

# ... magari!!!

www.RaccoltineMC.it - www.MC-online.it - www.digiTANTO.it



## #AmighinoRPi

### ... sogno di una notte di quasi estate

di Andrea de Prisco - AdP

O, come scrivevo sulla mia pagina FB, in uno dei tanti post dedicati al giocattolino in questione: "Se puoi sognarlo, puoi farlo". Dovrebbe essere un aforisma di Walt Disney in persona, ispiratore peraltro di tantissimi, entusiasmanti, sogni per molti.

Tutto è iniziato, nel mio caso, in seguito a uno dei miei tanti post leggeri che regolarmente pubblico con hashtag #MinkiateBelle, spesso abbinato a un cautelativo #FotoDalWeb, piazzato proprio per rispondere in anticipo alla domanda più comune e fugare ogni dubbio sul fatto che possa essere farina del mio sacco. Era la foto di un Amiga 500 in miniatura, di sicuro realizzato tramite stampa 3D, con dentro un Raspberry Pi sul quale, evidentemente, girava un emulatore, fine.

Fine???

Pubblicato solo come curiosità simpatica (per i pochi, come me, che non l'avevano mai visto prima) non avrei mai immaginato che sarebbe stata la scintilla in grado di (ri)accendermi più di un interesse. Tanti conti in sospeso, parcheggiati in qualche cassetto della testa, che aspettavano solo il momento giusto per saltare nel mio mondo reale. Tanto per essere chiari,



La «scintilla» colpevole di tutto...

fino a poche settimane fa non avevo mai programmato, né toccato con mano, un Raspberry. Durante lo stopcovid avevo giocherellato con Arduino e già sapevo che "prima o poi" avrei messo il naso anche nel suo fratello maggiore. Per non parlare del fatto che la stampa 3D, che pure mi incuriosiva da sempre, l'avevo vista solo in qualche tutorial di YT, relegando anche questa al capitolo "sì, poi, dopo".

Ritenevo quest'ultima, peraltro, ancora più irraggiungibile visto che non mi sarei mai limitato a stampare oggetti 3D altrui e quindi mi toccava esplorare preventivamente anche il mondo della modellazione 3D, aggiungendo al tutto un ulteriore (e oscuro) livello di complessità. E infatti, inizialmente, questo secondo aspetto l'ho ricacciato nel cassetto mentale dal quale era arbitrariamente schizzato fuori, ma lo sentivo scalpitare ancor più di prima.

Avete presente Jumanji? 🤣

#### Raspberry Pi: sì, ma quale?

Primo dubbio, tanto per ribadire quanto fossi a digiuno della doppia (ancora non tripla) materia: l'aspetto hardware, come al solito in funzione del software. Di RPi completi ne esiste più d'uno, diciamo che oggi il modello base è il 3, disponibile da alcuni anni in più versioni, ma c'è anche un più potente 4 e un mostruoso 5, che se non è un computer-computer, poco gli manca. Inutile dire che i due modelli maggiori scaldano parecchio, se non ho capito male è obbligatoria la ventola (specie per il 5), mentre il 3, con carichi non eccessivamente elevati, come sembra essere l'emulazione Amiga, può funzionare senza problemi perfino liscio, privo di alette di raffreddamento. Bene, per quanto alcuni storceranno il naso, posso dirvi che tenuto acceso con l'emulatore Amiga in funzione per ore, a far rimbalzare la famosa palla, sì e no i chip del Raspberry Pi 3+ utilizzato diventavano tiepidini, toccabili tranquillamente con le dita senza avvertire preoccupanti bollori. Sì, lo so, è un metodo poco scientifico... ma io sono fatto così e non posso farci nulla!

L'ho acquistato direttamente su Melopero.com a circa 38 euro, ai quali vanno aggiunte le spese di spedizio-



Vista d'insieme del «giocattolino» aperto con i componenti elettronici a vista. Notare l'assenza di dissipatori termici sul Raspberry Pi 3+

#### #AmighinoRPi

Prezzo componenti/materiali (05.2024)

Raspberry PI 3+	€ 38,50
Display LCD 5" Touch	€ 43,95
Miniamplificatore audio	€ 5,99
Mini cassa audio	€ 4,99
Trasformatore di isolamento	€ 2,50
Powerbank	€ 18,99
Filamento PLA (approx.)	€ 2,00





L'alimentatore originale, consigliatissimo!

ne. Il totale era comunque inferiore al prezzo su Amazon, dove era venduto a una cinquantina di euri spedizione inclusa, quindi affare fatto! Ho sbagliato a non prendere anche l'alimentatore originario, credendo che potesse andar bene uno qualsiasi... ma questo purtroppo non è vero in quanto il bestiolino in questione (o forse il suo controllore integrato), è piuttosto esigente riguardo all'erogazione di corrente. Alla fine per farlo funzionare senza l'odioso continuo alert (a parer mio spesso eccessivo) di tensione insufficiente ho dovuto scomodare l'alimentatore del Macbook Pro, da non so quanti gigowatt (cit. 🤣) di potenza!

