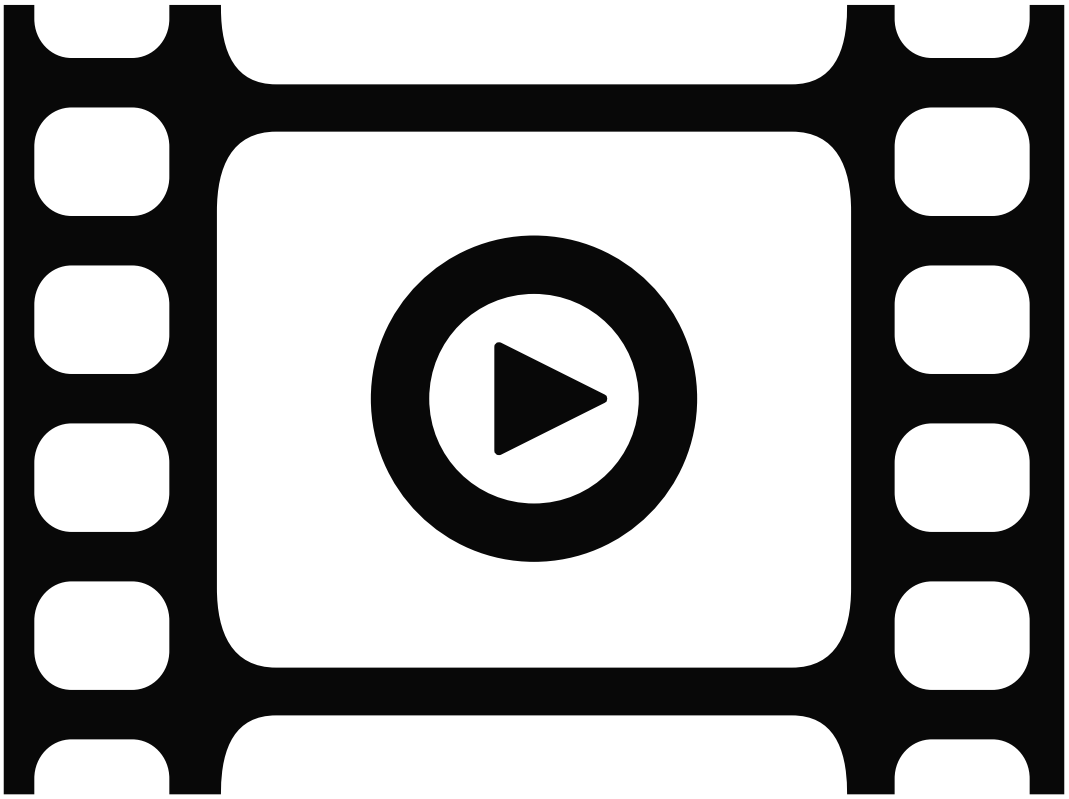
Pablo Rodríguez

GVoz

Record Voice and Times for Presentations



https://presvoz.gitlab.io

GVoz

Record Voice and Times for Presentations

Pablo Rodríguez

2024

https://presvoz.gitlab.io

Contents

Introduction

Why ConTEXt? *Why* Python *Instead of* Lua? *Sound Quality*
It's Up to You *Document Conventions* *License*

GVoz is a simple tool to record voice and times for presentations. With the help of *PresVoz*, it can also generate those presentations with voice.

GVoz has very basic functionality. Since it is a *Python* script, you can check yourself that my coding abilities are near to zero (if not below). Of course, any help is really appreciated.

Why ConTEXt?

ConTEXt is great at typesetting high-quality PDF documents. In fact, it is a "document generation system", as it used to be described. Although I'm aware that I may be abusing ConTEXt deploying it as a PDF library.

ConTEXt was my obvious choice to generate presentations with voice, being a ConTEXt user myself. All my presentations are generated with it. I'm also absolutely confortable using it to manipulate PDF documents: merging many PDF documents into a single file,[1] creating booklets, or simply imposing letter pages in A4 paper to print them.

1 There is a tool from *MuPDF* that may also merge documents, but it cannot set duplex and non-scale printing options in the resulting PDF document. This is only to mention one use case.

