Homework 4

C++ Interfaces – Stack Game

For this assignment you will create a Stack Game. (Not much of a game, but it will familiarize you with some of the design for the class project.)

A deck of 10 cards is created, where a card has a rank (but not a suit). The Game class supports two commands:

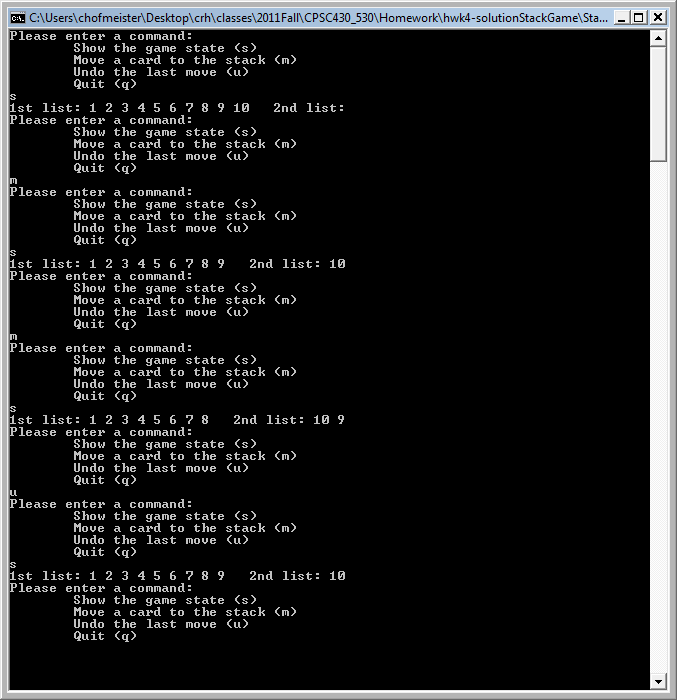
* move cards from the deck to a stack
* undo, which moves the top card from the stack back to the deck.

The GameView class displays the two stacks.

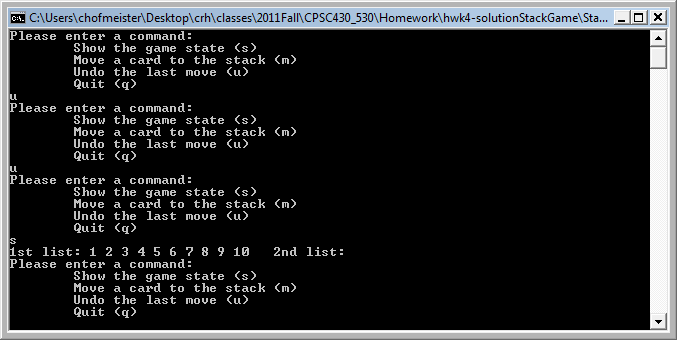
The Controller (main) creates the deck and stack, then passes them to both the Game and GameView. However, Game treats them as IStack objects, while GameView treats them as IReadCardList objects. The two interfaces limit what the Game and GameView can do with the deck and stack. IStack treats them both as stacks, and allows pushes and pops. IReadCardList supports sequential access to the cards, but does not allow any updates to the list.

Another important aspect of the design is the relationships between CardList, SStack, and Deck. CardList holds the actual cards and provides methods for reading and updating the CardList. SStack inherits from CardList. It provides the implementation of IStack, but simply uses CardList methods to accomplish the tasks. Deck inherits from SStack. It doesn’t implement any methods, but its constructor creates the Card objects and pushes them onto the stack (or appends them).

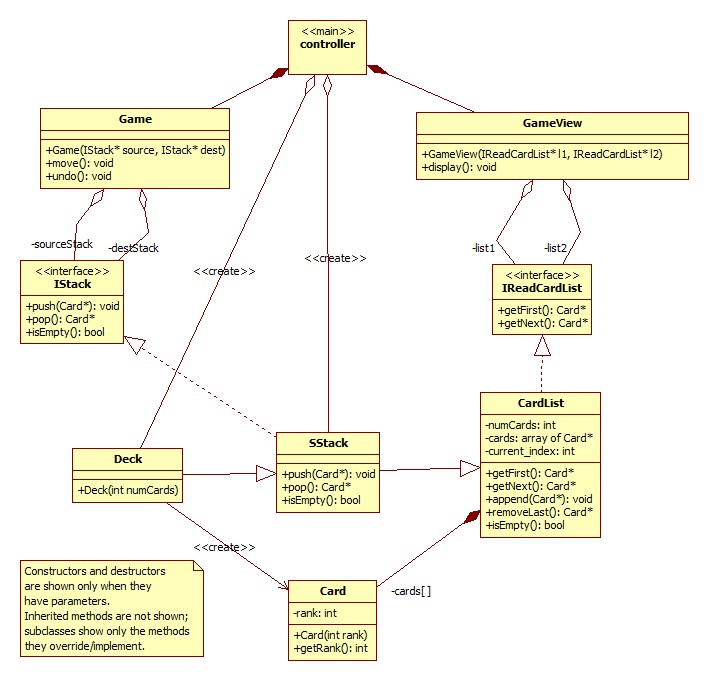
Here is an example of the output of the program:



Make sure your program handles boundary cases such as this:



The following class diagram specifies the classes you must provide.



### File Controller.cpp, method main

This method does the following (you can download it from D2L):

#include <string>

#include <iostream>

#include "SStack.h"

#include "Deck.h"

#include "Game.h"

#include "GameView.h"

using namespace std;

int main(int argc, char\* argv[]) {

SStack\* destStack = new SStack();

Deck\* sourceStack = new Deck(10); // put 10 cards in the deck

Game game(sourceStack, destStack);

GameView view(sourceStack, destStack);

bool done = false;

string command;

while (!done) {

cout << "Please enter a command:" << endl;

cout << "\tShow the game state (s)" << endl;

cout << "\tMove a card to the stack (m)" << endl;

cout << "\tUndo the last move (u)" << endl;

cout << "\tQuit (q)" << endl;

std::cin >> command;

if (command == "s") view.display();

else if (command == "m") game.move();

else if (command == "u") game.undo();

else if (command == "q") done = true;

else {

cout << "Invalid command: " << command << endl;

cout << "Commands are s, m, u, or q." << endl << endl;

}

}

return (0);

}

**General Information**

Make sure you follow all coding conventions given on webct in Course Documents. Also see Course Documents for the polymorphism example and the two interface examples (V1 and V2).

In Visual Studio .NET create a new **project named StackGame** and place all files for the program inside this project. When you are ready to submit, do the following:

* Make a copy of your entire solution folder.
* In the copy, delete everything except your source files, the .sln, .vcxproj, and .vcxproj.filters.
* Zip up the copied (and cleaned out) folder. USE WINZIP or 7-ZIP, NOT ANOTHER ARCHIVING PROGRAM.
* Check your zip file: copy it to a new spot on your machine, unzip, open the solution, build and run the program. Double-click on a few source files to make sure your file links work.
* Upload the zip file.