

RoboWare Viewer 软件使用手册

版本：0.10.38

时间：2017-09-21

目 录

1	介绍.....	3
1.1	编写目的.....	3
1.2	软件特性.....	3
2	软件安装.....	4
2.1	准备	4
2.2	安装	4
3	软件使用教程	4
3.1	指令区指令.....	5
3.2	组件区功能.....	5
4	生成独立的应用	13

1 介绍

1.1 编写目的

编写 RoboWare Viewer 使用说明的目的是充分叙述 RoboWare Viewer 所能实现的功能及其运行环境，以便 RoboWare Viewer 用户了解本软件的使用范围和使用方法，并为 RoboWare Viewer 软件的维护和更新提供必要的信息。

1.2 软件特性

RoboWare Viewer 是一个无需 coding 只需拖拽即可创建 ROS 界面的开发环境。它使 ROS 界面开发更加直观、简单，并且易于操作。

RoboWare Viewer 易于安装及配置，下载后双击即可安装，如需应用内编译全平台应用只需配置 NodeJS 开发环境即可，无需其它额外配置。这种“开箱即用”的特性能够帮助开发者迅速上手。

2 软件安装

2.1 准备

1. 安装 NodeJS 环境

- (1) 下载 NodeJS (<https://nodejs.org/en/download/current/>)
- (2) Windows 和 Mac 用户直接下载双击安装即可。
- (3) Linux 用户可以下载安装包安装，也可以使用包管理器（建议）进行安装
(<https://nodejs.org/en/download/package-manager/>)

2. ROS Master 所在的机器上安装中间节点

- (1) 安装 mongodb（如果机器上已安装，可略过此步骤）

```
apt-get install mongodb
```

- (2) 安装 Roboware_Viewer_Bridge

- clone 项目：

```
git clone https://github.com/tonyrobotics/RoboWare-Viewer-Bridge.git
```

- 进入 RoboWare-Viewer-Bridge 目录

```
cd RoboWare-Viewer-Bridge
```

- 安装依赖

```
npm install
```

- (3) 安装 rosbridge_suite

- 进入 ROS 工作空间的 src 目录

```
cd ~/catkin_ws/src
```

- 安装 rosbridge_suite 节点

```
sudo apt-get install ros-kinetic-rosbridge-suite
```

3. 启动

- (1) 启动 mongodb

```
/etc/init.d/mongodb start
```

- (2) 启动 roboware_viewer_bridge
node index.js
- (3) 启动 rosbridge_server rosbridge_websocket
roslaunch rosbridge_server rosbridge_websocket.launch

4. 网络

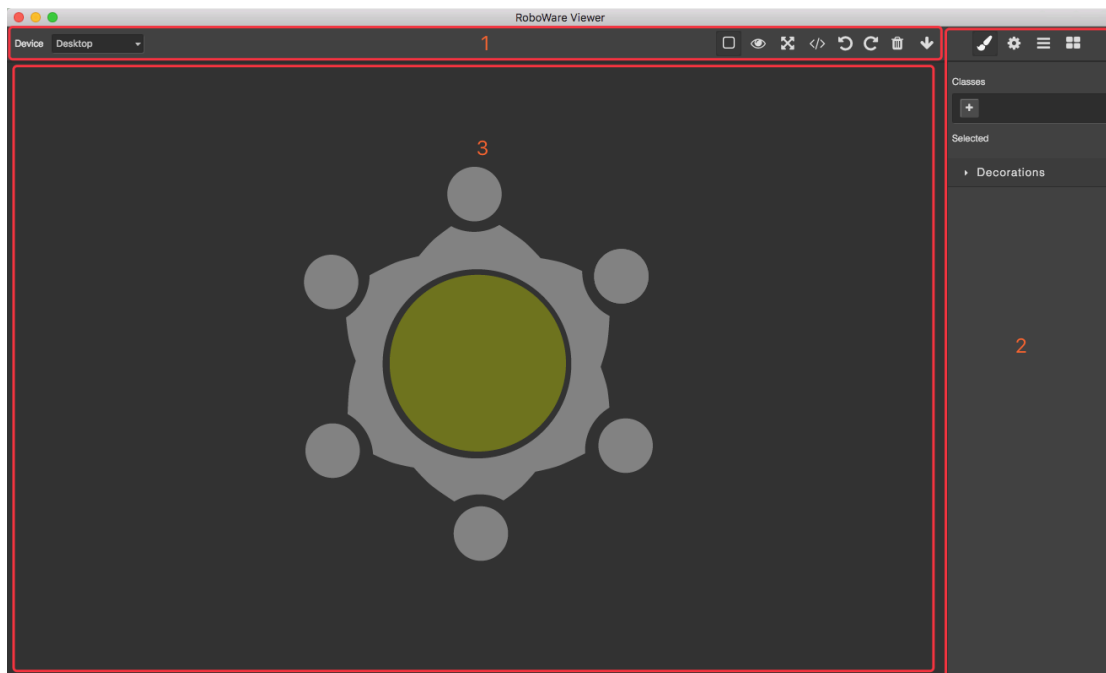
请确保本地机器 RoboWare Viewer 可以与 ROS master 通信。对于详细的教程，请参考以下网站: <http://wiki.ros.org/ROS/Tutorials/MultipleMachines>

