

LENGUAJES DE PROGRAMACIÓN I

TRABAJO PRÁCTICO FINAL: HOPPITY

SEGUNDO SEMESTRE – AÑO 2023

Profesores:

Vicente González
Fernando Caballero

Alumnos:

Alexander Bernal
Erison Acosta

DOCUMENTACIÓN:

1.Descripción de las estructuras de datos y las funciones.

Para la representación del tablero se utilizó una matriz de 17x17, la primera fila y columna se usó para enumerar el tablero, la parte superior desde 1 hasta 16 y el costado a la izquierda de la misma forma, esto para poder ver mejor y entender las ubicaciones (casillas) que ocupan cada ficha. La matriz en donde se colocaron las fichas quedaría entonces 16x16, hay dos fichas “las negras” y “las blancas”, las fichas negras se colocan en la parte superior izquierda en forma triangular (las fichas negras se representan con la letra “N” en la matriz), las blancas se colocan en la parte inferior derecha, también de forma triangular (las fichas blancas se representan con la letra “B” en la matriz) y las casillas vacías se representan con “.” puntos.

También se creó structs, esto para estadísticas.

```
typedef struct {  
    char nombre[50];  
    int partidas_jugadas;  
    int partidas_ganadas;  
    int partidas_perdidas;  
} Jugador;
```

Para ir almacenando los datos estadísticos del jugador.

Funcionamiento del juego.

Para el jugador (humano) le irá apareciendo varias ventanas para hacer la configuración de la partida (elecciones de modo de juego, turnos, color de ficha y nombres), se usan variables globales para ir cargándoles valores numéricos de acuerdo a la elección que se haya hecho, por ejemplo, si el jugador (humano) aprieta el botón “HUMANO” en la parte de elección de turno, entra en la función **aprieta_turno_humano** (**GtkWidget *event_box, GdkEventButton *event, gpointer data**), y a la variable que está dentro de esta función, en este caso “Turno” se le asigna 1, si el turno es de la computadora, entra en otra función similar, pero a Turno se le asignará 2, lo mismo pasa cuando se elige el color de ficha, se aprieta por ejemplo, color negro y entrará en esta función **aprieta_color_negro** (), donde se repite el mismo caso, solo que el nombre de la variable es “Color”, si se selecciona negra, a la variable se le asignará el valor 1, pasa lo mismo si selecciono blanca, entra en una función similar y a la variable se le asigna el valor 2.

Para la creación de tablero se usó un Grid y un EventBox.

```
i = (GUINT_FROM_LE(event->y) / 33); // en "i" se guarda valor de fila actual, 33  
pixels tiene la imagen de la ficha  
j = (GUINT_FROM_LE(event->x) / 33); // en "j" se guarda valor de columna actual,  
33 pixels tiene la imagen de la ficha
```

Y con estas funciones se recogen las coordenadas en donde se ha hecho “click” en la tabla, al presionar o darle click a cualquier parte del tablero (ya sea su ficha o casilla vacía) este entra en la

función `eventBox(GtkWidget *event_box, GdkEventButton *event, gpointer data)`, en donde está la función mas importante para que el jugador pueda hacer sus movimientos, revisa si ha seleccionado ficha negra o blanca, de acuerdo a la elección entra en un “if” (para este ejemplo será el de las negras), dicha ficha se pinta de otro color al darle click (seleccionado), el “if” se encargaria solamente de pintar/seleccionar las fichas, luego de haber seleccionado la ficha a mover una variable “bandera” cambia de valor para habilitar otro “if” que seria la revisión para casilla destino (a donde quiere mover su ficha seleccionada)

```
x = (GUIN_T_FROM_LE(event->y) / 33); // en x se guarda fila nueva
y = (GUIN_T_FROM_LE(event->x) / 33); // en y se guarda columna nueva
```

en estas dos variable “x” e “y” se guardan las nuevas coordenadas (coordenadas destino) después de hacer otro click en el tablero.

Ahora al tener estos dos datos importantes, coordenadas actuales y coordenadas destino, de acuerdo a la elección de turno y color se va revisando en la sentencia correspondiente los datos, si se ha seleccionado bien la ficha, si se ha seleccionado una casilla vacía para mover dicha ficha, si se ha generado un movimiento invalido, en este caso de movimiento invalido te muestra un mensaje y permite des-seleccionar la ficha y seleccionar otra ficha.

