PixelArtTikz [en]

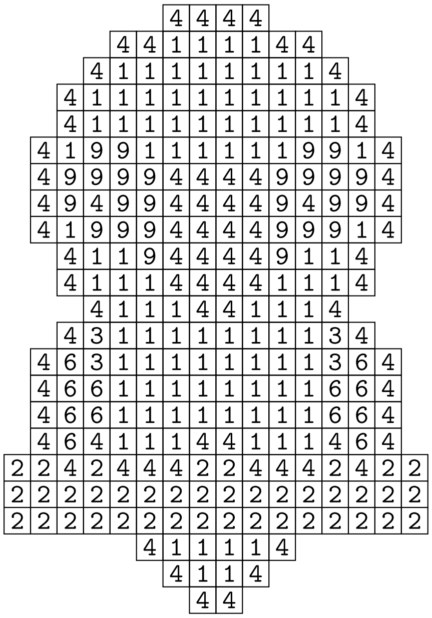
PixelArts, with Ti*k*Z,

with solution and colors.

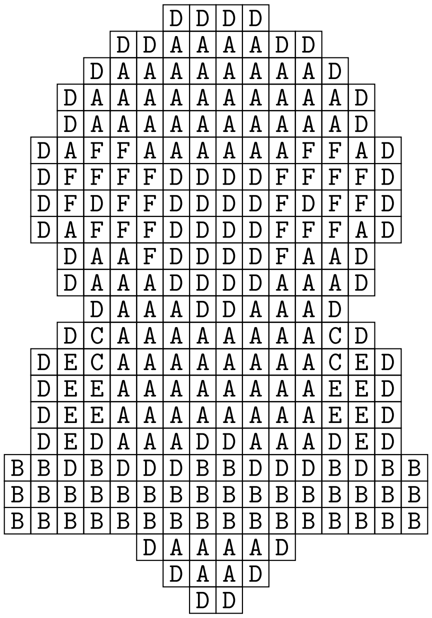
Version 0.1.4 - 16/04/2024

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

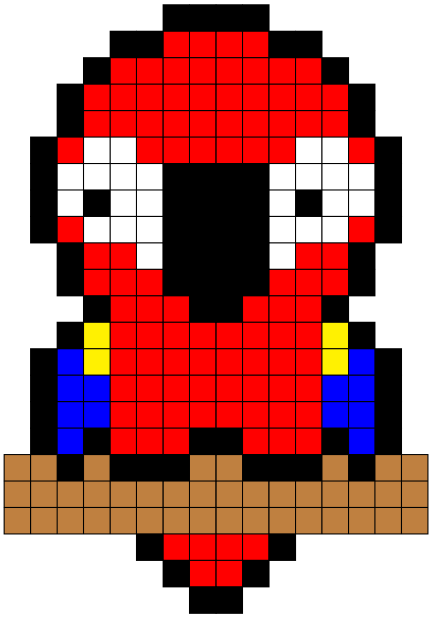
- Commands to display PixelArts.
- Environment to complete the PixelArt.



```
            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
```



```
            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 Ti*k*Z 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 "-".

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

The option **⟨[csvii]⟩** forces the usage of the LaTeX 2$_\varepsilon$ version.

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 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.

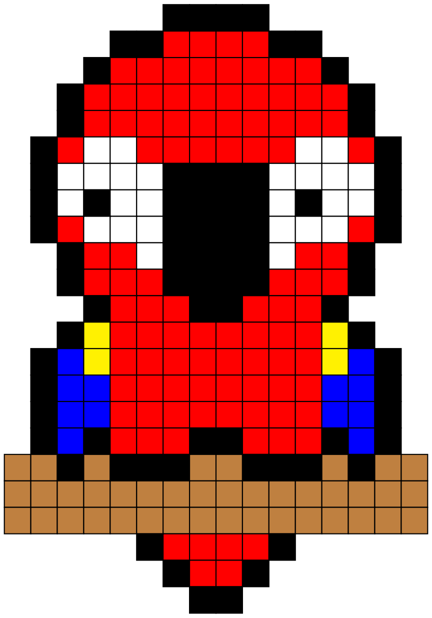
Code LaTeX

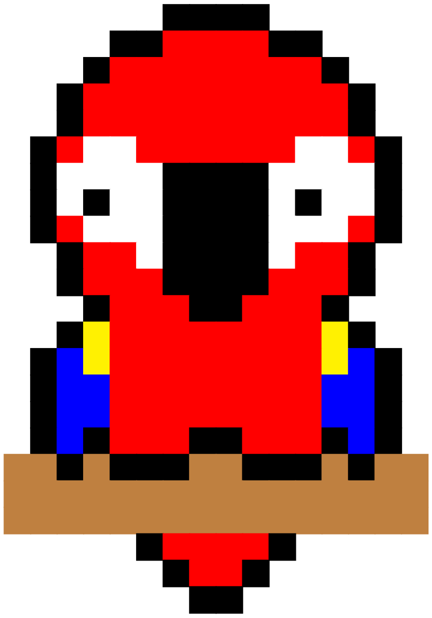
```
%creation of the csv
\begin{filecontents*}[overwrite]{test1.csv}
    -,-,-,-,-,-,4,4,4,4,-,-,-,-,-,-
    -,-,-,-,4,4,1,1,1,1,4,4,-,-,-,-
    -,-,-,4,1,1,1,1,1,1,1,1,4,-,-,-
    -,-,4,1,1,1,1,1,1,1,1,1,1,4,-,-
    -,-,4,1,1,1,1,1,1,1,1,1,1,4,-,-
    -,4,1,9,9,1,1,1,1,1,1,9,9,1,4,-
    -,4,9,9,9,9,4,4,4,4,9,9,9,9,4,-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

