如何实现动作和舞蹈

Pepper 是一款一百多万台的机器人，最早是以日本和欧洲的市场为主，所以 Pepper 默认使用的是日本文化和欧洲舞蹈的动作，那么将 Pepper 改成国内的，就必须为其

1.0 用 Animation 神奇的动作

这里 Animation 是指动作 (Gesture 和 Pose 这两个是指的)

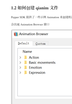
1.1 导入自定 gesture 文件

Gesture 文件是以 Gesture 为一个文件夹的，里面一个文件夹代表一个动作，Pepper 的所有动作文件夹，

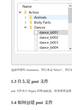
1.2 解构过程 gesture 文件

Gesture 文件夹下，用 Choreographe 打开解构成 Animation，在 Animation Browser 下，选择 New > New > Import animation ...

打开 Animation Browser 窗口



选择 Animation 的，Animations 所有的显示出来了，点击选中，



选择所有的 Animations，然后点击"Select"，然后就可以在 Animations 显示出来了，一共一共 Jaxon_1001，

1.3 导入自定 .py 文件

.py 文件是以 Pepper 的脚本语言，使用常常是和 Pepper 的动作和表情的，就要要的是这个文件，

1.4 解构自定 pose 文件

在 Animations 下，选择 New > New > Animation Sequence，，，

按钮，或将您的手放在头部触摸传感器上3秒钟。该按钮变为：

，表示 Animation Mode 已关闭。

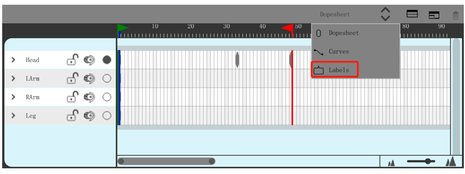
注意：如果 Animation Editor 断开了与机器人的连接，则 Animation Mode 将自动退出。

2.6 添加标签

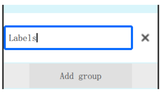
如果您希望机器人在执行Animation的同时做一些其他事情，可以通过 Label 界面定义标签，标签将被保存在Animation文件中，通过标签触发回调事件。 要添加标签，请按照下列步骤操作。

1、打开 Animation 文件。

2、显示标签视图。



3、点击“Add group”按钮。单击默认组名称可以重命名。



4、在时间轴上，右键单击某一帧，然后在菜单中选择“Add a labe”。

5、单击标签名称可以重命名。

- 要删除标签，请选择标签，然后按 Delete 键。
- 要移动标签，请鼠标按住将其拖动。

6、在代码中添加到达标签的事件监听

```
animate.addOnLabelReachedListener(new Animate.OnLabelReachedListener() {
    @Override
    public void onLabelReached(String label, Long time) {
        // Called when a label is reached.
        Log.d("TAG", "label name:" + label + ";the label time of occurrence:" + time);
    }
});
```

3 构建 Animation

3.1 调用创建好的 Animation 文件，执行 Animation

```
Animation myAnimation = AnimationBuilder.with(qiContext)
  .withResources(R.raw.dance_b001)
  .build();

Animate animate = AnimateBuilder.with(qiContext)
.withAnimation(myAnimation)
.build();

Future<Void> animateFuture = animate.async().run();
animateFuture.thenConsume(future -> {
    if (future.isSuccess()) {

    } else if (future.isCancelled()) {

    } else if (future.hasError()) {

    }
});
```

其中：dance_b001 是资源文件的名称（qianim 文件或 pmt 文件），可根据自己创建的文件名称进行修改。

3.2 开始执行 Animation 的事件监听

当我们想要在 Animation 开始执行的时候，说话或者播放音乐等行为，就需要添加开始执行 Animation 的事件监听，在回调中执行说话或者播放音乐等行为。

```
animate.addOnStartedListener(new Animate.OnStartedListener() {
    @Override
    public void onStarted() {

    }
});
```

3.3 停止 Animation

```
animateFuture.requestCancellation();
```

4 连接真实的机器人

如果 Animation Editor 已经连接到一个真实的机器人，则连接按钮显示为

，而不是

。

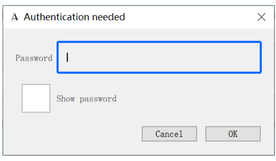
（1）如果没有连接到真实的机器人，确保机器人处于打开状态，选择“Robot” > “Connect to/disconnect from Robot”，或者点击

按钮。

（2）确保端口为**9503**。

（3）输入机器人 IP 地址（按一下Pepper胸部按钮会播报IP地址），点击“Connect”。

（4）输入机器人密码，点击“OK”。



此时连接按钮会显示为

。

将 Animation Editor 连接到真实的机器人后，您可以从 Animation Editor 控制真实的机器人：

5 使用 Animation 的注意事项

1、Pepper 在执行 Animation 时会自动避障，如果障碍物阻止他安全地完成整个 Animation，则 Animation 将部分执行。在执行轨迹时，如果Pepper在途中遇到障碍物，他将避开障碍物并尝试尽可能准确地执行轨迹，但是 Animation 的总时间保持不变。

2、当已经有一个 Animation 在运行的时候，无法同时运行另一个 Animation。Pepper 有很多 API 是涉及 Animation 的，比如 Say（默认情况下没有禁用 BodyLanguageOption）、LookAt，所以当调用这些 API 的时候，是无法执行 Animation 的。