

Documentation du Code de l'Application Geometry Dash

Ce code implémente une interface graphique pour un jeu de type *Geometry Dash*, avec l'utilisation de la bibliothèque Tkinter pour l'interface utilisateur et sqlite3 pour la gestion des enregistrements de score. Voici une présentation détaillée de son fonctionnement, incluant la gestion des niveaux et de l'interface utilisateur.

Interface Utilisateur (Fenêtre Principale)

L'interface est basée sur Tkinter et se compose principalement d'une fenêtre principale avec plusieurs boutons et champs d'entrée. Voici les principales parties de l'interface :

Fenêtre Principale (fenetre)

- **Titre** : "Geometry Dash"
- **Dimensions** : 350x400 pixels
- **Couleur de fond** : Gris (#606060)

```
import ...
#-----
# INTERFACE DE LA PAGE
#-----
fenetre = Tk()
fenetre.title("Geometry Dash")
fenetre.geometry("350x400")
fenetre.config(bg="#606060")
```

Champs et Boutons

- **Champ de texte (compte1)** : Un champ de texte où l'utilisateur peut entrer son nom d'utilisateur.
- **Label de nom (texteLabel2)** : Affiche le texte du nom d'utilisateur. Il est mis à jour lorsque l'utilisateur clique sur le bouton "Confirmer".
- **Bouton "Confirmer"** : Lorsqu'il est cliqué, il appelle la fonction `affichage()` qui met à jour le label `texteLabel2` pour afficher le nom de l'utilisateur saisi.

```
def affichage(): 1 usage
    name.set("Nom d'utilisateur : " +compte1.get()+"")
    compte1.delete( first: 0,END)
    name=StringVar()
    name.set("Nom d'utilisateur")
    texteLabel2 = Label(fenetre, textvariable = name)
    texteLabel2.pack()
    compte1 = Entry()
    bouton3= Button(fenetre , text= "Confirmer", command = affichage )
    compte1.pack()
    bouton3.pack()
```

Base de Données SQLite

Le code utilise **SQLite** pour gérer les enregistrements de score des joueurs :

- **Connexion à la base de données** : Une connexion en mémoire est établie via `sqlite3.connect(':memory:')`.
- **Création de la table record** : Une table est créée pour stocker l'ID, le nom du joueur, et son score record.
 - **MLD Records**(id , nom , record) , clé primaire id

```
connexion = sqlite3.connect(':memory:')
curseur=connexion.cursor()
curseur.execute('''CREATE TABLE IF NOT EXISTS records(
    id INTEGER PRIMARY KEY,
    nom TEXT,
    record INTEGER)''')
```

Fonctionnalités (Gestion des Niveaux)

L'application offre trois niveaux différents pour le jeu. Chaque niveau est représenté par une nouvelle fenêtre avec des éléments graphiques.

Fonction niva() : Niveau 1

- **Fenêtre** : Un Tk avec un canevas blanc de taille 1000x1000 pixels.
- **Graphique** : Un fond blanc, une ligne horizontale et un rectangle bleu représentant un obstacle.
- **Bouton "Quittez"** : Permet de fermer le niveau.

```
def niva(): 1 usage
class jeu(Tk):
    def __init__(self):
        Tk.__init__(self)
        self.canevas = Canvas(self,bg= "white", height = 1000, width= 1000)
        self.canevas.pack()
        self.canevas.create_line(4,600,10000,600, fill = "#606060",width=100)
        self.canevas.create_rectangle(19,535,29,535,fill = "Blue", width =30)
        fenniv1=jeu()
        fenniv1.title("Niveau1")
        bouton1= Button(fenniv1, text = "Quittez",bg="red", height = 4 , width = 13 , command = fenniv1.destroy )
        bouton1.pack()
        bouton1.place(x=800 , y= 20)

        nom = Label
        fenetre.mainloop()
```

Fonction nivb() : Niveau 2

- **Fenêtre** : Un Tk avec un canevas bleu de taille 1000x1000 pixels.
- **Graphique** : Une ligne horizontale violette et un rectangle bleu foncé.
- **Bouton "Quittez"** : Permet de fermer le niveau.

```
def nivb(): 1 usage
class oo(Tk):
    def __init__(self):
        Tk.__init__(self)
        self.canevas = Canvas(self,bg= "blue", height = 1000, width= 1000)
        self.canevas.pack()
        self.canevas.create_line(4,600,10000,600, fill = "purple",width=100)
        self.canevas.create_rectangle(19,535,29,535,fill = "dark blue", width =30)
        fenniv2=oo()
        fenniv2.title("Niveau 2")
        bouton1= Button(fenniv2, text = "Quittez",bg="red", height = 4 , width = 13 , command = fenniv2.destroy )
        bouton1.pack()
        bouton1.place(x=800 , y= 20)
        nom = Label
        fenetre.mainloop()
```

Fonction nivc() : Niveau 3

- **Fenêtre** : Un Tk avec un canevas rouge de taille 1000x1000 pixels.
- **Graphique** : Une ligne bleue et un rectangle bleu foncé.
- **Bouton "Quittez"** : Permet de fermer le niveau.