2.2 安装

下载对应操作系统 RoboWare Viewer 最新版，双击下载的文件即可完成安装。Linux 用户可在终端执行以下命令进行安装：sudo dpkg -i xxx (xxx 表示下载的文件)

3 软件使用教程

RoboWare Viewer 打开后，如下图：



如图所示，分为 3 个区域

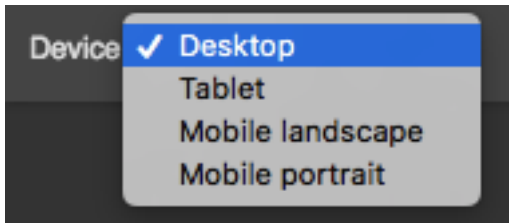
区域 1，指令区

区域 2, 组件区

区域 3, 内容区

3.1 指令区指令

设备：



可以以如上图所示 4 种设备类型进行查看内容区内容。

其余如下：



依次为，查看组件，预览，全屏，显示代码，撤销，恢复撤销，清除所有内容，编译

3.2 组件区功能

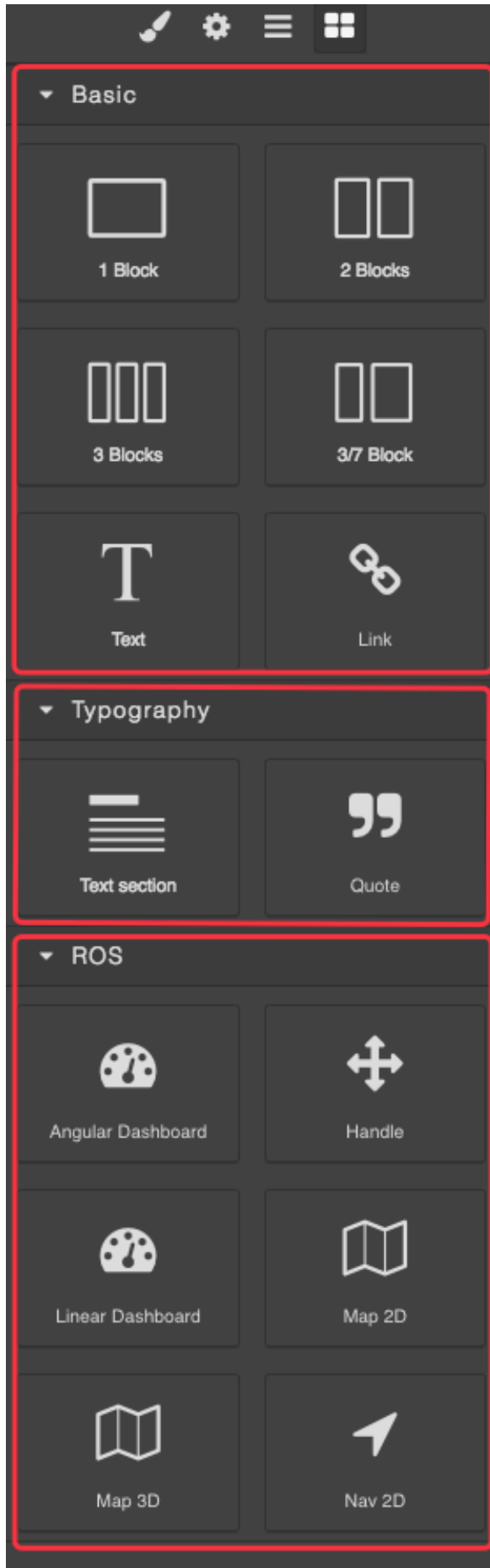
组件区有 4 种管理面板，如下图：



样式管理器，设置管理器，层次管理器，组件管理器

