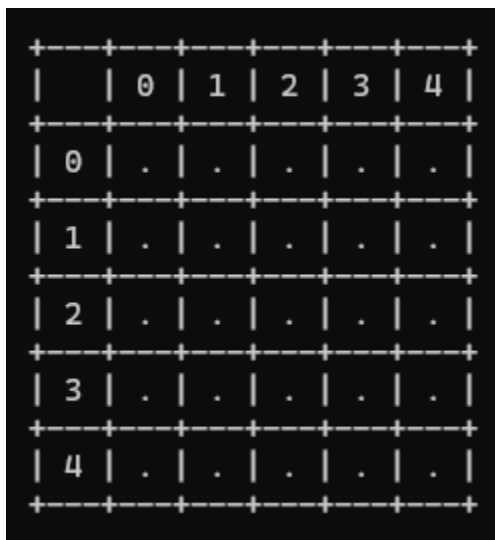




TRABAJO PRACTICO FINAL



TEMA 2: JUEGO DE BUSCAMINAS

Alumno: Eduardo Acuña González.

Carrera: Ingeniería en Informática

Materia: Lenguaje de Programación 1

Sección: TQ

Profesor: Diego P. Pinto Roa.

24 de julio de 2024.

San Lorenzo – Paraguay.

ÍNDICE

1. Introducción
2. Desarrollo
 - 2.1. Descripción General del Sistema
 - 2.2. Explicación de los Módulos y Algoritmos Utilizados
 - 2.3. Listado de Funciones del Programa
 - 2.3.1. Funciones del Inicio del Programa
 - 2.3.2. Funciones del Menú Principal
 - 2.3.3. Funciones de la Opción de Iniciar Juego
 - 2.3.4. Funciones de la Opción de Configuraciones
 - 2.3.5. Funciones de la Opción de Registro de Jugador
 - 2.3.6. Funciones de la Opción de Estadísticas
 - 2.3.7. Funciones de las Reglas del Juego
 - 2.3.8. Funciones Globales de Ayuda
 - 2.3.9. Funciones para el Juego
 - 2.3.10. Funciones de Gestión de Usuarios
 - 2.4. Metodología Utilizada
 - 2.5. Detalles Técnicos
 - 2.6. Resultados Obtenidos
 - 2.7. Análisis de los Resultados y Discusión
3. Conclusión

INTRODUCCIÓN

Este trabajo práctico se centra en el desarrollo de un juego de buscaminas utilizando el lenguaje de programación C. El objetivo principal es implementar un juego funcional con capacidad para gestionar usuarios, registrar puntuaciones y permitir la interacción entre el jugador y la computadora.

El propósito de este trabajo es crear un juego de campo de minas que no solo sea jugable y divertido, sino que también implemente una serie de funcionalidades adicionales como la gestión de usuarios y estadísticas. Los objetivos incluyen:

- Implementar un juego de campo de minas en C.
- Desarrollar un menú principal con varias opciones.
- Gestionar usuarios y puntuaciones.
- Mostrar las estadísticas del juego y proporcionar una experiencia de usuario completa.

El desarrollo de juegos es una manera de aprender sobre programación y lógica computacional. Implementar un juego de campo de minas en C no solo proporciona experiencia práctica en la programación, sino que también ayuda a entender conceptos importantes como la gestión de memoria, el manejo de estructuras de datos y la implementación de algoritmos.

DESARROLLO

2.1. Descripción General del Sistema

El sistema desarrollado es un juego de campo de minas en C que permite al jugador interactuar con un tablero, descubrir celdas y evitar minas. El juego incluye una interfaz de menú para gestionar las opciones del juego, configuración de usuarios y visualización de estadísticas.

2.2. Explicación de los Módulos y Algoritmos Utilizados

El proyecto está compuesto por tres archivos principales:

- **main.c:** Contiene el punto de entrada del programa y la lógica del menú principal.
- **funciones.c:** Implementa las funciones de juego y gestión de usuarios.
- **head.h:** Declara las estructuras de datos y funciones utilizadas en los otros archivos.