Se hace también una revisión constante de la función `revisarMovimientosPosibles (&x1, &x2, &z1, &z2, &puerta, &puerta2, &puerta3, &bandera, tablero);`, esta función hace que los movimientos sean correctos, por ejemplo hacer solamente un movimiento a una casilla adyacente, también revisa si es posible realizar saltos múltiples (todo esto se habilitan con variables, dependiendo del caso se habilita o se cierra).

Y para el movimiento solamente se hace intercambio de contenidos de la matriz, por ejemplo coordenadas actuales que contenía la ficha “N” o “B” se le asigna lo que contenía coordenadas destino, y a este se le asigna lo que contenía coord actuales.

Para la computadora, se usa una inteligencia para que pueda hacer su jugada, se seleccionan fichas que están mas adelante y va moviendo dichas fichas en dirección “diagonal” siempre apuntando al campamento enemigo, teniendo también la posibilidad de hacer saltos simples encima de la ficha propia o ficha enemiga, las fichas van avanzando y contando los pasos a la vez, cuando se llega a una cantidad “x” de pasos la ficha para, ya estando ubicada en una de las casillas del campamento enemigo, esto se repite mientras se van colocando de forma “triangular”.

Luego de cada turno se revisan estas dos funciones:

```
condicionQuienGana (tablero, &cont, &bandera5);
revisión_ganador(); //revisa esta funcion y si se cumple, muestra la ventana de
"VICTORIA"
```

si es que hay un ganador te muestra en pantalla una ventana de “VICTORIA” con el nombre del ganador.

Funciones importantes.

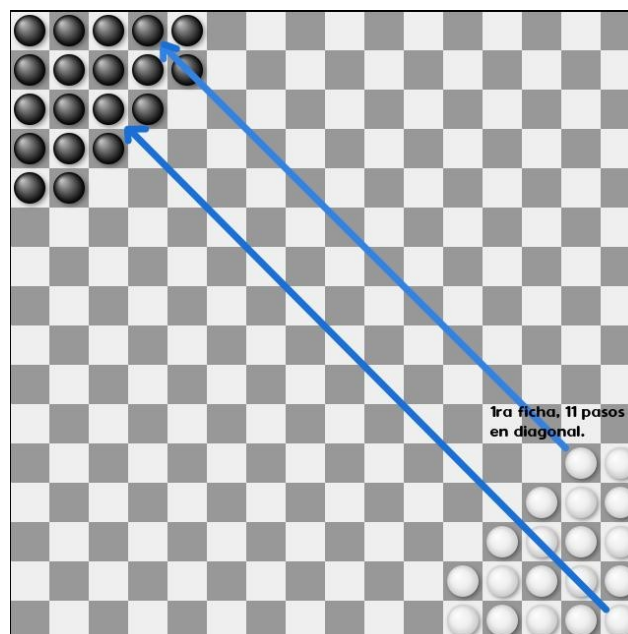
- **iniciartablero (tablero):** Se cargan todos los datos “N”, “B” y “.” en la matriz.

- **revisarMovimientosPosibles (&x1, &x2, &z1, &z2, &puerta, &puerta2, &puerta3, &bandera, tablero):** Revisa que los movimientos sean correctos, también revisa si es posible realizar saltos mutiples.

- **juegaMaquinaIA (tablero, &fila_actualM, &column_actualM, &fila_nuevaM, &column_nuevaM, &FichaColor, &reinicio, &aux2):** Inteligencia que usa el oponente para realizar su jugada.
- **solucionMov2fichas():** Parte de la IA, soluciona el error de mover dos fichas a la vez.
- **condicionQuienGana (tablero, &cont, &bandera5):** Revisa constantemente despues de cada turno, si hay un ganador a la variable “bandera5” se le asigna un valor.
- **revison_ganador():** Se revisa el valor asignado a “bandera5” para ver quien es el ganador y mostrar la ventana de “VICTORIA”.
- **guardarEstadisticas():** Se guardan los datos estadísticos del jugador.
- **actualizarEstadisticas(jugador.nombre, &jugador.partidas_jugadas, &jugador.partidas_ganadas, &jugador.partidas_perdidas):** Actualiza los datos estadísticos si es que lo ha jugado otro jugador, agregando su nombre y demás datos.

