

无限注德州扑克(No-Limit Texas Hold'em)

人机对战系统开发深度研究报告

1. 执行摘要与系统架构概论

本深度研究报告旨在为开发一款专业级“人机对战(Human vs. Bot)”德州扑克产品提供详尽的规则定义、逻辑架构与实现细节。不同于普通的玩家指南，本文档是从系统架构师与游戏逻辑工程师的视角出发，深入剖析德州扑克(特别是无限注德州扑克, No-Limit Texas Hold'em)的底层运行机制。

德州扑克看似规则简单，实则包含极其复杂的状态机流转、边缘情况处理(Edge Cases)以及严格的数学计算规则。对于开发人员而言，最大的挑战往往不在于“发牌”本身，而在于处理多边彩池(Side Pots)、特殊状态下的下注限制(如最小加注量与再加注权限)、平分彩池(Split Pot)时的余数分配等细节，以及如何构建一个既能容纳人类玩家不确定性又能支撑Bot逻辑严密性的后端系统。

本文档将依据TDA(Poker Tournament Directors Association)2024版规则¹、Robert's Rules of Poker³以及国际通用现金局(Cash Game)标准，提供一套完备的逻辑开发蓝图。系统设计必须考虑到服务器权威性(Server Authority)，确保所有游戏逻辑在服务端运行，客户端仅负责渲染状态和发送指令，以防止作弊并确保Bot获取的信息与人类玩家对等(或受控)。

1.1 开发目标与适用范围

本报告涵盖以下核心模块：

- 游戏实体定义：牌组数据结构、玩家状态模型、桌面元素属性。
- 有限状态机(FSM)设计：从盲注投放到摊牌结算的完整生命周期管理。
- 核心下注逻辑：包括动作校验、最小加注量计算算法、全压(All-in)引发的特殊规则判定。
- 结算算法：多边彩池(Side Pots)的动态构建与分割、手牌比对算法、奇数筹码(Odd Chip)分配规则。
- 特殊场景处理：两人对战(Heads-Up)的特殊位置规则、断线重连逻辑、防作弊(烧牌)机制。
- 机器人接口：Bot如何感知游戏状态与执行决策。

2. 游戏核心实体与数据结构定义

在编写任何逻辑代码之前，必须准确定义游戏中的物理实体及其在程序中的数字化映射。数据结构的合理性直接决定了后续逻辑判断的复杂度。

2.1 扑克牌(The Deck)与发牌机制

2.1.1 牌组构成

标准的德州扑克使用一副除去大小王的国际通用扑克牌，共 52 张⁵。

- 花色 (Suits): 黑桃 (Spades, ♠)、红桃 (Hearts, ♥)、梅花 (Clubs, ♣)、方片 (Diamonds, ♦)。
 - 逻辑要点: 在德州扑克的手牌大小比较中，花色之间严格没有大小之分⁶。这一点在开发比牌算法时至关重要: 同花顺中，皇家同花顺黑桃和皇家同花顺红桃是完全平局，必须平分底池。这与中国某些地方性扑克玩法 (如黑桃大于红桃) 不同，开发时需特别注意避免混淆。
- 点数 (Ranks): A (Ace, 最高), K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2 (Deuce, 最低)。
 - 特例处理: A 在顺子 (Straight) 中具有二象性。它既可以作为最大牌参与 A-K-Q-J-10 (Broadway Straight)，也可以作为最小牌参与 5-4-3-2-A (Wheel Straight)⁶。在数据结构设计中，通常将 A 的值设为 14，但在顺子检测算法中需特殊处理其作为 1 的情况。

2.1.2 洗牌与随机性 (RNG)

对于人机对战产品，发牌的随机性是公平性的基石。如果 Bot 在后台计算时利用了未公开的牌堆信息，或者人类玩家通过抓包预测了随机数种子，游戏将失去意义。

- 算法要求: 必须使用加密级安全的伪随机数生成器 (CSPRNG)，如 /dev/urandom (Linux) 或专门的加密库 (如 Python 的 secrets 模块)，而非各语言标准库中普通的 Math.random() 或 Random()⁵。普通的线性同余生成器 (LCG) 容易被破解。
- 牌堆数据结构: 建议使用一个包含 52 个整数的数组或列表，每个整数映射一张唯一的牌 (例如 0-51，其中 0-12 为黑桃 2-A，13-25 为红桃 2-A 等)。洗牌即对该数组进行 Fisher-Yates Shuffle 算法操作。

2.1.3 烧牌 (Burn Cards) 逻辑

虽然在纯数字游戏中“烧牌”没有物理意义 (物理烧牌是为了覆盖牌背可能有标记的下一张牌)，但为了模拟真实赌场流程或在 UI 上展示真实性，通常保留此逻辑¹⁰。

- 规则: 在发翻牌 (Flop)、转牌 (Turn)、河牌 (River) 之前，荷官必须先弃掉牌堆顶的一张牌，且该牌通常牌面朝下不公开 (除非出现甚至处理错误需要回溯)。
- 开发实现: 在状态机流转时，执行 Deck.pop() 操作并将该牌移入 BurnedPile 列表，而非直接发公共牌。如果游戏需要展示“发牌动画”，UI 层应渲染一张牌滑入废牌堆的动作。
- 异常处理: 若程序逻辑错误导致未烧牌，严格规则下这属于“Misdeal”或需特定修正¹²，但在电子游戏中应通过代码逻辑严格保证烧牌步骤的原子性执行。