Bibliotecas Incluidas

<stdio.h>: Proporciona funciones para la entrada y salida estándar en C. Mostrar menús, instrucciones y mensajes en la pantalla, y leer las opciones seleccionadas por el usuario.

<stdlib.h>: Incluye funciones para la gestión de memoria dinámica, control de procesos y otras operaciones generales. Manejo de memoria dinámica, generación de números aleatorios y operaciones de control de procesos.

<string.h>: Proporciona funciones para manipular cadenas de caracteres. Gestión de nombres de usuarios, contraseñas y otras cadenas de caracteres.

<time.h>: Contiene funciones para manipular y obtener la hora y la fecha. Inicialización del generador de números aleatorios y gestión de fechas y horas relacionadas con las partidas jugadas.

Algoritmos Clave:

- **Generación del Tablero:** Algoritmo para crear un tablero de juego con minas distribuidas aleatoriamente.
- **Descubrimiento de Celdas:** Algoritmo para descubrir celdas y mostrar el número de minas adyacentes.
- **Gestión de Usuarios:** Funciones para registrar, modificar y eliminar usuarios, así como para mostrar estadísticas.

2.3. Listado de Funciones del Programa

2.3.1. Funciones del Inicio del Programa

void inicio_programa(); Muestra un mensaje de bienvenida o información inicial al usuario cuando el programa comienza.

2.3.2. Funciones del Menú Principal

void menu_principal(); Muestra el menú principal del programa y gestiona la navegación entre diferentes opciones.

void mostrar_menu_principal(); Imprime en pantalla las opciones disponibles en el menú principal.

int validar_opcion_menu_principal(); Valida la opción seleccionada por el usuario en el menú principal para asegurar que es válida.

2.3.3. Funciones de la Opción de Iniciar Juego

void iniciar_juego(); Inicia el juego, llamando a las funciones necesarias para configurar y comenzar una nueva partida.

void configuracion_tablero(); Configura el tablero del juego, estableciendo el tamaño y el número de minas.

void registro(); Maneja el registro de la partida actual, incluyendo la configuración inicial y la introducción de datos.

void mostrar_resultado(); Muestra el resultado final de la partida al usuario, incluyendo el estado del juego (victoria/derrota) y otros detalles relevantes.

void guardar_resultado(); Guarda el resultado de la partida en un archivo o en una base de datos para su posterior consulta.

2.3.4. Funciones de la Opción de Configuraciones

void mostrar_menu_configuracion(); Muestra el menú de configuraciones, permitiendo al usuario ajustar varias opciones del juego.

int validar_menu_configuracion(); Valida la opción seleccionada por el usuario en el menú de configuraciones.

void cambiar_nombre_de_usuario(); Permite al usuario cambiar su nombre de usuario en el sistema.

void cambiar_contrasena(); Permite al usuario cambiar su contraseña.

2.3.5. Funciones de la Opción de Registro de Jugador

void mostrar_menu_registro_jugador(); Muestra el menú para registrar un nuevo jugador o gestionar los jugadores existentes.

int validar_menu_registro_jugador(); Valida la opción seleccionada por el usuario en el menú de registro de jugadores.

int buscar_jugador(); Busca un jugador en la base de datos o en la lista de jugadores registrados.

void lista_jugadores_mostrar_menu_registro_jugador(); Muestra una lista de jugadores registrados, permitiendo al usuario seleccionar o gestionar jugadores.

void resregistro_jugador_mostrar_menu_registro_jugador(); Muestra el menú para registrar un nuevo jugador en el sistema.

2.3.6. Funciones de la Opción de Estadísticas

void mostrar_estadistica(); Muestra estadísticas generales del juego, como el número de partidas jugadas, victorias y derrotas.

void mayor_puntaje(); Muestra el jugador con el mayor puntaje registrado en el sistema.

void estadistica_de_usuario(); Muestra estadísticas específicas para un usuario, como el puntaje promedio y las partidas ganadas.

2.3.7. Funciones de las Reglas del Juego

void regla_juego(); Muestra las reglas del juego al usuario, explicando cómo jugar y qué esperar.

