ex-hale / **instagiffer**

👁 Watch   1   ★ Star   13   ⑂ Fork   0

<> Code   ⓘ Issues **1**   Pull requests **0**   Projects **0**   Security   Insights

Instagiffer creates GIFs from YouTube, HD video files, Netflix or your screen.

🕑 **1** commit   ⑂ **1** branch   📦 **0** packages   🏷 **0** releases   👥 **0** contributors

Branch: **master ▾**   New pull request   Find file   Clone or download ▾

Justin Todd Initial import ⋯   Latest commit 704a8ab on May 4

📁 bindeps	Initial import	7 months ago
📁 doc	Initial import	7 months ago
📁 macdeps	Initial import	7 months ago
📄 .DS_Store	Initial import	7 months ago
📄 AppxManifest.xml	Initial import	7 months ago
📄 README.md	Initial import	7 months ago
📄 instagiffer.conf	Initial import	7 months ago
📄 instagiffer.icns	Initial import	7 months ago
📄 instagiffer.ico	Initial import	7 months ago
📄 instagiffer.py	Initial import	7 months ago
📄 installer.iss	Initial import	7 months ago
📄 main.py	Initial import	7 months ago
📄 make_app.sh	Initial import	7 months ago
📄 make_exe.bat	Initial import	7 months ago
📄 setup-mac-py2app.py	Initial import	7 months ago
📄 setup-win-cx_freeze.py	Initial import	7 months ago
📄 uninstall.ico	Initial import	7 months ago

📖 **README.md**

InstaGiffer

Whether you pronounce it GIF, or GIF, Instagiffer is the perfect tool for creating the best reaction moments to any situation. Sometimes, only a GIF will do in a conversation, and can say so much by saying so little. If you find yourself struggling to find the perfect GIF, look no more—just make your own. It's easy. Promise.

Instagiffer was designed during an era where websites would limit upload file sizes to 1MB, therefore, my primary goal was to generate highly-optimized GIFs that still look great. It started out as a CLI script which I used for my personal GIF creations and it grew organically from there—the user experience was never my primary focus, and because of this, the UI is a bit quirky. Help me improve it!

Instagiffer Architecture

Instagiffer is a monolithic Python script. The UI is developed in Tkinter. GIF generation is performed using FFMpeg which is used to extract frames from videos, and ImageMagik, which is used for effects, cropping, text, and GIF compression. Videos are downloaded from YouTube using youtube-dl. Instagiffer simply calls these three processes to generate GIFs.

The Instagiffer binary is generated using py2app (Mac OS) and cxFreeze (Windows). The Windows installer is assembled using the free Inno Setup utility.

Setting Up Your Development Environment

Windows

The following prerequisites are required to do a Windows build:

- Sublime Text 3 (Preferred editor, optional)
    - Package Control
- Inno Setup (Latest)
- Python (2.7.12)
- Github Client (Latest)
- Pillow 3.4.2
- cx_Freeze (4.3.x) - don't use version 5!
- PyWin32 (2.19)
- Update Instagiffer\bindeps
    - Latest FFMpeg 32-bit static build
    - Latest convert.exe and mogrify.exe from Imagemagick Portable x86 zip
    - Latest youtube-dl binary release

Mac

Macs require a little more work:

- A Mac or Mac VM
- Github Desktop - Download and clone instagiffer repo
- Sublime Text 3
- Python tools requirements:
  - pip: sudo easy_install pip
  - PIL: sudo pip install pillow
  - py2app: sudo pip install -U py2app
- Update instagiffer\macdeps with latest versions
  - ffmpeg: http://www.ffmpegmac.net
  - youtube-dl: curl https://yt-dl.org/downloads/2016.02.22/youtube-dl -o youtube-dl && chmod +x youtube-dl
  - ImageMagick (To see how I built a stand-alone version of IM, see below

Building ImageMagick (Mac)

