

MANUAL TECNICO DE JUEGO DE MEMORIA

by Juan Pablo Orizabal Gil

Carné: 201730318

MANUAL TÉCNICO

El software se puede utilizar en cualquier sistema operativo gracias a java, ademas se puede correr en la mayoría de computadoras que tengan instalado java 6 o superior. Se utilizo el JDK8 de java, y el IDE Netbeans.

El software es un juego de memoria puramente gráfico, se utilizo la librería Swing que viene incluido en el JDK 8 de Java, se tienen distintas dificultades, fácil(5 parejas), normal(10 parejas) y difícil(15 parejas), se guarda el historial de cada partida, en estos reportes se almacena el nombre del jugador ganador, del perdedor, el punteo del ganador, la hora y la fecha.

El software consta de varias clases entre las que esta: LogicaDelJuego, JugarJuego(main), Reportes, Jugador.

CLASE REPORTES

Se tienen 7 atributos privados que son los que sirven para moldear cada instancia de la clase Reportes.
Se tiene un constructor para facilitar la instanciación.

Y sus respectivos sets and gets para manipular el valor de los atributos.

CLASE JUGADOR

Se tienen 4 atributos privados que son los que sirven para moldear cada instancia de la clase Jugador.
Se tiene un constructor para facilitar la instanciación.

Y sus respectivos sets and gets para manipular el valor de los atributos.

CLASE LÓGICA DEL JUEGO

En esta clase se tiene la forma en la que se repartirán las cartas, es decir aquí es donde hace toda la lógica para repartir dos cartas de cada una a cada espacio de un vector, lo hace para las tres dificultades y las almacena en un vector que luego será copiado en la clase JugarJuego.

Método `int[] geNumeroCartasDifícil`: Ordena las cartas de dos en dos en la dificultad de difícil. Y devuelve el arreglo de enteros.

Método `int[] getNumeroCartasNormal`: Ordena las cartas de dos en dos en la dificultad de normal. Y devuelve el arreglo de enteros.

Método `int[] getNumeroCartasFácil`: Ordena las cartas de dos en dos en la dificultad de fácil. Y devuelve el arreglo de enteros.

CLASE JUGAR JUEGO(MAIN)

Tiene muchos atributos que no son para instanciar, esta clase es el main la cual ejecutara y llamara a todo lo necesario para hacer que el juego funcione perfectamente. Se tienen varias booleanas, varias arraylist que se crearon de las diferentes clases que existen. Y varias instancias de otras clases.

Método JugarJuego: Es el constructor que nos ayudara a inicializar todo el programa.

Método setCartasDifícil: Sirve para darle imagen a cada carta y repartirlas según el método de la clase logica del juego respectivo.

Método setCartasNormal: Sirve para darle imagen a cada carta y repartirlas según el método de la clase logica del juego respectivo.

Método setCartasFacil: Sirve para darle imagen a cada carta y repartirlas según el método de la clase logica del juego respectivo.

Método btnHabilitado: Sirve para ver si es la primera carta que se levanto o la segunda.

Método compararCartas: Este método es el encargado de comprobar si son parejas o no mediante su descripcion y si es así llama a los métodos respectivos de dar o quitar punteo.

Método repartirPunteoMalo: Como su nombre lo dice da el punteo malo si no se acerto la pareja.

Método repartirPunteoBueno: Como su nombre lo dice da el punteo bueno si se acerto la pareja.

Método actualizarPunteoJugadores: Sirve para ir actualizando el punteo de cada jugador luego de que cambie de turno y se salga de la carta que estaba seleccionada.

Método reiniciarPartida: Sirve para reiniciar la partida, es decir volver a cambiar el orden de las cartas.

Método mostrarQuienGanoLaPartida: Sirve para ver quien gano la partida y muestra una ventana emergente con el nombre del ganador y del punteo.

Método irAlPanelConJugadoresGanadores: Sirve para ir al panel donde estan los jugadores que han ganado una partida.

Método guardarHistorialJugadorGanador: guarda el nombre, clave, punteo del jugador ganador, así como la hora y la fecha.

Método juntarLosPunteosDeTodosLosJugadoresGanadores: Sirve para almacenar en un arreglo de enteros los punteos de los ganadores.

Método añadirAlHistorialJugadoresGanadores: sirve para añadir jugadores a la JTable donde estan los jugadores ganadores.

Método ordenarJugadoresGanadoresPorPunteo: Sirve para añadir a la JTable de jugadores ganadores de una forma ordena de mayor a menor punteo.

Método mostrarReporteMayorPunteo: arregla los punteos llamado al método ordenarJugadoresGanadoresPorPunteo y luego agarra el del mas alto y lo muestra en el panel.

Método ordenarHistorialJugadoresGanadores: Es un método ordenamiento, específicamente quicksort que ordena el arreglo de enteros que se creo anteriormente.

Método regresarASuEstadoNormalLosJugadoresGanadores: En el método anterior se valia de un atributo de los reportes y era el “estaActivo” por si existian dos punteos iguales, lo que hacia era cambiarlo a falso y en este método regresa a su estado verdadero.

Método verificarQuienGanoLaPartidaDificil: sirve para ver si ya se levantaron todas las cartas posibles , si es así muestra quien gano y su punteo.

Método verificarQuienGanoLaPartidaNormal: sirve para ver si ya se levantaron todas las cartas posibles , si es así muestra quien gano y su punteo.

Método verificarQuienGanoLaPartidaFacil: sirve para ver si ya se levantaron todas las cartas posibles , si es así muestra quien gano y su punteo.

Método mostrarInformacionParaNuevoJugador: Lo que hace es que si se tiene seleccionada la opcion de crear un nuevo jugador muestre donde se puede crear al jugador.

Método ocultarInformacionParaNuevoJugador: Lo que hace es que si se tiene seleccionada la opcion de no crear un nuevo jugador no se muestre donde se puede crear al jugador.

Método crearNuevoJugador: Crea a un nuevo jugador, como su nombre lo dice.

Método elegirTurno: Unicamente escoge quien empieza de los dos jugadores aleatoriamente.

Método verQuienLlevaElTurno: Actualiza en tiempo real la casilla donde se muestra quien lleva el turno.

Método cambiarTurno: Cambia luego de que finalice un turno la casilla donde se muestra quien lleva el turno.

Método mostrarInformacionDeLaAplicacion: Muestra una ventana emergente donde se muestra la informacion del desarrollador de la aplicación, como las circunstancias bajo las que se creo la app.

Método crearPartida: Crea una partida nueva, siempre y cuando se cumplan las condiciones alli especificadas.

Método verificarDatosParaEmpezarPartida: Verifica si se cumplen las condiciones para empezar la partida.

Método activarUnPanel: Activa el panel según el parametro que le manden, siempre mantiene activo el panel del menu principal.