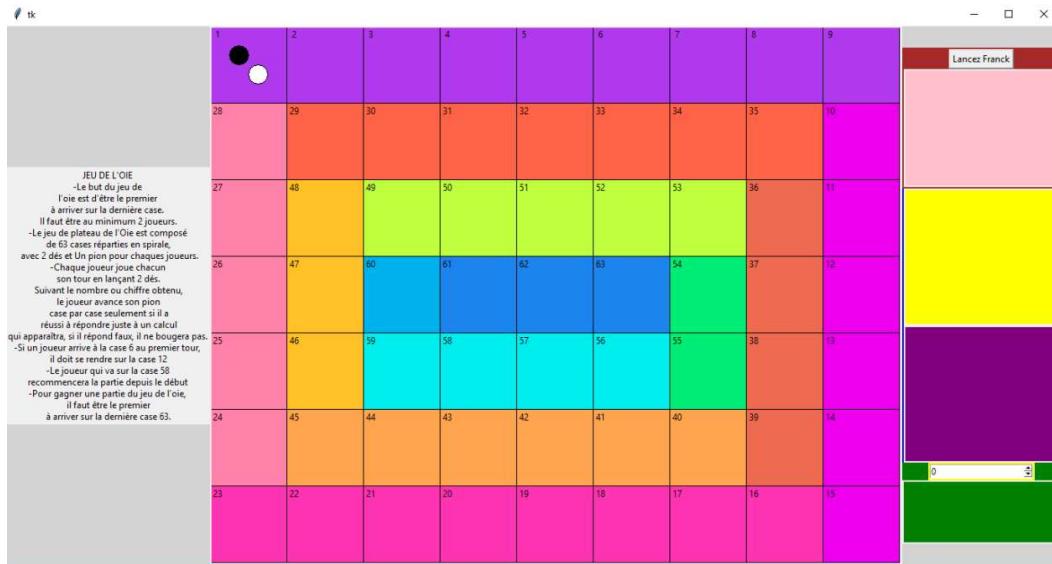


PROJET ISN 2019

Jeu de l'oie mathématique



Réalisé par Elliott FEBVRE LAPAUW, Nicolas GOUBET et Léo ESSIQUE

membres de l'équipe pédagogique: M. LEVEQUE

SOMMAIRE :

I/ Cahier des charges

- *La nature du projet*
- *Les contraintes techniques*

II/ Références et recherches documentaires

III/ Répartition du travail

IV/ La réalisation

V/ Le fonctionnement

VI/ Les prolongements envisageables

VII/ Bilans personnels

1/ Cahier des charges

Nature du projet :

Le projet “Jeu de l'oie mathématique” est un jeu numérique qui a été programmé par les élèves Elliott FEBVRE LAPAUW, Nicolas GOUBET et Léo ESSIQUE.

Nous avons choisi de faire notre projet en nous inspirant d'un jeu de société de notre enfance qui est extrêmement connu : “ Le jeu de l'oie”.

Le jeu de l'oie est à l'origine un jeu de société de parcours où l'on déplace des pions en fonction du résultat de deux dés. Le jeu de l'oie comprend 63 cases disposées en spirales enroulées vers l'intérieur et comportant un certain nombre de pièges. Le but est d'arriver le premier à la dernière case. Le jeu est un jeu de hasard pur.

Nous avons décidé de réduire le rôle du hasard en obligeant les joueurs à répondre à un calcul mathématique pour pouvoir lancer les dés. Ces calculs restent plutôt simples afin d'être à la portée des plus jeunes tout en les forçant ludiquement à un peu plus de réflexion que le jeu original.

Pour la réalisation de notre projet, nous avons utilisé la bibliothèque sur Python, Tkinter qui permet la création d'interfaces graphiques.

Les contraintes techniques :

Étant notre première réalisation avec Tkinter, notre principale contrainte fut de s'adapter à l'outil et de l'apprendre afin de pouvoir mieux estimer les contraintes de réalisation et la quantité de travail qu'il fallait pour la réussite de notre projet.

Après cette estimation, les contraintes étaient les suivantes :

- Adapter ce jeu en version informatique alors que nous n'avions joué qu'à la version sur table.
- Créer un jeu jouable pour 2 personnes.
- Créer une version de ce jeu en utilisant des calculs tout en restant instinctif pour un joueur.
- Apprendre les mathématiques au niveau primaire de manière ludique
- Rendre impossible une quelconque triche.

II/ Références et recherches documentaires

Nous avons utilisé Tkinter pour la réalisation de ce projet, ne connaissant pas cette outil, nous avons donc cherché beaucoup d'informations sur son utilisation.

Nous avons majoritairement utilisé le site :

<https://python-django.dev/page-tkinter-interface-graphique-python-tutoriel>

Afin de réussir dans les temps, nous avons dû travailler ensemble en dehors de nos séances au lycée. Pour cela, nous avons utilisé le site PythonTutor et l'application Discord qui nous a permis de travailler ensemble simultanément.

III/ Répartition du travail

Comme dit précédemment, nous avons pratiquement réalisé entièrement notre projet en groupe pour plusieurs raisons.

