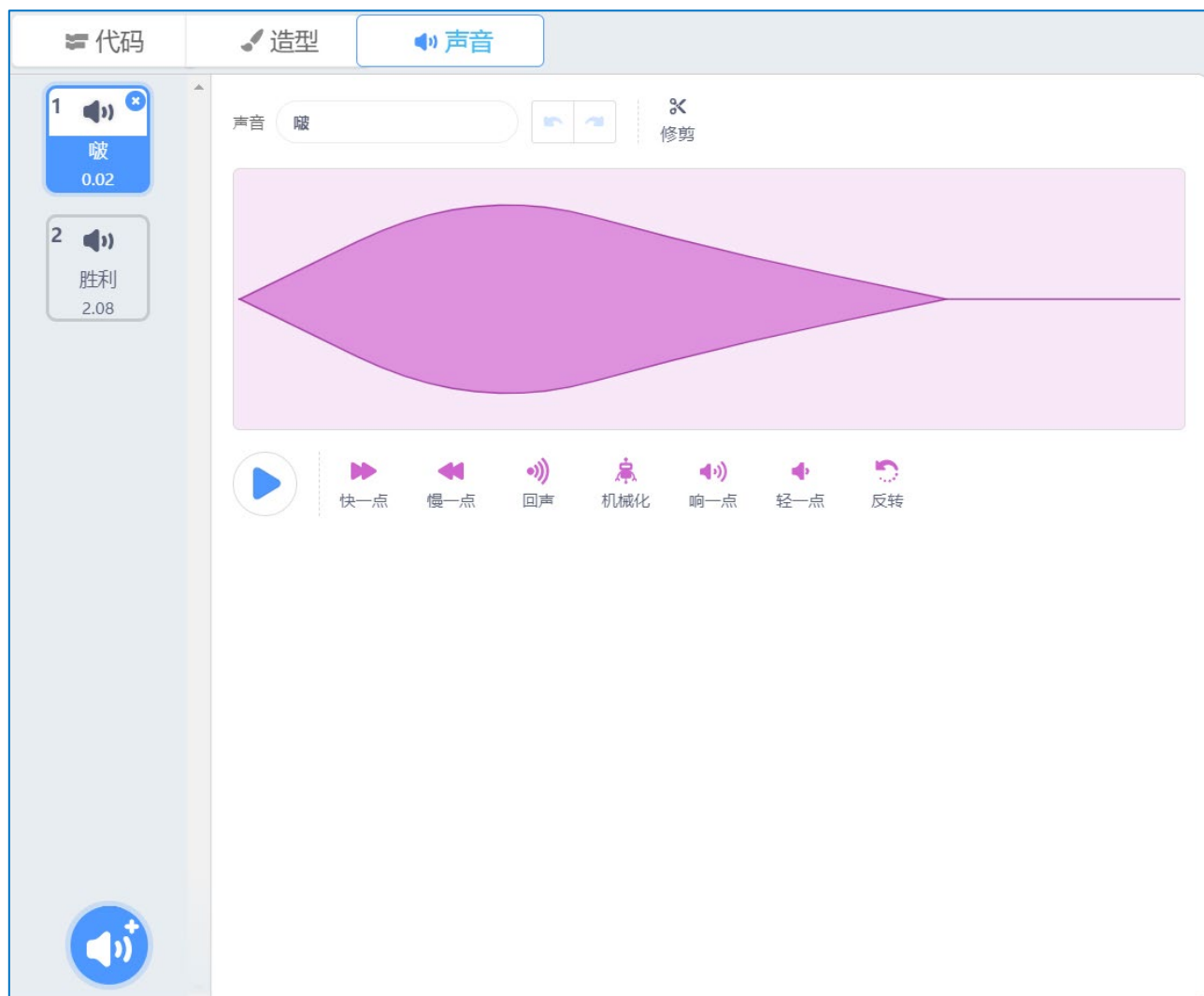
使用事件和侦测控制角色运动的脚本有何区别？使用事件模块“当按下空格键”的时候，按下空格键后，如果不松开按键，点点不会继续走动，在使用侦模块“按下空格键？”实现的时候，必须使用重复执行不断检测，如果按下空格键不松开，点点也会一直走动的。

四、编写迷宫游戏

导入你的声音标签页

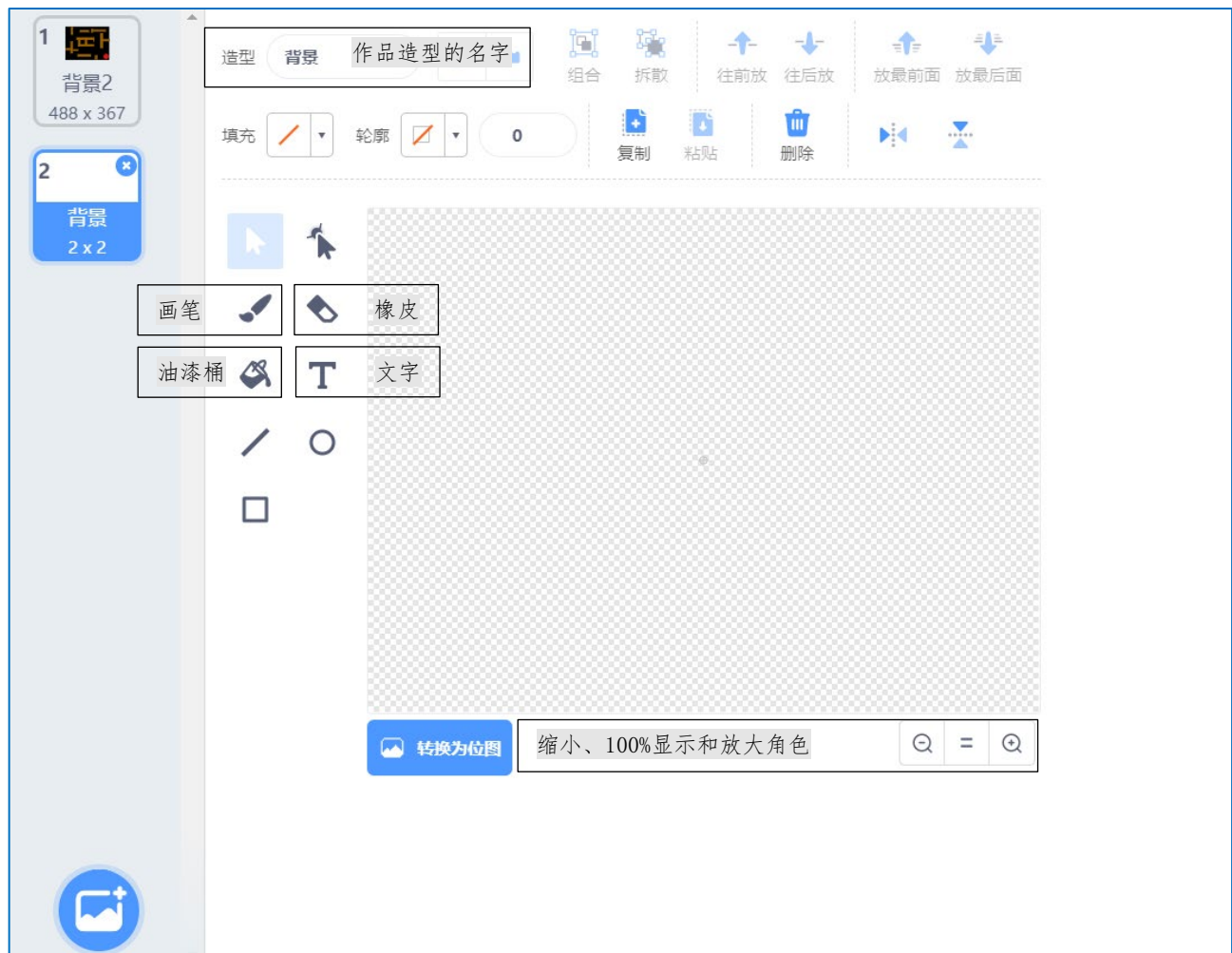
角色除了脚本和造型外，还可以播放声音：可会让你的作品栩栩如生。

声音标签页可以管理你的作品播放的声音，软件不仅可以导入、上传声音，甚至可以录制自己的声音。如图：



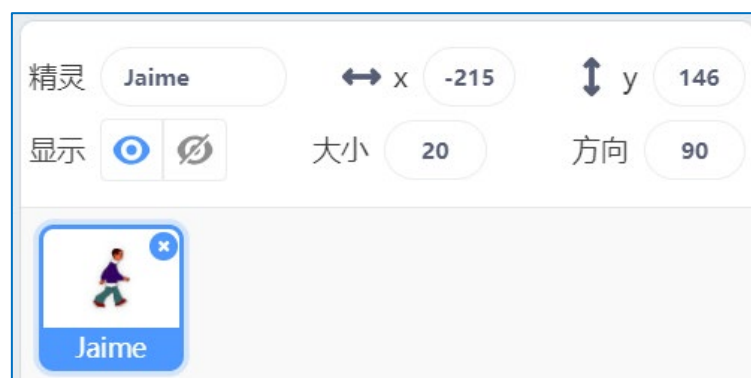
绘图编辑器---铅笔绘画彩虹花

使用绘图编辑器可以创建或编辑造型（背景），点击“绘制新造型”后大屏幕会出现软件自带的画图板：



结合这节课的知识，接下来，我们编写完整的迷宫游戏。

1. 新建一个项目，删除默认角色小猴子
2. 从角色库的“人物”分类中导入“Jaime”角色，并设置角色大小为 20%。



3. 使用画图板为舞台绘制背景图片，作为迷宫地图。绘制时，迷宫的入口在舞台的左边缘，胜利标志在舞台的右边缘，并绘制为红色区域，用于检测是否到达迷宫胜利处。参考如图 4-4 所示：

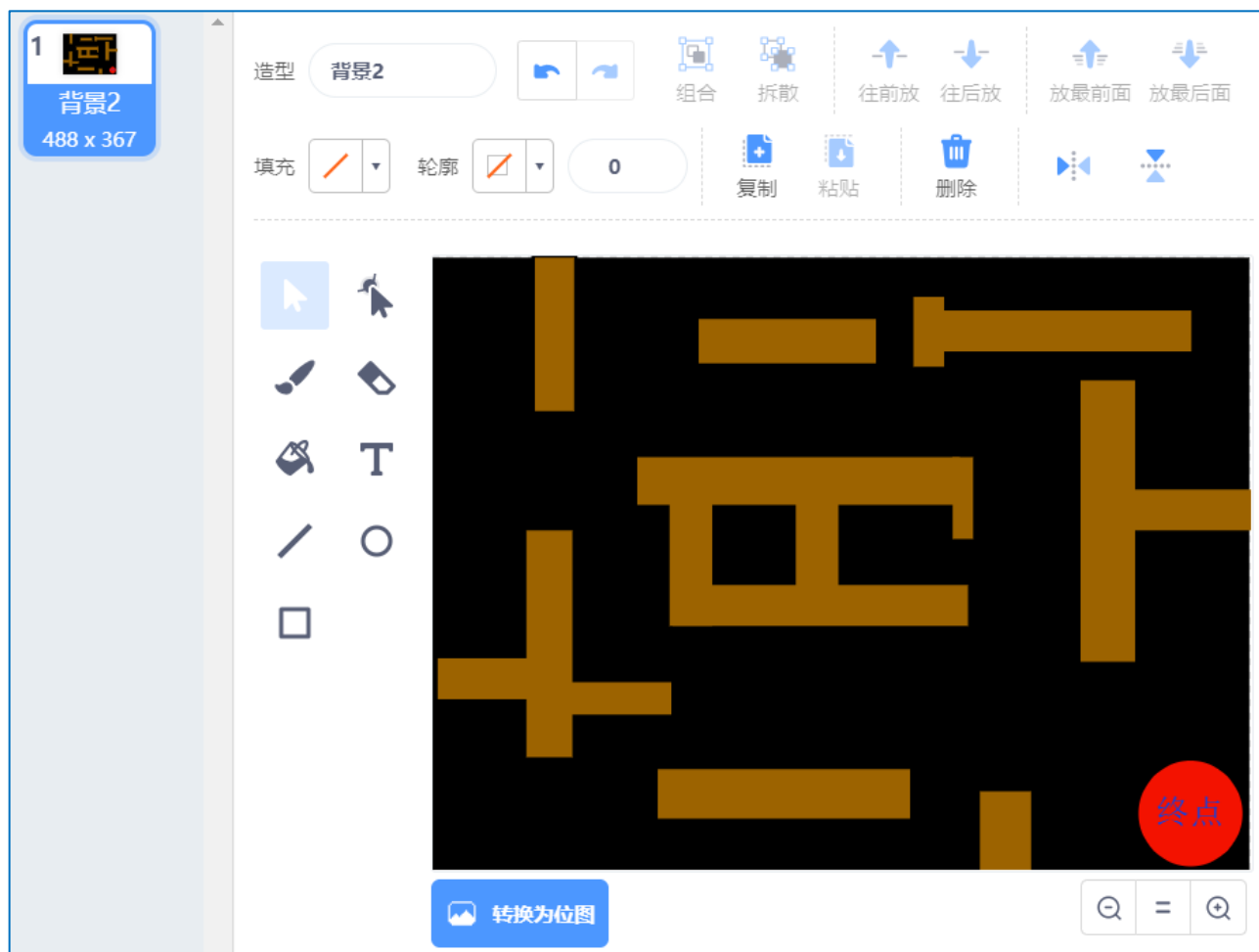


图 4-4 迷宫地图

4. 为了能够让角色通过迷宫，根据所绘迷宫宽度，修改角色的参数，来调整角色大小。将角色拖拽到迷宫入口处，并记录其位置坐标（称之为“初始坐标”）。最终舞台效果可参考图 4-5 所示：

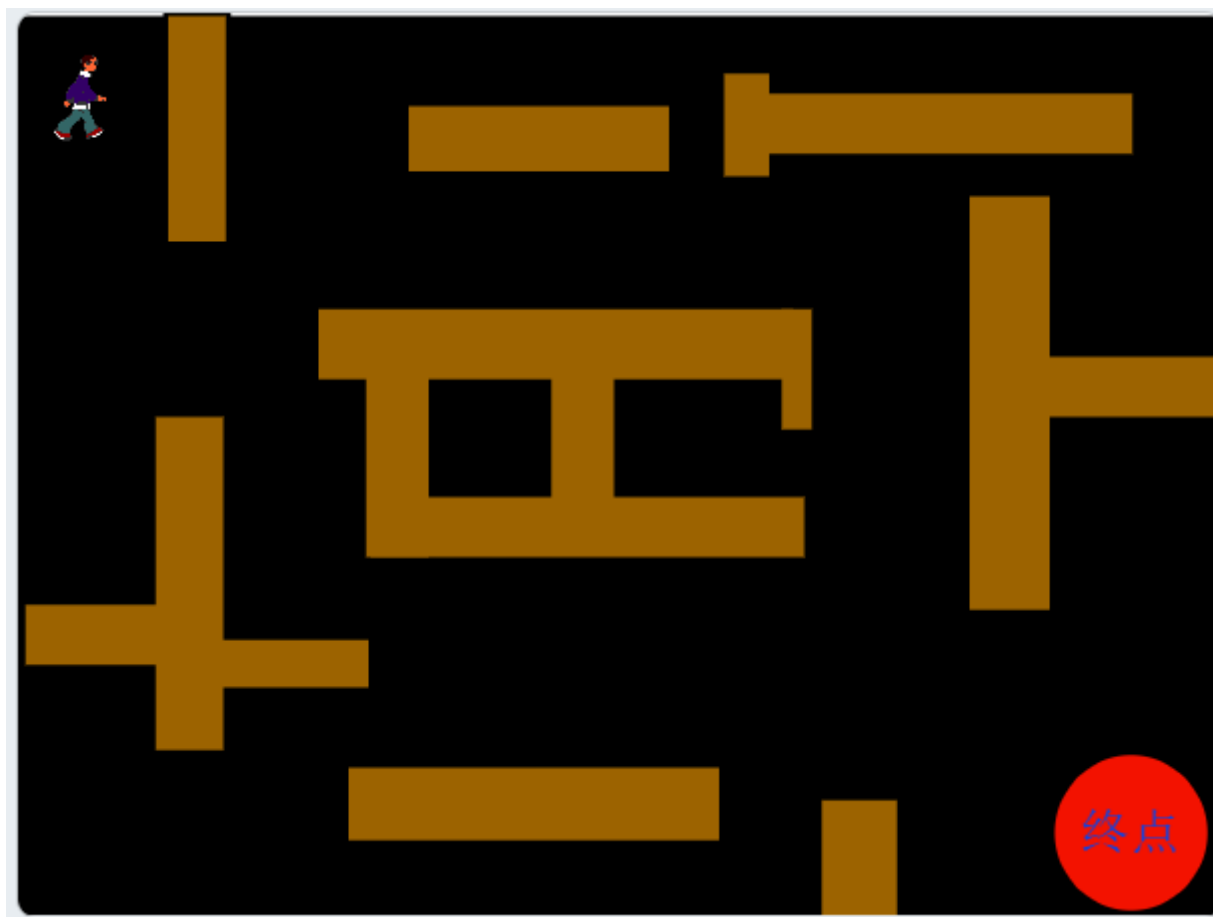


图 4-5 迷宫初始状态

5. 脚本编写

为小猫角色编写如下脚本：

<pre>当绿色旗帜被点击时 移到 x: -215 y: 146 换成 jaime walking-a 造型 面向 90 方向</pre>	设置角色初始位置：将角色移动到舞台左上角，并切换为未运动的 jaime 造型
---	--



设置角色的移动状态：当向下↓方向键被按下后，开行执行两组侦测判断：1) 如果碰到迷宫障碍物或者整个舞台的边缘，则一道最初始位置重新来过；否则按照相对应箭头的方向完成运动轨迹、并更改造型呈现运动效果；2) 如哦成功移动到终点，那么就执行胜利的相应功能。

其余三方向程序以此类推（仅更改当按下方向键的位置和面向的方向）。

取色器的使用方法-如果碰到舞台边缘或者褐色墙面就回到原点

侦测

- 碰到 鼠标指针 ?
- 碰到颜色 ?
- 颜色 碰到 ?
- 到 鼠标指针 的距离

→

碰到 鼠标指针 ?

碰到颜色 ?

→

碰到颜色 ?



→

碰到颜色 ?

碰到 舞台边缘 ?

利用取色器完成取色哟！要不然会不准确

点击小绿旗运行游戏，使用键盘上下左右按键控制点点，看看谁能最快达到迷宫出口吧！



Q: 为什么小明写的语句碰到墙面还有边缘不能够回到原点呢?



A: 不应该用“与”模块而应该用“或”模块，前者是同时发生才能回到原点，后者是只要一个发生就能回到原点。



小明在编写面朝方向的时候和大家起了争议，他说箭头有四种样式，分别对应上下左右，那么我只要切换一下造型就可以，小朋友们，你们认为小明的程序会是什么样的效果呢（）？



-
- A** 和原来的程序一样没变化（小明真聪明！）
 - B** 箭头虽然能左右移动，但是箭头的造型不会变化
 - C** 箭头造型会变化，但是不管怎么按只会向右边移动不会上下左移动
 - D** 箭头怎么按都不会动，只会改变形态。

答案：**C**

解析：大家可别被切换形态所迷惑，我们的角色虽然形态发生了变化，但是他的面朝方向是不会变化的，所以单纯的切换形态只会让箭头换一个样子，还是会按照系统默认的朝向移动，也就是只会向右边移动。

第五课 计算的乐趣



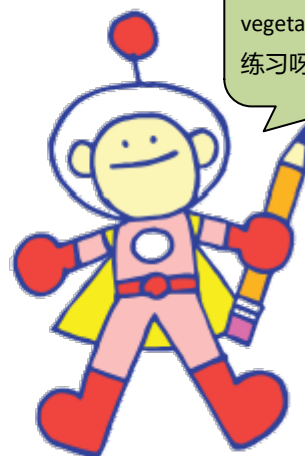
1. 学会变量的创建和对变量赋值的操作
2. 理解随机数的用法和意义
3. 学会 **询问** **你叫什么名字?** **并等待** 和 **回答** 模块的用法
4. 能够用数学运算模块进行变量间的运算操作



这几天数学老师正在教我们四则运算，好难啊！



别慌~可以用 Scratch-vegeta 编写程序进行数学练习呀！



数学对于计算机编程来说是一个重要的工具，无论是制作动画还是游戏，越要接近现实的效果，越需要数学模型来进行表达。接下来，我们使用 Vegeta 制作一个数学题库的程序：电脑自动出题，对答案进行自动判断并统计成绩。

参考如图 5-1 所示：



图 5-1 计算的乐趣



一、创建一个变量

“变量”是一个可以存储可变数的容器，就像一个篮子，你可以装一个猴子，也可以装两个梨，还可以随时拿出来一个或者放进去一个。所以“变量”在 Vegeta 中可以记录一个数字或者一个数学运算的结果。

1. 新建工程，并删除默认的小猴子角色
2. 添加“+”和“=”两个角色

使用画图板绘制两个角色，分别为“+”和“=”，并在舞台上调整到合适的位置。参考如图 5-2 所示：



图 5-2 添加+和=角色

3. 变量的创建

接下来，创建一个“加数”变量，操作如图 5-3 所示：

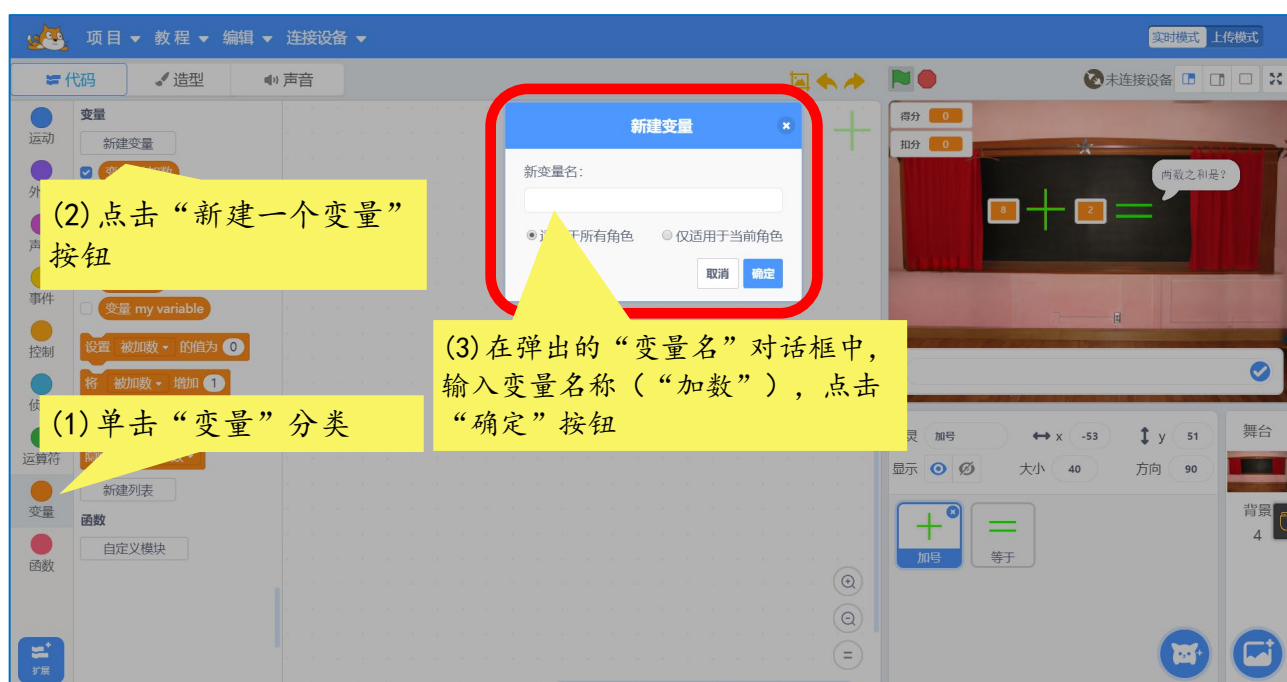


图 5-3 变量的创建

小贴士

在添加变量时需要选择变量类型为“适用于所有角色”或“仅适用于当前角色”，二者的区别是：“适用于所有角色”类型变量创建后可以在程序中的任何一个角色中使用；“仅适用于当前角色”类型的变量创建后只可以在创建时的角色内部使用，在其它角色中看不到这个变量。本课中，变量需要在所有角色中使用，所以需要选择“适用于所有角色”类型。

创建“加数”变量后，可以看到“变量”分类下，会出现以变量名为名称的模块“变量加数”，并且和变量控制相关的四个模块都可以对“加数”变量进行操作：



小贴士

在“加数”变量前有一个“√”，鼠标点击可以选择变量是否在舞台上显示，同时，也可以用编程模块 **显示变量** 和 **隐藏变量** 控制变量是否在舞台上显示。

4. 按照同样的方法，尝试添加“被加数”和“答案”这两个变量，并用鼠标将变量拖拽到舞台上合适的位置，使“+”和“=”组成一个加法算式。参考效果如图 5-4 所示：



图 5-4 变量添加后的效果

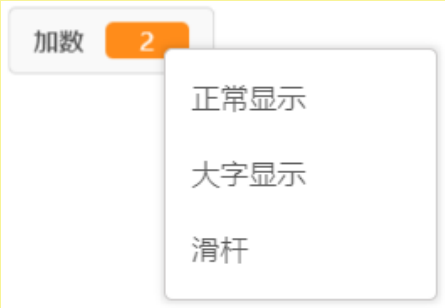
二、为变量赋值

通过创建变量，我们已经拥有了一个完整的加法计算公式，让我们为变量赋值，让它开始运算吧！记住加法的运算法则：“答案”等于“加数”加上“被加数”。

角色：“=”		
脚本功能	脚本实现	操作说明
为变量设置初始值		在“变量”中找到“设置（）的值为 0”模块，将变量设置初始值为 0
答案等于加数加上被加数		在“运算”中找到加法运算模块，并将加数变量和被加数变量拖拽到加法运算模块相应的位置
用随机数自动出题		在“运算”中找到随机数模块，为“加数”和“被加数”赋值

小贴士

变量在舞台上有多种显示效果，右键点击舞台上的变量可以看到选择菜单，菜单中各个选项的作用如下表所示：



名称	作用	效果
正常显示	默认的变量显示效果，同时显示变量名及变量值	
大字	只显示变量值，值的显示尺寸更大	
滑杆	显示变量名和变量值，同时在下方显示滑块，拖动可以调整变量的值	
滑杆的最小值	针对滑块形式，设置变量能够调整的最小值	
滑杆的最大值	针对滑块形式，设置变量能够调整的最大值	

尝试将变量显示效果设定为“大型”，可以看到显示效果更漂亮了：

10

+

7

=

17

重复点击，可以看到每次得到的数值都是不同的，这是因为程序中使用了“随机数”，它可以让计算机在指定的范围内随机产生一个数值，所以每次运行得到的等式都是随机生成的。

在 1 和 10 之间取随机数

用于产生随机数。随机现象在生活中非常常见，比如抛掷硬币猜正反面，每次抛掷的结果都是随机不确定的，想一想，生活中还有哪些现象用到了随机概念？你也可以用随机数编写一个模拟抛掷硬币的程序。

三、我来问你来答

为了让计算机判断用户的答案是否正确，程序首先要让用户输入自己的计算结果，如果输入的结果与答案相同则回答正确，否则回答错误。为角色“=”编写脚本：

脚本功能	脚本实现	操作说明
设置变量显示状态，隐藏正确答案		在“变量”中找到 显示变量 和 显示变量 模块
询问答案		在“侦测”中找到 询问 并等待 模块，在输入框内输入“两数之和是？”
判断回答是否正确		在“侦测”中找到 回答 模块，并利用“运算”中的 = 判断“回答”是否与“答案”相同
显示判断结果		如果判断正确，则利用“外观”中的 说 2 秒 ，显示“回答正确”，否则显示“回答错误”，同时展示正确“答案”

最终脚本如图 5-5 所示：



图 5-5 问答出题的脚本

点击  运行，分别尝试输入正确的答案和错误的答案，计算机准确告知回答是否正确，并对结果进行纠正。

四、统计得分

现在计算机已经能够自动出题，并判断用户输入的答案是否正确，继续对程序进行改造，可以让计算机连续出题，记录用户一共能答对多少。

选择“=”角色，完善程序实现重复出题并记录用户成绩：

脚本功能	脚本实现	操作说明
在得到正确答案时得分加 1		要记录用户答对题目的个数，需要增加一个“得分”变量，并利用“变量”中的值增加 1，在每道题答对时增加 1
在开始运行时将得分设置为 0		每次运行初始化都应该将“得分”清零，否则最终得分将会累计



图 5-6 计算的乐趣完整脚本

点击  运行，效果如图 5-7 所示：



图 5-7 运行效果



这节课我们学会了如何实现一个能够自动出题，并能判断用户结果的加法测试机，你能分别制作出减法、乘法和除法的测试机吗？



Imagine（想象）、Program（编程）、Share（分享）这三个词即为 Vegeta 的核心思想，寓意利用 Scratch 激发儿童的想象、培养儿童的思维，并在智慧分享中共同提高。

教学实践告诉我们，学生喜欢用 Vegeta 做有趣的程序，更喜欢玩自己的程序。学生根据经验想象自己的迷宫，应用 Vegeta 绘制舞台背景。再通过编程处理，设计迷宫关卡的形式、难度等。最后，交流自己的程序，玩一玩，在游戏中拓展自己的想象空间，提高自己的编程水平。同时，有些程序是结合学习、生活来开发的，如计算器、闹钟、台灯等。通过这样有趣的编程活动，让学生打好创造的基础。我们期望 Vegeta 课程能够更有持续性，并能让学生将软硬件结合起来学习。在后续内容，我们开始尝试引导从“趣味编程”引向“趣味创造”，在后续的高级课程中将体现这一变化。

第六课 画梅花



1. 学会画笔的使用，掌握画笔颜色及大小的变化控制
2. 掌握面向和平滑移动的用法
3. 学会图章功能的使用
4. 学习“落笔抬笔”、“画笔颜色设定”等新脚本

听说 Scratch-vegeta 还能画图？能教教我吗？



编程除了包含数学、工程等，还能培养艺术细胞呢！



梅原产中国南方，已有三千多年的栽培历史，无论作观赏或果树均有许多品种。许多类型不但可以露地栽培供观赏，还可以栽为盆花，制作梅桩。鲜花可提取香精，花、叶、根和种仁均可入药。果实可食、盐渍或干制，或熏制成乌梅入药，有止咳、止泻、生津、止渴之效。梅花是中国十大名花之首，与兰花、竹子、菊花一起列为四君子，与松、竹并称为“岁寒三友”。在中国传统文化中，梅以它的高洁、坚强、谦虚的品格，给人以立志奋发的激励。在严寒中，梅开百花之先，独天下而春。在今天的

课程中，我们就学习使用 Vegeta 的画笔功能画出一幅梅花图。并且我们不但可以使用画笔画出美丽的图片，还可以通过计算机编写程序，让计算机帮我们画图。



图 6-1 画梅花

一、导入新角色和背景

要画一棵漂亮的梅花树，首先要准备一朵漂亮的梅花，最后再编程让这朵漂亮的梅花开满整个树枝，现在就来把准备好的梅花图案导入程序吧。



在 Vegeta 中除了从软件自带图库中选择素材之外，也可以将电脑中的图片上传到程序中，操作如图 6-2 所示：



图 6-2 上传新角色

选择“上传新角色”功能后，可以从本地文件夹选择图片上传到程序作为一个新的角色，如图 6-3 所示：

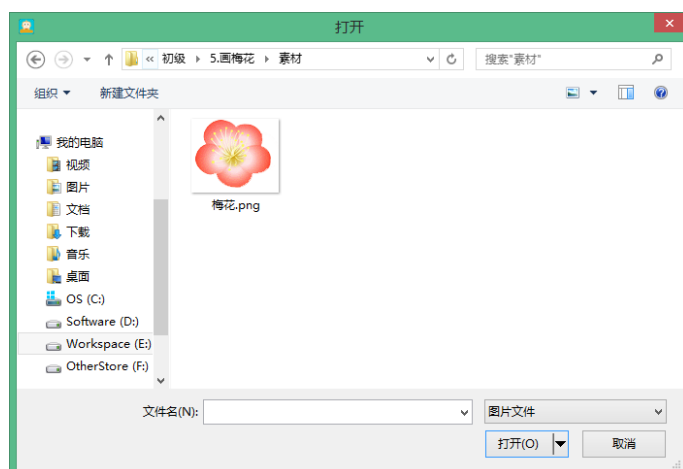


图 6-3 选择图片上传新角色

同样，也可以上传一副图片作为舞台背景：

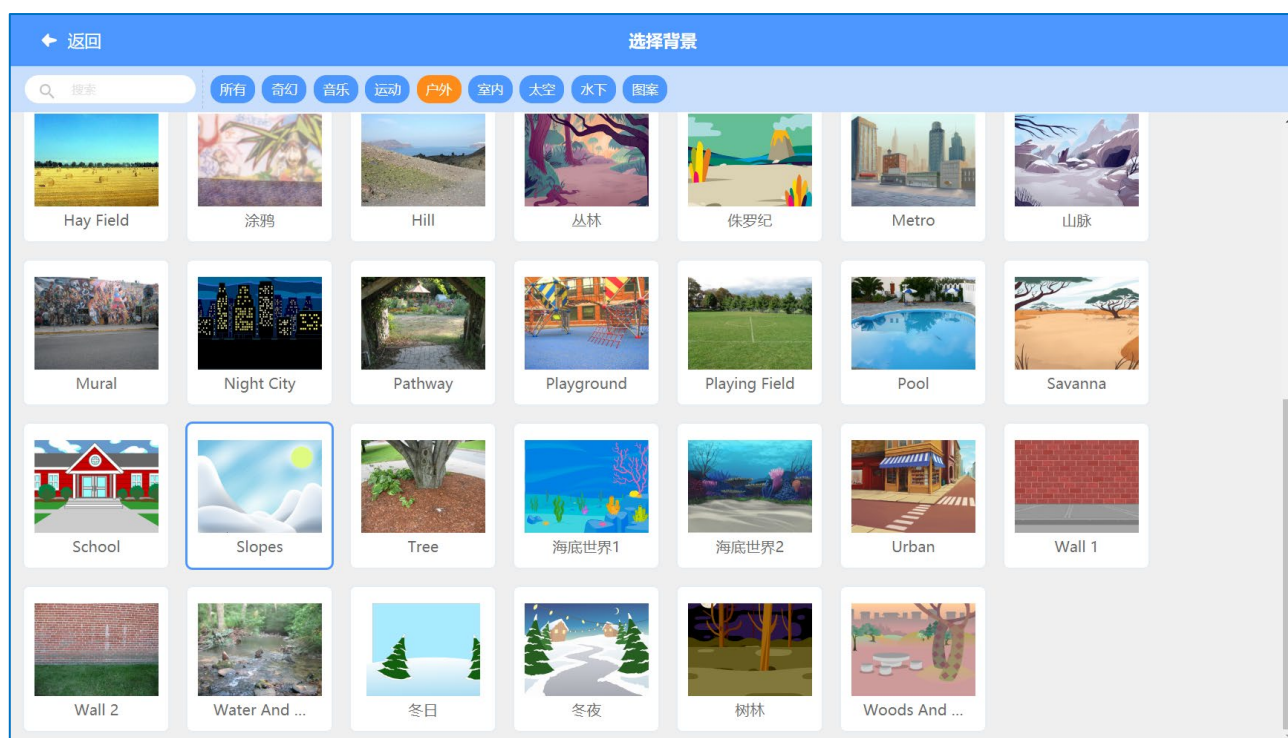


图 6-4 上传背景

自己从网上找到合适的照片或者使用课程提供的图片，参考上面的操作，导入舞台背景和梅花角色。

二、海龟作图法

海龟作图法最初来源于谷歌 LOGO 语言的设计，基本思想是，想象一只海龟在画布上爬行，海龟可以显示或者隐藏，当显示的时候海龟就会留下爬行的痕迹，隐藏时不会留下痕迹，这样就可以编程控制海龟在画布上爬行，画出我们需要的图案了，Vegeta 中画笔功能就是基于这样的思想设计的。需要在软件的扩展功能-人工智能里添加“画笔”：



现在就让我们用海龟作图法在舞台上画一个圆形，操作过程如下表所示：

角色: 		
脚本功能	脚本实现	操作说明
为了让画笔操作更加明了，增添角色铅笔，并设置笔尖为旋转中心点。		
清除画布准备绘图	 0 = red 20 = orange 35 = yellow 70 = green 130 = blue 150 = purple 175 = pink	找到全部擦除，该模块会清除画布，落笔控制开始绘图
设定画笔颜色和大小		找到将笔的颜色设为 () 和将笔的粗细设为 ()，分别设定画笔颜色和粗细度
控制铅笔龟旋转一圈		旋转一圈一共 360 度，可以控制每次旋转的度数和旋转次数，最终乘积要等于 360。每次移动 10 步可以控制圆的半径
抬笔结束绘图		抬笔用于结束绘图，与落笔一般成对出现，需要在移动时不留下痕迹就需要抬笔

通过实践，我们发现上面的程序只能画出一个单色调的图形，你能否利用

 来绘制一个颜色渐变的彩虹花呢？如图 6-5 所示，赶紧动手试试看吧！

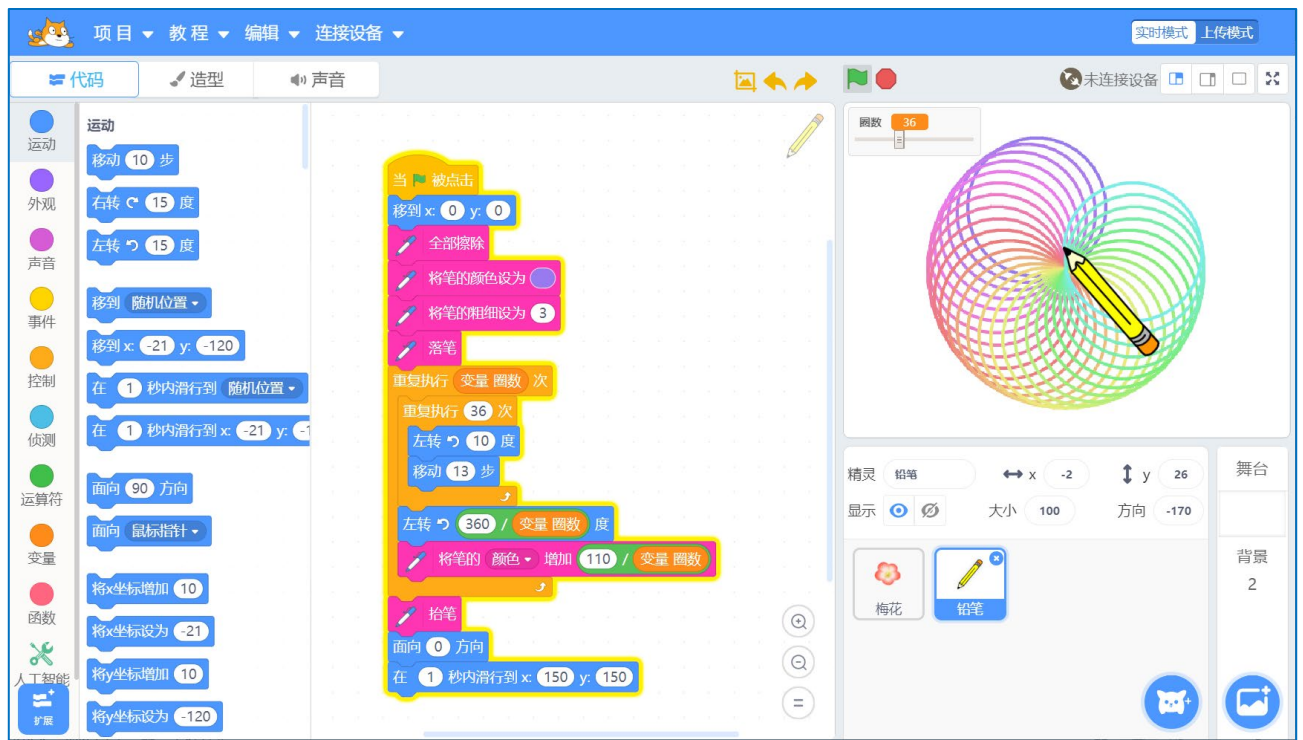
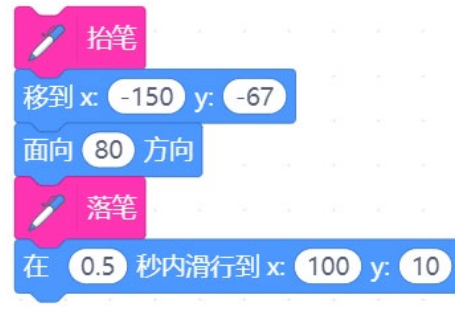
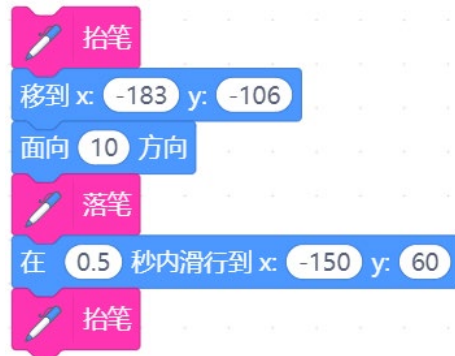


图 6-5 美丽的彩虹花（变量可先设为 36 圈）

三、绘制梅花树干

梅花开在枝干上，所以画一支梅花首先要画它的枝干，操作过程如下表所示：

角色： 		
脚本功能	脚本实现	操作说明
清 理 画 布 准 备 绘 图		设置合适的枝干颜色和粗细度
画 第 一 条 枝 干		在运动中 找到平滑移动，这样可以看到画图的过程

画 第 二 条 枝 干		在移动到第二条枝干起始点位置时，需要用到拾笔保证在移动过程中不留下痕迹
画 第 三 条 枝 干		同样方法绘制好第三条枝干后，拾笔结束画图

运行后，绘制效果如图 6-6 所示：



图 6-6 绘制树干运行效果



我们见到的树枝应该是什么样的？应该是主干粗，末梢细，你是否可以利用

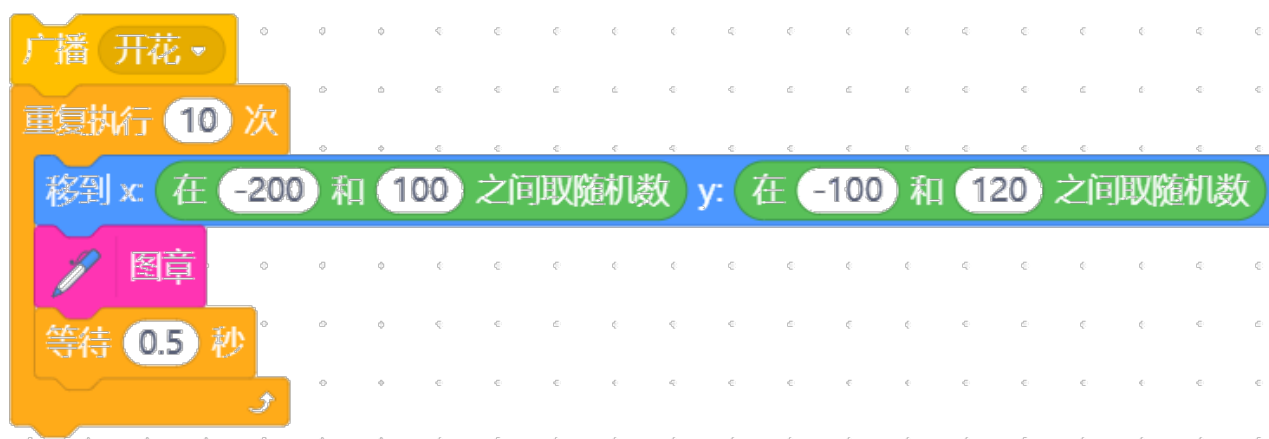
 将笔的粗细增加 1 功能实现这样的效果呢？

四、盛开的梅花

梅花枝干已经准备就绪，赶快让我们的梅花在枝干上盛开吧！为了让每次绘制的梅花图都独一无二，可以运用随机数控制梅花盛开的位置。

梅花的盛开主要利用画笔模块中的  图章 功能，图章功能就像我们生活中常见的盖章，可以对角色进行复制，然后粘贴到指定的地方。

具体操作过程如下表：



最终脚本图 6-7 所示：

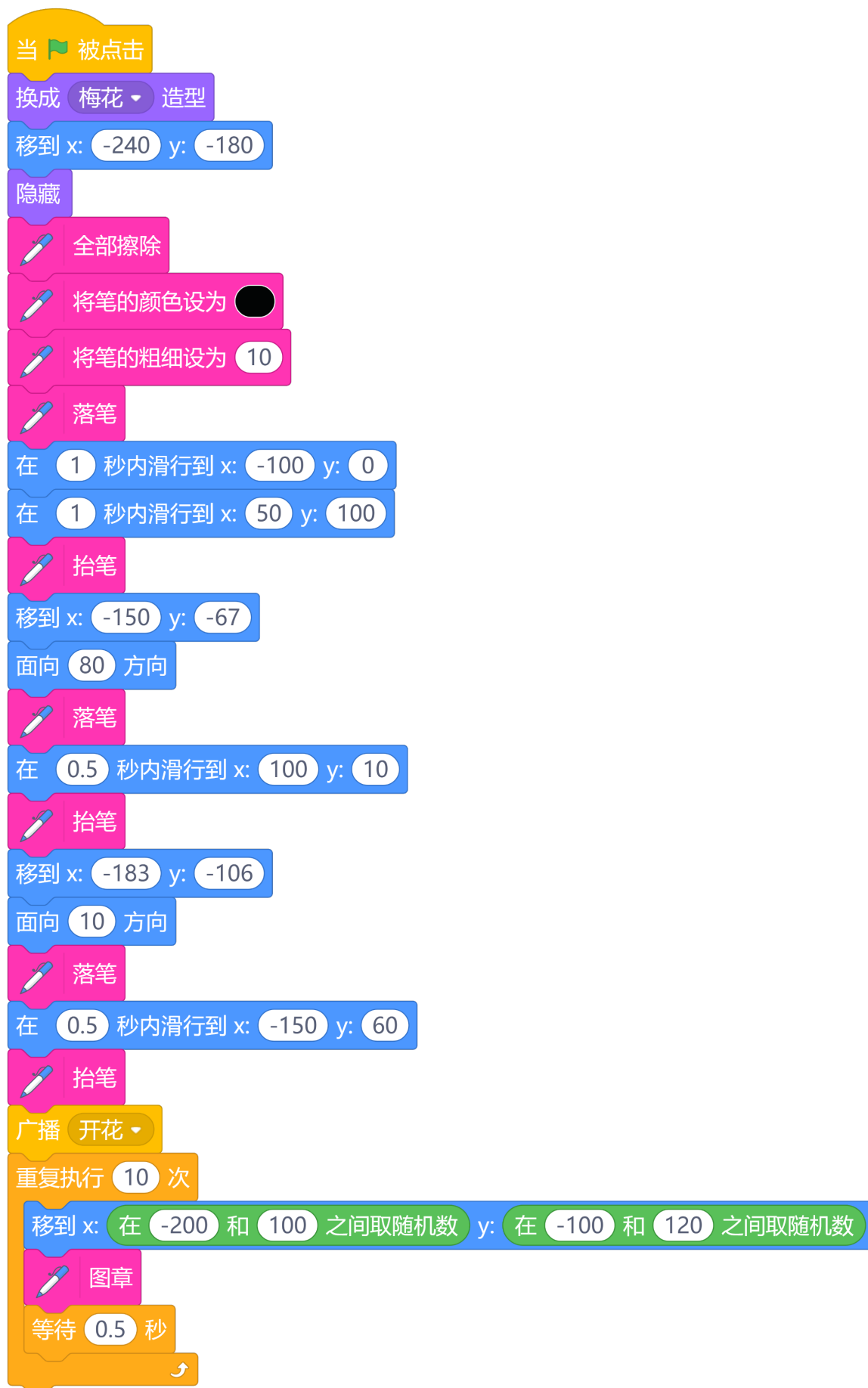


图 6-7 画梅花完整脚本

在这节课中，我们不但绘制了一朵美丽的彩虹花，还绘制了一枝盛开的梅花，你能用画笔绘制出其他的图形吗？比如，一个正三角形，正六边形，或者一棵猴子树？

关于克隆：

克隆 自己 ▾

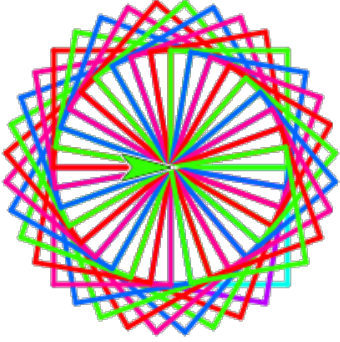
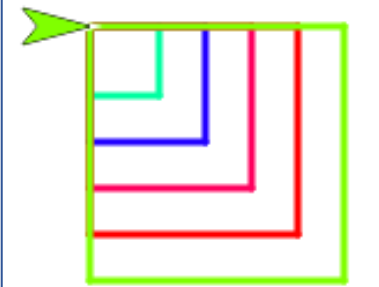
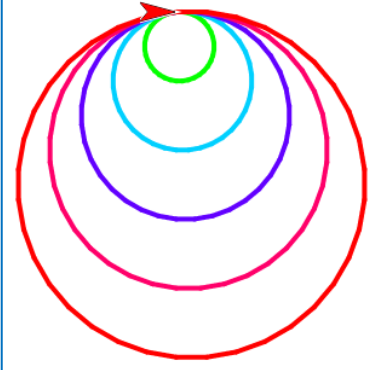
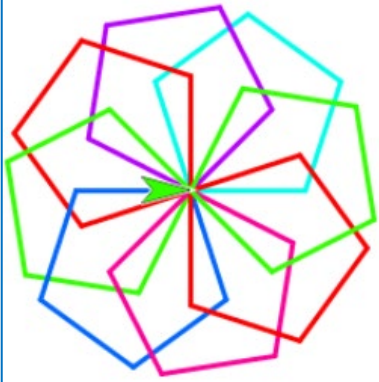
在现实生活中，克隆不是一个陌生的词语，比如科幻电影常会出现克隆人、克隆羊等，而在编程的语境中，克隆也有类似的含义。克隆总要有个母体、本体或者原始版，然后以它为模板创建、复制出几乎完全相同的克隆体。

克隆，是程序软件为我们提供的一项重要功能，它允许在程序运行过程中，创建某个角色的克隆体，克隆体与原始角色相同，但它们独立运行、互不影响。

任何角色都可以克隆出自己的克隆体，克隆体在创建时，完全相同地继承原始角色的造型、声音、属性和脚本，然后在接下来的运行中，克隆体可以对这些进行修改。

小贴士

“克隆”是一个科学术语，意指将一个个体完全复制，它与母体有完全一样的属性，但是可以根据自己意志活动。在编程软件中，克隆通常用于需要在舞台上同时出现多个相同角色的情况，比如发射子弹，只需要一个子弹角色，通过克隆就可以在舞台上同时发射无数多个子弹。

画成图形	请写出该程序的编写思路
	
	
	
	
	

第七课 贪吃的小老鼠



1. 学会广播功能的使用方法
2. 学会如何侦测颜色
3. 学会停止脚本

在编程软件中，如果多个角色之间要协调工作，怎么做呢？



哈哈！用广播啊！广播是很好的一個功能，能实现复杂的角色交互。



有只贪吃的小老鼠非常喜欢吃香甜的饼干，现在它又闻到了饼干的香味，你能帮它找到这块饼干吗？在本节课，我们就编写程序，让小老鼠沿着饼干的香味找到饼干。

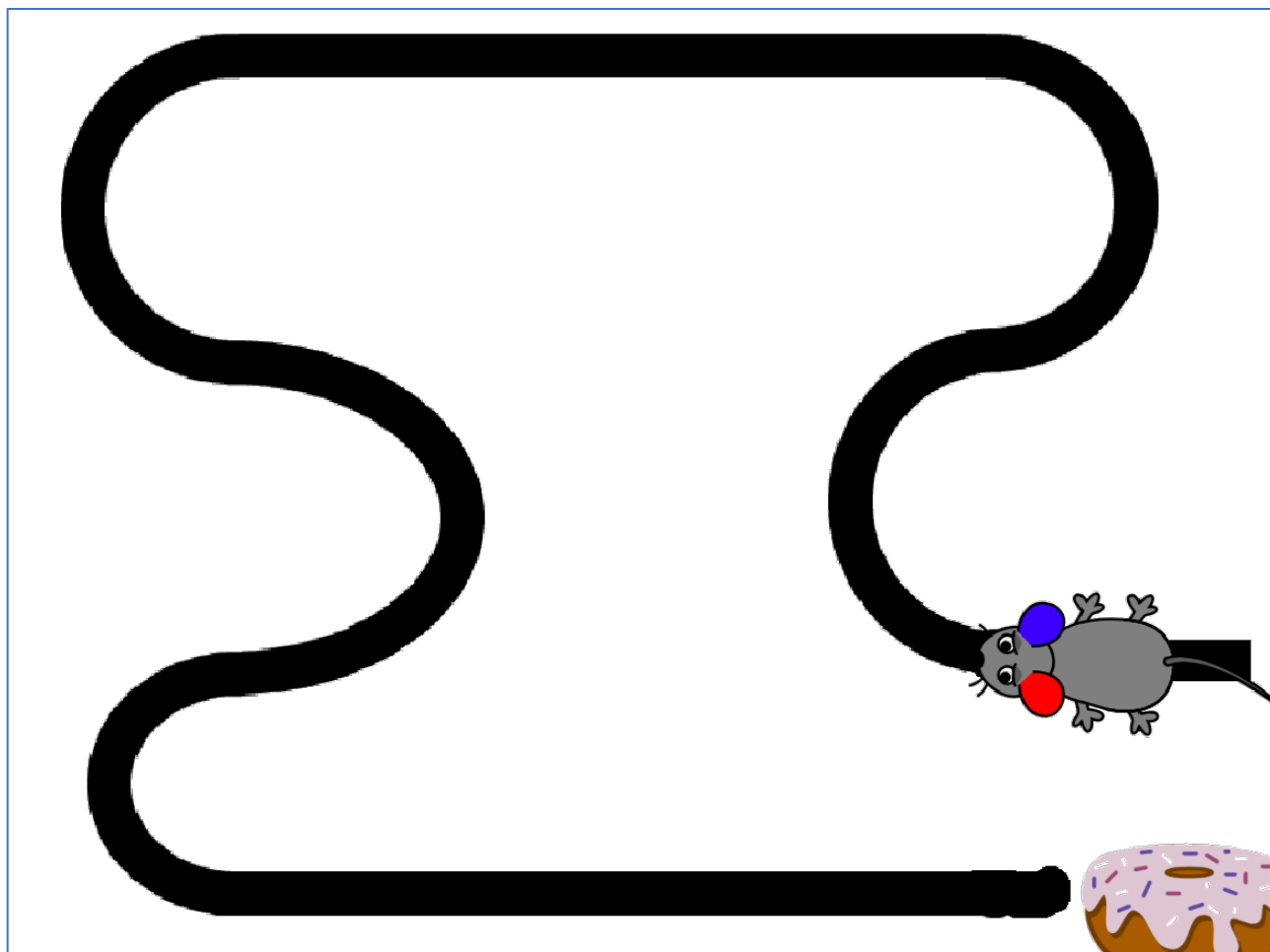


图 7-1 贪吃的小老鼠

一、找甜甜圈的路线

本课中，小老鼠要按照一条指定的线路走到终点，在终点有个大大的甜甜圈奖励它，首先让我们为它准备甜甜圈和寻找的路线吧。



首先，在背景栏，用画图板的“画笔”工具绘制一条路线，效果参考如图 7-2 所示：

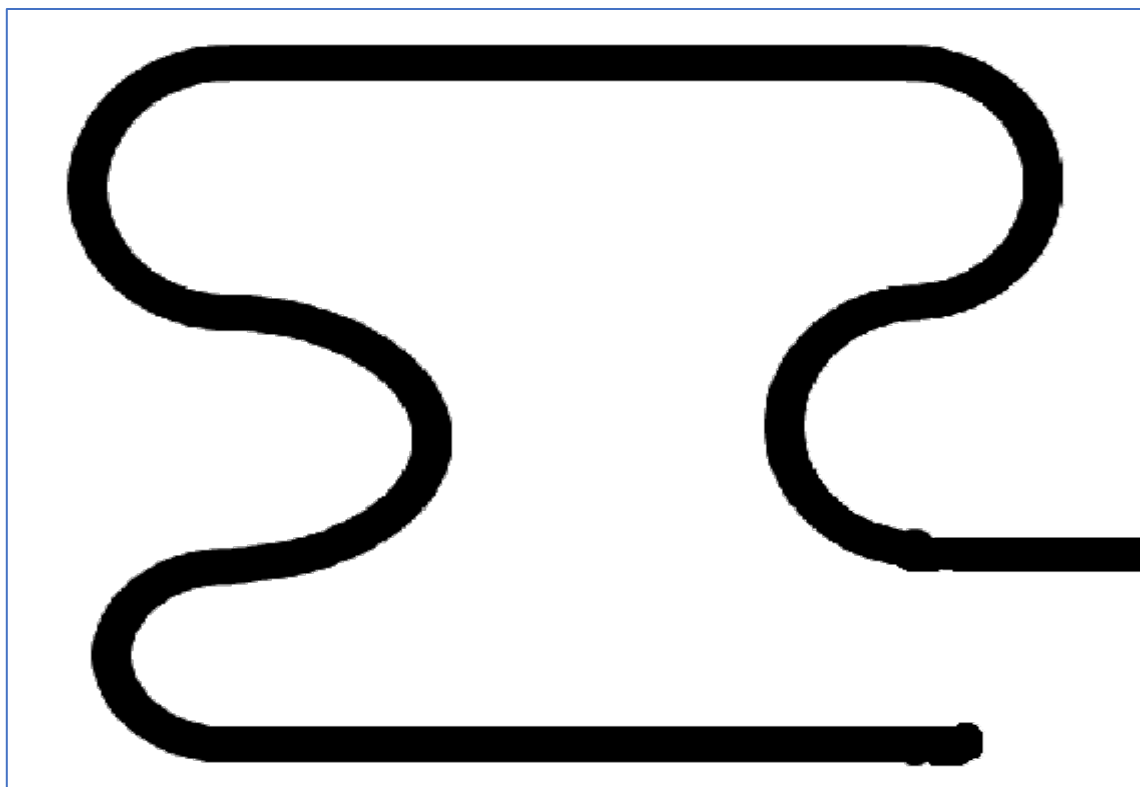
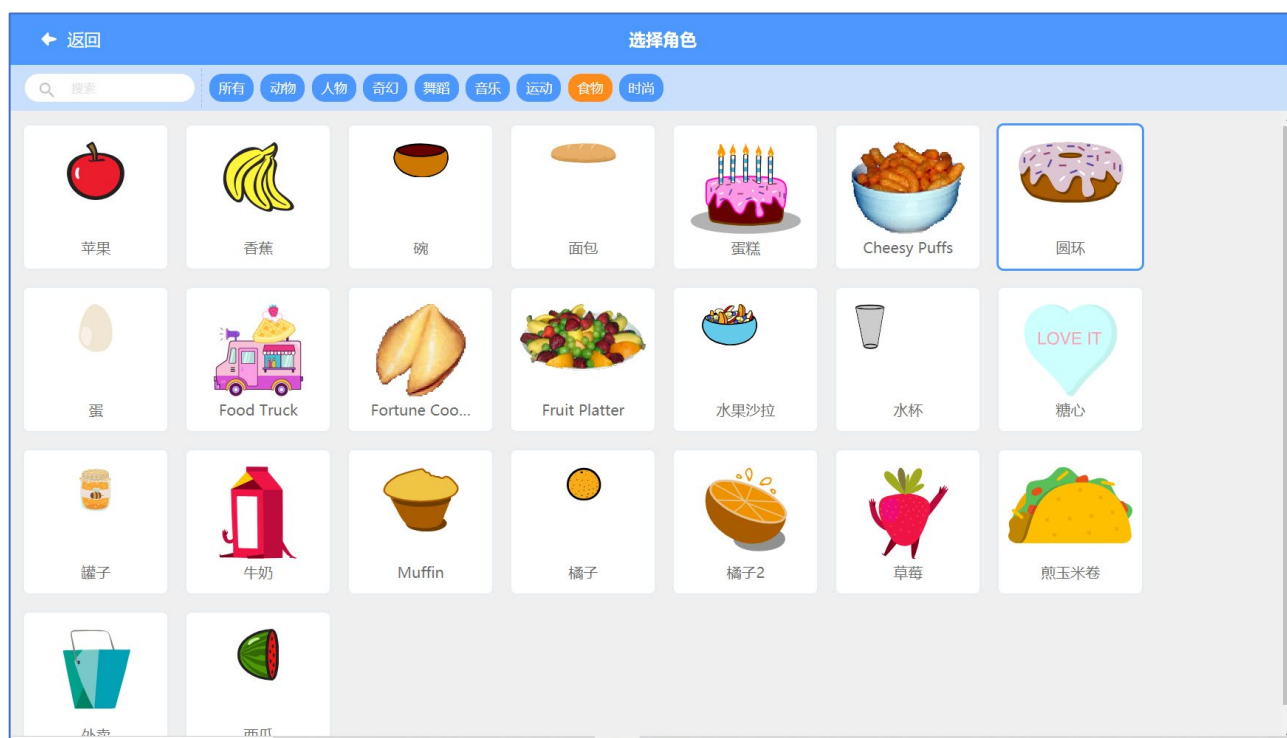


图 7-2 绘制路线

现在添加一个“甜甜圈”角色并放在路线终点处。我们可以在 Scratch 软件角色库中找到一个合适的角色，这里我们选中的角色是“食物”分类中的圆环角色，并摆放到路线的终点位置。为了保证饼干覆盖在路线上，编写脚本如下：



角色： 		
脚本功能	脚本实现	操作说明
将甜甜圈角色移动到最上层		在“外观”中找到  模块，保 证角色运行时不会被其他角色遮挡

最终效果如图 7-3 所示:

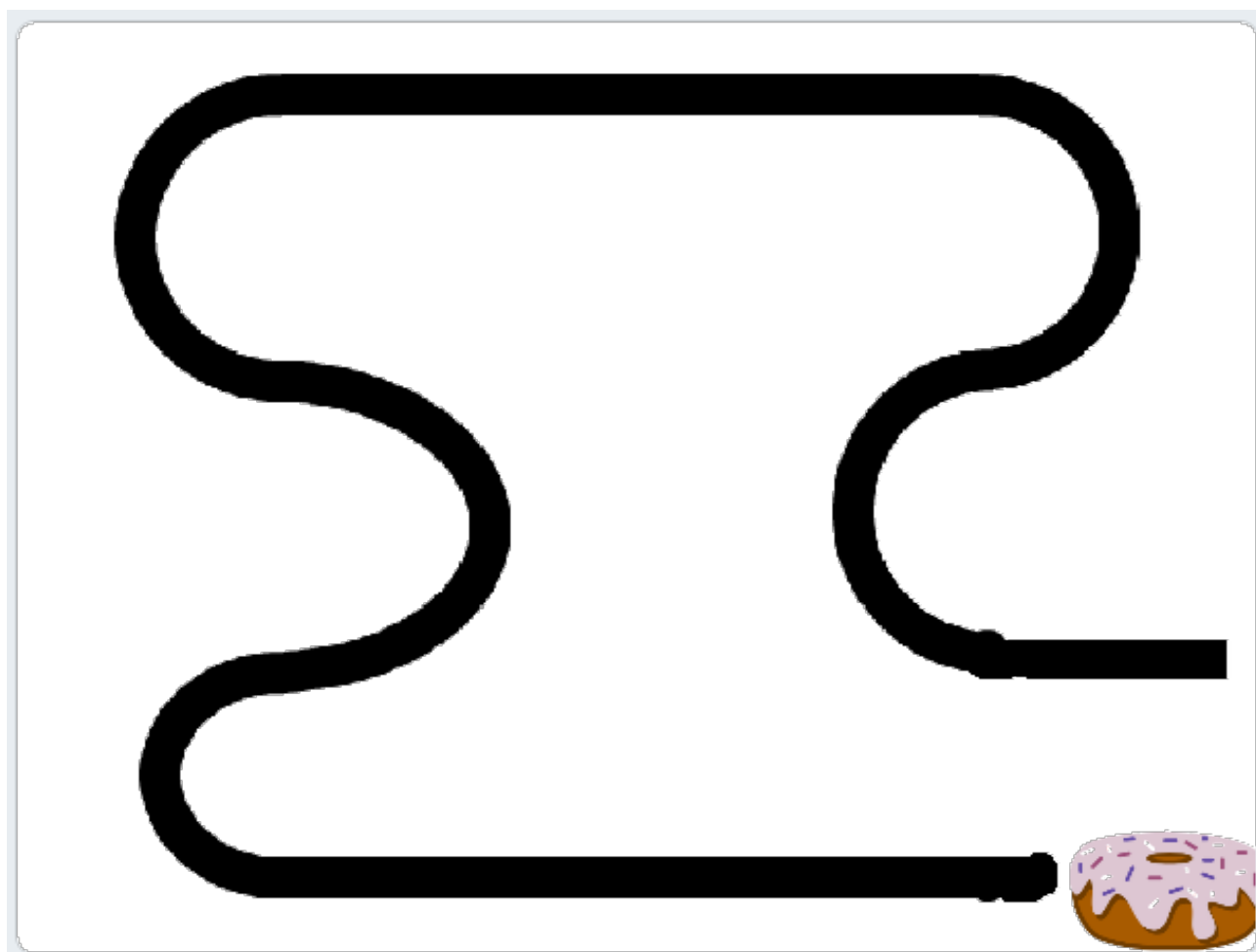


图 7-3 绘制路线和饼干

小贴士

移到最前面和前移一层可以控制角色之间相互遮挡的效果，具体作用如下图所示：



图 7-4 角色之间的层次关系

二、如何检测颜色

如何才能让小老鼠沿着线路前进呢？那就需要先判断小老鼠相对于路线是偏左还是偏右，然后根据情况修正它的方向。为了侦测小老鼠偏左还是偏右可以利用老鼠造型的两只耳朵，根据两只耳朵在路线的左右位置来判断。



利用“导入新角色”从软件角色库中“动物”分类中导入“老鼠” 角色，然后在“造型”界面对角色进行编辑，在画图板中利用“油漆桶”功能分别将小老鼠造型的两只耳朵填充为“红色”和“绿色”，效果如图 7-5 所示：

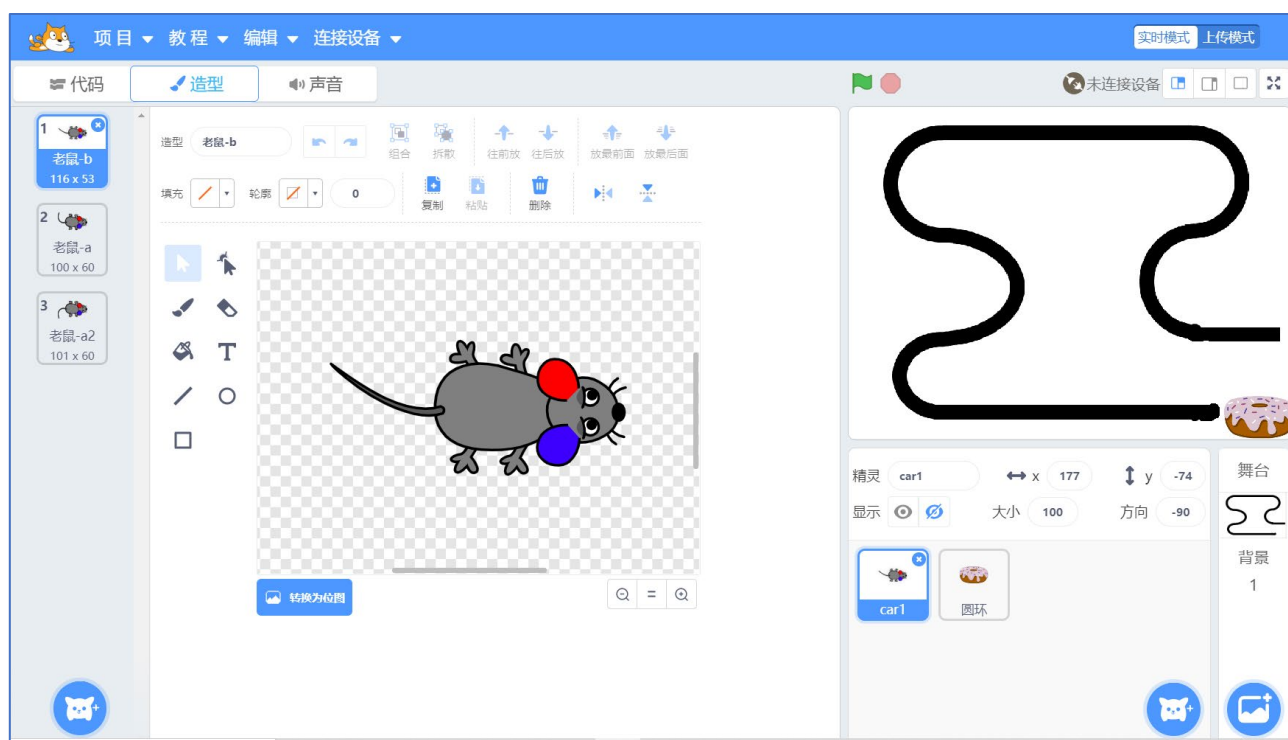


图 7- 5 填充耳朵后的小老鼠

编辑好之后可以把小老鼠放在“路线”角色上的合适位置，效果如图 7-6 所示：

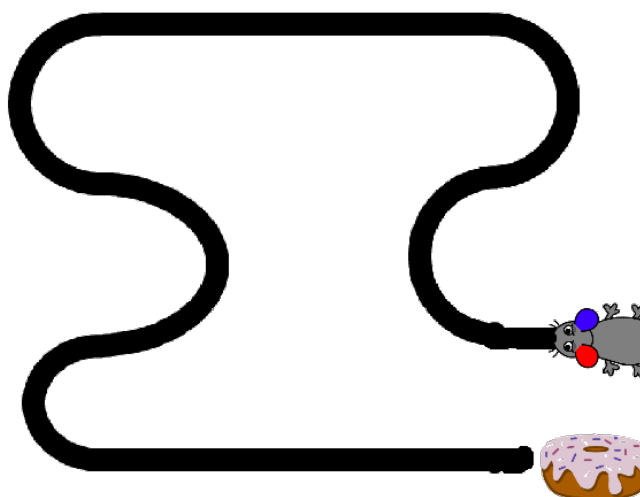
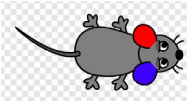


图 7-6 将老鼠放到路线的起始位置

现在，可以对“路线”角色进行编程，判断小老鼠应该左转还是右转了，操作过程如下表所示：

角色： 		
脚本功能	脚本实现	操作说明
左边红耳朵碰线左转		在“侦测”中找到
右边绿色耳朵碰线右转		 ;
重复检测		不断检测，以便及时告知小老鼠左转还是右转。

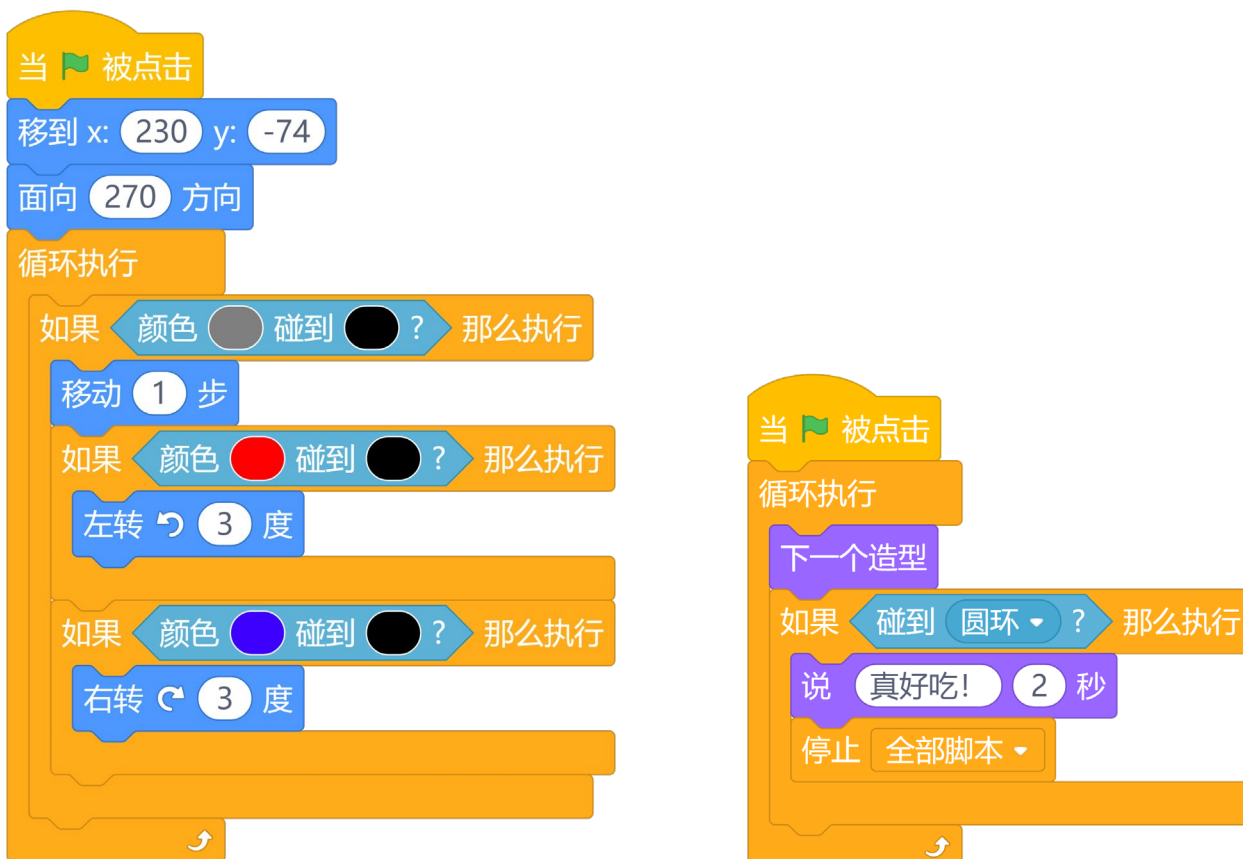
小贴士

在“侦测”中有  和  两个侦测模块，二者的不同是：
 只能侦测到是否碰到某个角色或鼠标，而  可以侦测到角色身上或者背景图案中的某一块颜色区域，功能更加强大，但是侦测速度比较慢，一般情况尽量使用 ，可以让你的程序运行更快。

三、听话的小老鼠

“路线”已经及时告知小老鼠应该左转还是右转，只要小老鼠能够按照指令行动就一定能够顺利找到甜甜圈。

选择  角色，按照下图所示代码进行操作：



- 1.使用  模块，思考如果不使用广播如何实现该程序。
- 2.本节课我们学会了怎么让小老鼠沿着知道路线前进，你能制作一个迷宫，并让小老鼠自己走出迷宫吗？

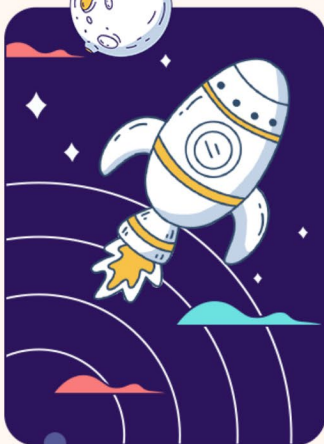
第二章节 机器人世界

// ——— 创想Vgeta-启程 ——— //

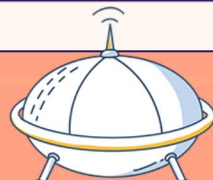




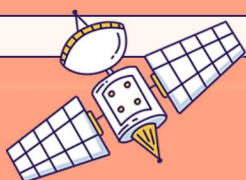
章节序言



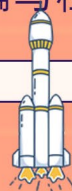
内容描述



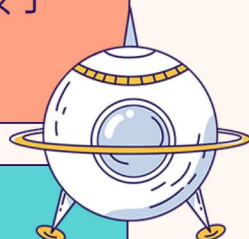
- 传感器（英文名称：transducer/sensor）是一种检测装置，能感受到被测量的信息，并能将感受到的信息，按一定规律变换成为电信号或其他所需形式的信息输出，以满足信息的传输、处理、存储、显示、记录和控制等要求。贝吉塔电子积木将传感器制作成积木的形式，用于更方便的搭建作品，编写程序。



内容介绍



- 此章节针对机器人入门的基础知识展开，以机械结构方面的物理知识为主要的介绍，为以后的智能机器人研究打下坚实的基础。从实际的应用生活场景出发，以实际案例为主，通过一个个作品展现机械知识及扩张孩子们的眼界。在以后的系列课程中，会出现越来越多的复杂机器人及智能机器人介绍及概念，让孩子们理解并掌握国际潮流下的机器人。

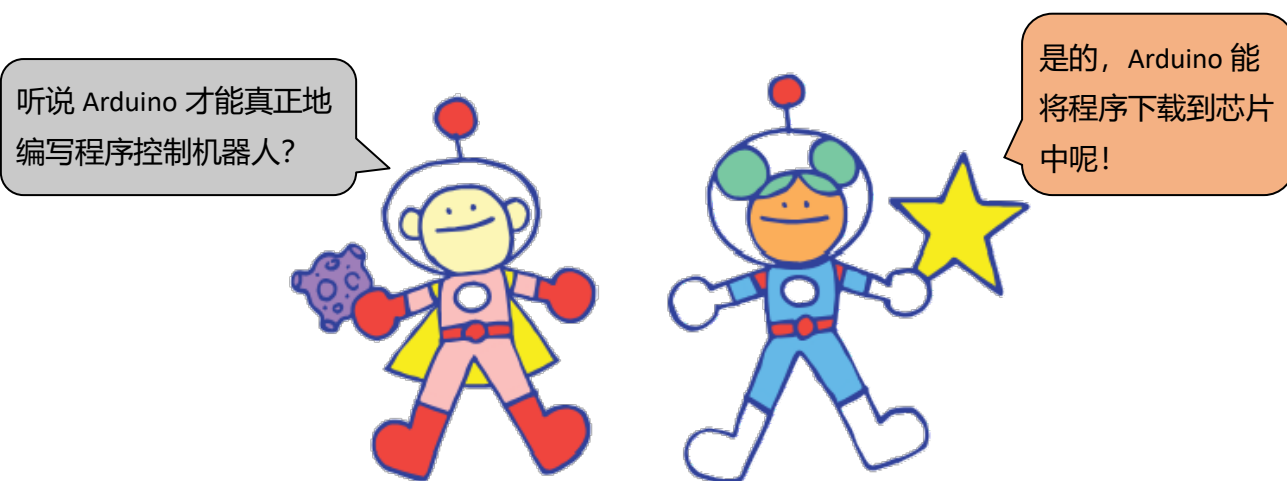


少年强则国强。作为国家的朝阳、希望与未来，让我们一同研究机器人、共同迎接机器人时代的到来。

第一课 进入传感器的世界



1. 了解智能设备的概念
2. 掌握 Arduino 程序的基本框架
3. 学会 Arduino 程序的上传和运行
4. 了解上传模式和实时模式的区别



一、切换到上传模式

点击菜单上传模式就能从实时模式切换到上传模式了，如图 1-1 所示：

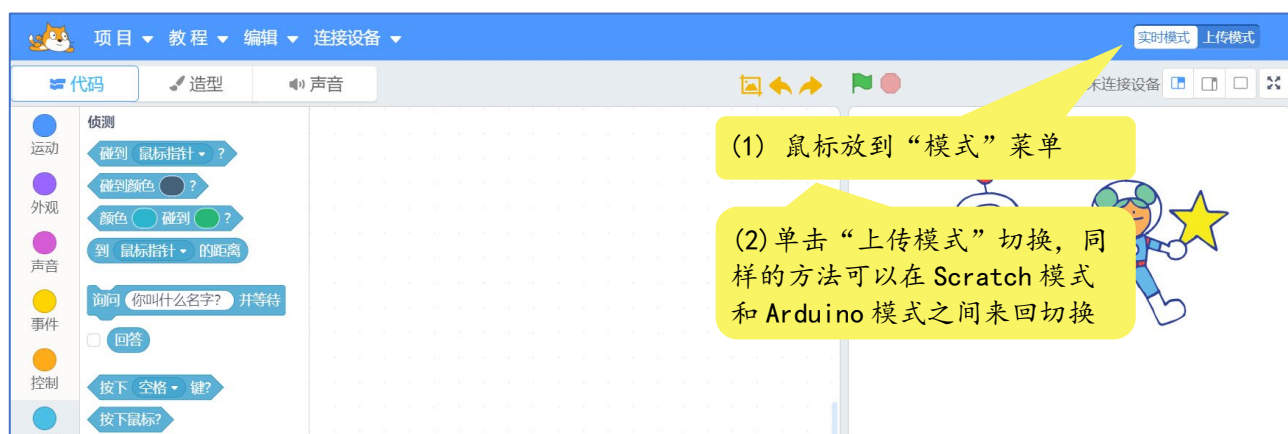


图 1-1 模式切换菜单

二、上传模式主界面

在上传模式下，软件的界面有很大的区别。舞台区、角色区、角色信息区等都被隐藏起来了。在模块分类中，部分分类以及模块被禁用了。造型面板和声音面板也被隐藏起来了，只剩下脚本面板了。在脚本面板右边，增加了一个代码区，里面有一些程序代码。界面各个分区如图 1-2 所示：

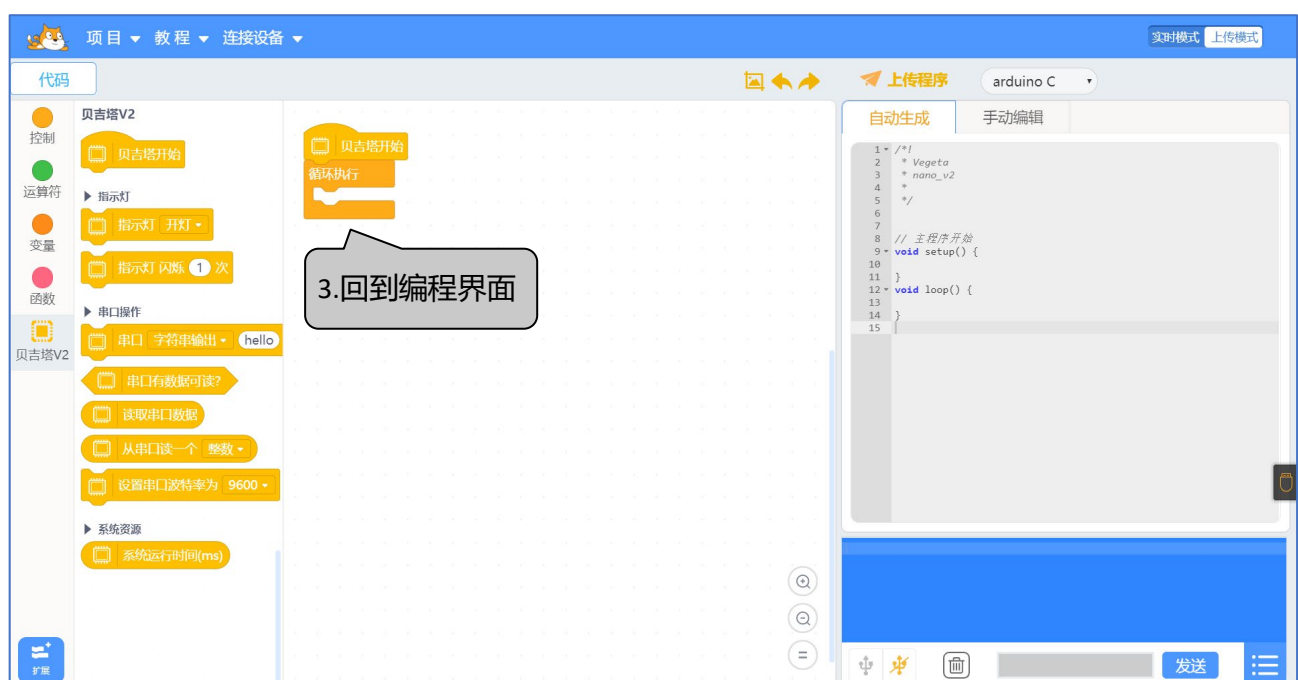


图 1-2 上传模式主界面

小贴士

为什么在上传模式下，有关动画的功能模块都被禁用了呢？因为在上传模式下，程序最终会上传到远程主控板的芯片中，并脱离电脑运行。由于远程控制板的芯片不具备电脑的所有功能，包括操作系统和显示器，所以不可能运行 Scratch 模式下的动画和游戏，因此只保留了和 Arduino 有关的较少模块，但是 Arduino 编程比 Scratch 编程更复杂，如何利用有限的模块完成复杂的功能呢？让我们一起来了解 Arduino 的优势吧！

三、Arduino 程序的基本框架



在脚本面板中，默认会有“贝吉塔开始-循环执行”这个模块，该模块是 Arduino 程序的基本框架。远程控制板（即 Arduino 主控板）上电之后，就会执行一次“Arduino 初始化”里面的内容，然后就会重复不断的执行“Arduino 主循环”里面的内容，这就是 Arduino 程序的基本框架。同时，Arduino 程序有且只能有一个主循环模块。



小贴士

上传模式中 Arduino 初始化所对应的代码为代码区中的“setup()”部分，Arduino 主循环对应的代码为代码区中的“loop()”部分。Arduino 的主循环会不断执行，相当于自带重复执行的功能。

小贴士

在实时模式下，电脑是利用蓝牙或串口与远程控制板进行通信的，由于电脑有很多其他任务需要执行，因此通信并不是实时发生的，所以，运行速度相对 Arduino 更慢，很多对响应时间有实时性要求的任务，实时模式无法完成，此时，就必须使用上传模式了。

第二课 控制 LED 灯



1. 学习可编程电子积木的使用方法
2. 学会编程控制 LED 灯的方法
3. 学会如何获取触碰开关状态，并根据状态控制 LED 灯

阿力麻球球哈丫丫！
亮灭随心控！



什么是 LED 灯？

在 19 世纪 70 年代末，爱迪生发明了电灯，点亮了漆黑的夜晚，照明技术进入了崭新的时代。20 世纪 60 年代，LED 灯（发光二极管）也被设计出来，然后就出现了各式各样的绚丽的 LED 灯产品，其应用范围也越来越广泛，比如路边的红绿灯，汽车仪表指示灯，都是通过小小的 LED 灯实现的。



电子积木名称	图例	用法说明
控制器	 奇颖主控器 奇颖主控器主频高达32MHz，内存32K，内置10个接口	贝吉塔控制器是进行编程指令处理的核心模块，所有电子积木都要听它指挥
LED 小灯	 小灯 高亮照明LED，控制小灯打开、关闭、闪烁等	使用  控制  LED 灯的亮和灭。
按钮/碰撞开关	 按钮/碰撞开关 读取按键状态，按键事件执行	使用   可以检测触碰开关的状态。

一、硬件连接



找到如下模块：主控板、LED 小灯、数据连接线、传感器连接线。

按图 2-1 所示，LED 灯模块连接到 1 号端口：



图 2-1 贝吉塔控制器连接 LED 灯，并将 USB 数据线插入电脑 USB 接口：

打开贝吉塔控制器开关，电源指示灯常亮：

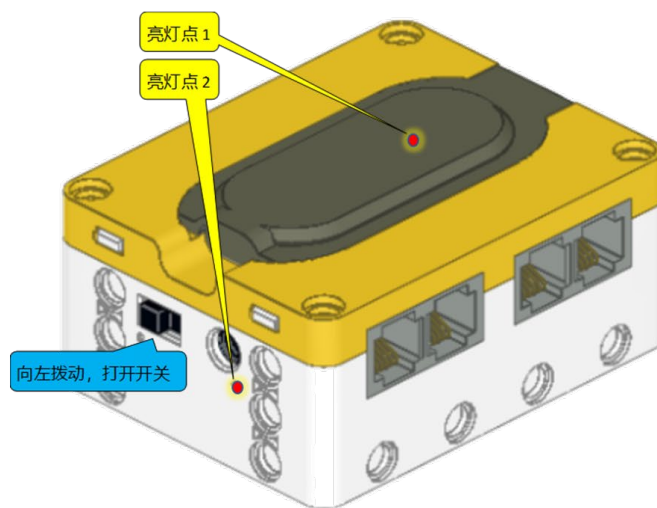


图 2-2 贝吉塔控制器电源指示灯亮

二、驱动安装

当程序编写完成后怎么能让电脑与主控产生联系呢？那就是安装驱动啦，有了这个信息，计算机就可以与设备进行通讯。可以说没有驱动程序，计算机硬件部分就无法正常工作了。下面让我们一起试试安装驱动吧！



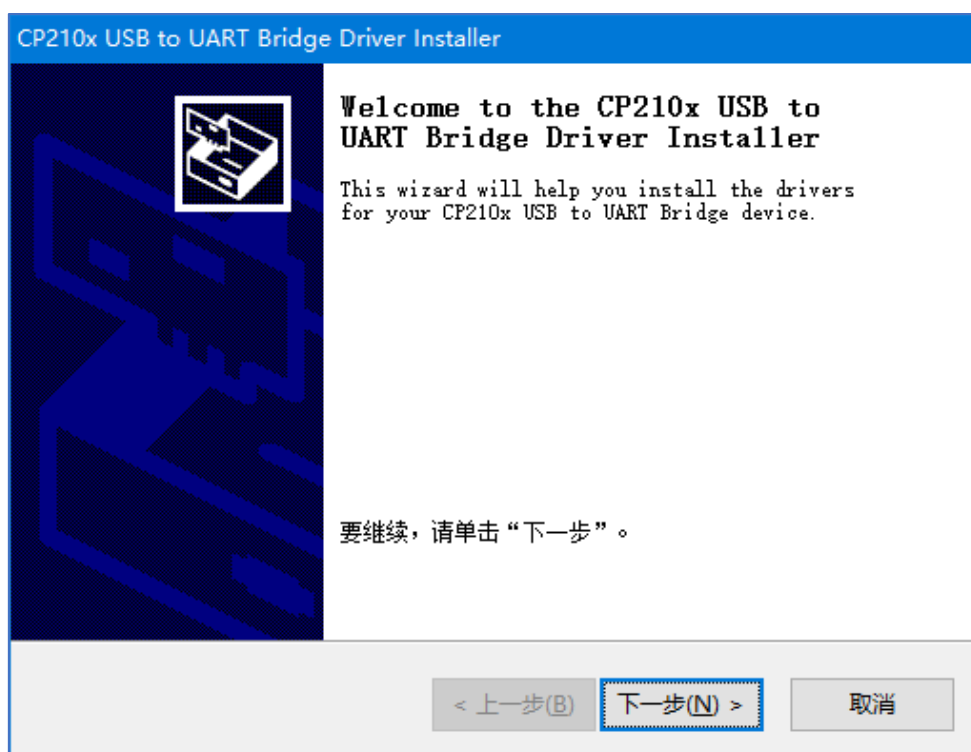
点击菜单“连接设备”->“安装驱动”：



小贴士

在一台电脑上，驱动只需安装一次即可，安装后，Scratch-vegeta 软件就能识别到控制器，以后使用就不需要再安装驱动了。

弹出如下窗口，按照提示完成安装即可。



三、串口连接

点击菜单下的“连接”->“串口”：COMx-CH340



在串口设备列表中选择想要连接的串口，点击“连接”，等待连接成功。

四、闪烁的 LED 灯

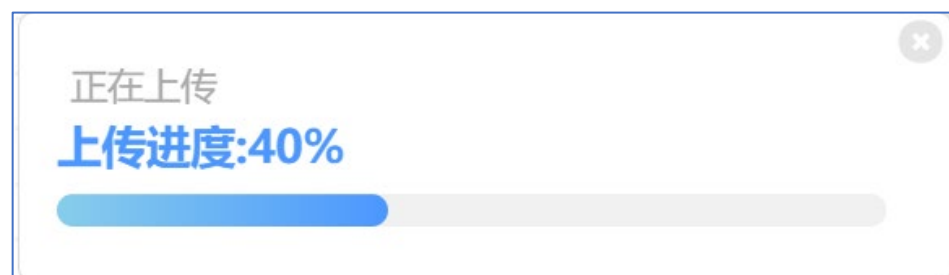
顺利连接电子积木后，就可以动手尝试编程控制 LED 灯了，一起动手编程来点亮它吧！



