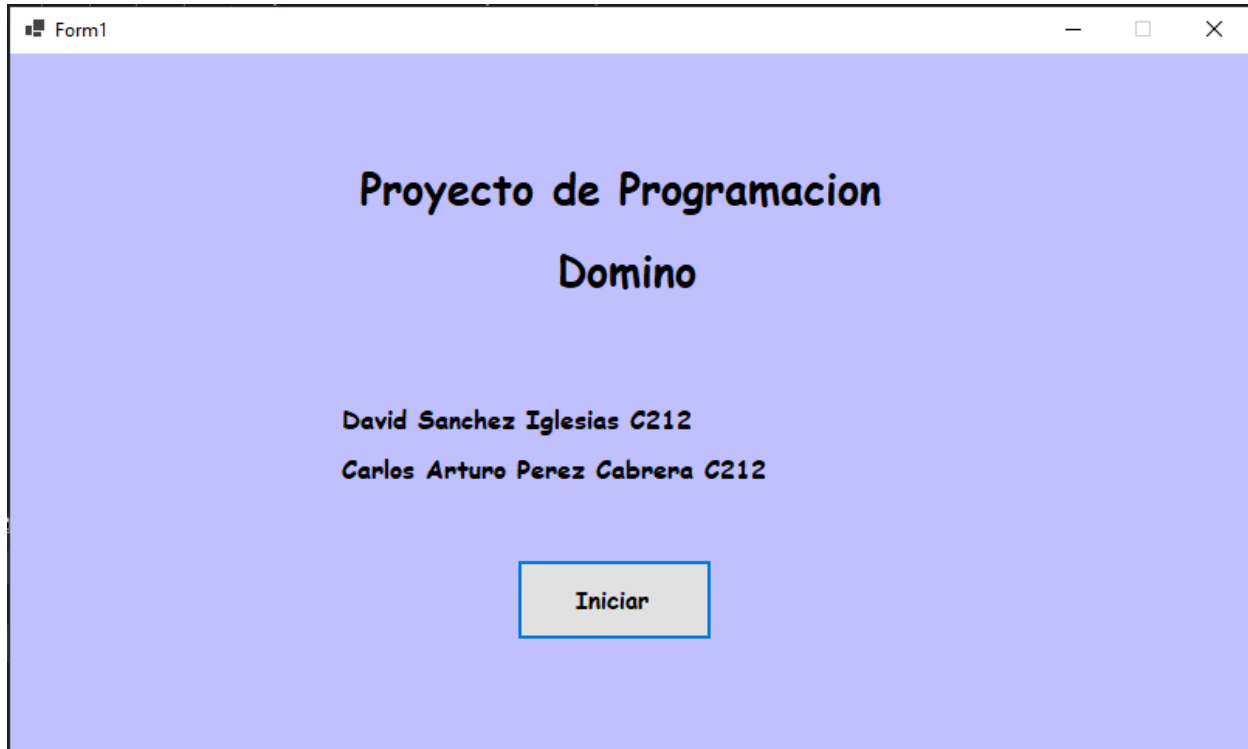


PROYECTO DE PRO DOMINO

David Sánchez Iglesias C 212

Carlos Arturo Pérez Cabrera C 212

MATCOM



Instrucciones:

Para usar esta aplicación y poder reproducir y observar el juego de dominó, deberá seguir unos sencillos y cortos pasos que aquí se explican.

Después de presionar el botón “Iniciar” de la pantalla de presentación, se abre la pantalla de configuración del juego. En esta nueva ventana, el usuario puede escoger y cambiar a voluntad los modos de juego, la condición para que finalice el juego, el orden en el que jugarán los jugadores, la forma en la que se repartirán las fichas a cada jugador y la forma de calcular la puntuación de cada jugador en dependencia de las fichas que haya jugado. En este proyecto se han implementado varias modalidades de cada una de las características modificables en esta ventana. Como modo de juego se han implementado los dominós, con fichas de dos números, hasta el doble 6 y hasta el doble 9.