Tornando brevemente alla questione *bollori*, sempre da poco esperto al riguardo, cautelativamente avevo preso anche un box in alluminio tutto aletta, risultato poi essere del tutto inutile... almeno nell'uso con l'emulatore Amiga. Parliamo di un computer di 40 anni fa che, dentro un Raspberry 3, a mio avviso ci "gira" impegnando una minima parte della potenza disponibile. "E ho detto tutto!" (altra cit.)

#### L'emulatore, questo sconosciuto...

Passiamo all'emulatore da utilizzare, altro terreno impervio (almeno per me). Ribadisco che stavo esplorando campi nuovi, ma la cosa in sé non rappresentava un problema, anzi era lo spunto per *studiare* le varie soluzioni, cominciando dal fatto che effettivamente ce n'era più d'una.

Come immaginavo, quella più semplice (detta anche *porta larga*) non mi ha convinto. Esistono alcune distribuzioni Linux specifiche per Raspberry con dentro già l'emulatore quasi-configurato. Dico *quasi* perché non viene mai fornito, per evidenti motivi di copyright, completo delle ROM di sistema Amiga, che sono da recuperare altrove.

Fortunatamente, avevo - anzi ho una licenza regolare di Amiga Forever, dalla quale è possibile scaricare legittimamente le ROM dei vari Amiga, dal 1000 in su.

Dettagli organizzativi a parte, la soluzione pappa pronta della distribuzione Linux, come facilmente immaginabile, non mi ha convinto. Non so di preciso il motivo, ma il fatto stesso di perdere una parte del controllo totale mi dava fastidio. Oltre al fatto che non funziona, comunque, al primo colpo e c'è da combatterci un po' per riuscire a farlo *ragionare*.

Alla fine ho deciso di fare di testa mia: distribuzione Linux standard (del Raspberry, scaricata sempre da Melopero.com) sulla quale ho semplicemente installato Amiberry, che mi risulta essere - ripeto, ne sapevo e continuo a saperne poco! - uno degli emulatori più diffusi.

#### Linux, questo sconosciuto...

Già, perché anche di Linux non ne so moltissimo, non avendolo mai utilizzato per davvero, che in altre parole equivale a dire che non mi ero mai scontrato con lo stesso per risolvere problemi... che subito si sono palesati nella mia prima esperienza RPi. Per cominciare: come *caxxarola* si fa per lanciare all'avvio automaticamente un'app? Chiedo a Nonno Google che le sa tutte?

Nì, perché se è vero che la soluzione (ovviamente) c'è, è altrettanto vero che non è unica e dipende da un sacco di fattori, a cominciare dalla distribuzione Linux utilizzata e dalle infinite possibilità che ognuna mette a disposizione. Parlo in generale, non solo in merito all'app da lanciare all'avvio. Così la prima dritta ricevuta riguardava il fatto di mettere il comando all'interno del file rc.local ma questo purtroppo non è sempre sufficiente se, come nel mio caso in una versione successiva dell'interfaccia utente, che parallelamente stavo buttando giù in Python, doveva partire prima la GUI X cosa che tramite il file in questione a quanto pare non avveniva. Per farla breve, alla fine la soluzione definitiva mi è stata suggerita nei commenti della mia pagina FB da chi, prima di me. s'era scontrato con il non-problema. C'è un'apposita cartella autostart in quella nascosta .config, nella propria cartella utente di home. nella quale inserire file con estensione .desktop che, per semplificare,

lancia l'eseguibile dediderato (python interfaccina.py). Se l'utenza è denominata pi e la cartella di lavoro amy, il file da salvare come interfaccina.desktop nella cartella autostart prima citata è il seguente:

[Desktop Entry] Type=Application Name=interfaccina Exec=/usr/bin/python /home/pi/amy/interfaccina.py Name[it\_IT]=interfaccina

#### Gli altri «pezzettini»

Finora, lato hardware, ho citato solo il Raspberry, come detto segliendo il modello 3+, a mio avviso sufficiente per le emulazioni Amiga. Almeno quelle fin qui esplorate da me. Già perché Amiberry, come immagino altri emulatori, permette di confezionare la macchina da emulare come ci pare, con il processore che vogliamo, tutta la memoria (chip e fast) di cui abbiamo bisogno e anche tanti altri add-on che un tempo, spesso, potevamo solo sognare. Nel nostro caso, ovviamente, ce la caviamo con gualche click nelle tante pagine di configurazione previste. Approfondirò anche questo a tempo debito (se non mi stufo prima!).

