**Instituto Tecnológico de Costa Rica**

**Unidad de Computación**

**Proyecto 1**

**Estudiantes:**

**Gherlan Steve Cerdas Rodríguez**

**Sede San Carlos**

**Fecha: 10/09/2024**



**Introducción**

Este proyecto implementa un crucigrama tridimensional (3D) utilizando Python y Pygame, permitiendo a los jugadores interactuar con una versión avanzada del clásico juego. El crucigrama está dividido en dos caras —frontal y lateral—, y los usuarios pueden navegar entre ellas usando las teclas de flechas. El sistema no solo permite a los jugadores resolver crucigramas, sino que también integra una funcionalidad para crear crucigramas personalizados, almacenar estos en archivos binarios y cargar partidas previamente guardadas.

La interfaz gráfica ofrece una experiencia visual similar a un crucigrama tradicional, pero con la novedad de poder moverse entre diferentes vistas, mientras que la estructura interna del juego gestiona la creación automática de las palabras en la cuadrícula, con la posibilidad de compartir los crucigramas entre jugadores.

**Análisis del Problema**

El desarrollo de este crucigrama 3D presenta varios retos que se abordan mediante distintas estrategias en el código:

Al tener dos caras conectadas, el jugador necesita moverse entre ambas de manera fluida, manteniendo el contexto de las palabras y definiciones. La implementación de eventos de teclado permite gestionar la navegación entre la cara frontal y lateral utilizando las flechas de dirección. Esto supone un desafío en cuanto al manejo simultáneo de dos matrices que representan cada cara del crucigrama, lo que se resuelve actualizando las entradas y dibujando la cara correspondiente según la selección del jugador.

La disposición automática de palabras en el crucigrama requiere encontrar coincidencias entre letras para formar una estructura lógica. La función acomodar\_palabras\_auto se encarga de posicionar las palabras en la matriz respetando las conexiones entre letras compartidas. Este proceso, aunque eficiente, conlleva complejidad debido a la necesidad de mantener la coherencia entre las posiciones de las palabras y las restricciones impuestas por las dimensiones limitadas de la cuadrícula.

Una característica clave del proyecto es la capacidad de almacenar y cargar crucigramas en formato binario. Este proceso incluye la serialización de las dimensiones del crucigrama, el número de palabras, las definiciones y la posición y dirección de cada palabra. La definición de un formato binario específico para los archivos introduce una capa adicional de complejidad en la gestión de la estructura de datos, pero permite una mejor optimización en cuanto a espacio y velocidad, además de ofrecer la posibilidad de compartir crucigramas entre diferentes usuarios o guardar progresos.

El uso de Pygame como motor gráfico añade el reto de representar dinámicamente el crucigrama en pantalla, mostrando las casillas y letras en tiempo real, además de las definiciones de las palabras. Es crucial que la interfaz se mantenga clara y accesible, especialmente al gestionar dos vistas del crucigrama. La función dibujar\_crucigrama se encarga de renderizar las casillas y letras correctamente, mientras que dibujar\_definiciones muestra las descripciones de las palabras en la interfaz de usuario.

Para comprobar si el jugador ha completado correctamente el crucigrama, se implementa un sistema de validación que compara las letras ingresadas con las palabras definidas originalmente. La función comprobar\_palabras valida si las palabras en el crucigrama coinciden con las correctas, brindando retroalimentación en pantalla para el jugador.

**Solución del Problema**

Este proyecto de crucigrama tridimensional en Python, desarrollado con la biblioteca Pygame, ha enfrentado varios retos en términos de interactividad, generación automática, almacenamiento y gestión de archivos binarios, así como en la representación visual y validación del juego. Para resolver estos problemas, se han implementado diversas soluciones que permiten ofrecer una experiencia fluida y eficiente para los jugadores.

Uno de los aspectos principales fue lograr la interactividad y la navegación entre las caras del crucigrama. El juego está dividido en dos matrices (frontal y lateral), y el jugador puede moverse entre ellas usando las teclas de dirección izquierda y derecha. Esto se resuelve mediante la captura de eventos de teclado en Pygame, que alternan entre las dos caras del crucigrama. Las entradas del jugador se sincronizan entre las matrices, lo que asegura que la transición entre vistas sea fluida y sin pérdida de datos.

Para la generación automática del crucigrama, se diseñó una función que acomoda las palabras de manera eficiente en la cuadrícula. Se solicita al usuario que introduzca las palabras y sus definiciones, y el sistema divide estas palabras entre las dos matrices. Luego, el algoritmo acomoda automáticamente las palabras basándose en las letras comunes entre ellas, utilizando estructuras como Counter de la biblioteca collections para encontrar coincidencias. Este enfoque garantiza que las palabras se conecten correctamente dentro de la matriz, formando un crucigrama coherente.