2.Descripción y justificación de la inteligencia/estrategia utilizada.

Esta inteligencia mueve primeramente las fichas que están mas adelante, las fichas que están mas cerca del campamento opuesto, con cada movimiento de ficha se cuenta la cantidad de pasos, hay una cantidad de pasos fija para las fichas, al hacer movimientos simples hay un contador de pasos que incrementa de uno en uno (por cada casilla visitada), si la ficha realiza un salto el contador aumenta en 2 (pasó 2 casillas), luego, al llegar a su paso limite la sentencia “if” ya no se cumplirá y dejara a la ficha en una casilla del campamento opuesto y hará lo mismo con la sgte ficha.



En caso de que la ficha se tranque encontrando dos fichas contrarias en la misma diagonal (no podrá saltar dos fichas contrarias), se usa una bandera que se le asigna un valor, con esta bandera permite revisar las otras fichas y mover una de esas mientras, igualmente cada vez que entra en la función de la IA, se revisa constantemente el estado de la ficha trancada, si ya puede hacer salto o no, en caso de que ya pueda esta ficha reanudará su movimiento hasta llegar al campamento opuesto, si es que no queda trancado de vuelta, si pasa de vuelta esto, se moverá la otra ficha que ya se había movido antes, así sucesivamente.

Justificación: Optamos por este criterio para la inteligencia, esto, por el tiempo y varios problemas que tuvimos con la inteligencia que realmente queríamos implementar, que era usando una matriz de peso para que realice una jugada mas interesante, pero, igualmente con la inteligencia que se eligió no ha generado problemas, hace jugadas validas hasta tratar de ganar.

3.Puntos fuertes y débiles del programa.

Puntos fuertes.

- Uno de los puntos fuertes del programa son las configuraciones/elecciones, las ventanas de configuraciones son bastantes claros, se configura correctamente, en caso de olvidarse y no hacer todas las elecciones pedidas, este mostrará un mensaje de aviso y no te permitirá avanzar hasta colocar todos los datos requeridos.
- Botón de “JUEGO NUEVO” inicializa perfectamente todo el juego.
- Estadísticas, carga perfectamente los datos de cada jugador ordenando de forma decreciente de acuerdo a la cantidad de partidas ganadas por cada jugador (persiste los datos después de cerrar el programa).

Puntos débiles.

- No posee la opción de “Computadora vs. Computadora”, en la interfaz se puede ver esta opción, pero no funciona, no se pudo llegar a implementar esta parte del juego.
- La inteligencia, a pesar de que hace jugadas validas y correctas, no era lo que teníamos pensado usar, la inteligencia que realmente queríamos implementar era mucho mas corto y eficaz, pero por problemas que se nos presentaba y el tiempo, decidimos optar por esta estrategia para la inteligencia de la computadora.

4. Dificultades encontradas durante el trabajo.

La mayor dificultad para nosotros fue la de implementar la opción de Computadora vs. Computadora, para empezar no teníamos muy claro lo que se tenia que hacer, se hizo el intento, pero igualmente no se llegó a nada, esto por el tema de como se tenia que hacer, respetando las especificaciones del trabajo (Anexo B), entonces, por el tiempo y por los problemas que se presentaban también en otras partes del código, decidimos dejar de intentar y priorizamos los otros problemas.

