PixelArtTikz [en]

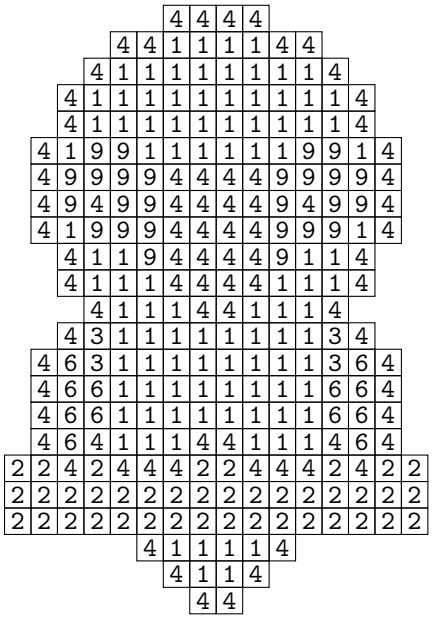
PixelArts, with Ti*k*Z,

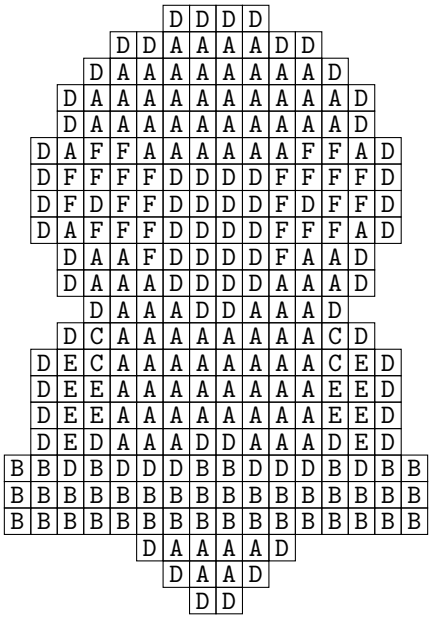
with solution and colors.

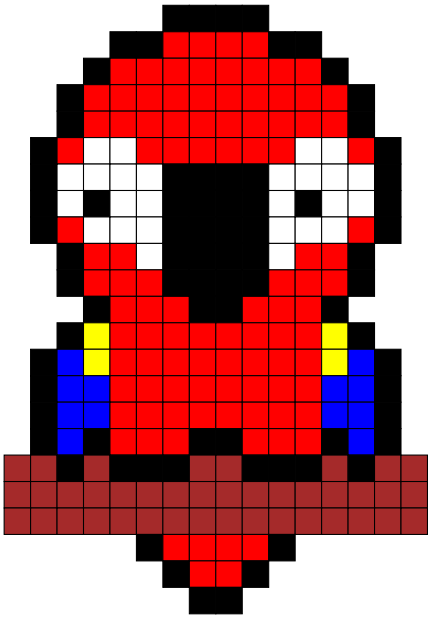
Version 0.1.0 - 23/01/2023

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

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







LaTeX

pdfLaTeX

LuaLaTeX

TikZ

TeXLive

MiKTeX

Contents

Part I
Introduction

1 The package PixelArtTikz

1.1 Introduction

The idea is to *propose*, within a Ti*k*Z 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 "-".

Code LaTeX
```
\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 *needed* package is here csvsimple, in order to read the csv file.
It's available for LaTeX 2_ε or for LaTeX3. By default, PixelArtTikz loads it for LaTeX3, but an *option* is available to work with LaTeX 2_ε.
The option **⟨[csvii]⟩** forces the usage of LaTeX 2_ε.

Code LaTeX
```
\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 copatible 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 Ti*k*Z environment in order to put code after.

