

# python-unity通信逻辑

## 一、通信协议

基于TCP的socket通信模型.

服务端  
与大模型交互

客户端

动画等反应

ServerSocket对象

accept 接请求

创建链接 socket

创建链接 socket 向服务端发信息

相互通信.

关闭 Socket & 相关资源

关闭 Socket & 相关资源

socketplus.py

定义了服务端与客户端的交互命令

main 处理了交互逻辑

AI

YY

Unity

任务安排  
日报生成  
请求

解析

请求

结项报告

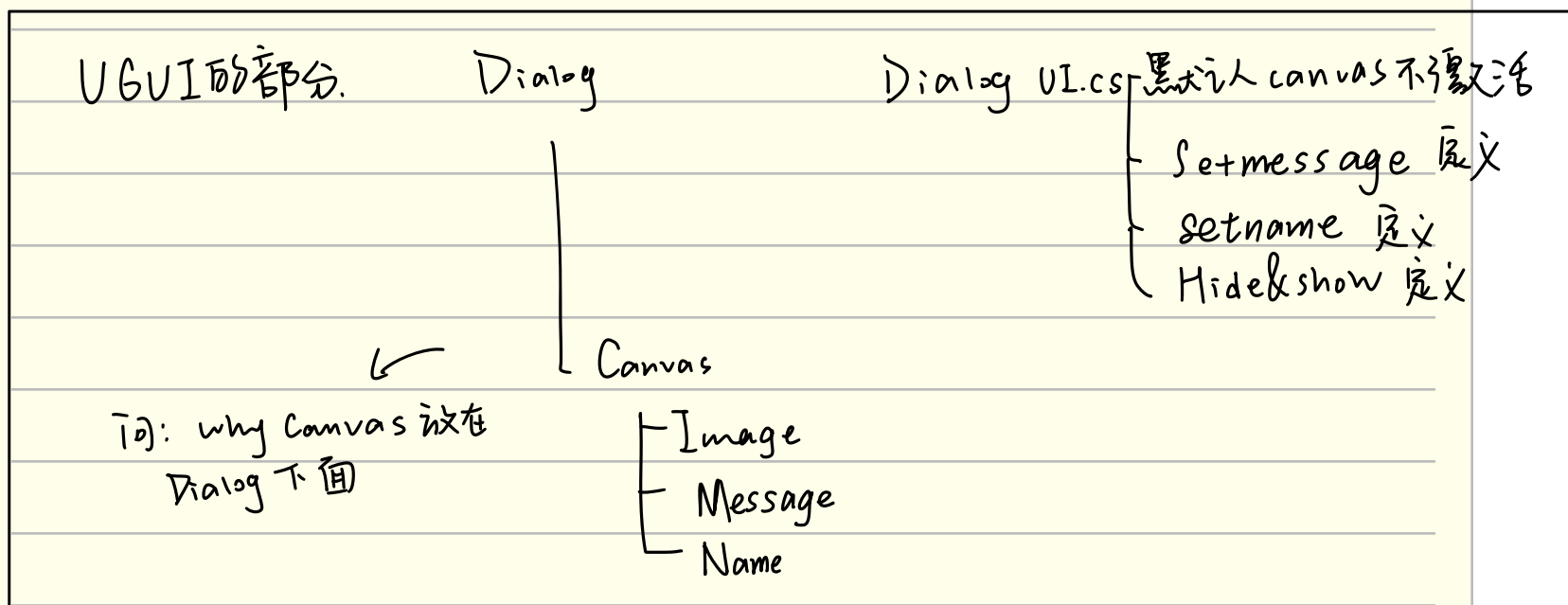
日报/结项报告

# 今日学习任务分解

- ✓ ① 理解 player、NPCs 的脚本组合控制方式 并写成自己程序的调式
- ✓ ② 理解 python、文心之间的通信
- ✓ ③ UGUI 以及
- ④ Unity端对消息的处理的细节

太多了

了解文心 python 库的使用  
跑通 PY端与文心的交互  
理解 Unity端的信息传递/处理



问: why Canvas 放在 Dialog 下面

## 角色挂载的脚本分析

Animator 通过绑定动画器来播放动画

clip Animator Controllers 设定函数, 更方便地通过 Animator 更改参数来影响动画器.

在创建时就已经绑定对象: eg. Object, Sprite 的 Sprite 组件, renderer

NPC AI . cs. 设置目标点让 Animation 动起来, 并实时监测是否到达以及动画的方向

Player 设置操作事件.

character ctrl ler

可以开发事件的触发.

学习:

- ① 互联网的机制
- ② Web的机制)

} NPC

总结



第一阶段的任务

- ① 跑通程序
- ② 做出一定的优化
- ③ 完成上传

SDK的更换.  
分散站位的代码写一下  
NPC随机目标卡顿闪烁的问题  
+ 如果有时间: NPC自由移动.

学习内容: ① Web的基础知识

- ② Unity操作与编程的熟练 ✓
- ③ 了解基本代码的组织逻辑 ✓ 细节有些暂时忽略了.