En cuanto a la gestión de las entradas del usuario, el juego permite que el jugador ingrese letras en las casillas correspondientes usando el teclado. Las casillas editables son aquellas que contienen palabras, mientras que las vacías están bloqueadas para evitar errores. Esta funcionalidad se implementa mediante la captura de eventos de teclado en Pygame, permitiendo ingresar o borrar letras, que luego se muestran visualmente en la cuadrícula actualizada.

Uno de los aspectos clave del proyecto es la gestión de archivos binarios para el almacenamiento y carga de crucigramas. El formato binario está diseñado para ser eficiente, incluyendo datos sobre las dimensiones del crucigrama, el número total de palabras, y para cada palabra, su longitud, definición, posición y dirección. Esto permite guardar crucigramas completos y luego cargarlos sin pérdida de información. El archivo binario tiene un encabezado que incluye la versión del formato y las dimensiones del crucigrama, lo que asegura que se pueda adaptar a futuras expansiones sin comprometer la compatibilidad.

La representación visual del crucigrama se maneja con Pygame, que permite renderizar las casillas y letras en tiempo real. Las casillas con palabras se diferencian de las vacías mediante colores, lo que facilita la interacción del jugador. Además, se incluye un sistema para mostrar las definiciones de las palabras en pantalla, lo que ayuda al jugador a completar el crucigrama. La pantalla también incluye un cuadro de mensajes que brinda retroalimentación, como "Presiona Enter para comprobar" o "¡Ganaste!" cuando el jugador completa correctamente el crucigrama.

Para validar si el crucigrama ha sido completado correctamente, se implementa un sistema que compara las letras ingresadas por el jugador con las palabras originales. Si todas las palabras son correctas, el jugador recibe un mensaje de victoria, lo que cierra el ciclo de juego de manera clara

**Seudocódigo**

INICIAR Pygame

DEFINIR colores: BLANCO, NEGRO, GRIS\_CLARO, GRIS\_OSCURO, VERDE

DEFINIR tamaño de casilla y margen

DEFINIR fuente para el texto

FUNCION crear\_crucigrama():

INICIALIZAR diccionario\_palabras vacío

MIENTRAS el usuario no escriba 'salir':

SOLICITAR palabra al usuario

SOLICITAR descripción de la palabra al usuario

AÑADIR palabra y descripción al diccionario

SI no hay palabras ingresadas:

RETORNAR matriz vacía y diccionario vacío

FUNCION encontrar\_letra\_comun(palabra\_nueva, palabra\_existente):

RETORNAR la primera letra que sea común entre dos palabras, o None si no hay coincidencia

FUNCION mejor\_palabra\_inicial(diccionario):

PARA cada palabra en el diccionario:

CALCULAR el total de coincidencias con otras palabras

RETORNAR la palabra con más coincidencias

FUNCION acomodar\_palabras\_auto(diccionario):

INICIALIZAR lista colocadas, y matriz vacía

SELECCIONAR la mejor palabra inicial

COLOCAR la mejor palabra inicial en la posición central de la matriz

PARA cada palabra en el diccionario:

SI ya fue colocada, continuar

PARA cada palabra ya colocada:

ENCONTRAR letra común entre palabras

TRATAR de colocar la palabra en la matriz si coincide una letra

AJUSTAR la matriz a las dimensiones mínimas necesarias

RETORNAR matriz ajustada

DIVIDIR las palabras en dos grupos: frontal y lateral

CREAR las dos matrices: frontal y lateral, usando acomodar\_palabras\_auto()

RETORNAR matriz frontal, matriz lateral y diccionario de definiciones

FUNCION dibujar\_crucigrama(screen, matriz, entradas, seleccionada):

PARA cada casilla en la matriz:

DIBUJAR casilla vacía o con letra según corresponda

SI la casilla está seleccionada, resaltar

FUNCION comprobar\_palabras(matriz, entradas):

PARA cada casilla en la matriz:

COMPROBAR si la letra ingresada coincide con la palabra correcta

RETORNAR True si todas las palabras son correctas, False si hay errores

FUNCION dibujar\_mensajes(screen, ancho, alto, juego\_ganado, fuente):

DIBUJAR cuadro de mensaje en la parte inferior de la pantalla

SI el juego está ganado:

MOSTRAR mensaje de victoria

SI no:

MOSTRAR mensaje de comprobación

FUNCION dibujar\_definiciones(screen, ancho, alto, definiciones, palabras\_ubicacion, fuente):

