

Introducción

Como parte del desarrollo de un nuevo producto en una organización para la cual usted trabaja, se le encomendó a usted la creación de una aplicación prueba de concepto que permita comprobar las capacidades del producto. Mientras usted desarrolla dicha aplicación, la organización decide que sería conveniente realizar otra aplicación similar. Las pruebas por realizar tienen el propósito de evaluar el desempeño en escenarios en que la cantidad de elementos en pantalla es mayor, y no saben si esto puede tener un efecto significativo sobre el desempeño en comparación a otros computadores similares.

Para el momento en que los requisitos para esta nueva aplicación de prueba son definidos, usted ya ha terminado la primera aplicación. Se considera que su experiencia le permitiría completar la nueva aplicación más rápidamente que otro de los trabajadores disponibles, por lo cual se le asigna a usted esta tarea. Se le provee la siguiente especificación de requisitos a cumplir.

Secciones de código

Para el movimiento de la nave se utilizó programación orientada a objetos, sumado a la clase `Sprite` que viene incluida en la biblioteca `pygame`. El proceso para lograr el movimiento de la nave consiste en extraer el rectángulo de su `Sprite` con la función `self.image.get_rect()`. Una vez obtenido dicho rectángulo se le da una posición inicial en el eje `X` y el eje `Y`. Para actualizar la posición de la nave se utiliza la función `update(self)` propia de `pygame`, dentro de esta función se define `pygame.key.get_pressed()` lo cual indica al programa que debe leer los cambios en el teclado del computador. Una vez hecho esto se define la velocidad de la nave, definida en 6 para este caso. Con la velocidad definida basta con indicar que al presionar las teclas arriba y abajo se actualice la posición `Y`, aumentado y disminuyendo su valor respectivamente, lo mismo para las teclas derecha e izquierda, pero cambiando la posición en `X`.

Para el movimiento de los asteroides también se recurrió a programación orientada a objetos y la clase `Sprite`, además se utilizó la biblioteca `random` para su velocidad y posición inicial en `x`. al crear un asteroide su velocidad se define con `random.randrange(2, 8)` lo cual hace que el asteroide aparezca con una

velocidad aleatoria entre 2 y 8. Una vez es creado el asteroide se mueve siempre hacia abajo a la derecha, una vez que choca con el borde de la pantalla su posición se invierte en función de la esquina golpeada, es decir si choca con el lado derecho pasara a moverse hacia la izquierda y viceversa, este comportamiento es el mismo para los borde superior e inferior. Una vez es asteroide toca un límite de la pantalla también cambia su velocidad volviendo a elegir entre un valor aleatoria entre 2 y 8.

Para el manejo de los puntajes y su organización se utilizó un algoritmo de Quicksort, para este se crearon las funciones con las cuales se realiza el Quicksort, la función Quicksort recibe una lista de entrada y retorna la misma lista ordenada. Cuando el jugador termina la partida, independientemente de si gana o pierde, su puntuación es añadida a la lista de puntajes y con la ayuda del Quicksort se obtiene una lista con los 7 mejores puntajes, misma que es añadida a un archivo txt para poder guardar dichas puntuaciones y que estas sean mostradas en la pantalla de mejores puntuaciones.

Resultados

La primera figura muestra la pantalla principal del juego desde la cual podemos acceder a las pantallas secundarias, además podemos iniciar el juego

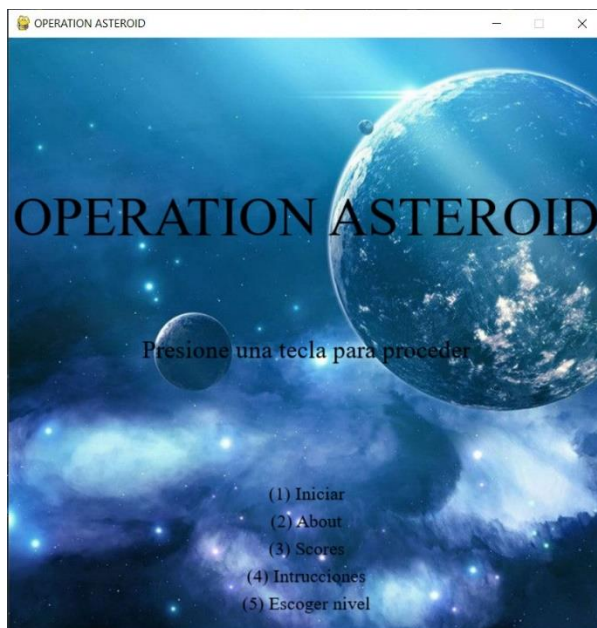


Figura 1. Menú principal de juego

En la figura 2 podemos observar la pantalla donde se digita el nombre del usuario, este nombre será el que se utilizará en caso de que el jugador supera alguno de los mejores 7 puntajes.

