	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Diseño de interfaces				CURSO: 2º
	PROTOCOLO:	Examen	AVAL:	2ª	DATA:	4/3/2022
	Nombre y Apellidos:					

Los juegos de cartas están sobrevalorados, son un hype actual y estamos aburridos y hastiados de tanta historia con las cartitas, que si Clash Royale, que si Gwent, que si la Escoba... ¡pues no! No puede ser, vamos a meternos a un juego más original, y aunque no sea especialmente novedoso si va a salir del sistema habitual de juegos de cartas. ¡Vamos a programa el piedra, papel, tijera! ¿El de siempre? ¡No! Uno especial que gestionaremosicon cartas!... original ¿eeeeehhhh? (codo, codo, guiño, guiño). Y así le damos experiencia a cada una de las cartas que tengamos ¡Vivan las cartas! ¡JA! ¡¡Cómo molan las cartas!! No se quién puede opinar de otra forma... parfavaaaaaar.

En fin agárrate a las teclas que nos vamos de viaje:

Crea una solución del tipo ClassLibrary o Biblioteca de Clases (Framework.Net) con el nombre **ApellidosNombre_DI2aEv**.

Se desea desarrollar un componente denominado **Carta** (Archivo *Carta.cs*) que simule una carta de personaje de juego con información de nivel y una barra creciente para indicar un valor o una experiencia. Puedes ver un ejemplo al final del enunciado.

Tendrá las siguientes características: Hereda de **UserControl** y contendrá un **PictureBox** denominado **pb** y una etiqueta denominada **lbl**. Puede tener cualquier tamaño.

(0.3p) Las distintas propiedades y eventos definidas a lo largo de este componente deben tener atributos de descripción y pertenecer a una categoría que se denominará Examen. Además se establecerá como evento por defecto del componente AlcanzaMax (Definido más abajo).

(1p) El **PictureBox** estará situado en la posición 0,0 del componente y ocupa todo el ancho del mismo y todo el alto menos la altura de **lbl**. Tendrá color de fondo blanco. La imagen, si la hay, se ajustará siempre al tamaño del mismo aunque se deforme.

La etiqueta estará pegada al margen izquierdo y justo debajo de **pb**. Pon su fuente en negrita (**Bold**) y el fondo de un color claro pero distinto del blanco y del gris de fondo del componente.

(1.5p) El componente Carta tendrá las siguientes propiedades públicas:

- **Imagen:** Da acceso a la propiedad Image de **pb**.
- **Nivel:** De tipo entero. Cuando se asigne un valor, además de guardarse en la variable privada se mostrará en **lbl**.
- **Valor:** De tipo entero. Será el porcentaje de la barra. Es decir, puede tener un valor entre 1 y 100. Si es mayor que 100 se queda en 100, si es menor que 0 se queda en 0.

(1p) Se definirá el evento **AlcanzaMax** con delegado *EventHandler* que será lanzado cuando *Valor* alcance el nivel 100 pero solo si además la propiedad *Enabled* del componente esté a true.

Se desea también que cuando se produzca el evento *DoubleClick* sobre **lbl**, se lance el *DoubleClick* de *Carta*.

(1p) En la parte inferior, a la derecha de la etiqueta, habrá una barra por defecto de color azul. Dibújala con *FillRectangle*. Su altura será la misma que la altura de **lbl**. Será creciente hacia la derecha según el dato Valor (si cambia Valor cambia la barra). Si Valor es 0, no se ve nada, si es 100 la barra ocupará todo. Cualquier valor intermedio pues ocupará la parte correspondiente del componente.

(0.5p) Además si Valor guarda justo el valor 100, la barra se verá verde en lugar de azul. Y si el componente Carta tiene el *Enable* a false, el color de la barra será *LightGray* (ojo, el efecto con el *Enabled* solo lo verás en ejecución, no en diseño).

Crea **en la misma solución** un proyecto denominado **PiedraPapelTijeraCards** del tipo Windows Forms.

El formulario principal (Form1) dispondrá de tres componentes Carta a la izquierda, cada uno con un RadioButton debajo (sin texto).