Against the use ConTEXt, non-TEXies may rightfully object it's a huge beast for such a specific use case.

There are PDF libraries under free licenses. Those accesible in *Python* are mainly *PyMuPDF* and *pyPDF2*. Before even asking their authors about individual features, one should not forget that multimedia is not the most common development in PDF, especially in open source software. There are good reasons for that, being one of the most important that PDF was mainly intended for printing.

In my case, it seems that none of these PDF libraries suppport what is required. Or at least, not all requirements are met by any single library. These are basically: *JavaScript* embedding, OCG layers and rendition objects. Rich media are not an option, because there is no way to check media position in them. This only seems to be available as a method for the media player object used for renditions.

Why Python *Instead of* Lua*?*

A legitimate question about *GVoz* would be why I use *Python* instead of *Lua* as the scripting language for *GVoz*. After all, ConTEXt generates the PDF document and invokes the required programs to create the *Flash* files.

GVoz was started from a sample file from the *Python* bindings for *Poppler-GTK+*. This was clearly before *GTK+ 3*. Both *Python* and *Lua* can access *GTK+ 3*, but *Python* has much more documentation and samples than *Lua*. At least, this is what it seems from the perspective of a total coding novice.

Since *GVoz* is shipped with *PresVoz* as part of a ConTEXt module, I might try to translate the *Python* script into *Lua*. This may make sense, but don't get me wrong: this won't happen in the near future. I even wonder whether this may happen in a foreseable future.

But before that, the maintenance status for LGI (**https://github.com/lgi-devs/lgi**) should improve. I'm afraid it seems at least problematic now.

Sound Quality

GVoz or *PresVoz* cannot achieve the impossible. One of their tricks for presentations with voice is to maximize the compression for recorded voice. How much is that? As much as it avoids too metalic or distorted sound.

Voice compression can be set as low as 32kbps in MP3 format. It might even go down to 16kbps using the *Opus* format. Although the default media player in *Windows* has no problem to play *Opus* sound files, *Acrobat* cannot handle the *Opus* media in *Windows*. I'm afraid that PDF presentations don't work in *Acrobat* for *macOS*.[2]

It's Up to You

There is nothing preventing you to recompress the sound recorded by *GVoz* with a higher bit rate, such as 320kbps. The

2 It seems impossible to check in *JavaScript* the current time for the media played. Without that information, it is impossible to set which page or slide should be displayed.

resulting sound file would be ten times bigger then, but you may have your reasons to increase the sound quality.

One of my main concerns with *GVoz* was providing the right files to generate presentations with voice using *PresVoz*. I think smaller files are better and wasting bytes is not really my thing. Besides, I don't have a voice that really needs more than 32kbps with MP3, or 16kbps with *Opus*.

But since *GVoz* is a script, you might go to the relevant line and set there a higher bit rate for all your voice recordings that fits your needs.

Document Conventions

This document is typeset using a slightly modified version of *Truetypewriter PolyglOTT* de Сергей Битов (also known as Sam_T). The italic font is only *slanted regular*. For the typewriter usage (where the keyboard is required to enter text, such as commands or internet addresses), the font is **slightly engrossed**.

Interactive links in this document are surrounded by a colored border. Internal links (pointing to other parts of this very same document) have green borders. External links (opened by your internet browser) have red borders. Both red and green colors are set to a bit darker tone than its pure values.

Interactive link borders are a PDF feature. They are displayed on screen, but the shouldn't be printed. As far as I know, the viewer in *Chrome* or *Edge* doesn't display them. *PDF.js* in *Firefox* displays them fine. If links lack colored

borders, you may blame your browser compilance with the PDF standard.

There are also page icons that include text to be simply copied and pasted. This avoids the extra format in the PDF body, which will cause errors when pasted in the terminal. Again, this is a pure interactive feature, so it won't be printed on paper.

License

GVoz is released under the third version of the *GNU General Public License.*[3]

It comes with no warranty and code may contain bugs. Unless you are willing to deal with glitches and faulty code, please don't use *GVoz*.