其中，样式管理器，设置管理器和层次管理器都是作用于组件管理器中的组件的。

1 组件管理器

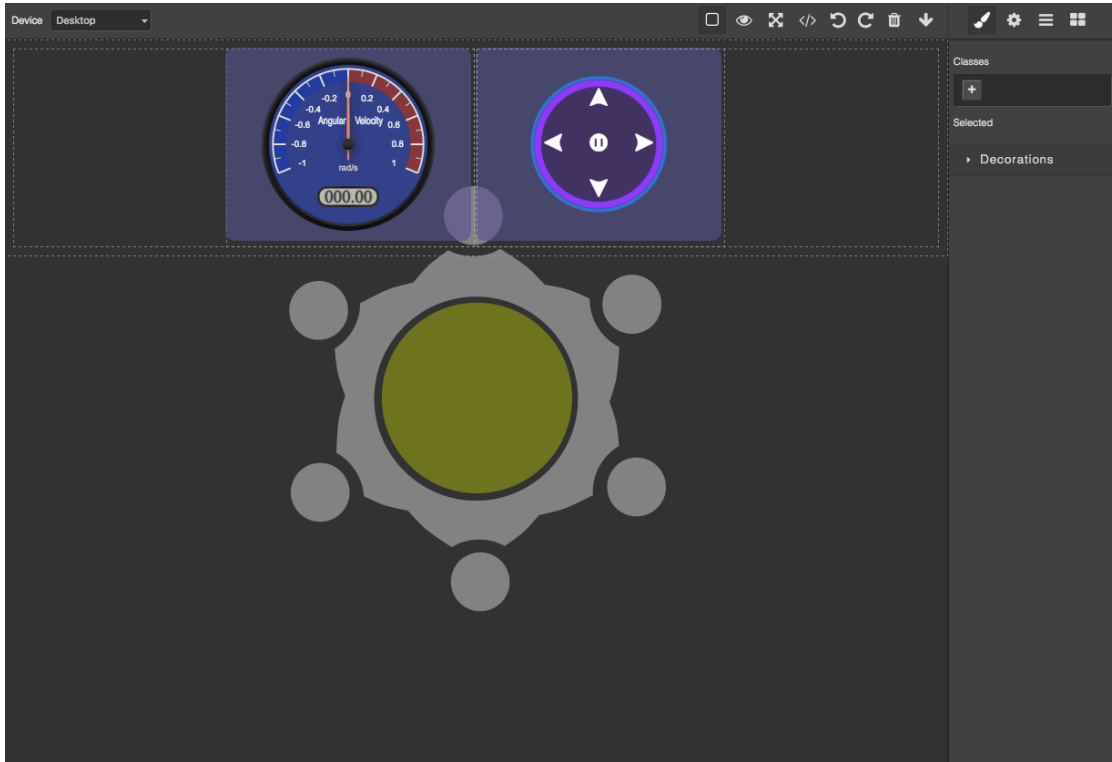


如上图所示，组件管理器中组件分三部分，基础组件，排版组件和 ROS 组件。

使用以上组件只需把他们拖入到内容区即可。

下面我们拖入内容区 “Angular Dashboard” 和 “handle” 两个组件来详细说明。

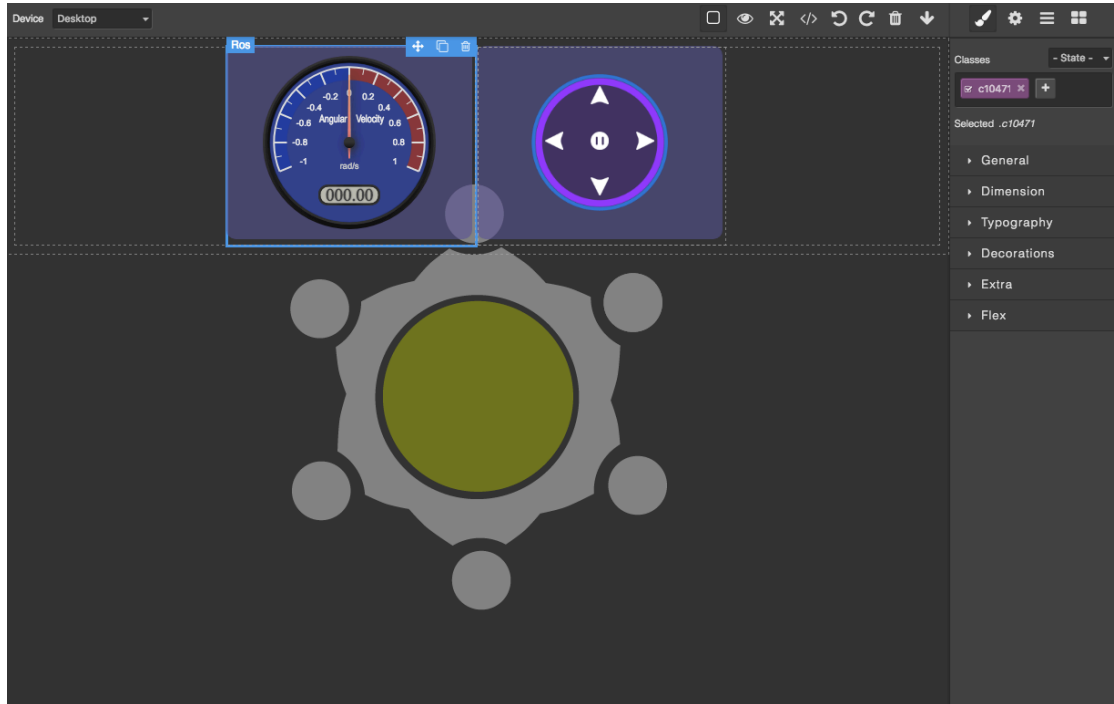
如下图：



5. 样式管理器

样式管理器可以很方便直观的的配置每个组件的任一样式。

当我选中其中一个组件后，样式管理器内容发生变化，如下图



可以看到，当前选中的组件 classes 为“c10471”，另一个组件的 classes 为“c10931”，该组件有 General , Dimension , Typography , Decorations , Extra , Flex 6 类属性。详细属性分别为：