2.3.8. Funciones Globales de Ayuda

void setConsoleUtf8(); Configura la consola para que use la codificación UTF-8, permitiendo el manejo adecuado de caracteres especiales.

void enter_continuar(); Pausa la ejecución del programa y espera que el usuario presione una tecla para continuar.

void limpiar_pantalla(); Limpia la pantalla de la consola para mejorar la legibilidad del programa.

2.3.9. Funciones para el Juego

void inicializarTablero(int fila, int columna, int numMinas); Inicializa el tablero del juego con el número de filas, columnas y minas especificadas.

void colocarMinas(int numMinas); Coloca minas en el tablero en ubicaciones aleatorias.

void calcularMinasAdyacentes(); Calcula y almacena el número de minas adyacentes a cada celda del tablero.

void mostrarTablero(); Muestra el estado actual del tablero en la consola, revelando celdas descubiertas y ocultas.

void descubrirCelda(int x, int y); Descubre el contenido de una celda específica en el tablero.

void marcarCelda(int x, int y); Marca una celda específica como sospechosa (posiblemente contiene una mina).

int verificarVictoria(); Verifica si el jugador ha ganado la partida, comprobando si todas las celdas sin minas han sido descubiertas.

void mostrarPuntaje(); Muestra el puntaje actual del jugador.

void turnoJugador(); Maneja el turno del jugador, permitiéndole hacer una jugada en el tablero.

void turnoComputadora(); Maneja el turno de la computadora, permitiéndole hacer una jugada en el tablero.

void mostrarPuntajeJugador1y2(); Muestra el puntaje de ambos jugadores (Jugador y Computadora) en la consola.

2.3.10. Funciones de Gestión de Usuarios

void cargarUsuarios(); Carga la información de los usuarios desde un archivo o base de datos.

void guardarUsuarios(); Guarda la información de los usuarios en un archivo o base de datos.

void registrarUsuario(); Registra un nuevo usuario en el sistema.

void actualizarJugadorActual(); Actualiza la información del jugador actual (como el puntaje y el progreso).

2.4. Metodología Utilizada

Para desarrollar el juego se utilizó la metodología de programación incremental. Se comenzó con la implementación de las funcionalidades básicas del juego y se fueron añadiendo características adicionales como la gestión de usuarios y el menú de opciones. Se realizaron pruebas continuas para asegurar el correcto funcionamiento de cada componente.

2.5. Detalles Técnicos y Teóricos Relevantes

El juego utiliza estructuras de datos como matrices para el tablero y listas para gestionar usuarios. Se implementaron algoritmos de búsqueda y generación aleatoria para las minas. La interfaz de usuario es en consola y permite al jugador interactuar con el juego mediante comandos.

2.6. Resultados Obtenidos

El juego cumple con los requisitos establecidos y proporciona una experiencia de usuario completa. La gestión de usuarios y estadísticas funciona correctamente y el juego se ejecuta sin errores significativos.

2.7. Análisis de los Resultados y Discusión

Los resultados muestran que el juego es funcional y responde bien a las entradas del usuario. Sin embargo, se identificaron algunas áreas para mejorar, como la optimización del algoritmo de generación de minas y la interfaz de usuario para hacerla más amigable.

CONCLUSIÓN

El proyecto logró implementar un juego de campo de minas con éxito, incluyendo todas las funcionalidades requeridas. La gestión de usuarios y las estadísticas están bien integradas en el juego.

Los resultados demuestran la capacidad del sistema para ofrecer una experiencia de juego completa y funcional. La implementación de características adicionales como la gestión de usuarios y las estadísticas añade valor al proyecto y permite una mayor interacción con el usuario.

Algunas limitaciones incluyen la simplicidad de la interfaz de usuario y la falta de funcionalidades avanzadas. La gestión de usuarios puede mejorarse con una interfaz gráfica y una base de datos más robusta.

