Cipher Crossword User Manual

This program gives the user a blank crossword to fill in. This is not the usual kind of crossword with clues, but is a Cipher Crossword where each letter in the grid is assigned a number, and the idea is to determine which number represents each letter. I give here the Wikipedia description of Cipher Crosswords.

> *A cipher crossword replaces the clues for each entry with clues for each white cell of the grid – an integer from 1 to 26 inclusive is printed in each square. The objective, as any other crossword, is to determine the proper letter for each cell; in a cipher crossword, the 26 numbers serve as a cipher for those letters: cells that share matching numbers are filled with matching letters, and no two numbers stand for the same letter. All resultant entries must be valid words. Usually, at least one number's letter is given at the outset. English-language cipher crosswords are nearly always pangrammatic (all letters of the alphabet appear in the solution). As these puzzles are closer to codes than quizzes, they require a different skill-set; many basic cryptographic techniques, such as determining likely vowels, are key to solving these.*

Obviously, this program will only be of interest to those who love word puzzles. The program gives you 10 puzzles to solve. All the crosswords in this program are pangrammatic (as they should be). The puzzles are taken from an august British weekly publication, and spellings are therefore British English, but generally there is no problem for international users.

**Installing the Program**

Use the URL to download the zip file and extract the folder “Cipher Crossword” and place that folder in your QB64 folder. Be careful that the extraction method does not create an extra folder layer. Open the file “cipherxword_v1.bas” in the IDE, and make sure that you have the Run Option “Save EXE in the source folder” checked. The program will work with GL 1.2 or higher.

https://www.dropbox.com/s/uw983ed2xacd1ge/Cipher Crossword.zip?dl=0

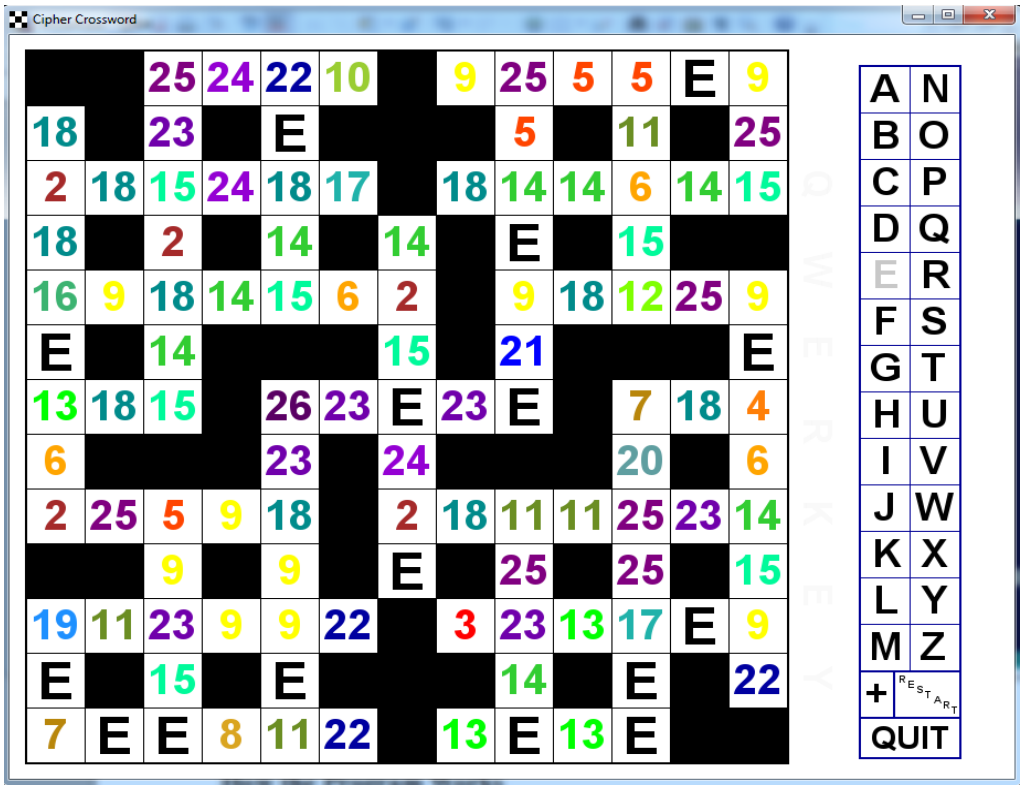
The program assumes a right-handed (default) mouse. If you use a left-handed mouse, then all the program lines containing “_MOUSEBUTTON(1)” will need to be altered – probably the “1” will need changing to “2”.