Premièrement, nous avions tous les trois une expérience très limitée dans Tkinter alors nous avons décidé d'affronter les problèmes ensemble pour les résoudre plus rapidement.

Deuxièmement, tout le groupe devait comprendre l'ensemble du programme.

Bien que nous nous sommes perpétuellement entraînés pour toutes les tâches, nous pouvons accorder le mérite de la réalisation du plateau et de la gestion des calculs à Eliott ; l'affichage des fenêtres et l'aspect du jeu à Nicolas ; la gestion des tours de chaque joueur à Léo.

IV| Réalisation

Lors de la réalisation de notre projet, nous nous sommes efforcés de garder un programme propre afin qu'une personne extérieure puisse lire et comprendre notre programme sans se perdre.

Nous avons donc tout d'abord importé les bibliothèques qui nous seront utile pour la suite du programme. Nous nous sommes servis de la bibliothèque Tkinter pour la réalisation graphique ; la fonction randint de la bibliothèque random afin de réaliser nos dés et générer nos calculs ; la bibliothèque math pour nos calculs.

1. Le programme commence tout d'abord par demander le nom des joueurs et déterminer aléatoirement grâce à la fonction lancer1 qui commencera la partie en premier et l'enregistre dans la variable m puis affiche le résultat dans une nouvelle fenêtre.
2. Après ces actions, le programme affiche la fenêtre de jeu avec la frame attribuée pour les règles, pour le plateau et pour les actions qui représente la zone d'interaction où se situe les calculs et les dés. Puis pour chaque Frame :
 - 2.1. Le programme affiche les règles grâce à un label dans la frame des règles.
 - 2.2. Puis le programme construit le plateau de jeu grâce à des boucles FOR dans un canvas. Il fait aussi apparaître 2 ovales comme pion sur la case 1 dans la frame du plateau.
 - 2.3. La majeure partie du programme se situe ensuite dans la frame actions : c'est ici que les joueurs peuvent interagir avec le jeu, cette frame a été divisée en 2 nouvelles frames : la frame pour les dés et la frame pour les calculs.
 - 2.3.1. La frame des dés est ensuite divisée en 2 autres frames :

* Une frame (**rouge**) pour le bouton du lancer de dés qui démarre une fonction lancerJ1 ou lancerJ2 qui alterne en fonction des tours des joueurs. La variable m qui a précédemment déterminé quel joueur allait démarrer permet au programme de commencer par l'une des 2 fonctions. Cette fonction "lancer" permet de lancer au hasard une génération de calcul.

Ce programme contient 3 générations différentes de calcul, soit une addition, soit une soustraction ou soit une multiplication. Ces fonctions génèrent 2 nombres entiers entre 1 et 100 ou 1 et 20 pour la multiplication puis la fonction récupère la réponse du joueur provenant d'une spinbox située dans la frame calculs et vérifie si le résultat est juste ou non.

Si la réponse est juste, la fonction est autorisée à lancer le déplacement qui est déterminé par 2 randint représentant les dés, le résultat est ensuite affichée dans la deuxième frame des dés. Il suffit ensuite au programme de vérifier si le résultat obtenu fait déplacer le pion sur un événement spécial afin de le déclencher comme pour la victoire. Le programme se sert aussi de ce résultat pour savoir si le joueur a fait un nombre supérieur à la ligne du plateau dans laquelle il se trouve afin de pivoter.

Cependant si la réponse est fausse, rien de tout cela n'est fait, le pion ne se déplace pas et le tour passe à l'autre joueur. Pour le changement de tour, le bouton du joueur précédent est simplement supprimé puis remplacé par celui de l'autre joueur.

* La frame **rose** correspond à l'affichage du résultat des dés. Pour cela, un canvas a préalablement été créé dans cette frame. Puis lorsque le programme a généré le résultat, il a aussi créé un texte ciblé contenant le résultat au centre de ce canvas.

2.3.2. La frame des calculs est divisée en 4 autres frames :

La frame **jaune** contient un canvas dans lequel est affiché suite à la vérification des fonctions additions / soustractions / multiplications si le résultat répondu par le joueur est juste ou faux.

La frame **violette** contient un canvas dans lequel est affiché le calcul généré par les fonctions additions / soustractions / multiplications

La frame contenant la spinbox permet aux joueurs d'écrire leurs réponses aux calculs générés. Cette réponse a besoin d'être validée par un bouton pour qu'elle soit envoyée en vérification aux fonctions additions / multiplications / soustractions

La frame **verte** contient le bouton 'Valider' qui envoi ensuite le résultat de la spinbox en vérification. Ce bouton ne peut être activé qu'une seule fois par tour pour éviter une tentative de triche.

3. Une fois qu'un pion arrive sur la dernière case, le programme déclenche la fin du jeu et créer une fenêtre qui annonce le nom du vainqueur de la partie. Un bouton est disponible pour fermer cette fenêtre et la fenêtre du jeu.

V/ Fonctionnement

Le programme complet fonctionne à chaque fois.