1. Download and install xcode from the App Store - 1 hour

   - Start Xcode - accept license agreement. exit
   - xcode-select --install # command line tools
   - sudo xcode-build -license

2. Download MacPorts package and install: https://www.macports.org/install.php

   - sudo port -v selfupdate

3. sudo vi `port file ImageMagick`

4. Add --enable-delegate-build, --disable-shared

5. sudo port install ImageMagick -x11 +universal # this takes about 2 hours

6. Copy ImageMagick binaries to macdeps/ using the following bash:

   IM_DIR=$HOME/instagiffer/macdeps/im

   rm -Rf $IM_DIR && mkdir $IM_DIR && cd $IM_DIR

   for d in bin etc usr lib; do mkdir -p $d done

   echo "Copy files" cp -LR /opt/local/etc/fonts /opt/local/etc/ImageMagick-6 $IM_DIR/etc cp /opt/local/bin/convert /opt/local/bin/mogrify $IM_DIR/bin cp /opt/local/lib/ImageMagick-6.9.3/config-Q16/configure.xml $IM_DIR/etc/ImageMagick-6

   echo "Update paths in config files" find $IM_DIR | grep -E '.xml|.conf' | xargs sed -ie 's#/opt/local#./macdeps/im#'

   function getdeps(){ for x in $(otool -L $1 | perl -ne 'if(/\s+(/opt.+?) /){print "$1\n"}'); do echo $x if [$x != $1] ; then getdeps $x fi done }

   for x in $(getdeps bin/convert | sort | uniq); do bn=$(basename $x) cp $x lib echo "copied $bn" done

   for binpath in bin/* lib/*.dylib ; do echo "Modifying $binpath" for dy in $(otool -L $binpath | perl -ne 'if(/\s+(/opt.+?) /){print "$1\n"}') ; do bn=$(basename $dy) echo " Update $bn" install_name_tool -change $dy @executable_path/../lib/$bn $binpath done done

Building a Release

Before you release, double-check:

- Version numbers updated?
- Help links work?
- Test everything on vanilla Mac/Windows VM?

Windows

- Run make_exe.bat to produce a Windows installer, and portable zip

Mac

- Run make_app.sh to produce a DMG installer. When the DMG pops up, move the icon into the right position.

Testing

Test URLS

- https://www.youtube.com/watch?v=EPP7WLuZVUk

Some Things I Test On Each Release

Here are some things I regression test on each release (based on bugs I've encountered in the past)

- Online downloads
  - Youtube (especially popular music videos)
  - VEVO
  - DailyMotion, and Metacafe
- Videos where h > w, w == h, w > h
- Invalid video formats
- Cellphone videos shot in portrait and landscape
- Corrupted video
- Any videos that revealed a bug previously
- Image sequence
- Image sequence with bad image in the middle
- Image sequence of 1
- Unicode paths
- Cinemagraphs
- Settings at extremes
- Specify time out of range
- GIF Overwrite on and off
- Different youtube qualities
- Screen capture
- Check for memory leaks
- Super-long GIF
- Help -> Generate Bug Report
- Make a 5 second GIF using screen capture feature. If you have a second monitor, capture something there.
- Frames -> Export Frames (pick a folder and ensure the frames are correctly exported)
- Frames -> Delete Frames. Delete all even frames
- Frames -> Import Frames - Use all of the frames you just exported (hint, you can multi-select)
- Create a GIF people having a dialog with text captions. Do this using a Youtube video.
- Create a GIF 10 seconds or longer, and under 1MB
- Input invalid data
- Unpopular video fomats or sites
- Button-smash the GUI (Hit escape to interrupt events)
- Load invalid or corrupted movie files
- Go into instagiffer.conf and muck around with configuration parameters (within reason) - requires you to close and re-open Instagiffer.

Look and feel

Color

Main Color: #395976

Font

"Bookman Old Style":

Bookman Old Style

Terms   Privacy   Security   Status   Help   Contact GitHub   Pricing   API   Training   Blog   About