3 **https://www.gnu.org/licenses/gpl-3.0.html#license-text.**

1 What is GVoz?

A Basic Description

GVoz is a very basic *Python* script that records sound and transition times, while it displays the pages from a given PDF document.

The main use case is obtaining the required files for *PresVoz*, to generate presentations with embedded voice.

It might be helpful, since it records sound and slide transitions in milliseconds from the start of voice recording.

Since presentations with voice have a different approach than standard video, voice syncing with slides has to be recorded in the form of a slide duration list.

Sound and image syncing is already given (and taken for granted) in screen plus voice recording.

Just in case you might wonder, the *GVoz* name comes from the Spanish *graba voz* ("record voice").[4]

GVoz is licensed to the public under the terms of the third or any later version of the *GNU General Public License* (**https://www.gnu.org/licenses/gpl-3.0.en.html**).

4 *PresVoz* comes also from the Spanish *presentaciones con voz* ("presentations with voice").

B Required and Generated Files

All that *GVoz* requires is a PDF document, which should contain slides for a presentation. It is irrelevant whether it has been generated by ConTEXt, as long as it is able to copy all its pages.

Because it records your voice from the default microphone (or the device set for sound capture by your operative system), it will generate and uncompressed sound file. It will be automatically compressed to MP3 format to embed it in the final presentation with voice.

GVoz will also generate a pure text file with transition times for each slides. Each line contains a single number, which are milliseconds. The number is the duration of each slide, when it had been advanced to the next one (or when the recording finished, after the advance of the last slide).

2 Installation

A Requirements

GVoz is written in *Python*, using *Poppler* and *GStreamer* (through *GTK+3* GObject Introspection). This is why you and I can get so much with only 500 lines of code.

On *Linux*, you can install these dependencies through your package manager. Since names differ depending on which distribution is used, you have to investigate what to install.

On *Windows*, you can install them using the MINGW64 environment from the MSYS2 project (**https://www.msys2.org/**).

On *macOS*, *Homebrew* seems to have all installation requirements (**https://brew.sh/**). Since I don't have access to a *macOS* computer, you will have to investigate on your own. But dependencies should be similar to the ones in *Windows*.

B Windows *Installation*

α Dependencies

You need a whole environment to run *GVoz*. Installing MSYS2 isn't so difficult at it might seem first.

1. Download the installer from **https://www.msys2.org/#installation**.

2. Unpack in the provided directory. The suggested **c:\msys64** is fine.[5]

3. Once finished, update the MSYS2 installation typing **pacman -Syu** in the main MSYS2 terminal.

   In some cases, the installer may update the base packages and quit. In that case, you would have to update again typing **pacman -Syu.**

4. Install all packages required by *GVoz* with the following command in the MINGW64 terminal.

pacman -S --needed pactoys && pacboy -S --needed poppler:x python-cairo:x gobject-introspection:x python-gobject:x gtk3:x gst-python:x gst-plugins-good:x gst-plugins-bad:x gst-plugins-ugly:x

GVoz is already installed with *PresVoz*, since it is bundled in the ConTEXt **presvoz** module.[6]

5 All references bellow use this path. If you happen to use another path, please adapt commands accordingly.

6 *PresVoz* is installed with the **install-modules** script from ConTEXt:

mtxrun --script install-modules --install presvoz

ConTEXt is assumed to be installed in your user root directory, that would make **%USERPROFILE%\context**. In that case, you should prepend to the previous command **cd %USERPROFILE%\context\tex**, with the final result:

cd %USERPROFILE%\context\tex\ && mtxrun --script install-modules --install presvoz

If you use a different path, please adapt it in the invocation.

β Drag-and-Drop

To enable drag-and-drop in *Windows*, you have to copy the following command to **gvoz.bat** (a pure text file, which has to be renamed to that):

```
set
PATH=%PATH%;%USERPROFILE%\context\tex\texmf-win64\bin
C:\msys64\msys2_shell.cmd -mingw64 -use-full-path -c
'$USERPROFILE/context/tex/texmf-
modules/tex/context/third/presvoz/gvoz.py
"%~dpnl"'
```

After that, you may drag and drop any PDF document on **gvoz.bat** and *GVoz* will start automatically. If you simply double-click on **gvoz.bat**, a dialog will prompt you to select the file.

