Undo/Redo System Implementation
Cpt S 322 Homework Assignment #8
by Evan Olds

Submission Instructions:

Submit source code (zipped) to Angel BEFORE the due date/time. If the Angel submission is not working, then submit to TA via email BEFORE the due date/time. "Angel wasn't working" is never an excuse.

Optional: Include a readme.txt file in the zip with any relevant information that you want the grader to be aware of.

Assignment Instructions:

Read each step's instructions *carefully* before you write any code.

In this assignment you will implement an undo and redo system for your spreadsheet application. You will also add the ability to choose the background color of a cell. Your undo system will support undo and redo actions for changing the cell's text or background color.

Part 1 – Add a background color property to the cells (4 points)

Make sure you implement this in the same way you've dealt with other cell properties. That is, you'll have a background color in the logic layer and changing that property will invoke a property-changed event that the UI will respond to.

In the logic engine:

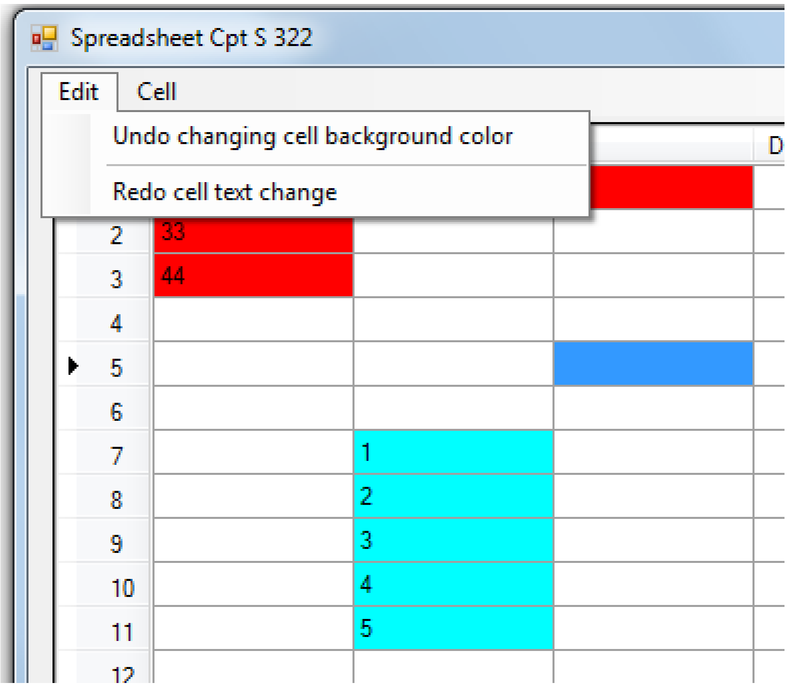
- Name the property something like "BackColor" or "BGColor"
- Make it a type that's not UI-technology dependent, such as **int** or **uint**
- Make sure that when it gets changed you invoke the property changed event

In the WinForms app:

- Extend your event-handling code to respond to changes to cell background colors
- Update cell backgrounds in the DataGridView accordingly
- Use the DataGridViewCell.Style.BackColor property
- That property is of type System.Drawing.Color which has a FromArgb method
- Add a button or menu option to change the background color of the selected cells
- Use a ColorDialog to prompt the user for a color
    - Remember that you should be setting the background color in the data/logic cell, then the UI should update in response to this change.

Part 2 – Implement the undo/redo system (6 points)

- Implement an undo/redo system as we discussed in class
- Support undoing cell text changes and cell background color changes
- Every undo that is executed should automatically push an item onto the redo stack
- Neither the undo or redo stacks should be publically exposed. That is, don't do:
  - public Stack<UndoRedoCmd> Undos
- Declare it privately then offer the ability to add and execute undos through public functions
- You can create a Workbook class like we discussed in class and store the undo/redo functionality there, or you can add it to the spreadsheet class.
- There must be menu options or buttons that allow the user to undo or redo at any time.
  - If the undo stack is empty then disable the undo menu item.
  - If the redo stack is empty then disable the redo menu item.
- Recall that each item on the undo or redo stack should be a collection of simple command objects with an accompanying title that tells the user what is going to be undone/redone. Display that information in the menu items as shown in the screenshot below.


Spreadsheet Cpt S 322
Edit
Cell
Undo changing cell background color
Redo cell text change
D
2
33
3
44
4
5
6
7
1
8
2
9
3
10
4
11
5
12