

H1 GoldMiner小游戏开发文档

H3 2019.7.26

H2 零、项目结构

- 体验版二维码：



- github仓库链接：<https://github.com/LosMerengues07/GoldMiner>
- Artwork 是我们自己的 原创图片素材 文件夹，使用Krita软件手工绘制而成，包含17张图片素材，并按照规定转为psd格式
- GoldMiner 是本游戏的主体工程文件夹
- subdomain 是微信开放数据域(即排行榜)的工程文件夹

H2 一、游戏开发背景

1.游戏选题

本次大作业的题目要求借助微信小游戏平台开发一款手机游戏，由于手机性能，屏幕大小的限制，以及微信小程序平台对项目大小的要求，我们最终选择了开发一款经典的小游戏：“黄金矿工（GoldMiner）”。为了更好地展现我们对游戏的设计，更加灵活地拓展功能，我们选择了自己手绘游戏图片资源，因此最终开发了“手绘版黄金矿工”这款游戏。

2.任务合作完成情况

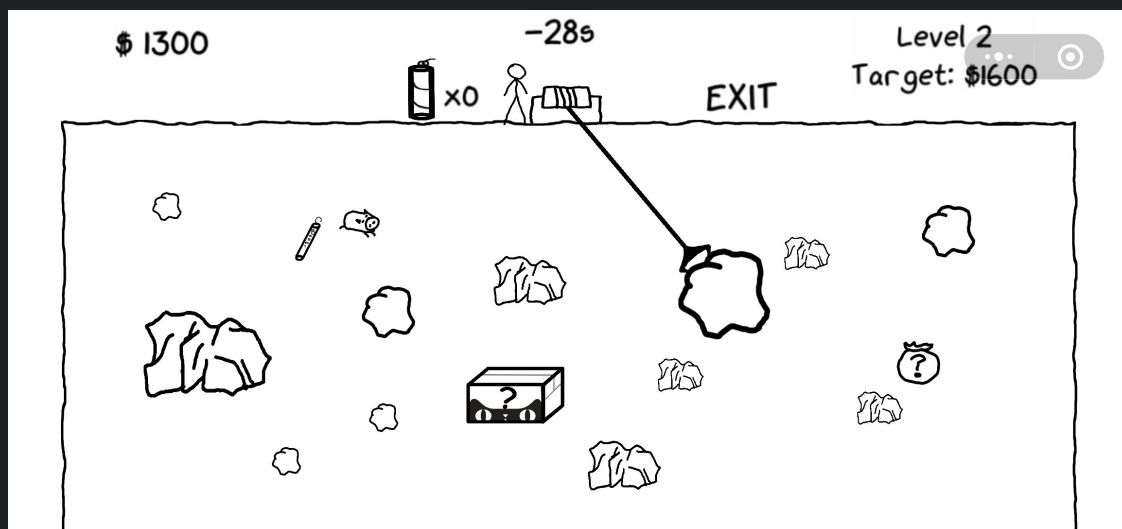
王世杰：游戏主UI与主逻辑功能设计与实现，游戏内所有动画效果实现，微信开放数据域排行榜，游戏菜单界面设计，游戏设置界面，音效实现，商店功能，游戏关卡设计

吴海隽：美工设计绘图资源，新手指引界面实现，游戏关卡设计，双人模式（基本完成 但手机版体验较差 故未发布）

3.游戏介绍

Gold Miner 是一款休闲游戏，其核心玩法是玩家扮演一名矿工，点击屏幕中间位置可以放下摇摆的钩子，用于勾取地下的各种资源，如不同大小的黄金与石块，会移动的猪，价值极高的钻石，背着钻石的猪，TNT炸药以及神秘宝藏等，不同的资源具有不同的价值与效果，在每回合的30s倒计时内完成目标钱数的收集即可通关。为了增加游戏性，游戏还设置了商店功能，商店内物品的种类和价格是随机的，具有大力水、炸药、钻石书、幸运草等效果各异的道具。

