



S E R A P H I N
. L E G A L

Mini Projet NSI 2023

The Knight's Quest

Dossier compte rendu

Rédigé le 19/04/2023
Dernière mise à jour : 09/05/2023

Des questions au sujet de notre mini projet ?

Contactez nous pour en discuter :

- enon.lois@nsi.fr
- didry.jules@nsi.fr

Sommaire

Sommaire	2
I- Descriptif du projet	2
II- Équipe du projet	2
III- Cahier des charges	3
IV – Algorithmes de points particuliers	4-5
V – Carnet de Bord	6-8
VI – Annexes	9

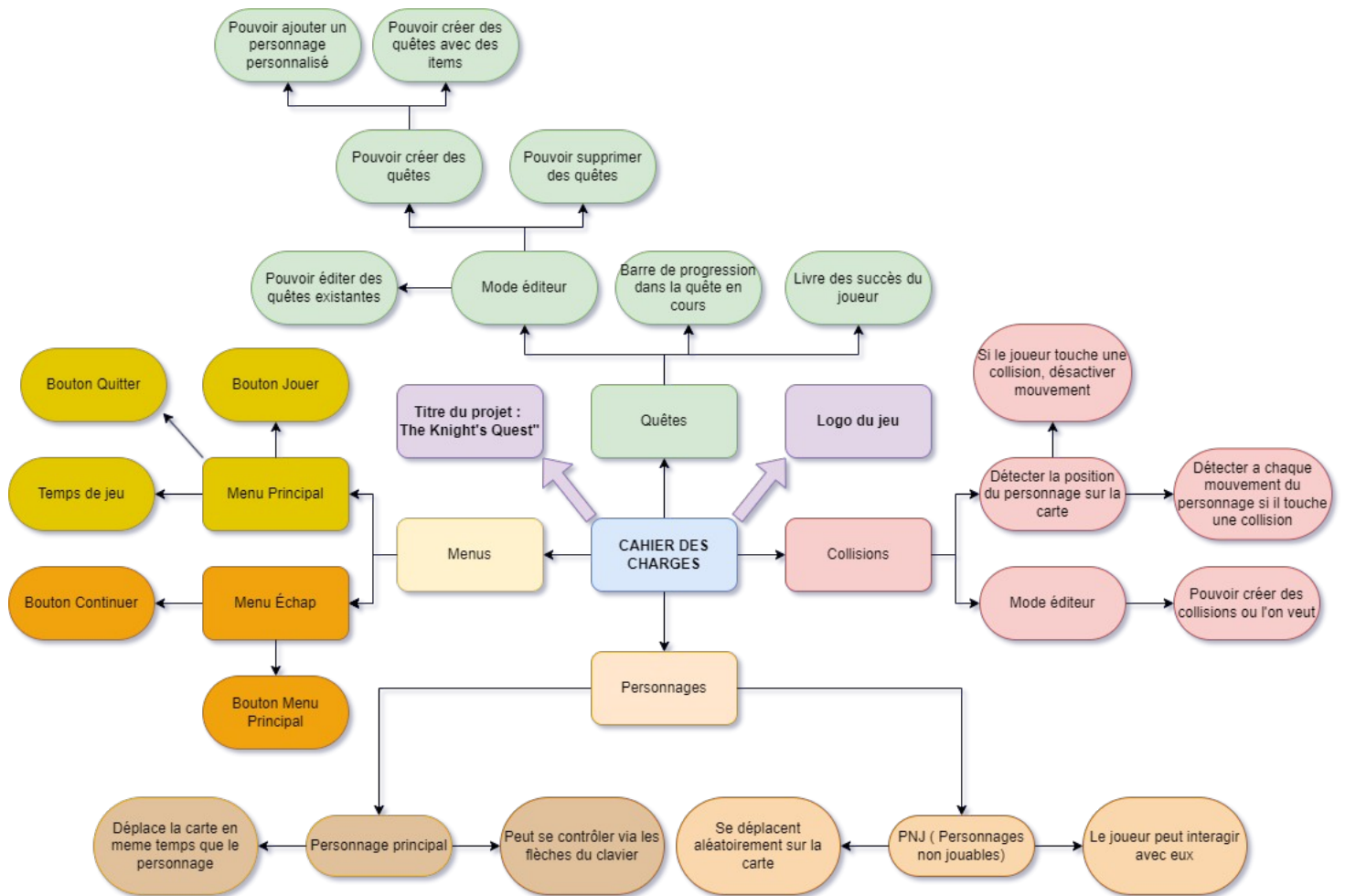
I- Descriptif du projet

Notre mini projet de NSI est un RPG (Role Playing Game) soit en français un jeu de rôle. Celui-ci à entièrement été codé et pensé par nous même.

