

Universidade Federal de Uberlândia
Engenharia de Controle e Automação / Engenharia Mecatrônica
Sistemas Embarcados II / Sistemas Digitais para Mecatrônica
Prof. Éder Alves de Moura
Semana 07 – Desenvolvendo IHM com Kivy



1. Como primeiro ponto de estudo, entenda sobre os conceitos de UI, UX e do projeto de Produtos, que são essenciais para quem desenvolve uma aplicação moderna. Veja os vídeos abaixo listados e produza um resumo de até duas páginas sobre os principais conceitos e aplicações. Converta o arquivo para o formato PDF.

- O que é UX e UI? - Semana do Design

https://www.youtube.com/watch?v=t_wlpJq9oZw

- Basics of UI, UX, and Product Design | What is Product Design?

<https://www.youtube.com/watch?v=gixxebwfqXM>

Introdução

UX e UI significam User Experience e User Interface. Os designs UX e UI são complementares e existem para responder as necessidades do usuário na definição de um produto.

O que UI significa ?

O design UI define a interface gráfica e o design visual das aplicações e dos aparelhos. O UI engloba todos os elementos visuais de uma aplicação, como os botões, o logo, as animações, a fonte, etc...

O UI designer se ocupa dos elementos visuais de uma aplicação. Ele quer criar um visual que facilita as interações no aplicativo. Ele sabe o comportamento do humano ao usar uma aplicação e sabe como eles vão reagir a um certo design. O usuário tem que fazer naturalmente o que o aplicativo quer que ele faz. O usuário não deve perceber que o aplicativo pode manipular a experiência dele.

O que UX significa ?

O design UX define a experiência que o designer quer compartilhar com o usuário. Ele é diferente dependendo de quem vai usar o aplicativo, que funcionalidades vão ser dentro do aplicativo, e que tipo de experiência o designer quer para o usuário. O design UX pode se focalizar na ergonomia para um lado prático, mas também pode ser feito para impulsionar o usuário a assistir publicidades, ficar no aplicativo, ou consumir. A experiência tem que ser otimizada para pode competir com outros aplicativos que oferecem os mesmos serviços.

O UX designer é responsável da satisfação do usuário com o aplicativo, ele determina a estrutura e as funcionalidades do produto, de tal forma que o produto tem que ser útil, agradável e fazer sentido para o usuário. O designer UX define o comportamento da interface e se interessa no retorno dos usuários.

Product Design

O product design é a criação do produto do começo até o fim. Indo da imaginação, da pesquisa e desenvolvimento, da industrialização até a estratégia de marketing e a durabilidade de vida do produto.

O product designer é responsável da criação do produto do começo até o fim. Ele sempre procura melhorar a experiência do usuário, as funcionalidades e o funcionamento do produto. Ele quer fazer o projeto ter sentido e se assegura que todos os componentes do produto interagem com coerência.

Conclusão

O design UI faz parte do design UX e o design UX faz parte do design do produto. O design do produto engloba toda a criação do projeto, enquanto o design UX engloba as funcionalidades e o funcionamento da interface. O design UI é focalizado na parte gráfica da interface do produto.

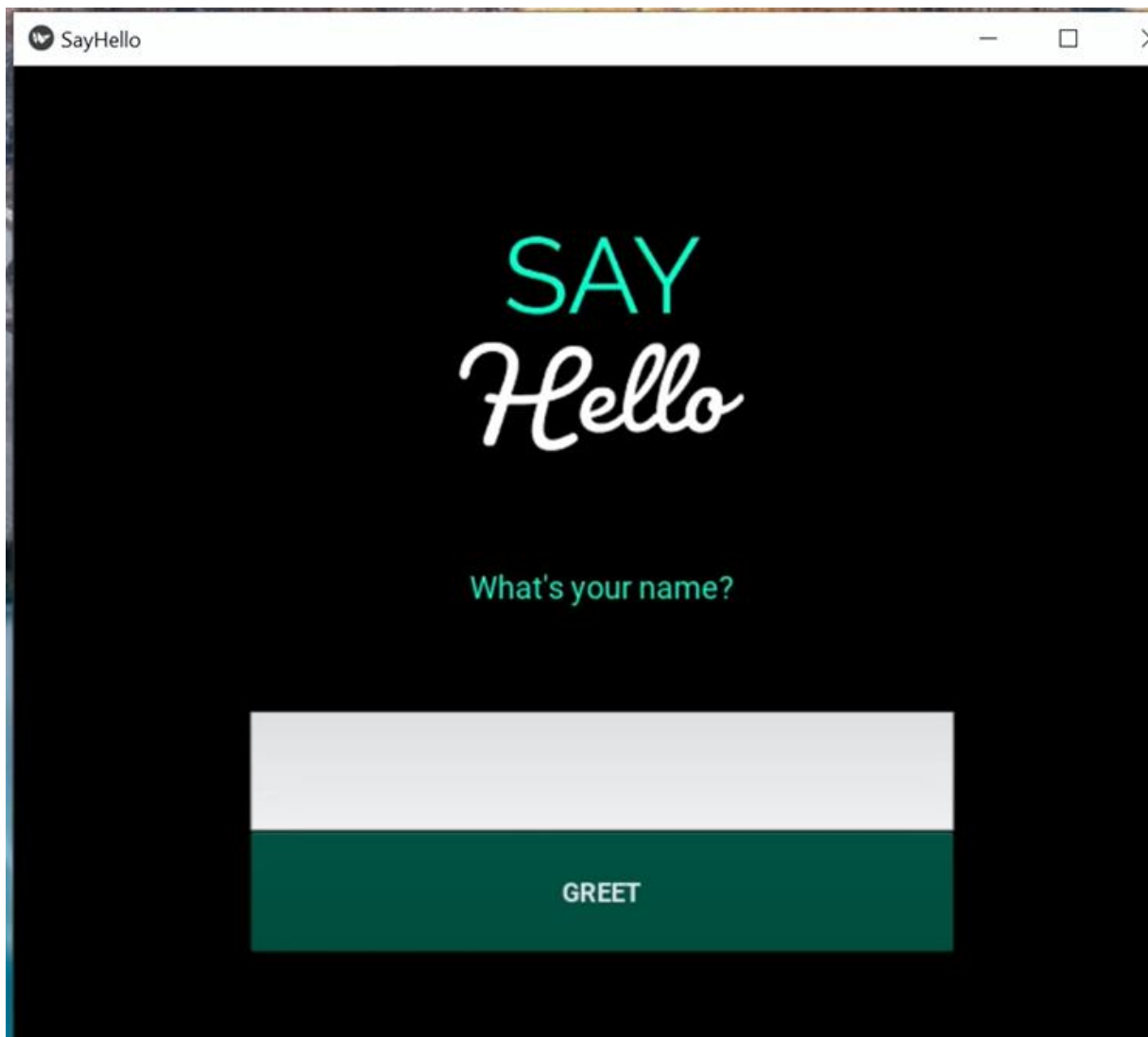
2. Neste primeiro exemplo você verá o processo de instalação do Kivy e serão discutidos alguns conceitos básicos sobre o uso dessa biblioteca em uma aplicação simples:

- Simple Python App with Kivy - Step by Step GUI Tutorial
<https://www.youtube.com/watch?v=YDp73WjNlSc>

Crie uma pasta 'problema02' para armazenar os códigos dessa atividade.

```
1 from kivy.app import App
2 from kivy.uix.gridlayout import GridLayout
3 from kivy.uix.label import Label
4 from kivy.uix.image import Image
5 from kivy.uix.button import Button
6 from kivy.uix.textinput import TextInput
7
8 class SayHello(App):
```

```
7
8  class SayHello(App):
9      def build(self):
10         #returns a window object with all it's widgets
11         self.window = GridLayout()
12         self.window.cols = 1
13         self.window.size_hint = (0.6, 0.7)
14         self.window.pos_hint = {"center_x": 0.5, "center_y":0.5}
15
16         # image widget
17         self.window.add_widget(Image(source="logo.png"))
18
19         # label widget
20         self.greeting = Label(
21             text= "What's your name?",
22             font_size= 18,
23             color= '#00FFCE'
24         )
25         self.window.add_widget(self.greeting)
26
27         # text input widget
28         self.user = TextInput(
29             multiline= False,
30             padding_y= (20,20),
31             size_hint= (1, 0.5)
32         )
33
34         self.window.add_widget(self.user)
35
36         # button widget
37         self.button = Button(
38             text= "GREET",
39             size_hint= (1,0.5),
40             bold= True,
41             background_color = '#00FFCE',
42             #remove darker overlay of background colour
43             # background_normal = ""
44         )
45         self.button.bind(on_press=self.callback)
46         self.window.add_widget(self.button)
47
48         return self.window
49
50     def callback(self, instance):
51         # change label text to "Hello + user name!"
52         self.greeting.text = "Hello " + self.user.text + "!"
53
54 # run Say Hello App Calss
55 if __name__ == "__main__":
56     SayHello().run()
```



Semana 07 – Desenvolvendo IHM com Kivy

3. Desenvolva a aplicação detalhada no seguinte vídeo:

- Como Criar Aplicativos e Programas com Python - Introdução ao Kivy

<https://www.youtube.com/watch?v=NkBcU8bIV4s>

Obs.: Para ver exemplos de diversas funcionalidades do Kivy, veja a playlist:

- Intro To Kivy - Installing Kivy on Windows - Python Kivy GUI Tutorial #1

https://www.youtube.com/watch?v=dLgqujoc5_U&list=PLCC34OHncOtpz7PJQ7Tv7hqFBP_xDDjqg

Crie uma pasta 'problema03' para armazenar os códigos dessa atividade.

```
1  # importar o App, Builder (GUI)
2  # # criar o nosso aplicativo
3  # # criar a função build
4
5  ✓ from kivy.app import App
6  from kivy.lang import Builder
7  import requests
8
9  GUI = Builder.load_file("tela.kv")
10
11
12  ✓ class MeuAplicativo(App):
13  ✓     def build(self):
14  |         return GUI
15
16  ✓     def on_start(self):
17  |         self.root.ids["moeda1"].text = f"Dólar R${self.pegar_cotacao(
18  |         self.root.ids["moeda2"].text = f"Euro R${self.pegar_cotacao(
19  |         self.root.ids["moeda3"].text = f"Bitcoin R${self.pegar_cotaca
20  |         self.root.ids["moeda4"].text = f"Ethereum R${self.pegar_cota
21
22  ✓     def pegar_cotacao(self, moeda):
23  |         link = f"https://economia.awesomeapi.com.br/last/{moeda}-BRL'
24  |         requisicao = requests.get(link)
25  |         dic_requisicao = requisicao.json()
26  |         cotacao = dic_requisicao[f"{moeda}BRL"]["bid"]
27  |         return cotacao
28
29
30  MeuAplicativo().run()
```

```
semana 7 > problema03 > ≡ tela.kv
1  ∨ GridLayout:
2      cols: 1
3  ∨  Label:
4      |      id: moeda1
5      |      text: "1º texto"
6  ∨  Label:
7      |      id: moeda2
8      |      text: "2º texto"
9  ∨  Label:
10     |      id: moeda3
11     |      text: "3º texto"
12  ∨  Label:
13     |      id: moeda4
14     |      text: "3º texto"
```