游戏主界面如下：



4.说明

我们的创意主要来自于经典小游戏黄金矿工，但我们的所有工作，包括素材绘制，游戏规则设定，道具实现，动画实现等都由自主完成，并未参考他人工作。

5.开发环境

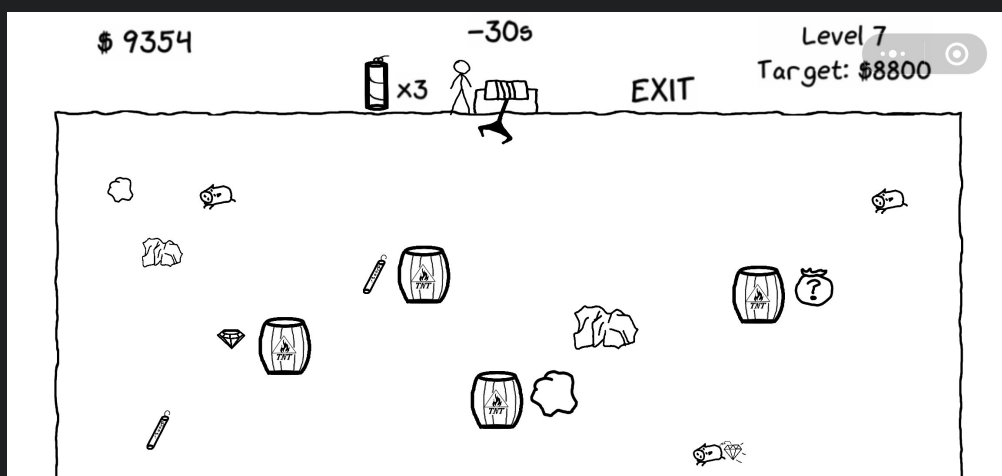
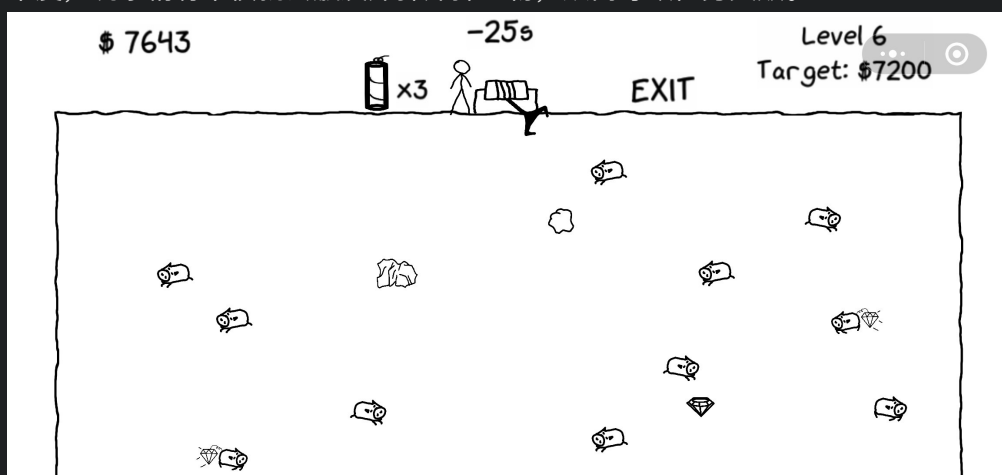
- IDE：Cocos Creator 2.0.10
- 编程语言：JavaScript (ES6)
- 操作系统：macOS 10.14 和 Windows 10 Pro Education
- 上传工具：微信开发者工具 Stable v1.02.1907160