Come display ho optato per un 5" touchscreen - qui in alto - collegabile direttamente al Raspberry tramite interfaccia DSI (utilizza un flat cable che finisce dritto dritto su un apposito connettore previsto sulla scheda) in un certo senso un collegamento nativo che ha il vantaggio di funzionare senza installare alcun drive complicato e quindi NON *pesa* sull'elaborazione del Raspberry come succede con quei display che si collegano tramite connettore GPIO.



Il modulo LCD che ho trovato, nelle dimensioni che mi interessavano, ha per certi versi lo svantaggio di essere in formato 16:9 (cosa che naturalmente l'Amiga non era) e se per alcuni è apparso come un errore di progetto, nell'uso risulta più indicato di un display 4:3 per quei giochi o guelle applicazioni, e non sono poche, che utilizzano il formato per l'NTSC. come noto, da 320x200px anziché 320x256px di quello per il PAL. Tradotto: in molti casi il formato 16:9 viene sfruttato se non al 100% almeno al 95% visualizzando realmente la schermata a pieno formato su quel display, mentre su un 4:3 (come molti ricorderanno) creava una corposa banda nera in basso, totalmente inutilizzata. In altre parole, non sono pentito affatto della scelta ... che posso nemmeno ritenere tale (un display 4:3 DSI, da pochi pollici, non sono riuscito a trovarlo nemmeno su Aliexpress, per esser chiari!).

Per l'audio ho optato per un amplificatorino interno collegato a una cassa in miniatura, pescata anche questa su Amazon.

Per semplificare il progetto non ho previsto l'audio stereo, pertanto (pur essendo stereo tanto il Raspberry, quanto l'amplificatore e l'emulatore, oltre che l'Amiga originario) ho usato un solo canale. Anzi, la somma di entrambi su un'unica uscita, altra impostazione prevista da Amiberry.





Qui però sì è palesato un secondo problema (dopo quello dell'avvio automatico dell'interfaccina utente), più di natura hardware, per di più analogica, che software. Collegando sia il Raspberry che l'amplificatorino alla medesima alimentazione si creava, come spesso avviene, un *loop di massa* con pesante - quanto inaccettabile - ronzio. Al solito ho chiesto aiuto sulla pagina FB e la soluzione è arrivata nel consueto millisecondo.

È bastato - si fa presto a dirlo! - interporre un minuscolo *trasformatore di isolamento* (in foto qui sotto) tra l'uscita audio del Raspberry e l'ingresso del piccolo ampli, per sentir sparire ogni disturbo e tirare un profondo sospiro di sollievo. Stavo per buttare tutto all'aria, per fortuna sono stato fermato in tempo!



Per l'alimentazione, visti anche i problemi riscontrati con quella esterna, ho optato per un *powerbank* di potenza sufficiente. Per questi è meno complicato di quanto possa sembrare, in quanto basta sceglierne uno recente in grado di caricare *in scioltezza* i moderni smartphone, anch'essi piuttosto esigenti in corrente, a meno di non aspettare tempi biblici per la ricarica. Ne ho scelto uno di dimensioni tutto sommato ridotte con integrato un display che indica la carica residua e soprattutto un pulsante di alimentazione per pilotare da questo l'accensione del tutto. Per lo spegnimento, no, meglio procedere da Linux con il più salubre e solito *Chiudi sessione* e poi *Arresta*.

#### Il primo mockup...

... faceva pena, non fatico ad ammetterlo. Era in cartoncino, pensando poi di riprodurlo in materiale plastico da incollare (avevo già acquistato in un negozio di modellismo un foglio di stirene) e non era altro che una serie di facce appena sagomate per inglobare un po' tutto. Ma ancora ero distante da una soluzione accettabile.



Mi è stato suggerito, sempre dalla mia pagina FB e sempre poche settimane fa, di *pensare* alla stampa 3D per realizzare qualcosa di meglio e che potesse in qualche modo richiamare la sua natura Amiga, caratteristica totalmente assente nel primo mockup, come ben testimonia la foto qui sopra.

Il problema, al solito, che di stampa 3D ne sapevo poco... ma *Jumanji* non dà tregua e i minacciosi colpi di tamburo si facevano sempre più insistenti!

Così, incoraggiato da alcuni, mi sono lanciato anche in questa *avventura nell'avventura* provando a capirne qualcosa di più, in primis, della modellazione 3D. Purtroppo (o per fortuna...) non potevo né volevo utilizzare un progettino già disponibile, avendo necessità troppo specifiche: l'unica parte, diciamo, standard di tutto il progetto era il Raspberry, tutto il resto era troppo personalizzato e quindi difficilmente avrei trovato - ma ripeto, non ho nemmeno provato a cercarlo - qualcosa già bell'e fatto!



