IT2105, Autumn 2006, Final Project: Texas Hold 'Em

1 Overview

Your assignment is to design and implement a poker-playing simulator for the (very popular) game of Texas Hold 'Em. A complete description of the game can be found at any number of web sites, including www.pokerhandsandrules.com/pokerrules.htm. You must use the programming language Scheme or any other variant of LISP for all aspects of the assignment. The project can be done alone or in groups of 2 students (but no more than 2).

There are 3 phases to the project, worth 10, 10 and 5 points, respectively.

1. The basic simulator for k-player Texas Hold 'Em
2. A *rollout simulator* to assist in the decision making needed to place bets.
3. A *player profiler* to keep track of each opponent's betting behavior and then influence the current betting.

2 Phase I: The Basic Texas Hold 'Em Simulator

As a short summary of the essentials of the game, each game consists of many *hands* and each hand consists of several steps:

1. The two players to the immediate left of the dealer put in small wagers, known as *blinds*, to insure that each player contributes some money to the pot, even when they have bad cards. **You can safely ignore this step in your simulator**.
2. Two *hole cards* are dealt to each player. These are not visible to the other players.
3. A round of betting is performed, with the player to the immediate left of the dealer beginning.
4. 3 shared cards, *the flop*, are dealt face up in the middle of the table. All players can use these, in combination with their hole cards, to make a 5-card poker hand.
5. A second round of betting is performed.
6. The 4th shared card, *the turn* is dealt face-up and added to the flop. Each player now makes a 5-card poker hand from these 6 cards: the flop, turn and the two hole cards.
7. A third round of betting is performed.
8. The 5th shared card, *the river* is dealt face-up and added to the flop and turn. Each player now makes a 5-card poker hand from these 7 cards: the flop, turn, river and the two hole cards.
9. A final round of betting is performed.
10. If 2 or more players have still not folded, they must show their hole cards. This is often called a *showdown*. The player with the highest-rated hand wins the entire pot. On ties, the pot is split among the winners. If all players except one fold, then that player is **not** required to reveal her hole cards before collecting the pot.

2.1 Betting

During betting, each player has the option to *fold*, *call* or *raise* the current bet. By folding, the player forfeits the current hand, thus losing any money that she has contributed to the pot so far. To call, the player matches the current bet. For example, if the bet for the current betting round is 100, and a player has already contributed 70 during that round, then 30 is needed to call.

When raising, a player contributes **more** than the current bet, thus increasing that bet. This forces all other players to contribute even more money to the pot if they wish to remain in the hand.

At the beginning of each betting round, the slate is wiped clean, the current bet is set to 0, and previous contributions to the pot have no effect upon the new round.

In simulation, it is wise to include a *maximum bet* parameter such that the betting in any given round does not escalate to unreasonable amounts.

The act of *bluffing* in poker involves aggressive betting when one has relatively bad cards. A bluffer will normally raise the bet in hopes of getting all other players to fold. The bluffer knows that if anyone continues to call her through all the betting rounds, she will probably lose. If everyone does fold, the bluffer need not show her hole cards, and thus nobody knows for sure if they have been bluffed. Bluffers are only exposed when they get caught in a showdown.

Good poker players will a) bluff occasionally, or even quite often, b) take note of how often other players bluff, and c) proceed to showdowns (largely by calling, not raising) even with relatively bad cards in order to force opponents to show their hole cards. Good poker players are experts at attaining, remembering, and using information about their opponents playing styles.

2.2 Simulation Details

Your simulator should accept any number of players. All can be computer players, although you are free to include options for one or a few human players.

Each player should begin with a sum of money and be allowed to play as long as desired. It is fine if players *go into debt* by having negative money; they can continue to play and place the same size bets as more successful players.

During each betting round, each remaining player (i.e., one who has not folded) should assess the power of their hand (using the power ratings from earlier homework assignments). This and any other information should be used to decide upon the current betting decision: fold, call or raise. In phase I of the project, this decision can be trivial, but it must at least be based on the hand's power rating. Note that in the first betting round, the power rating is based on only 2 cards, but after the flop, each player will have a 5-card hand to assess.

