**Code Assessment**

Forewords: As a beginner programmer, I’m unfamiliar with the structure and variables for the current game within this project. However as time progresses I should have built up enough knowledge to lay down the foundations on the first prototype.

As a visual reference, I had composed an image which displays the basic game-space and core elements required for minimal functionality. The portrait orientation indicates that it will be predominantly played on mobile/tablet devices and the grid system shows that the user will be performing actions along the idea of ‘tiles’. The grid can be designed in either an image-editing software and imported as an asset or created within a panel and called by the ArrayList function to construct a 5x6 table with shorter but longer rectangles that extend as far as the table’s width above and below the main table.

To demonstrate just how the main table will be constructed, I would enter the following lines:

*ArrayList ilTemp = new ArrayList ();*

*int h,v;*

*h=v=0;*

*for(int k = 0; k <30; k++)*

Inside the function, to order the dimensions properly, I would implement the following (the horizontal and vertical positioning here are just exemplary integers):

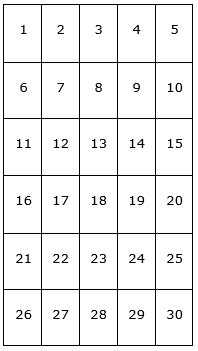
*ilTemp.Add(b);*

*h+=200;*

*if (h == 1000)*

*{h=0; v+=356;}*

This is a visual representation of the above:



The numbers in each cell wouldn’t appear in the proper table but rather are used to show the dimensions and controllable-character’s behaviour.

To display the menu bar (top rectangle) and the activity bar (bottom rectangle), specify the width of each as wide as the main table and as equal in height, which extends as far as the screen padding allows them - create a reference from the main table and apply +/- vertical translations to each respective rectangle.

This next assessment involves character movement, based on user controls in relation to a timer and turn-based system. In cells 27, 28 and 29, the three controllable-characters will be centred at the beginning of the level. Per turn, an outline glow will highlight the borders of the cell which indicates to the user that they can control that particular character. The outline glow selects one of the three cells that contain a controllable character in a pattern cycle until at least one collides with an obstacle which will then alternate selection between the remaining non-collided characters.

In addition, the current character will show a faint arrow for the user to gauge the cells travelled in one turn.

A timer will tick from a selected time until the user completes a move or the timer expires which will cause the controlled character at that time to move automatically which might not be the optimal choice given that the user can tell the character to stay in its position to avoid a collision.

The code below shows a timer of 15 second that counts down until 0:

*private System.Windows.Forms.Timer timer1;*

*private int counter = 15;*

*private void btnStart\_Click\_1(object sender, EventArgs e)*

*{*

*timer1 = new System.Windows.Forms.Timer();*

*timer1.Tick += new EventHandler(timer1\_Tick);*

*timer1.Interval = 1000; // 1 second*

*timer1.Start();*

*lblCountDown.Text = counter.ToString();*

*}*

*private void timer1\_Tick(object sender, EventArgs e)*

*{*

*counter--;*

*if (counter == 0)*

*timer1.Stop();*

*lblCountDown.Text = counter.ToString();*

*}*

A turn-based system is implemented to maximise the amount of time spent on a level before it requires a level-restart – this is used to add some complexity to the game.

The user can move a selected character by choosing the same cell, causing the character to remain in its current position and expend one turn or a move the amount of cells allowed, assigned to the character, vertically above until it reaches the final row.

When any character reaches the top-most row, the game must recognise the user has finished the level and display a congratulatory message. All of the above provide the user minimal functionality to be able to play through a prototype level of this game.

**References**

Agarwal, T. (2004). *Sliding Puzzle Game.* Available: http://www.codeproject.com/Articles/6557/Sliding-Puzzle-Game. Last accessed 25th Oct 2016.

Ormondroyd, S. (2016), *Level 1 Prototype [ONLINE]*. Available at: http://i66.tinypic.com/ncnyo8.png. Last accessed 25 Oct 2016.

Stecya. (2011). *Seconds Countdown Timer.* Available: http://stackoverflow.com/questions/6191576/seconds-countdown-timer. Last accessed 25th Oct 2016.