H2 二、游戏策划与功能

1. 游戏概要

- 名称：手绘版黄金矿工

- **概述：**《手绘版黄金矿工》是一款老少皆宜的休闲游戏，游戏的目标很简单：玩家扮演一名黄金矿工，在合适的时机放下钩子勾取底下的物品，不同的物品具有不同的价值和特殊属性，最终在时长30s的一回合内达到目标资金即可过关，而金钱在不同关卡之间是累计的，玩家也可使用累积的金钱在每回合结束后的商店中购买道具以获取不同加成。
- **特色：**本款游戏最大的特色就是我们使用的原创手绘素材，采用黑底白色线条的简约设计，手绘风格的界面更加随性与具有趣味性。而游戏中出现的不同的道具与物品也大大丰富了游戏内容。在关卡设定方面，我们一共提供了8个关卡，难度逐渐提升，其中有些特色关卡非常考验玩家的专注度与准确性，例如第六关玩家需要在很多小猪中抓到移动的钻石小猪，难度较大；第七关，通过TNT与宝物的绑定提升难度，玩家稍有不慎就会触发炸弹炸掉宝物，从而导致任务失败。



2. 界面场景设定

此部分请直接查看文档第三部分——界面布局设计

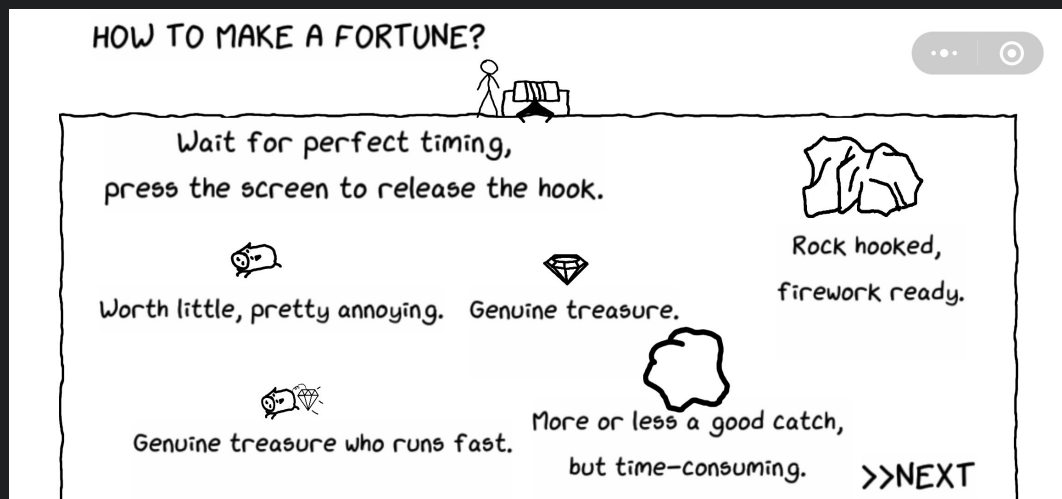
3. 游戏元素

游戏中具有以下几种可以交互的实体元素：

主游戏内元素：

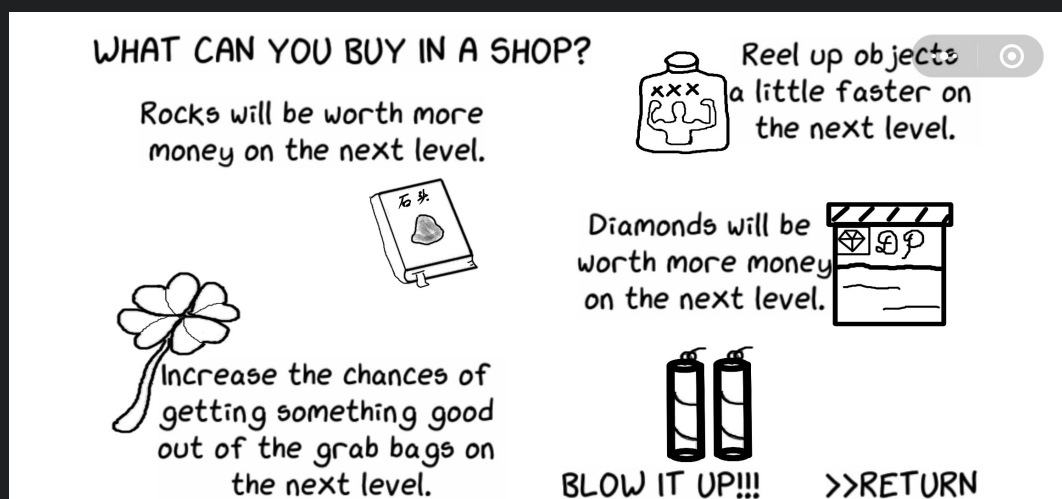
1. 钩子：玩家的工具，用于勾住宝物。
2. 金块：地下宝物之一，具有大中小三个尺寸，价值与拉取速度不同。

