Final Team Project Requirements

For this project you and your team will be creating a GUI simulation of the game *Durak*. Durak is a very popular card game in Russia and Eastern Europe[1]. The object of Durak is getting rid of all of your cards. The player that cannot is the *Durak* or *Fool*. Overviews of Durak and its major variants are easily found on-line[2], including at Wikipedia[3]. An excellent video tutorial is available on YouTube here[4].

Minimal Requirements (40 marks)

A project that implements ONLY the minimal requirements listed will be graded up to 67% (40/60).

	Unacceptable	Needs major improvement	Functional - significant issues	Functional - minor issues	Exemplary
Game-play logic	0	1-2	3-4	5-6	7-8
Computer A.I.	0	1-2	3-4	5-6	7-8
GUI	0	1-2	3-4	5-6	7-8
OOP Concepts	0	1-2	3-4	5-6	7-8
Internal Documentation	0	0	1	2-3	4
User Guide	0	0	1	2-3	4
					/40

- **Game-Play Logic (Two-Player).** Although typically Durak is played with between two and six people, for the purposes of this project you should limit the game to two-player, i.e. one human controlled player and one computer controlled player. You are not required to implement more than two players for this project.
- **Basic Computer Player A.I.** You are required to implement an A.I. logic structure that determines what action the computer-controlled player will make, taking into consideration whether the computer is attacking or defending, which suit is trump, and which cards (if any) in the computer's hand can be played.
- **GUI.** You are required to implement a graphical user interface (GUI) for your project. You are not required to design an especially artistic interface, but it should be simple, clean, and functional. You are required to make use of at least one custom control in your GUI. Note that if you are using graphical elements that you did not create from scratch yourself (e.g. playing card images), you **MUST** source these elements ethically and legally. This includes full and proper attribution to the artist in your project documentation.

[1] John McLeod, *Durak (Fool)*, http://www.pagat.com/beating/durak.html (Dec. 20, 2013).
[2] Google, *Durak card game,* https://www.google.ca/search?q=Durak+card+game (20 Dec 2013)
[3] Wikipedia, *Durak*, http://en.wikipedia.org/wiki/Durak (20 Dec 2013).
[4] homingpotato, *Durak Tutorial – Playthrough with Commentary*, http://www.youtube.com/watch?v=hQHW_CuGG2A (20 Dec 2013)

Final Team Project Requirements

- **Object-Oriented Concepts.** A major component of how you will be assessed is in the degree to which you are utilizing the object-oriented concepts and techniques presented in the course[5]. Your design should demonstrate *encapsulation*, *abstraction*, *polymorphism* and *inheritance* with a goal of ease of maintenance and reusability. Durak is a card game with elements that are common with many other card games (e.g. cards, decks, hands, etc). Any class you create that could be useful in other projects should be coded in one or more class libraries[6]. You are required to include at least one class library.
- **Internal Documentation.** At a minimum, your source code should be extensively documented with opening comments and in-line commenting. Like graphical elements you did not create from scratch, any code elements that you did not totally write yourself **MUST** be sourced ethically and legally. This includes full and proper attribution of the original code author in the comments.
- **User Guide.** You must include some form of user guide or tutorial the player can refer to that explains the functionality of the project. This may be text-based or video-based.

Recommended Features (20 marks)

A project that implements ALL of the minimal requirements and two or more of the recommended features listed will be graded up to 100% (60/60).

	Unacceptable / Not implemented	Needs major improvement	Functional - significant issues	Functional - minor issues	Exemplary
Transferring	0	1 - 3	4 - 6	7 - 8	9 - 10
Advanced A.I.	0	1 - 3	4 - 6	7 - 8	9 - 10
Number of Cards	0	1 - 3	4 - 6	7 - 8	9 - 10
Up to Six Players	0	1 - 3	4 - 6	7 - 8	9 - 10
Log and Statistics	0	1 - 3	4 - 6	7 - 8	9 - 10
Approved "Other" feature(s)	0	1 - 3	4 - 6	7 - 8	9 - 10
					/20

- **Transferring (a.k.a. *Perevodnoy* or *Passing* Durak).** In this variation of Durak, if on the initial attack the defending player is able to play a card of the same rank but a different suit (e.g. a six on a six), the attack transfers to that player in a two-player game. The new defending player has the option of transferring the attack again by playing a third card of the same rank. Another transfer is possible if a fourth card of the same suit is played. Transferring is not possible after a defense begins (i.e. the defending player plays a card of the same suit/higher rank or trump).[7]
- **Advanced Computer Player A.I.** Implement an A.I. logic structure that determines what action the computer-controlled player will make, taking into additional considerations such as whether it is

[5] Watson et al., Karli. "Chapter 8 - Introduction to Object-Oriented Programming". *Beginning Visual C# 2012 Programming*. Wrox Press. © 2013. Books24x7.
[6] Watson et al., Karli. "Chapter 9 - Defining Classes". *Beginning Visual C# 2012 Programming*. Wrox Press. © 2013.
[7] BoardGameGeek, *In Soviet Russia, cards play you!* http://boardgamegeek.com/thread/238664/in-soviet-russia-cards-play-you (20 Dec 2013)

Final Team Project Requirements

advantageous to pick-up even when a card could be played, how many cards are left in the draw pile, which cards have been played and discarded, and which cards the human player has picked up. If the *transferring option* is implemented, incorporate A.I. that considers transferring. The computer player should be difficult to beat.

- **Number of cards.** Durak is normally played with a deck of 36 cards. Give the player the option of playing with 20 (i.e. Ten to Ace in each suit), 36 (i.e. Six to Ace in each suit), or 52 (i.e. a standard deck) cards at the start of the game.
- **More than two players.** Give the user the option to play against up to six computer players. Note that this will significantly increase the complexity of the game-play.
- **Game-play Log and Statistics**
  - **Log.** As the game is being played, record the relevant game-play actions to a text-file log. This log should record the date and time the game was started, the initial hands dealt and the trump card at the game start. It should also record the actions of both the human player and the computer player for each round, including which cards they played, picked-up, and/or drew. Once the game is complete, it should record the results.
  - **Persistent Statistics.** Record and update the human-player's name and number of games played, wins, ties, and losses. This information should be stored in a text file and loaded every time the application is run. Provide the player the option to reset the name and statistics.
- **Other features.** If you have ideas for other features, they should be proposed in your milestone reports and consultations and approved by the instructor.

Evaluation Notes

Take note of the following points from the course outline regarding the final team project and in-progress team project activities:

- The final team project and in-progress team project activities will be completed by teams of students only. Students will be required to self-select their own team week 1. The instructor may assign students to teams in exceptional circumstances. Individual final projects will not be accepted.
- Equitable participation in all final project team activities is required. Any issues regarding student participation should be noted in the applicable progress report(s). Gross inequities identified will be dealt with on a case-by-case basis. This may include a non-participating student being removed from the team. In addition, all students will be given an opportunity to assess their own participation and that of their peers at the end of the course.
- The final team project and in-progress team project milestone reports are due by the due date assigned and posted on DC Connect and will not be accepted late.
- All in-progress team project consultations with the instructor will take place during designated class lab hours with feedback and the accompanying grade delivered as part of the consultation. Students must be present to partake in the discussion, and thus be eligible for these marks.