1. 支持自定义model，以资源方式定义构型工具导出的model文件，render piece的定义中可以指定model；

model同样需要加上progress属性，按进度获取不同的顶点坐标，可以通过设置初始model和结束model，中间过程的顶点通过差值得到，前提两个model的顶点数必须相同；

用于实现各种模型形变效果，比如挤压，碰撞，曲面等等；

1. 任意层次布局对象，当前仅支持默认的三个布局对象（section， line， char），并且定义与其中的render对象默认位于其中。

有两步可以做：第一步可以定义render对象依赖不同于定义其中的布局对象作为布局，比如在char中定义的render对象依赖于line或者section而非char；第二步，可以定义新的布局对象扩展布局层次，可以自定义新的非默认的布局对象，并可在其上加如依赖于它的render或布局对象（可能没有必要）；

用于方便实现（现有也可以做，只是坐标变幻和繁琐）画布范围的移动效果，比如某字的一部分从段落的左上移动到右下；

1. glyph资源支持自定义纹理，以自定义纹理的方式定义若干纹理，每个纹理关联到不同的字符上，用于取代字体glyph。

用于实现每个字符用关键帧改变外观。

1. 布局缓存和后处理。用于超过单个字符之外更大粒度的整体变化效果。比如整个段落整体的高斯模糊，或者整行的整体高斯模糊；甚至整行的形变，比如整行弯曲。实现方式是，posttotext中并非直接渲染到当前的gl context之上，而是创建frame buffer，并渲染到其上，然后将frame buffer作为纹理进入下一个场景进行后处理。
2. 自定义传入shader的变量值，自定义传入shader的变量。
3. 版本控制。后续版本Postto text和字效的版本兼容问题需要处理，特别是开始做字效下发的时候。
4. 更安全的加载。虽然字效包并非任意输入，也是存在各种原因导致字效包中的文件（特别是package.json）改变导致加载错误的问题，出现这样的错误时，尽量不要崩溃。
5. 性能优化，比如纹理的缓存，按需加载等等。
6. 现有配置方式实在不能支持的效果（典型的是需要在帧间共享变量，比如复杂的运动轨迹控制），发展到很后面，可以考虑引入脚本，以方便支持调用opengl接口的脚本为好。