

```

import pygame
from pygame.locals import *
import sys
from random import *

pygame.init() #Initialisation de Pygame.
fpsClock = pygame.time.Clock()
#Initialisation des variables.
IPS = 30 #Nombre d'images par seconde.
niveau = 0
CLIC = 0 #Représente le nombre de points que l'on gagne à chaque clic EN PLUS de nos clics normaux.
PIECES = 0 #Représente la valeur de nos points.
PPS = 0.0 #Représente le nombre de points par seconde que l'on gagne.
NOIR = pygame.Color(0, 0, 0) #Permet d'associer la couleur noire à une variable.
BLANC = pygame.Color(250, 250, 250) #Permet d'associer la couleur blanche à une variable.
POLICE = pygame.font.SysFont("Helvetica", 36) #Permet d'associer la police utilisée à une variable.
POLICE2 = pygame.font.SysFont("Helvetica", 17) #Police plus petite pour l'interieur des boutons.
POLICE3 = pygame.font.SysFont("Helvetica", 13) #Police encore plus petite pour les aides.
POLICE4 = pygame.font.SysFont("Helvetica", 15) #Police adaptée pour certains textes.
quitter = False #Variable permettant d'entrer ou de sortir de la boucle while de Pygame.
pause = 0 #Si pause = 0 alors le jeu est en cours, si elle est égale à 1 : le jeu est en pause.
MENU = 1 #Affiche le menu déroulant ou pas selon sa valeur.
AIDE = 0 #Affiche les aides selon sa valeur.
choix = 0 #Choix du mode de jeu.
choix_sur = 0 #Confirmation du mode de jeu.
jeu = 0 #Mode de jeu
compteur = 1 #Compteur de temps dans le jeu.
exploit = 0 #Accession à la liste des objectifs selon sa valeur (1 : accès)
nb_exploits = 0 #Nombre d'exploits réalisés.
fin = 0 #Si le joueur atteint la fin, fin = 1.
moins = 0 #Valeur totale de points perdus avec le malus.
a = 0 #Affichage du bonus ou non.
b = 0 #Variable intermédiaire à l'affichage du bonus.
d = 0 #Temps où le bonus est affiché.
X = 0 #Position en X de l'affichage du bonus qui est aléatoire.
Y = 0 #Position en Y de l'affichage du bonus qui est aléatoire également.
nb_bonus = 0 #Nombre de bonus attrapés
if quitter == True : #Dans le cas où une erreur apparaît, ferme le jeu.
    sys.exit(0)
def Nom(): #Fonction permettant de choisir son nom.
    global nom
    nom = str(input("Comment voulez-vous vous faire appeler ? (Pas plus de 10 caractères) : "))
    while len(nom)>10: #Si le nom possède plus de 10 caractères, il faut choisir un autre nom.
        nom = str(input("Le nom que vous avez choisi possède plus de 10 caractères. Veuillez recommencer : "))
    return nom
#Chargement d'images nécessaires au lancement du jeu.
vitesse_fond0 = pygame.image.load("Ressources/vitesse_fond0.png")
vitesse_fond1 = pygame.image.load("Ressources/vitesse_fond1.png")
succes_fond0 = pygame.image.load("Ressources/succes_fond0.png")
succes_fond1 = pygame.image.load("Ressources/succes_fond1.png")
catalan = pygame.image.load("Ressources/Succès/catala.png")
catalan_rect = Rect(128 + (8 - 6) * 156, 400, catalan.get_width(), catalan.get_height())

```

```

vitesse_fond_rect = Rect(0, 0, vitesse_fond0.get_width(), vitesse_fond0.get_height())
succes_fond_rect = Rect(320, 0, succes_fond0.get_width(), succes_fond0.get_height())
icon = pygame.image.load("Ressources/Vitesse/coin.png") #Icône du jeu.
menu_pause = pygame.image.load("Ressources/menu_pause.png")
bouton_croix = POLICE.render("X", True, BLANC)
bouton_egal = POLICE.render("=", True, BLANC)
bouton_egal_rect = Rect(0, 437, bouton_egal.get_width(), bouton_egal.get_height())
bouton_croix_rect = Rect(146, 437, bouton_croix.get_width(), bouton_croix.get_height())
#Création des listes :
objet = ["Mine de pièces", "Usine de pièces", "Excavateur"] #Liste des noms des objets
que l'on peut acheter.
prix = [1, 5, 10, 50] #Liste des prix des objets que l'on peut acheter.
