Daniel Goodnow
October 31, 2013
CS202

Project 8 Documentation

Summary: Project four is a simple card shuffling and dealing program. It reads in an Uno deck from a text file and places these cards into an array of card objects. The class contains the value, color, action, and location of the card. The list of cards can be printed to the screen. These cards can also be shuffled and stored in a separate array of card objects, which can then be printed to a user specified file. The program also loads in player profiles from a text file and stores these in an array of player objects. These classes store the name and ID number for each player. The program incorporates a basic menu system with options to print the unshuffled deck or shuffled deck, shuffle the deck, write the shuffled deck to a file, print the player information, and deal the cards. The latest functionality uses overloaded operators to simplify code and make it easier to adapt.

Important Design Concepts: Several key design factors were used in this program. The main function of the program uses a while loop with a switch statement to execute the functions. This runs until the user inputs a 'q', which ends the loop and closes the program. The use of classes allows each aspect of a player or card to be easily accessed and modified. It also makes the code simpler and easier to understand. The use of classes increases security of the program. All array manipulation and operations were done using only pointers. Pointers are now accessed using offset notation, which makes the code more readable. Also, all arrays are dynamically allocated. This helps manage memory of the program. The string functions were moved to a separate file in order to increase modularity. This means that the program now uses a make file to compile all necessary files. The latest addition is the use of overloaded operators. The '=', '<<', '>>', '<', and '>' operators were overloaded and functions with duplicate functionality have been removed.

Specifications and Limitations: The input file must be a simple text document. There must be six rows of cards. The first four consist of the four colors of cards and only contain the numbered cards. The fifth row contains the special action cards organized by color. The last row contains the eight wild cards. The value and color of the cards are separated by a space. These specifications are crucial to the proper operation of the program. The file will not be input correctly if these specifications are not used. The player input file consists of four rows, each with a player name and a 5-digit identification number separated by a space. Currently, the user interface is very basic and limited. The screen is not cleared between operations and the menu prints again once each operation is complete.

Future Improvements: With more time, the user interface could be largely improved. The screen can be cleared after each operation and cleaned up in general. At some point, the user interface could be changed into a graphical interface which could be much more user friendly. Another possible improvement would be to change the functionality of the deck loading function so that it can load different types of decks, as well as different formats of files. This would reduce the need for heavy specification of file format for the input file.