



TRABAJO INFORMATICA INDUSTRIAL Y DE COMUNICACIONES

AJEDREZ

MARIO GÓMEZ SÁNCHEZ-CELEMÍN - 55887

ÁLVARO GÓMEZ AGUDO - 55881

PEDRO ALVAREZ MONTEAGUDO - 55717

IÑIGO CASTELLS CASTRO - 55790

IKER CASTIELLA AGUIRREZABALA - 55791

GRUPO A302

FECHA: 26/05/2023

Índice

1.	¿QUÉ ES EL AJEDREZ?	3
2.	HISTORIA	3
3.	AJEDREZ CONTRA LA MAQUINA	4
4.	REGLAS	4
4.1	POSICION INICIAL DE LAS PIEZAS	4
4.2	MOVIMIENTOS	5
4.3	MOVIMIENTOS ESPECIALES	7
4.3.1	<i>Captura al paso</i>	7
4.3.2	<i>Promoción o coronación del peón</i>	7
4.3.3	<i>El enroque</i>	8
5.	DESARROLLO DEL CÓDIGO	9
5.1	ESTRUCTURA DE CLASES	10
5.2	CLASES PRINCIPALES	10
5.2.1	<i>Principal</i>	10
5.2.2	<i>Coordinador</i>	11
5.3	CLASES INTERMEDIAS	13
5.3.1	<i>Ajedrez</i>	13
5.3.2	<i>Menuinicial</i>	14
5.3.3	<i>Tablero</i>	15
5.4	PIEZAS	17
5.5	CLASES AUXILIARES	19
5.5.1	<i>ColorRGB</i>	19
5.5.2	<i>Casilla</i>	19
5.5.3	<i>Vector</i>	20
6.	RECURSOS AUXILIARES	20
6.1	SUPERPIEZA	20
6.2	SONIDOS AMBIENTALES E ITERACIÓN DEL RATON	21
7.	TERMINACIÓN DEL JUEGO	21
8.	LINEAS DE TRABAJO FUTURAS	21

1. ¿QUÉ ES EL AJEDREZ?

El ajedrez es un juego de estrategia considerado deporte mental en el que se enfrentan dos jugadores, cada uno de los cuales tiene 16 piezas de diversos valores. Un rey, una dama, dos alfiles, dos caballos, dos torres y ocho peones. Se pueden mover siguiendo ciertas reglas sobre un tablero cuadrado de 8x8 dividido en cuadrados blancos y negros, que constituyen las 64 posiciones posibles.

El objetivo final del juego consiste en “derrocar al rey” del oponente. Esto se hace amenazando la casilla que ocupa el rey con alguna de las piezas propias sin que el otro jugador pueda protegerlo interponiendo una pieza entre su rey y la pieza que lo amenaza, mover su rey a una casilla libre o capturar a la pieza que lo está amenazando. Como resultado se da el jaque mate y el fin de la partida.



Ilustración 1: Partida ajedrez

2. HISTORIA

Para encontrar la forma más similar al ajedrez actual tenemos que remitirnos a un legendario juego que surgió en el norte de India en el siglo VI. Conocido como chaturanga.

El chaturanga se jugaba entre cuatro personas en un tablero de 64 casillas, todas del mismo color. Los bandos norte y oeste eran aliados y luchaban contra los bandos sur y este. Sus piezas eran el rajá (de movimiento similar al del rey actual), el consejero (equivalente a la reina), el elefante, el caballo (de idéntico movimiento al nuestro), el carro y los soldados (que iban hacia delante paso a paso, como los peones).

En el año 600, un juego muy parecido al ajedrez, derivado del chaturanga de la India se propagó por Persia, alcanzó el mundo árabe y España, desde donde se introdujo en Europa.

El ajedrez adquirió las formas y reglas actuales en el siglo XV gracias a la invención de la imprenta. Posteriormente, en 1886 se celebró el primer campeonato del mundo de ajedrez. Y en el 1924 se creó la Federación Internacional de Ajedrez (FIDE), que rige desde entonces el mundo del ajedrez. [1]

3. AJEDREZ CONTRA LA MAQUINA

A principios del siglo XX, el español Leonardo Torres Quevedo creó el ajedrez mágico. Se trataba de un tablero de metal electromagnético con tres piezas. El autómata jugaba blancas sólo con el rey y una torre, mientras su contrincante jugaba con el rey negro, pudiendo desplazarlo como quisiera. Lo que hace que sus movimientos determinen los de las fichas blancas, que siempre ganan.



Ilustración 2: Ajedrez automático de Leonardo Torres Quevedo

Más tarde, en 1956, se desarrolló el MANIAC, el primer ordenador capaz de derrotar a un humano en una partida de ajedrez.

Durante mucho tiempo, en los años 1970 y 1980 se mantuvo abierta la cuestión de si cualquier programa de ajedrez sería capaz de derrotar a la experiencia de los mejores seres humanos. En 1968, el Maestro Internacional David Levy hizo una famosa apuesta que ningún ordenador de ajedrez sería capaz de vencerlo dentro de diez años. Ganó su apuesta en 1978 al vencer a ajedrez 4,7 (el equipo más fuerte en el momento), pero reconoció entonces que no pasaría mucho tiempo antes de que él fuera superado.