▼ General

Float

none

left

right

Display

block

Position

static

relative

absolute

fixed

Top

0

px

Right

0

px

Left

0

px

Bottom

0

px

▼ Dimension

Width

275

px

Height

220

px

Max width

auto

Min height

auto

Margin

Margin top

0

px

Margin right

0

px

Margin bottom

0

px

Margin left

0

px

Padding

Padding top

0

px

Padding right

0

px

Padding bottom

0

px

Padding left

0

px

济南汤尼机器人科技有限公司 (www.tonyrobotics.com)

▼ Typography

Font family

Font size

16 px

Font weight

Letter spacing

normal

Color

rgb(0, 0,)

Line height

normal

Text align

Text shadow

▼ Decorations

Border radius

0 px

Background color

none

Border radius

Border top left radius

0 px

Border top right radius

0 px

Border bottom left radius

0 px

Border bottom right radius

0 px

Border

Border width

medium px

Border style

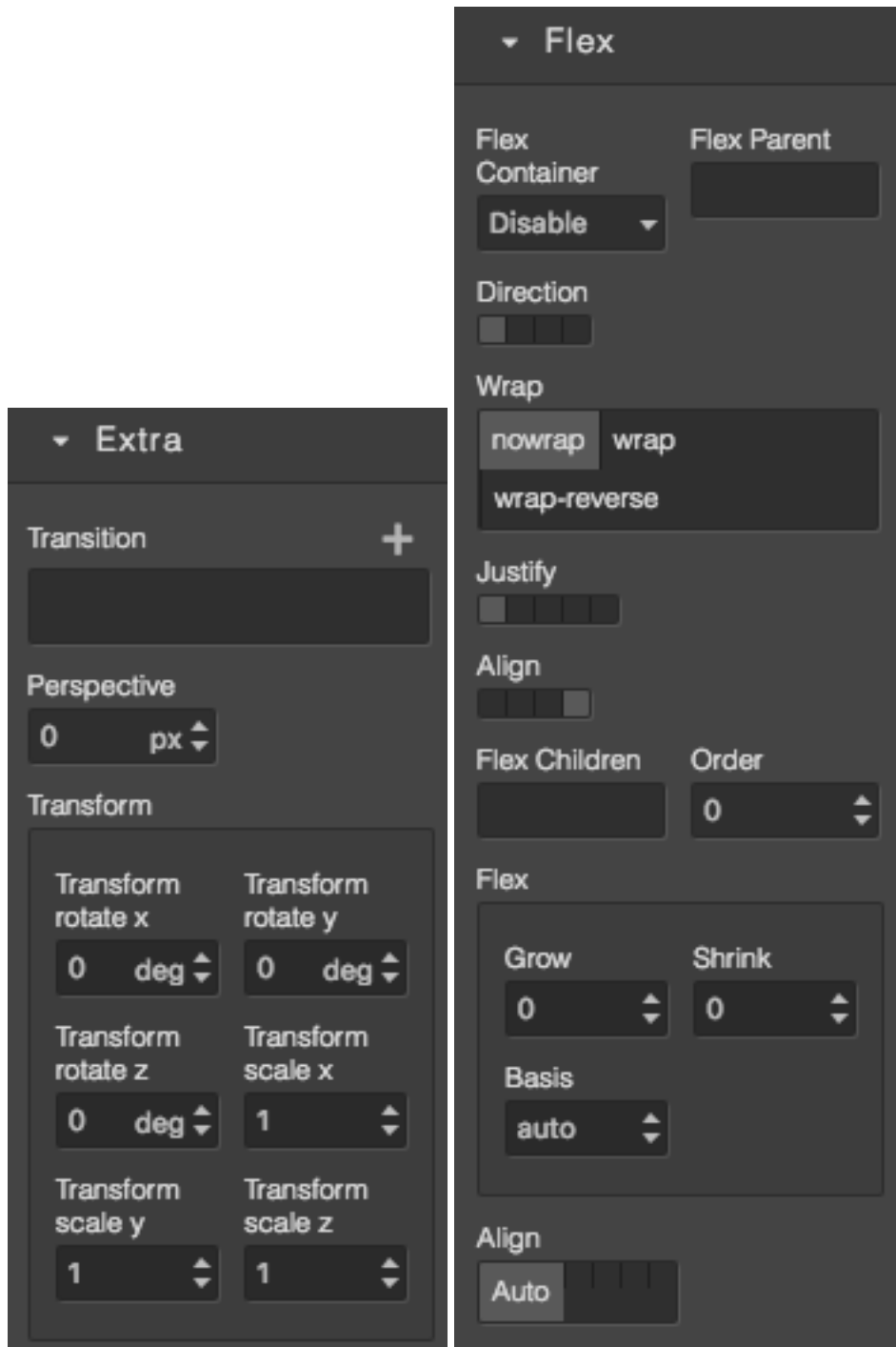
solid

Border color

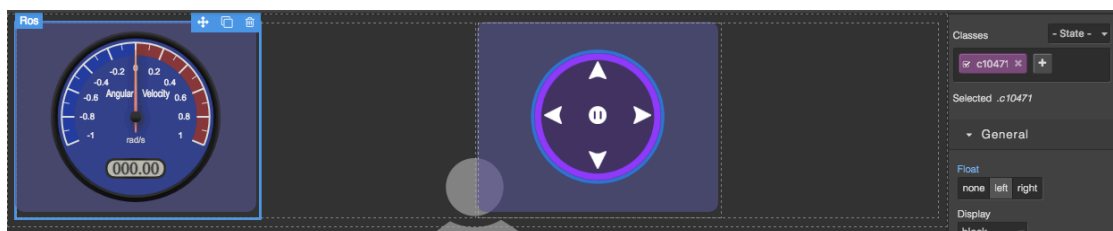
black

Box shadow

Background



此时，修改选中组件 General 属性下的 Float 属性为 left, 则，



该组件靠左显示。但是另一个组件并未有任何响应。

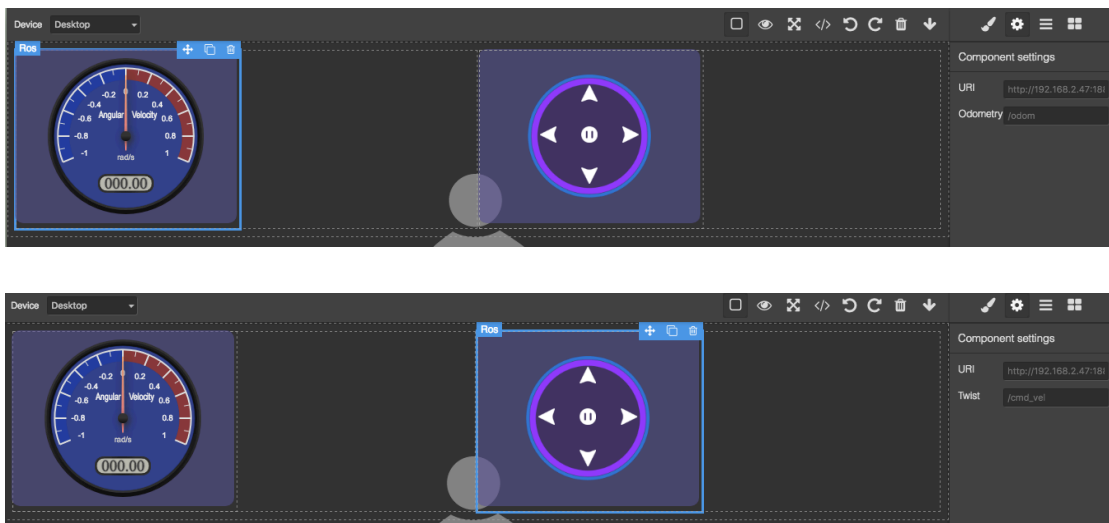
此时，将两个组件的 `classes` 都改为“`c10471`”，此时选中任意一个组件，再改变 General 下的 Float 属性，发现两个组件会有相同的响应。