在默认的角色下进行编程：

脚本功能	脚本实现	操作说明
LED 每 0.5 秒钟 闪烁一次		在声光分类中找到 设置参数为“远程 1”，结合 完成程序。

点击 运行程序，等待上传完毕后观察 LED 灯是否开始闪烁：



小贴士

在对电子积木进行编程的时候，根据电子积木连接所在的端口，设置控制模块的端口参数，二者必须相对应。在本课示例中，LED 灯连接到了控制器的 1 号端口，所以设置模块的参数为“P1”。



尝试将 LED 灯连接到 2 号端口，修改程序，再次实现闪烁的 LED 灯。

五、按键控制 LED 灯

学会了如何编程控制“LED 灯”电子积木，是否能够自行尝试如何使用“触碰开关”电子积木？一起动手实现一个由开关控制的 LED 灯吧！当然，在开始编程前，先来整理下编程的思路是成功的保障。

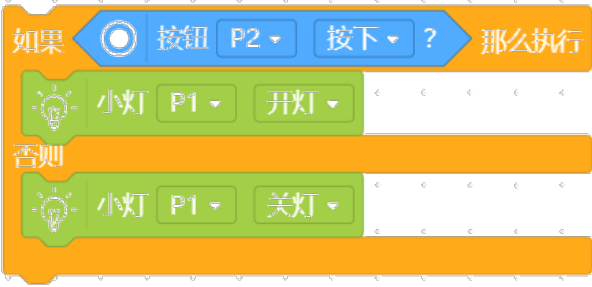
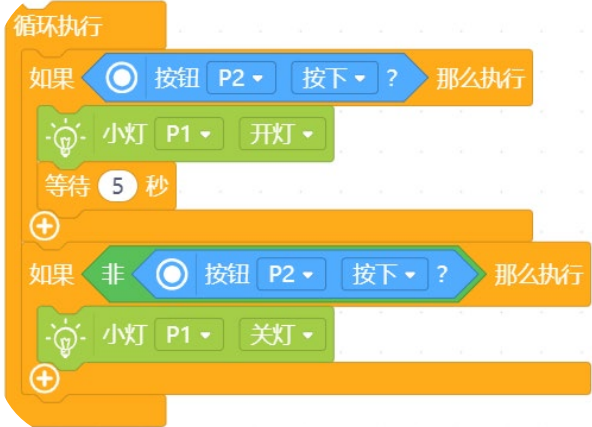
编程流程	流程说明
<pre>graph TD; Start([开始]) --> Decision{触碰开关是否被按下?}; Decision -- 是 --> On[点亮LED灯]; Decision -- 否 --> Off[熄灭LED灯]; On --> Decision; Off --> Decision;</pre>	开关控制 LED 灯的过程就是重复检测“触碰开关”的状态是否按下，如果按下就点亮 LED 灯，否则就熄灭。



在默认的角色下进行编程：

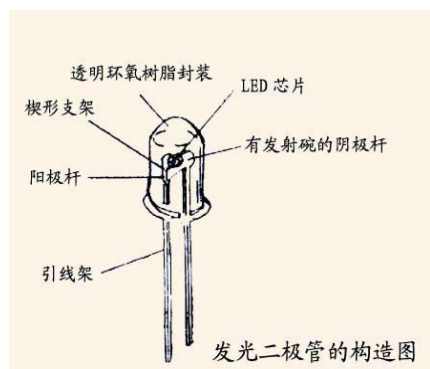
新增加按键模块，如图：



脚本功能	脚本实现	操作说明
根据按键状态控制 LED 灯		如果触碰开关被按下，就点亮 LED 灯，否则就熄灭 LED 灯。
重复检测开关状态实时控制 LED 灯亮灭		重复判断过程，实时控制 LED 灯状态
完成脚本		完成最终的脚本



1. 触碰开关按下并松开一秒后 LED 灯熄灭；
2. 触碰开关按一下 LED 灯点亮，再按一下 LED 灯熄灭。



LED 灯的学名叫“发光二极管”，顾名思义，就是会发光的二极管。它的基本结构是一块电致发光的半导体材料，置于一个有引线的架子上，然后四周用环氧树脂密封，用于保护内部结构。当有电流流过二极管时，就会把电能转换成光能，二极管就

发光了。

发光二极管有单向导电性，人体的血管就是只能从一个方向流到另一个方向，发光二极管也是如此，只能从正极流向负极。

第三课 交通灯

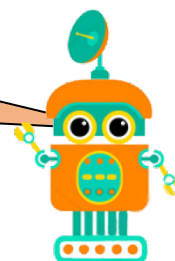


1. 观察交通灯的规则，思考其工作原理
2. 学会 RGB 灯的使用方法，了解光的三原色原理
3. 使用 Scratch-vegeta 编程实现交通灯的功能



我们的电子积木能不能做一个红绿灯呢？

很好的想法，当然可以，一起动手吧！



在古代的时候，灯语对人们的交流起到很大的帮助。现在的汽车也会经常打出“双闪”，以表示汽车可能出现故障或者其他情况。在十字路口，四面都悬挂着红、黄、绿、三色交通信号灯，它是不出声的“交通警察”。交通信号灯由红灯、绿灯、黄灯组成，红灯表示禁止通行，绿灯表示准许通行，黄灯表示警示。在这节课中，我们就用 Vegeta 编程实现一个交通灯。



电子积木名称	实例	电子积木说明
RGB 全彩灯	 彩色小灯 控制彩色小灯产生不同颜色、亮度等效果	该电子积木上有可以编程控制的灯，并可以根据光的三原色原理自由编程组合出任何你想要的颜色

一、硬件连接

找到如下材料：主控、彩色小灯、传感器连接线和白色数据线：

将彩色小灯模块连接到 1 号端口：

二、点亮 RGB 全彩灯

在每一个 RGB 全彩灯电子积木上，都能发出各种色彩的灯光。那么，如何点亮它呢？



将彩色小灯连接在远程控制板的 1 号端口上，使用串口连接远程控制板后，编写如下脚本：

脚本功能	脚本实现	操作说明
设置第一个灯为红色 设置第二个灯为绿色 设置第三个灯为蓝色		从声光模块中拖拽彩色小灯对应的模块到脚本区，设置端口参数为“P1”，亮度为 255

点击小绿旗运行程序，观察端口 1 上 RGB 全彩灯的效果。



修改彩色小灯参数的值，观察光线的色彩如何变化。

小贴士

RGB 的有效范围在 0 到 255 之间，分别表示红色、绿色、蓝色三个颜色分量的值，值越小，对应色彩亮度也会越小。根据光的三原色原理利用红绿蓝分量的不同，可以组合出任何你想要的颜色。

三、五彩斑斓

我们已经成功点亮了“RGB 全彩灯”，并学会了如何控制灯的颜色，接下来，我们利用“RGB 全彩灯”来实现五彩斑斓的灯光效果，体验 RGB 全彩灯的神奇之处。



尝试编写下面的脚本，观察 RGB 的效果。

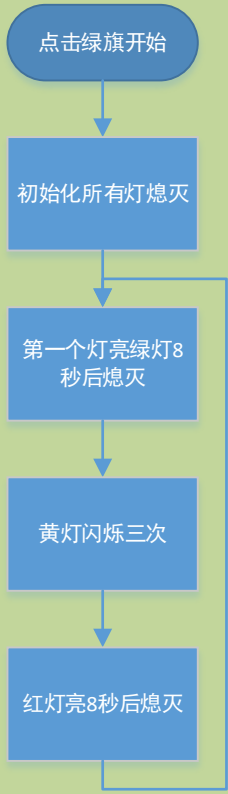
脚本功能	操作说明
让 RGB 全彩灯发出一个随机的颜色 每隔 0.5 秒变幻一种色彩	利用“随机数”分别设置 RGB 全彩灯三个颜色的亮度，从而产生一个随机的色彩 利用重复执行和等待功能实现每隔一段时间变幻一种色彩

四、交通灯的实现

为了实现交通灯，我们需要红色、绿色、黄色三种颜色的灯，RGB 全彩灯能轻易的在这几种颜色之间切换。



分析交通灯的逻辑，理解下面的流程图。

编程流程	流程说明
 <pre>graph TD; Start([点击绿旗开始]) --> Init[初始化所有灯熄灭]; Init --> Green[第一个灯亮绿灯8秒后熄灭]; Green --> Yellow[黄灯闪烁三次]; Yellow --> Red[红灯亮8秒后熄灭]; Red --> Green;</pre>	首先点亮第一个灯为绿色，8 秒后熄灭；继而设置为黄色，闪烁三次，然后转换为红色，等待 8 秒后熄灭，回到第一个绿灯，重复执行此逻辑。



在默认角色下进行编程：

脚本功能	脚本实现	操作说明
熄灭所有灯		程序开始后应该先进行初始化，清除上一次运行的结果。
点亮第一个灯		第一个灯为红灯，亮 5 秒
点亮第二个灯		第二个灯为黄灯，闪烁 6 次
点亮第三个灯		第三个灯为绿灯，亮 6 秒。
回到重复执行		重复执行，模拟交通灯效果



使用 RGB 全彩灯编程模拟警报灯的闪烁效果。



RGB 全彩灯的原理是通过光的红（R）、绿（G）、蓝（B）三原色之间不同的强度比例产生不同颜色。下图为 RGB 配色原理：



组合的颜色：

红+绿=黄（Yellow）

绿+蓝=青（Cyan）

红+蓝=品红（Magenta）

红+绿+蓝=白（White）

结合光的三原色原理，利用 RGB 全彩灯，尝试组合实现不同色彩的效果，验证光的三原色原理。

第四课 红灯停、绿灯行



- 1.RGB 灯的掌握及使用
- 2.红绿灯与动画相结合，创作场景故事动画
- 3.深入体会交通规则，融入场景，思考智能交通系统



马路上车水马龙，但他们都是很有规则的行驶着，这是为什么呢？对，就是因为有红绿灯的存在。红绿灯也被称为交通信号灯，一般由红灯、绿灯、黄灯组成。红灯表示禁止通行，绿灯表示准许通行，黄灯表示警示。当大家都听从红绿灯的指挥，遵

守交通规则，这样放学后我们才能安全的到家。那我们接下来就用 Vegeta 实现一个红绿灯，来指挥交通帮助行人安全过马路回家吧！



本节课程可分为 3 个阶段任务向学生进行讲解：

第一步，了解 RGB 灯的工作原理和拿出 RGB 全彩灯连接到远程 1 号端口上；

第二步，在实时模式下进行编程，首先，为舞台选择一个城市街道的背景，导入背景里面的户外分类中找到并选中 urban2 图片，添加角色“行人”；

第三步，切换到脚本区，为角色“行人”添加脚本，完成脚本编辑后，点击小绿旗运行看看效果吧。仔细观察红绿灯的变化，以及舞台上行人的动作，看看是否正确吧。



一、RGB 全彩灯

RGB 是从颜色发光的原理来设计定的，通俗点说它的颜色混合方式就好像有红、绿、蓝三盏灯，当它们的光相互叠合的时候，色彩相混，而亮度却等于两者亮度之总和，越混合亮度越高，即加法混合。

红、绿、蓝三色的叠加情况，中心三色最亮的叠加区为白色。

红、绿、蓝三个颜色通道每种色各分为 255 阶亮度，在 0 时“灯”最弱——是关掉的，而在 255 时“灯”最亮。当三色灰度数值相同时，产生不同灰度值的灰色调，即三色灰度都为 0 时，是最暗的黑色调；三色灰度都为 255 时，是最亮的白色调。

拿出 RGB 全彩灯连接到远程 1 号端口上。

二、添加背景角色

在实时模式下进行编程，首先，为舞台选择一个城市街道的背景，导入背景里面的户外分类中找到并选中 urban2 图片，添加角色“行人”。



The screenshot shows the Scratch stage with a city street background. A character named Jaime, a boy in a purple jacket and green pants, is walking across a crosswalk. The stage interface includes the following elements:

- 精灵 (Sprite):** Jaime
- 显示 (Display):** Visible (eye icon), Hidden (crossed-out eye icon)
- 大小 (Size):** 80
- 方向 (Direction):** 90
- 坐标 (Coordinates):** x: -84, y: -37
- 舞台 (Stage):** Background 2 (urban2 image)

三、交通灯下的行人

切换到脚本区，为角色“行人”编写脚本，利用 RGB 全彩灯模块设计出模拟过马路的画面，首先设定为熄灭所有灯，然后设定为点亮红灯，紧接是熄灭红灯点亮黄灯，最后是熄灭黄灯点亮绿灯，用动画完成整个红绿灯的使用过程。



完成脚本编辑后，点击小绿旗运行看看效果吧。仔细观察红绿灯的变化，以及舞台上行人的动作，看看是否正确吧。



程序最终运行图



1、通过这节课，我们学习了解了 RGB 工作原理和认识了红绿灯颜色缘由。利用 RGB 全彩灯模块编写脚本制作交通信号灯。

2、利用 RGB 全彩灯模块编写脚本制作交通信号灯。



1、完成这个作品之后，你应该掌握了红绿灯的工作原理和交通规则了吧！我们大家过马路都要遵守交通规则、注意安全。除了交通灯，我们的 LED 灯还能用到什么场合呢？大胆发挥你的想象力吧。

2、能否实现舞台灯？快速闪烁，制造绚丽的效果。和小朋友比赛看看谁做的更好看吧。

3、能否实现台灯的功能？按键按下，灯亮；再按一下，灯灭。

4、融入场景，思考智能交通系统，如何让交通信号灯可以检测到车流量多少来决定红绿灯的等待时间？

第五课 灵活小河马



1. 超声波传感器的掌握使用
2. 超声波传感器与动画的融合使用
3. 拓展超声波传感器模块



天气晴朗，鸟语花香的树林里，勤劳的小河马正在快乐的风来飞去，突然一个小皮球向小河马砸了过去，小河马跌落了在地上。小河马看见是一群调皮的孩子正在拿

皮球攻击它，心情很低落。你可以帮帮小河马躲过调皮的孩子的攻击吗？快使用 Vegeta 来实现小河马躲皮球的游戏吧。



本节课程可分为 4 个阶段任务向学生进行讲解：

第一步，将超声波传感器连接到远程 3 接口；

第二步，在实时模式下进行编程，首先，给舞台导入背景，然后，添加“小河马”和“小球”角色；

第三步，切换到脚本区，给“小河马”角色添加脚本；

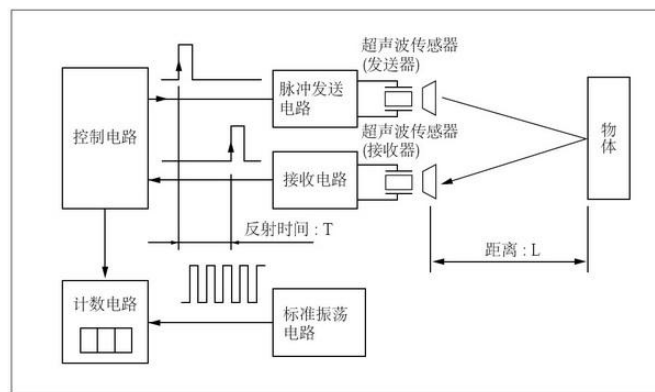
第四步，切换到角色“小球”，添加脚本，完成所有的脚本编辑后，点击小绿旗开始运行，看看能不能帮助小河马躲过调皮孩子的攻击。



一、认识超声波

声波是物体机械振动状态（或能量）的传播形式。超声波是指振动频率大于 20000Hz 以上的，其每秒的振动次数（频率）甚高，超出了人耳听觉的一般上限（20000Hz），人们将这种听不见的声波叫做超声波。

超声波传感器主要通过发送超声波并接受超声波来对某些参数或事项进行检测。发送超声波由发送器部分完成，利用振子的振动产生并向空中辐射超声波，接收超声波由接收器部分完成，主要接受由发送器辐射出的超声波并将其转换为电能输出；



二、添加背景角色造型

在实时模式下进行编程，首先，给舞台导入背景，然后，添加“小河马”和“小球”角色。如图所示。



图 5-1 导入角色添加背景界面

三、给小蜜蜂添加脚本

首先将小河马角色大小设定值设定为 50，选中将变量高度的值设定模块和超声波距离数值模块推算出小河马的飞行高度。根据以下脚本进行设定小蜜蜂是否能碰到小球，若碰到小球则游戏结束播放声音撞击模块。

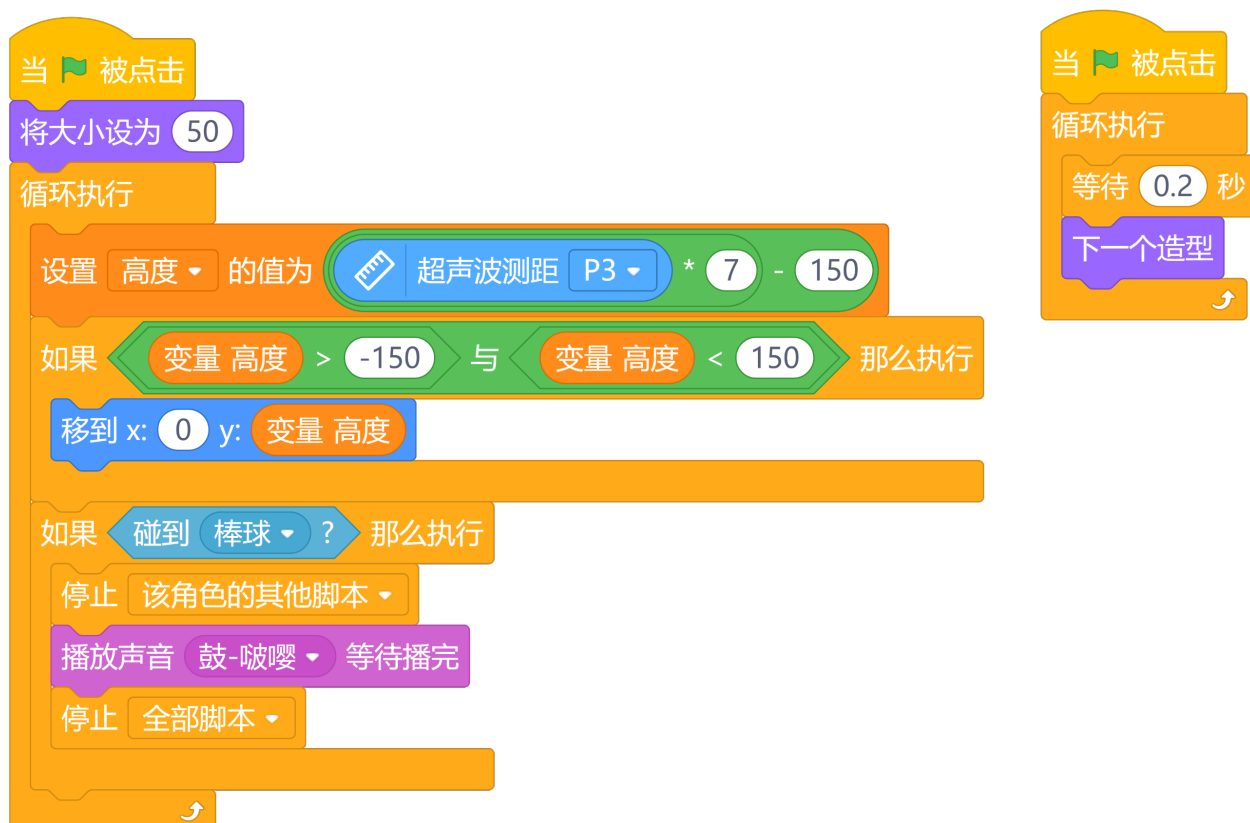


图 5-2 小河马脚本界面

四、给小球添加脚本并运行脚本

切换到角色“小球”，添加脚本，完成所有的脚本编辑后，点击小绿旗开始运行吧，看看能不能帮助小河马躲过调皮孩子的攻击。



图 5-3 小球脚本界面



总结汇报

1、通过这节课，我们学习了超声波是什么以及超声波传感器的原理，超声波的运用极其广泛，想一想在生活中哪些方面可以用到超声波。

2、运用超声波模块和其他变量值模块相结合来测算出小蜜蜂飞行高度以及帮助小蜜蜂躲避皮球的攻击。



举一反三

1、设计一个场景，用超声波模拟倒车的过程，在舞台上制作倒车的动画。

2、为盲人设计制作盲道障碍物提示器，根据障碍物距离发出不同声音提示。

第六课 打气球



- 1.深入认知按键传感器
- 2.学会角色的切换及声音的播放
- 3.按键传感器及动画的结合使用



气球是充满空气或某种别的气体的一种密封袋。气球不但可作为玩具，也可作为运输工具。如果气球足够大，里面气体又轻于同体积的空气，产生的浮力超过气囊和附带物体（如吊篮，热气球等）的重量时，气球就可上升。在生活中小朋友们都喜欢

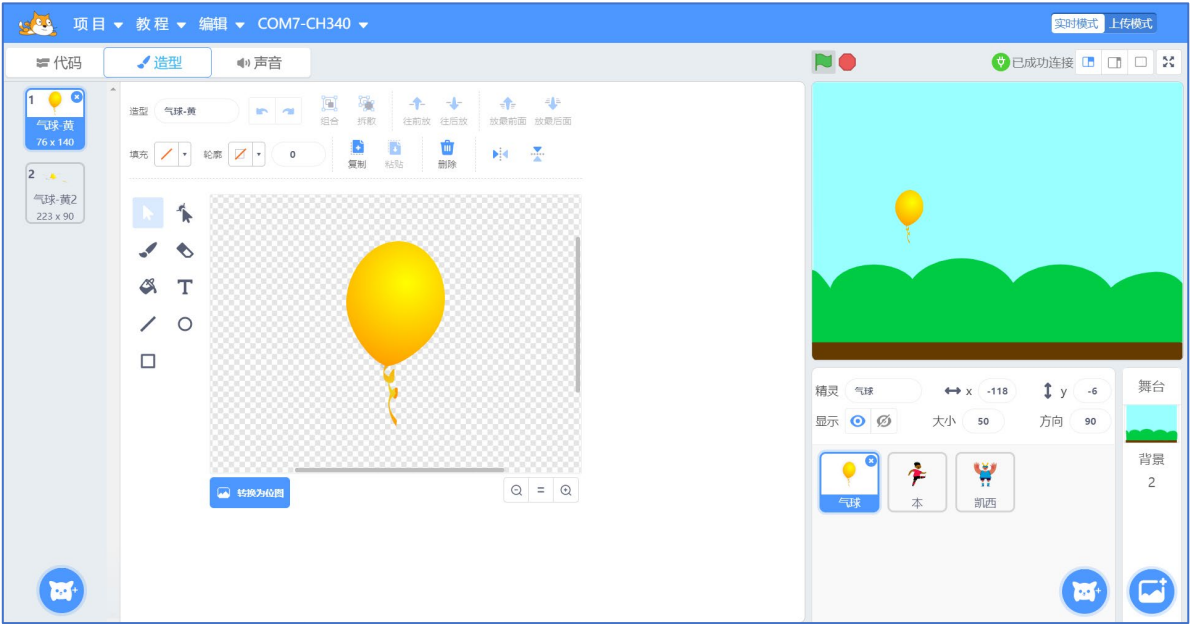
五颜六色的气球，能不能用编程软件实现吹气球呢？通过按钮，检测打气的过程，通过声音的大小，控制气球的大小，当气球大到一定程度，就会爆炸哦。这个可以用来测试小朋友们的肺活量呢，赶紧来试试吧！

一、声音传感器

电子积木名称	图例	用法说明
按钮/碰撞开关	 按钮/碰撞开关 读取按键状态，按键事件执行	使用   可以检测触碰开关的状态。

二、气球造型与爆炸

想要制作动画效果，就需在实时模式下进行编程。首先添加“气球”角色，分别导入“气球”和“爆炸”的造型（一个角色可以有很多造型），为下文将气球吹爆做铺垫：



三、爆炸气球之打气的过程

在脚本区，为“气球”角色添加脚本：

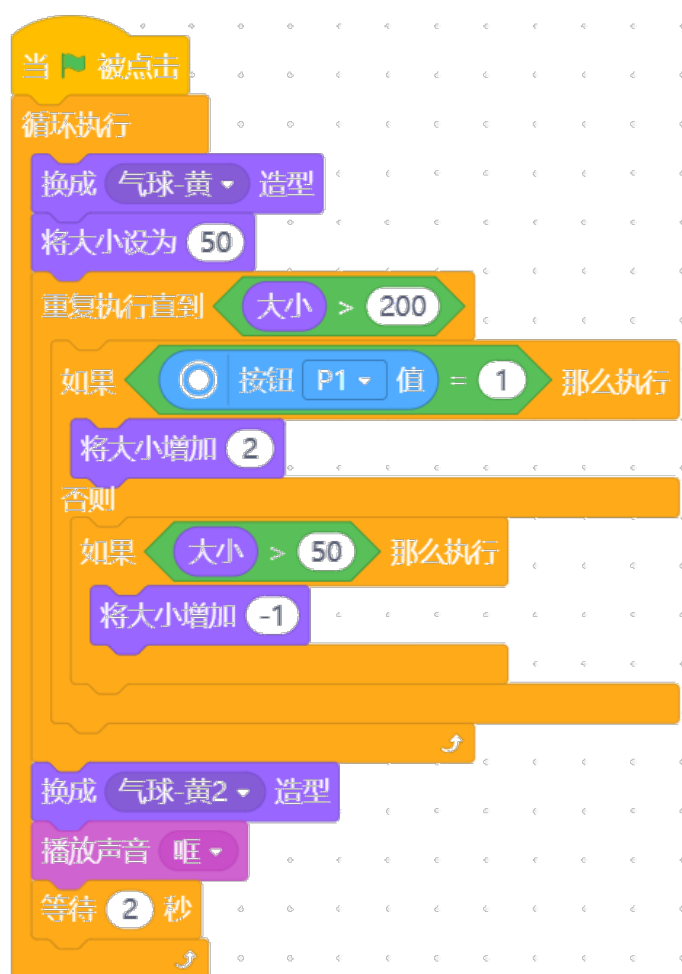


图 传感器感应及造型编程

程序编写完成，点击小绿旗开始运行。按压触碰开关，看看气球会如何变化？如果有问题，请仔细检查一下你的程序。当然，你也可以适当的调整一些参数，比如气球需要更大才会爆炸，让程序更符合你的设计要求。

Qichengjia

第三章 节

真实与生动

动力与结构

第一课 超声波台灯



1. 学习使用结构件搭建台灯造型；
2. 了解超声波传感器在生活中的用途；
3. 制作能够自动亮灭的感应台灯。



台灯是灯的一种，小巧精致，方便携带。此电器主要放置在写字台或餐桌上，以供照明之用。台灯的照射范围相对较小且集中，因此不会影响到整个房间的光线，光线作用于台灯周围，便于阅读、学习，节省能源。而智能感应台灯是一种通过感应模

块自动控制光源亮灭的新型智能照明产品。基于传感器的自动控制，当有人进入感应范围时，传感器探测到人和灯距离的变化，继而开关自动接通或断开。人不离开且在活动，开关持续导通；人离开后，开关自动关闭，实现人至灯亮，人离灯灭的功能，安全节能。本节课程我们就一起走进科技世界，通过学习编程软件，制作一台可以自动控制亮灭的感应台灯吧。



一、搭建台灯造型

利用套件中的乐高积木和传感器，观察书桌上小台灯的架构造型，自己设计搭建出智能台灯。

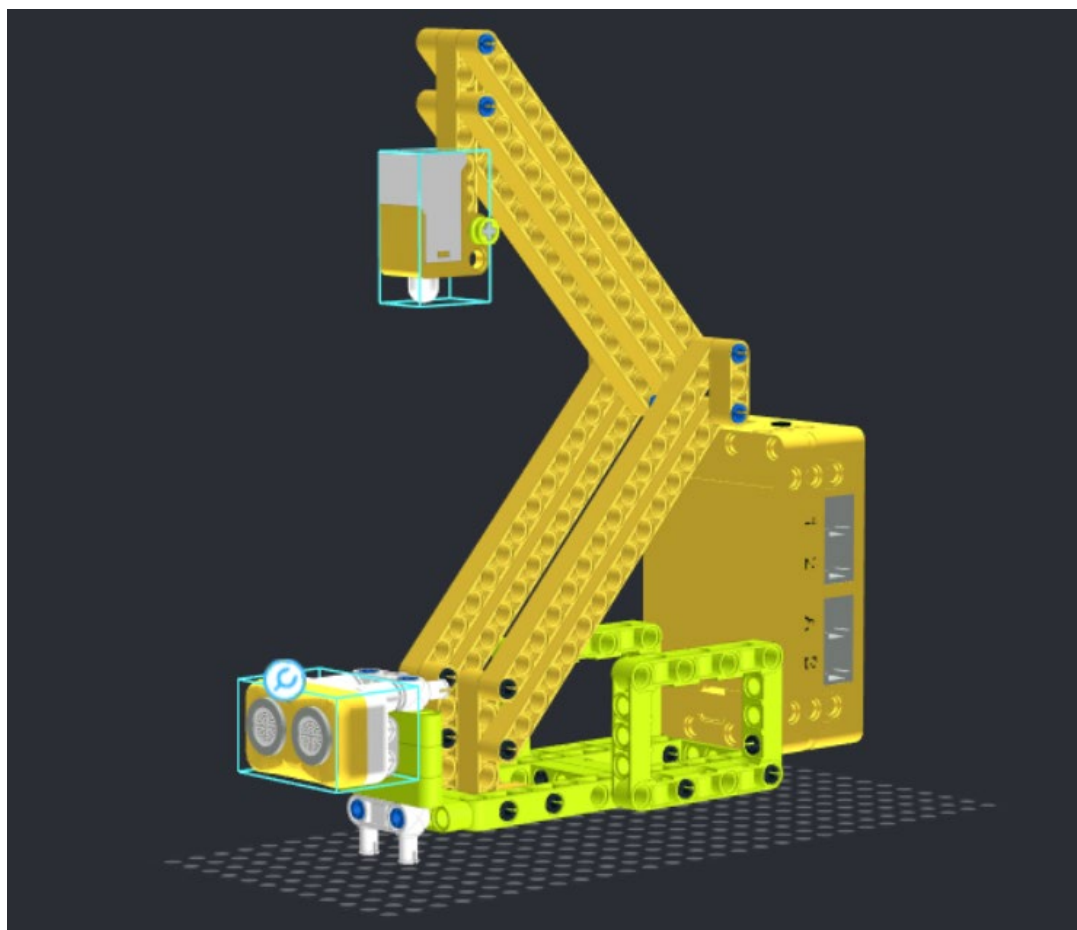


图 1-1 超声波台灯造型图片

搭建步骤、端口连接、注意事项请参考《可编程电子积木搭建步骤》。



本课所用到的电子积木如下表所示：

名称	图示	个数	作用
LED 灯	 VTMS0001 小灯 高亮照明LED，控制小灯打开、关闭、闪烁等	1	台灯的光源，通电使台灯发出光亮。
超声波传感器	 VTMS0027 超声波 感应角度 15°，检测距离 1cm 到 5 米，精度 3mm	1	感知人的距离以实现台灯的智能亮灭。

二、简易感应台灯

台灯造型搭建完毕后，利用编程对台灯的传感器进行编程，实现有人的时候台灯自动打开，人离开后自动熄灭的功能。



按照下表所示操作步骤，对搭建的台灯进行编程，实现感应台灯的功能：

脚本实现	操作说明
	初始化 RGB 灯熄灭
	如果超声波距离小于30cm，亮灯至少10秒，否则熄灭。实现有人来则开灯，人离开则关闭

三、实用感应台灯

通过测试感应台灯的效果，我们发现灯有时候会闪烁，可能是超声波因为障碍物移动而得到瞬间的错误值，可以对程序进行优化，添加误差处理算法：当发现距离大

于 30cm 时，重复检查 3 次，如果的确都是大于 30cm，说明人是真的离开，灯灭，否则持续亮灯。



切换到上传模式，按照下表所示编程过程，对感应台灯进行优化，解决超声波误差引起灯光闪烁的问题：

脚本功能	脚本实现	操作说明
当距离大于 30cm 时，重复检查 3 次加以确认		在否则脚本区中添加超声波重复检测判断，如果在重复检查时发现有小于 30cm 的数据，开灯，否则熄灭 RGB 灯。
确认离开后等待 3 秒关灯		在熄灭 RGB 灯前等待 3 秒，可以让人在离开时还有照明光源，更加合理安全。

串口连接控制板，点击“上传代码”功能将程序上传到控制板，用手挡在超声波传感器前方，观察 RGB 灯状态。

第二课 手持测距仪



- 1.深入认知超声波传感器及彩色小灯
- 2.了解仿生学的概念
- 3.拼搭简单结构，做出手持测距仪



生活中经常会出现对于距离、长度等的测量：房间高多少米？床的长宽是多少厘米？教室的长宽高是多少米？我们有很多测量方式，最常用的方式就是直接拉卷尺来进行测量，但卷尺的使用不方便。因此，在一些比较正规以及严肃的场合里面，对于

距离的测量，我们会用一些更加科学的方法，利用超声波测距仪测量就是这样的一种方法。今天我们也要通过零件和超声波传感器制造一个测距仪。

这个作品需要用到超声波传感器和彩色小灯两个电子模块。



本课围绕测距仪的组装进行设想，根据观察实物图片我们发现，测距仪需要将超声波传感器放置于前端，这样能更方便的对前方进行测量。测距仪上有一块可以显示数字的屏幕，但没有屏幕怎么办呢？我们可以利用上节课学习的 RGB 彩灯，将 RGB 彩灯安装在测距仪上，通过超声波测量的距离来控制 RGB 彩灯。



电子积木名称	实例	电子积木说明
超声波传感器	 VTMS0027 超声波 感应角度 15°，检测距离 1cm 到 5 米，精度 3mm	超声波传感器利用超声波发射原理实现距离探测，有效探测距离是 1cm 到 500cm。可连接 P3-8，精度为 3 密 mm。

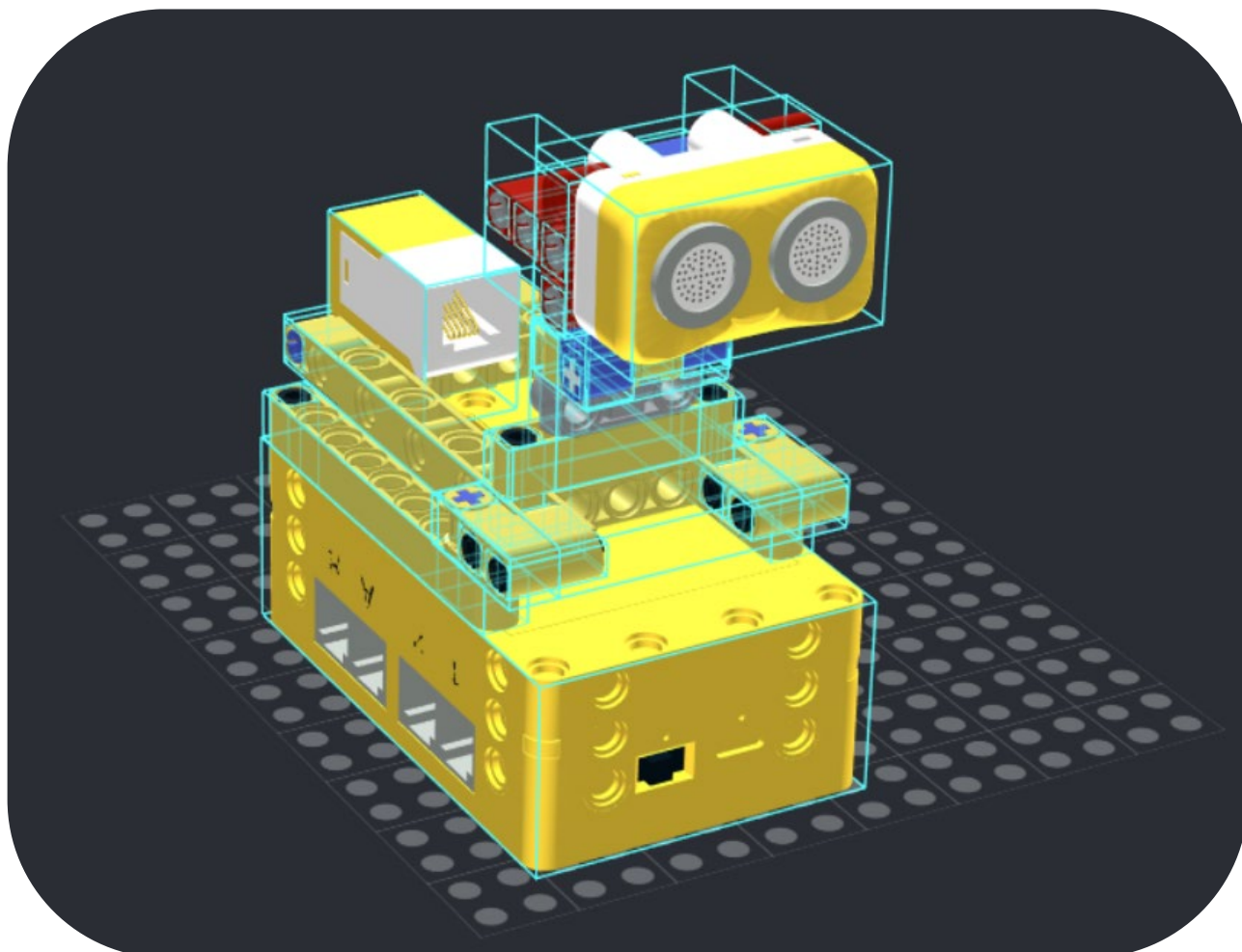
测距仪的功能：超声波测量的距离大于 30 时，RGB 彩灯亮绿灯。

超声波测量的距离小于 30 且大于 10 时，亮黄灯。

超声波测量的距离小于 10 时，RGB 彩灯亮红灯。

一、搭建结构造型

结合上节课所学完成结构的搭建，要求作品牢固稳定。



将“彩色小灯”连接到控制板的 P1 号端口，“超声波传”连接到 P3 号端口。在 Arduino 下对其编程。

作品外观造型没有限制，积极的大胆尝试，熟悉不同零件之间的造型组合，了解各部分材料的特点。

二、编程基础程序

编写如下“测距仪”的程序：



点击“上传程序”看看效果吧。拿着手持式测距仪，按下按键对着墙顶看看会有什么反应吧。



1、这节课，我们了解测量的本质和距离的概念。进而学习了超声波传感器的用原理和用法。

2、设计美观的手持结构。

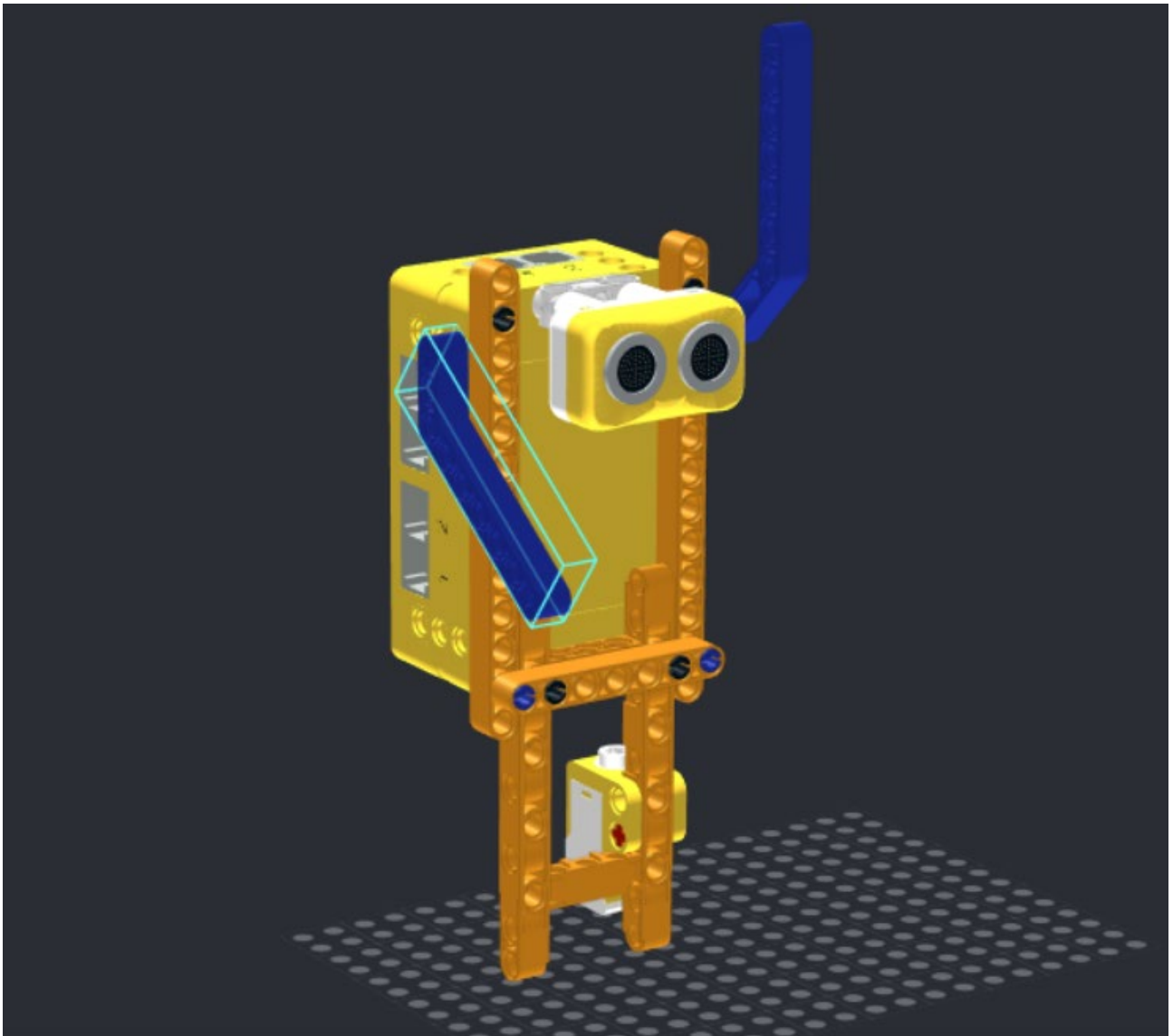
3、学习了传感器的结合。

三、总结与反思

时间到了，大家是不是都完成了测距仪的搭建和程序的编写了呀，接下来我们要梳理下这节课的学习内容。

- 1) 超声波的工作原理是什么？
- 2) 人类通过蝙蝠的超声波，创造了什么？
- 3) 学习超声波模块、“如果否则”逻辑模块的应用。
- 4) 学习运算类别中“与”模块的用法。

制作了手持式测距仪，解决了对触碰不到的地方的测量问题。想一想，你还能不能自己想到其他的类似应用呢？请尝试完成功能：按下按键，超声波反馈相应数值到软件中。



第三课 动力基础



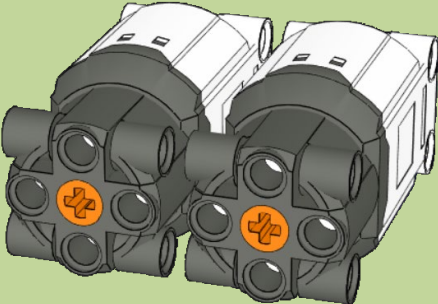
- 1. 学习掌握贝吉塔软件中直流电机的编程使用方法
- 2. 学习并尝试使用结构件搭建出你期望的作品样式
- 3. 学习单电机驱动作品的方式
- 4. 了解交互及烧录功能

电机（英文：Electric machinery，俗称“马达”）是指依据电磁感应定律实现电能转换或传递的一种电磁装置。

电机在电路中是用字母 M（旧标准用 D）表示，它的主要作用是产生驱动转矩，作为用电器或各种机械的动力源，发电机在电路中用字母 G 表示，它的主要作用是利用机械能转化为电能。



本课所用的电子积木如下表所示：

名称	图示	个数	作用
直流电机		2	带动传动机构旋转。 高力矩，速度连续可调。 转速： 0 ~ 200

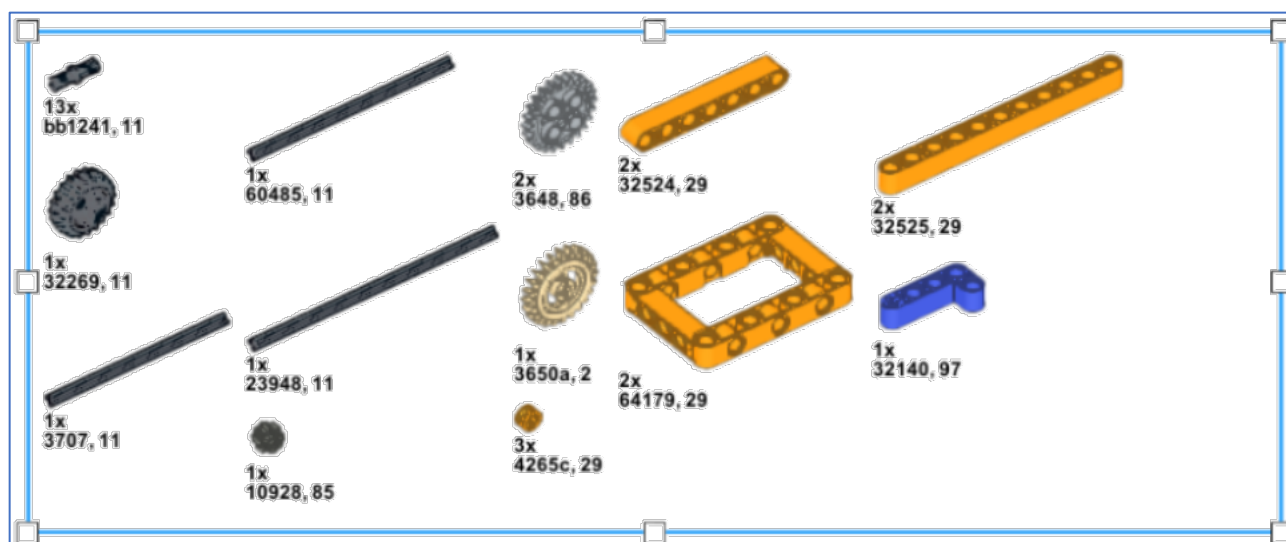
在接下来的课程内容中将会详细讲解结构与机械原理。

一、搅拌器



1. 学习理解齿轮传动的原理
2. 通过制作模型延伸理解齿轮的作用
3. 无限创意改善生活的理念培养

课前准备：教学基本信息——所需零件清单



手动打蛋器，用来将鸡蛋的蛋清和蛋黄打散充分融合成蛋液、单独将蛋清和蛋黄打到起泡的工具。可以简单而迅速地把蛋清和蛋黄打散充分融合成蛋液，以使用来制作蒸蛋。

使用手动打蛋器时，需要人们一直往同一方向快速搅拌蛋液，直到将蛋清搅拌至奶油状，相对于其他工具，手动搅拌器确实省了很大的力气，但搅拌需要快速、长时

间的操作，手部依旧会感到很酸痛。

为了解决这个问题，人们想到可以加入加速齿轮组，这样手腕转动速度就可以减慢，以减轻手部负担，最终设计人员还加上了把手，将手动打蛋器升级成为了图 4.2 中的手摇式打蛋器。在手摇式打蛋器中，我们加入了齿轮组，由大齿轮作为主动，啮合小齿轮。当我们转动大齿轮一周时，小齿轮却能转动好几周，从而搅拌叶转动几周。这样，就不需要我们迅速的转动手腕，也能将蛋清搅拌成奶油状。

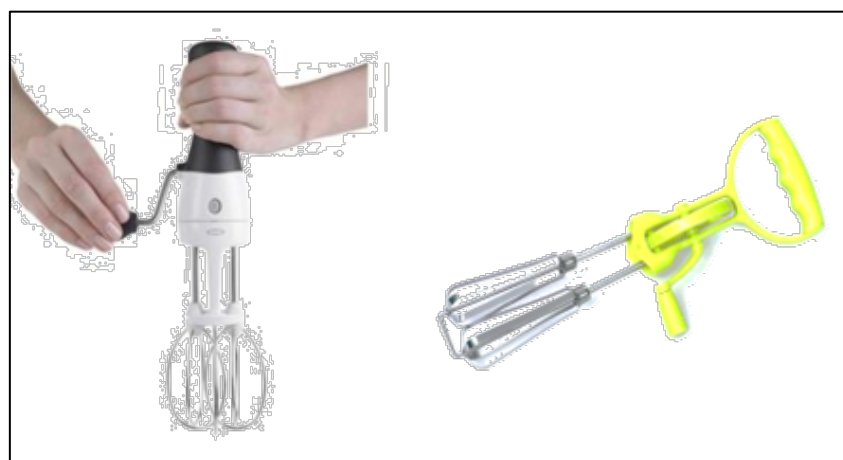


图 4-1 手摇式打蛋器



打蛋机是食品加工中常用的搅拌调和装置，用来搅打粘稠浆体，如糖浆、面浆、蛋液、乳酪等。工作原理：

- 1、打蛋机操作时搅拌器高速旋转，强制搅打，被搅拌的物料之间充分接触并剧烈摩擦，实现混合、乳化、充气及排除部分水分的作用。
- 2、调和物料的粘度，打蛋机的转速一般在 70~270r/min 范围内。
- 3、打蛋机为手持式，由搅拌器、转动装置等组成。

本课围绕打蛋器的组装设想、组装制作、设想改进展开教学活动，旨在引导学生了解打蛋器的结构与材料，讨论其加工及组装程序，并开展制作、评价和改进，以此

培养学生良好的组装制作习惯：

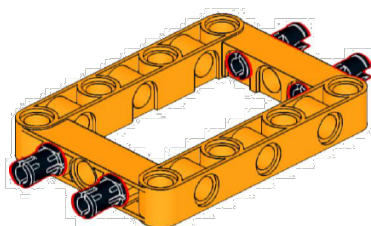
观察结构——了解材料——设想步骤——开展制作——试用评价——改进创新。

在充分观察的基础上引导学生认识打蛋器的结构，了解各部分材料的特点，小组讨论组装步骤，然后按步制作。在教学过程中，结构造型为次要的，主要的是教导齿轮传动的知识原理。在组装过程中，如果提出技术操作难度较大，可由个例示范操作，教师进行技术要点指导。完成作品后，由各小组开展自评和互评，并针对各自的“打蛋器”开展反思，提出改进设想，并各小组之间交流。

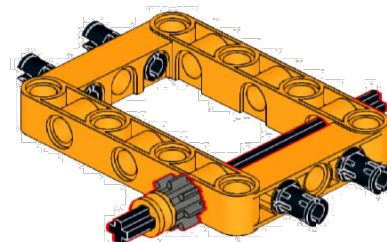
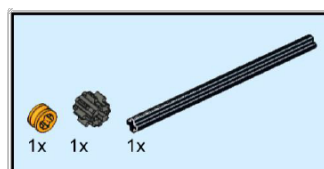


手动打蛋应该会很辛苦，蛋糕店一天要做很多很多蛋糕是不是很累啊？那我们有什么办法可以省力一点呢？现在我们来解密怎么样才可以让我的打蛋器更加省力吧：

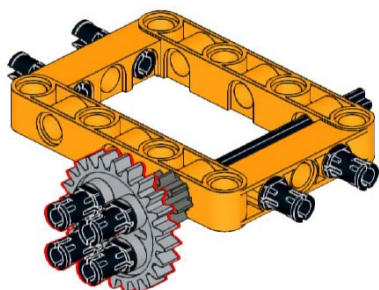
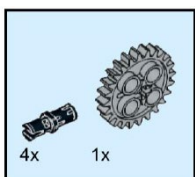
建构 (construct)



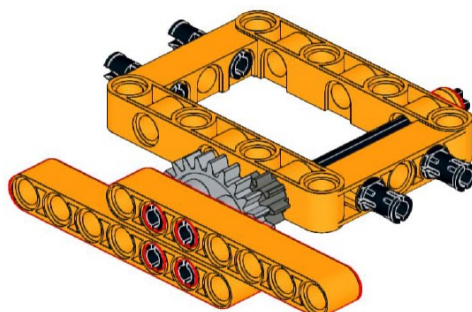
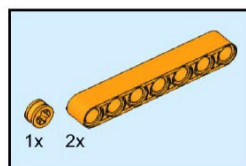
初步搭建的时候要知道上下左右前后的顺序，把作品分成几个部分搭建，每一个部分要清楚搭建的是什么东西。准备一个 5*7 特殊圈梁臂和四个黑色摩擦销（如图）连接



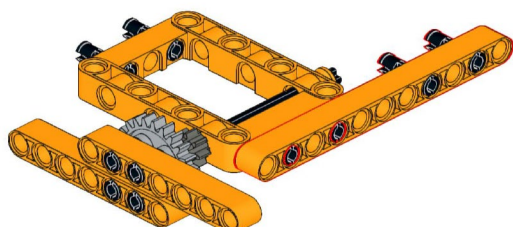
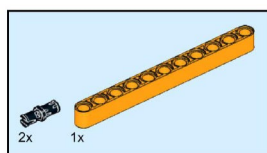
准备一根 8 单位轴，一个半轴套和一个 8 齿圆柱齿轮（如图）方式连接，注意连接的时候一定要固定牢固



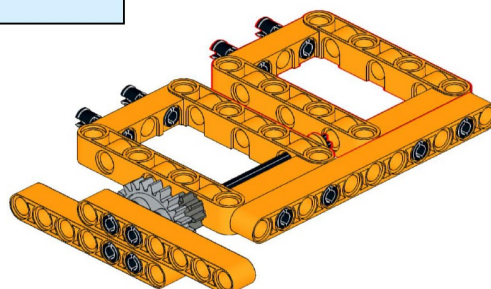
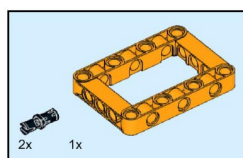
准备一个半轴套，一个 24 齿圆柱齿轮和四个黑色摩擦销，（如图）连接，注意先将半轴套连接在 8 齿圆柱齿轮的一侧。



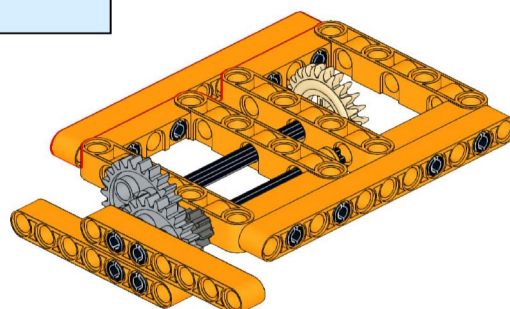
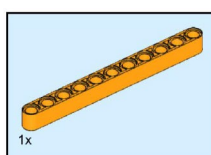
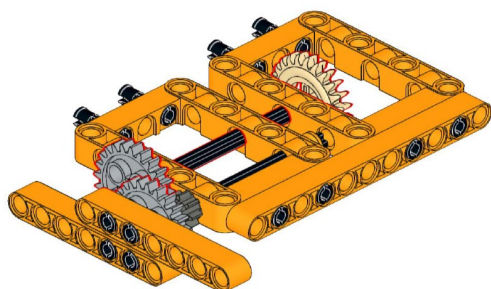
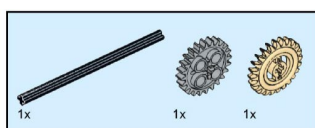
准备两个 7 孔洞梁（如图）连接



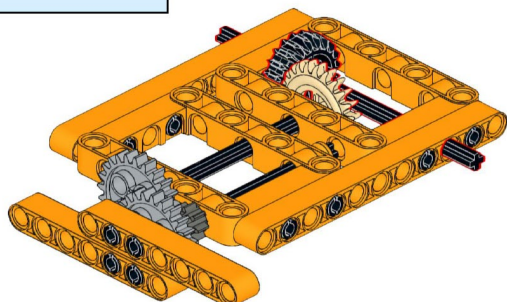
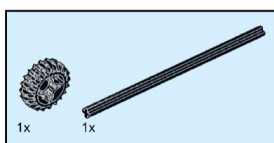
准备一根 11 孔洞梁和两个黑色摩擦销连接，注意黑色摩擦销安装在 11 孔洞梁的内侧



准备一个 5*7 特殊圈梁臂和两个黑色摩擦销（连接）。

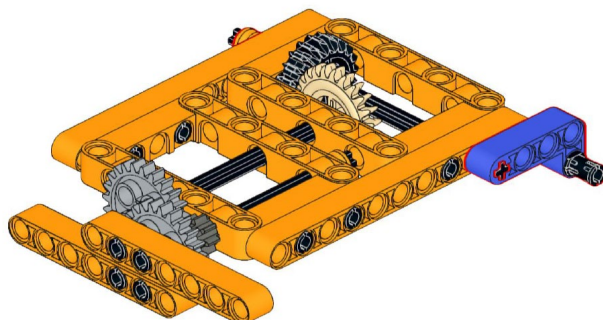
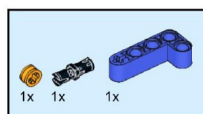


一个 24 齿圆柱齿轮和一个 24 齿冠齿齿轮进行连接。确保 24 齿圆柱齿轮与 8 齿圆柱齿轮啮合。

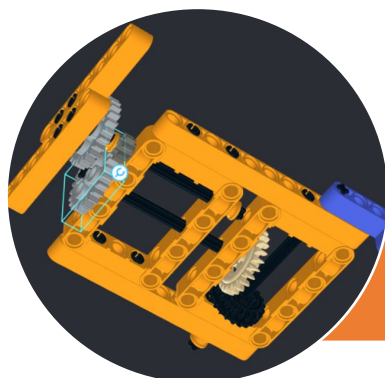


准备一个 8 单位轴和一个 20 齿双面斜齿轮（如图 9）进行连接，确保 20 齿双面斜齿轮和 24 齿冠齿齿轮垂直啮合。

准备一根 10 单位轴，准备一个一根 11 孔洞梁连接，注意这根 11 孔洞梁起到固定作用。



准备一个 2*4 直角洞梁和一个半轴套完成最后的固定，完成自己的作品，尝试活动一下自己的搅拌器。



搅拌器搭建

齿轮讲解：

齿轮是一种轮缘上有齿且能连续啮合传递运动和动力的机械零件。齿轮上每一个用于啮合的凸起部分称为轮齿，一般叫做齿。这些凸起部分一般呈辐射状排列，配对齿轮上的轮齿互相接触，可使齿轮持续啮合运转。在齿轮的整个圆周上轮齿的总数称为齿数。



齿轮啮合

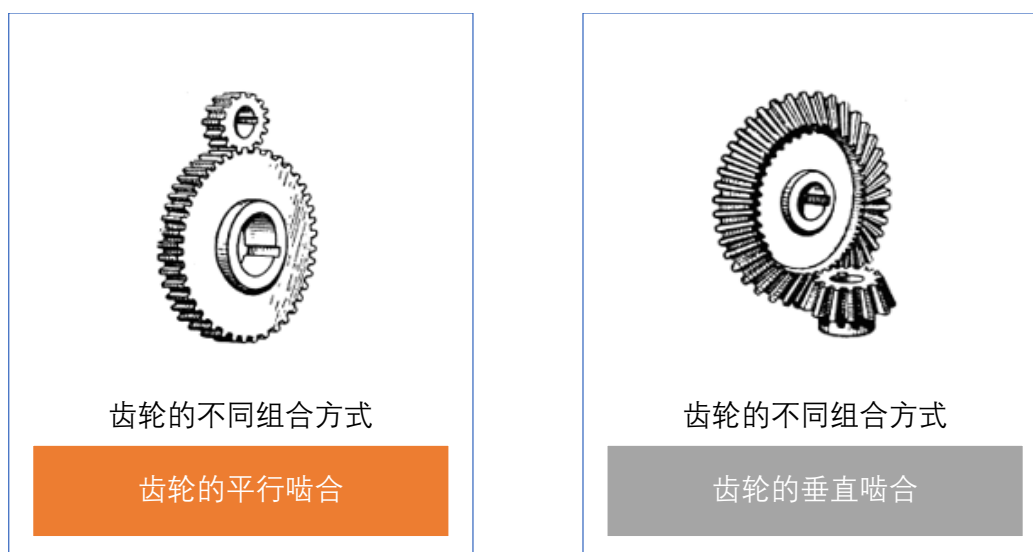
齿轮的最终目的是将发动机的特点通过最佳方式改造成我们想要的样子。传输动

力实际上是这个过程的一个边际效应。

能够相互啮合的轮齿必须是一样大小的，所以相互啮合的齿轮中直径大的齿数就会多些，直径小的齿数就会少一些。但齿轮在转动中转过的齿数是一样的，直径大的齿轮传动一周，直径小的齿轮会转动好几周，所以直径大的齿轮带动直径小的齿轮时，转速会变快，被称为加速齿轮装置。

齿轮啮合时可以平行啮合，也可以垂直啮合，对应是搅拌机中框架靠近叶片一端的两个齿轮，而垂直啮合，对应的是框架中横轴和竖轴交叉处的两个齿轮。

注意：平行啮合时，齿轮转动的方向在同一平面上，转动方向相反。垂直啮合时，齿轮传动方向在垂直的两个平面上，转动方向改变了，而力的方向也改变了。



齿轮传动的原理介绍：齿轮传动是机械传动中应用最广的传动装置，在所有的传动中占有最重要的地位。

1. 齿轮传动是动力传动的一种形式

2. 齿轮传动原理很简单。即一对相同模数（齿的形体）的齿轮相互啮合将动力由“甲轴”传送（递）给“乙轴”，完成动力传递的一种方式

3. 它的特点是：动力传递可靠、速度比可通过调整相对齿数来改变机械传动机构，可以将动力所提供的运动的方式、方向或速度加以改变，被人们有目的地加以利用

用。中国古代传动机构类型很多，应用很广，除了上面介绍的以外，像地动仪、鼓风机等等，都是机械传动机构的产物。中国古代传动机构主要有齿轮传动、绳带传动和链传动。

齿轮传动其出现时间不晚于西汉，西汉时的指南车、记里鼓车，东汉张衡发明的水力天文仪器上，都使用了相当复杂的齿轮传动系统。这些齿轮只用来传递运动，强度要求不高。至于生产上所采用的齿轮，要传递较大的动力，受力一般较大，强度要求较高。古代在利用畜力、水力和风力进行提水、粮食加工等工作时，都要应用此类齿轮。

轮轴：在搅拌机中还有一个轮轴的结构，那就是横轴上的把手，这种能绕共同轴线旋转的机械，都叫做轮轴。在轮轴中外环叫轮，内环叫轴。轮轴两个环是同心圆。

轮轴的实质实际是一个能够连续旋转的杠杆。这个杠杆的支点就在轴线上，轮轴在转动时轮与轴有相同的转速，动力点就是轮的外圆(搅拌机中就是把手的地方，动力臂就是把手到轴中心的距离)，而阻力点就是轴的外圆(阻力臂为轴的半径)，所以当动力作用在轮上，则轮轴为省力杠杆，轮和轴的半径相差地大则越省力，但越费距离。动力作用在轴上则轮轴为费力杠杆，轮和轴的半径相差越大则越费力，但更省距离。

齿轮种类：齿轮的种类多种多样，按照外形分一般分为平齿轮，冠齿轮和蜗轮。

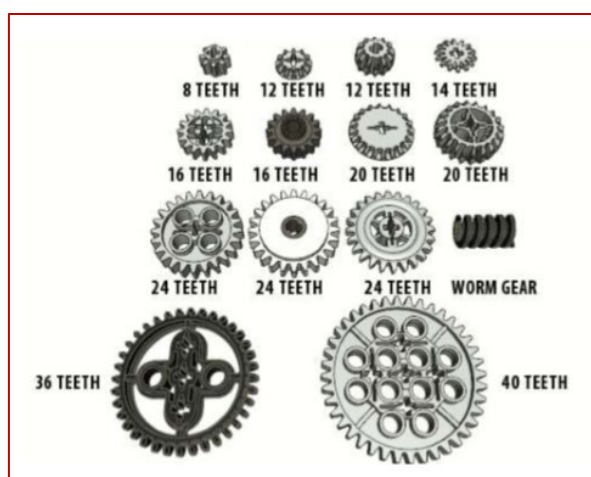


图 4-2 齿轮的种类

如图 4-2，这里有 13 种传统的圆形齿轮，和一种特殊的齿轮称为蜗杆。此外，圆形齿轮可以分为两组：拥有方形齿的常规齿轮和拥有圆形齿的斜面齿轮。这两组齿轮中的任何齿轮均可互相啮合。斜面齿轮的特殊之处在于它们即可水平啮合也可垂直啮合。由于尺寸合适，它们可以非常方便的和带孔洞梁一同使用。

之前我们用到的都属于平齿轮，还有一种齿轮称为冠齿轮，冠齿轮的齿分布在端面，而不是在轮缘上，一般用在机械的变速、换向离合器上，在玩具上也有广泛的应用。

而蜗轮上的齿看起来好像有点偏，并不是垂直于整个轮而的。这是因为蜗轮的用法和平齿轮的用法不太一样，蜗轮通常是和蜗杆一起使用的，如图 4-3 所示。

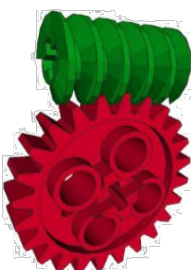


图 4-3 蜗轮与蜗杆

蜗轮蜗杆结构常用来传递两交错轴之间的运动和动力。蜗杆上有和螺丝上相似的螺旋，螺丝上的螺旋转动的时候会让螺丝进入被固定件，而蜗杆上的螺旋转动时会带

动蜗轮转动。蜗轮与蜗杆在中间平面内有点像齿轮与齿条。

蜗轮蜗杆结构有一个其他齿轮组所不具备的特性：蜗杆可以轻易转动蜗轮，但蜗轮无法转动蜗杆。这是因为当蜗轮尝试旋转蜗杆时，蜗轮与蜗杆之间的摩擦力会让蜗杆保持原位。



按照搭建步骤完成自己的作品后，尝试活动一下新搭建出的搅拌器。注意：本课程所有案例均为展示示例，鼓励并欢迎各位在此基础上修改优化。

【课堂讨论】我们的搅拌器使用到的是什么齿轮原理呢？

①小组成员之间相互交流、评价（从结构强度、稳定性、使用材料、连接方式以及美观、经济等角度进行交流和优化）

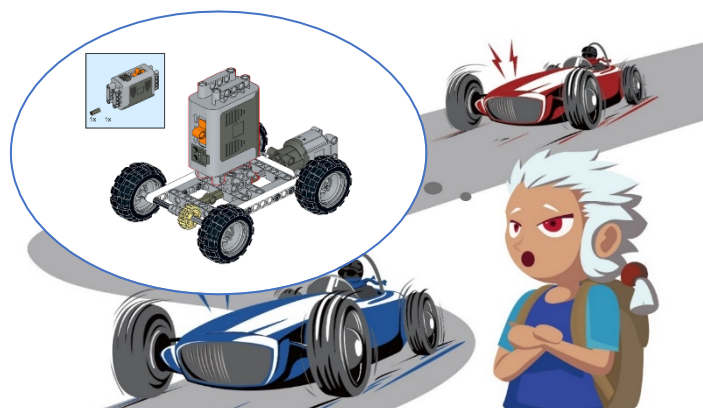
②大组交流、评价（要求每个小组选出一个代表，用几句话说出自己设计的特点。）

思考：玩一下我们的搅拌器想一下，满足搅拌器运行需要几个条件？

二、后轮驱动车



1. 研究汽车上的轮轴作用，并尝试使用轮轴解决问题
2. 学习了解流线型车体结构并搭建后驱车
3. 对马达的驱动结构有更深刻地掌握并利用齿轮加速作为后轮驱动



后轮驱动车就是发动机驱动后轮来负责进后退，前轮负责承重和转向的汽车。在这驱动形式中，发动机输出的动力全部输送到驱动桥上，也就是说实际上是后轮“推动”前轮，带动车辆移动。

我们生活中的汽车大部分都是后轮驱马车，汽车作为一种代步工具，给人们提供了很大的便利。在汽车漫长发展史中，汽车通过不断的改进与创新才演变成今天的样子。

轮子的使用能够使物体与地面的摩擦从滑动摩擦转变为滚动摩擦，这样就大大减小了摩擦力，不仅减少了磨损，而且随着阻力的减小，整个物体的运动也更加顺畅了。

最开始的时候，轮子都是木质的，后来发展为橡胶，再后来这些橡胶的轮子上还出现了很多花纹。



图 5-1 木轮车



设计并搭建：

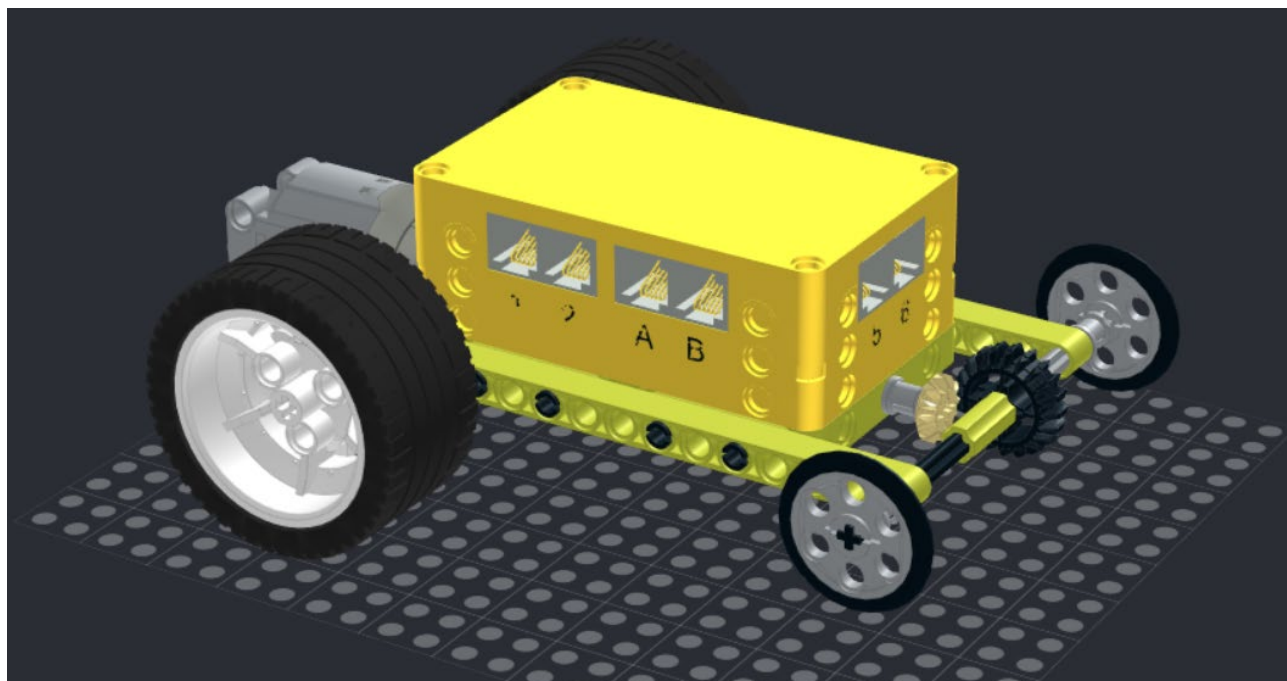
- I. 这辆后驱车有四只轮胎，一个发动机
- II. 发动机启动，后驱车可以直线运行
- III. 后驱车要求具备车体结构标准和较低的底盘

注意：孩子们对车体有很多自己的想法并充满了想象，但是一定注意器材使用的合理性，用有限的物料解决问题。

首先制作车的框架。我们实际上制作的就是车的底盘，具体制作中要考虑轮子和电机安装的位置。其次安装电机。电机安装在车的中后部。继而制作传动装置。安装车轮，完成模型。我们的车轮直接固定在前后的轴上。最后安装电源，控制模型。



下面我们来制作一个后轮驱动车。



底盘是用于支承、安装发动机及其各部件、总成，形成汽车的整体造型，并接受发动机的动力，使汽车产生运动，保证正常行驶的整体。它由传动机构、行驶机构、转向机构和制动机构四部分组成。也就是说除了引擎、轮胎和车身外壳，其余汽车部件几乎全部被纳入底盘的范畴。

摩擦力：两个相互接触的物体，当它们发生相对运动或者具有相对运动趋势时，在接触面上会产生一种阻碍相对运动的力，这种力叫做摩擦力。摩擦力的产生需要两个条件，第一是至少需要两个相互接触的物体，第二是物体之间要发生相对运动或者是相对运动趋势。而摩擦力的方向是阻碍物体运动的方向，根据物体的运动状态又分为静摩擦力、滑动摩擦力以及滚动摩擦力。

静摩擦力：当相互接触的物体之间没有相对移动，只有相对运动趋势时，在接触面上产生的摩擦力称为静摩擦力。静摩擦力的大小是不能直接得到的，不过我们可以根据力的平衡来计算。静摩擦时，虽然有运动趋势，但没有运动，所以物体是相对静止的，静止的物体是受力平衡的，所以可以根据力的平衡条件求得静摩擦力，这需

要知道物体上的其他力的大小和方向。而静摩擦力的方向是与其它作用在物体上力的总和的方向相反的。

当物体上的外力过大，超过接触面能产生的最大静摩擦力时，物体就会发生相对的滑动，此时摩擦力变为滑动摩擦力。最大静摩擦的大小要大于滑动摩擦力。

滑动摩擦力：两个相互接触的物体，当它们发生相对运动时，在接触面上会产生一种阻碍相对运动的力称为滑动摩擦力。滑动摩擦力的大小等于摩擦因数乘以压力，而摩擦因素与物体表面的粗糙程度有关。

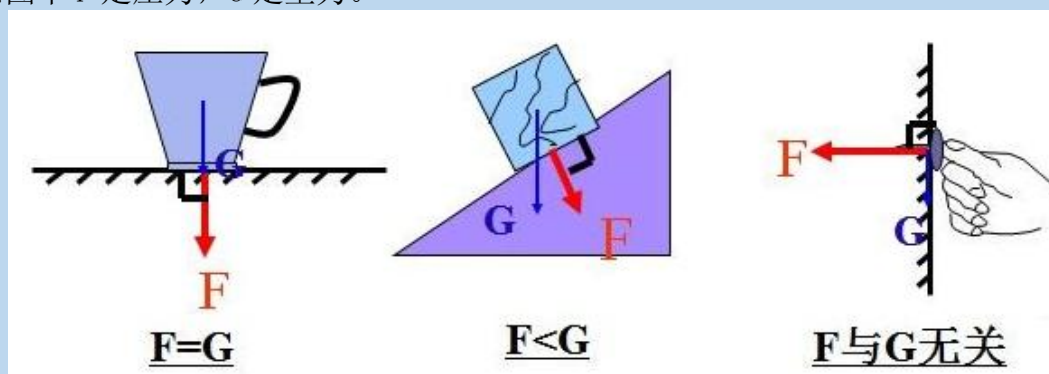
我们在模型制作中会用到一些圆形的成片状的垫片，一般垫在框架与转动装置之间。垫片的作用其实就是为了减小框架与转动装置之间的滑动摩擦力。

滚动摩擦力：当物体在另一物体表面做无滑动的滚动或有滚动的趋势时，在接触面上产生的阻碍滚动的摩擦力称为滚动摩擦力。

前面我们说现在的轮胎上都有花纹，这些花纹主要是为了让轮胎增加与地面之间的摩擦，防止打滑。

一般情况下，物体之间的滚动摩擦力远小于滑动摩擦力，只有滑动摩擦力的 $1/40 \sim 1/60$ ，所以在地面滚动物体比推着物体滑动省力得多。

注意：压力是垂直于接触面的力，而不是地面。
如图中F是压力，G是重力。



前驱和后驱：我们生活中的汽车基本分为前轮驱动、后轮驱动和四轮驱动。

前驱	后驱
前轮驱动的优点	1. 空间：由于动力系统都位于汽车前部的发动机舱中，省去了通往后轮的驱动轴和后差动器，不会占用空间，因此增大了车身内部的容积。 2. 成本较低：减少零部件的数量，所以成本较低。 3. 重量也随之减轻。 4. 提高燃油效率。 5. 提高了安装的效率。 6. 轮胎可以获得更多的牵引力(抓地力)。 7. 更稳定。
前轮驱动的缺点	● 重心分布：前轮驱动车的重心通常较后轮驱动车更偏离车体的中间。前轴往往会负担大部分的车重，与通常认为的理想前后轮平均分布有很大距离，这也是导致转向不足的原因。 ● 转弯半径：由于前轮兼有转向和驱动的功能，轮胎转向的角度可能因此受到限制。通常后轮驱动的车辆可以有更小的转弯半径。 ● 加速性：启动加速时，后轮驱动车的车重向后传递，从而加大后轮的牵引力。而前轮驱动车则没有这个条件，这也是几乎所有赛车都采用后轮驱动的原因。
后轮驱动的优点	1. 加速性：在启动加速时，车重向后传递，增加了后轮的压力，从而使得后轮获得更多的抓地力，提高了加速性。 2. 在机械上相对于前轮驱动车更简单，较易于拆卸和维护。 3. 没有前轮扭矩分配不均可能导致的偏向问题。

	4. 提高汽车的操控性。 5. 后轮驱动车通常有更小的转弯半径。 6. 有更好的转向性能和拖力。
后轮驱动的缺点	● 操控性：与前轮驱动相比，后轮驱动车容易出现转向过度的情况，一般驾驶技术的人较难控制。 ● 内部空间：对轿车而言，由于后轮驱动车的发动机和变速箱位置靠后，使得前座乘客的空间受到压缩。另外由于有通向后轮的传动轴，后座中间下方地板必须抬高，还有可能会使后备厢的容积减少。 ● 车重：尽管后轮驱动的机械设计较简单但是由于部件分布更分散的关系，动力系统的总重量增加。 ● 价格：由于在组装时，动力系统不能像前轮驱动车那样自成一体，在工作效率上较差，因此增加了组装成本。另外，动力系统的材料费用也较多。因此，一般后轮驱动车比同级别的前轮驱动车略贵。 ● 在力学中有得必有失，获得了速度，但同时也减少了扭矩，简单的说就是在力量上的损失，同时转化为速度——速度越快，扭矩就越小。如果获得了三倍的角速度，你的扭矩会减小到原来的 $1/3$。齿轮中有一个有趣的特性：扭矩和速度的转换是对称的，你可以将扭矩转换成速度，反之亦然。当系统增加速度而减小扭矩时，我们称为加速，反之我们称为减速。

chǐhuò

三、机械尺蠖



1. 掌握毛毛虫爬行的原理
2. 学会使用连杆结构解决问题
3. 锥形齿轮减速并改变旋转平面
4. 利用杠杆结构来驱动毛毛虫身体蠕动的方式
5. 棘轮对移动轮进行方向锁定



大部分女学生们都对毛毛虫充满了恐惧。尽管和虫子相比，这些女孩算得上是庞然大物，但是每当看到它们时，庞然大物们却总是发出毛骨悚然的尖叫。

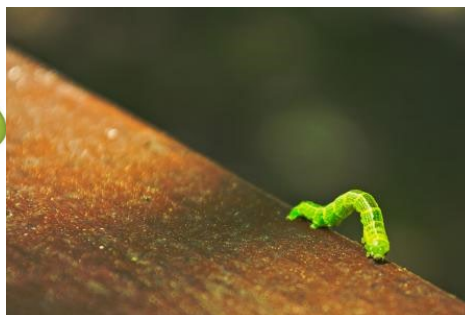


图 1 尺蠖

对于行动迟缓，没有翅膀的毛虫来说，生存就是一场战争。然而幸运的是，这些聪明的爬行动物精通伪装和防卫。研究发现，不同种类的毛虫都带有一对眼睛状的标记，且颜色和形状多种多样，有圆的也有狭长的瞳孔形。这些各式各样的图案表明毛虫们并没有刻意针对某种掠食者进行拟态伪装，以逃脱被捕食。那你有没有观察过毛毛虫是如何爬行的呢？本次课程就让我们走进仿生学的世界，来了解毛毛虫的运动方

式吧~

毛毛虫的身体特征给了仿生学家们极大的启示，着手创造一种毛毛虫机器人，不仅能够像活的毛毛虫一样灵巧，身体也能自由弯曲并且恢复原样，还可以沿路爬行。科学家们希望毛毛虫机器人可以在未来时代大放异彩，比如溜进核反应堆或雷区，获得危险地带的一手资料，甚至可以爬到太空检测板上对外部设备进行维修，人类将进入一个无孔不入的“毛毛虫”时代。

在机械上，也有一种装置可以实现这种毛毛虫只朝着一个方向的运动，这种装置叫做棘轮机构。请结合毛毛虫爬行方式提供的设计思路，设计出仿生运动机器人，实现蠕动爬行的展示。



提出了仿生毛毛虫机器人的运动原理,设计了新型基于曲柄滑块机构的仿生毛毛虫机器人运动机构。其具有反应速度快、结构稳定的特点。

本次课程案例可以从以下（4个）阶段任务像学生们进行讲解：

第一步详细介绍以毛毛虫为例的仿生学特点；

第二步讲解以毛毛虫为例的软体移动原理；

第三步灵活运用棘轮结构，完善搭建的创意与造型；

第四步总结加延伸拓展思考，开动脑筋，增加游戏趣味性。



蠕动原理：

毛毛虫的爬行行动在于其自由伸缩的身体和交替运动的多足之间的相互配合。因

此仿生毛毛虫机器人的设计关键是模仿具有拉伸收缩功能的身体结构，通过连杆结构可实现类似蠕动功效。

机器人由前后两节组成，机器人的后车轮各安装了十字轴套，作用于轮胎的防滑与固定。

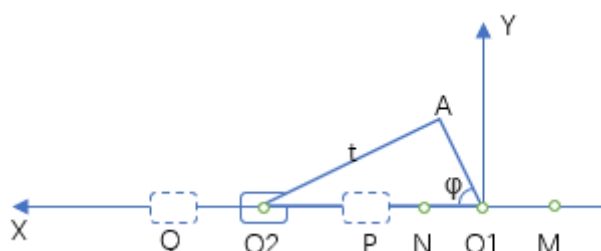
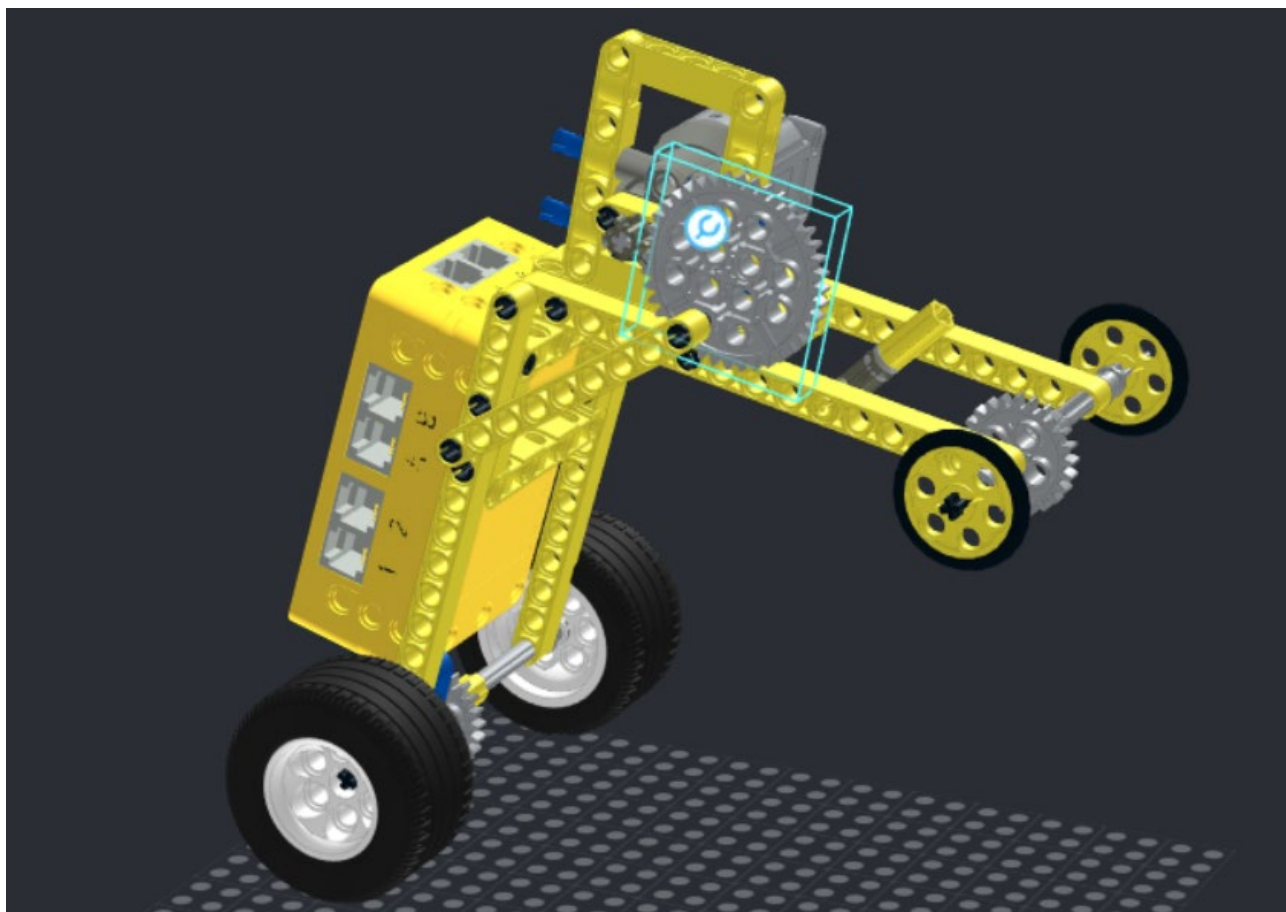


图 2 仿生毛毛虫机器人爬行原理

当机器人处于水平位置时，可简化为上图所示滑块机构。机器人前节即为滑块，当处于临界位置 M 点时， O_1O_2 之间的距离增加，此节后节车轮由于棘轮作用固定，因此连杆会推动前节向前滑行；当运动至 N 点时， O_1O_2 之间的距离会减小，此时车轮由于棘轮结构固定，因此连杆会拉动后节向前滑行，直至完成一个周期运动。因此机器人便像毛毛虫一样周而复始的向前伸缩滑行。

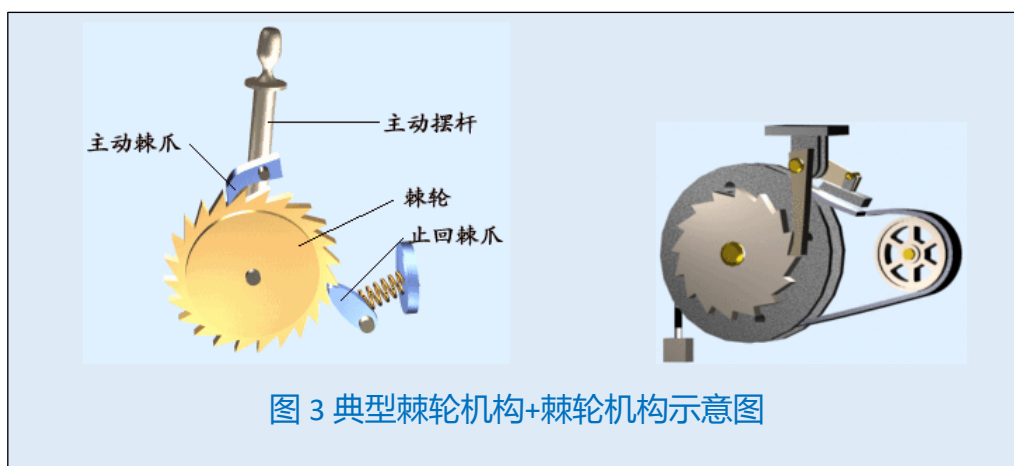
尺蠖行走的时候，身体像一个拱桥，所以尺蠖的身体框架我们需要制作前半身和后半身两部分，具体制作中要考虑电机的安装位置和棘爪的位置。

制作曲柄装置，连接身体框架的前半部分和后半部分。机械尺蠖是通过曲柄滑块机构将电机的运动传递出去的。



棘轮机构

棘轮机构用来将连续转动或往复运动转换成单向步进运动。典型的棘轮机构由四部分组成，分别为棘轮，主动棘爪，止回棘爪，主动摆杆，如图 3 所示。



主动件空套在与棘轮固连的从动轴上，并与主动棘爪用转动副相联。当主动件顺时针方向摆动时，驱动棘爪便插入棘轮的齿槽中，使棘轮跟着转过一定角度，此时，止回棘爪在棘轮的齿背上滑动。当主动件逆时针方向转动时，止回棘爪阻止棘轮发生

逆时针方向转动，而驱动棘爪却能够在棘轮齿背上滑过，所以这时棘轮静止不动。因此，当主动件作连续的往复摆动时，棘轮便作单向的间歇运动。

棘轮机构在生活 and 工业机械中都有很多应用。比如爸爸的腰带，用到的就是棘轮机构，通过拉动腰带，可以使棘爪在棘轮上划过，让腰带变紧，而想要松开腰带需要扳开棘爪，使其离开棘轮，让棘轮可以反向运动，才能解开腰带。除此之外，还有千斤顶、自行车等都运用棘轮机构将双向运动变为单向运动。

特殊的棘轮机构

棘轮机构可以按照不同的形式划分，按结构形式可以分为齿式棘轮机构和摩擦式棘轮机构；按棘轮棘爪的啮合方式可以分为外啮合棘轮机构和内啮合棘轮机构；按从动件运动形式分为单动式棘轮机构和双动式棘轮机构。每一种类型的棘轮机构都有不同的特点，下面我们依次讲解。

齿式棘轮机构：如图 4 所示，棘轮带有轮齿，棘爪可以插入棘轮的齿槽中。这种棘轮机构的优点是结构简单，制造方便，动与停的时间比可通过选择合适的驱动机构实现，缺点是动程只能作有级调节，噪音、冲击和磨损较大，故不适用于高速场合。



