**实验二 机器人语音对话**

**一、实验目的**

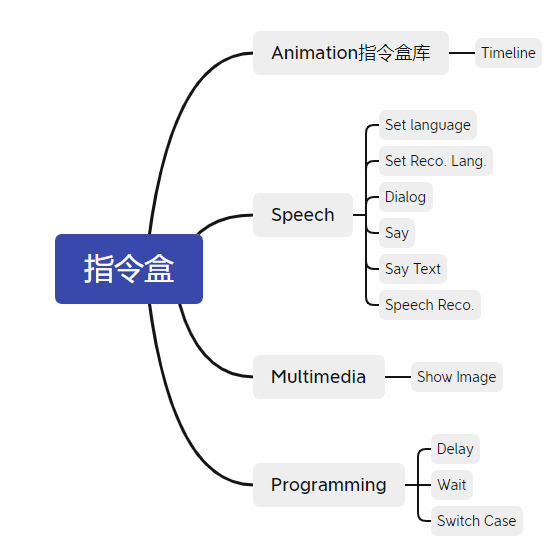
学习机器人自然语言交流功能，学会用两种方式实现机器人对话功能，并结合机器预设动作或编排动作、多媒体指令盒，实现机器人在一定场景下的语音交互。

**二、实验内容**

与机器人的语音对话，是类人机器人的一个重要功能。

设定一个场景，当机器人看到你，你和机器人打招呼，或摸它的头或者触碰肩膀或手臂时，机器人将与你进行交流，并根据交流内容给出不同的回答或行为反馈。

**三、使用到的指令盒**

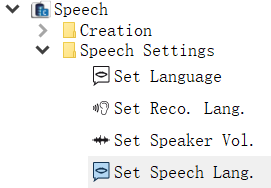


**四、实验步骤**

在choregraphe里，机器人的语音功能在Speech指令盒库，包括说和对话，说和对话的区别在于是否有交互，说仅仅是机器人说出设置好的语句，对话则可以设置对话的内容。

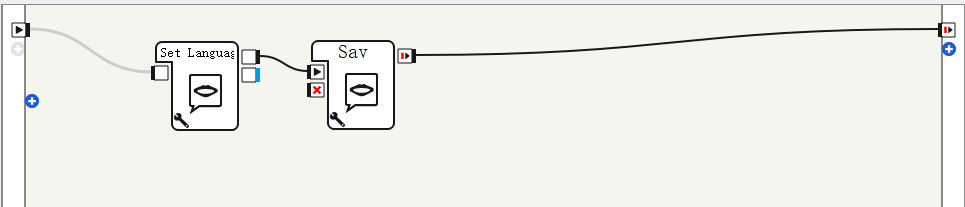
当然，用Say指令盒也是可以制作对话的，它需要结合Speech Reco指令盒来使用。

在Speech指令盒库中，还有Speech Settings指令盒库，用于设定说话的语言、语言识别、设定说话音量。

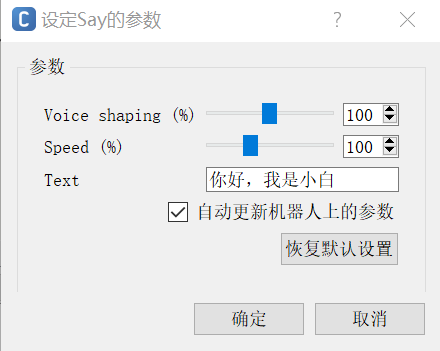
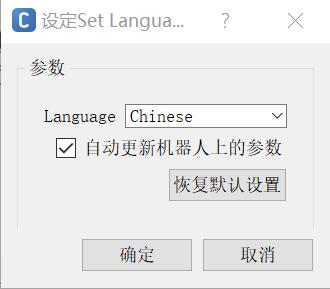


**1、让机器人介绍自己**

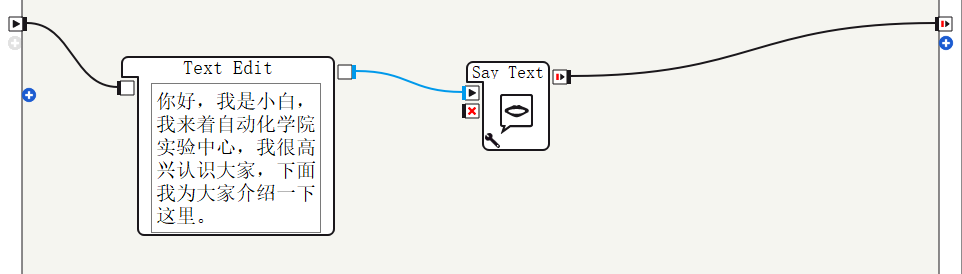
我们让机器人介绍自己，在流程图画板里拖入Set Language和Say指令盒，并将其输出顺序连接起来。