Though concepts such as *left of the dealer* may not make much sense in simulation, you should have some concept of playing order, and it should be varied with each hand. You should either a) keep a fixed ordering of players and simply alternate among who bets first in each hand, or b) randomly shuffle the player ordering after each hand.

There is no need to designate a simulated player as *dealer*. Rather, you will probably want to have a main

poker-manager object that handles all dealing and shuffling activities. A newly shuffled 52-card deck should be used for each hand.

The concept of *showing cards* during a showdown is also irrelevant in Phase I. The poker manager can simply compare the power ratings and distribute the pot to the winner(s). However, in Phase III of the assignment, the creation of player profiles involves taking account of the hands (and bluffs) that were revealed in showdowns.

The state of the table in terms of the shared cards, all hole cards, and the pot, should be displayed prior to each betting round. Then, during the betting, each player action should be displayed, as should the round's current bet. A fancy graphic interface is unnecessary, but all of this important game information should be printed in the command window.

In general, efficiency is not an important aspect of this assignment. Your simulator need only play a few hands of Hold 'Em during the demo. In phase I, time should be no problem: your simulator will probably be able to play thousands of hands in a few seconds. But in phase II, your system may only be able to play a few hands per minute due to the computational expense of rollout simulation.

Warning: If your hand-power-assessment code uses any list-sorting procedures that destructively modify card lists, then be sure to make copies of cards, particularly the shared flop, turn and river cards, when computing power ratings. Otherwise, you will run into all sorts of problems.

3 The Roll-Out Simulator

In Phase I, hand strength is based purely on the power rating of a player's **actual** cards, but good poker players think well beyond the present situation to the **potential future situations**. In these scenarios, the player assesses potential completions of her own hand and compares these to the possible hands that opponents may have. There are many ways to do these predictions and comparisons in a simulator. You will do one of the simplest.

In a roll-out simulator, each player has very limited knowledge: its own hole cards plus the shared cards. From this, it must come up with a reasonable estimate of the winning potential of those hole cards as compared to the possible hole cards that opponents may have.

For example, if the flop and turn consist of: $10\spadesuit$, $J\spadesuit$, $K\spadesuit$, $A\spadesuit$, while player P's hole cards are: $A\heartsuit$, $A\clubsuit$, then P can only be moderately optimistic, because even though she holds 3-of-a-kind Aces, there is a reasonably high probability that one of the other players holds the $Q\spadesuit$ and thus has a royal flush. This situation is depicted at the top of Figure 1.

To get some estimate of the winning probability for P, a roll-out simulator will deal many hands to the opponents (i.e., deal two hole cards to each, many times). In each round, the deck is reset to 52 cards minus player P's hole cards and minus the current shared cards. So in the example above, the reduced deck, D_r, would be every card except $10\spadesuit$, $J\spadesuit$, $K\spadesuit$, $A\spadesuit$, $A\heartsuit$, $A\clubsuit$. D_r is reshuffled after each round.

The bottom of Figure 1 shows the starting point for a roll-out simulation from the perspective of Player 1, after the turn card but before the river. In the figure, the question-marked cards are those to be dealt out during the simulation.

From this starting point, the question-marked cards are dealt out many times. After each dealing round, the

