Using a Raspberry Pi and PI Presents to Make a Touch Screen Media Show

Greg Reynolds   August 2016

The purpose of this guide is to show how to create touch screen presentations on a Raspberry Pi computer (RPi or just Pi) using Pi Presents.

This guide assumes the following.

1. You have an Rpi running with a display, keyboard, and mouse. The display does not have be a touch screen, but should be the same screen size as the touch screen you intend to use. (A touch screen normally has an USB output that is identical to a mouse. Clicking the right mouse button is the same as touching the screen at the cursor location.)

2. You have gone to the Pi Presents website and downloaded the program to your RPi.

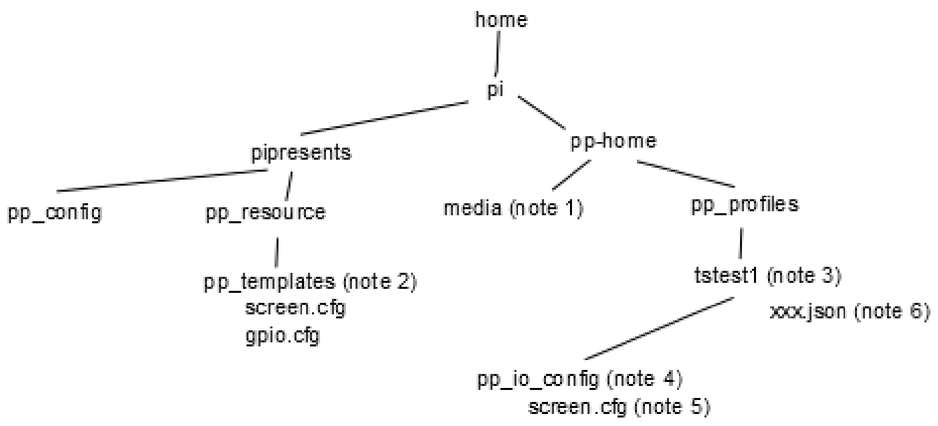
3. You have the videos you want to show under touch control on a USB Flash Memory stick. See **Appendix** to see what types of video files the RPi can play. You have transferred these files to folder home/pi/pp-home/media on the RPi. This directory is where Pi Presents looks for media files.

Pi Presents calls its programs “profiles”. The two terms are used interchangeably here.

4. You have some experience using the Pi Presents Editor, and you have transferred your videos into the media folder in the RPi.

The rest of this guide is to create a touch screen to run these videos on to the RPi.

The directory structure of Pi Presents is fairly complicated. To better understand this, a directory tree is helpful. A partial directory tree of Pi Presents is shown in Fig. 1. In this case, the name of the profile created is tstest1.



Note 1: This file is where Pi Presents looks for media files. Place all media files here.

Note 2: This file contains templates of the configuration files. For touch screen use screen.cfg

Note 3: Everything in this directory is specific to this Pi Presents program, namely tstest1.

Note 4: This directory (with screen.cfg in it) is necessary in a Pi Presents program if a touch screen is to be usec

Note 5: This file is the specific touch screen configuration for the touch screen profile.

Note 6: There are number of .json files in a Pi Presents program. They contain data exclusive to PI Presents. Don't touch them.

Fig 1 Partial Directory Tree of Pi Presents

Explore the Pi Presents directories using the File Manager on the desktop. The copy and paste functions are used in this guide.

1. Video Files

As mentioned above, the videos you want to play must be stored in the home/pi/pp_home/media directory. In this example, the media files are called vid0.AVI, vid1.AVI, and vid2.AVI, and vid3.AVI. We will come back to these later.

