

Webots搭建麦轮小车

19335112 李钰

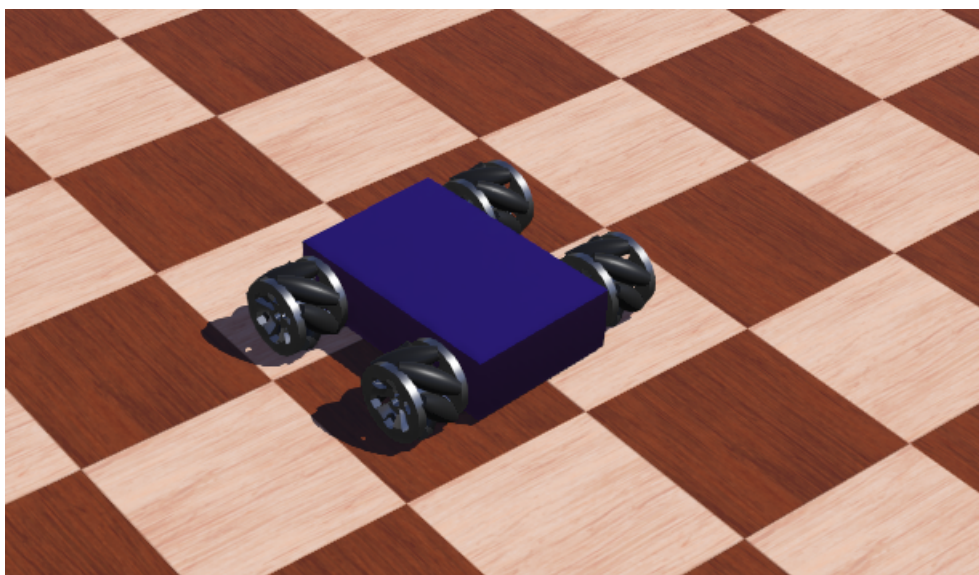
一. 实验目标

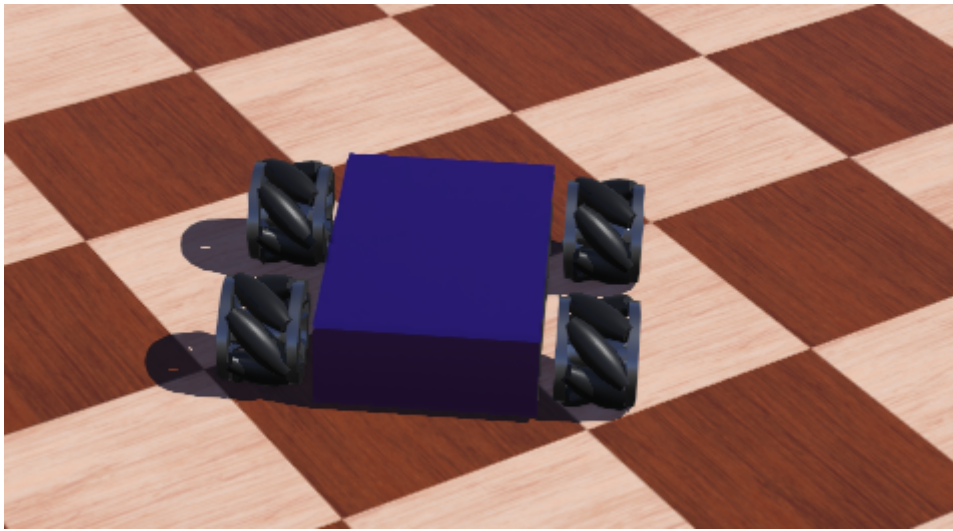
使用webots仿真软件，搭建仿真的场景（地面、光照），搭建麦轮小车，然后写一个控制器，控制麦轮小车能够八向移动（前后左右斜）和自旋。

二. 实验内容与步骤

1. 新建一个世界，修改世界坐标系：在WorldInfo中的coordinateSystem"ENU"中选择ENU选项，将世界坐标系改为z轴指向天空的右手坐标系。之后保存并重启。
2. 设置光源，添加直射和漫反射光源，调整视角，为了方便实验这里我使用的是top view来俯视整个小车。
3. 添加地面，设置大小、位置等参数。由于刚刚改了坐标系，这是地面还是以y轴的负方向为重力方向，所以我们给地板旋转一个角度
4. 添加机器人结点
 1. 修改外观，并且设置车体大小（ $x=0.3$ $y=0.2$ $z=0.08$ ）
 2. 定义物理碰撞边界以及给它物理属性，使其受重力作用影响
 3. 之后就是为车体安装铰链，一端连着电机，另一端连着轮子。这里我们采用主教提供的模型，在robot的child中添加节点时直接选择模型的轮子。
 4. 一共要添加4个轮子，修改相应的位置坐标，以及电机的名字。修改电机名字十分重要，因为在控制文件中，我们对电机的名字有引用，需要保持一致性。
5. 最后添加控制器文件，并将robot的controller选择到控制器文件。然后编译运行，小车即可以前后左右行进，也可以作右旋和左旋的操作。

