pokemon.zip sur site (pokedex.csv+dossier photos)

Pokémon Go
Christophe Viroulaud
Première NSI

Pokémon Go

Christophe Viroulaud

Première NSI

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Problématique

Le jeu pour smartphone *Pokemon Go* reprend l'univers du manga éponyme. Il utilise la réalité augmentée pour donner une expérience utilisateur nouvelle.

Figure – Illustration du jeu Pokémon GO

Problématique

Le jeu pour smartphone *Pokemon Go* reprend l'univers du manga éponyme. Il utilise la réalité augmentée pour donner une expérience utilisateur nouvelle.

Figure – Illustration du jeu Pokémon GO

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Devant le succès du jeu des communautés se créent et tentent d'établir des stratégies pour optimiser leurs résultats.

On se propose de construire un programme pour aider les joueurs dans leurs quêtes.

Devant le succès du jeu des communautés se créent et tentent d'établir des stratégies pour optimiser leurs résultats.

On se propose de construire un programme pour aider les joueurs dans leurs quêtes.

Pokémon Go

Problématique

Informations disponibles

Interface graphique

Bibliothèque tkinter

Ajouter un composant

Construire par bloc

Remplissage des labels

Gestion des images

Interaction entre les composants

Liste de choix : Combobox

Choisir un Pokémon : un événement

Informations disponibles

Quand on joue à *Pokémon Go* on trouve des Pokémon sur notre chemin, mais également des œufs. Il faut parcourir une certaine distance pour faire éclore un œuf. Enfin, il est possible de faire évoluer un Pokémon à l'aide de friandises. Un fichier de données (*pokedex.csv*) recense l'ensemble des Pokémons utilisables dans le jeu.

Informations disponibles

Quand on joue à *Pokémon Go* on trouve des Pokémon sur notre chemin, mais également des œufs. Il faut parcourir une certaine distance pour faire éclore un œuf. Enfin, il est possible de faire évoluer un Pokémon à l'aide de friandises. Un fichier de données (*pokedex.csv*) recense l'ensemble des Pokémons utilisables dans le jeu.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Attributs du fichier
- num : Number of the Pokémon in the official Pokédex
- name : Pokémon name
- img : URL to an image of this Pokémon
- type : Pokémon type
- height : Pokémon height (m)
- weight : Pokémon weight (kg)
- candy : type of candy used to evolve Pokémon or given when transfered
- candy_count : amount of candies required to evolve
- egg : Number of kilometers to travel to hatch the egg
- weakness : Types of Pokémon this Pokémon is weak to
- next_evolution : Number of evolution of Pokémon

Attributs du fichier

- num : Number of the Pokémon in the official Pokédex
- name : Pokémon name
- img : URL to an image of this Pokémon
- type : Pokémon type
- height : Pokémon height (m)
- weight : Pokémon weight (kg)
- candy : type of candy used to evolve Pokémon or given when transfered
- candy_count : amount of candies required to evolve
- egg : Number of kilometers to travel to hatch the egg
- weakness : Types of Pokémon this Pokémon is weak to
- next_evolution : Number of evolution of Pokémon

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Activité 1 :
1. Ouvrir le fichier *pokedex.csv* pour observer les données fournies.
2. Créer un programme *pokemon.py*.
3. Importer les données du pokedex dans un tableau de dictionnaires **pokedex**. Typer correctement les données de poids et de taille.

Activité 1 :

1. Ouvrir le fichier *pokedex.csv* pour observer les données fournies.
2. Créer un programme *pokemon.py*.
3. Importer les données du pokedex dans un tableau de dictionnaires **`pokedex`**. Typer correctement les données de poids et de taille.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction


```
fichier = open("pokedex.csv")
data = csv.DictReader(fichier)
pokedex = []
for pokemon in data:
    dico = {}
    for cle, val in pokemon.items():
        # validation des données
        if cle == "height" or cle == "weight":
            val = float(val)

        dico[cle] = val
    # ajout du pokemon dans le tableau
    pokedex.append(dico)
fichier.close()
```


Code 1 – Importation des données

Correction

```
1  fichier = open("pokedex.csv")
2  data = csv.DictReader(fichier)
3  pokedex = []
4  for pokemon in data:
5      dico = {}
6      for cle, val in pokemon.items():
7          # validation des données
8          if cle == "height" or cle == "weight":
9              val = float(val)
10
11         dico[cle] = val
12     # ajout du pokemon dans le tableau
13     pokedex.append(dico)
14 fichier.close()
```

Code 1 – Importation des données

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Interface graphique
L'interface permet à l'utilisateur d'interagir avec les données à l'aide de boutons, boîtes de dialogue...

