掘金

微信小游戏开发技巧

黄剑鑫 | 高级工程师 @ 腾讯互娱

TGideas

掘金

沙因Sign

DOLO

弹跳小卜

精灵之息

TGideas

掘金

沙因Sign

DOLO

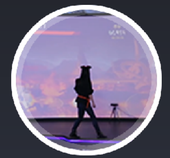
弹跳小卜

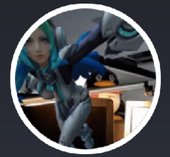
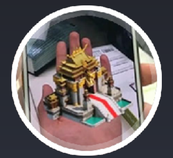
精灵之息

TGideas

掘金

TGideas 前端团队，负责腾讯互娱动漫游戏的 Web 开发支持
不局限于线上线下的 Web 互动体验制作




跳跃吧酷比


纪念碑谷


拳皇KO

TGideas

掘金

TGideas

掘金

小游戏制作技巧

对「小游戏」入门知识的普及

- 小游戏的基础概念
- 小游戏的架构设计
- 小游戏的部分算法简介
- 小游戏的微信API

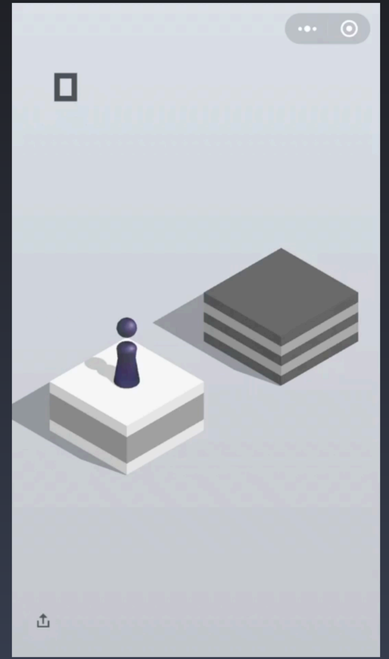
TGideas

掘金

1 小游戏的基础概念

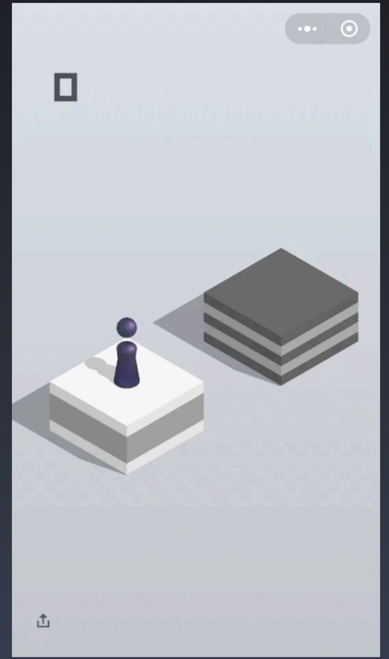
TGideas

掘金



TGideas

掘金



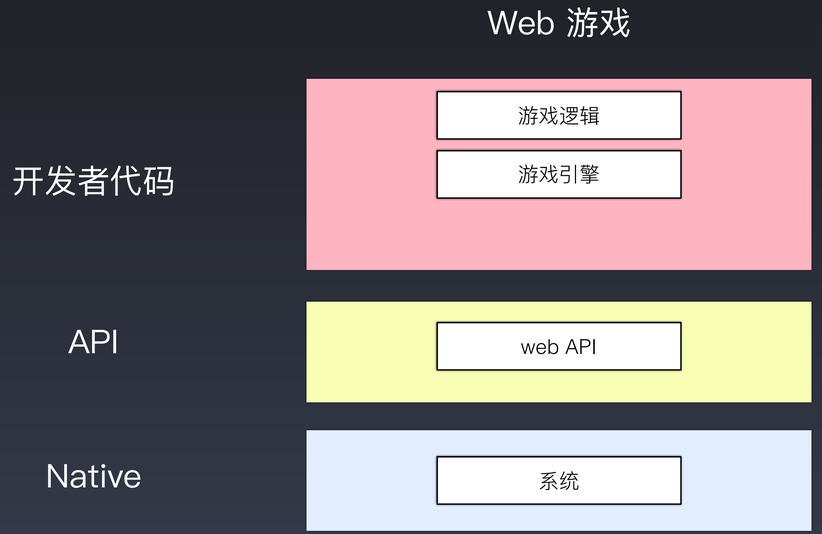
TGideas

掘金

「微信小游戏平台是一个不同于浏览器的运行环境」

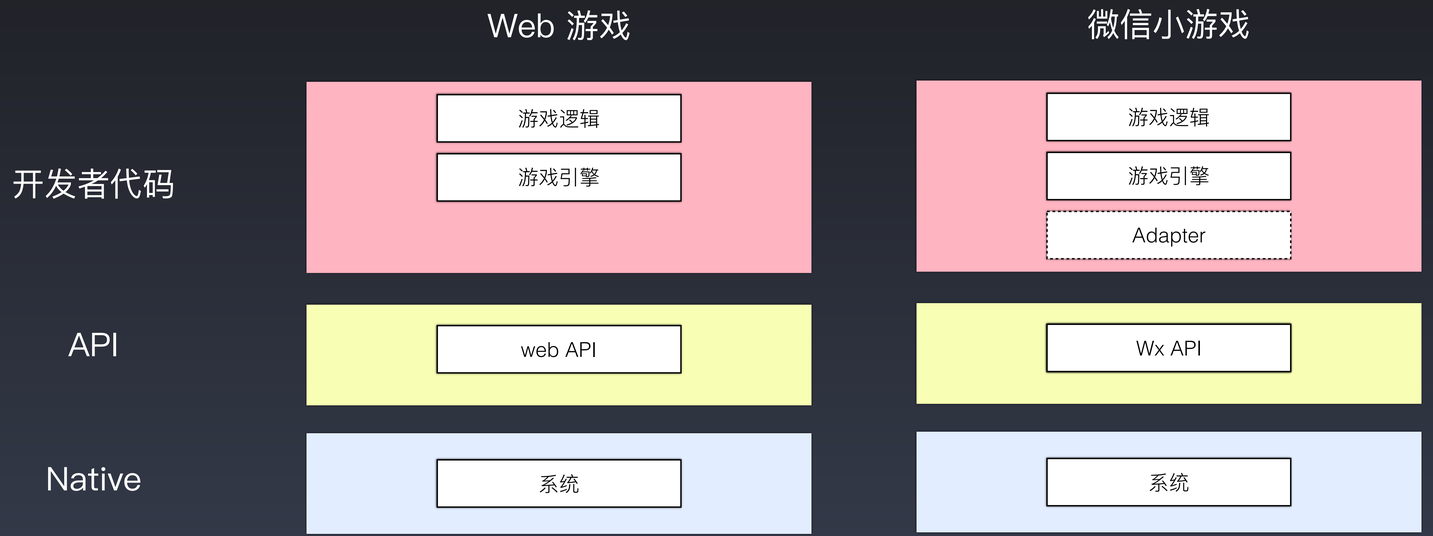
TGideas

掘金



TGideas

掘金


Web 游戏
微信小游戏
开发者代码
API
Native
游戏逻辑
游戏引擎
Adapter
web API
Wx API
系统
系统

TGideas

开发者代码

逐帧计算并渲染
requestAnimationFrame

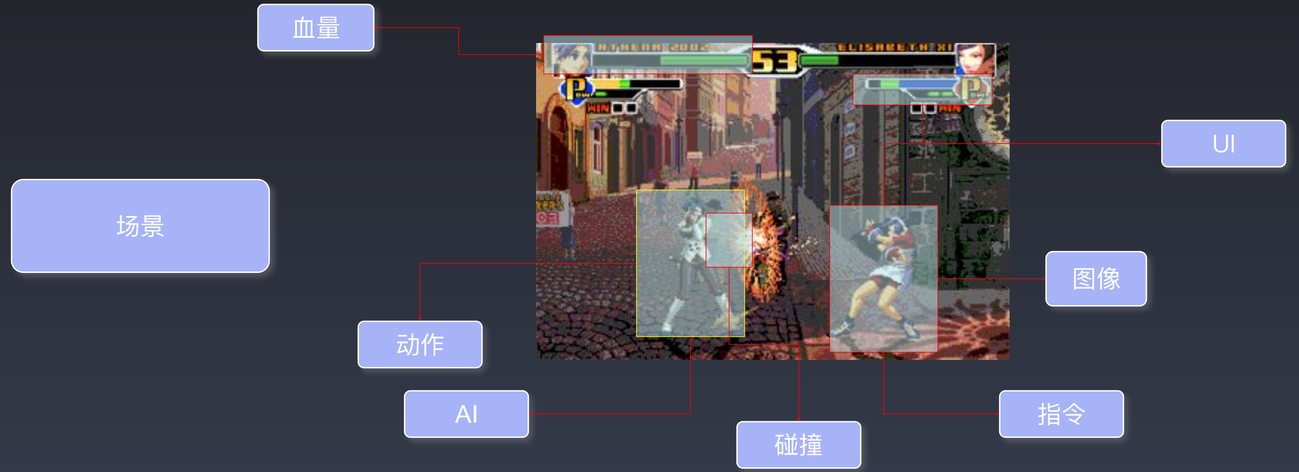
画布canvas内元素呈现

掘金

场景

TGideas

画布canvas内元素呈现



运行效果

运行效果

掘金

2 小游戏的架构设计

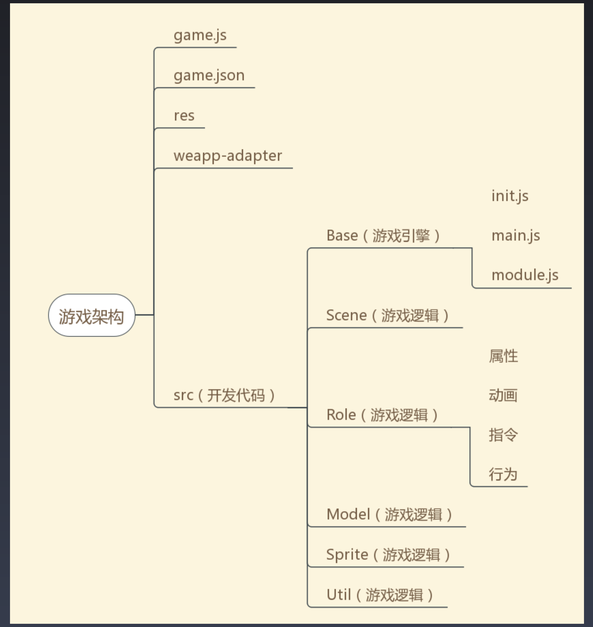
TGideas

掘金

TGideas

掘金

TGideas


游戏架构
game.js
game.json
res
