

DOCUMENTACIÓN TRABAJO PRÁCTICO FINAL

Cphalopod - Entrega Final

Alumnos: Enzo Erico y Julieta Monti

Estructuras de Datos Utilizadas y Funciones

Tablero Principal - Matriz 5x5 (Tablero)

Hemos empleado una matriz 5x5 denominada "tablero" como el tablero principal del juego. Esta matriz almacena los dados y sus respectivas cantidades, pero no atribuye la propiedad de pertenencia a un jugador en particular.

Tablero Secundario - Matriz 5x5 (TableroPosic)

Además, hemos utilizado otra matriz 5x5 llamada "tableroposic" como un tablero secundario. Esta matriz se encarga de registrar la propiedad de cada dado, es decir, a quién pertenece cada uno.

Vectores Locales para Coordenadas

En el proceso de entrada de coordenadas, hemos empleado dos vectores locales: uno para el usuario y otro para la computadora. Cada uno de estos vectores se utiliza para almacenar las coordenadas.

Uso de Punteros

Cada vector dispone de su propio puntero correspondiente, lo que nos permite acceder de manera eficiente a los elementos dentro de los vectores. El uso de estos punteros garantiza una manipulación efectiva de las coordenadas ingresadas por el usuario y la computadora durante el juego.

Gestión de Capturas Potenciales

Matriz de Capturas (4x3)

Para determinar y realizar un seguimiento de las posibles capturas en el juego, hemos implementado una matriz de dimensiones 4x3.

Contenido de la Matriz:

En la primera columna de la matriz, se utiliza un "activador" que puede tomar el valor 1 si se trata de una posible captura o 0 si no lo es.

La segunda columna almacena la coordenada en el eje X de la posible captura.

La tercera columna registra la coordenada en el eje Y de la posible captura.

Contador de Posibles Capturas (posibles capturas)

Además, hemos implementado un contador llamado "posibles capturas" que rastrea cuántas posibles capturas existen en cada turno. Esta variable nos permite mantener un seguimiento preciso de las oportunidades de captura disponibles.

Interfaz Gráfica con GTK 3.0

La interfaz gráfica se ha implementado utilizando GTK 3.0. Las signals (señales) de los elementos gráficos, como botones, están conectadas a funciones de callback específicas para manejar eventos. Esto asegura una interactividad adecuada entre la interfaz y la lógica del juego.

Configuración de la Partida

La configuración de la partida se realiza en las ventanas window2 y window3. Estas ventanas permiten al usuario establecer parámetros iniciales para la partida, como configuraciones de dificultad, selección de personajes, o cualquier otro aspecto relevante para el juego. Dependiendo de estas configuraciones, se inicia la partida con las condiciones específicas establecidas.

Función Principal: `on_button_tablero_clicked`

La función principal del código es `on_button_tablero_clicked`, la cual actúa como callback para los botones del tablero en la interfaz gráfica. La función toma dos parámetros: `GtkWidget *button` y `gpointer user_data`.

Descripción de la Función

Esta función se activa cuando se hace clic en un botón del tablero. Su objetivo principal es obtener las coordenadas del botón presionado y pasarlas como parámetro a la función `turnoUsuario`. Posteriormente, la función `turnoCompu` realiza su jugada, utilizando lógica de inteligencia artificial para tomar decisiones estratégicas. Luego se actualiza la parte gráfica según lo puesto en el backend (tablero y `tableroposic`).

Archivo `estadisticas.txt`

Las estadísticas del juego, como el número de partidas ganadas, perdidas y empates, son almacenadas en un archivo llamado `estadisticas.txt`. Este archivo sirve como un registro histórico de rendimiento del jugador a lo largo del tiempo. Se accede a este mediante lectura y escritura de archivos en C.

Mantenimiento de una Interfaz Consistente y Accesible

La implementación incluye un diseño de interfaz coherente que garantiza la accesibilidad a todas las ventanas en cualquier momento, evitando posibles errores en la navegación.

Descripción de la Inteligencia usada

La inteligencia artificial implementada en el juego se centra en la estrategia de captura de dados para la computadora. A continuación, se presenta una descripción detallada de las funciones clave que definen esta inteligencia:

1. Función `evaluarCaptura`:

- **Objetivo:** Evaluar la captura de dados del rival en una posición específica (x, y).
- **Metodología:**
 - Utiliza dos matrices, `tablero` y `tableroposic`, para representar el estado del juego.

- Verifica la presencia de un dado del rival en la posición actual y busca capturas en las cuatro direcciones adyacentes.
- Devuelve la cantidad total de puntos capturados como resultado de las capturas encontradas.

2. Función **estrategiaCapturaMejorada**:

- **Objetivo:** Buscar la mejor jugada de captura para la computadora.
- **Metodología:**
 - Itera sobre todas las posiciones del tablero ocupadas por dados de la computadora.
 - Llama a **evaluarCaptura** para cada posición y elige la jugada que maximiza la cantidad de puntos capturados.
 - Si no hay una jugada estratégica, busca entre los dados rivales y elige la posición con más dados rivales adyacentes.