所以，Classes 控制组件的样式，而样式管理器修改 Classes，从而对组件造成影响。

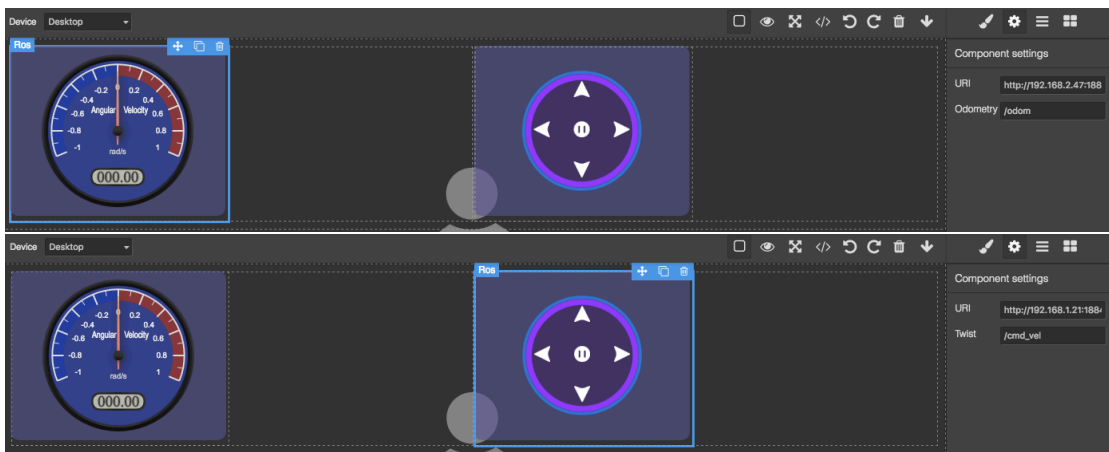
6. 设置管理器

设置管理器可以修改组件内参数，且不同组件显示参数不同。

和样式管理器不同的是，设置的参数只作用于当前选中的组件。，如下图



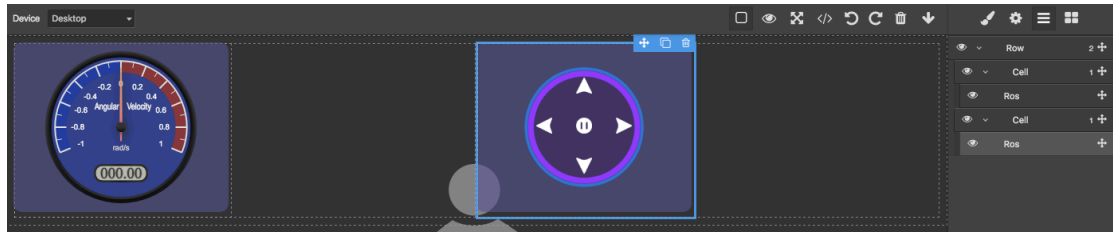
分别选中两个组件，发现设置的参数不同，分别设置后：



根据 URL 可见，参数不会互相影响。

7. 层次管理器

在使用组件时，您可以尽情的嵌套组件。层次管理器可以方便的管理、排列您的层次。

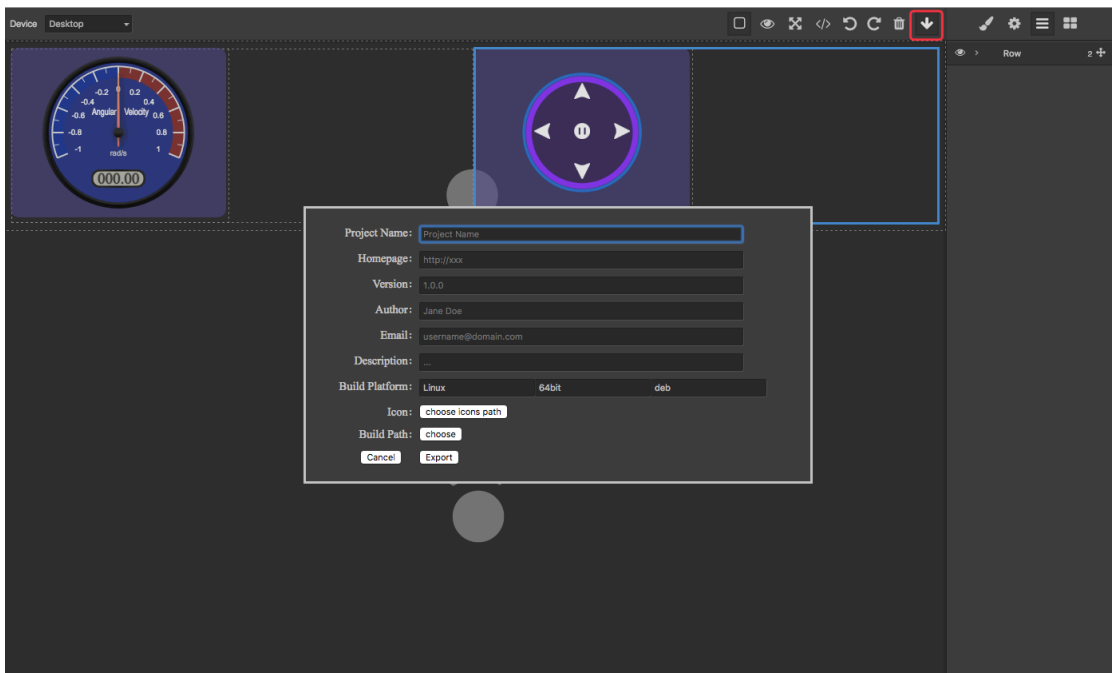


4 生成独立的应用

当您创建好您满意的 ROS 界面后，您还可以把它打包为 Linux、Windows、MacOS 下的独立应用。

生成步骤

1. 选择指令区内的编译按钮，会弹出如下界面



2. 按照提示依次填入内容，

其中，Build Platform，可选平台为 Linux, Windows 和 MacOS 三个平台。

每个平台都可以选择 64/32 位架构

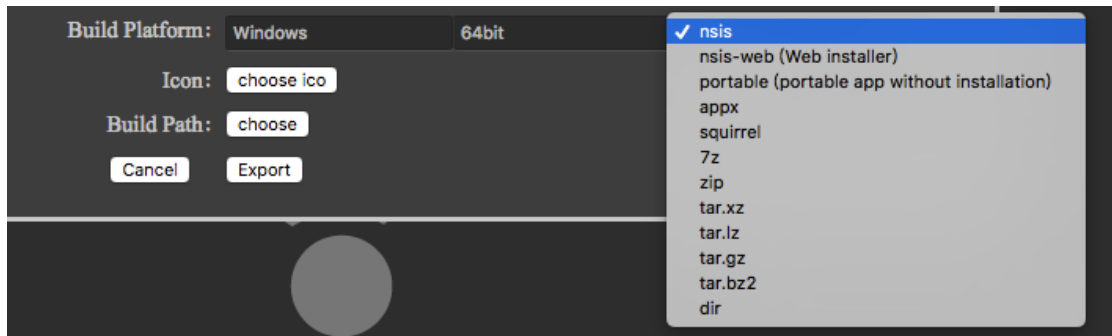