3. 石头：地下主要障碍物之一，具有大中小三个尺寸，重量较大，价值较低
4. 小猪：地下主要障碍物之一，会在地下水平运动以干扰玩家，价值极低
5. 钻石：地下宝物之一，尺寸很小，重量很轻，价值极高
6. 钻石猪：会运动的猪背着钻石，价格为猪与钻石的和
7. 宝物袋：地下宝物之一，随机获取金钱或道具，重量随机
8. 宝物盒：地下宝物之一，随机获取金钱或道具，重量随机，通常比宝物袋更好
9. 鞭炮：可拾取道具，拾取后自己的鞭炮道具数目+1
10. TNT炸药桶：不可拾取，一旦触碰会发生爆炸，使得周边物品消失



商店道具：

1. 石头书：商店道具，价格随机，购买后可在下一回合内增加石头的价值
2. 大力水：商店道具，价格随机，购买后在下一回合中可以以很快速度勾取任何物品，也可在宝箱中获取
3. 鞭炮：商店道具，价格随机，在回合间累积，可以用来炸掉勾取物品，例如石块等，可以在地下拾取，也可在宝箱宝物袋中获取
4. 幸运草：商店道具，价格随机，购买后可在下一回合内提升幸运度，在宝箱与宝物袋子中获取更好的物品与道具
5. 优质钻石：商店道具，价格随机，购买后在下一回合内提高钻石价值



4. 关卡设定

游戏具有精心设计的八个关卡，难度逐渐递增，其中有些特色关卡非常考验玩家的专注度与准确性，例如第七关，通过TNT与宝物的绑定提升难度，玩家稍有不慎就会触发炸弹炸掉宝物，从而导致任务失败。同时由于游戏的随机商店机制，如何合理分配资金购买道具获取加成也为游戏的提供了多样的玩法。

H2 三、界面布局和设计

为了契合游戏整体的黑白手绘简约风，我们的布局也力求简练友好，具有趣味性。

下面介绍几个用户界面的布局设计：

- 主菜单界面

主菜单界面如下图所示：为了提高趣味性，让主题界面不过与单调，我们采用了和主游戏相同的交互方法：点击屏幕方法钩子勾取不同选项，同时这样的设计也能够让新用户第一时间了解游戏的主要交互方式，起到了教学的作用，一举两特，颇具特色。