2. Creating a Configuration File

To make touch sensitive areas on a screen play a video, you must have in the profile folder called pp_io_config. In this folder there must be a file called screen.cfg. If you look at the directory tree in Fig. 1 you will see two files of this name. The first is a general purpose template under home/pi/pipresents/pp_resource/pp_templates. If you open this file with a text editor, you will see that it is filled with comments (# first symbol in the line) that tells you how to modify this file. It is also described in Section 13.2.2 of the Pi Presents Gapless Manual. The other screen.cfg file is the specific one for the tstest1 profile, so it is under home/pi/pp_home/pp_profiles/tstest1/pp_io_config. In the screen.cfg template below the comment section are groups that define each touch sensitive area. In this example, there are four touch areas defined, so four description sections are defined. All other sections in the template can be removed. Below are the four sections that were changed to make the four touch areas or buttons that play the four video files of the example.

```
[pp-play0
name=pp-play0
points = 800+100+75*75
fill-colour =
outline-colour =
text =
text-font = arial 15 bold
text-colour = black
image = +/resources/blank_pink.jpg
# image =
image-width = 75
image-height= 75


[pp-play1]
name=pp-play1
points = 975+100+75*75
fill-colour =
outline-colour =
text =
text-font = arial 15 bold
```

```
text-colour = black
image = +/resources/blank_green.jpg
# image =
image-width = 75
image-height= 75

[pp-play2]
name=pp-play2
points = 800+230+75*75
fill-colour =
outline-colour =
text =
text-font = arial 15 bold
text-colour = black
image = +/resources/blank_pink.jpg
# image =
image-width = 75
image-height= 75

[pp-play3]
name=pp-play3
points = 975+230+75*75
fill-colour =
outline-colour =
text =
text-font = arial 15 bold
text-colour = black
image = +/resources/blank_green.jpg
# image =
image-width = 75

image-height= 75
```

3. Discussion of the Touch Sensitive Area Descriptions

The touch sensitive areas or buttons are assigned names. The first one is called pp_play0, but any name will do. These names will be associated with video file reference names later.

The display screen positions are specified in “picture elements” or pixels. A standard HDTV screen is 1920x1080 pixels. The location 0x0 is the top left of the screen and the lower right is positon 1920x1080. The x positions are positive numbers going to the right. The y positions are positive numbers going down.

The “points” description is defined as x1+y1+width*height. x1 and y1 define the location of the top left corner of a button and the next two terms the width and height of the button. If you look at the points description of the button pp_play0, namely 800+100+75*75, it is a 75 by 75 pixel square located at 800x100. Note that each button is the same size but at a different location. The next five lines describe possible elements of the buttons that are not used in this example. The location of the appearance of the buttons is specified by “image”. Initially, these buttons must be able to be seen on the screen to adjust their location, but must become transparent so the background image (more about this later) is not covered by the buttons or touch areas. To do this, the image line becomes “image = “. The “image-width” and “image-height” lines should be the same size as the button width and height specified in the “points” line.

Both the comments in the screen.cfg template and section 13.2.2 of the Pi Presents manual described the purpose of these lines in detail.

All this fiddling around with numbers to position the touch area is why, in the beginning assumptions, it was suggested to use a display screen the same size as the actual touch screen that is going to be used. If you have a different size screen all the positioning of touch areas will have be redone.

The screen.cfg file is modified using a text editor**. This must be done carefully or an error message will occur.**

3. Creating a Background Image

You probably want to design a graphical image for the touch screen showing the name of the videos that will be played by that touch area button. This is why you will want the buttons to be transparent in the final profile.

