

PixelArtTikz [en]

PixelArts, with TikZ,
with solution and colors.

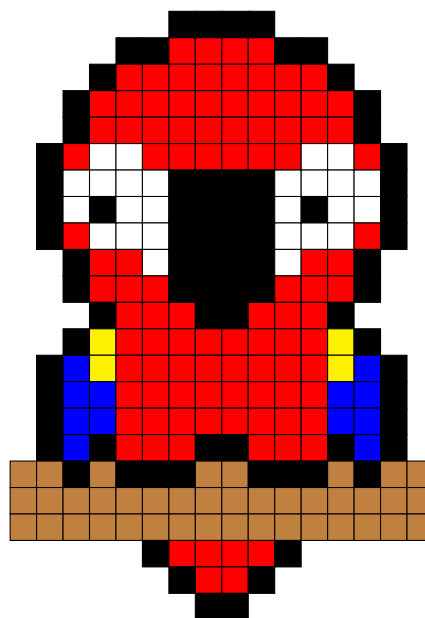
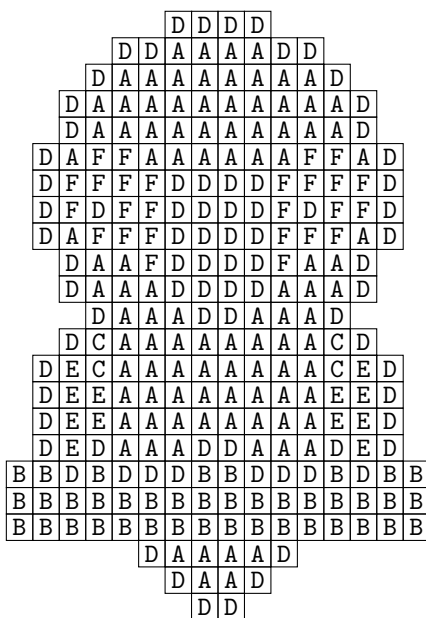
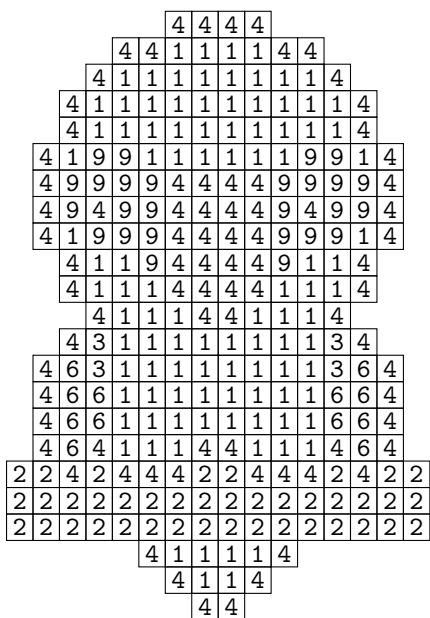
Version 0.1.1 - 19/04/2023

Cédric Pierquet

c pierquet - at - outlook . fr

<https://github.com/cpierquet/PixelArtTikz>

- Commands to display PixelArts.
- Environment to complete the PixelArt.



L^AT_EX

pdfL^AT_EX

LuaL^AT_EX

TikZ

T_EXLive

MiK_TE_X

Contents

Part I

Introduction

1 The package PixelArtTikz

1.1 Introduction

The idea is to *propose*, within a TikZ environment, a macro to generate PixelArt.

Datas are *red* by a csv file, already created and placed into the folder of the tex file, or directly created by filecontents.

Some advices about the cvs file :

- the csv file must use "," as separator ;
- empty cases are coded by "-".

```
\begin{filecontents*}{filename.csv}
  A,B,C,D
  A,B,D,C
  B,A,C,D
  B,A,D,C
\end{filecontents*}
```

Code \LaTeX

While compiling, the file filename.csv will be created, and the option \langle **[overwrite]** \rangle will propagate the modifications !

1.2 Loading of the package, and option

The *needed* package is here csvsimple, in order to read the csv file.

It's available for $\LaTeX 2_{\epsilon}$ or for $\LaTeX 3$. By default, PixelArtTikz loads it for $\LaTeX 3$, but an *option* is available to work with $\LaTeX 2_{\epsilon}$.