点击Set Language指令盒的参数设定选项，选择语言为中文，设置Say指令盒的参数设定选项，输入说的内容“你好，我是小白”。



如果需要表达的内容比较多，则可以用Say Text指令盒，Say Text指令盒前面的输入只接受字符串，所以一般与Text Edit指令盒组合使用，位于Programming-Data Edition-Text Edit，在Text Edit中设定要说的字符串。Say Text的参数中可以设定音调Voice shaping(%)与速度Speed(%)。



如果想要机器人在说话的同时伴随肢体动作，采用Animated Say指令盒可以简单的实现Pepper一边说话一边做出动作了。

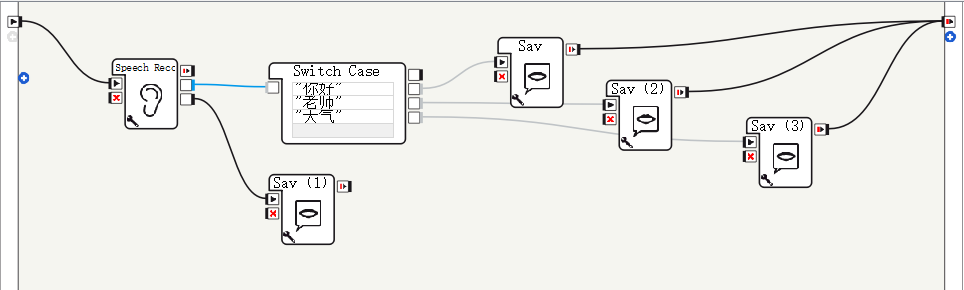
Pepper内置了丰富的肢体动作，例如鞠躬、招手、哭泣、摇头、弹吉他等等。

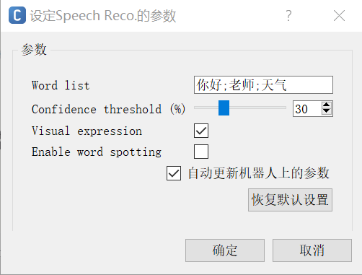
此外，如果是由多个指令盒共同完成一个功能，可以将这几个指令盒组合为一个指令盒，类似于PPT或Visio中的框图组合，具体方法为：选中需要组合的几个指令盒，右键选择“转化为指令盒”，重命名即可。

**2、使用Speech Reco.和Say指令盒实现简单自然语言对话**

我们不仅仅是希望机器人能说话，更希望机器人可以与我们对话，下面我们就用say指令盒来实现简单的自然语音对话，一般与Speech Reco.指令盒结合使用。

Speech Reco.指令盒可以识别声音中的关键字。在参数“Word list”中设置关键字，多个关键字可以用英文分号隔开。当识别到后会在指令盒后输出这个关键字字符串。一般可以与“Switch Case”指令盒组合使用。





另外，可以设定识别准确率的阈值。当阈值越高，识别不出的情况会增加。当阈值越低，被错误识别的情况会增加。Pepper听到每句话的识别准确率，可以在Choregraphe的对话面板中实时查看到。

通过这种方式，就可以设置具有一定交互性的对话内容，条件设置的越多，对话的内容也会更加丰富。

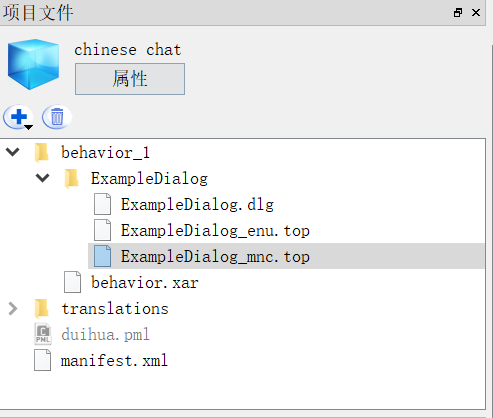
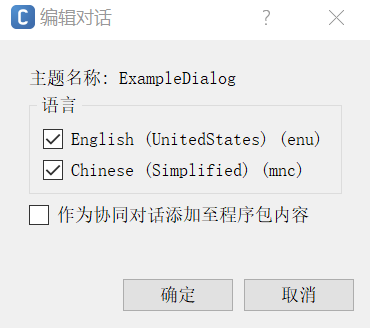
还有需要注意的是，建议和Wait指令盒组合使用。如果Pepper在20秒（推荐值）后仍然无法检测到正确的人声，请提示人类下一步如何做。防止出现人类反复向Pepper确认的情况。

