**PokerBots 2025 Course Notes**

**Florida International University**

**Competition Lecture 1 - Introduction To PokerBots & Poker**

Contents

1.1 Competition Overview and Logistical Details 2

1.2 Introduction to Poker 2

1.3 FIU PokerBots GitHub 2

1.4 Cross-Applications of Game Theory 3

1.5 Sam Ganzfried: “Introduction to Computational Game Theory 4

1.6 Texas Hold’em Rules 4

1.7 Hand Rankings & Types 4

1.8 Betting Round Structure 7

1.9 Overview of Skeleton Bot Architecture 8

1.10 Testing Your Bot 8

**1.1 Competition Overview and Logistical Details:**

The FIU PokerBots Hackathon will run for 6 weeks, from the first Saturday of May till through the first Saturday of June. Each Saturday of May will be an instructional lesson held here in pg6 room 116. The final competition will be held June 7th, also here in pg6 116.

Competitors may attend lessons in person or over zoom, we will take questions from the text chat of the zoom call. We ask that at least one member of each team attend the Final Tournament. If no member of your team can attend the Final Tournament, please reach out to Nick or Henrique so that we may see if we can make an accommodation.

**1.2 FIU PokerBots GitHub**

Check Github, we will upload the lesson notes, powerpoint, and sample code for each lesson the morning of the lesson.

Share github now for those unaware, make sure everyone find the first lesson, have all download the sample bot for this lecture

What you can expect to learn each lesson, lessons outline/agenda.

**1.3 Introduction to Poker**

* Why poker? Great domain for studying the influence of information & strategy in games
* Where did it come from? the modern form of poker emerged in the early 19th century along the Mississippi River
* Who created it? Nobody knows!
* How has it evolved? Poker was largely popularized with the introduction of Texas Hold’em. Which became the dominant variant after the first WSOP tournament in 1970, held in Binion’s Horseshoe Casino in Las Vegas, Nevada. The next year, the format for the Main Event as a freeze-out Texas hold’em game was introduced. In the early 2000s, The advent of online play brought sophisticated analysis tools, tracking software, and a new generation of players who approached the game with mathematical precision, leading to significant advances in optimal strategy.
* Parallels to other credit creation systems: Fractional Reserve Banking, Derivatives Markets i.e. Futures, Options, Carbon Credits, etc.
* For a human to be successful in poker, it requires a mix of mathematical and psychological skills. Professional Players must master mathematical concepts like pot odds and expected value while developing keen psychological awareness to read opponents' tendencies and tell-tale behaviors. Emotional control, or "tilt management," is crucial as players must be able to make rational decisions in high-pressure situations. Strategic thinking in poker is multifaceted, players must consider their own cards and the ranges of hands their opponents may have based on their opponents actions. Top poker players must be constantly adapting their strategies as the game evolves and their opponents adjust, making continuous learning and self-improvement essential traits for long-term success as a poker player.

**1.4 Cross-Applications of Game Theory & Reinforcement Learning**

1. Drug Discovery & Protein Folding

* The "game": Finding optimal molecular structures is framed as a game where the "player" (RL agent) tries to design molecules that maximize binding affinity while maintaining drug-like properties
* Game theory application: Multi-objective optimization treated as a competitive game between different molecular properties
* Real example: DeepMind's AlphaFold uses game-theoretical approaches to predict protein structures by treating amino acid interactions as a cooperative game
* Benefit: Dramatically speeds up drug discovery process and reduces costs

1. Autonomous Vehicle Navigation

* The "game": Each vehicle is a player trying to optimize multiple competing objectives (safety, speed, comfort, fuel efficiency)
* Game theory application: Traffic interactions modeled as a multi-agent game where vehicles must cooperate and compete for road space
* Real example: Tesla's autonomous driving systems use game theory to predict other drivers' behavior and optimize path planning
* Benefit: More robust and naturalistic autonomous driving behavior

1. Financial Market Trading

* The "game": Trading viewed as a game between market participants with imperfect information
* Game theory application: Zero-sum game framework for determining optimal trading strategies
* Real example: Susquehanna International Group uses game theory and RL to identify market inefficiencies and optimal trade timing
* Benefit: More stable trading strategies that account for other market participants' behavior

1. Software Testing & Bug Detection

* The "game": Bug finding treated as a game between a "defender" (test generator) and "attacker" (program)
* Game theory application: Adversarial testing where the RL agent learns optimal test case generation strategies
* Real example: Microsoft's SAGE system uses game theory for whitebox fuzz testing
* Benefit: More efficient discovery of edge cases and security vulnerabilities

**1.5 Sam Ganzfried: “Introduction to Computational Game Theory”**

**1.6 Heads Up Texas Hold’em Rules**

* Heads Up Texas Hold’em is played between two players, with one player designated as the dealer (button).
* The dealer position rotates after each hand.
* The player on the button posts the small blind, and the other player posts the big blind - twice the small blind.
* The dealer posts the small blind and acts first on the pre-flop betting round
* The non-dealer posts the big blind and acts first on all post-flop betting rounds

