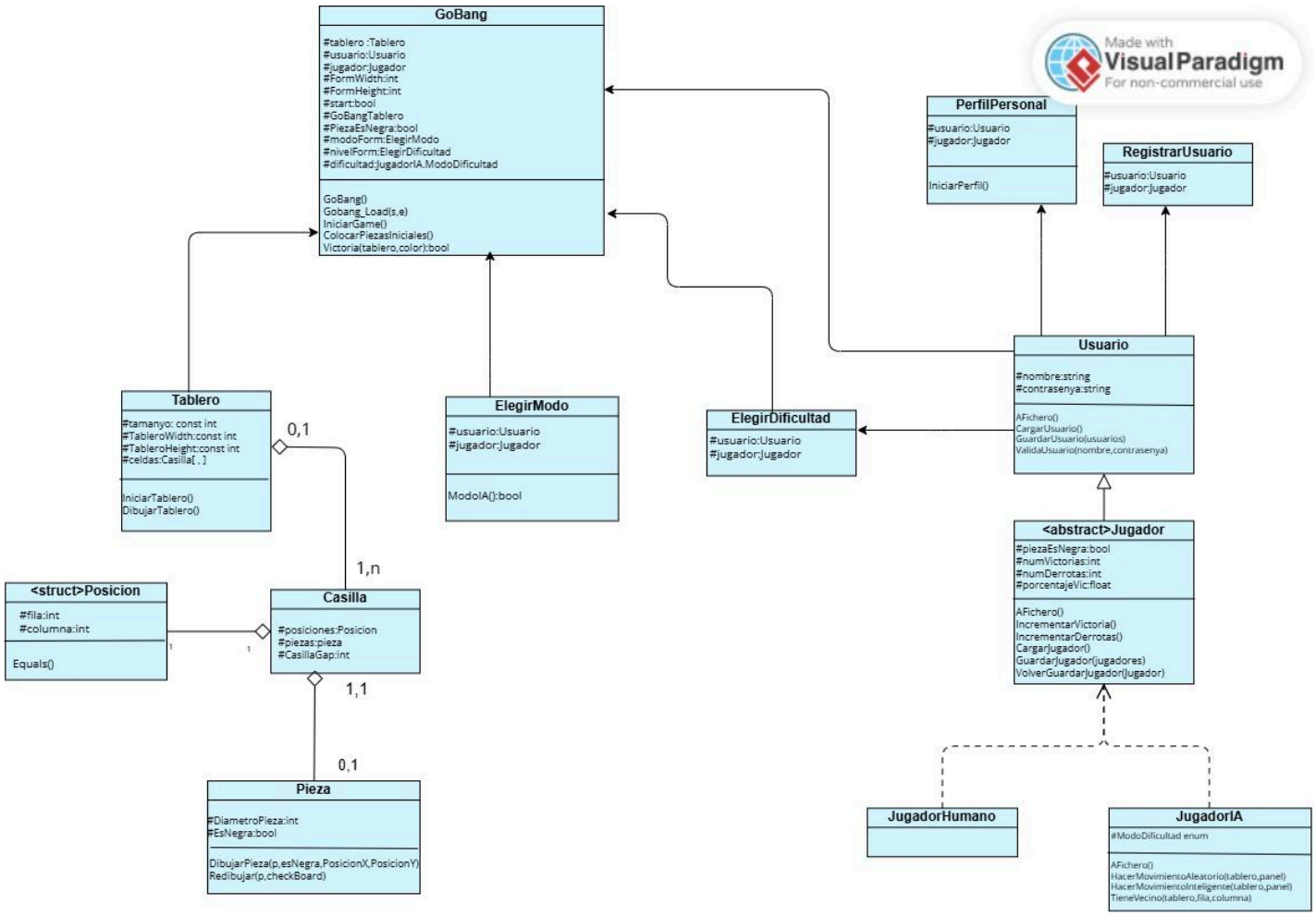
GOBANG

Kaxi Liu 1º DAM B Proyecto final

Diagrama de clases definitivo



Cambios con respecto a propuesta inicial

-Usuario.cs y Jugador.cs:

Se eliminaron del Usuario.cs los campos de num victoria/derrotas y la porcentaje de victoria del jugador, y se trasladaron a la clase Jugador.cs, ya que me di cuenta de que para el inicio de sesión del usuario solo se necesita el nombre y la contraseña.

También se eliminó el Nickname del jugador, y se hizo que la clase Jugador heredara de Usuario.

Después de que Jugador hereda de Usuario, los datos se almacenan por separado.

-Usuario.cs y RegistrarUsuario.cs

Sobre los métodos de registro e inicio de sesión de usuario, añadí una nueva clase llamada RegistrarUsuario, y utilicé botones para facilitar el registro y el inicio de sesión.A través del método ValidaUsuario(), se verifica si el nombre y la contraseña del usuario son correctos para poder registrarse o iniciar sesión.

El método anterior CerrarSesion() fue reemplazado por métodos más avanzados: CargarUsuario() y GuardarUsuario().

-Jugador.cs , JugadorHumano.cs y JugadorIA.cs

En la clase JugadorIA.cs se añadieron dos modos de juego: Fácil y Medio (incluyendo el método TieneVecino()), pero la lógica del modo de juego del JugadorHumano se trasladó directamente a la clase GoBang.cs.

El método GuardarVictorias() de la antigua clase Partido.cs fue reemplazado directamente por los métodos IncrementarVictoria() e IncrementarDerrota() en la clase Jugador.cs.

La información del usuario y del jugador se guarda por separado, y se añadieron nuevos métodos: CargarJugador() y GuardarJugador(). VolverGuardarJugador() solo se actualizan los datos del jugador actual y se actualizan con la lista de jugadores en el archivo.

-Pieza.cs

En la clase Pieza.cs se eliminó el uso del más complejo Color color, y se utilizó un bool para distinguir los colores. Además, se añadió el método DibujarPieza. Al principio, solo me enfoqué en la lógica del juego, pero olvidé la forma de dibujar las piezas.

-Tablero.cs

De manera similar, en la clase Tablero se añadieron los métodos IniciarTablero() y DibujarTablero(), y los métodos ColocarPieza() y Victoria() fueron trasladados a la clase GoBang.cs.Se hizo que la clase Tablero dejara de depender de la lógica del juego, convirtiéndola en una clase general de gestión de tablero.

-Casilla.cs

Esta clase no tuvo muchos cambios, solo se eliminó el bool estaVacia, ya que lógicamente se puede saber si una celda está vacía verificando el estado de pieza, es decir, si pieza == null.

-Partido.cs (anterio) y GoBang.cs

La clase Partido se encargaba de gestionar el estado independiente de una partida (como el tablero actual y el jugador actual), pero esto coincidía en gran medida con las funciones de la clase principal GoBang, por lo que decidí dejar de usar esta clase.

-Partido.cs(anterio) —> ElegirModo.cs y ElegirDificultad.cs

El modelo original de juego (modeloJuego: Amigo/IA como string, y dificultadIA como cadena) no se adapta bien a mi juego. Necesito controlar la operación mediante botones, y manejar tanto modoJuego como dificultadIA en una misma clase resulta confuso. Por eso, creé dos clases nuevas que son más intuitivas y tienen una lógica más clara.

-GoBang.cs

Antes solo existía un método IniciarSesion (que ahora es Gobang_load), utilizado para validar la identidad del usuario e inicializar los Usuario y Jugador.

Ahora se añadió el método IniciarGame(), que es el antiguo IniciarPartida() de la clase Partido. Su función es inicializar el estado dinámico del juego, es decir, antes de comenzar cada partida se limpian las piezas del tablero, pero sin afectar la información del usuario.

El método Pieza.DibujarPieza() se coloca dentro del evento Paint(panel1_Paint) para garantizar que se muestre correctamente en cada actualización.

ColocarPiezasIniciales() al principio era para pruebas, pero ahora funciona como una partida guardada para continuar el juego.

Uso de la videojuego

Al abrir el juego, veremos un tablero a la izquierda y algunos botones a la derecha.
Si intentamos colocar una pieza directamente en el tablero antes de comenzar, aparecerá el mensaje "Comience el juego". Sí, antes de iniciar la partida, debemos presionar el botón Comenzar.

Una vez que presionamos Comenzar, el sistema verifica si hay un usuario conectado. Si no has iniciado sesión, aparecerá la pantalla de inicio de sesión.

En la pantalla de inicio de sesión hay botones para Registrarse e Iniciar sesión (ya he guardado un usuario: nombre qweqwe, contraseña 1234), por lo que puedes acceder directamente con ese usuario guardado.

Después de iniciar sesión, al presionar el botón Comenzar, se te preguntará si deseas jugar con un amigo o con la IA.
Si eliges la IA, se te consultará nuevamente la dificultad del juego.

Luego ya podrás comenzar a jugar.

En la parte inferior derecha están los botones para reiniciar el juego y para ver el perfil personal.

Cuando termine la partida, el sistema reconocerá al ganador y guardará la victoria/derrota en la información del jugador, la cual podrás consultar en el perfil personal.