三. 实验结果与分析





四. 实验中的问题和解决方法

1. 添加铰链那里一开始没有弄清楚铰链、电机、轮子之间的关系，children之间的关系很混乱，甚至还出现了轮子下又增加了一个轮子的情况。后来理清条理之后，就顺利许多。
2. 结点的名字很重要，新建一个结点就要及时为其命名，不然影响后续使用。

控制器代码

```
#include <webots/Robot.hpp>
#include <webots/Motor.hpp>
#include <webots/Keyboard.hpp>
#include <iostream>
#include <algorithm>
#include <iostream>
#include <limits>
#include <string>

using namespace std;
using namespace webots;

int main() {
    Motor *motors[4]; //电机和键盘都要用webots给的类型
    webots::Keyboard keyboard;
    char wheels_names[4][8] = { "motor1", "motor2", "motor3", "motor4" }; //你在仿真器
    里面设置的名字

    Robot *robot = new Robot(); //使用webots的机器人主体
    keyboard.enable(1); //运行键盘输入设置频率是1ms读取一次

    double speed1[4];
    double speed2[4];
    double v = 10;

    //初始化
    for (int i = 0; i < 4; i++)
    {
        motors[i] = robot->getMotor(wheels_names[i]); //按照仿真器里面设置的名字获取句
        柄

        motors[i]->setPosition(std::numeric_limits<double>::infinity());
        motors[i]->setVelocity(0.0);
    }
}
```

```

        speed1[i] = 0;
        speed2[i] = 0;
    }

    double forward[4] = { v ,v ,v ,v };
    double backward[4] = { -v ,-v ,-v ,-v };
    double leftward[4] = { v ,-v ,v ,-v };
    double rightward[4] = { -v ,v ,-v ,v };

    double leftCircle[4] = { v ,-v ,-v ,v };
    double rightCircle[4] = { -v ,v ,v ,-v };

    int timeStep = (int)robot->getBasicTimeStep();//获取在webots设置一帧的时间
    cout << timeStep << endl;
    //仿真运行一帧
    while (robot->step(timeStep) != -1) {
        //获取键盘输入，这样写可以获得同时按下的按键（最多支持7个）
        int keyValue1 = keyboard.getKey();
        int keyValue2 = keyboard.getKey();
        cout << keyValue1 << ":" << keyValue2 << endl;

        //根据按键决定电机怎么样转动
        if (keyValue1 == 'w'){
            for (int i = 0; i < 4; i++){
                speed1[i] = forward[i];
            }
        }else if (keyValue1 == 's'){
            for (int i = 0; i < 4; i++){
                speed1[i] = backward[i];
            }
        }else if (keyValue1 == 'a'){
            for (int i = 0; i < 4; i++){
                speed1[i] = leftward[i];
            }
        }else if (keyValue1 == 'd'){
            for (int i = 0; i < 4; i++){
                speed1[i] = rightward[i];
            }
        }else if (keyValue1 == 'q'){
            for (int i = 0; i < 4; i++){
                speed1[i] = leftCircle[i];
            }
        }else if (keyValue1 == 'e'){
            for (int i = 0; i < 4; i++){
                speed1[i] = rightCircle[i];
            }
        }else{
            for (int i = 0; i < 4; i++){
                speed1[i] = 0;
            }
        }

        if (keyValue2 == 'w'){
            for (int i = 0; i < 4; i++){
                speed2[i] = forward[i];
            }
        }else if (keyValue2 == 's'){

```

```
        for (int i = 0; i < 4; i++){
            speed2[i] = backward[i];
        }
    }else if (keyValue2 == 'A'){
        for (int i = 0; i < 4; i++){
            speed2[i] = leftward[i];
        }
    }else if (keyValue2 == 'D'){
        for (int i = 0; i < 4; i++){
            speed2[i] = rightward[i];
        }
    }else{
        for (int i = 0; i < 4; i++){
            speed2[i] = 0;
        }
    }
    //让电机执行
    for (int i = 0; i < 4; i++)
    {
        motors[i]->setVelocity(speed1[i] + speed2[i]);
    }
}
return 0;
}
```