

frogFetch Version 0.1.0 Tutorial

Welcome to frogFetch! A terminal program with tools designed for terminal customization.
This initial release will only contain one tool: the “Pixel to ASCII” tool.
The primary goal of this tool set is simplicity and user-friendliness.
This should be considered a “testing release” and not the final product. Many more features and tools
still need to be implemented.

First:

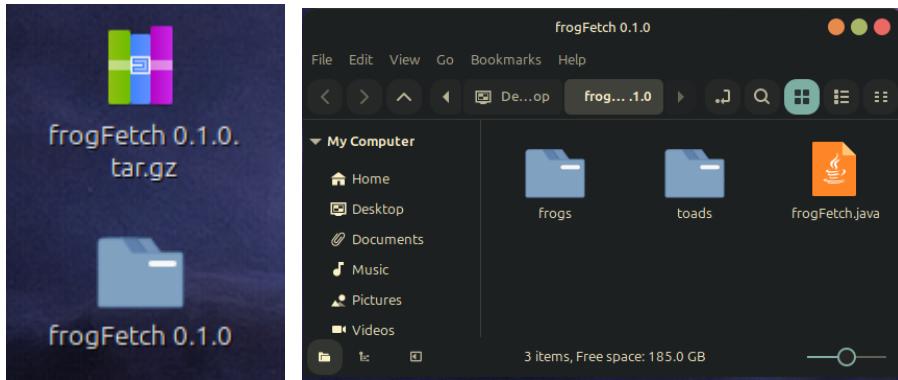
This program is written in Java. In order to run it, you will need to install the Java Development Kit (JDK). This can be done in the terminal.

For example, on **Debian** based systems:

Update your system by running “`sudo apt update`”
Then install the JDK by running “`sudo apt install default-jdk`”
Now we’re ready to begin using **frogFetch**!

Overview

Lets start by looking at the files.



Start by extracting the compressed folder and then open it.

Inside you will find the **frogFetch** java file, a folder named **frogs**, and a folder named **toads**.



Shockingly, inside the **frogs** folder you will find some **frogs**.

This folder is where you will place any pixel art you want to convert.

You’ll also find that the **toads** folder is currently empty. Good.

Getting started

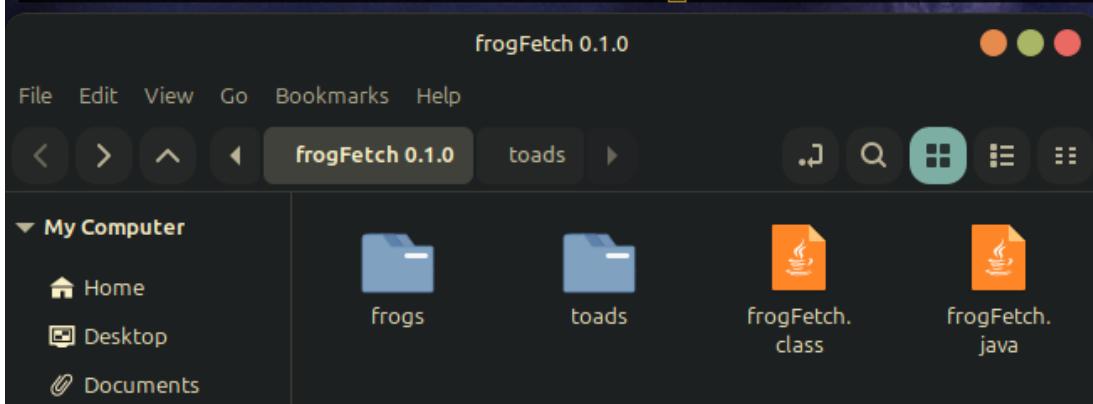
The only step left is to compile the program before we can use it.

Open up a terminal and navigate to the `frogFetch` folder. (Or open the folder, right click inside of it and select “open in terminal”).

To compile the program, run the command: `javac frogFetch.java`

This will create a `frogFetch.class` file.

```
daniel@Venusaur:~/Desktop/frogFetch 0.1.0$ javac frogFetch.java  
daniel@Venusaur:~/Desktop/frogFetch 0.1.0$
```



Running the program

Now, run the command: `java frogFetch` to start the program.