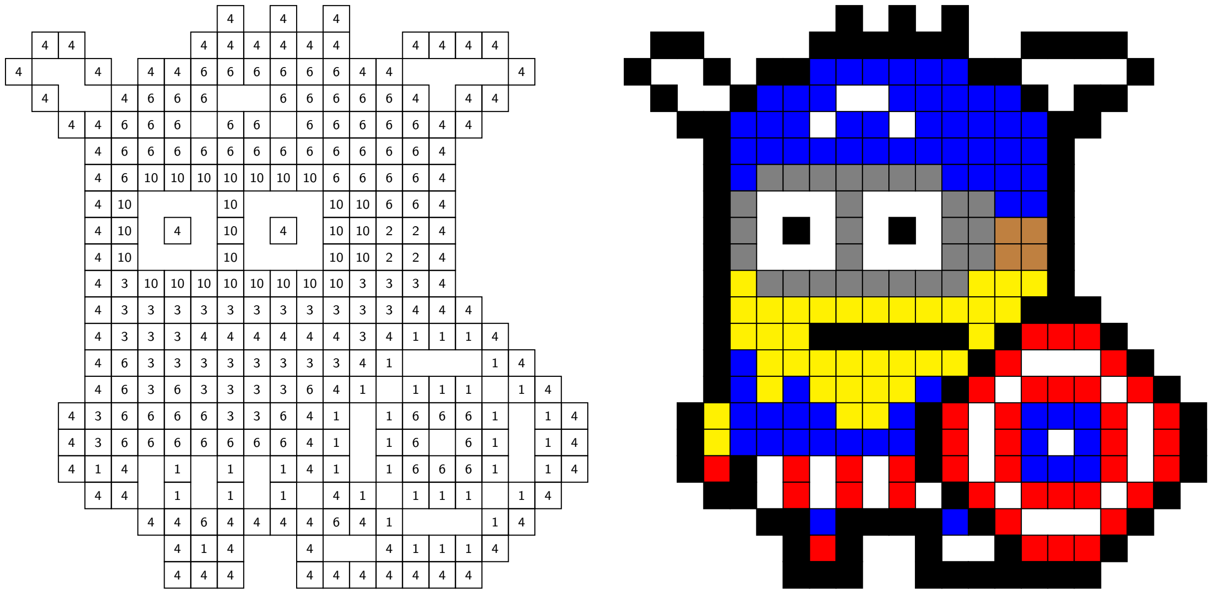




In the following example, the *symbols* to print can't be used for the *codes*, so we can use the keys ⟨**Symbols**⟩ and ⟨**Symb**⟩ to bypass this limitation.

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

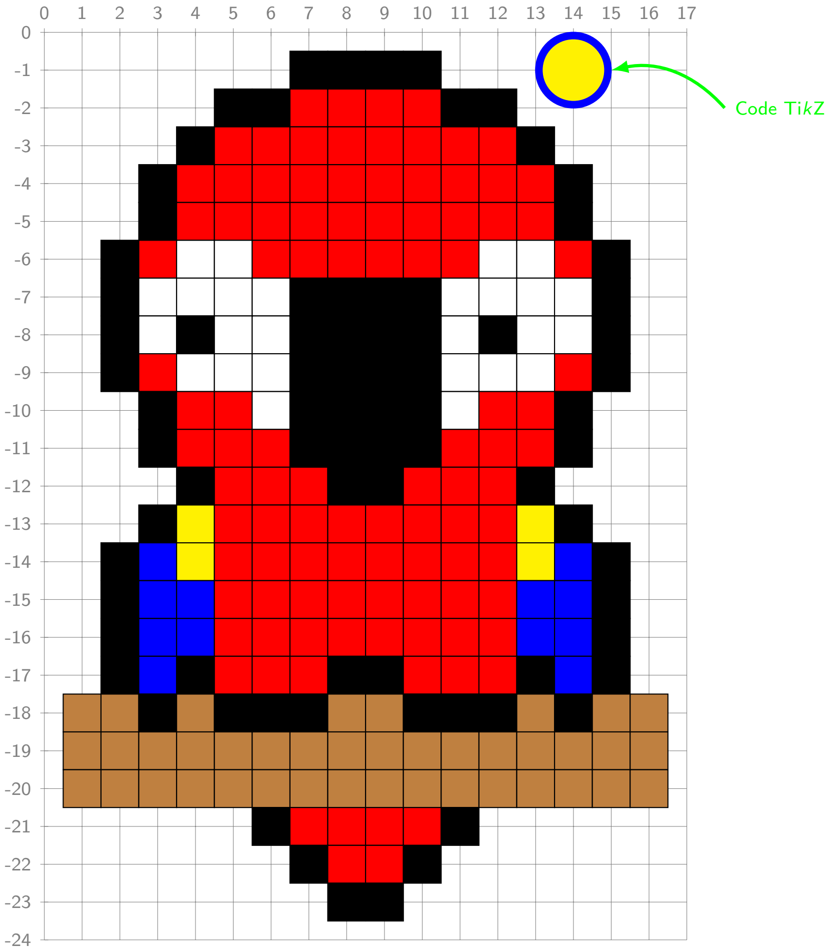
The starred macro `\PixlArtTikz*` is to be used within an already created environment. It can be useful for adding code after the PixelArt.