Figure – Exemple d'interface graphique

Interface graphique

L'interface permet à l'utilisateur d'interagir avec les données à l'aide de boutons, boîtes de dialogue...

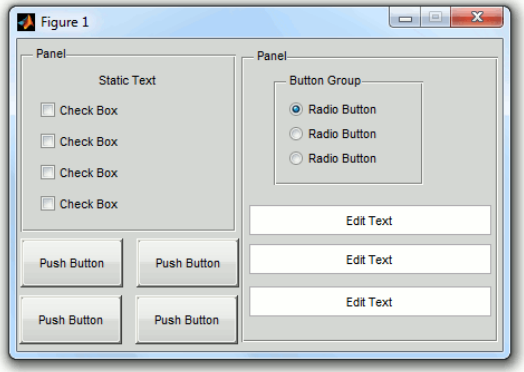


FIGURE – Exemple d'interface graphique

Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Bibliothèque tkinter
La bibliothèque tkinter fournit des *composants (widgets)* pour construire une interface graphique simple.

```
import tkinter
from tkinter import ttk

#création de la fenêtre
fenetre = tkinter.Tk()
fenetre.title("Pokemon Go")

# la construction des composants se placera ici

# dernière ligne du programme: met à jour les variables
fenetre.mainloop()
```

Code 2 – Créer une fenêtre d'interface

Bibliothèque tkinter

La bibliothèque tkinter fournit des *composants (widgets)* pour construire une interface graphique simple.

```
1  import tkinter
2  from tkinter import ttk
3
4  #création de la fenêtre
5  fenetre = tkinter.Tk()
6  fenetre.title("Pokemon Go")
7
8  # la construction des composants se placera ici
9
10 # dernière ligne du programme: met à jour les variables
11 fenetre.mainloop()
```

Code 2 – Créer une fenêtre d'interface

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Ajouter un composant

L'ajout d'un composant se déroule en trois étapes :
- Créer le composant.
- Placer le composant dans l'interface.
- Remplir le composant.

Ajouter un composant

L'ajout d'un composant se déroule en trois étapes :

- Créer le composant.
- Placer le composant dans l'interface.
- Remplir le composant.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Code 3 – Placer un *label*

dans la fenêtre


```
1 # création
2 etiquette = tkinter.Label(fenetre)
3 # placement
4 etiquette.pack()
5 # remplissage
6 etiquette["text"] = "Bonjour"
```

Code 3 – Placer un *label*

dans la fenêtre

Activité 2 : Construire une interface avec trois labels `nom, poids, taille`.

Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction

```
label_nom = tkinter.Label(fenetre)
label_nom.pack()
label_nom["text"] = "Bulbasaur"

label_poids = tkinter.Label(fenetre)
label_poids.pack()
label_poids["text"] = "6.9kg"

label_taille = tkinter.Label(fenetre)
label_taille.pack()
label_taille["text"] = "0.71m"
```

Code 4 – Création d'une carte Pokémon

Correction

```
label_nom = tkinter.Label(fenetre)
label_nom.pack()
label_nom["text"] = "Bulbasaur"

label_poids = tkinter.Label(fenetre)
label_poids.pack()
label_poids["text"] = "6.9kg"

label_taille = tkinter.Label(fenetre)
label_taille.pack()
label_taille["text"] = "0.71m"
```

Code 4 – Création d'une carte Pokémon

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction
Bulbasaur
6.9kg
0.71m
Figure – Interface obtenue