#### Arriviamo alla modellazione 3D

Ho scelto di utilizzare Onshape che, inaspettatamente, è un software che - adooooorooo - si utilizza interamente online, senza installare nulla sul proprio PC/Tablet/smartphone. La sua struttura è perfino responsive quindi entro certi limiti può essere utilizzato dappertutto tramite browser. Viene solo fatto un check su quest'ultimo ma, naturalmente, i vari Chrome, Edge, Safari, ecc. sono tutti pienamente compatibili. Onshape non è un giocattolino, tutt'altro! Vi dico solo che per l'utilizzo professionale è previsto un canone di 1500 dollari l'anno e la versione free non ha limiti particolari, se non per il fatto che i progetti salvati sulla loro piattaforma (unica modalità prevista) sono automaticamente visibili a tutti gli altri utenti, ovvero sono di pubblico dominio. Nel mio caso, non avendo questo lavoretto attese particolari (era mia intenzione sin da subito, come poi è avvenuto, rilasciare gratuitamente il tutto a beneficio di tutti) la cosa in sé non rappresentava un problema: ho aderito senza indugio.

Bene, ma come si modella un oggetto 3D con gli applicativi di questo tipo? Per semplificare, anche troppo, si parte sempre da un disegno 2D (ad esempio l'oggetto in pianta) e da questo si estrude il volume per aggiungere la terza dimensione. Volendo realizzare un (inesistente) Amiga *all-inone* con le sembianze dello storico monitor Commodore 1084S (prima versione) sono partito da alcune foto di quest'ultimo, riproducendo il suo



profilo, dalla forma inconfondibile... per chi sa di cosa stiamo parlando!



Con un click o poco più si trasforma la figura 2D appena tracciata in un solido pieno e successivamente, con meccanismi analoghi, prima si vuota l'interno (indicando la faccia coinvolta da cui *scavare* e lo spessore da lasciare, nel mio caso 1,5 mm).

Gli step a seguire, continuando a



semplificare, non sono altro che aggiunta o sottrazione di altri volumi, come per le porte laterali o per creare l'alloggiamento del display.



Naturalmente non siamo che all'1-2 per cento del percorso, la sto semplificando tanto altrimenti staremmo qui giorni e giorni, io a scrivere, voi a leggere, per illustrare in dettaglio tutti i passaggi... ma i meccanismi utilizzati sono praticamente tutti dello stesso tipo. Anche per costruire l'alloggiamento del powerbank, le griglie laterali (più che altro estetiche) e quella superiore che include i meccanismi di aggancio per l'altoparlante utilizzato, che dispone di un suo volume proprio: in pratica è una cassa acustica in miniatura.



#### Gli step successivi

Capito, più o meno, come guidare un CAD 3D (cosa che, ripeto, non avevo mai fatto prima di questa esperienza) poteva diventare realtà anche la stampa 3D, materia che altrettanto ignoravo totalmente, se non per sentito dire. Ad esempio una delle prime cose che avevo sottovalutato è che la progettazione 3D stessa, se il pezzo deve essere stampato con una macchina FDM (*Fused Deposition Modeling*, a filamento) va pensato nel giusto modo perché bisogna tenere conto dei vincoli specifici di tale tecnica. Non solo: anche il posizionamento



del pezzo sul piano di stampa può incidere sul risultato, oltre al fatto che variano di conseguenza anche i supporti aggiunti automaticamente (e obbligatoriamente) per produrre la stampa 3D. Ad esempio se una griglia bucherellata è stampata in piano viene bene, se la stampiamo in verticale non è detto che i forellini vengano tutti uguali (e precisi). Per farla breve ho dovuto sperimentare strada facendo diverse possibilità cercando di ottenere il risultato migliore, purtroppo non con il minimo sforzo. Per questo motivo le griglie sono tutti oggetti separati (quindi poi da incollare nelle sedi previste) e la faccia posteriore è separata dal resto proprio per evitare che fossero necessari



un'infinità di supporti interni, allungando di conseguenza anche i tempi di stampa. Ovviamente separare le parti spesso (non sempre) richiede anche di prevedere appositi rinforzi interni, da utilizzare per facilitare l'incollaggio o, quando possibile, l'utilizzo di micro viti per tenere unite le parti.

#### Abbiamo finito?

Macché! L'ultima *fatica* ha riguardato la finitura esterna: stuccatura, rasatura, trattamento con primer e verniciatura (spray).

Anche perché volevo una colorazione Amiga e, nonostante non sia esatta al 100%, il bianco crema utilizzato direi che ci assomiglia abbastanza. E poi gli Amiga variavano colore col tempo... chi lo stabilisce qual è lo stato corretto di invecchiamento?