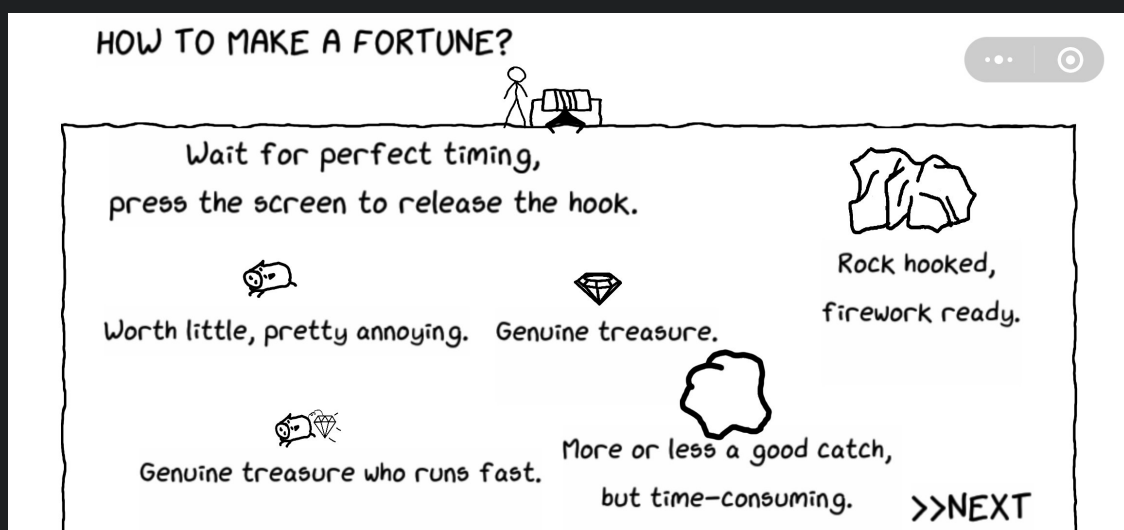
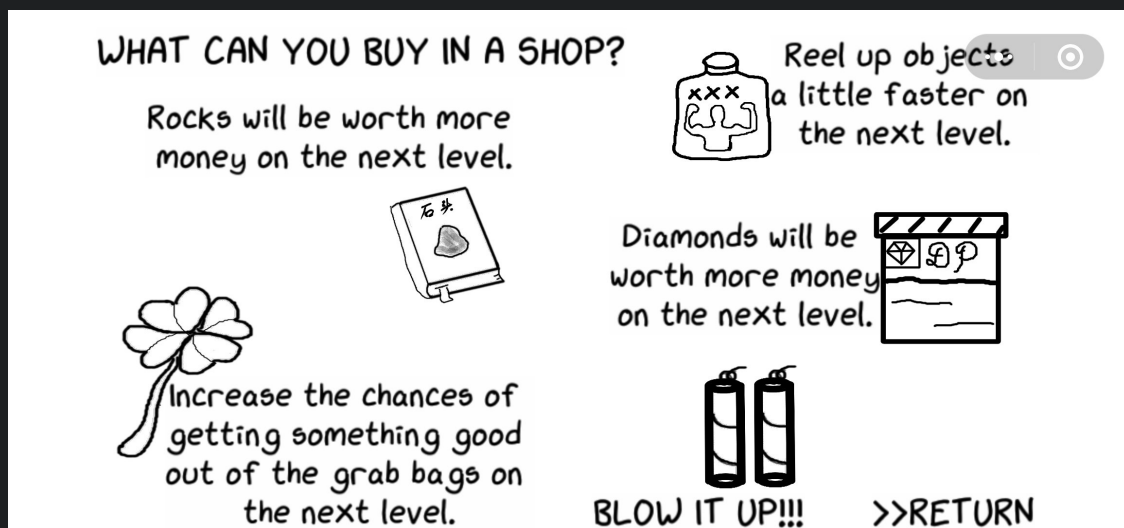
在授权登录后，右上角会显示用户微信头像。

主菜单界面具有四个选项卡，分别对应教程，开始，排行榜与设置界面，选取拖动后即可进入。



- 教程界面 (TUTORIAL)

