> VideoRenderer的目的是让链接定义他们自己的渲染行为，这个是通过回调产生的，这个方法同样体统了一个创建GUI的方法，用来创建GUI渲染器在各种各样的平台上面。 需要注意的是，frame只能通过native层进行构建。

```
  //这是I420的一个对象的类，I420是视频编码的一种方式
  public static class I420Frame {
    public final int width;
    public final int height;
    public int[] yuvStrides;      //信号的频幅
    public ByteBuffer[] yuvPlanes;    //平面的色差信号
    public final boolean yuvFrame;    //是否有帧的色差信号
    // Matrix that transforms standard coordinates to their proper sampling locations in
    // the texture. This transform compensates for any properties of the video source that
    // cause it to appear different from a normalized texture. This matrix does not take
    // |rotationDegree| into account.
    //抽样矩阵
    //将标准坐标转换为纹理中适当采样位置的矩阵。该转换补偿视频源的任何属性，使其与标准化纹理不同。这个矩阵不采取rotation
    public final float[] samplingMatrix;
    //结构Id
    public int textureId;
    // Frame pointer in C++.指针
    private long nativeFramePointer;

    // rotationDegree is the degree that the frame must be rotated clockwisely
    // to be rendered correctly.
    //旋转的角度应该是以顺时针的角度为标准
    public int rotationDegree;

    /**
     * Construct a frame of the given dimensions with the specified planar data.
     */
    //构造方法，并且旋转的角度必须是90的整数倍
    I420Frame(int width, int height, int rotationDegree, int[] yuvStrides, ByteBuffer[] yuvPlanes,
        long nativeFramePointer) {
      this.width = width;
      this.height = height;
      this.yuvStrides = yuvStrides;
      this.yuvPlanes = yuvPlanes;
      this.yuvFrame = true;
      this.rotationDegree = rotationDegree;
      this.nativeFramePointer = nativeFramePointer;
      if (rotationDegree % 90 != 0) {
        throw new IllegalArgumentException("Rotation degree not multiple of 90: " + rotationDegree);
      }
      // The convention in WebRTC is that the first element in a ByteBuffer corresponds to the
      // top-left corner of the image, but in glTexImage2D() the first element corresponds to the
      // bottom-left corner. This discrepancy is corrected by setting a vertical flip as sampling
      // matrix.
      // clang-format off
      samplingMatrix = new float[] {
          1,  0, 0, 0,
          0, -1, 0, 0,
          0,  0, 1, 0,
          0,  1, 0, 1};
      // clang-format on
    }

    /**
     * Construct a texture frame of the given dimensions with data in SurfaceTexture
     */
    //另一个构造方法，只是需要手动传入一个矩阵
    I420Frame(int width, int height, int rotationDegree, int textureId, float[] samplingMatrix,
        long nativeFramePointer) {
      this.width = width;
      this.height = height;
      this.yuvStrides = null;
      this.yuvPlanes = null;
      this.samplingMatrix = samplingMatrix;
      this.textureId = textureId;
      this.yuvFrame = false;
      this.rotationDegree = rotationDegree;
      this.nativeFramePointer = nativeFramePointer;
      if (rotationDegree % 90 != 0) {
        throw new IllegalArgumentException("Rotation degree not multiple of 90: " + rotationDegree);
      }
    }

    //获取宽和高
    public int rotatedWidth() {
      return (rotationDegree % 180 == 0) ? width : height;
    return (rotationDegree % 180 == 0) ? width : height;
  }

  public int rotatedHeight() {
    return (rotationDegree % 180 == 0) ? height : width;
  }

  @Override
  public String toString() {
    return width + "x" + height + ":" + yuvStrides[0] + ":" + yuvStrides[1] + ":" + yuvStrides[2];
  }
}
```

```
  //释放掉所有frame的做法
  public static void renderFrameDone(I420Frame frame) {
    frame.yuvPlanes = null;
    frame.textureId = 0;
    if (frame.nativeFramePointer != 0) {
      releaseNativeFrame(frame.nativeFramePointer);
      frame.nativeFramePointer = 0;
    }
  }
```

```
long nativeVideoRenderer;

//构造方法，需要传进来一个callbacks
public VideoRenderer(Callbacks callbacks) {
    nativeVideoRenderer = nativeWrapVideoRenderer(callbacks);
  }
```

```
  //销毁掉所有的方法
  public void dispose() {
    if (nativeVideoRenderer == 0) {
      // Already disposed.
      return;
    }

    freeWrappedVideoRenderer(nativeVideoRenderer);
    nativeVideoRenderer = 0;
  }
```

```
//native层的初始化的方法
private static native long nativeWrapVideoRenderer(Callbacks callbacks);
```

```
//销毁
private static native void freeWrappedVideoRenderer(long nativeVideoRenderer);
//释放
private static native void releaseNativeFrame(long nativeFramePointer);
```