#### L'interfaccina utente

Siamo guasi arrivati alla fine, dobbiamo solo dare uno squardo all' aspetto software (nel quale, tutto sommato, mi sono compiaciuto).

Ovviamente anche questa non è interamente farina del mio sacco, in quanto anche con Python ero alle prime armi - a differenza del resto comunque non totalmente a digiuno! ma soprattutto conoscevo poco le librerie utilizzate, principalmente tkinter per l'interfaccia vera e propria, e

Una roba che in HTML/PHP/ecc. avrei impiegato pochi millisecondi per buttarla giù, ma con Python condito di OOP è risultato essere un pelino più complicato del previsto. Ma sono stato aiutato anche in questo, come dichiarato anche nella pagina di ringraziamento che ho inserito nel codice.

In pratica su Amiberry ho creato tante configurazioni quante sono le icone presenti, ognuna corrispondente a un file nella cartella conf predisposta dall'emulatore. Per semplicità ho chiamato nello stesso modo il

file di configurazione (ad esempio FS.uae) e la corrispondente icona (FS.png) nel caso del Flight Simulator (FS, appunto). Come immaginabile posso rilasciare solo il codice in forma basica & testuale, chi volesse realizzare un #AmighinoRPi, anche vagamente simile a guesto, dovrà completarlo secondo le proprie necessità.

#### Concludendo...

natural-

oltre a

Cosa altro aggiungere, se non ribadire ancora una volta (come già fatto sulla mia pagina FB) che si è trattato di un'esperienza entusiasmante, ben oltre le più... catastrofiche aspettative. Già, non pensavo affatto di riuscirci, per quante erano le incognite e, soprattutto, le mie lacune di cui ero ben conscio.

Ci tengo a sottolineare ancora una volta che si è trattato di un lavoro di squadra: non credo infatti che sarei riuscito a portarlo a termine, senza l'aiuto di molti di voi.

#### GRAZIE!

```
import tkinter as tk
from tkinter import *
from tkinter import ttk
import svs
import os
import shlex, subprocess
import pygame
pygame.init()
amigray="#aaaaaa"
amigabluet="#6e8ebe"
sound=False
class SeaofBTCapp(tk.Tk):
  def __init__(self, *args, **kwargs):
    tk.Tk.__init__(self, *args, **kwargs)
    self.attributes("-fullscreen", True)
    self.container = tk.Frame(self)
    self.container.pack(fill="both", expand=True)
    self.current frame = None
    self.configure(background=amigray)
    self.show frame(Page 00)
  def show frame(self, new frame class):
    if self.current frame:
       self.current frame.destroy()
    self.current frame = new frame class(self.container, controller=self)
    self.current frame.columnconfigure(0, weight=1)
    self.current frame.columnconfigure(1, weight=1)
    self.current frame.columnconfigure(2, weight=1)
    self.current frame.configure(background=amigabluet)
    self.current frame.pack(fill="both", expand=True)
class Page_00(tk.Frame):
  def __init__(self, parent, controller):
    tk.Frame.__init__(self,parent)
    # ------ def button function ------
    def run_app(fileuae):
       global sound
       if sound:
         pygame.mixer.music.pause()
       subprocess.run(["/home/pi/amy/amiberry", "-f", "/home/pi/amy/conf/"+fileuae])
       if sound:
         pygame.mixer.music.unpause()
    def playstop_app():
       global sound
       if sound:
         pygame.mixer.music.pause()
         sound = not sound
       else:
         pygame.mixer.music.unpause()
         sound = not sound
```