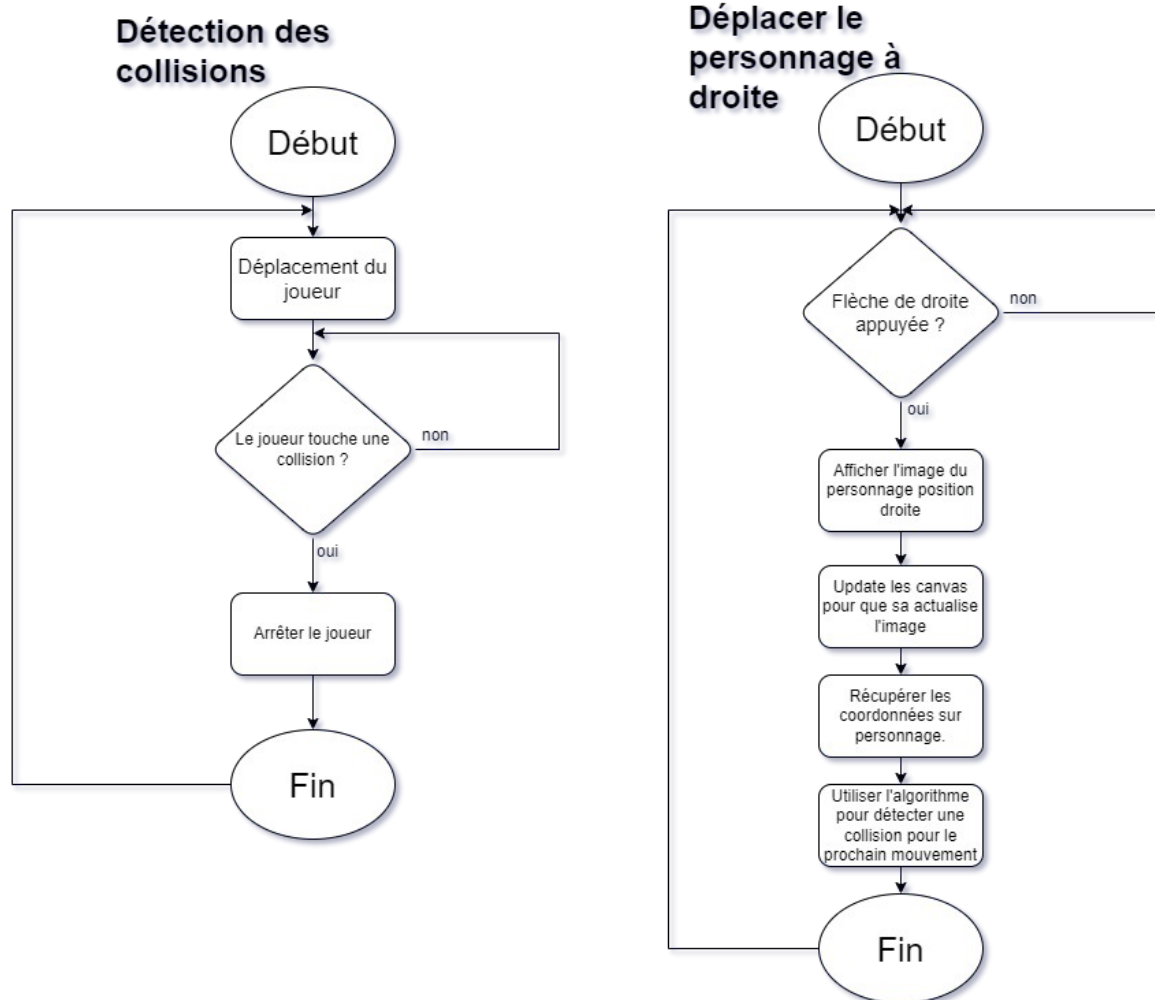
II- Équipe du projet

Nom – Prénom	Rôle projet	Société	E-mail de contact
ENON Loïs	Chef de projet	LPO- Turgot- Limoges	enon.lois@gmail.com
DIDRY Jules	Second chef de projet	LPO- Turgot- Limoges	didry.jules@nsi.fr

