



Trabajo Practico N°1 **Rompecabezas Deslizante**

Materia: Programacion III.

Profesor: Javier Marengo.

Patricia Bagnes.

Alumno: Ramos Aranibar Javier.



2017

Introducción

El presente trabajo practico tendrá como objetivo desarrollar el video juego Rompecabezas Deslizante, el cual consiste en una matriz compuesta por cuadrados, con una de ellas conmutable con la casilla contigua a la dirección del movimiento.

El objetivo es colocar los casilleros en un orden ya establecido.



Fig. 1



Fig. 2

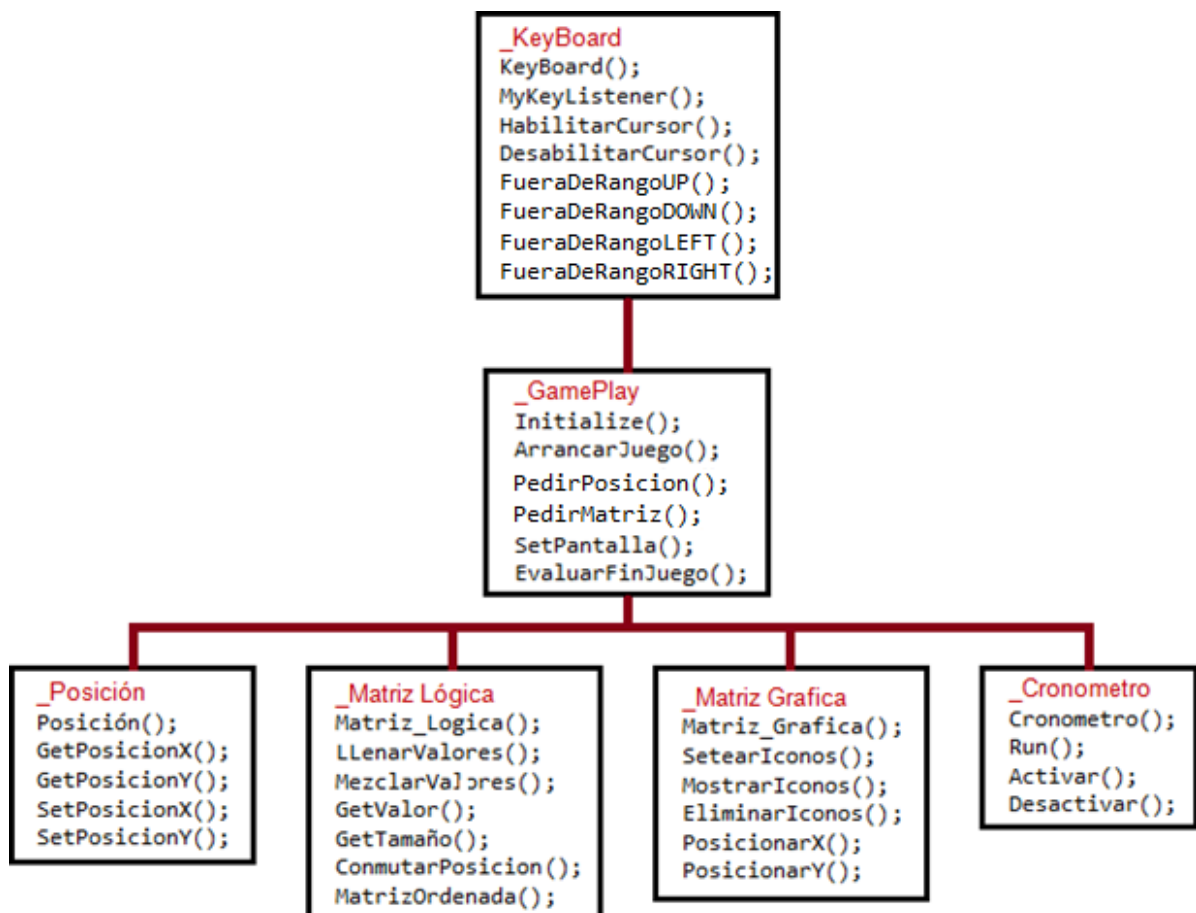


Fig. 3

Como podemos observar en la Fig. 1, el orden esta dada por números, por imágenes Fig. 2, o por colores Fig. 3. Este último, fue utilizado como criterio para desarrollar nuestro video juego. Haciendo alusión al juego más famoso del mundo, el Cubo Rubik.

Diagrama de Clases

El siguiente diagrama, representa todas las clases utilizadas y su relación entre ellas. A continuación se explicara con más detalles, las rutas tomadas.



El corazón del diagrama es GamePlay, de esta clase parte nuestro juego. Este crea todos los objetos que interactuarán con él para luego comunicarse con el resto.

Por Ejemplo: Ya inicializado todo, y a la espera de un movimiento...

- Se presiona la tecla/cursos arriba
- 1º Se verifica que el movimiento sea aceptable (Metodo: FueraDeRangoUP()), este necesita de dos datos, la longitud de la Matriz y la Posición en la que se encuentra, para analizarla, para ello hace uso de PedirMatriz() y PedirPosicion(), el cual retorna el objeto Matriz_Logica y el objeto Posicion, y con ellos sus métodos GetPosicionX(), GetPosicionY() (es Y lo que nos interés ya que se presiono UP) y GetTamaño().
- 2º Se analiza los datos y si no es aceptable no pasa nada. Si por el por contrario es un movimiento valido.
- 3º Se hace uso de PedirMatriz() y su método ConmutarPosicion(). Luego...
- 4º Pide SetPantalla() a GamePlay y este ejecuta EliminarIconos(), SetearIconos() y MostrarIconos() de MatrizGrafica. Por último...
- 5º Se ejecuta EvaluarFinJuego() de GamePlay y este ira al método MatrizOrdenada() de Matriz Logica.

Como se puede ver en este simple ejemplo, partiendo de KeyBoard todo se hace por intermedio de GamePlay.

Implementación

- Se opto por utilizar un único JFrame, ya que en caso de moverlo y luego crear otro, este nunca sabrá en qué posición quedo el anterior. Haciendo un juego poco atractivo a mí gusto.
- Sabiendo de antemano que se iba a usar el teclado como cursor y no el mouse. Se omitió los JBoton, ya que si bien este podía ser usado para insertar las imágenes, me pareció poco estético que se seleccionara cuando el mouse se posicionaba sobre él. Luego de investigar un poco se opto por JLabel's. ya que me facilitaba su introducción y extracción sobre el JFrame.
- El juego de armar un conjunto de Números o Imágenes, me pareció trillado. Y siendo un seguidor del Cubo Mágico, opte por este modelo. Aparte, para el ojo común, un conjunto de colores, atrae más que números o imágenes.
- Al ser un juego simple, y para hacer mayor la atracción del usuario, se creó tres niveles de rompecabezas.
- Se hizo uso de un cronometro para registrar la jugada, y así dos o más jugadores puedan competir su destreza.