Class in Full Screen & Full Screen Test:

1. DisplayMode:   
   DisplayMode (int width, int height, int bitDepth, int refreshRate)  
   field: REFRESH\_RATE\_UNKNOWN: Value of the refresh rate if not known.

The DisplayMode class encapsulates the bit depth, height, width, and refresh rate of a GraphicsDevice. The ability to change graphics device's display mode is platform- and configuration-dependent and may not always be available (see GraphicsDevice.isDisplayChangeSupported()).

1. public void run():

If this thread was constructed using a separate Runnable run object, then that Runnable object's run method is called; otherwise, this method does nothing and returns.

Subclasses of Thread should override this method.

1. GraphicsEnvironment:

The GraphicsEnvironment class describes the collection of GraphicsDevice objects and Font objects available to a Java(tm) application on a particular platform. The resources in this GraphicsEnvironment might be local or on a remote machine. GraphicsDevice objects can be screens, printers or image buffers and are the destination of Graphics2D drawing methods. Each GraphicsDevice has a number of GraphicsConfiguration objects associated with it. These objects specify the different configurations in which the GraphicsDevice can be used.

1. GraphicsDevice:   
   The GraphicsDevice class describes the graphics devices that might be available in a particular graphics environment. These include screen and printer devices. Note that there can be many screens and many printers in an instance of GraphicsEnvironment. Each graphics device has one or more GraphicsConfiguration objects associated with it. These objects specify the different configurations in which the GraphicsDevice can be used.
2. setUndecorated(Boolean undecorated): //method of JFrame  
   undecorated - true if no frame decorations are to be enabled; false if frame decorations are to be enabled  
   Disables or enables decorations for this frame.

This method can only be called while the frame is not displayable. To make this frame decorated, it must be opaque and have the default shape, otherwise the IllegalComponentStateException will be thrown. Refer to Window.setShape(java.awt.Shape), Window.setOpacity(float) and Window.setBackground(java.awt.Color) for details

Learn in Full Screen:

Class SimpleScreenManger 中先宣告GraphicsDevice，這個變數將會存放正在使用的環境下的status. Constructer中經由GraphicsEnvironment取得local的GraphicsEnvironment然後將得到的資訊的default狀況給GraphicsDevice的變數，這樣一來之前宣告的GraphicsDevice就得到了loacl環境下的資訊。

自定的SetFullScreen有兩個參數，DisplayMode決定顯示的方式，JFrame則會將要set的frame傳進來，GraphicsDeivce的setFullScreenWindow將這個JFrame的顯示方式設定成已全螢幕呈現，接下來的判斷式判斷若顯示方式不為空且此GraphicsDevice可備更改設定，則設定成我們自定的樣子。

RestoreScreen將使用的資源還給電腦，已此離開全螢幕。

Learn in FullScreenTest:

實作runnable 這個 interface，override run() 這個method，DisplayMode 將顯示模式的參數傳入，method內將前景設為白色、後景設為藍色，在進入full screen模式後，用thread.sleep()將程式停止5000 millisecond 若是上述程序失敗，則用InterruptedException接住例外，最後歸還資源給電腦。  
  
Learn in ImageTest( SimpleScreen跟 ImagTest 的ScreenManager 幾乎一樣，所以只列出ImageTest):

loadImage() 這個method吃檔名參數後回傳Image檔，loadImages()將需要的圖片載入，載入後將imagesLoaded設為true，imagesLoaded保護程式不會將空的圖檔repaint上去，因此不會產生錯誤。

Graphics2D g2 = (Graphics2D)g;

g2.setRenderingHint( RenderingHints.KEY\_TEXT\_ANTIALIASING, RenderingHints.VALUE\_TEXT\_ANTIALIAS\_ON);

上述兩行將之後繪出的圖設定為反鋸齒的status。

Sprite1：

Sprite這個class描述player，player物件有自己的animation 位置座標，以及運動的速度。

Animation描述Sprite的運動狀況，arrayList宣告的frame存放 AnimFrame這個自定義的資料型別，AnimFrame存放圖片以及該圖的存在時間，frame存有一系列的AnimFrame因此可製造出動畫的效果。

Synchronized使用時，需指定一個物件，系統會Lock此物件，當程式進入Synchrnoized區塊或Method時，該物件會被 Lock，直到離開Synchronized時才會被釋放。在Lock期間，鎖定同一物件的其他Synchronized區塊，會因為無法取得物件的 Lock而等待。待物件Release Lock後，其他的Synchronized區塊會有一個取得該物件的Lock而可以執行。

public synchronized void start() //在此物件備宣告時進入，並設定animTime = 0, currentFrameIndex = 0.

public synchronized void update(long elapsedTime) //進行一個frame的運作

ScreenManager in Sprite1:

BufferImage: The BufferedImage subclass describes an [Image](http://docs.oracle.com/javase/7/docs/api/java/awt/Image.html) with an accessible buffer of image data. A BufferedImage is comprised of a [ColorModel](http://docs.oracle.com/javase/7/docs/api/java/awt/image/ColorModel.html) and a [Raster](http://docs.oracle.com/javase/7/docs/api/java/awt/image/Raster.html) of image data. The number and types of bands in the [SampleModel](http://docs.oracle.com/javase/7/docs/api/java/awt/image/SampleModel.html) of the Raster must match the number and types required by the ColorModel to represent its color and alpha components. All BufferedImage objects have an upper left corner coordinate of (0, 0). Any Raster used to construct a BufferedImage must therefore have minX=0 and minY=0.

This class relies on the data fetching and setting methods of Raster, and on the color characterization methods of ColorModel.