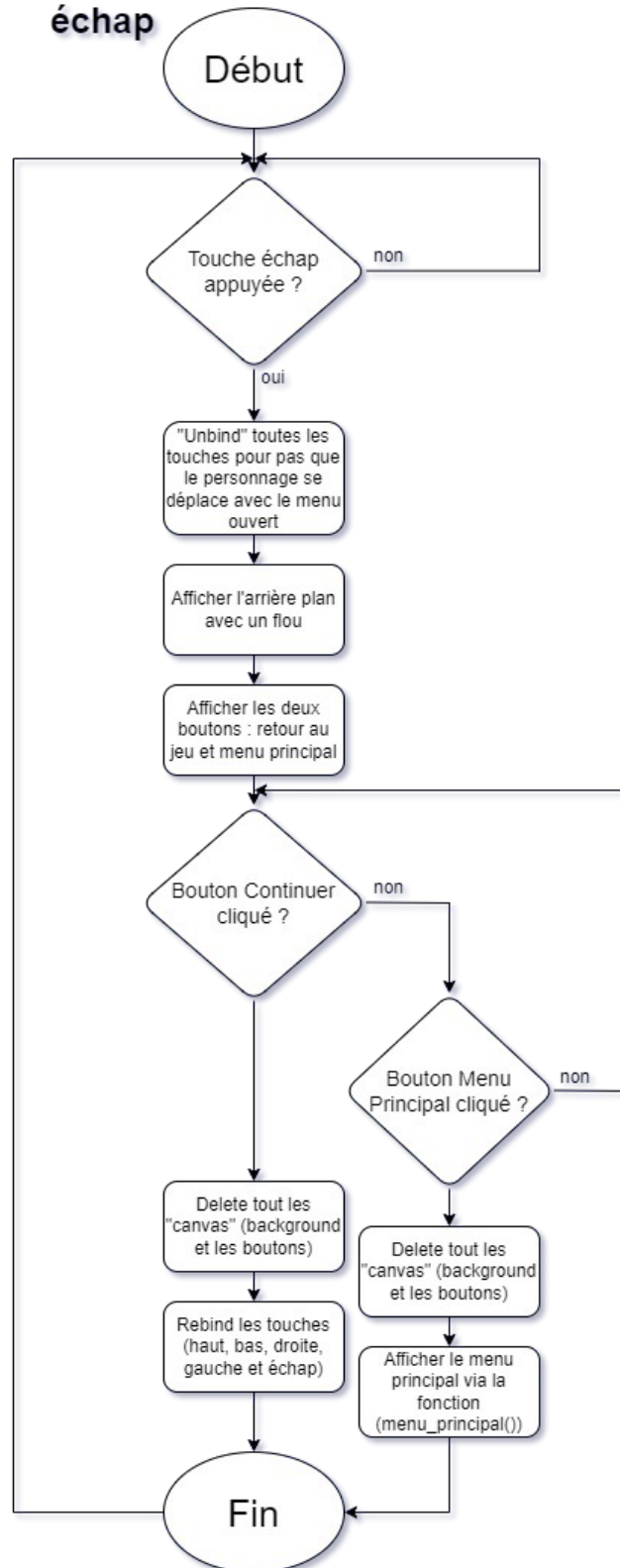
III – Cahier des charges



IV – Algorithmes et points particuliers



Afficher le menu échap



V – Carnet de bord

01/11/2022 : Découverte des mini-projets, mise en binôme.

03/11/2022 : Premières discussions a propos du projet, premières idées qui se dégagent du projet.

15/11/2022 : Concrétisation de l'idée du projet, nous avons donc décidé de partir sur un RPG (Role Playing Game).

17/11/2022 : Établissement de la première version du cahier des charges.

22/11/2022 : Recherche du nom du projet et du logo qu'on pourrait donner au projet.

24/11/2022 : Recherche des premières idées de quêtes mais surtout de la carte que l'on peut choisir pour le fond du jeu.