2.2 玩家状态 (Player States)

在任何给定的时间点，牌桌上的每个玩家 (无论是人类还是 Bot) 都必须处于且仅处于以下一种状态。状态机的设计应严格互斥，以避免逻辑冲突 (例如一个已弃牌的玩家被系统要求下注)。

1. **Active (参与中):** 玩家手中有牌，且筹码足以覆盖当前的最高下注，或者已经全压。这是玩家能赢得底池的唯一有效状态。
2. **Folded (弃牌):** 玩家已放弃当前手牌。
 - 逻辑推论: 弃牌玩家不再参与本局的任何结算，也不再拥有任何动作权。其之前投入底

池的筹码成为“死钱(Dead Money)”，归属于底池，玩家失去对该部分资金的所有权¹⁴。

3. **All-in (全压)**: 玩家已将所有筹码投入底池。
 - 特殊性: 此时玩家虽然处于“Active”状态(有权赢钱)，但失去了后续轮次的“动作权(Action)”。系统在轮到该玩家时应自动跳过，但在结算时需将其纳入比牌队列¹⁶。
4. **Sitting Out (离座)**: 玩家在桌上但暂时不参与发牌。
 - 资金局(Cash Game): 离座玩家不发牌，不收盲注。
 - 锦标赛(Tournament): 离座玩家依然会被发牌(随即被系统自动Fold)，并强制扣除盲注(Blinding off)，直到筹码耗尽¹⁸。
5. **Busted (淘汰)**: 筹码归零，被移除出游戏(锦标赛模式)。

数据结构建议:

JSON

```
Player {
  id: string;
  name: string;
  is_bot: boolean;
  stack: number; // 当前持有筹码(不含当前下注)
  bet_this_round: number; // 当前下注轮已投入的金额(用于计算加注额)
  total_bet_this_hand: number; // 本局游戏总投入(用于计算多底池权益)
  status: "ACTIVE" | "FOLDED" | "ALL_IN" | "SITTING_OUT";
  hole_cards: Card; // 2张底牌, 对其他玩家不可见
  has_acted: boolean; // 当前轮次是否已行动(用于判断轮次结束)
  position_index: number; // 相对位置标记
}
```

2.3 庄家位置与盲注(Dealer Button & Blinds)

德州扑克是强位置性游戏，所有行动顺序都严格参照庄家按钮(Dealer Button/BTN)。准确计算位置是实现游戏循环的第一步²⁰。

- 庄家(Button, BTN): 拥有最后一手行动权的优势位置(翻牌后)。
 - 小盲注(Small Blind, SB): 位于BTN顺时针下一位的玩家。通常强制下注金额为大盲的一半。
 - 大盲注(Big Blind, BB): 位于SB顺时针下一位的玩家。强制下注金额为全额基础注。
 - 枪口位(Under the Gun, UTG): 大盲注顺时针下一位的玩家，翻牌前(Pre-flop)首先行动。
-

3. 游戏流程控制与有限状态机 (Game Loop & FSM)

开发德州扑克的核心是构建一个健壮的有限状态机 (Finite State Machine, FSM)。每一局牌 (Hand) 都是一个新的状态循环。FSM的设计必须能处理正常流程和各种异常中断²²。

3.1 状态机顶层架构

游戏大循环可分为以下主状态：

1. **Hand Initialization** (牌局初始化)
2. **Pre-flop** (翻牌前)
3. **Flop** (翻牌圈)
4. **Turn** (转牌圈)
5. **River** (河牌圈)
6. **Showdown** (摊牌结算)
7. **Hand Cleanup** (清理与重置)

每个主状态内部包含子状态：PostingBlinds -> Dealing -> BettingRound -> StateTransition。

3.2 阶段一：初始化与按钮移动 (Hand Initialization)

3.2.1 按钮移动规则：死按钮 vs. 前移按钮

当有玩家在盲注位被淘汰或离座时，按钮和盲注的移动会变得复杂。主要有两种规则：

- **死按钮 (Dead Button)**²⁴：这是TDA和大多数锦标赛采用的规则。BTN移动到下一位玩家，如果该玩家因淘汰缺席，BTN可能指向一个空位（“死按钮”），或者由于SB位置无人，导致该局没有SB。
 - 规则核心：每位玩家每轮必须支付一次大盲，一次小盲。大盲位优先保证有玩家支付。
- **前移按钮 (Forward Moving Button)**²⁶：常见于现金局 (Cash Games)。BTN总是移动到下一个活跃玩家。这可能导致某些玩家在一轮中避开小盲，但在简化的人机对战逻辑中，这更容易实现且体验流畅。
 - 开发建议：对于App类人机对战产品，强烈推荐使用前移按钮规则，即BTN总是指向下一个Player，SB是BTN+1，BB是BTN+2。这避免了复杂的“谁补盲”逻辑。