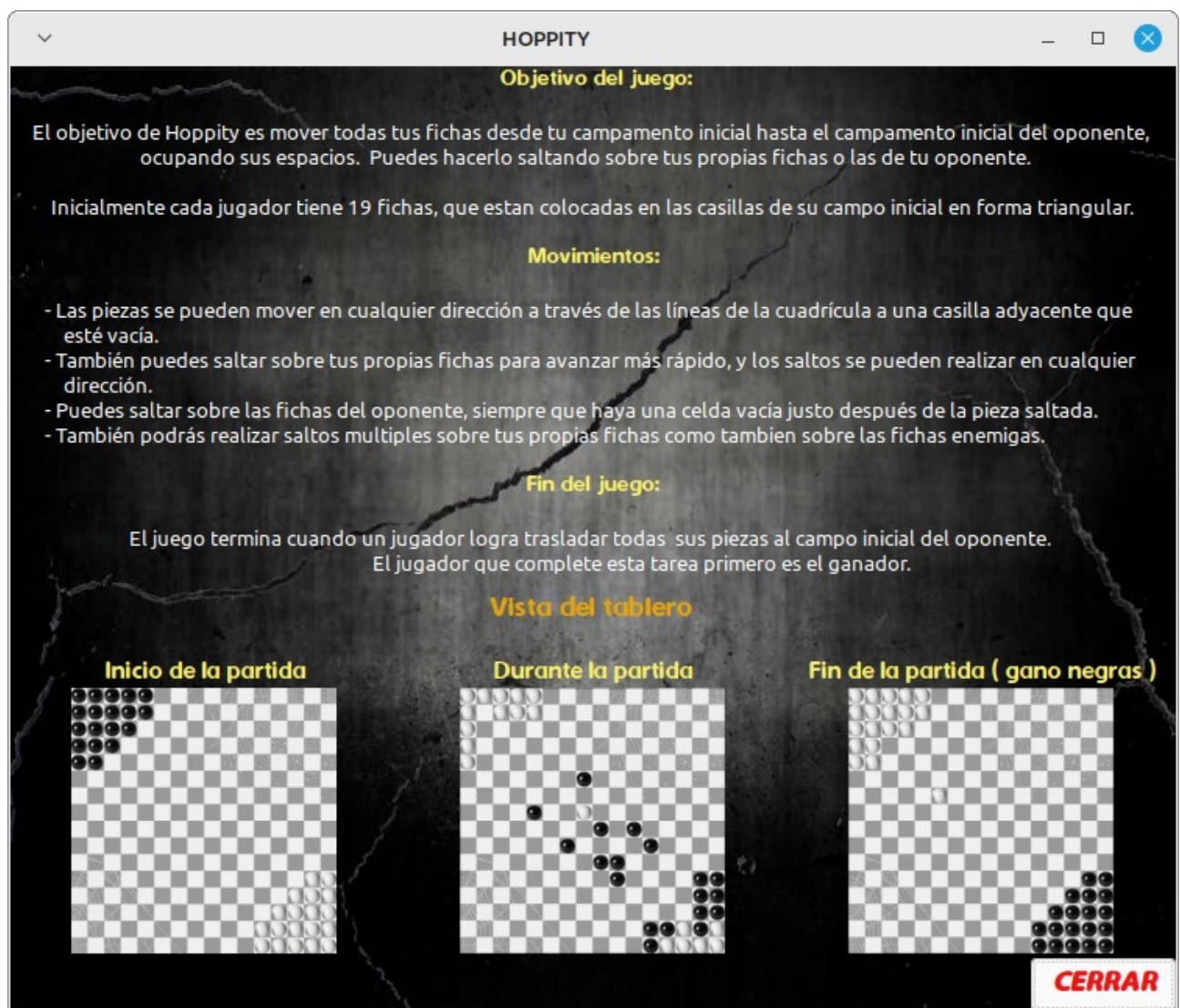
5. Guía de utilización del programa



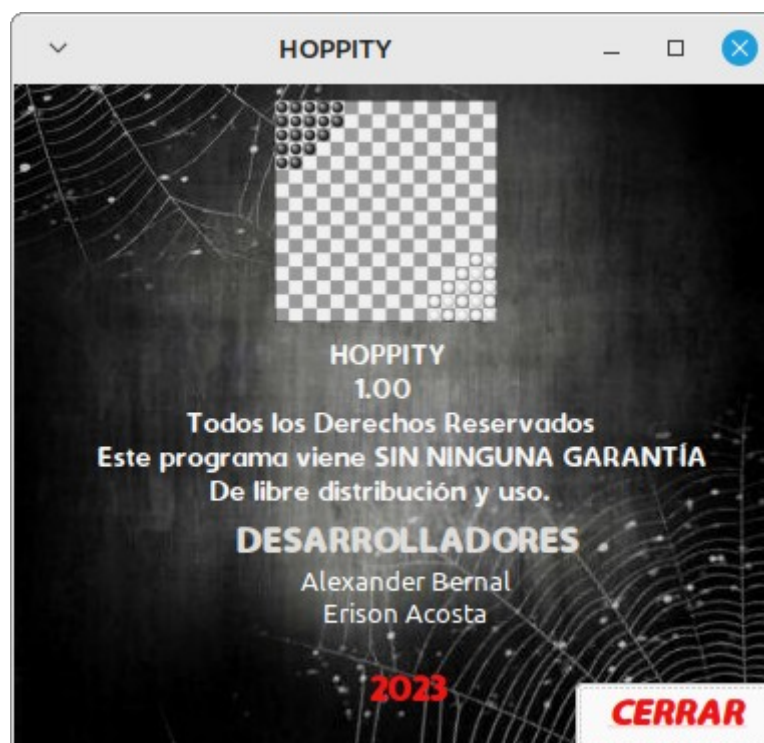
Al iniciar el juego aparece esta ventana con esas 4 opciones.

