



Tirumala Engineering College



Kivy

A Cross Platform Python UI Library

**Presentation By:-
Jyothi Prakash Muddana
17NE1A0571**

CONTENTS

1

INTRODUCTION

What and why kivy?

2

SETUP KIVY

setting kivy environment in PC

3

Kv LANGUAGE

How to work with kv language

4

KIVY IMPLEMENTATION

Kivy applicaiton example

5

PACKAGING KIVY APPLICATION

pack resouces required to run application

6

SELF LEARNING

How to learn kivy by self

INTRODUCTION

Kivy is a graphical user interface open-source Python library that allows you to develop multi-platform applications on Windows, macOS, Android, iOS, Linux, and Raspberry Pi.

Kivy is a free and open-source Python library used for developing mobile applications and other multitouch application software with a Natural User Interface.

Creating Kivy apps is fun , You will require a basic knowledge of Python and OOP's concepts to make it work



SETUP KIVY

Installation of kivy on Windows

- Need at least python 3.6 and above to work
- Upgrade pip ,wheel,setuptools of python
cmd : >python -m pip --upgrade pip wheel setuptools
- Use the below command to install kivy
cmd : > pip install kivy

KV LANGUAGE

Graphics instructions in Python

```
class MyWidget(Widget):  
    def __init__(self, **kw):  
        super(MyWidget, self).__init__(**kw)  
        with self.canvas:  
            # add your instruction for main canvas here  
            Color(1, 0, .4, mode='rgb')  
            Line(points=(x1, y1, x2, y2, x3, y3))  
        with self.canvas.before:  
            # you can use this to add instructions rendered before  
        with self.canvas.after:  
            # you can use this to add instructions rendered after
```

Graphics instructions in KV language

```
MyWidget:  
    canvas:  
        Color:  
            rgba: 1, .3, .8, .5  
        Line:  
            points: zip(self.data.x, self.data.y)
```

KIVY IMPLEMENTATION

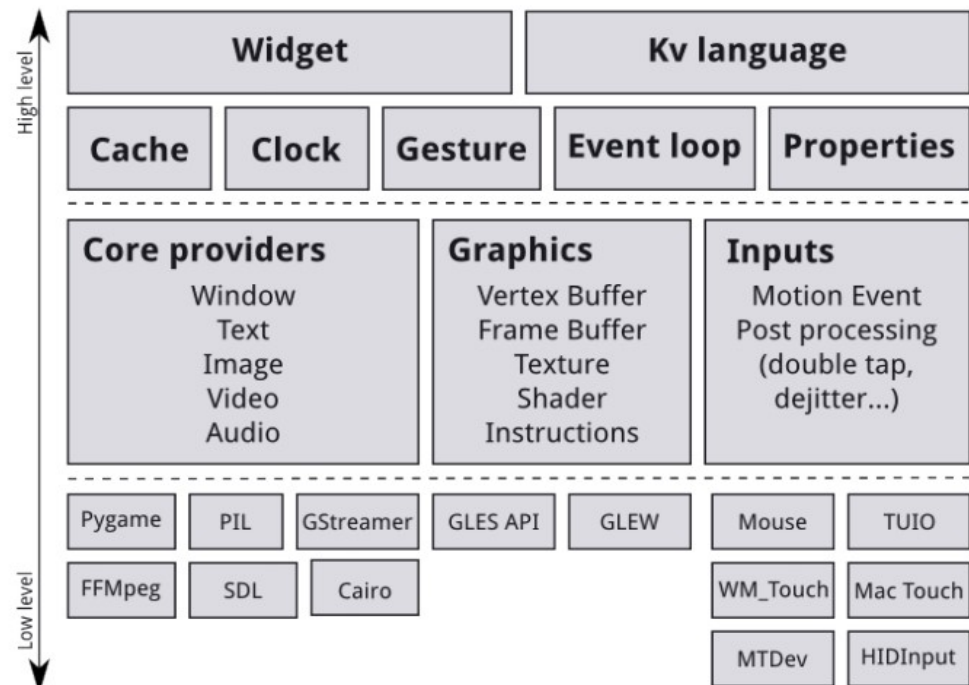
To work with kivy
No need of any low level
behaviour of the kivy

By inheriting the class from
the kivy module we can easily
modify and use them
through method overriding

This Architecture will helps us
to know how the whole model
was developed from scratch