3.2.2 盲注投放

系统强制从SB和BB的筹码中扣除相应金额并放入底池。

- **破产处理**：如果玩家筹码不足以支付盲注，则视为All-in。例如，SB应付10，但只有5，则SB All-in 5，底池增加5。这不会影响后续玩家的动作要求 (BB仍需付全额)。

3.3 阶段二：翻牌前流程 (Pre-flop)

1. **发底牌 (Dealing Hole Cards)**：从SB开始，顺时针每人发一张，循环两轮，每人共2张。Bot的底牌数据应在此时生成，但仅Bot自己 (AI模块) 可见，UI层对人类玩家隐藏。
2. **行动开始**：

- 先手:从**UTG**(BB的下一位)开始行动。
 - 可用动作: Fold, Call(跟大盲), Raise(加注)。
 - 注意:此时不能Check, 因为大盲注被视为一笔强制的“Bet”。UTG必须至少匹配大盲金额才能留存。
3. 大盲特权(**Option**): 如果所有玩家都Fold或Call回到大盲位(即没有加注), 大盲玩家拥有“特权”。他可以选择:
- **Check**: 因为他已经支付了大盲, 且没人加注, 他可以免费看牌。
 - **Raise**: 他也可以选择加注。

3.4 阶段三至五: 翻牌、转牌、河牌(**Post-flop Streets**)

这三个阶段逻辑高度相似, 统称为“翻牌后”。

1. 烧牌与发公共牌:
 - Flop: 烧1张, 发3张。
 - Turn: 烧1张, 发1张。
 - River: 烧1张, 发1张。
2. 行动顺序变更:
 - 与Pre-flop不同, 翻牌后的行动总是从**BTN**左手边第一个依然**Active**的玩家(通常是SB)开始²⁰。
 - 如果SB已Fold, 则由BB先动; 若BB也Fold, 则由UTG先动。
3. 下注轮(**Betting Round**):
 - 第一个行动的玩家可以选择 **Check**(因为本轮目前下注额为0)或 **Bet**。
 - 一旦有人Bet, 后续玩家只能 Call, Raise, 或 Fold。

3.5 阶段六: 摊牌(**Showdown**)

- 提前结束: 如果在任何轮次, 除了一名玩家外其他人全部Fold, 则该玩家直接获胜, 无需摊牌, 也不发剩余的公共牌(Uncontested Pot)。
- 进入比牌: 若河牌圈结算后仍有2人或以上Active(或All-in), 则进入摊牌。
- 摊牌顺序(**Order of Showdown**)²⁸:
 - 这是一条重要的礼仪规则, 虽然在程序中可以同时亮牌, 但为了模拟真实体验:
 - 主动权原则: 若河牌圈有下注/加注行为, 最后发起主动动作(**Bet**或**Raise**)的玩家必须先亮牌。因为其他玩家是付钱看他的牌, 他有义务证明自己值得赢得底池。
 - 被动原则: 若河牌圈所有人都Check, 则从最早位置(SB位方向)开始顺时针亮牌。
 - **Muck**(埋牌): 后位玩家看到前位玩家亮出的牌比自己大时, 可以选择不亮牌直接认输(Muck), 以保护隐私。Bot应具备此逻辑, 除非规则强制亮牌(All-in通常强制亮牌)。

4. 盲注与特殊对局模式

4.1 两人对战(**Heads-Up**)的特殊规则

两人对战(Heads-Up)是人机对战中最常见的模式, 其盲注与行动规则与多人局有根本性的反转

, 这是开发中最容易出错的硬编码逻辑²⁰。

- 位置定义:
 - **BTN**(庄家)同时也是 小盲注(**SB**)。
 - 对手 是 大盲注(**BB**)。
 - 原因:在多人局中, SB在BTN左边。但在两人局若照此排列, BB将是BTN, 意味着BB在翻牌前最后行动, 翻牌后也拥有位置优势(最后行动)。为了平衡, 将BTN设为SB, 使其翻牌前先行行动(劣势), 翻牌后后行动(优势)。
- 行动顺序:
 - 翻牌前(**Pre-flop**):BTN/SB 先行行动(Call/Raise/Fold), BB 后行动。
 - 翻牌后(**Post-flop**):BB 先行行动(因为他在BTN的“左边”/顺时针下一位), BTN 后行动。
- 代码实现警告:不要试图用通用的 $(btn_index + 1) \% num_players$ 公式来计算SB位置。必须加判断 `if num_players == 2: sb_index = btn_index`。

5. 下注逻辑核心引擎(Betting Engine)