静态的新手教程页面，介绍了游戏的主要交互方式，道具与商店系统，图片与文字相结合，简单易懂。界面如下图所示：



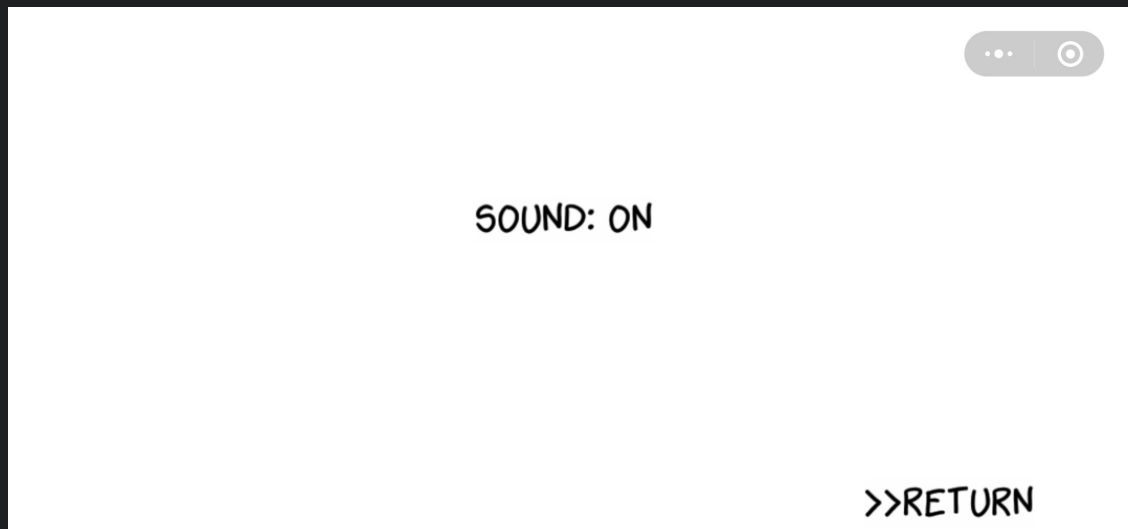
- 排行榜界面 (RANK)