In this case:

- the *optional* argument between `<...>` is discarded;
- the key ⟨**Unit**⟩ is discarded 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.

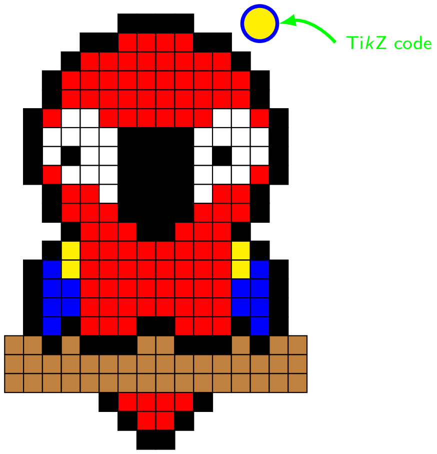
The third argument, *mandatory*, is the filename of the csv.

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}
```



5 Macro for *mini*-PixelArt

5.1 Idea

The idea is tu propose a macro to insert, without csv file, a small PixelArt with small colors list.

Code LaTeX

```
\MiniPixlArt[keys]{list of colors}
```

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

- the key ⟨**Unit**⟩ for dimension of the cells ; default 0.25em,
- the boolean ⟨**Border**⟩ to print a small border for the cells. default false

The second argument, *mandatory* and between `{...}`, is the colors of the cells :

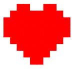
- each color is *coded* by a letter :
  - `R : red`
  - `G : green`
  - `C : blue`
  - `Y : yellow`
  - `B : black`
  - `L : gray`
  - `. : white`
  - `M : maroon`
  - `O : orange`
  - `P : purple`
- each linebreak is done by `,` ;
- the thickness of the borders are 10 % of the unit.

The last argument, *optional* and between `<...>`, proposes options for the tikz environment.

5.2 Examples

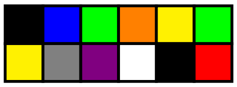
Code LaTeX

```
\MiniPixlArt{%
    ..RR..RR..,
    .RRRRRRRR.,
    RRRRRRRRRR,
    RRRRRRRRRR,
    RRRRRRRRRR,
    .RRRRRRRR.,
    ..RRRRRR..,
    ...RRRR...,
    ....RR....
}
```



Code LaTeX

```
Inline, we can give \MiniPixlArt[Unit=5mm,Border]{BCGOYG,YLP.BR}<baseline=(current bounding box.center)>
this miniPA.
```

Inline, we can give  this miniPA.

6 Macro for cutting PixelArt

6.1 Idea

The idea is to offer commands to create *collaborative* PixelArts, to form a *large* image from several small ones (of the same size).

💣 For practical reasons (related to the use of `filecontents`) the file `csv` must end with an empty line (it is created automatically via `filecontents`, and the code uses this specificity).

The available ⟨**keys**⟩ are exactly the same as those in classic displays.

6.2 Global usage

The argument `cutting` can be given in the form :

- ⟨**<number of block length>x<number of block col>**⟩ to specify the size of the blocks;
- ⟨**<number of V blocks>+<number of H blocks>**⟩ to specify the number of blocks.

Code LaTeX

```
\CutPixlArtTikz(*)[keys]<tikz options>{file.csv}{cutting}
%(*) := change layout formatting (A1 or 1.1)
%1   := keys
%2   := tikz options
%3   := csv file
%4   := block size (LxC) or block number (L+C)

\HelpGridPixelArtTikz(*)[scale]{file.csv}{cutting}
%(*) := change layout formatting (A1 or 1.1)
%1   := scale of notice
%2   := csv file
%4   := block size (LxC) or block number (L+C)
```

6.3 Example

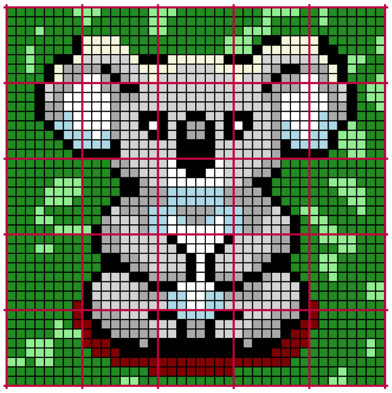
Code LaTeX

```
\def\listcoulkoala%
    {VertForet,Noir,GrisClair,GrisFonce,Beige,Blanc,BleuClair,VertClair,Marron}
```

Code LaTeX