Correction

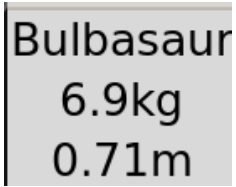
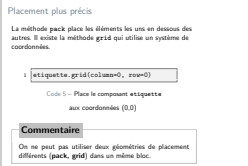


Figure – Interface obtenue

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement


Placement plus précis

La méthode pack place les éléments les uns en dessous des autres. Il existe la méthode grid qui utilise un système de coordonnées.

```
1 etiquette.grid(column=0, row=0)
```

Code 5 – Place le composant etiquette aux coordonnées (0,0)

Commentaire

On ne peut pas utiliser deux géométries de placement différents (pack, grid) dans un même bloc.

Placement plus précis

La méthode **pack** place les éléments les uns en dessous des autres. Il existe la méthode **grid** qui utilise un système de coordonnées.

```
1 etiquette.grid(column=0, row=0)
```

Code 5 – Place le composant **etiquette** aux coordonnées (0,0)

Commentaire

On ne peut pas utiliser deux géométries de placement différents (**pack, grid**) dans un même bloc.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Activité 3 : Présenter les trois labels sous la forme d'une grille.

nom
poids taille

Activité 3 : Présenter les trois labels sous la forme d'une grille.

nom
poids taille

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction

```
label_nom = tkinter.Label(fenetre)
label_nom.grid(column=0, row=0)
label_nom["text"] = "Bulbasaur"

label_poids = tkinter.Label(fenetre)
label_poids.grid(column=0, row=1)
label_poids["text"] = "6.9kg"

label_taille = tkinter.Label(fenetre)
label_taille.grid(column=1, row=1)
label_taille["text"] = "0.71m"
```

Code 6

Correction

```
label_nom = tkinter.Label(fenetre)
label_nom.grid(column=0, row=0)
label_nom["text"] = "Bulbasaur"

label_poids = tkinter.Label(fenetre)
label_poids.grid(column=0, row=1)
label_poids["text"] = "6.9kg"

label_taille = tkinter.Label(fenetre)
label_taille.grid(column=1, row=1)
label_taille["text"] = "0.71m"
```

Code 6

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Construire par bloc

Pour organiser les composants de manière plus ordonnée, il est préférable de ne pas tous les plaquer directement dans la fenêtre *tkinter* principale. On crée alors des blocs (`Frame`) pour découper notre interface.

Construire par bloc

Pokémon Go

Problématique

Informations disponibles

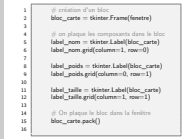
Interface graphique

Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images

Interaction entre les composants

Liste de choix : `Combobox`
Choisir un Pokémon : un événement

Pour organiser les composants de manière plus ordonnée, il est préférable de ne pas tous les plaquer directement dans la fenêtre *tkinter* principale. On crée alors des blocs (`Frame`) pour découper notre interface.



Code 7 – Création d'un bloc

```
1   # création d'un bloc
2   bloc_carte = tkinter.Frame(fenetre)
3
4   # on plaque les composants dans le bloc
5   label_nom = tkinter.Label(bloc_carte)
6   label_nom.grid(column=1, row=0)
7
8   label_poids = tkinter.Label(bloc_carte)
9   label_poids.grid(column=0, row=1)
10
11  label_taille = tkinter.Label(bloc_carte)
12  label_taille.grid(column=1, row=1)
13
14  # On plaque le bloc dans la fenêtre
15  bloc_carte.pack()
16
```

Code 7 – Création d'un bloc

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Remarque
Dans le code 7, la *Frame* bloc_carte a utilisé une géométrie grid alors que la fenêtre principale c'est pack qui est utilisé. Il n'y a pas d'incompatibilités car il s'agit de deux blocs différents.

Remarque

Dans le code 7, la *Frame* **`bloc_carte`** a utilisé une géométrie **`grid`** alors que la fenêtre principale c'est **`pack`** qui est utilisé. Il n'y a pas d'incompatibilités car il s'agit de deux blocs différents.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Remplissage des labels

Pour remplir le texte des composants on peut choisir de créer une fonction.

