ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS
UNIVERSITÉ HASSAN II DE CASABLANCA

RAPPORT TECHNIQUE

Application StudyHub

Application d'Apprentissage Gamifiée en Python

Suivi de progression • Quiz interactifs • Gamification

Membres du Projet

KAZOURY CHAIMAE
FATIMA-EZZAHRA LAGDEM
BIBY MARYAM

Encadré par : Mustafa Hain
ENSAM Casablanca

Année Universitaire 2025 - 2026

Table des matières

1 Introduction

StudyHub est une application éducative complète développée en Python avec l'interface graphique Tkinter. Elle a été conçue pour offrir une expérience d'apprentissage interactive de la Programmation Orientée Objet (POO) à travers un système gamifié motivant.

Objectifs Clés

- Faciliter l'apprentissage progressif de la POO en Python
- Offrir une interface moderne et intuitive
- Motiver les apprenants par la gamification (points, niveaux, séries)
- Suivre la progression individuelle de chaque utilisateur
- Proposer des contenus adaptés au niveau de compétence

Technologies Utilisées

- **Python 3** : Langage principal
- **Tkinter** : Framework d'interface graphique
- **SQLite3** : Base de données embarquée
- **ttk** : Widgets modernes et stylisés

2 Architecture et Base de Données

Structure Modulaire

L'application est organisée en trois modules principaux :

- `constante.py` : Configurations, couleurs, questions de quiz
- `base_de_donnee.py` : Gestion de la persistance avec SQLite
- `interface.py` : Interface graphique et logique métier

La classe `BaseDeDonnees` centralise toutes les opérations de stockage avec trois tables essentielles :

- **joueurs** : informations utilisateur (nom, score, niveau, date de dernière connexion, série de connexion)
- **inscriptions_formations** : inscriptions des utilisateurs aux différentes formations
- **progres_lessons** : suivi de l'avancement des leçons (non commencé, en cours, terminé)

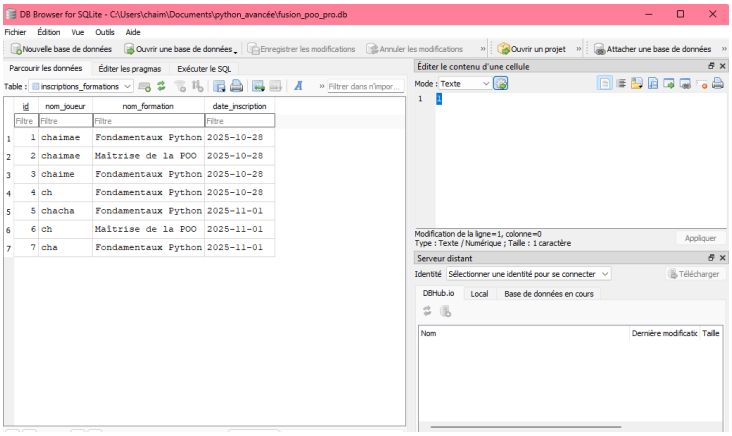

FIGURE 1 – Schéma de la base de données SQLite

```python
def creer_table(self):
    cursor = self.connexion.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS joueurs (nom TEXT PRIMARY KEY, score
        INTEGER DEFAULT 0, niveau INTEGER DEFAULT 1)''')
    cursor.execute('''CREATE TABLE IF NOT EXISTS inscriptions_formations (id INTEGER
        PRIMARY KEY AUTOINCREMENT, nom_joueur TEXT, nom_formation TEXT, date_inscription
        TEXT, UNIQUE(nom_joueur, nom_formation))''')
    cursor.execute('''CREATE TABLE IF NOT EXISTS progres_lessons (id INTEGER PRIMARY KEY
        AUTOINCREMENT, nom_joueur TEXT, nom_formation TEXT, lesson TEXT, statut TEXT
        DEFAULT 'not_started', UNIQUE(nom_joueur, nom_formation, lesson))''')
        self.connexion.commit()
def charger_joueur(self, nom):
    cursor = self.connexion.cursor()
    cursor.execute('SELECT score, niveau, last_login_date, login_streak
                    FROM joueurs WHERE nom = ?', (nom,))
    resultat = cursor.fetchone()
    if resultat:
        return resultat[0], resultat[1], resultat[2], resultat[3]
    else:
        cursor.execute('INSERT INTO joueurs (nom) VALUES (?)', (nom,))
        self.connexion.commit()
        return 0, 1, None, 0
```