图4 齿式棘轮机构

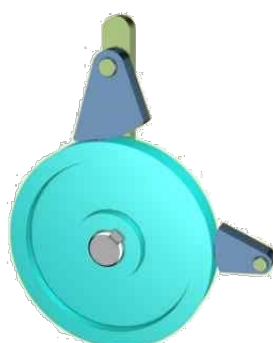


图5 摩擦式棘轮机构

摩擦式棘轮机构：如图 5 所示，用偏心扇形模块代替齿式棘轮机构中的棘爪，以无齿摩擦代替棘轮。这种棘轮机构中，棘轮没有轮齿，棘爪控制棘轮反转摩擦力。摩

擦式棘轮机构的优点是传动平稳、无噪音，动程可无级调节！缺点是靠摩擦力传动，会出现打滑现象，虽然可起到安全保护作用，但是传动精度不高，只适用于低速轻载的场合。

外啮合棘轮机构：这种棘轮机构中，棘轮和棘爪啮合在外侧，它的特点是加工、安装和维修方便，应用较广。

内啮合棘轮机构：棘轮的轮齿在棘轮的内侧，棘爪与之啮合在整个结构的内侧。这种棘轮机构的特点是结构紧凑、外形尺寸小。

单动式棘轮机构：当主动件按某一个方向摆动时，才能推动棘轮转动。

双动式棘轮机构：在主动摇杆向两个方向往复摆动的过程中，分别带动两个棘爪，两次推动棘轮转动。双动式棘轮机构常用于载荷较大，棘轮尺寸受限，齿数较少，而主动摆杆的摆角小于棘轮齿距的场合。

双向式棘轮机构：以上介绍的棘轮机构，都只能按一个方向作单向间歇运动，都属于单向式棘轮机构。棘轮机构中还有一类双向式棘轮机构，可通过改变棘爪的摆动方向，实现棘轮两个方向的转动，如图 5 所示。双向式棘轮机构在制作时必须采用对称齿形，才能实现双向运动。

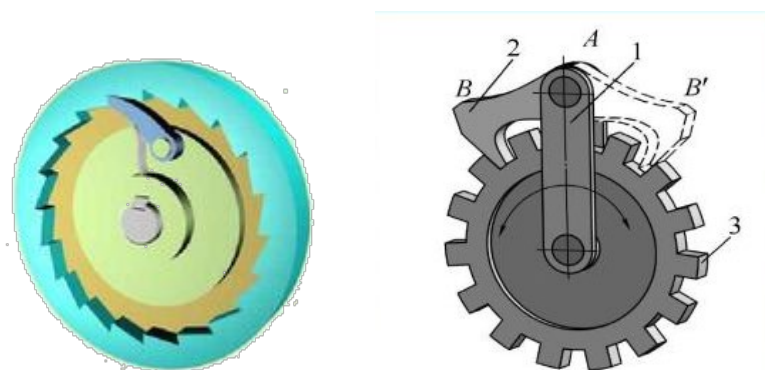


图 5 内啮合棘轮机构 + 图 6 双向式棘轮机构

曲柄滑块机构

在毛毛虫的制作中，传动装置中有一部分是利用曲柄滑块机构传动的。毛毛虫中

曲柄滑块机构的作用是将电机的圆周运动转化为直线运动。

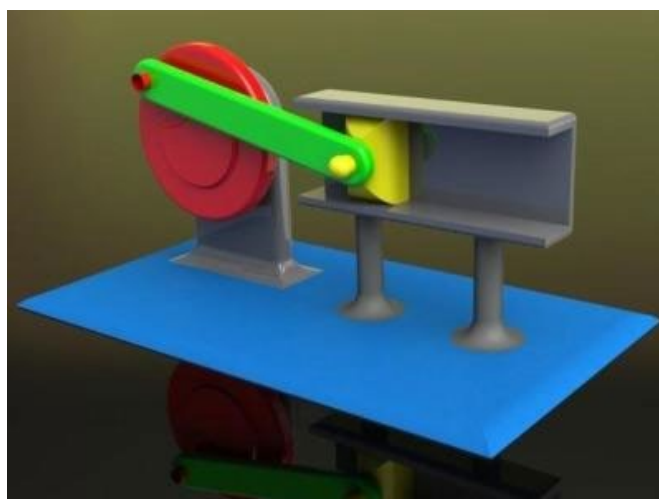


图 6 曲柄滑块机构

曲柄滑块机构是用曲柄和滑块来实现圆周运动和直线移动相互转换的平面连杆机构。曲柄滑块机构的示意图如 12.6 所示，当曲柄为主动件，滑块作为输出件时，曲柄滑块机构的作用是将匀速圆周运动(等速回转运动)转化为直线往复运动\$当滑块作为主动件，曲柄作为输出件时，曲柄滑块机构的作用是将直线往复运动转化为圆周运动。

曲柄滑块机构属于连杆机构的一种特殊形式，关于连杆机构的各种形式将在之后具体讲解。



给你们每个小毛毛虫作品来一场化妆舞会吧，看看谁的毛毛虫打扮是最漂亮的？

(使用纸板、画笔、水彩等物料)

chōng

四、春米机



1. 了解春米的过程及原理
2. 搭建一台“春米机模型”
3. 了解机器是如何将圆周运动转换为上下运动的
4. 了解凸轮机构的组成及作用

古时候，没有电力，春米完全靠人力，相当辛苦。农民用土砬（lóng）将稻米的壳磨掉之后，这时糙米和皮壳便混在一起，因此需要用风鼓将壳皮，也就是粗糠（kāng）统统吹走，然后再把糙米摆进“春臼”里，再用“春白槌”不断捣米，这就叫春米。

春米是为了将附在糙米的上一层糠皮捣去，这层皮就是米糠，去糠皮的米就是白米了。米糠可以用来当饲料，拌在剩菜剩饭里喂养家禽，鸡、鸭和鹅吃了拌米糠的饲料，不但有了充分的营养，还不会得脚气病（即俗称“软脚病”）。因为米糠里有相当丰富的维他命 B。



春米就是把打下的谷子去壳的过程，“春”出来的壳就是米糠，剩下的米粒就是我们吃的白米，春米的工具有点像捣药罐，有一个棒槌、一个盛器。用棒槌砸谷子，把米糠砸掉。

春米，谈不上什么工艺，过程也并不复杂，但绝对是个力气活。一口臼（jiù），

用一大块整石慢慢凿成。它形同一只巨大的酒盅，“臼口”的直径不小于 70 ~ 80 公分。一只大的石臼，一般要四个人才能抬得动它。臼内凿有铳纹，臼的身子须埋在地下，“臼口”露出地面，四周用陶瓷贴平，以便于清扫。在臼的上面，架着用一棵大树段做成的“碓身”，“碓”的头部下面有杆杵，杵的嘴子上按了铁牙。“碓”肚的中部，两边有支撑翘动的横杆，就像玩杂技的“翘翘板”中间支撑那玩意儿，“碓”尾部的地下挖一个深坑，当把重心移到“碓”尾并将其向坑里踩压下去，“碓头”便抬了起来，然后“舂”下去，抬起来，再“舂”下去，就这样机械地重复，便是舂米。

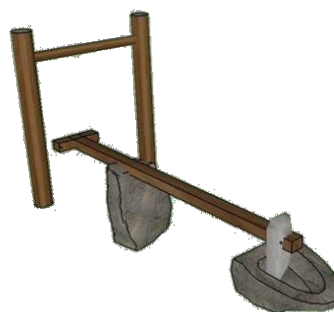


图 1 古老的舂米与工具

我们发现轴上的木板在作匀速圆周运动而“舂臼”在作间歇性往复摆动运动，“舂臼槌”随之上下摆动。这种由凸轮带动从动件将匀速圆周运动转换为间歇性往复摆动的机构叫做凸轮机构。（“舂米机”不仅更加省力，而且大大提高了效率。）



1. 基础构造：利用杠杆的基本形态，将舂米的锤子固定在支架上，类似于一个翘翘板的结构。
2. 操作这样的简易机械，只需要踩住一端让它反复的上下运动就可以了。
3. 锤子下端搭建一个装米的容器。

4. 锤子头利用重块来搭建，可以增加它舂米的力量。

注意：利用杠杆和连杆装置实现最基本的“舂米机”的构造。同时增加足够多的辅助设施，为舂米劳动服务。

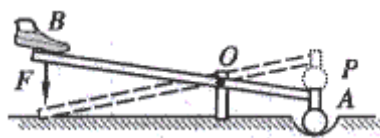


图 2 “舂米机”原理图



今天我们要制作的“舂米机”就是使用凸轮机构将电机的转动转化为“舂米槌”一下下的敲动，我们可以参考脚踩式舂臼和水碓的构造来制作舂米机。

1.制作“舂米机”的框架。“舂米机框架”需要一个平稳的底座和支撑凸轮机构的支架，还要为前面从动件的杠杆支点留出可穿轴的位置。

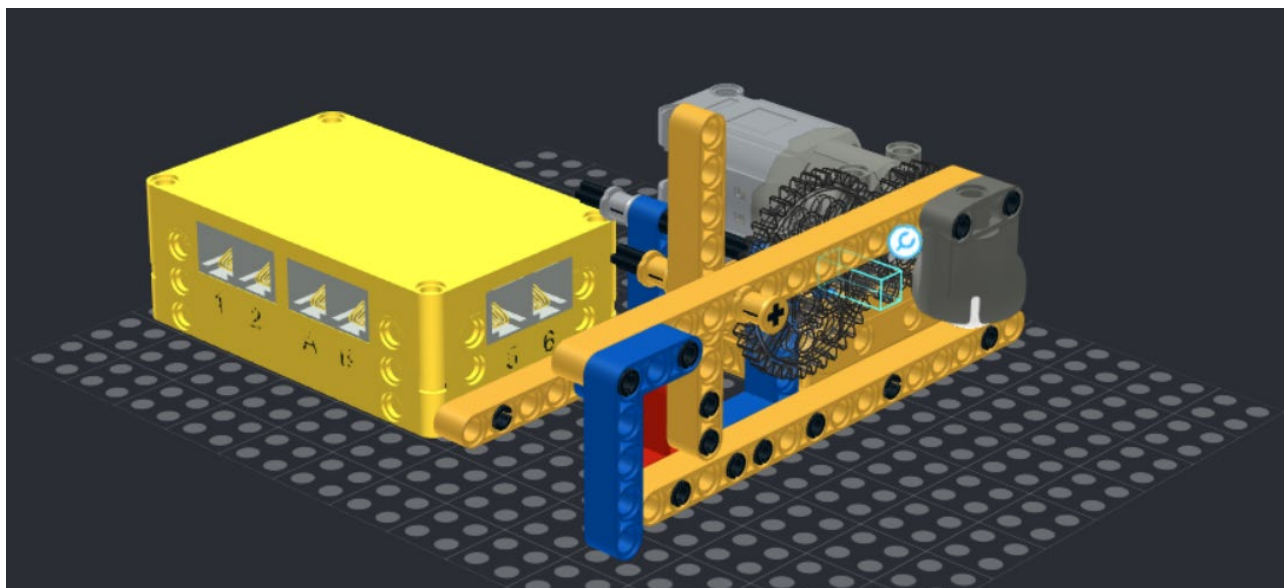
2.制作凸轮机构。

3.安装电机。电机安装在底座上。

4.制作传动装置。

5.连接从动件，完成模型。

6.连接电源，控制模型。



凸轮机构

“舂米机”中使用到的凸轮机构也常常应用在工业机械中。这节课让我们来认识一下凸轮机构。

凸轮机构是由凸轮，从动件和机架三个基本构件组成的高副机构。凸轮是一个具有曲线轮廓或凹槽的构件，它一般为主动件，作等速回转运动。从动件一般为长杆型，作为传递动力和实现预定的运动规律的构件。从动件一般作连续或间歇性的往复直线运动或摆动，从动件直线运动或者摆动的最大距离叫做行程。

凸轮机构在应用时衍生各种各样的状态。可以根据不同的方式分类，分别介绍如下：

按照凸轮形状不同，可以分为盘形凸轮，移动凸轮和圆柱凸轮三种，如图 3 所示。

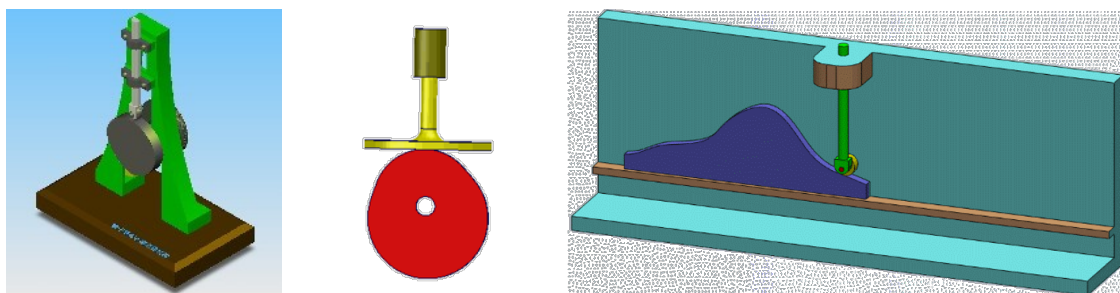


图 3 凸轮形状不同

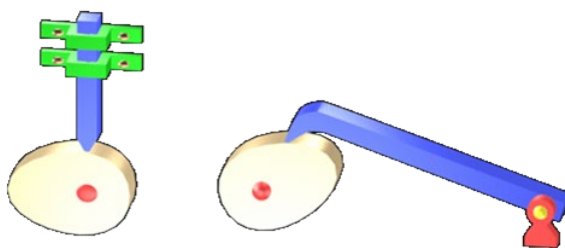
盘形凸轮是凸轮形状像盘子一样的凸轮机构，是凸轮机构中最基本的形式。盘形凸轮结构简单，应用最为广泛，凸轮一般作匀速圆周运动(等速回转运动)。

移动凸轮中凸轮形状可能是个“山”形或斜面形，凸轮相对机架做直线移动。

圆柱凸轮中凸轮形状为圆柱形，这种凸轮机构属于空间凸轮机构，凸轮绕自己中心轴作圆周转动。

按从动件形状不同，可以将凸轮机构分为尖顶从动件，滚子从动件和平底从动件。

尖顶从动件是最基本从动件的形式，从动件与凸轮接触的部分为尖顶形状。如图15.14 分别为尖顶从动件盘形凸轮，尖顶从动件圆柱凸轮，尖顶从动件移动凸轮。尖顶从动件能够与任意复杂的凸轮轮廓保持接触，从而使从动件实现任意的运动规律。但尖端处摩擦较大，极易磨损，所以尖顶从动件只能用在传动力不大的低速机构。



4 尖顶从动件凸轮机构

滚子从动件是将从动件与凸轮接触的顶端变为滚轮，将滑动摩擦变为滚动摩擦，依次为滚子从动件盘形凸轮，滚子从动件移动凸轮，滚子从动件圆柱凸轮。滚子从动件的优点是凸轮与从动件之间为滚动摩擦，因此摩擦磨损较小，所以滚子从动件适用于传动力较大的中、高速机构，应用十分广泛。

平底从动件将从动件与凸轮接触的顶端做成一个平底，从动件与凸轮之间易形成油膜，润滑状况好，受力平稳，传动效率高。但从动件做成平底后与之相配合的凸轮轮廓须全部外凸，所以一般只与盘形凸轮搭配。平底从动件适用在高速，传动力大的

机构。

根据运动形式的不同，可以将凸轮机构分为直动从动件和摆动从动件。

直动从动件是指从动件作往复移动，其尖顶的运动轨迹为一段直线。直动从动件有两种形式，一种是对心直动从动件，一种是偏置直动从动件。对心直动从动件就是尖顶或者滚子或者平底的中心的轨迹通过凸轮的轴心，偏置直动从动件尖端或滚子中心的轨迹不通过凸轮的轴心。

摆动从动件是指从动件作往复摆动，其尖端的运动轨迹为一段圆弧，分别为摆动平底从动件，摆动尖顶从动件和摆动滚子从动件。



1. 支点的位置如何选择才是最佳方案？（锤子离支点较远时运行轨迹较大，所转化的重力势能也就越强）
2. 如何增加“舂米机”锤子的力量？让它变得更有效率。
3. 在不破坏自然环境的前提下合理利用自然资源，应该做哪些准备工作？
4. 能否使用齿轮传动的方式，在齿轮上安装拨动锤子的销，或者做成连杆装置？
5. 能否在手把处增加一组减速齿轮，使舂米更省力？

五、旋转杯子



1. 了解旋转杯子运动的原理
2. 初步了解旋转杯子中杯子旋转的基本方式
3. 掌握自转和公转的知识
4. 了解惯性在生活中的广泛应用



旋转杯子是一种新型游乐设施。旋转杯子造型优美，在乘坐旋转咖啡杯时，一边公转，一边自转，游客的愉悦感油然而生。旋转杯子作为新兴游乐设施，吸引了很多游客，和其他游乐设施相比，既安全自在，又比旋转木马多了自传的功能，所以受到了游客的追捧。



图 1 游乐项目——旋转杯

旋转杯子的运动使用的是齿轮装置，通过齿轮之间的啮合将运动传递出去。那么游乐场的旋转杯子如何实现同时公转和自转的呢？



随着我国经济的快速发展，人们对于孩子的期望，也变得与以前不大相同，尤其是那些独苗家庭。我们知道以前，我国儿童正统教育中，人们对孩子寄予的期望是“乖”、“听话”、学习成绩好等等相对单一而静态的衡量指标；但是如今就不同了，家长们越来越意识到儿童们的创新精神和动手能力的重要性。

旋转杯子中使用了“行星齿轮”，顾名思义，行星齿轮就像行星一样，一边自转，一边围绕着恒星转动。下面我们学习行星齿轮的制作。

1.制作旋转杯子的底座。

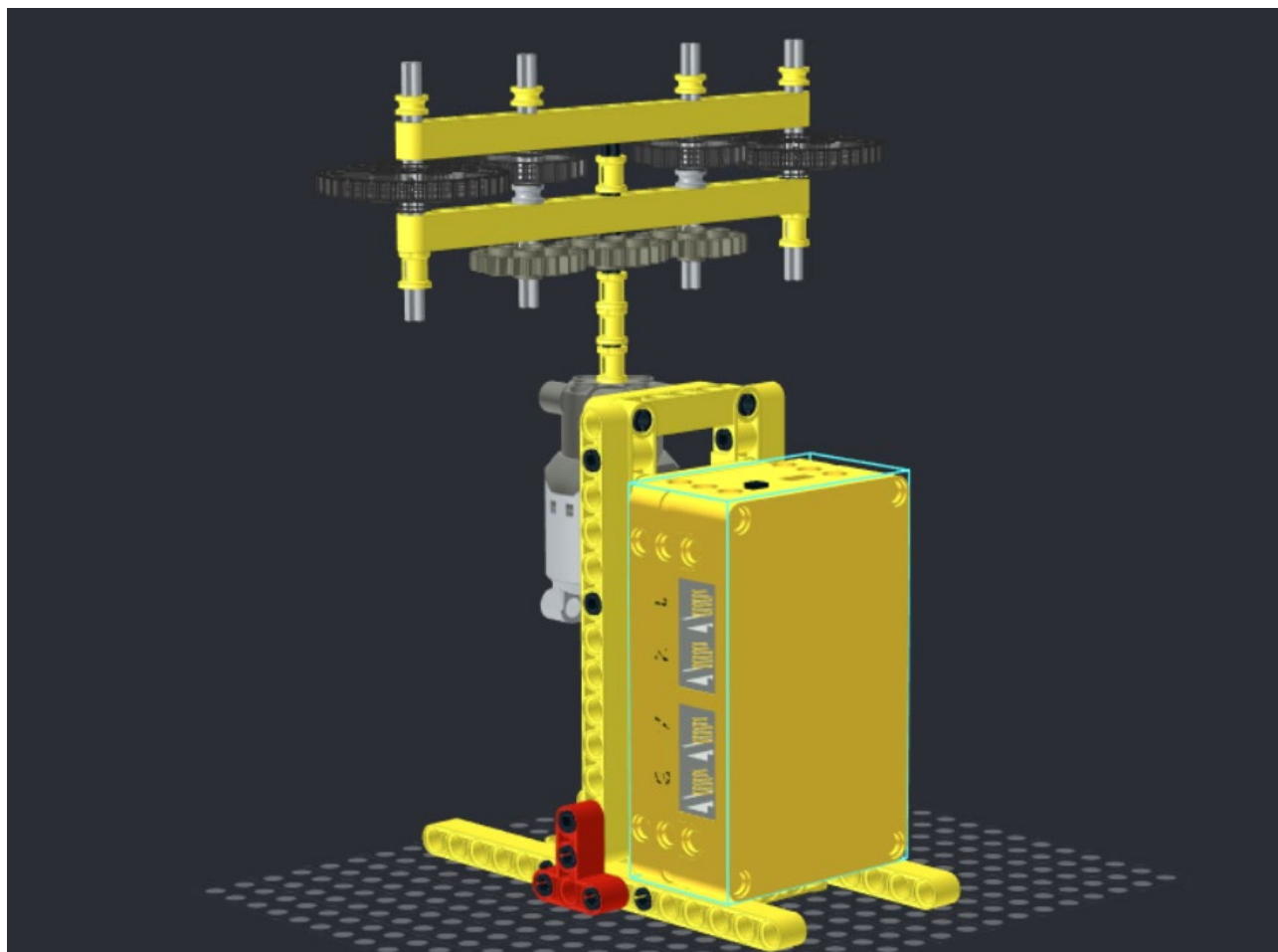
2.安装电机。电机安装在旋转杯子支架的一侧。

3.制作行星齿轮。两根长条之间要安装四根轴，轴上要安装齿轮，滑轮等传动装置。

4.制作传动装置，并连接行星齿轮和杯子，完成模型。传动装置为齿轮啮合传动，在旋转杯子支架中间穿插中心轴，在轴上安装齿轮与电机上齿轮啮合。并在中心轴的上方安装行星齿轮机构，将双臂曲柄固定在轴上，使得行星齿轮机构上的小齿轮与作为恒星的齿轮啮合。注意：作为恒星的齿轮不固定在轴上，它是固定在支架上的。

5.连接电池盒，控制模型。用电池盒控制电机的转动，通过啮合齿轮带动中心轴转动，从而带动中心轴上方的整个行星齿轮结构转动，形成公转。由于整个行星齿轮结构绕中心轴转动，此结构上的小齿轮与大齿轮啮合，所以小齿轮围绕大齿轮转动的同时，必须自转，由小齿轮的转动带动同轴的滑轮，然后通过皮带，带动杯子自转。所以我们会看到杯子一边公转，一边自转。





旋转杯子的公转和自转

旋转杯子的运动方式是公转同时自转。公转是指一件物体以另一件物体为中心所作的循环运动。自转是指物体自行旋转的运动，物体会沿着一条穿越身体本身的轴进行旋转，这条轴被称为“自转轴”。

旋转杯子实现公转自转的主要因素是行星齿轮，行星齿轮除了能像定轴齿轮(位置不变的齿轮)那样围绕着自己的转动轴转动之外，它们的转动轴还随着支架(称为行星架)绕其它齿轮的轴转动。绕自己轴线的转动称为“自转”，绕其它齿轮轴线的转动称为“公转”，就象太阳系中的行星那样，因此得名。

旋转杯子公转自转的原理如下：

旋转杯子中，电机上的齿轮通过啮合将动力传递，齿轮固定在轴上，带动轴一起转动。轴上固定的双臂曲柄，它将中心轴的转动传递给长条，使长条绕着中心轴转

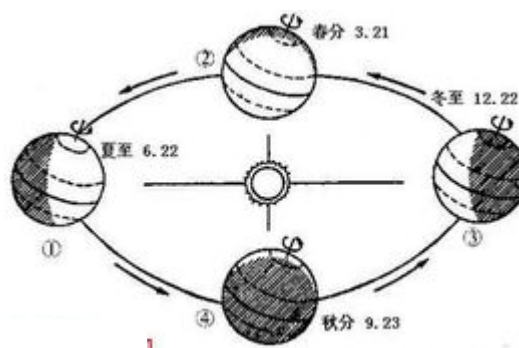
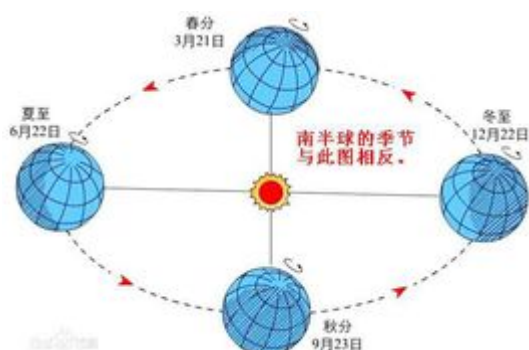
动，从而带动两端的杯子围绕中心轴转动，形成杯子的公转。

支架上的恒星齿轮不与中心轴固定，而是固定在框架上，它本身不转动，只是提供轮齿，就像一个跑道一样，其他齿轮在围绕它转圈时，由于轮齿啮合在一起，就只能转动自身。公转时，行星齿轮上的小齿轮与恒星齿轮啮合，所以自身转动，再通过皮带装置将自身转动传递出去，最终实现杯子的自转。

这样，杯子就完成了一边公转一边自转的活动。

地球的公转和自转

公转是一件物体以另一件物体为中心所作的循环运动。比如太阳系里的行星绕着太阳转动，各行星的卫星绕着行星而转动，公转所沿着的轨道可以为圆、椭圆、双曲线或抛物线，公转方向为自西向东。



地球环绕太阳的运动称为地球公转，地球公转一周的周期称为 1 太阳年，约为 365 天又 5 小时 48 分 46 秒。因为地球围绕着太阳转，轨道是一个椭圆，而太阳不是在正中心。我们的地球在围绕太阳不停地公转的同时，也在绕自身的地轴自转，不过地轴并不垂直于公转轨道面，而是有“倾角”。正是因为这个倾角的存在，才会使太阳在地球表面的直射点在南、北回归线之间移动，从而形成了春夏秋冬四个季节。

地球公转的地理意义：

1. 太阳直射点的回归运动北半球夏至(6 月 21 日或 22 日)，大阳直射北地球回归

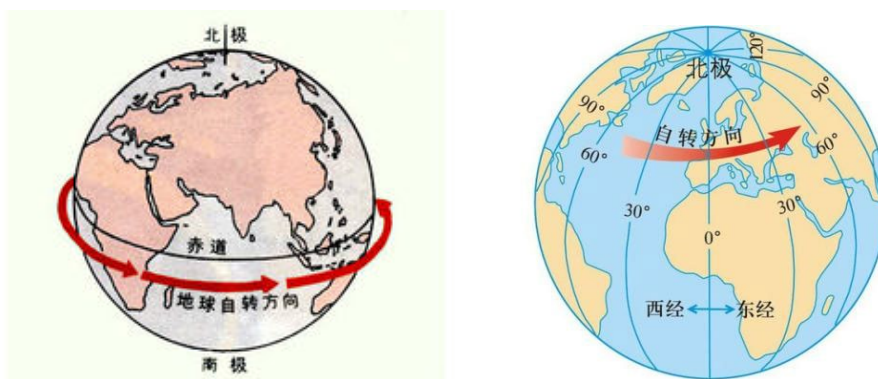
线;北半球冬至(12月21-23日), 太阳直射地球南回归线春分, 秋分, 太阳直射地球赤道。

2.昼夜长短变化太阳直射北半球时, 南半球黑夜长白昼短, 北半球白昼长黑夜短。太阳直射南半球时, 北半球黑夜长白昼短, 南半球则反之。太阳直射赤道时, 地球大部分昼夜等长。

3.正午太阳高度的变化以中国为例, 夏季、太阳正午时, 几乎在头顶正上方, 影子很短, 因为此时太阳直射北半球, 冬季, 太阳斜上方, 感觉较远, 影子较长, 因为这时太阳直射南半球。

4.四季和五带划分当太阳直射在北回归线时, 北半球为夏季。当地球绕太阳再公转半圈时, 太阳的直射点由北回归线移向南回归线, 由夏季进入秋季, 进而转入冬季, 而南半球却正好相反。

自转指物件自行旋转的运动, 物件会沿着一条穿越身体本身的轴进行旋转, 这条轴被称为自转轴。所有卫星、行星、恒星、星系绕着自己的轴心转动。



地球同太阳系其他行星一样, 在线太阳公转的同时, 围绕着一根假想的自转轴在不停地转动, 这就是地球的自转。地球的自转是自西向东。地球自转一周的时间是 23 小时 56 分 4 秒。

地球自转的意义:

1.产生昼夜交替地球自转时，地球上向着太阳的一面为白昼，背对太阳的一面为黑夜。

2.沿地表水平运动的物体的偏移物体受地球自转偏向力的作用，运动方向偏向一侧，比如弯曲的河流。纬度越高（越接近南北极），偏向越明显。

3.产生地方时经度(地球仪上纵线)不同，各地方的时间不同。由于地球自西向东自转，同纬度(地球仪上横线)偏东位置的地方总比偏西位置的地方要先见到日出，时刻较早。经度每隔 1° ，地方时相差 15 分钟每隔 15° ，地方时相差 1 小时。

六、稻草人



1. 学习了解不同传动的作用及用处
2. 学习比较各运动方式的区别
3. 了解齿轮、皮带及传动链的作用



稻草人是农人为守护田地，以防鸟雀糟蹋粮食一类的庄稼，而立于田边用稻草做的人像。因为用稻草做的故叫其“稻草人”。



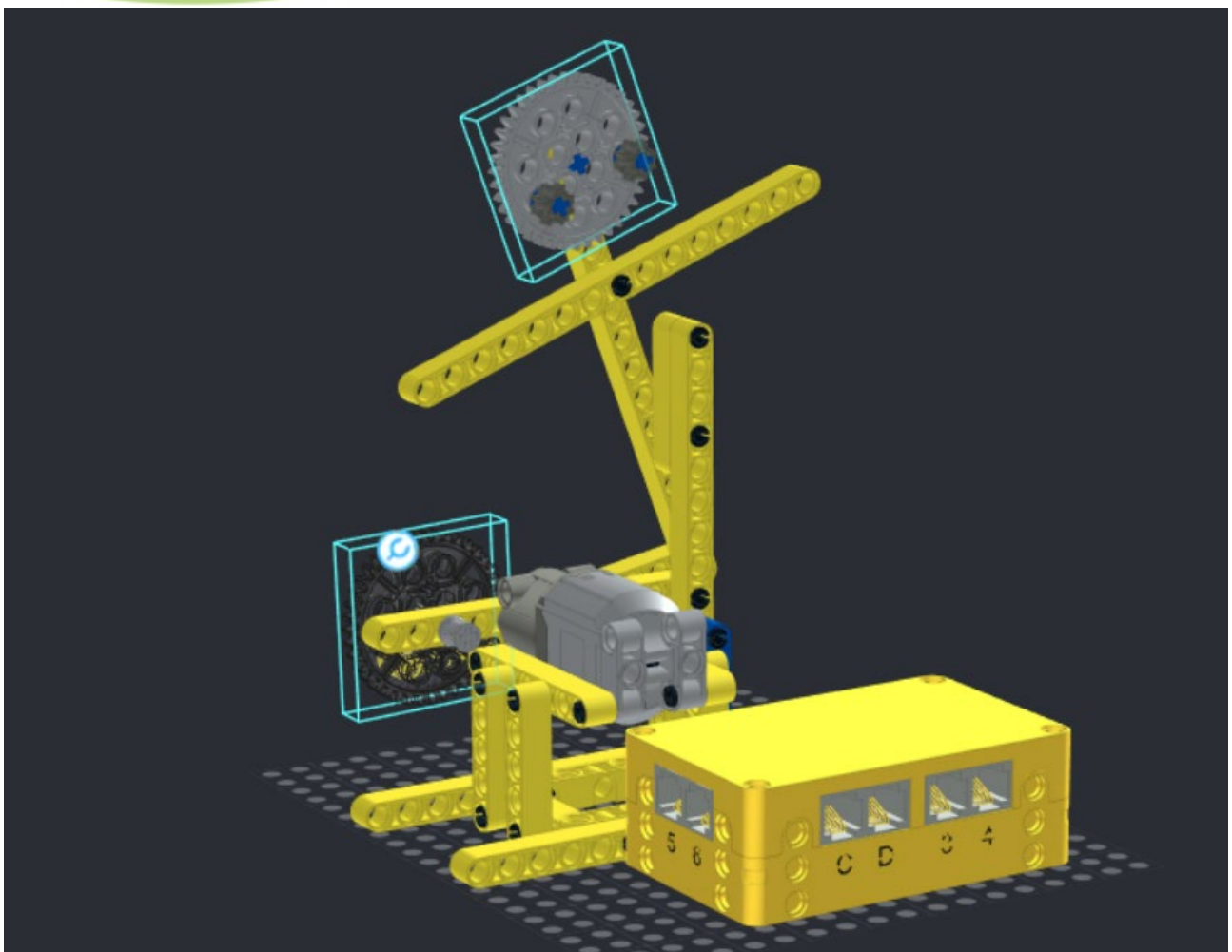
图 2 动画稻草人

很久以前，农民伯伯在农场上耕种，春天播撒种子，秋天收获粮食。但是好景不长，天空来了一群不速之客，一群乌鸦偷走了农民伯伯的粮食，整个农场被糟蹋得杯盘狼藉。后来，农民伯伯想了一个办法：发挥他的聪明才智用稻草制作了一个逼真的假人，这个假人就是稻草人。农民伯伯把稻草人插到农场里，乌鸦以为是真的人在看守粮食，于是再也没有糟蹋过这片上地。

大家有没有想过如果我们能让农场上的稻草人不停的晃动，这样就与人类更接近，会帮助农民伯伯赶走更多的麻雀和乌鸦，减少他们的损失。



- 1.制作稻草人的支架。稻草人需要一个平稳的底座。
- 2.制作稻草人的头和身体。
- 3.安装电机，将电机安装在底座上。
- 4.制作传动装置。
- 5.连接各部分，完成模型。
- 6.连接电池及控制器，控制模型。



皮带传动

皮带传动是两个传动轮之间有一定距离，能够生产连续的旋转运动，将力从一个传动轮传导到另外一个传动轮的传动装置。在不同的传动场合，皮带传动常用的方式有三种平行传动，交叉传动，半交叉传动，下面我们依次讲解。

平行传动：皮带直接绕过两个传动轮，这种传动方式也称作开口传动。如图 3 所示，传动时两个滑轮转动方向相同。当小滑轮作为主动轮顺时针转动时，大滑轮也顺时针转动。

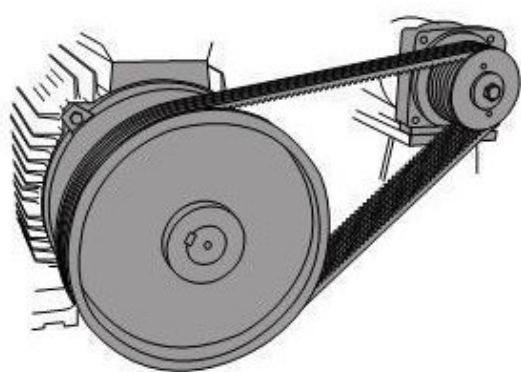


图 3 传动方式

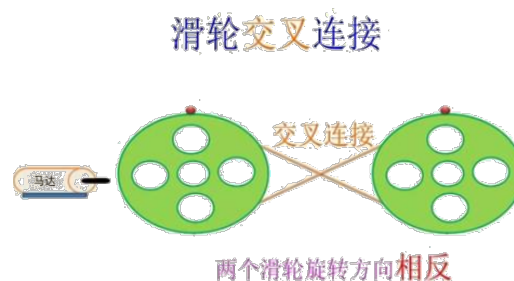


图 4 交叉传动

交叉传动：皮带反方向 180° 绕过两传动轮，如图 4 所示，这种传动方式叫交叉传动。交叉传动时，两个滑轮的转动方向相反，当小滑轮顺时针转动时，则大滑轮逆时针转动。

半交叉传动：皮带交叉 90° 绕过两传动轮，穿过两个传动轮的轴位置异面垂直，这种传动方式叫半交叉传动。两个滑轮不在同一平面上，所以不讨论转动方向。半交叉传动使用较少，轴的位置需要异面垂直，两个传动轮的中心位置连线又要和两个轴都垂直，限制较多，不容易调节。

在机械中，皮带传动装置应用很多，它拥有以下优点和缺点。

皮带装置的优点：

可实现远距离传动，皮带是有一定的长度的，这样就可以只利用两个滑轮和一根皮带将力传递到较远距离。在突然施加外力或者突然变速时，可以保护机械，不会损坏零件。噪音小，相对于齿轮啮合，皮带传动的噪音较小。

皮带装置的缺点：

在传递动力过程中，会有时间上的延缓。皮带有弹性，在主动轮从静止到转动的一刻，皮带被拉动伸长，从动轮并没有马上运动，会有一瞬间的延缓。

不能准确无误的传动，我们讲过减速和加速装置，在传动时，齿轮速度和齿轮半径成反比，可以得到准确的数值，比如半径为 3:1 的齿轮，大齿轮为主动轮时，为加速装置，大齿轮速度与小齿轮速度比应该为 1:3，小齿轮速度为大齿轮速度的三倍。而皮带由于弹力和打滑的问题，在传动时，不能准确无误的传递运动与力，同样的大滑轮为主动轮的加速装置，小滑轮的速度没有达到大滑轮速度的三倍。并且在传动时，能量损失比较大，同理。皮带容易破损。

三种运动方式的比较：

通过前面的课程我们一共讲述了三种传动方式，分别是齿轮传动，皮带传动，链传动。每种传动方式都有各自的特点，它们的使用场合也不尽相同，具体使用哪种传动方式要根据模型要求和动力，速度等多种情况综合考虑。三种传动方式的区别详见下表所示。

齿轮、皮带、传动链之间的区别					
传动装置	安装方向	转动方向	传动距离	优点	缺点
齿轮	同一平面（轴平行）	两齿轮转动方向相反	近距离：两个齿轮直接咬合 远距离：多个齿轮咬合	可以准确无误的传递动力；齿轮传动力大；结构紧凑	噪音大； 易损坏零件、电机
	轴同面垂直：冠齿轮与平齿轮	不在同一平面不讨论方向			
	轴异面垂直：蜗轮和平面轮				
	可任意角度咬合：多用途齿轮				
传动链	必须同一平面	同向	远距离传递动力	准确无误的传递动力； 远距离传递	噪音大
皮带	同一平面：皮带平行	两滑轮转动方向相同	远距离传递动力	噪音小； 保护机械； 在皮带的弹性范围内远距离传递动力	有时间的延缓，无法准确传递； 有能量损失，传递力量小； 皮带速度过大或突然加速会导致皮带断裂
	同一平面：皮带交叉	两滑轮转动方向相反	/		
	垂直平面（轴异面垂直）：皮带半交叉	不在同一平面不讨论方向	/		

七、缝纫机



1. 了解并掌握滑杆结构
2. 了解并掌握曲柄连杆机构
3. 学习了解皮带装置



缝纫机是用一根或多根缝纫线，在布料上形成一种或多种线迹，使一层或多层缝料交织或缝合起来的机器。缝纫机能缝制棉、麻、丝、毛、人造纤维等织物和皮革、塑料、纸张等制品。



图 1 缝纫机

缝纫机比起手工缝制不仅提高了效率，缝纫速度变快，并且缝纫机缝出的线迹整齐美观、平整牢固。缝纫机的使用也非常简便。

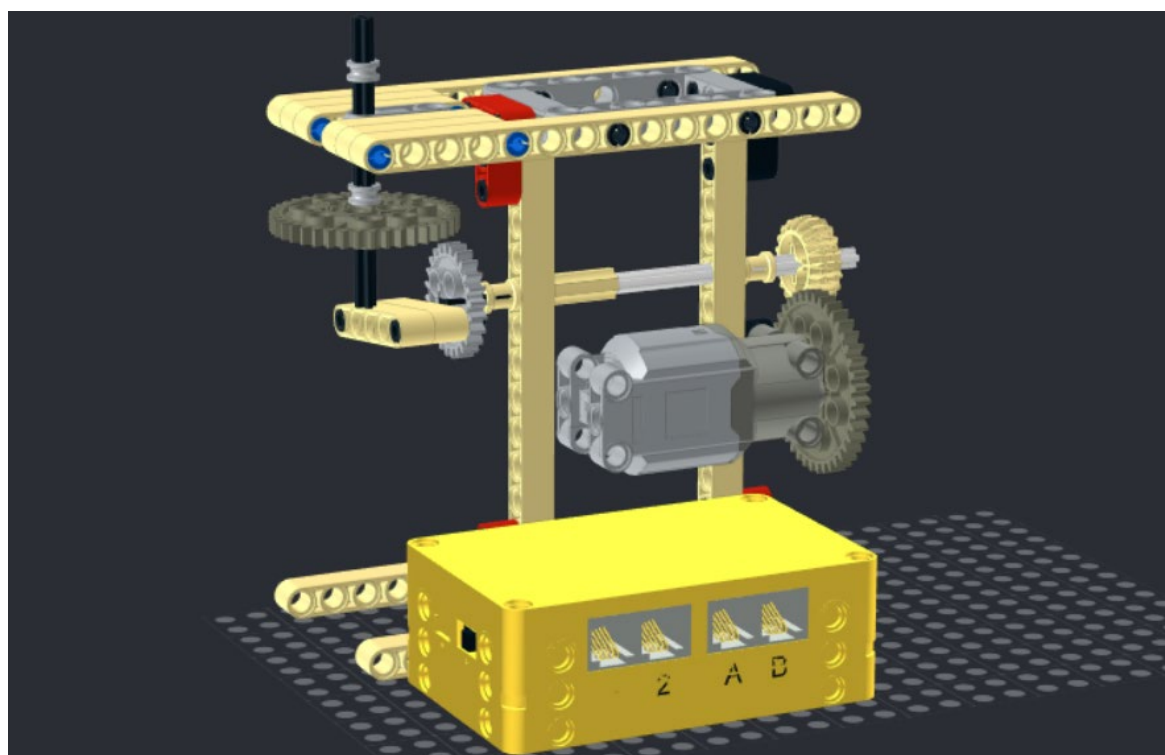
小时候的脚踏式缝纫机，下面通过脚的踩动，带动与其连接的滑轮转动，滑轮通过皮带装置带动上方的滑轮转动，实现了远距离传递动力；而滑轮的转动通过缝纫的主体部分，最终转换为针头的上下穿梭。

电动式缝纫机，直接将电机轴上带动的滑轮转动，通过皮带传递出去，最终转化为针头的上下穿梭。

在上面的介绍中，我们依次使用到了几种特殊结构，一是曲柄连杆机构(注意此时是连杆作为主动件，由连杆带动滑轮转动)，二是皮带装置(模型中用传动链代替)，三是曲柄连杆机构(此时曲柄作为主动件)，四是滑杆结构。前三种机构前面已经学习过，只作复习和总结，这节课我们主要讲解滑杆机构。



- 1.制作缝纫机的机架。缝纫机的机架需要底座和机身。
- 2.安装电机。电机安装在缝纫机的底座上。
- 3.制作传动装置。
- 4.制作曲柄机构和滑杆，完成模型。
- 5.连接电池盒，控制模型。

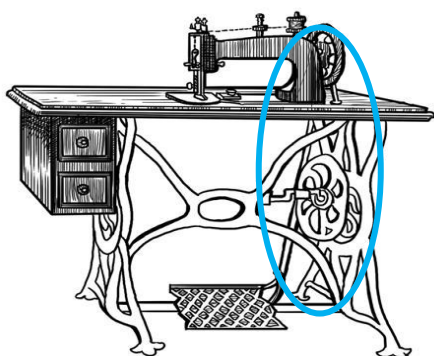




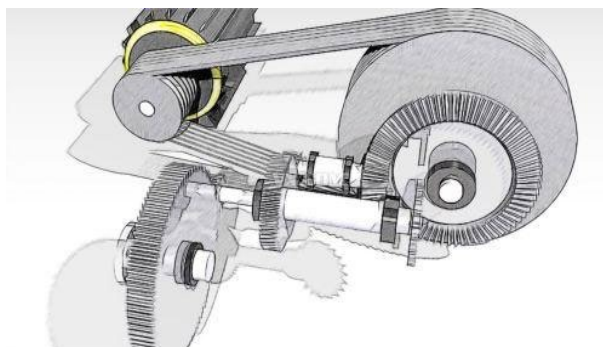
机械原理分析

缝纫机中的几处机械原理：

曲柄摇杆机构。如图 2 所示，在脚踏式缝纫机的踏板处使用的机械结构为曲柄连杆(摇杆)机构。通过脚步用力，使踏板作摆动，带动滑轮作圆周运动。在这里踏板是摇杆并作为主动件，滑轮是曲柄作为从动件。



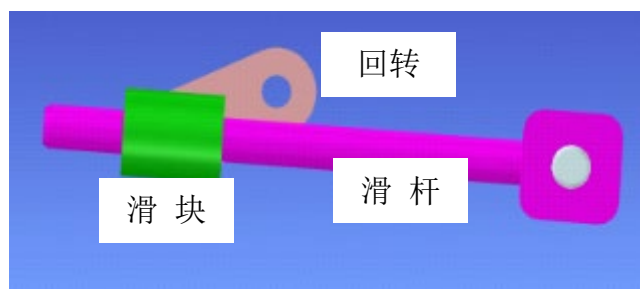
3 缝纫机曲柄摇杆机构



4 缝纫机传动链装置

传动链装置。如图所示，传动链可以实现远距离传递动力，比起齿轮啮合，传动链更加的适合远距离的时候，虽然齿轮啮合也能实现，却要在中间加很多个轴和齿轮，过于繁琐。传动链装置在使用时需要注意的是，两个齿轮必须在同一个平面上，以防止传动链左右受力不均而容易断裂并影响传递运动的准确性。

滑杆机构。滑杆机构由滑块、滑杆、回转体和机体组成的一种机械装置。如图 5 所示。滑杆机构中回转体为主动件，作匀速圆周运动(等速回转运动)，滑块从动，并带动滑杆作往复直线运动或者往复摆动。图中，两个圆孔处都是固定点，当回转体作圆周运动时，滑块跟着回转体运动，带动滑杆上下摆动。



5 滑杆机构

间歇运动机构

通过前几节课的学习，我们掌握了几种特殊的机构，比如凸轮机构，棘轮机构等。我们发现这些机构可以将连续转动转化为从动件间歇性或者周期性的运动，像这样的机构统称为间歇运动机构。

间歇运动机构是能够将原动件的连续转动转变为从动件周期性运动和停歇性运动的机构。有些机械需要其构件周期地运动和停歇，例如牛头刨床工作台的横向进给运动，电影放映机的送片运动等都用到间歇运动机构。

间歇运动机构按照运动方向可分为单向运动和往复运动两类。

凸轮机构。凸轮每转动一周，凸起的部分带动从动件运动，无凸起的部分从动件不动作。如图 6 所示。

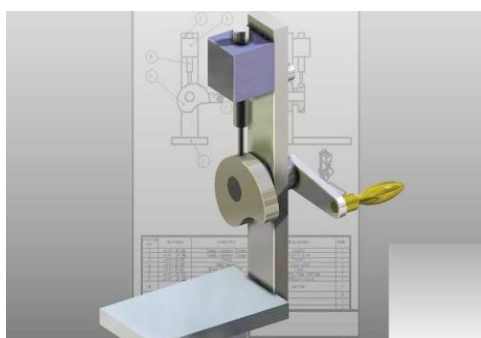


图 6 凸轮机构

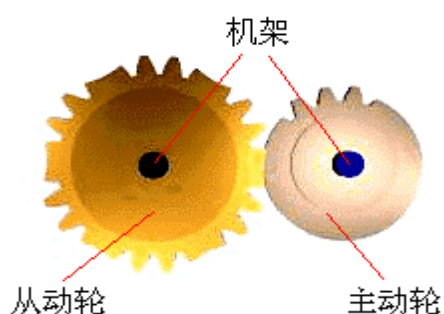


图 7 不完全齿轮机构

不完全齿轮机构：用一个没有布满圆周的齿轮做主动轮，则没有齿的一段圆弧就不会带动从动轮转动，而实现间歇运动。如图 7 所示。

槽轮机构：一个带槽的槽轮与一个带圆销件的机构组成。当圆销插入槽轮的槽

中，带动槽轮转动，而圆销离开槽时，槽轮停止转动，如图 8 所示。

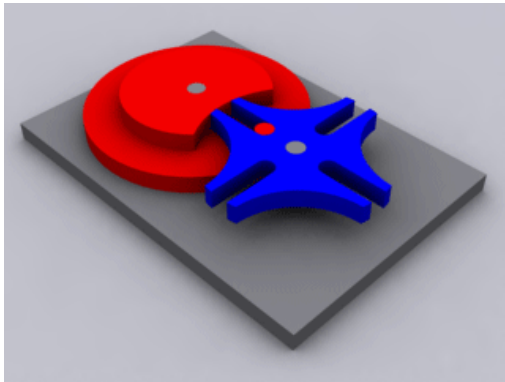


图 8 槽轮机构



图 9 棘轮机构

棘轮机构。通过棘爪使棘轮达到间歇性运动。如图 9 所示。

间歇运动的其它方法。主要是通过电流来改变动作，使通过电流的时有时无，得到间歇运动。

八、六足步行机器人



1. 学习了解六足运动的方式
2. 学习掌握差速运动控制
3. 掌握爬虫的前进原理及了解机械模仿动物行走的基本方式



仿生六足机器人又叫蜘蛛机器人，顾名思义，六足机器人架构中借鉴了自然界中昆虫的行走方式，是多足机器人的一种。

六足步行机器人的行走方式多样，最典型的行走方式就是模仿六足纲昆虫的三角步态。



图 1 六足步行机器人

三角步态是什么样子呢？

六足昆虫行走时，一般不是六足同时直线爬行，而是将三对足分成两组，每组三只足，以三角形支架结构交替前进，如图 2 所示。分组情况是这样的身体左侧的前后、足及右倒中足为一组；右侧的前足、后足和左侧的中足为另一组，分别组成两个

三角形支架。行走时接触地面的腿如黑方块所示，形成稳定的三角结构，这样模型通常会保持直立平稳的走姿而不会走路时跌跟头了。

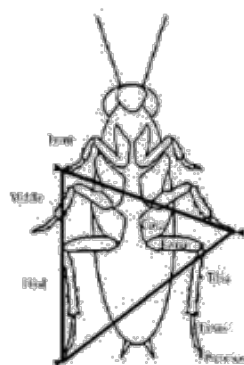


图 2 三角步态的分组

这种六足机器人控制简单，不过动作单一，因为他的每个关节都是通过机械连接完成，最后由一个统一的动力源控制。

另外，我们还能看到很多复杂的六足机器人每一个关节都是分开控制的。参照关节类动物的运动形式，这个六足机器人的每条腿的关节都是相对独立的，每个关节都是用 180 度的舵机来实现的，每条腿上三个关节，对应就是 3 个舵机，六条腿就是 18 个舵机。

不过这种用 18 个舵机支撑起来的机器人控制复杂，对电源的要求太高，还需要通过程序来控制，调试时间长。

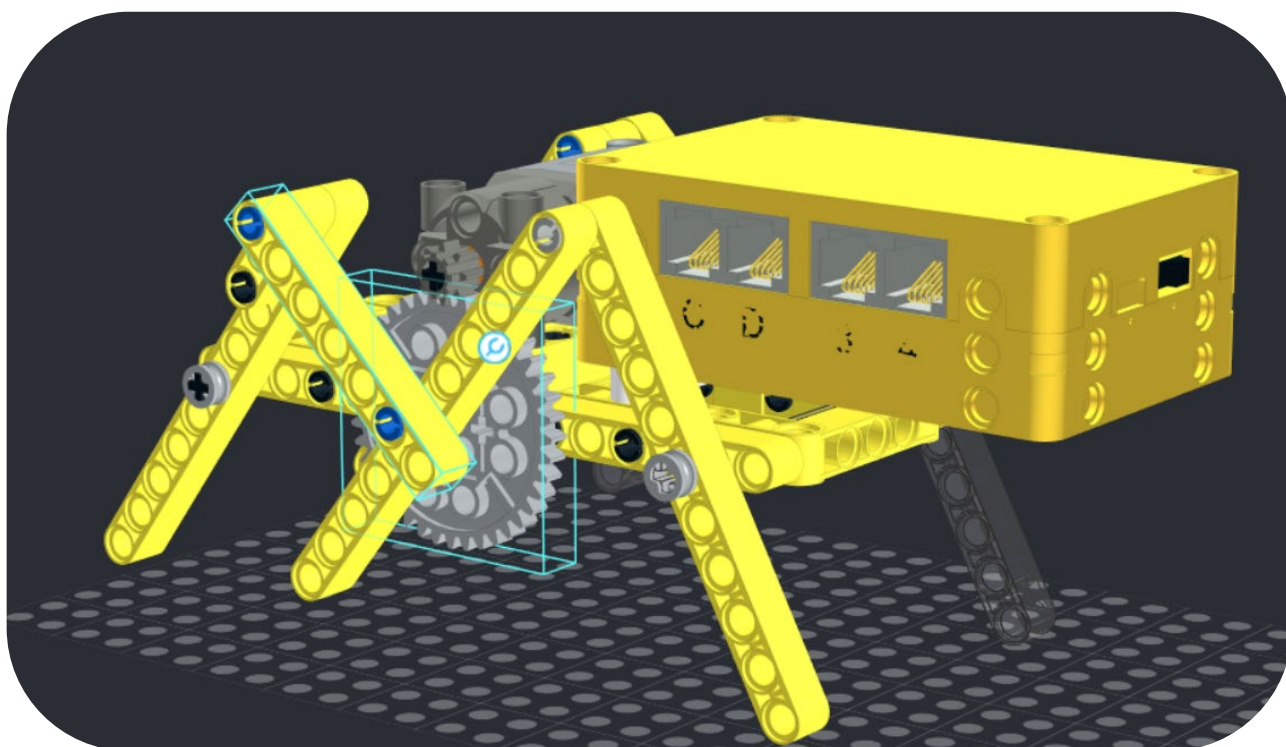
我们综合以上的知识内容，制作一个双电机的六足移动机器人，即保证了机器人的控制简单，也能实现机器人前后左右、左转右转的移动效果。



- 1) 从应用的角度理解联动杠杆系统的涵义。
- 2) 通过简单的系统设计案例的分析，初步学会简单系统设计的基本方法。
- 3) 注意：对爬行类作品有很多自己的想法并充满了想象，但是一定注意器材使

用的合理性，用有限的物料解决问题。制作步骤如下：

- 1.完成一个基本的框架。
- 2.在这个框架上，我们要安装电机。
- 3.在电机上我们安装一个多孔的齿轮。多孔滑轮实际上能够完成一个曲柄。
- 4.在框架的两头安装两根轴，然后在四个角分别实现一条做圆弧运动的“足”，这四个足的一端也通过一个铁条连接到滑轮上。



角上的四个“足”通过固定器固定在轴上，避免运动时从轴上脱落。在连接两个铁条以及连接铁条和滑轮时要注意，要留有一定空间让各个部件能够灵活转动，必要时还要在部件之间增加垫柱或垫片。

用电池控制两个电机转动。当两个电机都向前转动，六足机器人前进；反之，两个电机都向后转动，六足机器人后退当左侧电机停止或向后转动，右侧电机向前转动时，六足机器人左转反之，当右侧电机停止或向后转动，左侧电机向前转动时，六足机器人右转。



六足机器人运动方式

六足机器人的运动方式为三角步态，上面我们简单介绍过，三角步态就是六足机器人的六只足分成了两组，组成了两个三角支架。

当一组三角形支架所有的足同时提起，另一组三角形的足原地不动，支撑身体并以中足为支点。如图 4(a) 中所示。这一组的三角架抬起到下落的过程中，前肢的构件向前迈进拉动身体向前，后足的构件将机器人向前推，如图 4(b) 所示。这一组落地，另一组的三只足抬起，重心落在这一组三角形支架的三足上，然后重复前一组的动作，达到图 4(c) 的效果。之后互相轮换周而复始。

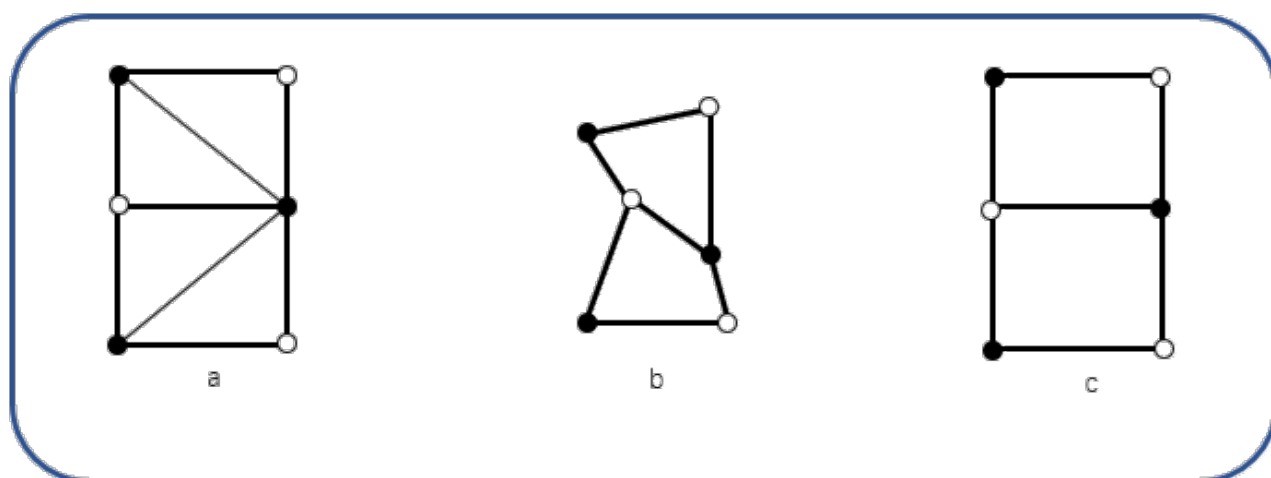


图 4 三角步态

差速运动控制

六足机器人中使用了两个电机，分别控制六足机器人的两边。六足机器人的转向原理为差速转向。差速转向是指两侧速度不同，有速度差而产生转向的方式。

六足机器人前进。两边电机以同样速度向前转动时，六足机器人会向前进，如图 5 所示。

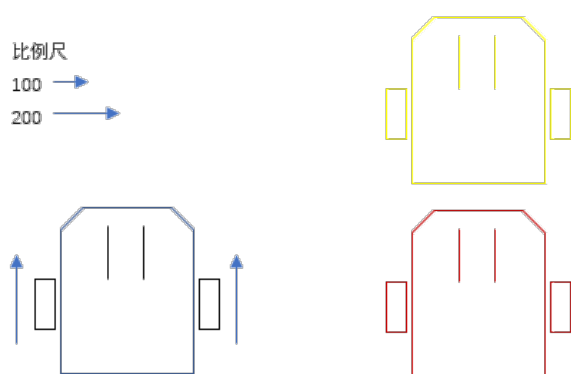


图 5

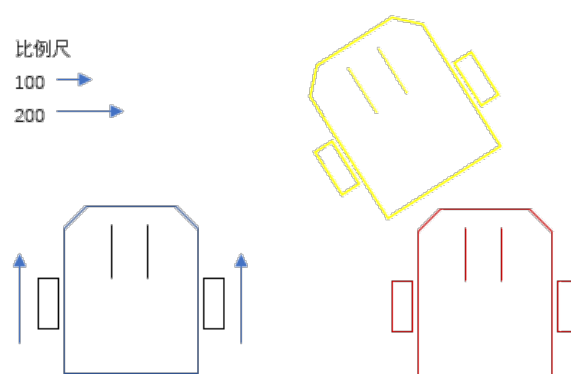


图 6

当左侧电机向前速度 100，右侧电机向前速度 200 时，六足机器人会向左转，并且是一边前行一边左转，如图 6 所示。

当左边电机的速度被设置为 0 时，电机将停止转动，同时将右边电机的速度被设置为 200 时，电机的方向会向前转动，这时六足机器人会向左转，并且是以左边中间的足部为圆心向左转，如图 7 所示。

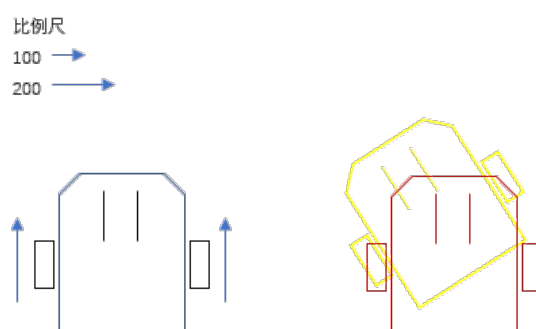


图 7

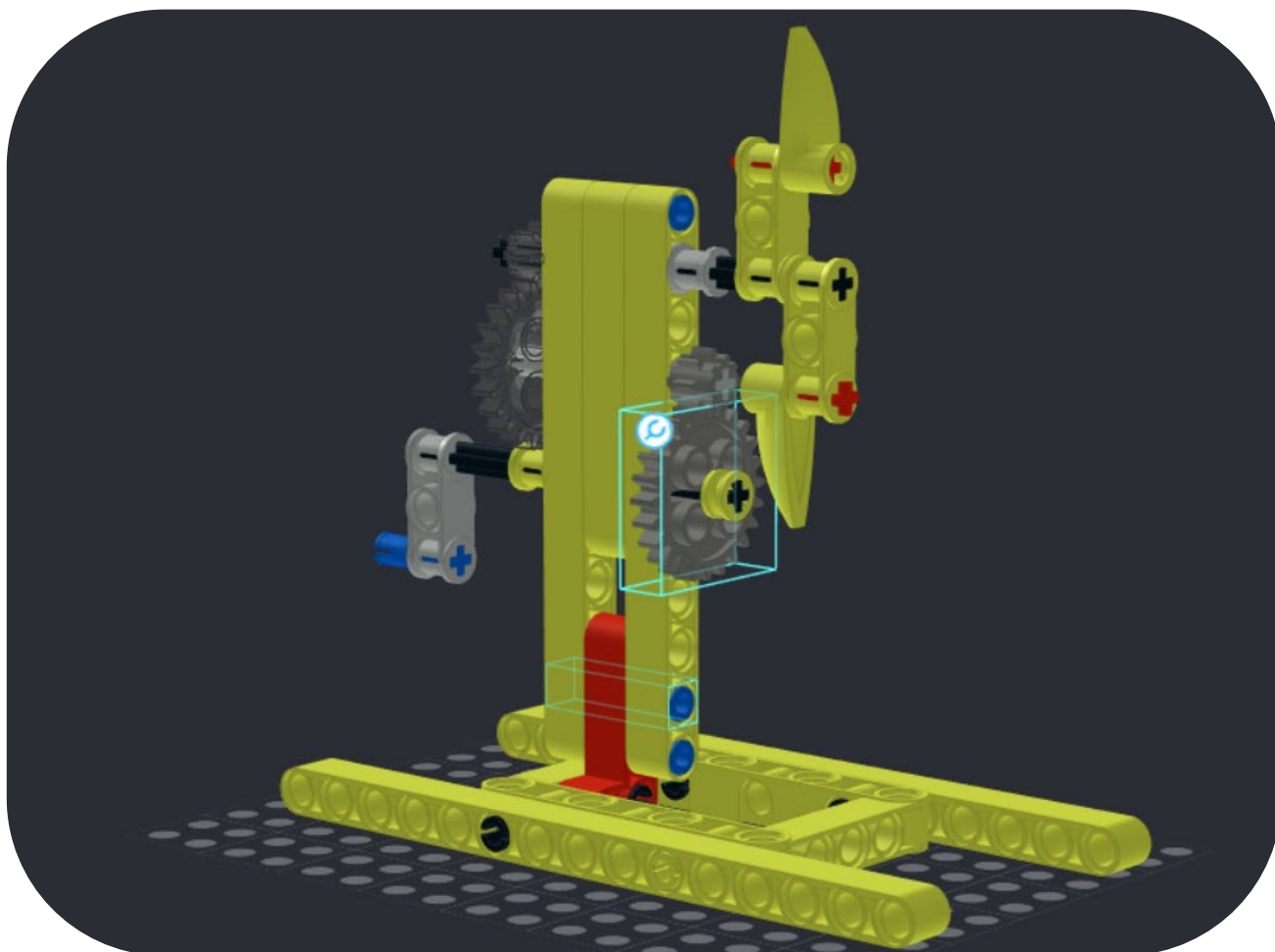
当左边的电机方向 向后退时，右边电机方向向前进，请问六足机器人会如何运动呢？

当一个机器人中两侧电机的转动速度不同时，则可实现转向的功能，这就是所谓的差速转向。

九、手动风扇



1. 知识目标：理解并学习什么是齿轮组
2. 认识齿轮和齿轮传动的特点+掌握齿轮在机械传动中的应用
3. 能力目标：学习了解电风扇的工作原理，并通过作品论证实验原理
4. 情感目标：通过制作模型延伸理解功率的概念



一年四季当中哪个季节最热呀？夏天最热是不是，那夏天的时候大家一般都是怎么给自己降温解暑的呢？吃冰镇西瓜吃冰激凌确实可以解暑，但是记住一次不要吃太

多哦，对小朋友们的身体可不好。除了这些还有什么办法呢？对，开空调开电风扇。今天我们这节课的任务就跟风扇有关哦，这节课我们来研究一下风扇的演变史，有谁知道风扇是怎么演变来的吗？电风扇是从古至今都有的嘛？在电力还不发达的古代，人们是怎么在炎热的夏天给自己降温的呢？

对，在科技不发达的时代，人们用扇子扇风来消暑，但是随着科技的发展，出现了手摇风扇和电风扇也随之诞生(见图 8.1)。今天我们这节课的第一个任务呢就是大家来自由搭建一个可以手摇转动的风扇。

机械风扇最早是安装在天花板上的，1829 年，一个叫詹姆斯·拜伦的美国人从钟表的结构中受到启发，发明了一种可以固定在天花板上，用发条驱动的机械风扇。这种风扇转动扇叶带来的徐徐凉风使人感到凉爽，但每次必须爬上梯子去上发条，非常不方便。



图 1 手动风扇

1872 年，一个叫约瑟夫的法国人又研制出一种靠发条蜗轮启动，用齿轮链条装置传动的机械风扇，这个风扇比拜伦发明的机械风扇精致多了，使用也方便一些。

直到 1880 年，美国人舒乐首次将叶片直接装在电动机上，再接上电源，叶片飞速转动，阵阵凉风扑面而来，这就是世界上第一台电风扇。

本次课程我们就一起尝试制作出一架手摇风扇吧！



教学重点	学习使用结构件拼搭出合适的造型；
教学难点	灵活运用齿轮传动在具体项目中的运用；
教学模式	建构学习，项目式教学。

这个手摇转动的风扇需要哪些部分组成呢？大家可以先思考一下也可以和你的小伙伴讨论一下，讨论好后告诉老师。

扇叶；2.底座；3.支架；4.可以转动的轴。

接下来请大家将你在脑海中建构的风扇模型的设计草图画在活动单上。



现在我们来解密电风扇的传动力：

介绍一个手动风扇中的齿轮组，在这组齿轮传动中，首先是把手带动另一端的大齿轮转动，然后大齿轮带动旁边的小齿轮转动。因为这两个齿轮是平行啮合，所以大齿轮和小齿轮的转速比是 1:3，而这个小齿轮和对面的大齿轮是同轴，所以转速一样，而对面的大齿轮和旁边的小齿轮的转速比又是 1:3,所以最终扇叶的转速就是把手转速的 9 倍。

我们前面说过功的概念，我们转动风扇的时候，在观察转速变化的同时要体会一下手部的力量变化。

我们可以将把手和扇叶的位置调换，这样模型中传动方式就变成了两副小齿轮带动大齿轮的减速装置。此时我们再用同样的速度匀速转动把手，观察把手转动的速度和扇叶转动的速度，同样感受手部力度。

在平行啮合的齿轮中，大齿轮带动小齿轮的加速结构，和小齿轮带动大齿轮的减速机构对比关系如表 8.1 所示，这里小齿轮与大齿轮半径比为 1:3.

齿轮啮合	半径	直径	周长	齿数	转速	传动力	作用
大齿轮带动小齿轮	3: 1	3:1	3:1	3:1	加速 1:3	变小 3:1	加速 装置
小齿轮带动大齿轮	1:3	1:3	1:3	1:3	减速 3:1	变小 1:3	减速 装置

表 1

功率的介绍：这里我们再介绍一个新的概念——功率。功率就是单位时间内所做的功，通常用 P 表示。功率的单位是瓦特，因为功的大小是力乘以距离，而距离又等于速度乘以时间，所以：

$$\text{功率} = \frac{\text{功}}{\text{时间}} = \frac{\text{力} \times \text{距离}}{\text{时间}} = \frac{\text{力} \times \text{速度} \times \text{时间}}{\text{时间}} = \text{力} \times \text{速度}$$

这里的距离指的是：沿力的方向产生的距离，速度指的是沿力的方向移动的速度。

风扇为什么能降温？

风扇其实并不像空调能够产生凉气，风扇只是一个加快空气流动的设备。快速流动的空气能够带走了你体表的部分热量，于是你就觉得凉快了。

当气温比人体表面温度还高时，皮肤没有汗水的情况下用电风扇是毫无用处的，只会感觉更热。因为风扇造成的空气流动，让我们直接接触空气，由于空气更热，所以我们感觉更热。

那夏天吹风扇为什么感觉凉快呢？因为皮肤表层都有汗渍，风扇吹动增加了空气流动，促进了汗水蒸发，汗水蒸发时会吸热带走一部分热量，所以我们就感觉凉快了。



按照搭建步骤完成自己的作品后，尝试活动一下新搭建出的电风扇。注意：本课

程所有案例均为展示示例，鼓励并欢迎各位在此基础上修改优化。

【课堂讨论】风扇的转速，应该考虑哪些因素？具体有哪些措施？

①小组成员之间相互交流、评价（从结构强度、稳定性、使用材料、连接方式以及美观、经济等角度进行交流和优化）

②大组交流、评价（要求每个小组选出一个代表，用几句话说出自己设计的特点。）

思考：那为什么我们吹电风扇就能降温呢？



孩子们投入精力沉迷在项目固然重要，但是停下来反思一下整个项目中发生了什么同样重要。你可以通过提一些关于项目的问题来促进孩子们反思自己的项目。如通常会问“你是怎么想到这个主意的？”这个问题促使孩子们反思到底什么促使和鼓舞了他们。

另外一个是“最让你惊奇的是什么呢？”，这个问题可以让他们不仅仅是描述自己的项目而且能够反思自己的创作经历。如果项目出问题了，要问“你想要实现什么样的功能”在描述他们的目标的过程中，他们经常能够认识到问题出在哪里，不需要教师更多的启发和帮助。

无论对于大人还是小孩，对每个人来说思考都是一件不容易的事情，互相交流关于项目的策略和解决问题的思路是非常有好处的。要更加积极主动和开放的反思自己的想法。把孩子们当作善于创造性思考的学徒，而老师是正在通过动手示范和讨论来帮助他们成为更有创意的工匠。

第四课 幸运大转盘



1. 理解按键按下一次的检测机制；
2. 了解随机数在生活中的用途和使用方法；
3. 制作随机结果的幸运大转盘。



追溯人类赌博史，早在旧石器后期便有“碰运气”的习惯。中国史前文明中大量运用“抓签”筮（shì）卜的方式来判断凶吉，包括重大战事等均通过卜卦来决定。东汉许慎《说文》载“古者乌曹作博”（乌曹是夏桀的臣子），由此看来，赌博在中国的夏朝就有出现。5000年前在伊拉克和印度出现了骰子，而且它的足迹遍布希腊、罗马、中东和玛雅古文明社会。凭借运气决定结果，这种听天由命式的赌博能够大众化的原因

很简单，它体现了一种“天意”的公允和权威，与人类原始意识和宗教观念具有同性关系。因为人们觉得自己永远是运气最好的那个，所以出现了类似大转盘的娱乐产品。本课将制作一个简易的幸运大转盘，用于分配节日小礼物。让我们通过简易机器人来实现它吧！

一、搭建抽奖机造型

观察大转盘的样式特点，利用套件中的乐高积木和传感器搭建一台幸运抽奖机的模型。

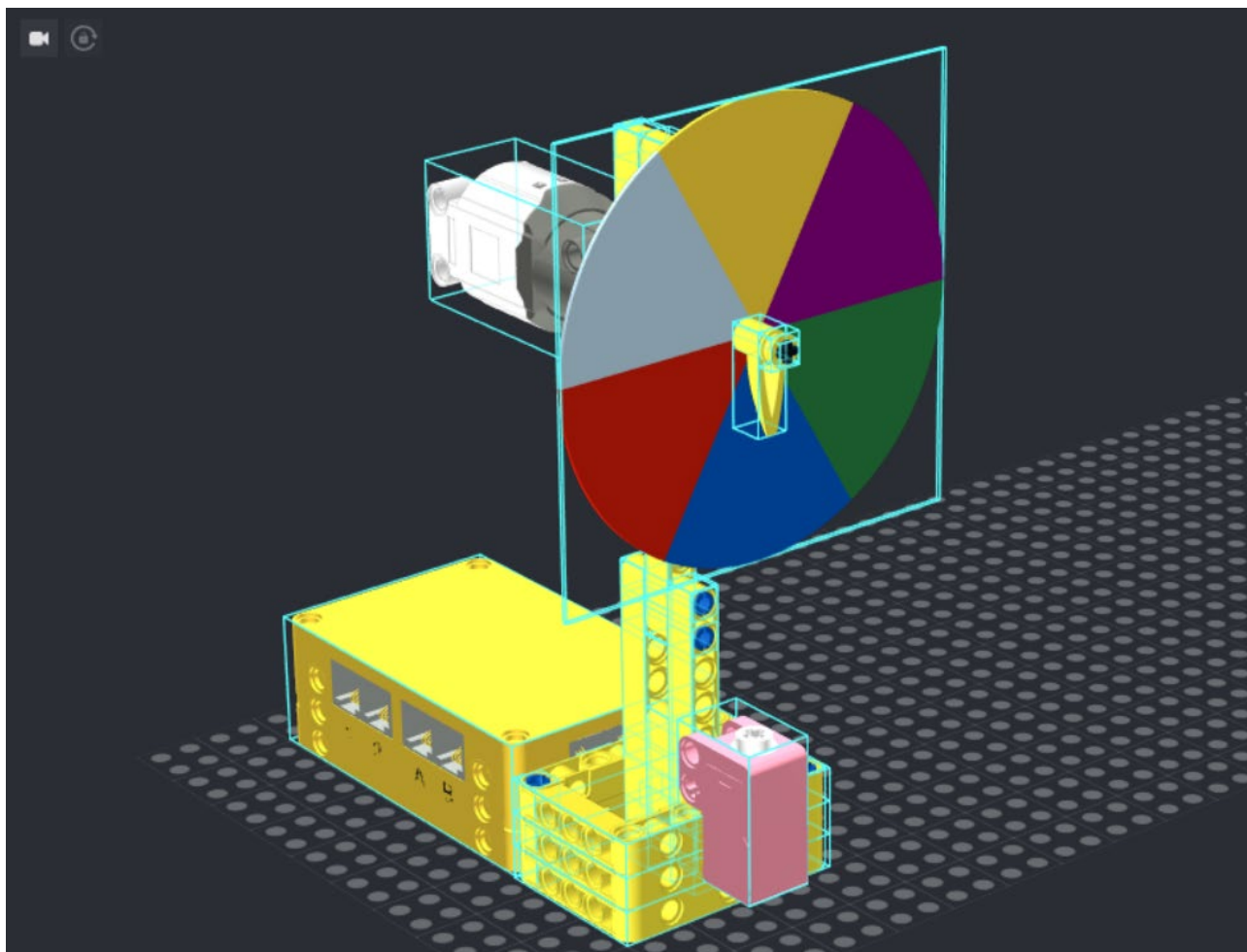
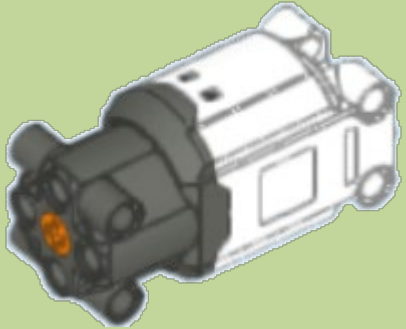


图 1 幸运大转盘造型



本课所用到的电子积木如下表所示：

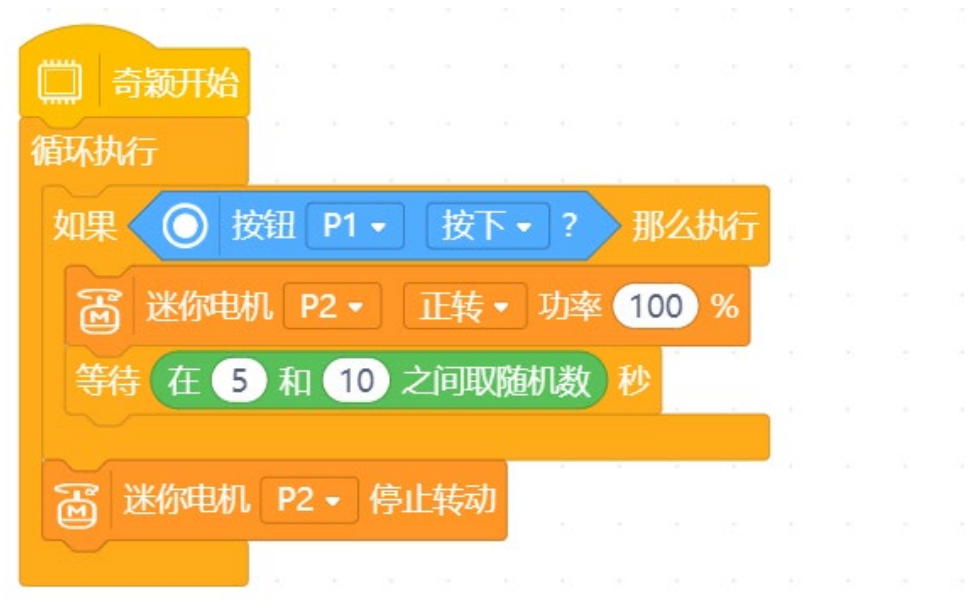
名称	图示	个数	作用
按钮/碰撞开关	 VTMS0010 按钮/碰撞开关 读取按键状态，按键事件执行	1	使用 可以检测触碰开关的状态。
直流电机		1	使电机工作，为幸运转盘的转动提供动力

二、幸运大转盘

搭建完毕后通过编程实现简单大转盘转动与停转的功能，并且利用随机数控制电机停转，最终产生随机的抽奖结果。



运用 Vegeta 编程实现简单的大转盘效果，空格键按下大转盘转动，松开后停止转动。

脚本功能	脚本实现
空格按下电机转动	 The script is written in Scratch and is as follows: 奇颖开始 (When green flag clicked) 循环执行 (Loop) 如果 (If) block with conditions: 按钮 P1 (Button P1) and 按下 ? (Pressed ?), followed by 那么执行 (Then execute). - Inside the if block: 迷你电机 P2 (Mini motor P2) with 正转 (Forward) and 功率 100 % (Power 100 %). 等待 (Wait) block: 在 5 和 10 之间取随机数 (Pick random number between 5 and 10) 秒 (seconds). 迷你电机 P2 (Mini motor P2) with 停止转动 (Stop turning).

使用串口连接控制板，点击绿旗，按动空格键，观察大转盘效果。

三、随机的结果

Arduino 模式下，利用随机数控制电机停转，最终产生随机的抽奖结果。



按下表所示操作步骤进行编程，实现随机抽奖功能。

最终程序如图所示：

- 1.将“按键开关”连接到控制板的 1 号端口，“迷你电机”连接到控制板的 2 号端口
在 Arduino 下对其编程。
- 2.编写如下“测试串口”的程序：



幸运转盘的实现

幸运转盘的原理就是利用触发开关，当检测到的按钮的值与设定的阈值不符之后，让电机随机转动 5-10 秒的时间在停止。

提示：

如果转盘触发条件过于灵敏，可以提高设置的阈值条件。增大阈值。如果要想电机转动的更久，就可以在随机模块更改随机数。

根据幸运转盘的编程思路，按照下面的过程，实现幸运转盘的程序：

最终的程序脚本如图 3 所示：



时间到了，大家都完成了幸运转盘的搭建和程序的编写了呀，接下来我们要梳理下这节课的学习内容。

学习串口实时读取传感器的数值；转盘启动过于灵敏的原因是什么呢？



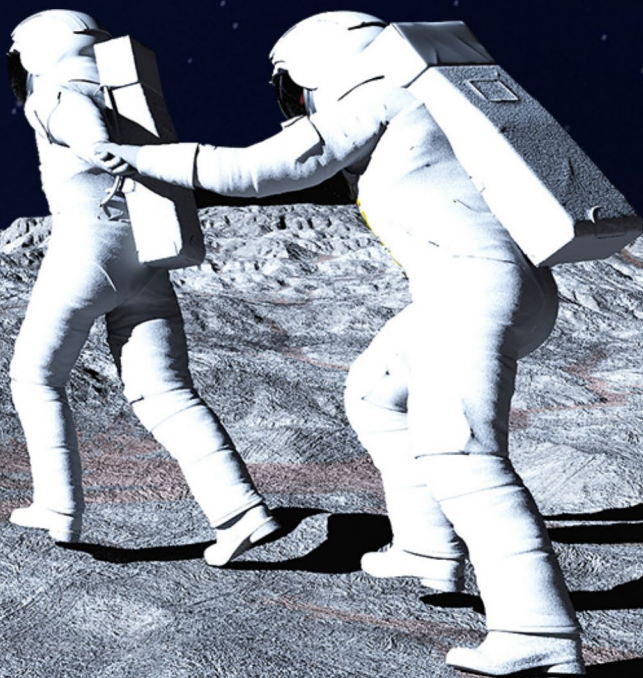
- 1.如果想让幸运大转盘呈现出炫酷的特效（比如灯光效果，音乐效果），该怎么
做呢？
- 2.能否使用按钮控制实现抽奖？



掌控 机器人交互

第四章

探索，前进，从未停止



第一课 模拟呼吸灯



1. 了解 LED 灯的呼吸控制原理
2. 对 LED 灯的呼吸控制进行编程
3. 理解毫秒级时间在 Arduino 中的应用



课前准备：教学基本信息——自备所需零件清单

听说有一种灯叫呼吸灯？难道和人一样呼吸吗？



哈哈！我们可以用奇颖控制器实现一个呼吸灯的效果！

呼吸灯是指灯光在微电脑的控制之下完成由亮到暗的逐渐变化，感觉好像是人在呼吸。其广泛应用在手机上，比如手机有未接来电的时候，为了起到通知提醒的作用，比如说未接来电，未查收的短信等等，呼吸灯就会有节奏的由暗变亮。像呼吸一样那么有节奏，起到一个通知提醒的作用。

电子爱好者可通过简单的电路来实现呼吸灯的视觉装饰效果。



一、视觉残留

视觉暂留现象即视觉暂停现象，又称“余晖效应”，1824年由英国伦敦大学教授皮特马克罗葛特在他的研究报告《移动物体的视觉暂留现象》中最先提出。

人眼在观察景物时，光信号传入大脑神经的过程需经过一段短暂的时间，光的作用结束后，视觉形象并不立即消失，这种残留的视觉称“后像”，视觉的这一现象则被称为“视觉暂留”。

我们可以利用 LED 灯来研究视觉残留现象。



- 1. 将“LED 灯”连接到远程控制板的 1 号端口，在 Arduino 下对其编程。
- 2. 编写如下“闪烁的 LED 灯”的程序：



图 1 闪烁的 LED 灯

小贴士

在脚本区空白区域，鼠标右键可以为程序添加注释，上面脚本中“点亮持续的时间”和“熄灭持续的时间”为注释文字。注释对程序的运行不会产生影响，主要用于对程序进行说明，便于他人理解程序。

3. 分别按照下表设置点亮持续的时间和熄灭持续的时间，上传程序后观察 LED 灯的效果：

点亮持续的时间	熄灭持续的时间	效果
1 秒	1 秒	每 2 秒亮灭一次，非常慢的闪烁。
0.5 秒	0.5 秒	每 1 秒亮灭一次，较慢的闪烁。
0.1 秒	0.1 秒	每 200 毫秒亮灭一次，较快的闪烁。
0.05 秒	0.05 秒	每 100 毫秒亮灭一次，非常快的闪烁。
0.02 秒	0.02 秒	每 40 毫秒亮灭一次，更快的闪烁。
0.01 秒	0.01 秒	每 20 毫秒亮灭一次，几乎看不到闪烁，LED 灯一直亮着。
0.005 秒	0.005 秒	每 10 毫秒亮灭一次，LED 灯完全点亮，人眼无法看到闪烁。
.....		