Finalmente, en mayo de 1997 el programa Deep Blue, desarrollado por IBM derrotó a Kaspárov, el vigente campeón mundial de ajedrez. [2]

4. REGLAS

4.1 POSICION INICIAL DE LAS PIEZAS

Al principio del juego, las piezas deben disponerse como se muestra en la siguiente imagen:

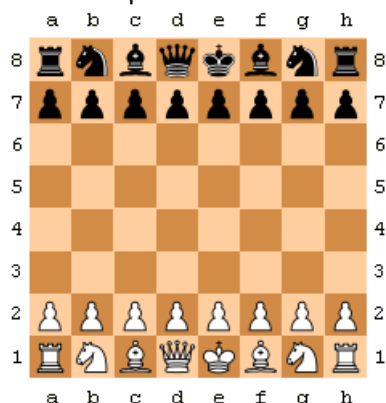


Ilustración 3: Posición inicial de las piezas

4.2 MOVIMIENTOS

La torre se mueve en direcciones ortogonales, es decir, por las filas (horizontales) y columnas (verticales).

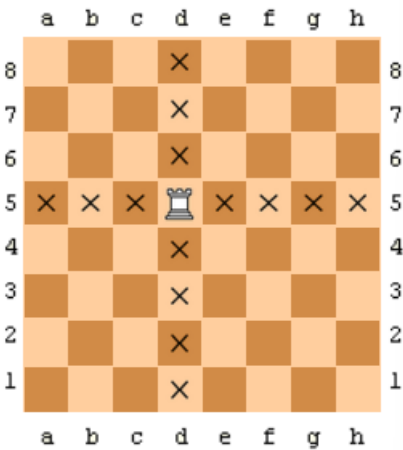


Ilustración 4: Movimiento de la torre

El alfil se mueve en direcciones diagonales, es decir, en la dirección de las casillas del mismo color.

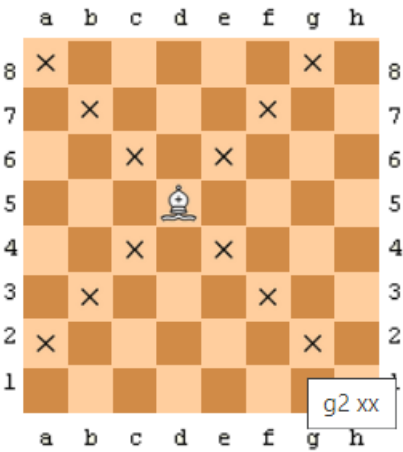


Ilustración 5: Movimiento del alfil

La dama puede moverse en horizontal, vertical o diagonal, pero solo en una dirección en cada turno.

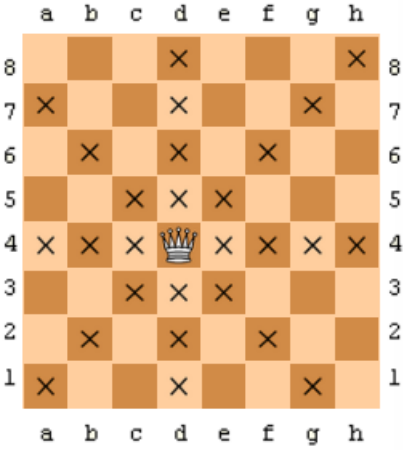


Ilustración 6: Movimiento de la reina

El rey puede moverse en todas las direcciones un solo escaque cada vez, mientras el movimiento no sea para un escaque amenazado por una pieza adversaria. Un rey no puede dar jaque a otro rey.

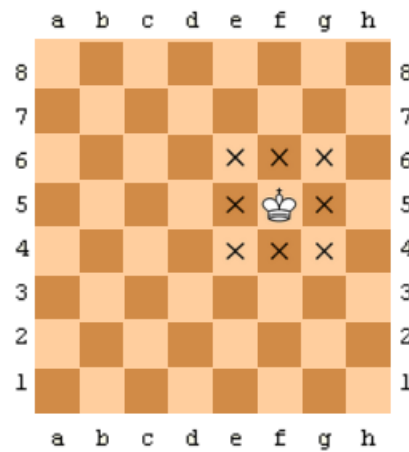


Ilustración 7: Movimiento del rey

El caballo puede moverse a cualquiera de los escaques más próximos que no sean de la misma fila, columna o diagonal. Normalmente se dice que mueve en forma de L, esto es, mueve dos casillas en horizontal o vertical y después una casilla vertical u horizontal, o viceversa. El caballo puede saltar sobre cualquier pieza suya o del adversario. La captura ocurre cuando una pieza del adversario se encuentra en la casilla de destino del movimiento realizado por el caballo.

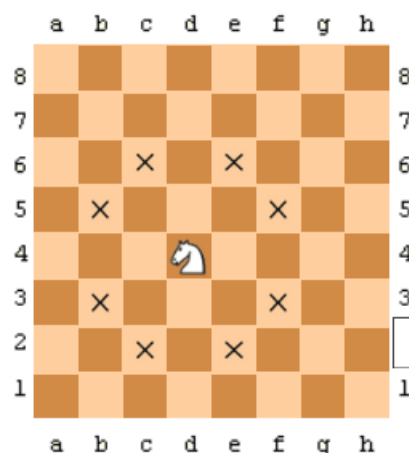


Ilustración 8: Movimiento del caballo

El peón tiene las reglas de movimiento más complejas del ajedrez:

Un peón se mueve hacia adelante una casilla, si este escaque está vacío. Si el peón todavía no se ha movido, en su primer movimiento también tiene la opción de mover dos casillas hacia adelante, siempre que ambas casillas estén vacías. Los peones no se pueden mover hacia atrás. Los peones son las únicas piezas que capturan de manera diferente de cómo se mueven. Un peón puede capturar una pieza adversaria en cualquiera de las casillas diagonales en frente del peón (pero no se puede mover a esas casillas si están vacías).

