Hidden Hands: The Skeleton of a Poker Simulator

For this project, I set out to build a poker probability simulator – an application that would calculate probability of winning for a first-person player (aka "Hero") based upon the cards visible to Hero and the actions of an opponent. However, it quickly became clear to me that this project was beyond what I could accomplish in one week; so I scaled down the project to build a scaled down calculator that creates some basic utilities needed to build a full poker simulator later and that demonstrates some basic probabilistic aspects of the game of poker. [I put important poker terms in bold below.]

My simulator (called "Hidden Hands") is based upon a simplified version of Texas Hold'em. **Texas Hold'em** is a poker game where each player tries to make the best five-card poker hand using a combination of two private cards and five shared community cards.

- Each player receives **two private face-down cards**, called **hole cards**. Only that player sees these cards. The first-person player is called the **Hero** and his/her cards are referred to as **Hero Cards**.
- Then five shared cards, called **community cards** or **board cards**, are dealt face up in the middle of the table. Every player can use these cards.
- In total each player can see a total of 7 cards and can use 5 to create his/her best poker hand.

The board cards are revealed in stages:

1. **Pre-flop**: Each player receives two private hole cardss
2. **Flop**: Three community cards are dealt face up.
3. **Turn**: A fourth community card is dealt face up.
4. **River**: A fifth and final community card is dealt face up.

Players can bet during each stage. A player must choose one of the following actions:

check = decline to bet if no bet is required
bet = put money into the pot
call = match another player's bet
raise = increase the bet
fold = give up the hand – foregoing current/future bets, but also any chance to win

At the end, if more than one player remains, there is a **showdown**. Each player makes the best possible five-card hand using any combination of their 2 hole cards + the 5 community cards. So each player has 7 total cards available, but only the best 5-card hand counts.

This simplified app focuses only on **flush draw scenarios**. A flush is five cards of the same suit. A flush draw is an incomplete hand that can become a flush if future cards of the same suit are dealt to the player – or in the case of Teas Hold'em, if the necessary community cards are turned over after the Flop, on the Turn, or on the River.. A **Frontdoor Flush Draw** means that the Hero already has four cards of the same suit and needs at least one more of that same suit to complete a flush. A **Backdoor Flush Draw** means Hero has only three cards of a suit on the Flop and needs BOTH the Turn and River to be that same suit to complete a flush. It computes the probability that a player completes a flush draw, compares this exact calculated probability to a Monte Carlo simulation, uses pot odds and expected value to make a simplified call/fold recommendation, and applies Bayes' theorem to update the probability that an opponent has a strong hand after observing his/her action.

My current demo runs **five scenarios (scenarios.py)**, including two frontdoor flush draws, a dead-card example (a dead-card happens when a card is accidentally exposed by another player, e.g. when folding), a large-bet expected value example, and a backdoor flush draw example. The final command

line interface ("CLI") output shows the program running "scenario 1 of 5" through "scenario 5 of 5," summarizing the probability concepts for each scenario.

To build this simulator, I built the following Python functions:

- Card utilities (cards.py) that create a 52-card deck, a validator to validate card names, functions that remove known cards from the deck, and functions that count the number of unknown cards and the number of unknown cards of the target suit. These utilities can be used as the foundation for a full poker simulator later, if and when I am able to complete this app.
- An exact probability calculator (draws.py) that computes exact probabilities for frontdoor and backdoor flush draws. This function uses the complement rule for frontdoor flush draws. A successful completion of a flush is called a Hit. Pr(Hit) = 1 – Pr(Miss every future card). As an example, if Hero Cards (Hero's two Hole cards) are As and Ks (Ace and King of Spades) and the Board or Community Cards after the Flop are Qs, Js, and 2d (Queen and Jack of Spades and 2 of Diamonds), then Hero has 4 spades between his hole cards and the board and thus only needs one more spade on the Turn or the River. There are now 47 unknown cards left (52- 5) and 38 unknown non-Spades (47-9). So Hero's probability of not drawing a spade on the Turn is (38/47) and then not drawing a spade on the River is (37/46). So the Hero's probability of success = 1 – (38/47)(37/46) = 0.3497.

