tvPlayer as digital props for film and TV

This is basically a media player.

Power it and it will play the first image or video on the USB stick.

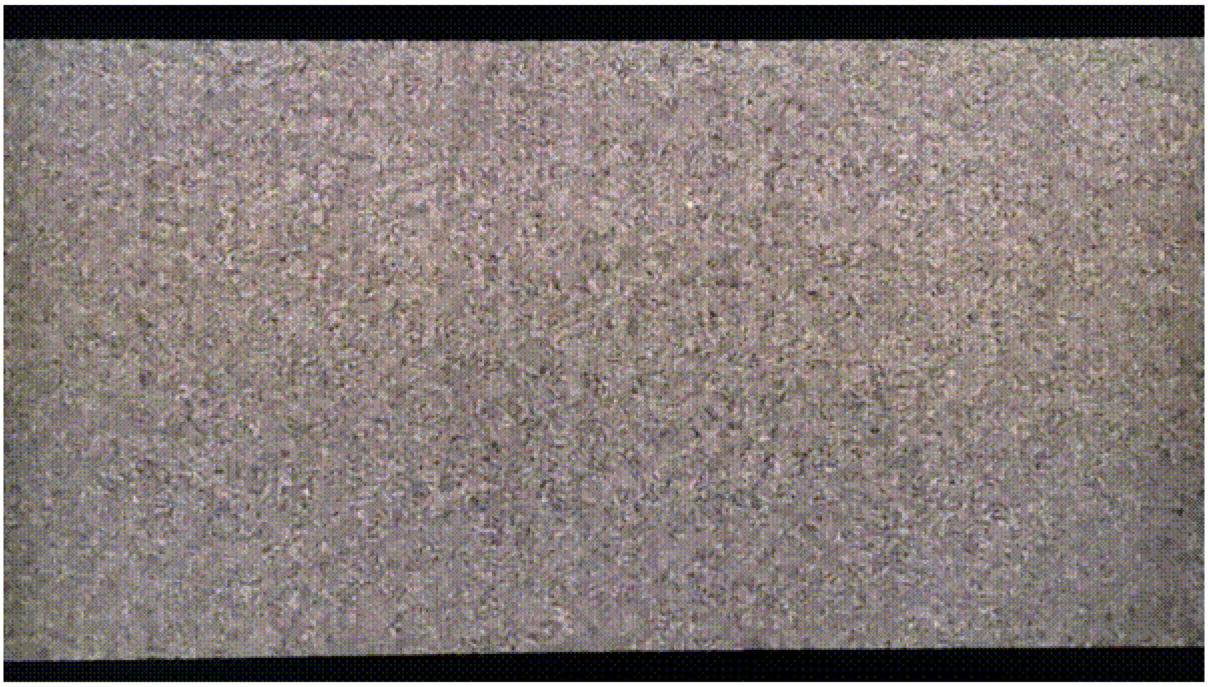
With some keyboard strokes you "change the channel", now you have an interactive TV station.

Fill a USB stick with video (or *image!*) files; plug it in; have a fake TV.
Channel number is shown when switching them, white noise is shown when nothing is available and the volume bar is there for you when you expect it. Both can be turned off on-the-fly.

Control it with a wireless keyboard and your actress ort actor can press what they want, you'll be sure they find their program as *per the script*.

What you can do

- You can set inpoints for each video, so when you switch channels, it starts directly at the key moment, like when the journalist says, "*...some minor accident today in the hospital of Dunwich was contained and we have nothing to worry about. Now to the weather.*."
- You can turn the TV fake-off ("blackout").
- You can switch video-fitting-mode from contain to stretch to cover, to get rid of potential letterboxes.
- You can turn off the fancy effects.
- The files do not auto-progress from one to another. Each video loops itself until you "switch the channel" (play the next file on the USB device).
- Change the brightness of the video to match exposure times of the camera.
- If everythings lost, theres a green screen button



Hardware

Tested successfully on Raspberry 4 (Debian bullseye).

Raspberry pi 3 (Debian Bookworm 32bit) did ***not*** work. Very stuttery but very plausibly i'm doin something wrong.

Combine it with a HDMI-to-RCA adapter **and** a RCA-to-RF adapter to display on an old TV. Even sound works!