weapp-adapter
src (开发代码)
Base (游戏引擎)
init.js
main.js
module.js
Scene (游戏逻辑)
Role (游戏逻辑)
属性
动画
指令
行为
Model (游戏逻辑)
Sprite (游戏逻辑)
Util (游戏逻辑)

掘金

简易的小游戏结构

TGideas

简易的小游戏结构

掘金

TGideas

简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件

简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件
weapp-adapter	使用wx API模拟BOM和DOM

简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件
weapp-adapter	使用wx API模拟BOM和DOM
res	游戏资源，图片，音频等……

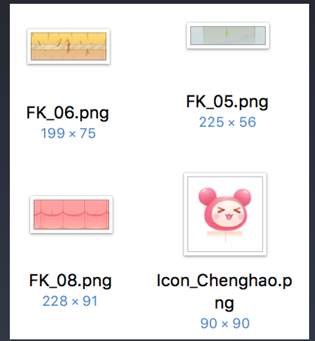
简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件
weapp-adapter	使用wx API模拟BOM和DOM
res	游戏资源，图片，音频等……

掘金

简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件
weapp-adapter	使用wx API模拟BOM和DOM
res	游戏资源，图片，音频等……

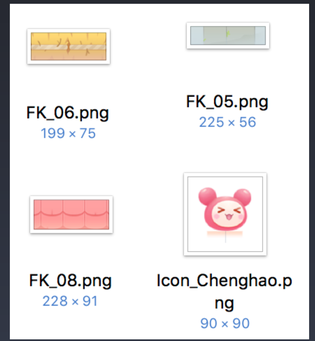


TGideas

掘金

简易的小游戏结构

game.js	小游戏入口文件
game.json	配置文件
weapp-adapter	使用wx API模拟BOM和DOM
res	游戏资源，图片，音频等……
src	



TGideas

简易的小游戏结构

game.js

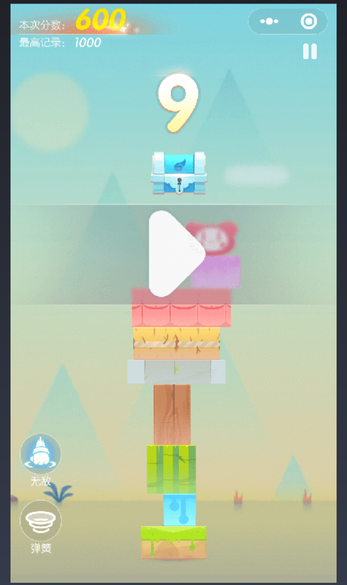
game.json

weapp-adapter

res

src

掘金



简易的小游戏结构

Base	模块管理脚本
Scene	场景模块
Role	角色模块
Model	模版模块（无具体属性及行为的角色）
Sprite	材质模块
Util	一些当前游戏所需要的脚本

