

# Práctica 2: Kinect Puzle

Alumno: José Antonio Larrubia García

Alumno: José Miguel Navarro Moreno

## **1. Introducción.**

En esta práctica hemos desarrollado una aplicación que consiste en la realización de un puzle, para ello el usuario deberá coger la piezas con las manos, moverlas y colocarlas en la posición correcta. El usuario tendrá un tiempo determinado para la realización del puzle, y una vez que se acabe el tiempo se le mostrara la puntuación que ha obtenido.

## **2. Como lo hemos abordado.**

Para realizar la aplicación lo primero que hicimos fue una clase que está formada por una imagen y un rectángulo, esta clase se contiene los métodos necesarios para poder cogerla, moverla, soltarla y dibujarla en un objeto `DrawingContext`. (Para más información ver archivo `RectImagen.cs` del código).

Una vez que tuvimos un objeto que pudiéramos coger y mover, añadimos en el archivo `MainWindow.xaml.cs` un evento de `InteractionStream` para detectar cuando la mano está abierta o cerrada, el código necesario lo saquemos de la referencia [1] y [5] y lo integramos en nuestro código.

Después hemos hecho la clase `puzzle` detallada en el archivo `Puzzle.cs`. Esta clase recibe una imagen en el constructor y se encarga de partirla, también crear tantos objetos de la clase `RectImagen` como se haya especificado con cada una de las partes de la imagen y se desordenan. Además incluye los métodos necesarios para poder coger y dejar las pizas del puzle, así como el de dibujarlo.

## **3. Uso de la aplicación.**

Cuando empieza la aplicación nos tiene que detectar el esqueleto, una vez que nos detecte te mostrara el tutorial de cómo funciona la aplicación, para pasar el tutorial habrá que pasar la mano por el círculo rojo de la esquina superior derecha.

Luego se procede a resolver el puzle cogiendo las piezas con las manos y colocándolas donde se desee.

Una vez que se acabe el tiempo del que se dispondrá para resolver el puzle se mostrara la puntuación obtenida.

Tutorial que sale al principio de la aplicación:



#### 4. Problemas encontrados.

- Al detectar si las manos están cerradas o abiertas no conseguíamos adaptar los códigos que encontramos que solucionamos añadiendo código del proyecto Depth-Basics-WPF que es necesario para el evento de InteractionStream y uniando y adaptando los códigos de la Referencia [1] y [5].
- Partir la imágenes para las piezas del puzle que solucionamos usando el código de Referencia [2].
- Transformar los trozos de la imagen partida en un objeto BitmapImagen para poder dibujarlo en un objeto DrawingContext que solucionamos con la Referencia [3].
- Kinect no detecta bien el cierre de la mano para posiciones bajas.

## 5. Referencias.

- [1] <http://dotneteers.net/blogs/vbandi/archive/2013/05/03/kinect-interactions-with-wpf-part-iii-demystifying-the-interaction-stream.aspx>
- [2] <http://stackoverflow.com/questions/13625891/cut-an-image-into-9-pieces-c-sharp>
- [3] <http://joe-bq-wang.iteye.com/blog/1685024>
- [4] [https://msdn.microsoft.com/es-es/library/system.windows.media.formattedtext\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.windows.media.formattedtext(v=vs.110).aspx)
- [5] <http://stackoverflow.com/questions/18729142/how-to-detect-open-closed-hand-using-microsoft-kinect-for-windows-sdk-ver-1-7-c>