Q = [0, 0, 0] #Quantité d'objets achetés.
texte_objet = [0, 0, 0] #Liste des noms des objets.
clac_par_s = [0.0] # PPS lorsque mis en pause.
compteur_s = [0] # Compteur lorsque mis en pause.
e = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #Liste de si l'exploit est réalisé ou non.
v = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #Liste de si le son est réalisé lors de la réalisation
de l'exploit.
E = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #Liste permettant d'afficher l'image de l'exploit si
celui-ci a été réalisé.
t = [POLICE3.render("Cumuler 100 pièces.", True, BLANC), POLICE3.render("Cumuler 500
pièces.", True, BLANC), POLICE3.render("Cumuler 1000 pièces.", True, BLANC),
POLICE3.render("Posséder un PPS de 5.", True, BLANC), POLICE3.render("Posséder un
PPS de 10.", True, BLANC), POLICE3.render("Posséder un PPS de 15.", True, BLANC),
POLICE3.render("Attraper 1 bonus.", True, BLANC), POLICE3.render("Attraper 3
bonus.", True, BLANC), POLICE3.render("Perdre 100 pièces.", True, BLANC)] # Liste des
textes correspondants aux exploits.
Nom() #Choix du nom
#Création et modification du titre ainsi que de l'icône de la fenêtre.
fenetre = pygame.display.set_mode((640, 480))
pygame.display.set_caption("Clickos")
pygame.display.set_icon(icon)
#Création des procédures.
def points(): #Permet de calculer le nombre de points lorsqu'on clique.
    global PIECES
    PIECES += 1 + CLIC #On ajoute au nombre de pièces actuel, la valeur du clic.
def malus(): #Fait perdre un nombre de points aléatoire en cas d'erreur.
    global PIECES, moins
    clicdroit = randint(0,10) #On affecte une valeur aléatoire entre 0 et 10 au clic
droit.
    moins += clicdroit #On ajoute à la variable moins la valeur du clic droit
    if PIECES-clicdroit >= 0:
        PIECES -= clicdroit #On soustrait au nombre de pièces, la valeur du clic
droit.
    else :
        PIECES = 0 #Si le clic droit à une valeur plus grande que le nombre de pièces,
le nombre de pièces sera égal à 0.
def objet1(): #Définit l'achat de l'objet 1.
    global PPS, PIECES
    pps = PPS
    if PIECES >= prix[0]:
        Q[0] += 1
        pps += 0.1
        PPS = round(pps,1) #Arrondi pps à une décimale.
        PIECES -= prix[0]
        prix[0] *= 1.5
        p = prix[0]
        prix[0] = round(p) #Arrondi p en entier.
def objet2(): #Définit l'achat de l'objet 2.
    global PPS, PIECES

```

```

pps = PPS
if PIECES >= prix[1]:
    Q[1] += 1
    pps += 1
    PPS = round(pps,1)
    PIECES -= prix[1]
    prix[1] *= 1.5
    p = prix[1]
    prix[1] = round(p)
def objet3(): #Définit l'achat de l'objet 3.
