### First Project Deliverable: Requirements Specification (RS)

For your class project you will develop four card games: two versions of solitaire and two versions of rummy. Although you will eventually create four executables, much of the code will be common across the four games.

For the first Project Deliverable each group will write a Requirements Specification (RS) for two of the card games. Even-numbered groups will write a requirements specification covering the two solitaire games, and odd-numbered groups will write a requirements specification covering the two rummy games.

Although your group will be developing just part of these four games, for this assignment your group will write the RS for two of the games from start to finish. Information on the games is given below.

Use IEEE Standard 830 as the format for your RS. Use italicized text to record open issues in the requirements. These will of course have to be resolved as the project progresses.

Your group should turn in a hardcopy of your deliverable (the RS). Attach to the deliverable your meeting minutes. (And you probably will meet more than once for each deliverable, so each meeting should have minutes.)

Each person should turn in an individual report, but do not attach it to your group’s deliverable. (See the syllabus for more details about the meeting minutes and individual report.)

### Solitaire Games

Solitaire games are for a single player. You will develop both standard (klondike) solitaire and spider solitaire.

Here are version of both games to try:

<http://www.klondikesolitaire.net/>

<http://games.aarp.org/games/spider-solitaire/>

Here are some sample rules for the games:

<http://www.123freesolitaire.com/games/klondike.htm>

<http://www.123freesolitaire.com/games/spider_solitaire_two_suits.htm>

Your games should behave similarly, with a few exceptions: the game may have a slightly different look, you will use just mouse clicks to move cards, not click-and-slide (drag and drop), and you’ll save the game as follows.

To save a game, there will be a button to click. When the button is clicked, the game is saved in its current state to a file. To keep it simple for now, the file is placed in the local directory, and its name is automatically generated and contains the date and time. The name of the game should be the first thing in the file.

To load a saved game, the name of the file containing the game is given as a command line argument when starting up the game. Make sure the saved game is the right game before trying to read it in. When no command line argument is given, the program always starts up a new game.

### Rummy Games

The rummy games are normally for two or more people, but both your games will be for exactly two players, one of which is the ”AI” player. Here is a version of gin rummy to try out the game:

<https://www.aol.com/games/play/masque-publishing/gin-rummy/>

<https://www.solitaireonline.com/game/Gin+Rummy>

For the specific rules for each game, use the following web sites.

Outline of rules for rummy:

<http://www.pagat.com/rummy/rummy.html#basic>

Outline of rules for gin rummy:

<http://www.pagat.com/rummy/ginrummy.html>

The behavior of the AI player can be pretty simple; it just needs to be good enough so that the other player doesn’t always win.

Use only mouse clicks for playing the game—not click-and-drag.

To save a game, there will be a button to click. When the button is clicked, the game is saved in its current state to a file. To keep it simple for now, the file is placed in the local directory, and its name is automatically generated and contains the date and time. The name of the game should be the first thing in the file.

To load a saved game, the name of the file containing the game is given as a command line argument when starting up the game. Make sure the saved game is the right game before trying to read it in. When no command line argument is given, the program always starts up a new game.