26/11/2022 : Jules : Changement de la carte de fond du jeu, car elle était trop petite et pas bien optimisée pour ce que l'on voulait faire.

Loïs : Recherche de la photo du personnage principal, et détournement du personnage pour les futures collisions.

13/12/2022 : Mise en place du dossier du projet, en mettant les assets (image d'arrière plan, personnage principal)

20/12/2022 : Loïs : Mise en place des premiers bouts de code : importation des bibliothèques et importation des assets (personnage principal et fond du jeu)

Jules : Recherche sur les premières quêtes du jeu (quête quand le joueur arrive sur le jeu)

05/01/2023 : Loïs : Mise en place de l'écran de chargement du jeu, via la bibliothèque tkinter.

Jules : Recherche des futurs PNJ (Personnage Non Jouable) qui seront mis dans le futur sur la carte.

10/01/2023 : Suppression de l'écran de chargement, car il ne servait à rien.

12/01/2023 : Réflexion sur le menu principal et ses fonctionnalités
(Bouton Jouer pour lancer le jeu, Bouton quitter et un texte indiquant le temps de jeu du joueur)

17/01/2023 : Loïs : Mise en place du menu principal, placement des boutons et ajouts du texte uniquement du temps de jeu.
 Jules : Création du design des deux boutons du menu principal (Bouton Jouer et Bouton Quitter)

19/01/2023 : Loïs : Correction des premiers bugs.
 Jules : Modification du texte permettant de faire le temps de jeu du joueur.

31/01/2023 : Modification du nom du projet (The Knight's Quest) et modification du logo.

02/02/2023 : Loïs : Ajout de la fonction permettant de cliquer sur le bouton jouer du menu principal, qui détruit les boutons et le fond transparent et met la carte du jeu.
 Jules : Ajout de la fonction permettant de cliquer sur le bouton quitter permettant de quitter le jeu et implémentation du personnage dans le jeu de façon optimisée.

21/02/2023 : Expérimentations sur le déplacement du personnage uniquement.

23/02/2023 : Le personnage peut maintenant se déplacer dans la fenêtre et expérimentations du déplacement du fond en même temps que le personnage de façon plus lente.

28/02/2023 : Intégration des deux codes de test (mouvement du personnage et mouvement du fond) dans le code principal en appuyant sur le bouton Jouer.

07/03/2023 : Optimisation du code et mise en place des premiers commentaires.

16/03/2023 : Réflexion sur le menu échap du joueur (Boutons, fonctionnalités)

30/03/2023 : Loïs : Mise en place du menu échap : ajout du bind dans le code et positionnement des boutons.
 Jules : Création design des deux boutons du menu échap (Bouton Continuer et Bouton Menu Principal)

16/04/2023 : Loïs : Mise en fonction du bouton du menu échap : Continuer, permettant de revenir sur le jeu et de cacher les boutons et le fond flou.

18/04/2023 : Jules : Mise en fonction du bouton du menu échap : Menu Principal, permettant de revenir dans le menu principal du jeu et de réinitialiser la position du personnage.

25/04/2023 : Premières réflexions sur le moyen de crée des collisions au moment du déplacement du personnage.

27/04/2023 : Premiers essais sur les collisions, on a dupliquer le projet pour garder une version sans bugs et sans collisions. Ce premier essai n'est pas une grande réussite.

02/05/2023 : Loïs : Second essai sur les collisions, celles-ci fonctionnent, implémentation dans le code principal.
Jules : Reflexion sur les quêtes principales du jeu.

04/05/2023 : Loïs : Ajout d'une fonction permettant de créer des collisions rapidement et facilement en faisant la touche espace.
Jules : Réflexion et ajout du moyen d'enregistrer dans un fichier .json les collisions crée en faisant la touche espace.
Ajout de la fonctionnalité que des que le bouton Jouer est cliqué, les collisions chargent via le fichier .json via une boucle dans le code.

09/05/2023 : Finalisation des quêtes et correction des derniers bugs du jeu.