# ------ label conf ------

self.barra = tk.PhotoImage(file='/home/pi/amy/puls/barra.png') label = tk.Button(self, command=lambda: controller.show frame(Page GR), image=self.barra, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) #label = ttk.Label(self, text="#AmighinoRPi", background=amigabluet, foreground='black', font=LARGE\_FONT) label.grid(column=0, row=0, columnspan=3, sticky=W, padx=73) # ----- move button conf ----- command=playstop app, self.photo0 = tk.PhotoImage(file='/home/pi/amy/puls/file0.png') button = tk.Button(self, command=self.quit, image=self.photo0, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button.grid(column=0, row=0, columnspan=3, sticky=W) self.photoD = tk.PhotoImage(file='/home/pi/amy/puls/fileD.png') buttonD = tk.Button(self, command=lambda: controller.show frame(Page 01), image=self.photoD, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonD.grid(column=0, row=0, columnspan=3, sticky=E) self.photoST = tk.PhotoImage(file='/home/pi/amv/puls/fileST.png') buttonST = tk.Button(self, image=self.photoST, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonST.grid(column=0, row=0, columnspan=3, sticky=E, padx=73) self.photoA = tk.PhotoImage(file='/home/pi/amy/puls/fileA.png') buttonA = tk.Button(self, command=playstop\_app, image=self.photoA, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonA.grid(column=0, row=0, columnspan=3, sticky=E, padx=146) # ------ app button conf -----self.photo1 = tk.PhotoImage(file='/home/pi/amy/puls/A1.png') button1 = tk.Button(self, command = lambda: run app("A1.uae"), image=self.photo1, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button1.grid(column=0, row=1) self.photo2 = tk.PhotoImage(file='/home/pi/amy/puls/A2.png') button2 = tk.Button(self, command=lambda: run\_app("A2.uae"), image=self.photo2, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button2.grid(column=1, row=1) self.photo3 = tk.PhotoImage(file='/home/pi/amy/puls/A3.png') button3 = tk.Button(self, command=lambda: run app("A3.uae"), image=self.photo3, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button3.grid(column=2, row=1) self.photo4 = tk.PhotoImage(file='/home/pi/amy/puls/A4.png') button4 = tk.Button(self, command=lambda: run\_app("A4.uae"), image=self.photo4, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button4.grid(column=0, row=2) self.photo5 = tk.PhotoImage(file='/home/pi/amy/puls/A5.png') button5 = tk.Button(self, command=lambda: run\_app("A5.uae"), image=self.photo5, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button5.grid(column=1, row=2) self.photo6 = tk.PhotoImage(file='/home/pi/amy/puls/A6.png') button6 = tk.Button(self, command=lambda: run\_app("A6.uae"), image=self.photo6, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button6.grid(column=2, row=2) class Page 01(tk.Frame):