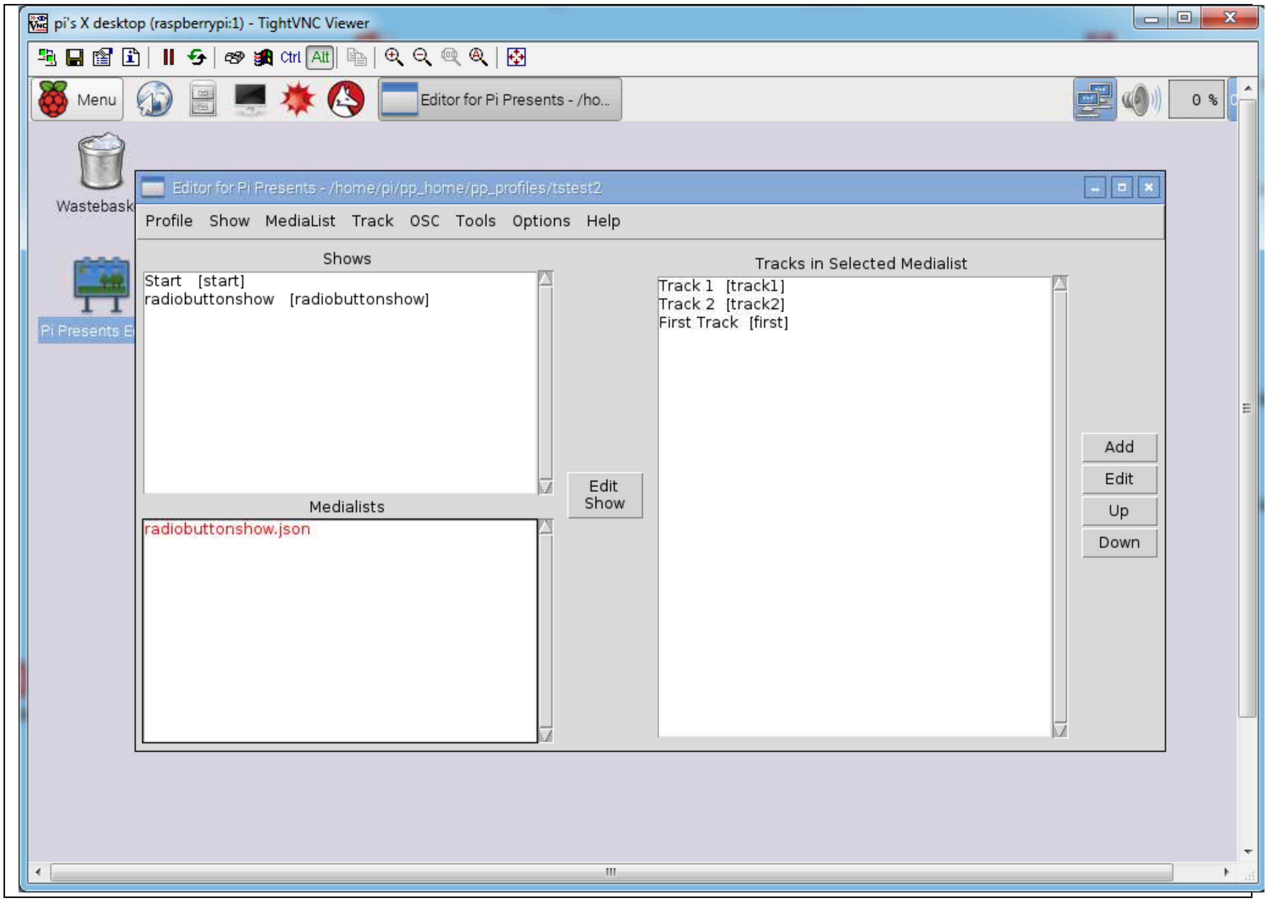
Pi Presents accepts image files of the following type:

'Image files','.gif','.jpg','.jpeg','.bmp','.png','.tif'

If the graphics design tool you are using does not save in any of these formats, you can print the graphic on paper and scan it into your computer saving as a .bmp type. The graphic for this example was created in MS Word and then scanned to convert to a .bmp file. The file must then be placed in the media folder along with your video files.

4. Creating the Touch Screen Profile

The profile of the touch screen media show can now be created. Open the Pi Presents Editor, pull down the Profile menu (top left of the tool bar) and select the New from Template tab and select Radiobutton Show. Enter your new profile name. Items will appear in the Editor windows. In the Medialist panel select radiobuttonshow.jason. This will display the tracks of this template in the right hand window Tracks in Selected Medialist. You should now have the window in Fig. 2.


pi's X desktop (raspberrypi:1) - TightVNC Viewer

Editor for Pi Presents - /home/pi/pp_home/pp_profiles/tstest2

Shows

Start [start]
radiobuttonshow [radiobuttonshow]

Medialists

radiobuttonshow.json

Tracks in Selected Medialist

Track 1 [track1]
Track 2 [track2]
First Track [first]