The option \langle **[csvii]** \rangle forces the usage of $\LaTeX 2_{\epsilon}$.

```
\usepackage{PixelArtTikz}           %package latex3
%which loads
%\RequirePackage{expl3}
%\RequirePackage[13]{csvsimple}

\usepackage[csvii]{PixelArtTikz}    %package latex2
%which loads
%\RequirePackage[legacy]{csvsimple}
```

Code \LaTeX

1.3 Used packages

It's fully compatible with usuals compilations, such as latex, pdflatex, lualatex or xelatex.

It loads the packages and libraries :

- tikz, xintexpr et xinttools;
- xstring, xparse, simplekv and listofitems.

1.4 Macros and environment

There's two ways to create PixelArt :

- by an independent macro ;
- by a TikZ environment in order to put code after.

Code \LaTeX

```
%Independent macro
\PixelArtTikz[keys]<options tikz>{file.csv}

%Semi-independent macro, in a tiks environment
\PixelArtTikz*[keys]{file.csv}

%environment
\begin{EnvPixelArtTikz}[keys]<options tikz>{file.csv}
  %tikz code
\end{EnvPixelArtTikz}
```

2 Colors

Concerning colors, the user can use the colors loaded by the loaded packages !

Without extra package, avaiaibles colors are :

magenta	cyan	blue	green	red	darkgray	olive	lime	brown	lightgray
white	gray	black	yellow	violet	teal	purple	pink	orange	

Part II

Macros and environment

3 Main macro

3.1 Example

The macro `\PixlArtTikz` needs :

- the file `csv` ;
- the list (by a string) of codes used in the file `csv` (eg `234679` or `ABCDJK...`);
- the list of symbols (if needed) to print in the cases, eg `25,44,12` or `AA,AB,AC` ;
- the list of colors (for the correction), same order as the codes.

We can begin by creating the file `csv`, directly within the `tex` code, or with a external file.

```
%creation of the csv
\begin{filecontents*}[overwrite]{base.csv}
  A,B,C,D
  A,B,D,C
  B,A,D,C
  C,A,B,D
\end{filecontents*}
```

Code \LaTeX

```
%instructions and pixelarts
\begin{center}
  \begin{tblr}{colspec={*{4}{Q[1.25cm,c,m]}},hlines,vlines,rows={1.15em}}
    \SetCell[c=4]{c} Instructions & & & \\
    A & B & C & D \\
    45 & 22 & 1 & 7 \\
    Black & Green & Yellow & Red \\
  \end{tblr}
\end{center}

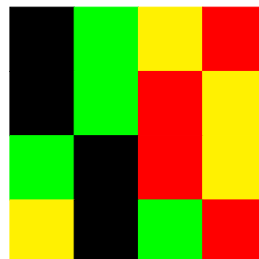
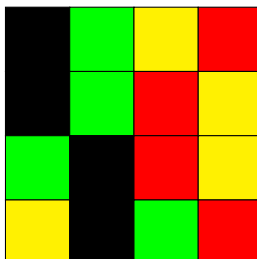
\PixlArtTikz[Codes=ABCD,Style=\large\sffamily,Unit=0.85]{base.csv}
~~
\PixlArtTikz[Codes=ABCD,Symbols={45,22,1,7},Symb,Style=\large\sffamily,Unit=0.85]{base.csv}
~~
\PixlArtTikz[Codes=ABCD,Colors={black,green,yellow,red},Correction,Unit=0.85]{base.csv}
~~
\PixlArtTikz[Codes=ABCD,Colors={black,green,yellow,red},Correction,Border=false,Unit=0.85]{base.csv}
```

Code \LaTeX

Instructions			
A	B	C	D
45	22	1	7
Black	Green	Yellow	Red

A	B	C	D
A	B	D	C
B	A	D	C
C	A	B	D

45	22	1	7
45	22	7	1
22	45	7	1
1	45	22	7



3.2 Options and keys

```
\PixlArtTikz[keys]<options tikz>{file.csv}
```

Code \LaTeX

The first argument, *optional* and between [...] proposes the keys :

- the key **<Codes>** with the *string* of *simple* codes of the csv file ;
- the key **<Colors>** with the *list* of colors ;
- the key **<Symbols>** with the *optional list* of alt. symbols for the cases ;
- the boolean **<Correction>** to color the PixelArt ; default false
- the boolean **<Symb>** to print the symbols ; default false
- the boolean **<Border>** to print borders of the cases ; default true
- the key **<Style>** to specify the style of the text. default \scriptsize

The second argument, *optional* and between <...> are options – in TikZ – to parse to the environment which create the PixelArt.