The program will prompt you to enter the name of a png image.

Let's type in “frog”



Ta-da! The ASCII art is now generated. It should display on your terminal with a “Gap size”.

Please enter the name of the png image. (Example: "frog" not "frog.png") (image must be in the "frogs" folder)

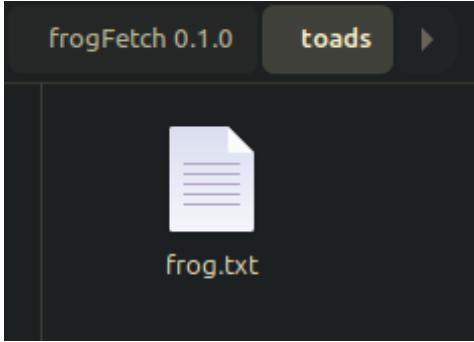
frog

Gap size = -394

Keep the Gap size number in mind. We will need it later.

Now would be a good time to check the `toads` folder.

Inside you will find a brand new text file. This is the ASCII art formatted in **BASH** script which should work on any **Linux** distribution, **Mac**, and **Windows** (if you set it up through WSL or whatever).



A screenshot of a Linux desktop environment. At the top, there is a horizontal menu bar with the following items: File, Edit, View, Search, Tools, Documents, and Help. Below the menu bar is a toolbar with several icons: a document icon, a search icon, a magnifying glass icon, and a refresh/circular arrow icon. The main window shows a file named "frogs.txt" with a small thumbnail preview. In the bottom right corner of the window, there is a close button (an 'X').

Neofetch time!

frogFetch aims to stand on its own and not rely on any other program.

However, we are not there yet.

Lets see how to add our ASCII art to our neofetch.

Note: this program has only been tested using the **Gnome terminal** and **Neofetch**.

It **should** work on most terminals and with other similar terminal programs like fastFetch, but I have not had time to try it. The method of modifying the ASCII art may be different for fastFetch **idk**.

Anyway, lets navigate to the neofetch config file. (`./config/neofetch/config.conf`).

Open this up with your favorite text editor.

Navigate to the `image_source="auto"` line. (Ctrl +F is your friend).

```
##----- Backend Settings

# Image backend.
#
# Default: 'ascii'
# Values: 'ascii', 'caca', 'chafa', 'jp2a', 'iterm2', 'off',
#          'pot', 'termix', 'pixterm', 'tycat', 'w3m', 'kitty'
# Flag:    --backend
image_backend="ascii"

# Image Source
#
# Which image or ascii file to display.
#
# Default: 'auto'
# Values: 'auto', 'ascii', 'wallpaper', '/path/to/img', '/path/to/ascii', '/path/to/dir/'
#          'command output (neofetch --ascii "$(fortune | cowsay -W 30)")'
# Flag:    --source
#
# NOTE: 'auto' will pick the best image source for whatever image backend is used.
#       In ascii mode, distro ascii art will be used and in an image mode, your
#       wallpaper will be used.
image_source="auto"
```

Change this to the file path of the ASCII art .txt file.

```
image_source="/home/daniel/frogFetch 0.1.0/toads/frog.txt"
```

Note: You don't have to keep it in the **toads** folder. You can keep your ASCII art anywhere you'd like. However, I recommend keeping the files in the **toads** folder as this will be more efficient for quick changes to the ASCII art in your neofetch.

Be sure to save your changes before closing the config file.