  The exact probability calculator also handles backdoor flush draws. For example, if Hero Cards are As and Ks and the Board after the Flop is Qs, 7d, and 2c, then Hero has only 3 spades and needs both the Turn and River to be spades. There are 47 unknown cards left and 10 remaining spades. Thus, Hero's probability of drawing a spade on the Turn is (10/47), and then drawing another spade on the River is (9/46). So Hero's probability of success = (10/47)(9/46) = 0.0416, or 4.16%.
- A Monte Carlo simulation function (monte_carlo.py) that estimates these same probabilities by removing known cards from the deck, randomly dealing future cards, checking whether the draw succeeds, counting the number of successes, and then dividing this count by the number of simulations. For example, if the program runs 10,000 samples and records 409 "hits" (aka successes), then this produces a Monte Carlo estimate of 0.0409. This program also uses the Monte Carlo simulations to calculate a 95% confidence interval around the sample estimate. For example, this 95% confidence interval might be 0.0370 to 0.0448. In this case, the exact probability of 0.0416 falls inside the confidence interval, validating my Monte Carlo simulation function.
- An expected value and pot odds function (ev.py). This function begins to connect probability to decision-making, which is the essence of poker. This function specifically calculates the breakeven probability that Hero needs to breakeven from calling – or matching the bet required to continue or complete the game. EV(Call) = Pr(Hit/Success) * Final Pot Value – Call Amount. For example:
  - Final Pot After Call: $180
  - Call Amount: $ 40
  - Required Prob: 22.22% (=40/180)
  - Hero Success Prob: 35%
  - EV(Call) $23 (= .35 * 180 - 40)
  - Recommendation: Call because EV(Call) is positive

  Or a different example:
  - Final Pot After Call: $180
  - Call Amount: $ 40

    - Required Prob: 22.22% (=40/180)
    - Hero Success Prob: 10%
    - EV(Call) -$22 (= .10 * 180 - 40)
    - Recommendation: Fold because EV(Call) is negative
- A Bayesian opponent update function (bayes.py) that uses a very simple Bayesian model of opponent strength to update prior beliefs based upon observed actions. My prior and conditional probabilities are created simply to illustrate how this function might work in a more sophisticated model. Specifically, an opponent is in one of two hidden states: strong or weak. This model starts with a prior, Pr(Opponent = Strong) = 0.30. This function then has two likelihood assumptions: Pr(Raise | Strong) = 0.70 and Pr(Raise | Weak) = 0.20. Using Bayes' Theorem, this function updates the posterior using:
    - Pr(Strong | Raise) = Pr(Raise | Strong)*Pr(Strong) / Pr(Raise)
- In addition, I generated a series of tests, one set for each file above.

Hidden Hands is not a full poker probability tool/simulator. It cannot evaluate full poker hands; it only computes flush draw probabilities. It cannot calculate true showdown equity (where showdown equity is defined as expected share of pot given that the game goes to a final showdown). Instead, this project estimates the probability that a draw completes a flush – it cannot calculate the true probability of winning a poker hand versus an opponent.

To complete this project, I will need to create a additional functions. First would be a "hand evaluator" function that would be able to compare multiple poker hands and determine which is best, or the winner, including the ability to handle tiebreakers (i.e. if two hands are two pairs, which is the winner?). Second, I would build a full Monte Carlo equity simulator, which would be able to deal complete games to both Hero and his/her opponent(s), determine the winner in each sample, and accumulate the counts to determine the probability of Hero win, loss, or tie from any point of a game:

- Pr(Hero wins) = hero wins / trials
- Pr(Hero ties) = hero ties / trials
- Pr(Hero loses) = hero losses / trials

where Hero Equity = Pr(Hero wins) + 0.5 * Pr(Hero ties). After completing these two functions, I would do some work to determine whether it would be feasible to generate exact probabilities of drawing specific hand types (from among straight flush, four of a kind, full house, flush, straight, three of a kind, two pair, one pair, and high card).

I would then use the odds above to determine the optimal actions for each player at each step of the game. But the optimal action would not necessarily be to play the odds/probabilities – I would test "stochastic" strategies that would entail bluffing, or making each player's actions less predictable, perhaps based upon visible cards as well as private cards and perhaps based upon some degree of randomization where the randomization probabilities are a function of the simulated probabilities of winning given each player's hand.

