Luke Lavallee

ELEC3225-01

Poker Incremental Model

Preliminary Phase

1. Research rules of Poker
   1. Will use this to decide on 5 Card Stud of Texas Hold’em.
      1. Texas Hold’em has more movement and complexities, therefore I will choose 5 Card Stud
2. Research the Poker Python library
   1. Python has a prebuilt library to aide in constructing a poker game and therefore I believe I can work with it.
3. Continue learning basics of Python

Phase 1: Playing a hand

1. I want the first phase to simply play a hand. The game will start with 3 simulated players that all have the same behavior.
   1. Figure out how to instantiate and deal cards
      1. Must use multiple decks in casino poker
   2. Set up basic structure (i.e. – Ante, Deal, Bet, Trade, Bet, Show)
2. Set up player class to be elaborated on later
   1. Will hardcode 3 players in Version 1

Phase 2: Elaborate players for simulation

1. Since multiple styles of player must be implemented, create different kinds of players
   1. Randomize difficulty, buy in, balance on creation
      1. High difficulty = bets in a smart, systematic way, add bluffing
      2. Medium Difficulty = aggressive better but not very safe with money
      3. Low Difficulty = not aggressive, safe with money, folds often to conserve winnings
   2. Keep track of casino winnings as well as player winnings
2. Add winner selections
   1. This is easier than betting selections to implement and will start to show the flow of a game
   2. These formulas can be found online most likely after research into them
      1. They were eventually found and cited in code, cited again here: (http://pythonfiddle.com/poker-game/)

Phase 3: Implement Betting

1. The hardest thing to implement will be betting. First, extensive models will be made to simulate betting on paper.
2. The UML modeling will be implemented and manipulated as needed to design working flows for all three user types. They will be implemented as described in phase 2 (phase 2 was to build the structure to make the betting formulas work easier later on).
3. Players will bet by folding, raising or calling. If a raise is made after player 1, the loop must begin again until all players have called or folded.
4. These processes should be modulated for easier implementation in the two rounds pf betting required.
5. Regression test the winner selection to make sure it is still selecting the appropriate winner. With folding now in place, a player may not win automatically with the high hand if strategic betting came into play.

Phase 4: Create Simulation environment

1. Create a menu that allows the user to passively watch the game play in normal time or automatically simulate a desired number of games.
2. This will show statistics on the casino and all of the games players over time
   1. May be placed in a Database so that the leaderboard can reference it later
3. This will require more dynamic player instantiation. The player list must grow and shrink dynamically from the set list or 3 that testing was conducted in. This should not be hard but has been in mind when designing.