可以看出，当时间越来越短，LED 灯的闪烁越来越快，当超过一定的闪烁速度后，人眼就看不出闪烁了。这就是视觉残留导致的结果，在熄灭 LED 灯之后，之前点亮 LED 灯会留下“后像”，如果“后像”持续的时间比熄灭的时间长，那么，人眼将看

不到熄灭的过程，所以灯一直处于“点亮”的状态。实际上，灯是有熄灭状态的，只是人眼无法区分而已。

小贴士

到底 LED 灯熄灭多短的时间，才会出现视觉残留？实际上，视觉残留的时间和光的亮度有关，另外，视觉残留的时间因人而异。对于我们的 LED 灯来说，一般时间低于 0.008 秒，就无法看到明显的闪烁了。

二、LED 灯亮度的改变



分别按照下表设置点亮持续的时间和熄灭持续的时间，上传程序后观察 LED 灯的效果：

点亮持续的时间	熄灭持续的时间	效果
0.02 秒	0 秒	LED 灯亮
0.01 秒	0.01 秒	LED 灯较亮
0.005 秒	0.015 秒	LED 灯偏暗
0.001 秒	0.019 秒	LED 灯很暗
0 秒	0.2 秒	LED 灯熄灭

测试结果表明，在同样的 0.2 秒的时间内，点亮和熄灭的时间比例的改变，会影响 LED 灯的亮度。

三、呼吸灯的实现

呼吸灯的原理就是根据一定的节奏改变 LED 灯的亮度，呈现出呼吸的效果，由暗到亮的过程，相当于呼吸中的“吸气”，由亮到暗的过程，相当于呼吸中的“呼气”的过程。



在实现呼吸灯之前，先理清编程思路。首先，设定每一个亮灭周期为 0.02 秒，即亮灭总时间为 0.02 秒，为了实现亮度不断改变的效果，将 0.02 秒平均分为 100 份（即总份数为 100）。假设点亮的份数为 10，那么熄灭的份数为 90；假设点亮的份数为 30，那么熄灭的份数为 70，依此类推，即熄灭的份数为总份数减去点亮的份数，每一份的时间为 0.02 除以 100，因此，点亮和熄灭的时间分别为：

点亮的时间= 点亮的份数 * (总时间 / 总份数)

熄灭的时间=总时间- 点亮的时间

小贴士

熄灭的时间= (总份数- 点亮的份数) * (总时间 / 总份数) = 总时间- 点亮的时间



根据呼吸灯的编程思路，按照下面的过程，实现呼吸灯的程序：

功能	脚本实现	操作说明
创建变量		创建相关的变量。
初始化变量		初始化设置亮的份数为 0，总份数为 100，总时间为 0.02 秒。
呼吸灯亮的时间和熄灭的时间		根据前面的编程思路的公式，递增“亮的份数”，然后计算 LED 灯点亮的的时间和熄灭的时间。

呼吸灯由暗变亮的实现		重复执行，不断递增亮的份数，直到超过总份数。
呼吸灯由亮变暗的实现		同样的思路，递减亮的份数，直到小于0，计算点亮和熄灭的时间。

将呼吸灯由暗到亮的过程和由亮到暗的过程结合后，最终的程序脚本如图 2 所示：

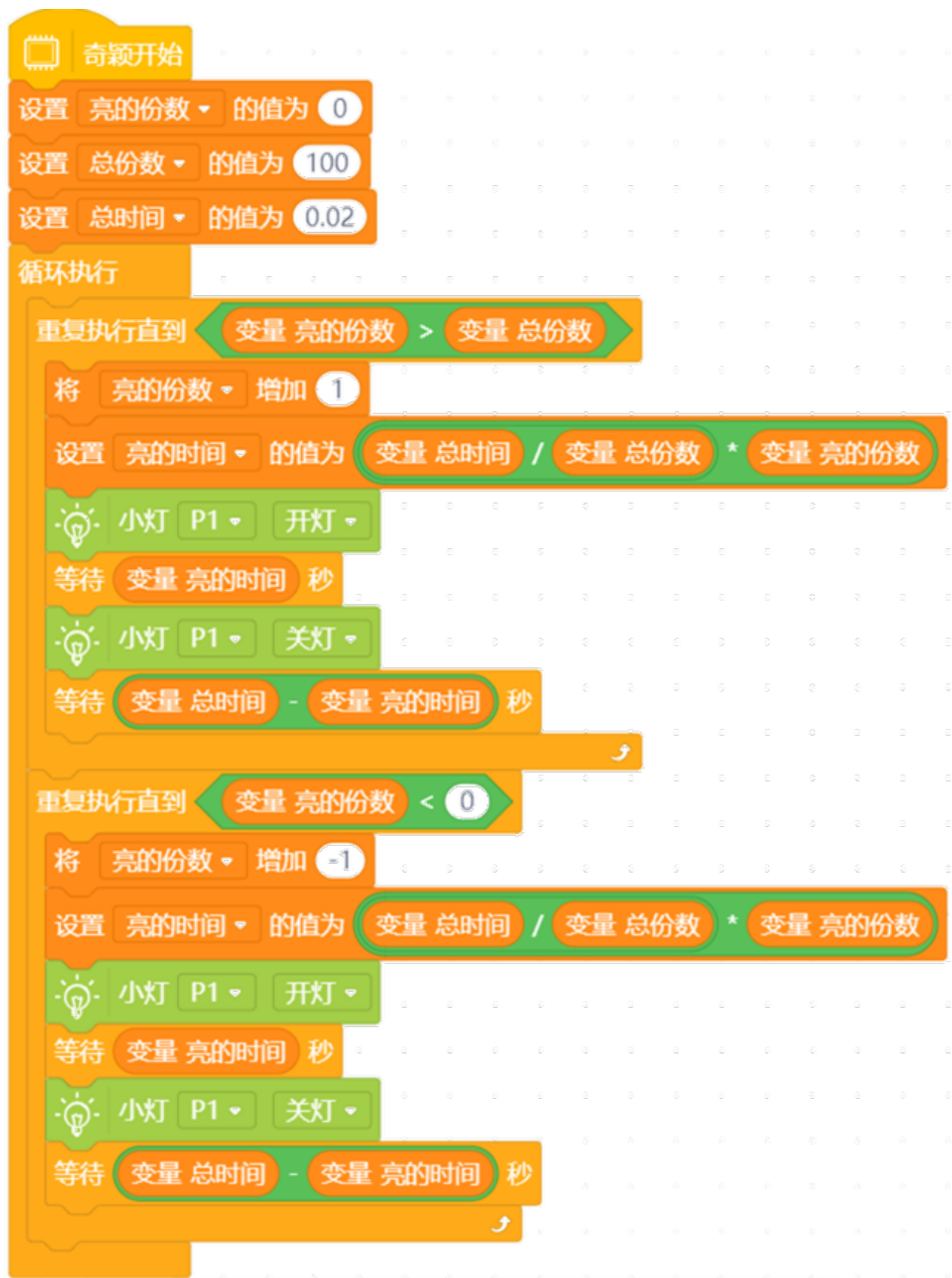


图 2 呼吸灯程序

小贴士

1. 在呼吸灯程序中，大量用到了毫秒级别的时间，对于 Arduino 而言，毫秒级时间是很常见的，但是呼吸灯的程序是无法使用实时模式实现的，因为在实时模式下，模块的执行速度无法达到毫秒级别，所以无法利用视觉残留的相关特性。

2. **等待 1 秒** 模块在上传模式中运行时，最低精度可以达到 1 毫秒级别，也就是说，即使时间设置小于 1 毫秒，实际的延时也是 1 毫秒。因此，在呼吸灯程序中，如果总时间为 0.02（即 20 毫秒），每一份时间为 $20/100 = 0.2$ 毫秒，因此当份数小于 5 的时候，实际上都是同样的亮度。

3. 如果时间精度更高（比如微秒级精度），可以将时间份数分得更多，这样呼吸灯能达到更好的过渡效果，由暗变亮和由亮变暗的过程会更加顺畅。



1. 尝试调整呼吸灯程序中的总份数和总时间参数，观察呼吸灯的变化。

2. 尝试使用 **等待 10 微秒** 模块，实现更顺畅的呼吸灯效果。理解时间精度在 Arduino 中的应用。

第二课 避障机器人



1. 学习使用结构件制作贝吉塔小车的方法；
2. 学习通过两个电机转速比控制三轮小车运行的方法；
3. 实现能够在碰到障碍后自动转弯的避障机器人。



超声波传感器和红外避障传感器都可以感应前方的障碍物，但是由于红外避障传感器距离的限制，可见光中的红外线也会对它有影响，因此超声波传感器是我们的最佳选择。将超声波传感器安装在机器人前方，可以实时监测前方障碍物距离，为机器

人搜索开阔路径。当机器人离障碍物还有一定距离时，超声波传感器就能够检测到相关信息，并据此控制机器人避障。下面让我们一起动手，制作一个属于自己的贝吉塔避障机器人吧。



一、组装贝吉塔小车

发挥想象，利用套件中的乐高积木和传感器搭建出一个你认为最酷炫的机器人小车。

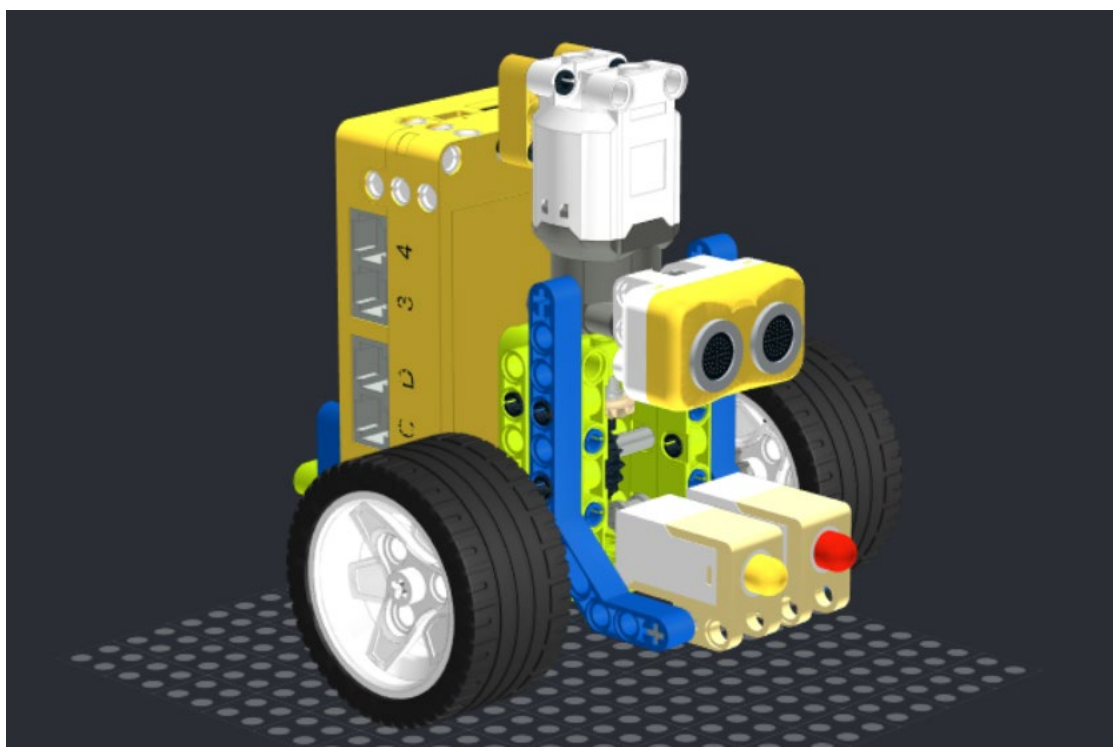


图 1 避障机器人参考模型

搭建步骤、端口连接、注意事项请参考《可编程电子积木搭建手册》。

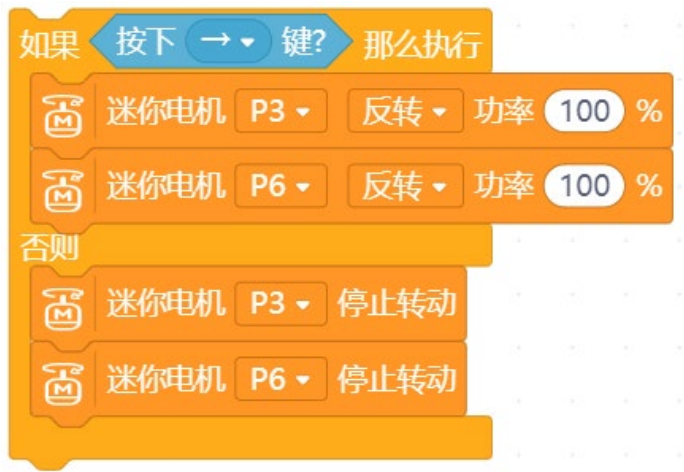
二、贝吉塔动起来

小车搭建完毕，如何让小车动起来呢？这就需要编程控制小车的两个电机。



按照下表所示操作过程，在 Scratch 模式下，编程实现可以用键盘远程控制的贝吉塔小车：

脚本功能	脚本实现	操作说明
按上移键， 小车前进		在“侦测”分类中找到 按键 ▼ 是否按下？ 。
按下移键， 小车后退		将脚本添加到上个脚本的“否 则”判断中。如果下移键按 下，小车后退。
按左移键， 小车左转		将脚本添加到上个脚本的“否 则”判断中。如果左移键按 下，小车左转。

脚本功能	脚本实现	操作说明
按右移键， 小车右转		将脚本添加到上个脚本的“否则”判断中。如果所有按键都没按下，小车停止。
重复监测按键状态		将以上脚本添加到“重复执行”中。

连接贝吉塔，点击小绿旗，使用键盘上下左右键控制贝吉塔运行。

小贴士

由于小车结构中的两个电机是反向安装的，因此在设置前进的时候，两个电机的转向是相反的。同理，当设置后退，左转和右转的时候，电机转向也和正常情况相反。如果发现自己的小车运行情况与预想情况相反，可以修改程序改变两个电机正反转方向，通过调试修正小车运行方向。

三、避障机器人

遥控小车还是需要人工控制才能工作，如果它能在发现障碍物时自动转弯，这才是智能化的体现！接下来，为贝吉塔编程实现自动避障功能吧。



按照下表所示操作过程，在 Arduino 模式下，编程实现自动避障小车：

脚本功能	脚本实现	操作说明
初始化小车前进		将电机控制程序块加入“Arduino 初始化”脚本中。
根据超声波距离探测障碍		如果超声波距离小于 20cm 说明前方有障碍。
如果有障碍，后退再转向		将脚本加入到“如果”判定脚本中，当发现障碍时控制小车转向。
否则继续前进		将脚本加入到“否则”判定脚本中，如果没有发现障碍则继续前进。

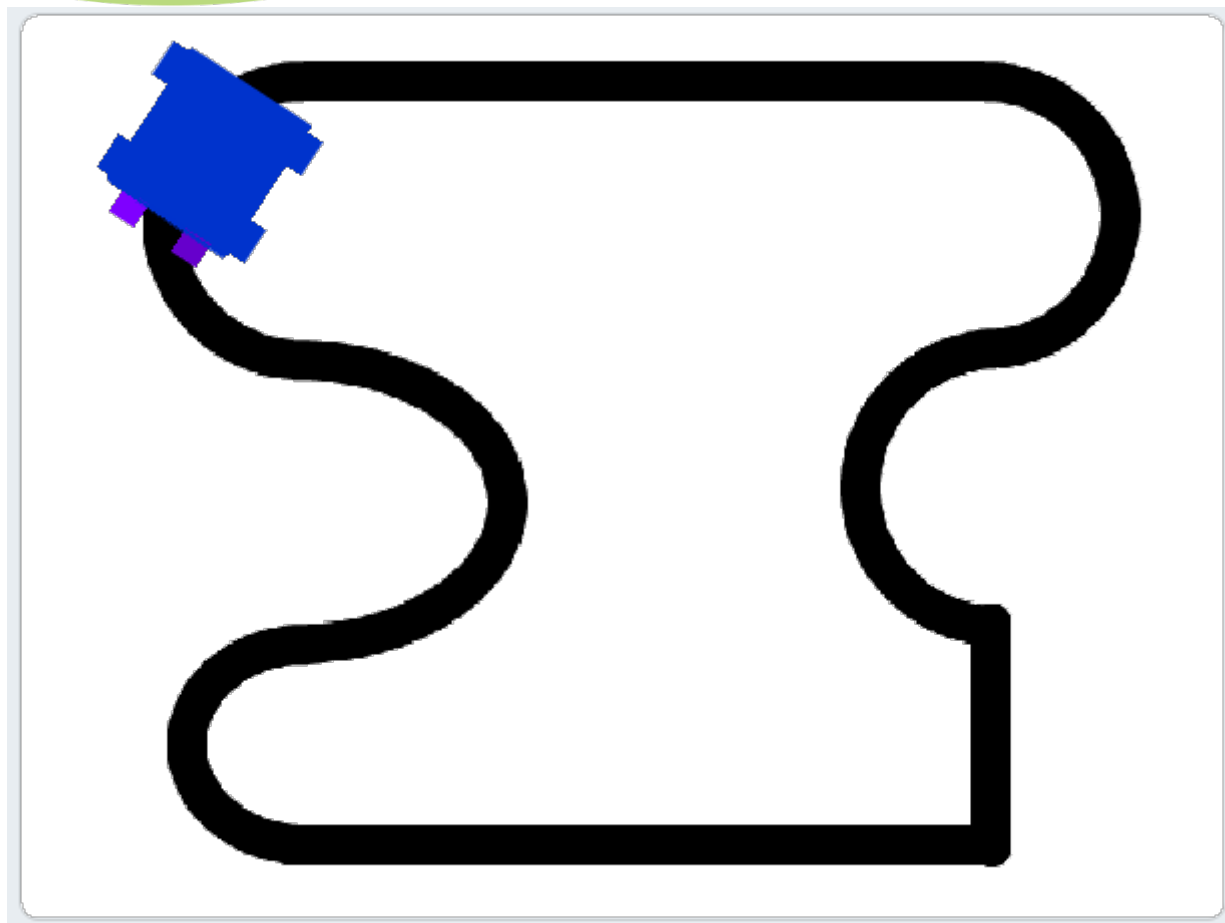
串口连接贝吉塔小车控制板，点击“上传代码”按钮将程序上传到控制板，将贝吉塔小车放在障碍物前，观察小车避障效果。

- 1.如果想让贝吉塔在避障时不是一直朝右转向，而是随机的左转或者右转，该如何编写这个程序呢？
- 2.如果想制作一个倒车雷达，即贝吉塔在开始过程中迅速前进，车速随着小车与前方障碍物距离的减小而减小直到停止，该如何编写这个程序呢？
- 3.改变算法，使贝吉塔机器人转弯时直到发现出口才向前走，否则一直转弯直至发现出口。

第三课 循迹机器人

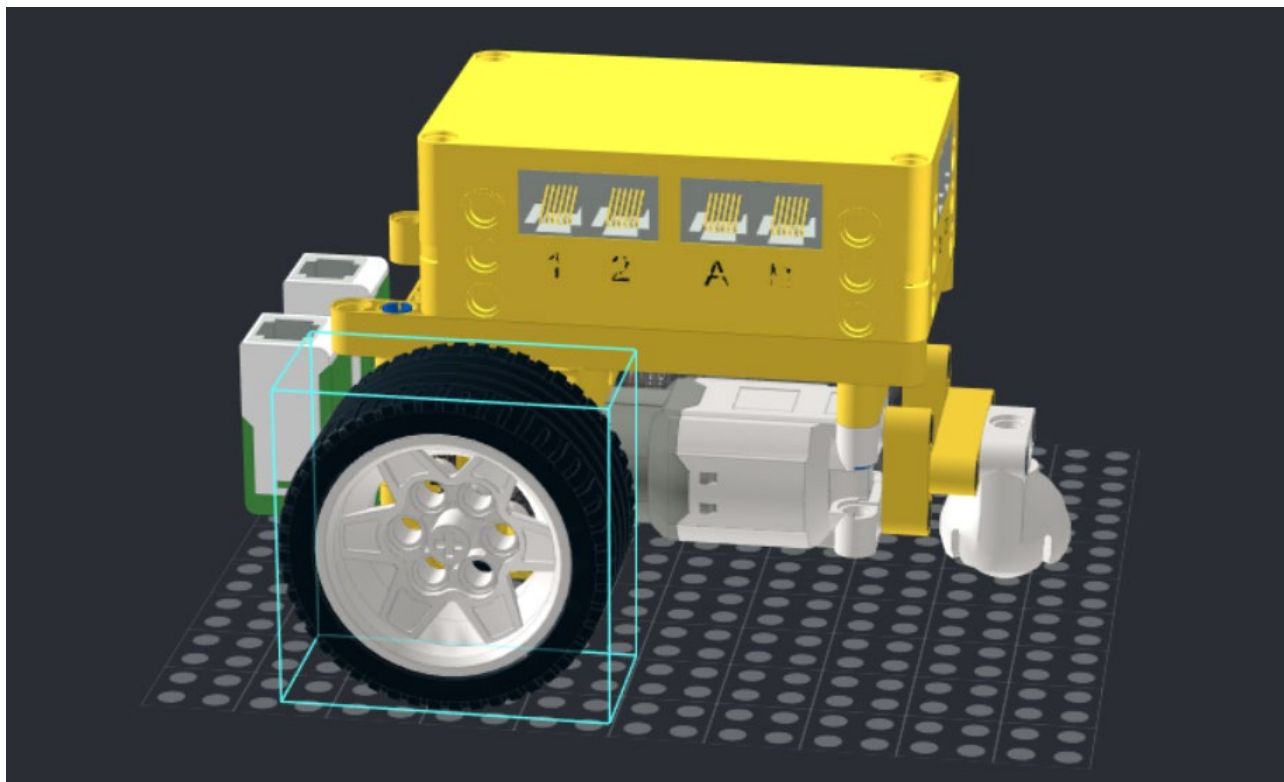


1. 学习红外循迹传感器的原理及使用方法
2. 学习黑线轨迹探测的方法和技巧
3. 编程实现能够沿黑线自动运行的机器人



循迹机器人是一种能够自动按照给定路线进行移动的机器人，它是一个实现路面探测、障碍检测、信息反馈和自动行驶的技术综合体。循迹机器人在军事、民用和科学研究等方面已得到了广泛的应用。例如自动化生产线的物料配送机器人，医院的机

机器人护士，商场的导游机器人等。在本课中，我们可以对贝吉塔小车进行编程，实现一个自动沿黑线行走的循迹机器人。



一、红外循迹传感器

为了实现贝吉塔的自动循迹功能，我们应当使用“红外循迹传感器”。“红外循迹传感器”有两个间距为 0.3cm 的红外探测器，负责发射和接受红外线，利用红外线在不同颜色物体表面具有不同反射强度的特点，在小车行驶过程中不断的向地面发射红外光，当红外光遇到白色纸质地板时发生反射，反射光被装在小车上的接收管接收；如果遇到黑线则红外光被吸收，小车上的接收管接收不到红外光。因此，我们利用这种原理来判断外界的颜色是黑色还是白色。通过循迹让小车能够沿着黑线（粗线或细线）运行，地图如图 1 所示：

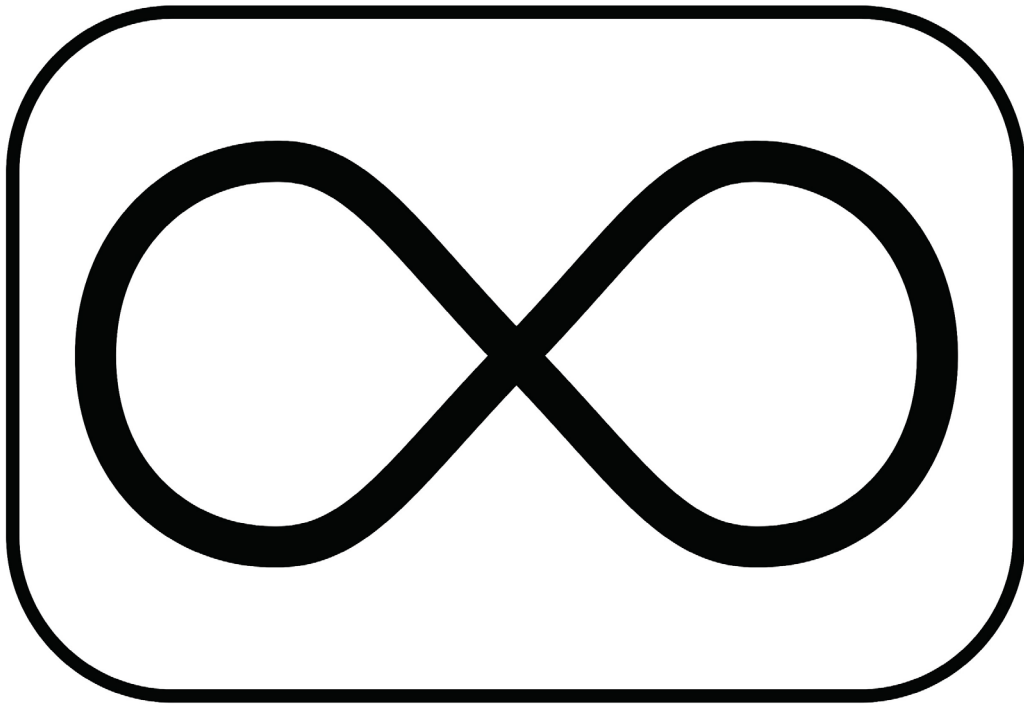


图 1 循迹地图

小车循迹运行的基础是检测前方纸片是黑色还是白色。



图 2 红外循迹传感器

如图 2 所示，在红外循迹传感器上有 1 和 2 两个探测器，它们能够感知探测器下方区域为黑色还是白色，现在将贝吉塔放在地图上，让前方的“红外循迹传感器”探头分别对准白色区域和黑色区域，观察 1 和 2 号探测器状态指示灯的状态。



如图 3 所示，根据实验的反馈，可以得到结论：

当循迹 1 和循迹 2 都不在黑线内-----小车应该直行；

当只有循迹 1 在黑线内-----小车应该左转；

当只有循迹 2 在黑线内-----小车应该右转；

当循迹 1 和循迹 2 都在黑线内-----小车需要停止。

二、自定义程序块

接下来要在 Arduino 模式下编程，这样贝吉塔就可以脱离电脑自由运行了。

为了让程序更加简单明了，可以利用自定义程序块的功能：在“更多模块”分类下，有一个“新建程序块”按钮。利用这个功能，我们可以根据需要定制自己的程序块，如图 4 所示。



图 4 自定义模块界面

自定义模块可以创建自己指定的程序块，该程序块的功能可以由其他程序块组合实现，便于将重复使用的多个程序块封装为一个自定义程序块，简化程序的复杂度，方便阅读和理解。

在本课中，贝吉塔的“前进”功能需要分别控制两个电机正转和反转，如果程序中需要多次使用“前进”功能，那么程序会变得复杂和臃肿，如果将两个电机的控制功能封装为一个自定义的“前进”模块，可以简化程序的编程，并且让程序逻辑更加清晰。创建“前进”程序块的过程如下文所示：

首先，点击“更多模块”分类下的“新建程序块”。

在“程序块名称”处输入“前进”，表示“前进”模块，点击“确定”按钮，界面弹出“程序块编辑器”，可以在编辑器中编辑对应功能脚本，如图所示：



将电机驱动控制脚本拖到“前进”下面，然后点击“确定”按钮即可。

利用这种方法，创建前进、后退、左转、右转、停止五个新的程序块，分别控制小车的五个动作。

三、循迹机器人



按照下表操作过程，在上传模式下，编程实现根据红外循迹传感器状态指挥小车运动的功能：

脚本功能	脚本实现
如果 1、2 号传感器都不在黑线内，前进	定义 前进 迷你电机 P1 反转 功率 100 % 迷你电机 P2 正转 功率 100 %
如果只有 1 号传感器在黑线内，左转	定义 后退 迷你电机 P1 正转 功率 100 % 迷你电机 P2 反转 功率 100 % 定义 左转 迷你电机 P1 正转 功率 100 % 迷你电机 P2 正转 功率 100 %
如果只有 2 号传感器在黑线内，右转	定义 右转 迷你电机 P1 反转 功率 100 % 迷你电机 P2 反转 功率 100 %
如果 1、2 号传感器都在黑线内，后退	贝吉塔开始 循环执行 如果 红外循迹 P1 <= 0 与 红外循迹 P2 <= 0 那么执行 后退 如果 红外循迹 P1 = 1 与 红外循迹 P2 <= 0 那么执行 左转 如果 红外循迹 P2 = 1 与 红外循迹 P1 <= 0 那么执行 右转 如果 红外循迹 P1 = 1 与 红外循迹 P2 = 1 那么执行 前进

如何让贝吉塔机器人按照规划好的路线自行行动呢？我们已经能够方便的控制小车前后左右运动，再结合红外循迹算法，尝试让贝吉塔沿黑线前进吧。

串口连接贝吉塔小车控制板，点击“上传代码”按钮，将程序上传到控制板，把贝吉塔小车放到地图外圈的细线轨道上，观察贝吉塔运行情况。

思考：如何让贝吉塔沿地图上的黑色宽线部分运行？

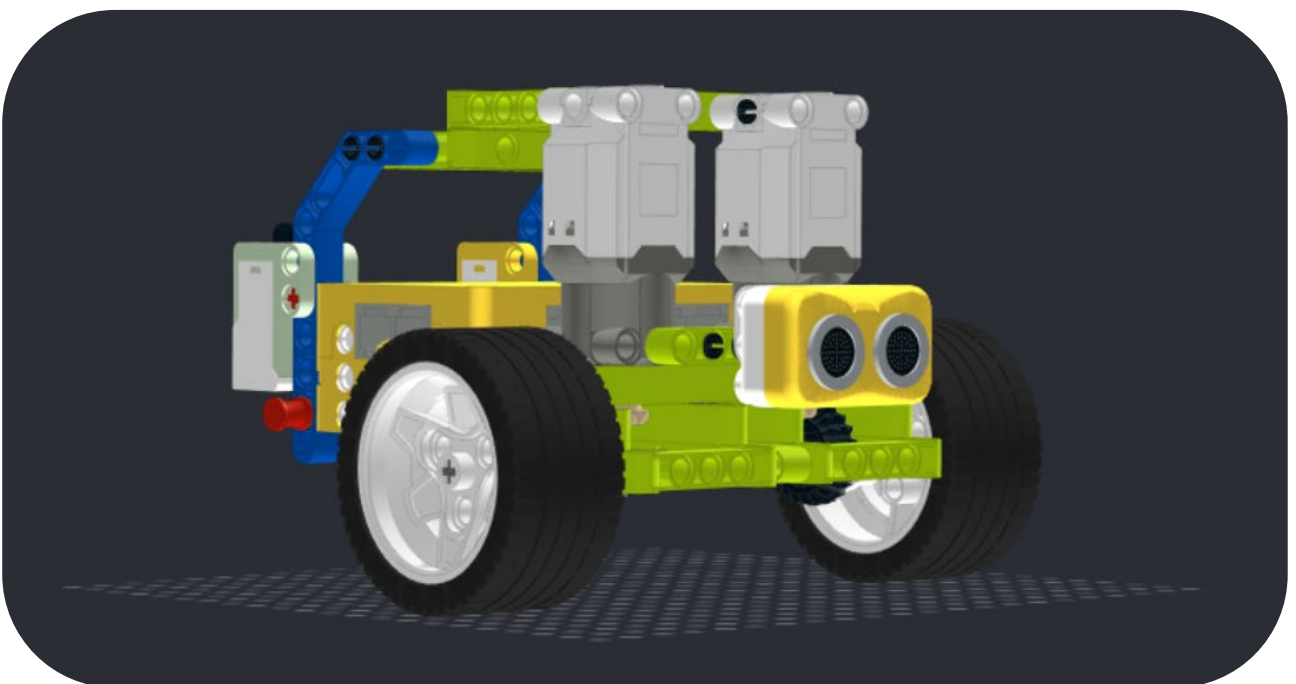
第三课 遥控警车

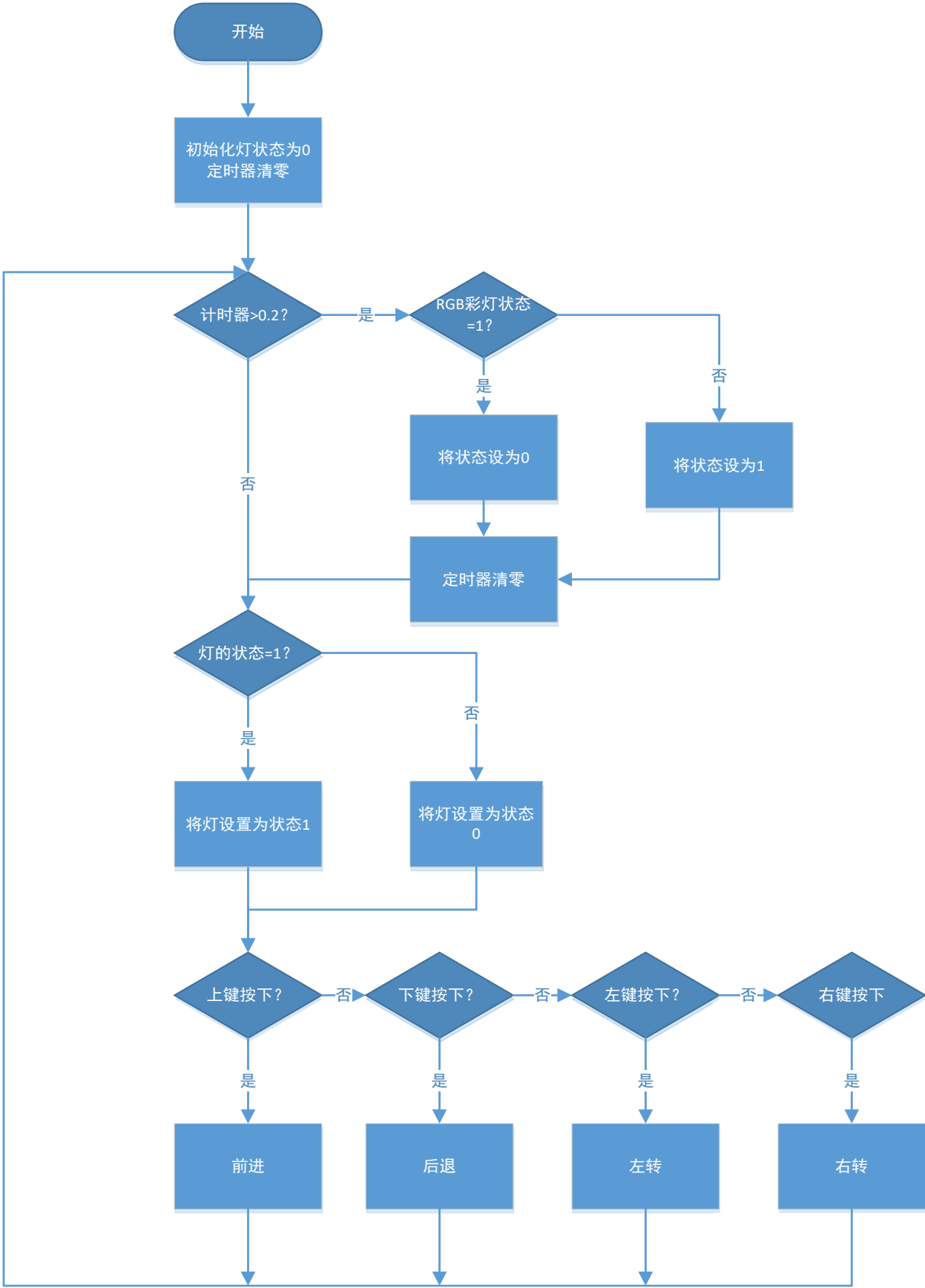


1. 学习红外遥控功能的使用方法；
2. 理解遥控器查询与警车闪烁的时序问题；
3. 理解计时器的概念，掌握计时器在 Arduino 中的应用；
4. 理解变量在状态或标志记录中的应用，深入理解和掌握变量的用法。



商店的遥控车是孩子们最喜欢的玩具之一，使用遥控器可以方便的控制小车前后左右前进，有些遥控车还带有警灯功能，简直跟真的警车一样酷炫。那么，你知道它是怎么实现遥控功能的吗？在本课中，就为大家揭开谜底，一起动手制作一辆带有警灯的遥控警车。





一、使用计时器实现闪烁的灯光



警灯由红蓝两色灯组成，两种颜色灯光快速切换就能实现酷炫的闪烁效果。在软件中可以利用“RGB 全彩灯”模块来实现灯光颜色的切换闪烁。

假设红蓝灯光分别点亮 0.1 秒，那么程序很容易使用 **等待 1 秒** 模块实现。

将 RGB 全彩灯连接到远程控制板的 1 号端口，编写脚本如图 2 所示：



图 1 RGB 灯红蓝交替

除了使用 **等待 1 秒** 实现灯光交替效果，还可以使用计时器实现。计时器是一个很常用的功能，能实现一些 **等待 1 秒** 实现不了的复杂逻辑。

在  分类中，可以找到  和  模块，计时器用于记录

程序运行的时间，通过计时器归零重新计时，可以很容易实现精确的延时操作。

计时器的好处是，远程控制板内部会自动计时且不会等待，只是查询当前的时

间，而 **等待 1 秒** 模块会阻塞程序的执行。

脚本功能	脚本实现	操作说明
初始化计时器		初始化的时候，将计时器归零，开始计时
点亮红灯		计时器 模块会返回当前的运行时间，直接可以用来比较，也可以使用变量保存后使用。如果计时器时间大于 0.1，点亮红灯
点亮蓝灯		如果计时器时间大于 0.2，点亮蓝灯，并且将计时器归零，重新开始计时，当再次达到 0.1 的时候，会切换到红灯，依次类推，就能实现红蓝灯光的交替切换效果

观察贝吉塔上 RGB 全彩灯的闪烁效果。

二、遥控警车

现在贝吉塔已经拥有了酷炫的警灯，如何实现遥控功能呢？尝试利用红外遥控器和红外收发器来实现对警车的遥控吧。



红外收发器连接在贝吉塔小车的 2 号端口上。按照下表所示操作过程，在 Arduino 模式下，编程实现遥控警车（功率可以自己调整把控）：

脚本功能	脚本实现	操作说明
如果遥控器“上”键按下，小车前进		在“通信功能”中找到  , 将脚本放进“循环执行”中，判定红外遥控器按键是否按下。
如果遥控器“下”键按下，小车后退		将脚本放在否则的判定中，“上”键未按下的情况下再判定下键是否按下。
如果遥控器“左”键按下，小车左转		将脚本放在否则的判定中，如果上下键都未按下则判断左键是否按下。

如果遥控器“右”键按下，小车右转，没有按键按下则停止		将脚本放在否则的判定中，如果上下左右键都没有按下，就控制小车停止。
----------------------------	---	-----------------------------------

遥控代码如图 2 所示（单独测试记得增添重复执行框架）：



图 2 遥控警车程序

串口连接贝吉塔控制板，点击“上传代码”按钮，将代码上传到控制板，利用红外遥控器控制贝吉塔警车。

小贴士

1. 使用程序处理快慢不一致的任务切换时，计时器是必备的工具。

2. 合理的使用状态变量能简化复杂的程序，分析清楚时间切换的时序是写好这类程序的关键。

3. 微型计算机的工作原理本质上也是通过时间段的切分来完成的，当时间段的切分足够多，处理器的速度足够快，就能实现并行完成很多任务的效果。

三、计时器的妙用

在实验遥控警车效果时，我们发现贝吉塔小车有时会反应迟钝。仔细分析程序就可以找到原因：警灯闪烁是每 0.2 秒一次，这样每次主循环至少需要 0.4 秒时间才能查看一次红外遥控器状态，因此反应迟钝。要解决这样的问题，可以利用**计时器**和一个变量来实现非堵塞的等待。

根据以下表格的操作过程修改程序，尝试利用计时器解决等待问题：

脚本功能	脚本实现	操作说明
创建新变量，记录 RGB 灯的状态		在“变量”分类中，利用 新建变量 创建一个“RGBState”变量。

初始化变量及定时器		将变量初始值设定为 0，同时将计时器归零。
每 0.2 秒切换一次 RGB 灯状态		计时器每计 0.2 秒后归零，重新计时，同时根据 RGBState 状态切换。
根据变量切换 RGB 灯状态		如果 RGBState=0,那么 1、3 号灯亮红色，2 号灯亮蓝色；如果 RGBState=1,那么 1、3 号灯亮蓝色，2 号灯亮红色，实现每 0.2 秒灯切换一次状态效果。

最终的程序脚本如图 3 所示：

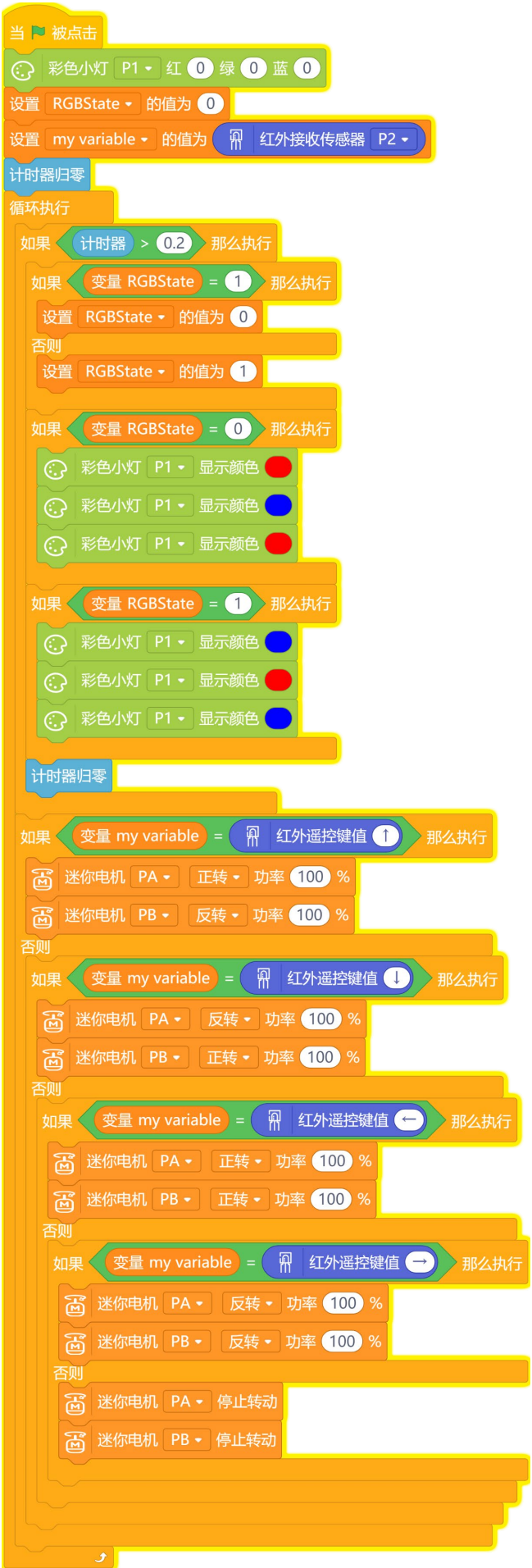


图 3 遥控警车脚本

第四课机器人足球挑战赛

一、机器人工程控制赛

（一）比赛简介

本赛项是为了丰富青少年电子信息相关知识，培养青少年创新思维方法，锻炼青少年动手制作能力，发掘青少年电子信息人才而设立的，主要考核参赛选手对电子信息、智能控制等相关知识的综合运用。

本竞赛项目参赛选手基于智能硬件和编程软件两大工具，利用智能硬件及提供的材料完成外观设计，通过编写程序，对搭建的作品进行控制，完成比赛项目设定的任务。

（二）比赛主题

“创意电子 智造生活”

（三）比赛题目

智能避障车

（四）比赛流程

小学组比赛分为两个环节。第一个环节为现场制作：参赛选手使用自备器材进行计算机编程与作品搭建与调试，时间为 120 钟。第二个环节为竞技展示：在竞技区内进行比赛，时间为 3 分钟。

注意：只有完成现场制作环节（计算机编程和作品搭建）才可进入下一个竞技环节。

（五）比赛规则

现场制作环节：

1.作品搭建：围绕智能避障车的题目利用自备及器材进行作品搭建。

2.作品要求：作品外观符合车辆的特征，尺寸不能超过长 250mm*宽 150mm*高 150mm，必须包含 RGB 灯传感器、触碰传感器、侦测障碍（超声波传感器）、可编程控制器、马达电机块等电子模块。场地内设置测试区，供选手对作品进行调试。

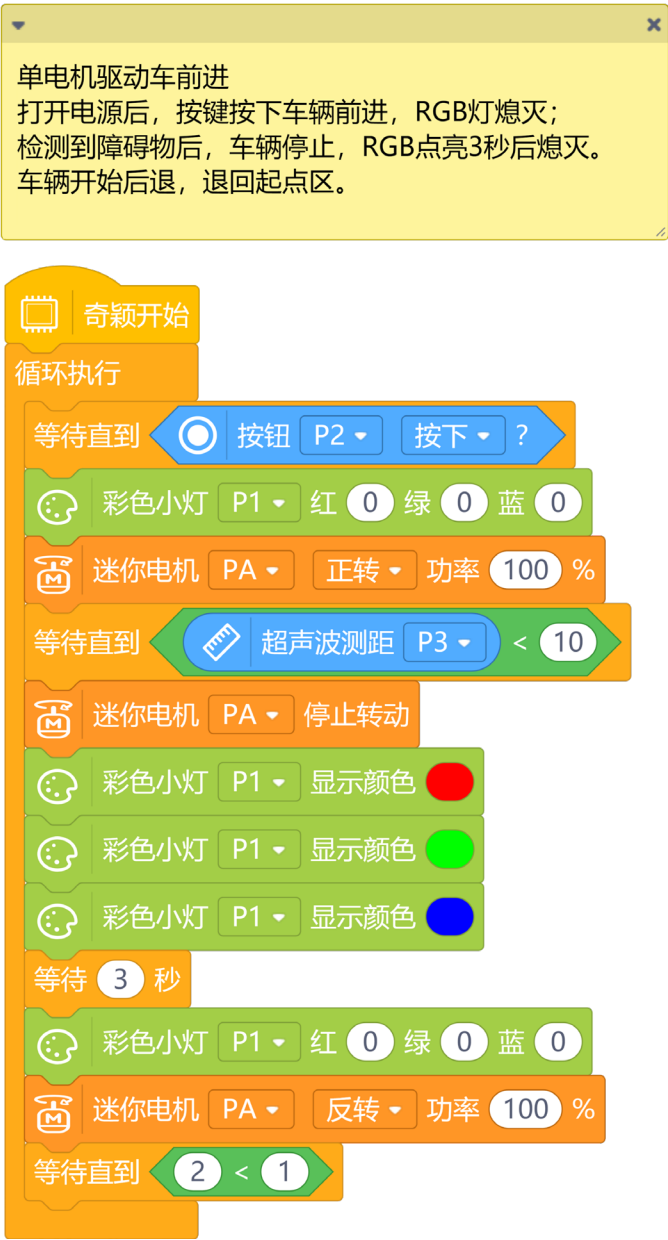
3.程序要求：打开电源，按下按键后车辆前进，RGB 灯熄灭；当检测到障碍物，车辆停止，RGB 灯点亮 3 秒（灯珠颜色自定）后熄灭，车辆开始倒退，一直退回到起点区，任务完成。

4.任务书填写：参赛队员在搭建完成后需完成任务书的填写，包括本小组作品的程序流程图、作品创新点（至少 2 条）等（详见附件一）。

5.评分内容：作品搭建完成后评审会根据任务书以及评审问答两方面进行评分（详见评分标准）。

竞技展示环节：

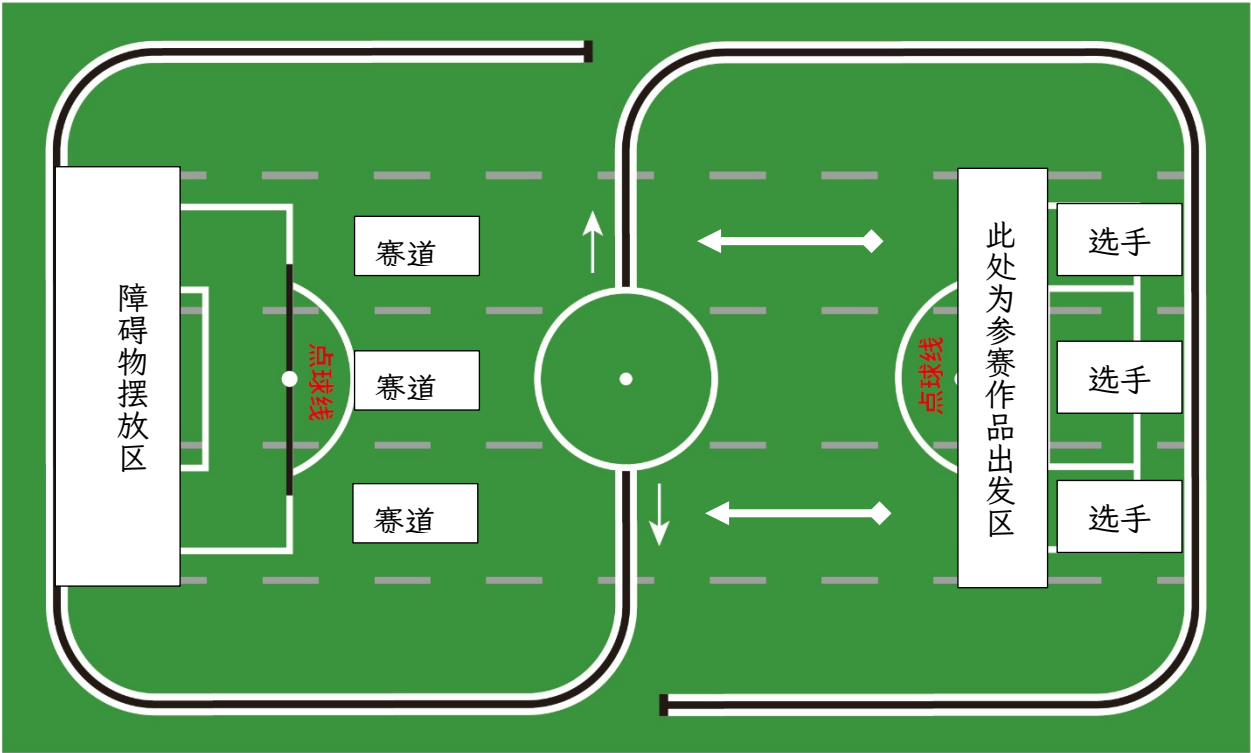
1.竞技要求：比赛开始前，在起点区域放置所搭建车辆，做好准备，裁判信号响起时，进行启动，RGB 灯熄灭，车辆前进，须在 60 秒内检测到障碍物，车辆停止，RGB 灯点亮 3 秒后，RGB 灯熄灭，车辆开始后退，并退回到起点区，完成任务。竞技环节中不可以调试车辆程序。程序参考（单电机走的直）：



2.人员要求：比赛开始前，每组推荐一名队员进入竞技场地，比赛过程中队员不可以触碰正在竞技的车辆。

3.比赛机会：在竞技时间内，每组有两次比赛机会。当竞技车辆偏离赛道（车身超过赛道 $\frac{1}{3}$ ），队员可以示意裁判重新开始。如果本组车辆被偏离赛道车辆碰撞无法继续比赛，该组队员可以示意裁判重新开始，不计入比赛机会。

4.竞技场地：



5.评分内容：（详见评分标准）。

（七）评分标准

评分分为两部分共 100 分，其中现场制作环节满分 40 分，竞技展示环节满分 60 分。

现场制作环节评分标准

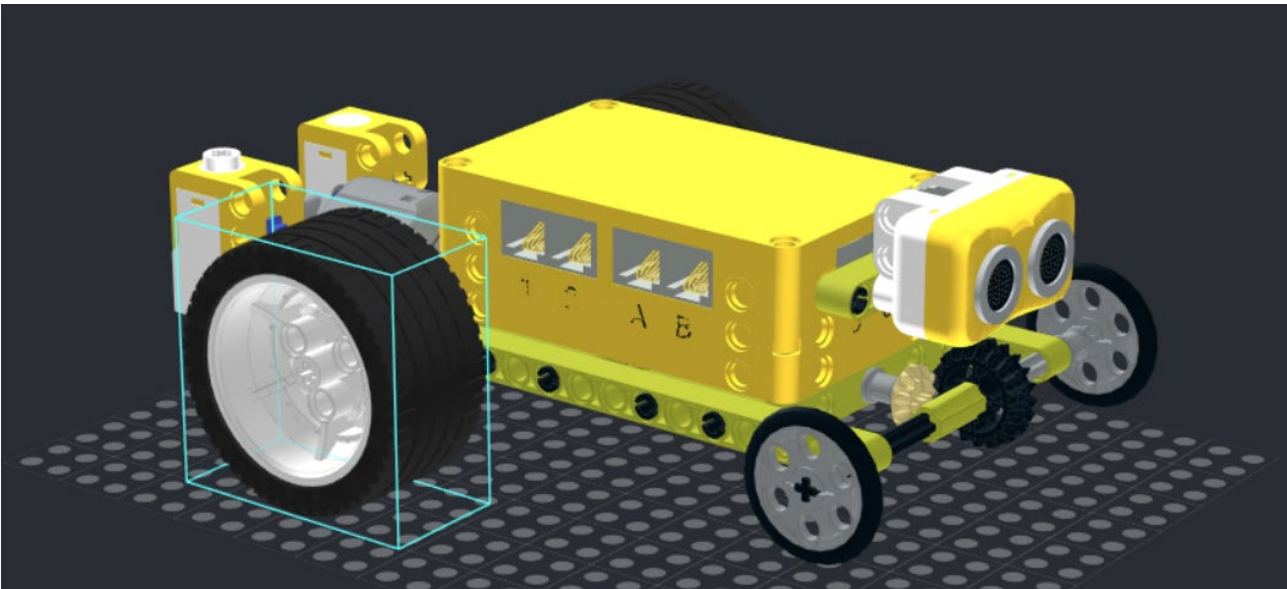
序号	评分项	标准	分值
1	任务书	流程图合理，创新点合理	10
2	评审问答	1．结构新颖，设计巧妙，有一定的创新。制作过程中工具和相关器材使用规范。 2．各类传感器应用合理，程序编写逻辑合理、规范。	30

		3. 现场操作娴熟，演示过程完整；作品陈述语言精炼准确，答辩思路清晰；回答问题逻辑严谨思维清晰、分工明确。	
--	--	---	--

竞技展示环节评分标准

序号	任务完成情况	扣分
1	任务完成	0
2	车辆去程压线（没超过车身三分之一）	5
3	车辆回程压线（没超过车身三分之一）	5
4	RGB 灯没按要求亮	5
5	RGB 灯没亮	10
6	车辆检测到障碍物停下过程中与障碍物有碰撞	10
7	完成去程任务（没压线，检测到障碍物停下且与障碍物无碰撞，没有完成回程任务）	30
8	车辆偏离赛道（车身超过赛道三分之一）	60
9	车辆碰到障碍物没有停下，继续行驶	60

（八）车体形态



(九) 附件：任务书

智造机器人全球足球挑战赛

机器人工程控制赛任务书

日期：__年__月__日 参赛人员：_____ 组号：_____

机器人编程与作品搭建部分			竞赛展示部分（60分）	总分	裁判签字
任务书（10分）	制作时间（20分）	团队协作（20分）			

1.请参赛选手简要说明主题及创新之处（至少两条）。

2.请在虚框处画出本作品的程序流程图。

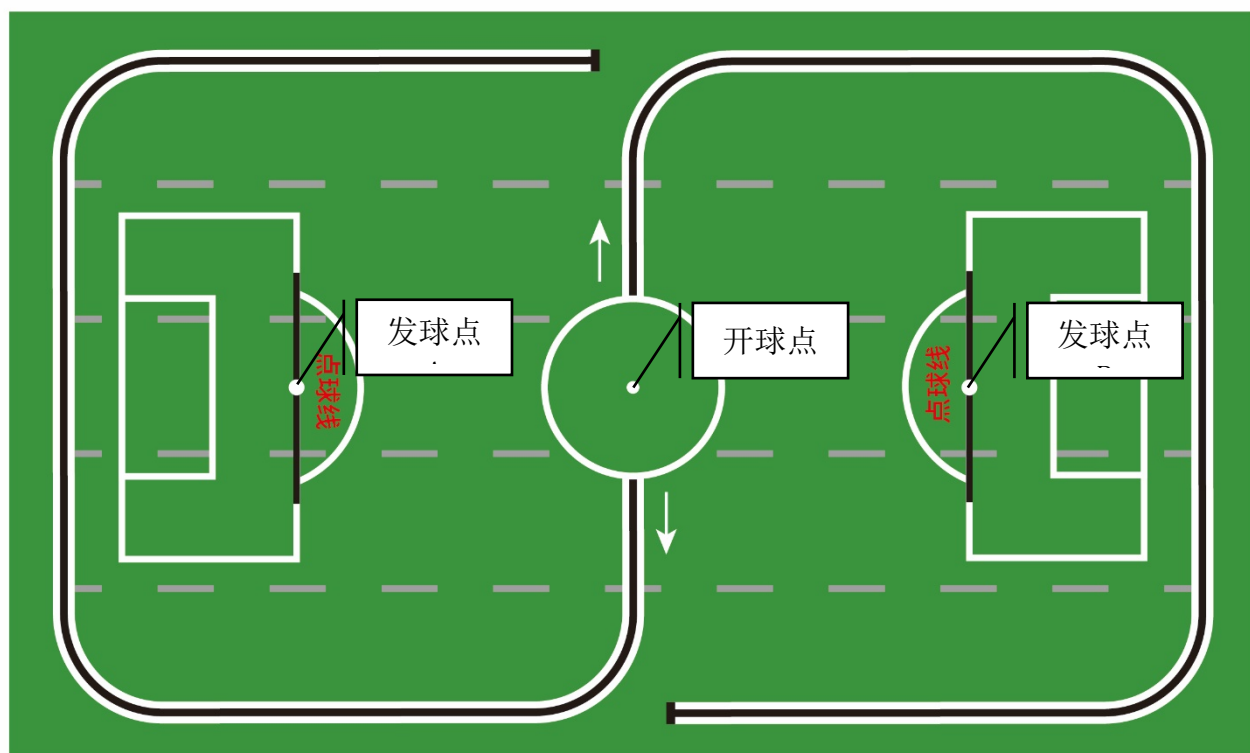
二、点球机器人大赛（含守门员）

（一）比赛简介

机器人点球大赛具有传统机器人足球的对抗性、竞技性和游艺性，也更能激发儿童的创造力，同时能够训练参与者控制力和逻辑思考能力等。其宗旨是培养青少年具有国际视野和科技创新能力，促进跨科技和文化领域的友好交流合作，让热爱机器人活动的青少年能够体会到更多的乐趣，同时了解到多元化的机器人发展形式。

（二）比赛场地和足球

点球大赛赛台的球场区长 2M、宽 1.2M。赛台尺寸如图所示。赛台用刀刮布制成。足球采用仿真皮质足球。绿色球场应水平和平整。球门由 APS 塑料及 PVC 管结构搭建而成，组装尺寸为 55.5x28x35cm。



（三）比赛说明

此赛内容为从开球点控制参赛作品启动，到达发球点后通过遥控控制发射动作，将足球踢进球框内。

发球点与开球点：场上有两个发球点和一个开球点，图中有文本说明，在场地纸上由白点标记。（每场开赛前，裁判都要检查足球是否损坏）

机器人尺寸规格：机器人将在直立和所有部件完全伸展的状态进行测定。（直立的机器人必须能自由纳入内径为 150mm 的开口圆筒中，机器人的高度必须小于 150mm、重量不得超过 1kg）

搭建：机器人部件不可以使用胶水、螺钉等永久固定。可以用扎带或胶带固定电线。

如果机器人具有活动部件，并在前后或左右的方向上延伸，它需要操作这个部分将其运转起来进行检查。部件活动时，机器人必须不接触量筒。

（四）比赛规则

1. 在比赛当天，机器人的所有部件必须是零散状态（出厂模式），比赛前将给予指定的安装时间，机器人应在指定的安装时间内被组装起来，机器人可以在比赛的时间内进行修改。（以组委会规定为准）

2. 机器人必须为遥控方式控制，一个机器人由一名参赛选手控制。

3. 参赛选手须装饰或标记自己的机器人，能明确区分出它们属于哪一个团队，这些装饰与标记不能影响比赛，并且不会在高度上予以限制。

4. 机器人在射球时，必须能够释放球，有明显的“踢”的动作，否则进球无效。

（撞球不能算做“踢”的动作）

5.进入比赛场地的机器人的所有状态必须与检录时完全一致，否则不允许其参加比赛。

6.守门员机器人只做单向或左右移动。守门员在比赛开始前，可以手动放置在球门处。（守门员机器人可有主办单位统一提供，赛事开始时应是完整的、具备功能的模型）

（五）参赛队伍规则

1.每支参赛队由 1-2 名学生队员和 1 名教练组成。

2.在通常情况下，不允许参赛队员任意用手移动机器人。每场比赛前，参赛队应指派一名队员担当队长，在规则允许范围内或在裁判员的指示下队员可以拿走、移动、重新放置机器人。

3.参赛队员应熟知比赛的有关规定，所有活动及行为必须遵循规则，服从赛场工作人员的指示。

4.以四个参赛队为一组，分组循环赛。每组对战三场，一般共赛 5 轮,5 轮结束之后以累计进球数多的一方获胜。胜方积 3 分，负方积 0 分；如果平局，双方各积 1 分。

5.按分组循环赛的积分确定每组出线的两支参赛队，如果出现积分并列，按累计进球数确定先后，如果仍然并列，再按累计净胜球数确定先后。

6.复赛中如果出现平局，该场比赛将加时。每队各加时 1 次点球机会。若加时 2 次后双方仍是平局，则机器人总重量轻的队获胜。

7.允许在限定时间内对不合格的机器人加以调整，调整时间不得超过 10 分钟且不能影响正式比赛的安排。如果修改后的机器人仍不符合要求，将取消比赛资格。

（六）点球规则

1.裁判员鸣笛前，参赛队有 1 分钟的调整时间。

2.整个足球完全越过球门线，自由滚入球门且碰撞球门后壁即判为进球，裁判将鸣裁判将鸣笛示意。判定进球后，裁判应鸣笛示意，进球方即得到一分。

3.比赛结束：裁判员吹响终场哨音后，参赛队员除应立即关断机器人的电源外，不得与场上的机器人或任何物品接触。

4.裁判员填写记分表。参赛队员应确认自己的得分并立即将自己的机器人搬回准备区。

（七）犯规和取消比赛资格

任何不尊重裁判、不服从裁决的行为，将给予黄牌警告，若纠缠不止，则给予红牌并取消其比赛资格。

任何严重违背公平竞争精神的行为（例如，故意干扰并再三损坏其它机器人、损坏比赛场地或足球，采用不符合规定的机器人等）将被取消比赛资格。

（八）程序参考

踢球机器人图左、裁判图右

奇颖开始

设置 my variable 的值为 红外接收传感器 P1

循环执行

如果 变量 my variable = 红外遥控键值 9 那么执行

迷你电机 PA 正转 功率 100 %

否则

如果 变量 my variable = 红外遥控键值 8 那么执行

迷你电机 PA 反转 功率 100 %

否则

如果 变量 my variable = 红外遥控键值 7 那么执行

迷你电机 PB 正转 功率 100 %

否则

迷你电机 PA 停止转动

迷你电机 PB 停止转动

奇颖开始

循环执行

迷你电机 PA 正转 功率 100 %

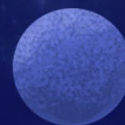
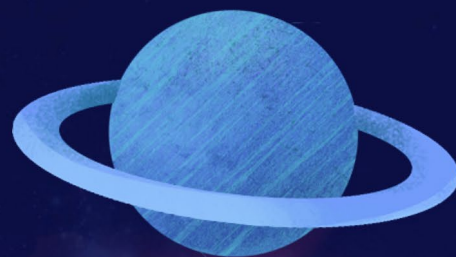
等待 2.5 秒

迷你电机 PB 反转 功率 100 %

等待 2.5 秒

注：足球挑战赛（对抗）及循迹遥控赛等详见比赛手册。

启程 近地轨道卫星回收

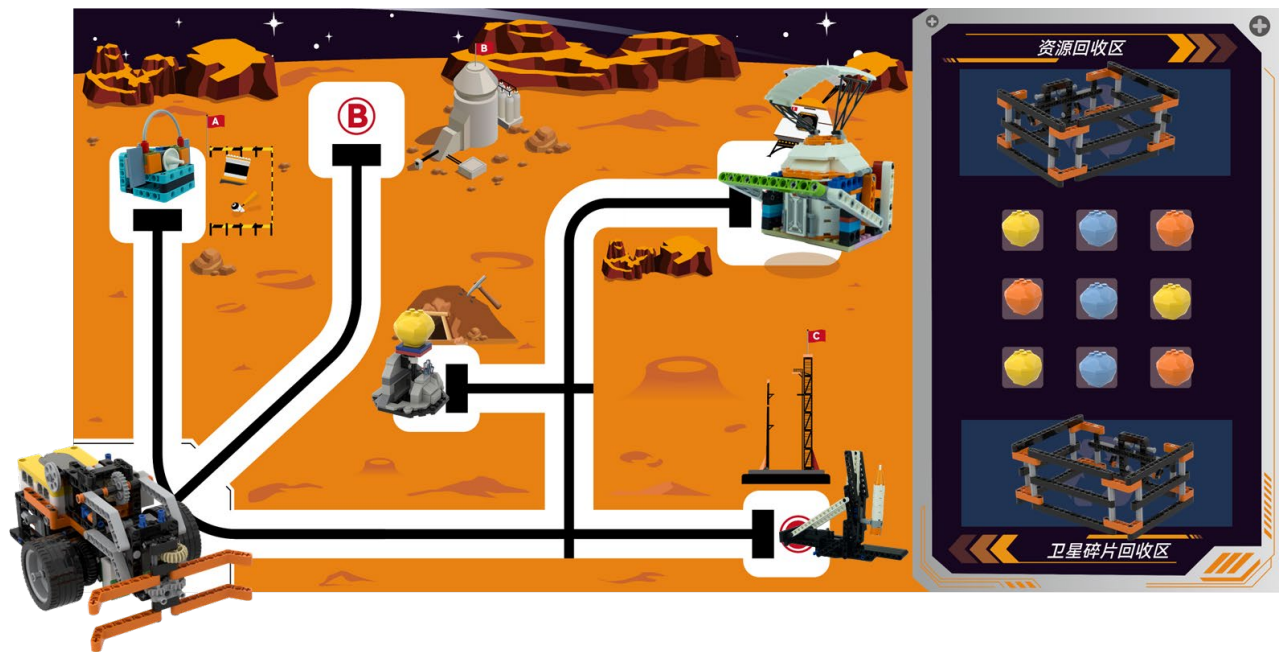


赛事竞技课

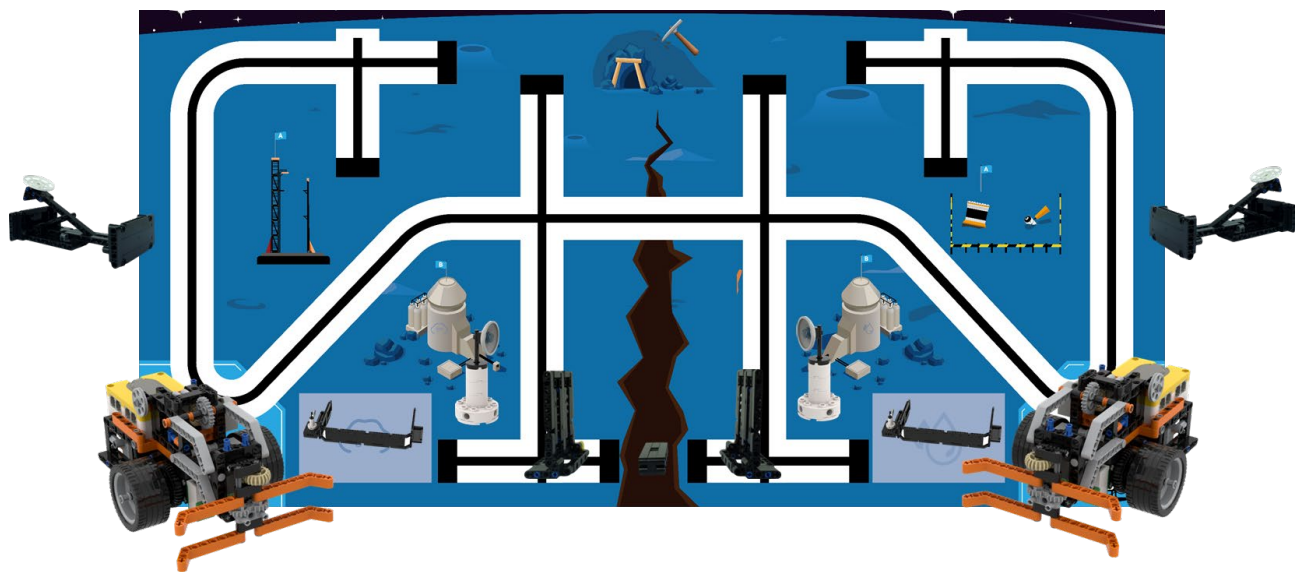


根据《教育部办公厅关于公布2022-2025学年面向中小学生的全国性竞赛活动的通知》（教监管厅函【2022】13号），经教育部批准，中国下一代教育基金会举办“全国青少年科技教育成果展示大赛”。该竞赛被评定为2022-2025年教育部公示的《面向中小学生的全国性竞赛活动名单》（教育部白名单比赛）。

第一课 赛场布置与道具

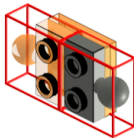
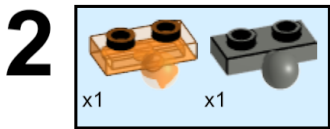
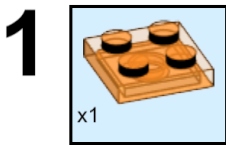


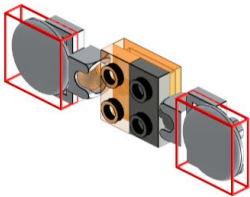
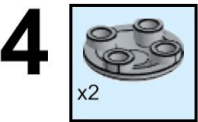
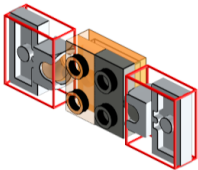
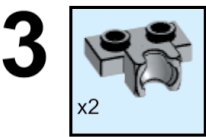
小学组、初中组场地布置

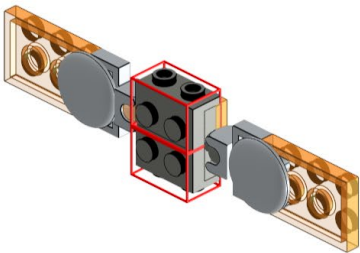
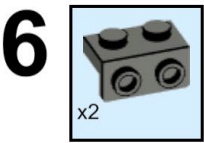
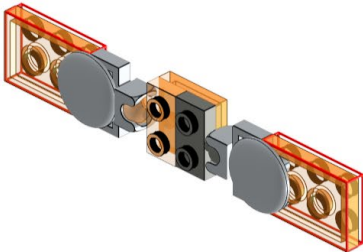
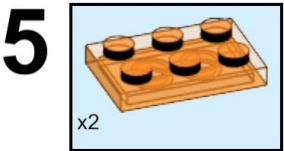


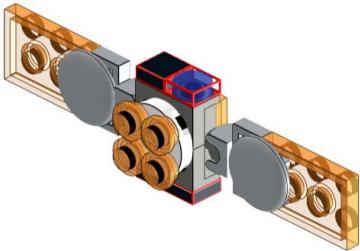
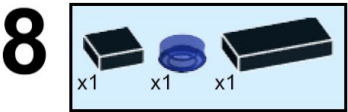
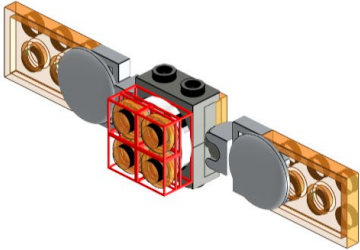
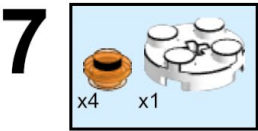
高中组场地布置

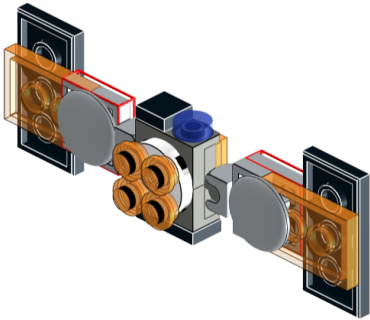
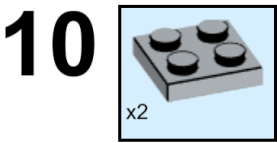
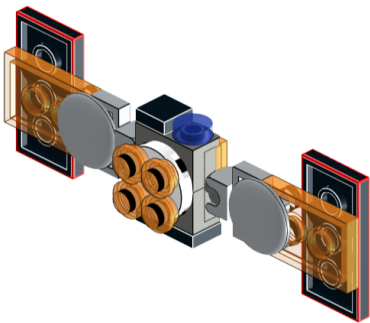
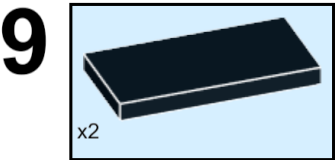
一、小学组卫星搭建



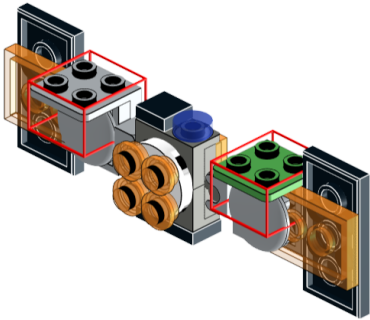
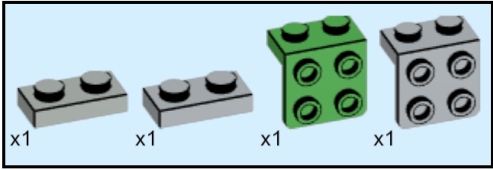




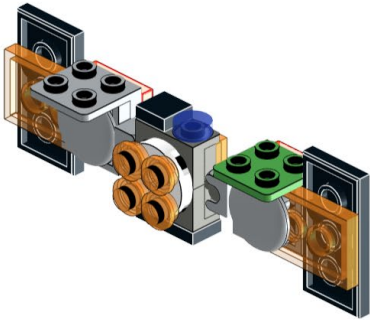
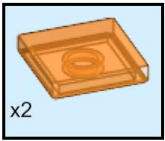




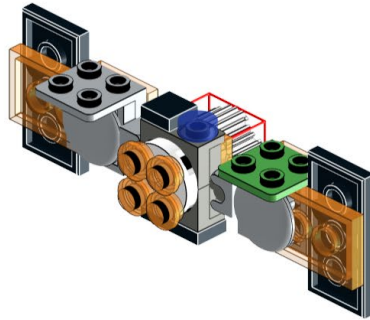
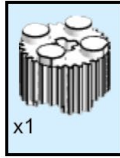
11



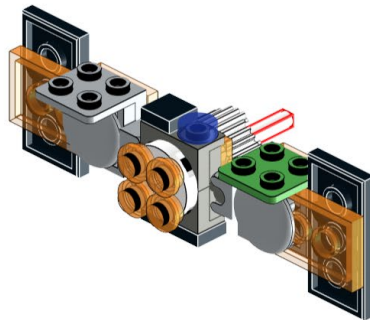
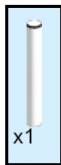
12

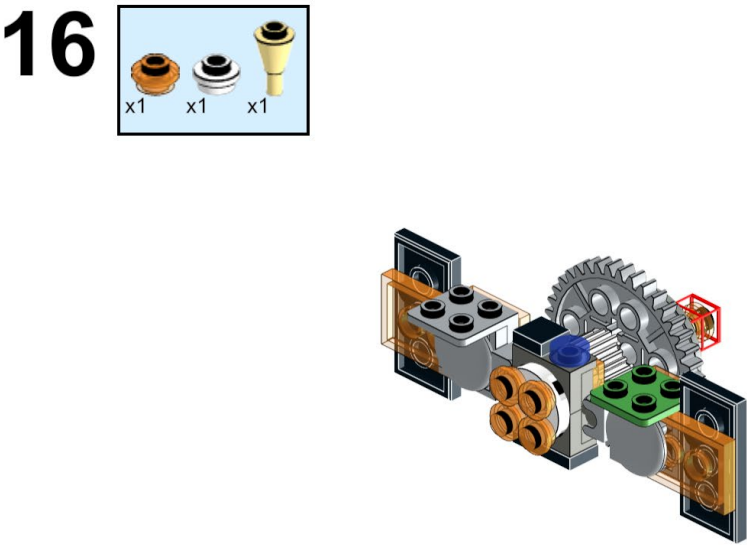
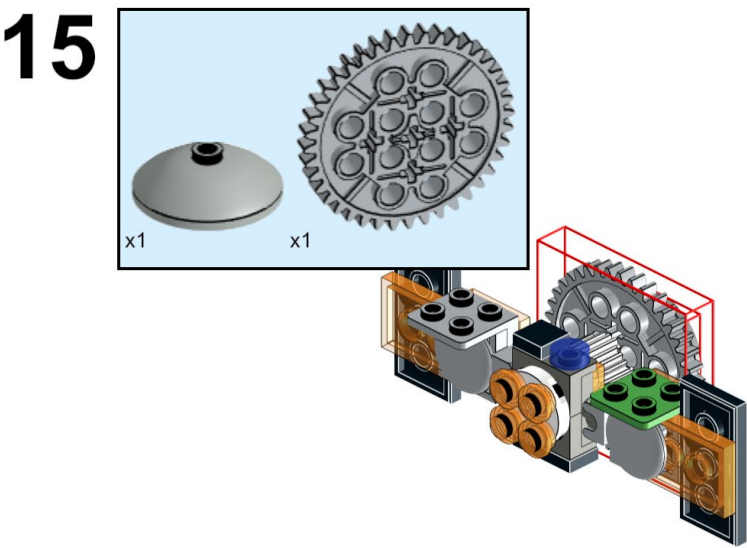


13

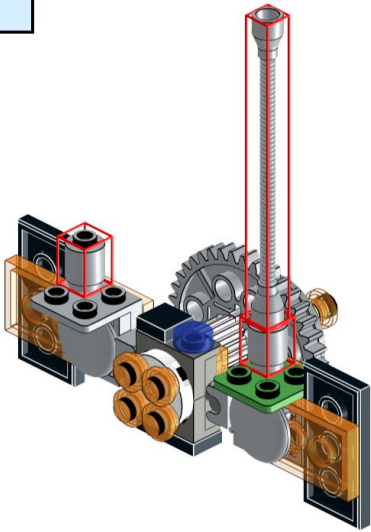
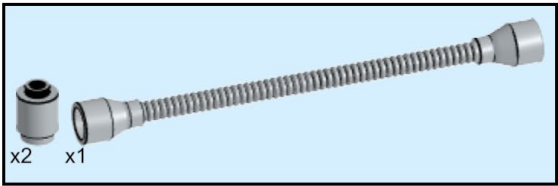


14

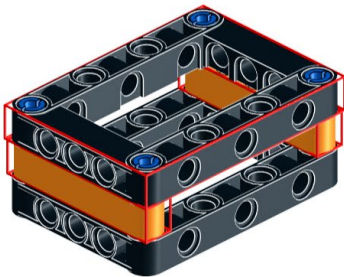
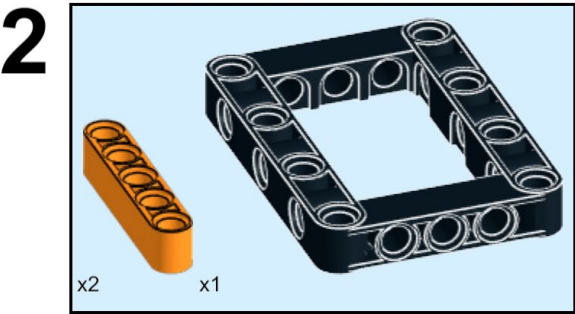
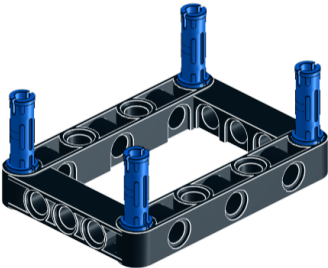
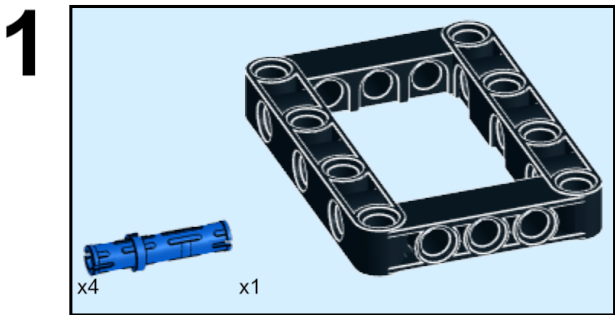




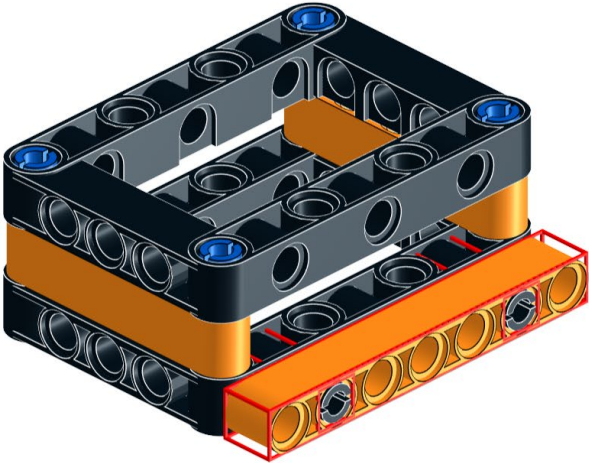
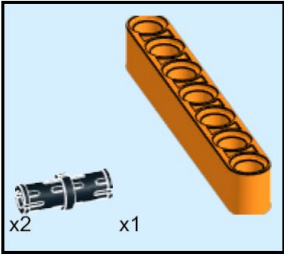
17



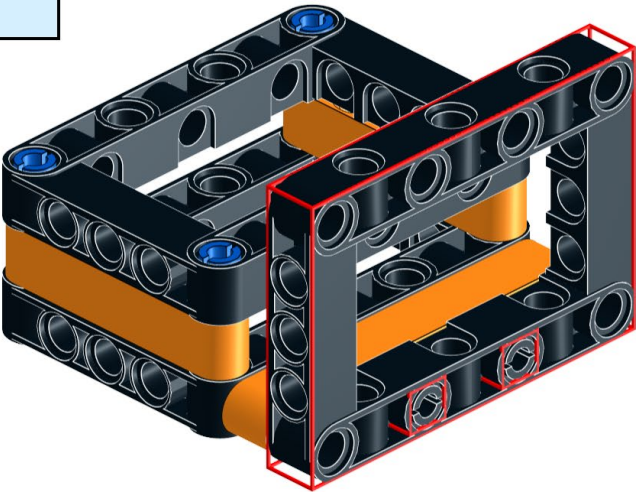
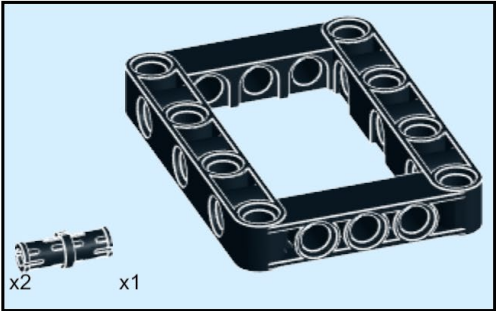
二、小学组卫星支架

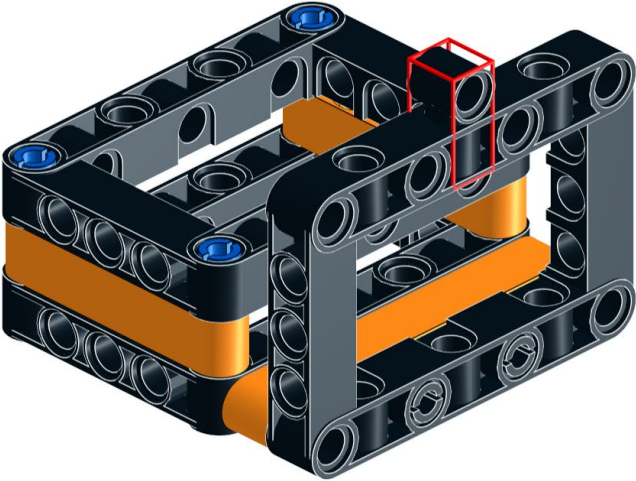


3

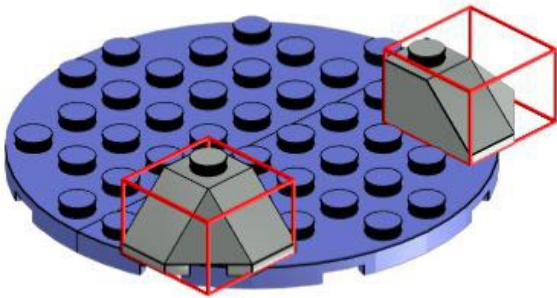
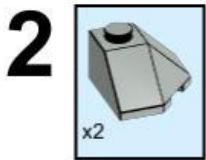
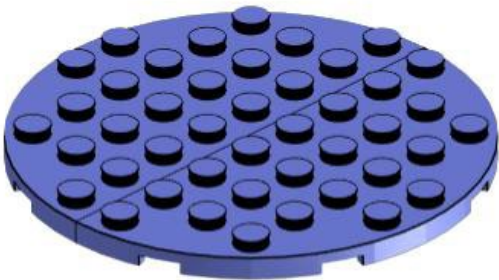
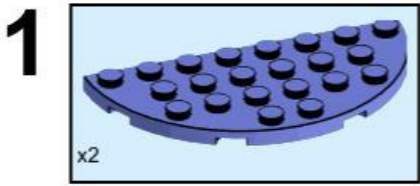


4

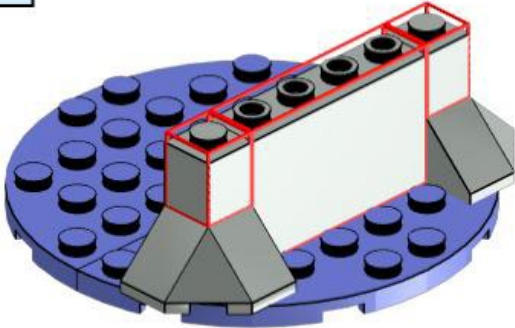
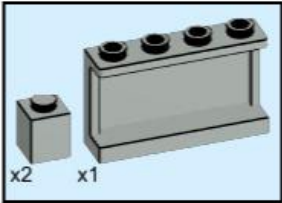




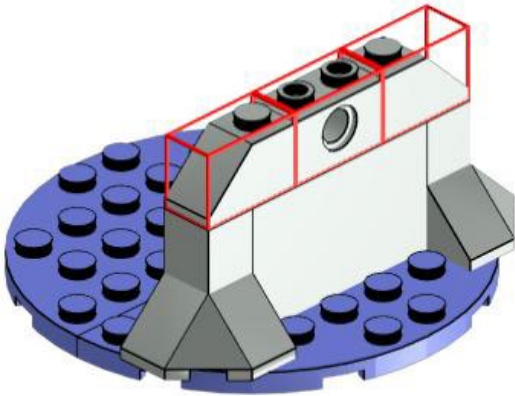
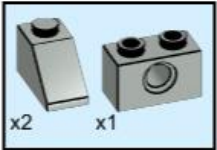
三、小学组矿石支架

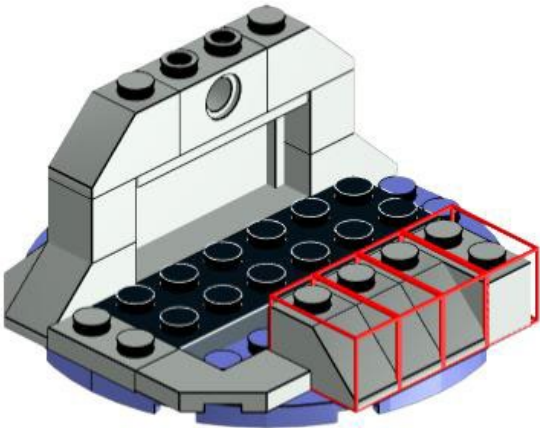
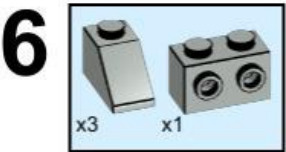
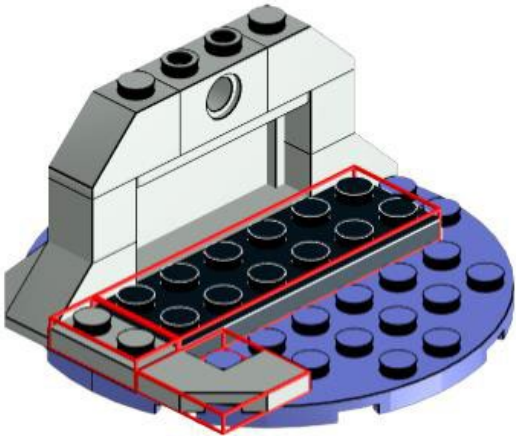
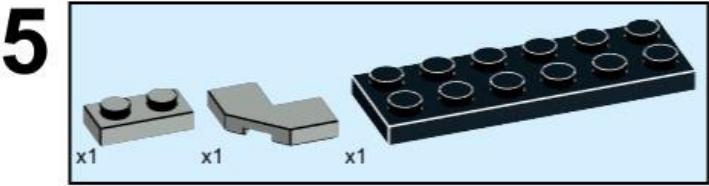


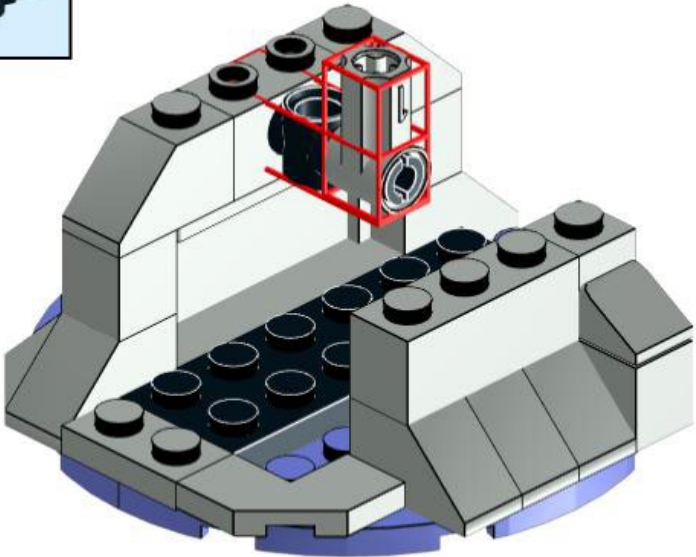
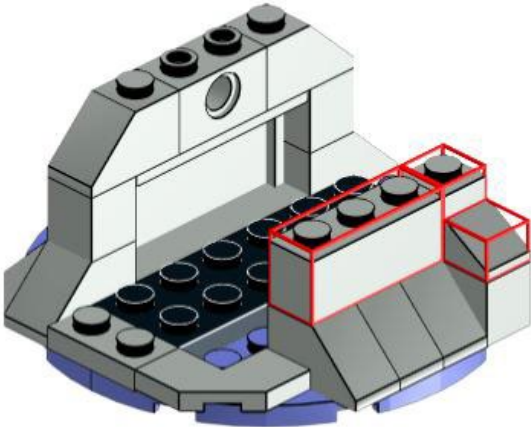
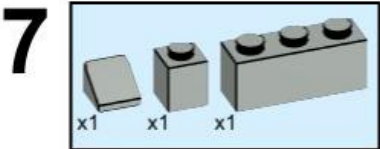
3



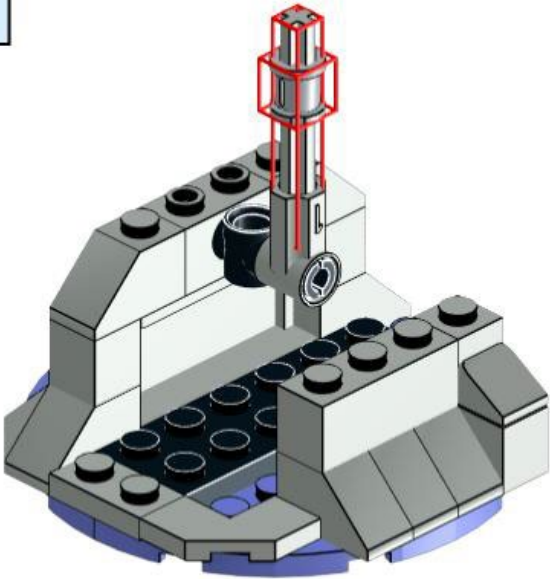
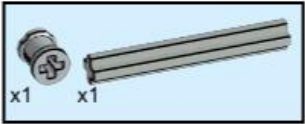
4



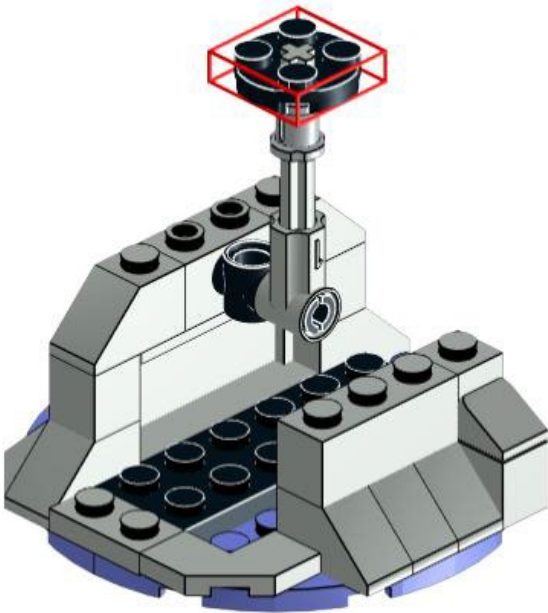




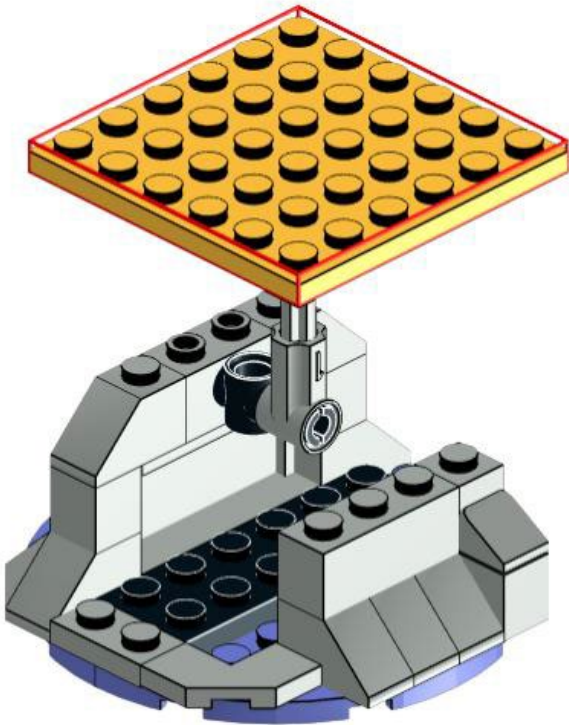
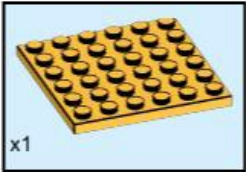
9



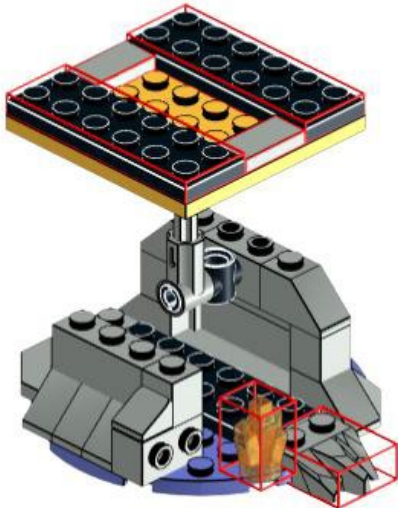
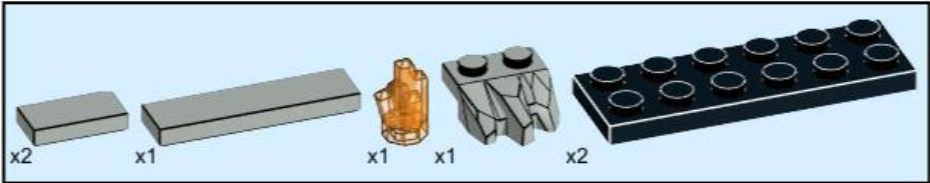
10