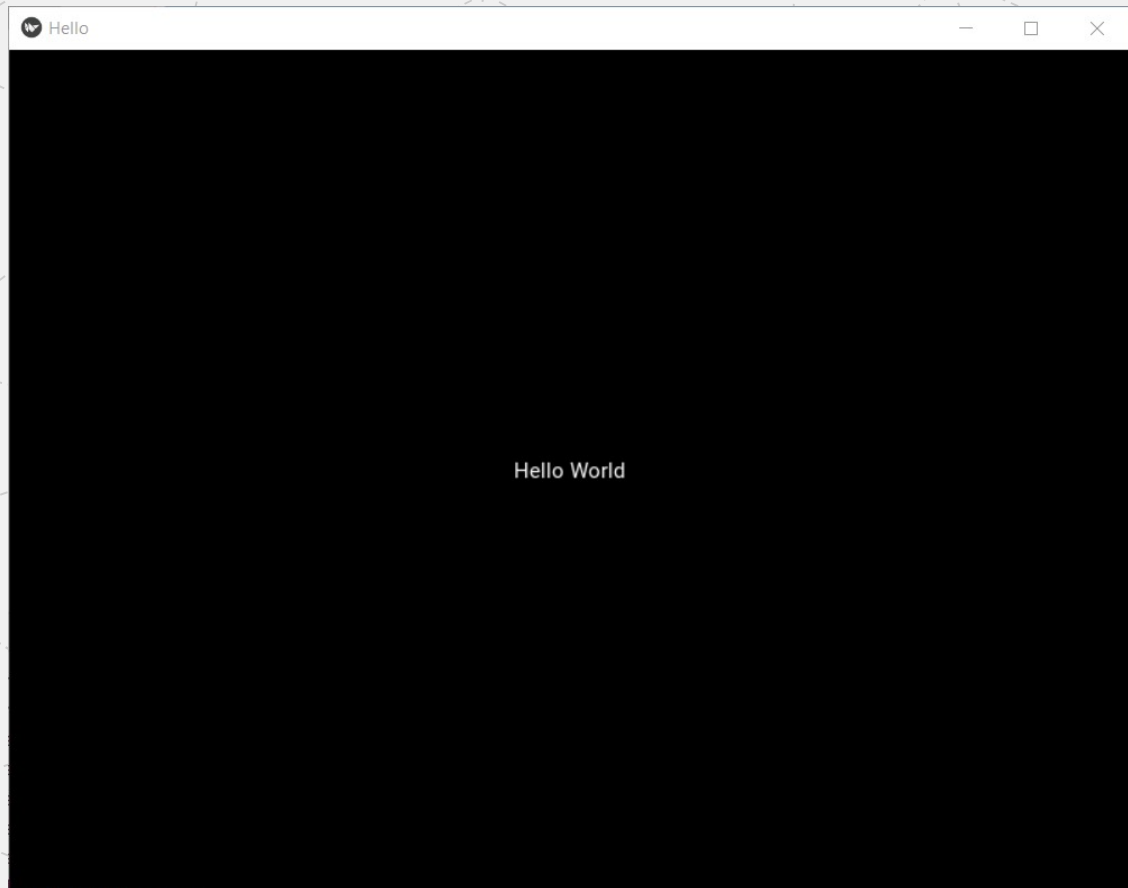


Kivy Architecture



Hello World application

```
from kivy.app import App  
from kivy.uix.label import Label  
  
class HelloApp(App):  
    def build(self):  
        return Label(text = "Hello World")  
  
helloApp = HelloApp()  
helloApp.run()
```



