成果：

1. 步兵实现了自动瞄准，可以较准确击打低速移动的目标（前提是不受外界干扰，如撞击的情况）
2. 英雄实现了自动发射和自动瞄准
3. 哨兵、无人机实现了自动瞄准以及自动发射
4. 在官方开源代码的基础上进行了改进，例如将部分参数放进xml、地址采用相对路径、新增相机测试及标定程序、增加主程序参数便于适应启动脚本，大幅提高调试便利性
5. 完成了大风车击打（粗暴版）的核心部分（还有部分需电控传入的参数（比如旋转方向）未写、未测试）
6. 完成了小电脑的选型
7. 通过增加顶层PID（已加入代码，可选电控或视觉来调）解决去年和电控对接时云台鬼畜的情况
8. 哨兵代码中加入了数字识别，能排除掉工程车

不足之处：

1. 装甲识别不够稳定
2. 对高速移动目标命中率较低
3. 更换阵营时需要手动修改xml中的颜色参数

TODO：

1. 更换工业相机
2. 根据情况新增运动预测
3. 完善大风车识别算法
4. 通过电控读取裁判系统传入颜色信息
5. 用树莓派以及高清小屏幕给工程写一个画面切换，摄像头用今年换下来的（可作为面试题）
6. 为工程车加入视觉辅助对位（取弹、英雄补弹）（根据情况可考虑使用深度相机，如Intel Realsense D435）
7. 电控判断视觉是否成功识别，利用操作界面的数据显示功能
8. 完善数字识别的功能
9. 处理对方摆尾和小陀螺的情况

培训方面：

今年在windows平台用vs作为开发环境进行培训，主要讲了opencv的一些常用函数，布置了识别特定颜色目标的小任务。

在windows下用vs配置环境的一大缺点是环境配置经常出现各种奇怪的问题，而且偏离实际开发情况。因此明年培训时必须在linux环境（可利用小电脑、学员电脑装linux、虚拟机等方式）下操作，可使用clion等编译器，同时让学员接触一下纯命令行编译代码的操作。具体步骤为：1. 安装ubuntu的教学，解决常见问题。2. 在ubuntu下搭建opencv开发环境。3. opencv常用内容教学。4. 小项目制作。5. 往年代码学习（只针对战队成员）

另外可以利用小学期进行一次招新。因为大部分同学的c语言都学了，上手视觉代码只要再学一下class以及几个常用的模板就可以，而且小学期课较少，便于培训开展以及任务布置。

人员安排方面：

今年人员安排工作做得非常不好，导致组员经常没事干，也不知道干什么。明年建议在学习代码时就开始分组（不同车，以及特殊功能开发）。鉴于不同车的代码大体一样，差别基本在云台尺寸信息、标定信息、识别参数、硬件信息、通信协议等与程序跑起来无关紧要的内容（infantry-TX2和UAV要注意USE\_NEON的问题），因此学习过程可以互相讨论。分组之后，各组成员进入对应车的群，了解机械和电控的进度并进行必要的交流。