En effet, pour le prouver, nous pouvons lancer le programme :

- Premier joueur aléatoire :

Le joueur qui débute la partie doit être tiré aléatoirement car le premier joueur aura un tour d'avance sur le deuxième s'il finit en premier la partie. Le premier joueur ne doit pas être déterminé en fonction de s'il est joueur 1 ou joueur 2. Pour le vérifier, il suffit simplement de relancer plusieurs parties.

- L'aspect mathématique :

Le joueur ne doit pouvoir avancer que seulement s'il répond correctement au calcul affiché. Pour le vérifier, il suffit d'appuyer sur lancer des dés puis d'appuyer sur valider sans entrer de valeur dans la spinbox.

- L'alternance des joueurs :

Chaque joueur ne doit pouvoir déclencher la génération d'un calcul qu'une seule fois par tour avant que le tour de l'adversaire ne se déclenche. Pour le vérifier il suffit de faire une partie.

- Les événements spéciaux :

Selon la version originale du jeu de l'oie, il doit y avoir des événements qui se déclenchent quand on s'arrête sur certaines cases et qui impliquent un déplacement du pion (6 = Allez à la case 12 ; 58 = retourner au départ ; 63 = victoire). Pour le vérifier, il suffit de faire une partie et de tomber sur ces cases.

Il est possible de rajouter dans le programme autant d'événements spéciaux que nous le désirons.

- La victoire :

La partie doit se terminer lorsqu'un joueur atteint la dernière case du plateau et doit annoncer que le joueur a gagné. Pour le vérifier, il suffit de finir une partie.

VII / Les prolongements envisageables

Ceux-ci sont de plusieurs natures ;

- Amélioration de l'esthétique du jeu
 - Enlever les couleurs qui nous ont permis de nous repérer avec les frames.
 - Personnaliser le plateau en améliorant le design des cases et en ajoutant des éléments graphiques sur les cases qui déclenchent des événements spéciaux.
 - Voir le déplacement des pions à travers le plateau, sans voir le pion simplement disparaître d'une case à la case qu'il a atteint avec les dés.
 - Ajouter une animation pour le lancement des dés, comme un gif lorsque qu'on appuie sur le bouton
 - Que chaque joueur puisse choisir sa couleur de pion.
 - régler les derniers problèmes (il en reste 2 mineurs)
- Ajouter des événements spéciaux :
 - Devoir arriver pile sur la case 63 pour gagner la partie.
 - Ajouter les cases de la version originale du jeu : la prison, le labyrinthe et l'hôtel.
- Ajouter la possibilité de jouer à 3 ou 4 joueurs.

VIII / Bilans Personnels

Nicolas :

Ma collaboration avec Léo et Elliott dans ce projet d'ISN m'a beaucoup apporté. Bien qu'absent lors de la majorité des séances prévues pour avancer sur le projet en cours, j'ai totalement adoré réaliser ce projet durant les nombreuses fois que nous nous sommes vus en dehors des cours.

Travailler sur Tkinter était vraiment très enrichissant car l'interface graphique nous permettait de suivre en direct nos actions et de pouvoir comprendre l'utilité de ce qu'on avait pu apprendre durant l'année.

Notre dynamique de travail en groupe nous a permis de gagner beaucoup de temps et, il faut bien l'avouer, fut très plaisante lorsque nous nous sommes retrouvés face à des obstacles.

Ce projet m'a conforté dans mon envie future de travailler dans le domaine informatique. Bien que ce programme n'était pas d'une très grande complexité, j'ai vraiment hâte de me lancer dans un nouveau projet !

Elliott :

La réalisation de ce projet d'ISN m'a beaucoup aidé à travailler en groupe et à réussir à mieux connaître ceux avec qui je travaillais et aussi apprendre leur manière de fonctionner. Globalement ce projet m'a plonger dans cette nouvelle matière que j'ai beaucoup apprécié et

dans laquelle je me suis investi et m'a permis de m'investir dans un groupe.

Ce travail m'a aussi permis de comprendre le langage de tkinter et a aussi maîtriser des notions vu au cours de l'année que je n'avais pas forcément compris totalement à force de les manipuler comme par exemple les définitions dont je ne comprenais pas totalement le principe ni le fonctionnement.

j'ai aussi découvert ce que pouvait apporter la programmation. Que ce soit lorsque l'on se retrouve bloqué à cause de quelque chose qu'on ne connaît pas et que l'on arrive pas à résoudre et où on s'énerve. Où quand on réussit à surpasser cette erreur et la satisfaction de voir son programme marcher et évoluer.

Nicolas et moi étions plus ou moins au même niveau cependant avec l'aide de Léo qui avait fait l'option ICN et qui connaissait un peu mieux cet espace nous avons pu nous en sortir et nous avons fini par travailler tous ensemble en même temps afin de tout comprendre et de s'aider mutuellement afin de terminer ce projet.

Léo :

Ce travail d'équipe a été enrichissant pour moi autant sur la méthode de travail qui permet des échanges passionnants que sur le travail lui-même qui m'a permis d'approfondir l'utilisation de Tkinter. J'ai adoré construire ce jeu de l'oie mathématique.

J'ai également adoré faire le html.

