**Minefield – Task 2**

The next major task I decided to tackle was to get a “sprite” or “character” to appear on the grid. Its position will be defined in terms of a Row and Column number, but to actually display it, I need to identify the specific label that corresponds to this location.

After this task has been solved I can progress on to getting the sprite to move around the grid. This will be achieved in a similar way to how you can create an animation by flicking through a set of cards having successive pictures drawn upon them. To make the sprite ‘move’ we must make it disappear from one label and then reappear on another adjacent label.

Furthermore, as it moves it should leave a ‘trail’ behind to show where it has already been…

Give it some thought: how do you code a solution to achieve what is seen in the executable?

**Firstly, what are the main tasks involved; what key problems or actions need to be solved?**

* We need to be able to see where the sprite is at any point in time. We could change the colour of a label (say to White) to indicate this, but using an image would be a lot nicer
* We will need to ‘keep track’ of where the sprite is at all times. The most obvious way is to use two global variables to record in which column and row on the grid it is currently located. We need to decide if the coordinates will be row,column from the top-left, or x,y from the bottome-right, and whether they start at (0,0) or (1,1). All of these are equally possible – you just need to pick one and stick to it – I chose to use (Row,Column) coordinates with the values (0,0) identifying the top-left grid cell.
* A tricky issue is that we need to find a way to identify, and get hold of, the specific label that corresponds to any particular (row, column) location on the grid.
* We need to find a mechanism to show the history trail. This could display an image again, but this time the use of a plain backcolor is simpler and will be quite adequate.
* We need some way for the user to instruct how to ‘move’ the sprite. We could use buttons objects on the Form to control its movement. You could also consider keyboard press events: so, if the user presses the up-arrow key on the keyboard, for instance, the sprite moves up one row (this is a potential future development).
* What is the algorithm (or what key steps involved) in making a sprite to appear to ‘move’? Essentially, it must be hidden/removed from the label where it is currently positioned, then shown on an adjacent label according to the move direction
* We also need to ensure you cannot move the sprite off the edge of the grid

**My Solutions:**

I will need to keep a track at all times of the “current position” of the sprite. I set up two variables (atRow and atCol) to record this information. However, as you know, the grid cells are actually label objects referenced by a name (label1, label2, …label400). So I need a way to convert from any given (atRow, atCol) coordinate to the associated label object.

I will use the backcolor property of a label to show those already visited

I will use the image property of a label to show the sprite at its current location (this could use backcolor again, but an image is more interesting and looks nicer)

Therefore, I need to source an image. It must be small, as the labels are only 20x20 pixels. I grabbed a simple icon via a quick web search (look for a nicer one for your version).

I make use of “Resources” to store the image internally in the program (to be demo’d)

I want the sprite to be displayed in the centre of the bottom row of the labels when we start

For now (again this is something that can be improve upon later if you wish) I will control the sprite’s movements using four buttons controls. I have left these plain – you can find some pretty images to use on them to make the game look more attractive.

**My Solutions:**

1. Use two global variables to store the sprite’s location

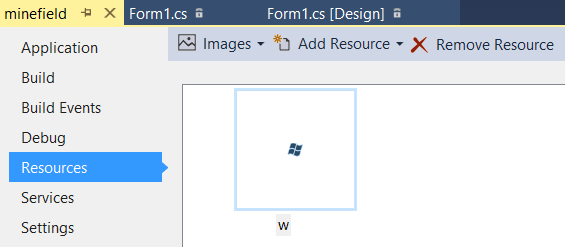
//start location of sprite

int atCol = 10;

int atRow = 19;

2. Then set up the sprite image in Resources (an image is supplied on Bb; it is deliberately not particularly attractive, and is something for you to change in your final submission)

To use Resources: Right-click on the project, go to properties, then the resources tab; and then in add resource - add existing file; and finally select the file “w.PNG”



The next question is, how do you access a Resource from within your code? Do some research, for example, a google search on “c# use image from resources” found me…  
<http://stackoverflow.com/questions/1192054/load-image-from-resources-area-of-project-in-c-sharp>

This tells us you can access the image using code that looks something like:  
 **Properties.Resources.*image\_name*;**

2. Next task: I need to set the image of the label positioned at location (atRow,atCol) to show this sprite. To do so we need a wat to establish which label corresponds to any particular (atRow,atCol) position.