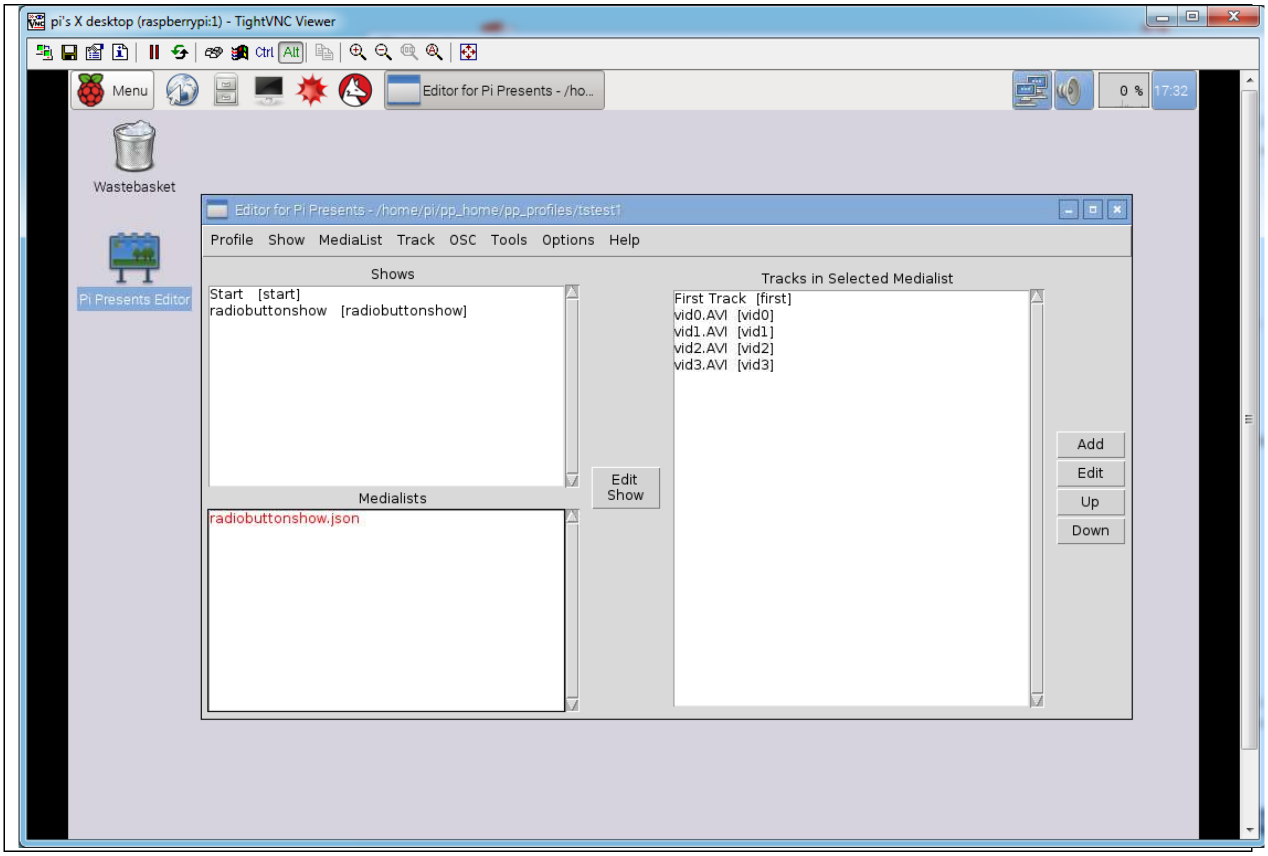
Fig. 2 The radiobutton template displayed

Select radiobuttonshow in the Shows window and click Edit Show button in the center of the Editor window. Select the Egg Timer tab and delete the text in the Egg Timer window. This is a fully editable window. Select the tab Show Background and Text and click the button to the right of top line Background Image. A widow to the media folder will appear and select the background image you placed there by pressing Open. The path to you image will be entered. Press OK. Select the Track Default tab. In the line Video Window, change entry to warp 0 0 1850 950. This will make your video fill most of the screen when it is played. If you are using a HDTV display, the entry should be warp 0 0 1920 1080.

Another example of the advantage of developing the profile on a display that is the same size as the one that will really be used.

Now we have to modify the tracks in the profile. Select radiobuttonshow.json in the Medislist window to display the tracks in the Tracks in Selected Medialist window. Although it's not absolutely necessary, we will delete the tracks from the template, Track 1 and 2. Select Track 1, then pull down the Track menu in the top left tool bar and select Delete. Do the same for Track 2. Now bring in your videos to the profile. Press the Add button at the very right of the Editor window. A window displaying the contents of the media folder will again appear. Select your video, and press open. Your video will now appear in the Tracks window of the Editor window. In this example, there are four videos, vid0, 1, 2, and 3.

