# EECS 22L: Project 1 Grading Criteria

Prepared by: Weiwei Chen, Che-Wei Chang, and Prof. Rainer Dömer

February 04, 2013

**The following functions are mandatory for the *final* version of the *Chess* program:**

1. A user interface for the game
2. Follows the rules of the Chess game:
   - Allow all the legal moves, including the special ones, and can make the moves
   - Recognize illegal moves, and NOT make the moves (warn the user if necessary)
   - Recognize the “checkmate” situation and declare the winner
3. The game supports four modes:
   - player vs. player
   - player vs. computer (player moves first and takes the white pieces)
   - computer vs. player (computer moves first and takes the white pieces)
   - computer vs. computer
4. The program keeps a log of all the moves
5. The program can load a specific board setup and continue to play in the aforementioned four modes (user will make the choice)
6. The program can support different levels of players
7. The auto player (computer) makes quick moves

**The following items are mandatory for deliverables of the *final* version of the *Chess* program:**

1. A complete and clean tarball, namely `Chess_Source.tar.gz` as the project source deliverable with:
   - Proper project file hierarchy (as what was presented in week3’s discussion session)
   - An `INSTALL` file with the descriptions of two installation options:
     - tarball extraction
     - CVS checkout information, i.e. linux command that can be used to get the project checkout

     Proper instructions on how to install the program is also expected.
   - A `README` file with the information of the authors, program version, date, and general information / description about the software
   - A `COPYRIGHT` file with authors and copyright information
   - A top-level `Makefile` with at least three targets, i.e. ‘all’, ‘test’, ‘install’, and ‘clean’.  
     **Note:** this file is different from the `Makefile` in the *src* directory.
   - The `src` directory with all the properly documented program source code files.
   - The `bin` directory where the binary executable file of the `Chess` game and the related resources, such as images for the pieces, will be put after ‘make all’.

    - The `doc` directory with all the documentation files for this project, i.e. `Chess_SW_Spec.pdf`
    - An ASCII text file named `Chess.1` in directory `man/cat1/` as the static text file for the *Chess* game program's manual page
    - The `test` directory with instructions and input files for testing

2. A complete and clean tarball, namely `Chess.tar.gz` as the user deliverable with:

    - The `bin` directory where the binary executable file of the `Chess` game and the related resources, such as images for the pieces, are put
    - A *pdf* file named `Chess.pdf` in the `doc` directory as the user manual of the *Chess* game program. Screenshots of the program functions are desirable to have in this document

**The following test cases will be used for grading the *beta* version of the *Chess* program :**

1. make some legal moves

2. make some illegal moves, the program should recognize this, warn the user, and NOT make the move

3. make the following specific board setups and then make moves

    - **Position No. 6241** in "`LA_Times_Article_010613`"
    - **Position No. 6242** in "`LA_Times_Article_011313`"
    - **Position No. 6243** in "`LA_Times_Article_012013`"
    - **Position No. 6244** in "`LA_Times_Article_012713`"
    - **Position No. 6245** in "`LA_Times_Article_020313`"

    **Note:** each team **should** provide the instructions on how to setup a specific board, and the input files to load the specific setup in the `test` directory.

    This test should be supported in the `Makefile`, i.e. the 'make test' should run these tests automatically (in both `Makefile`s in `test/` and in the top level).

4. run a *computer vs. computer* game in less than 1 hour with moves properly displayed