

Autonomous Surface Vehicle Simulator 使用说明

Version2.0 BY HeZe at 2020/06/21

一、运行环境与软件准备

- macOS 、 Windows 、 Linux 均可以运行
- 需要下载软件 JetBrains 的 IntelliJ 以及 JDK8 , 注意版本要正确
- JDK8 下载地址: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- 下载、安装完毕后, 按照视频 <https://youtu.be/weAlRonzOd0> 提供的方法配置环境以及编译运行 (压缩包里也有这个视频)

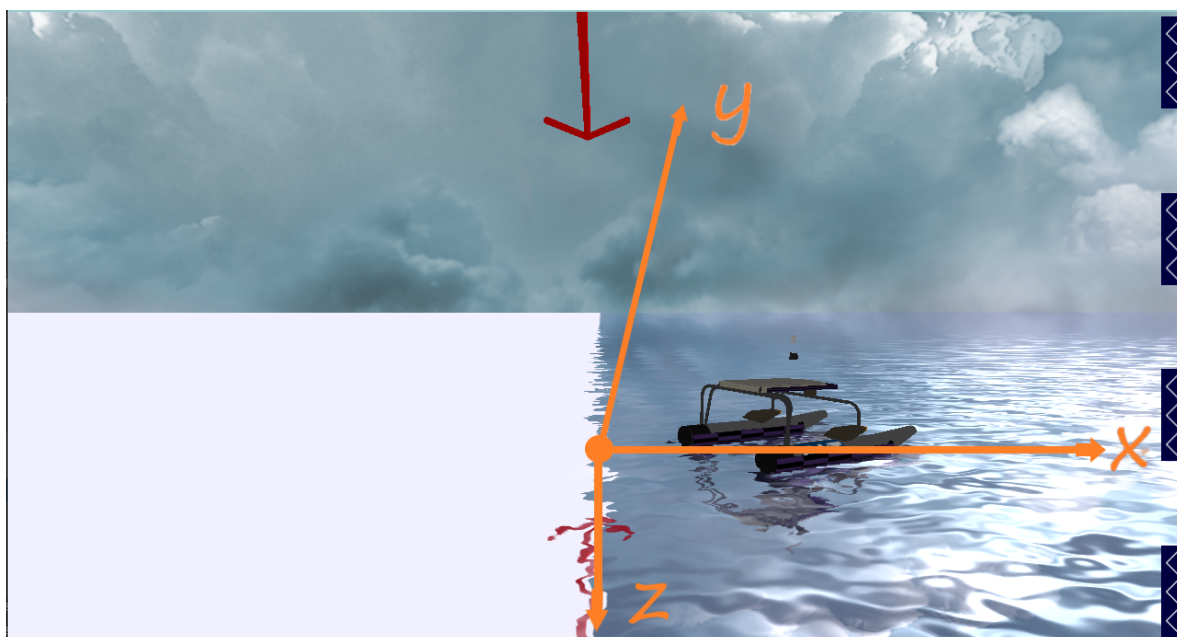
需要注意以下几点:

- 视频里的代码是在 GitHub 下载, 因为我做了改动, 所以直接解压我的压缩包之后记住文件夹路径即可, 可跳过视频前一小段下载代码的过程
- 有一步是根据自己的操作系统选择不同的文件, 需要留意