VI – Annexes

```
#####
# IMPORTATION DES BIBLIOTHÈQUES
# NÉCESSAIRES AU PROGRAMME
#####
from tkinter import *
from PIL import Image, ImageTk
from time import *
import json

#####
# DELARATION DE LA FENETRE
# AVEC LA TAILLE, LE TITRE
# ET L'ICONE
#####
fenetre = Tk()
fenetre.geometry("1920x1080")
fenetre.title("The Knight's Quest")
fenetre.iconbitmap("assets/icon/chateau.ico")

#####
# CRÉATION DE LA FENÊTRE
# GRACE A CANVAS
#####
canvas = Canvas(fenetre, width=1920, height=1080, bg="white")
canvas.pack()

def quit_button_event(): #définiton de la fonction quit_button_event qui sert
a fermer le programme
    exit()

#####
# DELARATION DES IMAGES
#####
background_img_normal_src =
ImageTk.PhotoImage(Image.open('assets/background/background.png'))
background_img_blur_src =
ImageTk.PhotoImage(Image.open('assets/background/background_blur.jpg'))
personnage_avant_src =
ImageTk.PhotoImage(Image.open('assets/personnages/marche_avant.png'))
personnage_arriere_src =
ImageTk.PhotoImage(Image.open('assets/personnages/marche_arrière.png'))
personnage_gauche_src =
ImageTk.PhotoImage(Image.open('assets/personnages/marche_gauche.png'))
```

```
personnage_droite_src =
ImageTk.PhotoImage(Image.open('assets/personnages/marche_droite.png'))
play_button_src =
ImageTk.PhotoImage(Image.open("assets/buttons/start_button.png"))
quit_button_src =
ImageTk.PhotoImage(Image.open("assets/buttons/quit_button.png"))
resume_button_src =
ImageTk.PhotoImage(Image.open("assets/buttons/resume_button.png"))
menu_principal_button_src =
ImageTk.PhotoImage(Image.open("assets/buttons/menu_principal_button.png"))

def play_button_event(variable) :
    global personnage
    canvas.delete(background_img_blur)
    canvas.delete(play_button)
    canvas.delete(quit_button)

    background_img_normal = canvas.create_image(0, 0, anchor=NW,
image=background_img_normal_src)
    personnage = canvas.create_image(960, 540, anchor=NW,
image=personnage_avant_src)
    canvas.move(background_img_normal, -2760, -4240)

    f = open('data.json')
    data = json.load(f)

    rectangles = []

    for i in data['collisions']:
        rect = canvas.create_rectangle(i["x1"], i["y1"], i["x2"], i["y2"],
fill=i["color"], outline=i["color"])
        rectangles.append(rect)

    f.close()

    coord_carres = []
    for i in data['collisions']:
        coord_carres.append([i["x1"], i["y1"], i["x2"], i["y2"], i["id"]])

    def detect_collision(canvas, mover, data, x, y, coord_carre_move):

        new_coord_carre_move = [coord_carre_move[0] + x, coord_carre_move[1] + y,
coord_carre_move[2] + x, coord_carre_move[3] + y]
        can_move = True
        for i, coord_carre in enumerate(coord_carres):
```

```
    if new_coord_carre_move[0] < coord_carre[2] and new_coord_carre_move[2] >
coord_carre[0] and new_coord_carre_move[3] > coord_carre[1] and
new_coord_carre_move[1] < coord_carre[3]:
    can_move = False
    break
    if can_move:
        x_perso = x/5
        y_perso = y/5
        canvas.move(mover, x_perso, y_perso)
        canvas.move(background_img_normal, -x, -y)
        for i, rect in enumerate(rectangles):
            canvas.move(rect, -x, -y)
            coord_carres[i] = [coord_carres[i][0]-x, coord_carres[i][1]-y,
coord_carres[i][2]-x, coord_carres[i][3]-y]

def deplacer_droite(e):
    canvas.itemconfig(personnage, image=personnage_droite_src)
    canvas.update()
    coords_perso = canvas.bbox(personnage)
    detect_collision(canvas, personnage, data, 10, 0, coords_perso)

def deplacer_gauche(e):
    canvas.itemconfig(personnage, image=personnage_gauche_src)
    canvas.update()
    coords_perso = canvas.bbox(personnage)
    detect_collision(canvas, personnage, data, -10, 0, coords_perso)

def deplacer_bas(e):