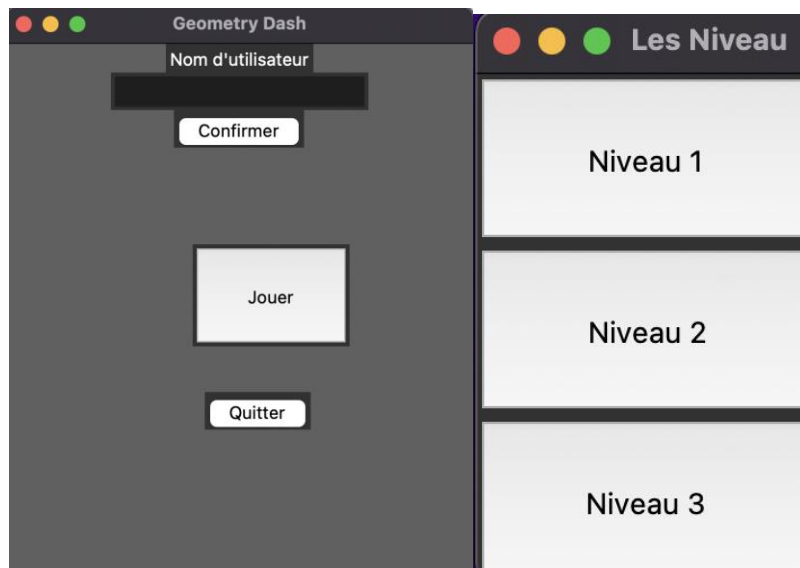
```
def nivc(): 1 usage
class zz(Tk):
    def __init__(self):
        Tk.__init__(self)
        self.canevas = Canvas(self,bg= "red", height = 1000, width= 1000)
        self.canevas.pack()
        self.canevas.create_line(4,600,10000,600, fill = "blue",width=100)
        self.canevas.create_rectangle(19,535,29,535,fill = "dark blue", width =30)
fenniv3=zz()
fenniv3.title("Niveau 3")
bouton1= Button(fenniv3, text = "Quittez",bg="red", height = 4 , width = 13 , command = fenniv3.destroy )
bouton1.pack()
bouton1.place(x=800 , y= 20)
nom = Label
fenetre.mainloop()
```

Gestion des Niveaux

Le bouton "Jouer" dans la fenêtre principale permet à l'utilisateur de choisir un niveau parmi trois disponibles :

1. **Niveau 1** : Appel de la fonction niva() pour ouvrir le premier niveau.
2. **Niveau 2** : Appel de la fonction nivb() pour ouvrir le deuxième niveau.
3. **Niveau 3** : Appel de la fonction nivc() pour ouvrir le troisième niveau.

Chaque niveau est une nouvelle fenêtre avec son propre canevas pour dessiner des éléments graphiques comme des lignes et des rectangles représentant le gameplay.



```

def page_des_niveau(): 1 usage
class niv0(Tk):
    def __init__(self):
        Tk.__init__(self)
        self.canevas = Canvas(self, bg="blue", height=100000, width=10000)

fena = niv0()
niv1 = Button(fena, text = "Niveau 1", bg = "white", height=4,width=13, command =niva)
niv1.pack()
niv2 = Button(fena, text = "Niveau 2", bg = "white", height=4,width=13, command =nivb)
niv2.pack()
niv3 = Button(fena, text="Niveau 3", bg="white", height=4, width=13, command= nivc)
niv3.pack()
fena.title("Les Niveau")
fena.mainloop()

```

Gestion des Boutons

- **Bouton "Jouer"** : Affiche la fenêtre de sélection des niveaux lorsqu'il est cliqué.
- **Bouton "Quitter"** : Permet de fermer l'application principale (fenêtre principale).

```

bouton2= Button(fenetre,text="Jouer", heigh="4",width="9", command = page_des_niveau )
bouton2.pack()
bouton2.place(x= 140 , y=150)
bouton1= Button(fenetre,text="Quitter", command = fenetre.destroy , bg = "red")
bouton1.pack()
bouton1.place(x=149, y=260)

connexion.close()
fenetre.mainloop()

```

Conclusion

Ce code implémente une interface de jeu de type *Geometry Dash* avec une gestion des utilisateurs et des niveaux. La base de données SQLite permet de stocker les scores, et l'interface graphique réalisée avec Tkinter permet d'interagir avec l'utilisateur pour choisir un niveau et jouer.