def init (self, parent, controller): tk.Frame, init (self.parent) # ------ def button function -----def run app(fileuae): global sound if sound: pygame.mixer.music.pause() subprocess.run(["/home/pi/amy/amiberry", "-f", "/home/pi/amy/conf/"+fileuae]) if sound: pygame.mixer.music.unpause() def playstop app(): global sound if sound: pygame.mixer.music.pause() sound = not sound else: pygame.mixer.music.unpause() sound = not sound # ------ label conf -----self.barra = tk.PhotoImage(file='/home/pi/amy/puls/barra.png') label = tk.Button(self, command=lambda: controller.show\_frame(Page\_00), image=self.barra, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) label.grid(column=0, row=0, columnspan=3, sticky=W, padx=73) # ------ move button conf -----self.photo0 = tk.PhotoImage(file='/home/pi/amy/puls/file0.png') button = tk.Button(self, command=self.quit, image=self.photo0, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button.grid(column=0, row=0, columnspan=3, sticky=W) self.photoD = tk.PhotoImage(file='/home/pi/amy/puls/fileD.png') buttonD = tk.Button(self, command=lambda: controller.show frame(Page 02), image=self.photoD, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonD.grid(column=0, row=0, columnspan=3, sticky=E) self.photoST = tk.PhotoImage(file='/home/pi/amy/puls/fileS.png') buttonST = tk.Button(self, command=lambda: controller.show frame(Page 00), image=self.photoST, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonST.grid(column=0, row=0, columnspan=3, sticky=E, padx=73) self.photoA = tk.PhotoImage(file='/home/pi/amy/puls/fileA.png') buttonA = tk.Button(self, command=playstop\_app, image=self.photoA, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonA.grid(column=0, row=0, columnspan=3, sticky=E, padx=146) # ----- app button conf -----self.photo1 = tk.PhotoImage(file='/home/pi/amy/puls/B1.png') button1 = tk.Button(self, command = lambda: run app("B1.uae"), image=self.photo1, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button1.grid(column=0, row=1) self.photo2 = tk.PhotoImage(file='/home/pi/amy/puls/B2.png') button2 = tk.Button(self, command=lambda: run\_app("B2.uae"), image=self.photo2, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button2.grid(column=1, row=1)

```
self.photo3 = tk.PhotoImage(file='/home/pi/amy/puls/B3.png')
     button3 = tk.Button(self, command=lambda: run app("B3.uae"), image=self.photo3, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button3.grid(column=2, row=1)
     self.photo4 = tk.PhotoImage(file='/home/pi/amy/puls/B4.png')
button4 = tk.Button(self, command=lambda: run_app("B4.uae"), image=self.photo4, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button4.grid(column=0, row=2)
     self.photo5 = tk.PhotoImage(file='/home/pi/amy/puls/B5.png')
     button5 = tk.Button(self, command = lambda: run app("B5.uae"), image=self.photo5, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button5.grid(column=1, row=2)
     self.photo6 = tk.PhotoImage(file='/home/pi/amv/puls/B6.png')
button6 = tk.Button(self, command=lambda: run_app("B6.uae"), image=self.photo6, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button6.grid(column=2, row=2)
class Page 02(tk.Frame):
  def init (self, parent, controller):
     tk.Frame. init (self,parent)
     # ------ def button function ------
     def run app(fileuae):
        global sound
        if sound:
          pygame.mixer.music.pause()
        subprocess.run(["/home/pi/amy/amiberry", "-f", "/home/pi/amy/conf/"+fileuae])
        if sound:
          pygame.mixer.music.unpause()
     def playstop app():
        global sound
        if sound:
          pygame.mixer.music.pause()
          sound = not sound
        else.
          pygame.mixer.music.unpause()
          sound = not sound
     # ------ label conf ------
     self.barra = tk.PhotoImage(file='/home/pi/amy/puls/barra.png')
label = tk.Button(self, command=lambda: controller.show_frame(Page_00), image=self.barra, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     label.grid(column=0, row=0, columnspan=3, sticky=W, padx=73)
     # ------ move button conf ------
     self.photo0 = tk.PhotoImage(file='/home/pi/amy/puls/file0.png')
     button = tk.Button(self, command=self.quit, image=self.photo0, borderwidth=0, highlightthickness=0, bg=amigray,
activebackground=amigray, border=0)
     button.grid(column=0, row=0, columnspan=3, sticky=W)
     self.photoDT = tk.PhotoImage(file='/home/pi/amy/puls/fileD.png')
buttonDT = tk.Button(self, command=lambda: controller.show_frame(Page_03), image=self.photoDT, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     buttonDT.grid(column=0, row=0, columnspan=3, sticky=E)
```

```
self.photoS = tk.PhotoImage(file='/home/pi/amv/puls/fileS.png')
    buttonS = tk.Button(self, command=lambda; controller.show frame(Page 01), image=self.photoS, borderwidth=0.
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    buttonS.grid(column=0, row=0, columnspan=3, sticky=E, padx=73)
    self.photoA = tk.PhotoImage(file='/home/pi/amy/puls/fileA.png')
    buttonA = tk.Button(self, command=playstop_app, image=self.photoA, borderwidth=0, highlightthickness=0,
bg=amigray, activebackground=amigray, border=0)
    buttonA.grid(column=0, row=0, columnspan=3, sticky=E, padx=146)
    # ------ app button conf ------
    self.photo1 = tk.PhotoImage(file='/home/pi/amy/puls/C1.png')
    button1 = tk.Button(self, command = lambda: run app("C1.uae"), image=self.photo1, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button1.grid(column=0, row=1)
    self.photo2 = tk.PhotoImage(file='/home/pi/amy/puls/C2.png')
    button2 = tk.Button(self, command=lambda: run app("C2.uae"), image=self.photo2, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button2.grid(column=1, row=1)
    self.photo3 = tk.PhotoImage(file='/home/pi/amy/puls/C3.png')
    button3 = tk.Button(self, command=lambda: run app("C3.uae"), image=self.photo3, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button3.grid(column=2, row=1)
    self.photo4 = tk.PhotoImage(file='/home/pi/amy/puls/C4.png')
    button4 = tk.Button(self, command=lambda: run app("C4.uae"), image=self.photo4, borderwidth=0.
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button4.grid(column=0, row=2)
    self.photo5 = tk.PhotoImage(file='/home/pi/amy/puls/C5.png')
    button5 = tk.Button(self, command = lambda: run_app("C5.uae"), image=self.photo5, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button5.grid(column=1, row=2)
    self.photo6 = tk.PhotoImage(file='/home/pi/amy/puls/C6.png')
    button6 = tk.Button(self, command=lambda: run app("C6.uae"), image=self.photo6, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    button6.grid(column=2, row=2)
class Page 03(tk.Frame):
  def __init__(self, parent, controller):
    tk.Frame.__init__(self,parent)
    # ------ def button function ------
    def run app(fileuae):
       global sound
       if sound:
         pygame.mixer.music.pause()
       subprocess.run(["/home/pi/amy/amiberry", "-f", "/home/pi/amy/conf/"+fileuae])
       if sound:
         pygame.mixer.music.unpause()
    def playstop app():
       global sound
       if sound:
         pygame.mixer.music.pause()
         sound = not sound
       else:
```

