Deel 04

GUI's maken met Python: 's Werelds slechtste GUI

Leer een goede GUI ontwerpen door het eerst helemaal fout te doen!

MAKER

Laura Sach

Laura leidt het A Level-team bij de Raspberry Pi Foundation, waar ze middelen maakt voor leerlingen om te leren over computerwetenschappen.

@CodeBoom

MAKER

Martin O'Hanlon

Martin werkt in het leerteam van de Raspberry Pi Foundation, waar hij online cursussen, projecten en leermiddelen creëert.

@martinohanlon

Het is tijd om echt aan de slag te gaan met je GUI's en te experimenteren met verschillende widgets, kleuren, lettertypes, en features. Zoals bij de meeste experimenten, zul je het waarschijnlijk niet meteen de eerste keer goed doen! In feite, ga je de verkeerde manier ontdekken om je GUI te maken.

Moeilijk te lezen

De juiste keuze van GUI-kleur en -lettertype zijn belangrijk. Het is belangrijk dat het contrast tussen achtergrond en tekstkleur er voor zorgt dat je GUI makkelijk leesbaar is. Wat je niet moet doen is twee kleuren gebruiken die erg op elkaar lijken.
Importeer de widgets bovenaan in de code:

```
from guizero import App, Text
```

Maak een app met een titel:

```
app = App("it's all gone wrong")
title = Text(app, text="Some hard to read text")

app.display()
```

Experimenteer door de kleuren, het lettertype en de tekstgrootte te veranderen (zie listing **worst1.py**). Onze keuzes zijn niet de beste!

```
app = App("it's all gone wrong", bg="dark green")
title = Text(app, text="Some hard-to-read text", size="14", font="Comic Sans", color="green")
```

Het is belangrijk dat tekst op een GUI ook lang genoeg blijft staan om gelezen te worden. Hij moet zeker niet verdwijnen of gaan knipperen.
Alle widgets in guizero kunnen onzichtbaar (of weer zichtbaar) gemaakt worden met de functies `hide()` en `show()`. Door de functie `repeat` in guizero te gebruiken om elke seconde een functie uit te voeren, kun je je tekst laten verdwijnen en verschijnen, zodat hij lijkt te knipperen.

Maak een functie die de tekst verbergt als hij zichtbaar is en toont als hij dat niet is:

```
def flash_text():
    if title.visible:
        title.hide()
    else:
        title.show()
```

Voordat de app wordt weergegeven, gebruik je `repeat` om de functie `flash_text` elke 1000 milliseconden (1 seconde) te laten lopen.

```
app.repeat(1000, flash_text)

app.display()
```

Je code zou er nu uit moeten zien als **worst2.py**. Test je app: de titeltekst moet knipperen, en elke seconde verschijnen en verdwijnen.

De verkeerde widget

Het gebruik van een geschikte widget kan het verschil betekenen tussen een geweldige GUI en een die compleet onbruikbaar is.
Welke widget zou je gebruiken om een datum in te voeren? Een TekstBox? Meerdere Combo's? Een TextBox zou flexibeler zijn maar zou toetsing en formattering vereisen. Meervoudige Combo's voor

Figuur 1

Figuur 1 Combo's om letters te kiezen.

jaar, maand en dag hoeven niet getoetst te worden, maar zijn langzamer in gebruik.

Het gebruik van een Slider om een datum en tijd in te stellen (**Figuur 2**), zoals in het codevoorbeeld **worst3.py**, is echter niet zo'n goed idee.

De Slider widget geeft een getal tussen 0 en 999.999.999 terug. Dit is het aantal seconden sinds 1 januari 1970. De functie `ctime()` wordt gebruikt om dit getal om te zetten in een datum en tijd.

Tekst-input krijgen van je gebruiker is eenvoudig: een TextBox of een multi-line TextBox zou aan al je behoeften moeten voldoen. Maar is dit te simpel? Vergt het te veel typwerk?

Hoe zit het met de gebruiker die alleen de muis wil gebruiken? Misschien is een reeks Combo's met in elk alle letters van het alfabet beter (**Figuur 1**)? Begin met het importeren van de guizero widgets en `ascii_letters`.

```
from guizero import App, Combo
from string import ascii_letters
```

`ascii_letters` is een lijst met alle 'afdrukbare' ASCII-tekens die je kunt gebruiken als de opties voor de Combo.

Maak een enkele Combo die alle letters bevat en laat de app weergeven.

```
a_letter = Combo(app, options=" " + ascii_
letters, align="left")

app.display()
```

Je programma zou nu moeten lijken op **worst4.py**. Als je het uitvoert, zie je een enkele Combo die alle letters bevat plus een spatie en is uitgelijnd aan de linkerkant van het venster.

