

Pensamiento Computacional
Prof. Darío Creado F.
Fecha: 05/05/22

Nombre: _____
Prueba 2
Tiempo límite: 1 hr 15 min

- **NO ABRAS el cuadernillo de la prueba hasta que se les dice que comiencen. Escribe tu nombre en la parte superior y lee las siguientes instrucciones.**
- Organiza tu trabajo, de forma razonablemente ordenada y en el espacio provisto. Si deseas que algo no sea revisado, por favor táchalo pulcramente. Sólo se calificará lo respondido en estas hojas, si necesitas más espacio para el desarrollo puedes adicionar hojas con previo aviso durante el desarrollo de la evaluación.
- Puedes usar cualquier apunte o resultado de clase, actividades o información anotada en tu cuaderno.
- La entrega deberá ser por medio de un archivo comprimido (.rar o similar) que contenga los códigos con extensión .py. El comprimido deberá ser enviado al mail del profesor y subido al repositorio de GitHub personal.
- No es necesario tiempo reescribiendo definiciones o axiomas en la prueba. Puedes usar una calculadora convencional. No se puede usar un teléfono celular, una computadora o una calculadora programable.
- Es obligatorio realizar los problemas 1 y 2 de la evaluación. La resolución de la primera parte de la prueba les permitirá obtener una bonificación extra en la nota.

Problem	Points	Score
1	2	
2	2	
3	2	
Total:	6	

1. (2 points) Una empresa de videojuegos lo contrata por su habilidad al programar, y le piden que quede a cargo de la creación de los personajes de dicho videojuego. El juego consiste de varios personajes, los cuales tienen ciertas características, como su nombre, su poder de ataque, su resistencia a los golpes, su velocidad y su vida. Su jefe le pide que la vida sea 100 por defecto. Además, los personajes interactúan entre estos, de manera que pueda atacar a otro personaje, o ver si es más rápido que otro. Le dicen que ocupe la siguiente formula para el daño.

$$\text{Daño} = \left\lfloor \frac{\text{Ataque}}{\text{Defensa Oponente}} \right\rfloor$$

Para simplificar esta creación, decide programar una clase que genere a dichos personajes del videojuego.

2. (2 points) Las imagenes se representan como una lista de listas rectangular de pixeles, donde cada pixel es una lista de 3 colores: rojo, verde y azul (esto se traduce en ingles como RGB = R: red, G: green, B: blue). Cada componente del pixel toma un valor entero de 0 a 255 que indica la intensidad de cada uno de los colores. Un ejemplo de imagen es el siguiente:

```
1 imagen = [  
2 [[58,23,125], [150,255,0], [0,0,0] , [14,0,99] ],  
3 [[0,0,200] , [34,25,0] , [0,65,0] , [8,0,45] ],  
4 [[58,0,1] , [70,25,0] , [0,80,165], [255,255,255]]  
5 ]
```

Debes escribir primero una función que reciba una imagen y retorne la imagen invertida, es decir, cada valor de un pixel se debe recalcular de la siguiente manera: **valor nuevo = 255 - valor antiguo**. Tomando el ejemplo anterior, la imagen inversa es:

```
1 inversa = invertir_imagen(imagen)  
2 inversa = [  
3 [[197, 232, 130], [105, 0, 255] , [255, 255, 255], [241, 255, 156]],  
4 [[255, 255, 55] , [221, 230, 255], [255, 190, 255], [247, 255, 210]],  
5 [[197, 255, 254], [185, 230, 255], [255, 175, 90] , [0, 0, 0] ]  
6 ]
```

Por ultimo, escribe una función que rote la imagen en 90 grados en sentido horario, es decir, que la ultima fila pase a ser la primera columna, la penúltima fila la segunda y así sucesivamente. No es lo mismo que trasponer la imagen. Rotando el ejemplo:

```
1 rotada = rotar_derecha(imagen)
2 rotada = [
3   [[58, 0, 1] , [0, 0, 200], [58, 23, 125]],
4   [[70, 25, 0] , [34, 25, 0], [150, 255, 0]],
5   [[0, 80, 165] , [0, 65, 0] , [0, 0, 0] ],
6   [[255, 255, 255], [8, 0, 45] , [14, 0, 99] ]
7 ]
```

Una ultima consideración es que las funciones no deben alterar la imagen original, solo deben retornar una copia modificada de esta.

3. (2 points) Se pide dibujar una línea de piso, junto a una caja (cuadrado) que contenga dentro un **sprite** a elección (personaje de videojuego, película, etc.) utilizando la librería Pygame.

Solución Problema 1

Solución Problema 2