PixelArtTikz [en]

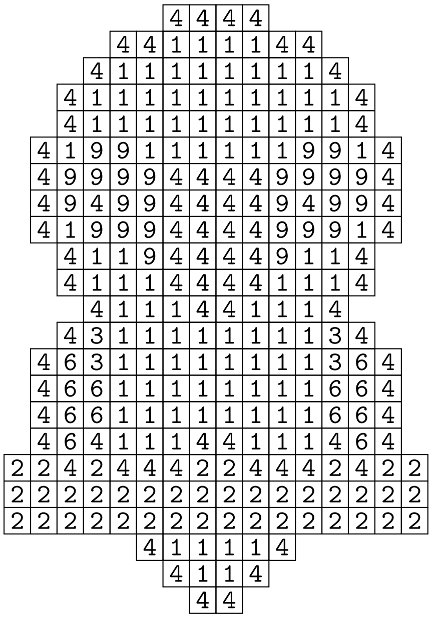
PixelArts, with Ti*k*Z,

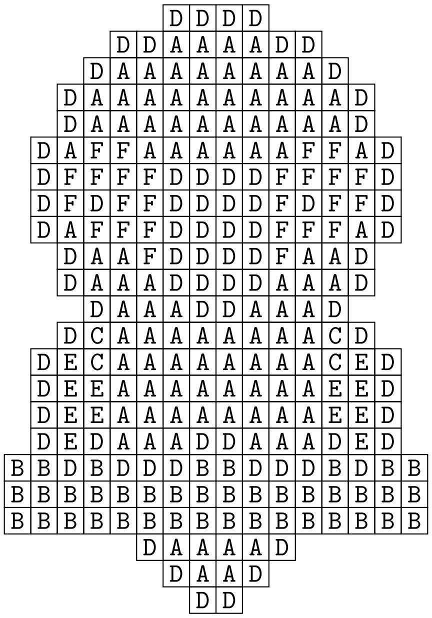
with solution and colors.

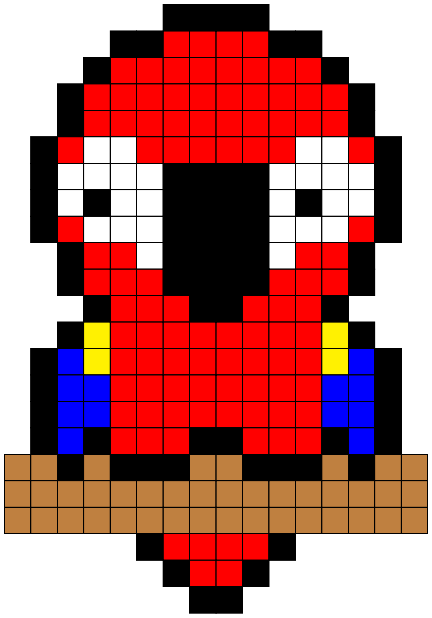
Version 0.1.1 - 19/04/2023

Cédric Pierquet
c pierquet - at - outlook . fr
https://github.com/cpierquet/PixelArtTikz

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




D D D D
D D A A A A D D
D A A A A A A A A D
D A A A A A A A A A A D
D A A A A A A A A A A D
D A F F A A A A A A F F A D
D F F F F D D D D F F F F D
D F D F F D D D D F D F F D
D A F F F D D D D F F F A D
D A A F D D D D F A A D
D A A A D D D D A A A D
D A A A D D A A A D
D C A A A A A A A A C D
D E C A A A A A A A A C E D
D E E A A A A A A A A E E D
D E E A A A A A A A A E E D
D E D A A A D D A A A D E D
B B D B D D D B B D D D B D B B
B B B B B B B B B B B B B B B B
B B B B B B B B B B B B B B B B
D A A A A D
D A A D
D D



LaTeX

pdfLaTeX

LuaLaTeX

TikZ

TeXLive

MiKTeX

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.
The data is *read* from a csv file, already existing in the folder of the tex file, or created on-the-fly by filecontents.

Some advices about the cvs file :

- the csv file must use "," as separator;
- empty cells 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*}
```

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

1.2 Loading of the package, and option

The package csvsimple is necessary in order to read the csv file.
The package is available in two versions, one written in LaTeX 2ε and the other in LaTeX3. By default, PixelArtTikz loads the LaTeX3 version, but an *option* is available to work with the LaTeX 2ε version.

The option **⟨[csvii]⟩** forces the usage of the LaTeX 2ε version.

```
\usepackage{PixelArtTikz}                   %package latex3
%which loads
%\RequirePackage{expl3}
%\RequirePackage[l3]{csvsimple}

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

1.3 Used packages

It's fully compatible with usual LaTeX engines, such as latex, pdflatex, lualatex or xelatex.