Listing 1 – Méthode de chargement d'un joueur

3 Interface Utilisateur - Écran d'Accueil

Fonctionnalités d'Accueil

- Design épuré avec les couleurs du projet
- Champ de saisie du prénom avec validation
- Bouton "Commencer"
- Gestion automatique des nouveaux utilisateurs

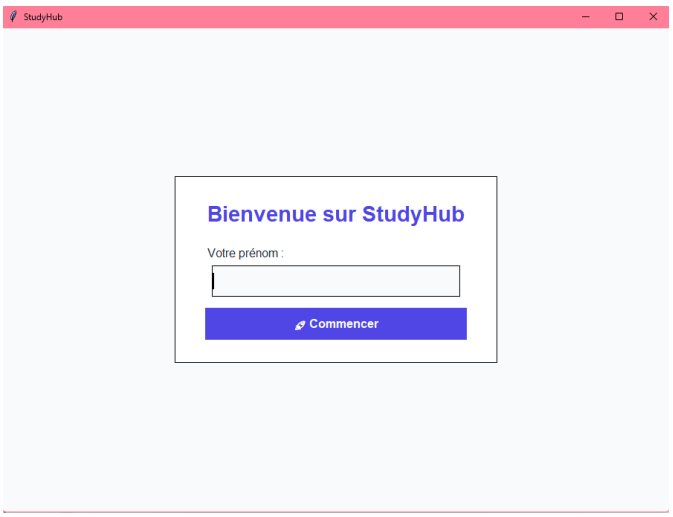

FIGURE 2 – Écran d'accueil de StudyHub

```
1
2       def ecran_dashboard(self):
3           self.nettoyer_contenu()
4           tk.Label(self.frame_contenu, text=f"Bienvenue, {self.nom} !", font=("Arial",22,"
                bold"), bg=COULEUR_FOND, fg=COULEUR_TEXTE).pack(anchor="w")
5           tk.Label(self.frame_contenu, text="Pr t    relever un nouveau d fi aujourd'hui
                ?", font=("Arial",12), bg=COULEUR_FOND, fg=COULEUR_MUTED).pack(anchor="w",
                pady=(0,20))
6           chart_frame = tk.Frame(self.frame_contenu,bg=COULEUR_CARD,relief="solid",bd=1,
                padx=20,pady=20); chart_frame.pack(fill="x", pady=10)
7           tk.Label(chart_frame,text="       Votre Progression",font=("Arial",16,"bold"),bg=
                COULEUR_CARD,fg=COULEUR_PRIMAIRE).pack(anchor="w")
8           canvas = tk.Canvas(chart_frame,width=600,height=200,bg=COULEUR_CARD,
                highlightthickness=0); canvas.pack(pady=10)
9           max_niveau = 3; bar_width = 80; spacing = 120; max_height = 150
10          for i in range(1,max_niveau+1):
11              x = 50 + (i-1)*spacing
12              height = (self.score/(max_niveau*10))*max_height if i<=self.niveau else 20
13              color = COULEUR_SUCCES if i<=self.niveau else COULEUR_MUTED
14              canvas.create_rectangle(x,180-height,x+bar_width,180,fill=color,outline="");
                    canvas.create_text(x+bar_width//2,195,text=f"Niveau {i}", font=("Arial"
                    ,10), fill=COULEUR_TEXTE)
15          tk.Label(chart_frame,text=f"Score Total: {self.score} | Niveau Actuel: {self.
                niveau} | S rie: {self.login_streak} jours",
16                   font=("Arial",11),bg=COULEUR_CARD,fg=COULEUR_MUTED).pack(pady=5)
17          card_quiz = tk.Frame(self.frame_contenu,bg=COULEUR_CARD,relief="solid",bd=1,padx
                =20,pady=20); card_quiz.pack(fill="x",pady=10)
18          tk.Label(card_quiz,text="       Prendre un Quiz",font=("Arial",16,"bold"),bg=
                COULEUR_CARD,fg=COULEUR_PRIMAIRE).pack(anchor="w")
19          tk.Label(card_quiz,text=f"Testez vos connaissances et passez au niveau {self.
                niveau+1} !",font=("Arial",11),bg=COULEUR_CARD,fg=COULEUR_MUTED).pack(anchor=
                "w",pady=5)
20          self.creer_bouton(card_quiz,"Commencer le Quiz",self.demarrer_quiz,'primary').
                pack(anchor="e", pady=10)
```

Listing 2 – Création de l'écran d'accueil

4 Interface Principale - Dashboard

Composants du Dashboard

- **Sidebar de navigation** : Nom, score, niveau, série de jours et boutons
- **Graphique de progression** : Barres visuelles montrant l'avancement
- **Carte Quiz** : Invitation à passer un quiz pour progresser

