PRÁCTICA FINAL PROGRAMACIÓN 2

EL PUZZLE

Sebastian Serra Tirado
Cristobal Albertí Martinez

Programación 2 - Bartomeu Extrañ

INTRODUCCIÓN

Esta práctica consiste en diseñar un puzzle creado a traves de imagenes que deberá ser resuelto en un tiempo determinado .Al acabar el tiempo si el usuario no ha resuelto el puzle se muestra un mensaje que pone no lo ha conseguido, en caso contrario si se ha completado el puzzle se muestran por pantalla los puntos recibidos . Pero en ambos casos la partida acabará con la visualización de la ordenación correcta del puzzle. Además de la opción de nueva partida habrá otras opciones ,una de ellas es la de historial general que consiste en la lectura de un fichero en el que se habrán almacenado las diferentes partidas que se hayan realizado. Otra de las opciones es buscar partidas basándose en el nombre del jugador , que básicamente será una búsqueda dentro de un fichero.Otra de las funciones será la de cambio de directorio de imágenes, que se podrá acceder mediante un icono en la parte de arriba de la pantalla o mediante un menú desplegable , que servirá para seleccionar la imagen que uno quiera usar en el puzzle accediendo a diferentes directorios. Y por último pero no menos importante un botón para salir que simplemente cerrará la ventana.

DISEÑO

1-AdicionObjectOutputStream

Clase que extiende la casa madre ObjectOutputStream ,utilizada para poder escribir en fichero de texto utilizando ObjectOutputStream sin eliminar el contenido previo del fichero, para ello elimina la cabeza que se suele poner al escribir con esta clase.

2-PartidaObjetoEscritura

Clase que utilizamos para escribir en fichero de textos utilizando objectoutputstream. Dicha clase consta simplemente de 3 métodos muy sencillos y muy triviales, un constructor, escribirpartida y cerrafcihero.

3-PartidaObjetoLectura

Clase que se basa en la lectura de objetos partida de un fichero de texto, contiene 3 métodos ,un constructor , el método leerpartida y cerrarfichero.

4-Subimagen

Clase que define objetos subimagen, esta clase tiene un constructor cuya función será la instanciación de objetos subimagen. Además de dos métodos

setimagen para pasarle una imagen al objeto subimagen y getimagen que devuelve la imagen del objeto subimagen.

5-Partida

Clase creada para instanciar objetos clase partida .Cada partida tendrá dos atributos String y un atributo int. Y en esta clase habrá los getters y setters de dichos atributos además de un método toString .

6-Panel

Clase que extiende la clase madre JPanel. La utilizamos para definir paneles en los cuales dibujaremos imágenes que serán las piezas del puzzle .Al constructor le pasamos por parámetro una imagen y dos int que serán las filas y columnas. Utilizaremos los dos int para para saber las dimensiones de los paneles .Un método paintComponent que utilizaremos para dibujar la imagen en el panel. SeleccionarPanel y deseleccionarPanel que utilizaremos para remarcar el panel seleccionado.Y el método más importante redimesionarimagen que utilizamos para redimensionar la imagen que tenemos que usar en el puzzle utilizando el método getScaledInstance. Y por último cambiarimagen que sirve para cambiar la imagen de un panel

7-PanelSubimagen

Esta clase nos crea el tablero en el que jugaremos formados por paneles con subimagenes diferentes cada panel. El constructor de esta clase recibirá por parámetro un array de subimagenes y dos int que representan las filas y columnas de dicho tablero(panel) .El método inicialización es el método que se encarga de crear el tablero . Este método utiliza un random para colocar aleatoriamente los diferentes paneles con diferentes subimagenes en el panel principal, además de comprobar que no se repiten las subimagenes. En esta clase añadiremos un gestor de eventos para intercambiar paneles del panel general . Dicho método utiliza mousePressed y funciona detectando cuantos paneles has pulsado ,una vez se hayan pulsado dos paneles intercambia el primer panel seleccionado con el segundo utilizando un panel auxiliar. Además en este método cada vez que cambias un panel comprueba si la ordenación de los paneles concuerda con la solcuion.

8-Main

Esta clase es la clase principal de nuestro programa, se encarga de ejecutar el programa. Aqui definimos todos los componentes de la interfaz, incluyendo un gestor de eventos para la gestion de todos los botones que sobre todo sirve mucho para gestionar el cardlayout de diferentes contenedores. De entre ellos tenemos los del panel de botones, los de la toolbar que son representados con iconos y los del menu. Tambien tenemos el panel de visualizaciones donde se puede destacar que tenemos el apartado de jugar al puzle, que tiene una barra de progresion que indica el tiempo restante que varia dependiendo de las subdivisiones. Tambien en esta clase tenemos un metodo que nos ayuda con los recortes de la imagen dependiendo de las subdivisiones indicadas, este metodo recorta la imagen creando objetos subimagen con estos recortes y posteriormente creando un objeto panelsubimagen con estas subimagenes para luego añadir este panelsubimagen al panel de las partidas y poder utilizarlo para jugar.

Conclusiones

En este trabajo hemos aprendido muchos aspectos relacionados con la programación gráfica , utilizando una orientación de objetos. Hemos aprendido a afrontar problemas relacionados con la programación, es decir por una parte solucionar diferentes errores que nos iban surgiendo, que hemos solucionado investigando a fondo java y sus diferentes clases . Lo más difícil del trabajo a nuestro parecer es la gestión de eventos sobre todo en medio de la partida, además que hay muchos eventos a tener en cuenta y lo que más problemas nos ha dado es hacer que funcione el juego para cualquier imagen sin importar su dimensión.

https://youtu.be/tdTg_MjD7_8