TGideas

掘金

「较复杂的小游戏推荐使用一些专业的游戏引擎进行制作」

TGideas

「较复杂的小游戏推荐使用一些专业的游戏引擎进行制作」

掘金

TGideas

「较复杂的小游戏推荐使用一些专业的游戏引擎进行制作」

引擎	设计理念
Egret Engine	有一整套完整的工作流。老牌引擎，社区大，扩展工具多，支持3d。
LayaAir Engine	主打性能的游戏引擎。较新的热门引擎，社区活跃度高。3d支持最佳。
Cocos Creator	老牌引擎的底蕴，用户多，成功产品多（基于底层框架）。社区活跃，刚开始支持3d。
PixiJS	高性能的webgl模式2d渲染引擎，api简单。
CreateJS	基于HTML5开发的一套模块化的库和工具。较多的扩展库，有中文主页。
Phaser	快速开源的轻量级2d游戏引擎。
Three.js	知名的3d引擎，在webGL方面是第一优先选择，但不是个纯粹的游戏引擎。

「较复杂的小游戏推荐使用一些专业的游戏引擎进行制作」

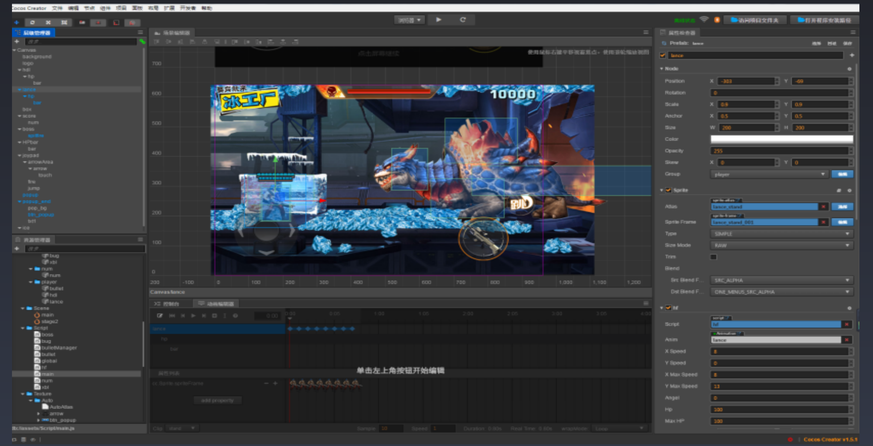
引擎	设计理念
Egret Engine	有一整套完整的工作流。老牌引擎，社区大，扩展工具多，支持3d。
LayaAir Engine	主打性能的游戏引擎。较新的热门引擎，社区活跃度高。3d支持最佳。
Cocos Creator	老牌引擎的底蕴，用户多，成功产品多（基于底层框架）。社区活跃，刚开始支持3d。
PixiJS	高性能的webgl模式2d渲染引擎，api简单。
CreateJS	基于HTML5开发的一套模块化的库和工具。较多的扩展库，有中文主页。
Phaser	快速开源的轻量级2d游戏引擎。
Three.js	知名的3d引擎，在webGL方面是第一优先选择，但不是个纯粹的游戏引擎。

示例：cocos creator制作游戏实际画面

掘金

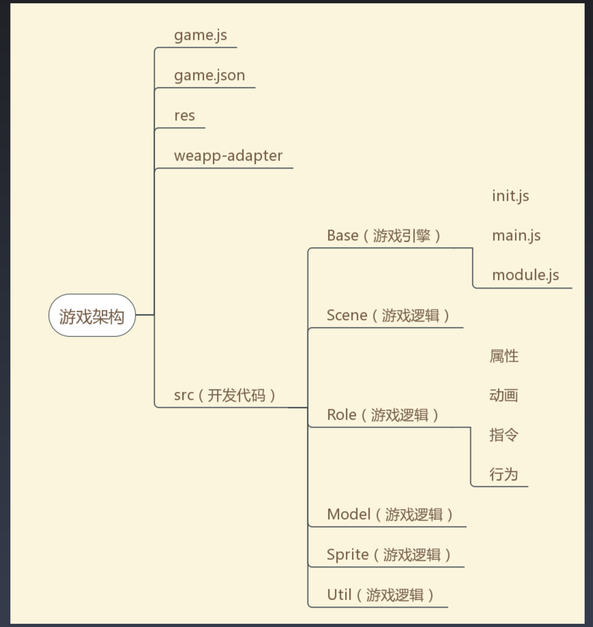
场景

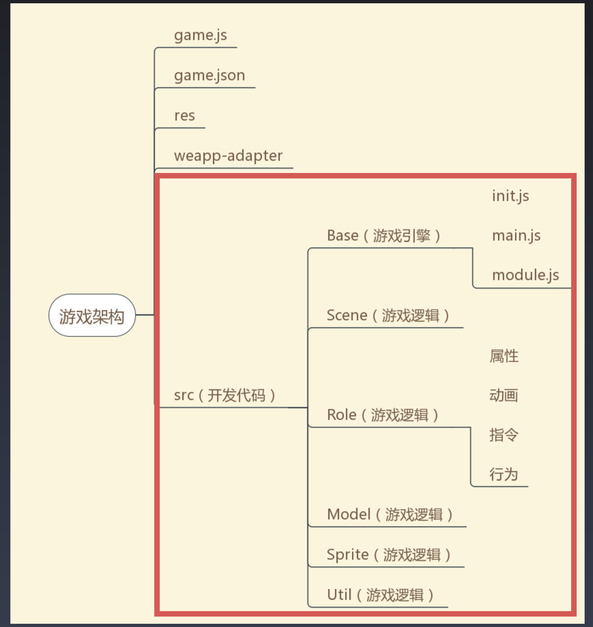
实体



游戏逻辑

TGideas


游戏架构
game.js
game.json
res
weapp-adapter
src (开发代码)
Base (游戏引擎)
init.js
main.js
module.js
Scene (游戏逻辑)
Role (游戏逻辑)
属性
动画
指令
行为
Model (游戏逻辑)
Sprite (游戏逻辑)
Util (游戏逻辑)


game.js
game.json
res
weapp-adapter
游戏架构
src (开发代码)
init.js
Base (游戏引擎)
main.js
module.js
Scene (游戏逻辑)
属性
动画
Role (游戏逻辑)
指令
行为
Model (游戏逻辑)
Sprite (游戏逻辑)
Util (游戏逻辑)

模块管理脚本

掘金

TGideas

模块管理脚本

Base init.js

main.js

module.js

...

掘金

TGideas

Base　　init.js　　模块管理脚本

main.js　　游戏入口脚本

module.js

...

```
requestAnimationFrame(render);
render();
```

调用main.js

模块管理脚本

Base

init.js

main.js

module.js

...

掘金

TGideas

模块管理脚本

Base	init.js	游戏逻辑脚本
	main.js	调用各个场景脚本
	module.js	场景脚本中调用角色脚本
	...	逻辑脚本控制整个游戏状态

模块管理脚本

Base

init.js

main.js

module.js

...

