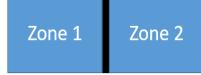
## **Présentation: Exercice Kivy**

## Présentée par: Hamza Ghanmi Ingénieur en mécanique et en thermique-énergétique



- L'exercice consiste à développer une interface graphique sur python en utilisant la bibliothèque Kivy
- le problème revient à saisir une valeur dans une zone, calculer l'autre et afficher sa valeur dans l'autre zone de manière instantanée
- 100 101.0
- La saisie d'une valeur dans la zone 1 engendre l'affichage d'une valeur dans la zone 2 selon cette expression: (Zone 2) = (Zone 1)+1
- Exemple: (Zone 1)=100, (Zone 2)=(Zone 1)+1=101
- 44.72135954999579 100
- La saisie d'une valeur dans la zone 2 engendre l'affichage d'une valeur dans la zone 1 selon cette expression: (Zone 1) = (Zone 2)/sqrt(5)
- Exemple: (Zone 2)=100, (Zone 1) = (Zone 2)/sqrt(5)=44,2

from kivy.app import App from kivy.uix.textinput import TextInput from kivy.uix.gridlayout import GridLayout from math import \*

- Importer l'ensemble des outils de la bibliothèque Kivy et math
- App: la création d'une application desktop
- TextInput: la création d'une zone de saisie
- GridLayout: la création d'une grille
- Math: ensemble des fonctions et opérateurs mathématique

- La définition du class MyGrid: une grille qui contient 2 colonnes, positionnée à (225, 250)
- La définition de la zone de saisie var1: la saisie est filtrée (seulement des réels), positionnée à (50, 50)

```
self.add widget(self.var1)
    self.var2 = TextInput(input filter='float',
                 multiline=False.
                 background color=(8 / 200, 136 / 200, 1, 1)
    self.add widget(self.var2)
    self.var1.fbind('focus', self.foc1)
    self.var2.fbind('focus', self.foc2)
  def foc2(self, obj1, obj2):
     if self.var2.focus:
       self.var1.text = ""
       self.var2.text = ""
       self.var2.fbind('text', self.change1)
     else:
       self.var2.funbind('text', self.change1)
  def foc1(self, obj1, obj2):
     if self.var1.focus:
       self.var1.text = ""
       self.var2.text = ""
       self.var1.fbind('text', self.change2)
     else:
       self.var1.funbind('text', self.change2)
  def change1(self, obj1, obj2):
       self.var1.text = str(float(self.var2.text)/sqrt(5))
    except ValueError:
       self.var1.text = ""
  def change2(self, obj1, obj2):
       self.var2.text = str(float(self.var1.text)+1)
    except ValueError:
       self.var2.text = ""
class MainApp(App):
                                   Définition de
                                   l'application MainApp
  def build(self):
                                  Construire l'interface en
    return MyGrid()
                                   retournant MyGrid
if name == " main
                                  • Lancer l'application
  MainApp().run()
```

- Accrocher la zone de saisie var1 à la grille MyGrid
- La définition de la zone de saisie var1: la saisie est filtrée (seulement des réels), positionnée à (50, 50)
- Accrocher la zone de saisie var1 à la grille MyGrid
- Rappeler les fonction foc1 ou foc2 suite à l'action d'activation de l'une des deux zone var1 ou var2
- Si le curseur est dans la zone 1 l'algorithme fait appel à foc1
- Si le curseur est dans la zone 2 l'algorithme fait appel à foc2
- Définition des deux fonctions foc1 et foc2, ce sont deux fonctions commandent l'appel et la fin d'appel aux fonction change1 et change2
- Foc1: Si le curseur change de position vers la zone1, la fonction vide les deux zone pour se préparer à la saisie de var1 et le calcule de var2 à travers change2
- Foc2: Si le curseur change de position vers la zone2, la fonction vide les deux zone pour se préparer à la saisie de var2 et le calcule de var2 à travers change2
- .fbind permet l'appel et .funbind coupe l'appel
- Définition des deux fonctions change1 et change2, ce sont deux fonctions permettent de calculer respectivement var1 et var2 à partir de var2 et var1

## **Conclusion:**

- Le point délicat dans ce problème est de trouver une méthode qui permet de calculer la valeur d'une variable à partir d'une valeur saisie sans affecter cette dernière et sans bloquer ou gêner la saisie
- La solution trouvée était de faire appel à deux fonctions qui permettent le calcul des variables à travers deux expressions différentes, l'appel est sensible à la saisie
- Les deux méthodes qui gèrent l'appel et l'arrêt d'appel sont .fbind et .funbind
- Je pense que j'ai réussi à résoudre ce problème, j'ai réussi à créer un algorithme fonctionnel comme demandé, il y a quelques points à améliorer (par exemple: réduire le nombre des fonctions