Cela m'a permis d'acquérir une connaissance à propos des id qui m'ont permis de pouvoir faire les ancrés de la page HTML.

Cela n'a pas été toujours simple mais c'est tellement valorisant lorsque l'on résout des difficultés de construction du programme qu'on en oublie les heures passées à travailler comme par exemple avec les déplacements ou le changement du tour des joueurs.

Pour conclure ce travail en équipe s'est très bien passé et je suis plutôt satisfait du résultat obtenu.

Annexes :

Programme Python :

```
from tkinter import *
from random import randint
from math import *

#Début de Jeu
joueur1 = input ("Joueur 1, Quel est votre nom ?")
joueur2 = input ("Joueur 2, Quel est votre nom ?")

def lancer1():
    global qui_commence
    global m
    bouton2 = Button(fenetre_debut_jeu, text = "SUIVANT", command = fenetre_debut_jeu.destroy)
    bouton2.pack(side = BOTTOM)
    bouton2.destroy()
    qui_commence = randint(1,2)
    if qui_commence == 1:
        m=joueur1
    else:
        m=joueur2

fenetre_debut_jeu = Tk()
label = Label(fenetre_debut_jeu, text = "Nous allons maintenant déterminer qui va commencer par un jet de dés et celui qui obtient le plus grand nombre commence!!!")
label.pack()
bouton = Button(fenetre_debut_jeu, text = "LANCER", command = lancer1)
bouton.pack()
fenetre_debut_jeu.mainloop()

#fenetre le résultat précédent :
fenetre_pileouface = Tk()
frame3 = Frame(fenetre_pileouface, width = 500, height = 500)
label1 = Label(frame3, text = (m,'va commencer la partie'))
label1.pack()
bouton = Button(frame3, text = 'Jouer', command = fenetre_pileouface.destroy)
bouton.pack()
frame3.pack()
fenetre_pileouface.mainloop()

#Gestion Fenêtre
fenetre_jeu = Tk()

jeu = Frame(fenetre_jeu, width = 1400, height = 700, bg = "lightgrey")
jeu.pack()

#Règles
regles = Frame(jeu, width=200, height=700, borderwidth=0,bg="purple")
regles.pack(side=LEFT)
label = Label(regles, text="JEU DE L'OIE\nLe but du jeu de l'oie est d'être le premier à arriver sur la dernière case.\n Il faut être au minimum 2 joueurs.\n Le jeu de plateau de l'Oie est composé de 63 cases réparties en spirale,\n avec 2 dés et Un pion pour chaque joueur.\n Chaque joueur joue chacun son tour en lançant 2 dés.\n Suivant le nombre ou chiffre obtenu,\n le joueur avance son pion\ncase par case seulement si il a réussi à répondre juste à un calcul\nqui apparaîtra, si il répond faux, il ne bougera pas.\n Si un joueur arrive à la case 6 au premier tour,\n il doit se rendre sur la case 12\nLe joueur qui va sur la case 58\nrecommencera la partie depuis le début\nPour gagner une partie du jeu de l'oie,\n il faut être le premier à arriver sur la dernière case 63.")
label.pack()
```

```

actions = Frame(jeu, width=200, height=700, borderwidth=0, bg="blue")
actions.pack(side=RIGHT)

plateau = Canvas(jeu, width = 900, height = 700, borderwidth = 0)

#Plateau de Jeu
for i in range (9):
    plateau.create_rectangle(0+100*i,0,100+100*i,100, fill = "darkorchid2")
    plateau.create_text(10+100*i,10, text = i+1)
for i in range (6):
    plateau.create_rectangle(800,100+100*i,900,200+100*i, fill = 'magenta2')
    plateau.create_text(810,110+100*i, text = i+10)
for i in range (9):
    plateau.create_rectangle(600-100*i,600,800-100*i,700, fill = 'maroon1')
    plateau.create_text(710-100*i,610, text = i+16)
for i in range (5):
    plateau.create_rectangle(0,500-100*i,100,600-100*i, fill = 'palevioletred1')
    plateau.create_text(10,510-100*i, text = i+24)
for i in range (7):
    plateau.create_rectangle(100+100*i,100,200+100*i,200, fill = "tomato")
    plateau.create_text(110+100*i,110, text = i+29)
for i in range (4):
    plateau.create_rectangle(700,200+100*i,800,300+100*i, fill = "coral2")
    plateau.create_text(710,210+100*i, text = i+36)
for i in range (6):
    plateau.create_rectangle(600-100*i,500,700-100*i,600, fill = "tan1")
    plateau.create_text(610-100*i,510, text = i+40)
for i in range (3):
    plateau.create_rectangle(100,400-100*i,200,500-100*i, fill = "goldenrod1" )
    plateau.create_text(110,410-100*i, text = i+46)
for i in range (5):
    plateau.create_rectangle(200+100*i,200,300+100*i,300, fill = "olivedrab1")
    plateau.create_text(210+100*i,210, text = i+49)
for i in range (2):
    plateau.create_rectangle(600,300+100*i,700,400+100*i, fill = "springgreen2")
    plateau.create_text(610,310+100*i, text = i+54)
for i in range (4):
    plateau.create_rectangle(500-100*i,400,600-100*i,500, fill = "cyan2")
    plateau.create_text(510-100*i,410, text = i+56)
for i in range (1):
    plateau.create_rectangle(200,300-100*i,300,400-100*i, fill = "deepskyblue2")
    plateau.create_text(210,310-100*i, text = i+60)
for i in range (3):
    plateau.create_rectangle(300+100*i,300,400+100*i,400, fill = "dodgerblue2")
    plateau.create_text(310+100*i,310, text = i+61)
joueur1_noir= plateau.create_oval(25,25,50,50, fill="black")
joueur2_blanc= plateau.create_oval(50,50,75,75, fill="white")

plateau.pack()