掘金

TGideas

模块管理脚本

Base

init.js

main.js

module.js

...

模块加载器
根据实际需要封装对应代码管理模块依赖

```
module.add();
module.load();

define(module);
require(module);
```

场景模块

掘金

TGideas

场景模块

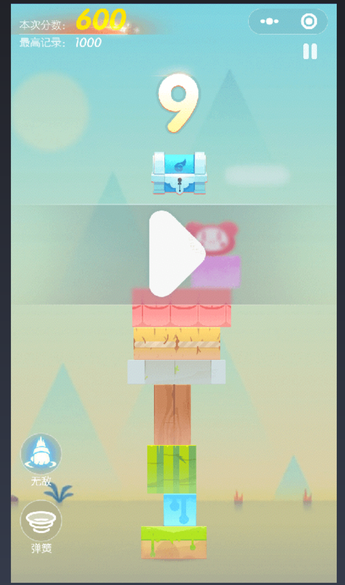
Scene

调用当前场景
需要的角色及背景

掘金

TGideas

掘金



场景模块

```
role1=module.load(role);

scene.init(backgroud);
scene.load(role1);
```

调用逻辑脚本main.js中的渲染函数，开始游戏

TGideas

角色模块

掘金

TGideas

角色模块

Role

通过model脚本
创建一个实体
object

掘金

TGideas

掘金

角色模块

Role

通过model脚本
创建一个实体
object

属性	x, y, hp, speed, sprite…
动画	调用动画及贴图脚本
指令	监控输入数据，并执行相应的行为
行为	通过控制属性，动画，来呈现

TGideas

掘金

角色模块

Role

通过model脚本
创建一个实体
object

属性　x, y, hp, speed, sprite…

动画　调用动画及贴图脚本

指令　监[illegible]据，并执行相应的行为

行为　通过控制属性，动画，来呈现

TGideas

掘金

角色模块

Role

通过model脚本
创建一个实体
object

属性	x, y, hp, speed, sprite…
动画	调用动画及贴图脚本
指令	监控输入数据，并执行相应的行为
行为	通过控制属性，动画，来呈现

TGideas

角色模块

属性

动画

指令

行为

掘金

TGideas

掘金

角色模块

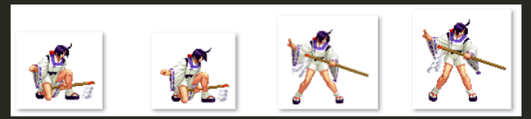
属性

动画

指令

行为

序列帧动画



```
wx.createImage();
context.drawImage(image, x, y);
```

TGideas

2d骨骼动画

可通过spine，dragonbone，live2d等软件制作

2d骨骼动画

掘金

TGideas

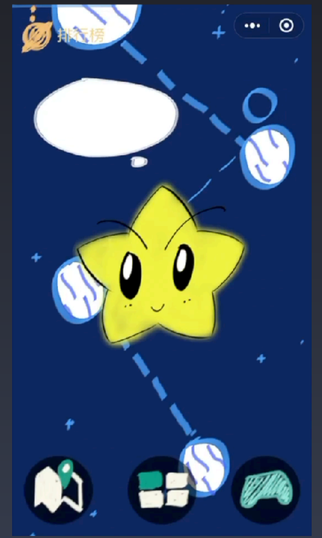
2d骨骼动画

掘金

TGideas

2d骨骼动画

掘金

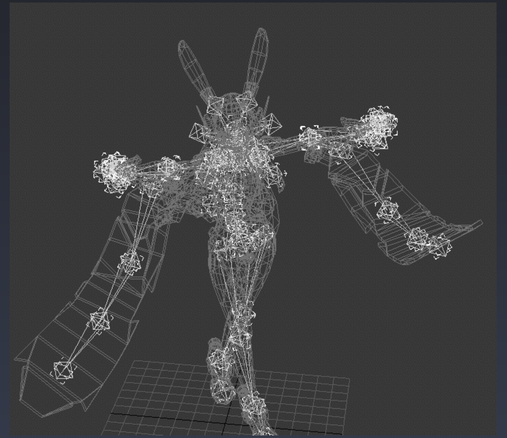


```
{
 "SubTexture": [{
  "width": 447,
  "y": 342,
  "height": 191,
  "name": "tail",
  "x": 1
 }, {
  "width": 270,
  "y": 1,
  "height": 273,
  "name": "chibangL2",
  "x": 326
 }],
 "name": "name",
 "imagePath": "tex.png"
}
```

TGideas

3d骨骼动画

可通过3dsmax, maya, blender等软件制作



3d骨骼动画

掘金

TGideas

3d骨骼动画

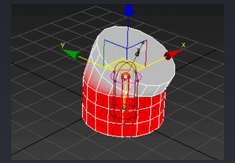
掘金

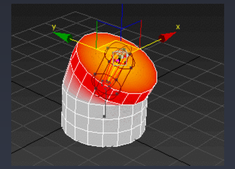
TGideas

掘金

3d骨骼动画


```
{
    "materials": [],
    "vertices": [-0.000000,10.206677],
    "morphTargets": [],
    "morphColors": [],
    "normals": [-0.70555,0.64668],
    "colors": [],
    "uvs": [[0.070949,0.94043],
    "faces": [42,730]
}
```



TGideas

掘金

角色模块

Role

通过model脚本
创建一个实体
object

属性	x, y, hp, speed, sprite…
动画	调用动画及贴图脚本
指令	监控输入数据，并执行相应的行为
行为	通过控制属性，动画，来呈现

TGideas

角色模块

属性

动画

指令

行为

掘金

TGideas

掘金

属性

动画

指令

行为

角色模块

指令

触控 wx.onTouchStart()

键盘 wx.onKeyboardInput()

加速计 wx.onAccelerometerChange()

…

各种输入手段的监听

TGideas

掘金

角色模块

Role

通过model脚本
创建一个实体
object

属性	x, y, hp, speed, sprite…
动画	调用动画及贴图脚本
指令	监控输入数据，并执行相应的行为
行为	通过控制属性，动画，来呈现

TGideas

角色模块

属性

动画

指令

行为

掘金

TGideas

掘金

属性

动画

指令

行为

角色模块

行为

移动　　x+=speed; y+=speed;

受伤　　hp-=damage; sprite.play(hurt);

攻击　　sprite.play(atk);
role.damage=init.damage;

…

通过属性及动画的变化实现角色行为

角色行为常常伴随着大量的算法

TGideas

角色模块

行为

移动　　x+=speed; y+=speed;

受伤　　hp-=damage; sprite.play(hurt);

攻击　　sprite.play(atk);
role.damage=init.damage;

…

通过属性及动画的变化实现角色行为

角色行为常常伴随着大量的算法

角色模块

行为

移动　　x+=speed; y+=speed;

受伤　　hp-=damage; sprite.play(hurt);