Form3

Modo de Juego(por cantidad de fichas)

FichasDe6
FichasDe6
FichasDe9

Condicion de finalizacion

FinalizacionPorPuntos

Orden de las jugadas

OrdenNormal

Forma Inicial de Repartir Fichas

RepartoAleatorio

Forma de Calcular la Puntuacion Final

ContarFichasDoblesPor2

Atras

Siguiente

Como condición de finalización del juego, también fueron implementadas dos modalidades: “PorPuntos” establece que, si un jugador alcanza determinada cantidad de puntos (50 puntos) durante el juego, este termina y dicho jugador se establece como ganador; y “PorPase” establece que, si un jugador se pasa dos veces seguidas, se termina el juego y se determina el jugador con la mayor cantidad de puntos hasta el momento, como ganador. Es importante aclarar que un jugador obtiene puntos al jugar una ficha, teniendo en cuenta la manera en la que escogió que se calculara la puntuación de las fichas; en otras palabras, la puntuación se decide evaluando la ficha que se juega en cada jugada, y sumando el resultado a los puntos que el jugador haya acumulado hasta el momento. En “Orden de las jugadas” se han hecho también dos variantes, “Normal”, que mantiene el orden de turnos de los jugadores hasta que acabe el juego; y “CambioSiSePasa”, que establece que, si un jugador se pasa (no juega ninguna ficha) entonces el orden de turnos se invierte, ejemplo: juegan “jugador1”, “jugador2”, “jugador3” y “jugador4” en ese orden, siendo el jugador1 el primero en jugar; si cuando debe jugar el jugador3 este se pasa, el siguiente en jugar sería el jugador2, pues se

invierte el orden de turnos. Si en el ejemplo anterior, después de que el jugador3 se pase, el jugador2 también se pasa entonces el algoritmo vuelve a cambiar el orden de los turnos y busca el siguiente jugador que pueda jugar.

Respecto a la forma de repartir, se debe aclarar algo primeramente, y es que una cosa son las fichas que se le dan a los jugadores para que ellos escojan, y otra es la manera en la que estos últimos escogen las fichas de su “mazo”. La forma de repartir no determina qué fichas le tocan a cada jugador, sino a partir de qué conjunto de fichas los jugadores pueden escoger, y si ellos pueden ver o no los números de estas fichas. Sabiendo esto, se programaron dos formas de repartir las fichas: aleatoriamente (equivale a darle a escoger a los jugadores de entre todas las fichas del juego, con las fichas boca abajo, o sea, el reparto usual), y el reparto boca arriba, que permite a los jugadores “ver” el contenido de las fichas y escogerlas a conveniencia. Por último, como forma de calcular la puntuación, se han creado tres criterios: las fichas dobles tendrán el doble de la suma de los números que posean, por ejemplo, el doble 6 valdría $2 * (6 + 6)$, o sea, 24; otro criterio es contar el valor de las fichas dobles como negativo, esto es restar los puntos que poseen las fichas dobles del total de puntos que posee el jugador hasta el momento; y un tercer criterio sería sumar ambos números de cada ficha a la puntuación del jugador. Luego, una vez configurado el juego, el usuario debe presionar “Siguiente” para acceder a la ventana de creación de jugadores.

En dicha ventana se encuentran dos campos a rellenar: el primero: “Tipo de jugador”, determina la estrategia que empleará el jugador durante el juego una vez creado; el segundo: “Nombre del jugador”, proporciona un nombre a dicho jugador para diferenciarlo de los ya creados anteriormente o de los que se crearán luego, si ese es el propósito del usuario. Respecto a los tipos de jugadores, en este proyecto se han implementado tres: el aleatorio, cuya estrategia es precisamente jugar una ficha aleatoria en cada jugada; el goloso, que siempre jugará la ficha con la que pueda obtener más puntos, dependiendo de la manera en la que se distribuyan los puntos; y el “inteligente”, que buscará la ficha que pueda jugar por un número y que el otro número sea lo más común posible dentro del resto de fichas que posee. Finalmente, después de escoger la estrategia del jugador y un nombre para él, se debe presionar “Agregar” para añadirlo a la lista de jugadores. El botón “Eliminar” se usa para borrar uno de los jugadores ya

creados y que se encuentre en la lista. Para borrar un jugador se debe dar click encima de un jugador en la lista de jugadores y luego presionar el botón “Eliminar”.

Después de seleccionados los jugadores, el usuario ha de presionar el botón “Siguiente” con el fin de pasar a la ventana de juego.

En esta ventana se puede ver el juego y cómo se desarrolla: lo que juega cada jugador y los puntos que estos tienen en cada jugada.

Aquí hay 4 botones: “Siguiente Jugada” muestra la ficha jugada por el jugador al que le toca ahora, o sea, le da el turno al siguiente jugador en la cola; “Juego Automático” reproduce cada jugada de manera autónoma hasta que termine el juego o hasta que se presione el botón de nuevo; “Mostrar Final” pone en pantalla el juego completo y su resultado de manera instantánea; y “Resetear” reinicia el juego, manteniendo las condiciones y configuraciones establecidas en la ventana de configuraciones del juego, con la diferencia de que a los jugadores se les vuelve a repartir las fichas para que escojan.