JUGAR: Al apretar “JUGAR” te muestra una ventana de “ELEGIR MODO DE JUEGO”

INSTRUCCIONES: Al apretar “INSTRUCCIONES” te muestra una ventana con una información del objetivo del juego y como hacer movimientos/jugadas.



CRÉDITOS: Al apretar “CRÉDITOS” te muestra una ventana con una información acerca del juego



SALIR : Sale del juego (Cierra la ventana)

Estando en la ventana de “ELEGIR MODO DE JUEGO”, puede seleccionar la opción “**HUMANO VS. COMPUTADORA**” o “**COMPUTADORA VS. COMPUTADORA**” o “**ATRÁS**” para volver a la ventana anterior.



Si apretó “**HUMANO VS. COMPUTADORA**” te mostrará la siguiente ventana



Acá puede seleccionar la opción “**ALEATORIO**”, para que las elecciones de turnos y color de ficha sean totalmente aleatorio o puede seleccionar la opción “**ELEGIR**”, para poder elegir a gusto los turno y color de ficha.

Si aprieta “**ATRÁS**” vuelve a la ventana de “ELEGIR MODO DE JUEGO”.

Si apretó “**ELEGIR**” le aparecerá la siguiente ventana.

The screenshot shows a window titled "HOPPITY" with a dark background and a grid of options. The window is divided into two main columns: "ELIGA TURNO" (Choose Turn) and "COLOR DE FICHA" (Choose Piece Color). Under "ELIGA TURNO", there are three buttons: "HUMANO" (Human), "COMPUTADORA" (Computer), and "ALEATORIO" (Random). Under "COLOR DE FICHA", there are two buttons: "HUMANO" (Human) and "COMPUTADORA" (Computer). Below these, there are two text input fields: "NOMBRE DEL JUGADOR 1 (ficha negra)" (Player 1 Name (black piece)) and "NOMBRE DEL JUGADOR 2 (ficha blanca)" (Player 2 Name (white piece)). The first field contains "Iron Man" and the second contains "Computadora". At the bottom, there are two buttons: "ATRÁS" (Back) and "LISTO" (Ready). Below the buttons, there is a summary section with labels and values: "Modo de juego:" (Game Mode) is "Humano vs. Computadora", "Turno:" (Turn) is "Humano", and "Color de ficha:" (Piece Color) is "Negra".

HOPPITY	
ELIGA TURNO	COLOR DE FICHA
HUMANO	HUMANO
COMPUTADORA	COMPUTADORA
ALEATORIO	ALEATORIO
NOMBRE DEL JUGADOR 1 (ficha negra)	Iron Man
NOMBRE DEL JUGADOR 2 (ficha blanca)	Computadora
ATRÁS	LISTO
Modo de juego:	Humano vs. Computadora
Turno:	Humano
Color de ficha:	Negra

En “ELIGE TURNO”

HUMANO: Si aprieta esta opción, el turno será del humano (empezará el humano)

COMPUTADORA: Si aprieta esta opción, el turno será de la computadora

ALEATORIO: La elección de turno sera totalmente al azar

En “COLOR DE FICHA”

Hay dos fichas, blanca y negra, usted puede seleccionar el color que mas le guste para jugar

