

Workshop activities – Naresuan University – 8th January 2010

Dependency and its role in modern programming languages

Try the following three activities, either in groups or individually. The first activity introduces the principle of dependency in spreadsheets. The second activity demonstrates dependency properties in Windows Presentation Foundation (WPF) and how they can be used to build an application. The third activity explores the use of dependency in tools developed by the Empirical Modelling Research Group at Warwick.

Activity 1: Simple dependency in a spreadsheet

This activity is an introduction to using dependency. To demonstrate the power of dependency, this activity shows how a standard spreadsheet application that utilises dependency (e.g. Excel) can be used to build a simple game. Requires Microsoft Excel (recommended 2003 or later).

Quick introduction to draughts (checkers)

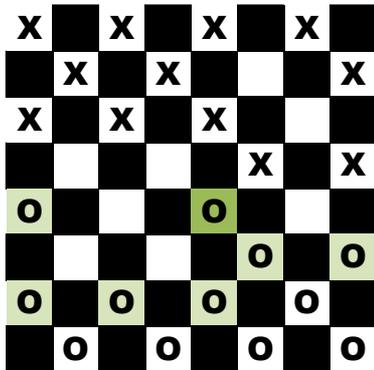
- Two player game on an 8x8 chequered board, using only one of the colours (e.g. only the white squares).
- Each player starts with 12 pieces arranged on the first three rows of their side of the board.
- Players take turns to make a move.
- Pieces can move diagonally left or diagonally right in the forward direction only.
- If there is an opponent piece diagonally left or right with a space the other side then you can 'jump' over the opponent's piece and remove it from the board.
- The game is won when a player removes all the opponent's pieces from the board.
- If a piece reaches the other edge of the board then it is 'crowned' and can then move in any diagonal direction.

Instructions

Open the Excel file 'Game.xls' and go to the first sheet called 'Play'. You will see a standard checkers board.

- a) You can rearrange pieces on the board by typing 'O' or 'X'. In order to move a piece you will have to first delete the piece and then type it in the new position. Try to play with you partner.
- b) If you were building a checkers game then you might want to have some indication of when a player has won. Create dependencies in cells M2, M3 & M4 that show the number of pieces left on the board and the winning player.
- c) Imagine you are playing as the 'O' pieces (starting at the bottom of the board) and you would like to know which pieces you can currently move. Create a dependency that determines whether a cell contains a piece that can be moved. Use the 'Possible moves for O' sheet, and place a formula in each of the cells (most of the formulas will be the same).
- d) You are still playing as the 'O' pieces and now you would like to know which pieces you can use to jump/take an opponent's piece. Use the 'Possible jumps for O' sheet, and place a formula in each of the cells.

- e) Adding colour to help you play! Excel allows you to add conditional formatting to cells, which means that you can change the style (e.g. background colour of a cell) depending on a formula (as discussed in the lecture). Try to add colours to your cells so that you can see which cells you can move, and which cells you can use to jump/take an opponent. Example:



- f) Can you think of (and implement!) other ways of using dependency to support playing checkers?

Hints

- You might want to use the following Excel functions: AND, OR, IF, COUNTIF.
- To reference cells on another sheet: Play!A1.
- If you need to use the same formula many times then you can copy/paste. If you only want to paste the formulas then you can do this using the 'Paste Special' function.
- Conditional formatting has changed between versions of Excel so you might want to look up how to do it in the version you are using.

Activity 2: Writing applications using dependency properties in WPF

In this activity you will be able to try out some of the WPF examples and begin to see how to build applications using dependency in .NET. For the activity you will need Visual Studio 2008 and .NET Framework 3.5.

Instructions

- Open Visual Studio and create a new project. Select the 'WPF Application' template under project type 'Visual C#', 'Windows'.
- The template contains a file Window1.xaml where you can begin to create an application by writing XAML code. Try the first example from the slides by pasting the following code:

```
<StackPanel>
  <Slider Name="FontSizeSlider" Minimum="10" Maximum="50" Value="20"
    Margin="3" />
  <TextBlock Name="MyTextBlock" Text="Sawasdee Naresuan!"
    FontSize="{Binding ElementName=FontSizeSlider, Path=Value}"
    Margin="3" />
</StackPanel>
```

- There are four examples of dependency in the slides (one-way, two-way, triggers and animation). Try each of these for yourself.
- In Activity 1 you began to create a helper for playing draughts using only dependency. Download the Draughts solution and its associated files to find WPF version of the draughts

board. The only dependency it contains initially is a trigger to change the background colour of the cells depending on the value of the TextBlock.

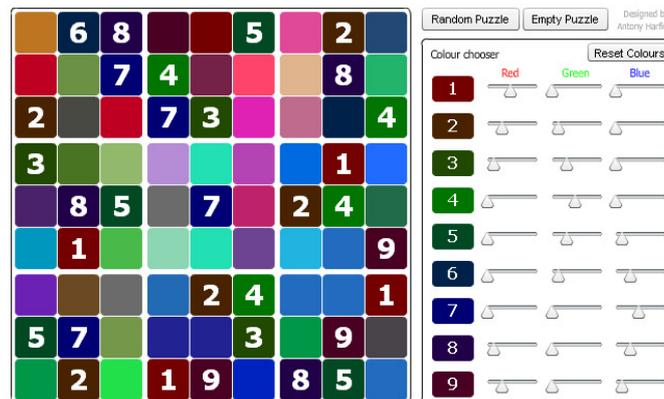
- i. How might dependency be used in WPF to provide similar help to that in Activity 1?
 - ii. What are the advantages of dependencies in spreadsheets over WPF dependencies?
 - iii. What are the advantages of dependencies and triggers in WPF over spreadsheets?
- e) If you have time, add bindings in the XAML code that create dependencies in the Draughts project to support a player in choosing a move.

Activity 3: Using dependency in EDEN to build colour Sudoku

In this activity you will use the Empirical Modelling tool called EDEN to build a colour version of a Sudoku game. The activity requires a web browser with Flash 9 or later.

Background

In 2008, whilst experimenting with dependencies in a model of Sudoku in EDEN, we found that colour recognition could be used to help solve Sudoku puzzles. The result was a colour version of Sudoku: www.warwick.ac.uk/go/sudoku



Dependency was the key to changing a normal Sudoku game into one which used colour as a support for solving. The following activities are designed to enable you to revisit the steps involved in building Colour Sudoku.

Instructions

- a) First run through the introduction to the Web Eden environment and find out how the Sudoku script was developed: <http://weden.dcs.warwick.ac.uk/webeden/runproject/intro/>
- b) To learn how colour was added to the Sudoku model: <http://weden.dcs.warwick.ac.uk/webeden/runproject/workshop2C/>

The two links above are taken from existing online workshops and use Web Eden. For more information, see the Web Eden page:

<http://www2.warwick.ac.uk/fac/sci/dcs/research/em/software/webeden/>