```
%block with 8x8 size
\HelpGridPixelArtTikz{PAkoala.csv}{8x8}
~~
\PixlArtTikz[Correction,Unit=0.125,Codes=ABCDEFGHI,Colors={\listcoulkoala},Grid=8x8]{PAkoala.csv}
```

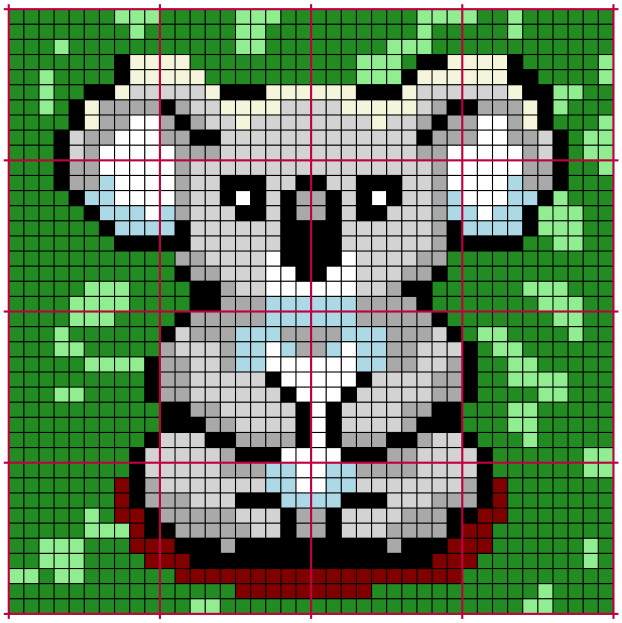
A1	A2	A3	A4	A5
B1	B2	B3	B4	B5
C1	C2	C3	C4	C5
D1	D2	D3	D4	D5
E1	E2	E3	E4	E5



Code LaTeX

```
%16 blocks (4L 4C)
\HelpGridPixelArtTikz*[2]{PAkoala.csv}{4+4}
~~
\PixlArtTikz[Correction,Unit=0.2,Codes=ABCDEFGHI,Colors={\listcoulkoala},Grid=4+4]{PAkoala.csv}
```

1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4
3.1	3.2	3.3	3.4
4.1	4.2	4.3	4.4



Code LaTeX