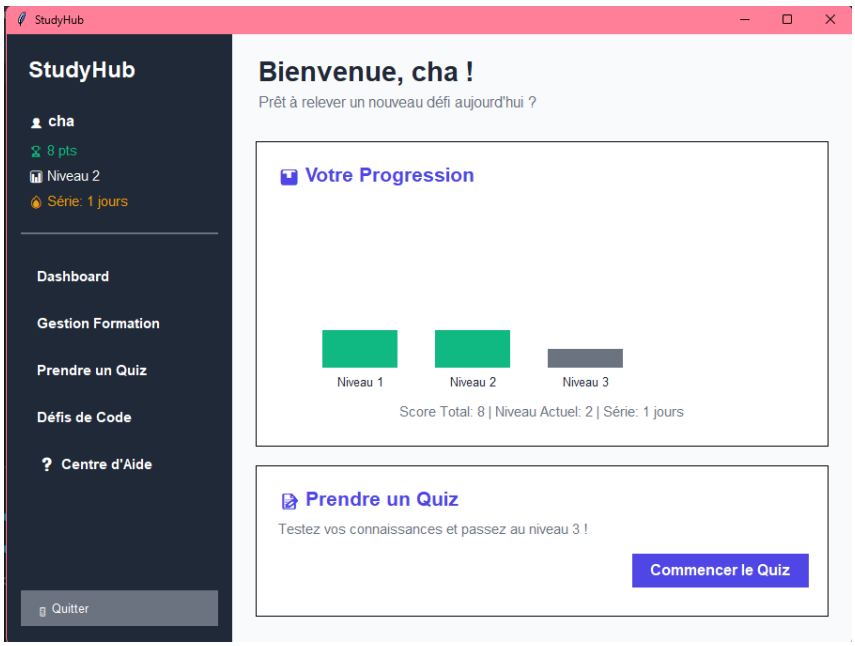

FIGURE 3 – Dashboard principal avec graphique de progression

```python
def ecran_dashboard(self):
    self.nettoyer_contenu()
    tk.Label(self.frame_contenu, text=f"Bienvenue, {self.nom} !",
             font=("Arial",22,"bold"), bg=COULEUR_FOND,
             fg=COULEUR_TEXTE).pack(anchor="w")

    chart_frame = tk.Frame(self.frame_contenu,bg=COULEUR_CARD,
                           relief="solid",bd=1,padx=20,pady=20)
    chart_frame.pack(fill="x", pady=10)

    canvas = tk.Canvas(chart_frame,width=600,height=200,
                       bg=COULEUR_CARD,highlightthickness=0)
    canvas.pack(pady=10)

    max_niveau = 3
    bar_width = 80
    spacing = 120

    for i in range(1, max_niveau+1):
        x = 50 + (i-1)*spacing
        height = (self.score/(max_niveau*10))*150 if i<=self.niveau else 20
        color = COULEUR_SUCCES if i<=self.niveau else COULEUR_MUTED
        canvas.create_rectangle(x,180-height,x+bar_width,180,
                                fill=color,outline="")
        canvas.create_text(x+bar_width//2,195,text=f"Niveau {i}",
                           font=("Arial",10), fill=COULEUR_TEXTE)
```

Listing 3 – Création du dashboard avec graphique

5 Système de Quiz Interactif

Explication

Le système de quiz est le coeur pédagogique de l'application. Son objectif est d'évaluer les connaissances de l'utilisateur sur la POO et de le faire progresser de niveau en niveau. Les questions sont adaptées au niveau actuel, garantissant une difficulté croissante.