这是整个系统的“心脏”, 负责校验玩家动作的合法性、计算下注金额、处理分池。任何逻辑漏洞都会导致游戏崩溃或被利用。

5.1 动作校验与参数

每个状态下, 系统需计算以下参数供前端和AI使用:

- `current_bet`: 本轮当前单人最高下注额。
- `min_raise`: 当前允许的最小加注额。
- `legal_actions`: 玩家当前允许的动作集合(如 或)。

5.2 无限注(No-Limit)的加注计算算法

在无限注德州扑克中, “加注(Raise)”的金额限制有严格的数学定义, 特别是最小加注额(**Minimum Raise**)³²。

5.2.1 最小加注规则

- 规则:加注的增量(Raise Amount)必须至少等于上一次加注的增量(Delta)。如果这是本轮第一次加注, 则增量必须至少等于大盲注或上一个Bet的金额。
- 公式:

$$MinRaiseTotal = PreviousBet + (PreviousBet - BetBeforeThat)$$

或者:

$$MinRaiseTotal = CurrentBet + LastIncrement$$

场景演练(盲注 10/20):

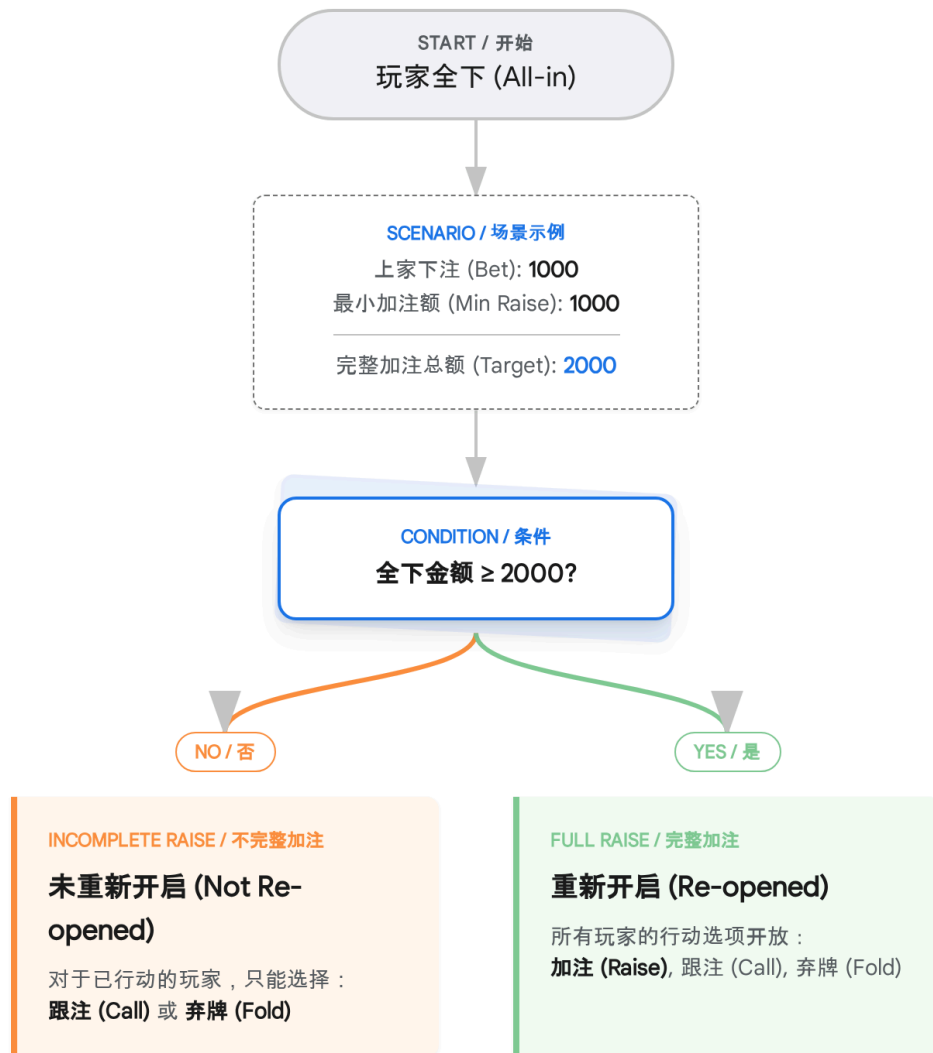
1. **Player A** 下注 (Bet) 50。
 - 此时 $\text{CurrentBet} = 50$ 。
 - $\text{LastIncrement} = 50 - 0 = 50$ 。
2. **Player B** 想要加注 (Raise)。
 - 他必须至少加注 50 (增量)。
 - 最小总下注额 = 50 (A的Bet) + 50 (最小增量) = **100**。
3. **Player C** 想要再加注 (Re-raise)。
 - 假设 B 实际加注到了 150。
 - 此时 $\text{CurrentBet} = 150$ 。
 - $\text{LastIncrement} = 150 - 50 = 100$ 。
 - C 的最小加注增量是 100。
 - C 的最小总下注额 = $150 + 100 = \mathbf{250}$ 。

