

# ET 框架拖拉机项目源码与设计重构

deepwaterooo

May 18, 2023

## Contents

<b>1</b>	<b>DefinedConstant   CardCommands   CurrentState</b>	<b>1</b>
<b>2</b>	<b>CurrentPoker : Diamonds   梅花   红桃   黑桃</b>	<b>1</b>
<b>3</b>	<b>源码分析与重构</b>	<b>2</b>

## 1 DefinedConstant | CardCommands | CurrentState

```
// 程序常量
class DefinedConstant {
    // 时间常量
    internal const int FINISHEDONCEPAUSETIME = 1500; // 每圈暂停时间
    internal const int NORANKPAUSETIME = 5000; // 流局时间
    internal const int GET8CARDSTIME = 1000; // 摸 8 张底牌的时间
    internal const int SORTCARDSTIME = 1000; // 我的牌排序时间
    internal const int FINISHEDTHISTIME = 2500; // 每局暂停时间
    internal const int TIMERDIDA = 100; // 系统滴答
}
// 命令状态, 指示下一步动作
enum CardCommands {
    ReadyCards, // 发牌命令
    DrawCenter8Cards, // 画 8 张底牌的命令
    WaitingForSending8Cards, // 等待扣底的命令
    DrawMySortedCards, // 排序我的牌的命令
    Pause, // 通用暂停命令
    WaitingShowPass, // 显示流局的命令
    WaitingShowBottom, // 翻底牌的命令
    WaitingForSend, // 等待出牌
    WaitingForMySending, // 等待我出牌的命令
    DrawOnceFinished, // 出完一圈后的命令
    DrawOnceRank, // 出完一局后的命令
    Undefined // 未定义的命令
}
// 保存当前游戏状态的对象
[Serializable]
struct CurrentState {
    internal int OurCurrentRank; // 自己当前的牌局
    internal int OurTotalRound; // 总轮数
    internal int OpposedCurrentRank; // 对方的牌局
    internal int OpposedTotalRound; // 总轮数

    // 当前的【庄家】: 未定 0, 自己 1、对家 2、西 3、东 4
    internal int Master;

    // 当前的【花色】: 未定 0、红桃 1、黑桃 2、方块 3、梅花 4、无主 5
    internal int Suit;

    internal CardCommands CurrentCardCommands; // 当前命令
    internal CurrentState(int ourCurrentRank, int opposedCurrentRank, int suit, int master, int ourTotalRound, int opposedTot
}
}
```

## 2 CurrentPoker : Diamonds | 梅花 | 红桃 | 黑桃

- 这个东西重复四遍，什么意思嘛。就是因为这么初始化，没法支持用户配置【2 为常主】
- 设置为一个广谱的 OneSuit 之类的类，可以实例成四种类型，并且根据 2 是否为常主来设置大小

```
#region 方块
// 方块 (2,3,4,5,6,7,8,9,10,J,Q,K,A)
private int[] diamonds = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
internal int[] Diamonds {
    get { return diamonds; }
    set { diamonds = value; }
}
// 不带主的方块
private int[] diamondsNoRank = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
internal int[] DiamondsNoRank {
    get { return diamondsNoRank; }
    set { diamondsNoRank = value; }
}
// 方块 Rank 数
internal int DiamondsRankTotal = 0;
// 方块非 Rank 数
internal int DiamondsNoRankTotal = 0;
// 排序的牌型
internal int[] SortCards = new int[56];
#endregion // 方块
```

## 3 源码分析与重构

- 还是需要相对事理一个源码里必要的关键类。因为变量太多，容易忘记。不知道哪个变量取什么值，是什么意思
- 源码主要特点是：没有设计。像是没学过 OOP/OOD 的小屁孩写的。既然今天下午是看这个项目的源码与设计重构，就可以用好电脑，要比这个舒服多了。【爱表哥，爱生活!!! 活宝妹就是一定要嫁给亲爱的表哥!!!】没有分层，找不到 Model 层，控制层在哪里？源码设计不功能模块化。。