C Linux *Installation*

α Dependencies

In order to run, *GVoz* depends on:

- GObject Introspection for GTK+3 with Python bindings.
- Cairo with Python bindings.
- Poppler.
- GStreamer with Python bindings and all its plugins.

I don't know how the required packages are named in each distribution. Search for the names using the engine you please. I would be surprised if they weren't already packaged for your distribution.

β *Symbolic link*

As *GVoz* is part of *PresVoz* and it is installed as a ConTEXt module,[7] **gvoz.py** should be found where *PresVoz* is installed.[8] All you need is a link to a path known to your system.

Making a symbolic link for *GVoz* is easy. Just type in your terminal:

```
ln -s ~/context/tex/texmf-
modules/tex/context/third/presvoz/gvoz.py
~/bin/gvoz
```

In that case, your **$PATH** variable should include **$HOME/bin/.**[9] After that, *GVoz* can be invoked on the terminal, such as in:

```
gvoz document.pdf
```

Please link to **gvoz.py** contained on the installed package and avoid copying it. Otherwise, it won't be updated when a new version is included with *PresVoz*.

7 If you have ConTEXt in your user root directory (**$HOME/context/**), the easiest way to install the module is to type:

```
cd $HOME/context/tex/ && mtxrun --script install-modules --install
presvoz
```

8 Again, if you have ConTEXt in your user root directory (**$HOME/context/**), the path to *GVoz* should read **$HOME/context/tex/texmf-modules/tex/context/third/presvoz/gvoz.py**. If ConTEXt isn't on that path, please replace **$HOME/context/** in the previous full path with your actual directory.

9 You might include after editing **$HOME/.bash_profile** to include **export PATH=$PATH:$HOME/bin** and the end of it.

γ Desktop Icon

You may create a desktop entry, with contents similar to:

```
[Desktop Entry]
Categories=Education;Utility;Office;AudioVideo;Graph-
ics;Presentation;Recorder;GTK;
Keywords=Presentation;Recording;
Comment=Create presentations with voice
Exec=gvoz %f
Icon=/home/$USER/context/tex/texmf-
modules/tex/context/third/presvoz/gvoz.svg
MimeType=application/pdf;
Name=GVoz
Terminal=false
Type=Application
```

Save a pure text file to something similar to **gvoz.desktop**. You may want to place it on your desktop.

The image file **gvoz.svg** lives in **tex/texmf-modules/tex/context/third/presvoz/gvoz.svg** (same directory as for **gvoz.py**). You may want to adapt the **Icon** entry on your **gvoz.desktop** file to reflect the change.[10] If you followed the path installation suggestions for *PresVoz*, the absolute path for the SVG icon should read:

10 It only accepts absolute paths. I mean, **/home/user/context/tex/texmf-modules/tex/context/third/presvoz/gvoz.svg** works (given that **user** is the actual value of the **$USER** variable; otherwise, replace **user** with your actual user name).
But neither shortcuts such as **~/context/tex/texmf-modules/tex/context/third/presvoz/gvoz.svg**, nor variables such as **$HOME/context/tex/texmf-modules/tex/context/third/presvoz/gvoz.svg** would work.

/home/$USER/context/tex/texmf-modules/tex/context/third/presvoz/gvoz.svg

Don't forget that you have to give the actual value of the **$USER** variable, since the desktop entry won't be able to solve it.

3 Usage

A Basic Invocation B Options C Generated Files
D Presentation Generation E Sample Presentation F Issues

A Basic Invocation

If you are more a keyboard than a mouse person, you can always type (given that the path is know to your operative system):

gvoz document.pdf

Otherwise, you may just double-click or drag-and-drop a PDF document on your batch or desktop file, depending on being *Windows* or *Linux* your operative system.

