Homework 9:
Final Project, Phase 2:
Example of Analysis and Design

Programming Logic and Design
ITEC 1150
Spring Semester, 2014
Eric V. Level, Instructor

Here is an example of what I want for HW 9: the card game "Acey-Deucey" or "In-Between", which is a simple version of poker. Your game and design won't be this complex, likely…

Read the rules here: http://www.onlinepoker.net/Card-Games/In-Between.php

First design = "Functional analysis and design"

Acey-Deucey pseudocode:

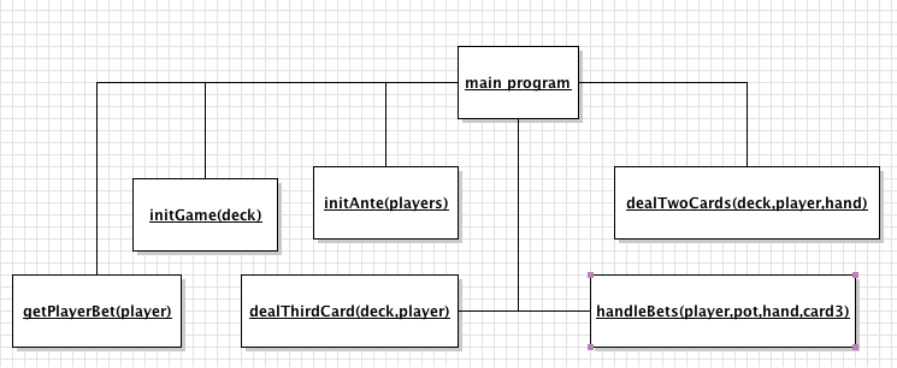
```
(0) Game deck initialized, shuffled (cards are ranked 1..13, with 1-
Ace, 11-Jack, 12-Queen, 13-King), number of players set, pot set to 0
and amount won by each player set to 0.

(1) Each player adds their ante (forced bet) of 1 chip to pot.

(2) For each player, play one round:

   (2.1) Dealer gives two face-up cards to player.
   (2.2) Player bets from 1..pot that next card will be numerically
            between first two, inclusive.
   (2.3) If player declines to bet (bet of 0), they fold and are out of
      the game with no bet added.
   (2.4) Dealer gives third face-up card to player.
   (2.5) Bets collected or paid according to rules; if third card is:

         -- Strictly in-between first two, player wins and takes
         bet from pot.
         -- Outside range of first two, player loses and adds bet
         to pot.
         -- Same as either two cards, player loses and adds double
         his bet to pot.

(3) Repeat 2 until (a) pot is empty or (b) all players fold.
```

Function candidates highlighted above.

Function header design:

```
def initGame(deck):
def initAnte(players):
def dealTwoCards(deck,player,hand):
def getPlayerBet(player):
def dealThirdCard(deck,player):
def handleBets(player,pot,hand,card3):
```

Structure chart:



Second design = "Domain object discovery"

Candidate classes, along with purpose ("representing") and attributes:

`Game`:
-- Represents one entire game in progress
-- Attributes and multiplicities:
`_players: Player(0..5)`
`Pot(1)`
`Deck(1)`

`Deck`:
-- Representing collection of Cards used to play game, with dealing and shuffling
-- Attributes and multiplicities:
`_cards: Card(0..51)`

`Card`:
-- Representing single card of given rank and suit
-- Attributes and multiplicities:
`_rank: 1..13 (11--Jack, 12--Queen, 13--King, 1--Ace)`
`_suit: 'Spades','Hearts','Diamonds','Clubs'`

`Player`:
-- Representing single player with name, amount won, and current hand (0,2,or 3 cards)
-- Attributes and multiplicities:
`_name: string(1)`
`_winnings: currency value(1)`
`_hand: current Hand(1)`

`Pot`:
-- Representing current pot with $
-- Attributes and multiplicities:
`_value: currency value(1)`

`Bet`:
-- Represent what player bets
-- Attributes and multiplicities:
`_amount: currency value(1)`

`Hand`:
-- Represents current cards for single player
-- Attributes and multiplicities:
`_cards: Card(0..3)`

UML Class Diagram (NOT required - but shown here as useful visual map):