1. 系统发版本信息【DOUDIZHUVER xxx】
2. 保存版本信息
3. 回复NAME给系统
4. 系统发送轮局信息【INFO 1,4,1,6,9,2100,15】
5. 保存轮局信息
6. 回复系统OK INFO
7. 系统发牌【DEAL B0,4,5,7,...】
8. 获取自己位置【cdir】
9. 保存牌到手牌中【iOnHnad】
10. 回复系统OK DEAL
11. 调用pokersort()把手牌从小到大排序
12. 叫牌
13. 若是系统转播其他人叫分【BID XX】

①保存其他人的叫分信息到iBid[]

1. 若是系统提问自己叫分【BID WHAT】

①调用CalBid函数叫分【有BUG，叫相同分或低分报错】

（3）回复平台OK BID

1. 系统发送底牌信息【LEFTOVER B27,48,53】
2. 确定地主方位并保存到cLandLord
3. 地主方的牌数+3
4. 如果地主是自己，则保存底牌到手牌
5. 回复平台OK LEFTOVER
6. 用pokersort()把手牌从小到大排序
7. 出牌
8. 若是其他人出牌【PLAY XX】

①将另外两方所出的牌储存到iOnTable数组中

②回复平台OK PLAY

1. 若是自己出牌【PLAY WHAT】

①调用CalPlay函数

1. 调用Line确定阵型
2. 把二维数组iOnTable按照最新值依次赋给一维数组iOnTableW
3. 调用pokersort()把手牌从小到大排序
4. 牌型初始化(struct PaiXing)
5. 调用Query()储存各种牌型，顺序按

**火箭>炸弹>三顺>双顺>单顺>三张>对子>单张**

**【无法拆分存储，如33445567，只能保存为334455，不能一并存34567】**

并对单牌，对牌按升序排

F. 计算手牌还有可以多少手就能出完【px->Time】

G. 调用helpdz()确定出牌内容【\*\*\*\*\*\*\*\*这个就是出牌策略\*\*\*\*\*\*\*\*\*】

1. carttype()根据其他人出牌的张数和内容确定出牌类型

**1.火箭2.炸弹3.单牌4.对牌5.三张策略6.三带X 7.连续单顺策略8.连续双顺策略9.连续三顺策略10三带一单策略11.四带二12.放弃**

b.使用switch判断出牌类型是什么选择对应策略

H. 出牌内容存到iOnTable

②调用iTos()将iToTable的内容输出到sCommandOut

③调用TakeOff()将要出的牌从手牌剔除

1. 出牌次数加1【iOTmax】
2. 游戏决出胜负，系统发送信息【GAMEOVER X】
3. 保存胜利者方位编号【cWinner】
4. 回复系统消息OK GAMEOVER
5. 判断是否所有轮次结束

①是，则引擎状态置0，main函数while循环结束，程序退出

②否，重新初始化所有参数，回到步骤**1**