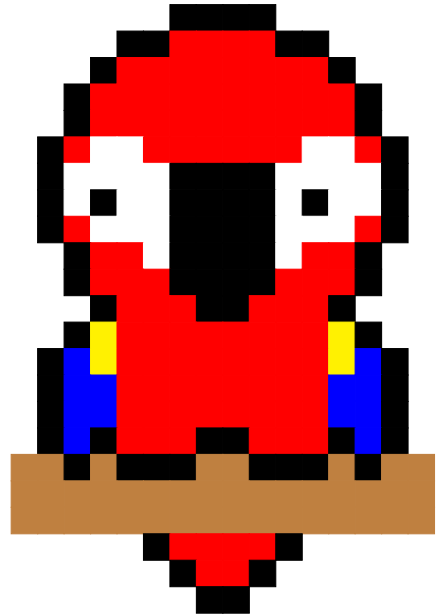
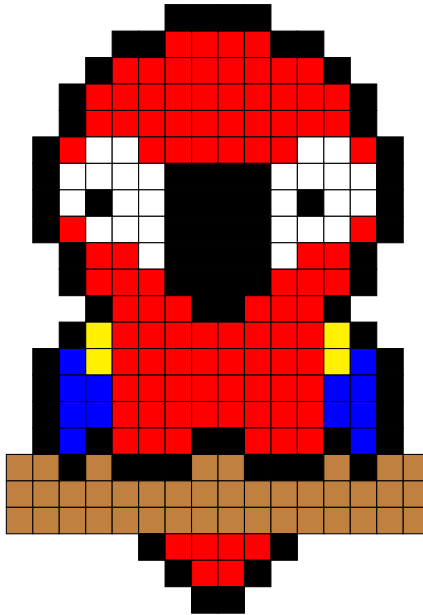
The third argument, *mandatory*, is the filename of the csv.

```
%creation of the csv
\begin{filecontents*}[overwrite]{test1.csv}
  -,,-,-,-,4,4,4,4,-,-,-,-,-
  -,,-,-,4,4,1,1,1,1,4,4,-,-,-
  -,,-,-,4,1,1,1,1,1,1,1,1,4,-,-,-
  -,,-,4,1,1,1,1,1,1,1,1,1,4,-,-
  -,,-,4,1,1,1,1,1,1,1,1,1,1,4,-,-
  -,4,1,9,9,1,1,1,1,1,1,9,9,1,4,-
  -,4,9,9,9,9,4,4,4,4,9,9,9,9,4,-
  -,4,9,4,9,9,4,4,4,4,9,4,9,9,4,-
  -,4,1,9,9,9,4,4,4,4,9,9,9,1,4,-
  -,,-,4,1,1,9,4,4,4,4,9,1,1,4,-,-
  -,,-,4,1,1,1,4,4,4,4,1,1,1,4,-,-
  -,,-,-,4,1,1,1,4,4,1,1,1,4,-,-,-
  -,,-,4,3,1,1,1,1,1,1,1,1,3,4,-,-
  -,4,6,3,1,1,1,1,1,1,1,1,3,6,4,-
  -,4,6,6,1,1,1,1,1,1,1,1,6,6,4,-
  -,4,6,6,1,1,1,1,1,1,1,1,6,6,4,-
  -,4,6,4,1,1,1,4,4,1,1,1,4,6,4,-
  2,2,4,2,4,4,4,2,2,4,4,4,2,4,2,2
  2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
  2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
  -,,-,-,-,4,1,1,1,4,-,-,-,-,-
  -,,-,-,-,-,4,1,1,4,-,-,-,-,-
  -,,-,-,-,-,4,4,-,-,-,-,-,-
\end{filecontents*}
```

Code \LaTeX

```
%simple codes
%empty case with -
\PixelArtTikz[Codes=123469,Style=\ttfamily,Unit=0.35]{test1.csv}
~~
\PixelArtTikz[Codes=123469,Colors={red,brown,yellow,black,blue,white},Correction,Unit=0.35]{test1.csv}
~~
\PixelArtTikz[Codes=123469,Colors={red,brown,yellow,black,blue,white},Correction,Unit=0.35,Border=false]{test1.csv}
```