排行榜界面使用微信开放数据域，可以读取好友中该游戏的TOP5排名， 点击右下可返回主菜单



- 设置界面

可以设置是否静音，点击右下角可返回主菜单



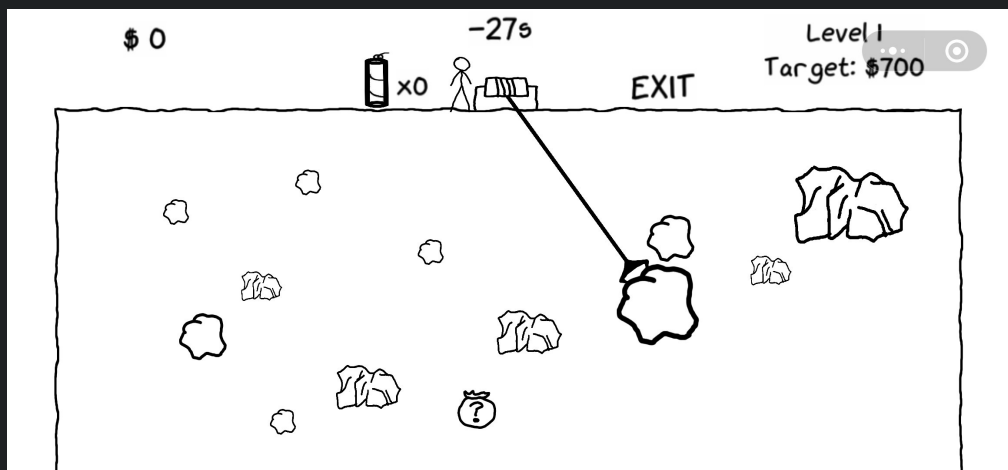
- 主游戏

主游戏界面又分为几个界面，下面逐个展开介绍：

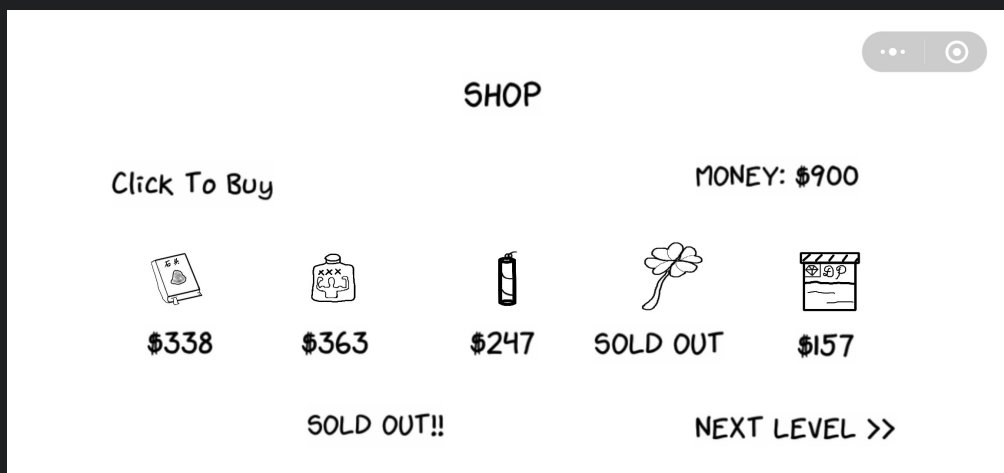
- 关卡载入界面：显示当前关卡和目标，并显示玩家当前具有道具数目，在3s后自动进入游戏界面



- 主游戏关卡界面：主要游戏界面，右上角显示游戏当前关卡和目标数，上方显示倒计时数目，小人左面又鞭炮计数器 and 闪电形状的力量表示，用以提醒玩家，点击鞭炮即可炸掉当前物品。左边为当前收集钱数。“EXIT”按钮可以点击后放弃当前游戏返回主菜单，其中鞭炮数目和钱数增加都有动画效果以提示状态改变。



- 关卡结束界面，有失败，成功，通关三种状态，可以展示分数，可以实现重新游戏，返回主菜单等操作。
- 商店界面：在商店内显示物品的价格，而是否售光也是随机的。玩家可以点击图标购买，购买后该物品会消失，同时左下角会根据购买状态提醒玩家，如“已售光”、“钱不够”或“获得一本石头书”等



H2 四、技术实现方案及重点与难点与亮点

1. 场景切换时信息的保留与传递

Cocos Creator中最基本的可视载体是场景文件（scene），而scene之间的数据不能相互访问，因此为了场景切换之间的信息传递是一个重要的问题。

在本游戏中，我采用了全局变量的方法来实现，即定义脚本 `Global.js`，记录一些游戏中需要跨场景的全局信息，以及一些游戏参数设定，例如当前金币数，当前鞭炮数，各个关卡的目标钱数，这样通过对全局变量的访问与修改，即可实现不同场景之间的信息传递与交流。

2. 游戏场景的主逻辑

本游戏中的核心元素为用户控制的钩子，主逻辑都围绕钩子进行，而钩子的运动可以分为三个状态：

- 未释放，围绕轴点做周期性转动
- 已释放，正在向下运动抓取宝物，此时应该检测碰撞
- 已释放，正在向上往回收，此时不应该检测碰撞

根据分析可以得出，钩子的运动状态一直处于1->2->3->1....的循环中，因此交互与游戏状态的改变就应该集中于钩子的三种状态转换的临界条件：

- 1->2：当用户点击地下部分的屏幕时，点击事件发生，停止钩子的转动动画，并以当前角度以一定的速度下沉，
- 2->3：从下降状态转到上升，当且仅当两种情况：1) 钩子碰触到某个宝物（item Group）2) 钩子碰到屏幕边缘。以上两种情况互斥。当钩子与物品发生碰撞时，为了使得回收动画更加真实，我将物品做了一个小位移，使得在视觉上物品确实被钩子勾住了，而不是悬空状态。随后需要根据钩子碰触物品的情况设置钩子回收速度，还要考虑碰到TNT，中途物品被炸，出于“大力”状态等特殊情况。其中为了使得物品被炸的情况下效果更加真实，我在爆炸动画结束帧上绑定了回调函数，当该回调函数执行时删除被炸