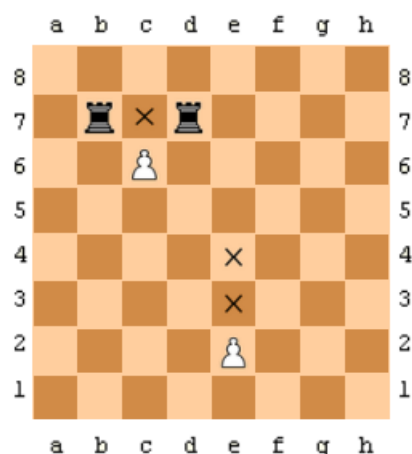


Ilustración 9: Movimiento del peón

El peón también participa en dos movimientos especiales: captura al paso y promoción.

4.3 MOVIMIENTOS ESPECIALES

Los movimientos especiales en ajedrez son jugadas legales que, aunque no forman parte de los movimientos básicos de las piezas, son parte de su repertorio y son fundamentales para generar dinamismo en las partidas. Existen 3, la captura en paso, la coronación del peón y el enroque. [3]

4.3.1 Captura al paso

Consiste en que, si un peón en su primer movimiento avanza dos casillas, puede ser capturado por un peón contrario colocado en la misma fila y columna adyacente. Este movimiento solo se puede realizar en la jugada siguiente a este movimiento.



Ilustración 10: Captura al paso

4.3.2 Promoción o coronación del peón

Según el libro de normas de la FIDE, cuando un peón alcanza la fila más alejada de su posición inicial (la octava para las blancas y la primera para las negras), debe cambiarse como parte del mismo movimiento en la misma casilla por una nueva reina, torre, alfil o caballo del mismo color.

4.3.3 El enroque

Es el movimiento del rey desde su casilla original hacia dos casillas a la torre, y luego esa torre se mueve a la casilla que el rey acaba de cruzar.

Para realizar este movimiento se debe cumplir que ni el rey ni la torre se hayan movido anteriormente, no haya piezas entre ellos, y que la casilla en la que se encuentra el rey, o por donde cruza no este atacado por una pieza rival.

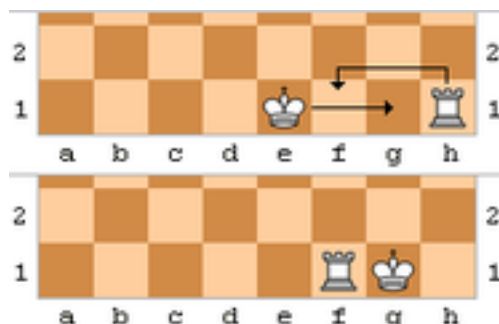


Ilustración 11: Enroque corto

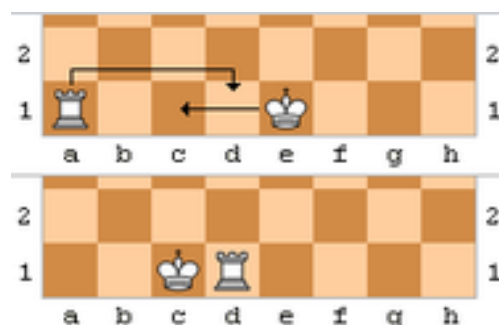


Ilustración 12: Enroque largo

5. DESARROLLO DEL CÓDIGO

Se ha desarrollado la modalidad de dos jugadores, por lo que el caso de uso se define brevemente como:

“Dos jugadores se reúnen y comienzan una partida. Se establecerá una secuencia de turnos, jugador1 fichas blancas y jugador2 fichas negras. Ambos jugadores realizarán los movimientos de las piezas alternativamente. La partida terminará cuando uno de los dos jugadores consiga un Jaque Mate.”

El diagrama UML que representa el caso de uso es el siguiente:

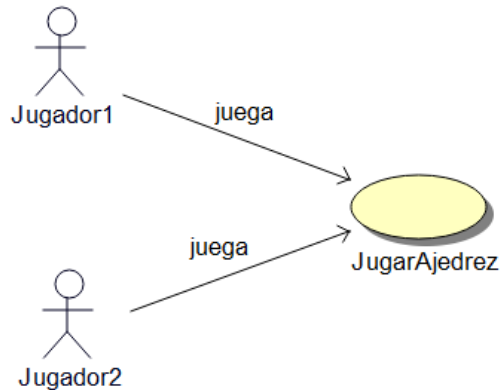


Ilustración 13: Caso de uso principal de la aplicación

En la primera versión el movimiento de las piezas consistía en desplazar la casilla utilizada como selector por el tablero. Una vez seleccionada la pieza y el destino se comparaba la posición con los posibles movimientos de dicha pieza.

Esta estructura generó numerosos problemas a la hora de extender el código e implementar nuevas funciones como los posibles movimientos de una pieza o el jaque mate.

En la versión final se mejoró el diseño notablemente, creando un tablero “tonto”, en el que se utiliza un puntero de tipo pieza a una matriz 8x8. En las casillas vacías el puntero apuntará a nullptr. Además, se incluye la comprobación para el movimiento dentro de la propia pieza pasando el tablero como argumento.

Esto permitió solucionar aspectos que en el primer diseño resultaban muy tediosos de manera sencilla.

5.1 ESTRUCTURA DE CLASES

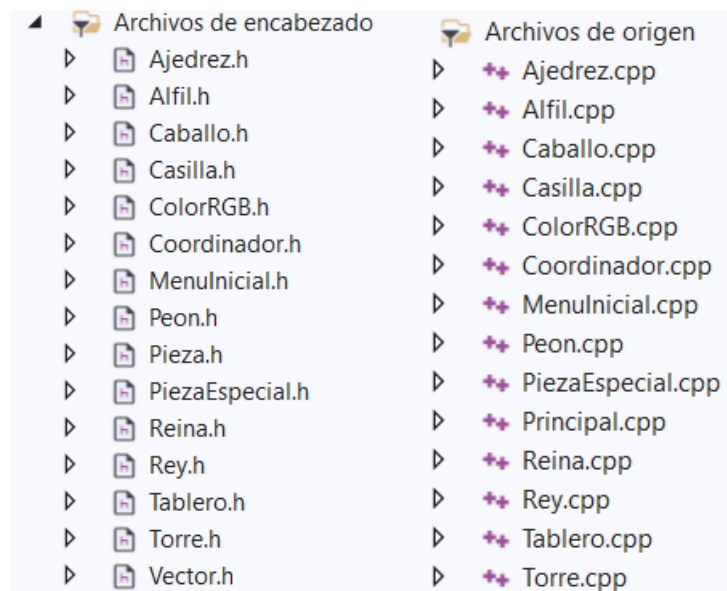


Ilustración 14: Archivos de origen y encabezado

Para facilitar la comprensión se han dividido las clases utilizadas en cuatro grandes grupos: clases principales, clases intermedias, clases auxiliares y Piezas.