```

```

#gestion Dés

des = Frame(actions, width = 200, height=350, borderwidth=2, bg='brown')
des.pack(side=TOP)

lancer_des = Frame(des, width = 200, height=175 ,borderwidth=0, bg='pink')
lancer_des.pack(side=TOP)

#Système de Tour du joueur

def lancerJ1():
    global bouton_joueur1,bouton_joueur2
    hasardJ1=randint(1,3)
    if hasardJ1==1:

```

```

        additionJ1()
    elif hasardJ1==2:
        soustractionJ1()
    else:
        multiplicationJ1()

bouton_joueur2 = Button(lancer_des, text = ("Lancez", joueur2), command = lancerJ2)
bouton_joueur2.pack()
bouton_joueur1.destroy()

def lancerJ2():
    global bouton_joueur1,bouton_joueur2
    hasardJ2=randint(1,3)
    if hasardJ2==1:
        additionJ2()
    elif hasardJ2==2:
        soustractionJ2()
    else:
        multiplicationJ2()

bouton_joueur1 = Button(lancer_des, text = ("Lancez", joueur1) , command = lancerJ1)
bouton_joueur1.pack()
bouton_joueur2.destroy()

if m==joueur1:
    bouton_joueur1 = Button(lancer_des, text = ("Lancez", joueur1), command = lancerJ1)
    bouton_joueur1.pack()
else:
    bouton_joueur2 = Button(lancer_des, text = ("Lancez", joueur2), command = lancerJ2)
    bouton_joueur2.pack()

```

#Lancer dés

```

N1=1
N2=1
def dés1 ():
    global joueur1
    global N1,X1,Y1,N2,X2,Y2
    affichage_des2.create_rectangle(0,0,210,160, fill = "pink")
    de = randint(1,6)+randint(1,6)
    score=Canvas.create_text(affichage_des2, 100,100, font = 'Arial 24 italic' ,text = de)
    N1=N1+de

X1=25
Y1=25

if N1==6:
    N1=12
    X1=825
    Y1=325
    fenetre_chance=Tk()
    chance= Canvas(fenetre_chance,height=100,width=400,bg="white")
    chance.create_text(200,50,font='Arial 12 italic', text='Vous avancez de 6 cases supplémentaires')
    bouton_chance = Button(fenetre_chance, text = ("SUIVANT"), command = fenetre_chance.destroy)
    bouton_chance.pack(side = BOTTOM)
    chance.pack()
elif N1<10:
    X1=(N1-1)*100+25
    Y1=25
elif N1<16:
    X1=825
    Y1=(N1-9)*100+25
elif N1<24:
    X1=825-100*(N1-15)
    Y1=625
elif N1<29:
    X1=25
    Y1=625-100*(N1-23)
elif N1<36:
    X1=100*(N1-28)+25

```

```

Y1=125
elif N1<40:
    X1=725
    Y1=100*(N1-35)+125
elif N1<46:
    X1=725-100*(N1-39)
    Y1=525
elif N1<49:
    X1=125
    Y1=525-100*(N1-45)
elif N1<54:
    X1=125+100*(N1-48)
    Y1=225
elif N1<56:
    X1=625
    Y1=225+100*(N1-53)
elif N1==58:
    N1=1
    X1=25
    Y1=25
    fenetre_malchance=Tk()
    malchance= Canvas(fenetre_malchance, height=100,width=400,bg="white")
    malchance.create_text(200,50,font='Arial 12 italic', text='Vous êtes tombés dans le puits.\nVous retournez au point de départ.')
    bouton_malchance = Button(fenetre_malchance, text = ("SUIVANT"), command = fenetre_malchance.destroy)
    bouton_malchance.pack(side = BOTTOM)
    malchance.pack()

elif N1<59:
    X1=625-100*(N1-55)
    Y1=425
elif N1==60:
    X1=225
    Y1=325
elif N1<61:
    X1=225
    Y1=425-100*(N1-58)
elif N1<63:
    X1=225+100*(N1-60)
    Y1=325
else:
    X1=425+100
    Y1=325
    fenetre_victoire=Tk()
    victoire= Canvas(fenetre_victoire, height=100,width=300,bg="white")
    victoire.create_text(150,50,font='Arial 12 italic', text= "Vous avez gagné")
    victoire.create_text(150,25,font='Arial 12 italic', text= joueur1)
plateau.coords(joueur1_noir,X1,Y1,X1+25,Y1+25)
victoire.pack()
def quitter_jeu():
    fenetre_victoire.destroy()
    fenetre_jeu.destroy()
bouton_victoire = Button(fenetre_victoire, text = ("SUIVANT"), command = quitter_jeu)
bouton_victoire.pack(side = BOTTOM)

def dés2():
    global joueur2
    global N1,X1,Y1,N2,X2,Y2
    affichage_des2.create_rectangle(0,0,210,160, fill = "pink")
    de = randint(1,6)+randint(1,6)
    score=Canvas.create_text(affichage_des2, 100,100, font = 'Arial 24 italic' ,text = de)
    N2=N2+de
    X2=50
    Y2=50

    if N2==6:
        N2=12
        X2=850
        Y2=350
        fenetre_chance=Tk()