攻击　　sprite.play(atk);
role.damage=init.damage;

…

通过属性及动画的变化实现角色行为

角色行为常常伴随着大量的算法

功能模块

Util

根据游戏不同，经常会有需要一些独特的脚本

碰撞检测脚本
音频管理脚本
UI整合脚本
适配脚本
…

掘金

游戏引擎将模块用组织架构的形式管理

模块管理脚本作为主线贯穿整个游戏架构

场景模块加载角色模块

角色模块加载动作逻辑

TGideas

掘金

3 小游戏的部分算法简介

TGideas

游戏中算法运行的机制

渲染器（requestAnimationFrame） → 逐帧计算

物理速度方程

$S = v * t$

游戏内

$S = v + v + v + \dots$

T 持续中

```
render() {
    S += v;
}
```

掘金

TGideas

掘金

游戏中算法运行的机制

渲染器（requestAnimationFrame） → 逐帧计算

加速度方程

S= v0 * t + 1/2 * a * t * t;

游戏内

v = a + a + a + …

S = v + v + v + …

T 持续中

```
render() {
    v += a;
    S += v;
}
```

TGideas

算法的应用 —— 碰撞算法

掘金

TGideas

掘金

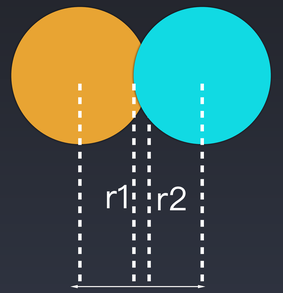
圆形碰撞

r1 + r2 < d

TGideas

掘金

圆形碰撞

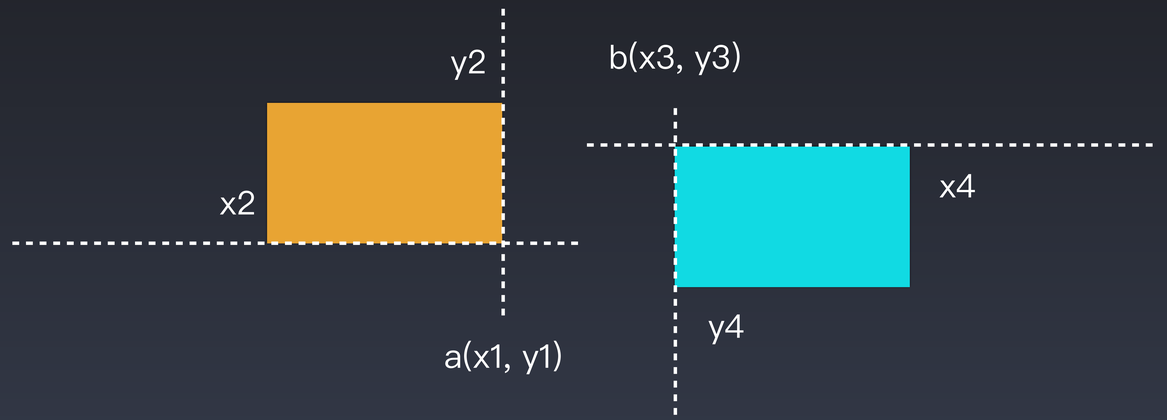


r1 + r2 >= d

TGideas

掘金

AABB包围盒碰撞

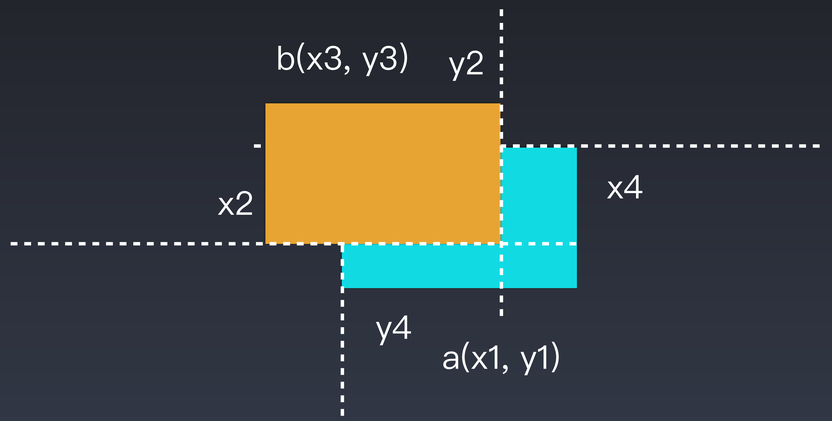


(x3 < x1 < x4 && y4 < y1 < y3) false

TGideas

掘金

AABB包围盒碰撞



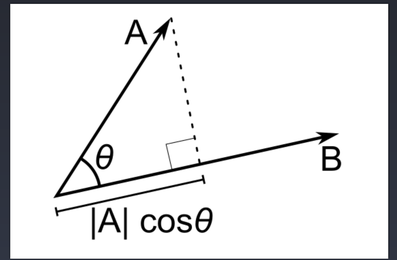
(x3 < x1 < x4 && y4 < y1 < y3) true

TGideas

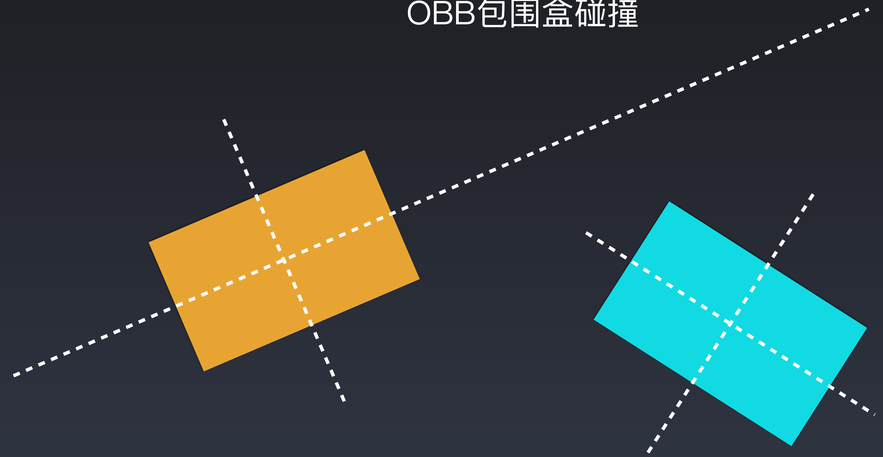
不同的算法，实体属性的定义很重要

仅仅以坐标表示的矩形很难进行更复杂的运算

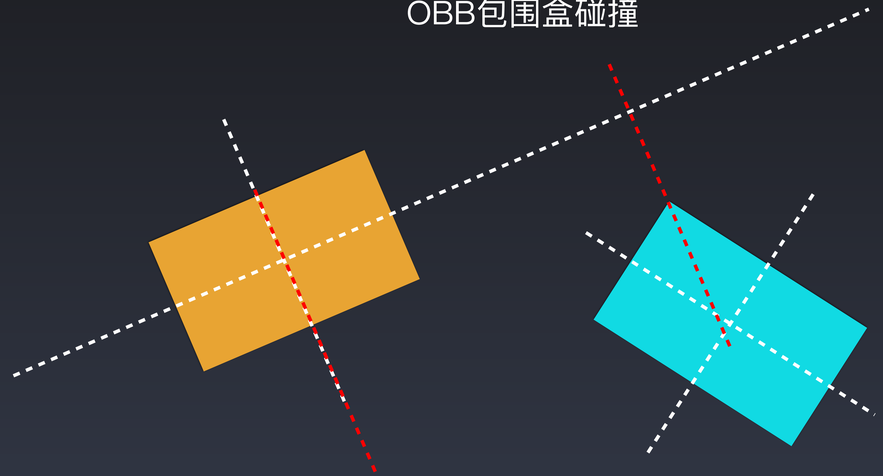
```
vector2( x, y )

vec1.dot( vec2 )

dot : function( v ) {
  return this.x * v.x + this.y * v.y;
}
```