Las clases principales son 'Principal' y 'Coordinador', estas serán las encargadas de ejecutar las funciones principales, la primera de ellas contiene el "main" y las funciones de dibujado y de interacción con las teclas y el ratón. La que la segunda de ellas contiene la máquina de estados con las transiciones entre estos.

Las clases intermedias son 'MenuInicial', 'Ajedrez' y 'Tablero' estas son las que contienen funciones básicas como la lógica del juego, la gestión de turnos o los equipos del menú.

Las clases auxiliares contienen los objetos que se utilizarán como atributos para la creación de nuevos objetos, 'ColorRGB', 'Casilla', 'Vector'...

Y por último las 'Piezas' que heredarán de una clase base.

De tal forma que podemos ver la estructura como una secuencia de clases que se van especificando y conteniéndose unas a otras.

Posteriormente se explicará más a fondo los métodos y atributos empleados en cada una de las clases.

5.2 CLASES PRINCIPALES

En este apartado se comentará más a fondo los atributos y métodos de cada una de las clases que componen este grupo.

5.2.1 Principal

La clase 'Principal' contiene las funciones principales de la librería "freeglut.h" que se utilizará para facilitar el dibujado, gestión del ratón y teclado.

5.2.2 Coordinador

La clase 'Coordinador' contiene la máquina de estados con las diferentes funcionalidades del juego: INICIO, MENU, PARTIDA, PAUSA, GANA_B, GANA_N y TABLAS.

A continuación, se muestra el diagrama DCD de dicha clase:

Coordinador
<ul style="list-style-type: none">- sobrecargaintro : int- x_ojo : float- y_ojo : float- z_ojo : float# estado : Estado# ajedrez : Ajedrez
<ul style="list-style-type: none">+ inicializa()+ dibuja()+ tecla(inout key : uchar)+ teclaEspecial(inout key : uchar)+ Raton(in button : int, in state : int, in x : int, in y : int)

Ilustración 15: Diagrama DCD clase 'Coordinador'

Al comenzar el juego aparece una ventana emergente en la que se explica el modo de juego y varios comandos que serán imprescindibles para poder utilizar todas las funcionalidades.

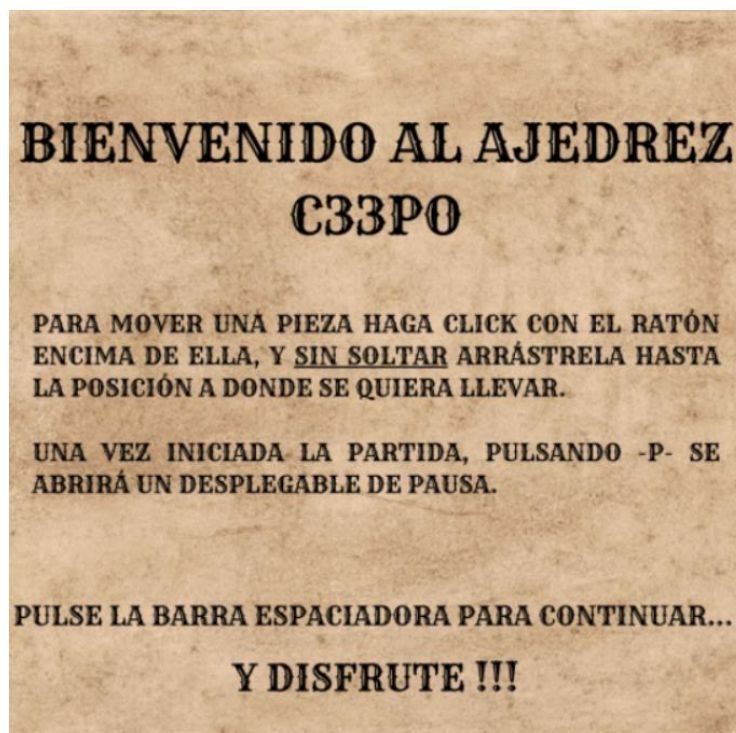


Ilustración 16: Pantalla de bienvenida

La función pausa es otro estado imprescindible de esta clase. Al pulsar la tecla 'p' el juego se detiene y se imposibilita la movilidad de las piezas.



Ilustración 17: Menú Pausa

Con el juego detenido se pueden implementar varias funciones más. En primer lugar, al pulsar la tecla 's' se saldrá del juego automáticamente. Por otra parte, con el teclado, se pueden acordar las tablas, pulsando 't', o rendirse y declarar que ha ganado la partida el jugador blancas, pulsando 'b', o el jugador negras pulsando 'n'.



Ilustración 18: Estados al pulsar una de las teclas citadas

5.3 CLASES INTERMEDIAS

Estas clases servirán como enlace entre las clases superiores y las piezas, controlarán que se seleccionan fichas correctas y que se muevan sobre el tablero.