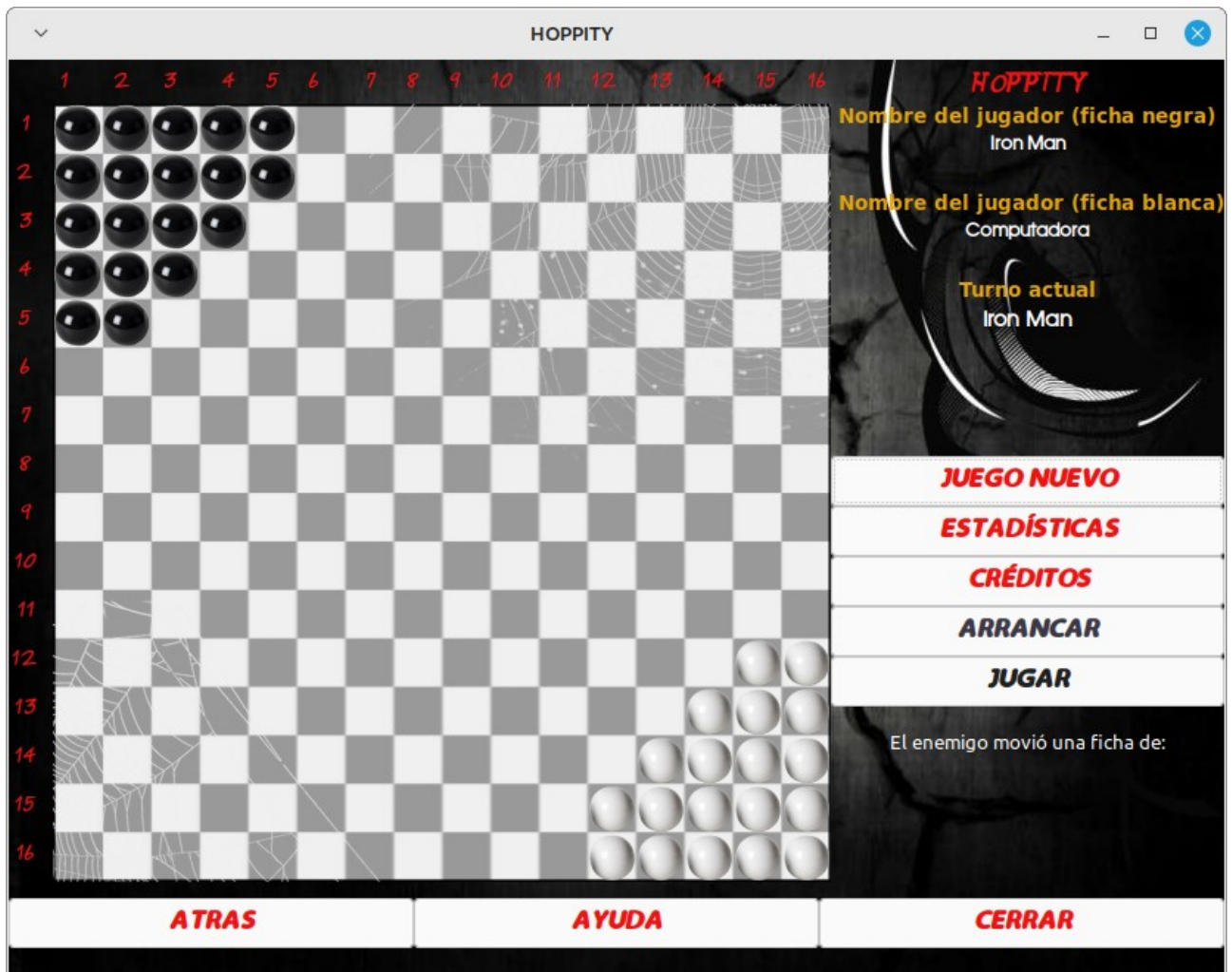
ALEATORIO: La elección de color de ficha es totalmente al azar

Puede ingresar los nombres, para los dos jugadores tanto para ficha negra como para ficha blancas.

Usted si eligió jugar con ficha negra, ingrese el su nombre en “NOMBRE DEL JUGADOR 1”, si eligió ficha blanca, ingrese su nombre en “NOMBRE DEL JUGADOR 2”.

Si aprieta “ATRÁS” vuelve a la ventana anterior de “COMO QUIERE INICIAR EL JUEGO”

Luego de hacer todas las elecciones, puede apretar “**LISTO**” para que le muestre la ventana en donde se podrá apreciar el tablero ya con las configuraciones hechas.



JUEGO NUEVO: Esto es para inicializar una nueva partida.