Activité 4 : Écrire la fonction `remplir_carte(num_pok: int)` → `None` qui remplit les trois labels créés en fonction du numéro de Pokémon choisi dans le Pokédex.

Remplissage des labels

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Pour remplir le texte des composants on peut choisir de créer une fonction.

> **Activité 4 :** Écrire la fonction `remplir_carte(num_pok: int)` → `None` qui remplit les trois labels créés en fonction du numéro de Pokémon choisi dans le Pokédex.

Correction

Code 8 – Fonction de remplissage

Commentaire

La fonction utilise ici des variables du programme principal. Avec nos connaissances actuelles, c'est la manière la plus simple de procéder.

Correction

```
1  def remplir_carte(num_pok: int) -> None:
2      # choix du pokemon dans le pokedex
3      pok = pokedex[num_pok]
4
5      label_nom["text"] = pok["name"]
6      label_poids["text"] = str(pok["weight"])+"kg"
7      label_taille["text"] = str(pok["height"])+"m"
```

Code 8 – Fonction de remplissage

Commentaire

La fonction utilise ici des variables du programme principal. Avec nos connaissances actuelles, c'est la manière la plus simple de procéder.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction

Dans le programme principal, l'appel 9 après la création des labes, affiche les informations du Pokemon 12.

1 remplir_carte(12)

Code 9 – Remplissage des labels

Correction

Dans le programme principal, l'appel 9 après la création des labes, affiche les informations du Pokemon 12.

```
1 remplir_carte(12)
```

Code 9 – Remplissage des labels

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement


Gestion des images

L'affichage d'une image demande un peu plus de travail. Il faut créer un objet PhotoImage qui sera ensuite affiché dans un composant label_photo. Commençons par créer le composant.

```
1 label_photo = tkinter.Label(bloc_carte)
2 label_photo.grid(column=1, row=0)
```

Code 10 – label_photo est placé à droite de label_nom

Gestion des images

L'affichage d'une image demande un peu plus de travail. Il faut créer un objet **`PhotoImage`** qui sera ensuite affiché dans un composant **`label_photo`**. Commençons par créer le composant.

```
1 label_photo = tkinter.Label(bloc_carte)
2 label_photo.grid(column=1, row=0)
```

Code 10 – **`label_photo`** est placé à droite de **`label_nom`**

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Il faut ensuite modifier la fonction **`remplir_carte`** pour afficher la photo.

```
1  def remplir_carte(num_pok: int) -> None:
2      global photo # garde une référence de l'image
3
4      pok = pokedex[num_pok]
5
6      # affichage de l'image
7      photo = tkinter.PhotoImage(file=pok["img"])
8      label_photo["image"] = photo
9
10     label_nom["text"] = pok["name"]
11     label_poids["text"] = str(pok["weight"])+"kg"
12     label_taille["text"] = str(pok["height"])+"m"
```

Commentaire

La ligne 2 évite que le *garbage collector* efface l'image à la sortie de la fonction. Cette notion est hors programme.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

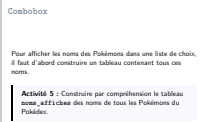
Interaction entre les composants

Notre programme n'est pour l'instant que peut utile : il faut changer *en dur (c'est à dire dans le programme)* le numéro du Pokémon à afficher. La prochaine étape consistera à créer une liste de choix dans l'interface graphique pour changer dynamiquement la carte à afficher.

Interaction entre les composants

Notre programme n'est pour l'instant que peut utile : il faut changer *en dur (c'est à dire dans le programme)* le numéro du Pokémon à afficher. La prochaine étape consistera à créer une liste de choix dans l'interface graphique pour changer dynamiquement la carte à afficher.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement


Combobox

Pour afficher les noms des Pokémons dans une liste de choix, il faut d'abord construire un tableau contenant tous ces noms.

Activité 5 : Construire par compréhension le tableau `noms_affiches` des noms de tous les Pokémons du Pokédex.

Combobox

Pour afficher les noms des Pokémons dans une liste de choix, il faut d'abord construire un tableau contenant tous ces noms.