```

```

chance= Canvas(fenetre_chance,height=100,width=400,bg="white")
chance.create_text(200,50,font='Arial 12 italic', text='Vous avancez de 6 cases supplémentaires')
bouton_chance = Button(fenetre_chance, text = ("SUIVANT"), command = fenetre_chance.destroy)
bouton_chance.pack(side = BOTTOM)
chance.pack()
elif N2<10:
    X2=(N2-1)*100+50
    Y2=50
elif N2<16:
    X2=850
    Y2=(N2-9)*100+50
elif N2<24:
    X2=850-100*(N2-15)
    Y2=650
elif N2<29:
    X2=50
    Y2=650-100*(N2-23)
elif N2<36:
    X2=100*(N2-28)+50
    Y2=150
elif N2<40:
    X2=750
    Y2=100*(N2-35)+150
elif N2<46:
    X2=750-100*(N2-39)
    Y2=550
elif N2<49:
    X2=150
    Y2=550-100*(N2-45)
elif N2<54:
    X2=150+100*(N2-48)
    Y2=250
elif N2<56:
    X2=650
    Y2=250+100*(N2-53)
elif N2==58:
    N2=1
    X2=50
    Y2=50
    fenetre_malchance=Tk()
    malchance= Canvas(fenetre_malchance, height=100,width=400,bg="white")
    malchance.create_text(200,50,font='Arial 12 italic', text='Vous êtes tombés dans le puits.\nVous retournez au point de départ.')
    bouton_malchance = Button(fenetre_malchance, text = ("SUIVANT"), command = fenetre_malchance.destroy)
    bouton_malchance.pack(side = BOTTOM)
    malchance.pack()
elif N2<59:
    X2=650-100*(N2-55)
    Y2=450
elif N2==60:
    X2=250
    Y2=350
elif N2<61:
    X2=250
    Y2=450-100*(N2-58)
elif N2<63:
    X2=250+100*(N2-60)
    Y2=350
else:
    X2=450+100
    Y2=350
    fenetre_victoire=Tk()
    victoire= Canvas(fenetre_victoire, height=200,width=300,bg="white")
    victoire.create_text(150,100,font='Arial 12 italic', text= "Vous avez gagné")
    victoire.create_text(150,50,font='Arial 12 italic', text= joueur2)

    victoire.pack()
def quitter_jeu2():
    fenetre_victoire.destroy()
    fenetre_jeu.destroy()
bouton_victoire = Button(fenetre_victoire, text = ("SUIVANT"), command = quitter_jeu2)

```

```

bouton_victoire.pack(side = BOTTOM)

plateau.coords(joueur2_blanç,X2,Y2,X2+25,Y2+25)

#Affichage dés
affichage_des = Frame(des,width = 200,height=175 ,borderwidth=0, bg='yellow')
affichage_des.pack(side=BOTTOM)

affichage_des2 = Canvas(affichage_des, width = 200, heigh =150, borderwidth=0, bg = 'pink')
affichage_des2.pack()

#fenetre graphique pour calculs

calcul = Frame(actions,width = 200,height=300,borderwidth=0, bg = "green")
calcul.pack(side=BOTTOM)

calcul_entree = Frame(calcul, width = 200, heigh = 150, borderwidth=2, bg= 'yellow')
calcul_entree.pack(side = TOP)

calcul_validation = Canvas(calcul,width = 200, heigh = 75, borderwidth=2, bg ='green' )
calcul_validation.pack(side=BOTTOM)

affichage_calcul = Canvas(actions, width = 200,height = 175, borderwidth = 0, bg = "purple")
affichage_calcul.pack(side = BOTTOM)

affichage_confirmation=Canvas(actions, width =200, height=175,borderwidth=0,bg = "yellow")
affichage_confirmation.pack(side=BOTTOM)

#spinbox pour répondre
entree_valeur=Spinbox(calcul_entree,from_=-9999,to=9999)
entree_valeur.pack()

#gestion des calculs
##additionJ1
def additionJ1():
    global i,c,b,f,entree_valeur
    global bouton_valider_addition
    c = randint(1,100)
    b = randint(1,100)
    f=c+b

    def validerJ1_addition():
        global c,b,f,entree_valeur
        global bouton_valider_addition
        i=int(entree_valeur.get())
        if f==i:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
            bouton_valider_addition.destroy()
            dés1()
        else:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
            affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
            affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
            bouton_valider_addition.destroy()

    affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
    affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "+", b))


