



MEMORIA TRABAJO WPF



Manuel Clerigué García

DNI: 71035275M

Índice

Manual de usuario:	2
Ventana principal:	2
Ventana de configuración:	4
Manual de programador.....	5
Clases principales	5
Card	5
Dealer	5
Game	6
AuxWindow	6
MainWindow	6
Otros	6
Bibliografía	7

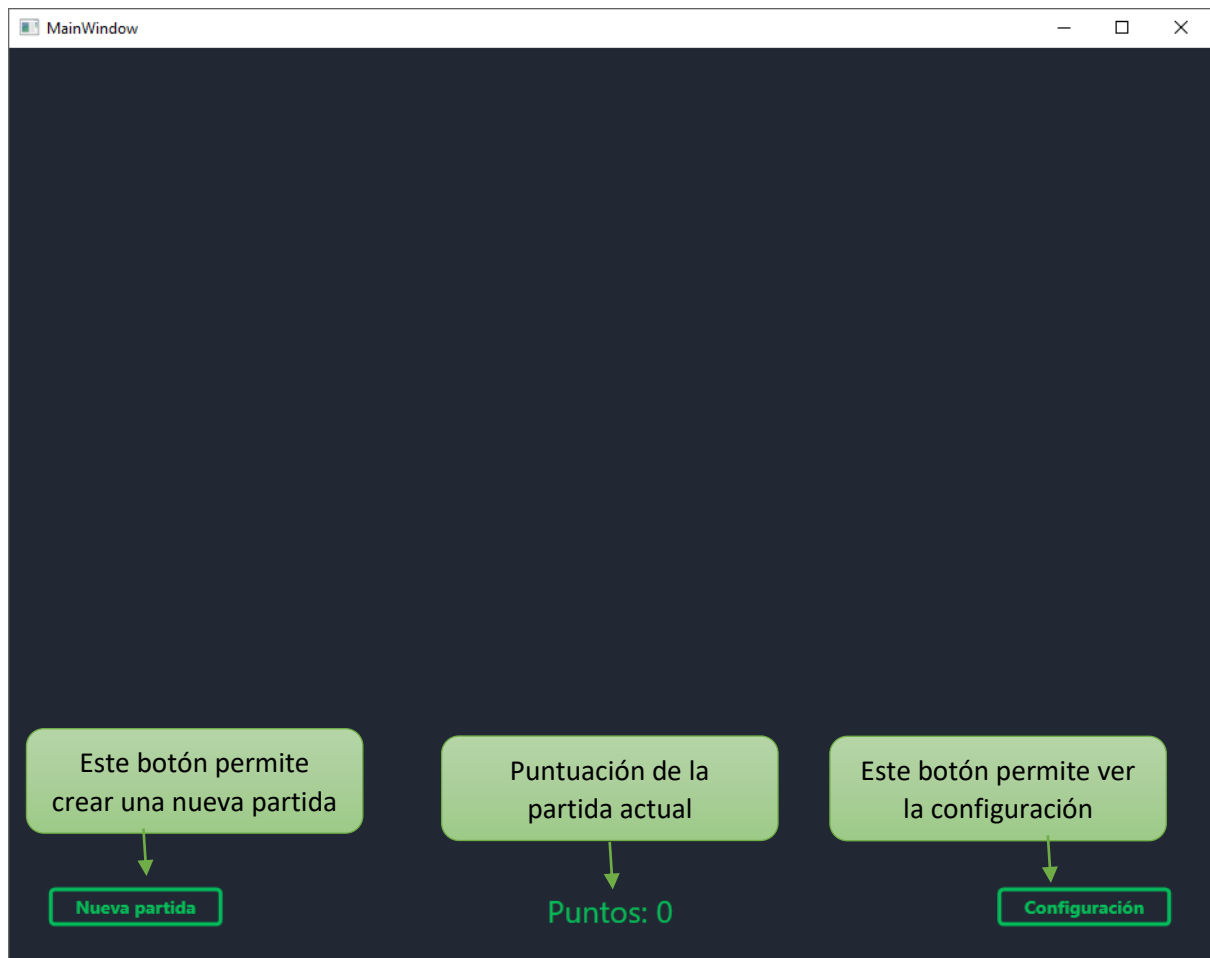
Manual de usuario:

Este programa es una implementación del clásico juego de memoria de emparejar cartas:

Se presentan un número par de cartas en pantalla de las que se desconoce el valor, al hacer click sobre una carta se muestra su valor brevemente. Una vez se ha vuelto a ocultar el valor de esa carta se puede mirar el de otra. El objetivo es encontrar todas las parejas de cartas con el mismo valor en el menor número de clicks posible.