Now we have to add Track Reference names to the videos in the Tracks window. Select the video, and press the Edit button to the right of the Tracks window. In this editor window, select the track tab. Delete the text in the Message Text window. In the Track Reference line add a reference name to your video. It can be named anything. In this example for convenience, the reference names were made the same as the track names, namely vid0, vid1, vid2, and vid3. Press OK. The reference name will appear in square brackets after the track name in the Track window. Below in Fig.3 is the final Editor window for this example.


pi's X desktop (raspberrypi:1) - TightVNC Viewer
Ctrl
Alt
Menu
Editor for Pi Presents - /ho...
0 %
17:32
Wastebasket
Pi Presents Editor
Editor for Pi Presents - /home/pi/pp_home/pp_profiles/tstest1
Profile
Show
MediaList
Track
OSC
Tools
Options
Help
Shows
Medialists
Edit Show
Tracks in Selected Medialist
Add
Edit
Up
Down

Start   [start]
radiobuttonshow   [radiobuttonshow]

radiobuttonshow.json

First Track   [first]
vid0.AVI   [vid0]
vid1.AVI   [vid1]
vid2.AVI   [vid2]
vid3.AVI   [vid3]

Fig. 3 The Modified Profile Editor Window

Now we have to connect the playing of the videos to the previously defined touch buttons in the screen.cfg file. In the Shows window select radiobuttonshow, and press Edit Show button in the center.

Select the Controls tab. Delete all the text defining the template controls in the Controls window. Put in the following control text. This window is fully editable.

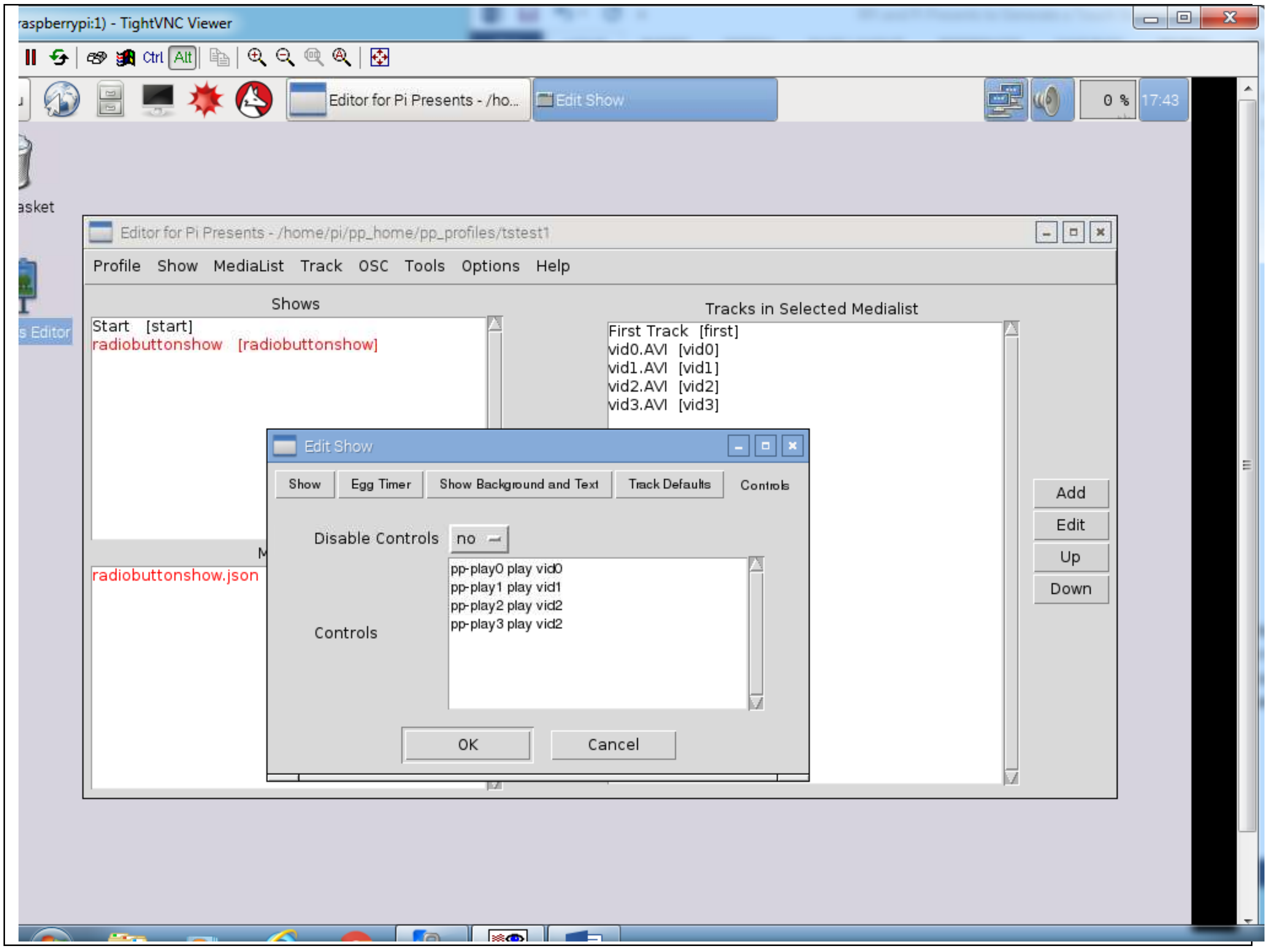
```
pp-play0 play vid0

pp-play1 play vid1

pp-play2 play vid2

pp-play3 play vid3
```