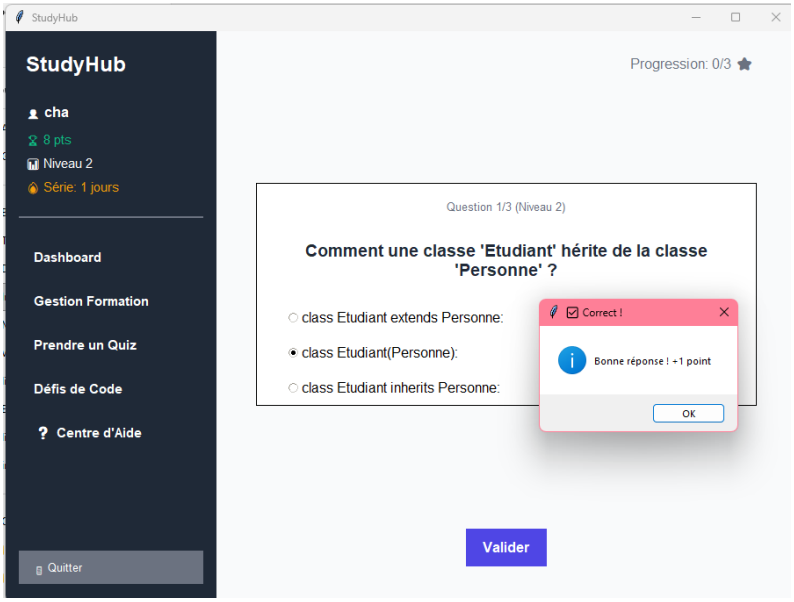

FIGURE 4 – Interface du quiz avec progression

```python
def verifier_reponse(self):
    r = self.var_option.get()
    q = self.questions_niveau_actuel[self.index_question]

    if r == -1:
        messagebox.showwarning("Attention","Selectionne une reponse !")
        return

    if r == q["reponse"]:
        self.score += 1
        self.points_du_niveau += 1
        messagebox.showinfo("Correct !","Bonne reponse ! +1 point")
    else:
        messagebox.showerror("Incorrect",
            f"La bonne reponse etait : {q['options'][q['reponse']]}")

    self.db.sauvegarder_joueur(self.nom, self.score, self.niveau)
    self.mettre_a_jour_stats()
    self.index_question += 1
    self.afficher_question()

def passer_niveau_suivant(self):
    self.niveau += 1
    self.points_du_niveau = 0
    self.db.sauvegarder_joueur(self.nom, self.score, self.niveau)
    self.mettre_a_jour_stats()
    messagebox.showinfo("Bravo !",f"Niveau {self.niveau} debloque !")
    self.ecran_dashboard()
```

Listing 4 – Validation et progression dans le quiz

6 Gestion des Formations

Explication

Le module de formations offre un parcours d'apprentissage structuré. L'objectif est de fournir des contenus théoriques organisés en formations progressives, chacune composée de plusieurs leçons. Les formations sont débloquées selon le niveau atteint par l'utilisateur. Trois formations sont disponibles :

- **Fondamentaux Python (Niveau 1)** : Variables, boucles, fonctions
- **Maîtrise de la POO (Niveau 2)** : Classes, héritage, polymorphisme
- **Python Avancé (Niveau 3)** : Décorateurs, générateurs, métaclasses

