PROG MiniProjet 1

Jeu en programmation objet

Nous allons utiliser la bibliothèque pygame pour créer un jeu en programmation orientée objet
Les images, polices de caractères et le fichier jeu.py sont présents sur http://palmerio.drak.fr/

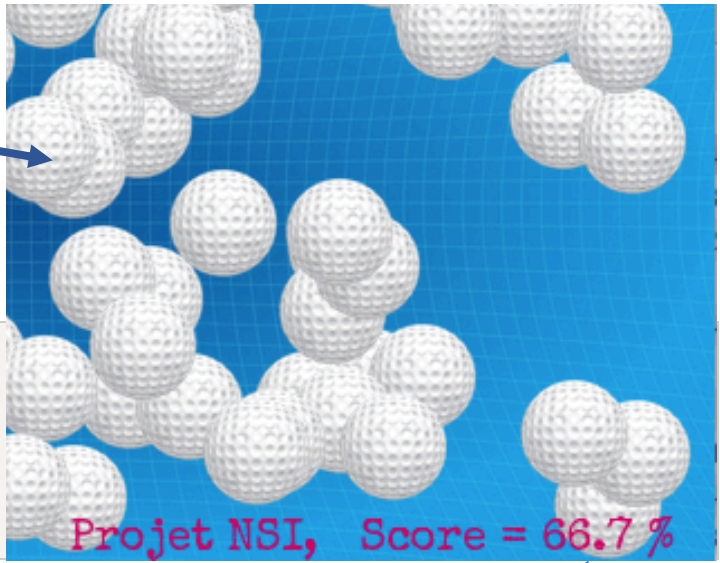
Défi 1

Transformer le script jeu.py en programmation orienté objet sous cette forme :

```
import pygame
import math
from random import *

#  ------------------------------------------------------------------------
#                      Implémentation de la classe Balle

class Balle():
        def __init__(self,largeur,hauteur):
                pass

        def updatepos(self):                    # ne retourne rien
                pass

        def getX(self):                         # accesseur
                pass

        #  .... et les autres méthodes de classse

#  ------------------------------------------------------------------------


# Création de l'instance
maballe=Balle(largeur,hauteur)

# Boucle while du jeu
```

Défi 2

Créer un tableau contenant 50 balles différentes
en mouvement désordonné.



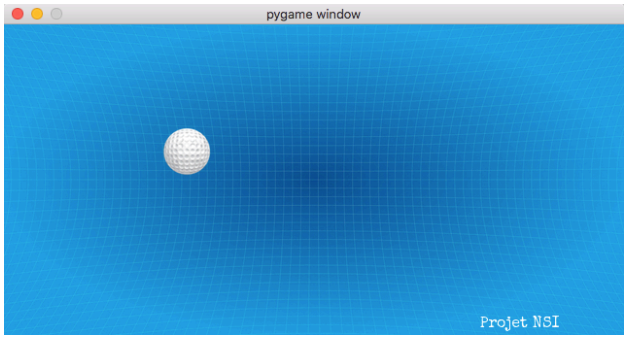
Défi 3

Un clic de souris sur une balle doit la faire disparaître
de l'affichage.

```
#   Pour vous aider, rechercher :

# for event in pygame.event.get():

# event.type == pygame.MOUSEBUTTONDOWN:
```

Afficher le score en bas à droite sous forme de pourcentage de réussite.

Script " jeu.py "




```
import pygame
import math
from random import *

# Initialisation de pygame
pygame.init()
clock = pygame.time.Clock()

# Dimensions de la fenetre
largeur = 638
hauteur = 320
screen = pygame.display.set_mode((largeur, hauteur))

# Chargement des images et de la police de caractères
fond = pygame.image.load('img/fond.jpg').convert()
balle = pygame.image.load('img/balle.png').convert_alpha()
font = pygame.font.Font('font/elite.ttf', 16)

# Texte qui sera affiché au bas de la fenetre
WHITE = pygame.Color(255, 255, 255)
text = font.render('Projet NSI', True, WHITE)

# Coordonnees de la POSITION initiale de la balle (générees aléatoirement)
x = random()*largeur-55
y = random()*hauteur-55

# Coordonnees de la VITESSE initiale de la balle (générees aléatoirement)
angle = 2*math.pi*random()
deltax = 5*math.cos(angle)
deltay = 5*math.sin(angle)

#  Boucle exécutée pendant toute la partie (Barre espace pour quitter le jeu)
continuer = True
while continuer:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            continuer = False
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
            continuer = False

    #calcul de la nouvelle position de la balle, la balle avance de deltax et de deltay suivant les axes
    x = x + deltax
    y = y + deltay

    # Rebonds sur les côtés du cadre pour gérer les changements de directions
    if (x>largeur-50 and deltax>0) or (x<0 and deltax<0):
        deltax = -deltax
    elif ((y>hauteur-50 and deltay>0) or (y<0 and deltay<0)):
        deltay = -deltay

    # Affichage des images à l'écran
    screen.blit(fond, (0,0))
    screen.blit(text, (490,300))
    screen.blit(balle, (x, y))

    # Affichage de l'image avec une cadence de 50 images par seconde (modifiable )
    pygame.display.update()
    clock.tick(50)

#  Fin de la boucle
```