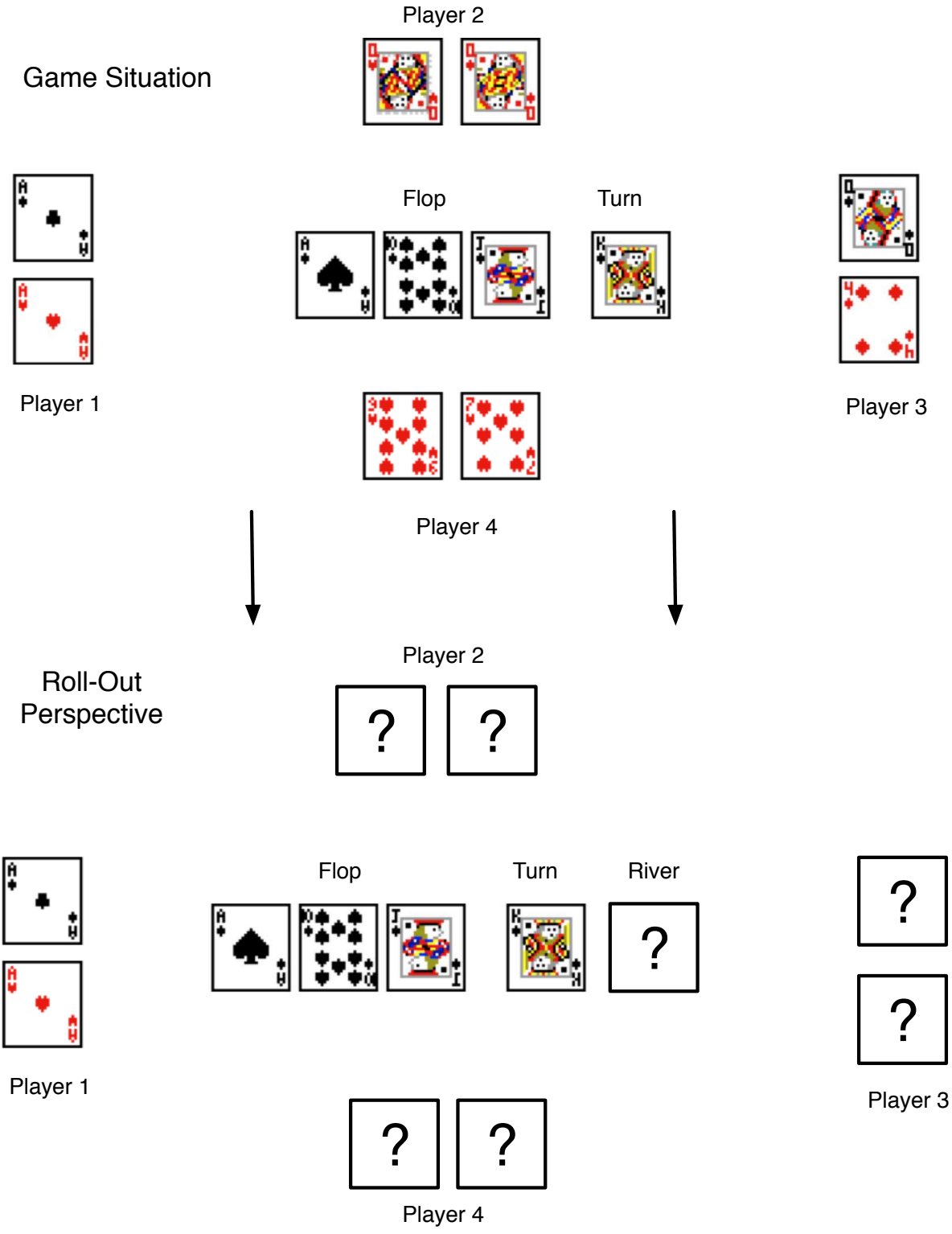

Figure 1: A game situation and the corresponding roll-out simulation starting point from the perspective of player 1.

resulting hands are compared for power, and the winner(s) is/are noted. The simulator keeps track of the number of cases in which the focal player, Player 1 in this case, has the best hand. The larger that number, the better grounds that player has for calling and raising bets.

This roll-out approach assumes that each opponent will remain in the hand until the showdown. Of course, this is somewhat unrealistic, but it provides a reasonably good first-cut estimate of a hand's potential. This approach also ignores all betting behavior, which, in a real game, gives some indication of opponent strength. These and other factors explain why this particular roll-out methodology is one of the simplest.

Each simulated player should perform a roll-out simulation, using his hole cards and the shared cards as the basis for D_r. For each player, a good number of roll-outs should be performed to get a reasonably accurate statistic. Although industrial-strength poker simulators use tens of thousands of roll-out rounds, you can get by with a few hundred for this project.