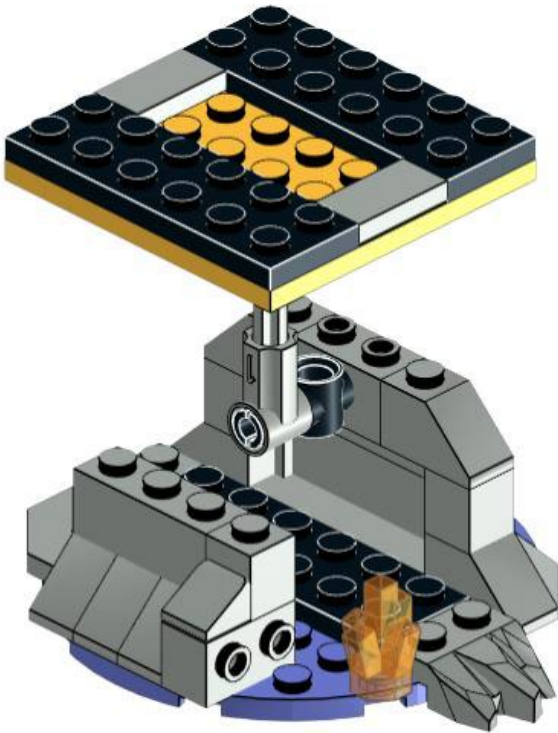
11



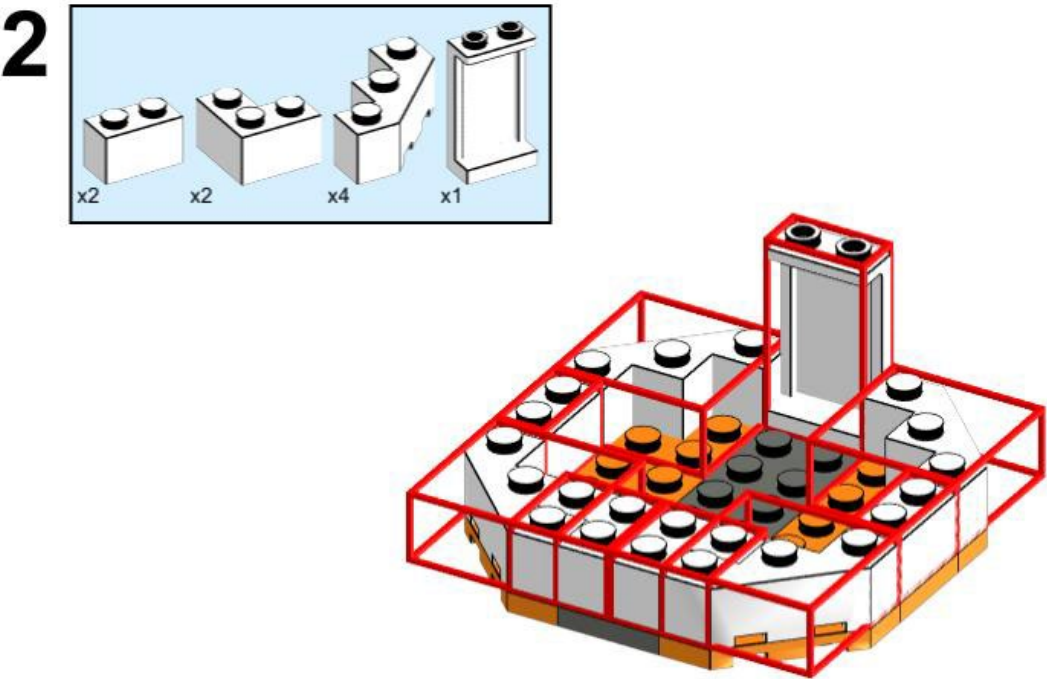
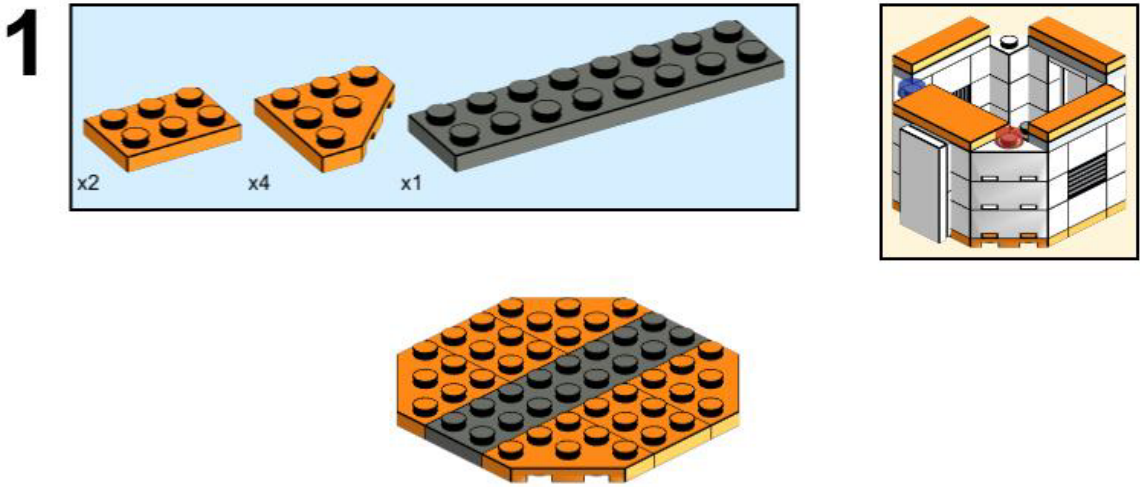
12



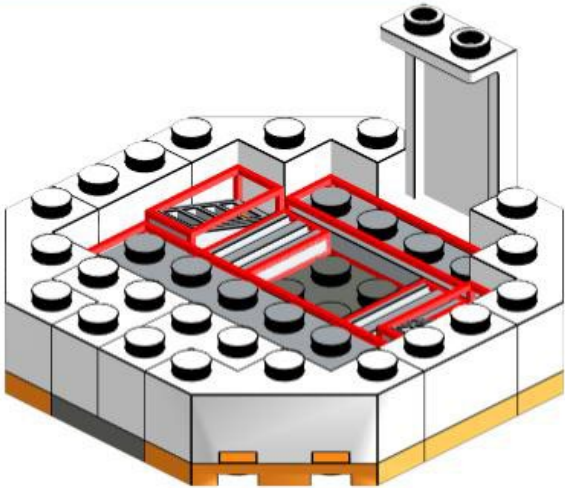
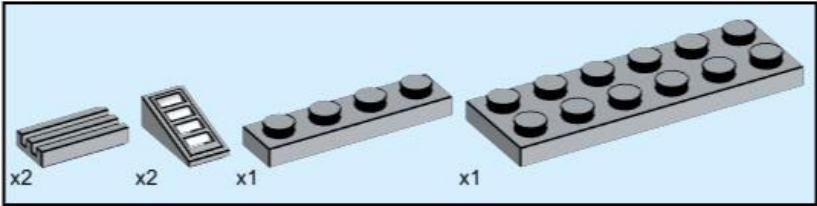
13



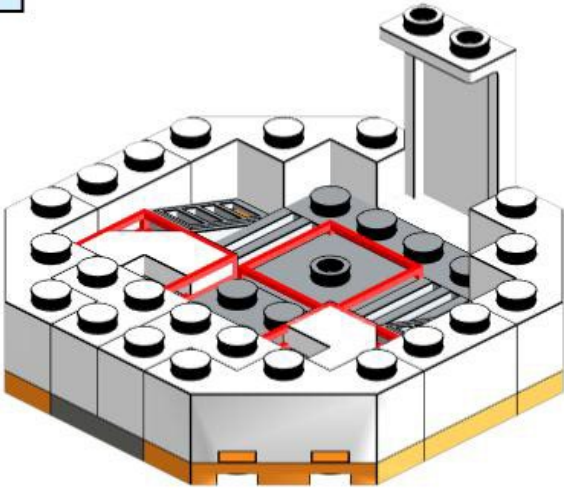
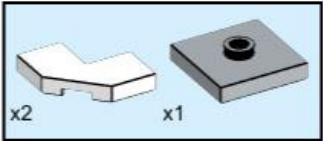
四、小学组回收舱



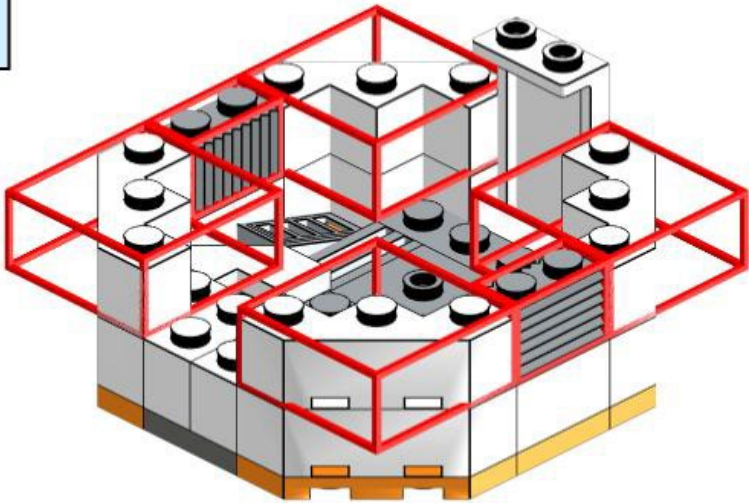
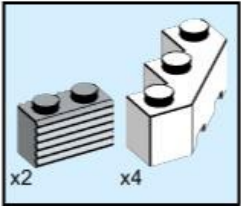
3



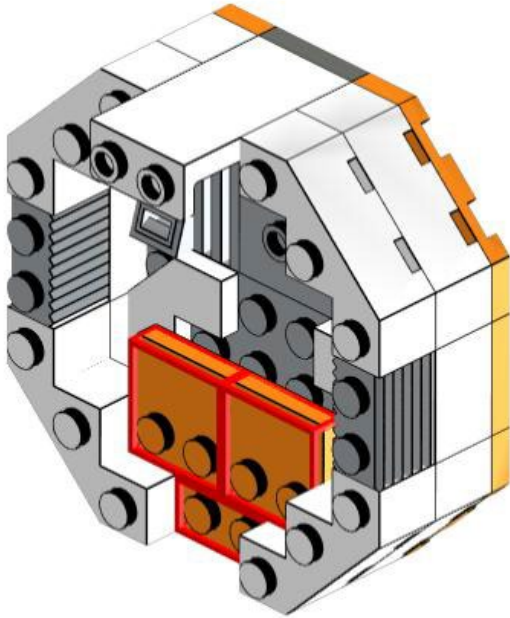
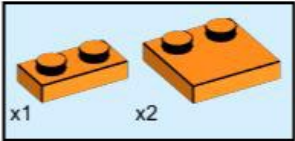
4



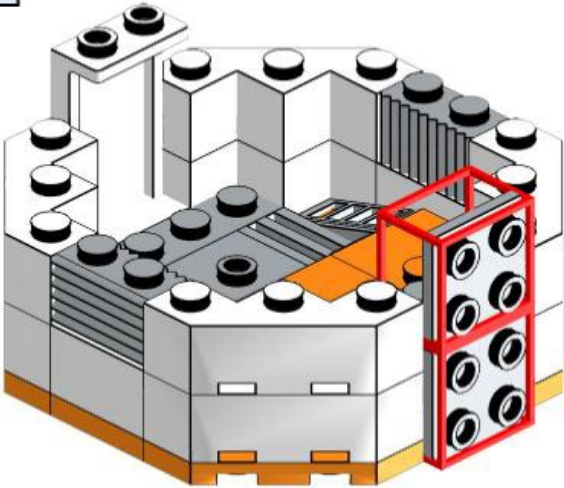
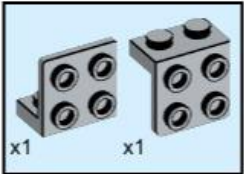
5



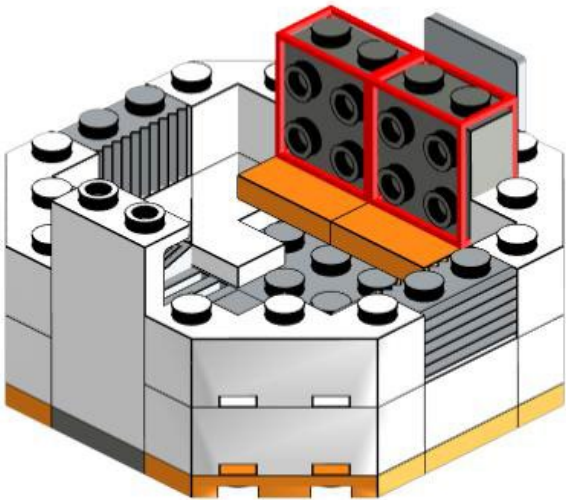
6



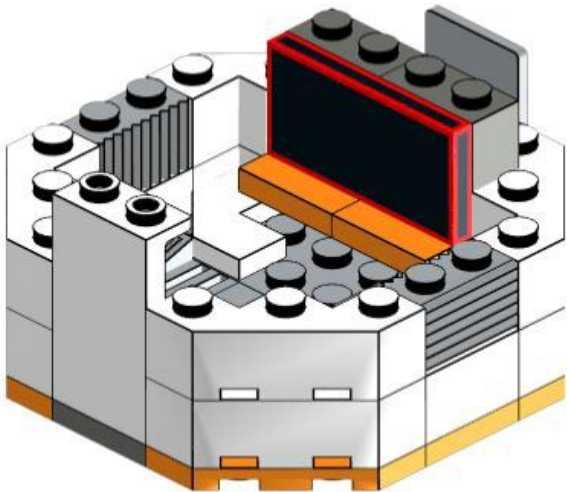
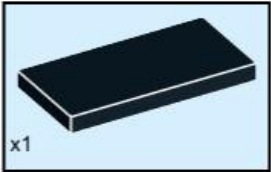
7



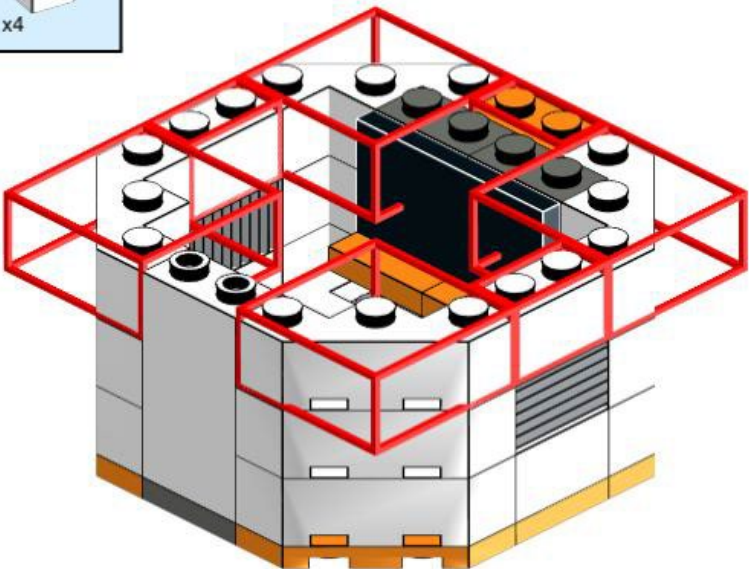
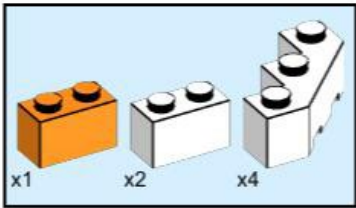
8



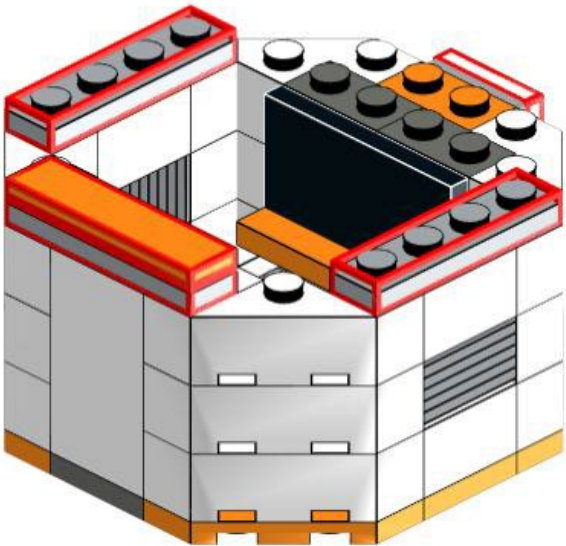
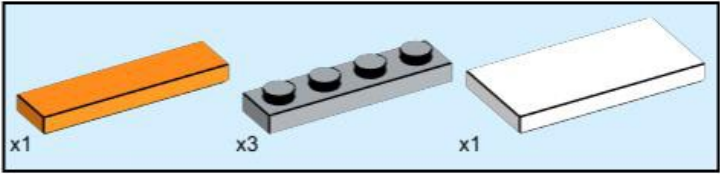
9



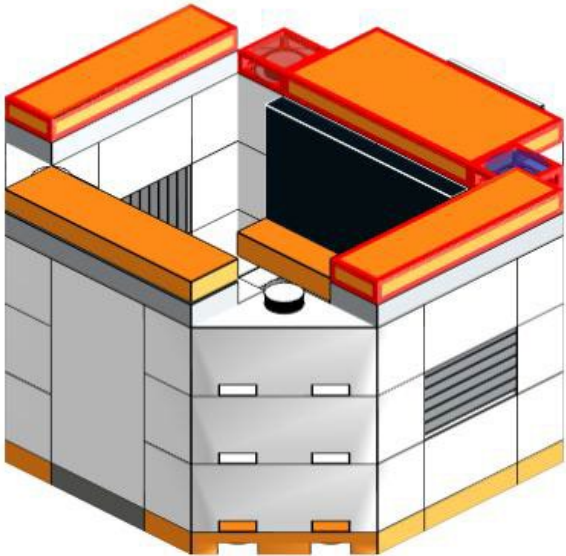
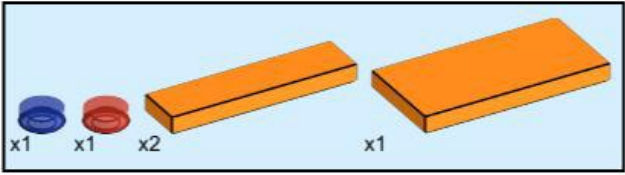
10



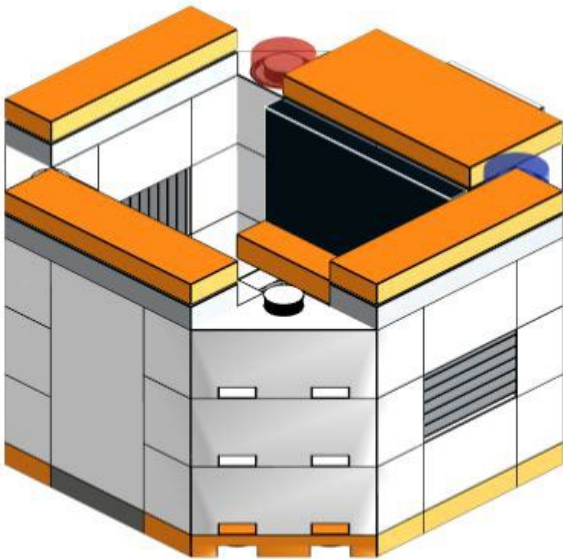
11



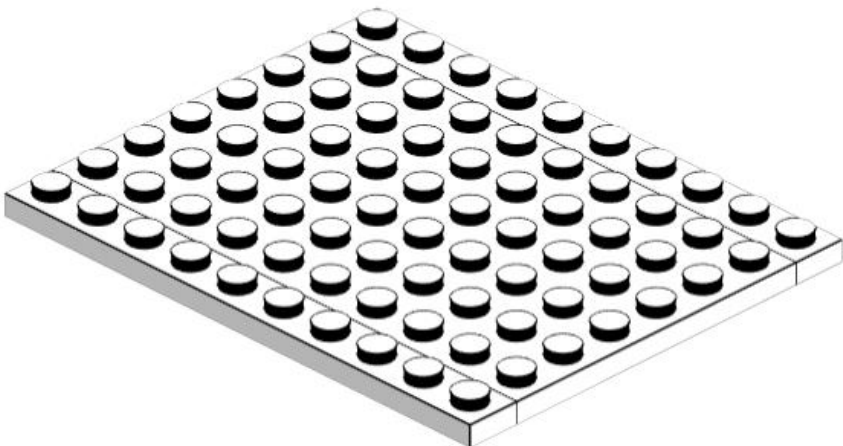
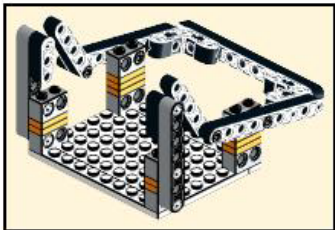
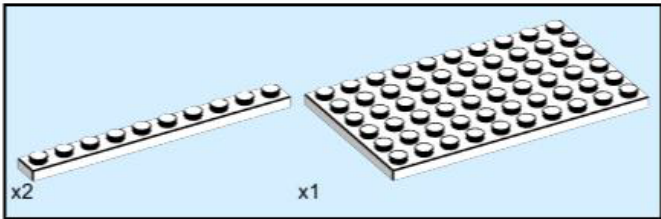
12



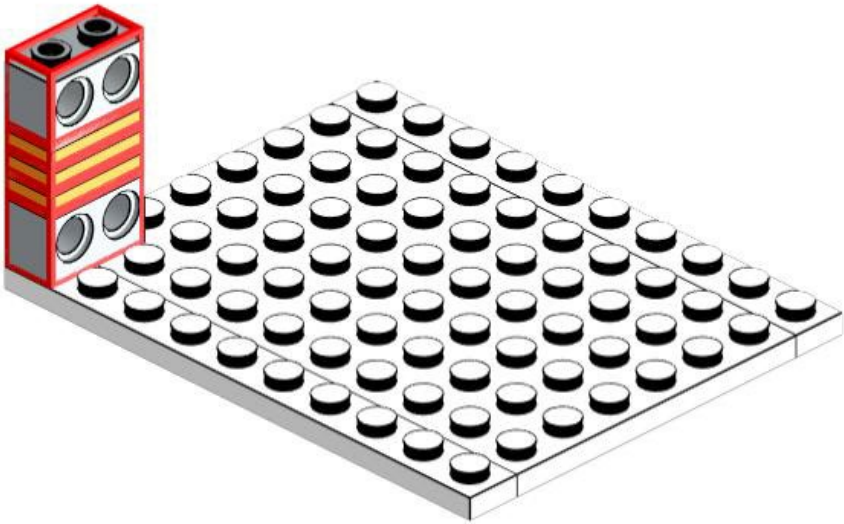
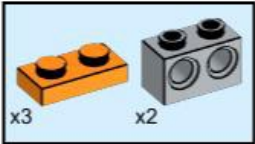
13



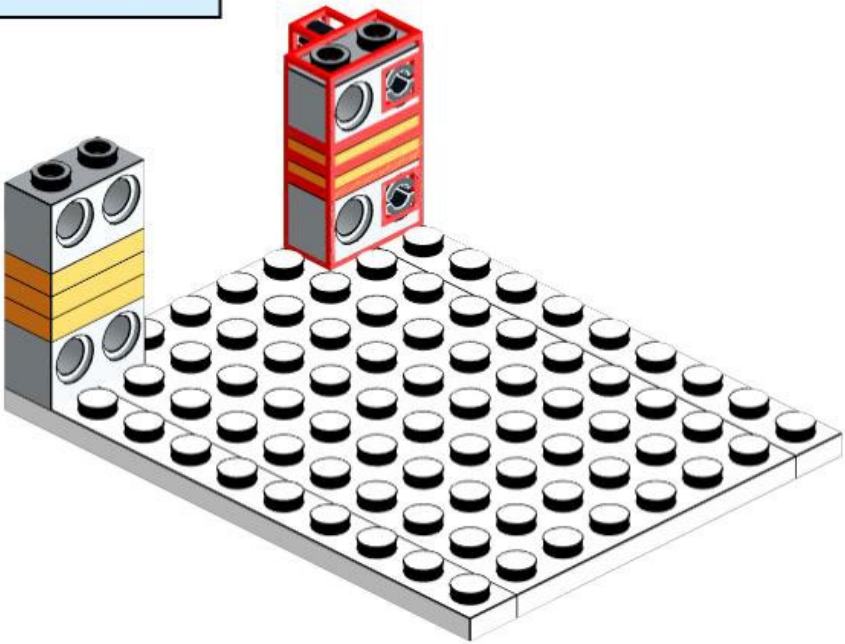
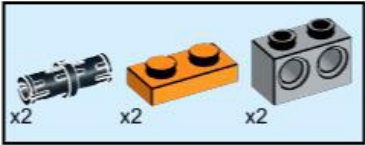
14



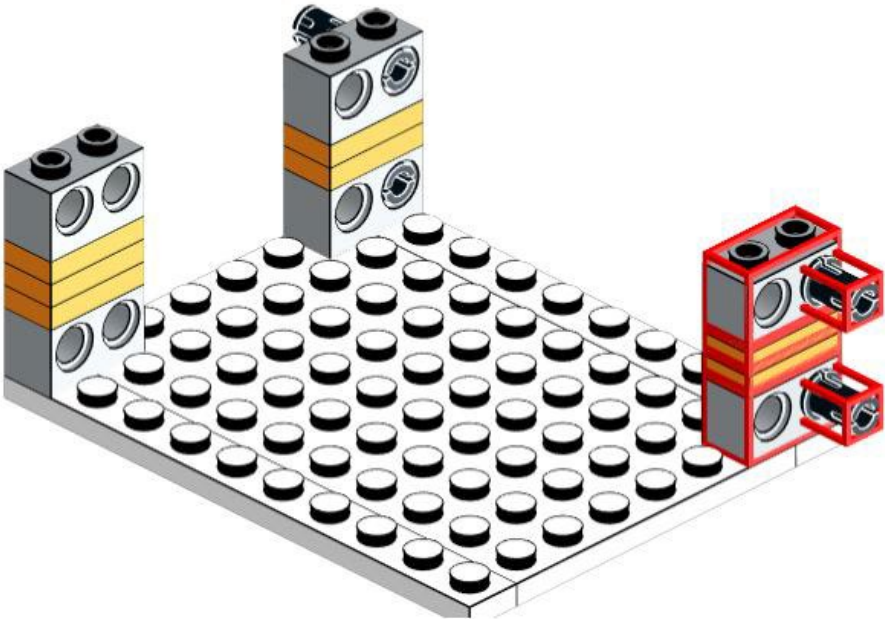
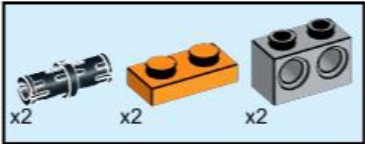
15



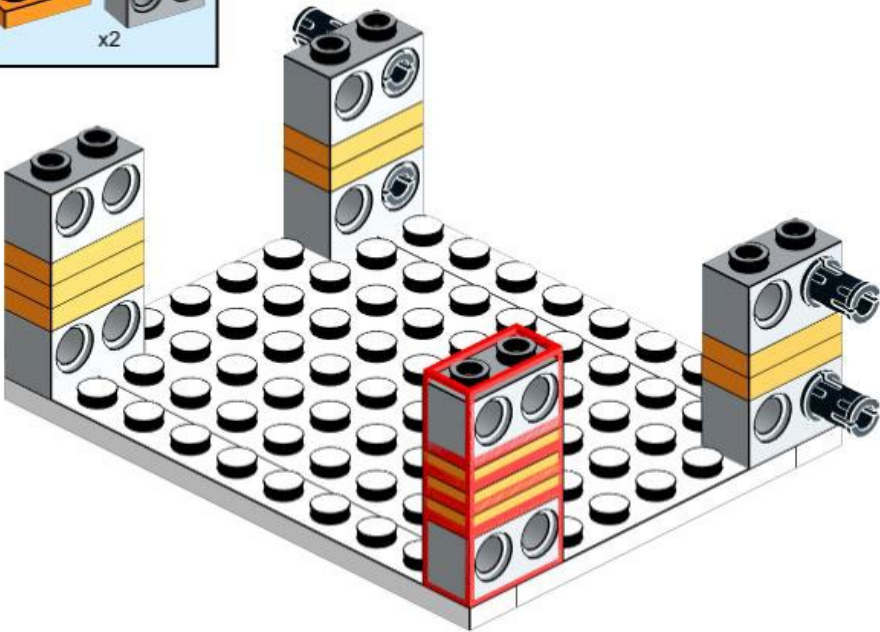
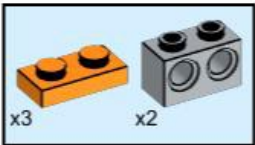
16



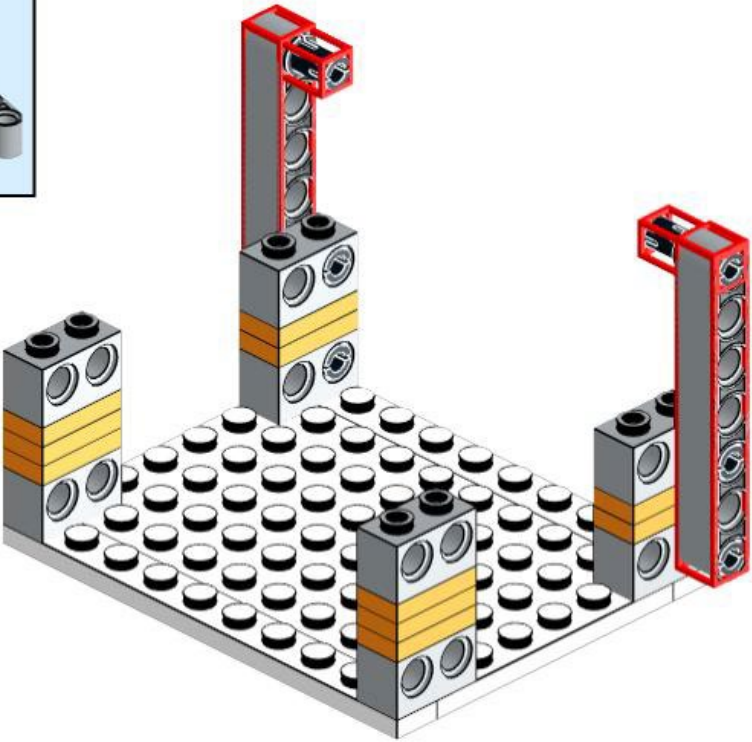
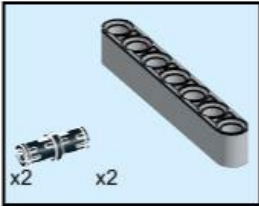
17



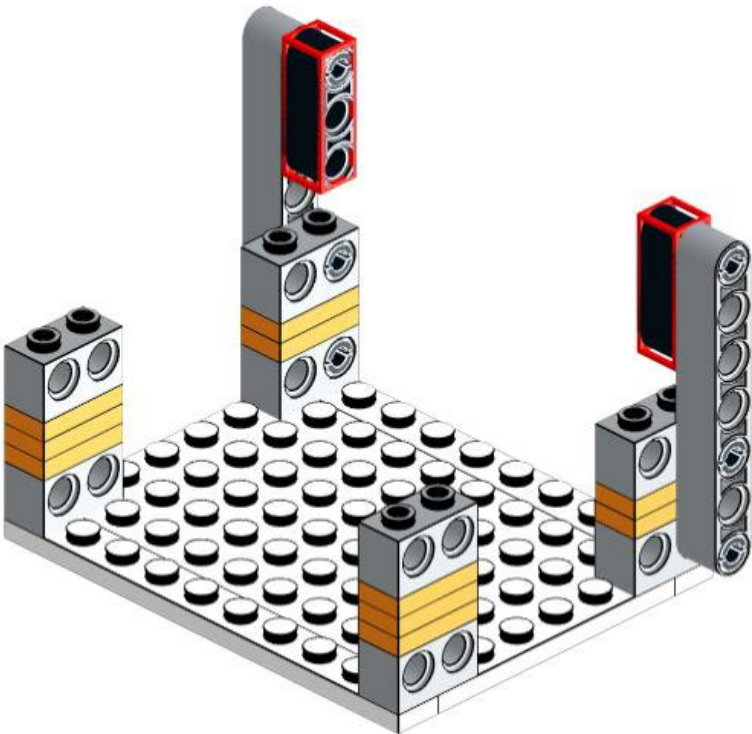
18



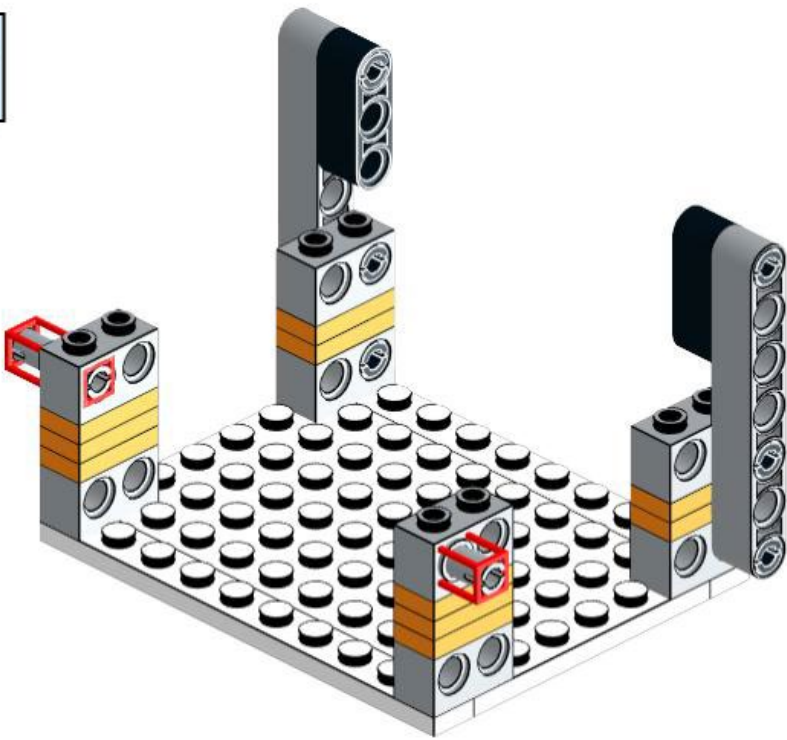
19



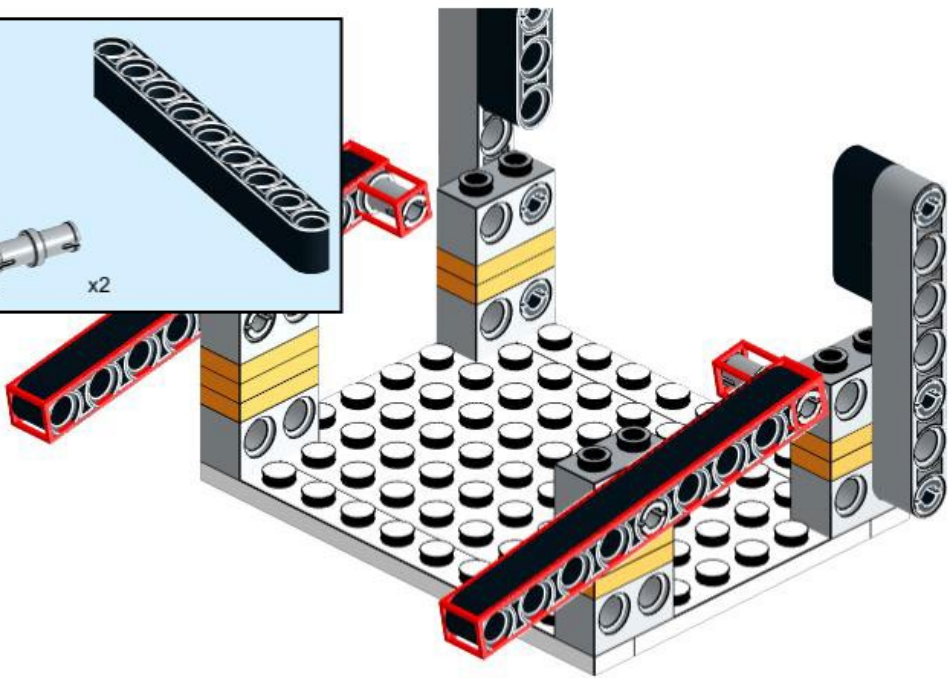
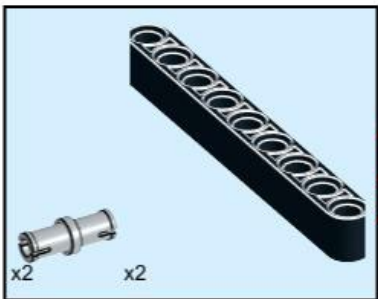
20



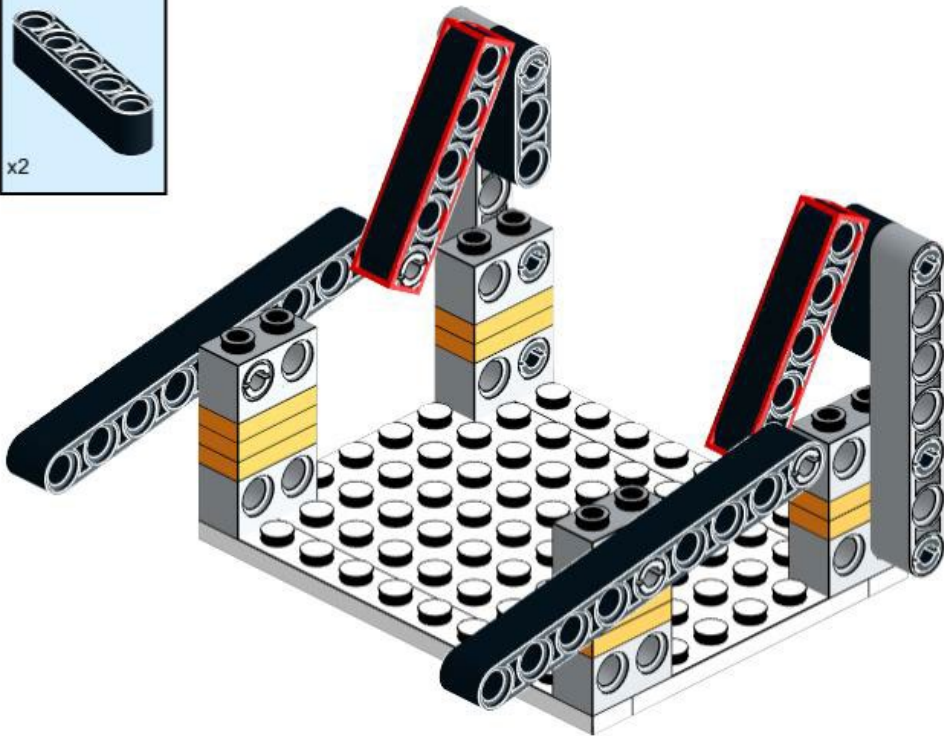
21



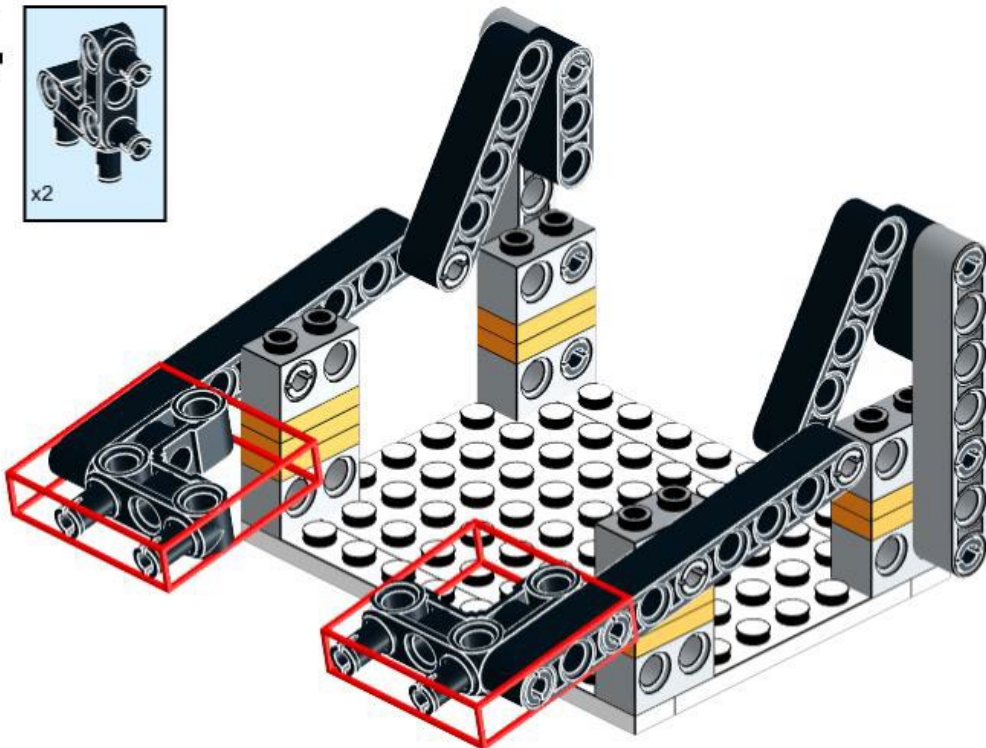
22



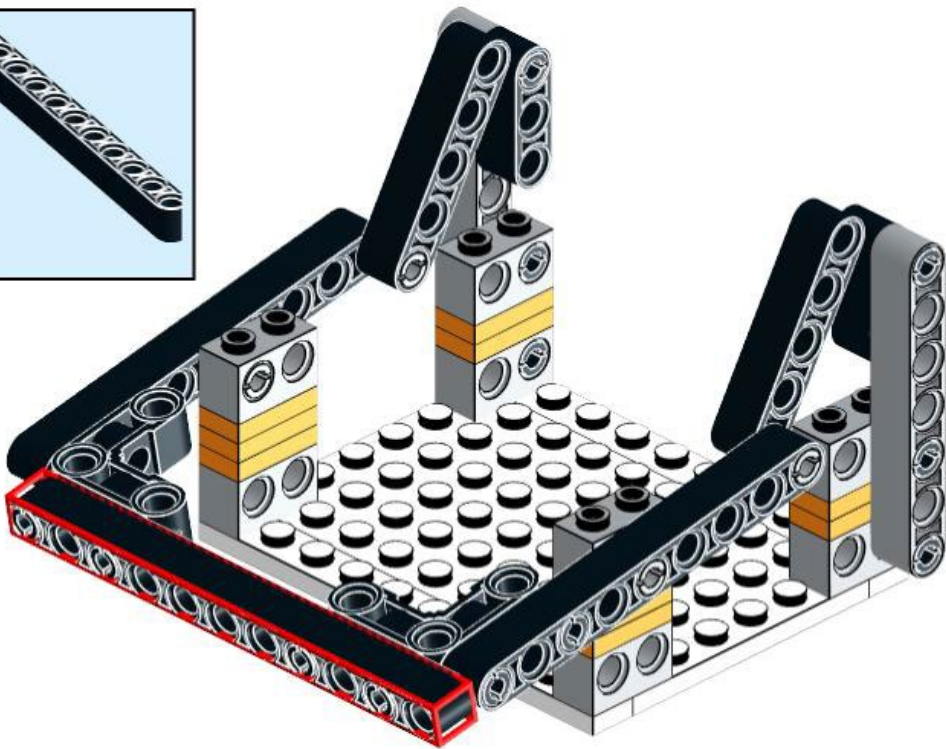
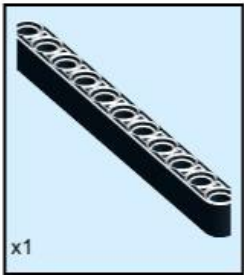
23



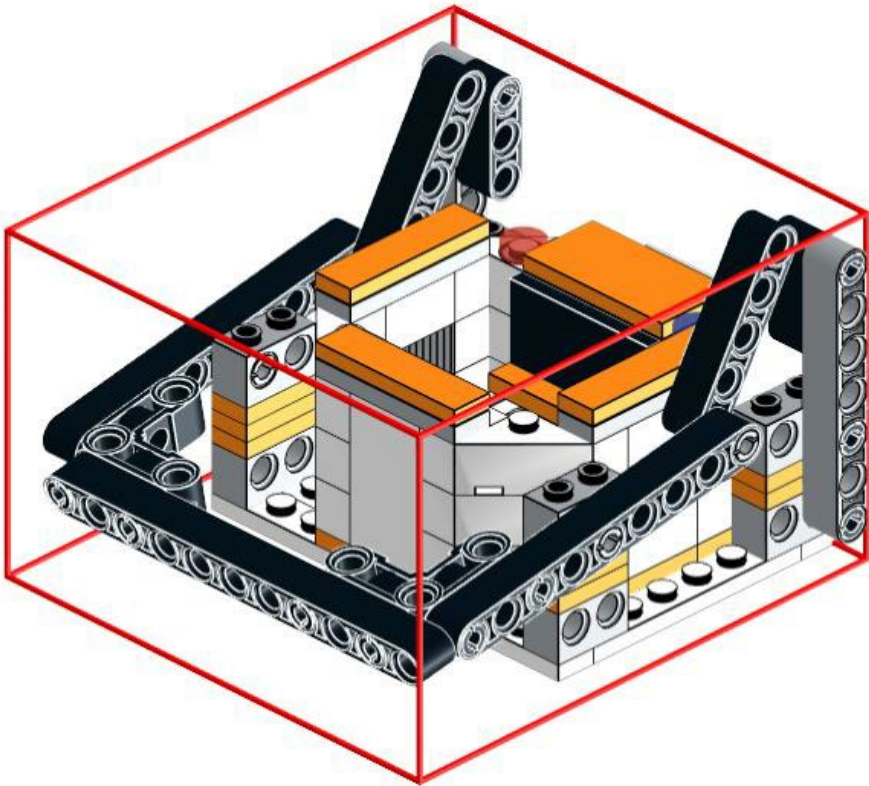
24



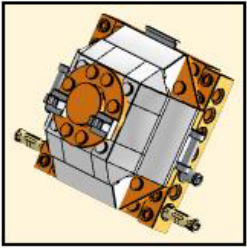
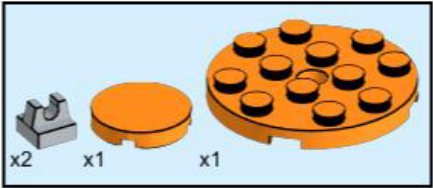
25



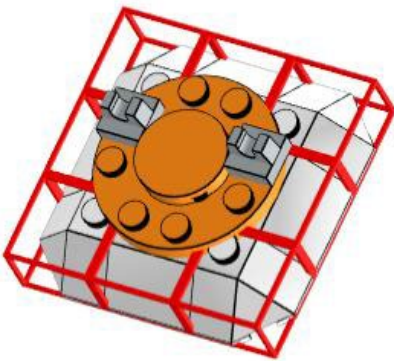
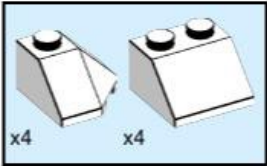
26



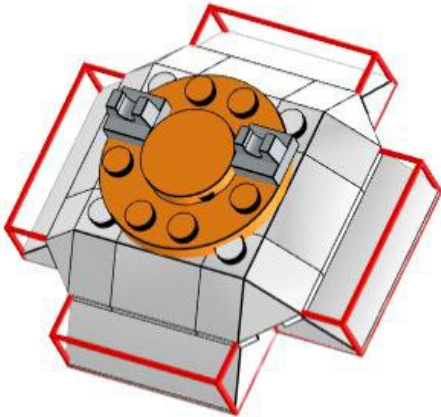
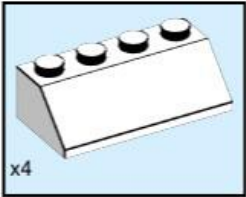
27



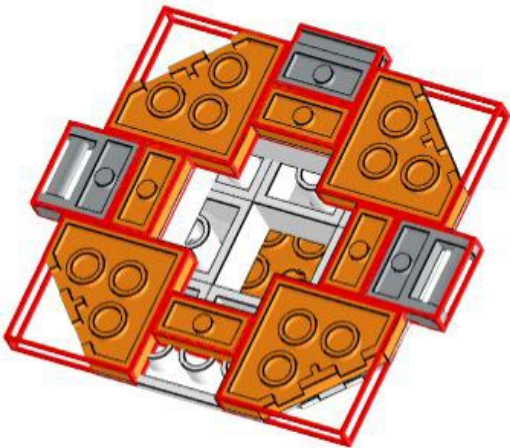
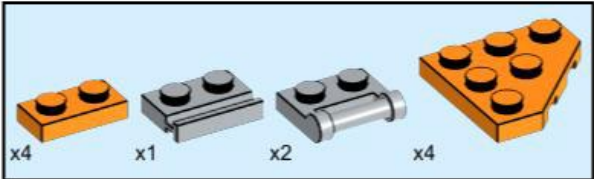
28



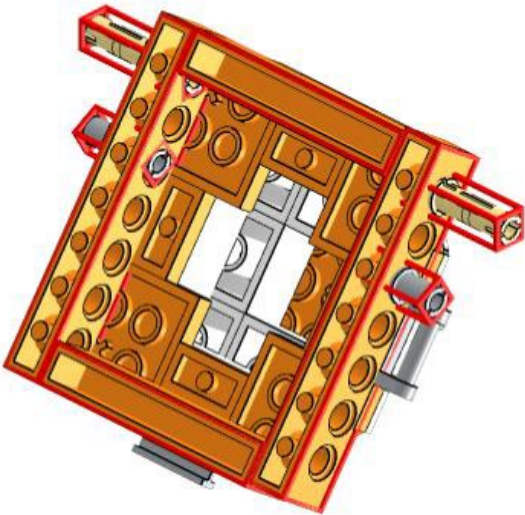
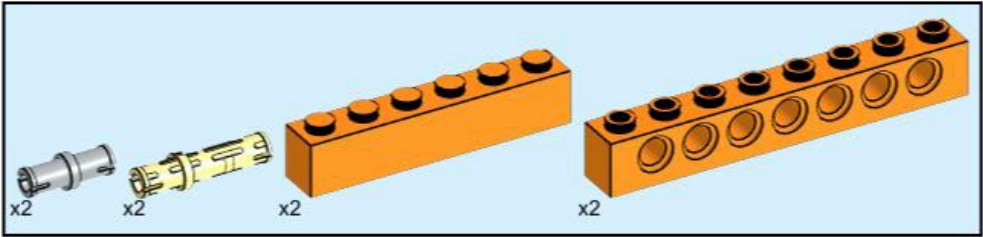
29



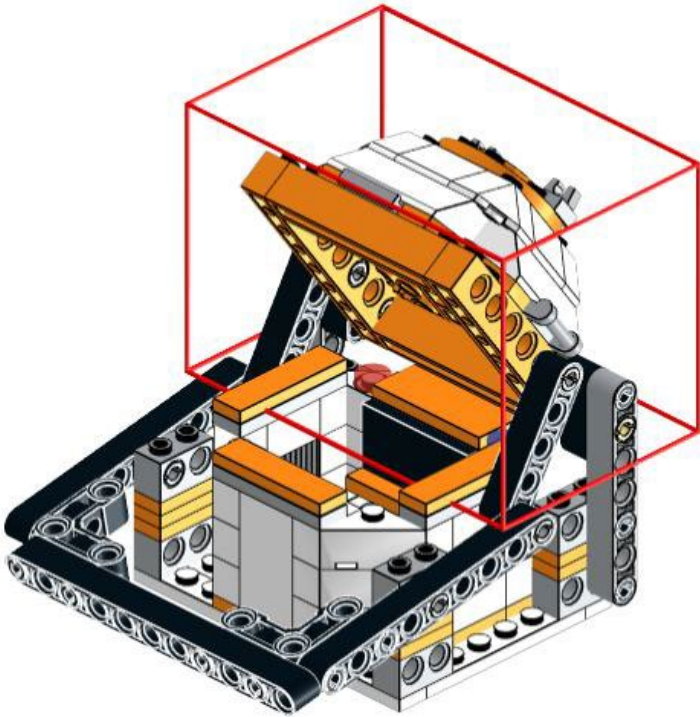
30



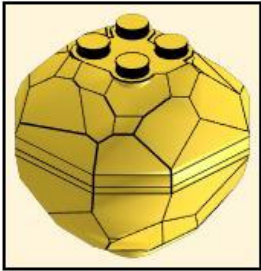
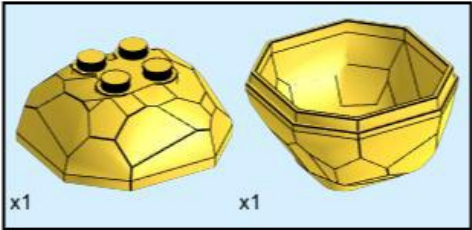
31



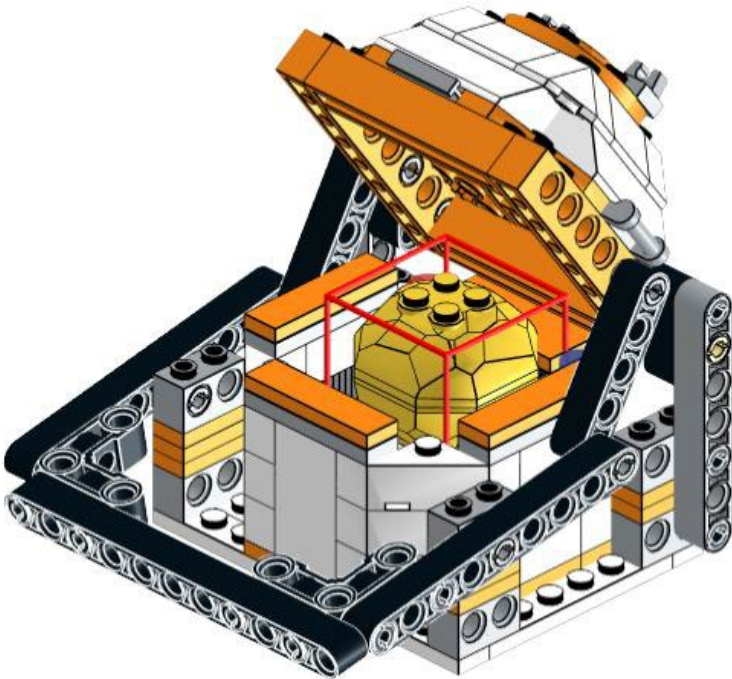
32



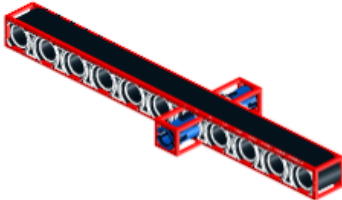
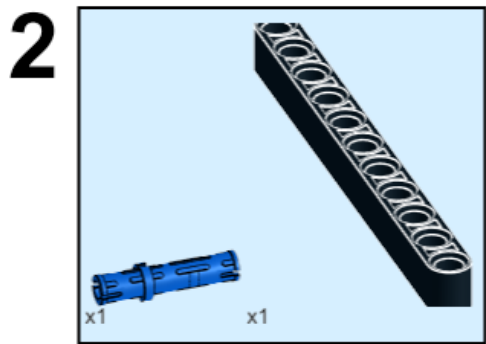
33



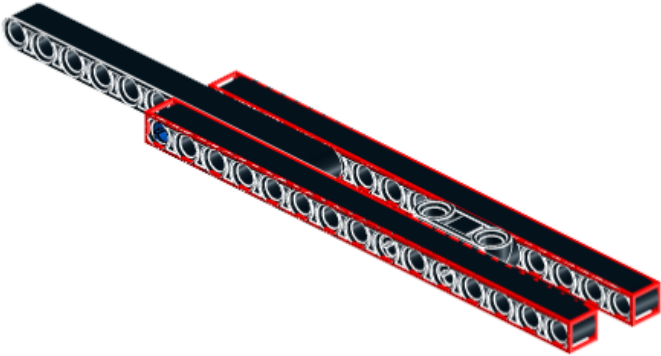
34



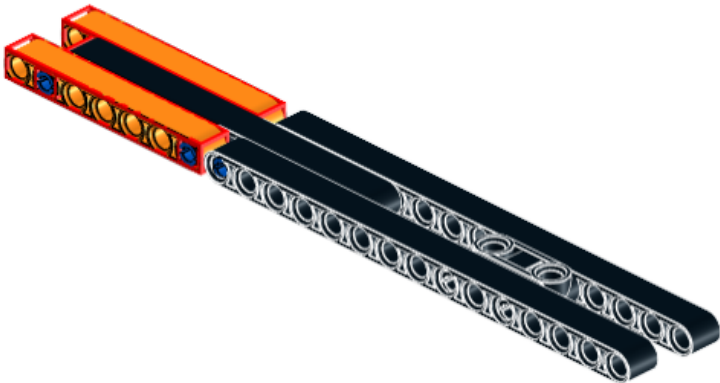
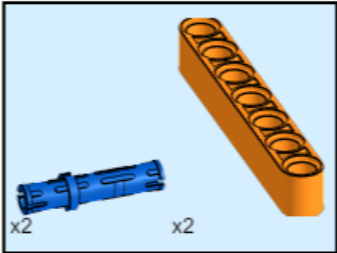
五、小学组火箭发射器

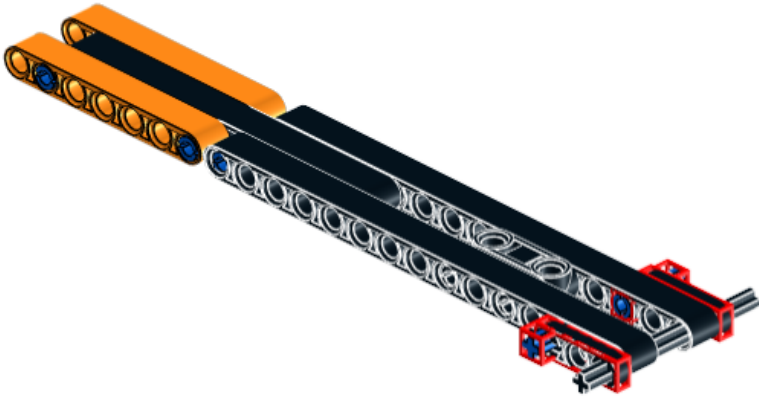
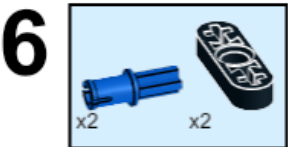
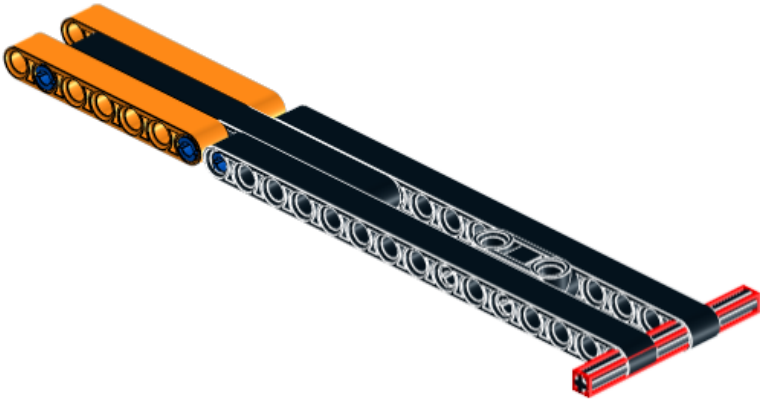
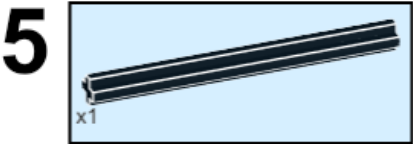


3

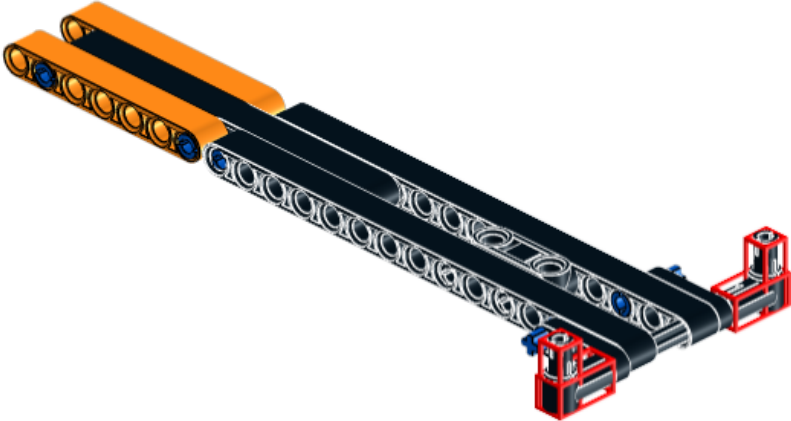
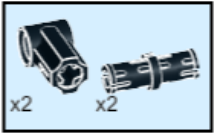


4

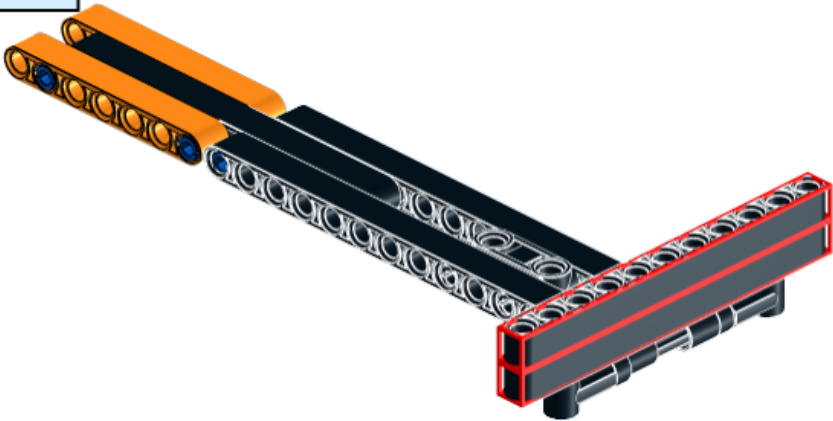


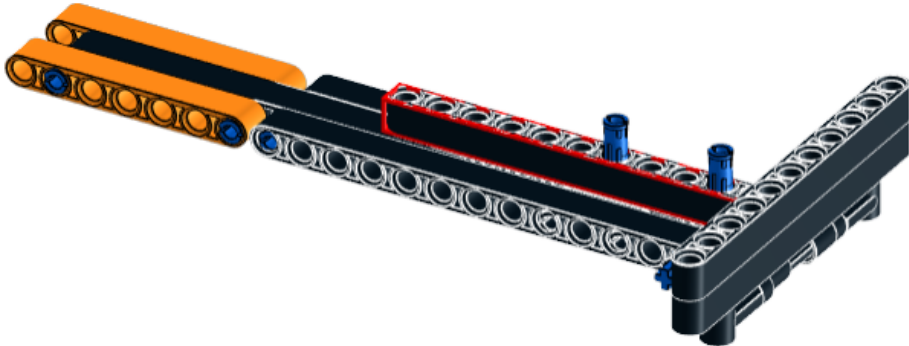
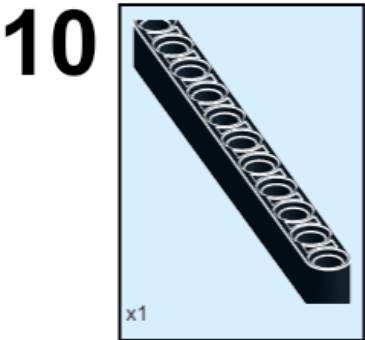
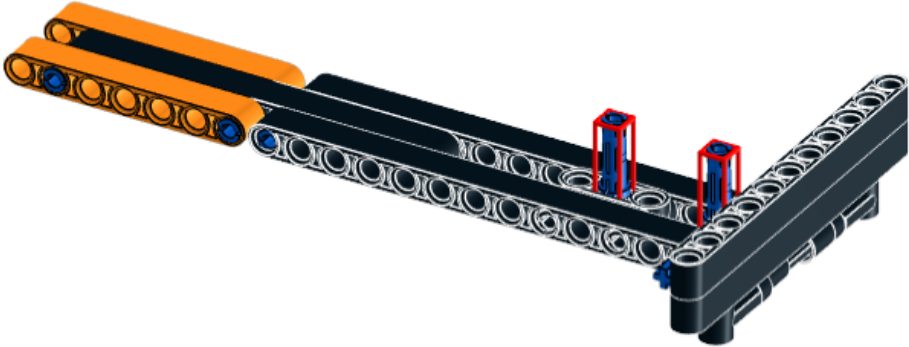
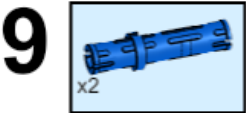


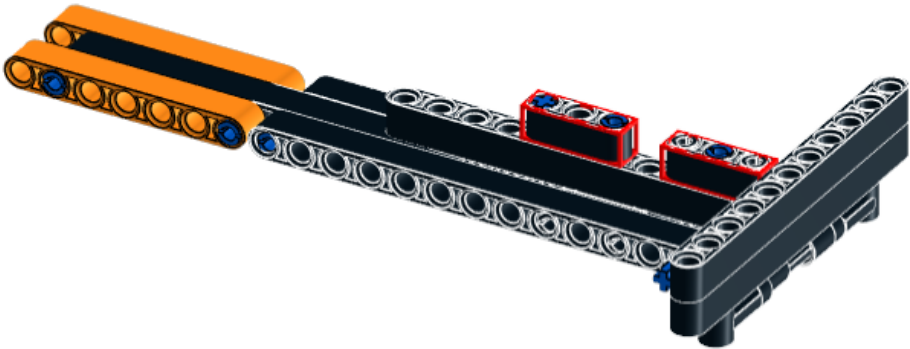
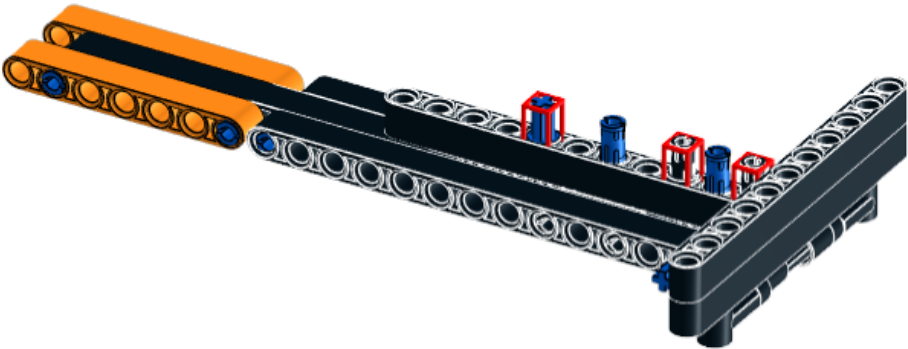
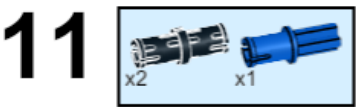
7



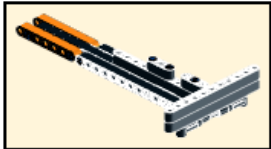
8



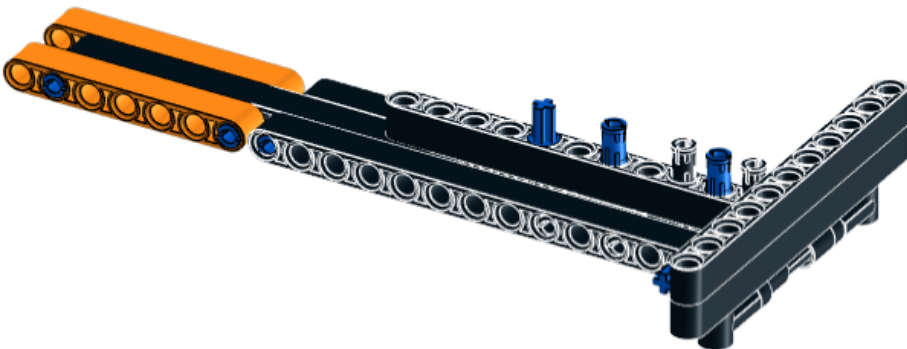




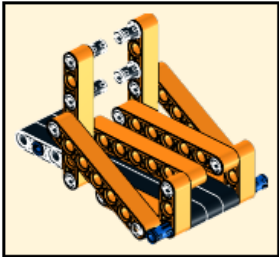
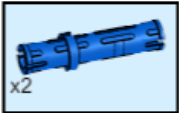
13



14



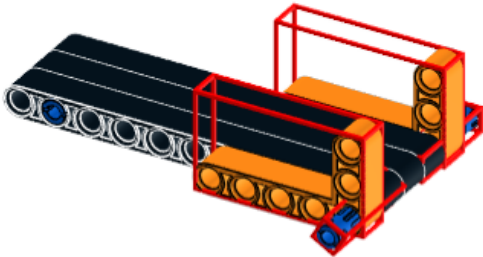
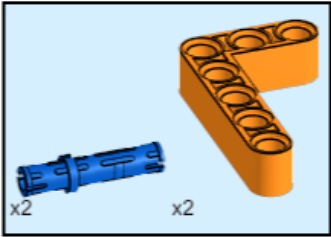
15



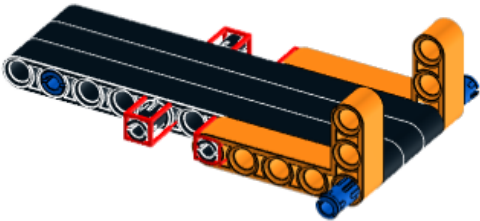
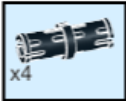
16



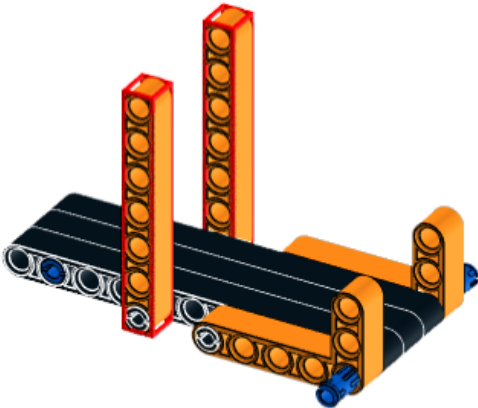
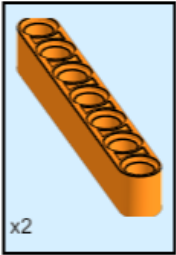
17



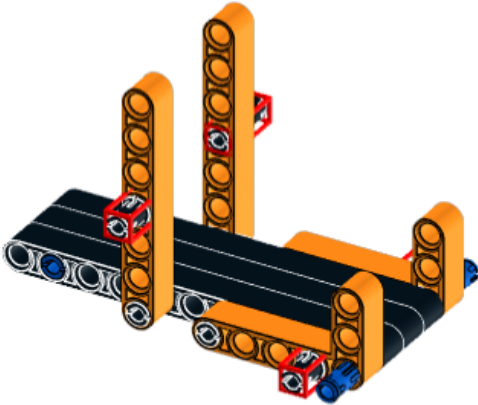
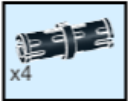
18



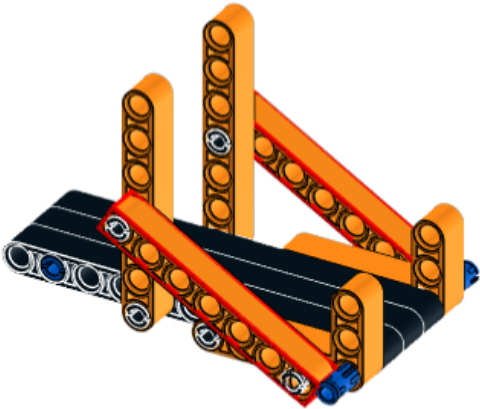
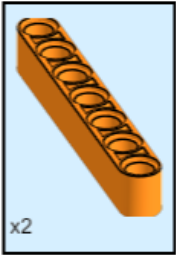
19



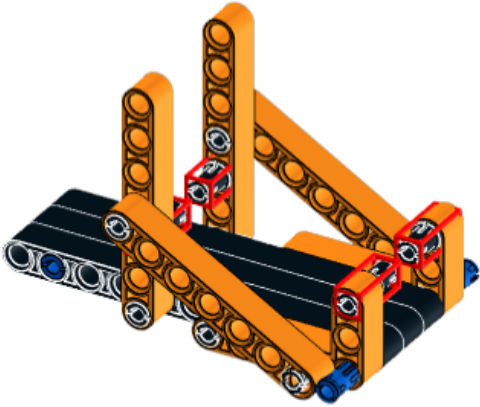
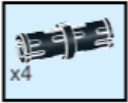
20



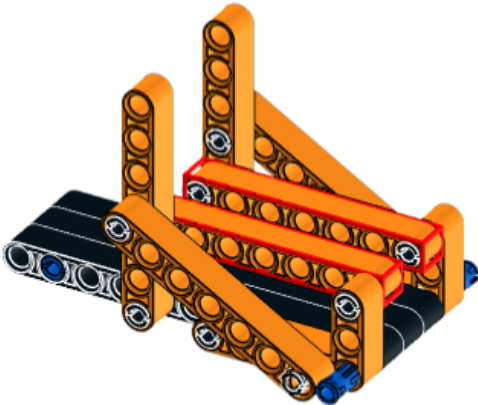
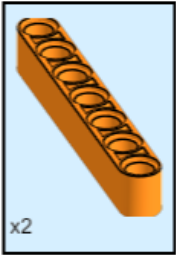
21



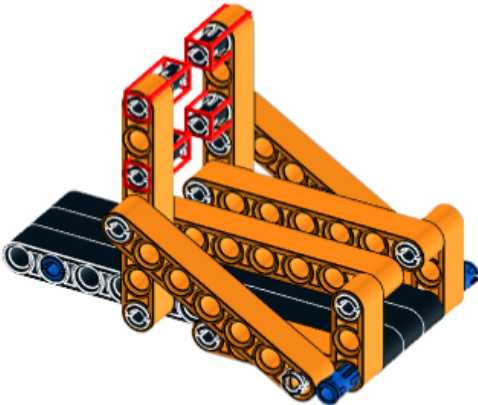
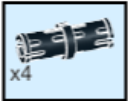
22



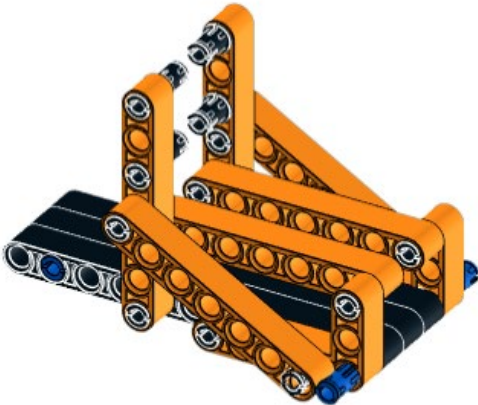
23



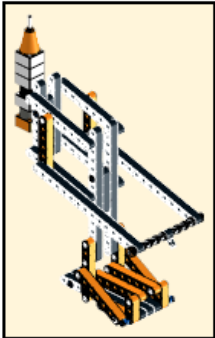
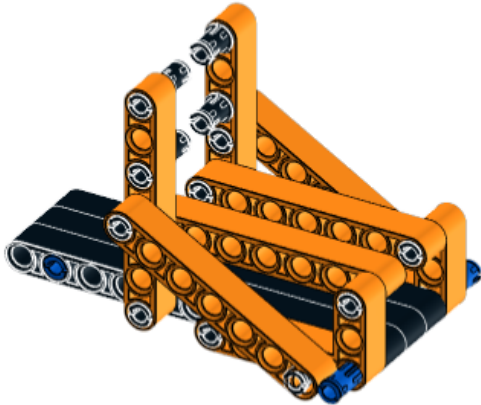
24



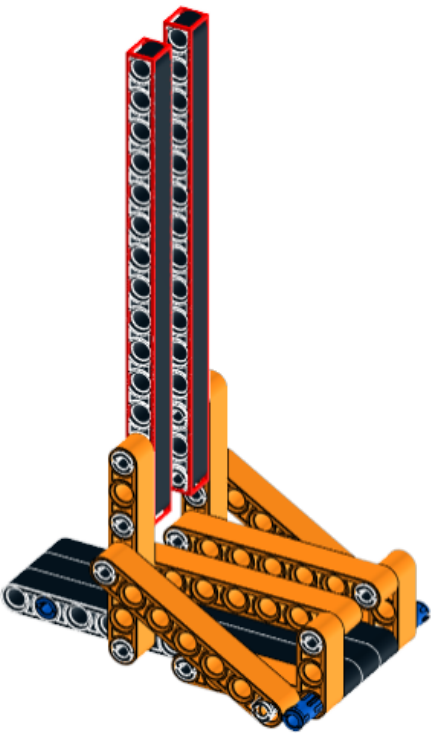
25



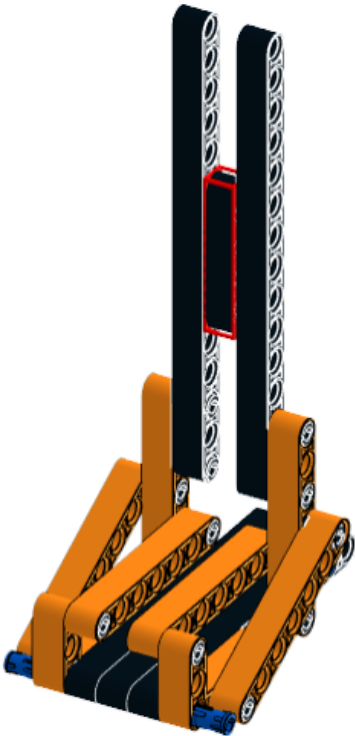
26



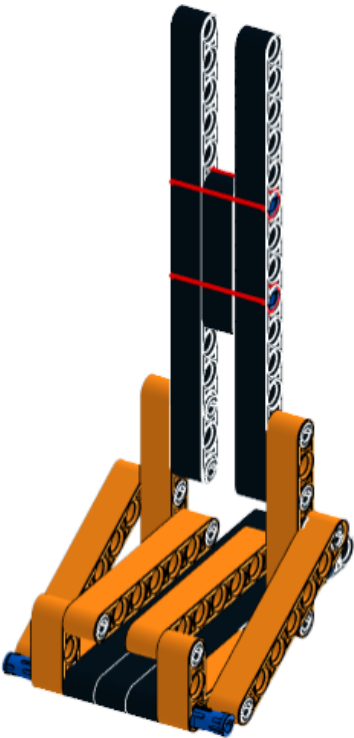
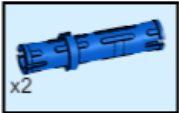
27



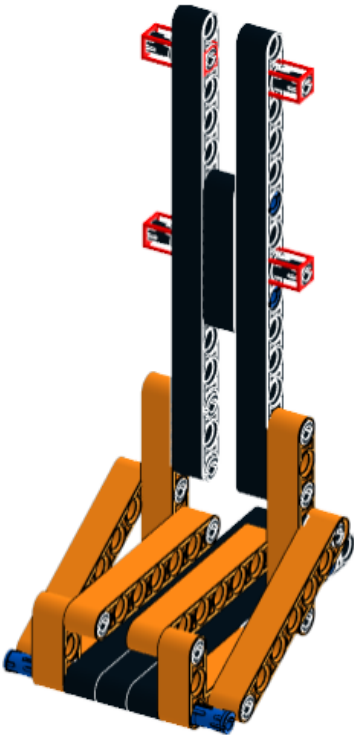
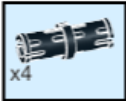
28



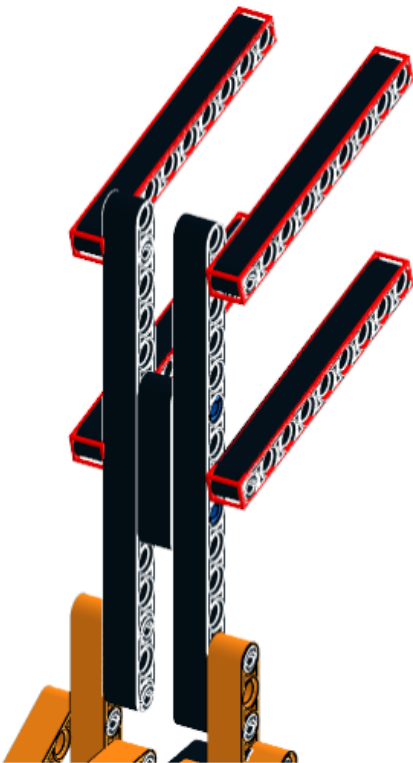
29



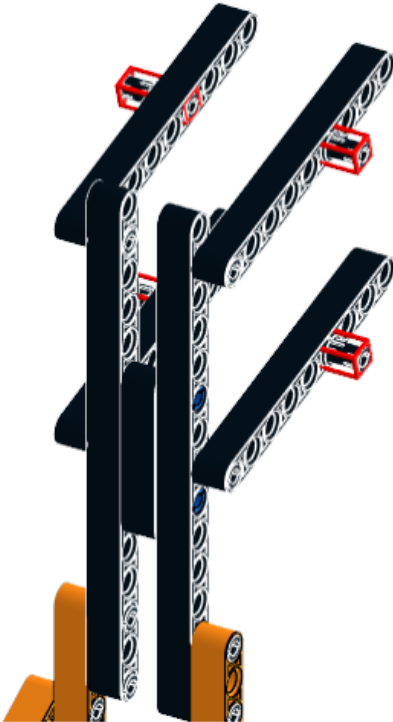
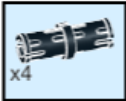
30



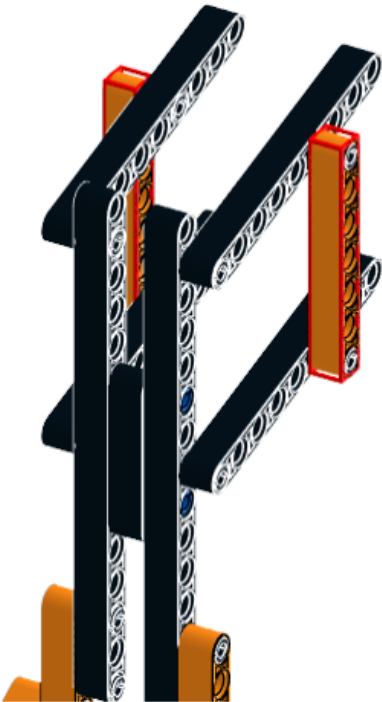
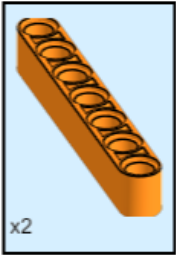
31



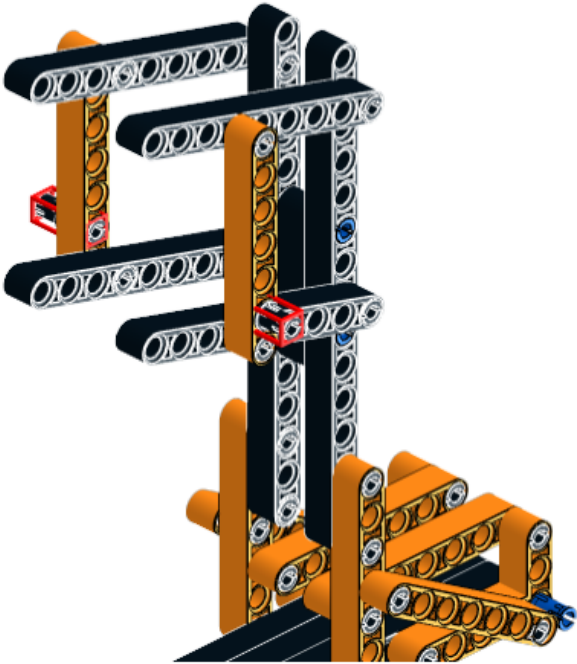
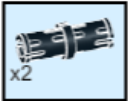
32



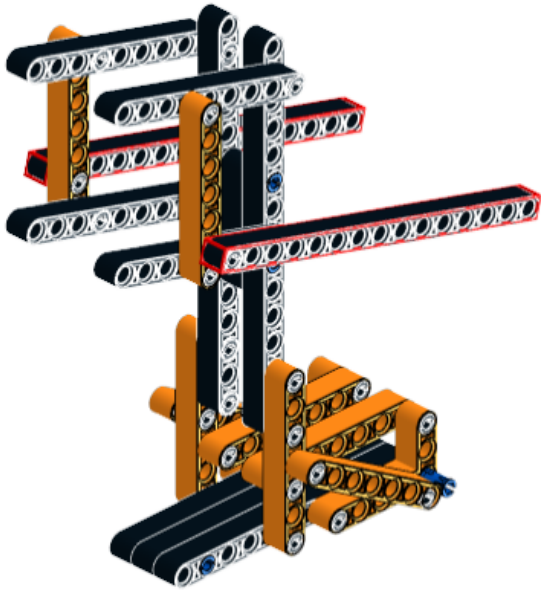
33



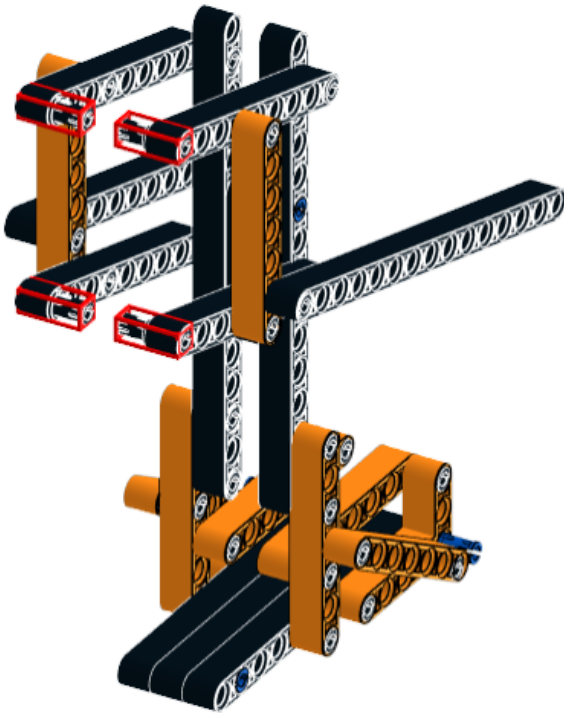
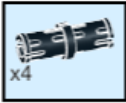
34



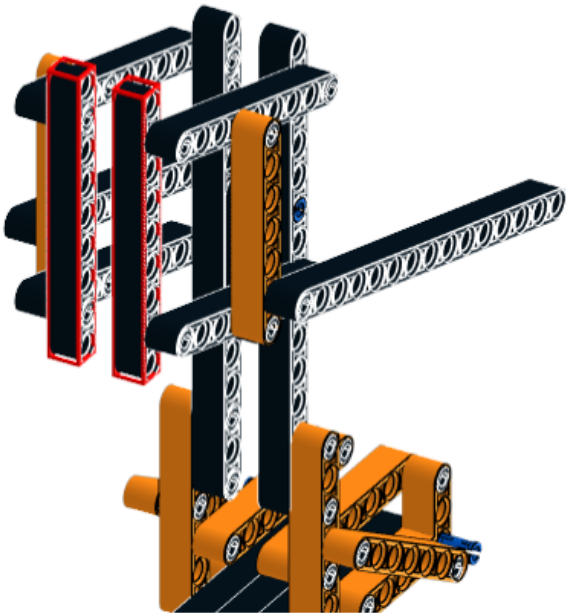
35



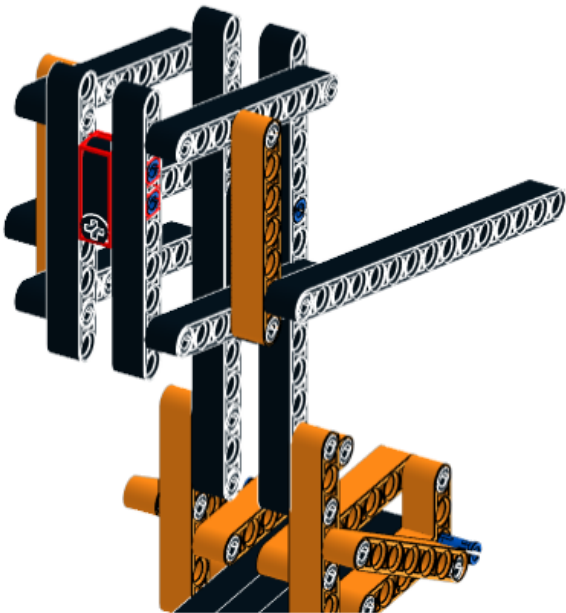
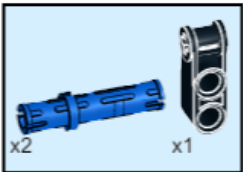
36



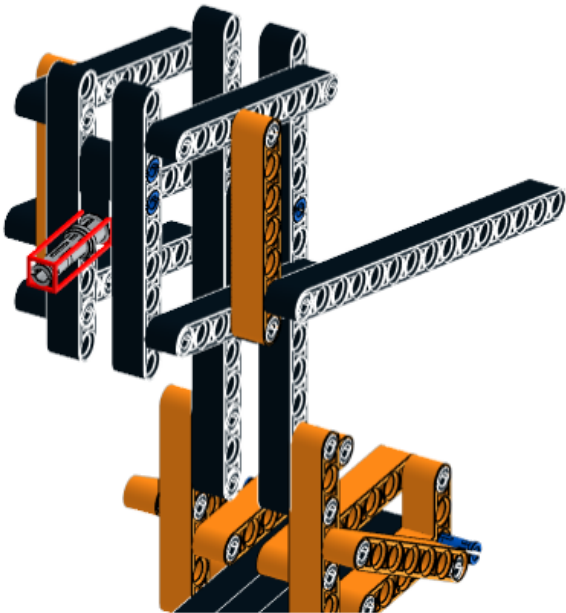
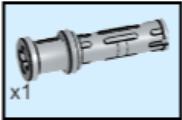
37



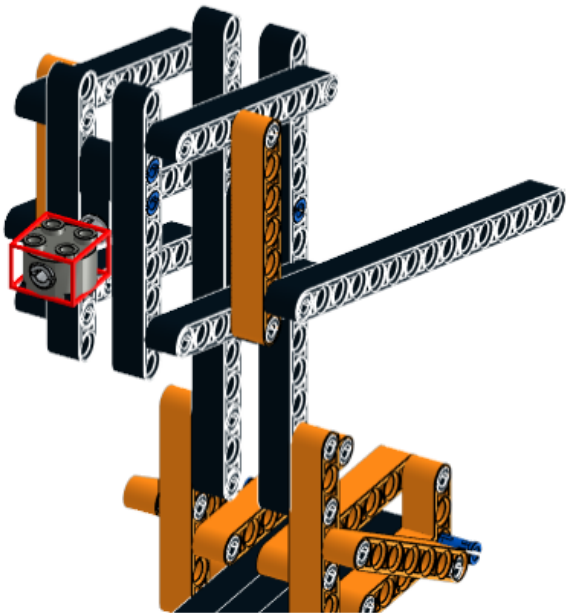
38



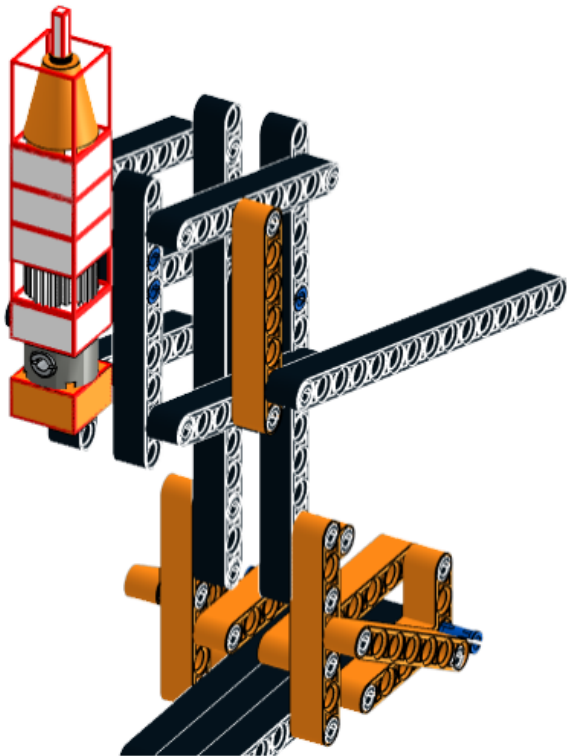
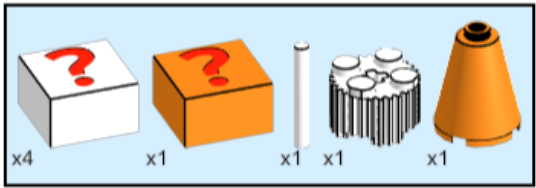
39



