CSE 341, Spring 2004, Assignment 7
Due: Friday 4 June, 9:00AM

Last updated: 20 May

In this assignment, you will write a Smalltalk player for a simple card game. You must also complete a short English description of your player. This assignment has several accompanying documents:

- A description of the game's rules.

- A description of the interface between your player and the classes for implementing the game.

- An email with your "player number".

The purpose of this document is to outline what you must do to receive 100% credit, how we will reward extra credit, etc.

Note: Clearly this assignment gives you considerably more flexibility than previous ones. That does *not* give you license to submit a jumbled mess of unreadable code. Style matters!

To receive full credit your solution must meet several criteria, which are explained in more detail below:

- Not have a Smalltalk error

- Not consume too much time or space

- Not submit illegal bids

- Not cheat

- Have all your classes named appropriately

- Have an accompanying description that concisely and accurately describes your code

- **Play "optimally" in several "guaranteed win" and "guaranteed lose" situations**

Not have a Smalltalk error: Your player should never cause a `MessageNotUnderstood` error to occur. That means your player should never send a bad message or interact with the provided code in an incorrect way.

Not consume too much time or space: Roughly, you should not worry about very much about efficiency. However:

- Do not go into infinite loops.

- Do not implement algorithms that time proportional to 2^n where n is the number of rounds or shuffles or cards in a deck because it is impractical to do so.

- While good players will maintain a "history" of what has happened in a match, do not build data structures that keep growing with every round forever. (It's fine to have them grow as a shuffle proceeds, but there may be many, many shuffles in a match. So any history about previous shuffles will have to be some sort of concise summary. For example, you might have a number representing how many times player-2 had the highest card, but you shouldn't have a list with all the rounds where player-2 had the highest card.)

Not submit illegal bids: Remember, if your bid is not "quit for 2" then your bid must be at least as high as any previous non-quitting bid. And it must be 2, 4, 6, 8, or 10.

Not cheat:

- You must not send messages to the dealer classes except for those described in the API. (Else you could cheat by reading other people's cards or reshuffling the deck!)

- You must not send messages to objects "belonging" to other players. All communication is with the dealer.

- You *may*, in fact, *should* use Squeak's extensive libraries. But you must not mutate things that would harm other players, such as removing classes, renaming methods, etc.

Have all your classes named appropriately:

- You should have a class named `PxxPlayer`. We will send `new` to this class to create your player.

- Every class you define should begin with `Pxx`.

Have an accompanying description: Compose a concise and accurate description of your player. Focus on the strategy your player implements and an overview of how you used the Squeak libraries and your own classes to implement the design. We except descriptions to be 1–2 pages, but these are not strict limits. You do not need to describe most individual methods in detail. You should assume your reader understands the game, Smalltalk, and the Squeak libraries.

Play optimally in certain situations:

1. Never be the third or fourth player in a round to "quit for two"

2. If the other 3 players swap during a round, you must do the following:

   (a) If you have yet to decide whether to swap and the dealer-card is highest-or-lowest and your card is neither highest-nor-lowest, then you must swap.

   (b) If you have yet to decide whether to swap and the dealer-card is neither highest-nor-lowest and your card is highest-or-lowest, then you must not swap.

   (c) If you have the highest-or-lowest, you must bid 10.

   (d) If you have neither highest-nor-lowest, you must "quit for 2".

3. If considering every card you have *seen* during the current shuffle, you must have the highest-or-lowest card, then you must not swap. (You have *seen* a card if you ever held it, it was ever the dealer-card, or a player who did not quit had the card at the end of a round.) Simple example: If you hold the 2 of hearts, but you saw the 2 of clubs and 2 of diamonds in earlier rounds, you know you have the lowest card.

4. Similar to the previous rule, if the dealer-card must be highest-or-lowest, then you must swap for it (unless your current card must also be highest-or-lowest).

5. If given the cards you have seen, you must have the highest-or-lowest, then you must bid 10.

Note: Item 2(d) prevents you from *bluffing* (trying to trick other players into thinking you have a good card when you don't) in certain situations. You still must meet the requirements above; there are plenty of other opportunities to bluff.

Note: In certain situations, you may be able to use the cards you have seen to determine you are guaranteed to lose. In this situation, it is wise to "quit for 2" but this is not required.

Late Days:

- If you use late days for this assignment, you will not participate in the tournament.

- Using late days is probably unwise because studying for finals is important.

Extra Credit: There are two separate ways to get a small amount of extra credit:

- Use and describe an especially interesting solution strategy (using good OO design) *and* do more than the "minimum" strategy for 100% credit.

- Finish in the top-3 in the tournament.

What to turn in:

- File-out your classes to a `.st` file.

- Rename your `.st` file to `lastname-hw7.st`, where `lastname` is your last name.

- Email your `.st` file to `daverich@cs.washington.edu` as an attachment.

- Include your description in the same email.

- Your description may be in the body of your email, a text attachment, a PDF attachment, a Postscript attachment, or a Microsoft Word file. For other formats, ask ahead of time.