```
%block with 8x8 size
\CutPixlArtTikz[Unit=0.3,Codes=ABCDEFGHI]{PAkoala.csv}{8x8}
```

Sample A1

A	A	A	A	A	A	A	H
A	A	A	A	A	A	A	A
A	A	A	H	A	A	A	A
A	A	A	A	A	A	A	B
A	A	H	A	A	A	A	B
A	A	H	A	A	B	B	E
A	A	H	A	B	E	D	D
A	A	A	A	B	E	D	D

Sample A2

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

Sample A3

H	H	A	A	A	A	A	A
H	A	A	A	A	A	A	A
H	A	A	A	A	A	A	A
A	A	A	A	A	A	A	H
A	A	A	A	A	A	A	H
B	E	E	E	E	E	E	B
E	E	C	C	C	C	E	E
C	C	C	C	C	C	C	C

Sample A4

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

Sample A5

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

Sample B1

A	A	A	B	C	D	D	F
A	A	A	B	C	D	F	F
A	A	A	B	D	D	F	F
A	A	A	B	D	D	G	F
A	A	A	A	B	G	G	F
A	A	A	A	A	B	G	G
A	A	A	A	A	A	B	G
A	A	A	A	A	A	A	B

Sample B2

F	F	D	D	B	B	C	C
F	F	F	D	D	D	C	C
F	F	F	D	C	C	C	C
F	F	F	D	C	C	B	B
F	F	F	D	C	C	B	F
G	F	G	D	C	C	C	B
G	G	G	B	C	C	C	C
B	B	B	B	C	C	C	C

Sample B3

C	C	C	C	C	C	C	C
C	C	C	C	C	C	C	C
C	C	C	C	C	C	C	C
B	C	C	B	B	C	C	B
B	C	B	D	D	B	C	B
B	C	B	D	D	B	C	B
C	C	B	B	B	B	C	C
C	C	B	B	B	B	C	C

Sample B4

C	C	C	B	D	D	D	F
C	C	C	D	D	F	F	F
C	C	C	C	D	F	F	F
B	B	C	C	D	F	F	F
F	B	C	C	D	G	F	F
B	C	C	C	D	G	G	F
C	C	C	C	D	B	G	G
C	C	C	C	D	B	B	B

Sample B5

F	D	D	C	B	A	H	H
F	F	D	C	B	A	H	H
F	F	D	D	B	A	A	H
F	G	D	D	B	A	A	A
F	G	G	B	H	A	A	A
G	G	B	H	H	H	A	A
G	G	B	H	H	H	A	A
B	B	A	A	H	H	A	A

Sample C1

A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	H	H	H
A	A	A	A	H	H	H	H
A	A	A	A	H	H	H	A
A	A	A	H	A	A	A	A
A	A	A	H	H	A	A	A
A	A	A	A	A	H	H	H

Sample C2

A	A	A	D	D	C	C	C
A	A	A	B	D	D	C	C
A	A	A	A	B	B	D	C
A	A	A	A	B	B	D	D
A	A	A	B	D	D	D	D
A	A	B	C	D	D	D	G
A	A	D	C	C	C	D	G
H	B	D	D	C	C	D	G

Sample C3

C	F	B	B	B	B	F	C
C	F	F	B	B	F	F	C
C	F	F	F	F	F	F	C
D	G	G	G	G	G	G	D
D	G	G	G	G	G	G	D
G	G	D	D	D	D	G	G
G	F	G	D	D	G	F	G
G	F	F	F	F	F	F	G

Sample C4

C	C	C	D	D	A	A	A
C	C	D	D	B	A	A	A
C	D	B	B	A	A	A	A
D	D	D	B	A	A	A	A
D	D	D	D	B	A	A	A
G	D	D	D	C	B	A	H
G	D	D	C	C	C	B	A
G	D	D	C	C	D	B	A

Sample C5

A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	H	H	H	A	A	A
A	A	A	H	H	H	A	A
A	A	A	A	H	H	A	A
H	A	A	A	A	H	A	A
H	H	A	A	A	A	A	A
A	H	H	A	A	A	A	A

Sample D1

A	A	A	A	A	A	A	A
A	A	A	H	H	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	I

Sample D2

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

Sample D3

C	B	F	F	F	F	F	B
C	C	B	F	F	F	B	C
C	C	C	B	F	B	C	C
C	C	C	B	F	B	C	C
D	D	D	B	F	B	D	D
B	B	F	F	F	F	B	B
D	G	G	F	F	G	G	D
C	G	G	F	F	G	G	C

Sample D4

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

Sample D5

A	H	H	H	H	A	A	A
A	A	A	H	H	A	A	A
A	H	H	A	A	A	A	A
A	H	H	A	A	A	A	A
A	A	H	A	A	A	A	A
A	A	A	A	A	A	H	H
A	A	A	A	A	A	H	H
I	A	A	A	A	A	A	A

Sample E1

A	A	A	A	A	A	A	I
A	A	A	A	A	H	A	I
A	A	A	A	A	H	H	H
A	A	H	H	H	A	A	A
A	A	H	H	H	A	A	A
H	H	A	H	H	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A

Sample E2

B	D	D	D	C	C	C	B
I	B	D	D	D	D	D	C
I	B	B	D	D	D	D	D
I	I	I	B	B	B	D	B
A	I	I	I	B	B	B	B
A	A	A	I	I	I	I	I
A	A	A	A	A	A	A	I
A	A	A	A	H	H	A	A

Sample E3

B	B	G	G	G	G	B	B
C	C	B	D	D	B	C	C
C	C	B	D	D	B	C	C
B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I
A	A	A	A	A	A	A	A

Sample E4

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

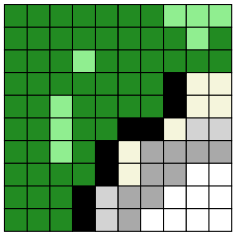
Sample E5

I	A	A	A	A	A	A	A
I	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A
A	A	A	A	A	A	H	A
A	A	A	A	A	A	H	A
A	A	A	A	A	A	A	A
A	A	A	H	A	A	A	A
A	A	H	H	A	A	A	A

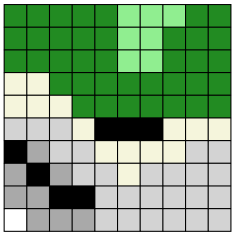
Code LaTeX