			4	4	4	4									
		4	4	1	1	1	1	4	4						
	4	1	1	1	1	1	1	1	4						
	4	1	1	1	1	1	1	1	4						
4	1	9	9	1	1	1	1	1	9	9	1	4			
4	9	9	9	9	4	4	4	4	9	9	9	9	4		
4	9	4	9	9	4	4	4	4	9	4	9	9	4		
4	1	9	9	9	4	4	4	4	9	9	9	1	4		
	4	1	1	9	4	4	4	4	9	1	1	4			
	4	1	1	1	4	4	4	4	1	1	1	4			
		4	1	1	1	4	4	1	1	1	4				
		4	3	1	1	1	1	1	1	1	3	4			
	4	6	3	1	1	1	1	1	1	1	3	6	4		
	4	6	6	1	1	1	1	1	1	1	6	6	4		
	4	6	6	1	1	1	1	1	1	1	6	6	4		
	4	6	4	1	1	1	4	4	1	1	1	4	6	4	
2	2	4	2	4	4	4	2	2	4	4	4	2	4	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		4	1	1	1	1	4								
		4	1	1	4										
		4	4												



In the following example, les *symbols* to print can't be used for the *codes*, so we can use the keys **(Symbols)** and **(Symb)** to bypass this limitation.

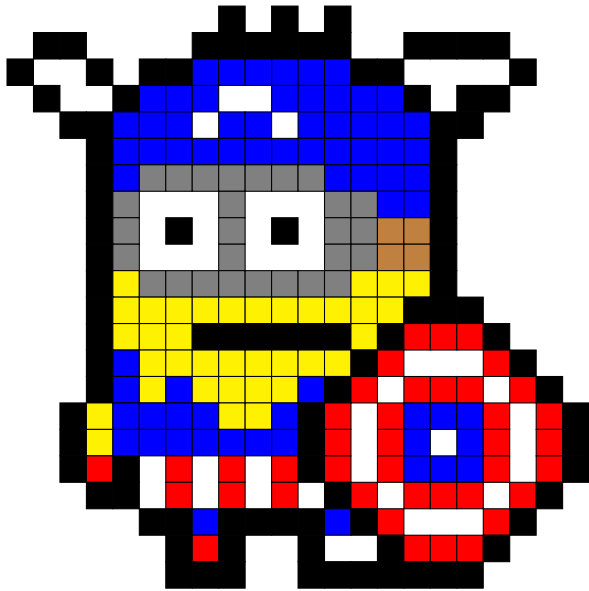
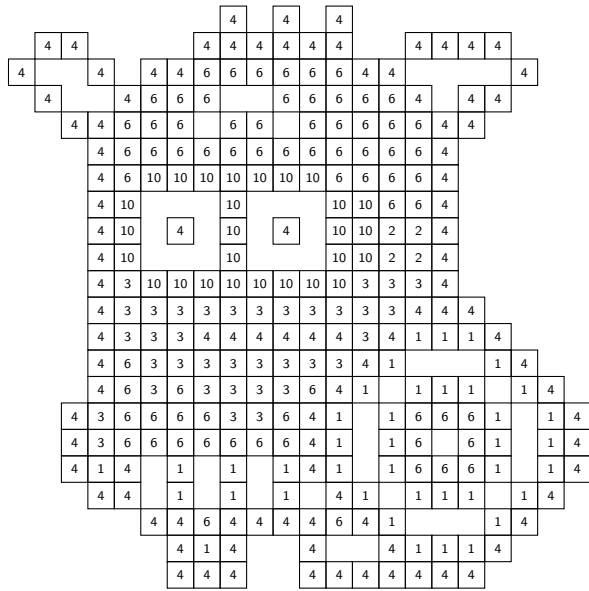
%symbols associated to codes

```

\begin{filecontents*}[overwrite]{cap.csv}
-,-,-,-,-,-,-,D,-,D,-,-,-,-,-,-,-,-,-,-,-,-,-
-,D,D,-,-,-,-,D,D,D,D,D,-,-,-,D,D,D,D,-,-,-
D,-,-,D,-,D,D,F,F,F,F,F,F,D,D,-,-,-,-,D,-,-
-,D,-,-,D,F,F,F,-,-,F,F,F,F,F,D,-,D,D,-,-,-,-