    global CLIC, PIECES
    if PIECES >= prix[2]:
        Q[2] += 1
        CLIC += 1
        PIECES -= prix[2]
        prix[2] *= 1.5
        p = prix[2]
        prix[2] = round(p)
        if Q[2] % 2 == 0 and Q[2] != 0:
            f = randint(1, 4)
            if f == 1:
                pygame.mouse.set_cursor(*pygame.cursors.broken_x)
            elif f == 2:
                pygame.mouse.set_cursor(*pygame.cursors.diamond)
            elif f == 3:
                pygame.mouse.set_cursor(*pygame.cursors.tri_left)
            elif f == 4:
                pygame.mouse.set_cursor(*pygame.cursors.ball)
def objet4(): #Définit l'achat de l'objet 4.
    global PIECES, niveau
    if PIECES >= prix[3]:
        niveau += 1
        PIECES -= prix[3]
        prix[3] *= 1.5
        p = prix[3]
        prix[3] = round(p)
def Bonus(): #Apparition des bonus aléatoire (Position+temps)
    global a, b, X, Y, d
    c = int(compteur)
    if c % 20 == 0:
        a = 1
        if b == 0:
            X = randint(0, 398) # Génération de la position aléatoire du bonus sur
l'axe des X
            Y = randint(0, 438) # Génération de la position aléatoire du bonus sur
l'axe des Y
            d = int(compteur)
            b = 1
        if c == d+5:
            a = 0
            b = 0
def exploits():
    global exploit
    if exploit == 0:
        exploit = 1
    elif exploit == 1:
        exploit = 0
def verif_exploits():
    global PIECES, PPS, moins, nb_bonus, nb_exploits
    son = pygame.mixer.Sound("Ressources/son.wav")

```

```

if PIECES >= 100:
    e[0] = 1
    if v[0] == 0:
        nb_exploits += 1
        son.play()
        v[0] = 1
if PIECES >= 500:
    e[1] = 1
    if v[1] == 0:
        nb_exploits += 1
        son.play()
        v[1] = 1
if PIECES >= 1000:
    e[2] = 1
    if v[2] == 0:
        nb_exploits += 1
        son.play()
        v[2] = 1
if PPS >= 5:
    e[3] = 1
    if v[3] == 0:
        nb_exploits += 1
        son.play()
        v[3] = 1
if PPS >= 10:
    e[4] = 1
    if v[4] == 0:
        nb_exploits += 1
        son.play()
        v[4] = 1
if PPS >= 15:
    e[5] = 1
    if v[5] == 0:
        nb_exploits += 1
        son.play()
        v[5] = 1
if nb_bonus >= 1:
    e[6] = 1
    if v[6] == 0:
        nb_exploits += 1
        son.play()
        v[6] = 1
if nb_bonus >= 3:
    e[7] = 1
    if v[7] == 0:
        nb_exploits += 1
        son.play()
        v[7] = 1
if moins >= 100:
    e[8] = 1
    if v[8] == 0:
        nb_exploits += 1
        son.play()
        v[8] = 1
def Fin():
    global nom, compteur, fin, choix
    if jeu == 1:
        mode = "Vitesse"
    else :

```

```

        mode = "Succès"
        bravo = POLICE.render("Bravo " + nom, True, BLANC)
        MODE = POLICE.render("Vous avez fini Clickos en mode " + mode, True, BLANC)
        temps = POLICE.render("En " + str(int(compteur)) + " secondes", True, BLANC)
        if fin == 1:
            fenetre.fill(NOIR)
            fenetre.blit(bravo, (0, 0))
            fenetre.blit(MODE, (0, 200))
            fenetre.blit(temps, (0, 400))
def vitesse():
    global niveau, PIECES, PPS, pause, MENU, AIDE, event, compteur, fin, nom
    fenetre.fill(NOIR)
    bouton_niveau0 = pygame.image.load("Ressources/Vitesse/bouton_niveau0.png")
    bouton = pygame.image.load("Ressources/Vitesse/fond_bouton.png")
    bouton_rect = [Rect(220, 220, bouton_niveau0.get_width(),
bouton_niveau0.get_height()),
                    Rect(440, 50, bouton.get_width(), bouton.get_height()),
                    Rect(440, 150, bouton.get_width(), bouton.get_height()),
                    Rect(440, 250, bouton.get_width(), bouton.get_height()),
                    Rect(440, 350, bouton.get_width(), bouton.get_height())]
    #Affichage des images préalablement chargées à des positions définies
    if niveau == 0:
        fenetre.blit(bouton_niveau0, bouton_rect[0])
        texte0 = POLICE2.render("Niveau 1 : 0$", True, NOIR)
        fenetre.blit(texte0, (265, 235))
    elif niveau >= 1:
        fonds = [pygame.image.load("Ressources/Vitesse/fond_clicker.png"),
                  pygame.image.load("Ressources/Vitesse/fond_clicker2.png"),
                  pygame.image.load("Ressources/Vitesse/fond_clicker3.png"),
                  pygame.image.load("Ressources/Vitesse/fond_clicker4.png"),
                  pygame.image.load("Ressources/Vitesse/fond_clicker5.png")]
