

第十九届全国大学生智能汽车竞赛设计文档

翻斗花园大队*

*Zhejiang University

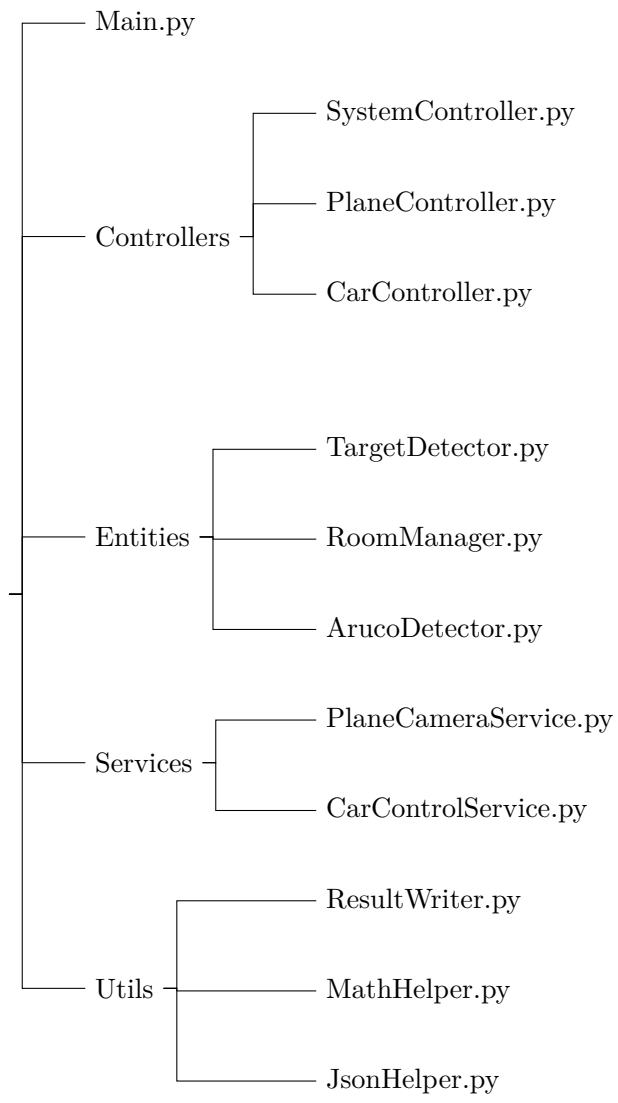
July 22, 2024

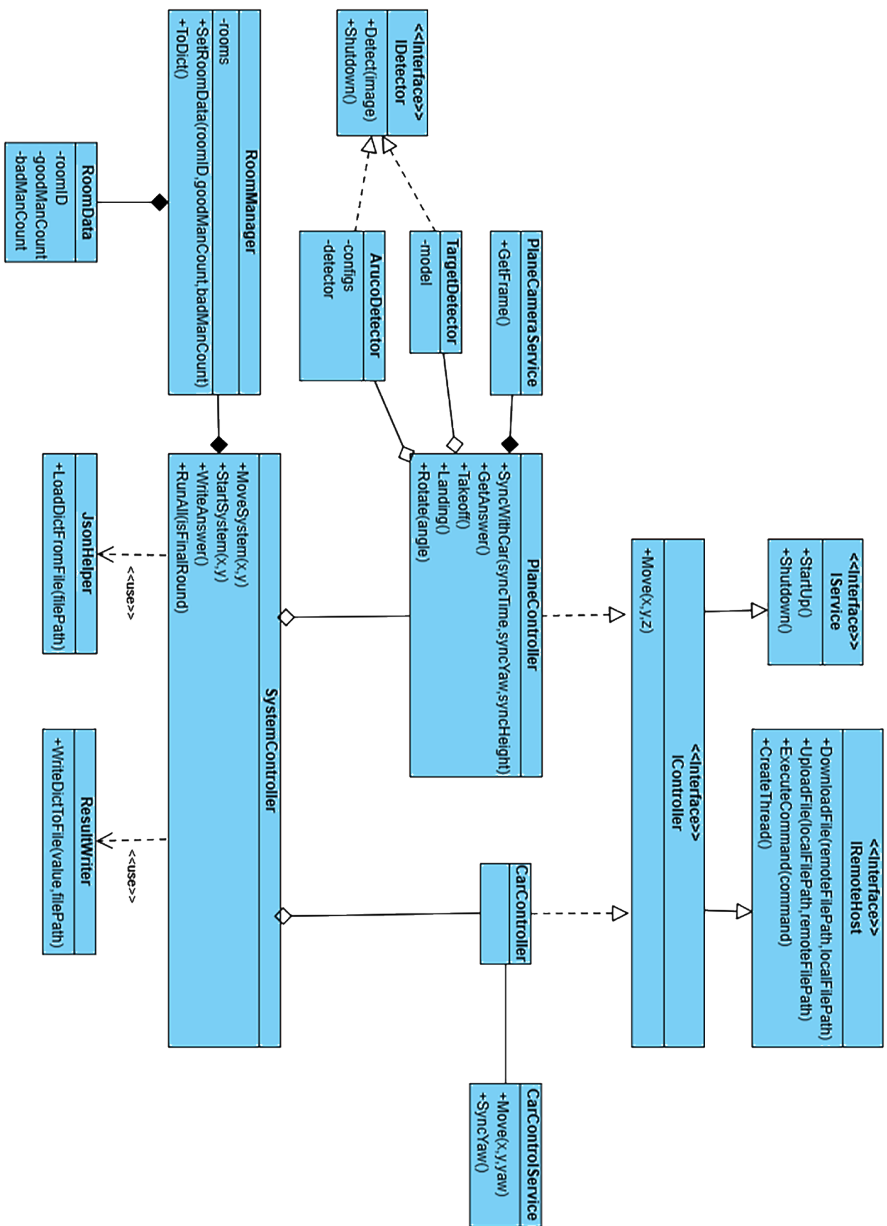


带队老师：孙磊

参赛队员：楼宸豪 胡家欣 郑乔译 徐屹寒 张柏铭 梁熙媛

1 代码框架





2 设计亮点

2.1 架构

1. 整体控制代码设计遵循面向对象的开发规范，而且考虑了”组合优于继承“的 ECS 开发范式；
2. 代码中出现的**几乎所有**参数全部以配置文件的方式加载，为系统提供了良好的可调试性；
3. 常用工具类被封装在 Utils 包中，方便不同的模块调用。

2.2 数学方法

1. 使用 `cv2.solvePnP()` 方法在二维图像中解算图像的平移向量；
2. 为了方便使用和更加贴合赛事场景，使用计算几何学方法重新对旋转角进行二维坐标内的解算。

2.3 SystemController

以异步的方式实现了小车和无人机的同时移动。

2.4 CarController

1. 使用 SSH channel 将小车的控制流程进行封装，通过与机载算力盒子进行明文通讯实现；