5.2.2 全压 (All-in) 引发的“投注重开”规则 (Re-opening the Bet)

这是TDA Rule 47中最晦涩难懂的部分，也是开发重灾区¹。当一个玩家筹码不足以支付“完整最小加注”时，他可以All-in任何剩余金额。这个“不完整加注 (Incomplete Raise)”是否会重新激活之前已经行动过的玩家的加注权？

- 规则判据：
 - 如果全压金额的增量 \geq 完整最小加注增量：视为完整加注 (**Full Raise**)，所有玩家的加注权重置，可以再加注。
 - 如果全压金额的增量 $<$ 完整最小加注增量：视为行动而已 (**Action Only**)。对于已经行动过且现在只需跟注 (Call) 的玩家，下注权不重启——他们只能 Call 或 Fold，不能再 Raise (除非有其他玩家做了完整加注)。对于尚未行动的玩家，依然可以自由 Raise。

逻辑决策树：全下是否重新开启投注？



根据 TDA 规则第47条，只有当全下金额代表一个完整的、合法的加注时，才会对已行动的玩家重新开启投注。如果全下金额少于完整加注，这些玩家只能选择跟注或弃牌。

Data sources: [TDA Rules](#), [Poker TDA Forum](#), [Ebro Rules](#)

代码逻辑参考：

Python


```
# 伪代码:判断是否允许 Re-raise
def can_player_reraise(player, current_bet, last_full_raise_increment):
    if not player.has_acted_this_round:
        return True # 尚未行动的玩家总是可以加注

    amount_to_call = current_bet - player.current_bet_in_round
    # 如果面临的加注额包含了至少一个完整的增量
    if amount_to_call >= last_full_raise_increment:
        return True
    else:
        return False # 仅面临All-in带来的零头加注, 不可再加注
```

6. 底池管理与分配(Pot Management)

当涉及多名玩家All-in且筹码量不等时, 单一底池无法满足分配需求, 必须构建主池 (**Main Pot**) 和多个边池 (**Side Pots**)。

6.1 边池构建算法

基本原则: 玩家只能赢取他所投入的那部分筹码对应的对手筹码。如果A下注100, B下注200, A赢了也只能从B那里拿走100, B剩下的100要么退回(若无其他人), 要么归属其他边池¹⁴。

算法步骤:

1. 收集阶段: 本轮下注结束, 所有玩家投入筹码进入暂存区。
2. 检测**All-in**: 若无All-in, 所有筹码直接合入主池。若有All-in, 进入分池逻辑。
3. 排序: 将本轮所有参与游戏的玩家(包括已Fold但投了钱的)按“本局总投入(Total Contribution)”从小到大排序。
4. 迭代剥离:
 - 设 active_players 为当前参与分池的玩家集合。
 - 找到 active_players 中投入最少的非零金额 min_stack。
 - 创建底池: 该池金额 = min_stack * len(active_players)。
 - 标记资格: 只有在本轮未Fold的玩家有资格争夺此池。
 - 扣除筹码: 所有 active_players 的暂存投入减去 min_stack。
 - 移除玩家: 投入已归零的玩家移出 active_players。
 - 重复上述步骤, 直到所有筹码分配完毕。

实战案例:

- A (All-in 100)
- B (All-in 300)
- C (All-in 200)

- D (All-in 400) - D筹码最多, Cover所有人
- 计算过程:
 1. **Pot 1 (Main Pot):** 最小栈是A (100)。4人参与。
 - 金额 = $100 * 4 = 400$ 。
 - 争夺者: A, B, C, D。
 - 剩余待分: A=0, B=200, C=100, D=300。
 2. **Pot 2 (Side Pot 1):** 剩下最小栈是C (100)。B, C, D参与 (A已剔除)。
 - 金额 = $100 * 3 = 300$ 。
 - 争夺者: B, C, D。
 - 剩余待分: B=100, C=0, D=200。
 3. **Pot 3 (Side Pot 2):** 剩下最小栈是B (100)。B, D参与。
 - 金额 = $100 * 2 = 200$ 。
 - 争夺者: B, D。
 - 剩余待分: D=100。
 4. 退款: D剩下的100没有对手匹配, 直接退还给D。

6.2 奇数筹码分配(The Odd Chip)

当底池无法被平时(例如Pot 25, 2人平分, 每人12.5, 但筹码最小单位是1), 多出的最小单位筹码(Odd Chip)给谁?

