

**Object Oriented Programming Using C#**

**Resit Assignment 2012**

**Dr M. Spann**

1. **Aims and Objectives**

### The aim of this assignment is to produce an *ImageToy* game. This is an electronic version of an old children’s game which re-creates a jumbled up picture by moving one piece at a time. Figure 1 shows an original image and it’s jumbled up version with one piece of the original image missing thus creating a gap in which to shuffle individual pieces. By convention the piece at the bottom right of the original image has been removed and then the image shuffled. The image can be re-constructed by moving the mouse pointer into a piece of the shuffled image adjacent to the gap and then clicking the mouse button. This then swaps the image piece and the gap.

1. **Preparatory Work**

Thus assignment will make use of many of the programming techniques we discussed in the lectures such as the use of the *OpenFileDialog* box to read in an image and display it in a simple form, the use of the Image class to handle images and the use of event handlers to handle mouse button presses. You should read through your lecture notes once again to make sure you understand the material. Also I would suggest you make extensive use of visual programming to design your user interface.

1. **Lab Work**

Your program should ask the user for the number of pieces in which to shuffle the image. It should then randomly shuffle all of the pieces of the image. The number of pieces in the shuffled image, and hence the difficulty of reconstructing the image, should be left as a user choice. (Think carefully about how to shuffle the image. Your program must be able to ‘un-shuffle’ the shuffled image back to its original form by moving image blocks into the adjacent gap. Just randomly moving image blocks around won’t necessarily give you this!)

Finally it should then allow the user to play the game by clicking on pieces to move them. Clearly, only if the user clicks on a piece adjacent to the gap should the piece be moved.

Think carefully how you want to represent the shuffled image. You should certainly avoid having to store the original image and its shuffled version as separate images as this is very inefficient! To shuffle an image, you could use a Bitmap class and access individual image pixels. However, this would be extremely inefficient. Try and think of a better way of doing it.

**4. Assessment**

The assessment will be based on a submitted design document (**maximum length 3 pages**) from which you will receive feedback as well as my assessment of your program’s functionality. These will be in proportion report 25%, program functionality 75%.

Please submit your program **written under Visual Studio 2010** on cd. Please include all of the solution files under a single solution directory. Make sure your cd has your name/ID on it in case it.

The deadline for submission of your design document will be **2nd July at 12pm.** I will provide summative feedback within 3 working days to help with your program implementation.

The deadline for submission of your cd is **16th July at 12pm.**

Please submit both the report and the cd to the postgraduate office.

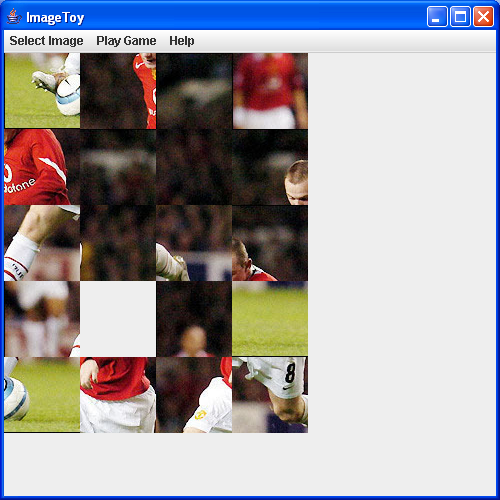
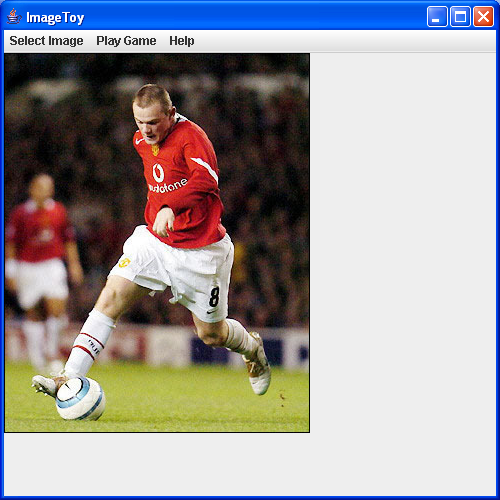


Figure 1

Object-Oriented Programming and Design

Programming Assessment

Dr M Spann

**Student's Name: …………………….…..**

**Grade (%): ……………………………**

**Assessor: ………………………………..**

|  |  |  |
| --- | --- | --- |
| **Design Report ---- /25**  Structure, style and clarity  Class, object diagrams  Data structures  Discussion of design issues inc. object oriented design |  |  |
| **Program Functionality------/75**  Graphical user interface  Correct operation (specification fulfilled, extended features) |  |  |
| **Total** |  | **/100** | |

Any evidence of plagiarism Yes No