



ROCOS Slide 2

Play 调试

Created by [Mark](#) - [ZJUNlct](#)

课程总纲

- Part1 lua layer
- Part2 lua-c++ interaction
- Part3 c++ layer
- Appendix A debug engine

Task

- 介绍Debug Engine
- 在你的play中调试输出
- 回顾lua的报错信息

Debug Engine

- Client接收绘制
- 支持的类型
 - 文字、点x、线段、圆环、机器人shadow
- 支持的颜色
 - 简易 ： 0-9
 - 复杂 ： RGB值
- 细节文件 ： zss_debug.proto
- 配合 Log Module 使用

调试的使用

- 输出变量的值
- 输出play、state、role、target
- 输出视觉信息 (player.lua enemy.lua ball.lua)
- 输出机器人反馈信息
- 输出路径规划
- *染色图

如何调试你的play

- switch的调用逻辑
- 查看Role来调试match
- 查看目标点判断书写正确性
- 直接打印变量

lua的报错信息如何处理

- 尝试copy其他可用代码并比较报错代码
- 了解task层以及skill层的逻辑
 - 尝试重新封装task层 (task.shoot)
- YOURPLAY.lua / task.lua / play.lua
- 可能的几种错误
 - curPlay、curState相关
 - 参数为nil或类型不对、解包问题
 - 与c++的skill衔接问题