```
%16 blocks (4L 4C)
\CutPixlArtTikz*[Unit=0.3,Codes=ABCDEFGHI,Correction,Colors={\listcoulkoala}]{PAkoala.csv}{4+4}
```

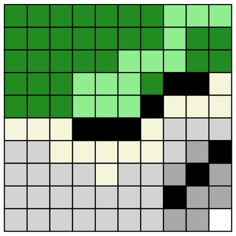
Sample 1.1



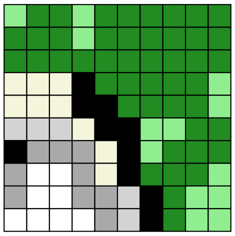
Sample 1.2



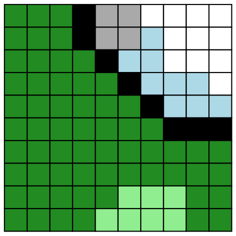
Sample 1.3



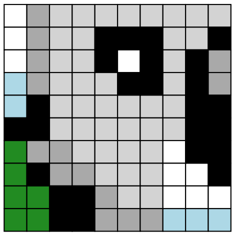
Sample 1.4



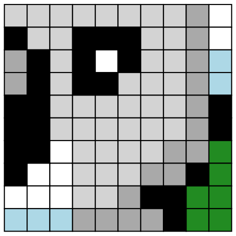
Sample 2.1



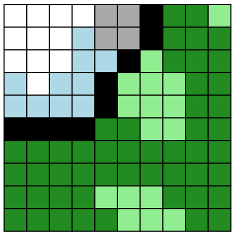
Sample 2.2



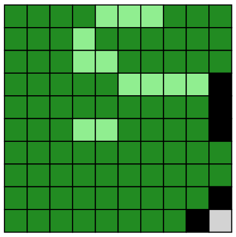
Sample 2.3



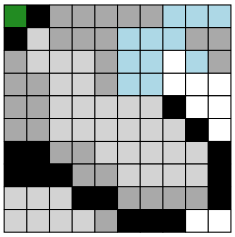
Sample 2.4



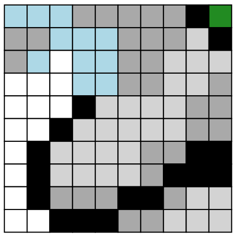
Sample 3.1



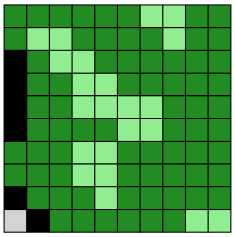
Sample 3.2



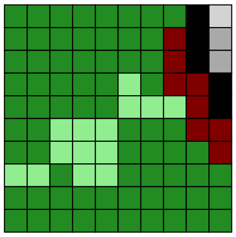
Sample 3.3



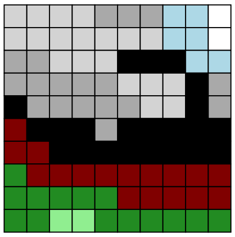
Sample 3.4



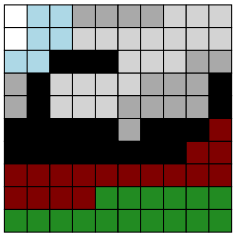
Sample 4.1



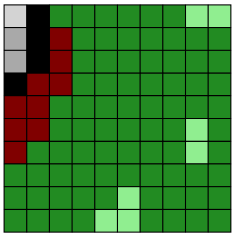
Sample 4.2



Sample 4.3



Sample 4.4



7 Anamorphic cylinder PixelArt

7.1 Idea

The idea is to provide something to create PixelArt with the aim of using a cylindrical anamorphosis.
On https://www.youtube.com/watch?v=PT8KUozBg3I, there is a video *demonstration*, proposed by Jean-Yves Labouche.
The overall operation is similar to that of the *main* command, however there are some adjustments:

- the possibility of giving the file csv in *normal* or *inverted* mode;
- the dimensions (width & middle) must be specified to produce the PixelArt;
- the command is standalone (for the moment) so no addition(s) later.

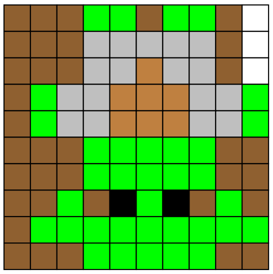
Code LaTeX

```
\PixlArtTikzCylindric[keys]{file.csv}
```

The files illustrating this paragraph are given below.

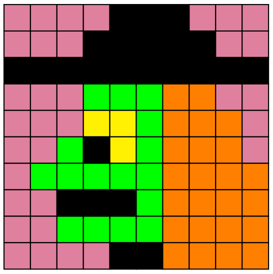
Code LaTeX

```
%version with inverted data
\begin{filecontents*}[overwrite]{PAyoda.csv}
E,E,E,A,A,E,A,A,E,D
E,E,E,F,F,F,F,F,E,D
E,E,E,F,F,C,F,F,E,D
E,A,F,F,C,C,C,F,F,A
E,A,F,F,C,C,C,F,F,A
E,E,E,A,A,A,A,A,E,E
E,E,E,A,A,A,A,A,E,E
E,E,A,E,B,A,B,E,A,E
E,A,A,A,A,A,A,A,A,A
E,E,E,A,A,A,A,A,E,E
\end{filecontents*}

\PixlArtTikz[%
    Codes=ABCDEF,
    Colors={green,black,brown,white,brown!75!black,lightgray},
    Correction,Unit=0.35]%
    {PAyoda.csv}
```



Code LaTeX

```
%version with normal data
\begin{filecontents*}{PAsorciere.csv}
V,V,V,V,N,N,N,V,V,V
V,V,V,N,N,N,N,N,V,V
N,N,N,N,N,N,N,N,N,N
V,V,V,G,G,G,O,O,V,V
V,V,V,J,J,G,O,O,O,V
V,V,G,N,J,G,O,O,O,V
V,G,G,G,G,G,O,O,O,O
V,V,N,N,N,G,O,O,O,O
V,V,G,G,G,G,O,O,O,O
V,V,V,V,N,N,O,O,O,O
\end{filecontents*}