5.3.1 Ajedrez

La función principal de la clase Ajedrez es la de ser el intermediario entre la clase principal y el tablero. Esta clase controla más específicamente la inicialización del programa, que cosas se dibujan (menú y tablero) y la implementación del ratón y teclado para avanzar en el juego.



Ilustración 19: Diagrama DCD 'Ajedrez'

El siguiente diagrama muestra la secuencia de funcionamiento de la selección y movimiento de las piezas a través del ratón.

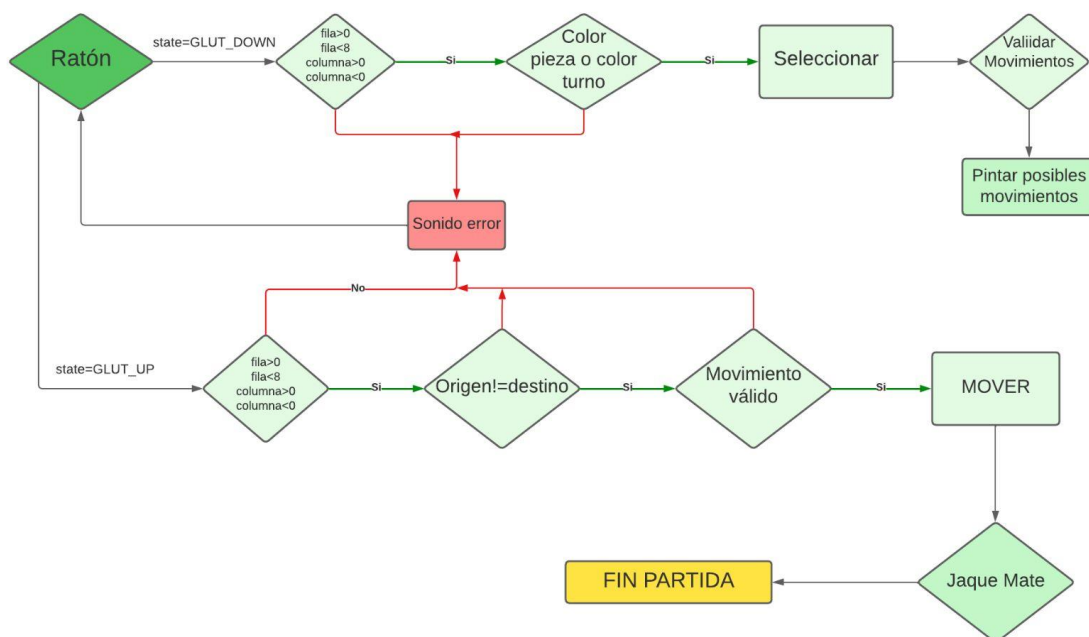


Ilustración 20: Secuencia de funcionamiento

5.3.2 Menuinicial

‘MenuInicial’ es una clase que genera una serie de pantallas emergentes al compilar el programa mediante la interacción con el teclado. Consta de varias imágenes, “menu1”, “menu2”, “etsidi”, “atletico” y “espana”.

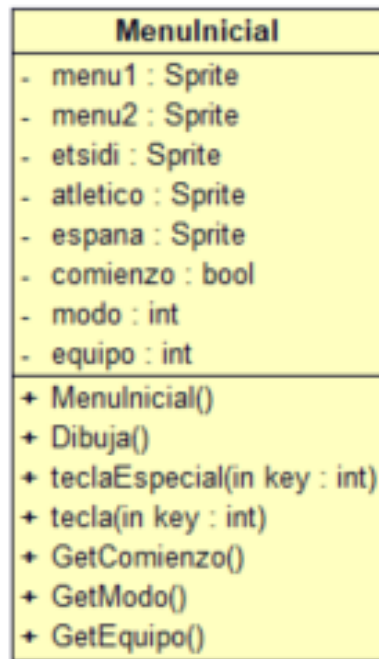


Ilustración 21: Diagrama DCD ‘MenuInicial’

En primer lugar, aparece “menu1”, una pantalla en la que usando las flechas del tablero el usuario decide comenzar una partida nueva o salir del juego.



Ilustración 22: Pantalla “menu1”

A continuación, se selecciona una de las opciones con la tecla ‘enter’, al pulsar sobre “SALIR DEL JUEGO” el programa se detiene. Por el contrario, al seleccionar “NUEVA PARTIDA” aparece un nuevo menú, “menu2”. En dicha ventana, se selecciona mediante las flechas del teclado un equipo.



Ilustración 23: Pantallas selección de equipos

Esta selección define los colores de las casillas del tablero según los colores característicos de cada uno de los equipos. Por último, tras elegir el equipo, se genera el tablero y las piezas en sus posiciones iniciales y comienza la partida.

5.3.3 Tablero

La clase ‘Tablero’ tiene como función definir toda la logística del juego. En ella se implementan los turnos (con sus respectivas imágenes), las casillas del tablero y las de movimientos posibles, y la “matriz” de las piezas.

Tablero
<ul style="list-style-type: none"> - casilla : Casilla - tablero : Pieza - NPieza : int + turno : int
<ul style="list-style-type: none"> + inicializa() + DibujaTablero(inout color1 : ColorRGB, inout color2 : ColorRGB) + EliminarContenido() + agregar(inout : Pieza, in i : int, in j : int, out : bool) + eliminar(in i : int, in j : int) + eliminar(inout p : Pieza) + mover(inout origen : Vector, inout destino : Vector) + DibujaPieza() + getTurno()

Ilustración 24: Diagrama DCD ‘Tablero’

Esta clase contiene los siguientes métodos:

- Constructores y destructores: Un constructor por defecto, otro constructor de copia (para el jaque mate) y un destructor.
- Función inicializa para establecer todas las piezas inicialmente como “nullptr”.
- Funciones para dibujar el tablero utilizando una matriz de casillas y elegir el color según el equipo seleccionado, y para dibujar el turno.