Now the Control tab window should look like Fig.4 below



raspberrypi:1) - TightVNC Viewer
Ctrl
Alt
Editor for Pi Presents - /ho...
Edit Show
0 %
17:43
asket
s Editor

Editor for Pi Presents - /home/pi/pp_home/pp_profiles/tstest1

Shows

Start [start]
radiobuttonshow [radiobuttonshow]

radiobuttonshow.json

Tracks in Selected Medialist

First Track [first]
vid0.AVI [vid0]
vid1.AVI [vid1]
vid2.AVI [vid2]
vid3.AVI [vid3]

Edit Show

Disable Controls: no

Controls:

```
pp-play0 play vid0
pp-play1 play vid1
pp-play2 play vid2
pp-play3 play vid2
```

Fig. 4 Completed Control Tab Window for the Example

Examine the line “pp-play0 play vid0”. This says that when touch button pp-play0 (defined in screen.cfg) is touched play vid0 (defined by the Track Reference name). See section 5.2.4.1 of the PI Presents manual for more detail

That’s it, so shut down the Editor window

5. Running the Profile

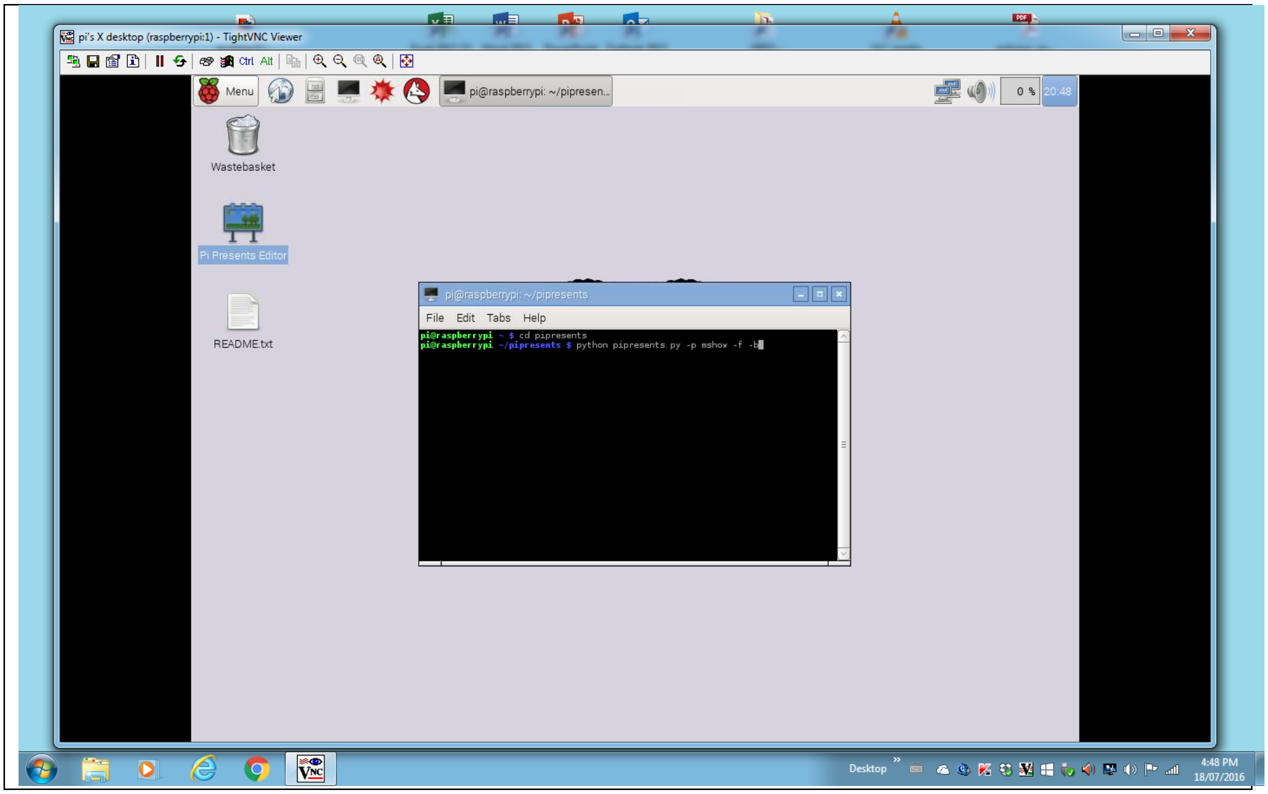
You have to start the profile using a command line. Click on the LXTerminal icon at the top of the Rpi desktop. It is the fourth icon in the tool bar and looks like a small black screen. This provides you with the classic Linux command line widow. You first need to change to the directory that contains the Pi Presents software by entering

```
cd pipresents
```

Now enter the command that should start your profile, as follows

```
python pipresents.py  –p yourprofilename
```

Fig. 5 shows this terminal widow.



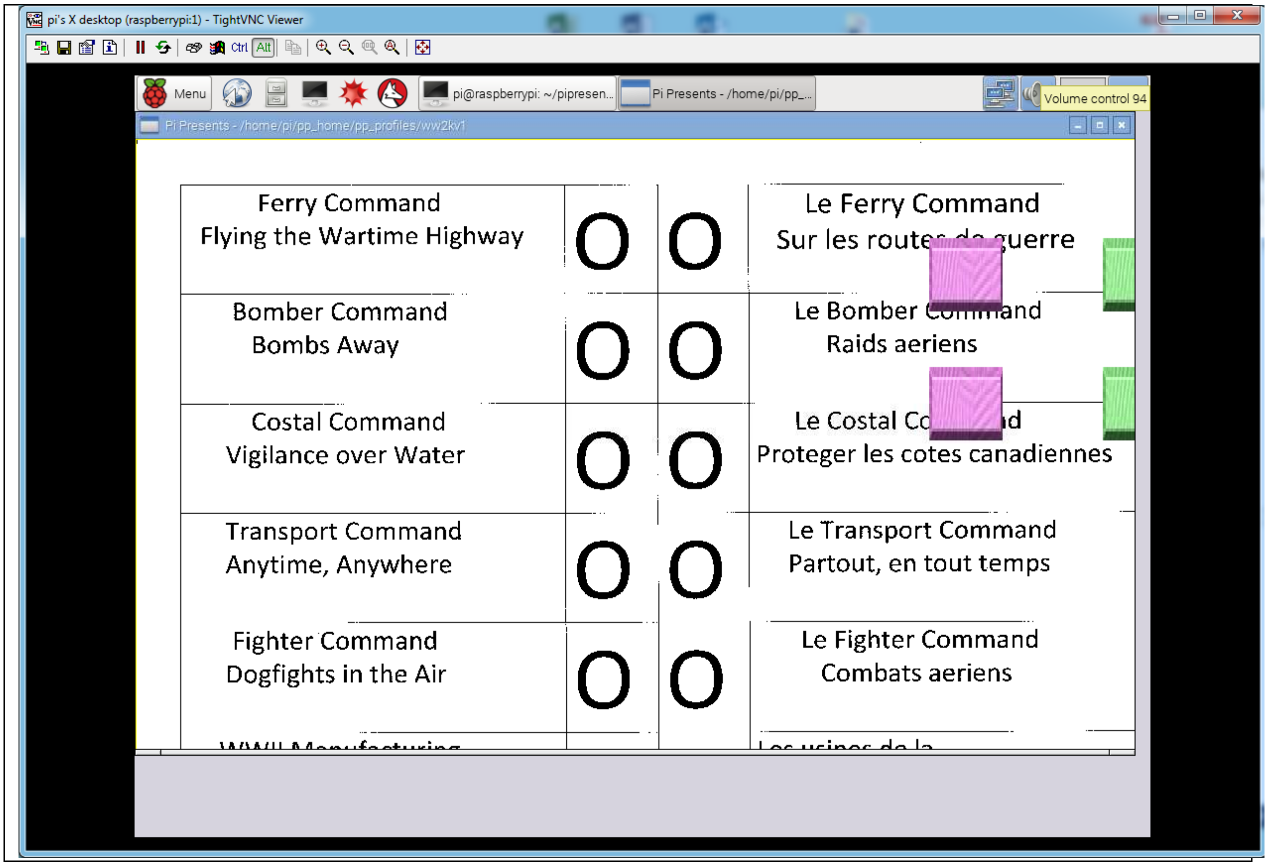
pi's X desktop (raspberrypi:1) - TightVNC Viewer
Menu
pi@raspberrypi: ~/pipresen...
0 %
20:48
Wastebasket
Pi Presents Editor
README.txt
pi@raspberrypi: ~/pipresents
File
Edit
Tabs
Help
Desktop
4:48 PM
18/07/2016

```
pi@raspberrypi ~ $ cd pipresents
pi@raspberrypi ~/pipresents $ python pipresents.py -p mshow -f -b
```

