Informe del Proyecto.

Nuestro humilde proyecto consiste en un juego de dominó que posee la característica de poder ser personalizado por el usuario. El juego enfrenta a distintos jugadores virtuales con diferentes estrategias programadas en un modo de juego previamente elegido por el usuario de entre las opciones anteriormente programadas. La estructura del código tiene como objetivo además facilitar la adición de nuevos estilos de juego que puedan ser programados en un futuro.

La estructura del proyecto posee una amplia biblioteca de clases e interfaces de c# que contine toda la lógica para realizar una partida y una aplicación de blazor desde la que el usuario puede interactuar y personalizar (con ciertas limitaciones) la partida.

Clases e Interfaces

Ficha:

La interfaz IFicha$< T >$ recoge las funcionalidades importantes que a nuestro parecer debe tener una ficha de dominó. Posee un valor T para cada una de sus caras y un peso determinado. En el caso de la Ficha clásica que implementa IFicha$< int >$ el valor de su peso es la suma de ambas caras.

Tablero:

El tablero es la pieza central de la partida. Este es el encargado de guardar la información de cada movimiento realizado durante el juego de modo que pueda accederse a ella de manera organizada. En nuestra implementación de la clase decidimos crear un árbol que tuviera como valor en cada nodo la información relativa al movimiento en el que fue creado y una lista donde se guardarán los movimientos realizados posteriormente por su rama. Para la construcción de este árbol tenemos un nodo raíz, y la ficha de salida conectada a este nodo raíz. Tuvimos la necesidad de crear 2 nodos, o los necesarios dependiendo de las reglas para las ramas del juego.

Jugador:

El jugador es el encargado de en su turno correspondiente realizar una jugada en base a una estrategia definida. Nuestra clase Player$< T >$ se instancia en su constructor con un string Nombre, que permita identificarlo y un delegado que será su estrategia. Entre las estrategias implementadas está: la aleatoria, la botagordas y la de tratar de no pasarse, para esta hicimos uso de un diccionario donde ibamos poniendo las apariciones de cada valor de las caras de las fichas para luego jugar la q tenga las caras que más veces se repitan.

Mano:
La mano es un concepto sencillo. En la partida existe una por cada jugador y esta guardará una lista de las fichas que le corresponden.

Dealer:

El Dealer (Repartidor) es la clase encargada de, dado un conjunto de fichas, una cantidad a repartir y una lista de manos; añadir fichas de ese conjunto a las manos correspondientes.

Generador:

El Generador es el encargado de crear el mazo de fichas con el que se jugará la partida. Dentro de las implementaciones de nuestra interfaz IGenerador$< T >$ se encuentra el generador Clásico que genera el mazo usual de dominó de 6, de 9 o de cualquier número que se seleccione. El generador de fichas pares que solo genera fichas cuya suma de caras sea un número par. Y el generador de primos que genera fichas que en cada una de sus caras poseen como valor un número primo.

Matcher:

El Matcher (ni idea como llamarlo en español) es el encargado de conocer si una ficha puede jugarse en un lugar del tablero dada las condiciones de la partida. Lo cual es la base del reglamento de cualquier partida de dominó.Entre nuestras implementaciones de la interfaz IMatcher$< T >$ tenemos el matcher clásico, el matcher de longana, que tiene que tiene lo necesario para jugar una partida de este modo, como hacer jugable una ficha solo en la rama correspondiente a dicho jugador para el matcher de longana las fichas matchean por el criterio clásico, pero los métodos jugabilidad son peculiares. En el método jugabilidadPreJugada validamos la jugabilidad del último nodo de la rama del jugador que le toca el turno, y en el PostJugada, en caso de q no se haya pasado se vuelve a poner en falso la jugabilidad de dicho nodo, además de que si corresponde la jugada de salida crea tantas ramas como jugadores hayan; y por último el matcher de distancia 2 que hace que una ficha encaje con otra si la diferencia entre sus valores es 2 o menos.

Turner:

El Turner es el encargado de decidir a que jugador le corresponde jugar en cada turno. Entre nuestras implementaciones para Turner está el turner clásico que luego de un jugador le toca al siguiente en la lista de manera ciclíca, el turner inverso, que hace lo mismo... pero en sentido inverso, y el turner aleatorio, que decide a que jugador le toca jugar de manera aleatoria este tiene un método que devuelve un IEnumerable de enteros de forma "lazy", donde cada posición del enumerable contiene el número del jugador que le toca en ese turno.

EndCondicion:

La EndCondicion o "Condición de finalización" es la clase que determina bajo que condiciones una partida está finalizada. Entre las implementaciones de IEndCondicion$< T >$ se encuentra el EndCondicion clásico, que ocurre cuando todos los jugadores se pasaron en la misma condición del tablero o cuando uno

de los jugadores se queda sin fichas en su mano; el EndCondicion por puntos, que detiene el juego cuando la suma total de los números de las caras de las fichas en el tablero es igual o mayor a una cantidad previamente definida.

WinCondicion:

Esta clase una vez cumplido el Encondicion de una partida es la encargada de puntuar a los jugadores y establecer un ranking donde el ganador sea el primero de este. Entre las WinCondicion implementadas se encuentra la clásica, que da más puntos al jugador que tenga una menor suma de números en su mano, siendo el indiscutible ganador aquel que logre quedarse sin fichas en su mano. El ganador por fichas, que funciona de manera similar al clásico, solo que este no tiene en cuenta el peso de las fichas de la mano, si no el número de fichas en sí; y el ganador por jugadas, donde gana el jugador que más jugadas haya realizado en toda la partida.

Partida:

La partida es el objeto de comunicación con el usuario. Este recive los parámetros necesarios para realizar un juego de domino y guarda entre sus propiedades el tbalero, la lista de jugadores y sus manos, lo cual facilita implementar una muestra visual de los minsmos de ser deseado.

Referee:

El referee recibe todos los parámetros que le envía la partida y ejecuta los métodos de estos en el orden necesario para que la partida fluya correctamente. Posee 2 métodos principales. Run, que corre la patida hasta que se cumple la EndCondicion y el método RunTurn que juega un solo turno de la partida cada vez que se llama y devuelve un bool que indica si se cumplió o no la EndCondicion; ambos metodos del referee siguen casi de la misma forma una serie de pasos lógicos, los cuales se encargan de cambiar la jugabilidad en el tablero antes de la jugada, enviarle al jugador que le toque la lista de jugadas poisbles y luego poner la ficha en el tablero a través del método efectuar jugada. En caso de que la partida se acabe se reorganiza la lista de jugadores en forma de ranking. el referee consta también del método efectuar jugada, si la jugada es salida crea 2 ramas para que se pueda efectuar el juego, y si no, pone la ficha en donde corresponde y llama a la jugabilidad del matcher post jugada.

Movimiento:

La clase movimiento contiene como parámetros los datos necesarios para interpretar cada jugada que el jugador decida hace. Saber que ficha jugó, por donde lo hizo y que caras juntó. El Matcher en la partida crea una lista de Movimientos con todas las jugadas válidas que puede realizar el jugador en este turno, el jugador recibe dicha lista y en base a su estrategia decide cual de esos movimientos efectuar.

Interfaz Gráfica:

El apoyo visual de nuestro proyecto está hecho sobre una aplicación web de blazor. Se creó un componente .razor llamado Menu que, imprime un (valga la redundancia) un menú donde el usuario puede seleccionar los aspectos que variarán en la partida. Una vez seleccionado todo el usuario deberá pulsar el botón de "Aceptar" para guardar los cambios que seleccionó y el botón de "iniciar partida" para que desaparezca el menú y comience a correr la partida. Mientras se encuentre la partida iniciada en la página se encontrará un botón para ir avanzando los turnos y a medida que estos avancen se irán visualizando el tablero y las manos de los jugadores. Además de un botón para detener la partida en cuanto se desee.