B Options

If you want to enable mouse clicks to start recording and advance slides, there is an option pressing **m** to enable it. I'm afraid it isn't a good idea.[11]

The following keys are the way to use *GVoz*:

␣ starts sounds and times recording.[12]

11 Before you do it, please be warned that I never made it work. I mean, some single clicks were counted as two (or even more) slide advances. Not few of them taking place even on the same milisecond.
12 ␣ represents the space bar (or the empty space).
Once recording has started, pressing the space bar again advances the next slide and records its transition time.

a displays *GVoz* information.

c disables presentation generation by *PresVoz*.

d shows PDF metadata included in the MP3 file.

h displays keyboard help information.

m enables mouse to behave as the space bar.

p pauses and unpauses recording.

s converts sound to *Opus* format.

x removes metadata in generated files and presentation.

q quits *GVoz* inmediatly; no presentation is generated.

C Generated Files

From slides in PDF document named **document.pdf**, *GVoz* will record voice in **document-audio.wav** and times in **document-times.txt**.

When started, *GVoz* checks whether there is a previous voice recording or timeline from the same slides. If found, it prompts the user whether to proceed or quit. If the user starts recording, the MP3 audio file from the slides will be renamed too (along with the previous voice recording and timeline).[13]

13 Since *PresVoz* chooses MP3 over WAV files, this avoids embedding a wrong older recording in the final presentation.

If the users selects a *PresVoz* presentation to record sound and times, *GVoz* will prevent that.[14] Detection mechanism is not based on the final file name string **-presvoz** and the file extension, but on part of the subject metadata containing **Presvoz ✌🐞 - https://.**

D Presentation Generation

Advancing one more slide after the last one stops voice and times recording. Unless **c** has been pressed (to disable presentation generation), *GVoz* will automatically invoke *PresVoz* to generate the final presentations.

E Sample Presentation

The module includes three extra files in the documentation:

1. **presentation-presvoz.pdf**
2. **presentation-loader.swf**
3. **presentation-presentation.swf**

The first file is a PDF document with embedded sound and synchronized slides. Only *Acrobat* is able to play the sound and advance the slides automatically (only in *Windows*).

The other two files are *Flash* animations (or presentations, in this case). Just in case you might wonder, these can be

14 As a general rule, it is not a wise decision to use *PresVoz* presentations as source for *GVoz*. The main reason is that player buttons are already contained in the source. Buttons would appear in both resulting PDF and *Flash* presentations as source images. I mean, they wouldn't be operational and both play and pause, full screen and unfull screen would appear on top of each other.

played using *Ruffle*[15]. The first *Flash* file is only a loader, which only contains the first slide and a button. The second one is the presentation itself.

The PDF presentation may be controlled only with buttons. Its *Flash* version can be controled with both buttons and keystrokes.

The PDF presentation starts playing after pressing the play button (and permissions are granted to play multimedia files).

The *Flash* presentation starts playing automatically when invoked directly. Through the loader, it starts playing in full screen after pressing the central button.

Buttons and other controls in the *Flash* version are enabled either by pressing **c** or selecting *Enable visible controls* in the contextual menu.

Once enabled, they can be automatically hidden (only with playing sound), either by pressing **a** or selecting *Autohide controls* in the contextual menu.[16]

Other controls are:

␣ (and double-click) start/stop sound.

15 **https://ruffle.rs**. It comes as a desktop program, as a *Firefox* or *Chrome* extension, or as selfhosted to embed in your own pages. **https://ruffle.rs/demo/** may be the simplest way to check *Flash*, by dragging and dropping presentations generated by *PresVoz*.

16 Although it requires that the mouse is moved after in order to start disappearing. After that, only going to the slide bottom will display controls again.

f enables/disables full screen.

← (or **PgDn**) goes to previous slide.

→ (or **PgUp**) goes to next slide.

↑ (or **Home**) goes to last slide.

↓ (or **End**) goes to first slide (and starts again).

F Issues

If you experience an issue with *GVoz*, please report it at **https://gitlab.com/presvoz/gvoz/-/issues/new.** Patches are always welcome.

This document was generated with **pandoc**
(**http://pandoc.org/**) and typeset with
ConTEXt (**http://contextgarden.net/**).

TrueTypewriter PolygloTT
was used for this document.