ESTADÍSTICAS: Muestra en una ventana el registro histórico de los jugadores que jugaron el juego.

ARRANCAR: Este solo se usa una vez, se usa en caso de que el turno sea de la computadora, apriete una vez para que la computadora haga su movimiento.

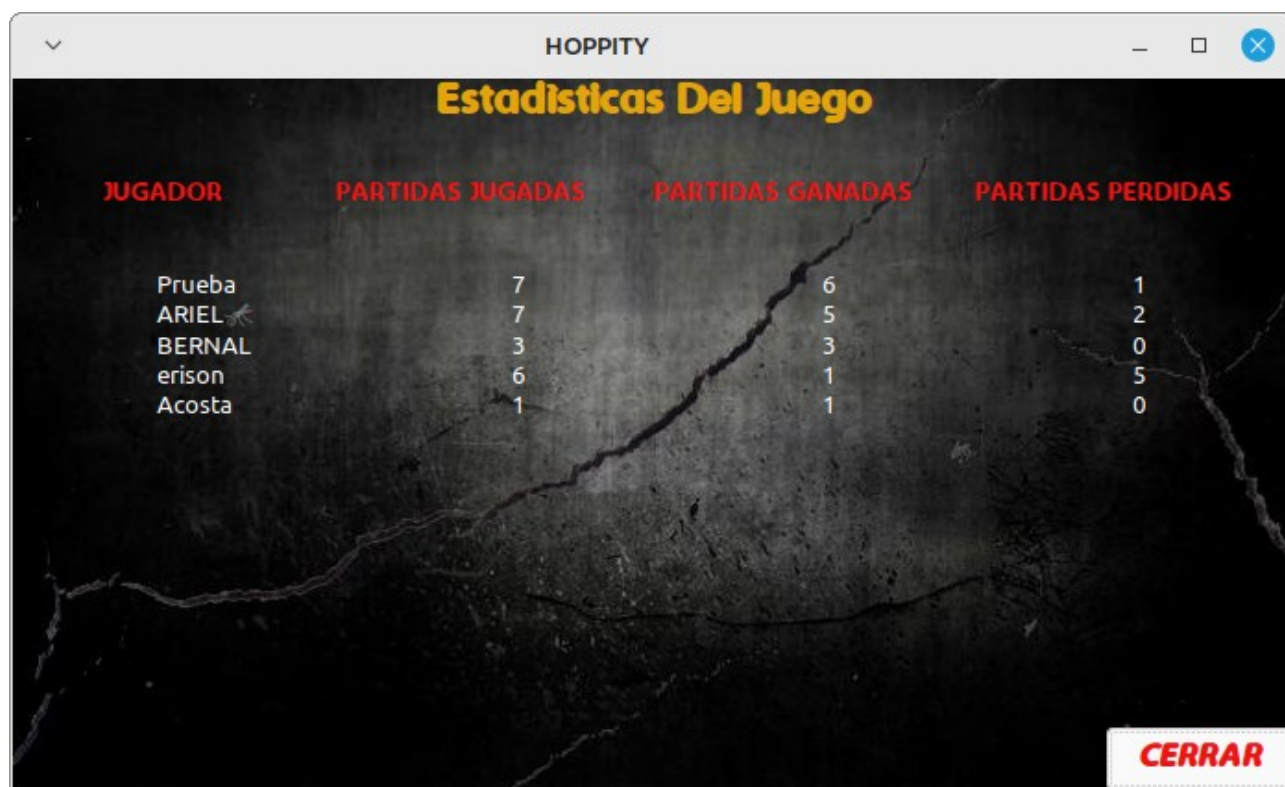
JUGAR: Con esto se van cambiando los turnos, usted hace un movimiento, luego debe apretar “JUGAR” para pasar el turno a su oponente y haga su jugada.

AYUDA: Este muestra una ventana con una información útil, para el uso del botón “ARRANCAR”

CERRAR: Si quiere cerrar totalmente el juego.

ATRÁS: Si quiere volver a la ventana anterior.

Muestra de los datos estadísticos guardados:



JUGADOR	PARTIDAS JUGADAS	PARTIDAS GANADAS	PARTIDAS PERDIDAS
Prueba	7	6	1
ARIEL	7	5	2
BERNAL	3	3	0
erison	6	1	5
Acosta	1	1	0

CERRAR