Yatris Documentation

Licence Information

A little note

Feel free to send any comment and/or suggestion about the program. Moreover, if you add any feature, correct any bug and/or add different piece and graphic sets, it would be very kind of you to send them to me (you don't really have to, it's just that I'd like to have anything concering this game on my hard disk). If you do this, you will be mentioned as co-author in future releases of this game. Thanks very much!

Requirements

The program has been created on Linux Mandrake 10.0 Community and so far tested on that distribution only. It needs Python and Pygame installed on the system. The game should be easily portable on other architetcures and operating systems provided that the above packages are installed. On Linux Python programs run just like any script, which means that you just have to give the file execution permission and then run it. Under Windows you have to run the command "python yatris.py", or execute the game from inside a Python shell. It is also possible to create an executable file from a .py by means of the py2exe tool, but I have not tried it yet. If you do and get it working, please send me a copy of the .exe along with an explanation of how you did it.
If you have any problem running the game please report it to me via e-mail giving as many information as you can regarding your system, specifying in particular: OS (name and version/distribution), Python version, Pygame version.

Description

Even though I assume everyone is quite familiar with a common game such as Tetris (a real masterpiece of computer games as far as I'm concerned) I just want to remind you how it works if you have forgotten (or if you are too young to have ever played it).
There is a set of object, each one with its own shape, that come from the upper part of the window one at a time. You can move the objects sideways and downwards and rotate it clockwise. Whatever you do, the object will go down at a speed that increases as the game goes. The object will stop (and a new one will appear) as soon as it touches the lower part of the window or other objects. When a horizontal line is filled with objects, it will be removed and everything standing on it will fall down. The game ends when an object touches the upper part of the window.

Commands

Standard commands are the following:

Player 1	Player 2	Effect
a	j	move left
d	l	move right
s	k	move down
w	i	rotate

It is possible to redefine the keys both as a static configuration and each time the game is run. In the latter case just choose "options" from the main menu, then "redefine keys" and follow the on-screen instructions. In the first case you must instead edit the file yatris.conf and use the $$$TODO$$$ appropriate commands.

Customization

The game can be customized by changing the background image and objects' shape and graphic appearence.

Changing the background image

To change the background image you can simply substitute the original *background.jpg* file with a new one, which should have a size of 800x600. Larger images are likely to work as well (even though they will be shown only partally) while smaller images may leave some parts of the screen black.

Changing the objects' shapes

You can define any sort of shape for the objects that appear in the game. The current set of objects is the one which appeared in the original Tetris.
To redefine the objects you have to edit the file *yatris.conf* and add/remove/modify the lines which begin with the keyword piece. Each piece is defined by a matrix, whose rows are separated with a semicolon (;), while the elements within each row are separated with a comma (,). The only elements which can appear in a matrix are 0 and 1 (other values may lead to unexpected results), where 1 corresponds to a solid part of the objects, whereas 0 corresponds to void. For example:
`1,0;1,0;1,1`
defines the following matrix

```
1 0
1 0
1 1
```

which corresponds to an L-shaped piece. You can define as many pieces as you want, but remember that for each of them a set of four images, corresponding to the four possible rotations, must be provided, as I will explain with more details in the next paragraph.

Changes the objects' graphics

Each object has its own graphic appearence, which can be changed by replacing the original piece*

images with new ones having the same name and size. If you add the pieces you have to take the following rules into consideration:

- The name of each image must be pieceX_Y.jpg where X is the number of the piece, starting from 0 and corresponding to the order the objects appear in the configuration file, and Y is the number of the position , which varies from 0 to 3 according to how the object is rotated (0, 90, 180, 270 degrees). Thus a valid entry would be piece0_0 (meaning the first piece in the original position, i.e. the one defined by the matrix in the configuration file). Only images with extension .jpg can be user.
- The size of each image should be 40*matrix_dimension, which means that, if the matrix is 2*3 as is the above example, the first image must be 80x120 pixels or larger. The second image will be 120x80 instead, for the matrix has got size 3*2 once it is rotated by 90 degrees. You can paint the whole image, not only the parts which will be shown; the program will think about this. The coefficient 40 works perfectly at the moment, but later version of the game are likely to be able to manage different grid sizes (you can redefine grid size as for now, but you will get a warning on startup and some bugs are likely to occur during the game).

Configuration commands reference

Here follows a list of all the commands that you can use within the file *yatris.conf* in order to set up the game according to your preferences.

piece

Defines the shape of a piece according to how I have already described in the previous paragraphs.

grid_width and grid_height

Both these parameters take an argument, which define the number of cells horizontally and vertically. Both player have the same values. Default values are grid_width=10 and grid_height=14. Due to a minor bug, the real value of grid_height is by one greater than what you write here (meaning that the default value is 15 actually)

key_[action][player]

Defines a default key, corresponding to the action and optionally the player specified. Actions can be **left**, **right**, **down** or **rotate**, in which case the player number (1 or 2) must be provided. Other actions are **fullscreen** and **pause**, which don't take any player argument. The parameter specifies the Unicode value of the key, allowing you to set any key on your keyboard. If you don't know a key ascii code, simply run the came, choose Options, then Redefine Keys and select the keys you want to use; in the shell the corresponding Unicode values will be printed so that you can use them inside the configuration file. For example, if you want to use 's' key for player 1 down command, type 'key_down1 115'.

start_speed

Defines the start speed by which the pieces fall down. A speed of 1 means that pieces fall down by one cell every second, while a speed of 0.5 means that they do every half a second and so on.

speed_increment

Defines how fast the speed is increased. This value must be between 0 (meaning that speed never changes) and 1 (which is not a correct value, for you would be saying that the speed must be increased by infinite). It's a good idea to keed this parameter quite low, otherwise games will end up going too fast in just a few seconds.

maximum_speed

Defines the maximum speed by which pieces fall.

theme

Defines a directory (which must be a subdirectory of where the game is run) to be used to get images (and sounds in future) from, so you don't have to copy files from a directory to another each time you want to change theme.