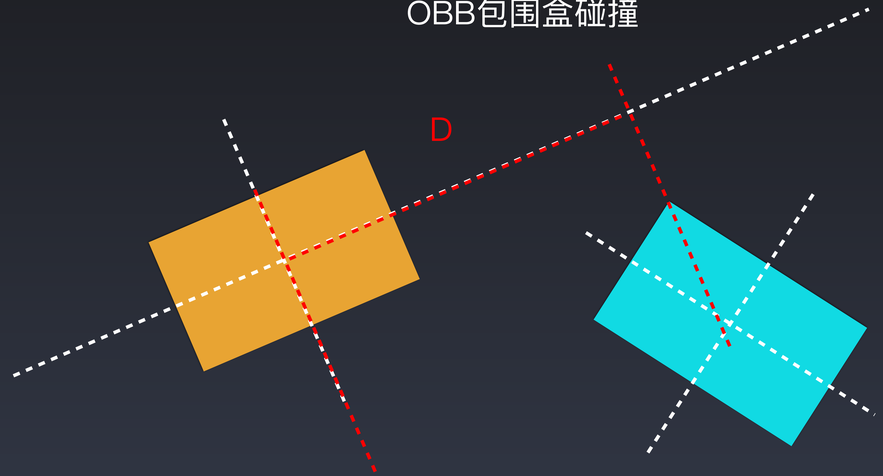
OBB包围盒碰撞



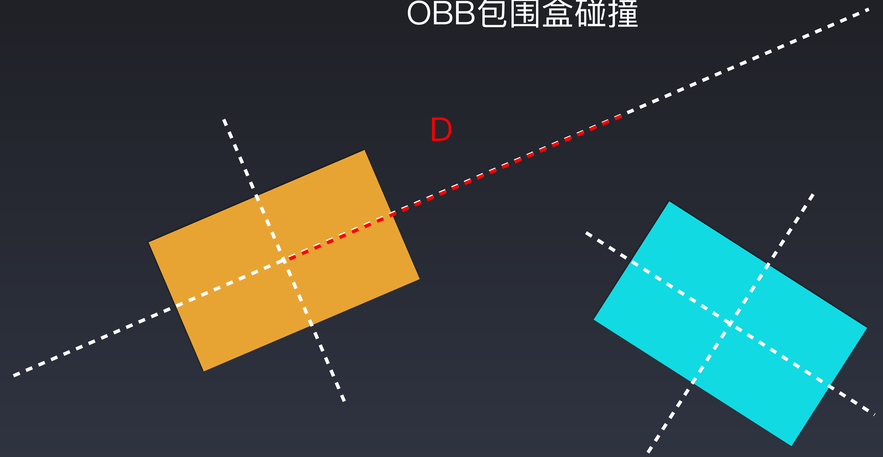
OBB包围盒碰撞



OBB包围盒碰撞



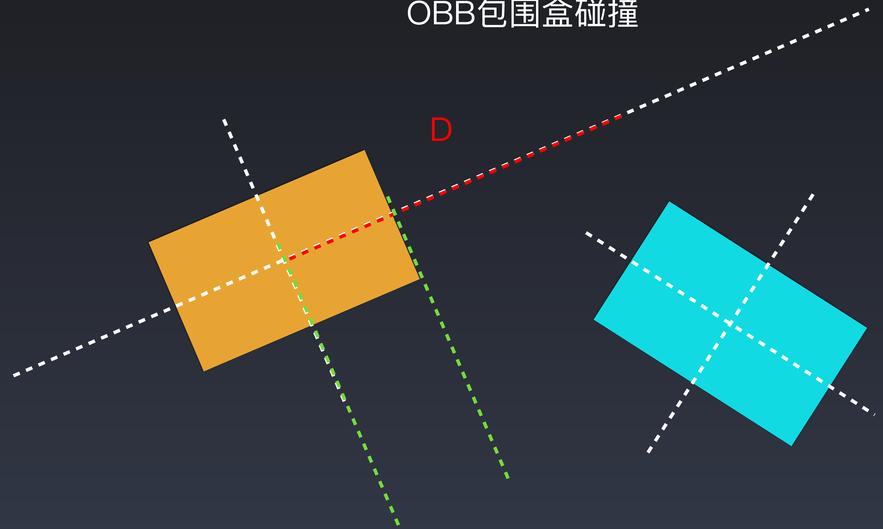
OBB包围盒碰撞



掘金

TGideas

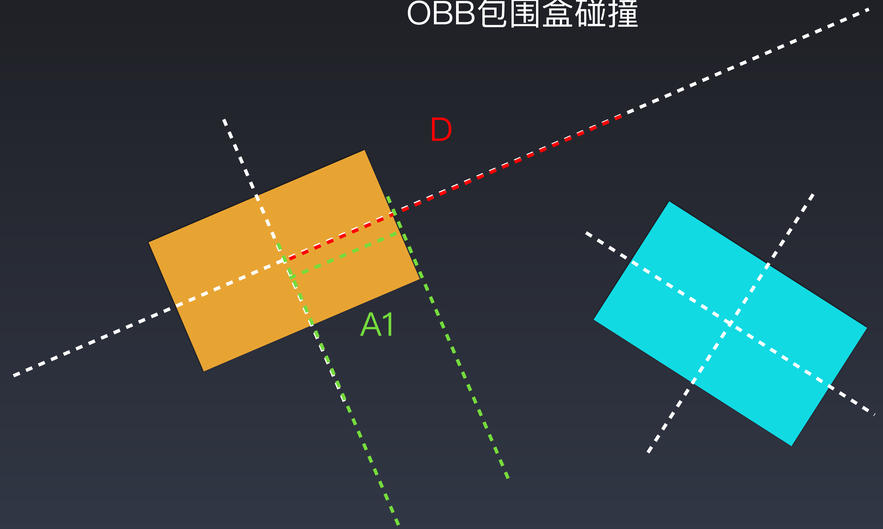
OBB包围盒碰撞



掘金

TGideas

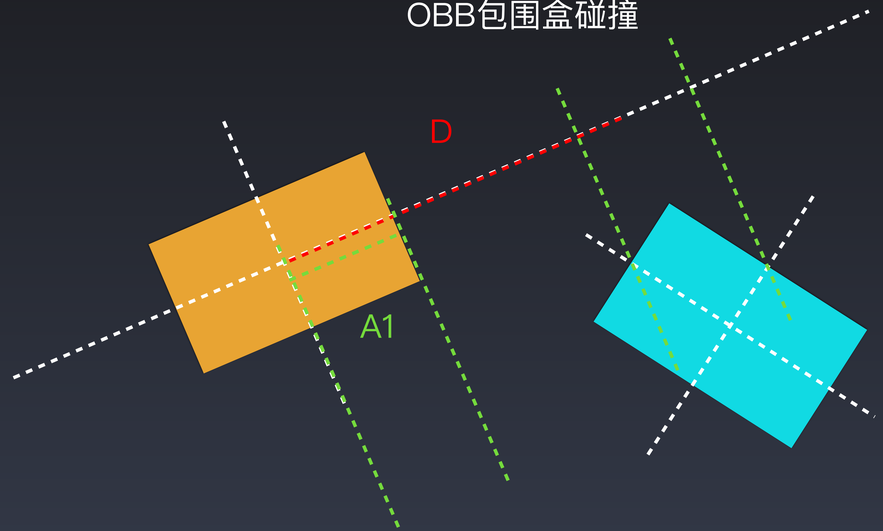
OBB包围盒碰撞



掘金

TGideas

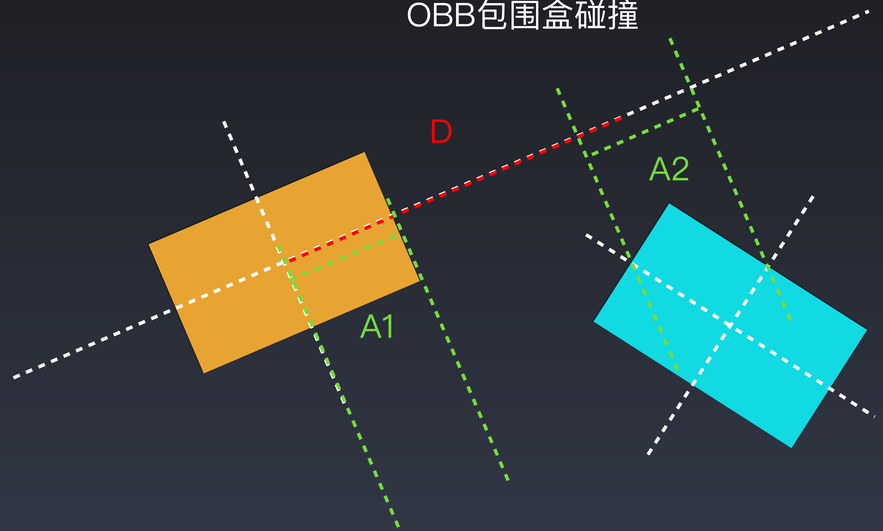
OBB包围盒碰撞



掘金

TGideas

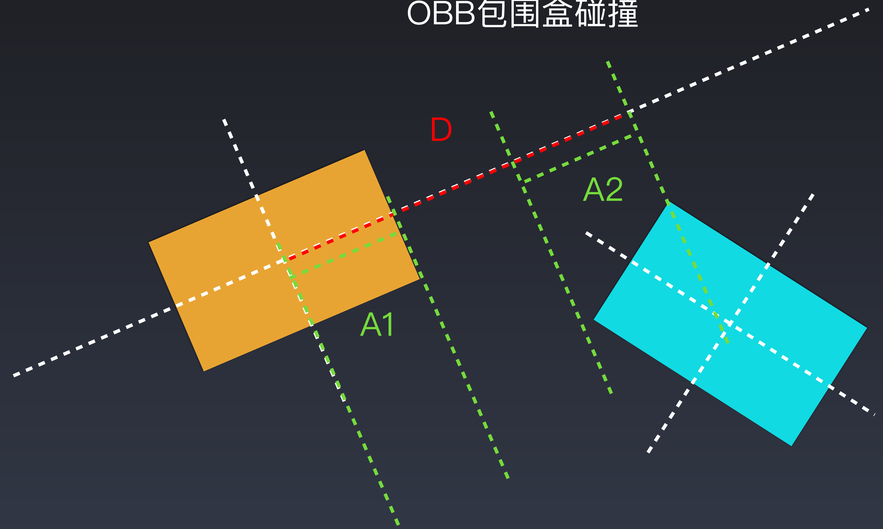
OBB包围盒碰撞



掘金

TGideas

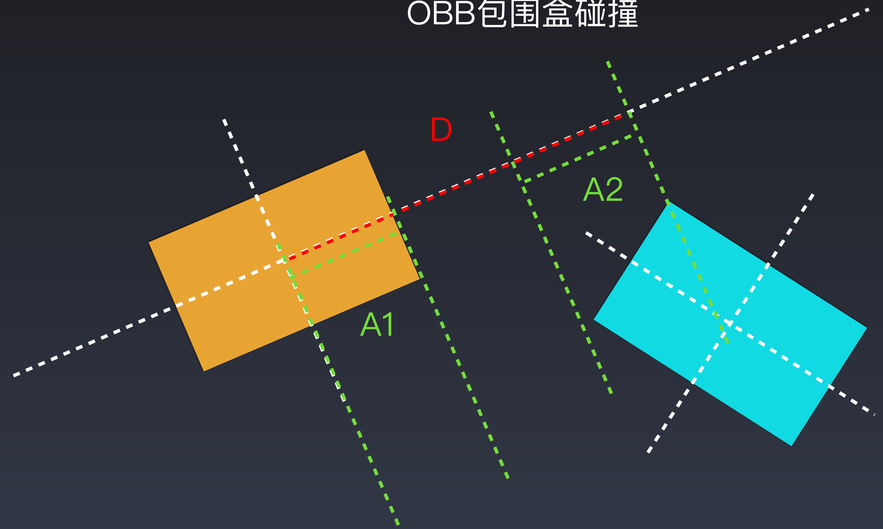
OBB包围盒碰撞



掘金

TGideas

OBB包围盒碰撞

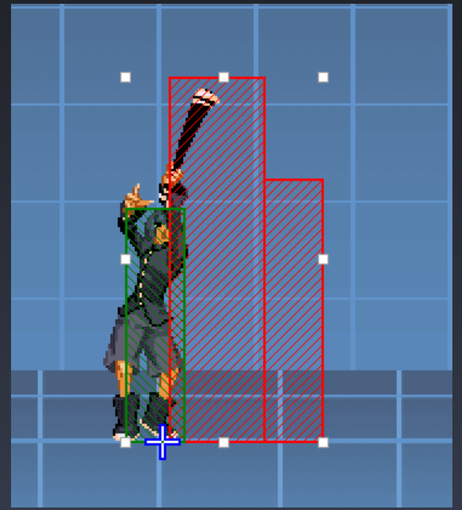


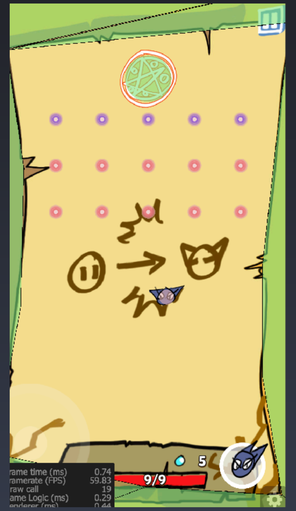
A1 + A2 < D

掘金

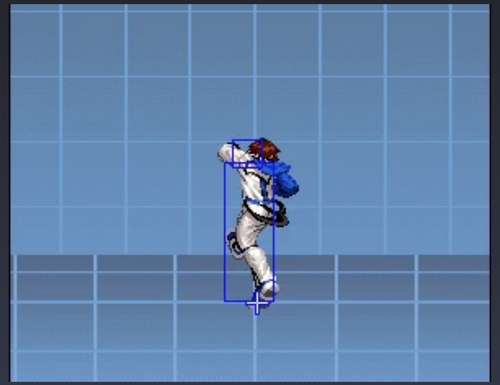
TGideas

实际游戏中碰撞的框是不可见的