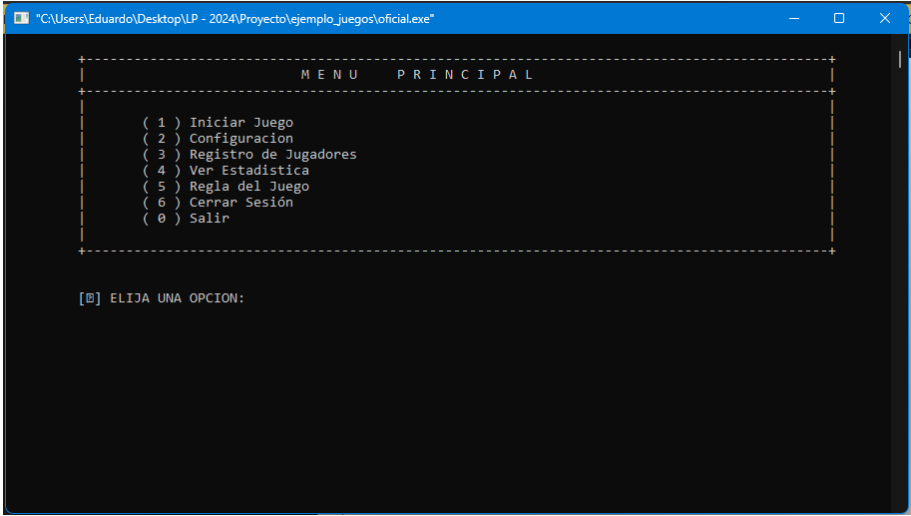
Bibliografía o Referencias

- **Deitel, P., & Deitel, H. (2015).** *C How to Program* (8th Edition). Pearson Education.
- **Joyanes Aguilar, L. (2006).** *Programación en C: Metodología, Estructuras de Datos y Algoritmos*. McGraw-Hill Interamericana.
- **Documentación de GCC.** Recuperado de sitio web de GCC.

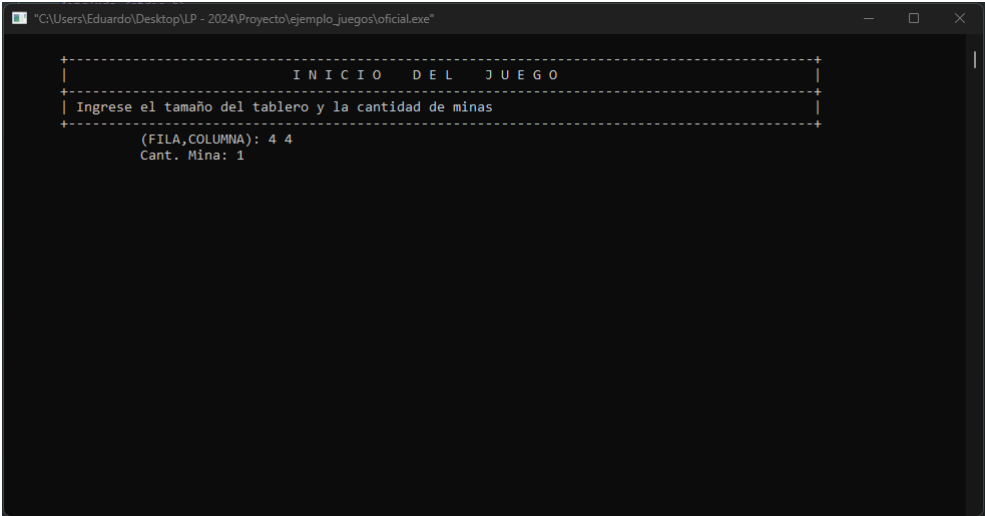
ANEXOS



Menú Principal del Juego de Buscaminas: La imagen muestra el menú principal del "Juego de Buscaminas". Esta es la primera pantalla que se presenta al usuario al ejecutar el programa. Desde este menú, los jugadores tienen la opción de iniciar sesión, registrarse o cerrar el programa.



Menú Principal del Juego de Buscaminas: La imagen a continuación muestra el menú principal del "Juego de Buscaminas" después de iniciar sesión o registrarse. Desde este menú, los jugadores pueden acceder a las principales funcionalidades del juego. Al seleccionar una opción, el usuario puede navegar a las diferentes secciones del juego, ya sea para jugar, ajustar configuraciones, o consultar estadísticas y reglas.



Inicio del juego de Buscaminas: La imagen muestra la pantalla de configuración del juego "Buscaminas" en modo consola, justo antes de iniciar una partida.