Ventana principal:

En esta ventana se desarrolla la actividad principal del juego. Presenta tres controles simples, uno que permite crear una nueva partida, otro que permite acceder a las opciones de configuración y otro que muestra la puntuación actual.



Ventana de configuración:

En esta ventana se muestran los controles para configurar diferentes aspectos del juego (dificultad, guardar/cargar partidas, mostrar el valor de las cartas...), así como una tabla en la que se pueden observar las cartas que hay actualmente en la ventana principal de juego.

The image shows a software window titled "AuxWindow" with a dark blue background. It contains a table with three rows and two columns. The first column is labeled "ID" and contains the number "0" in green. The second column contains three playing cards: a yellow card with two white circles, a red card with a black club symbol, and a red card with two white circles. Below the table, there is a checkbox labeled "Mostrar cartas" with a green arrow pointing to it. To the left of the checkbox is a slider labeled "Dificultad: Fácil" with a green arrow pointing to it. To the right of the checkbox is a text input field with a green arrow pointing to it. To the right of the text input field is a green button labeled "Guardar" with a green arrow pointing to it. Below the "Guardar" button is a dropdown menu with a green arrow pointing to it. There are five green callout boxes with arrows pointing to specific elements: one pointing to the table, one pointing to the "Mostrar cartas" checkbox, one pointing to the "Dificultad: Fácil" slider, one pointing to the text input field, and one pointing to the dropdown menu.

AuxWindow

Tabla con información de la partida actual (ID e imagen correspondiente a cada carta)

