

## Collection b\_fun\_1 – Help

### **credits**

Scripts in this collection were made by Bernhard Schaffer.

Email: [how.to.script@tavernmaker.de](mailto:how.to.script@tavernmaker.de)

Web : <http://DMScript.TavernMaker.de>

Version: 2014-01-19

### **purpose**

Scripts in this package serve no deeper purpose than showcasing the capabilities of DigitalMicrograph scripting. They are either little games or recreational simulations.

### **Contents**

Collection b_fun_1 – Help.....	1
credits.....	1
purpose.....	1
Contents .....	1
Setup .....	2
Contained scripts.....	3
Tile Puzzler .....	3
Game of Life .....	4

## Setup

Copy the *b\_fun\_1.gtk* file into the plugins folder and restart DigitalMicrograph. There should now be a new menu entry: **B\_**

It is possible, that other packages (of the same author) have already created a menu entry **B\_**. In this case, installed commands are added to the menu.

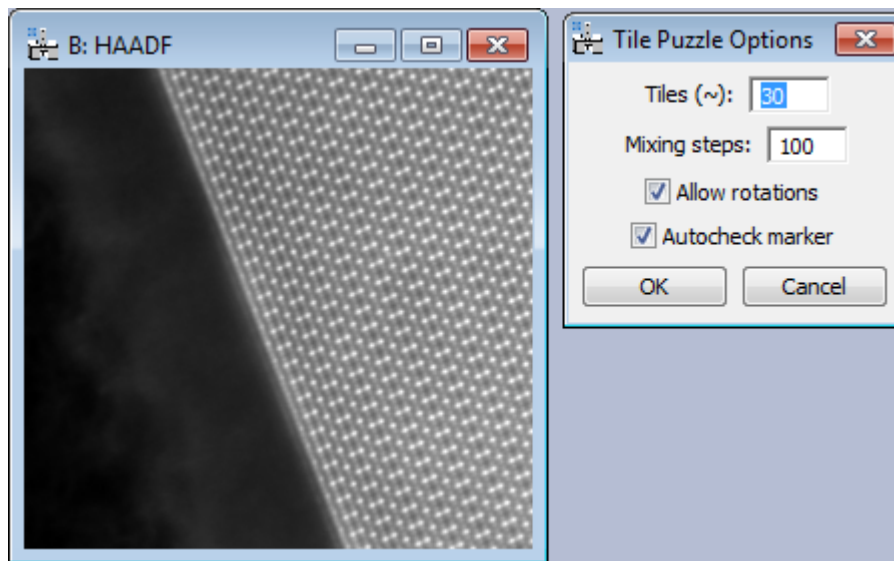
All scripts of this package install in the sub-menu of name **Fun**.

## Contained scripts

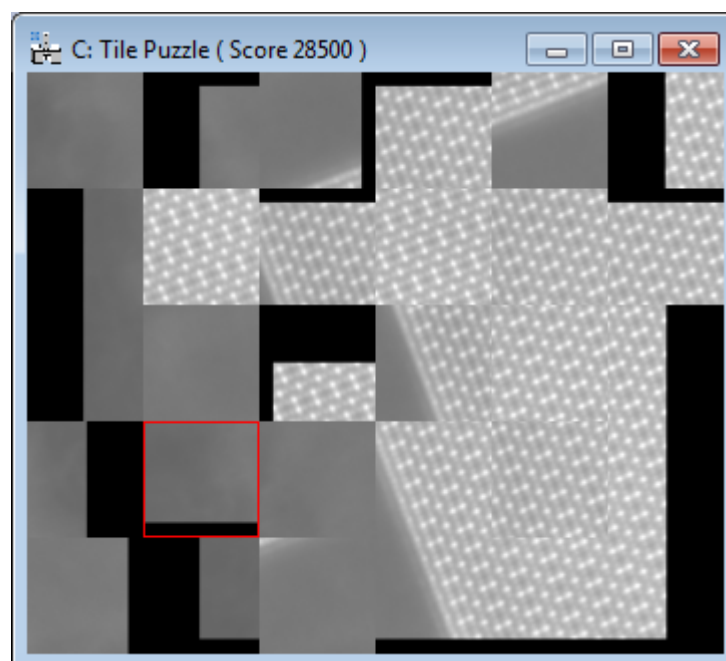
### *Tile Puzzler*

This script turns the front most displayed image into a tile-puzzle. The image can be any 2D raster display image ( also of type RGB ) and the original image is not changed.

