Kivy Ultimate Notes:

# Start your app

from kivy.app import App

from kivy.uix.label import Label

class Myapp(App):

def build(self):

return Label(text=**'Hello World'**)

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

# Create BoxLayout and widgets

from kivy.app import App

from kivy.uix.label import Label

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

class Myapp(App):

def build(self):

layout = BoxLayout(orientation=**'vertical'**) *# arrange widgets vertically*

label1 = Label(text=**'Label 1'**) *# create a label widget*

label2 = Button(text=**'Button 2'**) *# create a button widget*

layout.add\_widget(label1)

layout.add\_widget(label2)

return layout

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

# Positioning and Sizing

from kivy.app import App

from kivy.uix.label import Label

from kivy.uix.floatlayout import FloatLayout

from kivy.uix.button import Button

class Myapp(App):

def build(self):

layout = FloatLayout()

label1 = Button(text=**'Label 1'**, size\_hint=(.5, .5), pos\_hint= {**'center\_x'**:.3, **'center\_y'**:.7})

*# size\_hint(horizontal length, vertical length) (0~1) to set the size*

*# pos\_hint {'center\_x':0.5, 'center\_y':0.5}) meaning in the center,*

*# increase x to right, decrease x to left, increase y to up, decrease y to down*

label2 = Label(text=**'Label 2'**, size\_hint=(.1, .1), pos\_hint= {**'center\_x'**:.5, **'center\_y'**:.1})

layout.add\_widget(label1)

layout.add\_widget(label2)

return layout

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

# Button events

from kivy.app import App

from kivy.uix.label import Label

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

class MyLayout(BoxLayout):

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

super().\_\_init\_\_() *# inheritance*

self.button = Button(text=**'press me'**)

self.button.bind(on\_press = self.new\_label)

self.add\_widget(self.button)

def new\_label(self,button): *# this method needs second para as it is called by the button*

self.label = Label(text=**'A new label'**)

self.add\_widget(self.label)

self.remove\_widget(self.button)

class Myapp(App):

def build(self):

return MyLayout()

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

# Trace your cursor position (Draw shape)

from kivy.app import App

from kivy.uix.label import Label

class MyLabel(Label):

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

super().\_\_init\_\_()

self.text = text

*# To print out the position of your cursor when it is clicked and moved*

def on\_touch\_down(self, touch):

print(**'DOWN'**, touch)

def on\_touch\_up(self, touch):

print(**'UP'**, touch)

def on\_touch\_move(self, touch):

print(**'MOVE'**, touch)

class Myapp(App):

def build(self):

self.myLabel = MyLabel(text= **'hello'**)

return self.myLabel

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

## Likewise, we can draw shape using cursor

Draw a rectangle

from kivy.app import App

from kivy.uix.widget import Widget

from kivy.graphics import Rectangle

class MyWidget(Widget):

def on\_touch\_down(self, touch):

*# Draw a rectangle when you click your mouse*

self.canvas.add(Rectangle(pos= (touch.x, touch.y), size = (50,50)))

class Myapp(App):

def build(self):

return MyWidget()

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

Draw a line

from kivy.app import App

from kivy.uix.widget import Widget

from kivy.graphics import Line

class MyWidget(Widget):

def on\_touch\_down(self, touch):

*# Draw a line when you hold & move your cursor*

touch.ud[**'line'**] = Line(points = (touch.x, touch.y))

self.canvas.add(touch.ud[**'line'**])

def on\_touch\_move(self, touch):

touch.ud[**'line'**].points += [touch.x, touch.y]

class Myapp(App):

def build(self):

return MyWidget()

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

# Switching Screen

from kivy.app import App

from kivy.uix.button import Button

from kivy.uix.screenmanager import ScreenManager, Screen, SlideTransition

class FirstPage(Button):

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

super().\_\_init\_\_()

self.text = **'Hi'**

self.bind(on\_press = self.switch)

def switch(self, button):

myapp.screen\_manager.transition = SlideTransition(direction= **'left'**)

*# set the transition property moving to left*

myapp.screen\_manager.current = **'Second'**

class SecondPage(Button):

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

super().\_\_init\_\_()

self.text = **'Hi, there'**

self.bind(on\_press = self.switch)

def switch(self, button):

myapp.screen\_manager.transition = SlideTransition(direction= **'right'**)

myapp.screen\_manager.current = **'First'**

class Myapp(App):

def build(self):

self.screen\_manager = ScreenManager()

self.firstpage = FirstPage() *# create a variable of the first screen*

screen = Screen(name= **'First'**) *# create a screen named 'First'*

screen.add\_widget(self.firstpage) *# add the variable to the screen*

self.screen\_manager.add\_widget(screen) *# add the screen to main screen manager*

self.secondpage = SecondPage()

screen = Screen(name=**'Second'**)

screen.add\_widget(self.secondpage)

self.screen\_manager.add\_widget(screen)

return self.screen\_manager

if \_\_name\_\_ == **'\_\_main\_\_'**:

myapp = Myapp()

myapp.run()

# Coded in .kv language

If we code in kv language, it is easier for us to code, ( we don’t need to import uix widget manually. i.e. label, button) it is also easier to read.

from kivy.app import App

from kivy.lang import Builder

from kivy.uix.screenmanager import Screen

**"""**

**import Builder only when .kv file name doesn't match your app name**

**if you don't want to import Builder, make sure the name of your .kv file is**

**'my.kv' in this case as the main app is called 'MyApp'**

**"""**

Builder.load\_file(**'file.kv'**)

class MyBox(Screen):

def printOut(self):

txt= self.ids.TXTinput.text

*# in order to call widgets in .kv file, we need to give it an id*

*# do not add '' to id*

print(txt)

class Myapp(App):

def build(self):

return MyBox()

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

file.kv :

<MyBox>:

BoxLayout:

size: root.width, root.height

orientation: 'vertical'

TextInput:

id: TXTinput

Button:

text: 'Press me '

on\_press: root.new\_label()

Image:

source: 'blue\_monster.png'

# Import sound

from kivy.app import App

from kivy.core.audio import SoundLoader

class MyBox(Screen):

def play\_sound(self):

sound = SoundLoader.load(**'audio.mp3'**)

if sound:

sound.volume = 1

sound.play()

class Myapp(App):

def build(self):

return MyBox()

if \_\_name\_\_ == **'\_\_main\_\_'**:

Myapp().run()

file.kv :

<MyBox>:

BoxLayout:

size: root.width, root.height

orientation: 'vertical'

Button:

text: 'Press me '

on\_press: root.play\_sound()