Detalles técnicos:

Este proyecto se divide en dos partes: la parte gráfica, cuyas instrucciones para su uso fueron anteriormente descritas; y la parte lógica. La lógica se divide en tres partes más: los jugadores, el juego y las opciones. Cada parte posee una serie de interfaces y clases que posibilitan el funcionamiento del proyecto.

En la parte de las opciones se encuentran las clases que determinan las reglas del juego. Estas incluyen la naturaleza de las fichas, la puntuación de estas, el orden de los jugadores y las demás opciones que se pueden modificar en la sección de las opciones en la parte gráfica. Primero está `IFicha` que sirve de molde para definir, precisamente, lo que es una ficha como una tupla de dos variables tipo `object`; esto permite que se puedan crear fichas de cualquier par de variables de cualquier tipo. La clase `Ficha` define lo que es una ficha usando la anterior interfaz como base.

La interfaz `IFicha` se compone de una variable que define la cantidad de fichas que deberá poseer cada jugador como máximo, y de un método que genere todas las fichas con las que se jugará. Las clases que implementan esta interfaz fueron programadas para generar los juegos clásicos del dominó más conocidos: el dominó con fichas desde

el doble 0 hasta el doble 6, y hasta el doble 9. Es importante aclarar que, al crear una ficha, al constructor de la clase `Ficha` se le pasan dos números enteros, creando así las fichas numéricas.

La siguiente interfaz, `ICondicionDeFinalizacion`, y las clases que la implementan: `PorPuntos_50puntos` y `PorPase` definen las maneras en las que se puede calcular la puntuación. Estas se componen de un solo método que determina si con la última jugada el juego terminó de acuerdo a la condición que se haya escogido anteriormente. Cada clase determina una condición distinta.

La interfaz `IOrdenDeLasJugadas` consiste en dos métodos: `Siguiente`, que devuelve el jugador al que le toca después de la actual jugada, y `Reset()`, que restablece el orden inicial de los jugadores solamente cuando se presione el botón “Resetear”. Las clases que usan esta interfaz (`Normal` y `CambioSiSePasa`) describen dos maneras de establecer el orden de los jugadores.

`IFormadeRepartir` es una interfaz que es implementada por dos clases que definen cada una, una variante distinta para dar a los jugadores el conjunto de fichas del cuál ellos pueden escoger las que les convenga. Esto da más libertad a la hora de diseñar nuevos modos de juego. La interfaz y las clases que la implementan contienen una sola función: `Repartir`, que cumple solo cumple con el propósito ya descrito.

`IFormaDeCalcularPuntuacion` es una interfaz que establece que para calcular el valor de una ficha hace falta un método `Conversion` que define el valor numérico entero que tendrá cada elemento de cada ficha, lo que permite la creación de fichas que no necesariamente sean numéricas, solo si en `Conversion` se le da un valor a cada objeto que esté en las fichas; y otro (`CalcularPuntuacion`) que calcula el valor total de cada ficha que se le pasa como parámetro, utilizando la función anterior. El valor de una ficha se determina en el momento en el que esta ficha es jugada.

En el archivo “Jugadores.cs” se encuentra la implementación de los jugadores. Cada tipo de jugador es una clase que implementa la interfaz `IJugador`. Cada jugador debe tener un nombre, puntuación que se actualiza con cada jugada, una variable booleana `EsTurno` que permite determinar si a ese jugador le toca jugar, otra variable booleana `Ganador` que se revisa cuando se determina que el juego acabó para saber cuál jugador es el ganador (`true` implica que es ganador), una lista de fichas con las que puede jugar, una función

que calcula la puntuación de las fichas en dependencia de la opción que se haya seleccionado en las opciones del juego, y tres métodos: `Juega`, que devuelve la ficha que el jugador jugará cuando le toque, y si no puede jugar, devuelve una ficha nula; `seleccionar`, que escoge, al inicio del juego, las fichas con las que jugará de entre las que le dan; y `EsFichaJugable`, que es un método booleano auxiliar que devuelve *true* si la ficha se puede jugar por alguno de los dos extremos en el tablero.

Por último, en “Juego.cs” se encuentra el funcionamiento del juego como tal. Consiste en una clase `Jugada` y un “enumerable” `Juego`. La primera es una clase que se dedica solamente a escribir en pantalla la ficha que se ha jugado, si se acabó el juego, por qué se acabó el juego y quién ganó. En el segundo ocurre todo el juego: cada vez que se presiona el botón “Siguiete Jugada” se ejecuta el “`MoveNext()`”, donde se determina a qué jugador le toca, qué ficha juega, se actualiza la puntuación de dicho jugador, determina si se acabó el juego, y si es así determina cuál ganó e imprime la información en pantalla; por supuesto, todo esto llamando a los métodos y clases correspondientes, cumpliendo con la función de un árbitro.