

# 手指繪圖初體驗

王泓權

臺科大人工智慧研究中心 助理教授



010010111

# 手指繪圖初體驗

- ✔ 你就是控制器，使用 Kinect 感應器。
- ✔ 手指指尖為最接近感應器物體，在視窗顯示指尖移動的軌跡，就如同手指在空中繪畫一般



# 手指繪圖初體驗

- ✔ Step 01: 建立新專案-FingerDrawingTest
- ✔ Step 02: 建立使用者介面

```
<Window x:Class="DepthImageDisplay.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="480" Width="640" Loaded="Window_Loaded_1">
  <Grid>
    <Image x:Name="DepthImage"/>
  </Grid>
</Window>
```





# 手指繪圖初體驗

- ✔ Step 03: 初始化作業，登錄  
StatusChanged 事件及定義其事件處理常式  
KinectSensor\_StatusChanged
- ✔ Step 04: 串流初始化、啟動與停止作業



# 手指繪圖初體驗

```
//InitialStream函式
private void InitialStream()
{
    this._myKinect.DepthStream.Enable(); //要求Kinect感應器產生深度資料串流
    //建立WriteableBitmap物件及影像區域
    this._writeableBitmap = new WriteableBitmap(_myKinect.DepthStream.FrameWidth,
_myKinect.DepthStream.FrameHeight, 96, 96, PixelFormats.Gray16, null);
    this._imageRect = new Int32Rect(0, 0, _myKinect.DepthStream.FrameWidth,
_myKinect.DepthStream.FrameHeight);
    //指定Image控制項影像來源為WriteableBitmap物件,使Image控制項顯示內容隨
WriteableBitmap物件內容改變
    DepthImage.Source = this._writeableBitmap;
    //用影像資料像素資料大小定義暫存資料陣列長度
    _imageDataArray = new short[this._myKinect.DepthStream.FramePixelDataLength];
    //註冊Kinect_DepthFrameReady事件處理函式
    this._myKinect.DepthFrameReady += Kinect_DepthFrameReady;
    this._myKinect.Start(); //啟動Kinect感應器硬體
    //將DepthImage背景設為淡灰色
    for (int i = 0; i < this._imageDataArray.Length; i++)
    {
        this._imageDataArray[i] = 32767;
    }
}
```



# 手指繪圖初體驗

## 🟢 Step 05: 深度影格備妥事件處理常式

//影格備妥事件處理常式

```
private void Kinect_DepthFrameReady(object sender, DepthImageFrameReadyEventArgs e)
{
    using (DepthImageFrame frameData = e.OpenDepthImageFrame()) //取得傳遞的影格資料
    {
        if (frameData == null) //如果影格資料不存在,直接離開事件處理函式
        {
            return;
        }
        int minDepth = 4000; //預設最近深度
        int arrayIndex = 0; //最近深度所在陣列索引
        //將影格資料複製到暫存陣列
        short[] tempArray = new short[this._myKinect.DepthStream.FramePixelDataLength];
        frameData.CopyPixelDataTo(tempArray);
        for (int i = 0; i < tempArray.Length; i++)
        {
            int depthValue = tempArray[i] >> DepthImageFrame.PlayerIndexBitmaskWidth; //取得目前像素深度資料
            if (depthValue > 0 && depthValue < minDepth)
            {
                minDepth = depthValue;
                arrayIndex = i;
            }
        }
        this._imageDataArray[arrayIndex] = 0;
        this._writeableBitmap.WritePixels(this._imageRect, this._imageDataArray, frameData.Width * frameData.BytesPerPixel,
0);
    }
}
```



# 測試