On start, the user is presented with an option dialog in which he may enter the (approximated) number of tiles the image should be split into, a number of 'mixing steps' to scramble the tiles, whether or not tiles can be rotated, and whether or not correctly placed tiles are automatically indicated to the player.

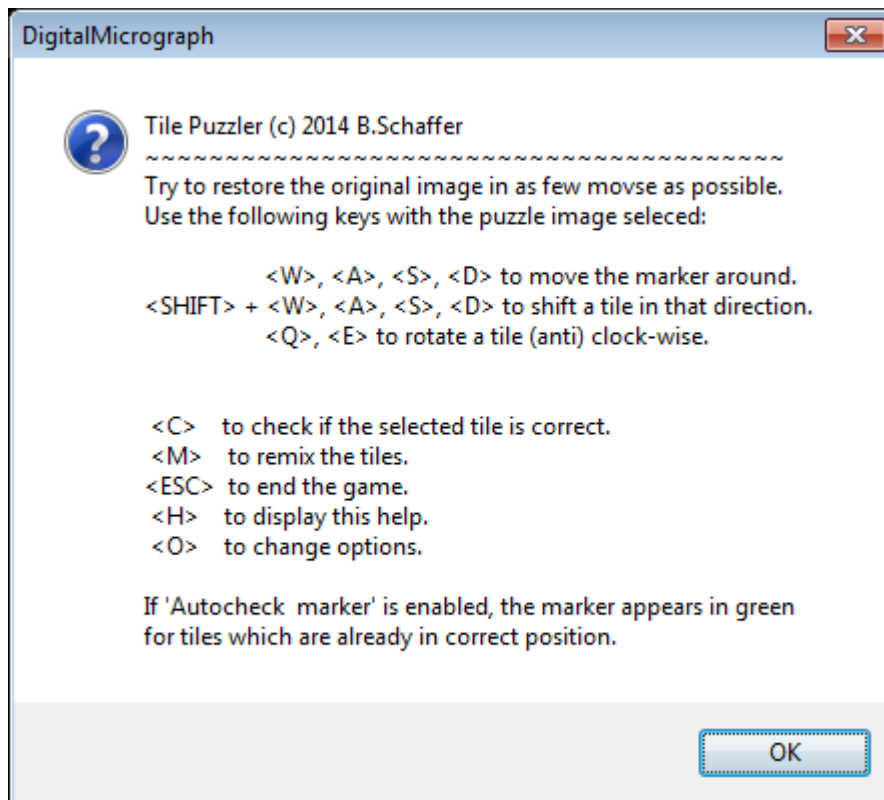


On okay, the game immediately starts. The image is split into the defined number of square tiles (some zero-padding is automatically applied, if needed.), and the tiles are scrambled and rotated. The puzzle-image is displayed with a tile marker on top of it. Note that the marker is generally red, but turns green if the 'Autocheck marker' option is activated and the highlighted tile is already correct in position and orientation.



The game is played via the keyboard with the puzzle image being selected. The goal is to put all tiles back to their correct position (and rotation) by swapping the highlighted tile with one of its neighbor tiles (and by rotating the highlighted tile.) The score is based on the number of used actions.

An overview of available keys is printed to the results window and may also be displayed as dialog by pressing the <H> key:

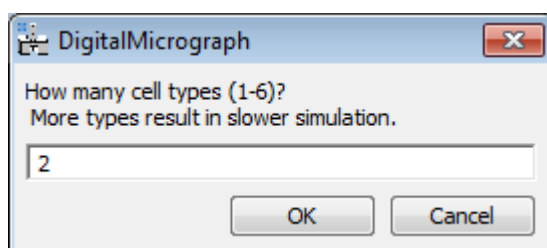


Once all tiles are correct, the winning dialog is shown and the player may restart another round or leave.