Om een regel letters bij elkaar te krijgen, zou je een heleboel Combo widgets aan je app kunnen toevoegen, bijv:

```
a_letter = Combo(app, options=" " + ascii_
letters, align="left")
b_letter = Combo(app, options=" " + ascii_
letters, align="left")
c_letter = Combo(app, options=" " + ascii_
letters, align="left")
```

Door elk Combo widget links uit te lijnen, worden de widgets naast elkaar tegen de linkerrand weergegeven.

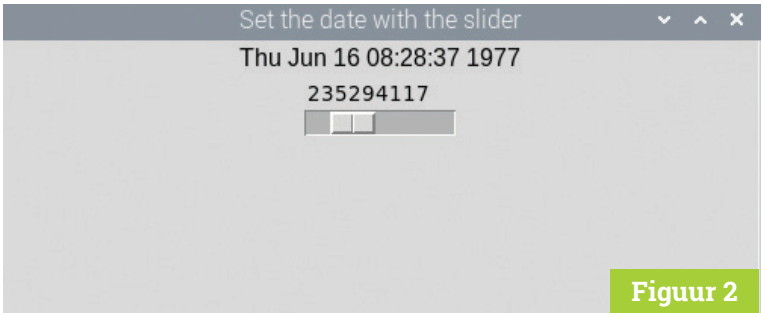
Als alternatief zou je een `for`-lus kunnen gebruiken, een lijst van letters maken, en elke letter

worst1.py

DOWNLOAD DE VOLLEDIGE CODE: magpi.cc/guizerocode

Taal: **Python 3**

```
001.  # Imports ---------------
002.
003.  from guizero import App, Text
004.
005.
006.  # App -------------------
007.
008.  app = App("it's all gone wrong", bg="dark green")
009.
010.  title = Text(app, text="Hard to read", size="14", font="Comic
011.  Sans", color="green")
012.
013.  app.display()
```



Figuur 2 Een schuifknop om datum en tijd in te stellen.

worst2.py

Taal: **Python 3**

```
001.  # Imports ---------------
002.
003.  from guizero import App, Text
004.
005.
006.  # Functions -------------
007.
008.  def flash_text():
009.      if title.visible:
010.          title.hide()
011.      else:
012.          title.show()
013.
014.
015.  # App -------------------
016.
017.  app = App("it's all gone wrong", bg="dark green")
018.
019.  title = Text(app, text="Hard to read", size="14", font="Comic
020.  Sans", color="green")
021.
022.  app.repeat(1000, flash_text)
023.
024.  app.display()
```

worst3.py

Taal: **Python 3**

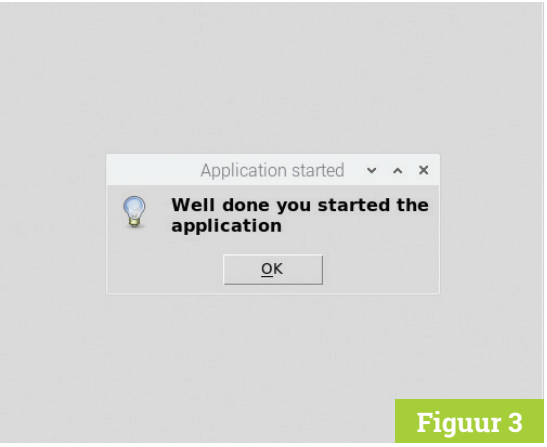
```
001.  # Imports ---------------
002.
003.  from guizero import App, Slider, Text
004.  from time import ctime
005.
006.
007.  # Functions -------------
008.
009.  def update_date():
010.      the_date.value = ctime(date_slider.value)
011.
012.
013.  # App -------------------
014.
015.  app = App("Set the date with the slider")
016.  the_date = Text(app)
017.  date_slider = Slider(app, start=0, end=999999999,
018.  command=update_date)
019.
020.  app.display()
```

worst4.py

Taal: **Python 3**

```
001.  # Imports ---------------
002.  from guizero import App, Combo
003.  from string import ascii_letters
004.
005.
006.  # App -------------------
007.
008.  app = App("Enter your name")
009.
010.  a_letter = Combo(app, options=" " + ascii_letters, align="left")
011.
012.  app.display()
```

Figuur 3
Zinloze pop-up.



Figuur 3

Window widget

Pop-up boxes kunnen worden gebruikt om gebruikers vragen te stellen, maar ze zijn erg eenvoudig.

Als je extra informatie wilt tonen of aanvullende gegevens wilt vragen, zou je de widget Window kunnen gebruiken om meerdere vensters te maken.