Thus, prior to each betting round, each player should perform a few hundred roll-out rounds to assess his own hand strength.

In Texas Hold 'Em, there are 4 betting rounds, where each is based on a slightly different set of visible (to any given player) cards. Hence, there are 4 opportunities for roll-out simulation:

1. After each player has received two hole cards, but before the flop, D_r consists of 50 cards: the full deck minus the focal player's hole cards. Using D_r, the simulator deals many rounds in which all opponents' hole cards as well as the flop, turn and river are dealt.
2. After the flop, D_r consists of 47 cards, since the flop cards are now *known* and no longer subject to stochastic prediction. However, the turn and river cards are still unknown and must be dealt out many times.
3. After the turn, D_r consists of 46 cards, since now the turn card is also known. Only the river and hole cards are dealt this time.
4. After the river, D_r consists of 45 cards, and now the simulator only needs to deal many rounds of hole cards to the opponents.

Your simulator should perform rollouts at each of these time points in the hand, and from the perspective of each of the simulated players. The percentage of roll-out wins, R, should then be a primary factor in making betting decisions. R replaces the less-informed hand-strength values used in Phase I. Note that R for any player, P, is the percentage of wins during the roll-outs **from P's perspective**. Statistics compiled during the roll-outs taken from his opponent's perspectives will not be known to P and thus cannot affect his betting decisions.

4 The Player Profiler

Up to this point, the system has essentially ignored any specific information about the opponents. Betting decisions are based purely on one's own cards and their potential strength compared to **possible** hands that opponents **may** hold. Of course, a player will never know her opponent's cards until an eventual showdown, but that information can be used to influence betting decisions in **future** hands.

For example, if a player is caught bluffing in a showdown, then good poker players will always note this and be wary of the bluffer, particularly when she makes a big raise in a future hand. Thus, they will be less

likely to fold when such a raise comes. A player's behavior is judged as a bluff in cases where they make one or more big raises during the betting rounds but then lose in a showdown with very weak cards. A bluffer cannot merely call with weak cards, since calling behavior rarely causes a lot of opponents to fold. Bluffers try to get everyone to fold in order to avoid a showdown and a highly probable loss.

Your player profiler will be a local component of each simulated player; it will keep statistics on all the other players. It need not be limited to bluffing behavior, although this is one of the most important factors to be noted. Other profiled traits may include general betting behavior. For example, players who fold a lot are often labeled as *conservative*: they don't like to risk playing when their cards are weak. When a conservative player does raise, it could signal that they have good cards. Conversely, a player who always seems to raise or call is less likely to arouse interest with yet another raise.

If you decide to base player behavior on money, then clearly, players with a lot of money have more freedom to play liberally: they can afford to hang around many hands until the showdown, and they risk very little by bluffing on occasion. Hence, a player might be less likely to fold when raised by a *rich* player.

Poker strategies, and poker psychology, are extremely complex subjects which you can only scratch the surface of in this project. However, to get credit for this portion of the project, you will need to cache some sort of player-behavior data and verify that it is being used in making betting decisions.

To showcase the player profiler, you will need to run your system through many hands, probably hundreds or thousands, in order to compile useful statistics on the different players. For these runs, you will probably want to use relatively few rollout rounds, say 50 or 100, depending upon the speed of your system.

You will probably want to include a *bluffing factor* in your simulation and give different values of this to each simulated player. Players with a high value will be more likely to raise on poor hands. If the other players have well-functioning profilers, then they should detect this general tendency after 50 hands or so and be less willing to fold when bets are raised by the bluffer.

In general, there is considerable room for creativity in this third phase of the project. Do whatever type of profiling seems useful, interesting, and fun.

5 Deliverables

Your entire system will be demonstrated at the end of November. For each portion of the project for which you intend to get credit, you must show a functioning program. A short report of approximately 5 pages must also accompany the project. The report should give a **high level** description of the system, with special focus on a) the basic structure of your code, and b) the logic behind the betting decisions made by the players in each of the three project phases. The report should not include code.