

CS111 Project Description

Fall 2013

With special thanks to Stephen Davies @ the University of Mary Washington

In this project you will be using the knowledge you gained from lectures and homeworks and applying them to a lengthier programming assignment. The answers to some of your questions do not strictly reside in this document. You are encouraged to look at other sources for inspiration. This semester's project will involve the popular card game UNO. This is an *individual* project, and as such you will be held to the University's academic integrity policies. You are welcome to discuss strategies and ideas with other students (and will be encouraged to in recitation), but you must code the project on your own.

The project will be divided into Weekly Milestones. You will have deliverables due every week as described by the milestone document. Below is a general overview of the project, as well as Milestone 1.

This project is what you make of it. There are points assigned to each milestone which are further broken down into tasks. However, you should not be limited by what we suggest the minimum work required here is. To get the most out of this project, I urge you all to push yourselves to find your limits. You might surprise yourself! Good luck!

Your assignment is to implement a sensible, strategic gaming algorithm that can compete in live game play against your fellow students and (ideally) slaughter them.

If you're one of the seventeen people in the U.S. who has never played Uno, you should read [the Wikipedia article](#) to get informed. Briefly, Uno is a popular card game in which each player holds a hand of cards, and tries to be the first one to "go out," or play all of their cards. Players are seated in a ring, and in the middle of this ring is an "up card," or a card placed on the table face up. Players take turns in sequence, clockwise around the ring. When it is your turn, you have the opportunity to play one of your cards on this up card (which will then become the new up card) and thus reduce the size of your hand. The card you play, however, must be playable on the up card, according to the following rules:

1. Most cards have a color -- red, green, blue, or yellow -- and you may play a card if it has the same color as the up card.
2. The colored cards each have a rank -- either a number from 0-9, or else a special "skip," "reverse," or "draw 2" rank. You may play a card if it has the same rank as the up card, even if it is of a different color.
3. There are two kinds of "wild" cards: ordinary wilds, and "draw 4" wilds. Either kind can be played on *any* up card. When you do so, you "call" a new color, specifying what color the next player must play.

Some special cards have an effect after being played, namely:

- If a "skip" card is played, the next player in sequence is skipped.

- If a "draw two" card is played, the next player in sequence must draw two cards from the deck, and is then skipped.
- If a "wild draw four" card is played, the next player in sequence must draw four cards from the deck, and is then skipped. (The player who played the "wild draw four" must then call a color as with a normal wild.)
- If a "reverse" card is played, the sequence of players is reversed to counterclockwise (or back to clockwise, after an even number of reverses.)

The object of the game is to run out of cards. When this happens, the player going out is awarded points based on **the cards remaining in the opponents' hands**. These points are calculated as follows:

1. For every numbered card held by an opponent, the winner of the round gets points equal to that number. (5 points for a 5, no points for a 0, etc.)
2. For every "special" colored card (draw two, reverse, and skip), the winner of the round gets 20 points.
3. For every "wild" card (either normal, or draw four), the winner of the round gets 50 points.

Normally, players continue playing hand after hand until one player reaches 500 points, and is declared the overall winner of the game.

Milestone 1: Due Week of Nov 4. (15pts total)

Task 1: Play UNO 0pts

This Milestone is the design phase of your project. However your first order of business is to actually play some UNO. You can pick up the physical game from many online and brick and mortar retailers and play with your fellow CS111 classmates, roommates, friends, family, etc. You'll want to get a sense of the rules and strategies that make a good UNO player. Alternatively, you may visit one of the many websites that offer UNO as an online game. Listed are a few to get you started: <http://www.memory-improvement-tips.com/uno-online-card-game.html> , <http://zone.msn.com/en/uno/> , <http://www.fupa.com/game/BoardGame-flash-games/uno-card-game.html>

Task 2: List information needed to play 5pts

After playing a good number of rounds of UNO, take a survey of the most basic information that you use to make your decisions on how you play one round of UNO. Make a listing of all of this data. Consider this as the "input" to your UNO playing algorithm. Put this into a file called <yournetid>-UNO-design.txt. Replace <yournetid> with your actual netid (not RUID!).

Task 3: Design a basic strategy for playing UNO 5pts

Using the input you derived from above, sketch out an algorithm to play just one hand of UNO. In other words, assume you are given the information you requested in Task 2, and design a list of steps that will allow you to choose a card you should play. In this phase of

the design, you may ignore any strategy, and merely select any legal card to play. Legal is defined by the rules of the game as listed above. Consider this the “output” of your UNO playing algorithm. You may choose to use a flowchart in this design, or write your design in paragraph form. Be very explicit about each step of your algorithm. Assume you are explaining your basic strategy to someone who has never seen or heard of UNO before. If you use a flowchart, name it <netid>-UNO-flowchart.pdf. If you use the paragraph form, add it to the file you created in Task 2.

Task 4: Optimizing Strategies for playing UNO 5pts

Now imagine a game where you are omniscient. You have full knowledge of all the information produced in the game. First, identify what these additional pieces of information are. What would you have liked to have known about the cards, the deck, or the other players when you were playing in Task 1? Write all of these down. Secondly, assume you have all of this new information that you don't have in a normal game of UNO. How does your strategy change now? Describe an algorithm that might make use of as much or as little of this newfound information to play a single hand of UNO. Explain why you think this information might help you. Again, include this in the file you produced in Task 2.