

Licenciatura en Sistemas

Trabajos Práctico

Adivina el Producto

(2do semestre, 2023)

Resumen: Juego Python con Pygame, Adivina el Producto

COM: 01

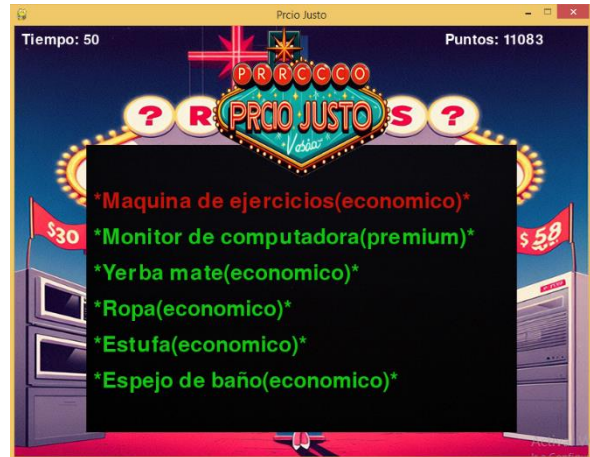
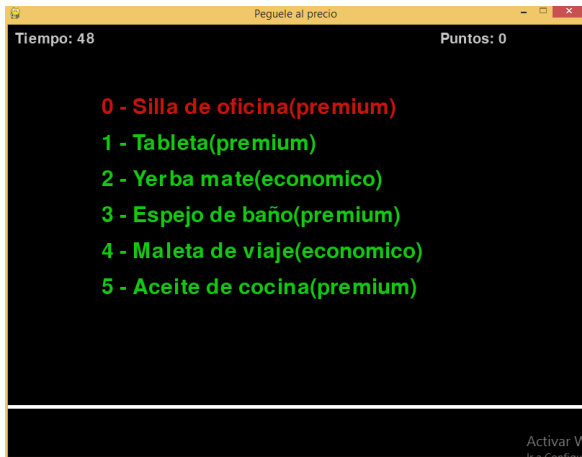
Profesores:

- Santiago Montiel (smontiel@ungs.edu.ar)
- Nora Martínez (namartin@campus.ungs.edu.ar)
- Jorgelina Rial (jrial@campus.ungs.edu.ar)

Integrantes:

- Ponce, Dylan Alexander
- Conde, Nahuel Agustín
- Lobos, Aaron Alexander
- Rojas, Thomas Daniel

1. Introducción En el presente informe se documenta el proceso de continuación y mejora del desarrollo de un videojuego. El proyecto originalmente se encontraba en fase intermedia, y el equipo asumió la tarea de finalizarlo e implementar nuevas funcionalidades, entre ellas. La inclusión de un menú, música, sistema de puntuación y cambios en el control mediante el clic del mouse, e incorporación de imágenes para mejorar la estética del juego.



2. Desarrollo El principal desafío se centró en comprender y programar las funciones clave del juego. La dificultad radico en entender el recorrido del programa, así como los enunciados. Una vez que conseguimos interpretar estos enunciados, logramos desarrollar las funciones.

2.1 Descripción general:

La solución se centra en el manejo de productos almacenados en un archivo de texto, donde obtenemos productos al azar y comparando precios, obtenemos puntos por aquellos productos "similares". Las funciones esenciales para lograr esto son: '*lectura*', '*buscar_producto*', '*dameProducto*', '*esUnPrecioValido*', '*procesar*', '*dameProductosAleatorios*', '*misSimilares*', '*noSimilares*'. El juego se complementa con un menú, para mejorar la experiencia visual de los jugadores, como también de un ranking que registra nuestras mejores puntuaciones.

2.2 Funcionalidades principales:

Def lectura (): Esta función lee un archivo de texto que contiene información sobre productos y los carga en una lista. Los datos se organizan en sub-listas que contienen el nombre del producto, su precio económico y premium.

Def buscar_producto (): Selecciona un producto de la lista al azar, y le asigna su calidad (económica o premium). Utiliza función **random.choise** para elegir producto, y le asigna aleatoriamente su calidad.

Def dameProducto (): Selecciona un producto valido, a partir de obtener un aleatorio de *buscar_producto*, verifica si es válido con la función *esUnPrecioValido*, si se cumple esto, este producto se convierte en mi referencia, mi producto_principal, si no se cumple sigue buscando, hasta obtener un producto valido.

Def esUnPrecioValido (): Su objetivo es verificar si un producto es válido, esto quiere decir que compara el producto dado, y va recorriendo la lista, con el fin de encontrar productos que tengan precios similares, teniendo en cuenta su margen. Si el contador

supera a 3, quiere decir que mi producto es válido, encontrando similares.

Def procesar (): Calcula los puntos del jugador a partir de las elecciones que realiza, compara mi producto_principal con el producto_candidato, si el candidato entra en el margen quiere decir que es correcto, obteniendo punto por acierto, de lo contrario no suma a mi puntuación.

Def dameProductosAleatorios (): Proporciona una lista de productos para que el jugador elija. Utilizar *misSimilares* y *noSimilares*, dos funciones implementadas por nosotros para obtener las condiciones específicas. Por otro lado mezcla y combina los productos para formar la lista final.

Def misSimilares (): Encuentra productos similares a uno dado. Utilizando aleatoriedad y margen para seleccionar productos similares, evitando repeticiones y garantizando 2 productos similares.

Def noSimilares (): Es similar a *misSimilares*, pero con el objetivo de encontrar 3 productos que no cumplan con la similitud.

2.3 EXTRAS Implementados:

Menú (Menú.py): Hemos implementado un menú interactivo que utiliza botones gráficos con efectos visuales y sonoros. Cambios en el color de los botones, sonidos al pasar el mouse y clics, proporcionando una mejor experiencia al usuario.

Ranking (Ranking.py): Se incorporó un módulo de ranking que muestra las mejores puntuaciones. Este módulo también utiliza efectos visuales y sonoros, junto con imágenes de fondo. Funciona leyendo un archivo de texto, en el cual se guardan todos los puntos que ha hecho el jugador en sus partidas y los muestra en una lista ordenada de mayor a menor.

Mouse: Con líneas de código como *e.type == MOUSEBUTTONDOWN* nos encargamos de que el juego funcione con el click izquierdo del mouse en vez de tener que escribir los números en el teclado. Esta función también aplica para el menú y el ranking.

Para implementación de la música y de la imagen, utilizamos funciones ya incluidas en Pygame, para la música usamos *pygame.init () - pygame.mixer.init () - pygame.mixer.music.load ('Soundtrack_Play.mp3')* que cumplen la función de inicializar la música que agreguemos. Por otro lado para las imágenes utilizamos *pygame.image.load ('Menu_Picture.png')*. Para la implementación de estos dos elementos no surgieron problemas, ya que encontramos foros donde mostraban ejemplos de cómo implementar estas funciones.

3. Conclusiones:

El trabajo fue bastante complejo, en un principio, por la falta de experiencia al usar una biblioteca como PyGame, así también el encontrar o descifrar algunas líneas del programa dado, tanto para realizar nuestras modificaciones como para comprender su recorrido. En grupo hemos desarrollado ideas, las dudas que teníamos pudimos resolverlas, para intentar y lograr comprender el programa, con el fin de que se nos haga más fácil la tarea y lograr los objetivos pedidos. A pesar de errores, y búsquedas de soluciones, el trabajo fue complejo pero divertido al mismo tiempo. Y una experiencia nueva el crear un videojuego.