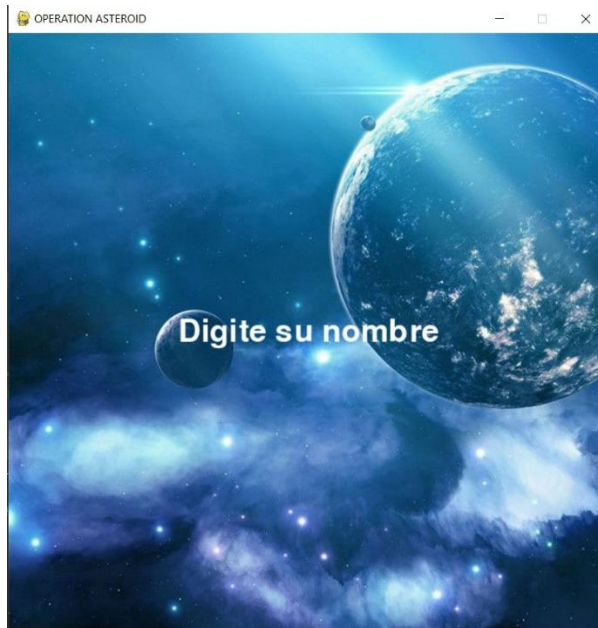


Figura 2. Pantalla para introducir el nombre

En las figuras 3, 4 y 5 podemos observar las diferentes pantallas secundarias que conforman el juego. Primero la pantalla de instrucciones donde se muestra las indicaciones básicas para jugar, segundo la pantalla complementaria, donde se muestra información importante acerca del desarrollo del juego y finalmente la pantalla de mejores puntajes, donde están los 7 mejores puntajes obtenidos por lo jugadores