- 规则:
 - 有公共牌的游戏(Hold'em, Omaha): 根据位置分配。奇数筹码发给BTN左手边顺时针方向最近的赢家(即最不利位置的玩家, Worst Position)³。
 - 逻辑原因: 这通常被视为对劣势位置的一种微小补偿, 或者是为了避免庄家(BTN)总是获利。
 - 花色规则(仅用于Stud/Razz): 注意, 在七张牌梭哈中, 奇数筹码给拥有最高花色单牌的玩家。但在德州扑克开发中, 切勿使用花色决定奇数筹码。
- 多边池分配: 每个边池单独结算, 每个边池都可能产生奇数筹码, 需分别按上述规则处理。

7. 牌型评估与结算(Hand Evaluation)

7.1 最佳5张规则(Best 5 Cards Rule)

这是新手理解和程序实现的易错点。

- 定义: 每位玩家从自己的2张底牌和5张公共牌(共7张)中, 选出数学上最大的5张组合。
- 组合机制:
 - 2张底牌 + 3张公牌
 - 1张底牌 + 4张公牌
 - 0张底牌 + 5张公牌(称为“Playing the Board”)
 - 系统实现: 遍历 $C(7, 5) = 21$ 种组合, 取最大值。对于高性能需求, 可使用位运算查表法(如 Cactus Kev 算法或 TwoPlusTwo Evaluator), 将7张牌编码为整数, 直接查表得

出Rank Value。

7.2 详细牌型等级与比牌逻辑(从大到小)

以下是必须实现的判断逻辑, 注意**踢脚牌(Kicker)**的处理⁶。

等级	牌型名称	英文	说明与比牌规则
1	皇家同花顺	Royal Flush	同花色的 A-K-Q-J-10。这是 最大的牌, 若多家持 有(仅可能在公牌就 是皇家同花顺时), 平分底池。
2	同花顺	Straight Flush	任意顺子且同花色 (如 9-8-7-6-5 ♡)。 比牌: 比最大牌点 数。
3	四条	Four of a Kind / Quads	四张同点数 + 1张杂 牌(Kicker)。 比牌: 先比四张的大 小, 若相同(公牌上 有四条), 则比第5张 单牌(Kicker)的大 小。
4	葫芦	Full House	三条 + 一对。 比牌: 先比三条部分 的大小; 若三条相同 , 再比对子部分的大 小。
5	同花	Flush	任意5张同花色, 不 连号。 比牌: 从最大单牌开 始逐张比对。若前4

			张都一样, 比第5张。花色不影响大小。
6	顺子	Straight	<p>5张连号, 花色不同。</p> <p>比牌: 比最大牌。注意 A-2-3-4-5 是5高顺子, 输给 2-3-4-5-6。</p>
7	三条	Three of a Kind / Trips	<p>三张同点数 + 2张杂牌。</p> <p>比牌: 先比三张的大小, 若相同, 依次比两张踢脚牌。</p>
8	两对	Two Pair	<p>两组对子 + 1张杂牌。</p> <p>比牌: 先比大对子, 再比小对子, 最后比踢脚牌。这是一个常见的**被公牌“伪造”**陷阱(Counterfeited)。</p>
9	一对	One Pair	<p>一组对子 + 3张杂牌。</p> <p>比牌: 比对子大小, 再依次比3张踢脚牌。</p>
10	高牌	High Card	<p>无上述牌型。</p> <p>比牌: 从最大牌开始逐张比对, 直到分出胜负。</p>

7.2.1 踢脚牌陷阱案例 (The Counterfeit Issue)

开发者常犯错误是只比较主要部分而忽略踢脚牌，或者比较了第6张牌。

- 场景：
 - Player A: 8 – 8
 - Player B: 8 – 9
 - Board: $A - A - K - K - Q$
 - 错误判断: A有一对8, B有一对8, A和B平分? 或者A有两对(A-A-8-8), B有两对?
 - 正确逻辑:
 - Player A 最佳5张: $A - A - K - K - Q$ (公牌本身的两对+Q踢脚)。一对8成了废牌。
 - Player B 最佳5张: $A - A - K - K - Q$ (同样是公牌)。
 - 结果: 平分底池 (**Split Pot**)。Player A手中的8-8完全不起作用, 因为它们比公牌上的Q还要小, 无法进入前5张。
-

8. 特殊状态与异常处理 (Edge Cases)

8.1 断线重连与状态快照

在人机对战中, 特别是涉及网络交互时, 必须支持断线重连。

- 快照 (**Snapshot**): 每一步状态转移后 (如玩家下注后), 系统应序列化当前 GameState 并持久化 (Redis或DB)。
- 包含数据:
 - 所有玩家的当前手牌 (加密存储)。
 - 当前底池、边池金额。
 - 公共牌。
 - 当前行动者索引 (Action Index)。
 - 剩余思考时间。
- 恢复逻辑: 玩家重连时, 下发完整快照。前端根据快照重建UI。

8.2 机器人思考时间模拟

如果是为了模拟真实对局, Bot不应瞬间行动。

- 随机延迟: Bot决策后, 加入一个符合正态分布的延迟 (如 mean=2s, std=1s), 再向服务器发送动作。
- 复杂局长考: 如果底池很大或局面复杂 (如面临All-in), Bot应展示更长的“思考”状态。

9. 机器人接口设计 (Bot Interface)

为了让不懂扑克玩法的开发人员能接入AI, 需要定义清晰的I/O接口⁴³。

9.1 输入: 游戏状态感知 (Observation)