I used an LLM as a programming tutor and a debugging assistant. In particular, the LLM helped me learn how to create project setup and configuration files, including a configuration for pytest that acted as a test harness. It also helped me learn how to use Git to store my most current code. It helped me plan the project structure, reason through probability formulas, write and revise tests, and debug Python/Git issues. I ran the code myself, tested the program, made the project scope decisions, and reviewed the final output.

Appendix – Model Output

```
MacBook-Air-8:hidden_hands williamduhamel$ python3 -m hidden_hands.cli

================================================================================
==
Hidden Hands: Poker Probability Tutor
================================================================================
==
This demo runs a set of poker probability teaching scenarios.
Each scenario uses the same probability engine but different inputs.

Running scenario 1 of 5

================================================================================
==
Scenario: Frontdoor Spade Flush Draw
================================================================================
==
Hero has four spades and needs one more spade by the river.

Cards and betting:
    Hero cards: As Ks
    Board cards: Qs Js 2d
    Extra known cards: none
    Draw type: frontdoor flush draw
    Target suit: spades
    Cards to come: 2
    Pot before opponent bet: $100.00
    Opponent bet: $40.00
    Opponent action: raise

================================================================================
==
1. Discrete and Conditional Probability
================================================================================
==
Known cards: As Ks Qs Js 2d
Known cards count: 5
Unknown cards remaining: 47
Spades already visible: 4
Spades remaining: 9

Probability lesson:
    We condition on the cards we can see.
    Known cards are removed from the deck before computing
probabilities.
```

```
================================================================================
==
2. Exact Draw Probability
================================================================================
==
P(spades on next card): 19.15%
P(frontdoor flush by river): 34.97%

Probability lesson:
    A frontdoor flush draw needs at least one card of the target suit.
    We use the complement rule:
        P(hit) = 1 - P(miss every future card)

================================================================================
==
3. Monte Carlo Simulation
================================================================================
==
Trials: 10000
Hits: 3459
Monte Carlo estimate: 34.59%
95% confidence interval:
    lower: 33.66%
    upper: 35.52%

Probability lesson:
    Each simulation randomly deals the remaining cards.
    The Monte Carlo estimate is the sample mean of hit/miss outcomes.

================================================================================
==
4. Pot Odds and Expected Value
================================================================================
==
Final pot after call: $180.00
Call amount: $40.00
Required probability to call: 22.22%
Hero success probability: 34.97%
EV(call): $22.94
Recommendation: CALL

Decision lesson:
    Pot odds tell us the break-even probability needed to call.
    Expected value compares our probability of success to the price.

================================================================================
==
5. Bayesian Opponent Update
================================================================================
==
```

```
Prior P(opponent strong): 30.00%
P(raise | strong): 70.00%
P(raise | weak): 20.00%
P(raise): 35.00%
Posterior P(opponent strong | raise): 60.00%

Bayes lesson:
    Opponent actions are evidence.
    Since some actions are more likely from strong hands than weak
hands,
    observing an action updates our belief about opponent strength.

Running scenario 2 of 5

================================================================
==
Scenario: Frontdoor Spade Flush Draw With One Dead Spade
================================================================
==
Same basic flush draw, but one additional spade is known to be
unavailable.

Cards and betting:
    Hero cards: As Ks
    Board cards: Qs Js 2d
    Extra known cards: 7s
    Draw type: frontdoor flush draw
    Target suit: spades
    Cards to come: 2
    Pot before opponent bet: $100.00
    Opponent bet: $40.00
    Opponent action: raise

================================================================
==
1. Discrete and Conditional Probability
================================================================
==
Known cards: As Ks Qs Js 2d 7s
Known cards count: 6
Unknown cards remaining: 46
Spades already visible: 5
Spades remaining: 8

Probability lesson:
    We condition on the cards we can see.
    Known cards are removed from the deck before computing
probabilities.
```

```
================================================================================
==
2. Exact Draw Probability
================================================================================
==
P(spades on next card): 17.39%
P(frontdoor flush by river): 32.08%

Probability lesson:
    A frontdoor flush draw needs at least one card of the target suit.
    We use the complement rule:
        P(hit) = 1 - P(miss every future card)

