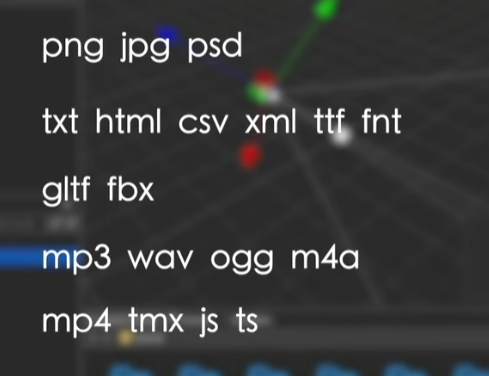
支持格式



调整摄像机】 ctrl+shift+F

吸附】按住V可以选中对应顶点 进行移动吸附

## 装饰器：

*const* { ccclass, property,executeInEditMode } = \_decorator;

\_decorator

不运行条件下执行组件

@executeInEditMode(true)

挂在的组件显示的名:

@ccclass('Launch')

公开组件到节点上

  @property

    public foo = 10;

    @property(Label)

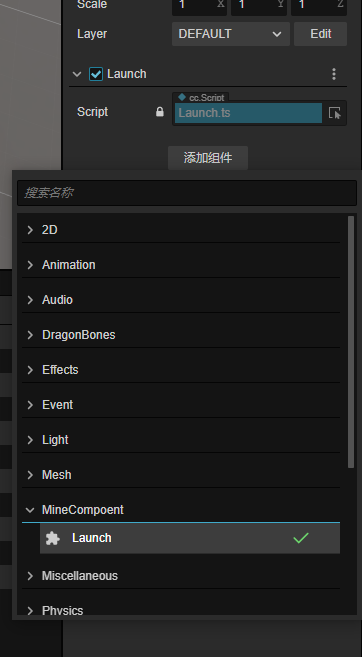
    public label: Label;

强依赖组件

@requireComponent(MeshRenderer)//强依赖组件

挂在组件显示的分类

@menu("MineCompoent/Launch")



## Tween

*var* targetTween: Tween<*any*> = tween(uiOpacity).to(1, { opacity: 2 }).delay(*delayTime*).start().call(*callBack*);

1. Tween类需要跟类型
2. 返回对应的Tween<T>
3. .start().delay()这些函数会先放在一个队列，然后在渲染层执行，

    export *enum* TweenEase{

        "linear", "smooth", "fade", "constant", "quadIn", "quadOut", "quadInOut", "quadOutIn", "cubicIn", "cubicOut", "cubicInOut", "cubicOutIn", "quartIn", "quartOut", "quartInOut", "quartOutIn", "quintIn", "quintOut", "quintInOut", "quintOutIn", "sineIn", "sineOut", "sineInOut", "sineOutIn", "expoIn", "expoOut", "expoInOut", "expoOutIn", "circIn", "circOut", "circInOut", "circOutIn", "elasticIn", "elasticOut", "elasticInOut", "elasticOutIn", "backIn", "backOut", "backInOut", "backOutIn", "bounceIn", "bounceOut", "bounceInOut", "bounceOutIn"

    }

*var* targetTween: Tween<*any*> = tween(uiOpacity).delay(*delayTime*).to(*showTime*, { opacity: 0 }, { easing: *tweenEasing* }).call(*callBack*).start();

Tween 的调用顺序 delay – to -call - start

## 生命周期

onLoad是代码层 相当于unity的awake

start 是渲染层

## JSon

Json转换对象内不能添加字段 不然会无效

JSon和Map或者手写的字典等类型不能直接转换 只能通过{key:value}的形式

## 物理碰撞系统

基础碰撞使用的BOX2D封装，其中有物理算法，如果项目不需要物理效果就自己写

使用矩形碰撞检测实现碰撞事件，然后维护触发的数组与上次碰撞的结果对比，如果发生了变化再派发碰撞的的事件

## 贴图纹理

在位图的使用中

有透明通道的贴图会比没有透明通道的贴图多占用一些内存

加载到内存中的大小取决于这张图片的像素的多少，而不是文件的大小

游戏中常用的贴图压缩技术主要是针对贴图的文件大小进行压缩，从而减少加载到内存中所占用的内存大小。这些压缩技术会压缩贴图文件中的颜色信息、透明度信息、纹理坐标等数据。一些常见的贴图压缩技术包括：

1. DXT压缩：主要用于压缩RGBA格式的贴图，压缩比例通常为4:1到6:1。
2. PVRTC压缩：主要用于移动平台上的贴图压缩，通常可以实现更高的、压缩比例，但会导致贴图在近距离观察时出现锯齿。
3. ETC压缩：也是用于移动平台上的贴图压缩，与PVRTC相比，ETC压缩可以在保证高压缩比的同时避免出现锯齿。

这些贴图压缩技术通常不会直接压缩像素，而是通过压缩贴图文件中的颜色、透明度等信息来减小贴图文件的大小，从而减少贴图在加载到内存中时所占用的内存大小。

## 加载UI和图集

1. 加载UI一个界面为不多余三张图集的界面
2. 没有直接显示的界面可以先不放在图集中，但是如果图集足够装得下 可以放在节点中显隐操作 ，否则可以动态加载大图或者另一个界面挂在到当前界面
3. 整个游戏只出现一次的界面，例如新手引导 新用户注册这种 最好单独一张图集
4. 特效和spine这种大图也拆在UI外边动态加载
5. 首页进入要缩短时间，进入游戏的时候将可以剔除的资源全都干掉

## 蒙皮骨骼

Mesh网格可以合批，蒙皮网格无法合批

## 声音

1. 声音过短的音效在抖音小游戏无法播放
2. Ios出现音效无法播放的问题

## 内存

一个imageAsset 占用的内存是一张图集的内存

新手引导这种占用单独内存图集或者视频的 打开和关闭就销毁UI