This is NOT easy to program because it must return a Label object. Something similar arose in the Calculator tutorials, where an object variable referred to a specific numeric button. I give you the code below to return a specific label object given a column and row coordinate. Feel free to try and code this yourself if you feel up to a challenge, but I am not expecting you to be able to do this (it is relatively tricky) so just expand and copy-and-paste my code if you prefer…

//function to return Label at (atCol, atRowl)

private Label getLabel(int atCol, int atRow)

{

int k = atRow \* 20 + atCol + 1;

string s = "label" + k.ToString();

foreach (Control c in panel1.Controls)

{

if (c.GetType() == typeof(System.Windows.Forms.Label))

{

if (c.Name == s)

{

return (Label)c;

}

}

}

return null;

}

We can call up this code in a subroutine that will now show the sprite at any stated grid location

//function to show sprite at (atCol, atRow)

private void showSpriteAt(int atCol, int atRow)

{

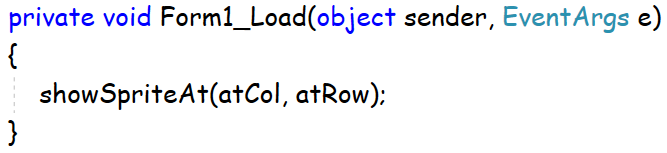
Label lbl = getLabel(atCol, atRow); //get label at (atCol,atRow)

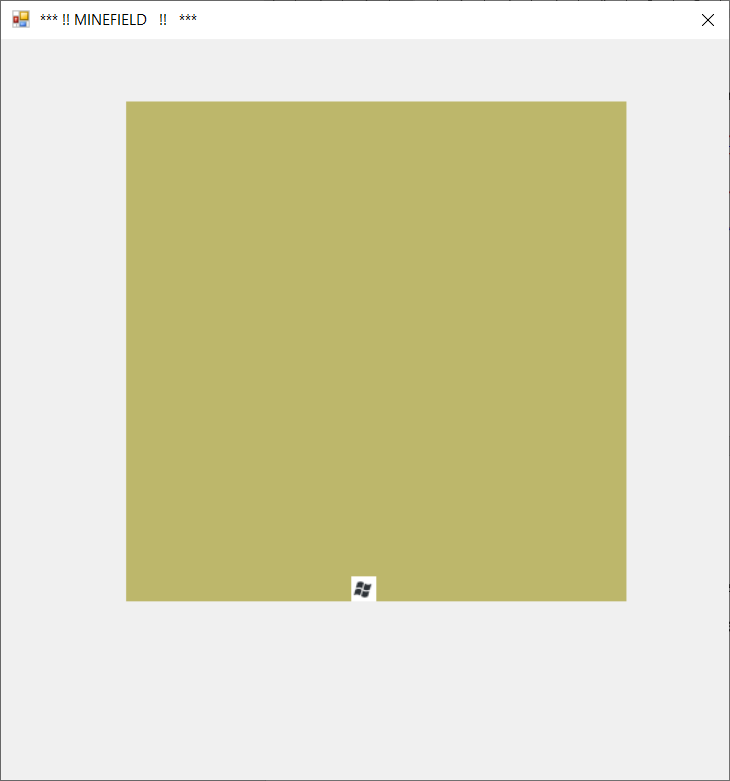
lbl.BackColor = Color.White; //set its backcolour to white

lbl.Image = Properties.Resources.w; //set it to show an image

}

When the form is first loaded, we wish to display the sprite at its starting position…



Give at a go… Voila! 

<commit2>