-,D,D,D,F,F,F,-,F,F,-,F,F,F,F,F,D,D,-,-,-,-,-
-,,-,-,D,F,F,F,F,F,F,F,F,F,F,F,D,-,-,-,-,-,-
-,,-,-,D,F,J,J,J,J,J,J,J,J,F,F,F,D,-,-,-,-,-,-
-,,-,-,D,J,-,-,-,J,-,-,-,J,J,F,F,D,-,-,-,-,-,-
-,,-,-,D,J,-,D,-,J,-,D,-,J,J,B,B,D,-,-,-,-,-,-
-,,-,-,D,J,-,-,-,J,-,-,-,J,J,B,B,D,-,-,-,-,-,-
-,,-,-,D,C,J,J,J,J,J,J,J,J,C,C,C,D,-,-,-,-,-,-
-,,-,-,D,C,C,C,C,C,C,C,C,C,C,C,C,D,D,-,-,-,-,-
-,,-,-,D,C,C,C,D,D,D,D,D,D,C,D,A,A,A,D,-,-,-,-
-,,-,-,D,F,C,C,C,C,C,C,C,C,D,A,-,-,-,A,D,-,-,-
-,,-,-,D,F,C,F,C,C,C,C,F,D,A,-,A,A,A,-,A,D,-,-
-,,-,D,C,F,F,F,F,C,C,F,D,A,-,A,F,F,F,A,-,A,D,-
-,,-,D,C,F,F,F,F,F,F,F,D,A,-,A,F,-,F,A,-,A,D,-
-,,-,D,A,D,-,A,-,A,-,A,D,A,-,A,F,F,F,A,-,A,D,-
-,,-,-,D,D,-,A,-,A,-,A,-,D,A,-,A,A,A,-,A,D,-
-,,-,-,-,D,D,F,D,D,D,D,F,D,A,-,-,-,A,D,-,-,-
-,,-,-,-,-,D,A,D,-,-,D,-,-,D,A,A,A,D,-,-,-,-
-,,-,-,-,-,D,D,D,-,-,D,D,D,D,D,D,D,-,-,-,-,-
\end{filecontents*}

\PixelArtTikz[Codes=ABCDJ,Symbols={1,2,3,4,6,10},Symb,Style=\tiny\sffamily,Unit=0.35]{cap.csv}
~~
\PixelArtTikz[Codes=ABCDJ,Colors={red,brown,yellow,black,blue,gray},Correction,Unit=0.35]{cap.csv}