pygame.mixer.music.unpause() sound = not sound # ------ label conf -----self.barra = tk.PhotoImage(file='/home/pi/amv/puls/barra.png') label = tk.Button(self, command=lambda: controller.show\_frame(Page\_00), image=self.barra, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) label.grid(column=0, row=0, columnspan=3, sticky=W, padx=73) # ------ move button conf -----self.photo0 = tk.PhotoImage(file='/home/pi/amy/puls/file0.png') button = tk.Button(self, command=self.quit, image=self.photo0, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button.grid(column=0, row=0, columnspan=3, sticky=W) self.photoDT = tk.PhotoImage(file='/home/pi/amy/puls/fileDT.png') buttonDT = tk.Button(self, image=self.photoDT, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonDT.grid(column=0, row=0, columnspan=3, sticky=E) self.photoS = tk.PhotoImage(file='/home/pi/amy/puls/fileS.png') buttonS = tk.Button(self, command=lambda: controller.show\_frame(Page\_02), image=self.photoS, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonS.grid(column=0, row=0, columnspan=3, sticky=E, padx=73) self.photoA = tk.PhotoImage(file='/home/pi/amy/puls/fileA.png') buttonA = tk.Button(self, command=playstop\_app, image=self.photoA, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) buttonA.grid(column=0, row=0, columnspan=3, stickv=E, padx=146) # ------ app button conf -----self.photo1 = tk.PhotoImage(file='/home/pi/amy/puls/D1.png') button1 = tk.Button(self, command = lambda: run\_app("D1.uae"), image=self.photo1, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button1.grid(column=0, row=1) self.photo2 = tk.PhotoImage(file='/home/pi/amy/puls/D2.png') button2 = tk.Button(self, command=lambda: run\_app("D2.uae"), image=self.photo2, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button2.grid(column=1, row=1) self.photo3 = tk.PhotoImage(file='/home/pi/amv/puls/D3.png') button3 = tk.Button(self, command=lambda: run app("D3.uae"), image=self.photo3, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button3.grid(column=2, row=1) self.photo4 = tk.PhotoImage(file='/home/pi/amy/puls/D4.png') button4 = tk.Button(self, command=lambda: run\_app("D4.uae"), image=self.photo4, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button4.grid(column=0, row=2) self.photo5 = tk.PhotoImage(file='/home/pi/amy/puls/D5.png') button5 = tk.Button(self, command = lambda: run app("D5.uae"), image=self.photo5, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button5.grid(column=1, row=2) self.photo6 = tk.PhotoImage(file='/home/pi/amy/puls/D6.png') button6 = tk.Button(self, command=lambda: run\_app("D6.uae"), image=self.photo6, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0) button6.grid(column=2, row=2)

class Page\_GR(tk.Frame):

```
def init (self, parent, controller):
     tk.Frame, init (self.parent)
     # ------ def button function ------
    def playstop app():
       global sound
       if sound:
          pygame.mixer.music.pause()
          sound = not sound
       else:
          pygame.mixer.music.unpause()
          sound = not sound
     # ------ label conf ------
     self.barra = tk.PhotoImage(file='/home/pi/amy/puls/barra.png')
label = tk.Button(self, command=lambda: controller.show_frame(Page_00), image=self.barra, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     #label = ttk.Label(self. text="#AmighinoRPi", background=amigabluet, foreground='black', font=LARGE_FONT)
    label.grid(column=0, row=0, columnspan=3, sticky=W, padx=73)
     # ------ move button conf ------ command=playstop app.
     self.photo0 = tk.PhotoImage(file='/home/pi/amy/puls/file0.png')
button = tk.Button(self, command=self.quit, image=self.photo0, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button.grid(column=0, row=0, columnspan=3, sticky=W)
     self.photoD = tk.PhotoImage(file='/home/pi/amy/puls/fileD.png')
     buttonD = tk.Button(self, command=lambda: controller.show_frame(Page_01), image=self.photoD,
borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
    buttonD.grid(column=0, row=0, columnspan=3, sticky=E)
     self.photoST = tk.PhotoImage(file='/home/pi/amy/puls/fileST.png')
     buttonST = tk.Button(self, image=self.photoST, borderwidth=0, highlightthickness=0, bg=amigray,
activebackground=amigray, border=0)
    buttonST.grid(column=0, row=0, columnspan=3, sticky=E, padx=73)
     self.photoA = tk.PhotoImage(file='/home/pi/amy/puls/fileA.png')
buttonA = tk.Button(self, command=playstop_app, image=self.photoA, borderwidth=0, highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     buttonA.grid(column=0, row=0, columnspan=3, sticky=E, padx=146)
    #-----app button conf ------ command=lambda: controller.show frame(Page 00),
     self.photo1 = tk.PhotoImage(file='/home/pi/amy/puls/greetings.png')
     button1 = tk.Button(self, command=lambda: controller.show frame(Page 00), image=self.photo1, borderwidth=0,
highlightthickness=0, bg=amigray, activebackground=amigray, border=0)
     button1.grid(column=0, row=1, columnspan=3,)
pygame.mixer.init()
pygame.mixer.music.load("/home/pi/amy/audio.mp3")
pygame.mixer.music.set volume(1.0)
pygame.mixer.music.play(loops=-1)
if sound==False:
  pygame.mixer.music.pause()
app = SeaofBTCapp()
app.mainloop()
```