Hello World application with kv language

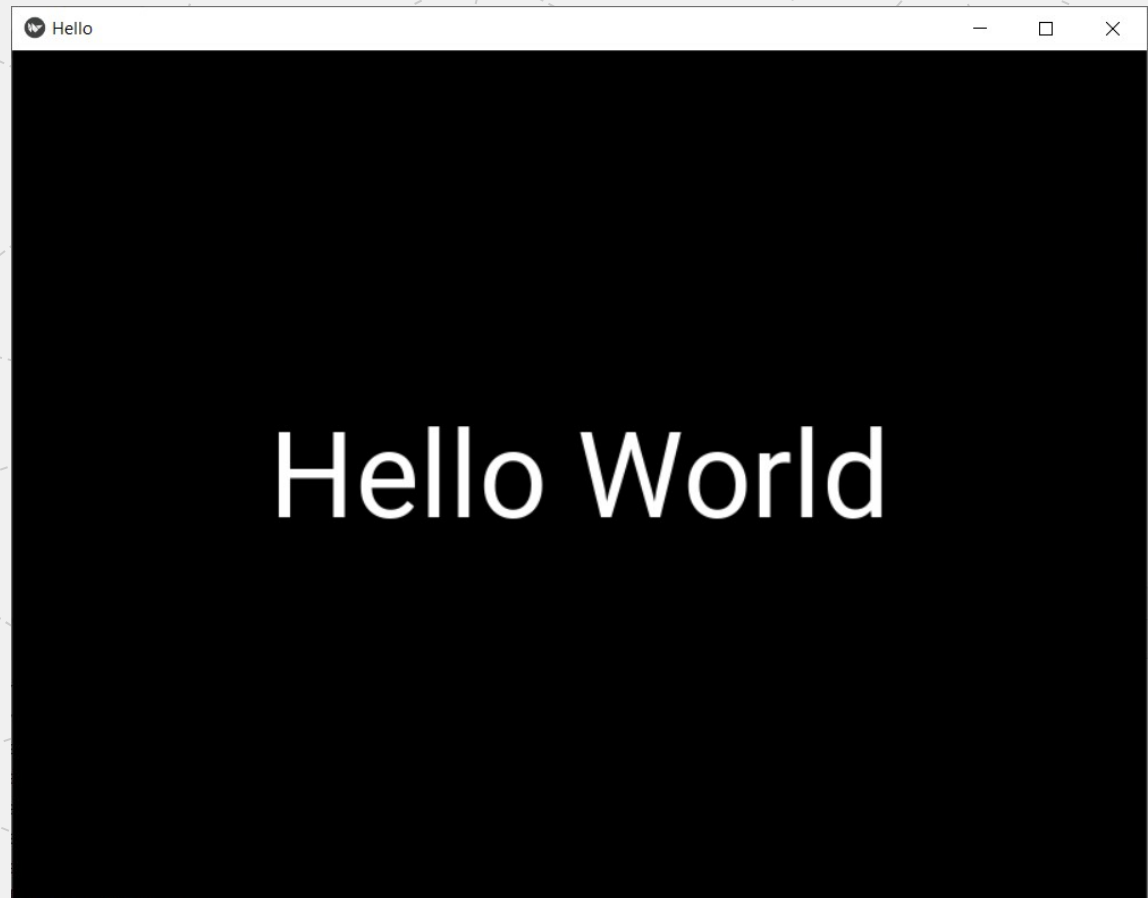
```
from kivy.app import App  
from kivy.lang import Builder
```

```
widget = Builder.load_file("filename.kv")  
class HelloApp(App):  
    def build(self):  
        return widget
```

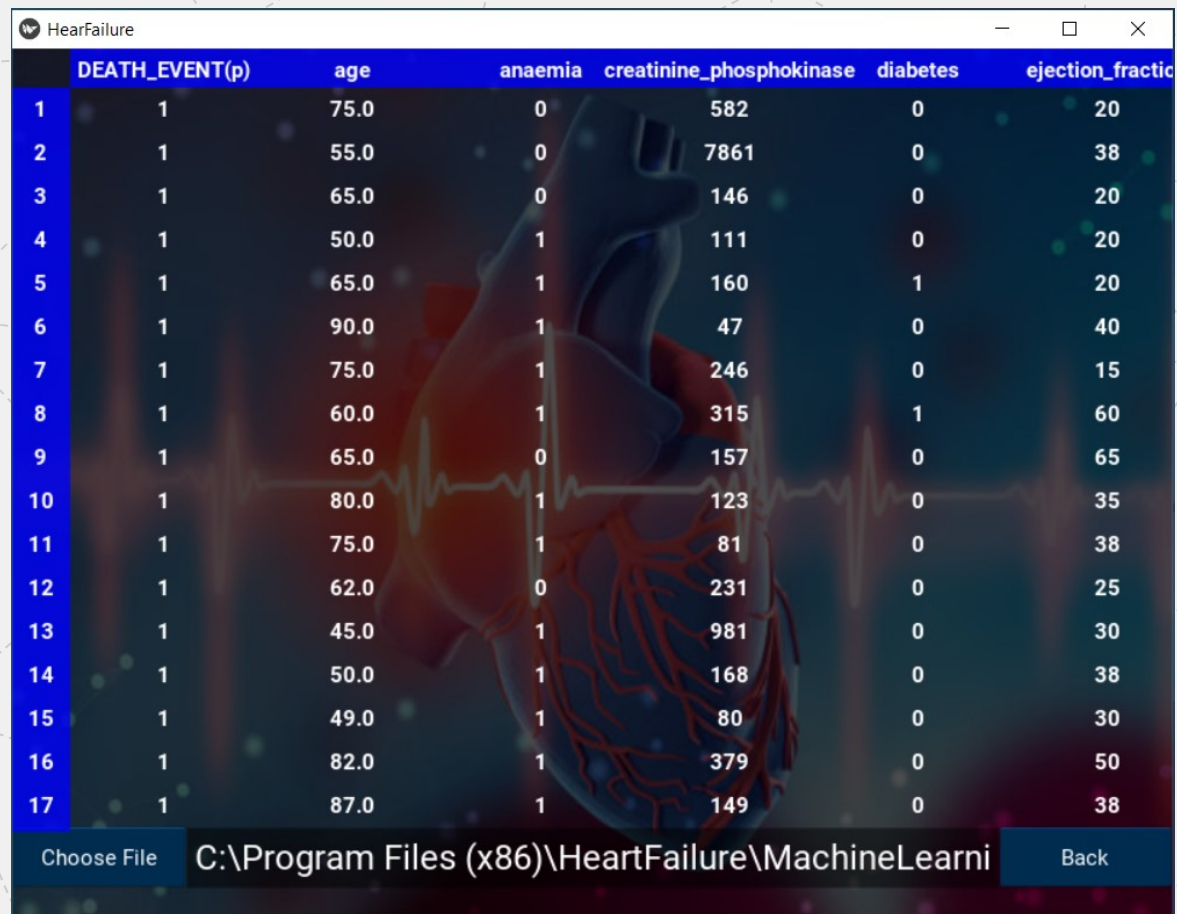
```
helloApp = HelloApp()  
helloApp.run()
```

```
#kv language  
Label:
```

```
text : "Hello World"  
font_size : "64pt"
```