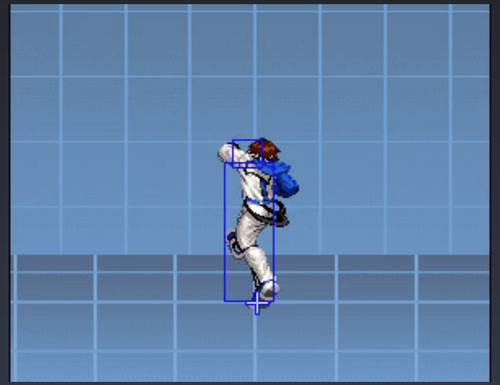




碰撞框在渲染过程中实时刷新



碰撞框在渲染过程中实时刷新



掘金

常用的游戏算法，状态机、字典树、寻路、排序……

TGideas

掘金

4 小游戏的微信API

TGideas

掘金

小程序中大部分操作界面的API小游戏中无法使用

操作WXML节点以及Page的API

TGideas

小游戏中特有的API

掘金

TGideas

掘金

小游戏中特有的API

画布
wx.createCanvas
…

分包加载
wx.loadSubpackage
…

帧率
wx.setPreferredFramesPerSecond
…

性能
wx.getPerformance
…

游戏圈
wx.createGameClubButton
…

虚拟支付
wx.requestMidasPayment
…

激励广告
wx.createRewardedVideoAd
…

防沉迷
wx.checkIsUserAdvisedToRest

TGideas

掘金

「子域」是一个很特殊的机制

开放数据

TGideas

小游戏的「开放数据」

掘金

TGideas

掘金

小游戏的「开放数据」

开放数据

wx.getFriendCloudStorage

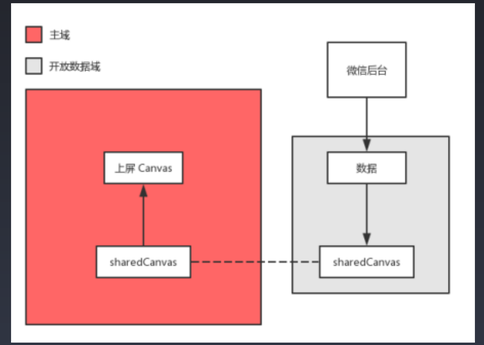
wx.getUserCloudStorage

wx.getGroupCloudStorage

wx.getSharedCanvas

KVData

…



TGideas

