GUI Code/images Version 0.5

import Tkinter

class databasePanel:

def \_\_init\_\_(self, frameL):

self.search = LabelFrame(frameL,text="search")

self.search.pack(fill=X)

self.sIn= Entry(self.search)

self.sIn.pack(fill=X)

self.searchButton = Button(self.search, text="go", command=self.updateSearch)

self.searchButton.pack(fill=X)

#DATABASE LIST

self.database = LabelFrame(frameL, text="Database")

self.database.pack(fill=X)

self.scrollbarL = Scrollbar(self.database)

self.mylistL = Listbox(self.database, yscrollcommand = self.scrollbarL.set, selectmode=SINGLE)

for line in range(100):

self.mylistL.insert(END, "This is line number " + str(line))

#CARDVIEW TEXT

self.cardview= LabelFrame(frameL, text="cardview")

self.cardview.pack(fill=X)

self.mylistL.pack( side = LEFT, fill = BOTH )

self.scrollbarL.pack( side = LEFT, fill=Y )

self.cardupdatebutton=Button(self.database, text="select",command=self.updateCard)

self.scrollbarL.config( command = self.mylistL.yview )

self.cardupdatebutton.pack()

self.card = Label(self.cardview,text="")

self.card.pack(fill=X)

def updateSearch(self):

self.mylistL.delete(0,self.mylistL.size())

newList= ['1','2','3','4']

for x in newList:

self.mylistL.insert(END,x)

def updateCard(self):

self.sel=self.mylistL.curselection()

#print(self.sel)

#print(self.mylistL.get(self.sel))

self.Cdata=self.mylistL.get(self.sel)

self.card.config(text=self.Cdata)

class collectionPanel:

def \_\_init\_\_(self, frameR):

self.colV=LabelFrame(frameR, text="collection")

self.colV.pack()

self.scrollbarR = Scrollbar(self.colV)

self.mylistR = Listbox(self.colV, yscrollcommand = self.scrollbarR.set )

for line in range(100):

self.mylistR.insert(END, "This is line number " + str(line))

self.mylistR.pack( side = LEFT, fill = BOTH )

self.scrollbarR.pack( side = LEFT, fill=Y )

self.scrollbarR.config( command = self.mylistR.yview )

self.buttons=Frame(self.colV)

self.buttons.pack(side=LEFT,fill=Y)

self.addC=Button(self.buttons, text="add Card", command=lambda:self.addCard)

self.remC=Button(self.buttons, text="remove Card",command=lambda:self.removeCard)

self.selC=Button(self.buttons, text="select",command=lambda:self.updateCard)

self.addC.pack(fill=X)

self.remC.pack(fill=X)

self.selC.pack(fill=X)

def hide(self):

self.colV.pack\_forget()

def show(self):

self.colV.pack()

def updateCard(self, databasePanel):

pass

def addCard(self, databasePanel):

pass

def removeCard(self, databasePanel):

pass

class deckPanel:

def \_\_init\_\_(self,frameR):

self.deckV=LabelFrame(frameR, text="deck")

self.deckV.pack(side=LEFT)

self.deckScroll = Scrollbar(self.deckV)

self.deckScroll.pack( side = LEFT, fill=Y )

self.deckList = Listbox(self.deckV, yscrollcommand = self.deckScroll.set )

for line in range(2):

self.deckList.insert(END, "This is line number " + str(line))

self.deckList.pack( side = LEFT, fill = X )

self.deckScroll.config( command = self.deckList.yview )

self.dcScroll=Scrollbar(self.deckV)

self.dcList = Listbox(self.deckV, yscrollcommand = self.dcScroll.set)

for line in range(60):

self.dcList.insert(END, "card"+str(line))

self.dcList.pack(side=LEFT,fill=X)

self.dcScroll.pack(side=LEFT,fill=Y)

self.dcScroll.config(command=self.dcList.yview)

self.buttons=Frame(self.deckV)

self.buttons.pack(side=LEFT,fill=Y)

