Revisión bibliográfica y tareas del periodo del 17 de mayo al 24 de mayo

Miranda Coloma

4° Medio B

REVISIÓN BIBLIOGRÁFICA

<https://www.youtube.com/watch?v=7AKatTpNSNQ&list=PLE3D1A71BB598FEF6&ab_channel=MarekBelski>

Es posible randomizar la aparición de objetos dentro de la pantalla de un juego. Para esto se usa la librería Random. Primero se establece el número de los objetos que se quiere que aparezcan (por ejemplo 10), después se declara una lista vacía que contendrá dichos objetos y después se declara un ciclo for que vaya a iterar en el rango del número de objetos que se declaró primero. Dentro de este ciclo se hace toda la randomización de la posición y tamaño de los objetos: primero se declara un número aleatorio para el tamaño del objeto (usando la función ##random.randint()##, con un rango estipulado por el programador) , aunque podría establecerse también un tamaño constante; segundo se definen números aleatorios (##randint##) para el lugar que ocupará dentro del espacio el objeto, primero en el eje x y después en el eje y, dejando como valor máximo el ancho y alto de la pantalla, respectivamente (para que el objeto no salga aleatoriamente fuera de la pantalla); tercero, se declara el color del objeto, que también puede ser randomizado a través de la función ##random.choice()## que elige un objeto aleatorio dentro de una lista (para este caso, se haría una lista que contenga los colores que se quieren usar), o bien se puede determinar el color de modo constante; en cuarto y último lugar, se usa la función ##list.append()## para agregar el nuevo objeto dentro de la lista vacía creada al principio (usando también dentro del objeto todos los parámetros anteriormente definidos), y así con todas las iteraciones.

Ejemplo del código:

colors=[black, red, green, blue] ##todos estos definidos anteriormente con sus valores RGB como variables

##también vamos a tener presupuesto que la clase del objeto ya está definida (que en este caso es un círculo)

number\_of\_circles=10

my\_circles=[]

for n in range(number\_of\_circles):

size=random.randint(10, 20)

x=random.randint(size, screen\_width-size)

y=random.randint(size, screen\_height-size)

color=random.choice(colors)

my\_circles.append(MyCircle((x, y), size, color)) ##MyCircle es la clase círculo

##después para mostrar estos objetos

for my\_circle in my\_circles:

my\_circle.display()

(Esta idea de randomizar la aparición de objetos en pantalla puede ser de utilidad para hacer que aparezcan plataformas de forma aleatoria en el juego) (o eventualmente para hacer que salgan enemigos de manera aleatoria)

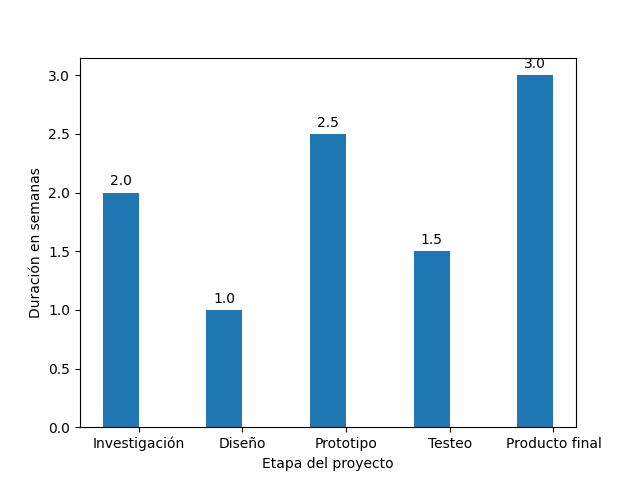
Para trabajar con vectores dentro del juego se puede usar la ayuda de la librería Euclid (anteriormente conocida como Pyeuclid) que contiene funciones relacionadas a las matemáticas de gráficas en 2D y 3D.

MÉTRICAS DE DESEMPEÑO DEL JUEGO

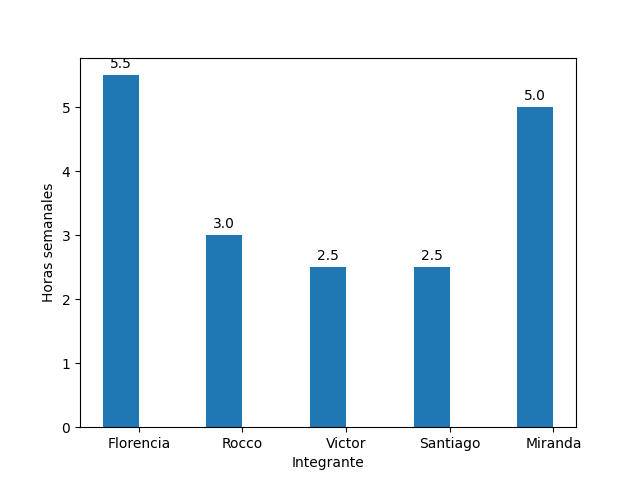
1. Diferencia de tiempo entre input-output
2. Calidad de nitidez de la imagen y el color (tal vez en una escala apreciativa de carácter subjetivo que respondan los sujetos de testeo del juego)

MÉTRICAS DE DESEMPEÑO DEL PROYECTO

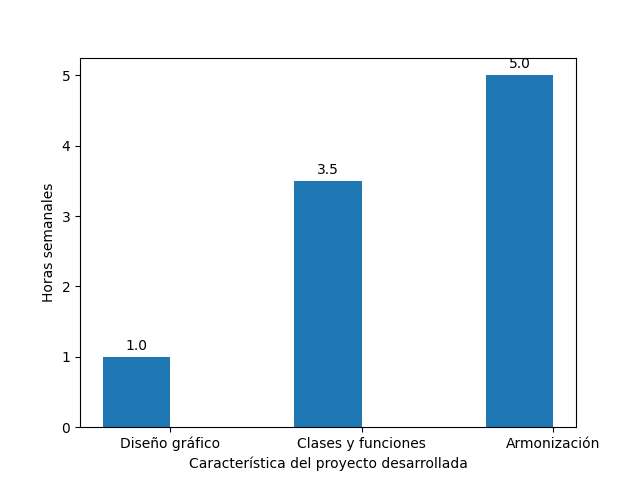
1. Duración en semanas de cada una de las etapas del proyecto (investigación, diseño, elaboración de prototipo, testeo, elaboración del producto final)



1. Horas trabajadas por cada integrante semanalmente



1. Horas dedicadas semanalmente a cada característica del proyecto (diseño gráfico, elaboración de clases y funciones, articulación de componentes)



¿CÓMO SE DEFINE LA VELOCIDAD DE ITERACIÓN EN UNA LISTA?

<https://stackoverflow.com/questions/44677606/how-to-measure-the-speed-of-a-python-function>

<https://docs.python.org/2/library/timeit.html>

Una de las formas para definir qué tan rápido va la iteración de una lista es con la función ##timeit##, que permite “cronometrar” pequeños pedazos de código en python.

Ejemplo:

>>> import timeit

>>> timeit.timeit('"-".join(str(n) for n in range(100))', number=10000)

0.8187260627746582