

# Webots 开发简易教程

---

本文包括有 Webots 开发的简易教程

由于学习目的主要为修改原来的代码

以从webots 中拿出摄像头图像进行处理

所以本文不会面面俱到，只会侧重于相关的 API

所使用语言为 C ++

## Webots 开发简介

---

核心是节点开发

项目中每个节点都有其对应的API

controller 需要做的本质就是将机器人再包装出来

然后在 run() 中放入主业务逻辑

通信层是由simulation-sdk 负责的，由qi::Application 调用

能做更改的只有 webots 相关代码

参数传递：

- 依赖于每一个robot下，controllerArgs 参数

## Webots Camera 相关 API

---

include

```
#include <webots/Camera.hpp>
```

APIs

```
namespace webots {  
  class Camera : public Device {  
    virtual void enable(int samplingPeriod);  
    virtual void disable();  
    int getSamplingPeriod() const;  
    // ...  
    int getwidth() const;  
    int getHeight() const;  
    // ...  
    const unsigned char *getImage() const;
```

```

        static unsigned char imageGetRed(const unsigned char *image, int width, int
x, int y);
        static unsigned char imageGetGreen(const unsigned char *image, int width,
int x, int y);
        static unsigned char imageGetBlue(const unsigned char *image, int width, int
x, int y);
        static unsigned char imageGetGray(const unsigned char *image, int width, int
x, int y);
        // ...
        int saveImage(const std::string &filename, int quality) const;
        // ...
    }
}

```

其中使用到的应该只有 getImage 和 saveImage

Plus：获取某一个camera名字的方法：可以使用Camera 类从 Device 类继承的 name() method

## Webots 和 vs2019 联调

方法其实不难

首先在已经 build 完成整个项目之后

运行 webots · 启动对应的 controller

然后Pause整个进程 · 用 VS2019 的 `Debug->Attach to Process`

Attach 到进程 `MyController.exe_webots`

此时可以添加断点进行调试

提示：

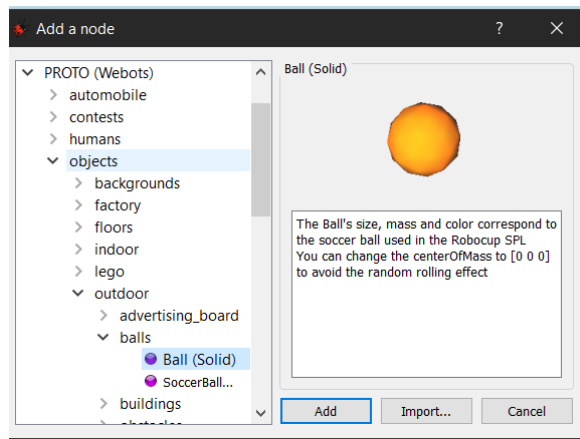
现在可以用这套方式看看以前为什么

自己编译出来的 controller 会 crash

找出来的原因是：应该使用VS2010 进行编译 · VS2019的SDK过新不兼容

## Webots 物体

球体



小车建模：

<https://blog.csdn.net/crp997576280/article/details/105537639>

## 补充：Webots 更改摄像头清晰度

---

因为现实中 nao 的分辨率是 1280x960

但是仿真中默认为 160x120

所以在代码中 Camera 类的初始化里我做了一点改变

## Appendix

---

### 学习资料

- 官方开发文档：<https://cyberbotics.com/doc/guide/controller-programming#using-controller-arguments>
- 官方node 手册：<https://cyberbotics.com/doc/reference/node-chart?tab-language=c++>