Fig. 5 The LXTerminal window

An error message might appear in the terminal window. The appendix note below may be helpful.

If all goes well you will get a screen with your background overlaid with your touch buttons. Fig. 6 is the display of the example.


pi's X desktop (raspberrypi:1) - TightVNC Viewer
Ctrl
Alt
Menu
pi@raspberrypi: ~/pipresen...
Pi Presents - /home/pi/pp_...
Volume control 94
Pi Presents - /home/pi/pp_home/pp_profiles/ww2kv1

Ferry Command Flying the Wartime Highway	O	O	Le Ferry Command Sur les routes de guerre
Bomber Command Bombs Away	O	O	Le Bomber Command Raids aeriens
Costal Command Vigilance over Water	O	O	Le Costal Command Proteger les cotes canadiennes
Transport Command Anytime, Anywhere	O	O	Le Transport Command Partout, en tout temps
Fighter Command Dogfights in the Air	O	O	Le Fighter Command Combats aeriens
WWII Manufacturing			Les usines de la

Fig.6 Example Touch Screen

The example touch screen is not quite right because the method used to print the screen was a different size than the screen where it was developed. The four touch buttons need to be moved left and up a bit to be overlay the regions that are labeled in the background, then made transparent. Back to the screen.cfg file. A final graphic demonstration of the advantage of developing the profile on the final screen size.

Appendix

The Rpi plays many kinds of still, audio and video files. . A full list of file types is

VIDEO_FILES= ('Video Files','.asf','.avi','.mpg','.mp4','.mpeg','.m2v','.m1v','.vob','.divx','.xvid','.mov','.m4v','.m2p','.mkv','.m2ts','.ts','.mts','.wmv','.webm')

AUDIO_FILES=('Audio files','.mp3','.wav','.ogg','.ogm','.wma','.asf','.mp2')

WEB_FILES=('Web files','.htm','.html')

The file type used by Apple Computer (.mov) is **no**t in this list. Sometimes all you have to do is change the file name extension from .mov to .mp4. But other times you have to resort to the same methods you might need for older video files. See the following note.

NOTE: There is a possible wrinkle with playing old video files. Old ones can be encoded in a way that the RPi cannot handle. An error massage will appear in the LXTerminal window after you try to run the profile. It ends “Have a nice day”. A way around this is to load an open source transcoder called Handbrake (Google it) onto your PC and convert the old video file to a newer format called H.264. This appears to be Handbrake’s default format. This format works with the RPi.