- Funciones para el manejo del vector de piezas mediante el empleo de arquitectura de “poo”. Estas funciones sirven para agregar y eliminar Piezas y tienen un proceso de seguridad para no apuntar a punteros que no existen. Además, cuenta con la coronación del peón.
- Funciones para el movimiento y dibujado de las piezas.
- Funciones de interfaz para obtener el tipo de pieza o el turno.
- Funciones para la comprobación de posibles movimientos (se especifican en cada pieza) y agregado de casillas para dichos movimientos.
- Funciones para la obtención de la posición del rey y comprobación del jaque.
- Además, contiene un operador “=” para la clonación del tablero (usado para el jaque mate).

Dentro de todos estos métodos cabe destacar la función de mover, pues es la más determinante para el óptimo funcionamiento del videojuego. Esta función será la encargada de gestionar el movimiento de las piezas. Controla el buen funcionamiento de las fichas y restringe el movimiento si se detecta Jaque.

```

196 //en esta funcion controlamos el movimiento de las piezas, así como si se produce jaque en algún momento, limitando el movimiento hasta que se des
197 bool Tablero::mover(Vector origen, Vector destino) {
198     if (origen.x < 0 || origen.y < 0 || origen.y >= 8 || origen.x >= 8) return false;
199     if (destino.x < 0 || destino.y < 0 || destino.y >= 8 || destino.x >= 8) return false;
200
201     if (tablero[origen.x][origen.y] == nullptr) return false;
202
203     Pieza* piezaMovida = tablero[origen.x][origen.y];
204     Pieza* quieroComer = nullptr;
205     quieroComer = tablero[destino.x][destino.y];
206
207
208     tablero[destino.x][destino.y] = tablero[origen.x][origen.y];
209     tablero[origen.x][origen.y] = nullptr; // IMPORTANTE -> poner el puntero a nullptr después de eliminar
210
211     // Verificar si se pone en jaque al propio rey
212     if (turno % 2 == 1) {
213         if (HayJaqueBlancas()) {
214             // Movimiento inválido, pone en jaque al rey blanco
215             tablero[origen.x][origen.y] = tablero[destino.x][destino.y];
216             tablero[destino.x][destino.y] = quieroComer;
217             ETSIDI::play("sonidos/error.wav");
218             return false;
219         }
220         else {
221             if (quieroComer) {
222                 ETSIDI::play("sonidos/PiezaComida.wav");
223                 delete quieroComer;
224                 NPiezas--;
225
226                 // Actualizar el turno
227                 turno++;
228                 return true;
229             }
230         }
231     }
232     else {
233         if (HayJaqueNegras()) {
234             // Movimiento inválido, pone en jaque al rey blanco
235             tablero[origen.x][origen.y] = tablero[destino.x][destino.y];
236             tablero[destino.x][destino.y] = quieroComer;
237             ETSIDI::play("sonidos/error.wav");
238             return false;
239         }
240         else {
241             if (quieroComer) {
242                 ETSIDI::play("sonidos/PiezaComida.wav");
243                 delete quieroComer;
244                 NPiezas--;
245                 // Actualizar el turno
246                 turno++;
247                 return true;
248             }
249         }
250     }
251 }
252
253 coronacionPeon();
254 if(quieroComer==nullptr) ETSIDI::play("sonidos/PiezaComida.wav");
255 // Actualizar el turno
256 turno++;
257 return true;
258

```

Ilustración 25: Función mover

5.4 PIEZAS

Todas las piezas se crearán por herencia desde una clase base llamada 'Pieza'.

Esta clase tendrá como atributos el color, y el tipo (ambas son de un tipo de datos creado con una enumeración), la imagen y el tamaño, todos ellos serán protegidos para permitir el acceso a todas las clases hijas.

Los métodos de la clase 'Pieza' se dividen entre funciones de interfaz, "getTipo" y "getColor" que se utilizarán para poder acceder a los atributos y métodos para la gestión de las piezas: "validarMovimiento" que será de tipo virtual para poder redefinir el movimiento de cada una de las piezas utilizando el polimorfismo.

En la ilustración --23-- se muestra el diagrama DCD de la clase Pieza y en la ilustración --24-- se puede apreciar el código de la definición de la clase implementado en el archivo 'Pieza.h'.

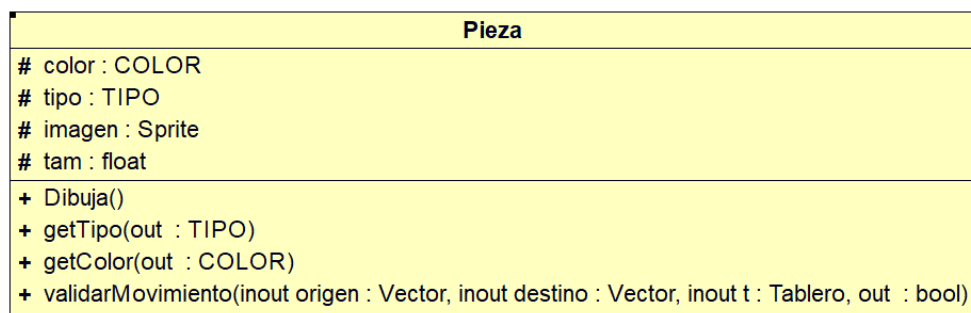


Ilustración 26: Diagrama DCD clase 'Pieza'

```

class Pieza
{
public:
    enum COLOR { Blanco, Negro };
    enum TIPO { PEON, REY, REINA, ALFIL, TORRE, CABALLO };
protected:
    COLOR color;
    TIPO tipo;
    const float tam = 1.0;
    Sprite imagen;

    //Constructor comun a las piezas
    Pieza(TIPO t, COLOR color, const char* fdibujo):tipo(t), color(color), imagen(fdibujo, 0, 0, tam, tam) {
        imagen.setCenter(0, 0);
    };

public:
    void Dibuja(){
        imagen.draw();
    }

    // FUNCIONES DE INTERFAZ
    TIPO getTipo() { return tipo;}
    COLOR getColor() { return color; }

    // Método para validar un movimiento de la pieza, se utiliza polimorfismo
    virtual bool validarMovimiento(Vector origen, Vector destino, Tablero& t) { return true; }
    virtual Pieza* clonar() { return new Pieza(*this); }
}
  
```

Ilustración 27: Definición clase 'Pieza'

Utilizando la clase base Pieza se han creado a través de la herencia las clases Peon, Alfil, Torre, Rey, Reina y Caballo.