        # Chargement des images que nous utiliserons selon le niveau.
        for i in range(0, 6):
            if niveau == i:
                fond = fonds[i-1]
        # Chargement des autres images ne variant pas.
        piece = pygame.image.load("Ressources/Vitesse/coin.png")
        menu_deroulant = pygame.image.load("Ressources/Vitesse/menu_deroulant.png")
        aide = pygame.image.load("Ressources/fond_bouton_aide.png")
        # Création des rectangles qui représenteront les zones où nous pourrions
cliquer.
        piece_rect = Rect(125, 150, piece.get_width(), piece.get_height())
        # Création des textes qui seront affichés.
        texte = POLICE.render(str(int(PIECES)) + " + " + str(PPS) + " PPS", True,
BLANC)
        texte_objet[0] = POLICE4.render(str(int(Q[0])) + "x " + objet[0] + " " +
str(prix[0]) + "$", True, NOIR)
        texte_objet[1] = POLICE4.render(str(int(Q[1])) + "x " + objet[1] + " " +
str(prix[1]) + "$", True, NOIR)
        texte_objet[2] = POLICE2.render(str(int(Q[2])) + "x " + objet[2] + " " +
str(prix[2]) + "$", True, NOIR)
        compteur_texte = POLICE2.render(str(int(compteur)), True, BLANC)
        if niveau != 5:
            texte_objet4 = POLICE2.render("Niveau " + str(int(niveau + 1)) + " " +
str(prix[3]) + "$", True, NOIR)
        else:
            texte_objet4 = POLICE2.render("Fin " + str(prix[3]) + " $", True, NOIR)
        texte_nom = POLICE.render(nom + " : niveau " + str(int(niveau)), True, BLANC)
        if niveau >= 1 and fin == 0: #Affichage de toutes les images et textes.
            fenetre.blit(fond, (0, 0))

```

```

fenetre.blit(piece, piece_rect)
for i in range(1, 5):
    fenetre.blit(bouton, bouton_rect[i])
fenetre.blit(texte, (125, 100))
for i in range(0, 3):
    fenetre.blit(texte_objet[i], (450, 65 + 100 * i))
fenetre.blit(texte_objet4, (450, 365))
fenetre.blit(texte_nom, (0, 0))
fenetre.blit(compteur_texte, (450, 450))
elif fin == 1:
    Fin()
if niveau >= 1 and fin == 0:
    if MENU == 1:
        fenetre.blit(bouton_egal, bouton_egal_rect)
    elif MENU == 2:
        fenetre.blit(menu_deroulant, (0,0))
        fenetre.blit(bouton_croix, bouton_croix_rect)
if niveau >= 1 and exploit == 0: #Affichage des aides.