\PixlArtTikz[%
    Codes=VNGOJ,
    Colors={purple!50,black,green,orange,yellow},
    Correction,Unit=0.35]%
    {PAsorciere.csv}
```



7.2 Keys and options

The first argument, *optional* and between `[...]` offers Keys necessary for the command to function properly:

- the key ⟨**Width**⟩ which defines the width (radius in cm) of the rendering; default 6
- the key ⟨**Center**⟩ which defines the width (radius in cm) of the *middle*; default 1.25
- the key ⟨**Codes**⟩ contains the *string* of the *simple* codes of the file csv;
- the key ⟨**Colors**⟩ which contains the *list* of associated colors;
- the key ⟨**Symbols**⟩ which contains the *possible list* of alternative characters to display in the boxes;
- the key ⟨**Style**⟩ which allows you to specify the style of characters. default normalsize
- the Boolean key ⟨**Correction**⟩ which allows you to color the PixelArt; default false
- the Boolean key ⟨**Symb**⟩ which allows you to display the *alternative* characters; default false
- the Boolean key ⟨**Solution**⟩ which displays the solution (with *mirror* effect); default false
- the Boolean key ⟨**Swap**⟩ which allows you to specify the data type (⟨**true**⟩ := normal; ⟨**false**⟩ := inverted). default false

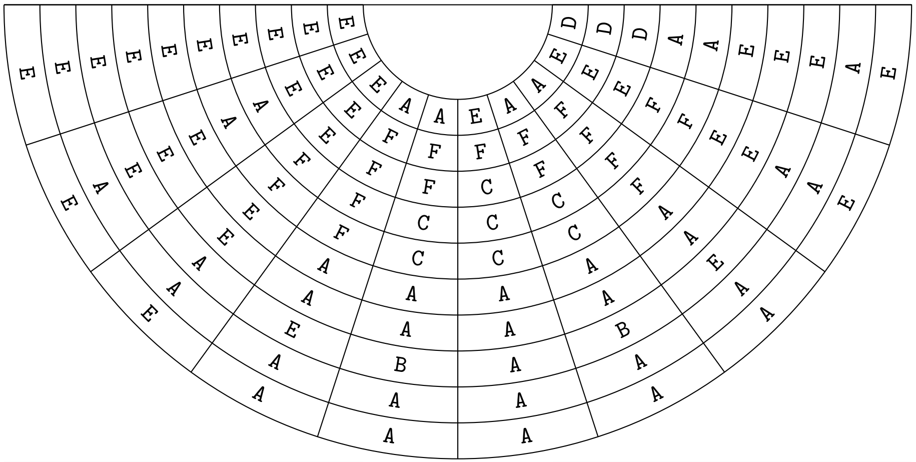
The second argument, *mandatory*, is the name of the csv file to use.

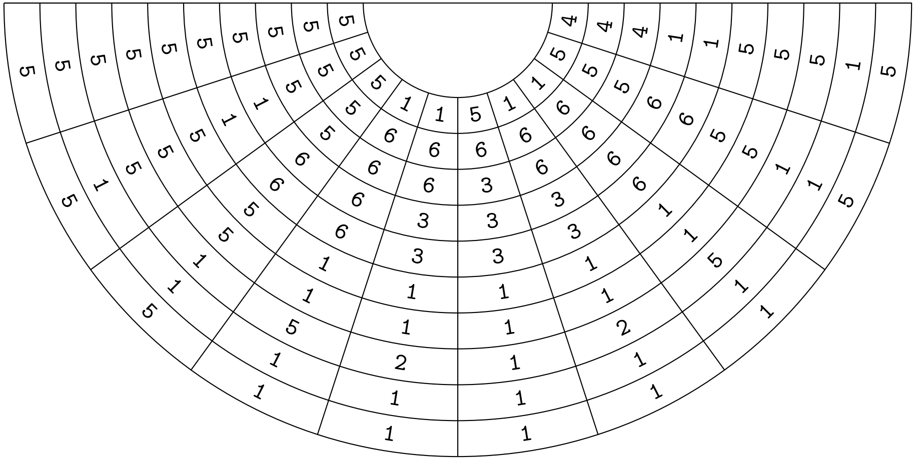
7.3 Example with inverted data (Yoda)

In this paragraph, we use the data `PAyoda`, which corresponds to the *inverted* layout, so the key **⟨Swap⟩** is not necessary.

Code LaTeX

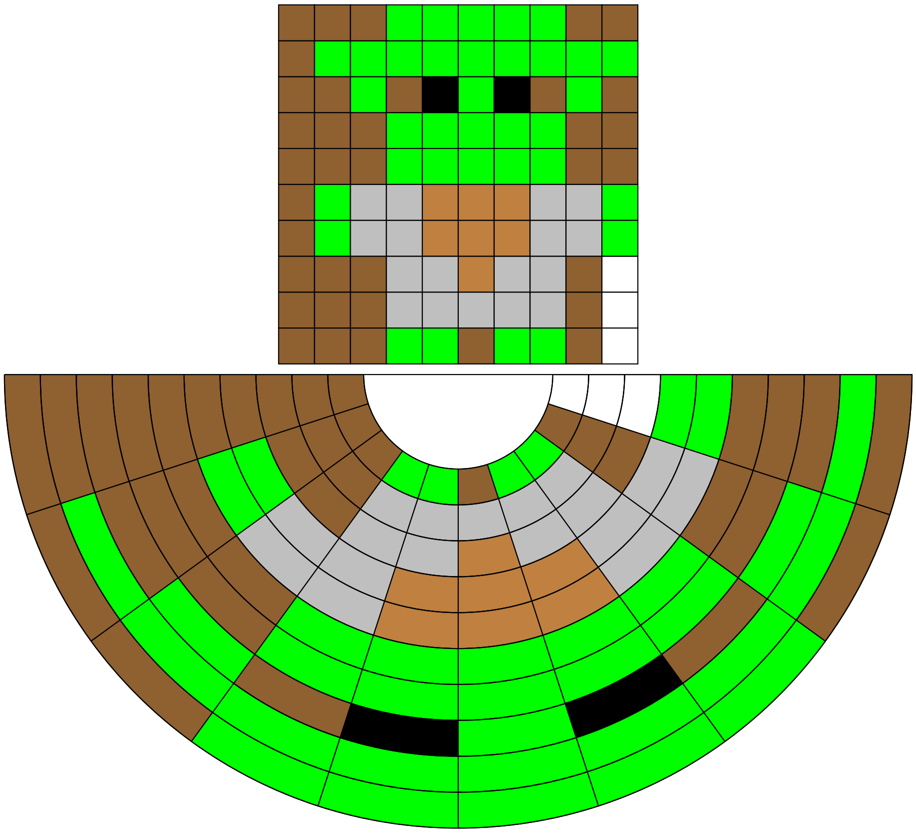
```
%classic version
\PixlArtTikzCylindric[Codes=ABCDEF,Style=\small\ttfamily]{PAyoda.csv}