**Playing the Game**

When the program starts, a completed cipher crossword of the type used is displayed. This special crossword grid uses some abnormal words. Look out for QB64 community members Steve McNeill, Fellippe Heitor, Galleon, Clippy, Johny B, Tempo di Basic and myself (this is a Vanity Project!). Central to this grid is our beloved QB64 in word-form. All other words are good English, and all letters have been used. Press any key to move onto the program proper.

| G | | T | | M | | | | F | | | | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | R | E | N | A | | Q | U | E | R | C | U | S |
| L | | M | | Z | | B | | L | | L | | K |
| L | I | P | | E | A | S | E | L | | I | C | E |
| E | | O | | | | I | | I | | P | | W |
| O | | | A | P | E | X | | P | I | P | E | |
| N | | J | | A | | T | | P | | Y | | Q |
| | D | O | O | R | | Y | L | E | M | | | W |
| E | | H | | A | | F | | | | C | | E |
| V | A | N | | B | O | O | S | T | | O | A | R |
| E | | Y | | O | | U | | E | | D | | K |
| N | E | B | U | L | A | R | | S | T | E | V | E |
| T | | | | A | | | | T | | D | | Y |

The program loads an unsolved crossword, and displays numbers in the available grid squares.



A single click on a letter in the right-hand table will pick up that letter (Drag-and-Drop is not implemented in this program). That letter will then move with the cursor. Moving the letter onto an available square in the crossword grid and single clicking will place that letter in all the squares of that number. An animation will move letters from the right-hand table to the crossword grid. When a particular letter of the alphabet has been placed, it becomes unavailable in the table.

The idea of the puzzle, as in standard Cipher Crosswords, is to use your judgement to determine which numbers represent which letters and to fill the grid. The program starts with one seed letter placed in the grid. This is to help you work out which letters go where – as the grid is filled it becomes easier to determine where the next letters can be placed.

In normal Cipher Crosswords, the compiler has taken the time to given one or more seed letters which aid the solution of the grid. Unfortunately, the program has no intelligence to determine such an approach! The given seed letter is generated randomly. Therefore, you may need to add additional seed letters. Clicking on the "+" in the right-hand table will add an additional seed letter. This will remove any letters which you have already placed.

If you wish to remove any letters which you have placed in the grid, click on any instance of that letter. An animation will replace that letter into the table. At any time, you may restart the puzzle by clicking on the RESTART in the table.

You may quit at any time without solving the puzzle (either Quit from the table or pressing Esc). If you solve the puzzle, the next time you run the program a different puzzle will be given.

**Further (Unimportant) Information**

This program effectively simulates a page from a puzzle magazine or a newspaper, and clearly everything is 2D. In implementing the animation of moving the letters to and from the grid, however, I have used 3D _MAPTRIANGLE (whereas 2D _PUTIMAGE would have been obvious). The mathematics is that of a projectile under (constant) gravity. Effectively, the letters are fired from a cannon out from the screen and then fall back into the screen under gravity and the motion is a parabola in 3D space.

For the 10 puzzles to be solved, I have used ones from a magazine which I have completed myself. If this results in me being taken through the Copyright Courts, I shall look forward QB64 Community crowd-funding for my defence costs.

Originally, I had tried to implement a crossword-compiler version where the computer generated a valid grid from an English dictionary. I could not come up with a clever enough algorithm where the program could find words to fill the grid successfully. Maybe some time later.

Richard Notley