Manual Técnico

El proyecto fue realizado en el lenguaje de programación C#, para dicha realización se utilizó la herramienta Visual Estudio 2013.

Clases desarrolladas:

Class Celda:

Atributos:

- Fila: de tipo entero, almacena la fila en la que se encuentra la celda.
- Columnas: de tipo entero, almacena la columna en la que se encuentra la celda.
- Valor: de tipo Object almacenará por lo general un dato de tipo entero que corresponde al valor numérico de la celda.
- Estado: de tipo enum, almacena el estado de la celda.
- Imagen: de tipo Image, almacena la imagen que corresponde a dicha celda.
- Tamaño: de tipo Size, almacena el ancho y alto de la celda.
- Ubicación: de tipo Point, almacena la ubicación de la celda.

Class NodoDoble:

Atributos:

- Celda: de tipo Celda, almacena un objeto de dicha clase.
- Li: de tipo NodoDoble, almacena un objeto del mismo tipo.
- Ld: de tipo NodoDoble, almacena un objeto del mismo tipo.

Class MatEnListF1:

Esta clase representa y manipula matrices dispersas con listas ligadas, de la siguiente forma:
-Por cada fila que tenga la matriz tendremos una lista simplemente ligada circular con registro cabeza
-Por cada columna que tenga la matriz también tendremos una lista simplemente ligada circular con registro cabeza.
-El registro cabeza de la lista de la fila **i** será el mismo de la lista de la columna **i**.
-Todo registro pertenecerá simultáneamente a dos listas: la lista de la fila **i** y la lista de la columna **j**.
-Habrá tantos registros cabeza como mayor sea el número de filas o columnas de la matriz a representar
-Con los registros cabeza conformaremos otra lista simplemente ligada circular con registro cabeza. En el registro cabeza de la lista de registros cabeza tenderemos las dimensiones de la matriz que se está representando.

Atributos:

- Mat: de tipo NodoDoble, almacena un objeto que apunta hacia el nodo cabeza de la lista de nodos cabeza.

Métodos:

- NumFilas: retorna el número de filas de la matriz.
- NumColumnas: reotrna el número de columnas de la matriz.
- ConstruyeNodosCabeza: crea los nodos cabezas de las listas simplemente ligadas de las filas y las columnas.
- NodoCabeza: retorna el objeto mat, objeto que apunta hacia el nodo cabeza de la lista de nodos cabeza.
- PrimerNodo: retorna el objeto que apunta hacia el nodo cabeza de la lista siplemente ligada correspondiente a la primera fila y columna.
- ConectaPorFilas: recibe como parámetro un nodo doble y lo conecta por filas en la posición correspondiente.
- ConectaPorColumnas: recibe como parámetro un nodo doble y lo conecta por columnas en la posición correspondiente.
- Conectar: recibe como parámetro un nodo doble y lo conecta tanto por filas como por columnas en la posición correspondiente

Class Game:

Derivada de la clase Form es una colase con interface gráfica de usuario que contiene toda la lógica del juego.

Atributos:

- Filas: de tipo entero, contiene el número de filas del campo de minas.
- Columnas: de tipo entero: contiene el número de columnas del campo de minas.
- NumMinas: de tipo entero, contiene el número de minas que hay en el campo de minas.
- Banderas: de tipo entero, contiene el número de celdas marcadas.
- TamCelda: de tipo Size, contien el ancho y alto de las celdas.
- CeldasSinMina: de tipo entero, contiene el número de celdas que no tienen minas.
- CeldasDescubiertas: de tipo entero, contiene el número de celdas que han sido descubiertas.
- IsPlaying: de tipo booleano, nos permite identificar si se está jugando o no.
- Tablero: de tipo MatEnListF1, contiene todas las celdas del campo de minas representada como una matriz en lista.

- PosicionesDisponibles: de tipo List<int[]>, contiene las posiciones de todas las celdas del campo de minas, lo cual nos será de gran utilidad a la hora de agregar aleatoriamente las minas en el campo.

Métodos:

- Canvas_Paint: dibuja el campo de minas.
- IniciarJuego: crea un nuevo juego.
- CrearTablero: crea un nuevo campo de minas.
- InsertarMinas: inserta minas en el campo de minas.
- AumentarValorVecinas: recibe como parámetro dos enteros, fila y columna de una celda, y aumenta el valor de todas las celdas circundantes.
- ActualizarImagen: recibe como parámetro una celda y actualiza la imagen de dicha celda.
- Descubrir: recibe como parámetro una celda y cambia su estado a descubierta.
- Marcar: recibe como parámetro una celda y cambia su estado a marcada.
- Desmarcar: recibe como parámetro una celda y cambia su estado a cubierta.
- ClickIzquierdo: recibe como parámetro dos enteros, fila y columna de una celda, identifica si se puede o no descubrise una celda, de poder hacerse identifica si está vacía, contiene un numero o una mina, si se ha ganado o perdido el juego e invoca el método correspondiente para cada caso.
- ClickDerecho: recibe como parámetro dos enteros, fila y columna de una celda, identifica si se puede o no marcar o desmarcar una celda e invoca en método correspondiente a cada caso.
- RevelarVecinas: recibe como parámetros dos enteros, fila y columna de una celda, descubre las celdas circundantes.
- GameOver:
- GrameWIn:
- RedibujarTablero: redibuja el campo de minas.
- Canvas_MouseUp: es un evento que se ejecuta cada vez que se da click sobre el campo de minas, identifica si se ha dado click derecho o izquierdo e invoca el método correspondiente.
- Btn_Reinicar_Click: se ejecuta cada vez que se da click sobre el botón reiniciar e inicia un nuevo juego.
- Btn_Volver_Click: se ejecuta al dar click sobre el botón volver y cierra el form.