%version with 'symbols'
\PixlArtTikzCylindric[Codes=ABCDEF,Symbols={1,2,3,4,5,6},Symb,Style=\small\ttfamily]{PAyoda.csv}
```





Code LaTeX

```
%Fix and solution
\begin{tabular}{c}
    \PixlArtTikzCylindric[%
        Codes=ABCDEF,
        Colors={green,black,brown,white,brown!75!black,lightgray},
        Solution]%
        {PAyoda.csv}
    \\
    \PixlArtTikzCylindric[%
        Codes=ABCDEF,
        Colors={green,black,brown,white,brown!75!black,lightgray},
        Correction]%
        {PAyoda.csv}
\end{tabular}
```

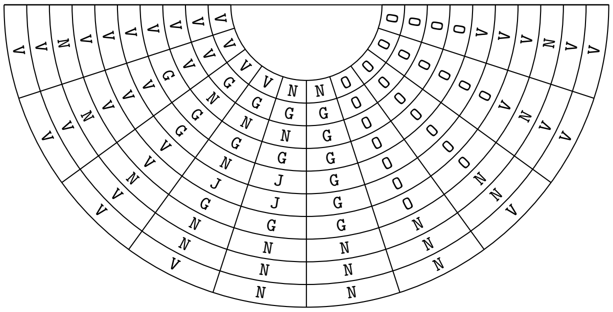


7.4 Example with classic data (Witch)

In this paragraph, we use the data `PAsorciere`, which corresponds to the *normal* layout, so the key ⟨**Swap**⟩ is necessary.

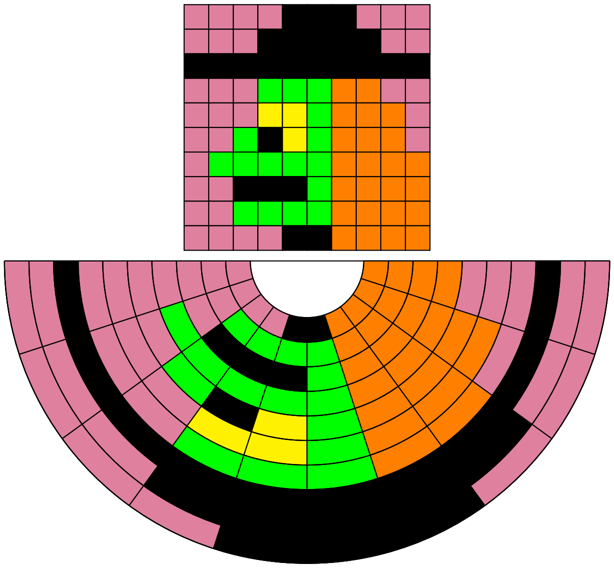
Code LATEX

```
%classic version
\PixlArtTikzCylindric[%
Width=4,Center=1,Codes=VNGOJ,
Colors={purple!50,black,green,orange,yellow},
Swap,Style=\ttfamily\scriptsize]%
{PAsorciere.csv}
```



Code LATEX

```
%Correction and solution
\begin{tabular}{c}
    \PixlArtTikzCylindric[%
        Width=4,Center=0.75,Codes=VNGOJ,
        Colors={purple!50,black,green,orange,yellow},
        Swap,Solution]%
        {PAsorciere.csv}
    \\
    \PixlArtTikzCylindric[%
        Width=4,Center=0.75,Codes=VNGOJ,
        Colors={purple!50,black,green,orange,yellow},
        Swap,Correction]%
        {PAsorciere.csv}
\end{tabular}
```



Part III
History

`v0.1.4` :	Anamorphic cylinder PixelArts
`v0.1.3` :	Cut PixelArts within several PixelArts
`v0.1.2` :	*mini*-PixelArts
`v0.1.1` :	Bugfix with color
`v0.1.0` :	Initial version