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}
```

For the colors, its depending from the loaded packages.
This documentation was compiled with xcolor, with ⟨**[table,svgnames]**⟩ options.

Part II
Macros and environment

2 Main macro

2.1 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.

Code LaTeX

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

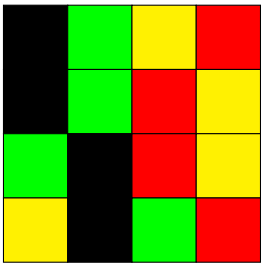
Code LaTeX

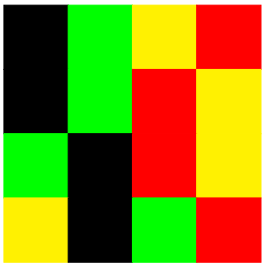
```
%instructions and pixelarts
\begin{center}
    \begin{tblr}{colspec={*{4}{Q[1.25cm,c,m]}},hlines,vlines,rows={1.15em}}
        \SetCell[c=4]{c} Instructions & & & \\
        A & B & C & D \\
        45 & 22 & 1 & 7 \\
        Black & Green & Yellow & Red \\
    \end{tblr}
\end{center}

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





2.2 Options an 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 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 specifythe style of the text. default \scriptsize

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

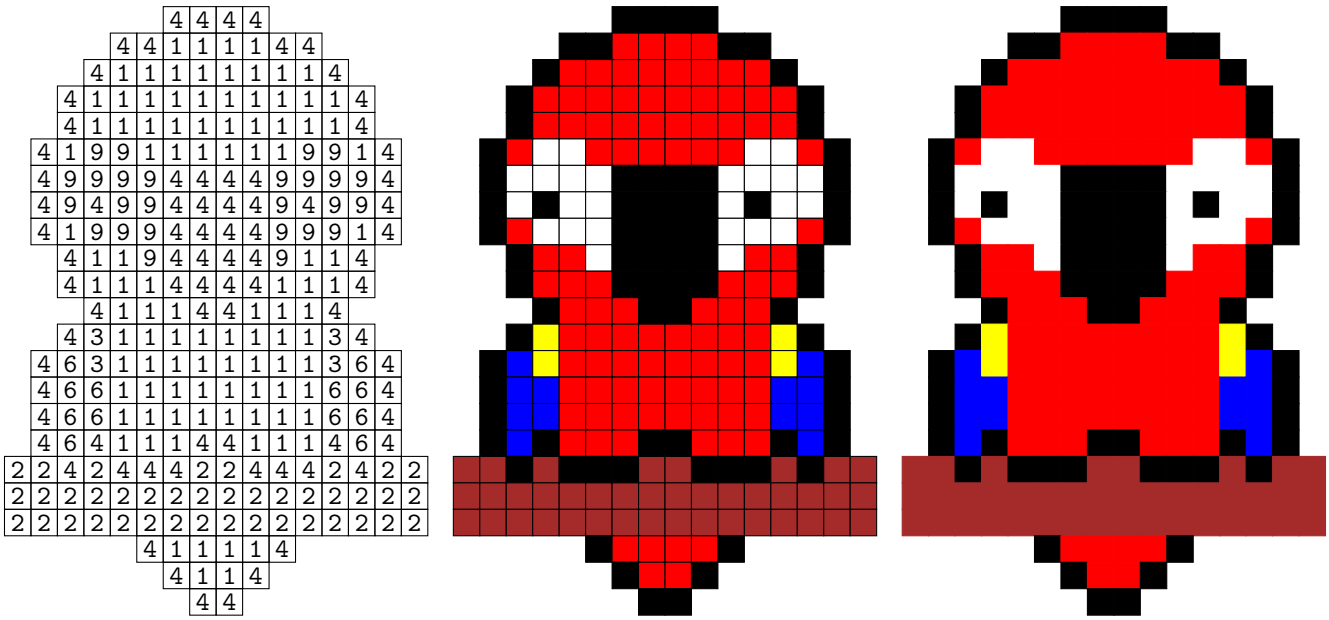
The third argument, *mandatory*, is the filename of the csv.