It loads the following packages and libraries:

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

1.4 Macros and environment

There are two ways to create PixelArt:

- with an independent macro;
- with a Ti*k*Z environment in order to add code afterwards.

Code LaTeX

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

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

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

2 Colors

Concerning colors: the user can use all colors provided by loaded packages!
Without extra packages, the available 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 (e.g. `234679` or `ABCDJK`...);
- the list of symbols (if needed) to print in the cells, e.g. `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.

Code LaTeX

```
%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*}
```

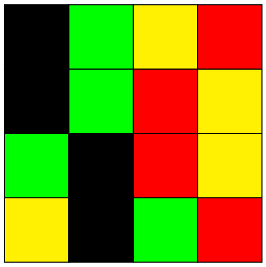
Code LaTeX

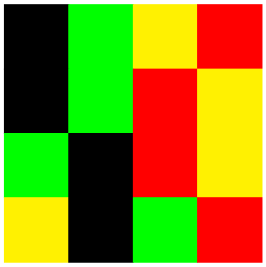
```
%instructions and pixelarts
\begin{center}
    \begin{tblr}{colspec={*{4}{Q[1.25cm,c,m]}},hlines,vlines,rows={1.15em}}
        \SetCell[c=4]{c} Instructions & & & \\
        A & B & C & D \\
        45 & 22 & 1 & 7 \\
        Black & Green & Yellow & Red \\
    \end{tblr}
\end{center}

\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}
```

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

Code LaTeX

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

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 cells;
- 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 cells; default true
- the key ⟨**Style**⟩ to specifythe style of the text. default \scriptsize

The second argument, *optional* and between `<...>`, are TikZ options to pass on to the environment which creates the PixelArt.

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

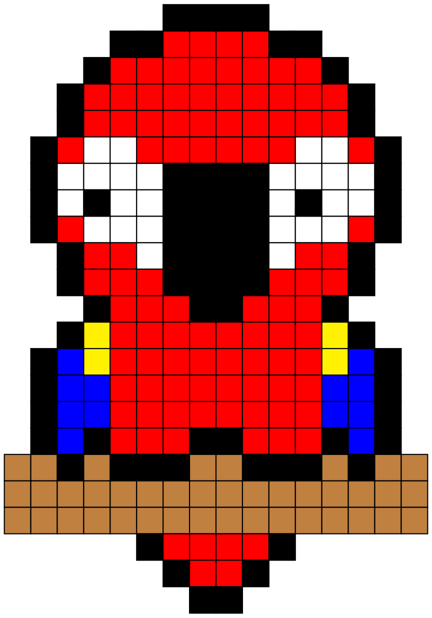
Code LaTeX

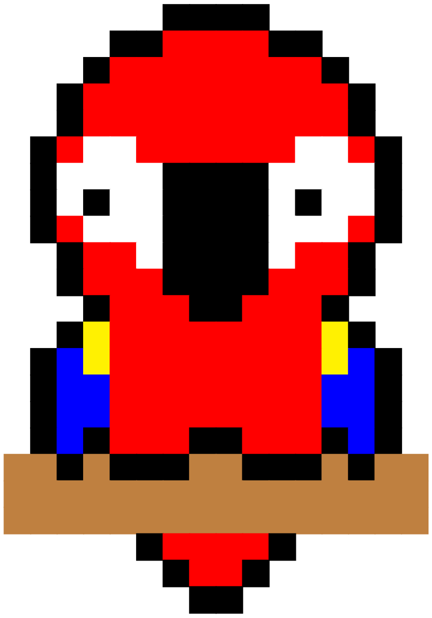
```
%creation of the csv
\begin{filecontents*}[overwrite]{test1.csv}
    -,-,-,-,-,-,4,4,4,4,-,-,-,-,-,-
    -,-,-,-,4,4,1,1,1,1,4,4,-,-,-,-
    -,-,-,4,1,1,1,1,1,1,1,1,4,-,-,-
    -,-,4,1,1,1,1,1,1,1,1,1,1,4,-,-
    -,-,4,1,1,1,1,1,1,1,1,1,1,4,-,-
    -,4,1,9,9,1,1,1,1,1,1,9,9,1,4,-
    -,4,9,9,9,9,4,4,4,4,9,9,9,9,4,-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

Code LaTeX

```
%simple codes
%empty case with -
\PixlArtTikz[Codes=123469,Style=\ttfamily,Unit=0.35]{test1.csv}
~~
\PixlArtTikz[Codes=123469,Colors={red,brown,yellow,black,blue,white},Correction,Unit=0.35]{test1.csv}
~~
\PixlArtTikz[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	1	4			
		4	1	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	2	2
					4	1	1	1	1	4					
						4	1	1	4						
							4	4							