二、仿真使用说明

1. 坐标系

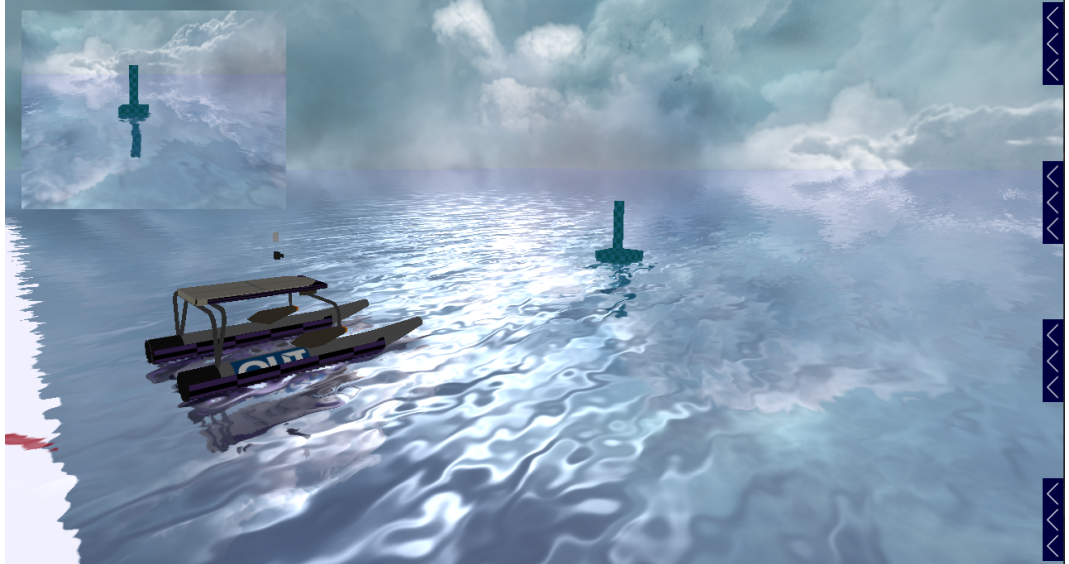
如下图:



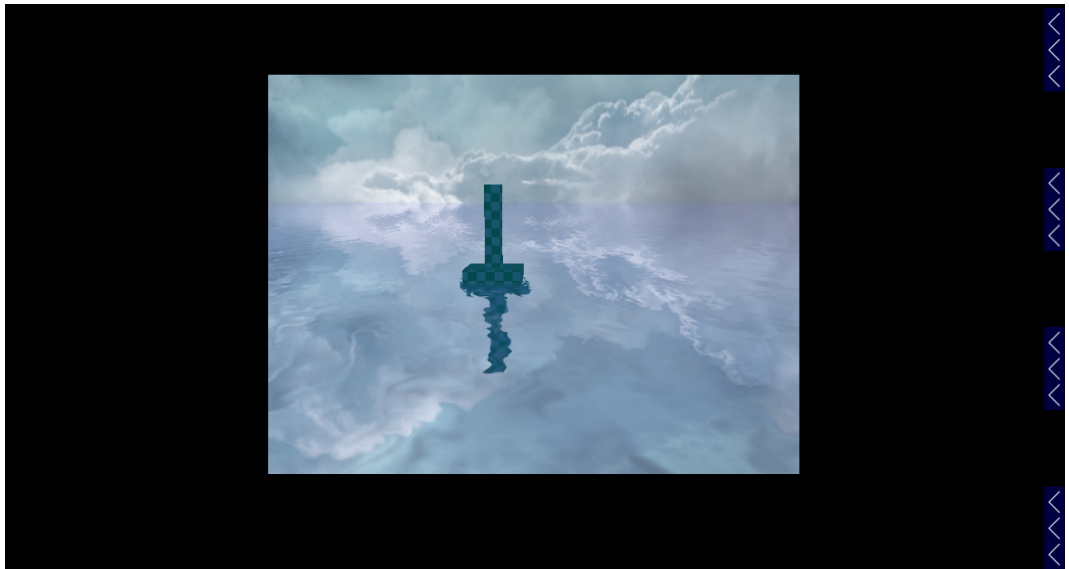
初始化的时候 $x=5$

2. 图形界面功能

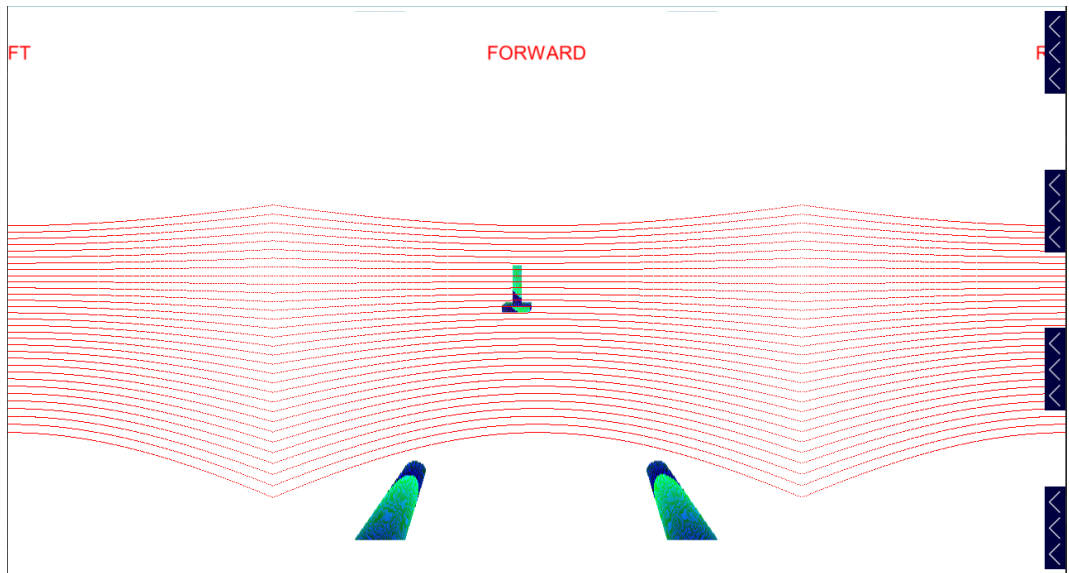
- 按住鼠标右键拖动可以改变视角
- 键盘a,s,d,w,q,e可以将镜头角度、方位改变
- 按下空格键可以循环显示以下模式
 - 上图的主界面
 - 左上角显示相机影像



- 纯相机影像



- 雷达图像



- 右侧有四个条形方块，点击后每一个菜单都可以进行不同的控制，如调节船的速度、位置以及环境（如水速、水向等等）等等，由于字面意思已经能表达功能，这里不再叙述
- 对于障碍物的添加也是在侧栏菜单中，有8种形状可以选择，同时可以设置位置、大小、颜色、各种转动等等

3.控制台功能

此版本支持以下命令：

- `position`
输入 `position`，返回三元坐标位置信息，`y` 为船的高度，默认值为 `0.8`，随水流上下波动，返回格式为 `(x,y,z)`
- `speed`
输入 `speed`，返回三元坐标速度信息，`y` 为上下速度，随水流上下波动会有变的速度，返回格式为 `(x,y,z)`
- `distance`
输入 `distance`，返回与 `(0, 0, 0)` 的距离，返回格式是一个浮点数
- `rotation`
输入 `rotation`，返回转动四元组向量

下面命令控制船运动，此仿真船的运动靠左右两个马达

- `setleft`
输入“ `setleft+空格+数值` ”，如“ `setleft 50` ”，代表将左马达数值设为 `50`
- `setright`
输入“ `setright+空格+数值` ”，如“ `setright 50` ”，代表将右马达数值设为 `50`
- `setall`
输入“ `setall+空格+数值` ”，如“ `setleft 50` ”，代表将左右马达数值同时设为 `50`
- `setspeed`
输入“ `setspeed+空格+数值1+空格+数值2` ”，如“ `setspeed 50 60` ”，代表将左马达数值设为 `50`，右马达数值设为 `60`
- `reset`

重置船的位置并将马达数值设为 0

雷达与相机

- 由于目前还没有和其他同学合作跑算法，每个人设计的算法需要的结构可能都不一样，所以在跑算法的时候需要因人而异重新设计这部分的数据结构
- 那么现在我设计的是这样的，不论是雷达还是相机，输入两个0至1之间的值，代表想测量的点在x、y的比例，调用相机则返回这一点的rgb值，具体格式为 (R, G, B)，调用雷达则返回这点的距离，注：现在仿真的雷达可以测的距离在1m至70m之间，在这个范围之外的都会返回0