ID
0

Indica si las cartas de la ventana principal muestran su valor o no

Mostrar cartas

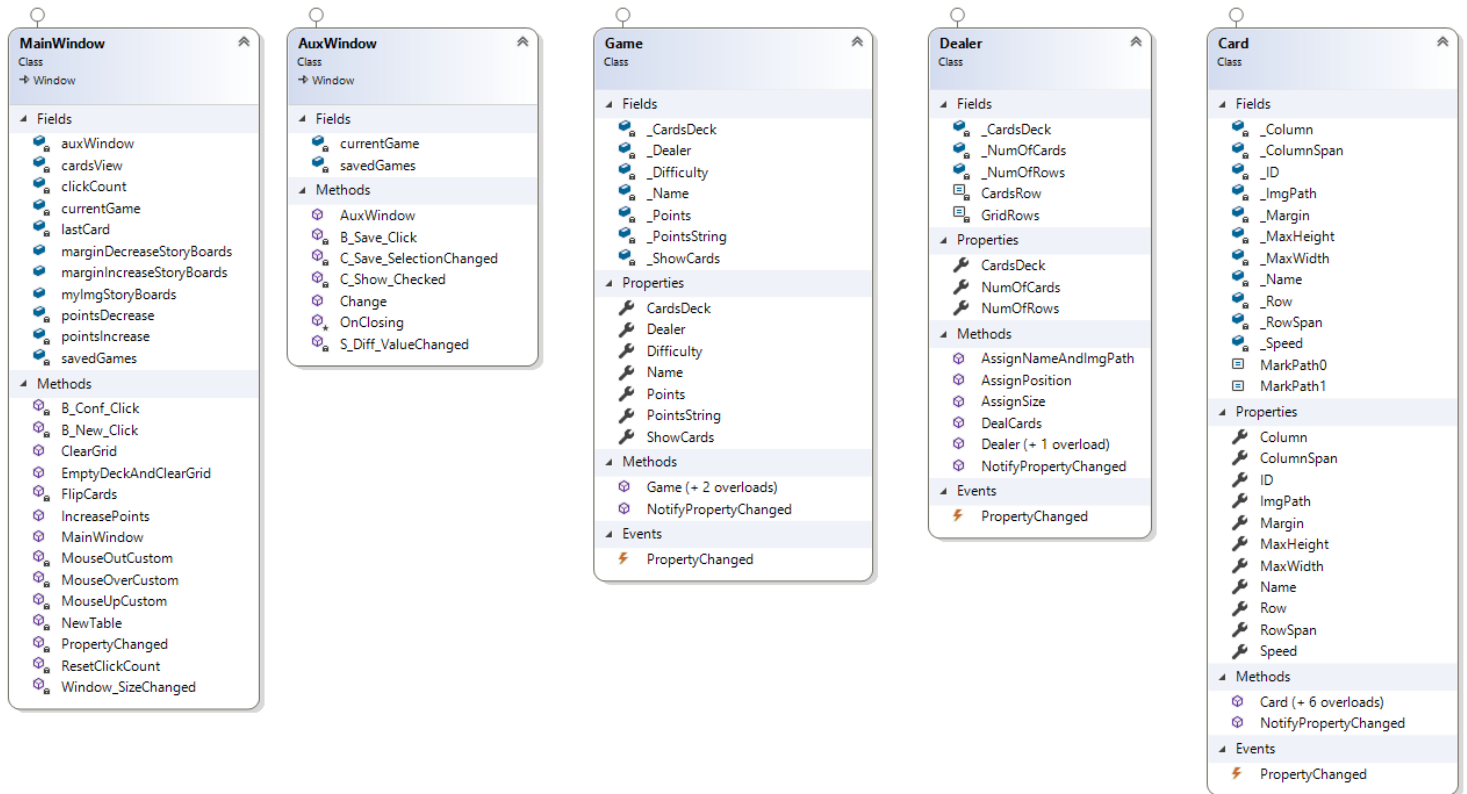
Dificultad: Fácil

Permite guardar la partida actual con el nombre dado

Guardar

Muestra una lista de las partidas guardadas permitiendo cargar cualquiera de ellas al seleccionarla

Manual de programador.



Clases principales

Card

Esta clase es la que representa el objeto básico de la aplicación, las cartas. Sus propiedades describen la que será su posición en el grid, así como su tamaño, y diferentes valores que permiten identificarla de manera única.

Los únicos métodos que incluye están relacionados con la creación de las propias cartas, son constructores.

Dealer

Esta clase es la encargada de organizar la “partida”. Tiene datos relevantes como el número de cartas totales, el número de cartas que habrá por fila, etc.

Sus principales métodos se encargan de crear una colección de cartas para los datos del juego que recibidos:

```
public ObservableCollection<Card> DealCards(double gridHeight, double gridWidth, Game currentGame)
```

Dependiendo del nivel de dificultad selecciona el nº de cartas que habrá en juego, las nombra, y llama a las funciones que se encargan de asignar el resto de propiedades de las cartas. Devuelve como resultado la colección de cartas creada (CardsDeck).

```
public void AssignNameAndImgPath()
```

Esta función se encarga de asignar un ID y una imagen para cada carta aleatoriamente, de manera que se asignará la misma imagen a dos cartas diferentes, y solo dos, recibiendo también cada una un ID diferente para poder diferenciarlas con mayor facilidad.

```
public void AssignPosition(double gridHeight, double gridWidth)
```

Esta función se encarga de, dados los parámetros del grid en el que se representarán las cartas, asignar una posición columna/fila a cada una. Esto también se hace de manera aleatoria, consiguiendo con esto que la distribución de cartas final en pantalla no se repita.

También se encarga de asignar un tamaño máximo en los ejes “x” e “y”.

```
public void AssignSize(double gridHeight, double gridWidth, Game currentGame)
```

Esta función se encarga de recalcular el tamaño máximo de cada carta en una partida determinada para un grid determinado.

Game

Esta clase es la que representa los datos de la partida, tanto la dificultad, como los puntos y las cartas en juego.

Al igual que la clase “Card”, sus métodos son constructores de clase.

AuxWindow

Esta clase representa a la ventana auxiliar de configuración de la partida. Sus métodos están asociados a los controles presentes en ella.

MainWindow

Es la ventana principal de la aplicación. En ella se controla la dinámica del juego:

Posee una colección de la clase “Image” (“cardsView”) en la que se guardan la imágenes asociadas a cada carta, de manera que el objeto “Image” sea la representación de la vista del objeto de la clase “Carta”.

Posee diferentes Storyboards y colecciones de ellas destinadas a animar el comportamiento de cada elemento del juego cuando se interactúa con él.

También contiene una colección de “Games”, la cual se utiliza para la gestión de partidas.

De todas las funciones que tiene cabe destacar la función registrada al evento de “MouseUp” sobre la imagen de cada carta. Esta función es la encargada de comprobar si la carta seleccionada actualmente es la pareja de la última carta seleccionada, en cuyo caso se borran ambas cartas de la vista y la colección de cartas, y se suman 50 puntos al contador. En caso de que no sea la pareja se restan 5 puntos.

Otros

Por último, cabe destacar la presencia de dos estilos personalizados implementados para reemplazar los controles “Button” y “CheckBox”.

En los ficheros de ambos estilos se incluye todo el código en xaml que les da forma y que permite que actúen como se espera cuando el usuario interactúa con ellos.

Bibliografía

<https://stackoverflow.com/>

<https://docs.microsoft.com>

Apuntes de la asignatura