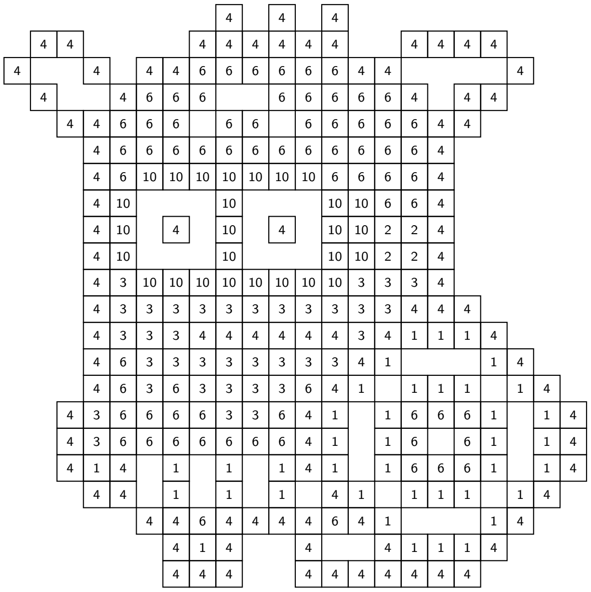


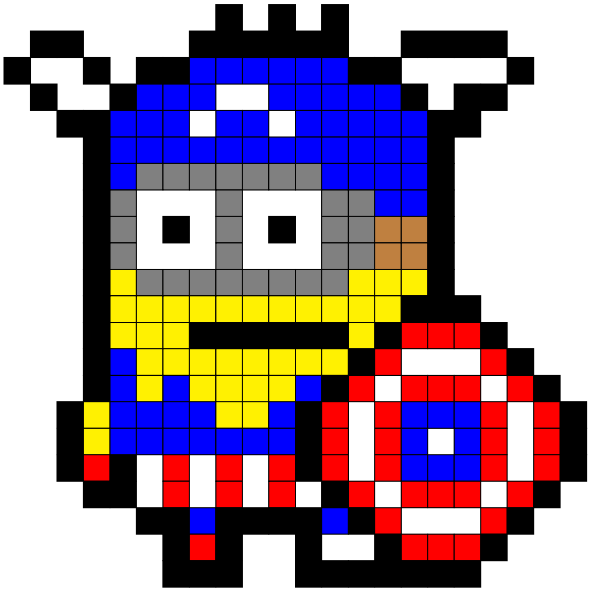


In the following example, the *symbols* to print can't be used for the *codes*, so we can use the keys ⟨**Symbols**⟩ and ⟨**Symb**⟩ to bypass this limitation.

Code LaTeX

```
%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,-,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,F,F,F,-,F,F,-,F,F,F,F,F,D,D,-,-,-,-
    -,-,-,D,F,F,F,F,F,F,F,F,F,F,F,F,D,-,-,-,-,-
    -,-,-,D,F,J,J,J,J,J,J,J,F,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,D,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*}

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





3.3 Starred macro

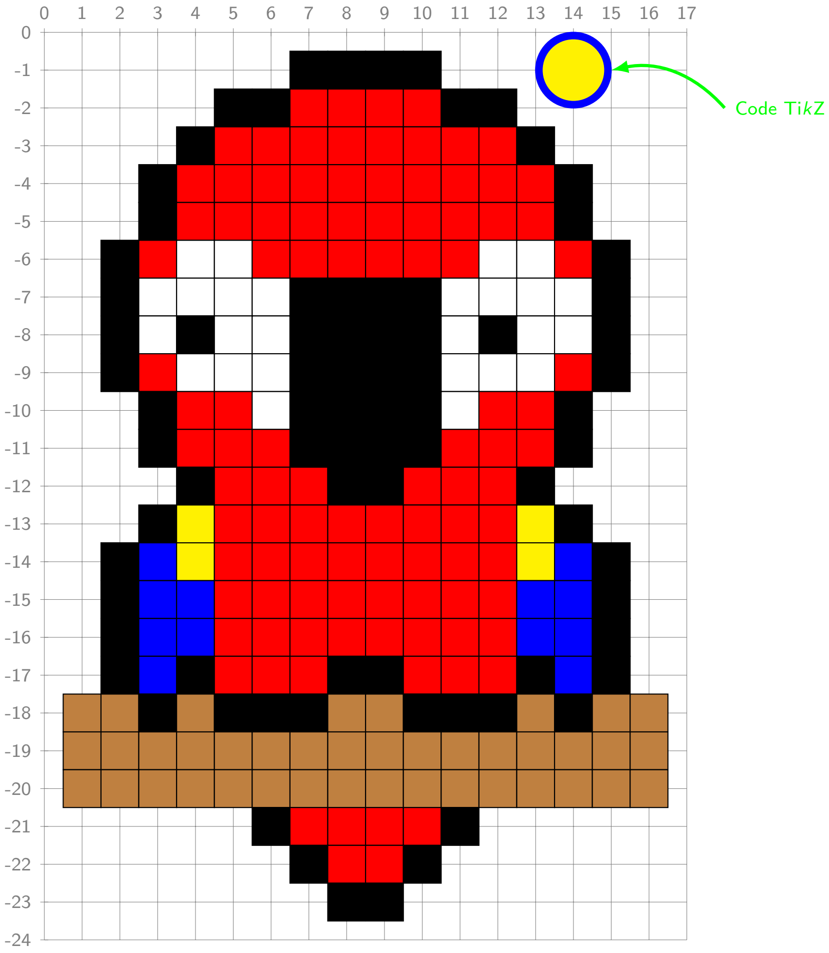
The starred macro `\PixlArtTikz*` is to be used within an already created environment. It can be useful for adding code after the PixelArt.

In this case:

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

Code LaTeX