DIBUJAR cuadro de definiciones en la parte inferior de la pantalla

MOSTRAR definiciones de las palabras junto con su ubicación (horizontal o vertical)

FUNCION obtener\_ubicacion\_palabras(matriz):

PARA cada fila en la matriz:

DETECTAR si hay palabras horizontales

PARA cada columna en la matriz:

DETECTAR si hay palabras verticales

RETORNAR diccionario de palabras con sus ubicaciones

FUNCION main():

CREAR crucigrama usando crear\_crucigrama()

OBTENER matrices frontal y lateral

INICIALIZAR la cara actual en "frontal"

CREAR la ventana de juego con tamaño basado en la matriz

INICIALIZAR matrices de entradas vacías para cada cara

INICIALIZAR variable de casilla seleccionada

MIENTRAS el juego esté corriendo:

DETECTAR eventos (teclado, ratón)

SI se presiona una tecla:

ACTUALIZAR entrada de la casilla seleccionada

CAMBIAR entre cara frontal y lateral si se presionan las flechas izquierda/derecha

SI se presiona Enter:

VERIFICAR si todas las palabras han sido ingresadas correctamente

DIBUJAR el crucigrama y las definiciones en pantalla

ACTUALIZAR la pantalla

SALIR del juego si se cierra la ventana

**Análisis de Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea/Requerimiento** | **Estado** | **Gherlan** | **Observaciones** |
| **Gestion del crucigrama** | Incompleto | 80% |  |
| **Guardado** | Completo | 80% |  |
| **Rejugabilidad** | Completo | 80% |  |

**Conclusiones**

El desarrollo de este proyecto ha permitido crear un crucigrama tridimensional interactivo en Python utilizando Pygame, que ofrece una experiencia visual dinámica y facilita la creación y resolución de crucigramas. El sistema es flexible al permitir la generación automática de crucigramas a partir de palabras ingresadas por el usuario, así como su almacenamiento y carga en formato binario. Esta solución brinda a los jugadores la posibilidad de crear y compartir crucigramas personalizados, proporcionando un balance entre jugabilidad y la complejidad que añade el espacio tridimensional.

Uno de los principales resultados obtenidos fue la integración de dos matrices (frontal y lateral) conectadas por el jugador mediante teclas de dirección, lo que ofrece una experiencia diferente a los crucigramas tradicionales. Además, el uso del formato binario para almacenar las partidas mejora la eficiencia en términos de espacio y velocidad, permitiendo que los juegos sean exportados e importados sin comprometer la integridad de los datos.

La implementación de funciones para validar las palabras ingresadas, gestionar las entradas del usuario y renderizar el crucigrama visualmente en tiempo real resultó ser una solución efectiva para asegurar una jugabilidad fluida. Asimismo, la integración de definiciones en pantalla y el manejo de eventos de teclado y ratón proporcionaron una interfaz amigable e interactiva.

En resumen, el proyecto cumplió con su objetivo principal de crear un crucigrama tridimensional interactivo que permite la creación, almacenamiento y resolución de crucigramas, utilizando técnicas eficientes tanto en el manejo de la interfaz como en el almacenamiento de datos.

**Recomendaciones**

Una posible recomendación sería continuar mejorando la interfaz gráfica, optimizando la distribución de las casillas y los cuadros de mensajes para proporcionar una experiencia visual más limpia y adaptada a distintos tamaños de pantalla. Esto podría mejorar la interacción del jugador, haciéndola más intuitiva y atractiva.

Otra recomendación sería la implementación de una funcionalidad de edición para que los usuarios puedan modificar un crucigrama previamente creado, ya sea añadiendo nuevas palabras o ajustando las posiciones de las existentes, sin tener que iniciar desde cero. Esto aumentaría la flexibilidad del sistema y permitiría una mayor personalización de los crucigramas.

También sería beneficioso explorar la opción de guardar y cargar múltiples estados del crucigrama. Implementar una funcionalidad que permita al jugador guardar el progreso a mitad de una partida, para retomarlo posteriormente, podría añadir una capa adicional de comodidad para los jugadores.

Finalmente, sería interesante considerar la incorporación de un sistema de puntuación o retroalimentación más avanzada, que motive a los jugadores a mejorar su rendimiento o competir entre ellos. Esto podría incrementar el atractivo del juego y su potencial como una herramienta de entretenimiento o aprendizaje.

# **Bibliografía**

Boddy, J. (2018). *PythonLoginAndRegister*. Obtenido de https://github.com/br34th3r/PythonLoginAndRegister.git

Singh, V. (11 de Julio de 2024). *Password Generator in Python*. Obtenido de Password Generator in Python