```

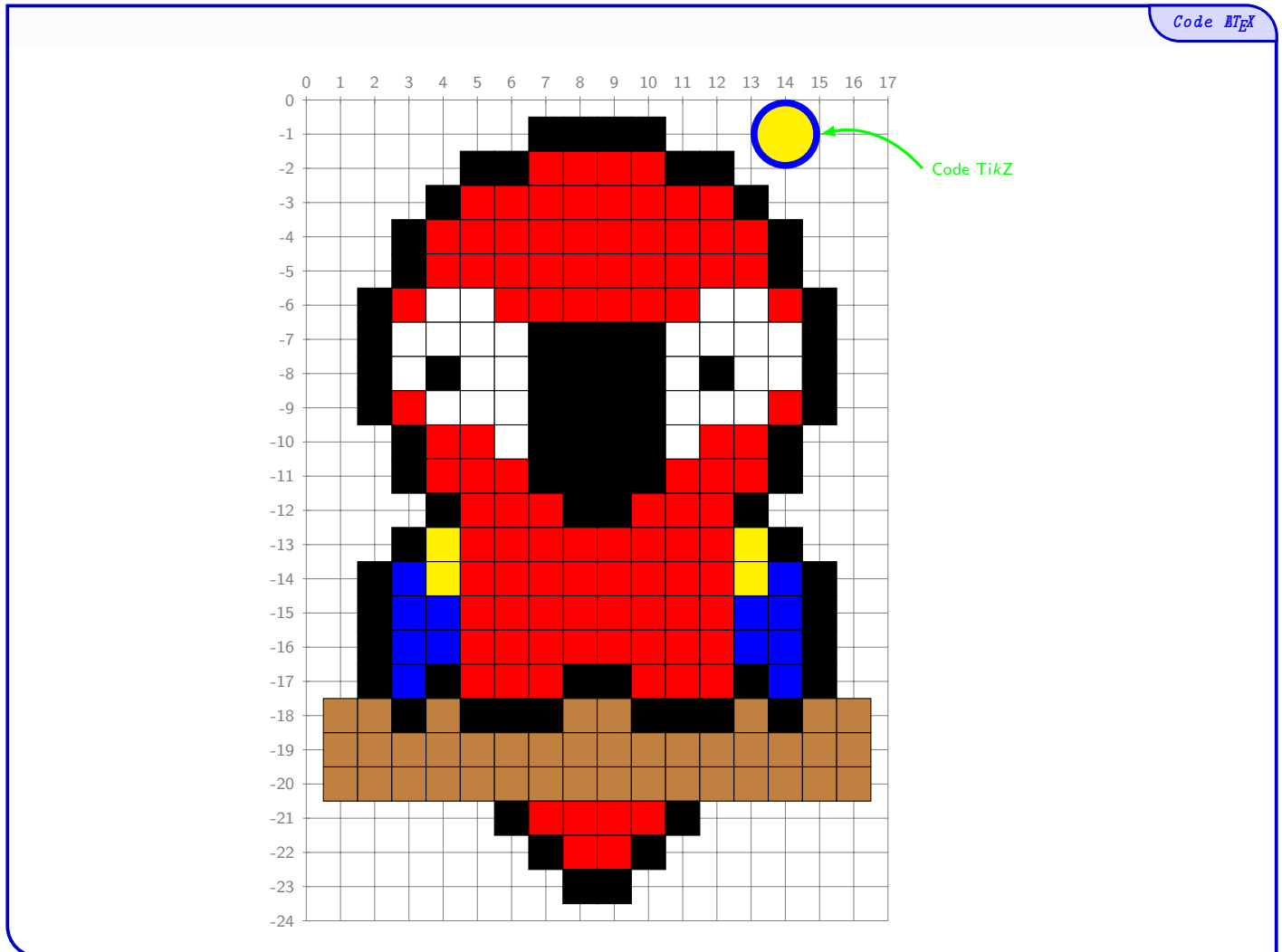


3.3 Starred macro

The starred *étoilée* macro `\PixelArtTikz*` is to be integrated within an environment already created. It can be useful to add code after the `PixelArt`.

In this case :

- the *optional* between `<...>` is useless ;
- the key `<Unit>` is useless too (units can be configured in the environment !)



4 PixelArt environment

4.1 Usage

The package PixelArtTikz proposes an environment to create a PixelArt, and to add code after.

- The environment is created within TikZ and added code is to give in TikZ !
- The added code will be print "above" the PixelArt !

Code \LaTeX

The first argument, *optional* and between [...] proposes the keys :

- the key **<Codes>** with the *string* of *simple* codes of the csv file ;
- the key **<Colors>** with the *list* of colors ;
- the key **<Symbols>** with the *optional list* of alt. symbols for the cases ;
- the boolean **<Correction>** to color the PixelArt ; default false
- the boolean **<Symb>** to print the symbols ; default false
- the boolean **<Border>** to print borders of the cases ; default true
- the key **<Style>** to specify the style of the text. default \scriptsize

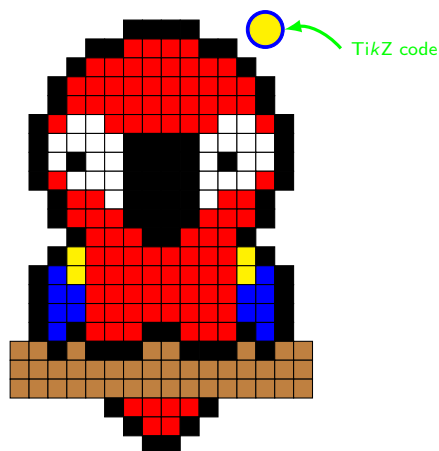
The second argument, *optional* and between <...> are options – in TikZ – to parse to the environment which create the PixelArt.

The third argument, *mandatory*, is the filename of the csv.

4.2 Exemple

The symbols are at the nodes ($c; -l$) where l and c are the row and column of the data in the csv file.

Code \LaTeX



Part III

Historique

v0.1.1 : Bugfix with color

v0.1.0 : Initial version