[SHOW PICTURE OF BB & SB]

**1.7 Hand Rankings**

* Royal Flush - Odds - 649,739 : 1 - 0.000154%
* Straight Flush - Odds - 72,192 : 1 - 0.00139%
* Four of a Kind - Odds - 4,164 : 1 - 0.0240%
* Full House - Odds - 693 : 1 - 0.1441%
* Flush - Odds - 508 : 1 - 0.1965%
* Straight - Odds - 254 : 1 - 0.3925%
* Three of a Kind - Odds - 46.3 : 1 - 2.1128%
* Two Pair - Odds - 20.0 : 1 - 4.7539%
* One Pair - Odds - 1.37 : 1 - 42.2569%
* High Card - Odds - 0.995 : 1 - 50.1177%

**1.7.1 Basic Poker Principles**

Strength Principle

* Raising with Strong Hands
* Checking with Middling Hands
* Folding/Bluffing Weak Hands

Playing Styles

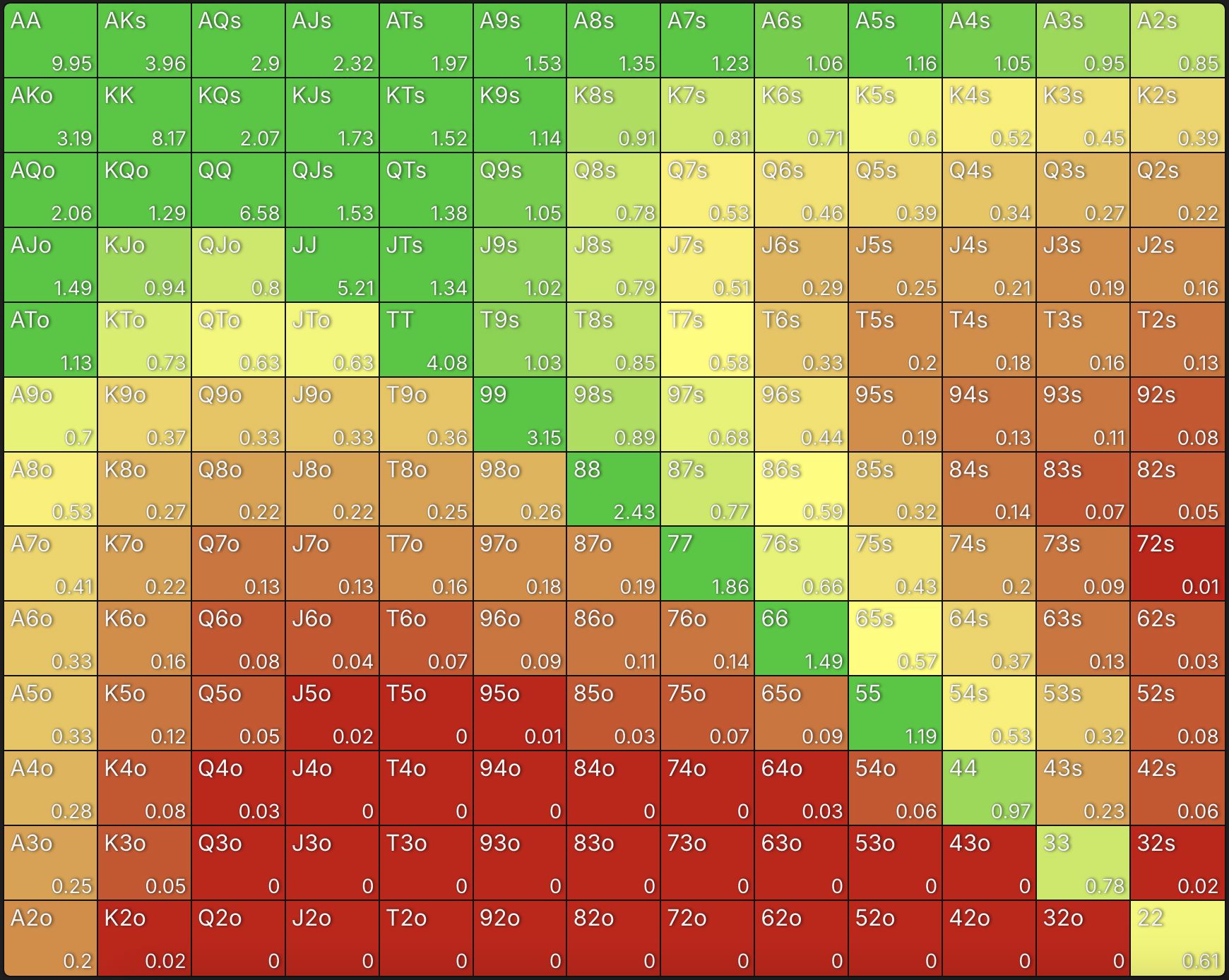
* Tight vs Loose
* Aggressive vs Passive
* In general, be tight and aggressive

Purpose of Betting

* Value Betting → want weaker hands to call
* Bluffing → want stronger hands to fold
* Betting for Protection→ want drawing hands to draw at unfavorable odds
* Sometimes bets can serve multiple purposes

Deception

* Easy to read = Exploitable
* Almost never want to do one action at 100% frequency
* Incorporate bluffs

**1.7.2 Hand Types** 

Starting Hand Types

* Paired vs Unpaired
* Suited vs Off-suit
* Connected vs Not connected
* High vs Low
* Best starting hands?