================================================================================
==
3. Monte Carlo Simulation
================================================================================
==
Trials: 10000
Hits: 3193
Monte Carlo estimate: 31.93%
95% confidence interval:
    lower: 31.02%
    upper: 32.84%

Probability lesson:
    Each simulation randomly deals the remaining cards.
    The Monte Carlo estimate is the sample mean of hit/miss outcomes.

================================================================================
==
4. Pot Odds and Expected Value
================================================================================
==
Final pot after call: $180.00
Call amount: $40.00
Required probability to call: 22.22%
Hero success probability: 32.08%
EV(call): $17.74
Recommendation: CALL

Decision lesson:
    Pot odds tell us the break-even probability needed to call.
    Expected value compares our probability of success to the price.

================================================================================
==
5. Bayesian Opponent Update
================================================================================
==
```

```
Prior P(opponent strong): 30.00%
P(raise | strong): 70.00%
P(raise | weak): 20.00%
P(raise): 35.00%
Posterior P(opponent strong | raise): 60.00%

Bayes lesson:
    Opponent actions are evidence.
    Since some actions are more likely from strong hands than weak
hands,
    observing an action updates our belief about opponent strength.

Running scenario 3 of 5

================================================================
==
Scenario: Frontdoor Heart Flush Draw
================================================================
==
Hero has four hearts and needs one more heart by the river.

Cards and betting:
    Hero cards: Ah Kh
    Board cards: Qh Jh 2d
    Extra known cards: none
    Draw type: frontdoor flush draw
    Target suit: hearts
    Cards to come: 2
    Pot before opponent bet: $120.00
    Opponent bet: $30.00
    Opponent action: raise

================================================================
==
1. Discrete and Conditional Probability
================================================================
==
Known cards: Ah Kh Qh Jh 2d
Known cards count: 5
Unknown cards remaining: 47
Hearts already visible: 4
Hearts remaining: 9

Probability lesson:
    We condition on the cards we can see.
    Known cards are removed from the deck before computing
probabilities.

================================================================
==
```

```
2. Exact Draw Probability
================================================================================
==
P(hearts on next card): 19.15%
P(frontdoor flush by river): 34.97%

Probability lesson:
    A frontdoor flush draw needs at least one card of the target suit.
    We use the complement rule:
        P(hit) = 1 - P(miss every future card)

================================================================================
==
3. Monte Carlo Simulation
================================================================================
==
Trials: 10000
Hits: 3439
Monte Carlo estimate: 34.39%
95% confidence interval:
    lower: 33.46%
    upper: 35.32%

Probability lesson:
    Each simulation randomly deals the remaining cards.
    The Monte Carlo estimate is the sample mean of hit/miss outcomes.

================================================================================
==
4. Pot Odds and Expected Value
================================================================================
==
Final pot after call: $180.00
Call amount: $30.00
Required probability to call: 16.67%
Hero success probability: 34.97%
EV(call): $32.94
Recommendation: CALL

Decision lesson:
    Pot odds tell us the break-even probability needed to call.
    Expected value compares our probability of success to the price.

================================================================================
==
5. Bayesian Opponent Update
================================================================================
==
Prior P(opponent strong): 30.00%
P(raise | strong): 70.00%
```

```
P(raise | weak): 20.00%
P(raise): 35.00%
Posterior P(opponent strong | raise): 60.00%

Bayes lesson:
    Opponent actions are evidence.
    Since some actions are more likely from strong hands than weak
hands,
    observing an action updates our belief about opponent strength.

Running scenario 4 of 5

================================================================================
Scenario: Frontdoor Diamond Flush Draw Against Large Bet
================================================================================
Hero has a diamond flush draw, but the opponent makes a larger bet.

Cards and betting:
    Hero cards: Ad Kd
    Board cards: Qd Jd 2c
    Extra known cards: none
    Draw type: frontdoor flush draw
    Target suit: diamonds
    Cards to come: 2
    Pot before opponent bet: $100.00
    Opponent bet: $120.00
    Opponent action: raise

================================================================================
1. Discrete and Conditional Probability
================================================================================
Known cards: Ad Kd Qd Jd 2c
Known cards count: 5
Unknown cards remaining: 47
Diamonds already visible: 4
Diamonds remaining: 9

Probability lesson:
    We condition on the cards we can see.