```

```

bouton_valider_addition=Button(calcul_validation, text = "VALIDER", command = validerJ1_addition)
bouton_valider_addition.pack(side=BOTTOM)

##soustractionJ1
def soustractionJ1():
    global i,c,b,f,entree_valeur
    global bouton_valider_soustraction
    c = randint(1,100)
    b = randint(1,100)
    f=c-b

    def validerJ1_soustraction():
        global c,b,f,entree_valeur
        global bouton_valider_soustraction
        i=int(entree_valeur.get())
        if f==i:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
            bouton_valider_soustraction.destroy()
            dés1()

        else:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
            affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
            affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
            bouton_valider_soustraction.destroy()

    affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
    affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "-", b))
    bouton_valider_soustraction=Button(calcul_validation, text = "VALIDER", command = validerJ1_soustraction)
    bouton_valider_soustraction.pack(side=BOTTOM)

##multiplicationJ1
def multiplicationJ1():
    global i,c,b,f,entree_valeur
    global bouton_valider_multiplication
    c = randint(1,20)
    b = randint(1,20)
    f=c*b

    def validerJ1_multiplication():
        global c,b,f,entree_valeur
        global bouton_valider_multiplication
        i=int(entree_valeur.get())
        if f==i:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
            bouton_valider_multiplication.destroy()
            dés1()

        else:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
            affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
            affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
            bouton_valider_multiplication.destroy()

    affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
    affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "*", b))
    bouton_valider_multiplication=Button(calcul_validation, text = "VALIDER", command = validerJ1_multiplication)
    bouton_valider_multiplication.pack(side=BOTTOM)

```

```

##additionJ2
def additionJ2():
    global i,c,b,f,entree_valeur
    global bouton_valider_addition2
    c = randint(1,100)
    b = randint(1,100)
    f=c+b

    def validerJ2_addition():
        global c,b,f,entree_valeur
        global bouton_valider_addition2
        i=int(entree_valeur.get())
        if f==i:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
            bouton_valider_addition2.destroy()
            dés2()

        else:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
            affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
            affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
            bouton_valider_addition2.destroy()

        affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
        affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "+", b))
        bouton_valider_addition2=Button(calcul_validation, text = "VALIDER", command = validerJ2_addition)
        bouton_valider_addition2.pack(side=BOTTOM)

##soustractionJ2
def soustractionJ2():
    global i,c,b,f,entree_valeur
    global bouton_valider_soustraction2
    c = randint(1,100)
    b = randint(1,100)
    f=c-b

    def validerJ2_soustraction():
        global c,b,f,entree_valeur
        global bouton_valider_soustraction2
        i=int(entree_valeur.get())
        if f==i:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
            bouton_valider_soustraction2.destroy()
            dés2()

        else:
            affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
            affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
            affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
            affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
            affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
            bouton_valider_soustraction2.destroy()

        affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
        affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "-", b))
        bouton_valider_soustraction2=Button(calcul_validation, text = "VALIDER", command = validerJ2_soustraction)
        bouton_valider_soustraction2.pack(side=BOTTOM)

##multiplicationJ2
def multiplicationJ2():
    global i,c,b,f,entree_valeur
    global bouton_valider_multiplication2
    c = randint(1,20)
    b = randint(1,20)
    f=c*b

```

```

def validerJ2_multiplication():
    global c,b,f,entree_valeur
    global bouton_valider_multiplication2
    i=int(entree_valeur.get())
    if f==i:
        affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
        affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Vous avez trouvé\n le bon résultat\n vous pouvez avancer")
        bouton_valider_multiplication2.destroy()
        dés2()

    else:
        affichage_confirmation.create_rectangle(0,0,210,300, fill = "yellow")
        affichage_confirmation.create_text(90,75,font='Arial 12 italic',text="Vous avez faux")
        affichage_confirmation.create_text(90,100,font='Arial 12 italic',text="Le résultat était")
        affichage_confirmation.create_text(90,125,font='Arial 12 italic',text=f)
        affichage_confirmation.create_text(90,150,font='Arial 12 italic',text="Vous restez sur place")
        bouton_valider_multiplication2.destroy()

affichage_calcul.create_rectangle(0,0,210,210, fill = "purple")
affichage_calcul.create_text(90,100,font='Arial 24 italic',text=( c, "*", b))
bouton_valider_multiplication2=Button(calcul_validation, text = "VALIDER", command = validerJ2_multiplication)
bouton_valider_multiplication2.pack(side=BOTTOM)

fenetre_jeu.mainloop()

```

Page HTML :