3. Función **pedirJugadaCompu**:

- **Objetivo:** Obtener una jugada estratégica para la computadora.
- **Metodología:**
 - Llama a **estrategiaCapturaMejorada** para buscar la mejor jugada de captura.
 - Si encuentra una jugada estratégica, la devuelve.
 - Si no encuentra una jugada estratégica, intenta ubicarse en las casillas adyacentes a los dados del usuario.
 - Si aún no hay una jugada válida, realiza una jugada aleatoria, asegurándose de que la casilla esté libre.

Enfoque Estratégico:

- **Captura Optimizada:** La inteligencia prioriza capturas que maximizan los puntos obtenidos para la computadora.
- **Adaptabilidad:** Si no hay oportunidades de captura estratégica, la inteligencia se adapta buscando entre los dados rivales y toma decisiones basadas en la presencia de dados enemigos adyacentes.

Esta estrategia proporciona a la computadora la capacidad de tomar decisiones informadas y adaptarse a diferentes escenarios de juego, mejorando así la experiencia del usuario al enfrentarse a un oponente virtual desafiante y estratégico.

Puntos Fuertes y Débiles del Programa

Puntos Fuertes del Programa:

1. **Buena Modularización:**

- *Fortaleza:* La estructura del código está bien organizada en módulos y funciones, facilitando la comprensión y el mantenimiento. La adaptabilidad del backend para integrarse con la interfaz gráfica indica un diseño modular eficiente.

2. Buena Adaptabilidad del Backend:

- *Fortaleza:* La capacidad del código del backend para integrarse con la interfaz gráfica demuestra una buena adaptabilidad. Esto sugiere que el código del núcleo del programa es lo suficientemente flexible como para ser utilizado en diferentes contextos sin problemas.

3. Buena Consistencia e Interconectividad de Interfaces:

- *Fortaleza:* La consistencia en el diseño de la interfaz gráfica y la interconectividad entre ventanas proporciona una experiencia de usuario coherente y fluida. La adaptabilidad del backend para soportar estas interfaces refuerza la consistencia.

4. Buena Interfaz Gráfica:

- *Fortaleza:* La implementación de la interfaz gráfica usando GTK 3.0 se destaca por su diseño atractivo y funcionalidad. La adaptabilidad del backend para conectarse sin problemas con la interfaz gráfica mejora la usabilidad del programa.

Puntos Débiles del Programa:

1. Mucha Cantidad de Líneas:

- *Debilidad:* La presencia de una gran cantidad de líneas de código puede dificultar la comprensión y el mantenimiento del programa. Se recomienda revisar el código en busca de oportunidades de refactorización y simplificación.

2. Uso Excesivo de Variables Globales:

- *Debilidad:* El exceso de variables globales puede afectar la legibilidad y mantenibilidad del código. Se sugiere limitar el uso de variables globales y considerar alternativas, como pasar parámetros a funciones, para mejorar la encapsulación.

3. Falta de Uso de Structs en Casos Óptimos:

- *Debilidad:* La no utilización de structs en situaciones donde serían óptimas puede afectar la organización del código. El uso de structs puede mejorar la modularidad y la claridad al agrupar datos relacionados.

Dificultades Encontradas Durante el Trabajo

1. Escasa Documentación de Glade y GTK:

- *Desafío:* Una de las principales dificultades fue la limitada documentación disponible para Glade y GTK. La falta de recursos detallados y ejemplos específicos dificultó la comprensión y la implementación efectiva de la interfaz gráfica.
- *Impacto:* Este desafío aumentó la curva de aprendizaje y ralentizó el desarrollo de la interfaz gráfica. La falta de información detallada también generó obstáculos al intentar solucionar problemas específicos de la interfaz.

Guía de Utilización del Programa:

El programa ha sido diseñado con una interfaz gráfica intuitiva que facilita su utilización. A continuación, se presenta una guía paso a paso para configurar partidas y realizar jugadas:

1. Interfaz Gráfica Intuitiva:

- Todas las funciones y opciones son accesibles de manera intuitiva a través de la interfaz gráfica. Botones, menús y mensajes proporcionan una experiencia de usuario fácil de entender.

2. Configuración de Partidas:

- **Ventana 2 - Modo de Juego:**
 - Abre la ventana 2 para elegir el modo de juego.
 - Actualmente, solo está disponible la opción "Compu vs Humano".
 - Más modos de juego pueden agregarse en futuras actualizaciones.
- **Ventana 3 - Configuración de Partida:**
 - Ingresa el nombre del usuario para registrar estadísticas.
 - Selecciona el color que deseas para tus dados.
 - Elige quién empieza el juego (Usuario o Computadora).
 - Confirma tus selecciones para avanzar a la ventana 4.

3. Resumen de Configuración - Ventana 4:

- En esta ventana, se presenta un resumen de la configuración de la partida elegida.
- Verifica que la información sea correcta antes de comenzar la partida.

4. Realizar Jugadas:

- **En el Tablero - Ventana Principal:**
 - Haz clic en la casilla del tablero donde deseas colocar tu dado.
 - La partida se desarrollará alternando entre turnos del usuario y la computadora.

5. Finalización de la Partida:

- Al finalizar la partida, se mostrarán mensajes indicando el resultado (ganador, perdedor o empate).
- Se registrarán las estadísticas automáticamente, considerando el rendimiento del usuario en la partida.

Notas Adicionales:

- Se espera que el usuario conozca las reglas del juego para realizar las jugadas de manera efectiva.

- La ventana 2 y 3 son cruciales para configurar la partida según tus preferencias antes de comenzar.