```
\begin{center}
    \begin{tikzpicture}[scale=0.5]
        %grid to show positionning
        \draw[very thin,gray,xstep=1,ystep=1] (0,0) grid (17,-24) ;
        \foreach \x in {0,1,...,17} \draw[very thin,gray] (\x,-3pt)--(\x,3pt)%
        node[above,font=\scriptsize\sffamily] {\x} ;
        \foreach \y in {0,-1,...,-24} \draw[very thin,gray] (3pt,\y)--(-3pt,\y)%
        node[left,font=\scriptsize\sffamily] {\y} ;
        %le PixelArt
        \PixlArtTikz*[Codes=123469,Colors={red,brown,yellow,black,blue,white},Correction]{test1.csv}
        %added code
        \filldraw[blue] (14,-1) circle[radius=1] ;
        \filldraw[yellow] (14,-1) circle[radius=0.8] ;
        \draw[green,very thick,<-,>=latex] (15,-1) to[bend left=30] (18,-2)%
        node[right,font=\scriptsize\sffamily] {Code Ti\textit{k}Z} ;
    \end{tikzpicture}
\end{center}
```



4 PixelArt environment

4.1 Usage

The package PixelArtTikz provides an environment to create a PixelArt and add code afterwards.

- The environment is created within Ti*k*Z and additional code is passed on to the Ti*k*Z environment!
- The additional code will be printed on top of the PixelArt!

Code LaTeX

```
\begin{EnvPixlArtTikz}[keys]<options tikz>{filename.csv}
    %tikz code(s)
\end{EnvPixlArtTikz}
```

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 cells;
- 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 cells; default true
- the key ⟨**Style**⟩ to specifythe style of the text. default \scriptsize

The second argument, *optional* and between `<...>`, is for Ti*k*Z options to be passed on to the environment which creates the PixelArt.

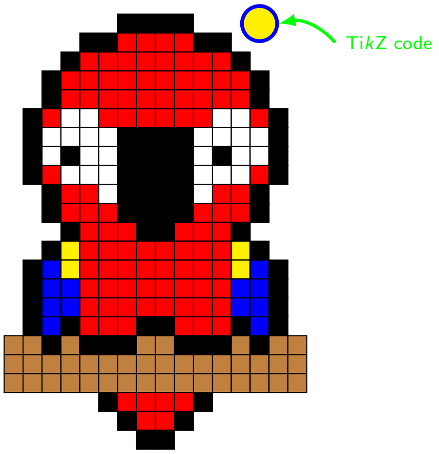
The third argument, *mandatory*, is the filename of the csv.

4.2 Example

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

```
\begin{center}
    \begin{EnvPixlArtTikz}%
            [Codes=123469,Colors={red,brown,yellow,black,blue,white},Correction,Unit=0.25]
            {test1.csv}
        \filldraw[blue] (14,-1) circle[radius=1] ;
        \filldraw[yellow] (14,-1) circle[radius=0.8] ;
        \draw[green,very thick,<-,>=latex] (15,-1) to[bend left=30] (18,-2)%
        node[right,font=\scriptsize\sffamily] {Ti\textit{k}Z code} ;
    \end{EnvPixlArtTikz}
\end{center}
```



Part III
Historique

`v0.1.1` : Bugfix with color
`v0.1.0` : Initial version