1：分析Challenge题目

通过分析题目，我们得出一下解决思路：机器人的行进路线始终维持在离围墙一定的距离范围内，我们获得的输入量只有机器人前方和左侧的雷达测量距离。通过这两个量机器人可以判断自身和围墙在各种情况下的相对位置。当离墙位置不同时，就可以通过调整两侧轮子的转速来维持离墙一定距离稳定顺时针前进。

2：编写代码：

我们的编写思路是：设置一个离左侧墙的距离范围区间li（最小值）-lo（最大值），设置一个离前方墙的距离ft。用雷达测定距前方的距离f，距左侧的距离l。当f>ft且li<l<lo，此时直行；当f>ft且l<li，此时右转；当f>ft且l>lo，此时左转，通过以上“if”逻辑，可以完成直线运动上的运动方向修正。当f<ft时，需向右转一个大角度以完成转弯。由于老师已经提供了前进与左转的class，我们只需简单给出相应速度参数即可完成challenge的程序编写。

3：调整参数

调整参数是完成项目过程中最耗时也是最需要运气的部分。我们可能会遇到直线路段方向修正不及导致撞墙的问题，也可能遇到转向路段转弯角度不足的问题。通过在终端上print出机器人运动过程的f，l数据。结合观察模拟器情况，不断修正，尝试，最终调整出了可以稳定运动的参数。但我们并不就此满足，我们希望以更快的速度完成挑战。为了以更快同时保持稳定性，我们额外增加了两个量li’和lo’(li’<li，lo’>lo)，若l<li或l>lo，则以更快的速度修正路线，防止撞墙。

4：总结

在本次challenge中，组员们分工合作，各个成员都积极建言献策，为优化方案，优化程序做出了积极贡献。同时在整个过程中，我们也加深了对于Linux，ROS2，Python3的理解与应用，收获颇丰。