

HenCoder Plus 讲义

范围裁切和几何变换

Canvas 的范围裁切

- clipRect()
- clipPath() clipPath() 切出来的圆为什么没有抗锯齿效果？因为「强行切边」
- clipOutRect() / clipOutPath()

Canvas 的几何变换

- translate(x, y)
- rotate(degree)
- scale(x, y)
- skew(x, y)

重点：Canvas 的几何变换方法参照的是 View 的坐标系，而绘制方法（drawXxx()）参照的是 Canvas 自己的坐标系。

关于多重变换：

Canvas 的变换方法多次调用的时候，由于 Canvas 的坐标系会整体被变换，因此当平移、旋转、放缩、错切等变换多重存在的时候，Canvas 的变换参数会非常难以计算，因此可以改用倒序的理解方式：

将 Canvas 的变换理解为 Canvas 的坐标系不变，每次变换是只对内部的绘制内容进行变换，同时把 Canvas 的变换顺序看作是倒序的（即写在下面的变换先执行），可以更加方便进行多重变换的参数计算。

Matrix 的几何变换

- preTranslate(x, y) / postTranslate(x, y)

- `preRotate(degree)` / `postRotate(degree)`
- `preScale(x, y)` / `postScale(x, y)`
- `preSkew(x, y)` / `postSkew(x, y)`

其中 `preXxx()` 效果和 Canvas 的准同名方法相同, `postXxx()` 效果和 Canvas 的准同名方法顺序相反。

注意

如果多次绘制时重复使用 Matrix, 在使用之前需要用 `Matrix.reset()` 来把 Matrix 重置。

使用 Camera 做三维旋转

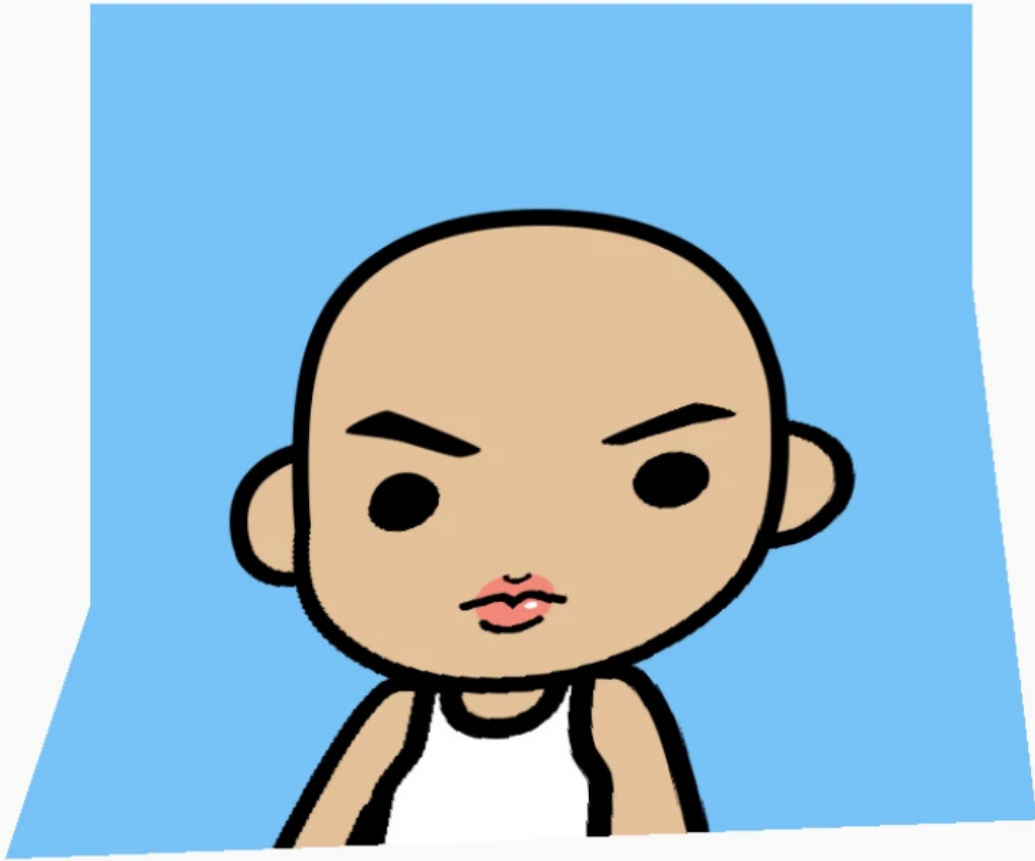
- `rotate()` / `rotateX()` / `rotateY()` / `rotateZ()`
- `translate()`
- `setLocation()`

其中, 一般只用 `rotateX()` 和 `rotateY()` 来做沿 x 轴或 y 轴的旋转, 以及使用 `setLocation()` 来调整放缩的视觉幅度。

对 Camera 变换之后, 要用 `Camera.applyToCanvas(Canvas)` 来应用到 Canvas。

`setLocation()`

这个方法一般前两个参数都填 0, 第三个参数为负值。由于这个值的单位是硬编码写死的, 因此像素密度越高的手机, 相当于 Camera 距离 View 越近, 所以最好把这个值写成与机器的 density 成正比的一个负值, 例如 `-6 * density`。



问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！