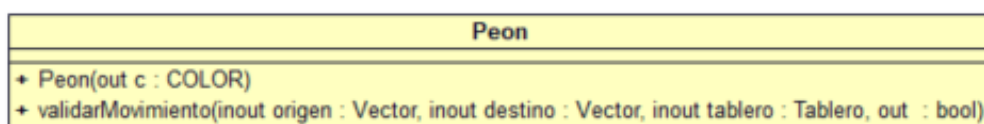


Ilustración 28: Diagrama DCD del 'Peon'



Ilustración 29: Diagrama DCD 'Alfil'

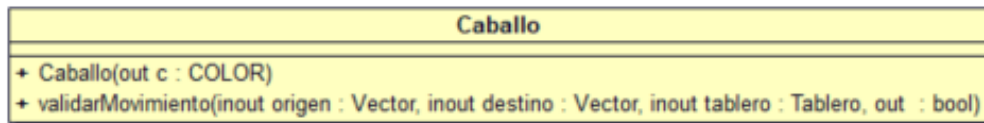


Ilustración 30: Diagrama DCD 'Caballo'

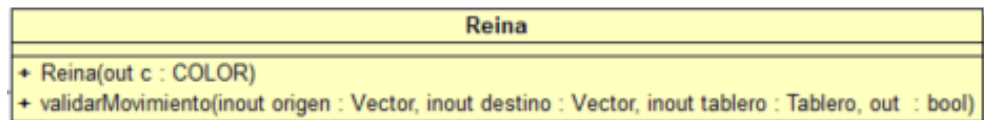


Ilustración 31: Diagrama DCD 'Reina'

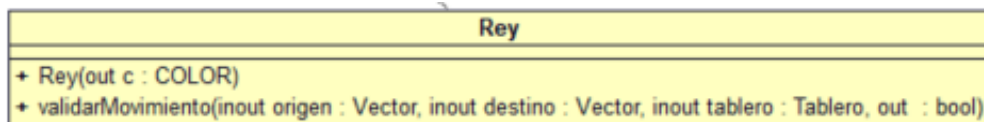


Ilustración 32: Diagrama DCD 'Rey'

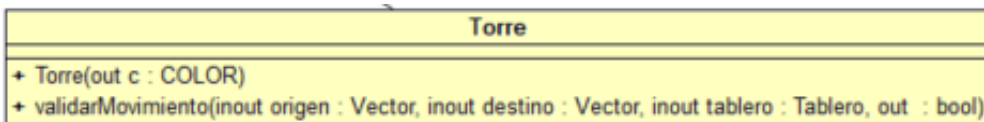


Ilustración 33: Diagrama DCD 'Torre'

Estas nuevas clases hijas solo contendrán únicamente su constructor que a su vez utilizará el constructor de Pieza como constructor delegante y el método “validarMovimiento” redefinido en función del movimiento de cada una de las piezas.

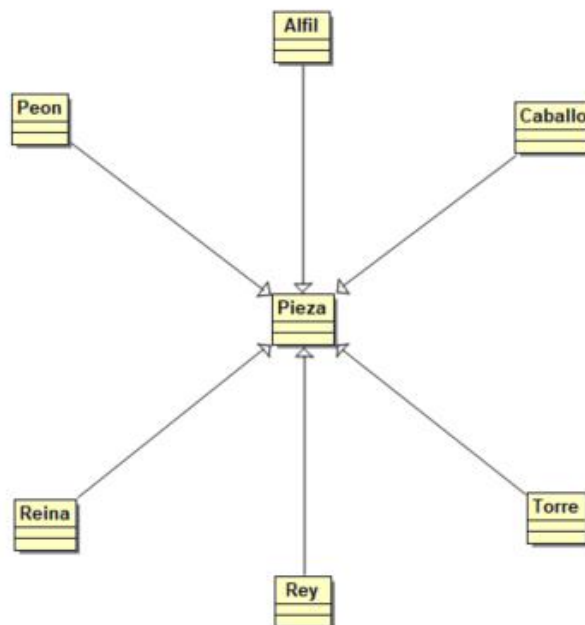


Ilustración 34: Diagrama DCD herencia

5.5 CLASES AUXILIARES

Se dividen en varias clases. ‘ColorRGB’, ‘Casilla’, ‘Vector’ y ‘Menuinicial’.

5.5.1 ColorRGB

‘ColorRGB’, clase que define y comprueba los colores de las casillas y las piezas.

Esta clase tiene atributos “char” privados para indicar el rojo, verde y azul. En la parte pública se encuentran funciones como el constructor, un “setColor”, y una función de seguridad que comprueba que los valores introducidos son correctos. Además, cuenta con funciones de interfaz que devuelven el valor de uno solo de los colores.