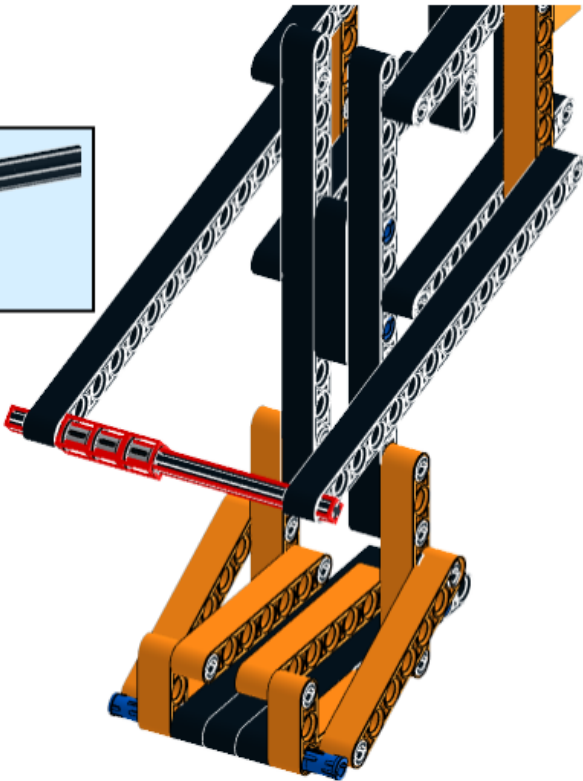
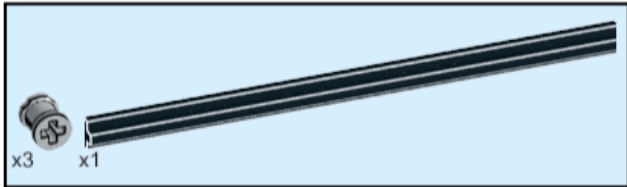
40



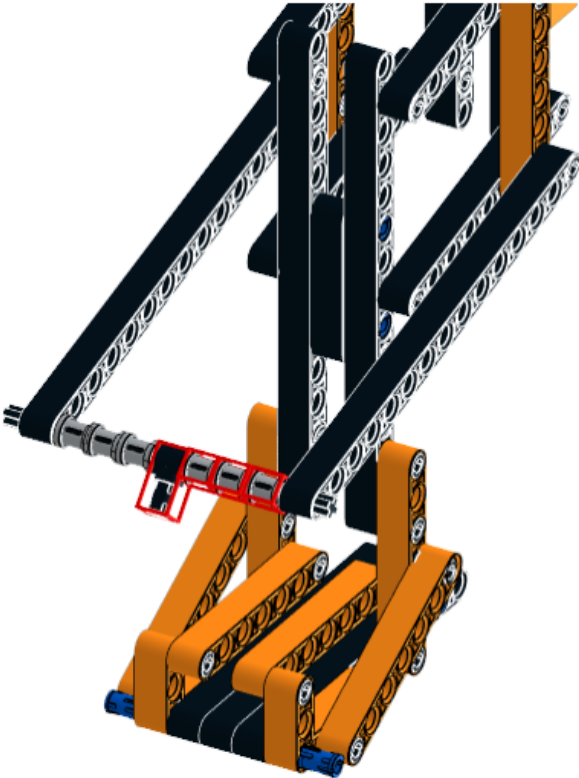
41



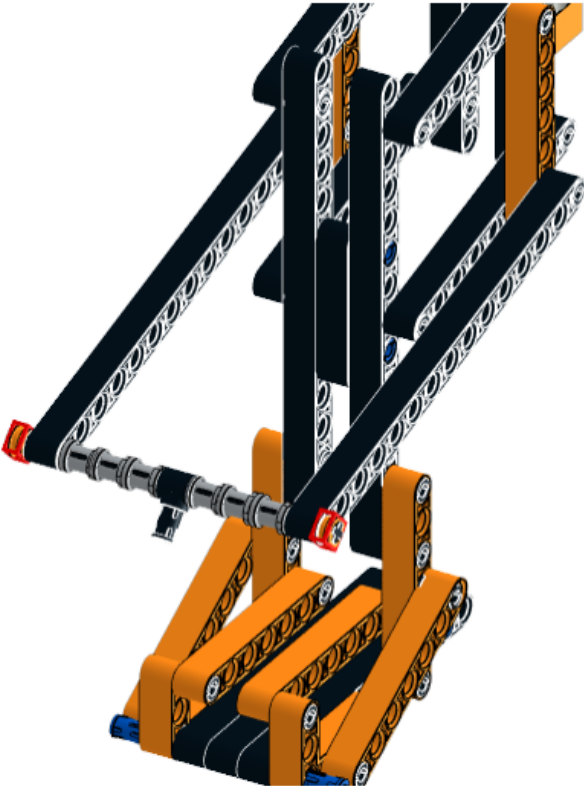
42



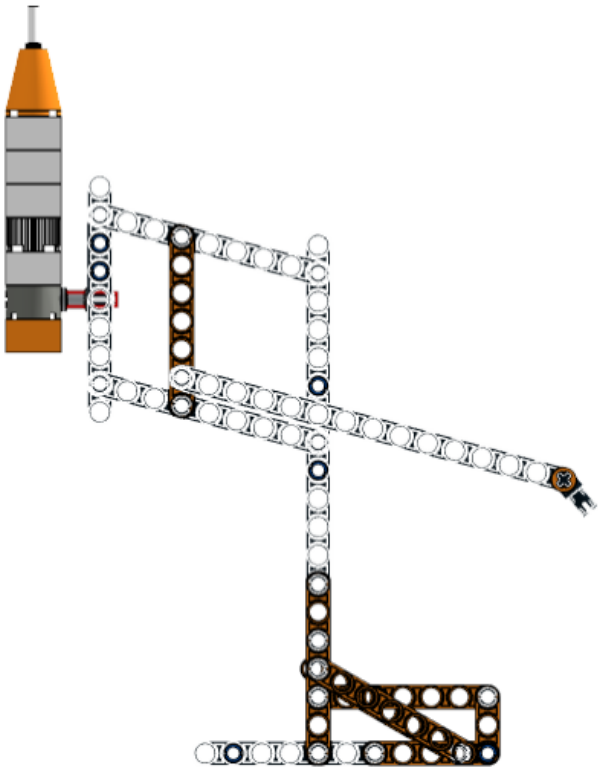
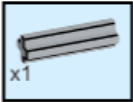
43



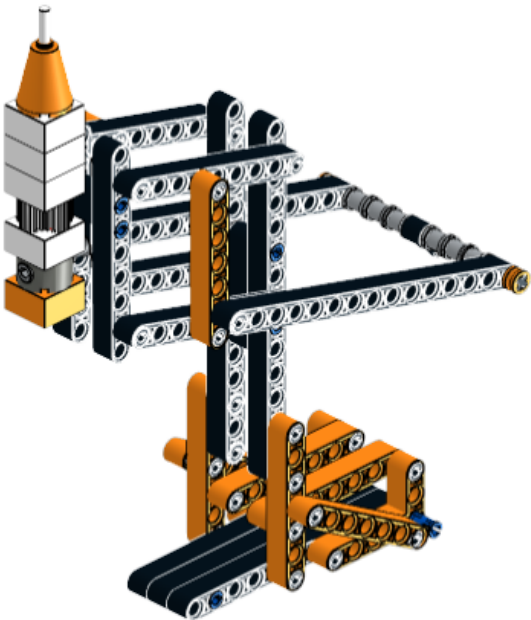
44



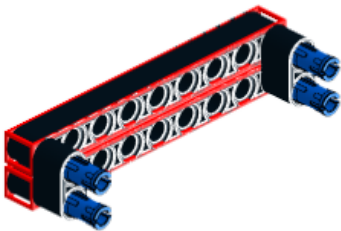
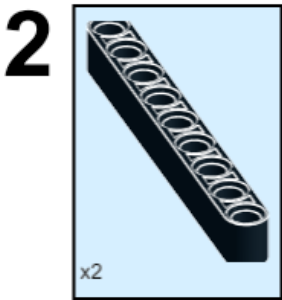
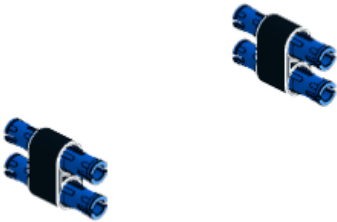
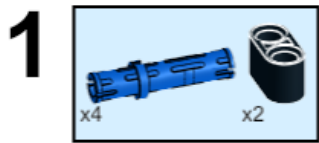
45



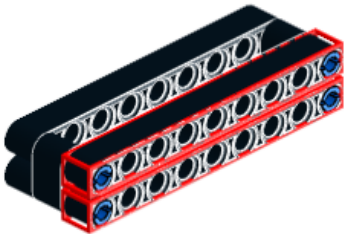
46



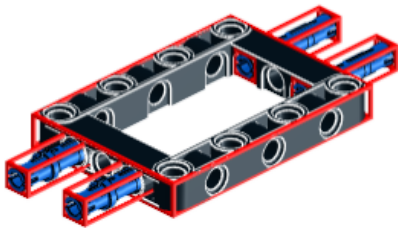
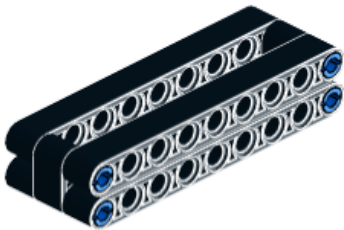
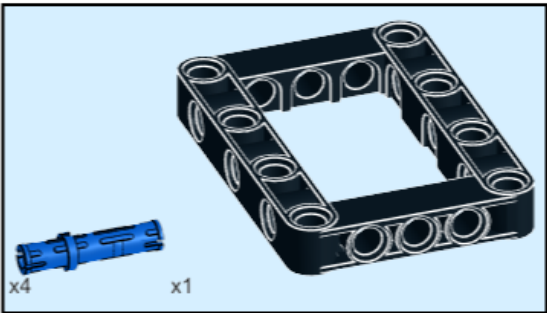
六、高中组桥梁

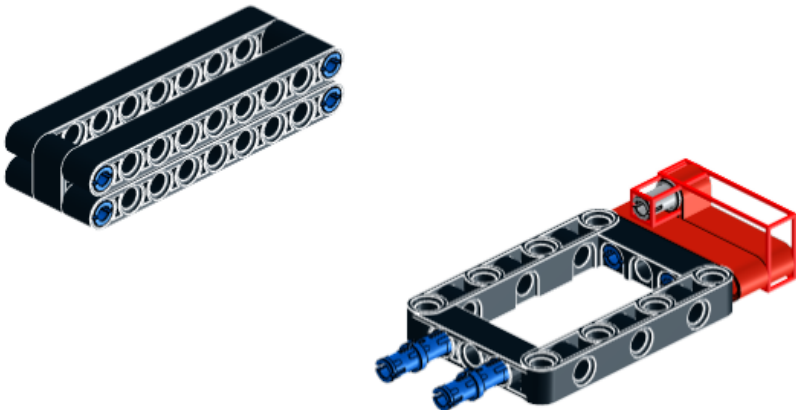
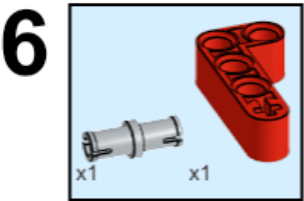
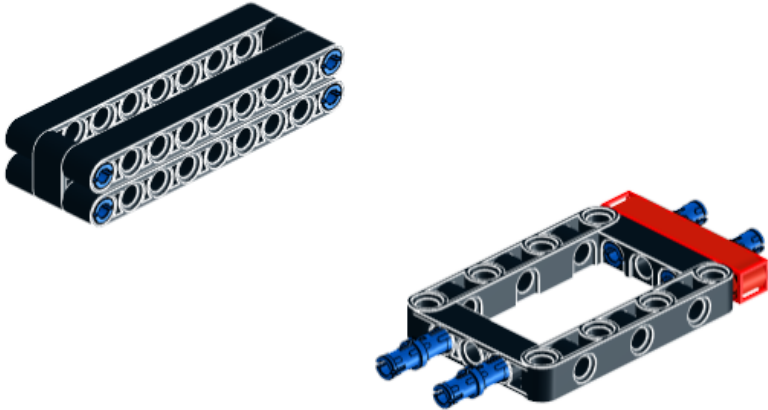
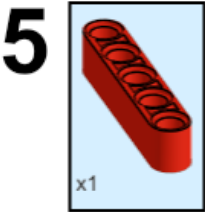


3

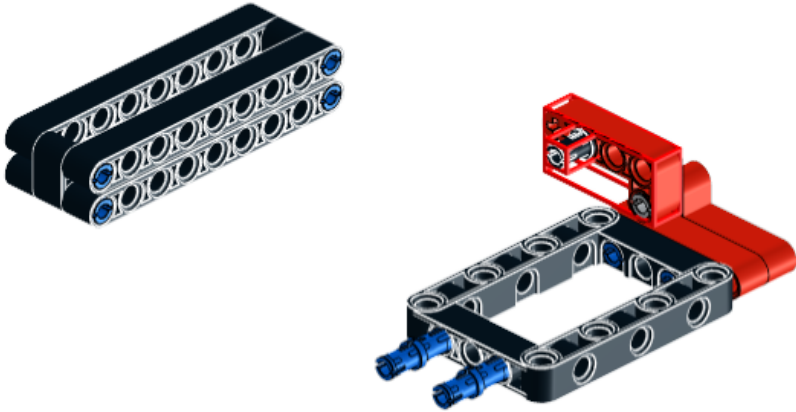
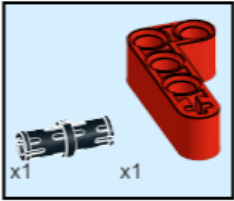


4

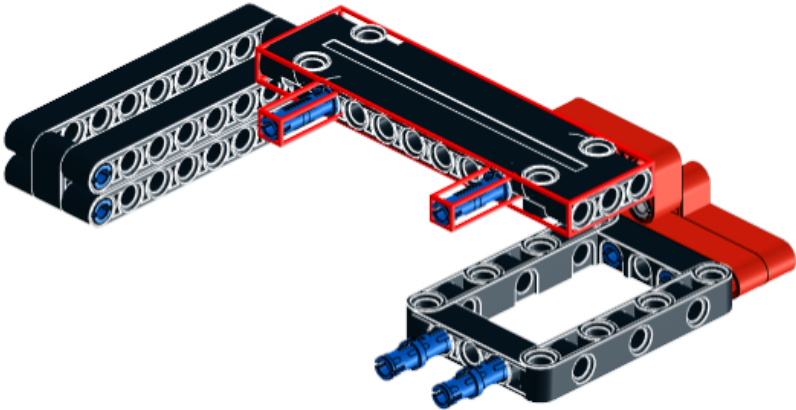
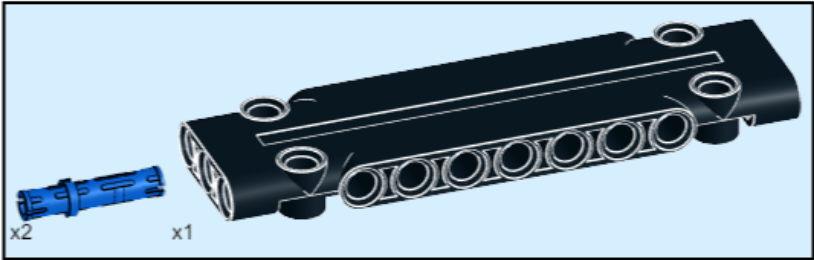


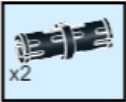


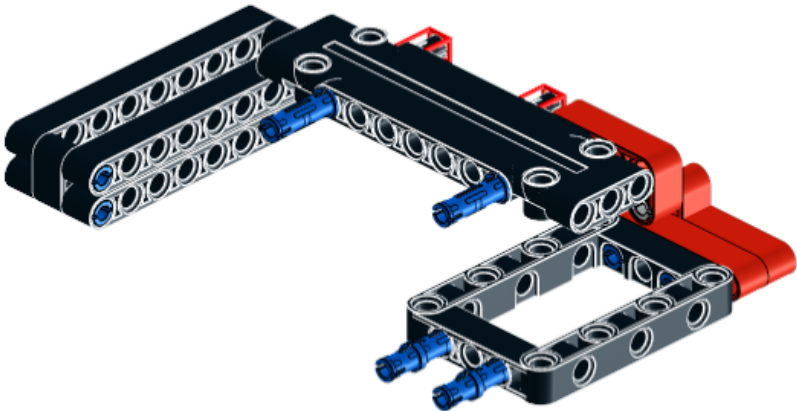
7

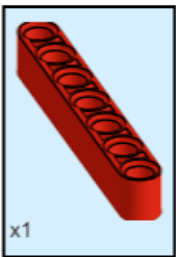


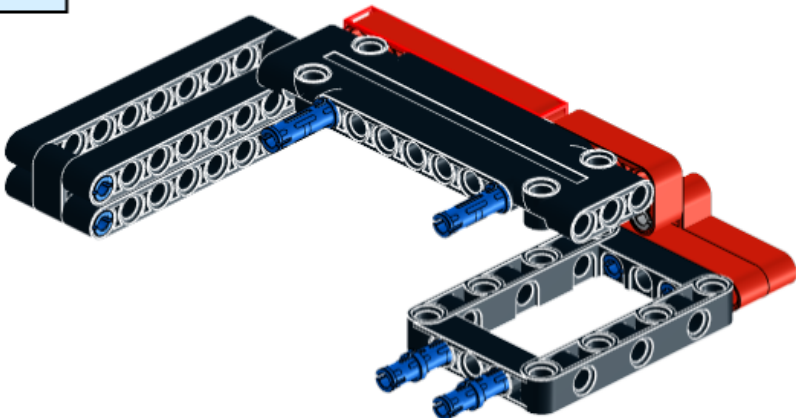
8



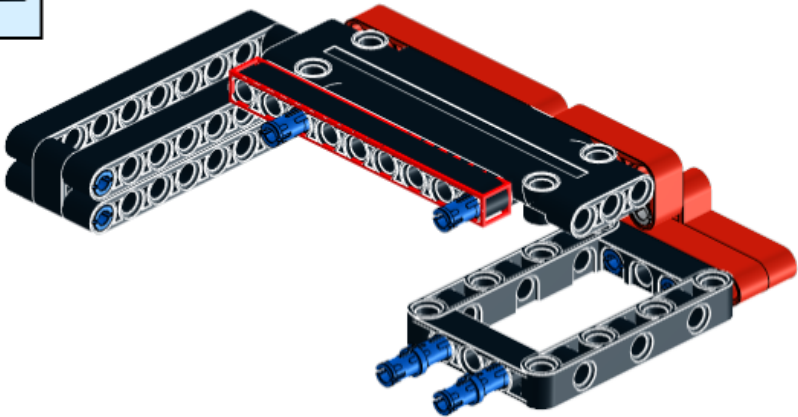
9  x2



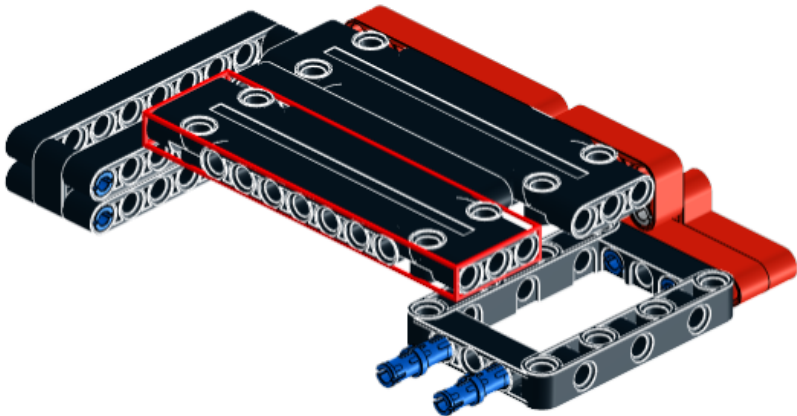
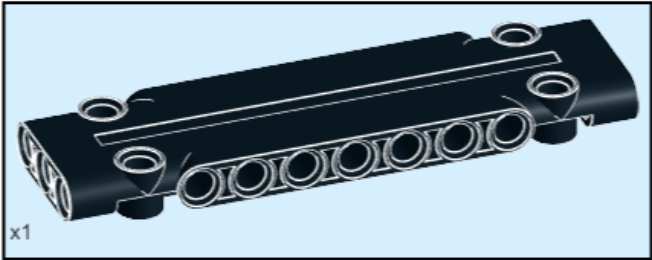
10  x1



11

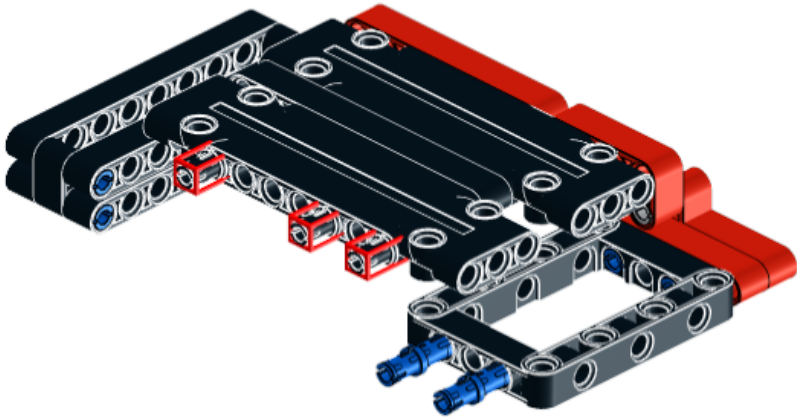


12



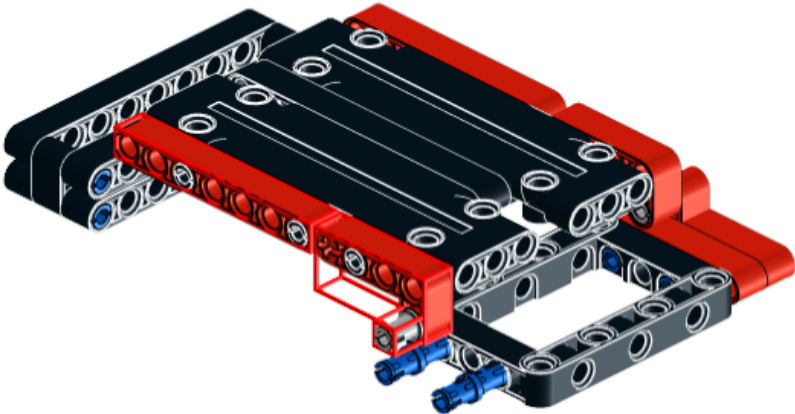
13

x3

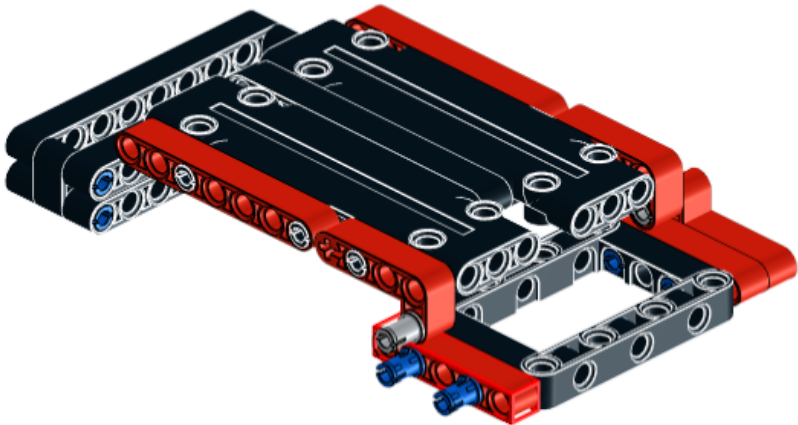
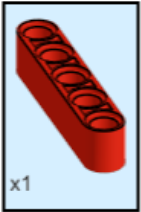


14

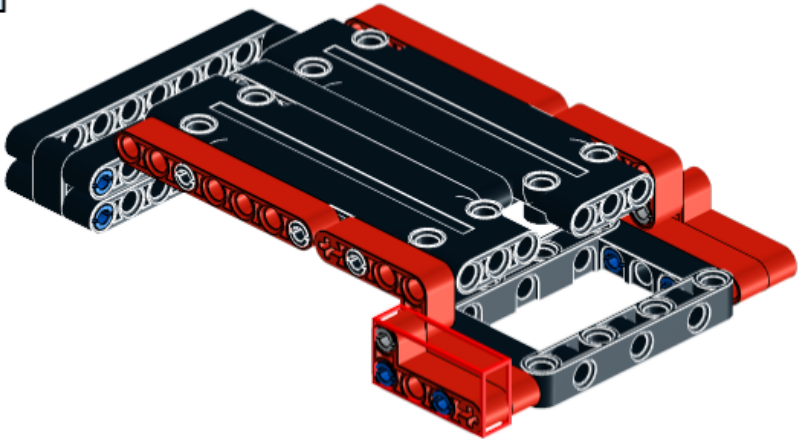
x1 x1 x1

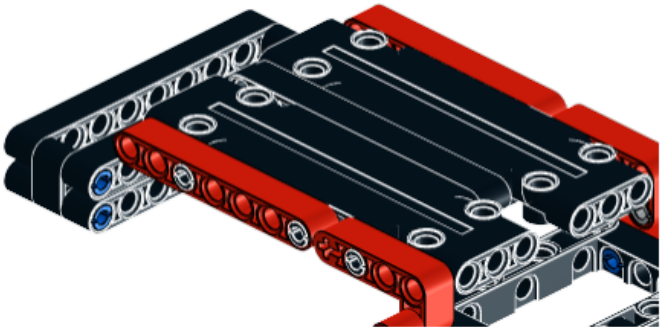
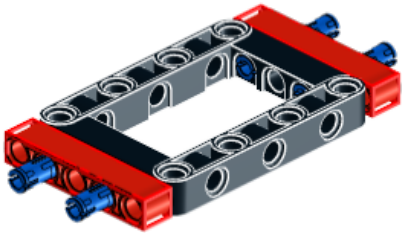
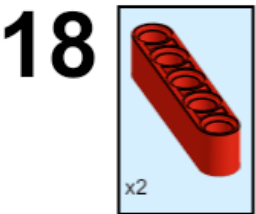
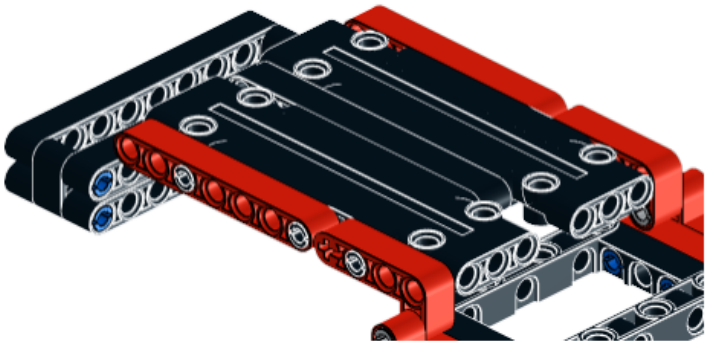
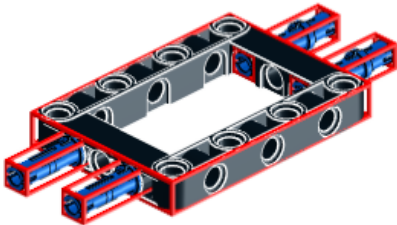
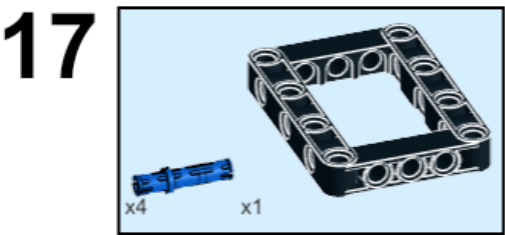


15

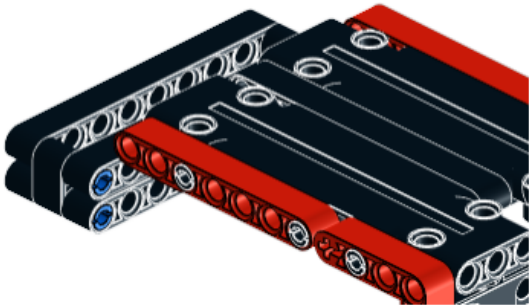
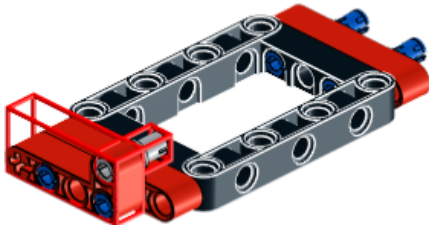
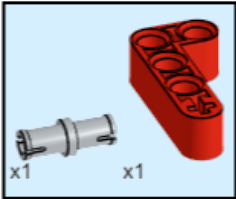


16

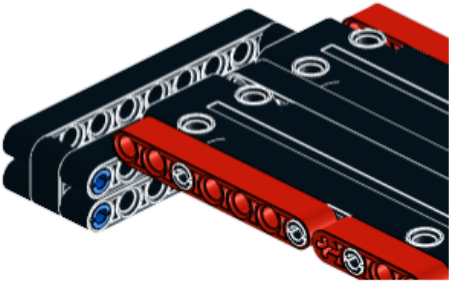
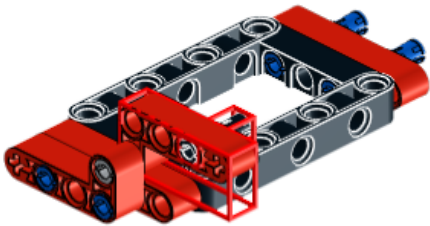
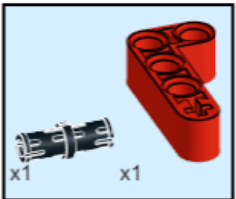




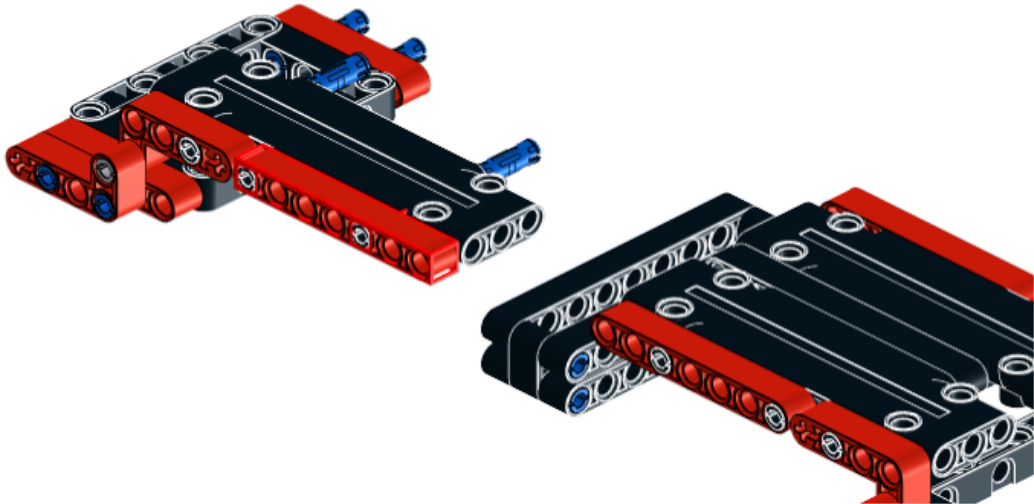
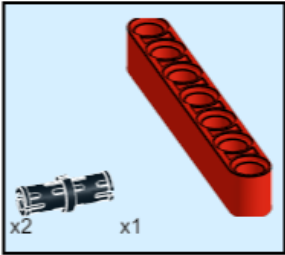
19



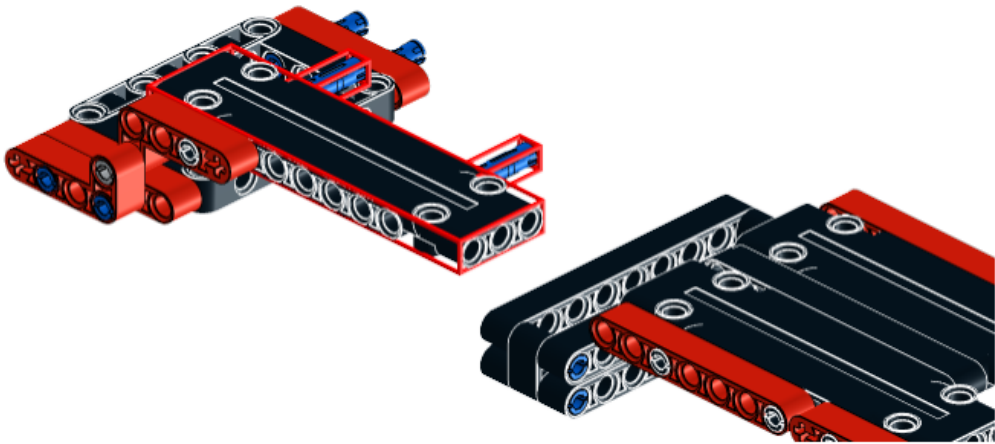
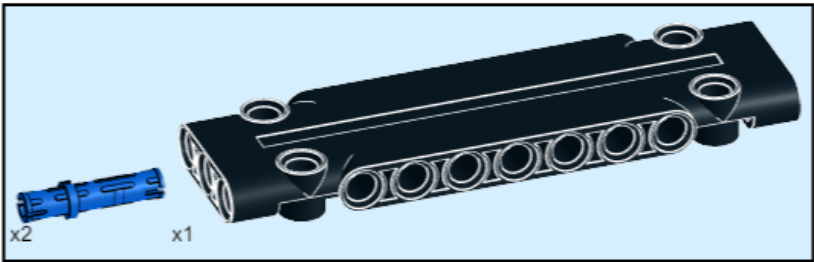
20



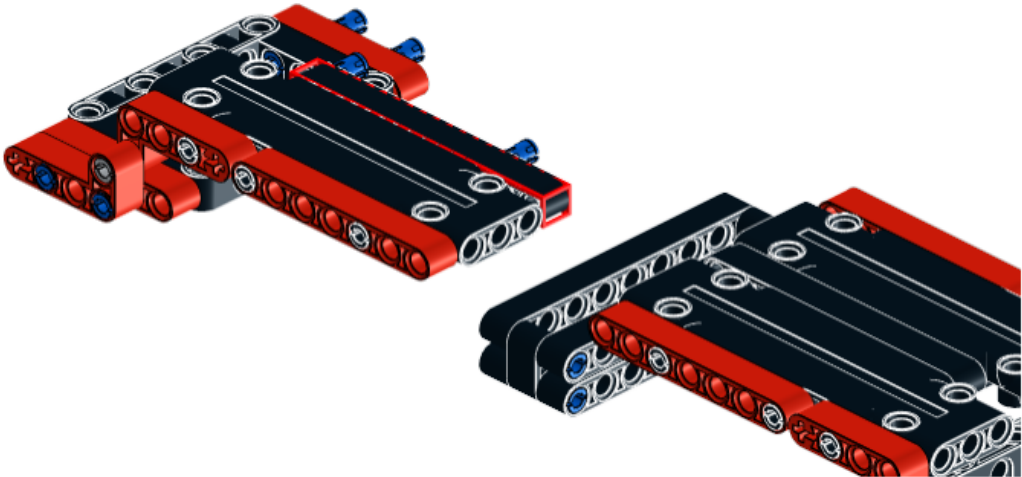
22



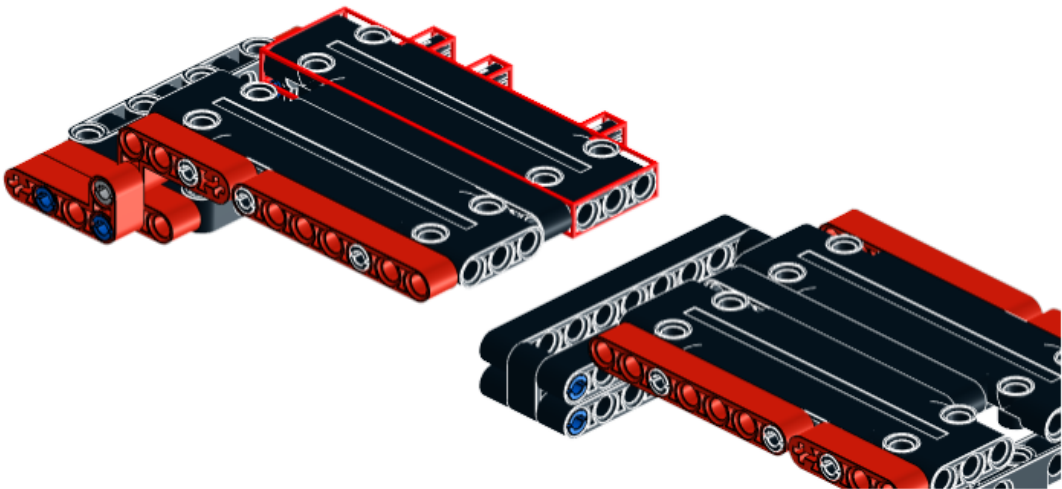
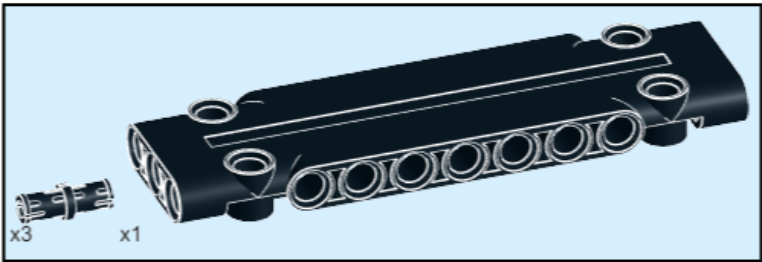
21



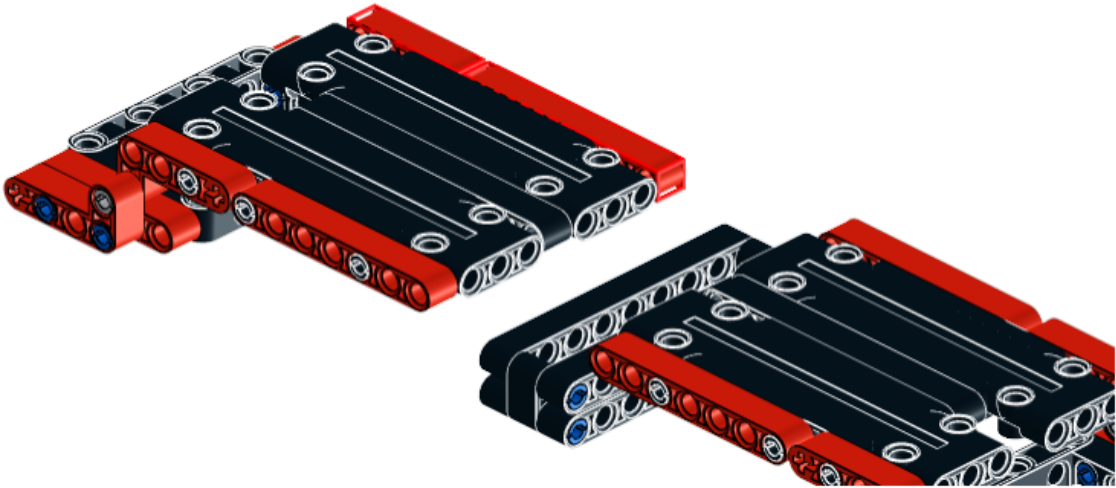
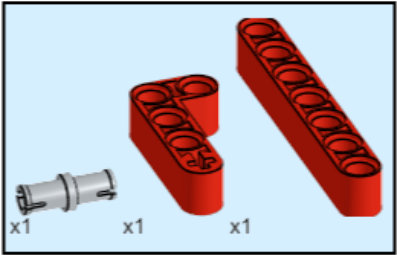
23



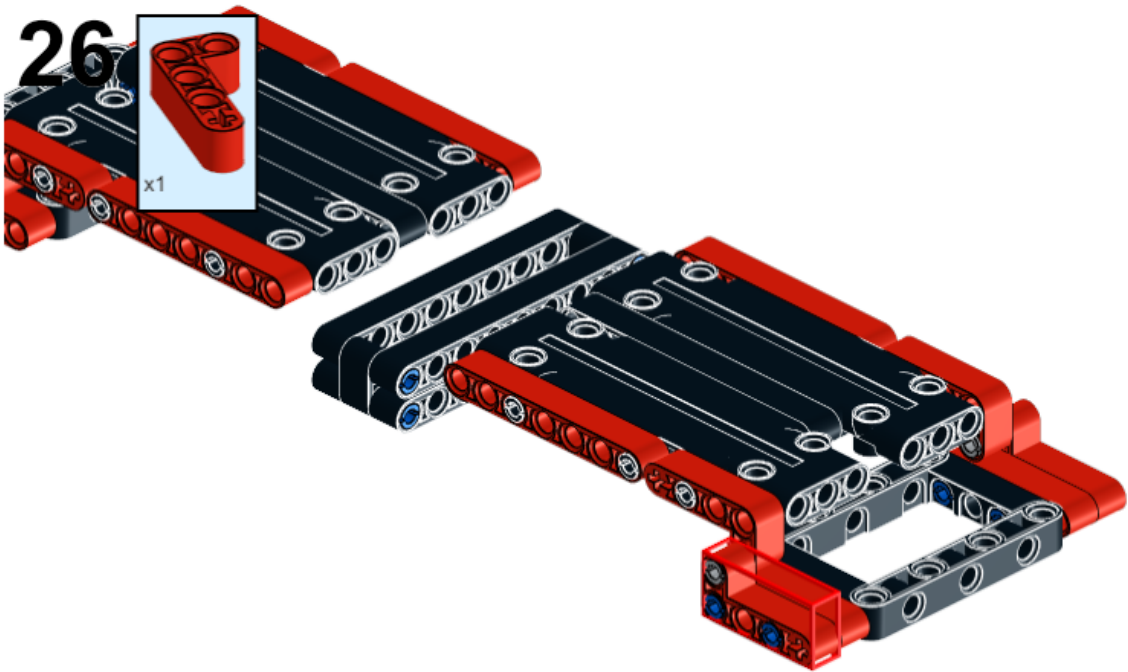
24



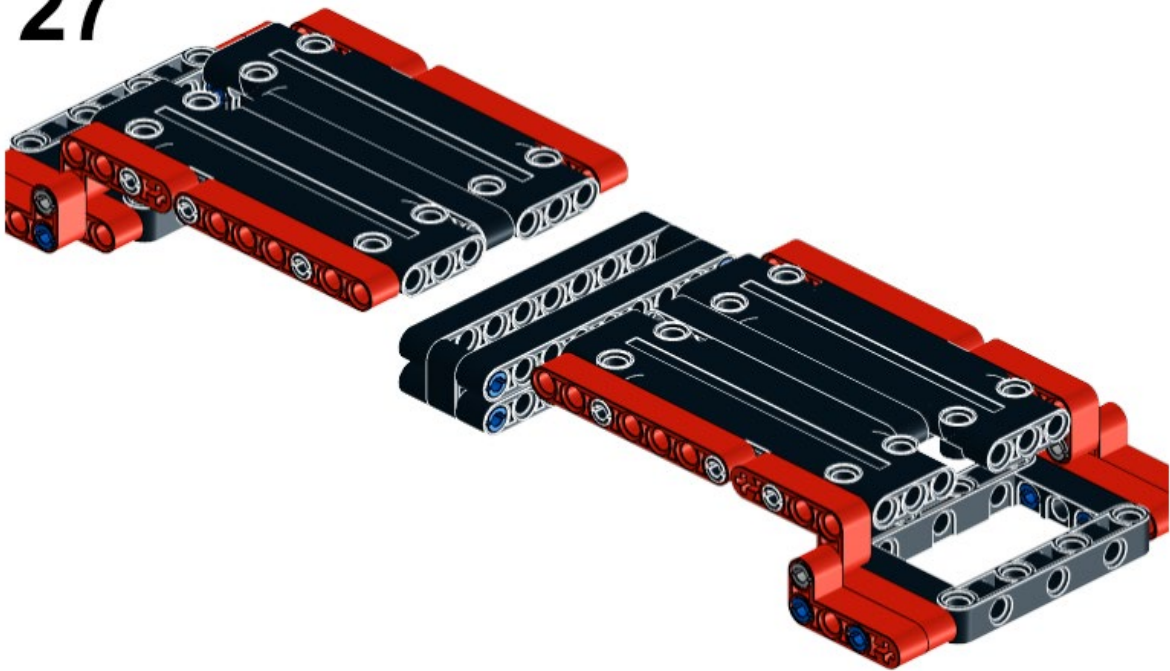
25



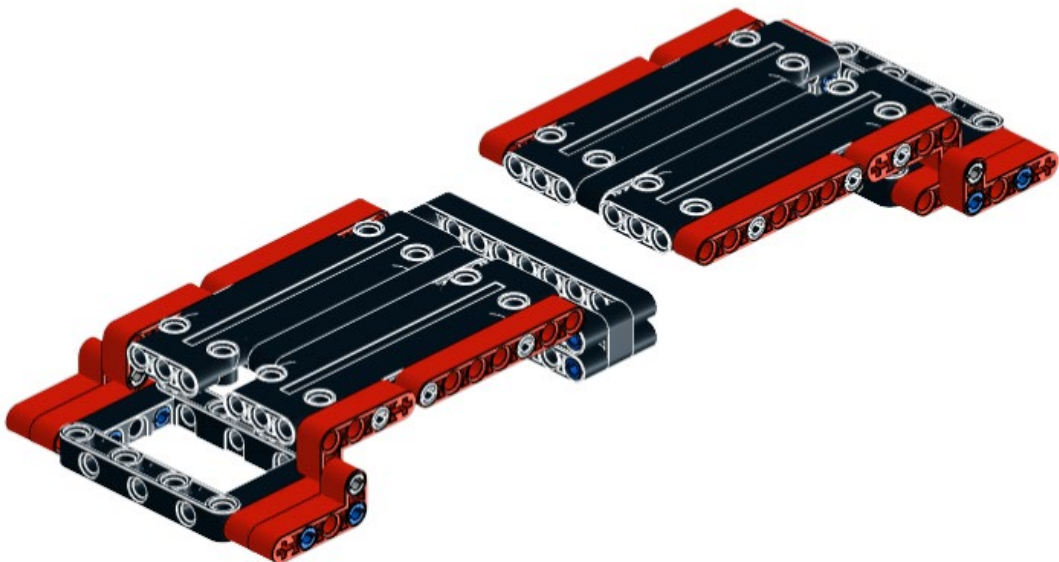
26



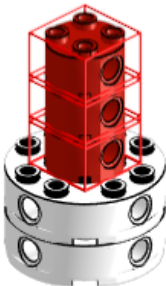
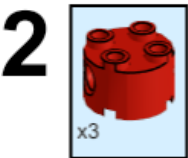
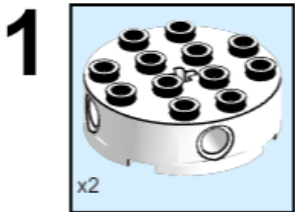
27



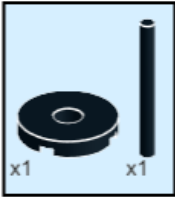
28



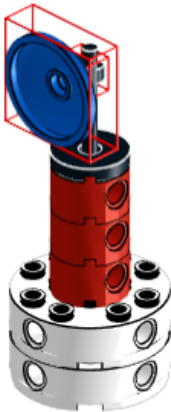
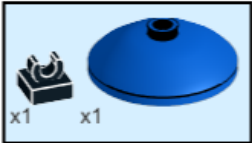
七、高中组旗帜

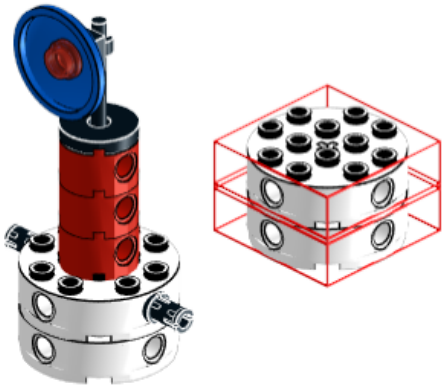
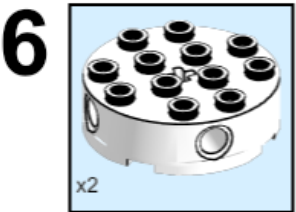
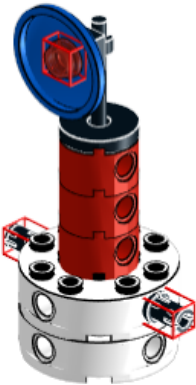


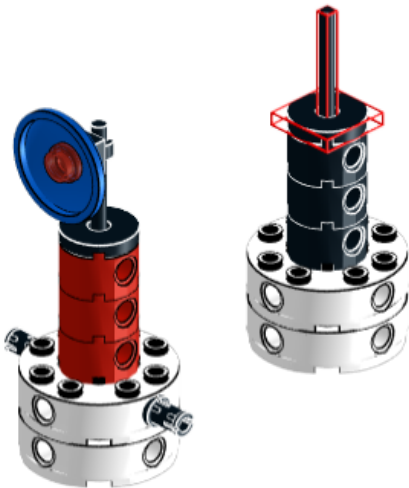
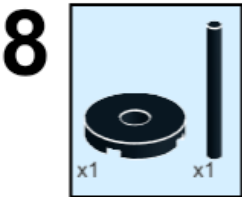
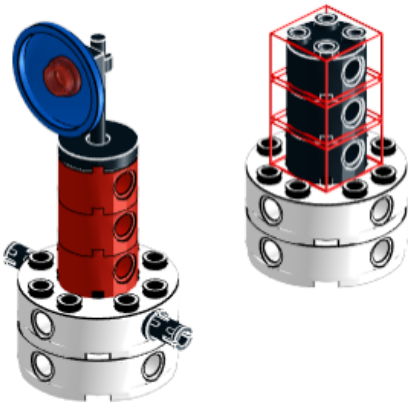
3

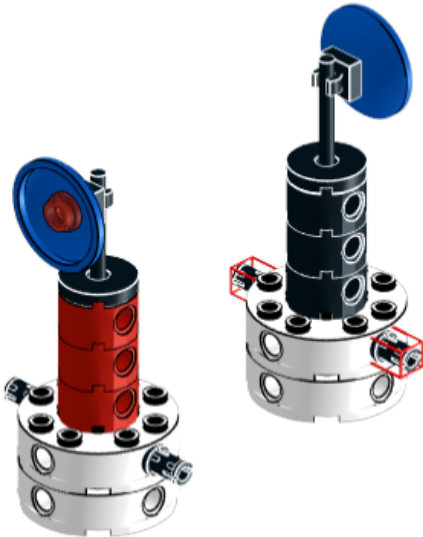
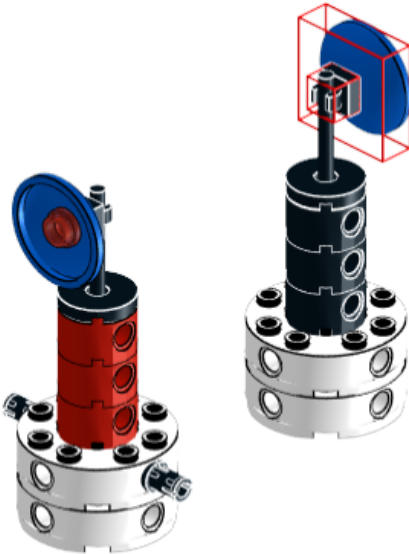
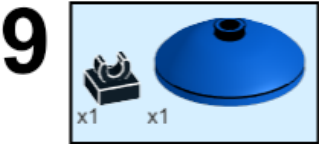


4

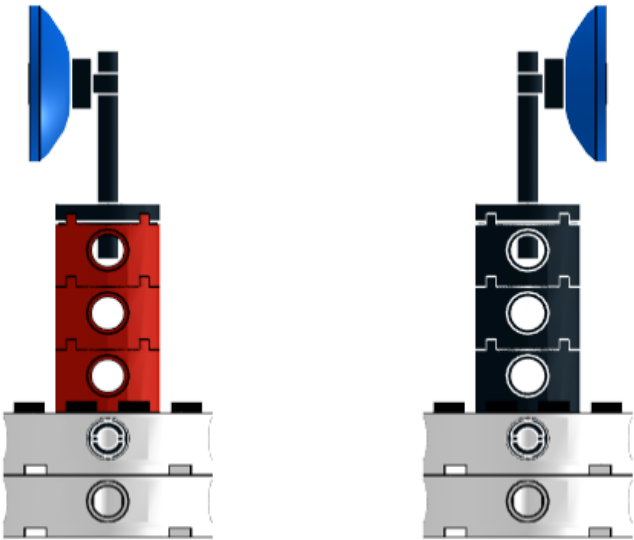




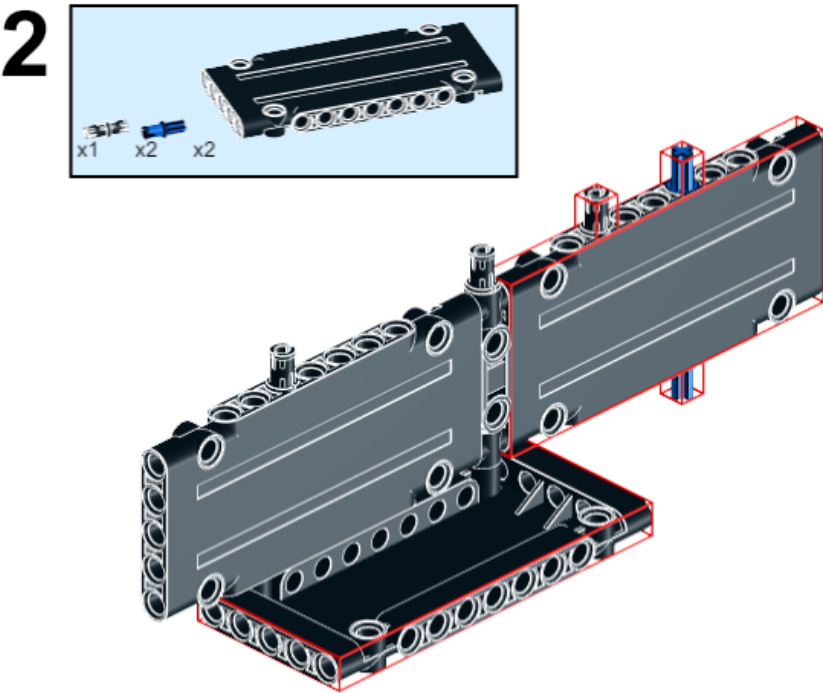
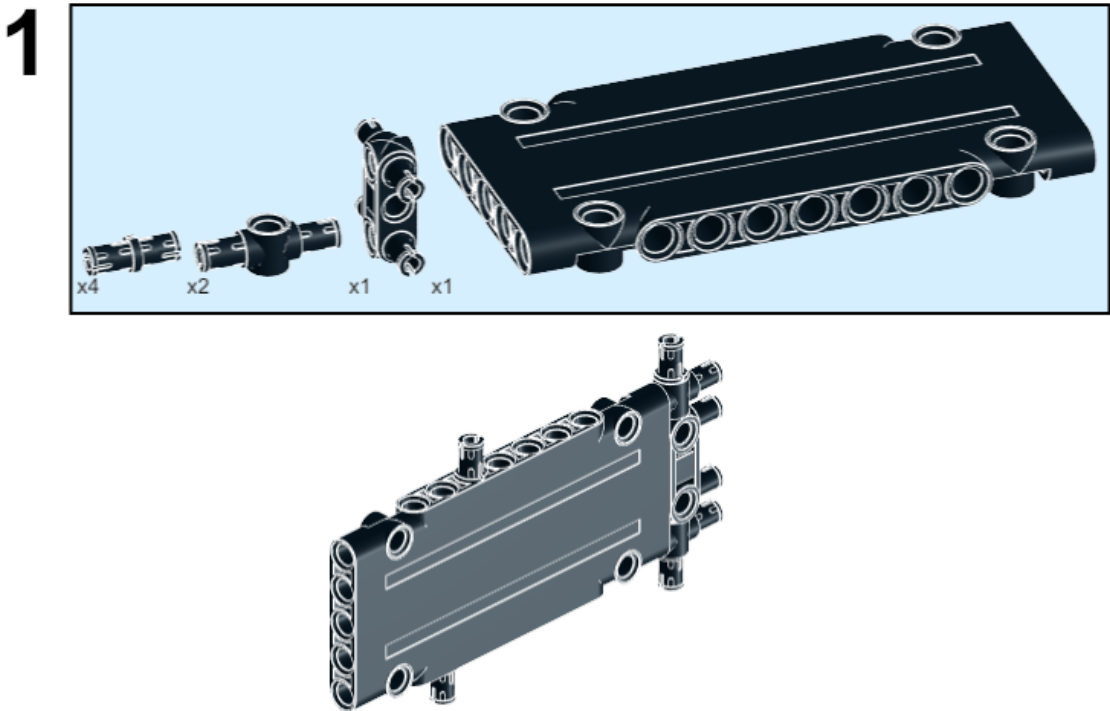




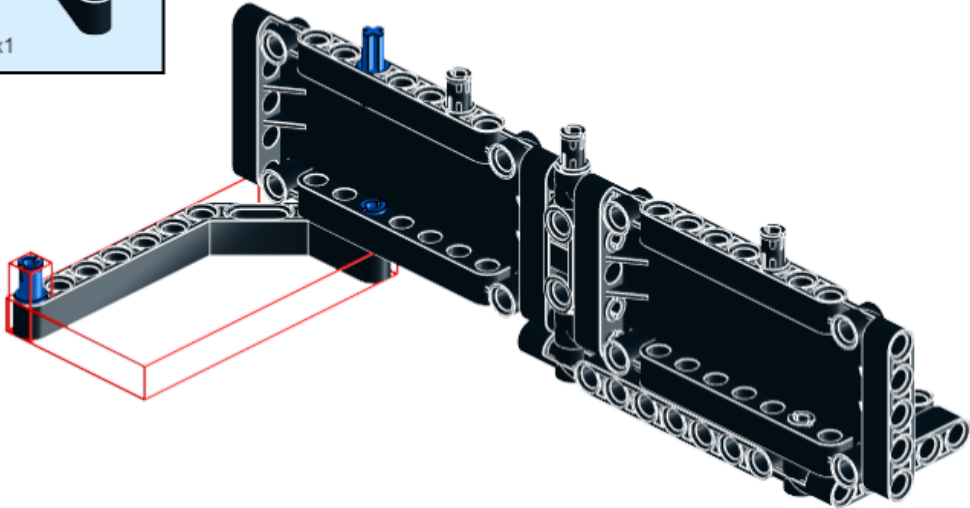
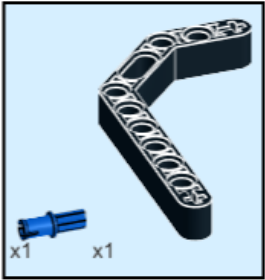
11



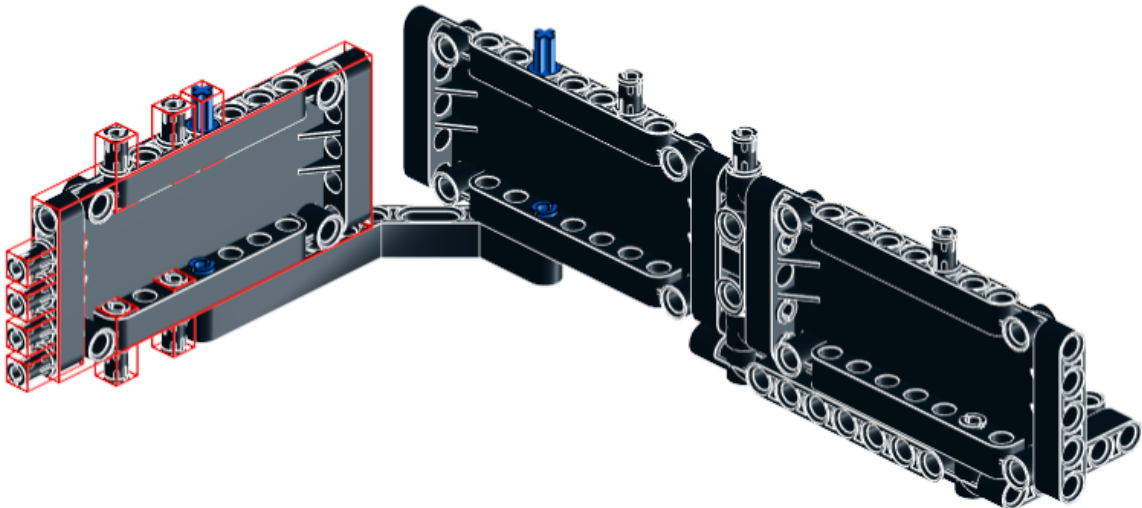
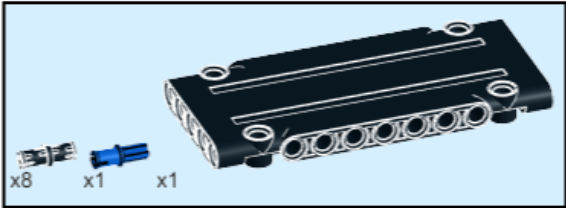
八、高中组收集框



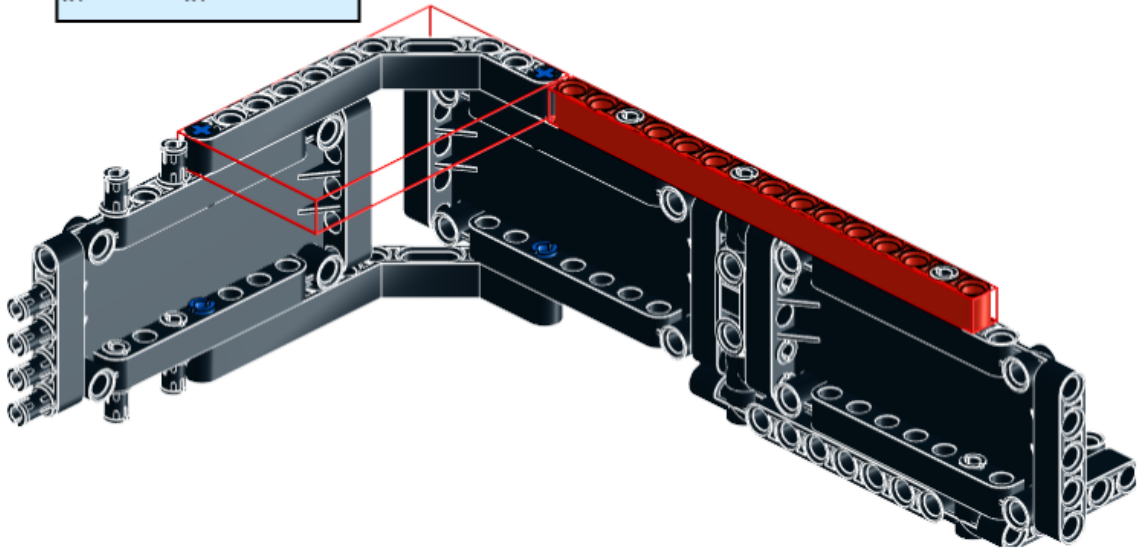
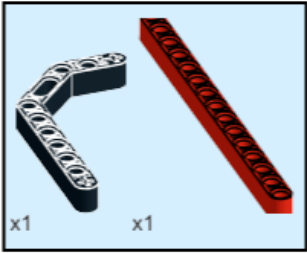
3



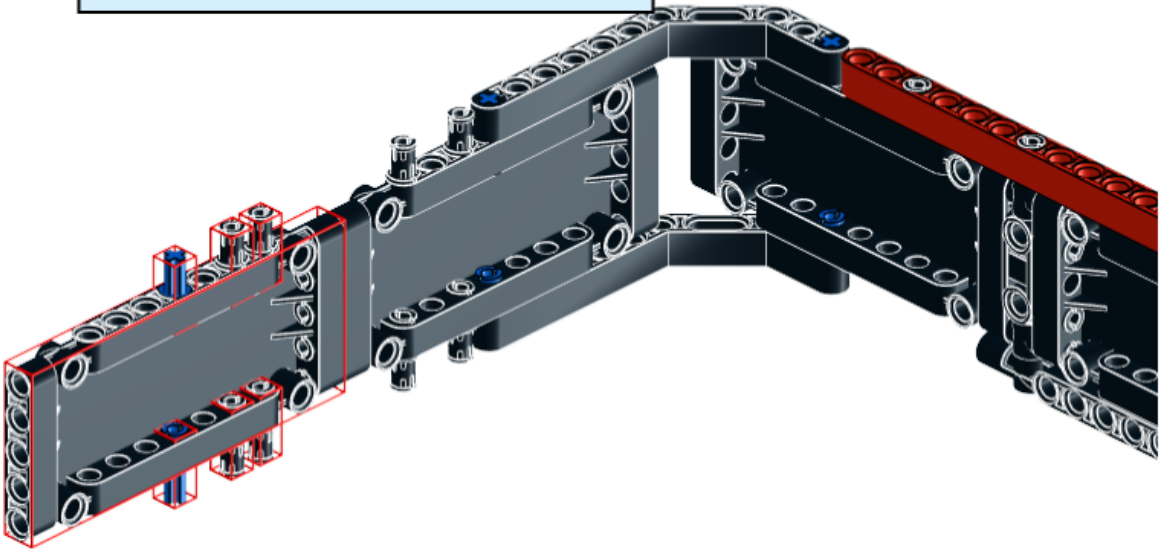
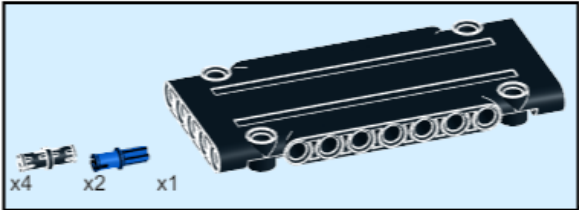
4



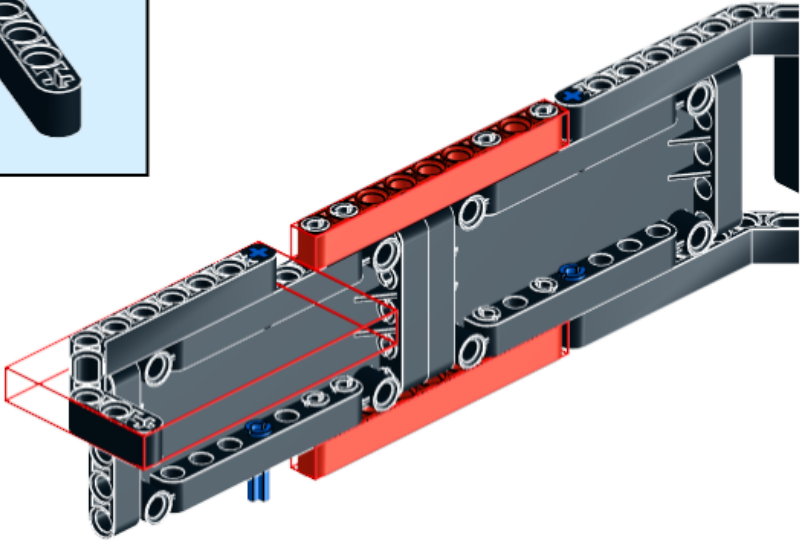
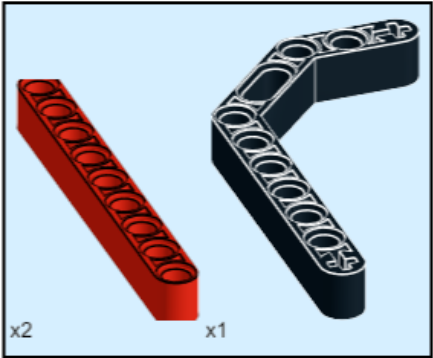
5



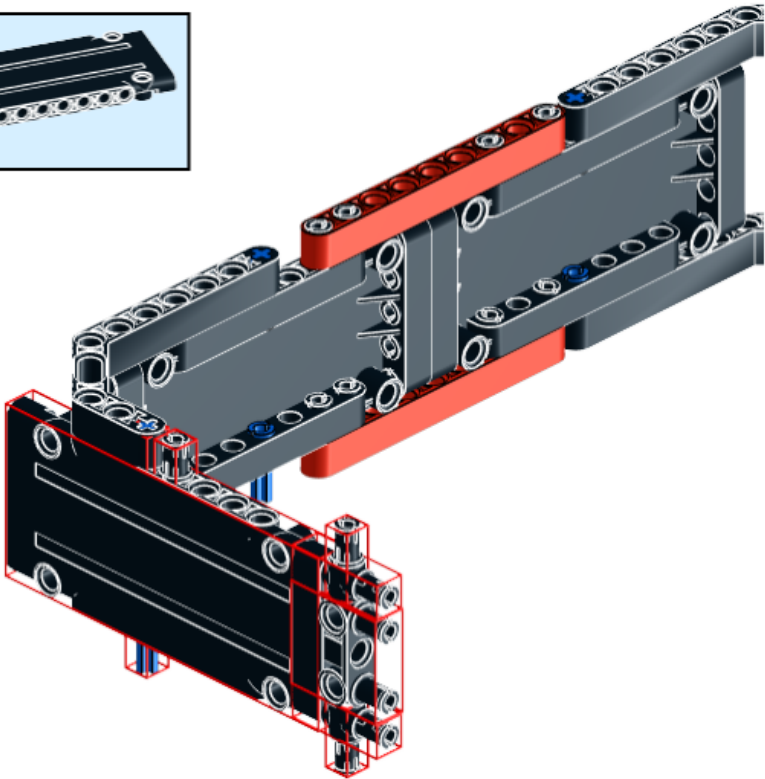
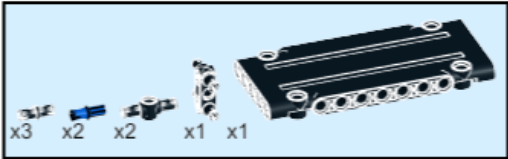
6



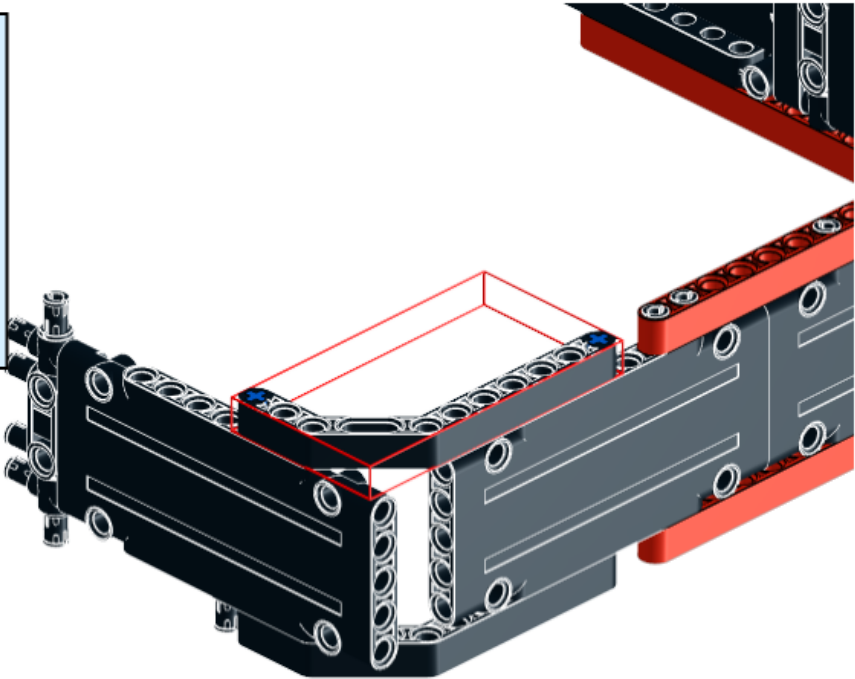
7



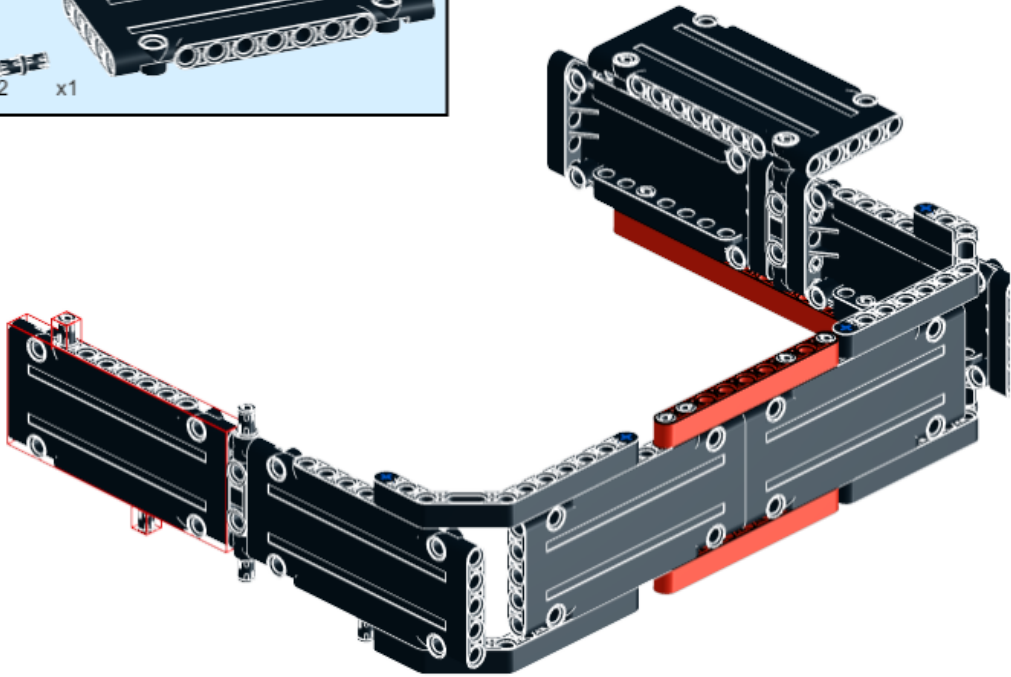
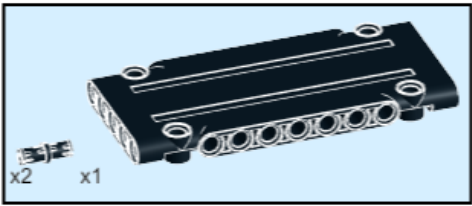
8



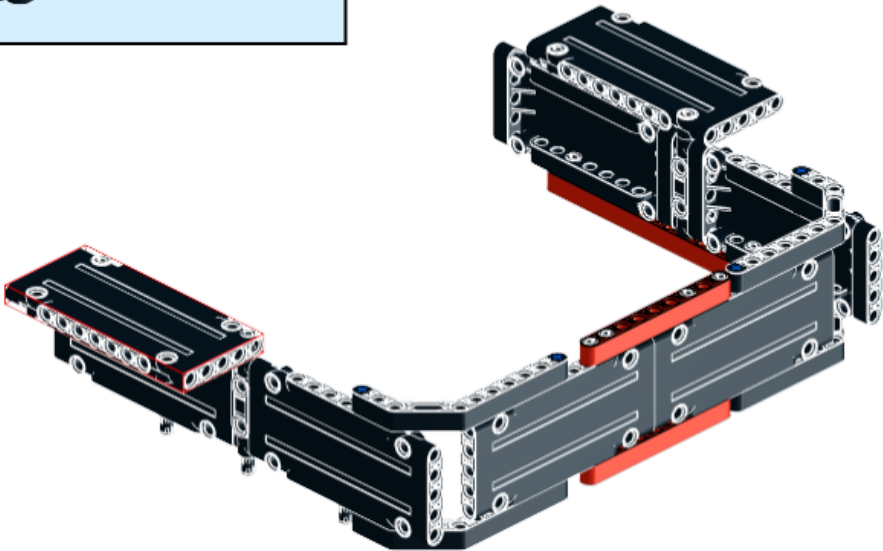
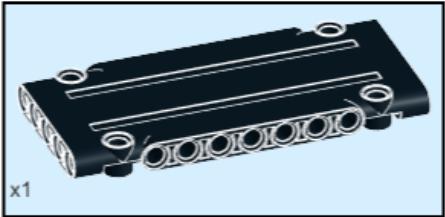
9



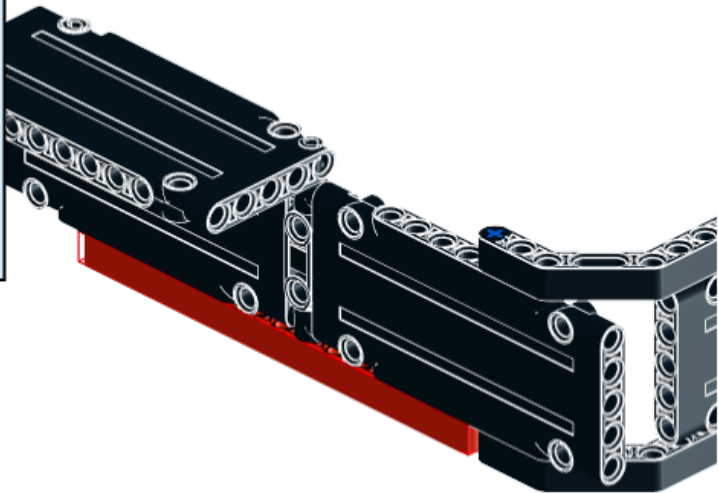
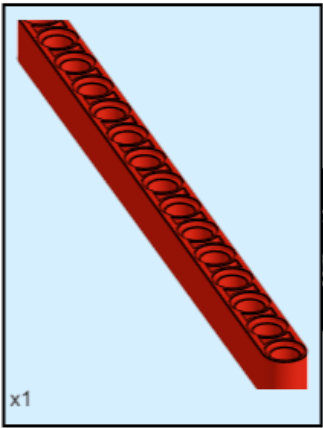
10



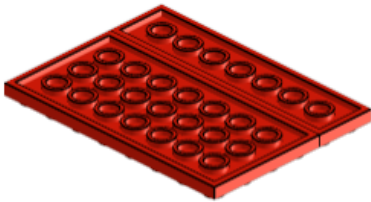
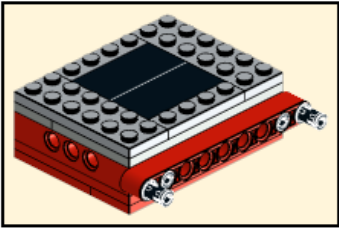
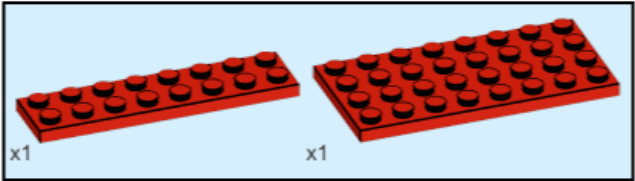
11



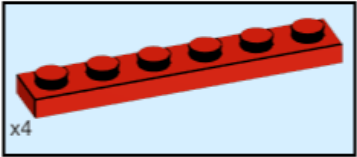
12



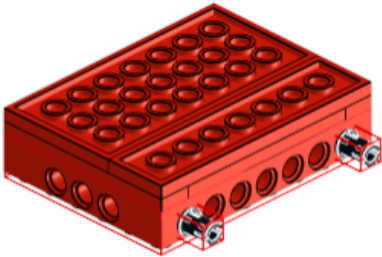
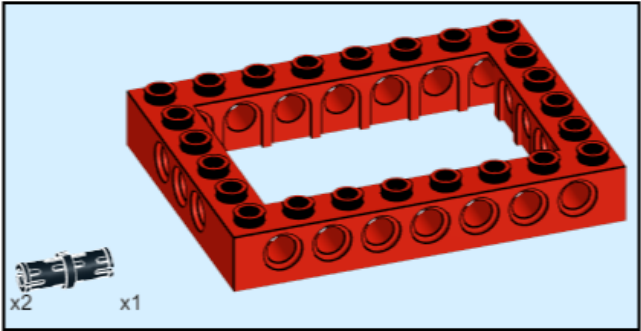
13



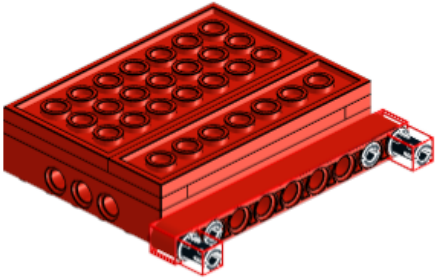
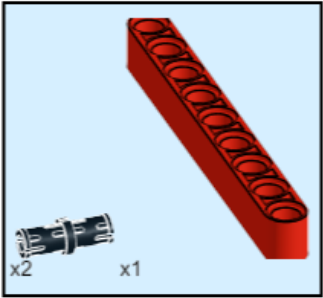
14



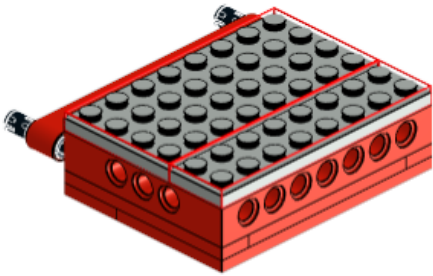
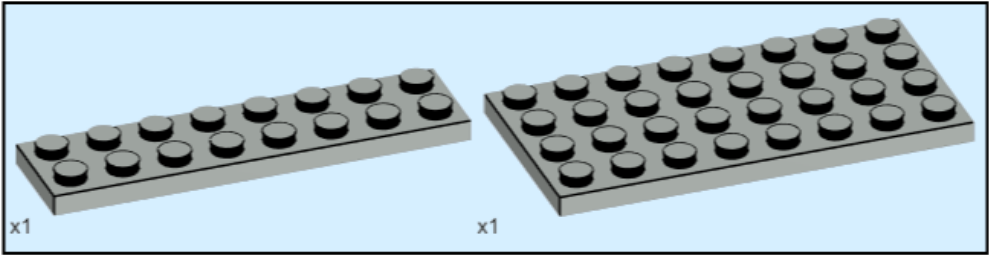
15



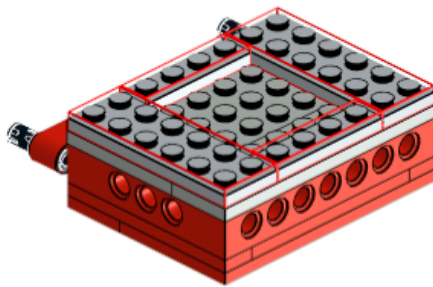
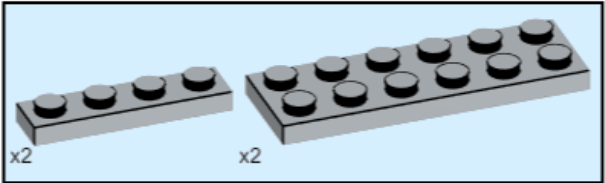
16



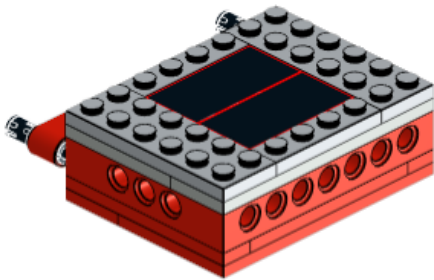
17



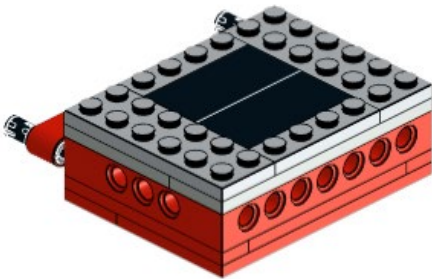
18



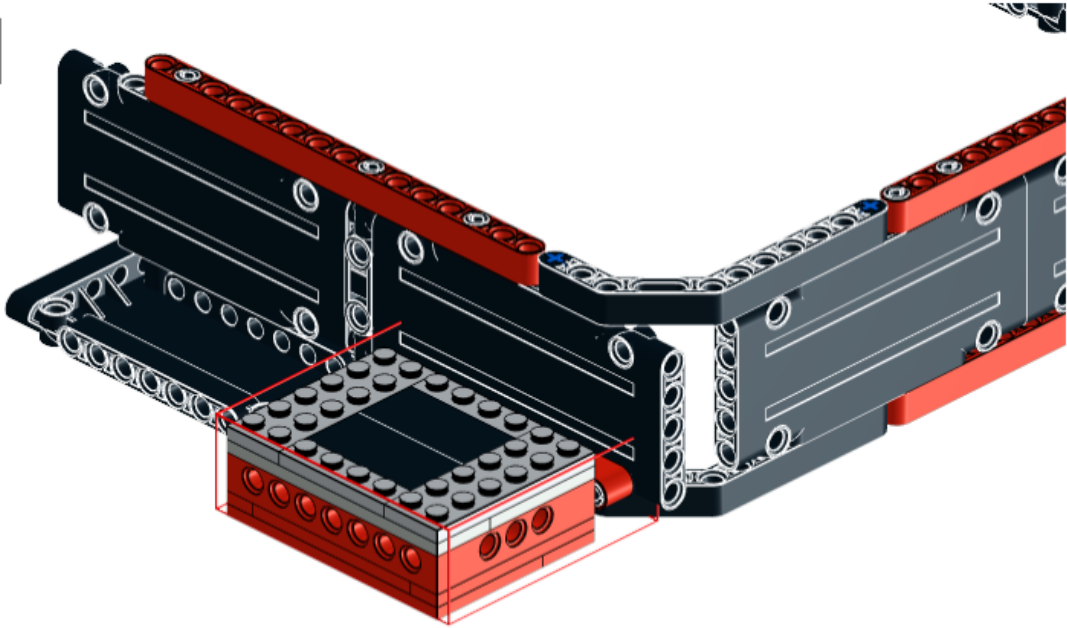
19



20

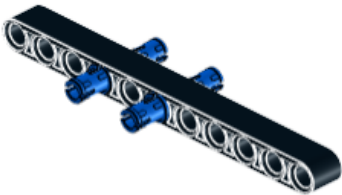
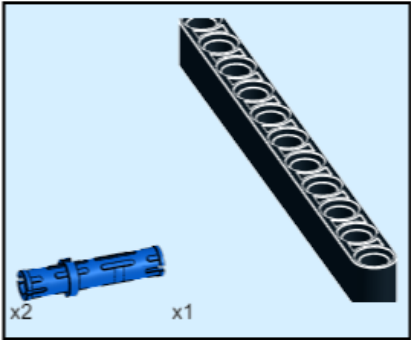


21

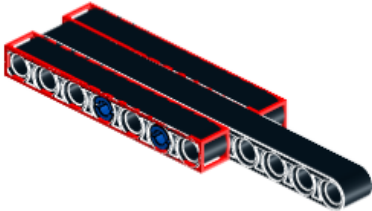


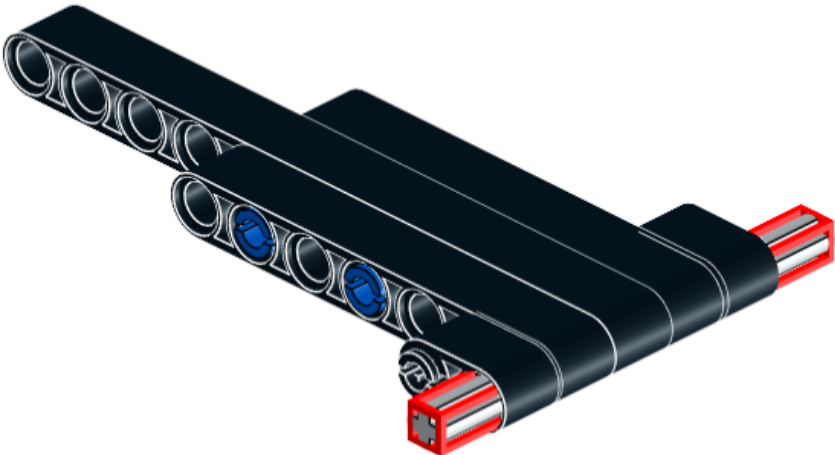
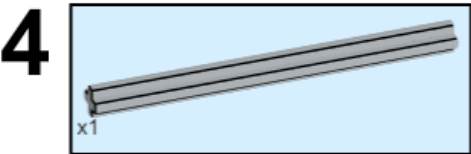
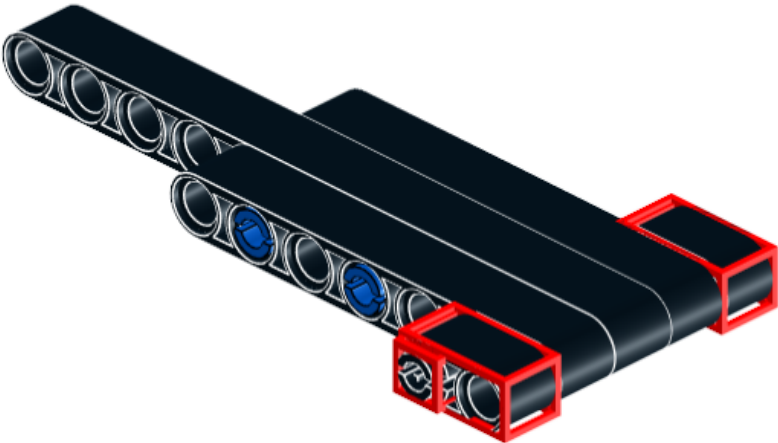
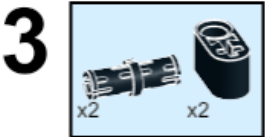
九、高中组通讯卫星