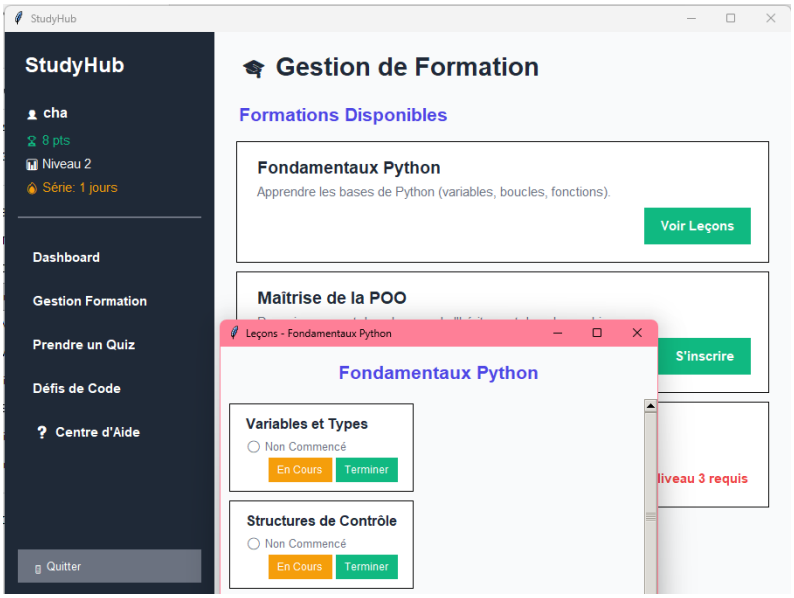

FIGURE 5 – Liste des formations avec statuts de déverrouillage

```python
def ecran_gestion_formation(self):
    self.nettoyer_contenu()
    tk.Label(self.frame_contenu, text="Gestion de Formation",
             font=("Arial",22,"bold")).pack(anchor="w", pady=(0,20))

    inscrites = [f[0] for f in self.db.get_formations_joueur(self.nom)]

    for f in FORMATIONS_DATA:
        card = tk.Frame(self.frame_contenu,bg=COULEUR_CARD,
                        relief="solid",bd=1,padx=20,pady=15)
        card.pack(fill="x",pady=5)

        tk.Label(card,text=f['nom'],font=("Arial",14,"bold")).pack(anchor="w")
        tk.Label(card,text=f['desc'],font=("Arial",11)).pack(anchor="w",pady=2)

        if f['nom'] in inscrites:
            self.creer_bouton(card,"Voir Lecons",
                lambda ff=f:self.voir_lessons(ff),'success').pack(anchor="e")
        elif self.niveau >= f['niveau_requis']:
            self.creer_bouton(card,"S'inscrire",
                lambda n=f['nom']:self.inscrire_formation(n),'success').pack(anchor="e")
        else:
            tk.Label(card,text=f"Niveau {f['niveau_requis']} requis",
                     font=("Arial",11,"bold"),fg=COULEUR_ERREUR).pack(anchor="e")
```

Listing 5 – Affichage et gestion des formations

7 Défis de Code et Centre d'Aide

Explication

Les défis de code permettent la pratique concrète des concepts de POO. L'utilisateur peut écrire son code dans un éditeur intégré, comparer avec la solution de référence, et passer au défi suivant. L'objectif est de renforcer l'apprentissage par la pratique.

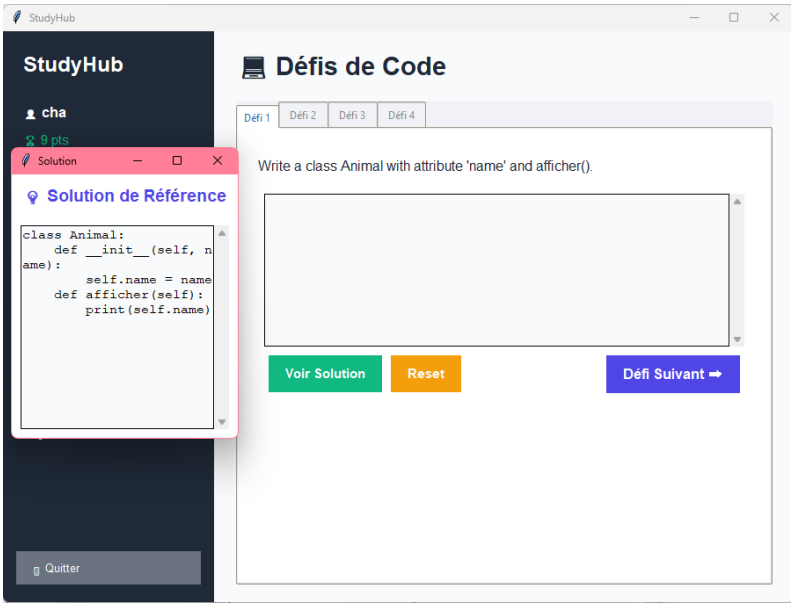

FIGURE 6 – Interface des défis de code avec onglets

```
def ecran_defis_code(self):
    self.nettoyer_contenu()
    nb = ttk.Notebook(self.frame_contenu)

    for i,(enonce,sol) in enumerate(CODE_CHALLENGES):
        f = tk.Frame(nb,bg=COULEUR_CARD,padx=20,pady=20)
        nb.add(f,text=f"Defi {i+1}")

        tk.Label(f,text=enonce,font=("Arial",12)).pack(pady=10,anchor="w")

        txt = scrolledtext.ScrolledText(f,height=10,font=("Courier New",11))
        txt.pack(fill="x",pady=10)

        bf = tk.Frame(f,bg=COULEUR_CARD)
        bf.pack(fill="x")
        self.creer_bouton(bf,"Voir Solution",
            lambda s=sol:self.voir_solution(s),'success').pack(side="left")
        self.creer_bouton(bf,"Reset",
            lambda t=txt:t.delete("1.0",tk.END),'warning').pack(side="left")

    nb.pack(expand=True,fill="both")
```

Listing 6 – Création de l'interface des défis

FIGURE 7 – Centre d'aide avec FAQ

8 Éléments de Gamification

Système de Récompenses

- **+1 point** par bonne réponse au quiz
- **Passage au niveau supérieur** après 3 bonnes réponses
- **3 niveaux progressifs** avec difficulté croissante
- **Déverrouillage** de nouvelles formations selon le niveau
- **Série de jours** pour encourager la connexion régulière

Remerciements

Nous exprimons nos sincères remerciements à notre enseignant pour son engagement, son encadrement et la qualité de ses explications tout au long de ce projet. Son accompagnement constant, ses conseils pertinents et sa disponibilité ont largement contribué à la réussite de ce travail.