This tutorial explains the creation of a game of Tetris in Excel using spreadsheet formulas and minimum VBA. It focuses on the principle of game creation rather than on syntax. The procedure used to create this game is original, derived by trial and observation and I don’t claim to have the best method.

You will later be able to use the same general principles to implement this or other games using the programming language of your choice.

This first section deals with the creation of the four different pieces or “tetrominoes” the way they are also called.

The parts:

Most people are familiar with Tetris, which was created by a bored Soviet academician in the early 80’s, so no formal introduction is presented here. You can simply Google the word “Tetris” and a bounty of results will cover the principles and the history of the game.

There are a total of seven parts involved, and they are denoted with the letters: “I”, “J”, “L”, “O”, “S”, “T” and “Z”. A photograph of the parts is shown below.
The parts (tetrominoes) – worksheet implementation:

- Open a new workbook, save it as “Tetris_Tutorial” and name the first worksheet “Tutorial_1”.
- In the first worksheet, in the range Z2:AG29 we will plug in the data for all the seven shapes involved in the game (I, J, L, O, S, T, Z), corresponding to four angles of rotation (0, 90, 180, 270). Each shape has four components and each component has a pair of coordinates (x,y).
- Based on the above statement the shape data area will cover 7 x 4 x 4 x 2 = 224 cells.
- The game has one active shape whose coordinates we choose to store in range W8:X11. The shape is determined randomly by the model out of the seven fixed options (“I”, “J”, “L”, “O”, “S”, “T” and “Z”). The shape starts from the top of a well then it “trickles” down with a fixed but adjustable speed, being able to be rotated and moved sideways by the player.
- When the shape reaches the pile of old “exhausted” shapes on the bottom of the well, it will freeze there and a new random shape will automatically be issued by the model.
- Let’s introduce the shape parameters in the range X2:X5. Name cell X2 “Shape”, cell X3 “Rotation”, cell X4 “x_offset” and cell X5 “y_offset”.
- Let’s also insert the active shape defining formulas in the range W8:X11:
  
  => W8: “=OFFSET(Z2,4*Shape,2*Rotation)+x_offset”
  => X8: “=OFFSET(AA2,4*Shape,2*Rotation)+y_offset”
- After inserting the two formulas above copy range W8:X8 down to W11:X11.
- The last thing we need to do is filling out the shape table (range Z2:AG29) with constant data by trial and error – see next page.
Filling out the shape data range:

- First let's create a 2D scatter chart based on the data from the “Active Shape” range (W8:X11).
- The chart will have both the x and the y axes scaled from -3 to +3.
- Since we will fill the data by trial and error we would like to have a button which changes the “Shape” type and the “Rotation” parameters with ease within the range 0, 1, 2 and 3.
- Since this is an advanced tutorial, the procedure for inserting the 2D scatter chart, the buttons and the associated macros will not be explained here.
- Fill the data area for the first shape. You do this by trial and error trying as much as possible to keep the shape centered in the origin regardless of the rotation.
- All the shapes must be aligned to a grid, in our case this means that the coordinate of all of the four components belonging to a tetromino must have integer coordinates. Whenever there is an even dimensionality (in this case there is a line 4-shapes long) there is no choice of centering the shape in the origin. To preserve consistency in this case, we could choose to have the shape slightly biased towards positive coordinates regardless of the rotation setting, but that's not a necessity.
- By trial and error I came up with the values to the right corresponding to the first shape (“I” or shape #0) for all four rotation settings.
- The shape in all the possible rotations settings is shown on the following slide =>

<table>
<thead>
<tr>
<th>W</th>
<th>Parameters</th>
<th>0_degrees</th>
<th>90_degrees</th>
<th>180_degrees</th>
<th>270_degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Shape</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Rotation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>x_offset</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>y_offset</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Active Shape</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tetrominoes “I” and “J”:

- The complete charted data for tetrominoes “I” and “J” is shown on this page.
- The color of the chart markers was adjusted in conformity to the color of each shape in the original game to make the chart look prettier.
Tetrominoes “L”, “O”, “S” and “T”:

- The charted data for tetrominoes “L”, “O”, “S” and “T” is shown on this slide.
- The color of the markers was adjusted in conformity to the color of each shape in the original game to make the chart look prettier. The size of the markers is 25.
- A complete tetromino data table is shown in the next page.
Tetromino “Z” and the full data table:

- The charted data for tetromino “Z” is shown to the right.
- The color of the markers was adjusted in conformity to the color of each shape in the original game to make the chart look prettier.
- A complete tetromino data table is shown here.