Bot需要接收“不完全信息”的状态包。

JSON

```
{
  "hand_id": "1024",
  "position": "BTN",
  "hole_cards": ["As", "Kh"], // 仅Bot自己的牌
  "community_cards":,
  "pot": 150,
  "current_bet": 50,
  "min_raise": 100,
  "opponents":,
  "legal_actions":
}
```

9.2 输出: 动作指令 (Action)

Bot计算后返回简化的指令。

JSON

```
{
  "action_type": "RAISE",
  "amount": 200 // 代表加注到总额200 (Raise To)
}
```

- **Raise Logic:** 通常建议Bot接口使用“Raise To”(加注到总数), 而非“Raise By”(增加数额), 这更符合服务器校验逻辑, 减少歧义。

10. 附录：规则参数配置建议

为了增加游戏的可玩性，建议将以下规则设为可配置项：

1. **Big Blind Ante (大盲前注)**: 为了加快游戏节奏，现代锦标赛常采用大盲支付所有人的Ante⁴⁴。
2. **Straddle (抓头)**: 允许枪口位 (UTG) 在发牌前自愿购买双倍大盲，从而获得翻牌前最后行动权⁴。
3. **Run It Twice (发两次牌)**: 在全压底池中，双方同意后，剩余公牌发两组，底池分两半结算，以降低波动 (Variance)。

通过遵循本文档中的架构设计、数学规则和异常处理方案，开发团队将能够构建出一个逻辑严密、符合国际标准且具备高度扩展性的德州扑克人机对战系统。

[报告结束]

Works cited

1. View Poker TDA Rules, Procedures, & Addendum - Poker ..., accessed on February 6, 2026, <https://www.pokertda.com/view-poker-tda-rules/>
2. TDA Poker Rules - Tournament Directors Association, accessed on February 6, 2026, <https://www.pokertda.com/poker-tda-rules/>
3. ROBERTS RULES OF POKER - Pagat, accessed on February 6, 2026, <https://www.pagat.com/docs/RobsPkrRules11.pdf>
4. Straddle Poker: Understanding the Voluntary Blind | esports.gg, accessed on February 6, 2026, <https://esports.gg/guides/betting/straddle-poker/>
5. Math of Poker - Basics | Brilliant Math & Science Wiki, accessed on February 6, 2026, <https://brilliant.org/wiki/math-of-poker/>
6. Poker Hand Rankings In Order - Downloadable Cheatsheet, accessed on February 6, 2026, <https://www.poker.org/poker-hands-ranking-chart/>
7. Poker Hands: The Complete Guide to Hand Rankings - Riverwind, accessed on February 6, 2026, <https://www.riverwind.com/blog/poker-hands-the-complete-guide-to-hand-rankings/>
8. Poker School 2, accessed on February 6, 2026, <https://gp-inc.jp/poker/school/lesson2.html>
9. Split Pots in Texas Hold'em - LiveAbout, accessed on February 6, 2026, <https://www.liveabout.com/split-pots-in-texas-holdem-537648>
10. Texas hold 'em - Wikipedia, accessed on February 6, 2026, https://en.wikipedia.org/wiki/Texas_hold_%27em
11. Understanding the Burn Card in Poker | PokerNews, accessed on February 6, 2026, <https://www.pokernews.com/pokerterms/burn-card.htm>

12. Burn Cards : r/poker - Reddit, accessed on February 6, 2026, https://www.reddit.com/r/poker/comments/112bf8a/burn_cards/
13. Misdeals, Premature Flops, Burn & Board Card Errors, accessed on February 6, 2026, <https://www.pokertda.com/forum/index.php?topic=836.0>
14. How to Figure Out Poker Side Pots: 7 Steps (with Pictures) - wikiHow, accessed on February 6, 2026, <https://www.wikihow.com/Figure-Out-Poker-Side-Pots>
15. Distributing side pots in poker - Stack Overflow, accessed on February 6, 2026, <https://stackoverflow.com/questions/62672186/distributing-side-pots-in-poker>
16. Betting in poker - Wikipedia, accessed on February 6, 2026, https://en.wikipedia.org/wiki/Betting_in_poker
17. How are side pots built? - Poker Stack Exchange, accessed on February 6, 2026, <https://poker.stackexchange.com/questions/462/how-are-side-pots-built>
18. Texas Hold'em Tournament Rules - Amazon S3, accessed on February 6, 2026, <https://s3.amazonaws.com/file.imleagues/Images/Schools/Uploaded/201702/201721510844.pdf>
19. POKER TOURNAMENT DIRECTORS ASSN. 2019 Rules Version 1.0 Sept 17, 2019 - RIS, accessed on February 6, 2026, https://ris.dls.virginia.gov/uploads/11VAC20/dibr/Poker%20Tournament%20Directors%20Assoc_2019%20Rules-20221228170122.pdf
20. What is the official procedure for starting a hand in texas holdem heads up?, accessed on February 6, 2026, <https://poker.stackexchange.com/questions/10366/what-is-the-official-procedure-for-starting-a-hand-in-texas-holdem-heads-up>
21. Blinds, Antes & The Button | Poker Rules, accessed on February 6, 2026, <https://upswingpoker.com/poker-rules/blinds-antes-button/>
22. Application Design Patterns: State Machines - NI - National Instruments, accessed on February 6, 2026, <https://www.ni.com/en/support/documentation/supplemental/16/simple-state-machine-template-documentation.html>
23. How to implement state machine to card games such as poker or russian poker?, accessed on February 6, 2026, <https://softwareengineering.stackexchange.com/questions/431848/how-to-implement-state-machine-to-card-games-such-as-poker-or-russian-poker>
24. Poker Rules - Explanations | PokerZone, accessed on February 6, 2026, <https://www.pokerzone.com/rules/explanations/>
25. Dead Button vs. Moving Button when blind is busted, accessed on February 6, 2026, <https://www.pokertda.com/forum/index.php?topic=479.0>
26. Dead Button or Forward Moving Button? - Poker TDA: Tournament Directors Association, accessed on February 6, 2026, <https://www.pokertda.com/forum/index.php?topic=234.0>
27. Dealer and Small Blind go bust - how does Dealer button move in this case?, accessed on February 6, 2026, <https://poker.stackexchange.com/questions/11039/dealer-and-small-blind-go-bust-how-does-dealer-button-move-in-this-case>
28. Tournament Rules: (Texas Hold'em) | PDF | Betting In Poker | Gaming - Scribd,