Code LaTeX

```
%creation of the csv
\begin{filecontents*}[overwrite]{parrot.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

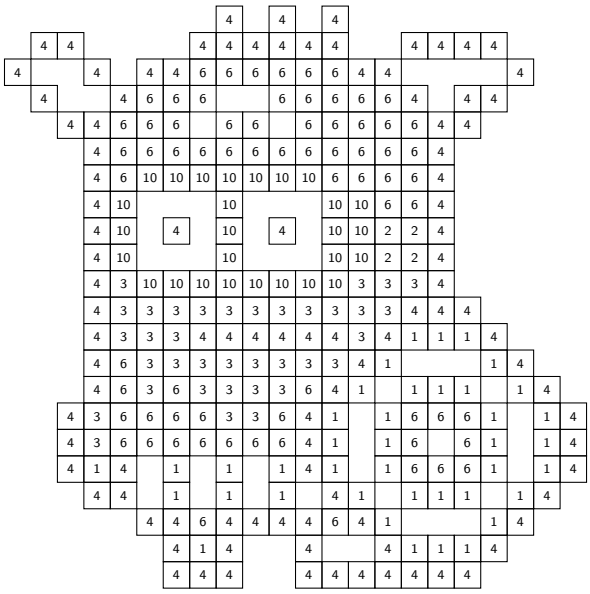
```
%simple codes
%empty case with -
\PixlArtTikz[Codes=123469,Style=\ttfamily,Unit=0.35]{parrot.csv}
~~
\PixlArtTikz[Codes=123469,Colors={Red,Brown,Yellow,Black,Blue,White},Correction,Unit=0.35]{parrot.csv}
~~
\PixlArtTikz[Codes=123469,Colors={Red,Brown,Yellow,Black,Blue,White},Correction,Unit=0.35,Border=false]{parrot.csv}
```

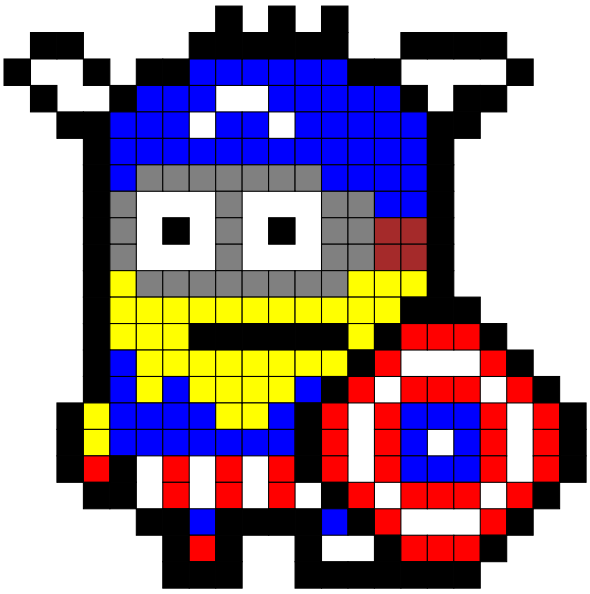


In the following example, les *symbols* to print can't be used for the *codes*, so we can use the keys ⟨**Symbols**⟩ and ⟨**Symb**⟩ to bypass this limitation.

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,-,-,-
    -,-,-,D,F,F,F,F,F,F,F,F,F,F,F,F,F,D,-,-,-,-
    -,-,-,D,F,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,C,C,C,D,-,-,-,-,-
    -,-,-,D,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,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}
```