小游戏的「开放数据」

小游戏的「开放数据」

小游戏的「开放数据」

小游戏的「开放数据」

context.drawImage(, 0, 0)

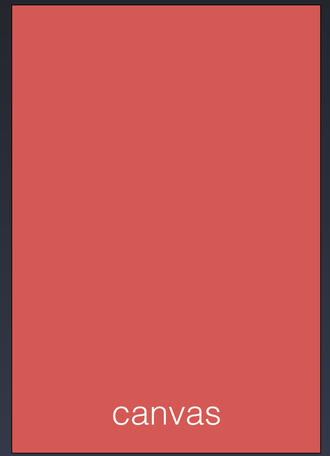
掘金

TGideas

小游戏的「开放数据」

context.drawImage(sharedCanvas , 0, 0)

小游戏的「开放数据」

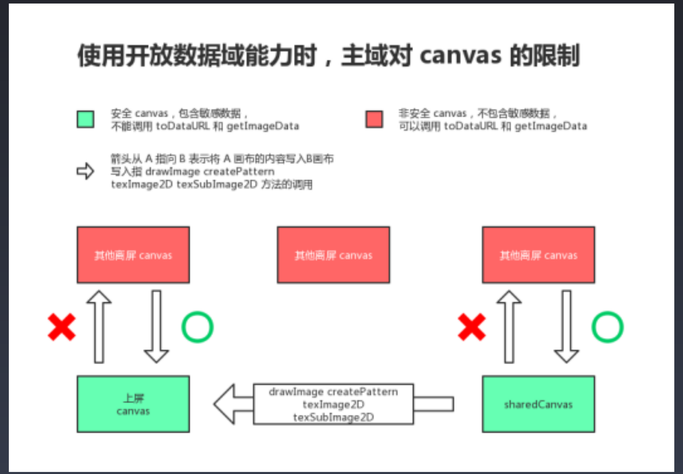


context.drawImage(sharedCanvas , 0, 0)

小游戏的「开放数据」

context.drawImage(sharedCanvas , 0, 0)

小游戏的「开放数据」



1. sharedCanvas 只能被绘制到上屏 canvas 上。
2. 上屏 canvas 不能调用 toDataURL，其 context 不能调用 getImageData。
3. sharedCanvas 不能调用 toDataURL 和 getContext。
4. 不能将上屏 canvas 和 sharedCanvas 以任意形式绘制到其他 canvas 上，包括 drawImage、createPattern、texImage2D、texSubImage2D。
5. sharedCanvas 的宽高只能在主域设置

总结

小游戏是通过模块管理来操作各个实体的属性产生的一种交互过程

微信的开放域数据是一个很奇特的机制

掘金

Thank you !

跟我在沸点 AMA 交流

TGideas