    if AIDE == 1:
        texte_aide = POLICE3.render("+0.1 PPS", True, BLANC)
        fenetre.blit(aide, (340, 50))
        fenetre.blit(texte_aide, (350, 65))
    elif AIDE == 2:
        texte_aide = POLICE3.render("+1 PPS", True, BLANC)
        fenetre.blit(aide, (340, 150))
        fenetre.blit(texte_aide, (350, 165))
    elif niveau >= 1 and AIDE == 3:
        texte_aide = POLICE3.render("+1 CLIC", True, BLANC)
        fenetre.blit(aide, (340, 250))
        fenetre.blit(texte_aide, (350, 265))
    elif AIDE == 4:
        texte_aide = POLICE3.render("+1 niveau", True, BLANC)
        fenetre.blit(aide, (340, 350))
        fenetre.blit(texte_aide, (350, 365))
for event in pygame.event.get(): #Permet de récupérer des événements, des actions
effectuées.
    if event.type == pygame.KEYDOWN and event.key == K_p: # Touche P enclenchée
(pause).
        if pause == 0 :
            clic_par_s[0] = PPS
            PPS = 0.0
            compteur_s[0] = compteur
            compteur = 0
            pause = 1
        elif pause == 1 :
            PPS = round(clic_par_s[0], 1)
            compteur = round(compteur_s[0])
            pause = 0
    if pause == 0:
        if niveau >= 1:
            if event.type == pygame.MOUSEBUTTONUP: #Nous récupérons l'événement du
clic de la souris qui est relâché.
                if event.button == 1: #Le chiffre 1 signifie qu'il s'agit du clic
gauche.
                    if piece_rect.collidepoint(event.pos):
                        points()
                    elif bouton_rect[1].collidepoint(event.pos):
                        objet1()
                    elif bouton_rect[2].collidepoint(event.pos):

```

```

        objet2()
    elif bouton_rect[3].collidepoint(event.pos):
        objet3()
    elif bouton_rect[4].collidepoint(event.pos):
        if niveau != 5:
            objet4()
        else :
            fin = 1
    elif bouton_egal_rect.collidepoint(event.pos):
        MENU = 2
    elif bouton_croix_rect.collidepoint(event.pos):
        MENU = 1
if event.button == 3:
    if piece_rect.collidepoint(event.pos):
        malus()
    if bouton_rect[1].collidepoint(event.pos):
        if AIDE != 1:
            AIDE = 1
        elif AIDE == 1:
            AIDE = 0
    if bouton_rect[2].collidepoint(event.pos):
        if AIDE != 2:
            AIDE = 2
        elif AIDE == 2:
            AIDE = 0
    if bouton_rect[3].collidepoint(event.pos):
        if AIDE != 3:
            AIDE = 3
        elif AIDE == 3:
            AIDE = 0
    if bouton_rect[4].collidepoint(event.pos):
        if AIDE != 4:
            AIDE = 4
        elif AIDE == 4:
            AIDE = 0

if niveau == 0:
    if event.type == pygame.MOUSEBUTTONDOWN:
        if event.button == 1:
            if bouton_rect[0].collidepoint(event.pos):
                niveau += 1
if pause == 1: #Affichage du menu pause.
    fenetre.blit(menu_pause, (0, 0))
def succes():
    global quitter, niveau, PIECES, PPS, pause, MENU, AIDE, event, exploite, fin, a, X,
Y, nb_bonus, nom, compteur, langue
    fenetre.fill(NOIR)
    verif_exploits()
    Bonus()
    # Chargement d'images nécessitant d'être chargées avant.
    bouton = pygame.image.load("Ressources/Succès/fond_bouton.png")
    bouton_niveau0 = pygame.image.load("Ressources/Succès/bouton_niveau0.png")
    bouton_rect = [Rect(220, 220, bouton_niveau0.get_width(),
bouton_niveau0.get_height()),
                  Rect(440, 50, bouton.get_width(), bouton.get_height()),
                  Rect(440, 150, bouton.get_width(), bouton.get_height()),
                  Rect(440, 250, bouton.get_width(), bouton.get_height()),
                  Rect(440, 350, bouton.get_width(), bouton.get_height())]

    # Affichage des images préalablement chargées à des positions définies
    if niveau == 0:

```



```

fenetre.blit(texte, (125, 100))
for i in range(0, 3):
    fenetre.blit(texte_objet[i], (450, 65 + 100 * i))
fenetre.blit(texte_objet4, (450, 365))
fenetre.blit(texte_nom, (0, 0))
fenetre.blit(compteur_texte, (450, 450))
if a == 1:
    bonus = pygame.image.load("Ressources/Succès/coffre.png")
    bonus_rect = Rect(X, Y, bonus.get_width(), bonus.get_height())
    fenetre.blit(bonus, bonus_rect)
#Affichage des textes des exploits.