2.3 Starred macro

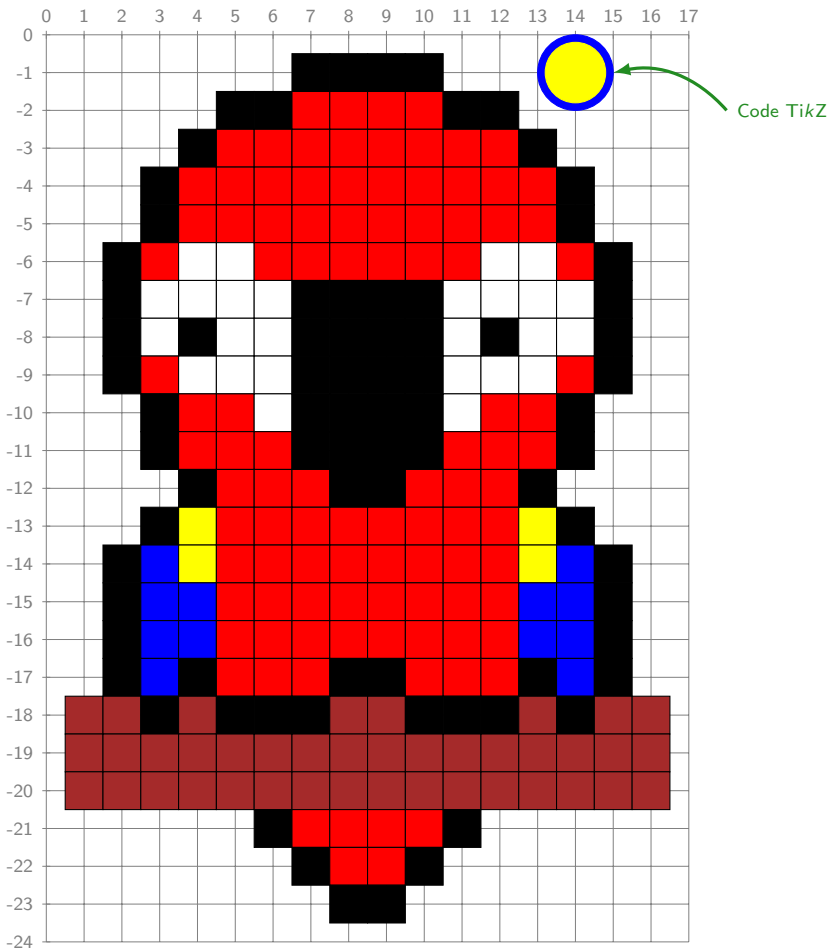
The starred *étoilée* macro `\PixlArtTikz*` is to be integrated within an environment already created. It cas be usefull to add code after the PixelArt.

In this case :

- the *optional* between `<...>` is useless ;
- the key ⟨**Unit**⟩ is useless too (uints 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]{parrot.csv}
        %added code
        \filldraw[Blue] (14,-1) circle[radius=1] ;
        \filldraw[Yellow] (14,-1) circle[radius=0.8] ;
        \draw[ForestGreen,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}
```



3 PixelArt environment

3.1 Usage

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

- The environment is created within Ti*k*Z and added code is to give in Ti*k*Z !
- The added code will be print "above" 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 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 specifythe style of the text. default \scriptsize

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

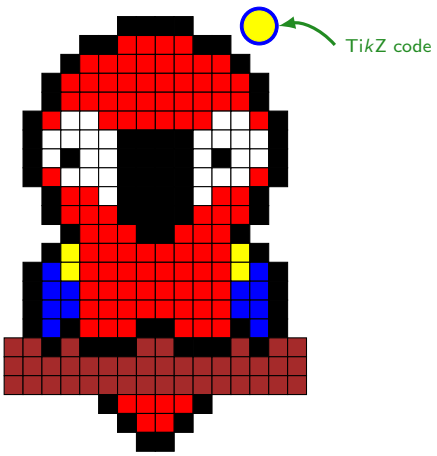
The third argument, *mandatory*, is the filename of the csv.

3.2 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

```
\begin{center}
    \begin{EnvPixlArtTikz}%
            [Codes=123469,Colors={Red,Brown,Yellow,Black,Blue,White},Correction,Unit=0.25]
            {parrot.csv}
        \filldraw[Blue] (14,-1) circle[radius=1] ;
        \filldraw[Yellow] (14,-1) circle[radius=0.8] ;
        \draw[ForestGreen,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

`v0.1.0 :` Initial version