    canvas.itemconfig(personnage, image=personnage_arriere_src)
    canvas.update()
    coords_perso = canvas.bbox(personnage)
    detect_collision(canvas, personnage, data, 0, 10, coords_perso)

def deplacer_haut(e):
    canvas.itemconfig(personnage, image=personnage_avant_src)
    canvas.update()
    coords_perso = canvas.bbox(personnage)
    detect_collision(canvas, personnage, data, 0, -10, coords_perso)

def menu_echap(e) :
    fenetre.unbind("<Left>")
    fenetre.unbind("<Right>")
    fenetre.unbind("<Up>")
    fenetre.unbind("<Down>")
    fenetre.unbind("<Escape>")
    background_img_blur = canvas.create_image(0, 0, anchor=NW,
image=background_img_blur_src)
```

```
canvas.move(background_img_blur, -980, -1980)

resume_button = canvas.create_image(150, 750, anchor=NW, image =
resume_button_src)

menu_principal_button = canvas.create_image(1270, 750, anchor=NW, image =
menu_principal_button_src)

canvas.tag_bind(resume_button, "<Button-1>", lambda event :
quit_escape_menu())
canvas.tag_bind(menu_principal_button, "<Button-1>", lambda event :
menu_principal())

def quit_escape_menu() :
    canvas.delete(background_img_blur)
    canvas.delete(resume_button)
    canvas.delete(menu_principal_button)
    fenetre.unbind("<Left>", deplacer_gauche)
    fenetre.unbind("<Right>", deplacer_droite)
    fenetre.unbind("<Up>", deplacer_haut)
    fenetre.unbind("<Down>", deplacer_bas)
    fenetre.unbind("<Escape>", menu_echap)

def menu_principal() :
    canvas.delete(personnage)
    canvas.delete(background_img_normal)
    background_img_blur = canvas.create_image(0, 0, anchor=NW,
image=background_img_blur_src)
    canvas.move(background_img_blur, -980, -1980)

    play_button = canvas.create_image(150, 750, anchor=NW, image =
play_button_src)

    quit_button = canvas.create_image(1270, 750, anchor=NW, image =
quit_button_src)

    canvas.tag_bind(quit_button, "<Button-1>", lambda event :
quit_button_event())
    canvas.tag_bind(play_button, "<Button-1>", lambda event :
play_button_event(canvas))
    fenetre.unbind("<Left>")
    fenetre.unbind("<Right>")
    fenetre.unbind("<Up>")
    fenetre.unbind("<Down>")
    fenetre.unbind("<Escape>")
```

```
playtime = canvas.create_text(960, 975, text="Temps de jeu : ",
fill="white", font=('Helvetica 15 bold'))
f = open('data.json')
data = json.load(f)
canvas.itemconfig(playtime, text="Temps de jeu : " +
str(data["playtime"]) + " min")
```

```
fenetre.bind("<Left>", deplacer_gauche)
fenetre.bind("<Right>", deplacer_droite)
fenetre.bind("<Up>", deplacer_haut)
fenetre.bind("<Down>", deplacer_bas)
fenetre.bind("<Escape>", menu_echap)
```

```
background_img_blur = canvas.create_image(0, 0, anchor=NW,
image=background_img_blur_src)
canvas.move(background_img_blur, -980, -1980)
```

```
play_button = canvas.create_image(150, 750, anchor=NW, image =
play_button_src)
```

```
quit_button = canvas.create_image(1270, 750, anchor=NW, image =
quit_button_src)
```

```
playtime = canvas.create_text(960, 975, text="Temps de jeu : ", fill="white",
font=('Helvetica 15 bold'))
f = open('data.json')
data = json.load(f)
canvas.itemconfig(playtime, text="Temps de jeu : " + str(data["playtime"]) + "
min")
```

```
canvas.tag_bind(quit_button, "<Button-1>", lambda event : quit_button_event())
canvas.tag_bind(play_button, "<Button-1>", lambda event :
play_button_event(canvas))
```

```
def task():
    with open('data.json', 'r+') as f:
        data = json.load(f)
        data['playtime'] = data['playtime'] + 1
        canvas.itemconfig(playtime, text="Temps de jeu : " + str(data["playtime"])
+ " min")
        f.seek(0)
        json.dump(data, f, indent=4)
        f.truncate()
    fenetre.after(60000, task)
```

```
fenetre.after(60000, task)
```

```
fenetre.mainloop()
```