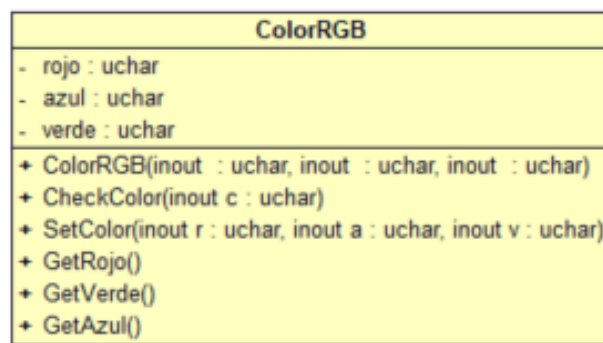


Ilustración 35: Diagrama DCD ‘ColorRGB’

5.5.2 Casilla

La clase ‘Casilla’ sirve para (como bien indica su nombre) construir una casilla con su respectivo tamaño, color (‘ColorRGB’) y posición (‘Vector’). Hay constructores para inicializar la casilla con diferentes valores. Además, cuenta con funciones para dibujar la casilla en una posición específica, establecer y obtener la posición, y cambiar el color de la casilla.

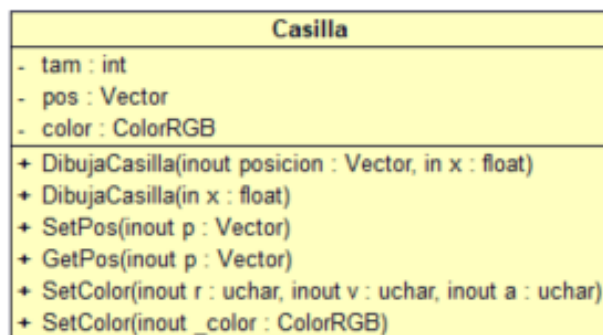


Ilustración 36: Diagrama DCD ‘Casilla’

5.5.3 Vector

Esta clase representa un vector bidimensional de parámetros x e y, y tiene un constructor por defecto que inicializa ambas variables a 0. Además, posee 4 sobrecargas de operadores para optimizar el código: "=", "!=", "<" y ">=".

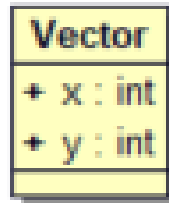


Ilustración 37: Diagrama DCD 'Vector'

6. RECURSOS AUXILIARES

Además de las reglas clásicas del juego del ajedrez se han añadido varios elementos adicionales entre los que destacan una pieza especial, inclusión de sonidos ambientales y circunstanciales, e interacción con ratón y teclado.

6.1 SUPERPIEZA

En el programa al ejecutarse la coronación del peón, éste se sustituye por una nueva pieza llamada "SuperPieza", la cual posee los movimientos de la reina y del caballo combinado y se representa con el logo de la ETSIDI.

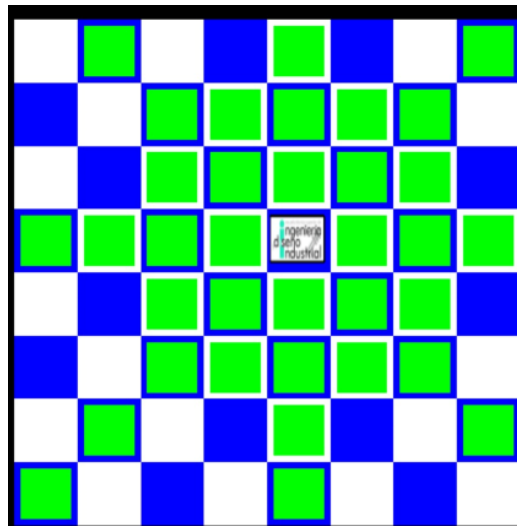


Ilustración 38: Movimientos posibles de la "SuperPieza"

6.2 SONIDOS AMBIENTALES E ITERACIÓN DEL RATON

Se han añadido varios sonidos: una melodía durante la partida, y dos sonidos circunstanciales. El primero suena cuando se comete un error, ya sea porque se selecciona una pieza del color que no posee el turno tanto como cuando se desplaza una pieza a un destino el cual no tiene permiso. Y el segundo cuando capturas una pieza rival. Por último, cuando uno de los jugadores gana la partida se reproduce un sonido de victoria.

Por otra parte, se ha desarrollado un sistema de selección de las piezas controlado por el ratón. Al hacer click izquierdo se selecciona la ficha que se quiere mover y se arrastra hasta la posición a donde se quiere llevar.

7. TERMINACIÓN DEL JUEGO

Para la terminación del juego se ha implementado el jaque mate, pero por problemas con la gestión de los recursos se ha decidido que el fin del juego sea detectado por el usuario. Se pueden ofrecer tablas o recibir jaque mate, ya que no se podrá mover ninguna pieza. En este momento el usuario deberá pulsar la tecla "1" o "2" en función del color del jugador ganador. Esta solución tiene como objetivo evitar alternativas como comerse al Rey lo que nos alejaría del ajedrez tradicional, que termina cuando uno de los dos reyes se encuentra en Jaque Mate.

Se ha añadido al repositorio de GitHub los archivos finales del Tablero y Ajedrez que contendrán la implementación del jaque mate.

8. LINEAS DE TRABAJO FUTURAS

- Solucionar problema al fallar un movimiento
- Mejorar y optimizar el diseño final ya que el código actual requiere mucha memoria y da lugar a diversos problemas como la saturación del alfil.
- Inclusión de una IA para el modo de juego maquina vs jugador.
- Creación de un ranking con puntuaciones de las distintas partidas.
- Adición de un cronometro y contador de movimientos.

9. BIBLIOGRAFIA

[1] A. G.M, La historia del ajedrez, un juego milenario, National Geographic, 2022.

[2] Historia del ajedrez, Curiosfera, 2020.

[3] D. Muñoz, Movimientos especiales del ajedrez, 2019.