    Known cards are removed from the deck before computing
probabilities.

================================================================================
2. Exact Draw Probability
```

```
================================================================================
==
P(diamonds on next card): 19.15%
P(frontdoor flush by river): 34.97%

Probability lesson:
    A frontdoor flush draw needs at least one card of the target suit.
    We use the complement rule:
        P(hit) = 1 - P(miss every future card)

================================================================================
==
3. Monte Carlo Simulation
================================================================================
==
Trials: 10000
Hits: 3506
Monte Carlo estimate: 35.06%
95% confidence interval:
    lower: 34.12%
    upper: 36.00%

Probability lesson:
    Each simulation randomly deals the remaining cards.
    The Monte Carlo estimate is the sample mean of hit/miss outcomes.

================================================================================
==
4. Pot Odds and Expected Value
================================================================================
==
Final pot after call: $340.00
Call amount: $120.00
Required probability to call: 35.29%
Hero success probability: 34.97%
EV(call): $-1.11
Recommendation: FOLD

Decision lesson:
    Pot odds tell us the break-even probability needed to call.
    Expected value compares our probability of success to the price.

================================================================================
==
5. Bayesian Opponent Update
================================================================================
==
Prior P(opponent strong): 30.00%
P(raise | strong): 70.00%
P(raise | weak): 20.00%
```

```
P(raise): 35.00%
Posterior P(opponent strong | raise): 60.00%

Bayes lesson:
    Opponent actions are evidence.
    Since some actions are more likely from strong hands than weak
hands,
    observing an action updates our belief about opponent strength.

Running scenario 5 of 5

================================================================
==
Scenario: Backdoor Spade Flush Draw
================================================================
==
Hero has three spades and needs both the turn and river to be spades.

Cards and betting:
    Hero cards: As Ks
    Board cards: Qs 7d 2c
    Extra known cards: none
    Draw type: backdoor flush draw
    Target suit: spades
    Cards to come: 2
    Pot before opponent bet: $100.00
    Opponent bet: $20.00
    Opponent action: check

================================================================
==
1. Discrete and Conditional Probability
================================================================
==
Known cards: As Ks Qs 7d 2c
Known cards count: 5
Unknown cards remaining: 47
Spades already visible: 3
Spades remaining: 10

Probability lesson:
    We condition on the cards we can see.
    Known cards are removed from the deck before computing
probabilities.

================================================================
==
2. Exact Draw Probability
================================================================
==
```

```
P(spades on next card): 21.28%
P(backdoor flush by river): 4.16%

Probability lesson:
    A backdoor flush draw needs every future card to be the target
suit.
    From the flop, that means target suit on both turn and river.

================================================================
==
3. Monte Carlo Simulation
================================================================
==
Trials: 10000
Hits: 409
Monte Carlo estimate: 4.09%
95% confidence interval:
    lower: 3.70%
    upper: 4.48%

Probability lesson:
    Each simulation randomly deals the remaining cards.
    The Monte Carlo estimate is the sample mean of hit/miss outcomes.

================================================================
==
4. Pot Odds and Expected Value
================================================================
==
Final pot after call: $140.00
Call amount: $20.00
Required probability to call: 14.29%
Hero success probability: 4.16%
EV(call): $-14.17
Recommendation: FOLD

Decision lesson:
    Pot odds tell us the break-even probability needed to call.
    Expected value compares our probability of success to the price.

================================================================
==
5. Bayesian Opponent Update
================================================================
==
Prior P(opponent strong): 30.00%
P(check | strong): 25.00%
P(check | weak): 65.00%
P(check): 53.00%
Posterior P(opponent strong | check): 14.15%
```

```
Bayes lesson:
    Opponent actions are evidence.
    Since some actions are more likely from strong hands than weak
hands,
    observing an action updates our belief about opponent strength.

======================================================================
==
Summary
======================================================================
==
Hidden Hands demonstrates:
    1. discrete probability
    2. conditional probability
    3. exact counting
    4. Monte Carlo simulation
    5. confidence intervals
    6. expected value
    7. Bayes theorem

Important limitation:
    This is an educational probability demo, not a complete poker
solver.
    It assumes that hitting the draw wins the pot, which is a
    simplifying assumption for teaching expected value.
```