## Game of Life

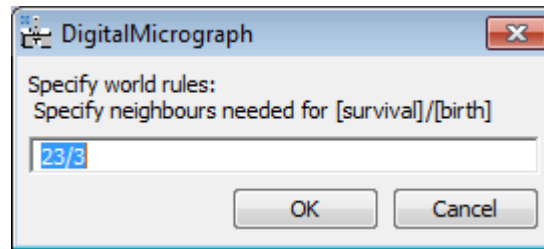
This script is a variant of the well-known '*Conway's Game of Life*' simulation<sup>1</sup>. It either uses the front most displayed image as source, or (if none found) creates a little simulation image with random start values. Different from the original simulation, this simulation allows up to 6 different cell types to be simulated simultaneously. When the simulation is launched, the user first specifies the number of cell types to be used.



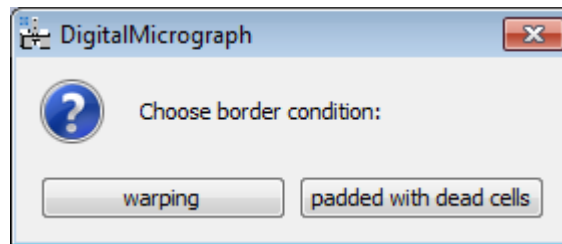
Then he specified the 'world rules' using the common world-name syntax string which consists of the 'survival' rule separated by a "/" from the 'birth' rule. Both birth and survival rule specify the necessary

<sup>1</sup> [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)

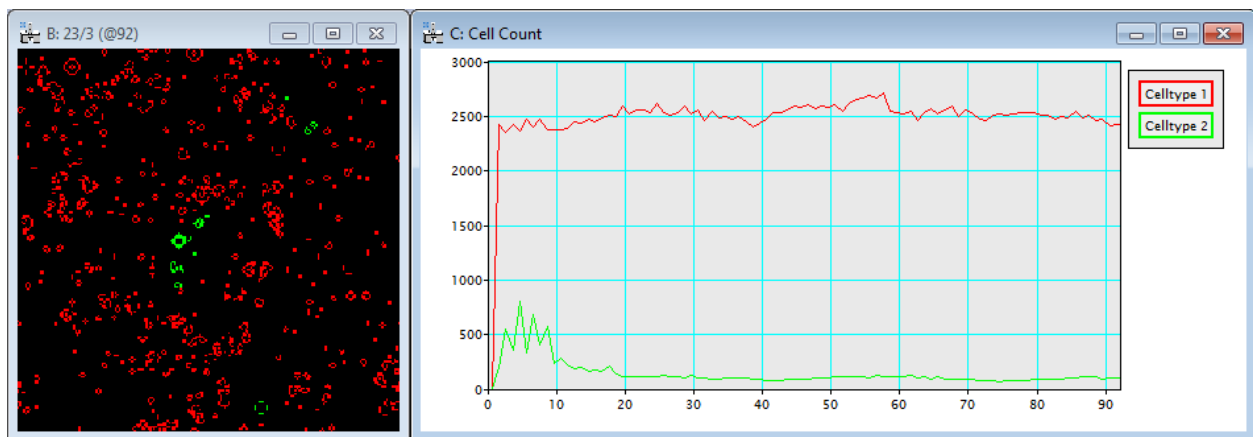
number of cell-neighbours for which the rule then applies. For 'birth' only neighbours of same cell type are counted, for 'death' cells of all types are counted. Hence, the rule set "23/3" will keep all cells alive which have 2 or 3 neighbours of any sort (but not more or less), and new cells are created if there are exactly three neighbours of specific cell type adjacent to a pixel.



Finally, the user chooses the border condition. Either the image is 'warped' around its edges, or it is 'padded' by dead cells.



Then the simulation immediately starts and continues until the user plays the SHIFT key. Two things are displayed during the simulation: The field with the current cell distribution, and a graph showing the history of the cell-count of various cell-types.



Final note: If you are interested in the script code for the simulation, it is explained as tutorial in the "How To Script" DigitalMicrograph tutorial e-book available on request from <http://DMScript.TavernMaker.de> .