**3、使用Dialog实现自然语音对话**

前面讲到的say指令盒对话形式一般比较简单，对于复杂的对话的场景，推荐使用Dialog指令盒。该指令盒使用一种叫做QiChat脚本的对话定义方式。

当我们向流程画板中拖入Dialog指令盒后，在工程中可以看到自动增加了top文件。双击后，即可以编辑QiChat脚本。

在这里需要注意的是，如果我们要设置中文对话，需要在项目文件属性里添加选择中文，然后再双击已经在流程图画板上的say指令盒，将中文语言选上，此时，可以看到在工程里出现了中文的top文件，双击后，即可编辑中文的QiChat脚本。



我们编写了如下一段这样的对话脚本来说明Qichat的对话编写规则。

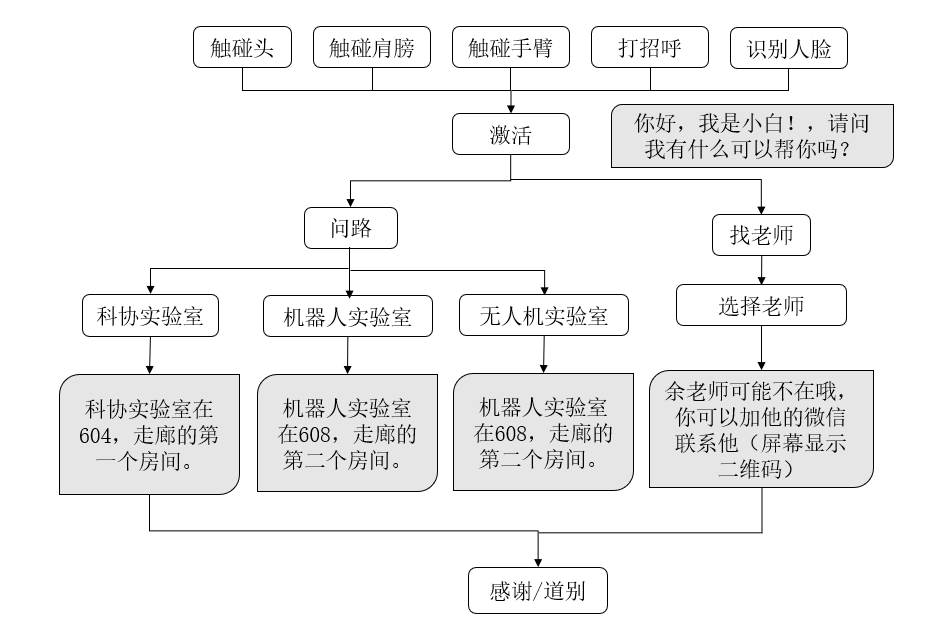
|  |
| --- |
| concept:(yes)[是 对 喜欢 是的 嗯]  u: 你好，你喜欢动物吗？  u1:(~yes)狗和猫，你更喜欢哪个？  u2:(狗)我也是，我喜欢金毛。$onStopped = 1  u2:(猫)一样啊，我喜欢布偶。$onStopped = 2  u1:([不 讨厌 不喜欢]) 原来是这样啊。$onStopped = 3 |

|  |  |
| --- | --- |
| 语法 | 说明 |
| concept | 代表一组短语赋值给一个变量，在用户规则中使用~符号调用。小括号内定义的变量名，后面可以定义一组同义词，例如是、对、喜欢都代表yes。只要Pepper听到这些关键字，就认为人的想法是yes。方括号[ ]里的为可替代词汇，也可以在方括号[ ]里加上大括号{ }，{ }里的为可省略词汇。 |
| u | u:(human input) robot answer  当Pepper听到(human input)时，他会说出robot answer。例如Pepper问你喜欢动物吗？人表达了“yes”的想法后，Pepper会说“狗和猫，你更喜欢哪个？” |
| uX | 反应选项，Pepper听到人回答后的反应选项。 |
| $onStopped = 1 | Dialog指令盒向onStopped输出数据1。指令盒运行结束。 |

语音语调的改变

**4、场景语音实现**

请你按照上面所讲的对话实现形式，完成下面流程图所示的场景对话。



**五、实验拓展**

请你在4.4场景语音实现的基础上，为每个分支对话设计一个机器人动作或多媒体展示，让机器人的对话场景更加合理和生动。比如，提示实验室位置时可以添加手部指引动作，找老师时可以报出老师的联系方式或展示老师的微信二维码等；在对话过程中，与机器人触摸情况下机器人的反应对话等。