1

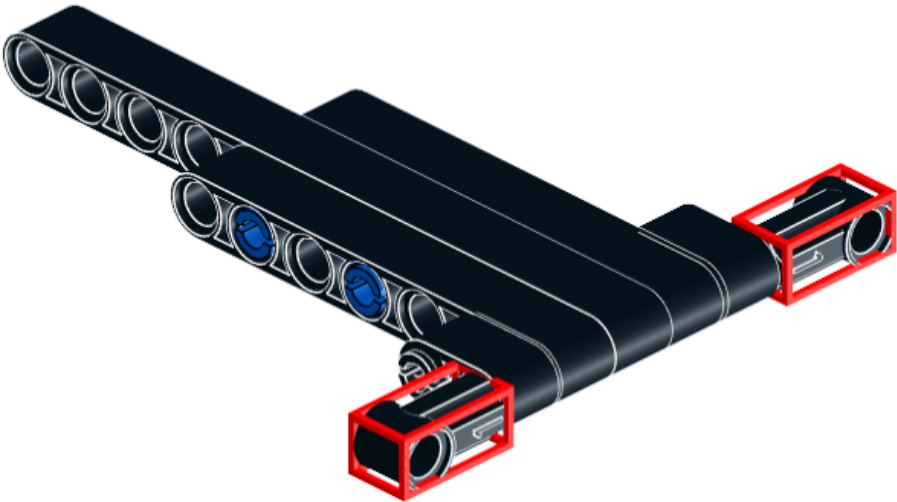


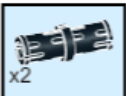
2

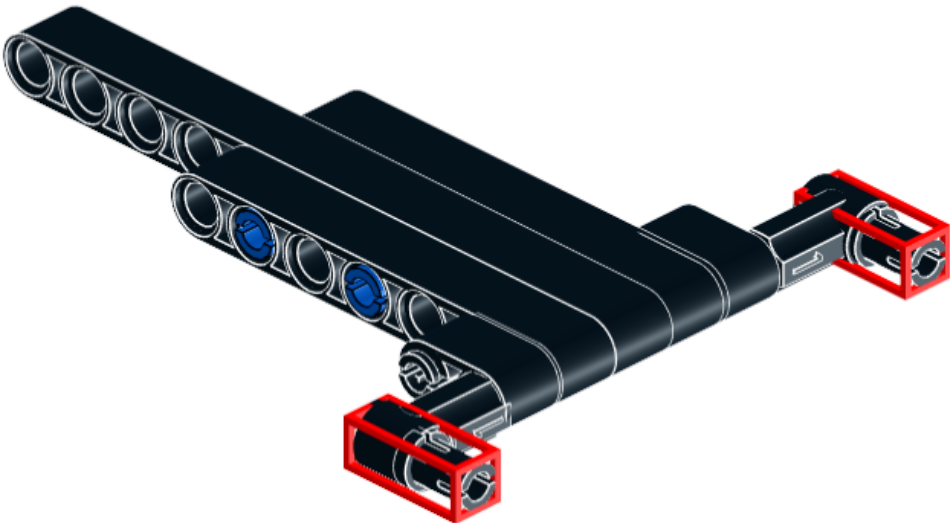


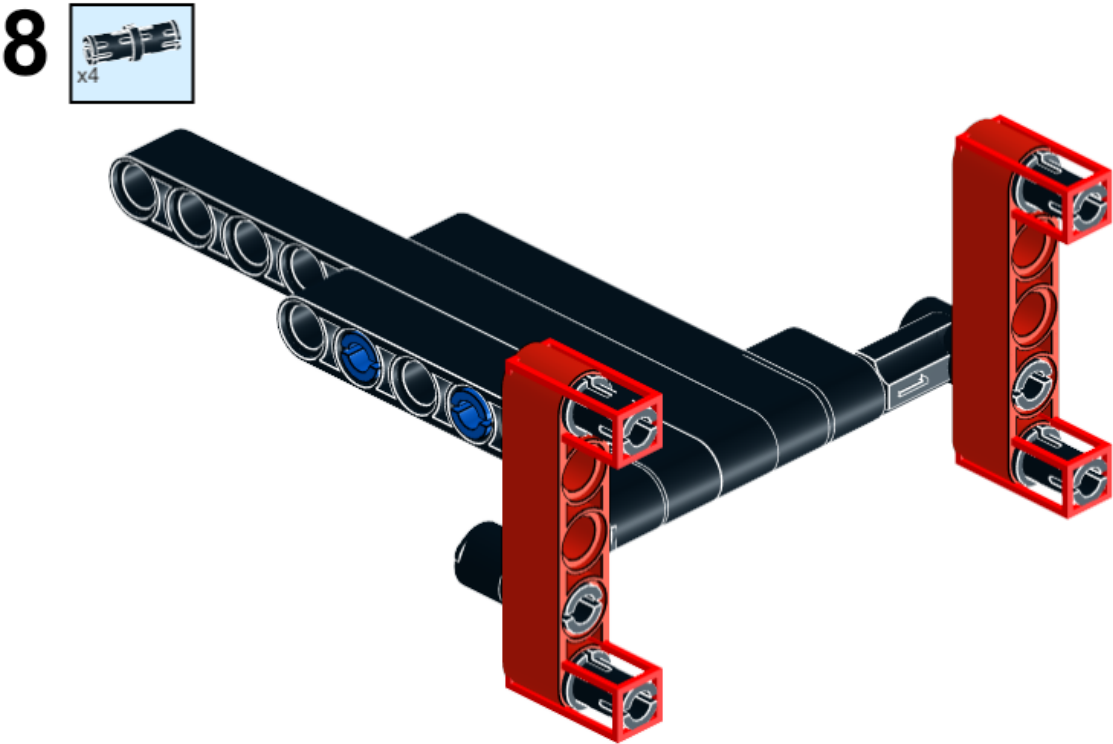
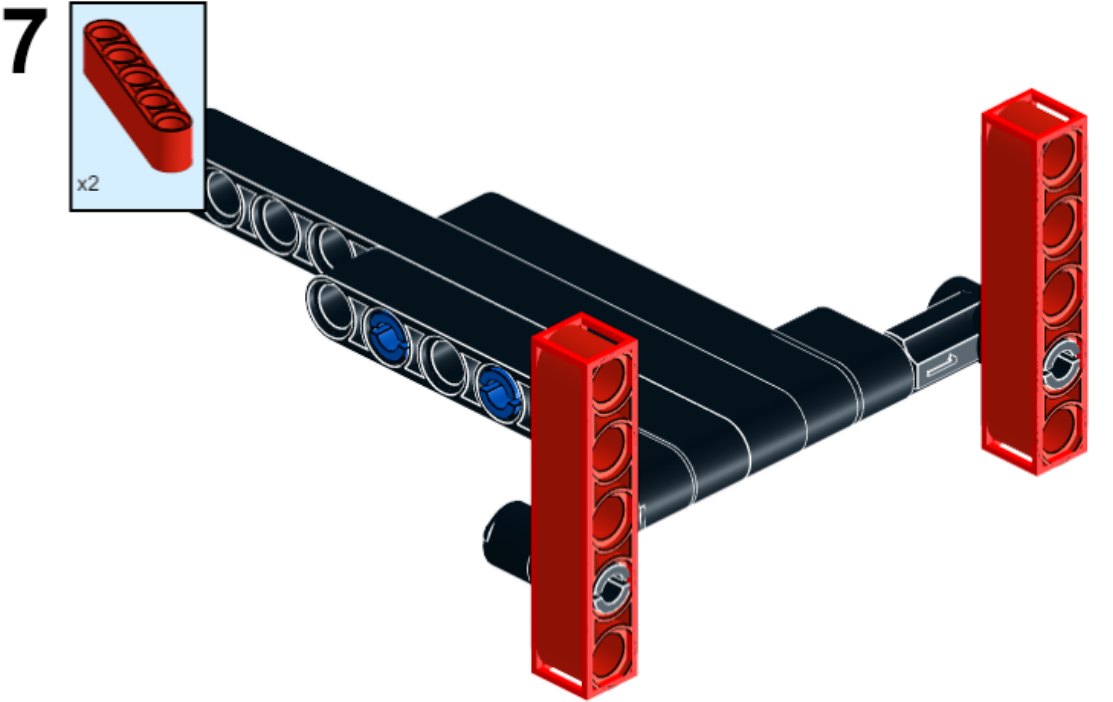


5 
x2

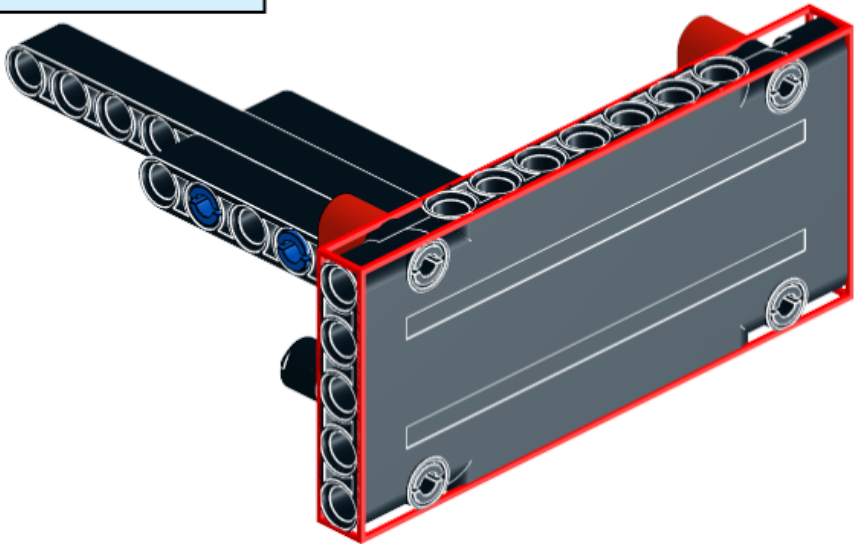
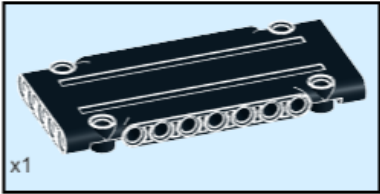


6 
x2

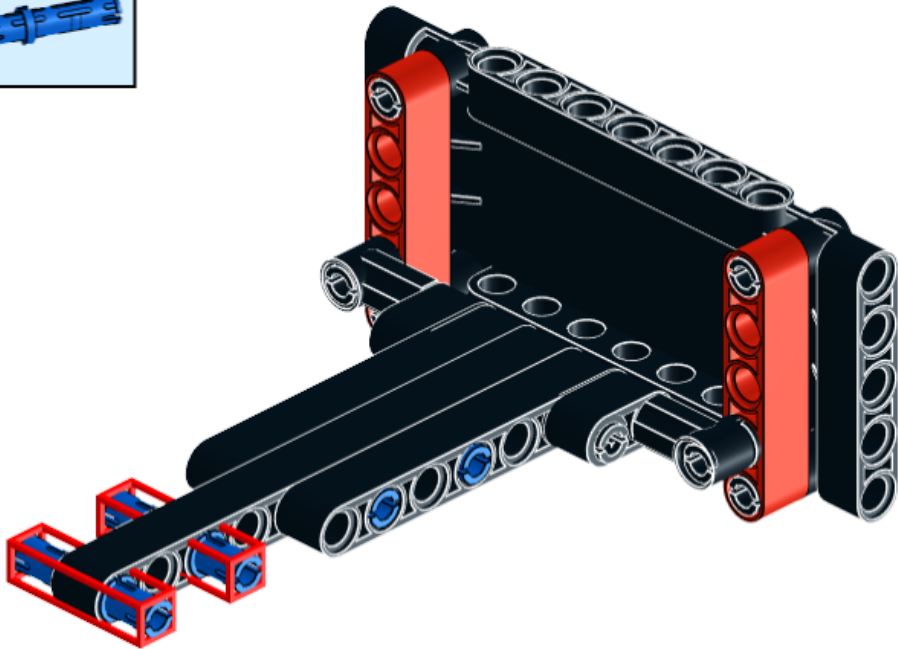
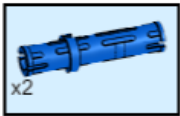




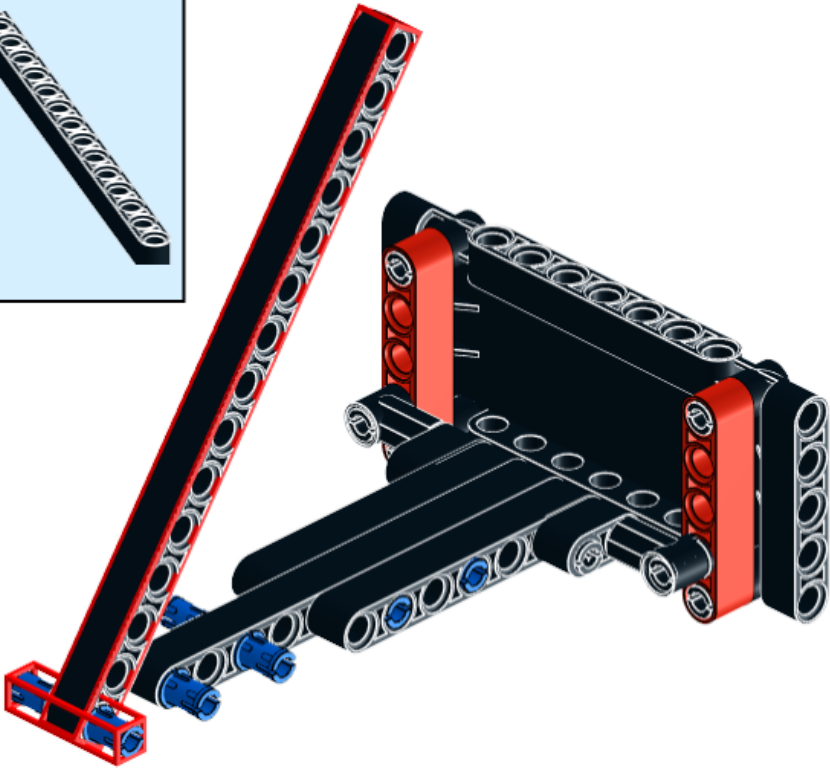
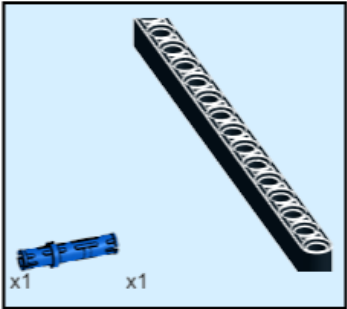
9



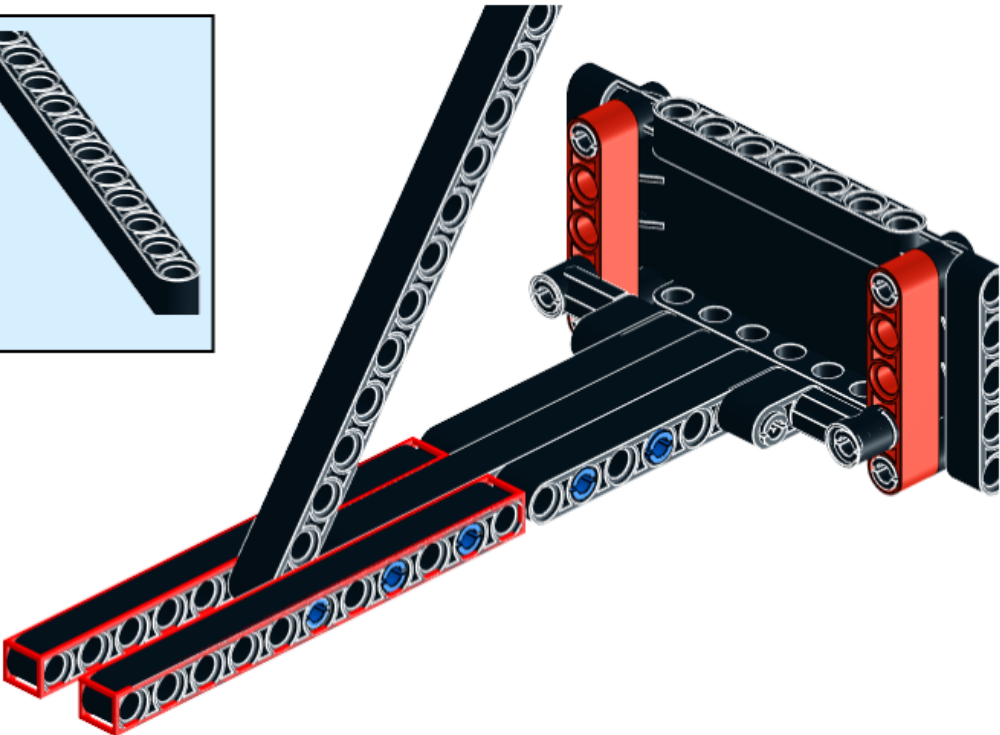
10

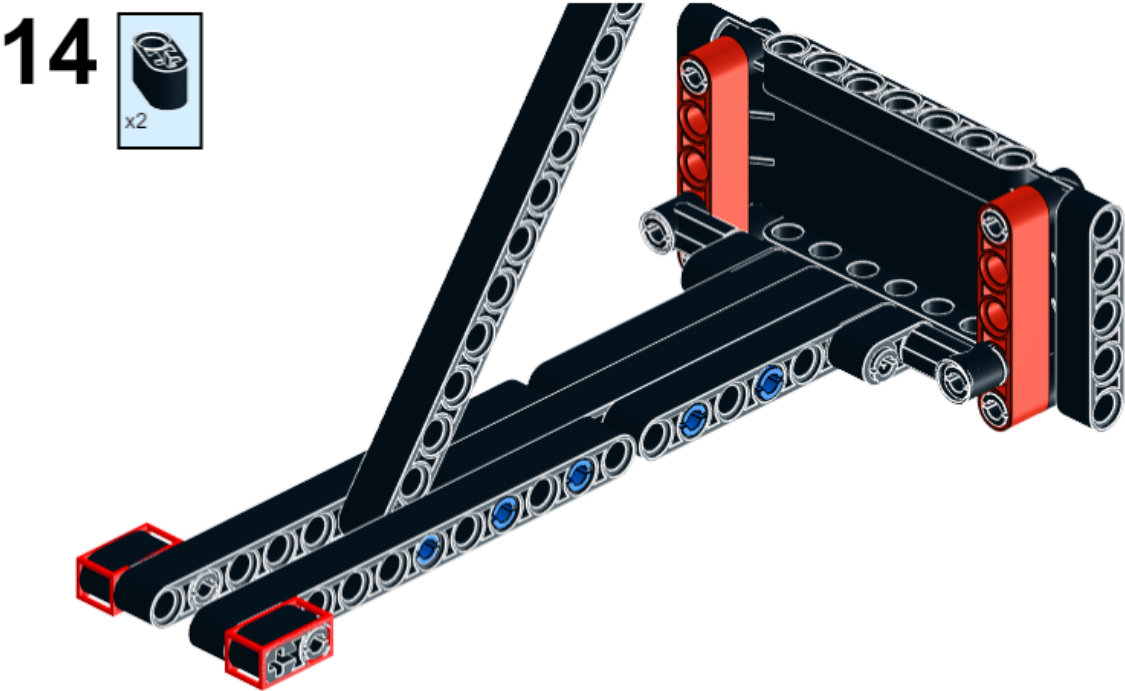
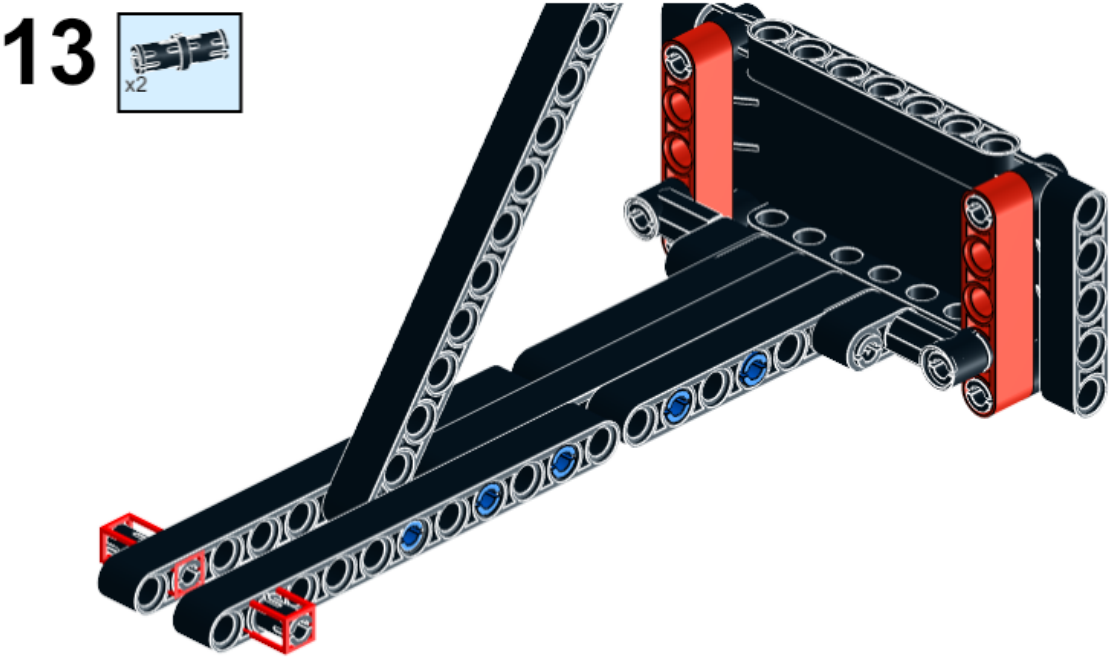


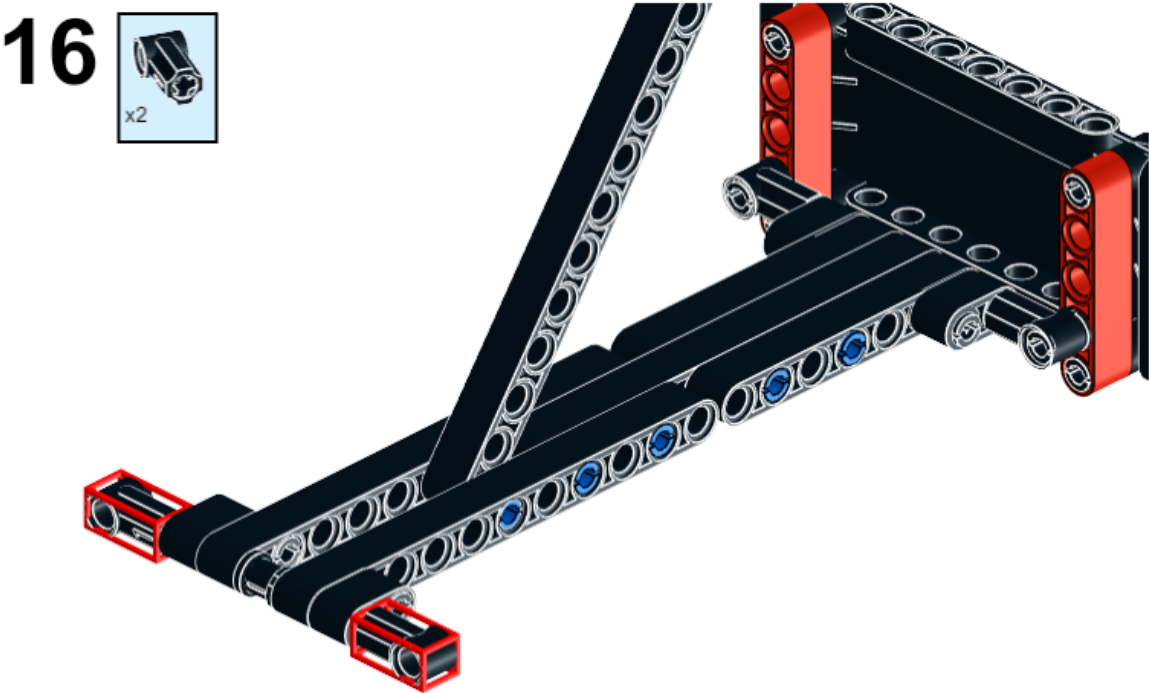
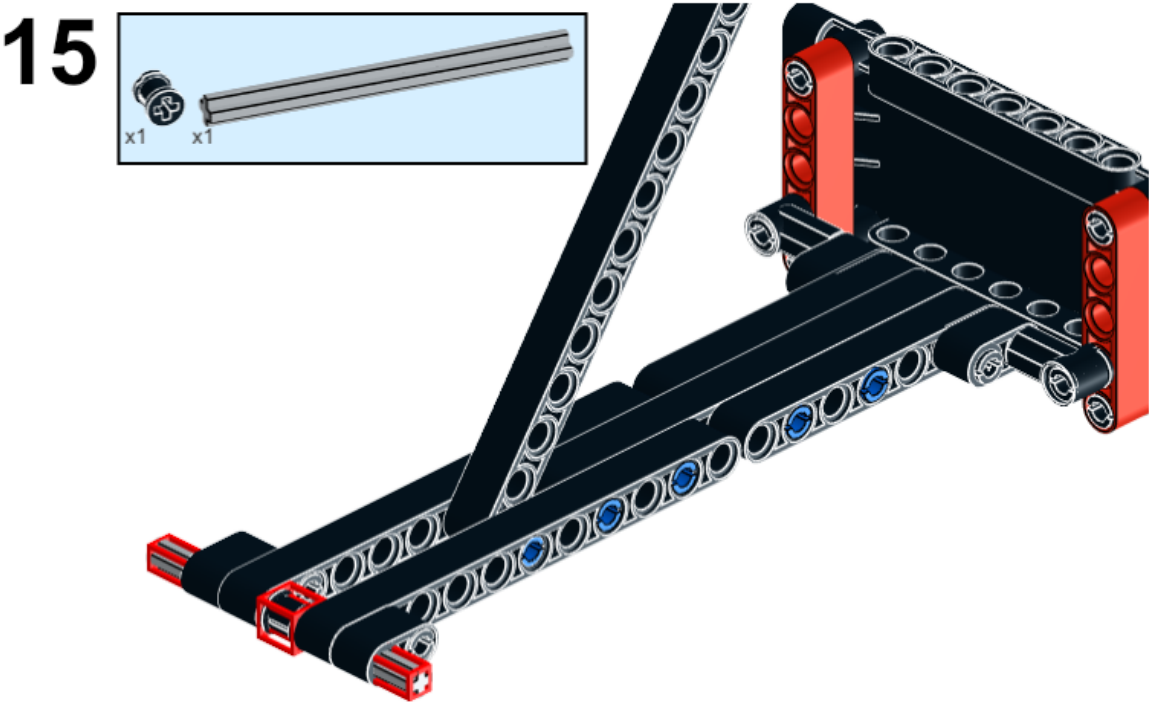
11



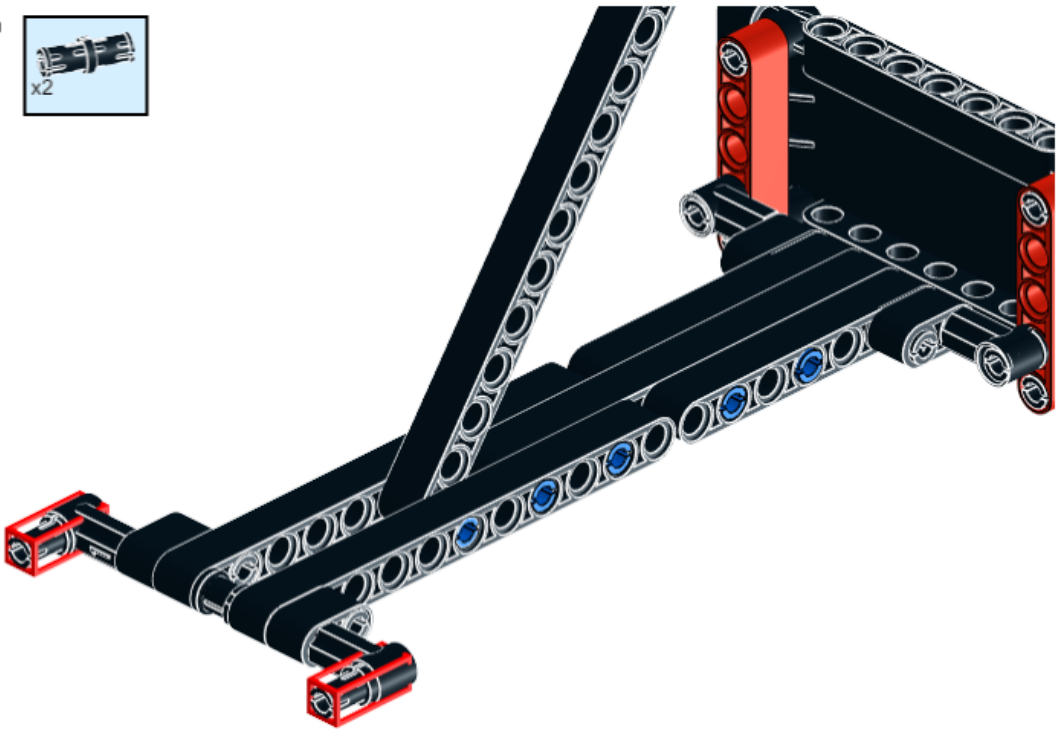
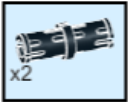
12



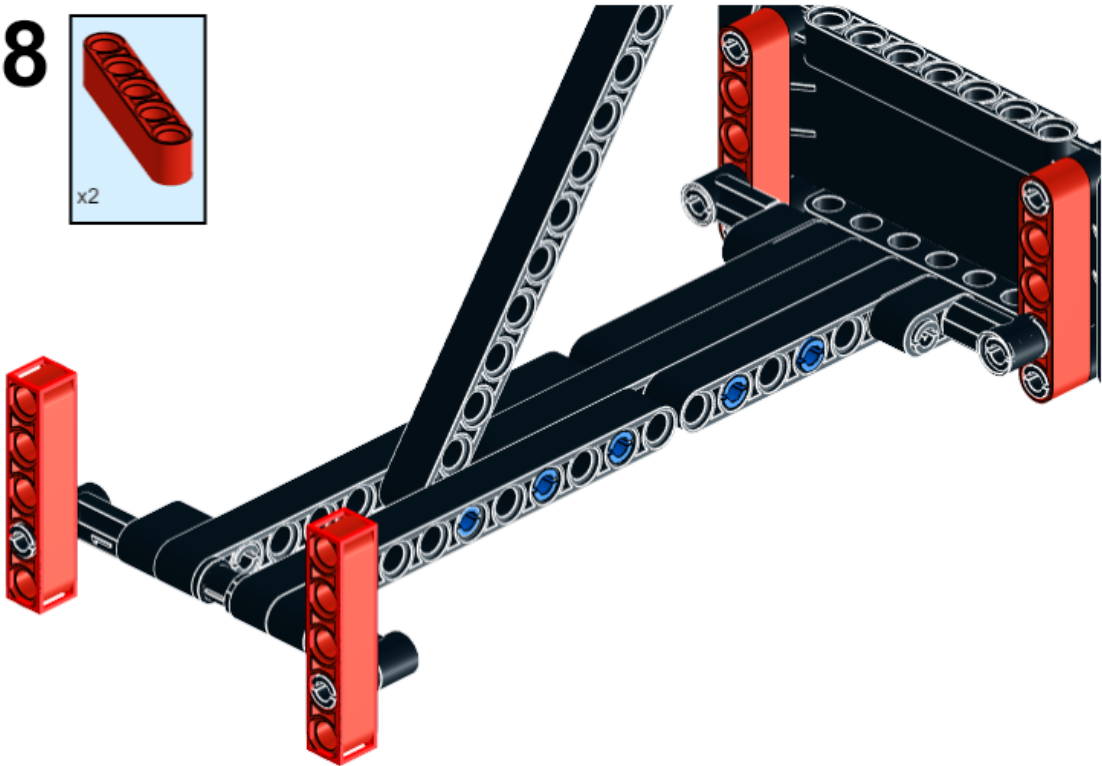
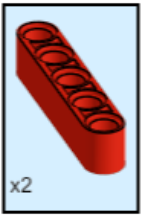


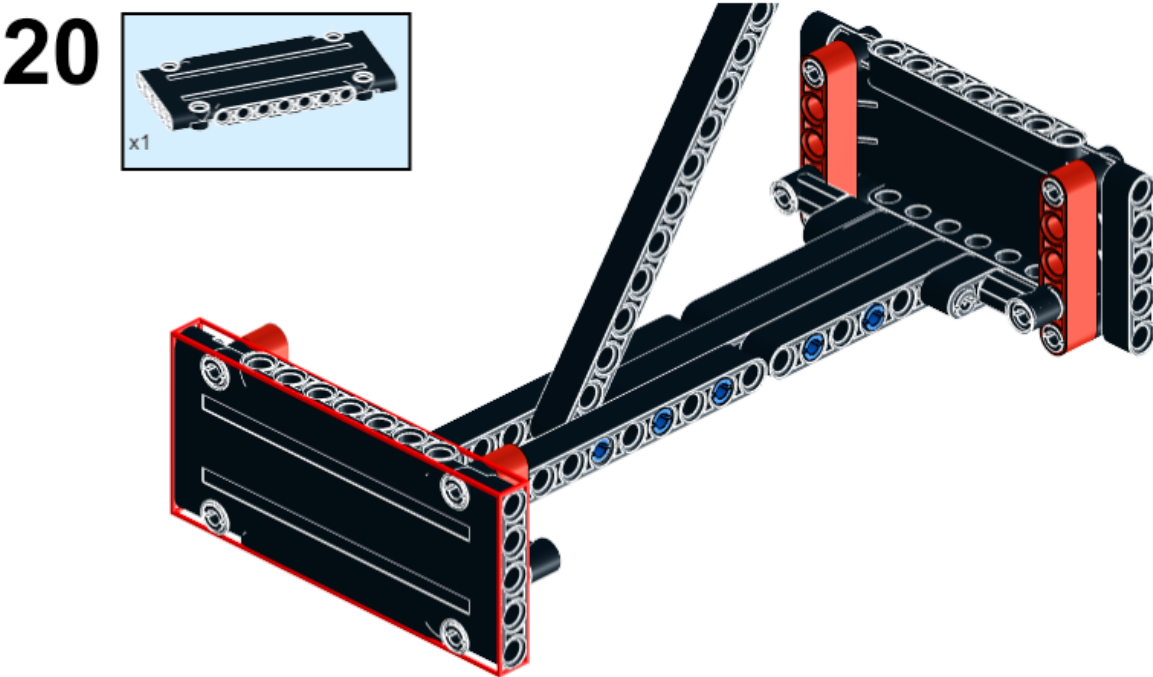
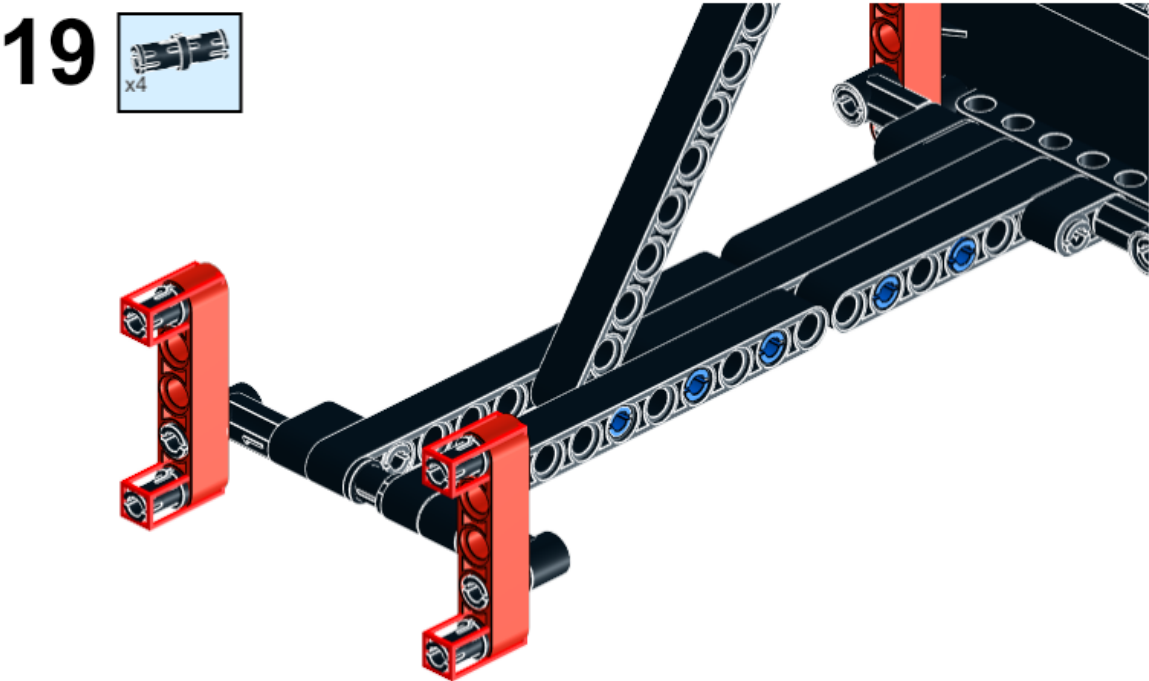


17

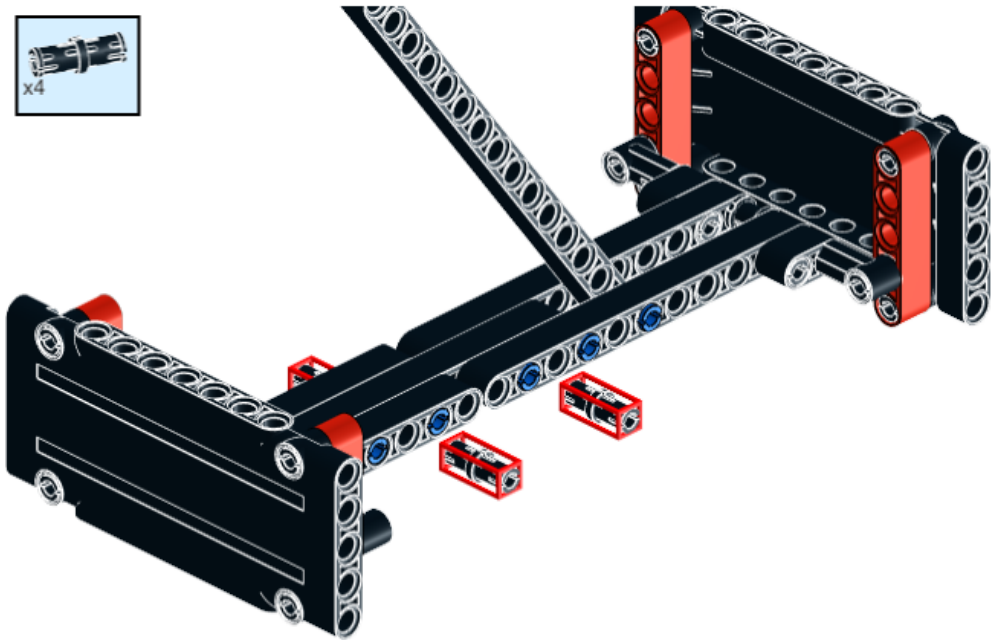
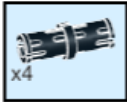


18

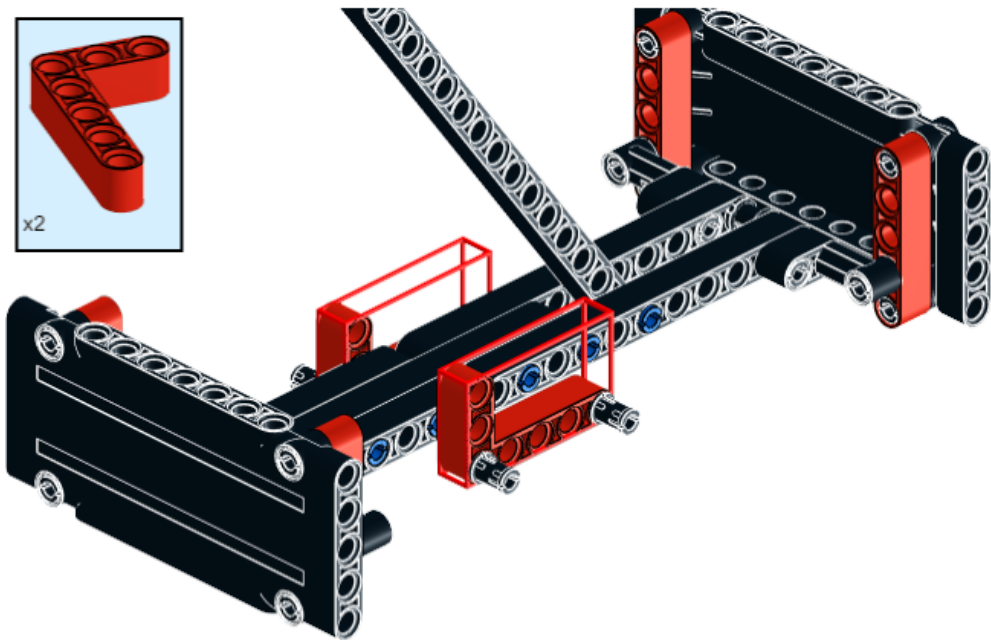
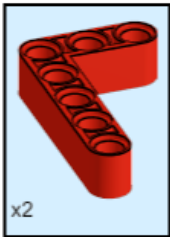


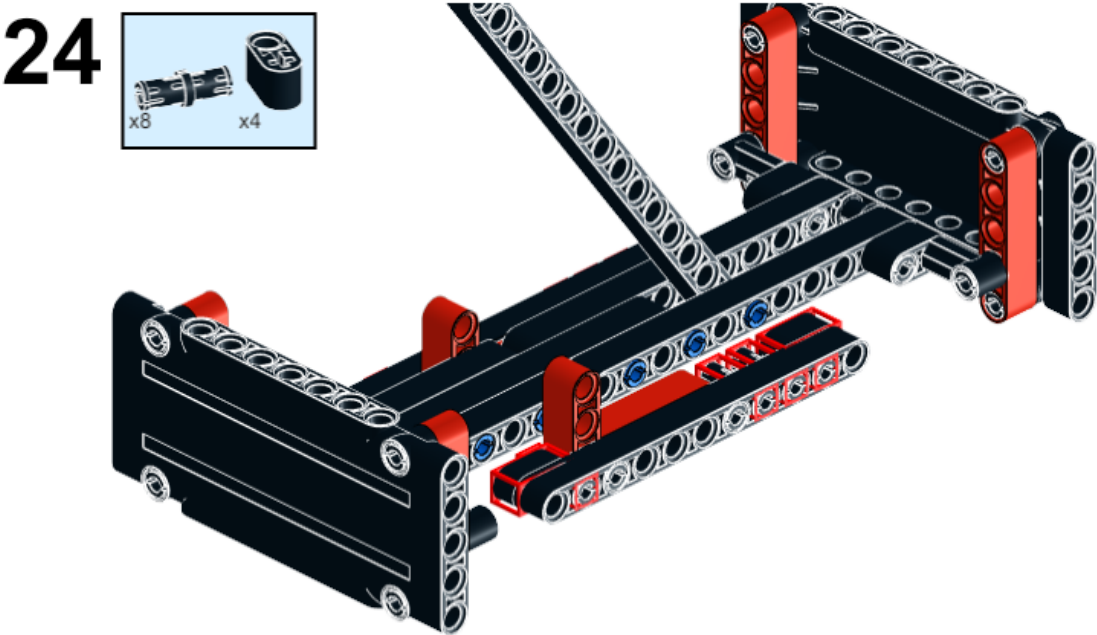
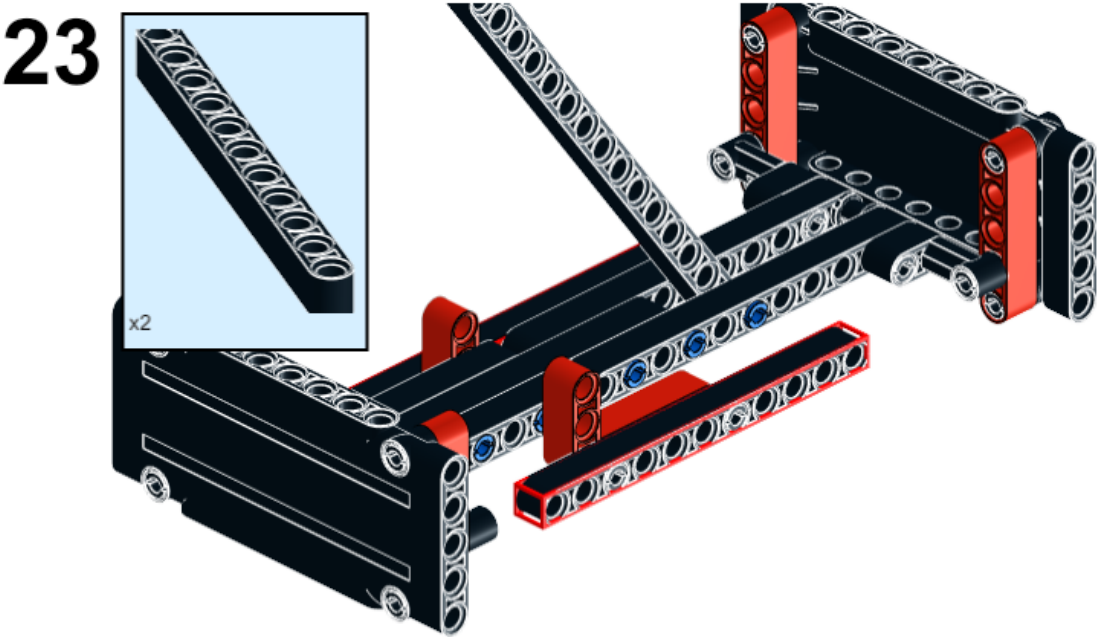


21

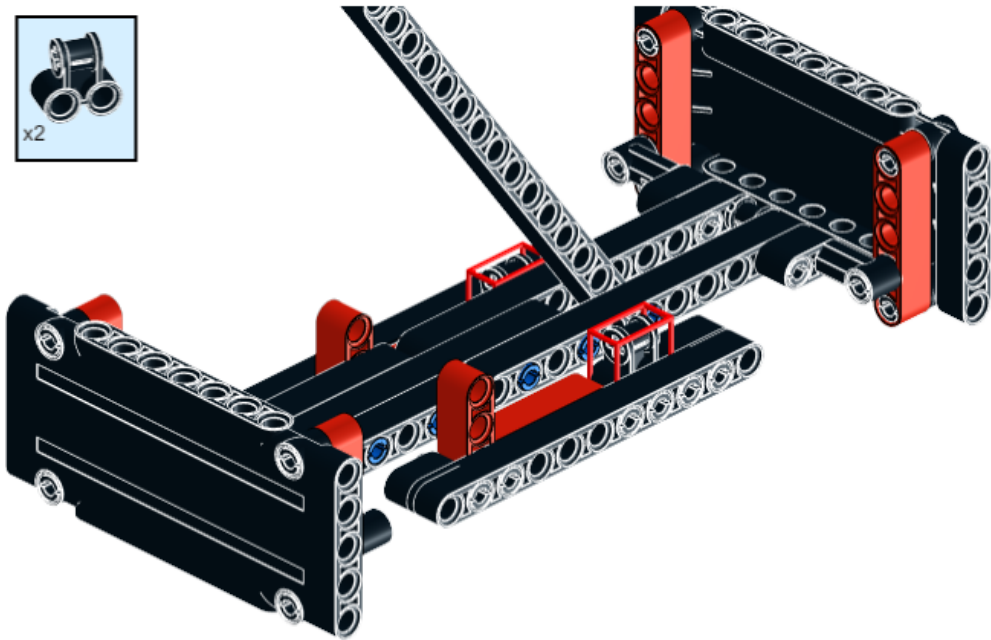


22

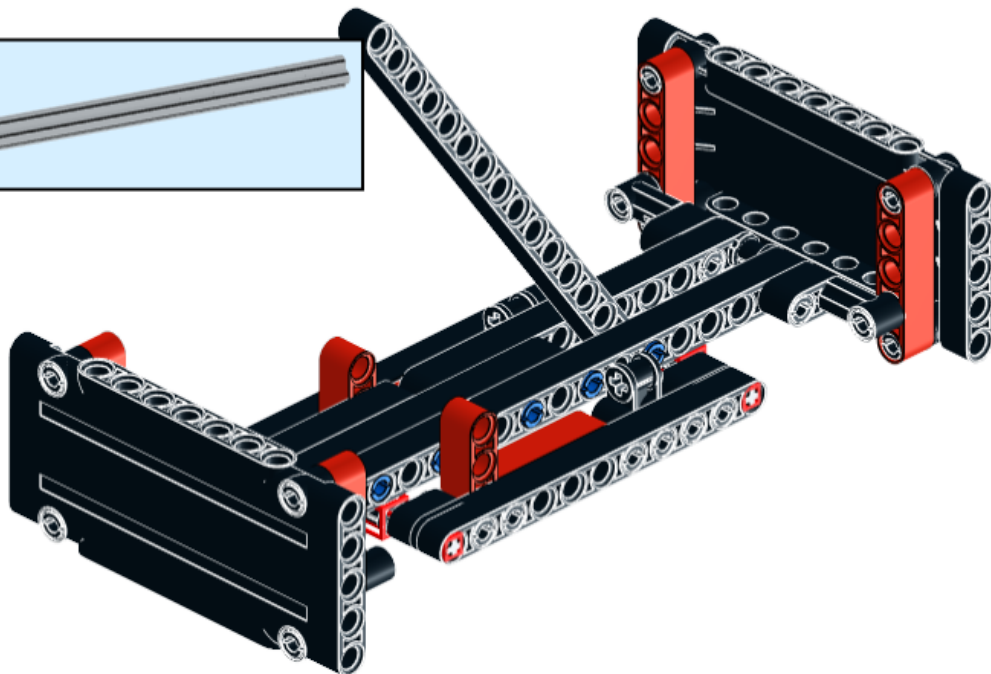
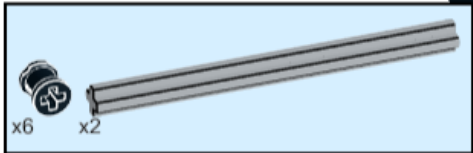




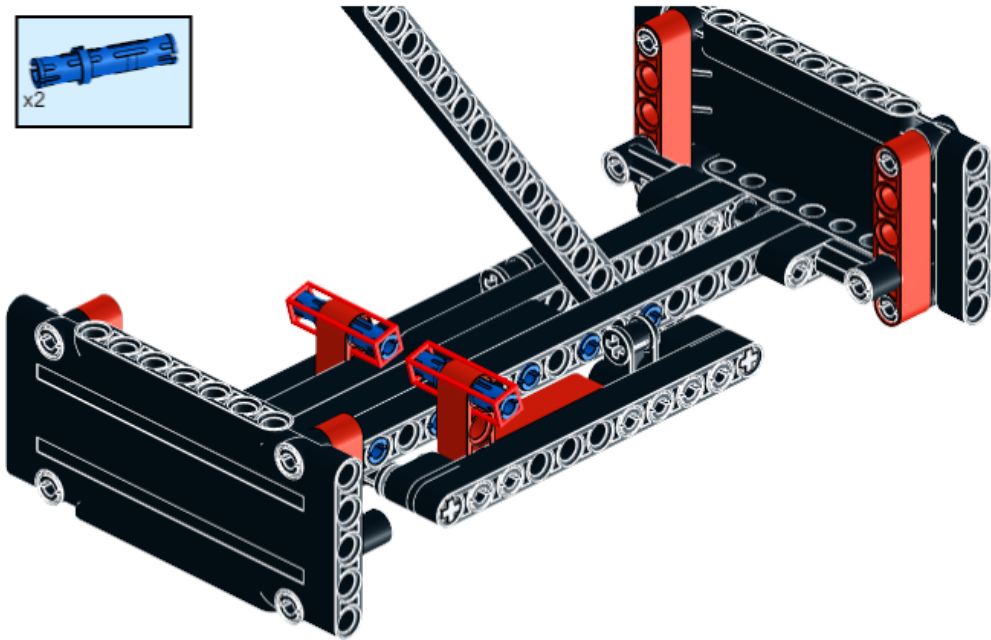
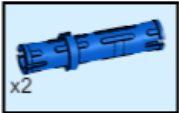
25



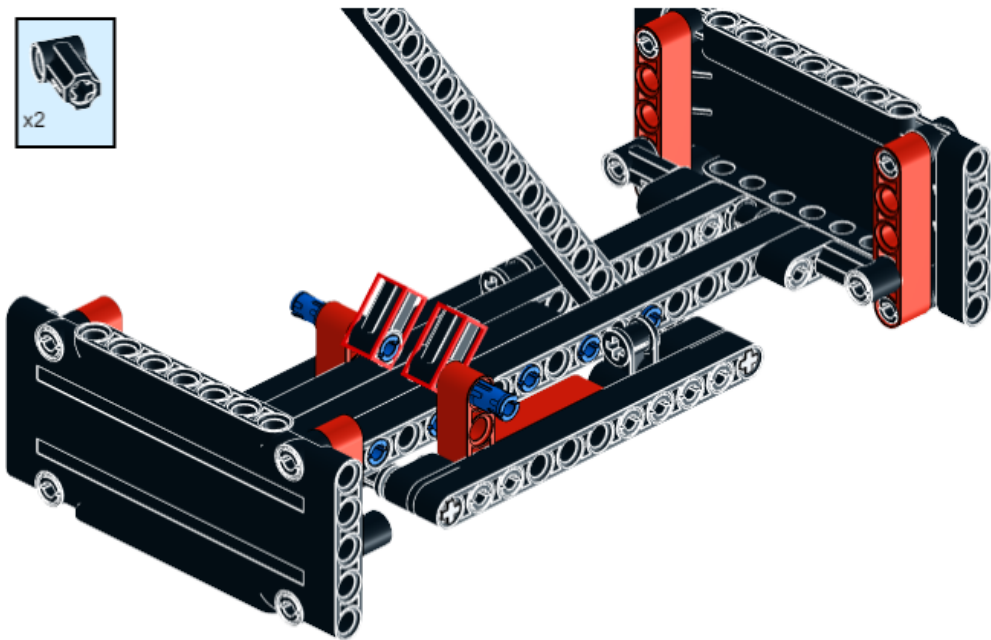
26



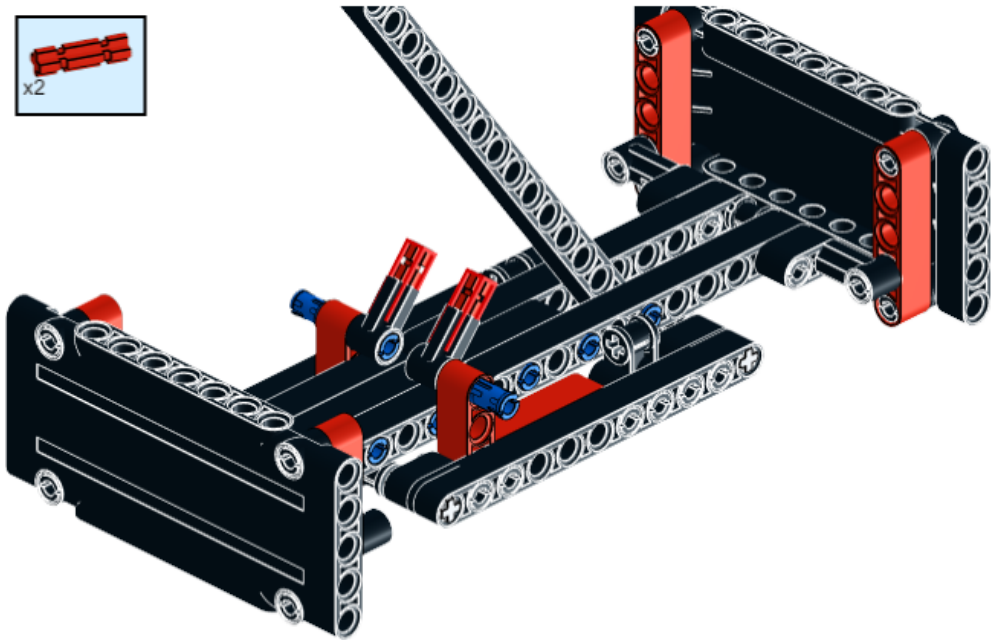
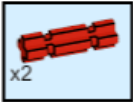
27



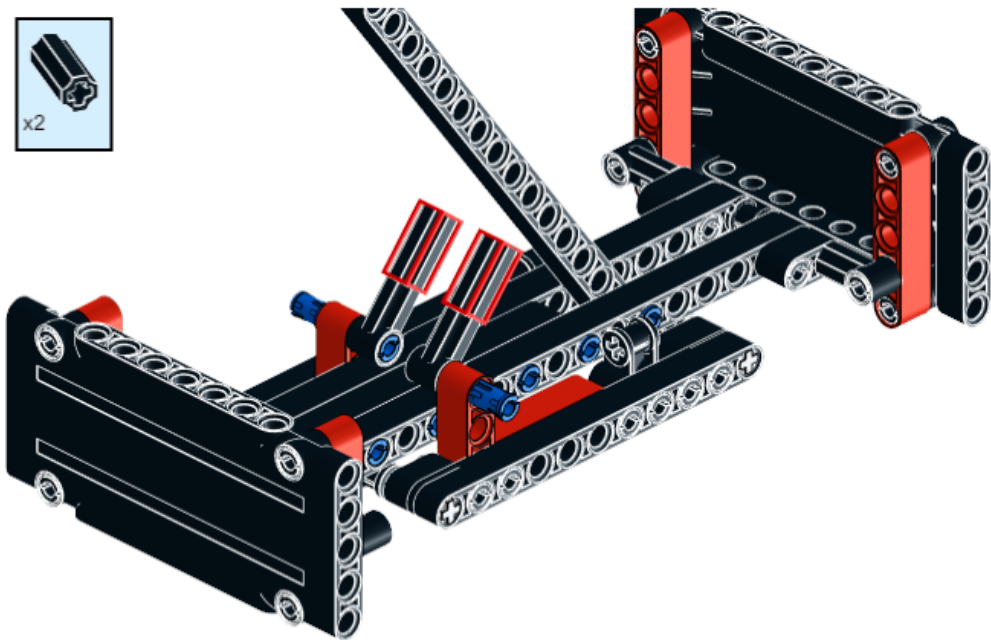
28



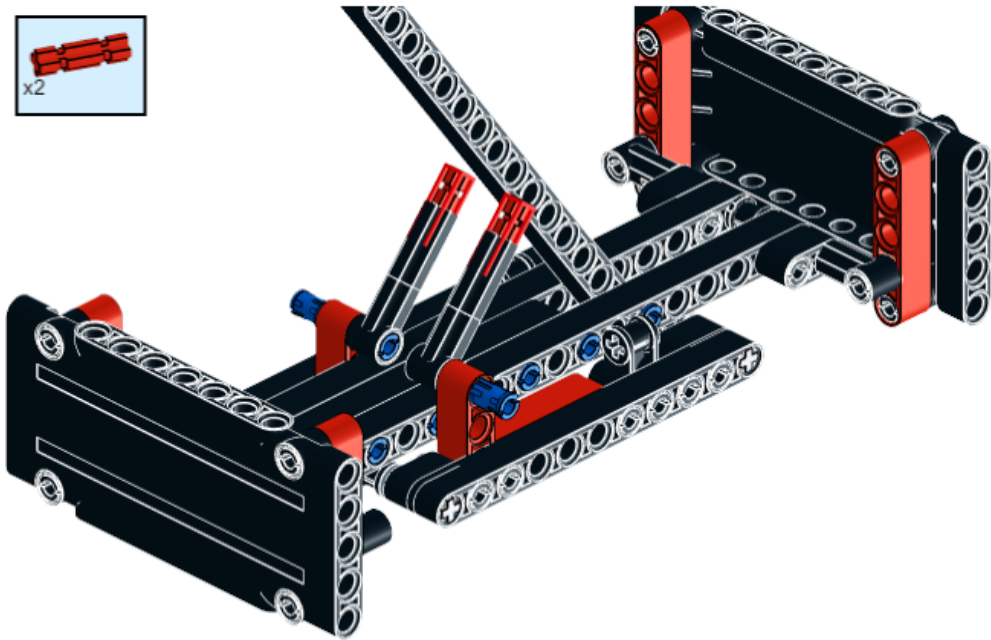
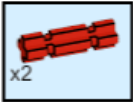
29



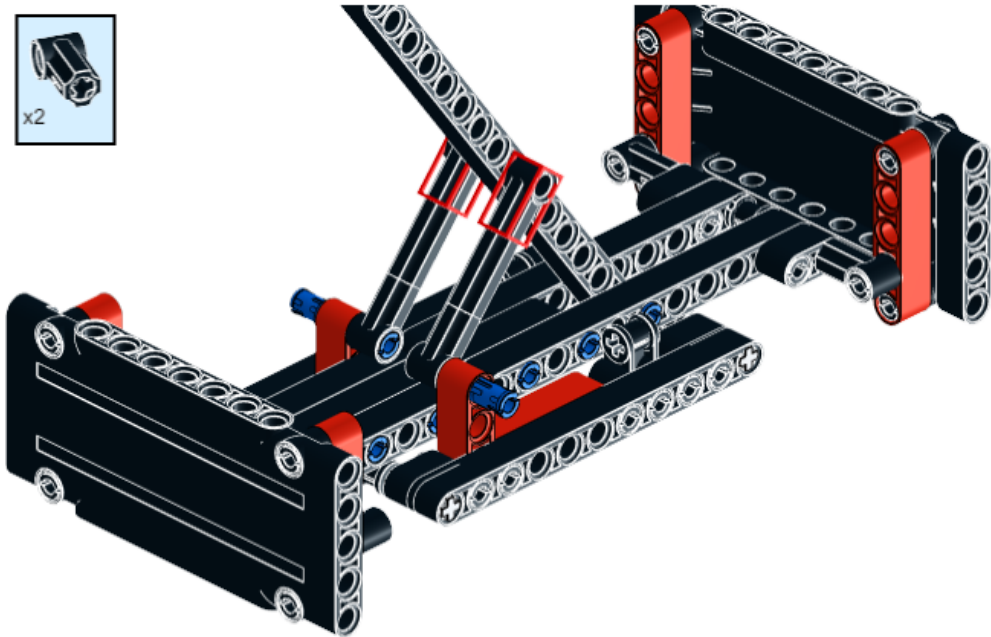
30



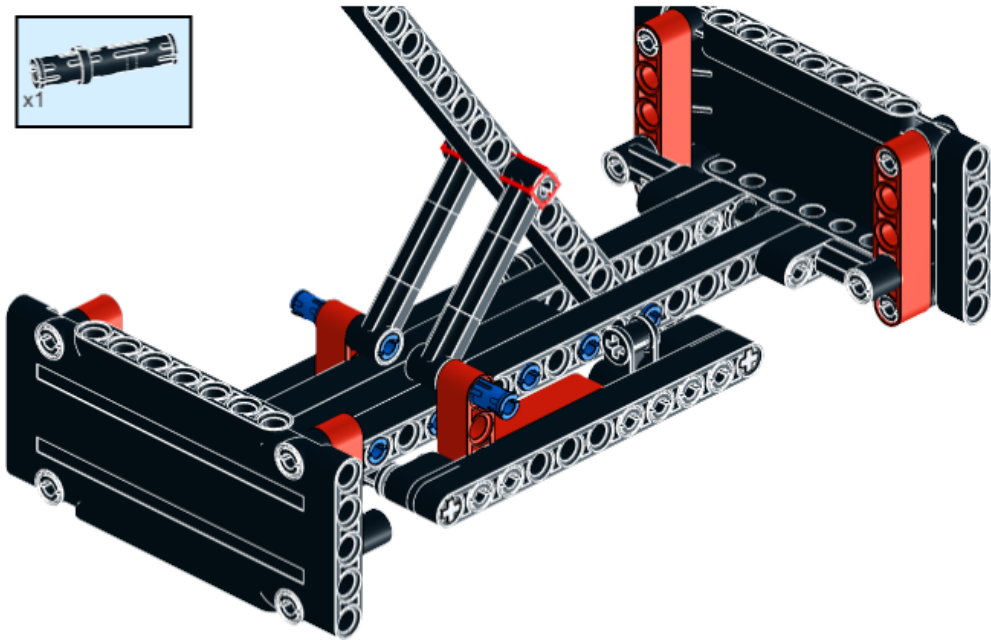
31



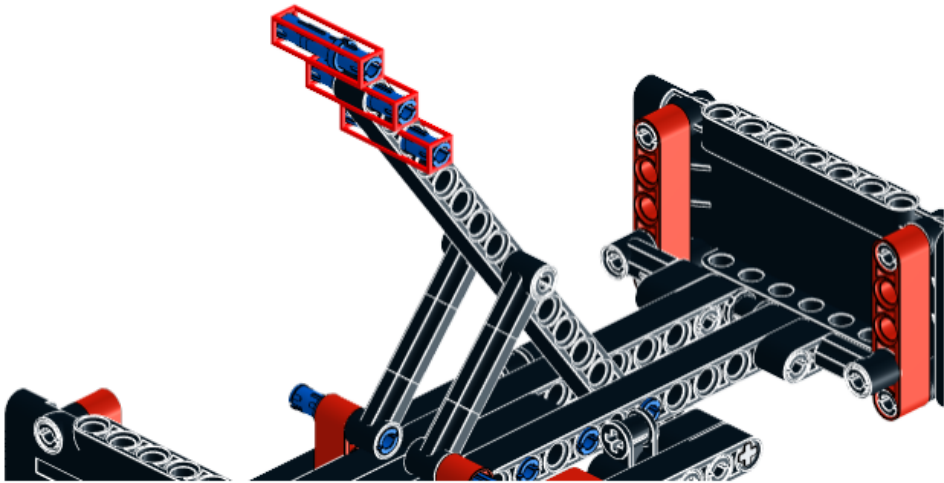
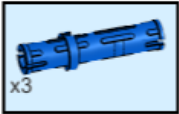
32



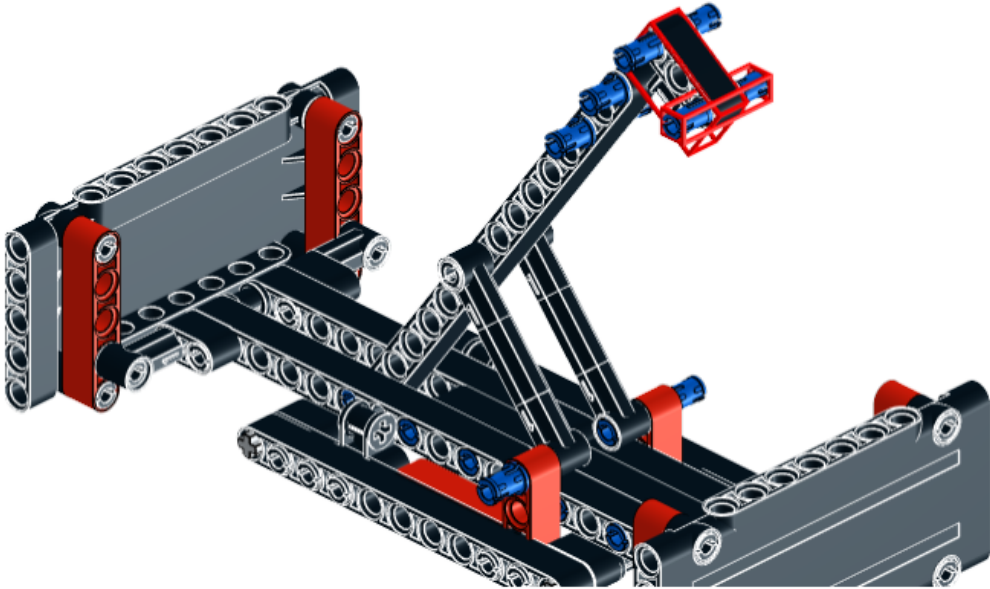
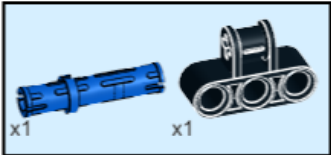
33



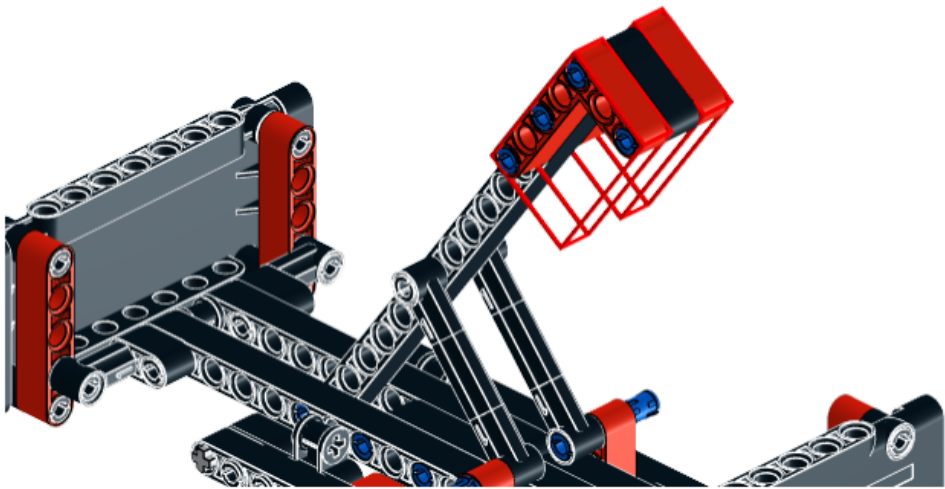
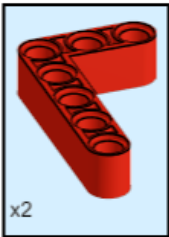
34



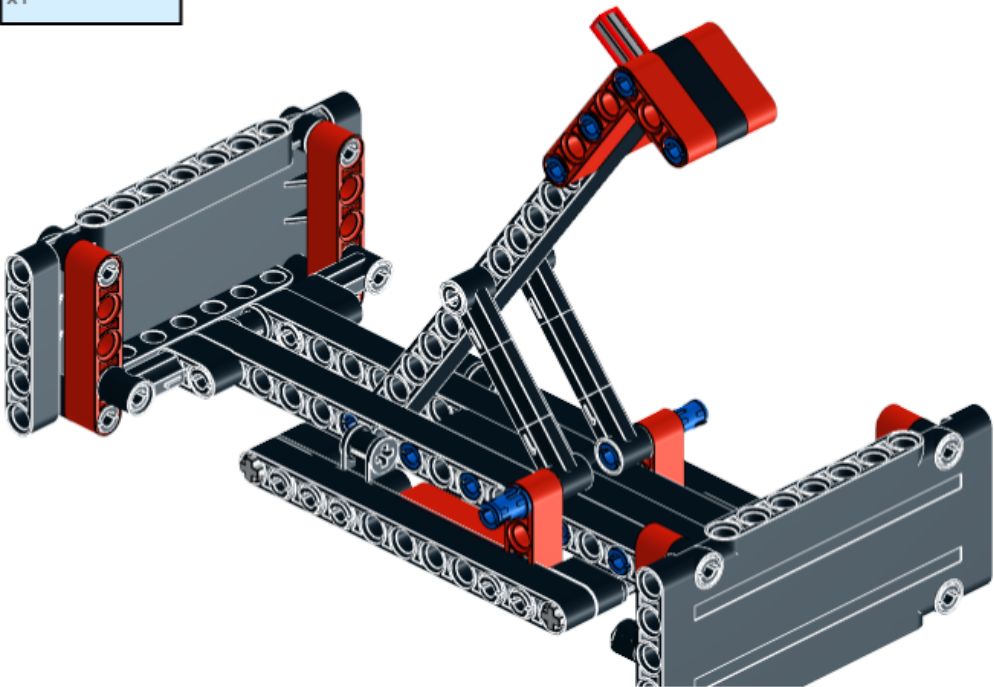
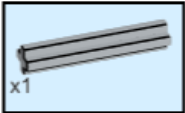
35



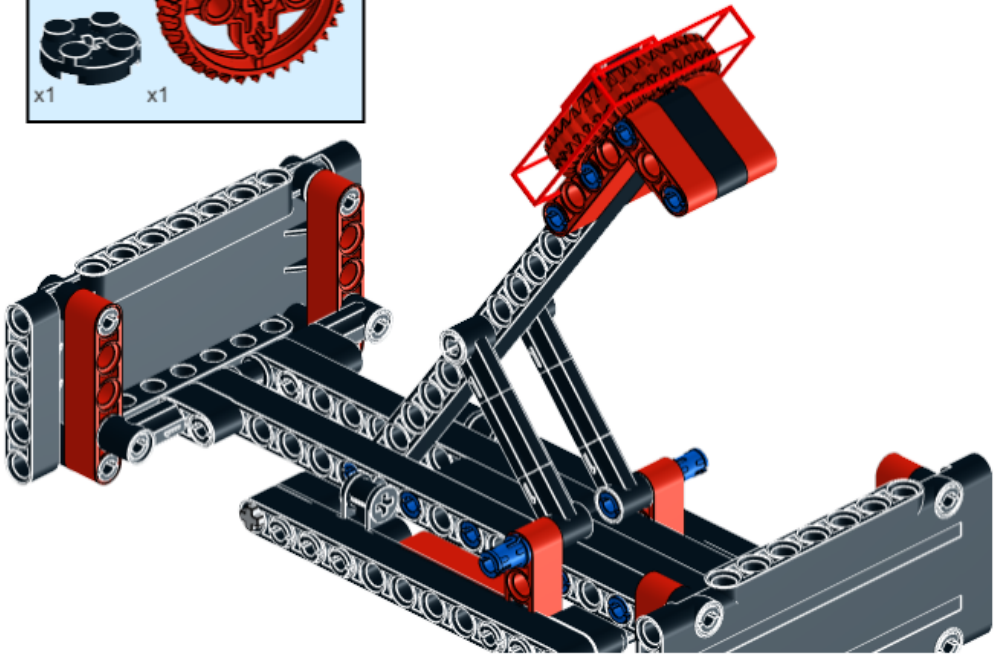
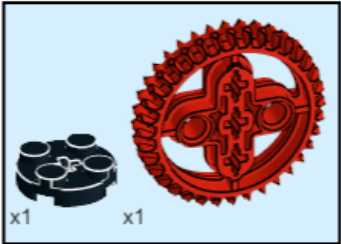
36



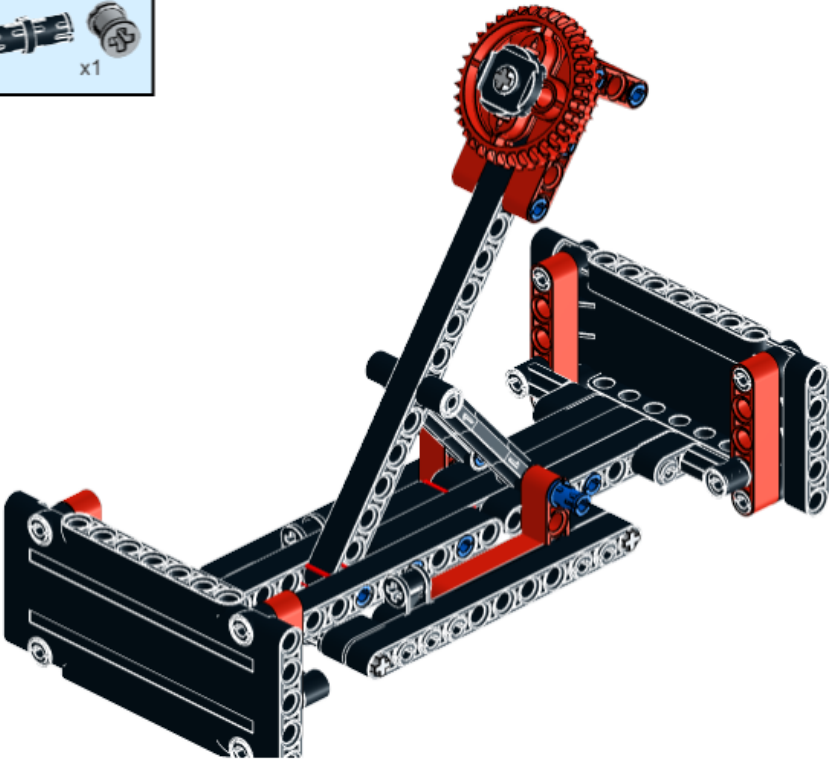
37



38

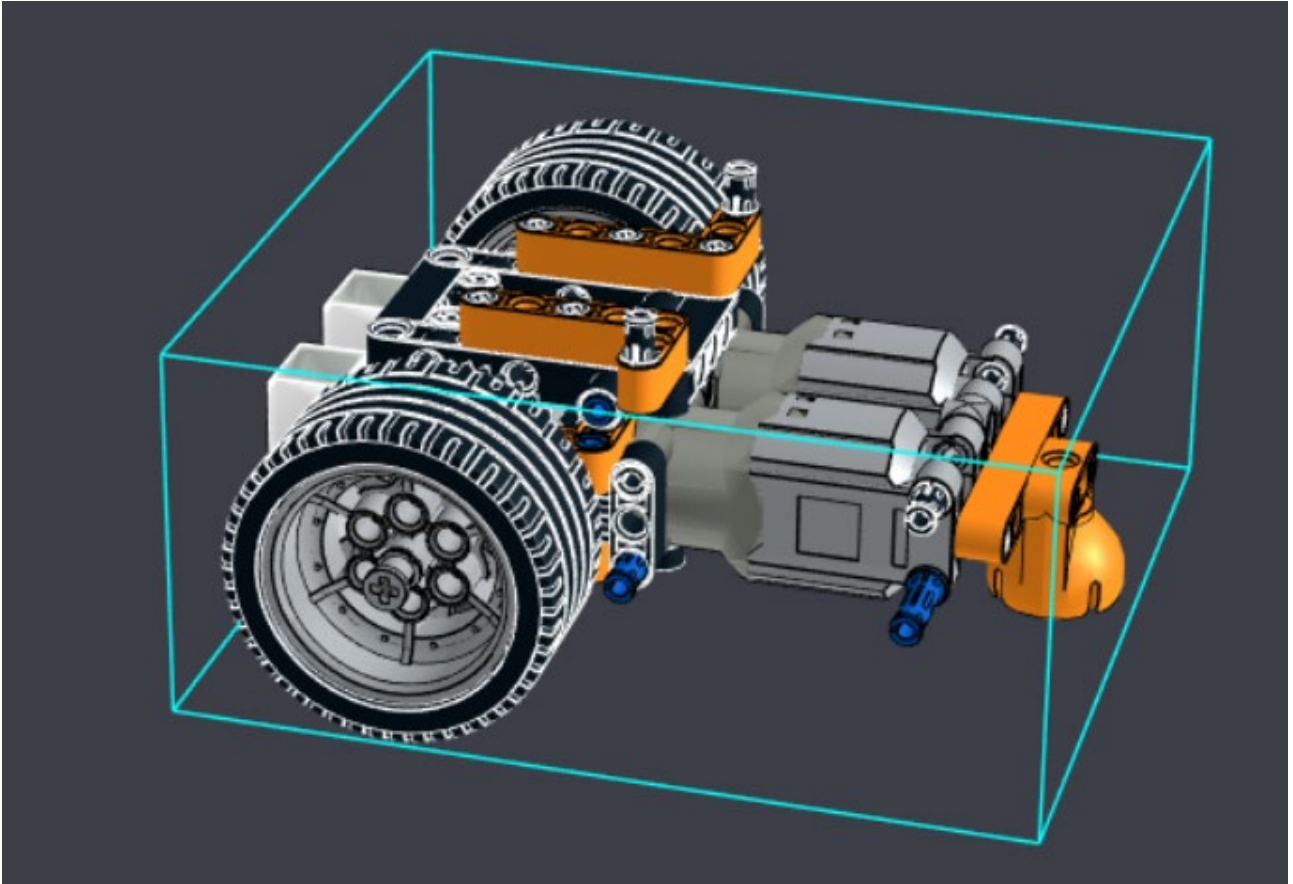


39

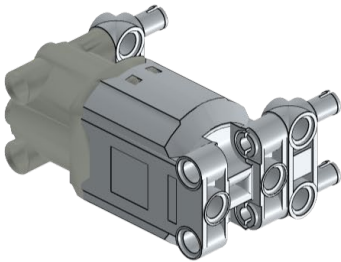
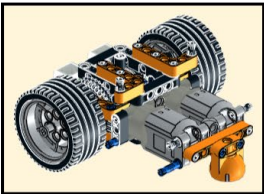
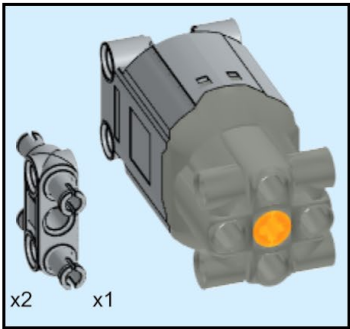


第二课 赛事车体推荐搭建指南

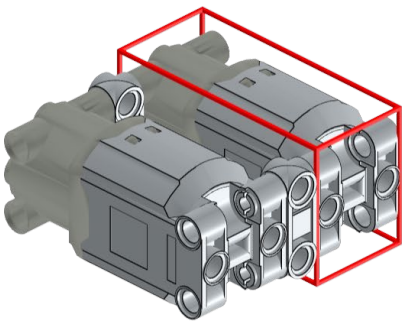
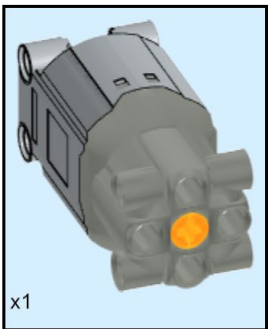
一、车的基础载体



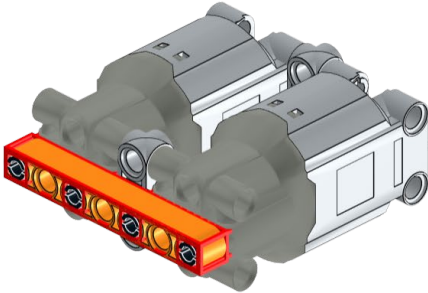
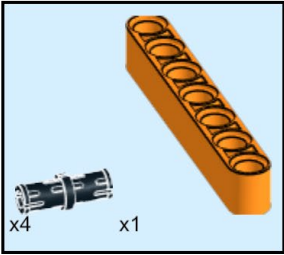
1



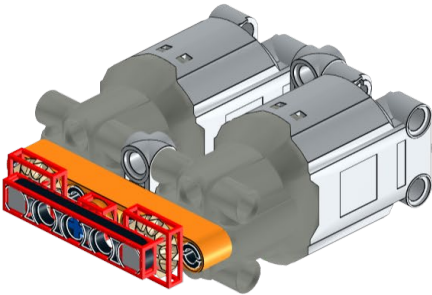
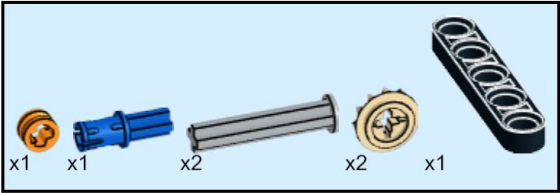
2



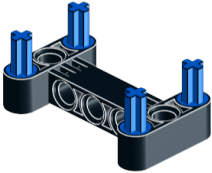
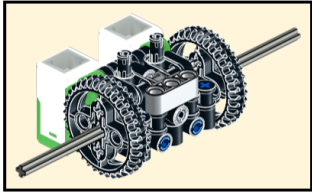
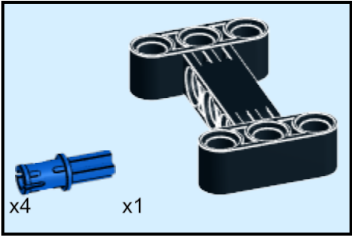
3



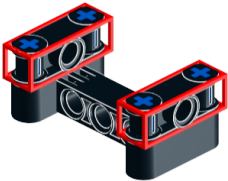
4

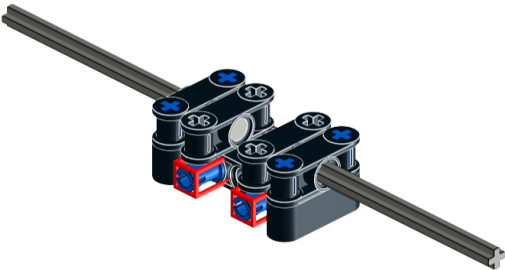
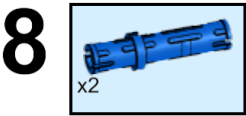
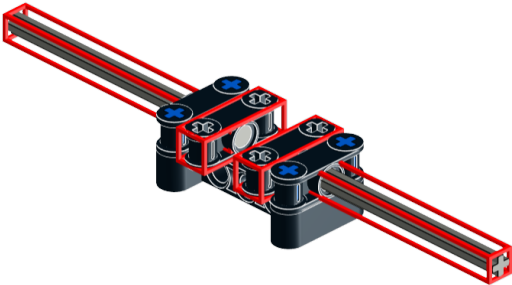
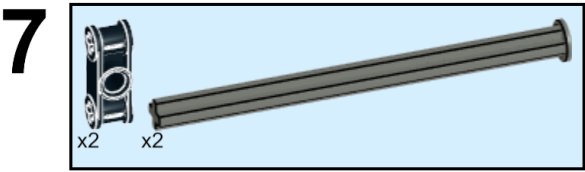


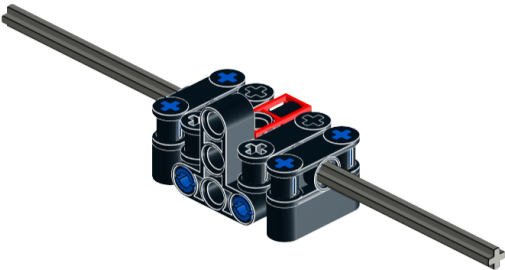
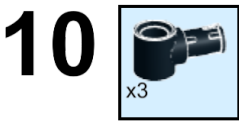
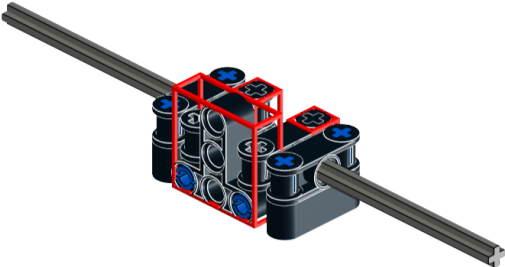
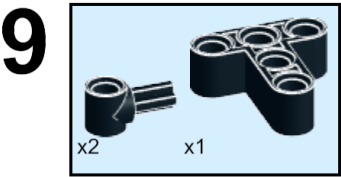
5



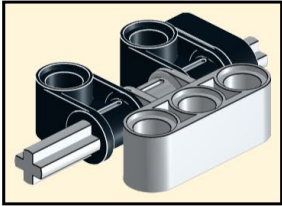
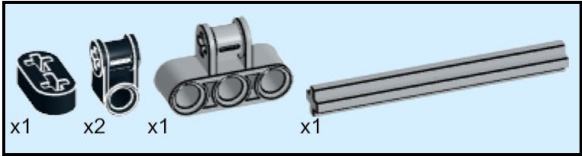
6



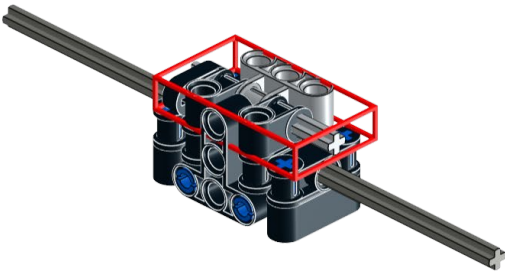




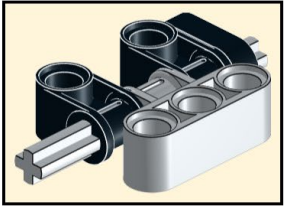
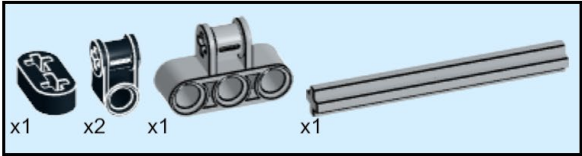
11



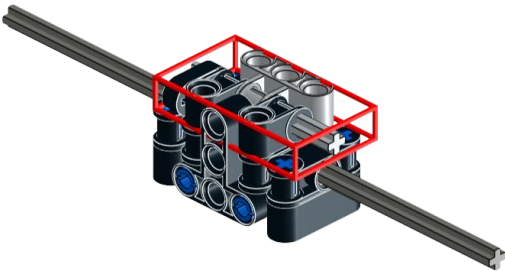
12



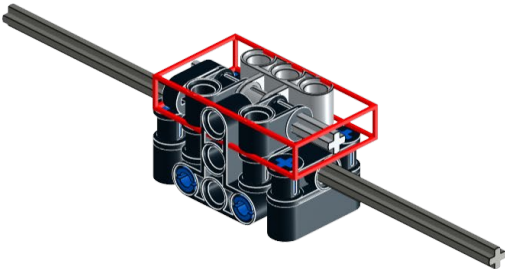
11



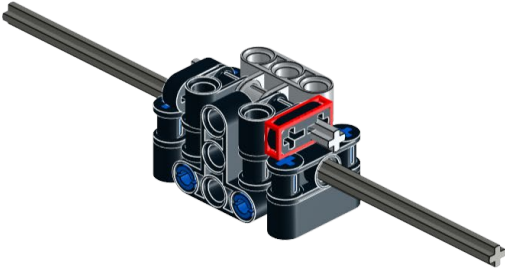
12



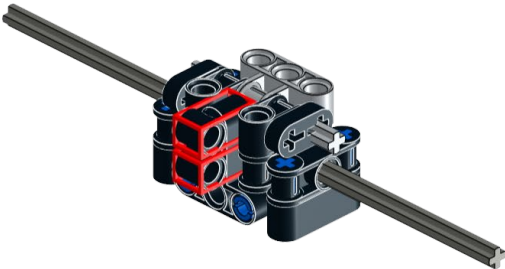
12



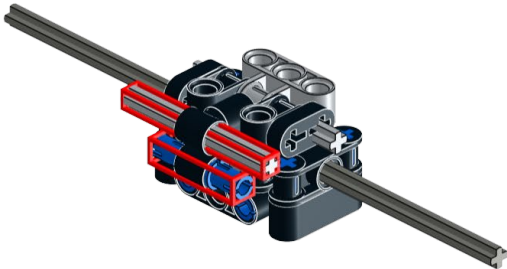
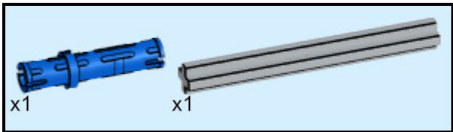
13



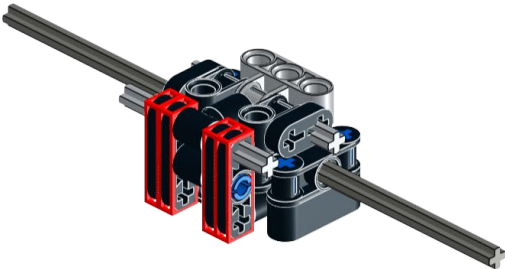
14



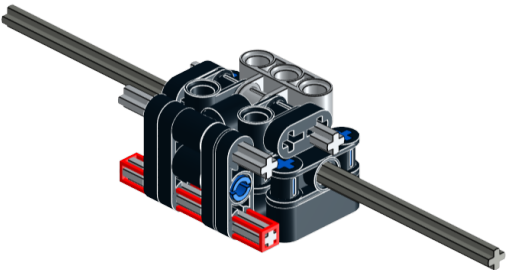
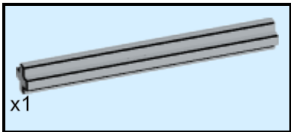
15



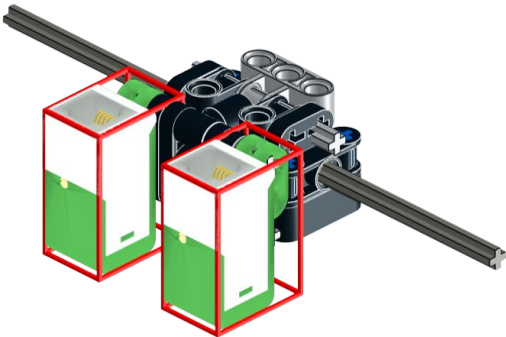
16



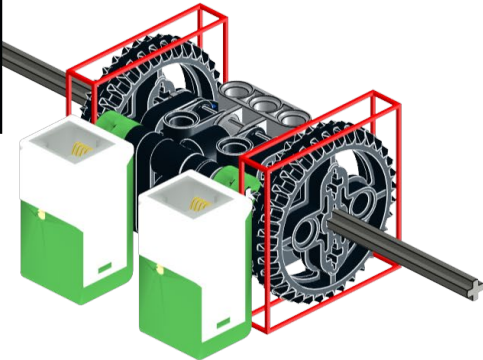
17

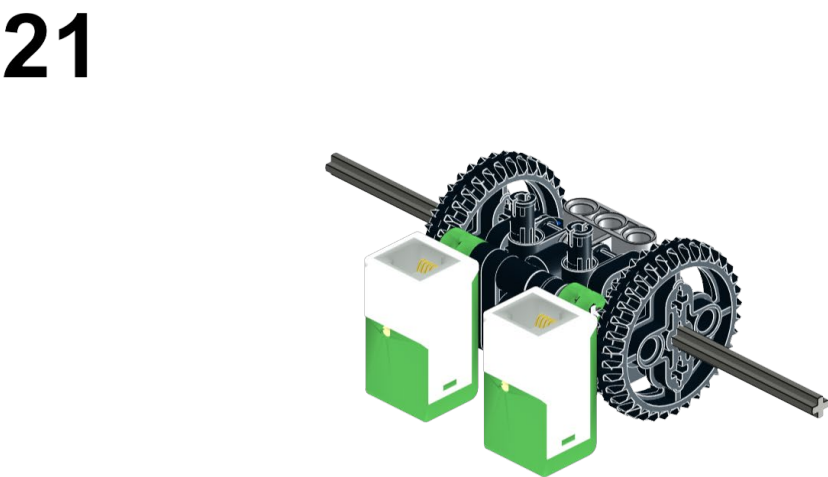
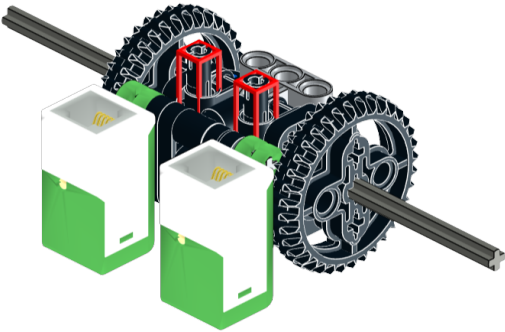
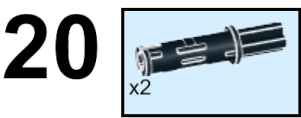