elif exploit == 1:
    fenetre.blit(fond, (0, 0))
    for i in range(0, 9):
        if i < 3:
            fenetre.blit(E[i], (156 + i * 156, 110))
        elif i >= 3 and i < 6:
            fenetre.blit(E[i], (156 + (i-3) * 156, 230))
        elif i >= 6:
            fenetre.blit(E[i], (156 + (i-6) * 156, 350))
    for i in range(0, 9):
        if i < 3:
            fenetre.blit(t[i], (123 + i * 156, 160))
        elif i >= 3 and i < 6:
            fenetre.blit(t[i], (113 + (i-3) * 156, 280))
        elif i >= 6:
            fenetre.blit(t[i], (128 + (i-6) * 156, 400))
    fenetre.blit(retour, retour_rect)
# Affichage ou masquage du menu déroulant.
if niveau >= 1 and MENU == 1 and exploit == 0:
    fenetre.blit(bouton_egal, bouton_egal_rect)
elif niveau >= 1 and MENU == 2 and exploit == 0:
    fenetre.blit(menu_deroulant, (0, 0))
    fenetre.blit(bouton_croix, bouton_croix_rect)
# Affichage ou masquage des aides.
if niveau >= 1 and exploit == 0:
    if AIDE == 1:
        texte_aide = POLICE3.render("+0.1 PPS", True, BLANC)
        fenetre.blit(aide, (340, 50))
        fenetre.blit(texte_aide, (350, 65))
    elif AIDE == 2:
        texte_aide = POLICE3.render("+1 PPS", True, BLANC)
        fenetre.blit(aide, (340, 150))
        fenetre.blit(texte_aide, (350, 165))
    elif niveau >= 1 and AIDE == 3:
        texte_aide = POLICE3.render("+1 CLIC", True, BLANC)
        fenetre.blit(aide, (340, 250))
        fenetre.blit(texte_aide, (350, 265))
#Récupération des touches et effectuation des atcions en lien avec la touche
pressée;
for event in pygame.event.get(): # Permet de récupérer des événements, des actions
effectuées.
    if event.type == pygame.KEYDOWN and event.key == K_p: # Touche P enclenchée
        if pause == 0:
            clic_par_s[0] = PPS
            PPS = 0.0
            pause = 1
            compteur_s[0] = compteur
            compteur = 0

```

```

        elif pause == 1:
            PPS = round(clic_par_s[0], 1)
            compteur = round(compteur_s[0])
            pause = 0
    if pause == 0:
        if niveau >= 1:
            if event.type == pygame.MOUSEBUTTONUP: # Nous récupérons l'événement
du clic de la souris qui est relâché.
                if event.button == 1: # Le chiffre 1 signifie qu'il s'agit du
clic gauche.

                    if exploit == 0:
                        if piece_rect.collidepoint(event.pos):
                            points()
                        elif bouton_rect[1].collidepoint(event.pos):
                            objet1()
                        elif bouton_rect[2].collidepoint(event.pos):
                            objet2()
                        elif bouton_rect[3].collidepoint(event.pos):
                            objet3()
                        elif bouton_rect[4].collidepoint(event.pos):
                            exploits()
                        elif bouton_egal_rect.collidepoint(event.pos):
                            MENU = 2
                        elif bouton_croix_rect.collidepoint(event.pos):
                            MENU = 1
                    if a == 1:
                        if bonus_rect.collidepoint(event.pos):
                            PIECES += randint(1, 15)
                            nb_bonus += 1
                            a = 0
                    if exploit == 1:
                        if retour_rect.collidepoint(event.pos):
                            exploits()
            if event.button == 3: # Le chiffre 3 signifie qu'il s'agit du
clic droit.