编译的目标根据平台的不同而不同：

选择 Linux 平台可以编译为

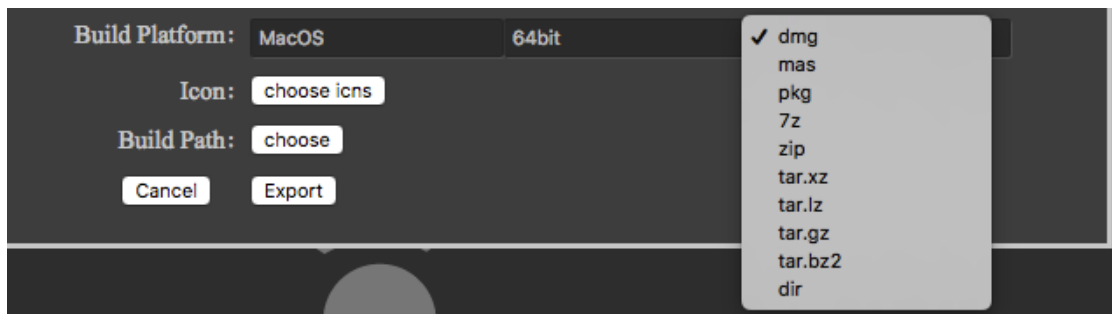
deb, AppImage, snap, rpm, freebsd, pacman, p5p, apk, 7z, zip, tar.xz, tar.lz, tar.gz, tar.bz2, dir
15 种目标。

选择 Windows 下可以编译为 nsis(exe), nsis-web, portable, appx, squirrel, 7z, zip,

tar.xz,tar.lz,tar.gz,tar.bz2,dir 12 种目标

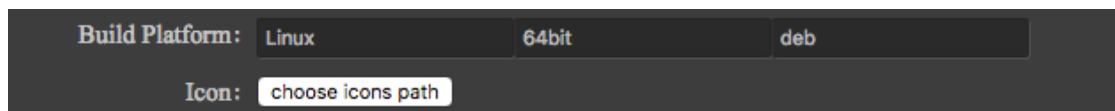


选择 MacOS 下可以编译为 dmg,mas,pkg,7z,zip, tar.xz,tar.lz,tar.gz,tar.bz2,dir 10 种目标。

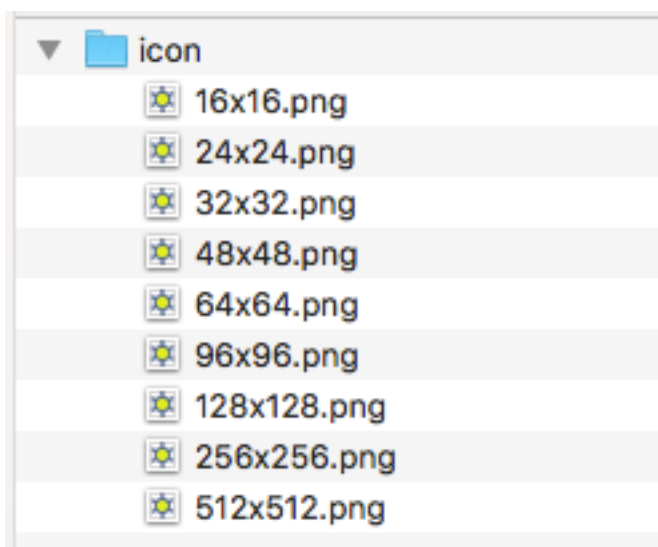


其中，Icon 的选择，

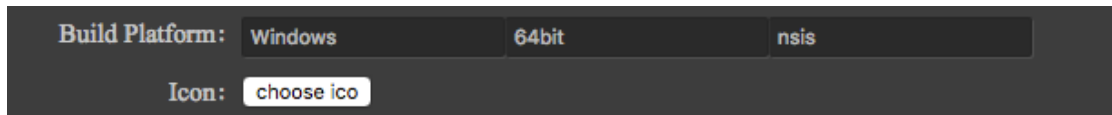
当 Build Platform 为 Linux 时，Icon 必须选择文件夹，