18

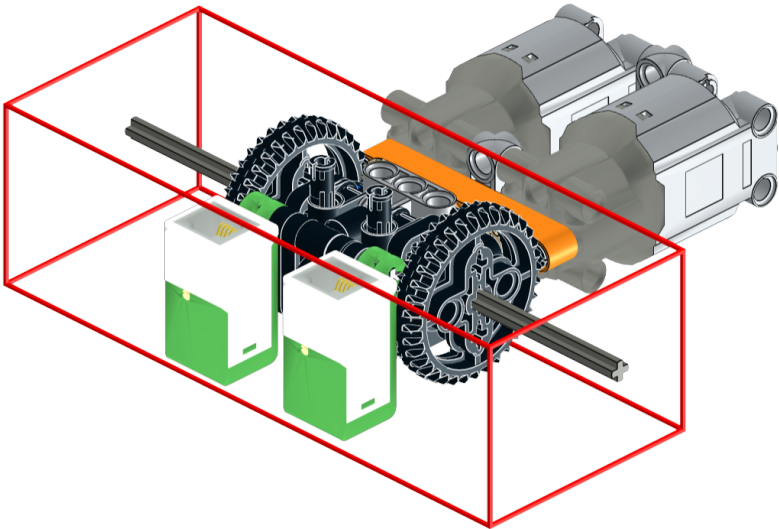


19

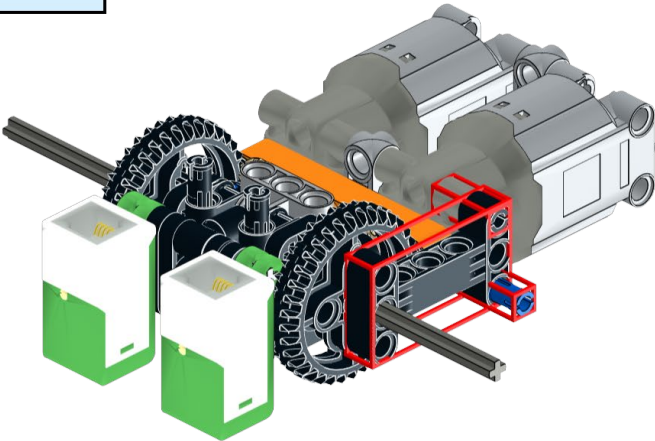
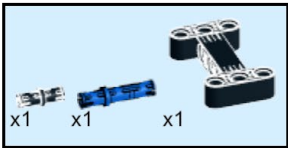




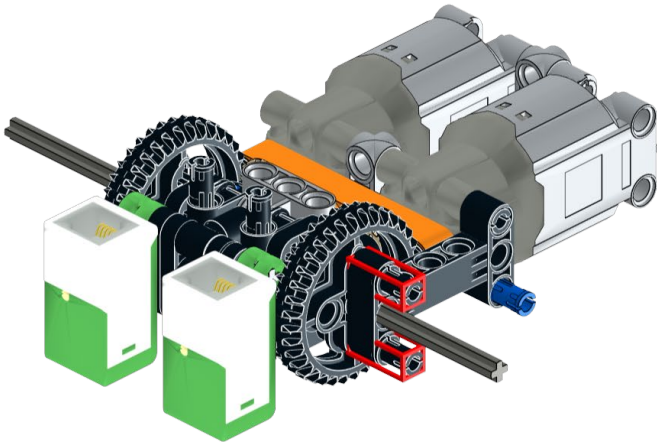
22



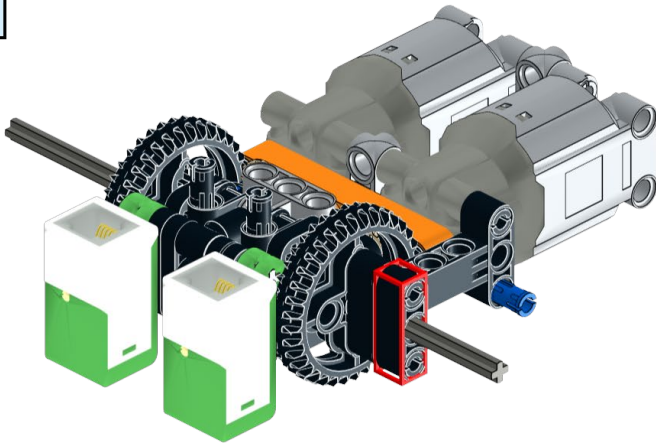
23



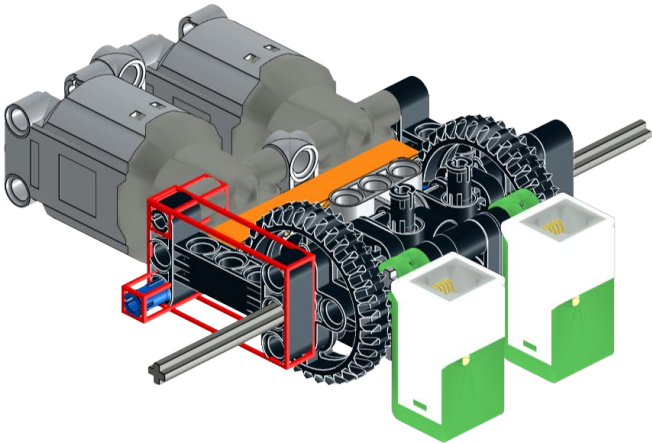
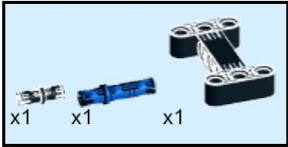
24  x2



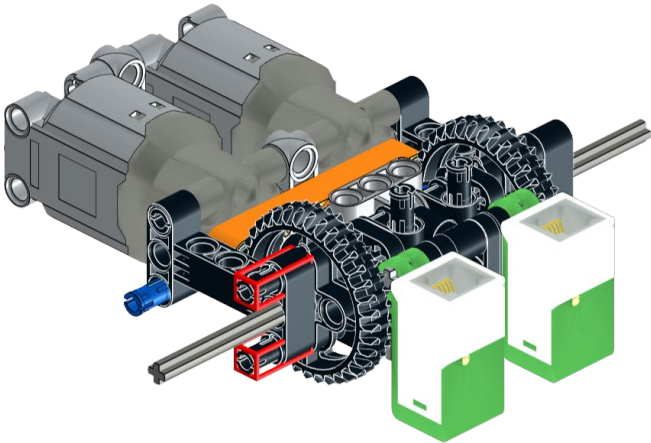
25  x1



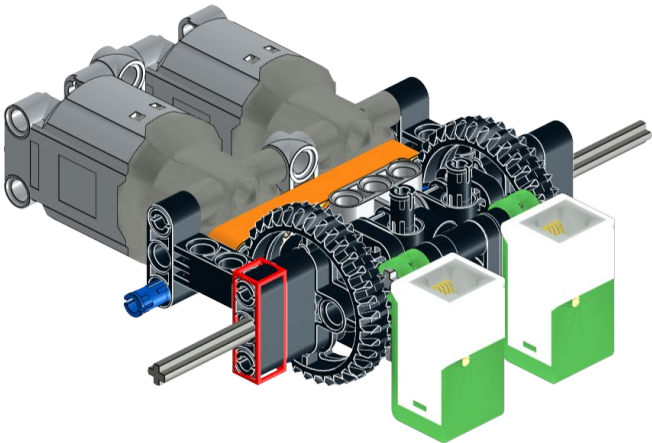
26



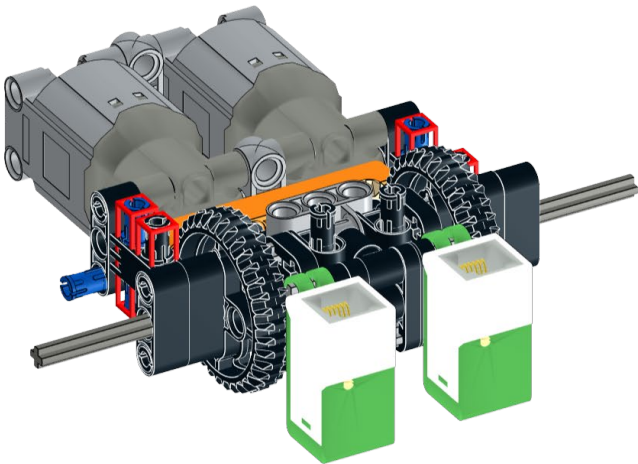
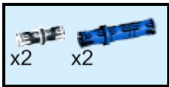
27



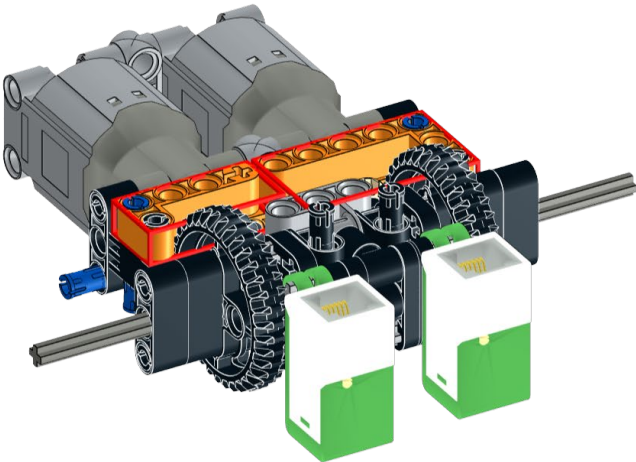
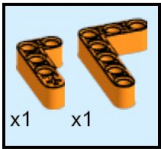
28



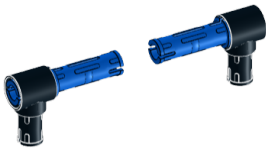
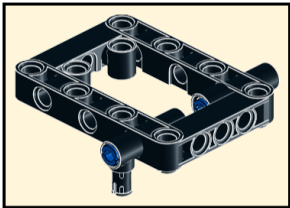
29

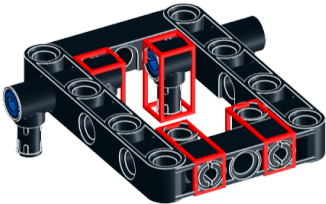
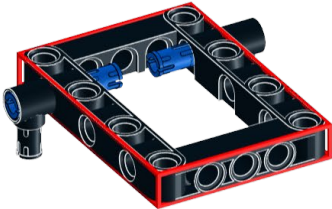
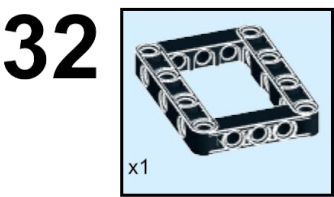


30



31

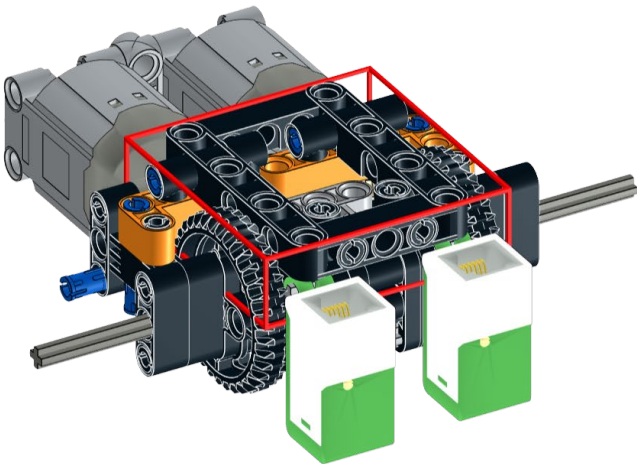




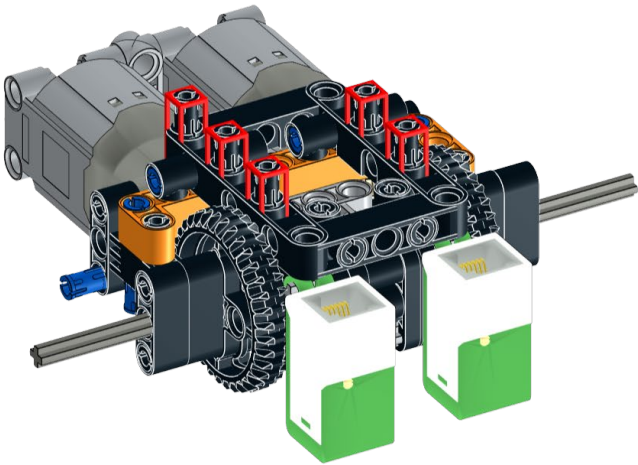
34



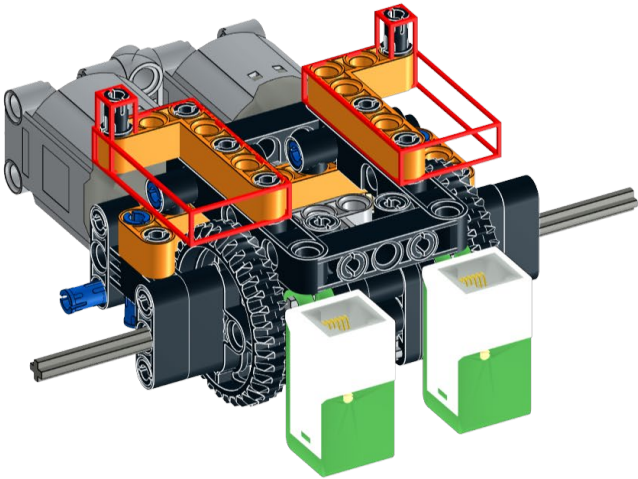
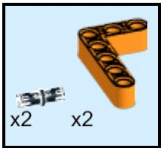
35



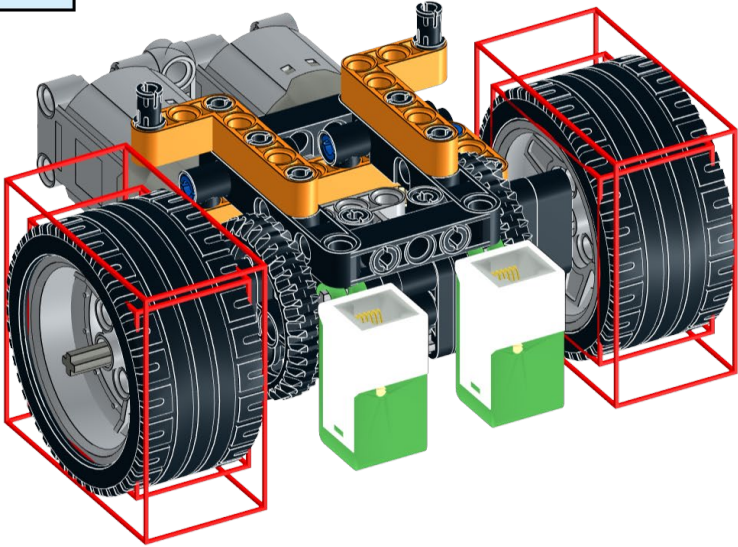
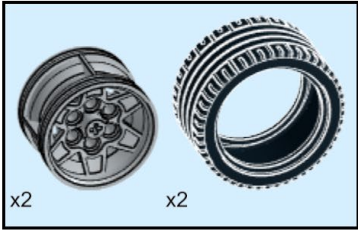
36



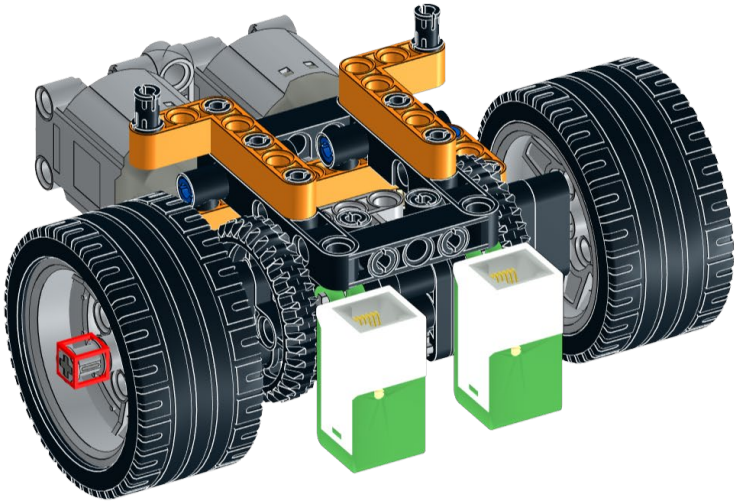
37



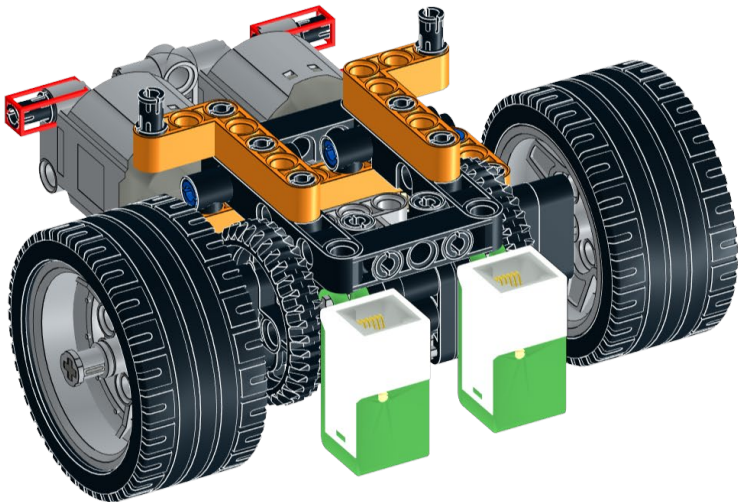
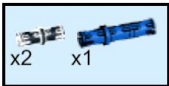
38



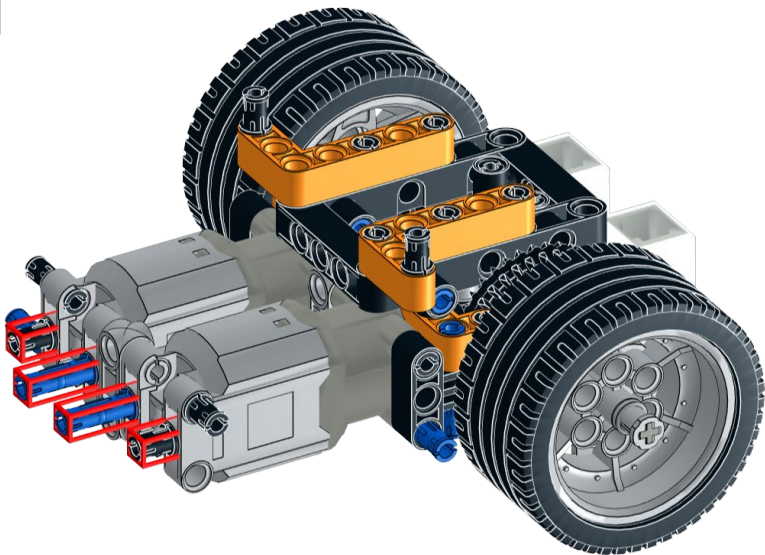
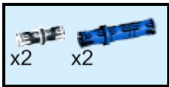
39



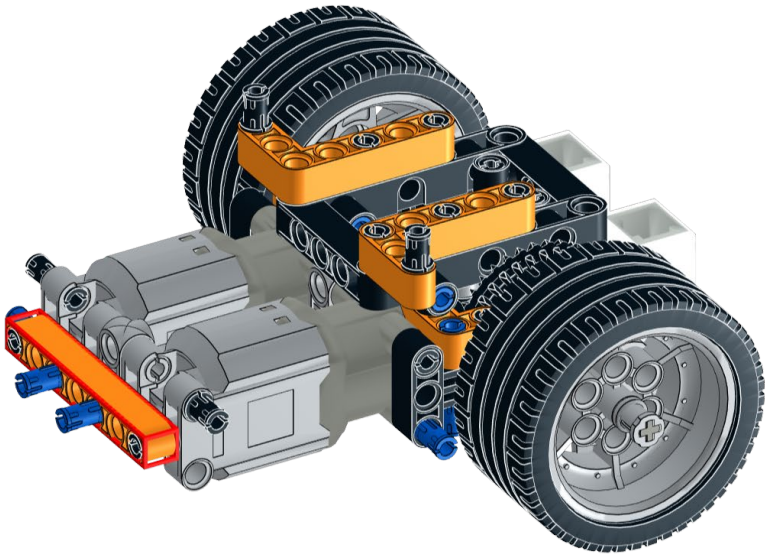
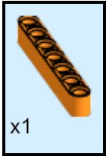
40



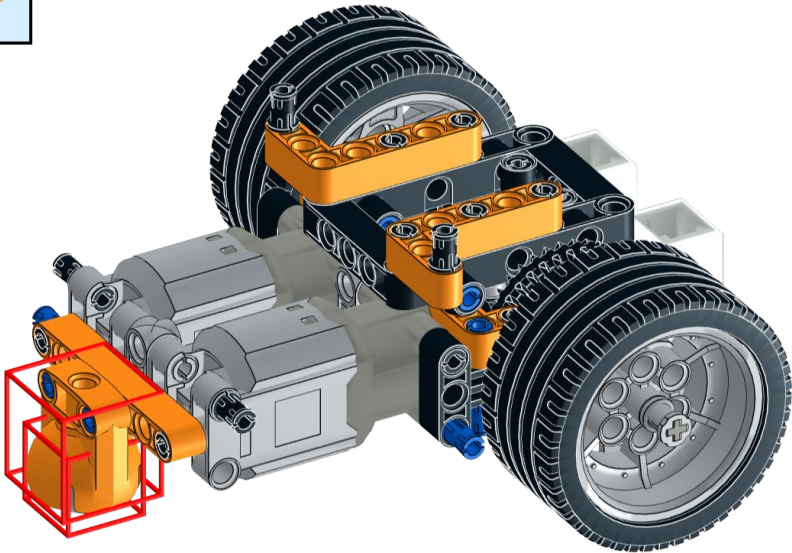
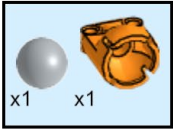
41



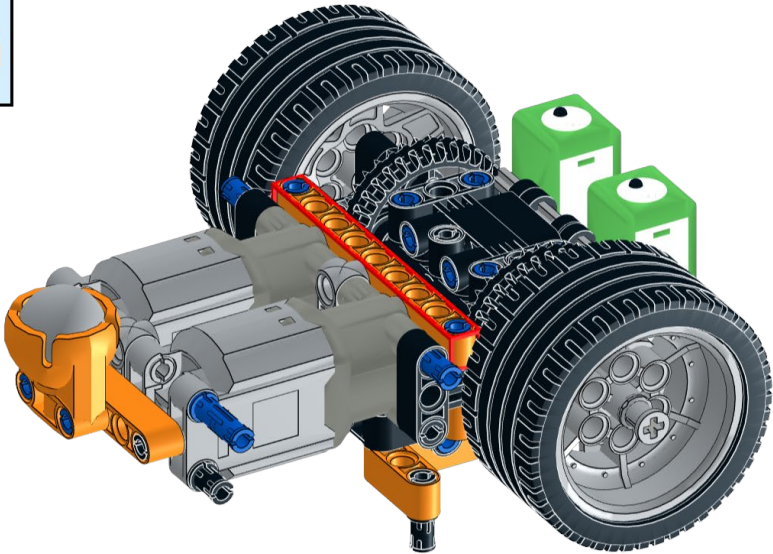
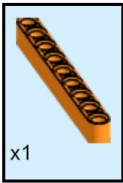
42



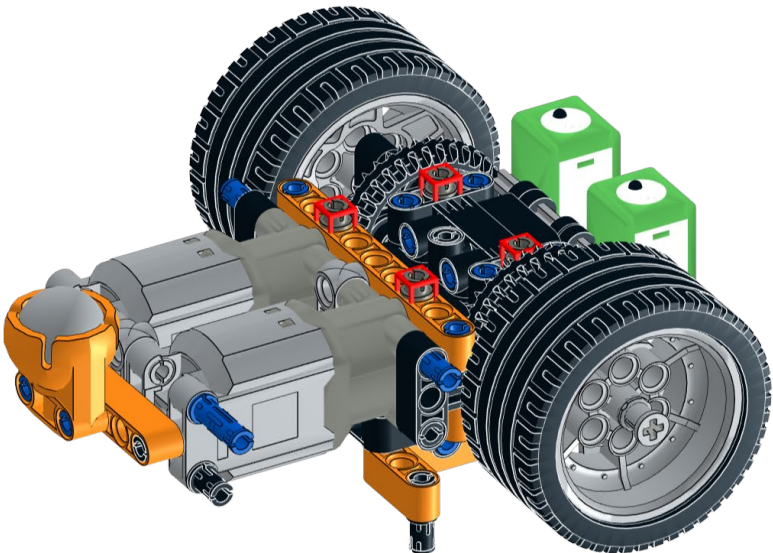
43



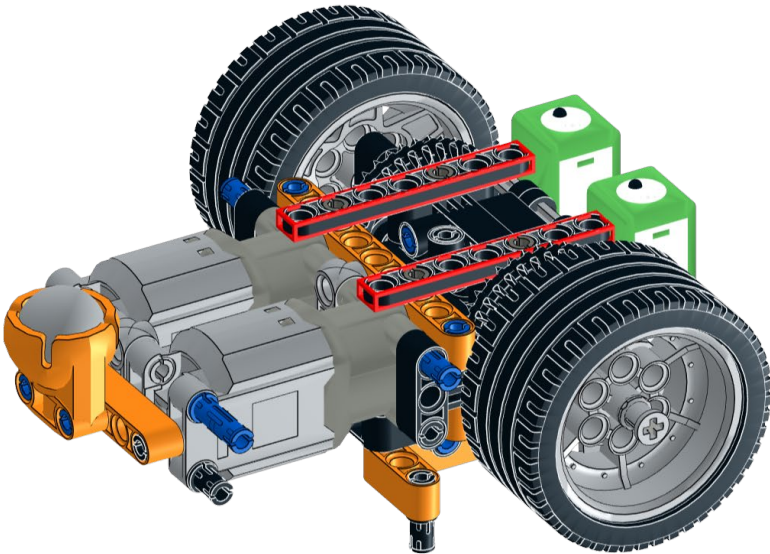
44



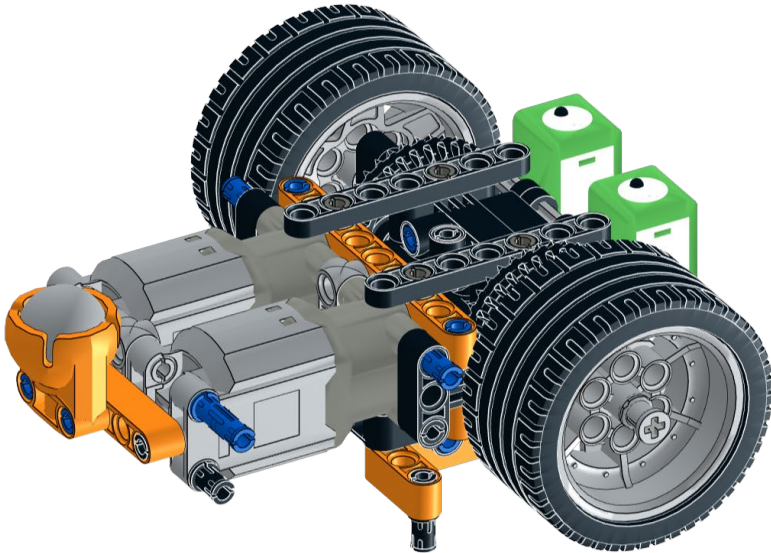
45



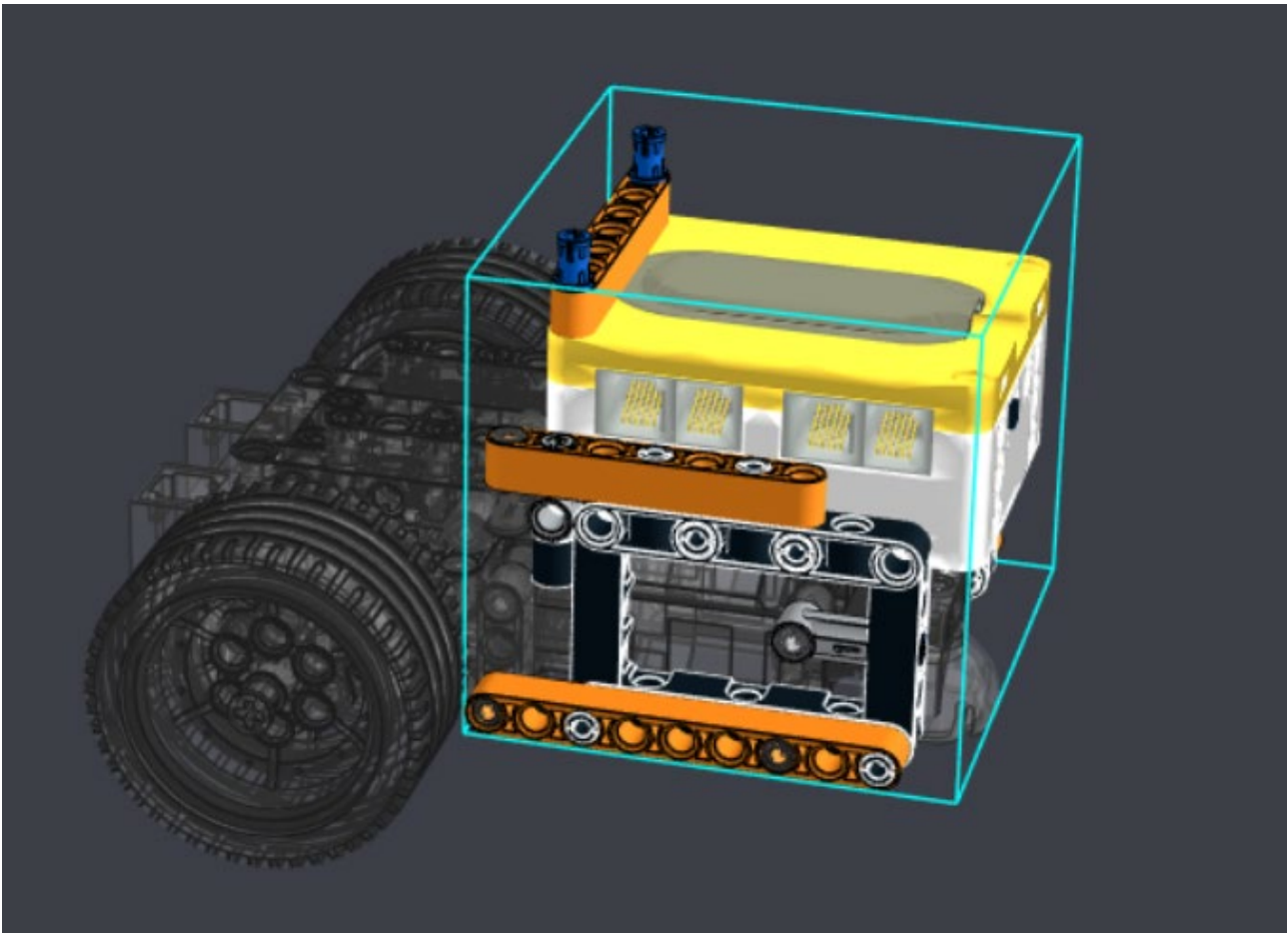
46



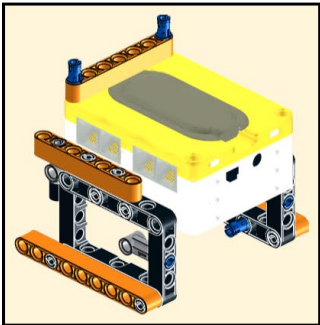
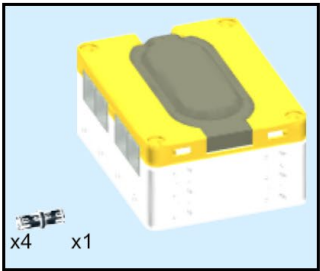
47



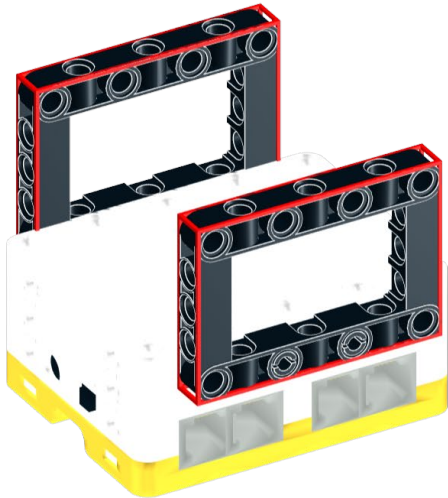
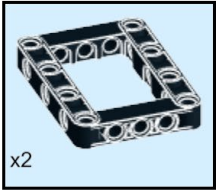
二、车身及框架



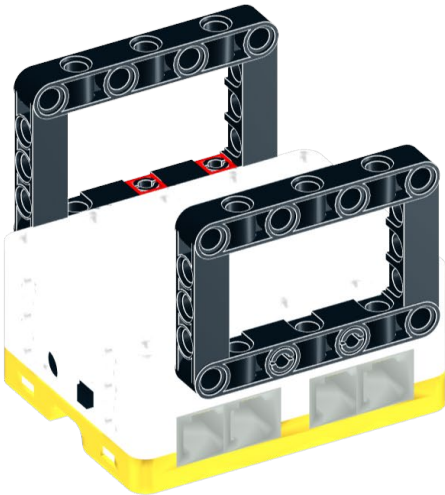
48



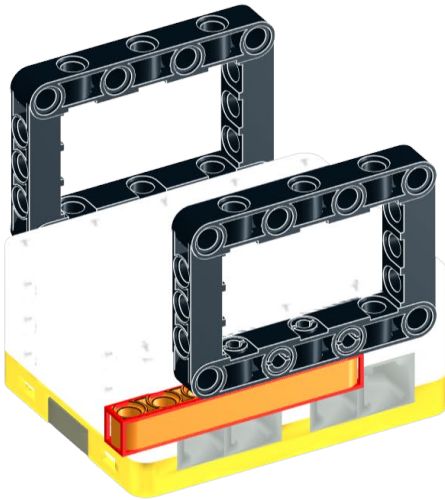
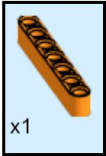
49



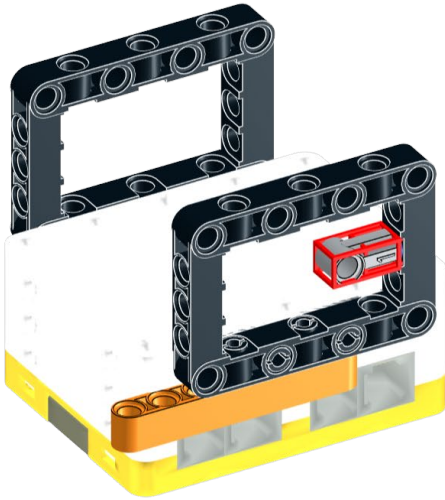
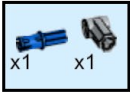
50



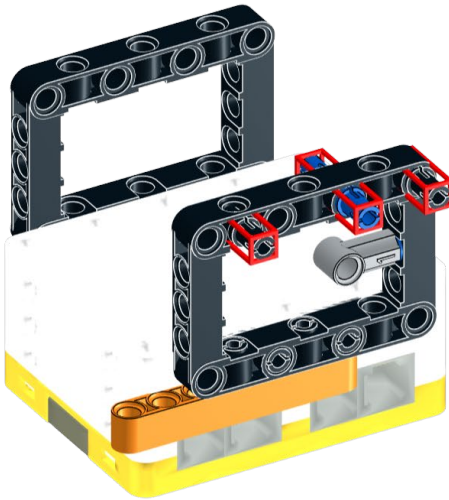
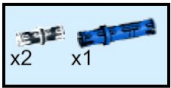
51



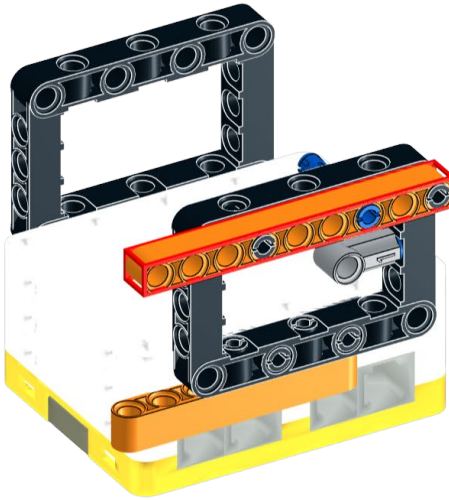
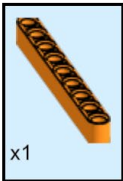
52



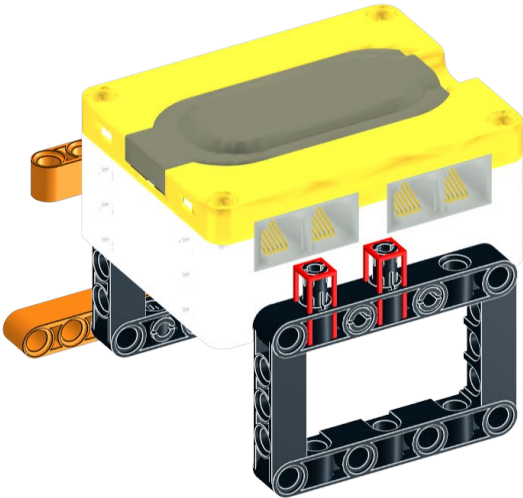
53



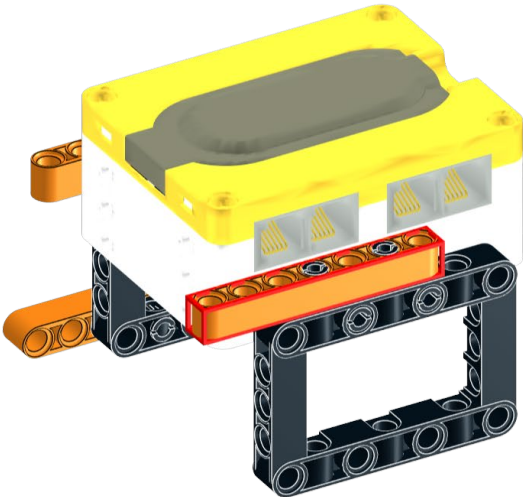
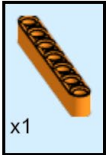
54



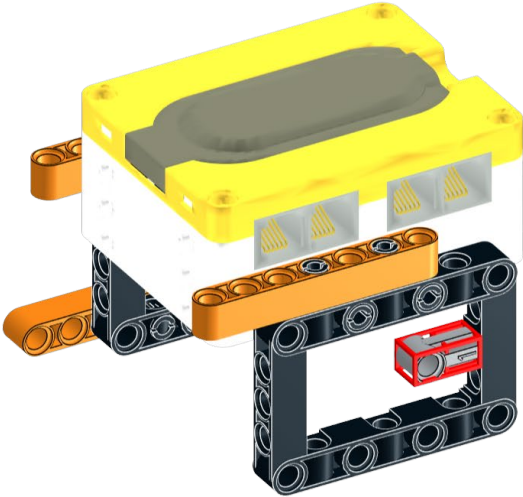
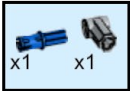
55



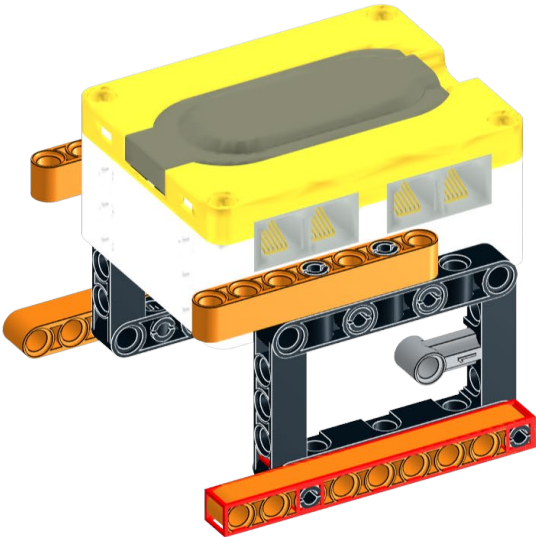
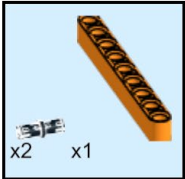
56



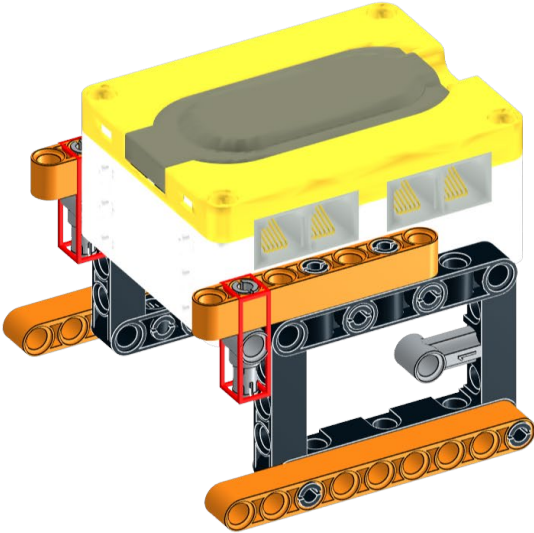
57



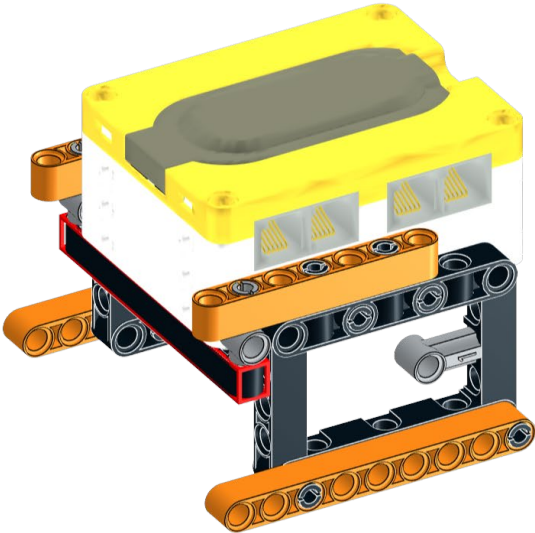
58

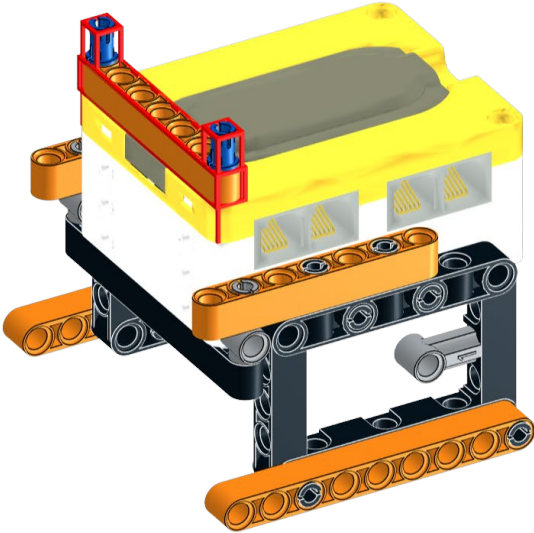
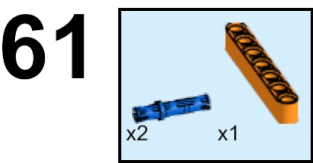


59

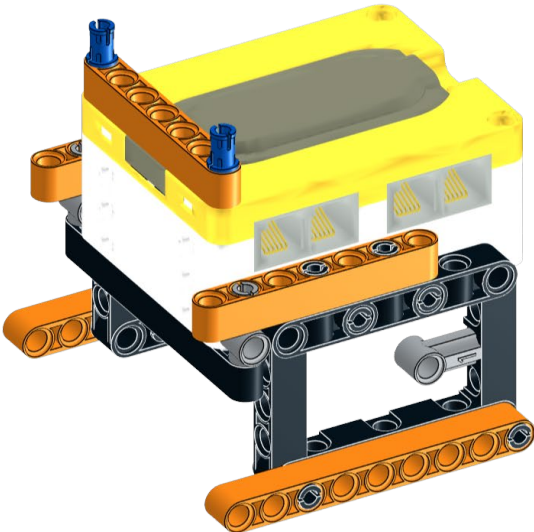


60

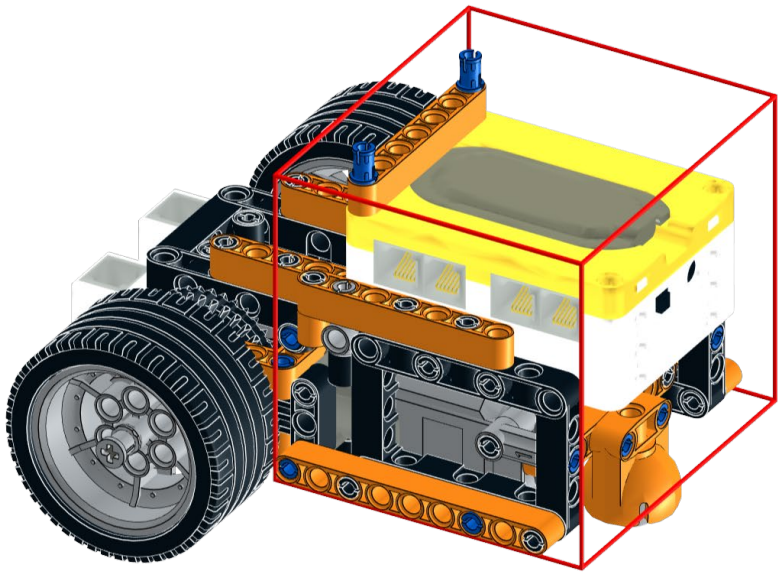




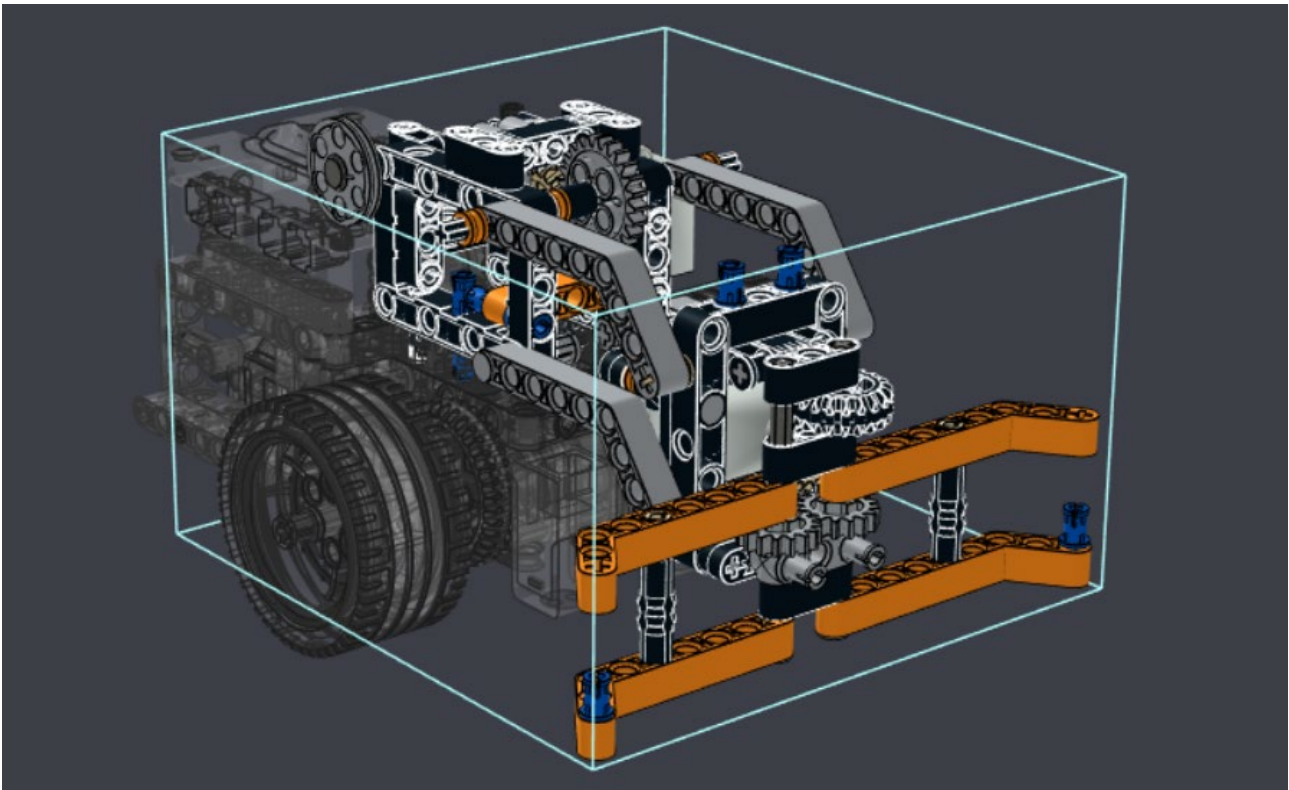
62



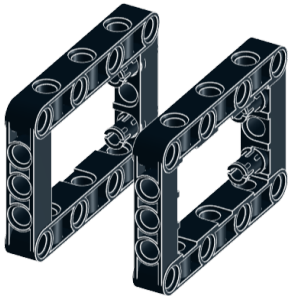
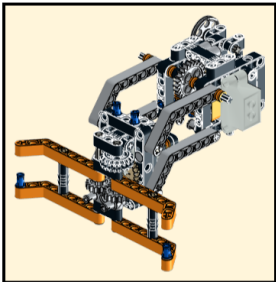
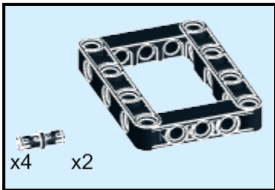
63



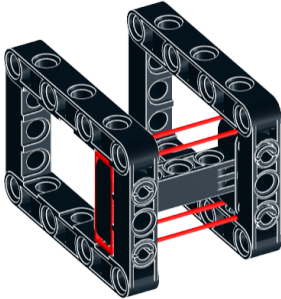
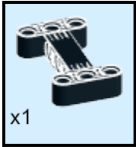
三、机械臂与机械爪



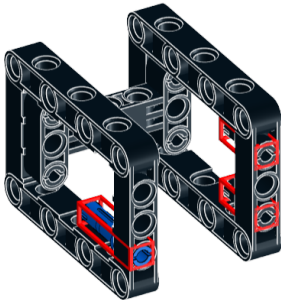
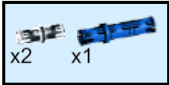
64

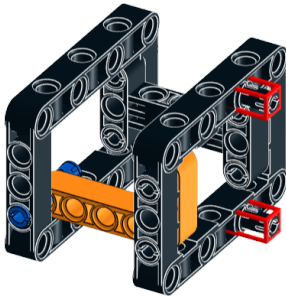
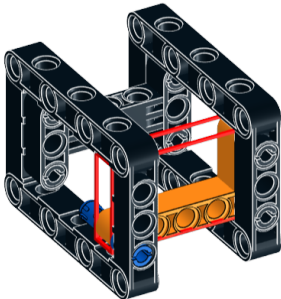
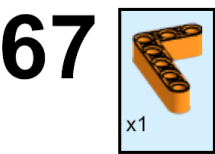


65

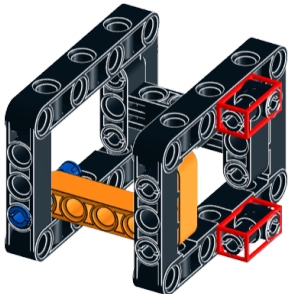


66

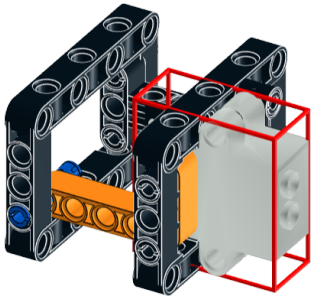
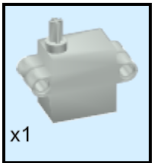




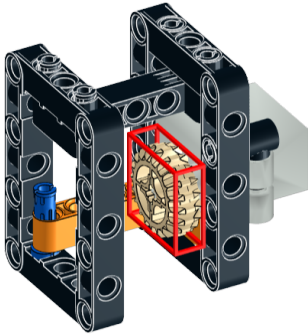
69



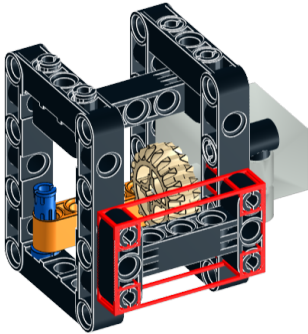
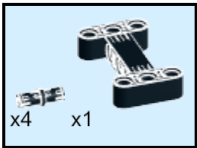
70



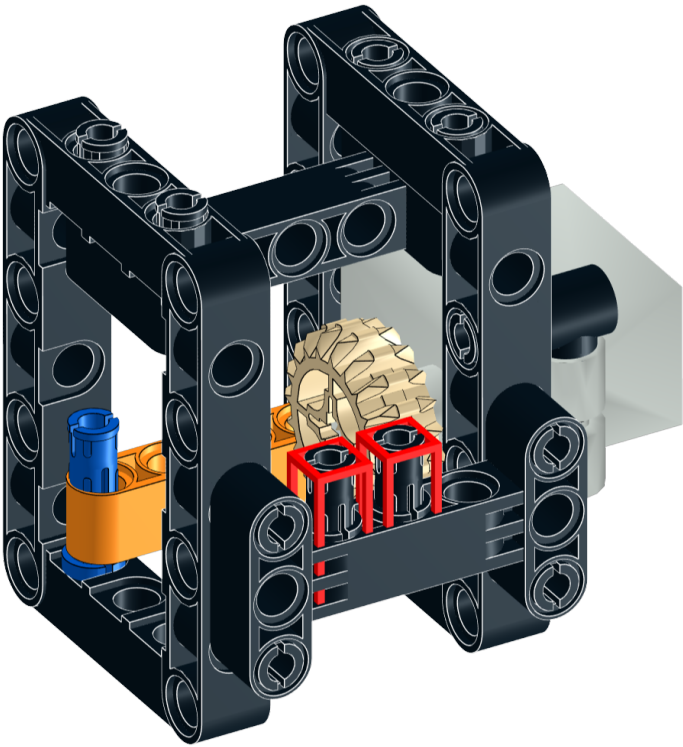
71



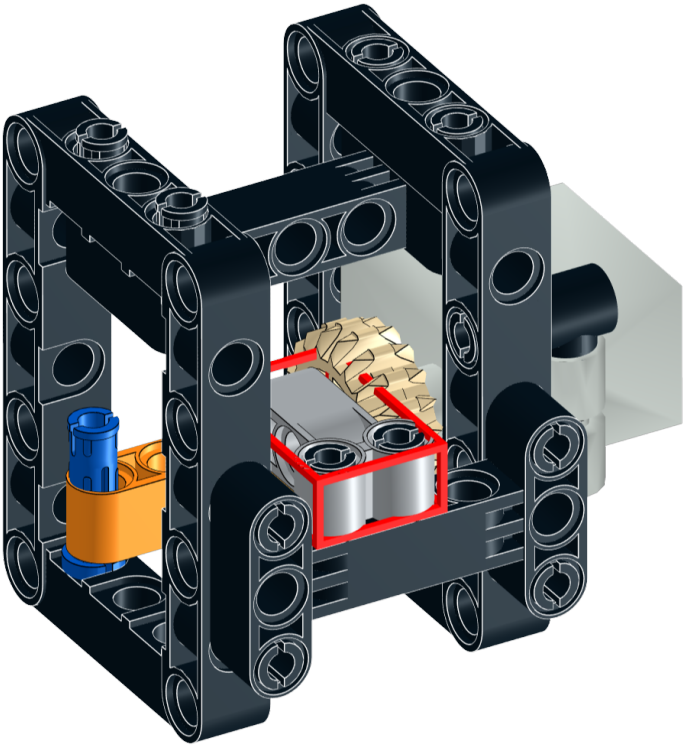
72



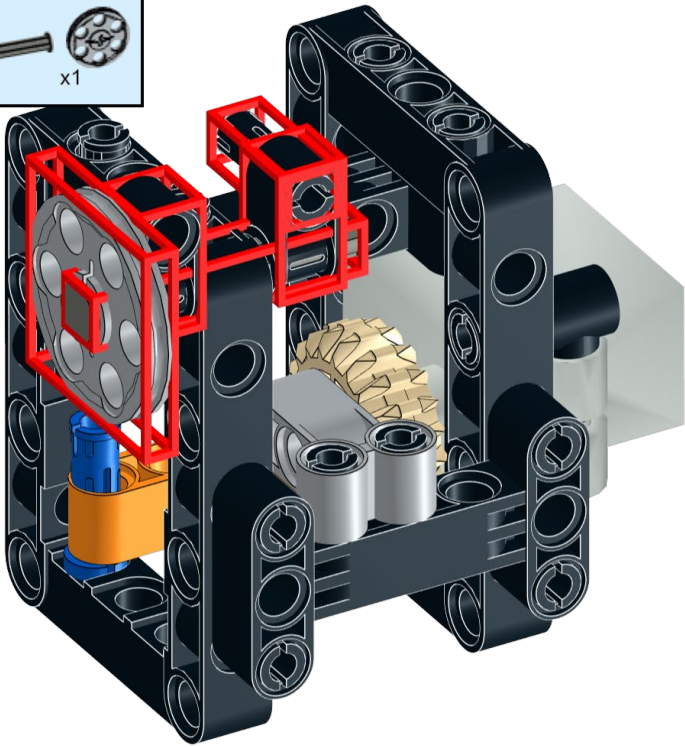
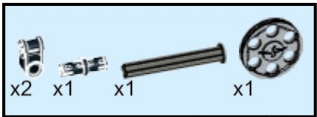
73



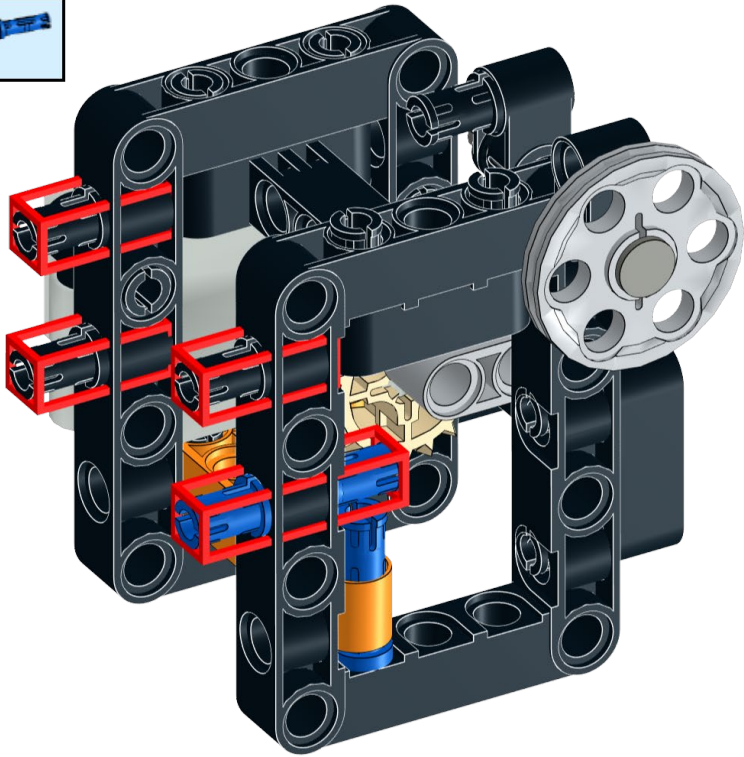
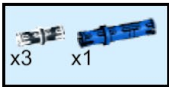
74



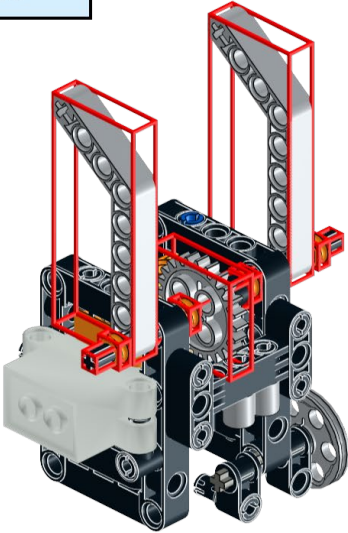
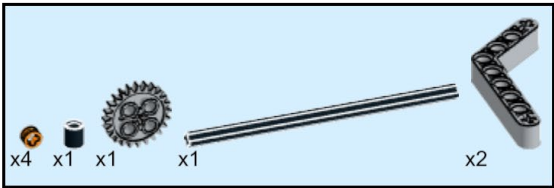
75



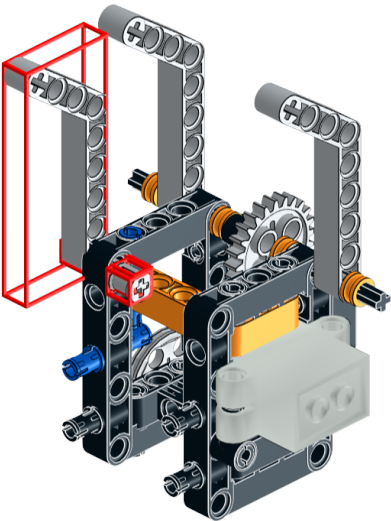
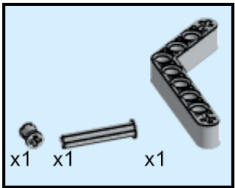
76



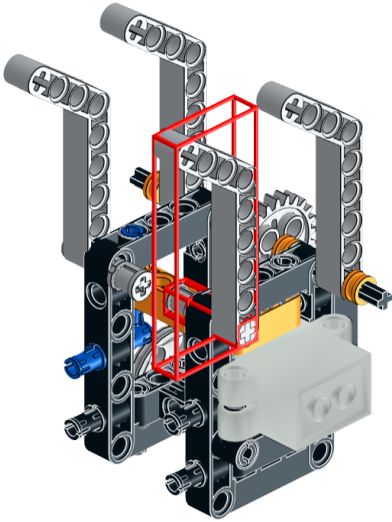
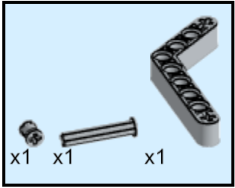
77



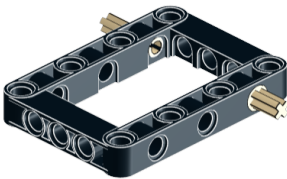
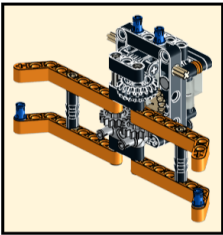
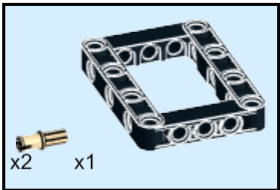
78



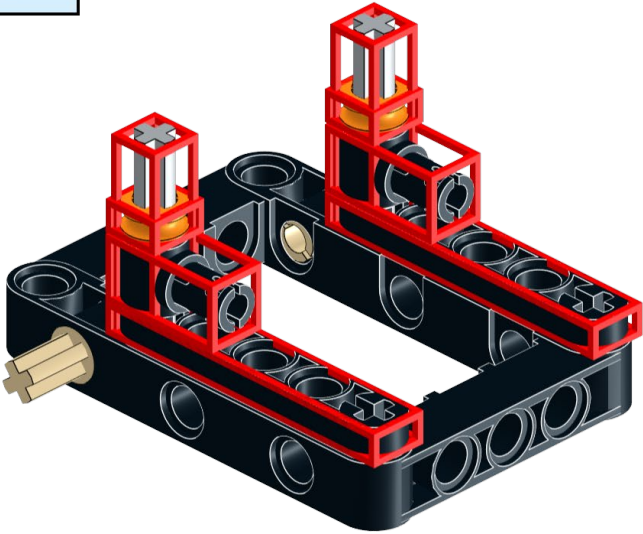
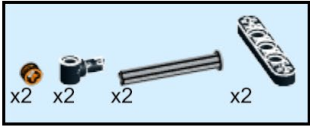
79



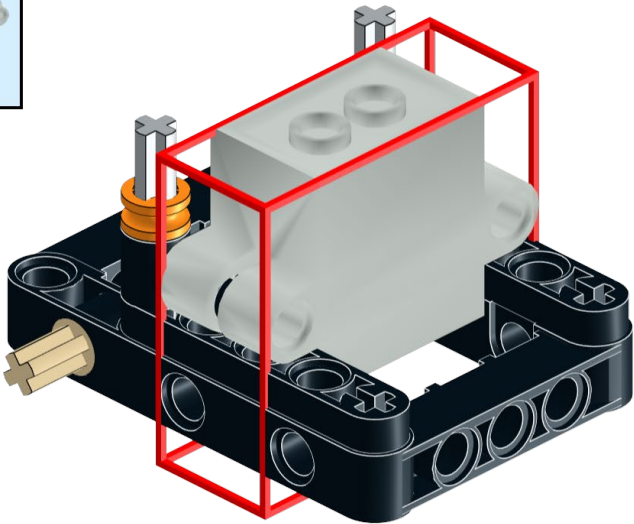
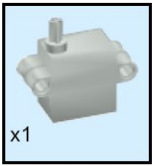
80



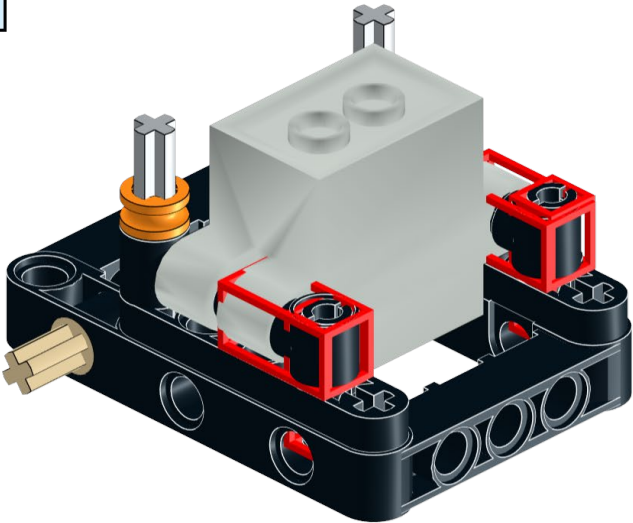
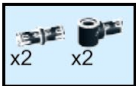
81



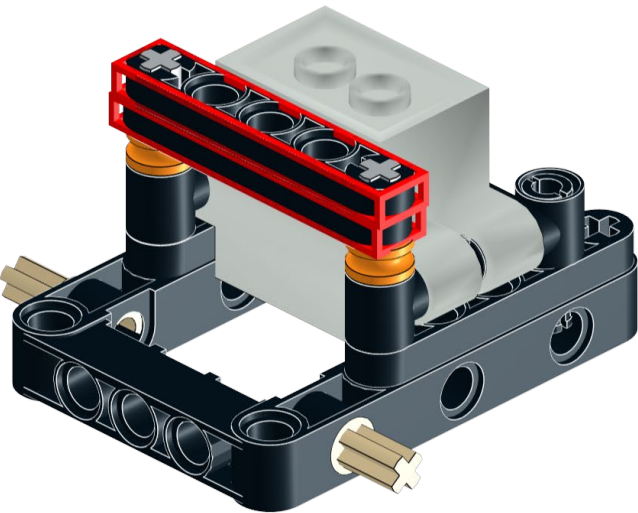
82



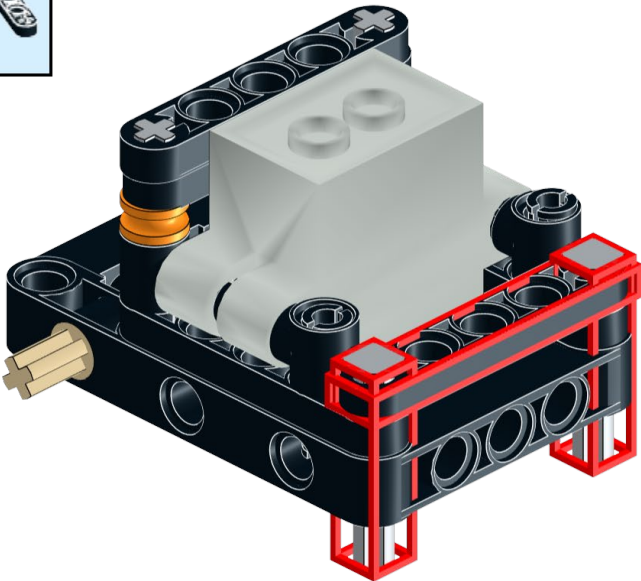
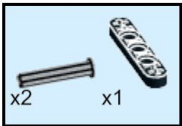
83



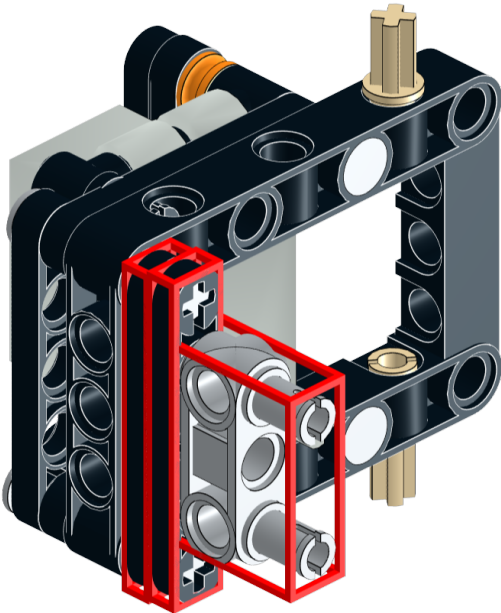
84



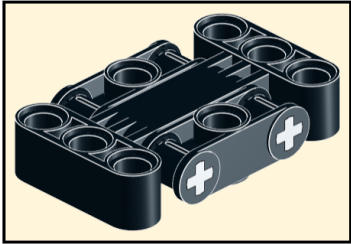
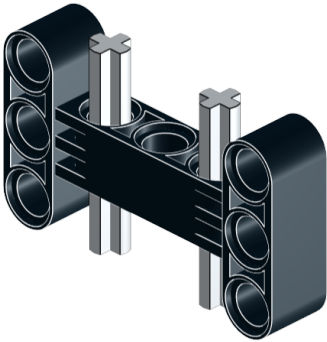
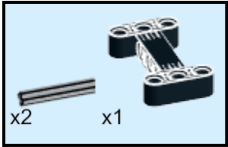
85



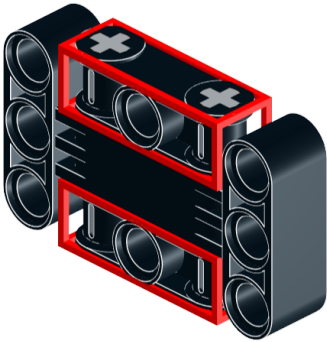
86



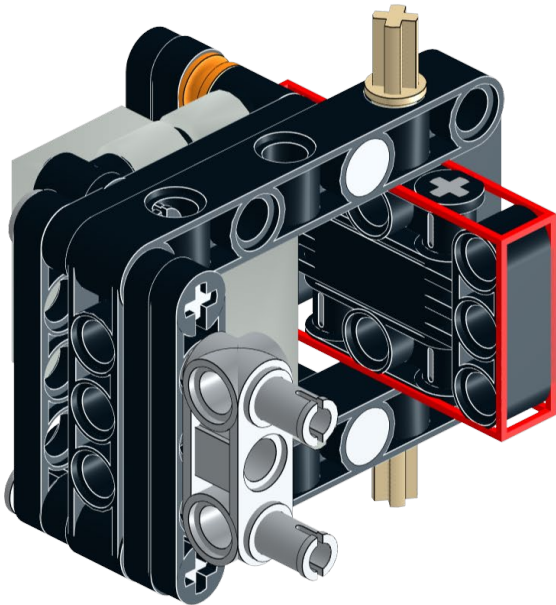
87



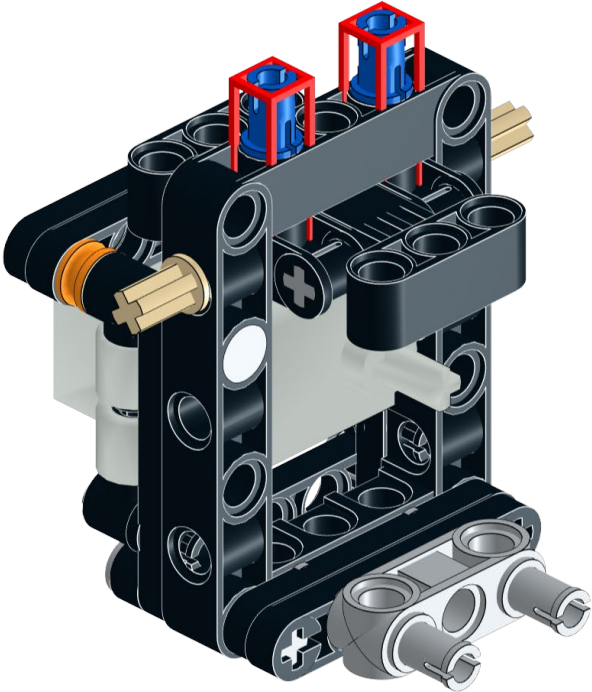
88



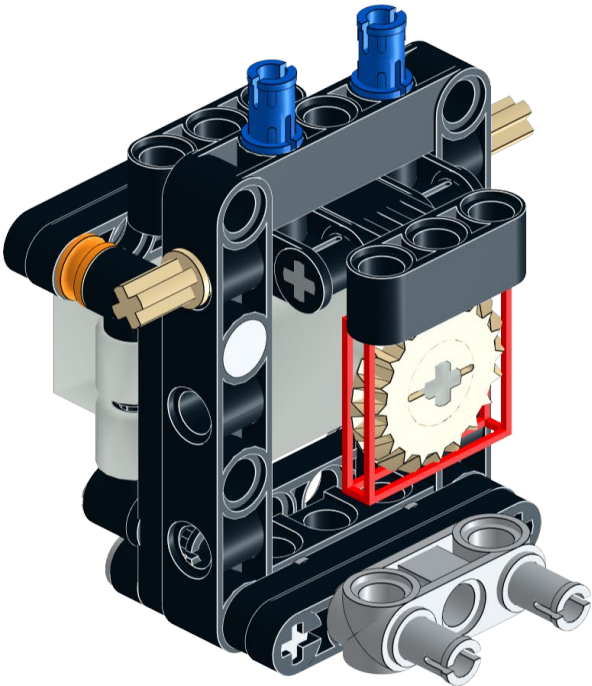
89



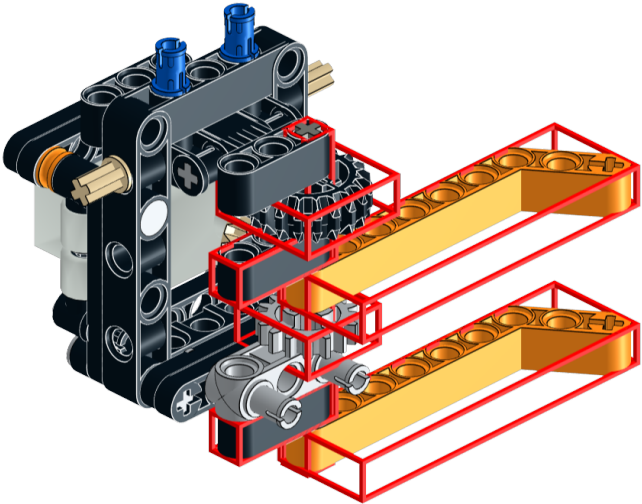
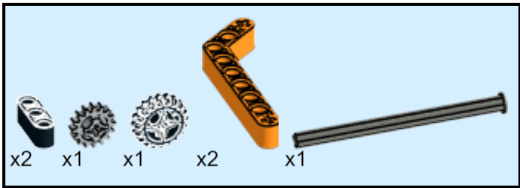
90



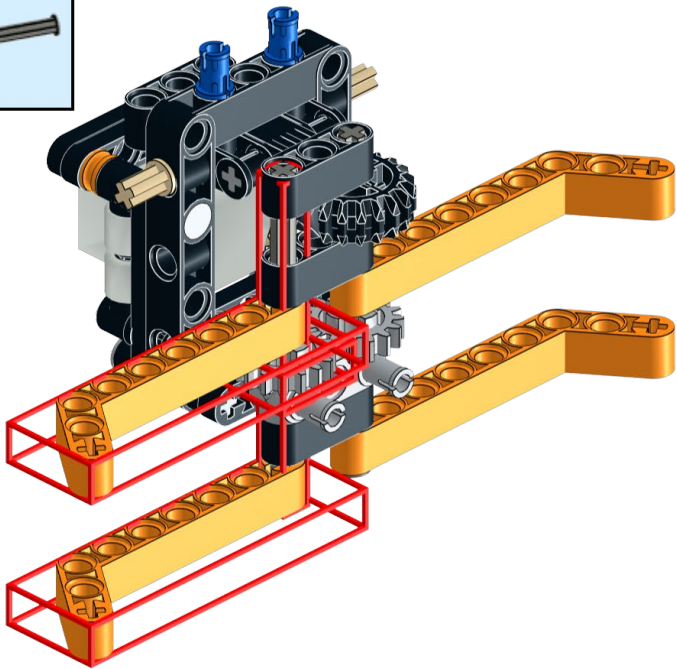
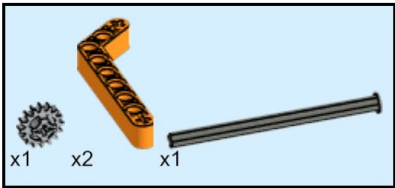
91



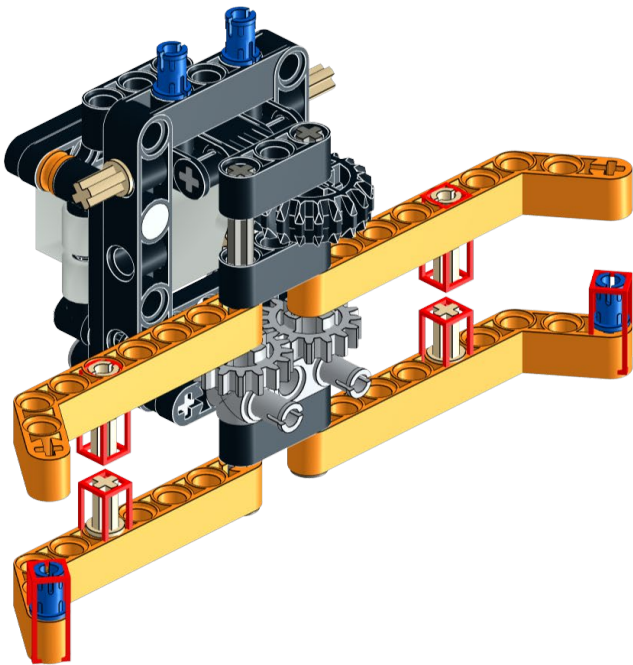
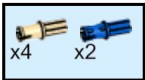
92



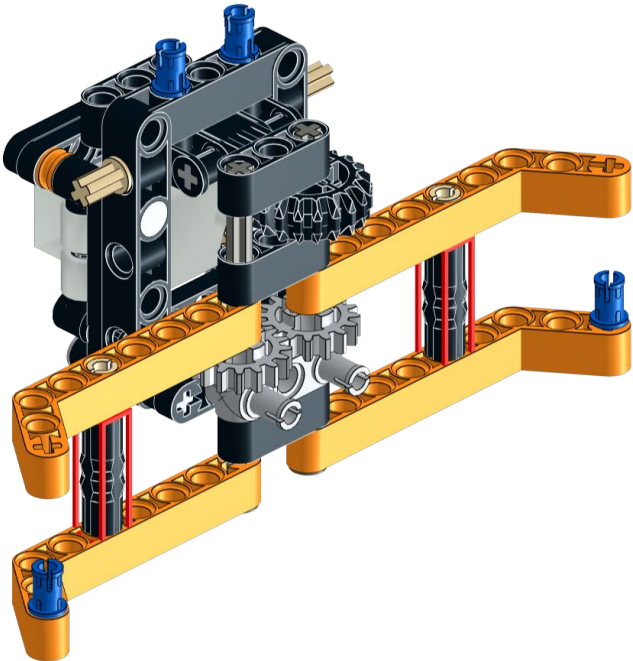
93



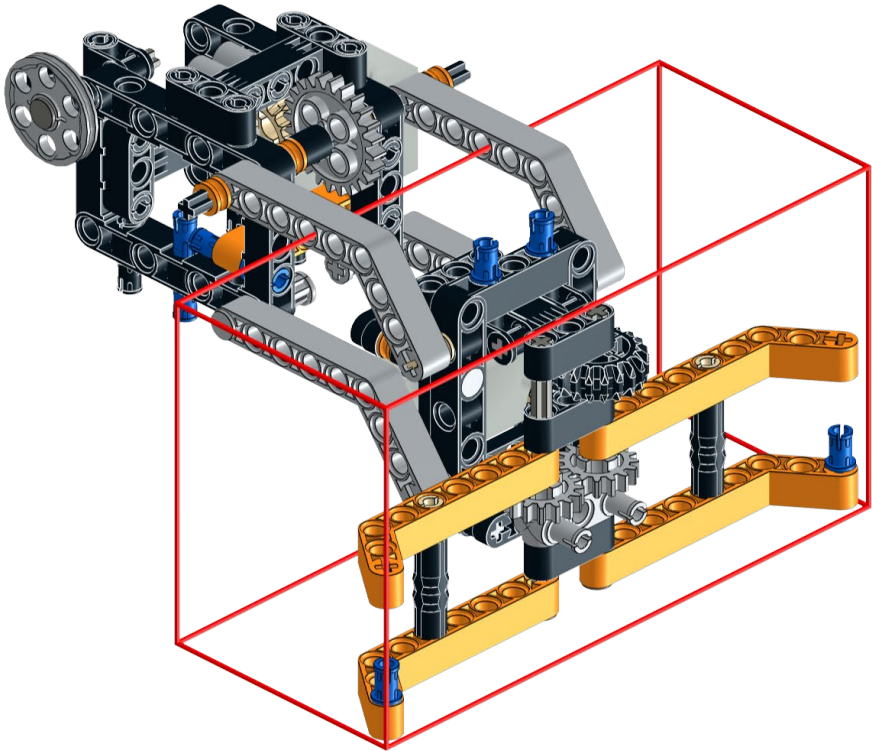
94



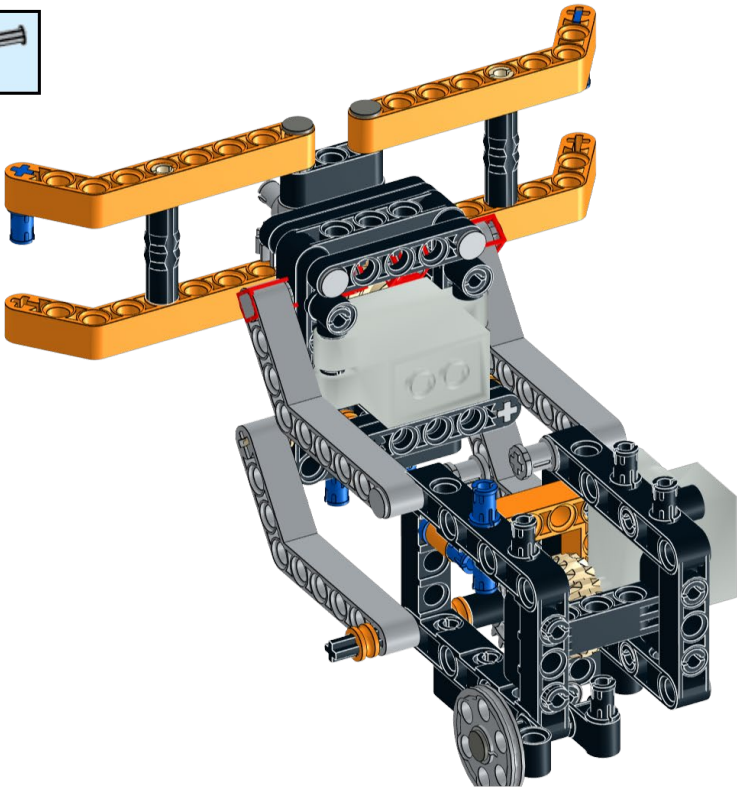
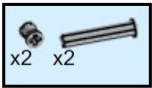
95



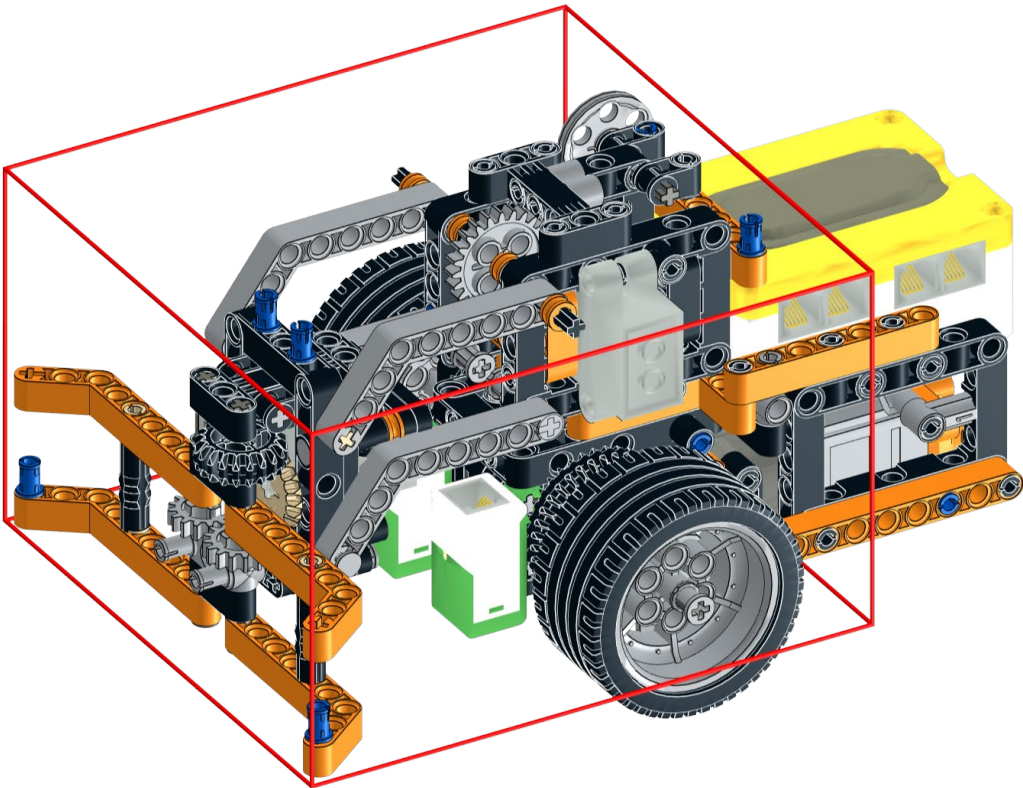
96



97



98



第三课 积木结构件基础

一、 齿轮组结构件特点

齿轮的分类有多种方法，如果按齿数来分，齿轮的齿数从 1 个齿的蜗杆到 40 齿齿轮，一共有 9 种齿数。如果按形状分类可分为直齿轮、双面齿轮、锥形齿轮、齿条、蜗杆等。

A. 40 齿齿轮

B. 24 齿齿轮

C. 8 齿齿轮

D. 双面 36 齿齿轮

E. 双面 20 齿齿轮

F. 双面 12 齿齿轮

G. 16 齿空心齿轮

H. 12 齿齿轮齿盘

I. 螺旋齿轮

J. 1x7 齿轮条

K. 1x13 齿轮条

二、 结构件-轴的特点

轴的横截面是十字形，与带有十字孔的零件连接时候不会产生打滑现象，可以进行稳定的动力传输。轴的主要按纵向长度进行分类：

A. 12# 十字轴

B. 10# 十字轴

C. 9# 十字轴

D. 7# 十字轴

E. 5# 十字轴

F. 3# 十字轴

G. 2# 限位轴

三、结构件-梁的特点

梁的典型外形是两端为半圆形的长方体，中间带有圆孔。按照梁的形状可以分为平（直）梁、角（L形）梁、弯梁、方框梁、T型梁、工字（H）梁等。

A. 15孔平梁

B. 13孔平梁

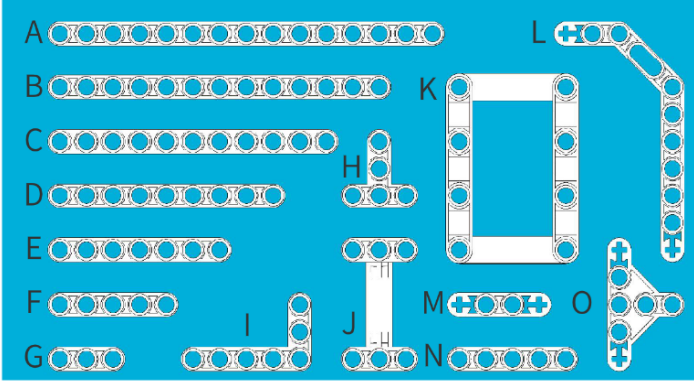
C. 11孔平梁

D. 9孔平梁

E. 7孔平梁

F. 5孔平梁

G. 3孔平梁



H. 3x3 T型梁

I. 3x5 L型梁

J. 3x5 H型梁

K. 5x7 方框梁

L. 3x7 带轴栓双弯梁

M. 1x4 带轴栓平梁（薄）

N. 1x5 平梁（薄）

O. 3x5 三角梁（薄）

认识积木——销、轴套、万向轮

A

B

C

D

E

F

G

H

I

J

K

A. 长两头栓摩擦销

B. 半十字轴半栓

C. 1x2 摩擦销

D. 1x2 活扣销

E. 带球栓球头

F. 3/4 栓

G. 十字轴套

H. 十字半轴套

I. 3# 十字轴带短栓

J. 十字轴带长栓

K. 万向轮