Get this up and running

1. Clone this repo `git clone https://github.com/falue/tvPlayer`
2. run the install script `sudo bash install.sh` to:
   - apt-get update
   - install dependencies (Packages: *mpv*, *socat*, *wmctrl*; Python: *pygame*, *natsort*)
   - auto-run `python3 autostart.py` on autostart[*]
   - ~~disable window "removable medium is inserted"~~ [BUG!]
   - create a desktop shortcut to the program
3. Set **audio output** to HDMI (right click on audio in toolbar, choose HDMI)
4. disable pop up window "**removable medium is inserted**": Open any folder > `Edit` > `Preferences` > `Volume Management` > uncheck `Show available options for removable media when they are inserted`
5. Insert USB, start watching TV.

> [*] The `autostart.py` script asks if you want to close all other autostarting windows and after a 12s timeout, it starts the main script `tvPlayer.py`. This is because other windows can overlap the tvPlayer and hinder the fullscreen mode. So without a keyboard and doing anything, the player goes to fullscreen on startup.

File handling

MPV player: Playable media

The media player MPV (doc wiki) used here can play pretty much everything - *however*, filename-endings are fixed to work with `.mp4`, `.mkv`, `.avi`, `.mxf` and `.mov` (case insensitive).

It uses ffmpeg to decode, so the list of playable media is huge.
According to the mpv.io website:

> File Formats: mpv supports a wide variety of media formats, including popular video files (e.g., MP4, MKV), audio codecs (e.g., AAC, MP3), and subtitles.

Some cherry picked examples:

> - Containers: MP4, MKV, AVI, WebM, OGG, FLV, and more.
> - Video Codecs: H.264, HEVC, VP8, VP9, AV1, MPEG-4, MPEG-2, and others.

> - Audio Codecs: AAC, MP3, Vorbis, FLAC, Opus, AC3, and DTS.
>   Subtitle Formats: SRT, ASS, SSA, VTT, and embedded subtitle tracks in containers like MKV.

Manually tested:

- ✓ `.mp4` MPEG-4 AAC, H264
- ✓ `.avi` MPEG-4 mp3
- ✓ `.mkv` h264, yuv420p
- ✓ `.mxf` mpeg2video (4:2:2), yuv422p
- ✓ `.mp4` HEVC, H265
- ✓ `.mov` H264
- ✓ `.mkv` "4k UHD" h264 yuv420p works (but stuttering on raspberry pi4 @8gb)

Also works with *images*!

Filetypes for images: `.png`, `.gif`, `.tiff`, `.bmp`

> *NOTICE*: Does ***not*** work with `.jpg`!

File and TV channel order

The order of the files on the USB device lead the order of the TV channels.

The alphabetically first file (case insensitive, numbers before letters) is the first channel, etc.

If you want to change the order, rename the files appropriently.

Channel number and volume bar styling

If you want to change the appearance of the channel numbers or the volume bar, you have to update the `.bgra` files in `/channel_numbers` and `/volume_bars`.

You can edit the pngs and convert them to `.bgra` files. Change the variable `input_folder` in it first, and then run bash script `python3 png_to_bgra.py` to convert the files.

> *Note*: For simplicity, keep the image sizes the same as they are hard coded. Otherwise you have make changes to the script (change the parameters of both `display_image(...)` instances)

Transparency

Complete black pixels will become transparent.
If you want "black" to show up, use `rgb(1,1,1)`.

Keyboard controls

Keypress	Action	Note
UP	next channel (next file)	

Keypress	Action	Note
DOWN	prev channel (next file)	
[number]	go to channel nr	
LEFT	jump -5 seconds	
RIGHT	jump 5 seconds	
SHIFT and LEFT	pause and one frame backwards	
SHIFT and RIGHT	pause and one frame forwards	
p *or* space	toggle play / pause	
ESC	toggle fullscreen	
q	shutdown raspberry pi	
Q (SHIFT and q)	exit program	
b	toggle black screen	
g	cycle through green screens →	
G (SHIFT and g)	cycle through green screens ←	
c	video fitting (contain, stretch or cover)	
i	set inpoint (where the file starts to play)	
I (SHIFT and i)	clear inpoint on this video	
a	toggle tv-animations (*pause in between channel changes*, *channel number*, *vol bars*)	
w	if tv-animations: toggle color of pause in between channel changes: *white noise* or *black*	
,	reduce video brightness by 5%	
.	increase video brightness by 5%	
+	reduce volume by 10%	
-	increase volume by 10%	