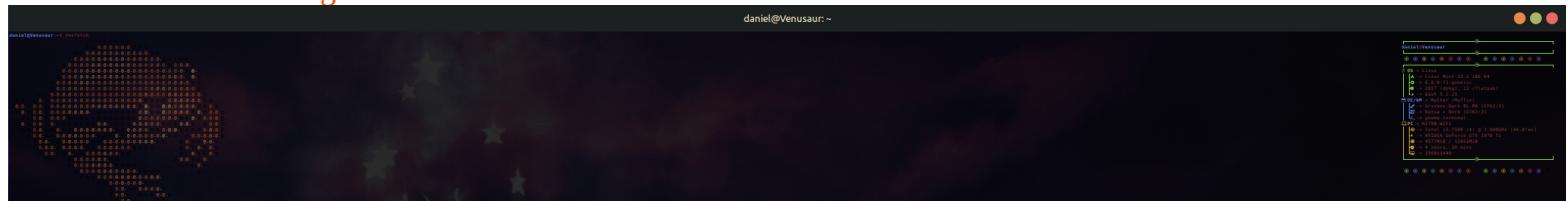
Now lets run neofetch to test it out:

```
daniel@Venusaur:~$ neofetch
```



It works! But where is all the info? Did the **tomato frog eat it?**

No. **Tomato frogs** are insectivores.



The info is still there, just **a little** far away.

How do we fix this?

Keep the Gap size number in mind. We will need it later.

Lets go back to that config file. Navigate to the **gap** section. Let's change this to **-394**. (The **gap** number is not stored anywhere. It is only shown in the terminal when the ASCII art is generated).

```
# Gap between image and text
#
# Default: '3'
# Values:  'num' , '-num'
# Flag:    --gap
gap=3
```

```
# Gap between image and text
#
# Default: '3'
# Values:  'num' , '-num'
# Flag:    --gap
gap=-394
```

daniel@Venusaur:~\$

Much better.
By default this gap size will place your info 3 spaces to the right of the rightmost “pixel.” This is the default behavior of neofetch

But be creative here!

You can use this to put the text to the left of your ASCII (You'll have to leave empty space in the pixel art to make room)

Or you can have 2 ASCII art images and have the info sit in between them.

How to format images for frogFetch

Image must be a png.

The background color must be #c8c8c8 (200,200,200 in RGB).

That's it.



Frog2.png (found in frogs folder) has a transparent background

That won't work properly:



```
Please enter the name of the png image. (Example: "frog" not "frog.png") (Image must be in the "frogs" folder.)  
frog2  
A 10x10 grid of colored dots representing a frog's body. The dots are arranged in a roughly triangular shape, with a concentration of blue and cyan dots forming the main body and a cluster of yellow and orange dots forming the head and front legs. The background is black.
```

What can I use to edit pixel art?

I use **Aseprite** on PC and **Pixel Studio** on my phone.

Aseprite is a powerful tool which you can get on Steam. It works on **Linux/Mac/Windows**.

Pixel Studio is on **Android/IOS/Linux/Mac/Windows**. This is a better tool to use if you are not experienced with pixel art.

You can also just use MS Paint lol.