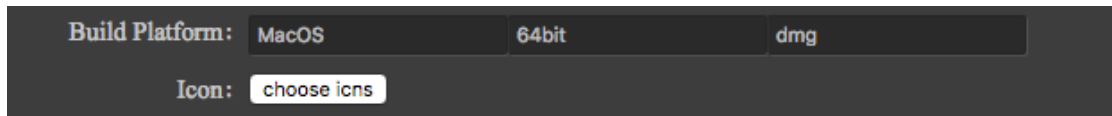
文件夹内容必须为不同尺寸的 png 图片：



当 Build Platform 为 Windows 时，Icon 必须选择 .ico 文件，

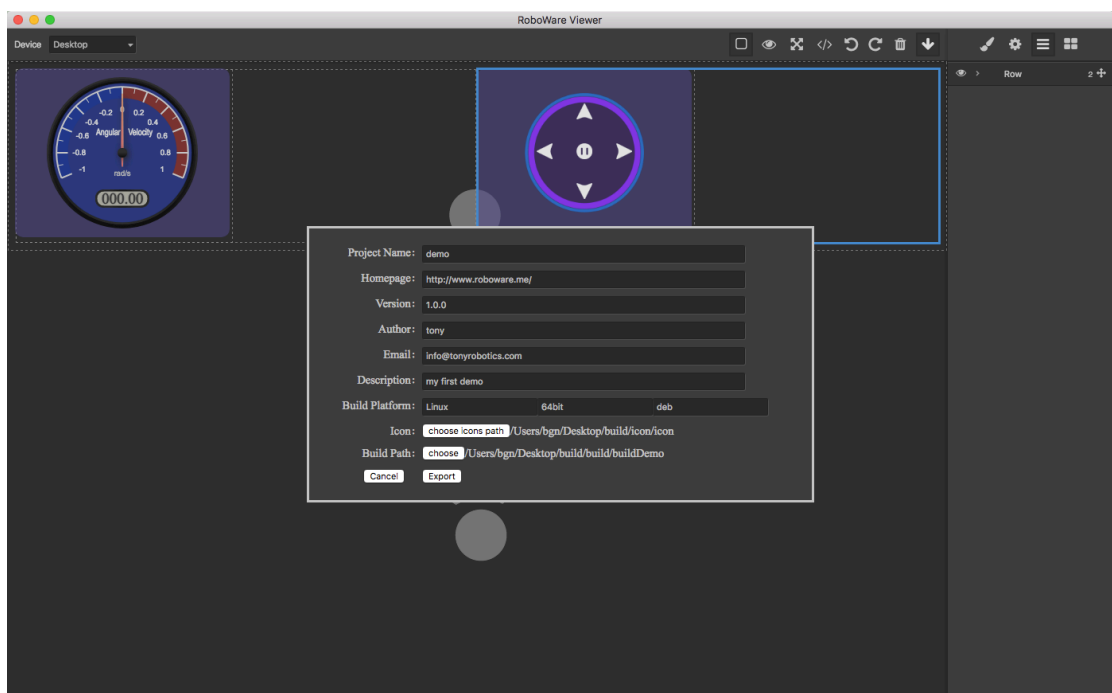


当 Build Platform 为 MacOS 时, Icon 必须选择.icns 文件



其中 Build Path 的选择, 将选择您要编译到的文件夹下。尽量不要选择带中文的文件夹, 选择的文件夹当前用户必须有读写权限。

最终填写信息如下:



点击 Export 按钮

会在选中的 Build Path 文件夹内以 build_\${Project Name}_\${Author}_时间戳 的格式生成文件夹和必要的 build 文件, 如下图:

buildDemo	今天 下午4:01	--	文件夹
build_demo_tony_1505721701848	今天 下午4:01	--	文件夹
apps	今天 下午4:01	--	文件夹
icon	今天 下午4:01	--	文件夹
node_modules	今天 下午4:01	--	文件夹
build.bat	今天 下午4:01	106 字节	Visual...pp 文稿
build.sh	今天 下午4:01	116 字节	Shell Script
icon.icns	今天 下午3:52	222 KB	Apple 图标图像
icon.ico	今天 下午3:52	370 KB	Windo...标图像
index.js	今天 下午3:52	1 KB	JavaScript
package.json	今天 下午4:01	750 字节	JSON

Mac 和 Linux 用户执行 build.sh 文件, Windows 用户执行 build.bat 文件, 即可生成独立的应用。

执行完毕后, 会在当前目录生成独立的应用程序, 如下图:

