Dommie Report

Preliminares :

Este proyecto está orientado a cumplir con la evaluación anual de la asignatura Programación en el segundo año de Ciencias de la Computación de la Universidad de la Habana. Su objetivo es desarrollar una aplicación que simule el juego de domino, además de contener ciertos aspectos que van a influir en la mecánica de un juego clásico de domino. El principal propósito es demostrar los conocimientos adquiridos durante el curso que respectan a un buen desarrollo de software, que sea mantenible y extensible, que cumplan en su mayor medida los principios del SOLID y otros principios de la programación en general; todo ello en parte, mediante el uso de interfaces y delegados.

Para ello hemos desarrollado una librería de clases `DominoLibrary` y una aplicación de consola `ConsoleApp`.

La librería de clases es la encargada de controlar toda la lógica del juego, mientras que la aplicación de consola mostrará al usuario los resultados del juego que haya decidido simular, y le permitirá ser partícipe durante la ejecución.

En la carpeta raiz está ubicado el archivo `README.md`, con la información necesaria para la ejecución del proyecto.

Aspectos variables :

Max Token:

Este aspecto se refiere al número del doble máximo que estará en el juego. Este parámetro influirá en la posterior generación de las fichas con que se va a jugar. Por ejemplo, la alternativa clásica del doble 9, va a generar 55 fichas donde la mayor de todas será el doble-9, ésta sería la ficha máxima. Su implementación viene dada como un parámetro de tipo `int`, que es pasado a las clases y métodos tras su selección.

Variantes:

- double - 6
- double - 7
- double - 8
- double - 9
- double - 10
- double - 11
- double - 12

De momento han sido las variantes implementadas, pero si fuera necesario se pudiera extender a más opciones en el futuro.

Number of Players:

De igual manera que el aspecto anterior, damos la libertad de elegir cuantos jugadores el usuario quiere sentar en la mesa, no todos tenemos solo cuatro amigos (yo tengo unos poquitos más). Hemos establecido un rango de entre 2 a 6 jugadores, tampoco queremos que la partida dure lo que un monopoly.

Hand Out:

No es más que las distintas formas en las que se pueden repartir las fichas del juego. Es el criterio que define la manera en la que las fichas se asignan a los jugadores. No queremos que siempre se jueguen todas las fichas sobre la mesa, por tanto no las vamos a repartir todas; para ellos hemos decidido que se repartan siempre el **71%** de las fichas del juego, dejando una parte fuera de partida, así complicamos un poco las posibles estrategias de jugadores más aventajados.

Variantes:

- Random: Esta variante es la clásica forma de repartir las fichas del juego. No es mas que ¨dar agua¨, frase típica referida a la acción realizada por los jugadores para mezclar las fichas que luego elegirán para jugar la partida.
- Bigger First Unfair:Aquí rompemos la tradicional regla de repartir la misma cantidad de fichas por jugador. Dejamos al azar decidir la cantidad de fichas que tendrán los mismos, eso sí, garantizamos siempre que al menos una ficha tendrán. Para hacerlo un poco más interesante no solo escogemos los jugadores de forma aleatoria, sino que las fichas a repartir se eligirán por la mayor puntuación de la misma.

Over Board Condition:

Es el criterio que se consulta para determinar si una ronda del juego ya llegó a su fin.

Variantes:

- Classic: Con esta variante te sentirás familiarizado. La partida llegará a su fin una vez que uno de los jugadores haya puesto en mesa todas sus fichas de la mano, o simplemente haya sido jugada una ficha capaz de pasar a todos los jugadores en mesa.
- Crazy Token: Una ficha entre las fichas de los jugadores será escogida al azar. Los participantes del juego no tendrán conocimiento de la misma. En caso de haber sido puesta en juego, la partida habrá llegado a su fin. Sin embargo, es posible que todos los jugadores de la mesa se pasen sin haber sido jugada la ficha, por lo que de igual modo la partida habrá llegado a su fin.

Winners Getter Judgment:

Se refiere a al criterio de escoger el ganador de una ronda una vez esta ha terminado.

Variantes:

- Classic: El ganador será aquel jugador que haya puesto todas sus fichas en mesa. En caso de no haber sido posible lo anterior y todos los jugadores tengan al menos una ficha en su poder, serán contados los puntos de las fichas y obtendrá la victoria aquel jugador que obtenga la menor puntuación. De haber dos jugadores con la mínima puntuación posible el juego será declarado empate, es decir, ninguno de los participantes habrá alcanzado la victoria.
- Random: Una vez más dejamos al azar decidir tu suerte. Independientemente de tu puntuación o de haber sido capaz de poner todas las fichas en mesa, el jugador será elegido de forma aleatoria. Ahora bien, miremos el lado positivo, si durante la partida no te está yendo demasiado bien, no todo está perdido, aún puedes ser tú quien obtenga la victoria.
- Smallest 5 Multiple: Si decides jugar escogiendo esta variante debes combinar muy bien tus jugadas, pues solo podrás obtener la victoria si eres capaz de lograr alcanzar en tu puntuación el menor múltiplo de cinco distinto de cero al final de cada partida. En caso de no haber algún jugador que cumpla la condición, el juego será declarado empate.

Points Getter Judgment:

Se refiere al criterio mediante el que se computa una puntuación que será asignada al jugador ganador de cada ronda una vez este se conoce y ya ha terminado la ronda.

Variantes:

- Classic: Si eres capaz de alcanzar la victoria lograrás obtener una puntuación igual a la suma de las puntuaciones de cada uno de los restantes jugadores. En caso de haber empate los jugadores no obtendrán ningun tipo de puntuación.
- 5 Multiples: Jugar con esta modalidad te permitirá alcanzar una puntuación con mayor rapidez o no, solo depende de tus habilidades para dejar al contrario con la mayor cantidad de fichas en mano. Tu puntuación en este caso será cinco puntos por cada una de las fichas que hayan quedado en manos de los adversarios.

Game Mode:

Hemos facilitado que el usuario no solo simule una simple partida de un juego de domino, sino que además pueda jugar torneos, o sea, un conjunto de varias partidas que al ganarlas un jugador, va acumulando puntos y la victoria se alcanzará una vez se haya obtenido una cantidad de puntos que puede ser especificada por el usuario antes de comenzar.

Teams Play:

Otro aspecto que hemos decidido extender es el juego en equipo ya que no hay un buen torneo de domino que no se juegue en parejas; por ellos hemos decidido llevar esta modalidad a Dommie. Nuestra aplicación no se limita a que solo sea mediante parejas, sino que también deja libertades al usuario respecto a la manera de formar los equipos, todo ello completamente personalizable desde los menús de configuración inicial, que bien explicamos en su sección. Para una mejor adaptación a la lógica, consideramos como un equipo de un solo jugador, a los jugadores cuando las partidas son sin equipos.

Human Player:

SÍ!!!, es los que estás pensando, ahora tú eres partícipe de nuestro juego. Te damos la opción de ser parte de Dommie e involucrarte a nuestros jugadores virtuales. Traza tus propias estrategias, demuestra tu destreza en el domino e intenta llevarte las mayor cantidad de partidas a tu palmarés.

Los jugadores :

HumanPlayer:

Implementación de IPlayer que permite al usuario ser partícipe del juego.

SingleStrategyPlayer:

Esta clase define a los jugadores que utilizan una única estrategia preestablecida para jugar.

Strategies:

- ***Botagorda***: El famoso "bota-gorda" tiene como estrategia jugar la ficha de mayor puntuación entre las posibles fichas válidas (dígase ficha válida aquella que cumple que uno de sus extremos coincide con alguno de los extremos de la mesa) en su mano.
- ***Mosaic***: Con esta estrategia el jugador trata de mantener un rango de fichas en su mano para poder acertar tantos números como sea posible. Si todas tus fichas tienen palos similares, te quedarás atrapado si eso es todo lo que está disponible en el tablero.

- ***Random***: Jugar de forma aleatoria no es más que seleccionar una ficha entre las posibles fichas válidas a jugar. Ten en cuenta que usar esta estrategia es impredecible. Con ella puedes o no alcanzar la victoria.

Cómo se elije una plantilla :

Una vez el juego le da la bienvenida al usuario, se muestran una serie de menús que permiten la elección de los aspectos que modifican el juego. Una vez determinado si el propio usuario va a participar en la experiencia de poner fichas sobre la mesa, y el tipo de competición, el motor del juego carga una serie de **plantillas** que el usuario podrá elegir para jugar directamente. Quedan explicadas a continuación:

- ***Classic double-9 (Teams)***: Es el clásico juego en parejas que se juega en las esquinas. Las partidas se juegan entre 4 jugadores, donde los que se sientan a lados opuestos de la mesa conforman los equipos. Se juega con una data máxima de doble-9 donde se reparten 10 fichas a cada jugador de manera pseudoaleatoria. Una ronda se acaba cuando un jugador pone todas sus fichas sobre la mesa o ninguno tiene una ficha capaz de encajar con los extremos de la mesa. ¿quién gana?, pues el jugador que se quedó sin fichas, y en caso de tranque, el que tenga la mano con menos puntos, donde los puntos los conforma la suma de ambos lados de todas las fichas. Una vez hay un ganador, los puntos que este recauda es la suma de las manos de todos los jugadores que se consideren contrarios. En caso de torneo, se ganará la competición cuando se reúnan 100 puntos.
- ***Classic double-9 (Single Player)***: Configuración muy similar a la anterior, solo que esta vez no habrá equipos, es una modalidad Deathmatch pero sobre la mesa.
- ***Classic double-6***: Siguiendo el hilo de una partida clásica, esta vez se juega con una data máxima de doble-6. Con el compañero del frente en el mismo equipo, y respetando la convención antes explicada de la cantidad de fichas a repartir, los jugadores buscan ganar puntos como lo juegan los clásicos chinos.
- ***Crazy Token***: Modalidad propia de Dommie, traemos esta modalidad donde las cosas se salen un poco de lo habitual. Para empezar esta vez se sentarán en la mesa 6 jugadores y con 10 fichas cada uno, aunque esta vez hasta el doble-12; con fichas obtenidas de manera pseudoaleatoria, intentarán ganar una partida quedándose sin fichas en mano o teniendo la menor cantidad de puntos. ¿equipos?, por supuesto, pero esta vez serán dos tríos donde nunca juegan dos compañeros de manera consecutiva. Pero ahora, una peculiaridad es que durante todo el juego, existe una ficha muy particular que será repartida a uno de los jugadores de manera asegurada. Nadie conoce esta ficha ni quien la porta, pero una vez que la ficha se ponga sobre la mesa, la partida terminará inmediatamente. Puede pasar que un jugador nunca llegue a poner la ficha y ninguno, ni siquiera él, lleve una ficha que pueda jugar por uno de los extremos, si esto ocurre

la partida acabará igualmente. Los puntos obtenidos para el torneo se computan de igual manera que el juego clásico.

- ***Customize your own template:*** Permite al usuario acceder a un nuevo menú de personalización del juego, a través del cual podrá combinar a placer, todas las opciones de personalización anteriores y hacer del juego de dominó una experiencia más que preferida.

Detalles de la implementación de los menús :

Para los menús del juego ha sido necesario la creación de clases que permitan un funcionamiento los más amistoso posible con el usuario. Durante la navegación por la interfaz gráfica el usuario interactúa con varios tipos de menús. La navegación por los menús se hace mediante las flechas de dirección del teclado, presionando la tecla **Enter** para confirmar selecciones.

El más común es el `SingleSelectionMenu<T>`, como su nombre indica es un menú de selección simple, que tiene un método llamado `Show()` que no termina hasta que el usuario confirma una selección. Sus opciones de tipo genérico son almacenadas en una lista y durante la navegación muta constantemente un valor que contiene a la selección actual. Como una extensión de este menú, se decidió implementar sobre él mismo una opción que mediante un valor booleano, habilita una opción extra llamada `Continue`, que en caso de usarla, es quien da la salida del método.

Quizás el más complejo sea el menú de personalización de plantilla: `CustomizeGame`. Este menú carga unos valores de los parámetros personalizables por defecto para cubrir el caso en que el jugador continúe a la partida sin establecer todos los valores. Sabiendo ya si el usuario va a poner fichas en juego y el modo de juego (partida o torneo), la configuración iniciaría de la siguiente manera:

Players: 4 (Generados de manera aleatoria)

Teams: No

Max Token: double - 6

Hand Out Judgment: Random (clásico)

Over Board Condition: Classic

Get Winner Judgment: Classic

Points Getter: Classic

Tournament Win Score: 100 points

En base a los preajustes anteriores el usuario puede modificar los aspectos del juego. Una vez establecido todo, se selecciona la opción `Continue` y tras confirmar, comienza el juego.

En el menu de personalización de jugadores no solo se eligen la cantidad de jugadores que rodearán la mesa, sino también las diferentes estrategias que estos van a adoptar durante todo el juego.

Detalles de la implementación lógica :

Una vez configurada por completo una plantilla, el siguiente paso sería ejecutar el IGame, donde IGame sería tanto un torneo como una simple partida.

El Torneo:

Un torneo no es más que una secuencia de partidas de la que los diferentes equipos buscan sacar las mejores puntuaciones y así ganar la competición. Su implementación está en el archivo `Tournament.cs`.

El Board:

Un Board no es más que una ronda del juego de domino. Una ronda es una iteración continua sobre los jugadores que solo tiene fin cuando esta se da por finalizada. Por defecto una jugada se preestablece como un pase, luego esta condición cambia una vez se le permita a un jugador realizar una jugada. Solo se deja a un miembro de la mesa jugar si tiene al menos una ficha válida de poner por alguno de los extremos. En caso contrario la jugada no cambiará su estado y se quedará como un pase. Si un jugador intenta poner 3 veces una ficha inválida en la mesa, su turno se cederá al siguiente jugador y su jugada durante la ronda será un pase. Tras completarse una jugada, la mesa se actualizará y se guardará la última jugada. Luego entra en juego el delegado que comprueba si la partida ha llegado a su fin; en dicho caso se escogerá el ganador de la ronda junto a sus puntos y se devolverá a la instancia anterior del torneo (fin en caso de jugarse solo una ronda).

Conexión con la interfaz gráfica :

La clase `GamePrinter.cs`, es la encargada de mostrar en consola todos los resultados durante el transcurso del juego. Para adjuntar el GamePrinter al juego es necesario: al crear cada instancia de un IGame, ejecutar el método

```
void SetGamePrinter(GamePrinter gamePrinter)
```

perteneciente a la interfaz IGame, de esta manera el GamePrinter se guardará como una propiedad de la clase que implementa IGame, y además se añadirá la instancia del Board o Tournament a dicha clase, así esta tendrá acceso a sus componentes para poder mostrar resultados en pantalla.

Esta clase tiene una serie de métodos que son llamados desde los diferentes momentos del juego y definen la forma en la que se muestra dicha información al usuario.

Por otra parte tenemos la clase `PlaySelectorMenu.cs` que es utilizada por el GamePrinter como el menú que permite al usuario ser el jugador humano.

Clases que están presentes durante el juego :

`CircularList.cs`

Esta clase es una implementación genérica de una estructura de dato que funciona de manera similar a una lista enlazada mediante nodos. Su principal razón de ser es que implementa la interfaz IEnumerable de manera que pasándole un IEnumerator por constructor nos permite variar la forma de recorrerla. La idea de su creación fue para realizar el ciclo `foreach` sobre los jugadores de manera circular, es decir, que tras el último jugador aparezca el primero en el ciclo y solo se termine el mismo al llamar a `break`.

`Token.cs`

Es la clase que englova el concepto de ficha. Además `Token_onBoard` define a una ficha una vez puesta en la mesa, una ficha que ahora tiene dueño, una orientación, y el lugar de la mesa por que fue jugada.

`Team.cs`

Define lo que es un equipo.

`Judge.cs`

Esta clase es la que contiene los delegados que el juego ejecuta para hacer cumplir las reglas. Además tiene un método `IsValid()` que determina si una jugada es válida.

`Setting.cs`

Esta clase es un simple contenedor de información que guarda los datos que necesitan el `Board.cs` y el `Tournament.cs`

`Lapse.cs`

Se encarga de relentizar la ejecución, el tiempo que se determina en su constructor.

`GameResult.cs`

Contenedor de la información de un ganador de ronda o torneo.