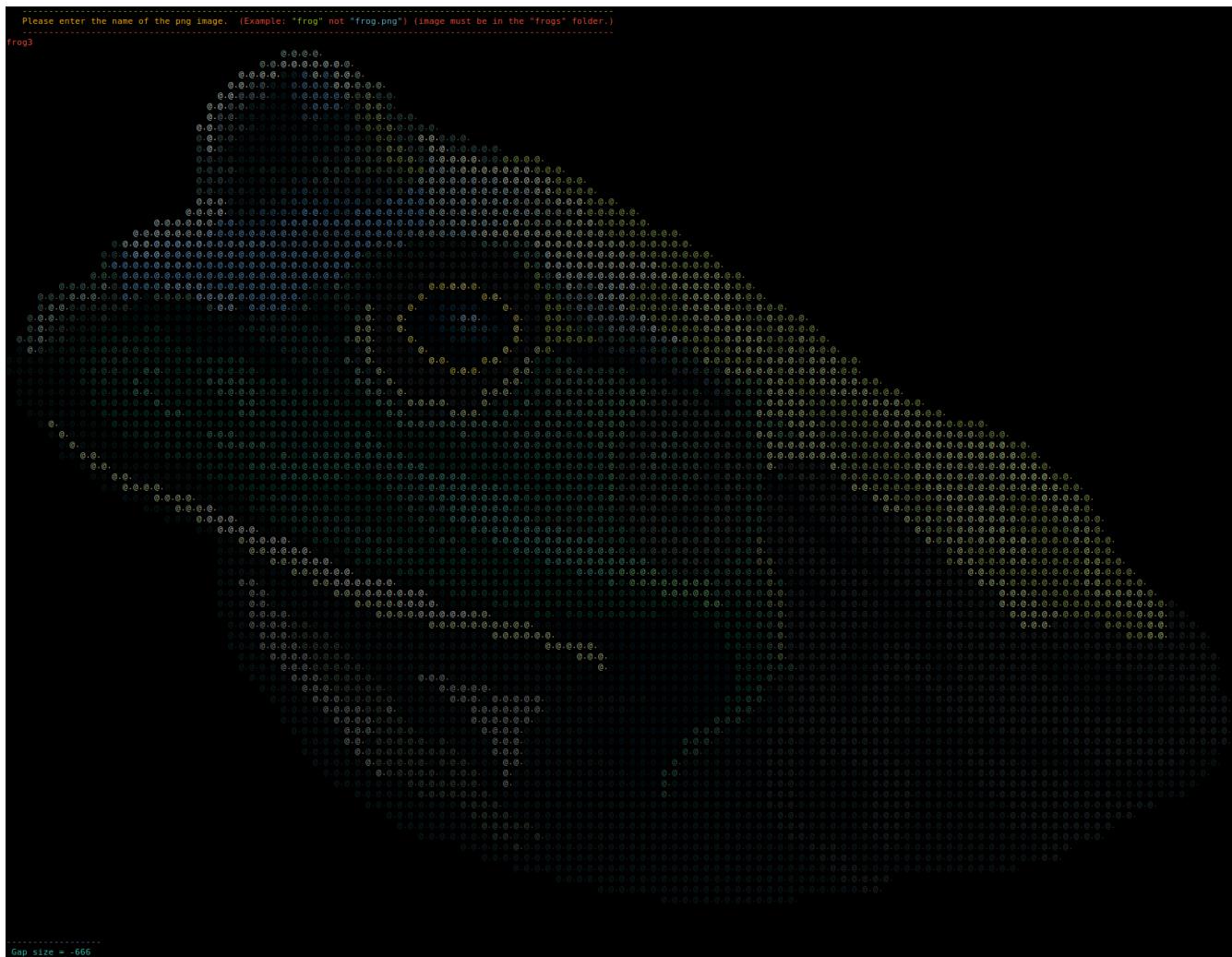
Is there a size limit?

There is no size limit (**I think**)

Frog3.png is huge!

Doesn't matter, it will work.

This program is intended for pixel art, but if you need large ASCII artwork, even if the png is not pixel art, it should work too.



Yay!

What's next for frogFetch?

I have a lot planned for the future of this project, but I am doing this as a hobby while working full time. Having recently graduated, I will also be spending time looking for a **better** job. So development for **frogFetch** will be slow and steady.

If this is version 0.1.0, what will version 1.0.0 look like?

That partially depends on you.

I need to hear feedback for how well (or not) the program works.

I would like the Pixel to ASCII tool to have a few more features, while keeping things very simple of course.

Also, I would like to have at least 2 or 3 more tools implemented before calling it version 1.0.0.

Lastly, I want to properly package this program by then too (apt install/dnf install, etc). No clue how to do that, but I'll figure it out.

What other tools are planned?

I won't go into too much detail because I don't want people to be waiting for a feature/tool that I may not be able to deliver, or that may take a long time before it is ready.

But:

I can say that a text formatting tool will probably be next.

Will this always be free and open source?

Yeah

How can I help?

I may implement a "buy me a coffee" thing on my GitHub in the future, but I don't have anything like that right now.

The big thing at this point would be feedback. I would like to know which features people would like, which ways the program can be improved, etc.

How can I reach you?

<https://github.com/Dannyimperial/frogFetch>

I don't know how GitHub works to be honest. I only used it once for school. But you can probably leave comments on there?

If not:

u/dannyimperial on [Reddit](#)

Just comment on the most recent post or send me a message.

Thank you.

All the pixel art included here is my own. I do not steal other people's art. No AI image generation was used.