- camera

输入“ camera+空格+x比例+空格+y比例 ”，如“ camera 0.5 0.5 ”，返回最中间这一点的rgb值

- lidar

输入“ lidar+空格+x比例+空格+y比例 ”，如“ lidar 0.5 0.5 ”，返回最中间这一点的距离

另外，如果想对这两个的格式进行改动，粗略的描述如下

- 这两个的调用在主函数的 1676 行，文件路径是 \Autonomous-Surface-Vehicle-Simulator-master\BoatVis\src\gebd\games\boat\BoatVis.java
- 也就是下面这段代码

```
1 public void getPhysicsBoatEntity2(int i,float q,float p) {
2     if (i==11) {
3
4         System.out.printf("%f\n",
5         LidarCalculationHandler.getDepthAtPixelPercentages(q, p));
6     }
7     if(i==12){
8         int qq=(int)(640*q);
9         int pp=(int)(480*p);
10        CameraCalculationThread.aaa(qq,pp);
11    }
```

i==11是雷达，i==12是相机，在IDE里鼠标放到调用的函数上面就可以编辑函数源代码

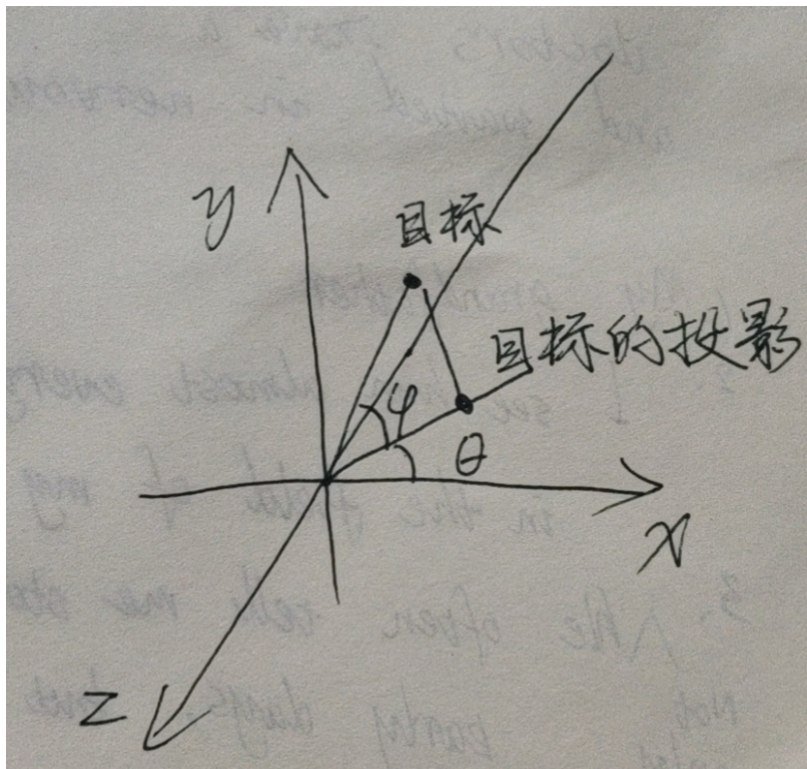
6.21添加

加入了探测点云以及TCP通信

- 点云

输入“ lidarpoint+空格+theta值+空格+phi值 ”，返回该方向障碍物坐标

theta和phi的物理含义如下：



- TCP通信

我设计的是模拟器作为服务器端，需要客户端与仿真连接，由于每个人算法使用的语言不同所以客户端使用的语言也不一致，压缩包里提供了Java的版本 `TCPCClient.java`，如果用Python或C++写也都是可以的，先启动仿真，再启动客户端，在客户端输入和上面一样的格式就可以。