Figura 3. Pantalla de instrucciones

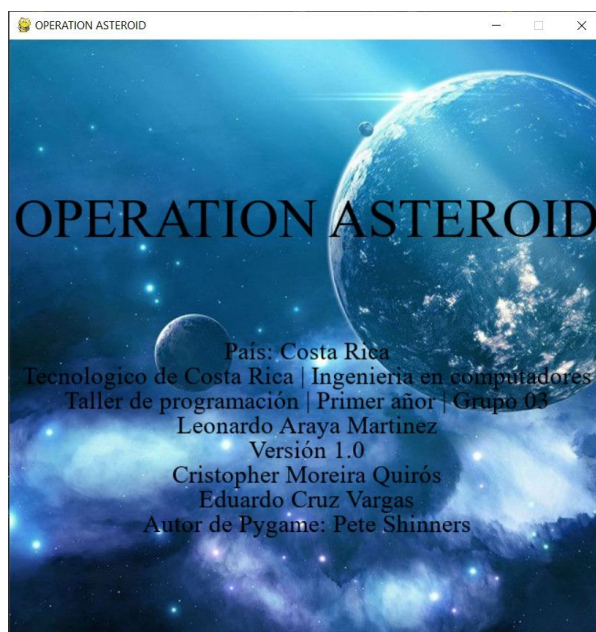


Figura 4. Pantalla complementaria

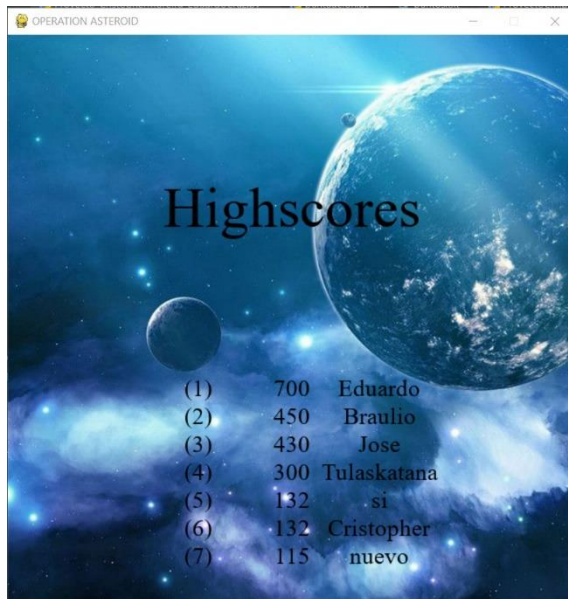


Figura 5. Pantalla de mejores puntajes

En la figura 6 se expone el menú de selección de niveles, siendo el nivel 1 el modo fácil, el nivel 2 el modo medio y el nivel 3 el modo difícil, si se supera cualquiera de estos niveles el juego avanzará automáticamente al siguiente y detendrá su ejecución al superar el nivel 3



Figura 6. Pantalla de selección de nivel

Por último, en las figuras 7, 8 y 9 vemos los diferentes niveles que posee el juego, se aprecia la nave del jugador y los asteroides moviéndose por la pantalla, además podemos observar como la cantidad de asteroides aumenta en función del nivel.

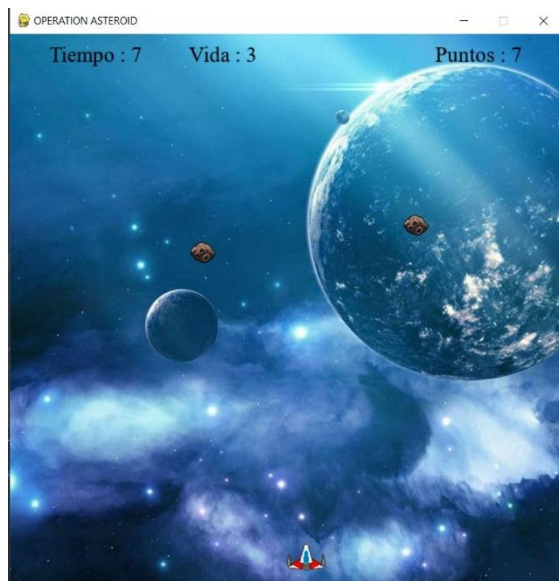


Figura 7. Nivel 1 del juego

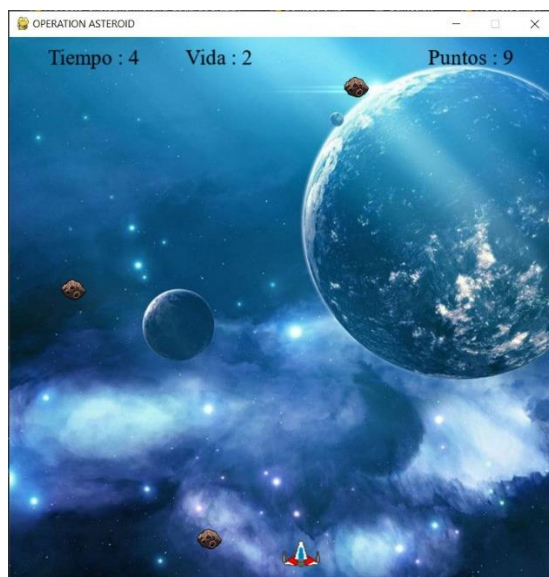


Figura 8. Nivel 2 del juego

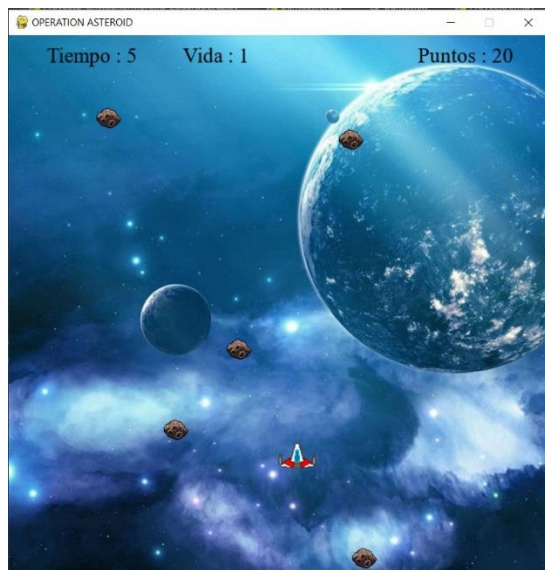


Figura 9. Nivel 3 del juego