<!DOCTYPE HTML >

```

<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Jeu de l'oie mathématique</title>
    <link rel="stylesheet" href="Projet_jeu_de_l'oie.css">
</head>
<body>
<h1 id="Jeu_de_l'oie_origine">Jeu de l'oie à l'origine</h1>

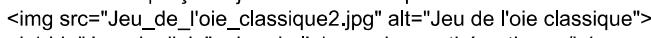
<a href="#Jeu_de_l'oie_origine">Le Jeu de l'oie à l'origine</a><br>
<a href="#Jeu_de_l'oie">Le Jeu de l'oie version mathématique</a><br>
<a href="#Image_jeu_de_l'oie">Image du Jeu de l'oie mathématique</a><br>

<p>Le jeu de l'oie classique consiste à faire des lancés de dés pour arriver jusqu'à la case 63.<br>
    Chaque joueur joue chacun son tour en lançant les 2 dés. Suivant le nombre ou chiffre obtenue, le joueur avance son pion case par case.<br>

    Il existe des règles du jeu de l'Oie à respecter selon le nombre que l'on fait ou de la case sur laquelle on tombe.<br><br>
    Voyons plus en détails ces règles :<br><br>

    Au commencement de la partie, si l'un des joueurs fait 9 avec un dé de 6 et un dé de 3, il doit avancer son pion immédiatement à la case 26.<br>
    S'il fait 9 avec un dé de 4 et un dé de 5, il ira à la case 53.<br><br>
    Si lors de la partie, le joueur tombe sur une oie, il avance de nouveau du nombre de points réalisés.<br><br>
    Si un joueur fait 6, il doit se rendre sur la case 12.<br><br>
    Le joueur qui tombe sur la case 19 correspondant à un hôtel devra passer son tour durant 2 tours.<br><br>
    Le joueur qui tombe sur la case 31 correspondant au puits attendra qu'un autre joueur arrive au même numéro et prendra sa place.<br><br>
    Celui qui tombe sur la case 42 correspondant au labyrinthe retournera obligatoirement à la case 30.<br><br>
    Celui qui ira en 52 correspondant la prison attendra qu'un autre joueur vienne au même numéro pour repartir.<br><br>
    Le joueur qui va sur la case 58 correspondant à la case Tête de mort recommencera la partie depuis le début.<br><br>
    Celui qui est rejoint par un autre joueur sur la même case devra se rendre sur la case où l'autre joueur se situait avant de jouer.<br><br>
    Pour gagner une partie de jeu de l'Oie, il faut être le premier à arriver sur la dernière case 63
    mais avec l'obligation d'arriver pile sur cette case.<br><br>

```

Au cas où le joueur fait un score de dés supérieur au nombre de case le séparant de la victoire alors il devra reculer d'autant de cases supplémentaires. </p>
<h3>Voici un aperçu du jeu de l'oie classique</h3>

<h1 id="Jeu_de_l'oie"> Jeu de l'oie version mathématique</h1>

<p>Ensuite on peut parler de notre jeu de l'oie numérique que l'on a créé tous ensemble et qui est moins développé que le jeu de l'oie classique.

Il reprend des bases des mathématiques.

Il a été conçu pour apprendre les bases des mathématiques pour le niveau primaire de manière ludique.

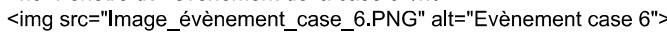
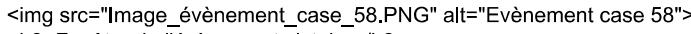
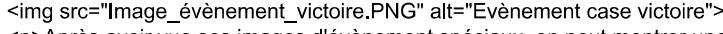
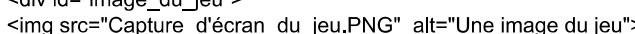
Les règles sont assez simples le but est de réussir des calculs pour avancer. Si l'on ne parvient pas à réussir le calcul qui nous est donné, notre pion ne bouge pas.

Le but est d'arriver jusqu'à la case 63 pour gagner.

Deux événements supplémentaires en plus des calculs sont ajoutés. Un à la case 6 et un autre à la case 58.

Celui à la case 6 permet d'avancer de six cases supplémentaires tandis que si l'on tombe sur la case 58, on tombe dans le puit et on repart à la case de départ.

Pour finir quand on franchit la case 63 ou plus nous avons une fenêtre de victoire qui s'affiche avec le nom du joueur qui a gagné et on peut quitter la partie.

</p>
<h3>Fenêtre de l'événement de la case 6</h3>

<h3>Fenêtre de l'événement de la case 58</h3>

<h3>Fenêtre de l'événement victoire</h3>

<p>Après avoir vu ces images d'événement spéciaux, on peut montrer une image du jeu</p>
<h3 id="Image_jeu_de_l'oie">Image du jeu de l'oie version mathématique</h3>
<div id="image_jeu">

</div>

</body>
</html>

CSS :

```
BODY {
background-color:rgb(64,190,213) /* C'est du turquoise */
}
p
{
color:black;
font-size:30px;
text-align:left
}
h1
{
color:red;
text-align:center;
font-size:50px;
text-decoration:underline;
}
img
{
width: 20%;
height: 20%;
position:relative;
left:40%
}
h3
{
text-align:center;
color:red;
font-size:40px;
text-decoration:underline;
}
#image_jeu
```

```
{  
width:400%;  
height:400%;  
position:relative;  
right:150%  
}  
a  
{  
font-size:30px;  
}
```