Some Application Developed by me



HeartFailure

	DEATH_EVENT(p)	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fractio
1	1	75.0	0	582	0	20
2	1	55.0	0	7861	0	38
3	1	65.0	0	146	0	20
4	1	50.0	1	111	0	20
5	1	65.0	1	160	1	20
6	1	90.0	1	47	0	40
7	1	75.0	1	246	0	15
8	1	60.0	1	315	1	60
9	1	65.0	0	157	0	65
10	1	80.0	1	123	0	35
11	1	75.0	1	81	0	38
12	1	62.0	0	231	0	25
13	1	45.0	1	981	0	30
14	1	50.0	1	168	0	38
15	1	49.0	1	80	0	30
16	1	82.0	1	379	0	50
17	1	87.0	1	149	0	38

Choose File C:\Program Files (x86)\HeartFailure\MachineLearn Back

PACKAGING APPLICATION

We need to pack our application so that in one go we can share and run our application on others computers also

for windows/linux to generate exe file we need

#pyinstaller

>pip install pyinstaller

>pyinstaller mainfilename.py

for android to generate apk file

#buildozer

>pip install buildozer

>buildozer init

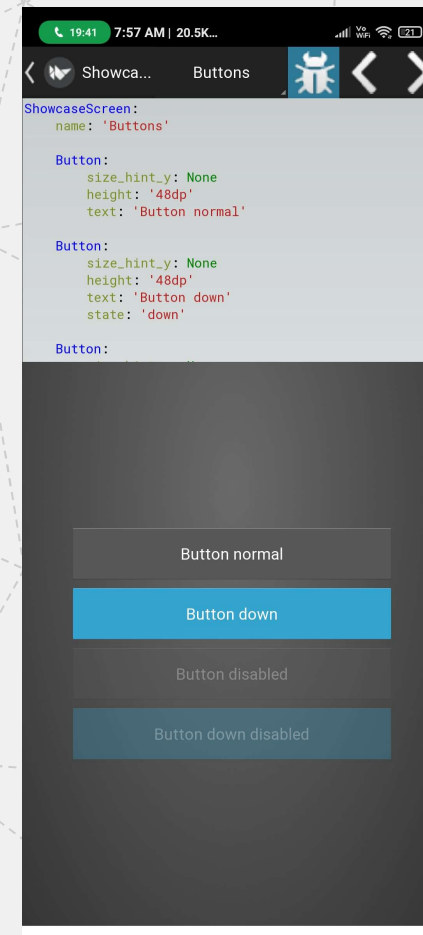
#modify spec file

>buildozer -V android debug

SELF LEARNING

How to learn kivy by self ?

* we can learn by using an application (showcase kivy)



ADVANTAGES

- Based on Python, which is the extremely powerful given it's library rich nature.
- Write code once and use it across all devices.
- Easy to use widgets built with multi-touch support.
- kv language is a language used to give the syntax of the kivy program a better view by representing the all the elements in the program like classes, the other classes it is inheriting, widgets and their properties and configurations etc. in a tree form
- You can run your app on desktop also, so there is no need to install some extra emulators/VMs to get it work

DIS - ADVANTAGES

- Not that much information in Internet, even on stackoverflow
- Pretty messy documentation
- No obvious way to test the application
- Not obvious mechanisms of placing widgets, especially in built in layouts, which causes situations like: you want place widget in the center of its parent, but kivy places it anywhere but not where you want it to be.
- Official examples are quite ugly, so you may get false vision of how your application could look like.

I believe you are
curious to learn
and implement
kivy to develop
small applications
which are useful
in daily life

START

A man in a dark suit stands with his back to the camera on a paved road. The word "START" is painted in large, white, italicized letters on the road surface just in front of him. The road stretches straight ahead into the distance, flanked by dry, scrubby vegetation. The sky is filled with heavy, grey clouds, and the overall lighting is dim, suggesting dawn or dusk.

THANK YOU