Window wordt op een vergelijkbare manier gebruikt als App en heeft veel van dezelfde functies.

```
from guizero import App, Window

app = App("Main window")
window = Window(app, "2nd Window")

app.display()
```

Je kunt bepalen of een Window op het scherm te zien is met de methodes **show()** en **hide()**.

```
window.show()
window.hide()
```

Je kunt een app maken om te wachten tot een venster is gesloten nadat het is getoond, door True door te geven aan de wait parameter van show. Bijvoorbeeld

```
window.show(wait=True)
```

Je kunt meer te weten komen over het gebruik van meerdere vensters in de documentatie van guizero: **lawsie.github.io/guizero/multiple_windows**.

aan de lijst toevoegen, zoals getoond in **worst5.py**.

Probeer beide concepten en kijk welke je voorkeur heeft. De `for`-lus is flexibeler omdat je daarmee zoveel letters kunt maken als je wilt.

Pop-ups

Geen verschrikkelijke GUI zou compleet zijn zonder een pop-up box. guizero bevat een aantal pop-up boxen, die je kunt gebruiken om gebruikers iets belangrijks te laten weten of om nuttige informatie te verzamelen. Maar je kunt ze ook gebruiken om gebruikers te ergeren en te irriteren!

Maak eerst een applicatie die aan het begin een zinloze box laat verschijnen om te laten weten dat de applicatie gestart is.

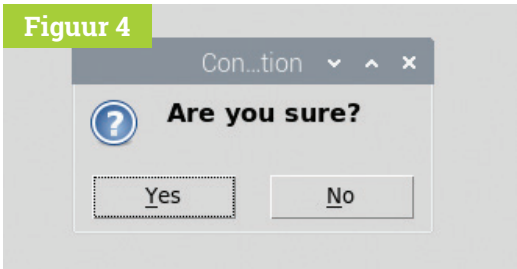
```
from guizero import App

app = App(title="pointless pop-ups")

app.info("Application started", "Well done
you started the application")
```

Noot van de redactie: dit artikel verscheen oorspronkelijk in MagPi tijdschrift. Elektor publiceert de Nederlandse, Franse en Duitse edities van MagPi.

Figuur 4




```
app.display()
```

Als je de applicatie uitvoert, zul je zien dat er een 'info'-box verschijnt (**Figuur 3**). De eerste parameter die aan `info` wordt doorgegeven is de titel van het venster; de tweede parameter is de boodschap.

Je kunt de stijl van deze eenvoudige pop-up veranderen door `warn` of `error` te gebruiken in plaats van `info`.

Pop-up boxes kunnen ook worden gebruikt om informatie van de gebruiker te krijgen. De eenvoudigste is een `yesno` die de gebruiker een vraag stelt en een True of False antwoord krijgt. Dit is handig als je een gebruiker wilt laten bevestigen voordat hij iets doet, zoals het verwijderen van een bestand. Maar misschien niet elke keer als ze op een knop drukken! Importeer de PushButton widget in je applicatie:

```
from guizero import App, PushButton
```

Maak een functie die de `yesno` pop-up gebruikt om een bevestiging te vragen.

```
def are_you_sure():
    if app.yesno("Confirmation", "Are you
sure?"):
        app.info("Thanks", "Button
pressed")
    else:
        app.error("Ok", "Cancelling")
```

Voeg de knop toe aan je GUI die de functie aanroept als hij wordt ingedrukt.

```
button = PushButton(app, command=are_you_
sure)
```

Je code zou nu moeten lijken op **05-worlds-worst-gui.py**. Wanneer je het programma uitvoert en op de knop klikt, zul je een pop-up zien die je vraagt om te bevestigen met Yes of No (**Figuur 4**).
Je kunt meer te weten komen over de pop-up boxen in guizero op lawsie.github.io/guizero/alerts.
Misschien kun je al deze 'features' kunnen combineren tot één geweldige GUI?

worst5.py

Taal: **Python 3**

```
001.  # Imports ---------------
002.
003.  from guizero import App, Combo
004.  from string import ascii_letters
005.
006.
007.  # App -------------------
008.
009.  app = App("Enter your name")
010.
011.  name_letters = []
012.  for count in range(10):
013.      a_letter = Combo(app, options=" " + ascii_letters,
014.  align="left")
015.      name_letters.append(a_letter)
016.
017.  app.display()
```

05-worlds-worst-gui.py

Taal: **Python 3**

```
001.  from guizero import App, PushButton
002.
003.  def are_you_sure():
004.      if app.yesno("Confirmation", "Are you sure?"):
005.          app.info("Thanks", "Button pressed")
006.      else:
007.          app.error("Ok", "Cancelling")
008.
009.  app = App(title="pointless pop-ups")
010.
011.  button = PushButton(app, command=are_you_sure)
012.
013.  app.info("Application started", "Well done you started the
014.  application")
015.
016.  app.display()
```