User Interface / User Experience(UI/UX)

Donghun Ha
Department of Computer Science & Engineering • Taylor University • 236 W. Reade Ave, Upland, IN 46989

Introduction

OpenSolver is an MS Excel extension created using a programming language called Visual Basic for Applications (VBA). In Excel VBA extension development, there are mainly two parts that developers are concerned with. The first component is Macro, which is the programming language. Another component is forms and dialog, which is a drag-and-drop GUI(Graphical User Interface) creating tool.

My role in the project was to create an appropriate interface and ensure good user experiences, which is a crucial part of any software development. I refered to an existing OpenSolver UI/UX as a sample and considered it as the ultimate project goal, and attempted implementing parts of its functionality throughout the project.

Demo Solver Interface

Although both Visual Basic and LibreOffice Basic provides forms and dialogs, the way their macros interact with the interface is quite different. Therefore, instead of performing direct porting of codebase, I created a demo interface and attempted to gradually implement more functionality with enough tests.

Demo solver interface has an intuitive look that any user who has experience with OpenSolver would find it easy to use. It has the ability to take cell references entered in the textfield and parse it into cell range references.

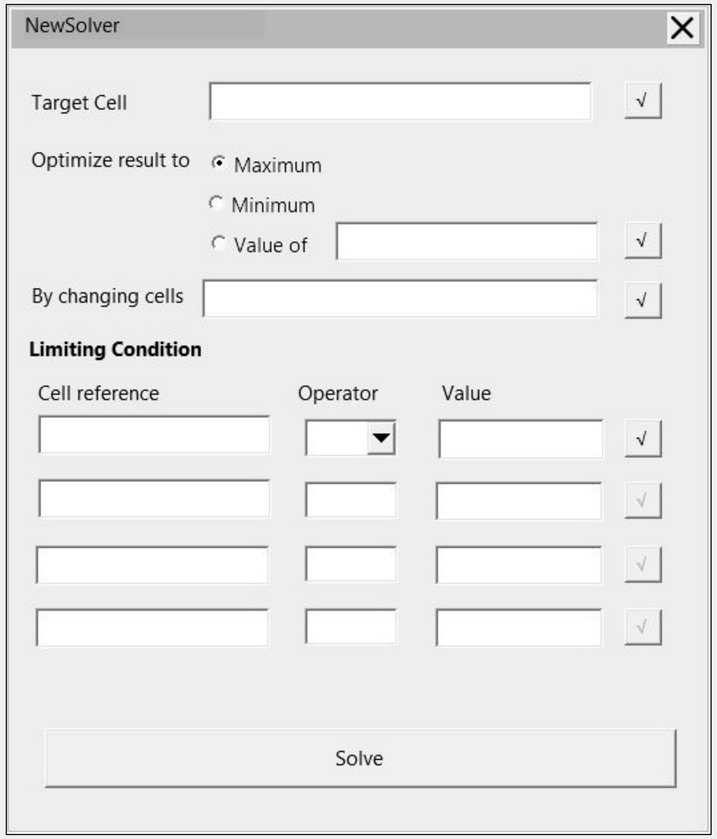

Figure 1. Demo solver Interface

Coloring Variable Cells

OpenSolver has the functionality of marking cells after setting a solver model. This was one of our client's major requests. This functionality was implemented by obtaining the location information of the cells (in coordinates format) and the size of them. Once the information was obtained, macros was able to create square-shaped objects and textboxes, as illustrated in Figure 2. Constraint cells had two different ranges, and LibreOffice Basic had "Glue" object that would link the two or more different shapes. With the combined use of these tools, we were able to mimic the OpenSolver user experience in terms of coloring variable cells.

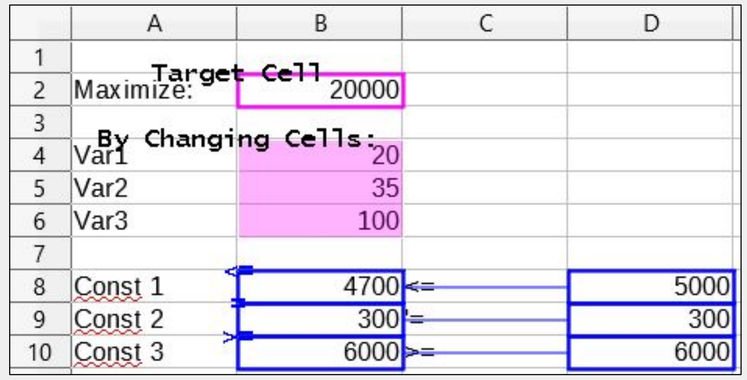

Figure 2. Colored variable cells

Conclusion

Although there is much room for improvement, the UI/UX of our demo has improved in many ways that meets some of our major requirements, such as taking inputs and coloring cells.

Future Work

In this project, the best user experience is defined as the one OpenSolver yields. In such manner, much of UI/UX could be improved by the future project team. For example, selecting cells with the mouse during dialog input would enhance the user experience substantially.
Improvement could also be made in linking ported macros. Our solver program would be much more powerful if all functionalities we implemented on macro could be linked to our UI.