pyglet$_{f}$ $fmpegDocumentation$

Release 0.1.14

Daniel Gillet

Oct 06, 2019

Contents:

CHAPTER 1

Installation

Warning: pyglet-ffmpeg currently only supports Windows (32 or 64 bits) and Mac OS X 10.5 and above.

pyglet-ffmpeg bundles the binaries for FFmpeg v4. On PyPI, pyglet current stable release (v1.3.2) does support FFmpeg at all. There is an unstable release (v1.4.0a1) but it's using FFmpeg v3.

In order to make things work, you will need to install pyglet from Bitbucket repository:

```
$ pip install -e hg+https://bitbucket.org/pyglet/pyglet/branch/default#egg=pyglet
```

You install pyglet-ffmpeg using pip:

```
$ pip install pyglet-ffmpeg
```

CHAPTER 2

Usage

Using pyglet-ffmpeg is really simple. In your code, simply call:

```
import pyglet_ffmpeg
pyglet_ffmpeg.load_ffmpeg()
```

FFmpeg will be loaded and ready to be used with pyglet or arcade.

CHAPTER 3

Example using Pyglet

Let's say you have an audio file *laser1.ogg* in the same directory as this script. Here is a minimal example which would play the sound.

```
import pyglet
import pyglet_ffmpeg
from pathlib import Path

pyglet_ffmpeg.load_ffmpeg()

window = pyglet.window.Window()

this_dir = Path(__file__).parent
soundfile = str(this_dir / 'laser1.ogg')
sound = pyglet.media.load(soundfile)
player = sound.play()

player.on_player_eos = window.close
pyglet.app.run()
```

CHAPTER 4

Development

Want to work on `pyglet_ffmpeg`? This document explains how to get a working sandbox, make releases, and some of the decisions made for good housekeeping.

CHAPTER 5

Sphinx

This project uses Sphinx for documentation. The docs are hosted on ReadTheDocs, connected by the RTD/GitHub integration. Local docs generation uses the `sphinx_rtd_theme`.

CHAPTER 6

Testing

This project uses `pytest`. The tests are under `tests`, with directories for unit test and other kinds of test. We also use `pytest-mock` and `pytest-cov`.

Run tests with coverage from the command line with the following, done from the root directory:

```
$ pytest --cov=pyglet_ffmpeg tests
```

CHAPTER 7

Version Numbering

This project uses semantic versioning. To ensure the scheme is enforced and to update all locations of the version number (setup.py, doc/conf.py, pyglet_ffmpeg/__init__.py), we use bump2version to set our version numbers.

Bump the version using bump2version commands:

- *bump2version patch: 0.1.0 -> 0.1.1.dev0*
- *bump2version release: 0.1.1.dev0 -> 0.1.1*
- *bump2version minor: 0.1.1 -> 0.2.0.dev0*
- *bump2version dev: 0.2.0.dev0 -> 0.2.0.dev1*
- *bump2version release: 0.2.0.dev1 -> 0.2.0#. Tag the release.*

It's a little bit of a hassle but better than doing it manually.

Also, we set the `bump2version` config file to automatically tag and commit.

CHAPTER 8

PyPI

Releases are pushed to PyPI from Travis, using the regular TravisCI-PyPI scheme. A push to a tag triggers not just the rest of the Travis build, but also the upload to PyPI of the new version. . . hands-free, from Travis.

CHAPTER 9

Compiling FFmpeg for Ubuntu

To make new binaries for Ubuntu, here are the required steps:

```
sudo apt-get update -qq && sudo apt-get -y install \
  autoconf \
  automake \
  build-essential \
  cmake \
  git-core \
  libtool \
  pkg-config \
  texinfo \
  wget \
  zlib1g-dev \
  yasm

mkdir -p ~/ffmpeg_sources ~/bin

cd ~/ffmpeg_sources && \
wget https://www.nasm.us/pub/nasm/releasebuilds/2.13.03/nasm-2.13.03.tar.bz2 && \
tar xjvf nasm-2.13.03.tar.bz2 && \
cd nasm-2.13.03 && \
./autogen.sh && \
PATH="$HOME/bin:$PATH" ./configure --prefix="$HOME/ffmpeg_build" --bindir="$HOME/bin"␣
↪&& \
make && \
make install

cd ~/ffmpeg_sources && \
wget https://nixos.org/releases/patchelf/patchelf-0.9/patchelf-0.9.tar.bz2 && \
tar xjvf patchelf-0.9.tar.bz2 && \
cd patchelf-0.9 && \
autoreconf --verbose --install --force --warnings=all && \
PATH="$HOME/bin:$PATH" ./configure --prefix="$HOME/ffmpeg_build" --bindir="$HOME/bin"␣
↪&& \
```

(continues on next page)

(continued from previous page)

```
make && \
make install

cd ~/ffmpeg_sources && \
wget -O ffmpeg-snapshot.tar.bz2 https://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2 &&
↪ \
tar xjvf ffmpeg-snapshot.tar.bz2 && \
cd ffmpeg && \
PATH="$HOME/bin:$PATH" PKG_CONFIG_PATH="$HOME/ffmpeg_build/lib/pkgconfig" ./configure␣
↪\
  --prefix="$HOME/ffmpeg_build" \
  --extra-cflags="-I$HOME/ffmpeg_build/include" \
  --extra-ldflags="-L$HOME/ffmpeg_build/lib" \
  --extra-libs="-lpthread -lm" \
  --bindir="$HOME/bin" \
  --disable-programs \
  --disable-doc \
  --disable-static \
  --enable-shared \
  --disable-avdevice \
  --disable-postproc && \
PATH="$HOME/bin:$PATH" make && \
make install && \
hash -r
```

This will create the needed so files in *~/ffmpeg_build/lib*. Move into this directory and use *patchelf* to add relative path to each **so** file so they can load their dependencies.

```
for file in *.so.*.*;
    do ~/bin/patchelf --set-rpath \$ORIGIN "$file";
done;
```

You can now copy those files to the `linux_x86_64` folder in pyglet-ffmpeg package.

Warning: Only copy the libraries, not the symlinks. The compilation step will have created for instance a file named `libavcodec.so.58.21.104`, but there will be two symlinks named `libavcodec.so.58` and `libavcodec.so`. Only copy `libavcodec.so.58.21.104`. The package will re-create the correct symlinks when running, but only if the symlinks are **not** initially present.

CHAPTER 10

Indices and tables

- genindex
- modindex
- search