self.addC=Button(self.buttons, text="add Card", command=lambda:self.addCard)

self.remC=Button(self.buttons, text="remove Card",command=lambda:self.removeCard)

self.selC=Button(self.buttons, text="select",command=lambda:self.updateCard)

self.addD=Button(self.buttons, text="new Deck", command=lambda:self.addDeck)

self.remD=Button(self.buttons, text="delete Deck", command=lambda:self.removeDeck)

self.addC.pack(fill=X)

self.remC.pack(fill=X)

self.selC.pack(fill=X)

self.addD.pack(fill=X)

self.remD.pack(fill=X)

def hide(self):

self.deckV.pack\_forget()

def show(self):

self.deckV.pack()

def updateCard(self):

self.sel=self.dcList.curselection()

#print(self.sel)

#print(self.mylistL.get(self.sel))

self.Cdata=self.dcList.get(self.sel)

self.car.config(text=self.Cdata)

def addCard(self, databasePanel):

pass

def removeCard(self, databasePanel):

pass

def addDeck(self):

pass

def removeDeck(self):

pass

class adminPanel:

def \_\_init\_\_(self,frameR):

self.admV=LabelFrame(frameR, text="admin")

self.admV.pack()

self.tools=LabelFrame(self.admV, text="admin tools")

self.tools.pack()

self.cName=Entry(self.tools)

self.cColor=Entry(self.tools)

self.cCost=Entry(self.tools)

self.cStr=Entry(self.tools)

self.cHP=Entry(self.tools)

self.cAbility=Entry(self.tools)

self.cFlavor=Entry(self.tools)

self.cName.pack()

self.cColor.pack()

self.cStr.pack()

self.cHP.pack()

self.cAbility.pack()

self.cFlavor.pack()

self.tools.pack\_forget()

self.pwLogin=LabelFrame(self.admV,text="login")

self.pwLogin.pack()

self.userIDin=Entry(self.pwLogin)

self.pwIn=Entry(self.pwLogin)

self.userIDin.pack()

self.pwIn.pack()

self.loginButton = Button(self.pwLogin, text="Login", command = lambda:

self.login(self.userIDin.get(), self.pwIn.get()))

self.loginButton.pack(side=LEFT)

def hide(self):

self.admV.pack\_forget()

def show(self):

self.admV.pack()

def updateCard(self, databasePanel):

pass

def newCard(self):

pass

def delCard(self):

pass

def adduser(self):

pass

def deluser(self):

pass

def login(self, usrID, pw):

#if info():

self.tools.pack()

self.pwLogin.pack\_forget()

class rightPanel:

def \_\_init\_\_(self,root):

self.frameR = Frame(root)

self.frameR.pack(side = RIGHT)

self.buttonList = LabelFrame(self.frameR, text="View Mode")

self.buttonList.pack(fill=X)

def colButton():

self.cp.show()

self.dp.hide()

self.ap.hide()

self.cButton = Button(self.buttonList, text="collection", command = colButton)

self.cButton.pack(side=LEFT)

def deckButton():

self.cp.hide()

self.dp.show()

self.ap.hide()

self.dButton = Button(self.buttonList, text="deck", command = deckButton)

self.dButton.pack(side=LEFT)

def admButton():

self.cp.hide()

self.dp.hide()

self.ap.show()

self.aButton = Button(self.buttonList, text="admin", command = admButton)

self.aButton.pack(side=LEFT)

#self.dp.deckV.pack()

#self.cp.colV.pack()

#self.ap.admV.pack()

self.dp= deckPanel(self.frameR)

self.cp= collectionPanel(self.frameR)

self.ap= adminPanel(self.frameR)

self.dp.hide()

self.ap.hide()

def refresh(self):

pass

from Tkinter import \*

root = Tk()

frameL = Frame(root)

frameL.pack(side = LEFT, fill=BOTH)

data= databasePanel(frameL)

frameR=rightPanel(root)

mainloop()