                if piece_rect.collidepoint(event.pos):
                    malus()
                if bouton_rect[1].collidepoint(event.pos):
                    if AIDE != 1:
                        AIDE = 1
                    elif AIDE == 1:
                        AIDE = 0
                if bouton_rect[2].collidepoint(event.pos):
                    if AIDE != 2:
                        AIDE = 2
                    elif AIDE == 2:
                        AIDE = 0
                if bouton_rect[3].collidepoint(event.pos):
                    if AIDE != 3:
                        AIDE = 3
                    elif AIDE == 3:
                        AIDE = 0

if niveau == 0:
    if event.type == pygame.MOUSEBUTTONUP:
        if event.button == 1:
            if bouton_rect[0].collidepoint(event.pos):
                niveau += 1
if pause == 1:
    fenetre.blit(menu_pause, (0, 0)) # Affiche le menu pause.

```

```

    if nb_exploits == 9: #Si le nombre d'exploits est égal à 9 alors le jeu affiche la
fin.
        fin = 1
        Fin()
    pygame.display.flip() # Commande permettant d'actualiser l'affichage.
#Boucle while permettant de tout activer.
while quitter == False:
    PIECES += (PPS / IPS * 2) # Clics automatiques
    if pause == 0 and niveau >= 1 and fin == 0 :
        compteur += (5/6) / IPS # Compteur de temps
    elif pause == 1:
        compteur += 0 # Arrêt du compteur car le jeu est en pause.
    if jeu == 0: #Affichage du choix du mode de jeu
        texte_mode1 = POLICE2.render("MODE VITESSE", True, BLANC)
        texte_mode2 = POLICE2.render("MODE SUCCES", True, BLANC)
        texte = POLICE.render("Appuyez sur Entrée pour confirmer", True, (255, 255,
255))
        fenetre.blit(vitesse_fond0, vitesse_fond_rect)
        fenetre.blit(succes_fond0, succes_fond_rect)
        fenetre.blit(texte_mode1, (100, 100))
        fenetre.blit(texte_mode2, (390, 100))
    if choix == 1:
        fenetre.blit(vitesse_fond1, vitesse_fond_rect)
        fenetre.blit(texte, (55, 380))
        fenetre.blit(texte_mode1, (100, 100))
    if choix == 2:
        fenetre.blit(succes_fond1, succes_fond_rect)
        fenetre.blit(texte, (55, 380))
        fenetre.blit(texte_mode2, (390, 100))
    for event in pygame.event.get():
        if event.type == MOUSEBUTTONUP and event.button == 1:
            if vitesse_fond_rect.collidepoint(event.pos):
                choix = 1
            if succes_fond_rect.collidepoint(event.pos):
                choix = 2
        if event.type == pygame.KEYUP and event.key == K_RETURN:
            if choix == 1:
                jeu = 1
                choix = 0
            elif choix == 2:
                jeu = 2
                choix = 0
            if (event.type == pygame.KEYDOWN and event.key == K_ESCAPE) or event.type ==
pygame.QUIT: # Nous récupérons l'événement d'une touche qui est appuyée ET qu'il
s'agit de la touche "echap" ou que l'on ferme le jeu avec la croix.
                quitter = True # quit prend la valeur True ce qui va nous faire sortir de
la boucle while.
            if jeu == 1:
                vitesse()
            elif jeu == 2:
                succes()
        pygame.display.update() #Permet d'effectuer une actualisation de ce qui va être
afficher dans la fenêtre dans le but de tout montrer.
        fpsClock.tick(IPS) #Définition du nombre d'images par secondes.
pygame.quit() #Ferme la fenêtre quand on sort de la boucle while.

```