- accessed on February 6, 2026,
<https://www.scribd.com/document/219251792/Tournament-Rules>
29. The Showdown Cards in Texas Hold'em - Dummies, accessed on February 6, 2026,
<https://www.dummies.com/article/home-auto-hobbies/games/card-games/poker/showdown-cards-texas-holdem-232939/>
 30. How to Play Heads-Up Poker: Rules, Strategy, and More - wikiHow, accessed on February 6, 2026, <https://www.wikihow.com/Heads-Up-Poker>
 31. How do blinds work when there are just two players in Texas Holdem? : r/poker - Reddit, accessed on February 6, 2026,
https://www.reddit.com/r/poker/comments/14hvslu/how_do_blinds_work_when_there_are_just_two/
 32. Minimum Raise Rules in No Limit Texas Hold'em - Pokerrrr2, accessed on February 6, 2026,
<https://www.pokerrrrapp.com/single-post/minimum-raise-rules-in-no-limit-texas-holdem>
 33. No Limit Texas Hold'em Rules (Beginner's Guide) - The Lodge Card Club, accessed on February 6, 2026,
<https://thelodgepokerclub.com/no-limit-texas-holdem-rules-beginners-guide/>
 34. What is the min-raise and min-reraise in Holdem No Limit? - Poker Stack Exchange, accessed on February 6, 2026,
<https://poker.stackexchange.com/questions/2729/what-is-the-min-raise-and-min-reraise-in-holdem-no-limit>
 35. The Ultimate Beginner's Guide to Raising and Re-Raising in Poker! - 888 Poker, accessed on February 6, 2026,
<https://www.888poker.com/magazine/strategy/raise-in-poker-beginners-guide>
 36. Player all-in for less than a min-raise. Bet re-opening & min re-raise questions, accessed on February 6, 2026,
<https://www.pokertda.com/forum/index.php?topic=1287.0>
 37. TDA Rules for Poker Tournaments in the Ebro Poker Room - Ebro Card Casino, accessed on February 6, 2026,
<https://www.goebro.com/ebro-poker-tournaments/ebro-poker-tda-rules.html>
 38. algorithms - Poker split / side pots - Software Engineering Stack Exchange, accessed on February 6, 2026,
<https://softwareengineering.stackexchange.com/questions/317640/poker-split-side-pots>
 39. Splitting up the pot in multiway all-ins where everyone had a different stack size : r/poker, accessed on February 6, 2026,
https://www.reddit.com/r/poker/comments/xla4pv/splitting_up_the_pot_in_multiway_allins_where/
 40. How to split a pot with an odd number? [duplicate] - Poker Stack Exchange, accessed on February 6, 2026,
<https://poker.stackexchange.com/questions/10928/how-to-split-a-pot-with-an-odd-number>
 41. Poker Rules - Ties | PokerZone, accessed on February 6, 2026,

- <https://www.pokerzone.com/rules/ties/>
42. Kicker (poker) - Wikipedia, accessed on February 6, 2026,
[https://en.wikipedia.org/wiki/Kicker_\(poker\)](https://en.wikipedia.org/wiki/Kicker_(poker))
43. Poker Game State Detection | CS231n - Stanford University, accessed on February 6, 2026,
<https://cs231n.stanford.edu/2024/papers/poker-game-state-detection.pdf>
44. A Guide to the Poker Button, Blinds, Ante, and Straddle | PokerNews, accessed on February 6, 2026,
<https://www.pokernews.com/strategy/poker-guide-button-blinds-ante-straddle-explained-39355.htm>