> **Activité 5 :** Construire par compréhension le tableau **`noms_affiches`** des noms de tous les Pokémons du Pokédex.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Correction
noms_affiches = [pokedex[i]"name"] for i in range(len(pokedex))]
Code 11 – Tableau des noms

Correction

```
noms_affiches = [pokedex[i]"name"] for i in range(len(pokedex))]
```

Code 11 – Tableau des noms

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Combobox

Le composant Combobox crée une liste de choix.

```
# création du composant
combo_pok = ttk.Combobox(fenetre, values=
    noms_affiches)
# valeur par défaut
combo_pok.current(0)
# placement
combo_pok.pack()
```

Code 12 – Création du composant

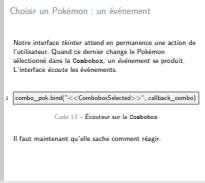
Combobox

Le composant **Combobox** crée une liste de choix.

```
1 # création du composant
2 combo_pok = ttk.Combobox(fenetre, values=
      noms_affiches)
3 # valeur par défaut
4 combo_pok.current(0)
5 # placement
6 combo_pok.pack()
```

Code 12 – Création du composant

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement


Choisir un Pokémon : un événement

Notre interface tkinter attend en permanence une action de l'utilisateur. Quand ce dernier change le Pokémon sélectionné dans la Combobox, un événement se produit. L'interface écoute les événements.

```
1 combo_pok.bind("<<ComboboxSelected>>", callback_combo)
```

Code 13 – Écouteur sur la Combobox

Il faut maintenant qu'elle sache comment réagir.

Choisir un Pokémon : un événement

Notre interface *tkinter* attend en permanence une action de l'utilisateur. Quand ce dernier change le Pokémon sélectionné dans la **`Combobox`**, un *événement* se produit. L'interface *écoute* les événements.

```
1 combo_pok.bind("<<ComboboxSelected>>", callback_combo)
```

Code 13 – Écouteur sur la **`Combobox`**

Il faut maintenant qu'elle sache comment réagir.

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement


Fonction de rappel (*callback*)

Quand l'utilisateur change la sélection dans la liste de choix, la fonction `callback_combo` est appelée. C'est une **fonction de rappel (callback)**. Un unique paramètre (noté ici `event`) est passé automatiquement à la fonction. Ce paramètre contient des informations sur le choix effectué.

```
1 def callback_combo(event):
2     # num contient l'indice de la ligne sélectionnée
      dans la liste
3     num = event.widget.current()
```

Code 14 – fonction de callback

Fonction de rappel (*callback*)

Quand l'utilisateur change la sélection dans la liste de choix, la fonction **`callback_combo`** est appelée. C'est une **fonction de rappel (callback)**. Un unique paramètre (noté ici **`event`**) est passé automatiquement à la fonction. Ce paramètre contient des informations sur le choix effectué.

```
1 def callback_combo(event):
2     # num contient l'indice de la ligne sélectionnée
      dans la liste
3     num = event.widget.current()
```

Code 14 – fonction de callback

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Nous pouvons alors utiliser cette information pour mettre à jour la carte Pokémon.

```
def callback_combo(event):
    """
    fonction de rappel quand on change de pokemon
    dans la combobox
    """
    num = event.widget.current()
    # mise à jour de la carte
    remplir_carte(num)
```

Nous pouvons alors utiliser cette information pour mettre à jour la carte Pokémon.

```
1  def callback_combo(event):
2      """
3      fonction de rappel quand on change de pokemon
4      dans la combobox
5      """
6      num = event.widget.current()
7      # mise à jour de la carte
8      remplir_carte(num)
```

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement

Code complet
Le code complet est récupérable à l'adresse suivante :
Code complet

Code complet

Le code complet est récupérable à l'adresse suivante :

Code complet

Pokémon Go
Problématique
Informations disponibles
Interface graphique
Bibliothèque tkinter
Ajouter un composant
Construire par bloc
Remplissage des labels
Gestion des images
Interaction entre les composants
Liste de choix : Combobox
Choisir un Pokémon : un événement