2. 使用激光雷达对场地边缘长度进行测量，通过数学方法解算小车的偏航角并实现自动校正。

2.5 PlaneController

1. 将无人机 *ros* 侧的 *service_client* 改写为在无人机端常驻运行的一个进程,同 *CarController*, 使用 SSH Channel 在控制端和执行端进行通讯，降低了启动进程所需要的操作系统的资源，提升了无人机控制的反应效率；
2. 使用单目相机视觉实现了对于张贴于车体上的 *Aruco* 特征码的识别与追踪功能；
3. 统计答案时考虑了由于环境因素导致答案的偶然误差，采用了统计学方法将答案的偶然误差降至最小。
4. 为减少 UDP 协议由于不检验数据完整性带来的数据丢失，在控制端和无人机端的 *PSDK* 侧建立了通讯通道，当 *PSDK* 接收到相应命令时，控制端会直接收到反馈，消灭了和 *ROS* 通讯造成的延迟，同时避免了由于丢包等原因造成的控制失败，大幅提升了无人机控制的稳定性

2.6 PlaneCameraService

设计了不阻塞主线程的，线程安全的帧同步队列，保证调用者在调用 *GetFrame()* 方法时收到的是从流收到的最新帧，避免了由于网络丢包或视频帧堆积造成的延迟。