Drawing Hand

A hand that isn't strong yet but has potential to become very strong on future streets. Drawing hands typically need specific cards to improve, like holding 9♥10♥ on a K♥2♥4♣ flop - you have nothing now but could make a flush with any heart or hit a straight if J & Q were to come out on the turn and river.

Low Pair

When you hold or make a pair that's lower than any card on the board. For example, having pocket 5s (5♣5♦) on a K♠Q♦8♣ board gives you a low pair that's vulnerable to any higher pair.

Top Pair

Making a pair with the highest card on the board. For example; holding A♠K♦ on a K♣7♥2♦ flop - you've paired your king with the highest board card - and have an ace kicker.

The Nuts

The absolute best possible hand at any given moment that cannot be beaten. If the board shows K♥Q♥J♥T♥2♣, holding the A♥ gives you the nut flush, which is the nuts since no better hand is possible.

Total Air

A hand that has completely missed the board with no pair, no draw, and no potential. Having 7♣2♦ on a A♠K♥Q♦ board is total air - you have no pair, no straight draw, no flush draw, just high card seven.

A “Kicker”

A kicker is a side/secondary card that comes into play when players have the same primary hand (like the same pair) to determine who wins. For example, if you hold A♠K♥ and your opponent holds A♣7♦ on a board of A♦5♣2♥, you both have a pair of aces, but your king kicker beats their seven kicker. Kickers are especially important when playing with ace-high or top pair hands, as many players will have the same pair but different kickers, leading to situations where the player with the highest kicker wins the pot.

**1.8 Betting Round Structure**

1. Pre-Flop

* Each player receives two private cards (hole cards)
* First action is to the small blind who can fold, call, or raise
* The big blind can then fold, check (if no raise), call, or raise
* Betting continues until all bets are equal or all but one player folds

1. Flop

* Three community cards are dealt face-up on the board
* First action is to the non-dealer
* Players can check, bet, raise, or fold
* Betting continues until all bets are equal or all but one player folds

1. Turn

* A fourth community card is dealt face-up
* Betting round proceeds same as flop

1. River

* The fifth and final community card is dealt face-up
* Final betting round proceeds same as previous streets

Betting Rules:

* Players can bet any amount from one big blind up to their entire stack
* The minimum raise must be at least the size of the previous bet or raise
* There is no maximum number of raises permitted
* If a player doesn't have enough chips to call a bet, they can go "all-in" with their remaining chips

Winning the Hand:

* If a player folds, the remaining player wins the pot
* If the players reach showdown, best five-card poker hand using any combination of hole cards and community cards wins
* In case of identical hands, the pot is split

**1.8.1 Pokerbots Game Iteration**

* 1000 rounds
* 400 chip stack, resets each round
* Small/big blinds: 1/2
* Chip profits/losses from all the rounds are totaled to determine winner of match

**1.9 Overview of Skeleton Bot Architecture**

Engine Structure

* Test\_engine.py - RoundState, Player, Game
* config.py - engine and testing settings configuration

Skeleton

* States.py: GameState, RoundState classes
* Bot.py: handle\_new\_round(), handle\_round\_over(), get\_action()
* Runner.py: communicates with game engine
* Player.py: the main brain of the bot for decision making

**1.9.1 ABC\_Bot**

* Relies on calculating hand strength to make decisions.
* In the get\_action() function the bot calculates the Expected Value (EV) of each action using the formula: EV = (Probability of Winning x Pot Size) - Cost of action.
* When EV > 0, the bot will raise with an amount proportional to the EV/pot\_size ratio. If it can’t raise it will call, if it can’t call it will check.
* When EV is negative but better than folding (EV > - cost\_to\_call), the bot will check, if it can’t check it will call, if not fold.
* When EV is worse than folding (EV < -cost\_to\_call), the bot will check, otherwise fold.

**1.9.2 OpenAI\_Bot**

* Connected to OpenAI API, can be reworked to use with other LLM APIs.
* Provides context to LLM about game state information and guides LLM response in returning poker actions.
* get\_action() creates a message with the Player’s Hole Cards, Community Cards, Stack Sizes, Current Pot Size, Cost to Call, Available Legal Actions, and Raise Limits.
* The bot also maintains a conversation history with the LLM through the messages list which includes: A system message defining the LLM’s role, Game Rules and Constraints, The bot’s acknowledgement of the rules, and ongoing game state and action updates.

**1.10 Testing Your Bot**

Show local test development

Show Scrimmage Server

Weekly scrimmages for progress

Upload Sample Bot 1 to scrimmage server