Inicializa las cartas con cada una de las imágenes que se entregan, con Nivel a 1 y Valor a 0.

Añádelas a un vector de forma que en la posición 0 está la carta de piedra, en la 1 la de papel y en la 2 la de tijera.

Dispón de un segundo vector con las imágenes o con los nombres de los archivos de imagen.

Ten una carta marcada por defecto. En la medida de lo posible usa el vector junto con los tags de los radiobutton si te hicieran falta para gestionar de forma más cómoda estos componentes evitando repetir código y usando los eventos adecuados.

A la derecha del formulario habrá un componente Carta más, que será la carta que vaya sacando la CPU. En principio déjalo vacío, se rellenará cuando se juegue.

Añade también un botón Jugar y una etiqueta donde se indicará el resultado de cada partida.

En la parte inferior habrá un panel de altura 200, que ocupe aproximadamente el ancho del formulario. El fondo será de un color distinto al formulario para que se vea aunque esté vacío.

Las imágenes a usar o las pones como recursos del proyecto o estarán situadas en el subdirectorio **images** dentro de la variable de entorno **userprofile**.

(0.7p) El usuario podrá elegir mediante los radiobuttons cual es la carta deseada para jugar. Luego pulsa jugar y sucede lo siguiente:

(1p) La CPU saca un número aleatorio entre 0 y 2 que corresponderá a una carta como se indicó arriba. Pon en la carta de CPU la imagen que corresponda.

El nivel de la carta de la CPU será aleatorio pero el mismo que tenga el jugador en la suya seleccionada ± 1 . Es decir, si por ejemplo el nivel de carta del jugador es 5, la CPU sacará una carta de nivel 4, 5 o 6 de forma aleatoria. El valor siempre sera 0.

(1p) La condición para ganar el jugador será: si el jugador tiene piedra y la cpu tijera, o que el jugador tenga papel y la cpu piedra o que el jugador tenga tijera y la cpu papel. Además el nivel de la carta del jugador debe ser el mismo o mayor que el de la CPU.

La condición para ganar la CPU es la misma pero a la inversa. Una vez hechas estas comprobaciones en la etiqueta se indica si gana el Jugador o gana la CPU.

Cualquier otro caso se considera empate.

(0.5p) Tras el juego se genera un número aleatorio entre 10 y 20 con el que se incrementa el valor de la carta seleccionada.

Cuando sea lanzado el evento AlcanzaMax de la carta, se incrementa el Nivel de la misma en 1 y se pone el Valor a 0.

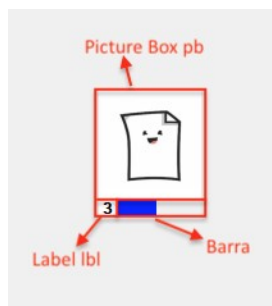
(1.5p) Finalmente en el panel se guardará un histórico de las jugadas mostrando cartas de ambos jugadores de la siguiente forma:

Partiendo de la posición 0,0 dentro del panel, en dicha posición se coloca una copia de la carta del jugador de esta última jugada y en la posición 0,90 una copia de la del ordenador. El tamaño de cada carta será 40 de ancho por 70 de alto. Luego simplemente cada nueva jugada se desplaza 50 pixels a la derecha. Si quieres haz un constructor a medida en Carta, como lo veas más cómodo.

Nota: Si no has hecho el componente completo o no te funciona, cada carta que necesites sustitúyela por un PictureBox y una label donde informas del Nivel y del valor de forma numérica. Si te fuera práctico crea una clase con la imagen el nivel y el valor.

Importante: Pon nombre y apellidos al principio de los archivos Carta.cs y Form1.cs.

COLEXIO VIVAS S.L.	RAMA:	Informática	CICLO:	DAM			
	MÓDULO	Diseño de interfaces				CURSO:	2º
	PROTOCOLO:	Examen	AVAL:	2ª	DATA:	6/3/2020	
	Nombre y Apellidos:						



Piedra, Papel, Tijera Cards

