

1. 分析与设计

1. 框架

使用python的rpc框架 rpyc 包

2. 分析

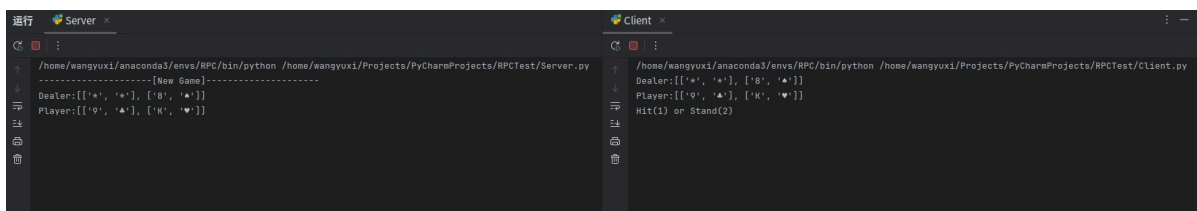
- 由于 Ace 的存在，使得当手中存在1张及以上 Ace 时，可能会有两种分数（至多两种）
- 当计算分数时，先判断是否存在 Ace，如果有，则选择其 ≤ 21 的最大值
- 判断是否 Bust 时，当存在 Ace 时，只需判断其最小值 ≤ 21 即可。
- Dealer 和 Player 每次抓牌之后，需要判断是否 Bust
- 在最后抓牌均结束且都没 Bust 下，比较其分数
- 由于 hole_card 的存在，需要一个 bool 变量 dealer_turn 保存是否是 Dealer 的回合。

3. Dealer策略

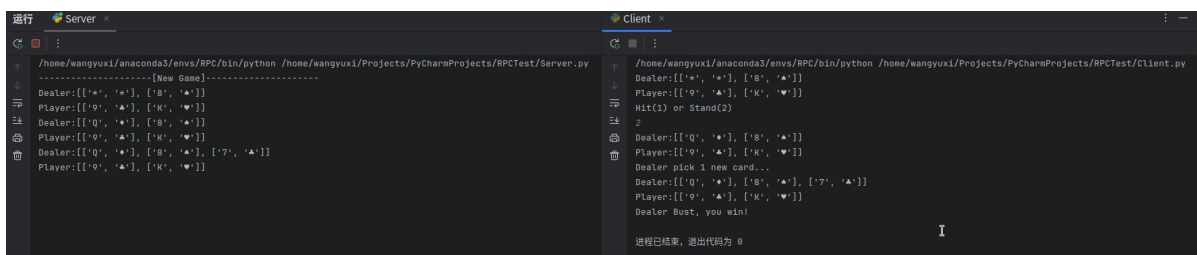
- Dealer 分数 ≤ 17 时: 抓牌
- 当 Dealer 的分数 \leq Player 分数时: 抓牌
- 当 Dealer 分数 $>$ Player 分数时: 停止

2. 实际演示

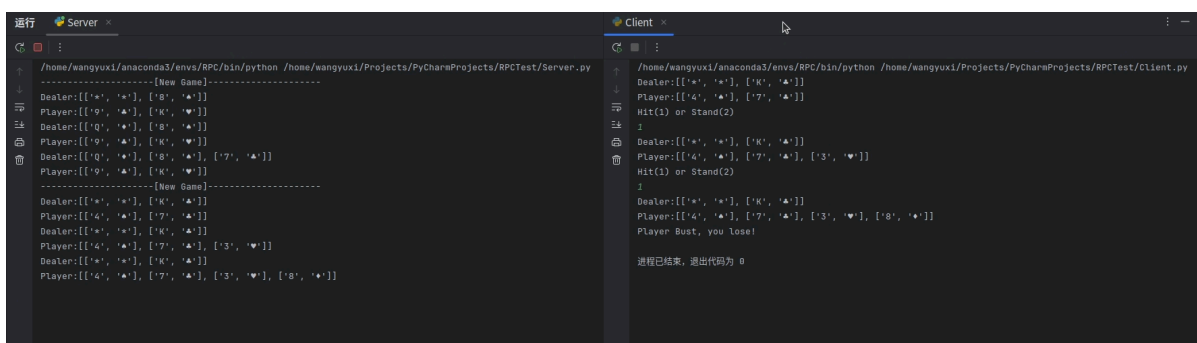
1. 程序运行, 玩家方显示双方卡片, hole_card 不显示, 输入 Hit(1) or Stand(2)



由于 player 已经19分，选择 Stand(2)，Dealer 显示 hole_card，并发现 Dealer 18分，选择继续抓牌，最后 Bust



2. Player 一直抓牌，最后 Bust



3. 存在 Ace 的情况

