介绍 基于http协议的前后端应用 ：

前端使用jq+js+canvas创建动态动画，使用 XMLHttpRequest对象用于在后台与服务器交换数据  ，后端使用flask提供http服务器接口，使用opencv 创建服务器端的游戏画面，使用tensorflow构建机器学习模型并预测，

使用方式：

运行server.py 然后打开浏览器访问本机ip 即可打开测试页面，输入id并开始，5人为一组，当超过设定时间会由机器人替补开始游戏,直致死亡

文件介绍：

Server.py flask接口文件 （主程序）

game 游戏相关文件

robot 机器人模型 （可删除）

saved\_networks 存放模型文件

static js+css+img文件

templates html文件

user 用户模型

utils 常用函数

接口：

/h 获取管道数据

/gp 分组

/pp 匹配

/img 上传画面数据

/getact 获取同步动作

/trainhtml 训练页面

/train 获取训练图片和返回结果

流程：

运行server.py 开启flask接口，并初始化训练网络，以层嵌套的思想初始化用户存储容器，等待前端调用接口。

当浏览器打开index页面，输入id 开始持续请求/pp接口 进行id分组，当服务器获得满足条件的五个用户或者计时超时，便开始当前组的游戏，首先用户前端开始请求/h接口获取当前组的管道初始参数，由前端渲染出画面，然后截取画面压缩上传/img接口，服务器通过请求控制当五个用户全部上传完毕，开放/getact接口，由前端请求获得小鸟的飞行动作，再由前端渲染出来