

理解 DrawEngine 中的坐标系

在 TdrawEngine 中，我们总是会看到这类声明

```
procedure DrawBox(r: TDERect; COLOR: TDECOLOR; LineWidth: TDEFLOAT); overload;  
procedure DrawBoxInScene(r: TDERect; COLOR: TDECOLOR; LineWidth: TDEFLOAT); overload;
```

InScene 后缀究竟有什么含义？

In Scene 分开来解释，In=在，Scene 场景，在场景中。这是一种 DrawEngine 自定义的规范坐标系。

在我们开发图形程序时，比如 FMX-Canvas 的坐标系是继承关系，当我们在一个 TpaintBox->OnPaint 画图时，x 为 0 坐标计算都是 TpaintBox 的 Left，最后 x 会被层层转换成 d2d,opengl,这些绘图 api 的绝对坐标系。

InScene 的后缀也相当于这种继承关系。

DrawEngine 本身只有两种坐标系，分别 Screen 和 Scene。

当我们使用 DrawBox 绘图时，坐标计算按 Screen 体系工作，如果使用了 DrawEngine for FMX 接口，Screen 就等同于 FMX 的控件坐标系。

当我们使用 DrawBoxInScene 绘图时，坐标系就会按 DrawEngine 内置的 Offset+Scale 两个参量进行换算，重新计算出一套 Screen 的坐标出来来画图。

以 FMX 为例，我们在数十万频率的绘制调用过程中，如果去创建数以万计的 Tcontrol 实现坐标系统一化显然很蠢，也不现实，因为 FMX 是作为 UI 界面使用图形框架，并非专业图形框架。InScene 的作用更偏向解决场景中绘图的问题，我们不必花心思去构建极易出错的坐标系系统，直接在后面加个 InScene 即可统一化坐标系，并且还具备剪裁、剪切，优化的功能，如果开发游戏，大规模图片绘制，InScene 体系会收到很好的效果。

在 zAI 的建模工具，手机麻将游，手机飞机游，坦克游中，InScene 坐标系被大量应用。

当我们理解了 InScene 的含义后，试想一下，我要在屏幕一千码的范围外画一张图，我们不必再去纠结坐标的来回转换，在 API 后面加上 InScene 即可，是不是很方便？

当我们理解了 InScene 的含义以后，再来理解 DrawEngine 中内置的一堆结构体

TDERect, Trect 的浮点版

TDEVec, TPoint 的浮点版

TDE4V, Trect 的浮点版，多了一个转动角度

TV2Rect4, 由 4 个 TPoint 的浮点版组成的 4 顶点框体结构

Tvec2 等同于 TDEVec

TrectV2 等同于 TDERect

这些坐标系，更容易找到来龙去脉，API 一目了然。

2019-7

By.qq600585