物品，重新设置钩子回收速度。

- 3→1：从上升状态转到转动状态，临界条件应该是钩子返回地面。因为在实际代码中钩子的坐标是采用速度x三角函数的形式，会产生近似误差，很难保证回收后坐标与原坐标严格相等，因此我采用了判断差值的方法，当钩子的纵坐标和初始纵坐标差距足够小，即认为已经归位，进入状态1。需要注意的是，在切换状态时，需要将钩子的 `occupied` 属性置为false，表示不再勾住物品。同时还要注意，在上升状态时，不应该在做碰撞检测以防止勾到小猪等活动物体。

以上就是钩子的核心逻辑，下面介绍游戏场景中的另外一种重要元素，地下物品：

首先为了对元素的碰撞属性做区分，我对元素进行了分组，如下图所示：

分组管理

添加分组

请注意! 分组添加后不可删除

Group 0	default
Group 1	hook
Group 2	items
Group 3	pig

允许产生碰撞的分组配对

	pig	items	hook	default
default	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
hook	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
items	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
pig	<input checked="" type="checkbox"/>			

可以看到，元素一共分为了4组，其中default代表不和任何物品碰撞的元素组，hook组只有hook一个元素，items组包括除了两种小猪之外的所有地下元素，他们的碰撞关系是基于这样的想法设计的：

- 钩子可以和小猪和地下物品发生碰撞
- 物品和小猪可以发生碰撞，当小猪碰撞时调转运动方向

因此地下物品的逻辑如下：

当钩子和物品发生碰撞时，分以下几类：

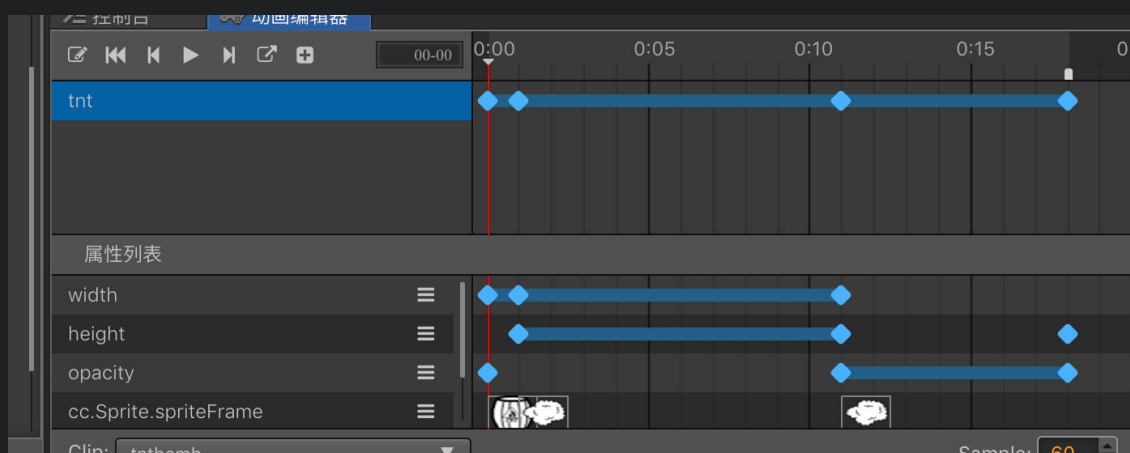
- 小猪和钻石猪：停止水平运动，跟随钩子以一定速度回收，再回收过程中为了增加效果依然播放小猪两腿爬行的动画，直至钩子转到状态1，小猪消失，钱数增加。
- TNT炸弹：开始TNT爆炸动画，播放爆炸音效，动画结束后执行回调函数删除自己和周边元素
- 其余元素：跟随钩子以一定速度回收，直到钩子转为状态1，获得金钱或道具

在回收的过程中以上物品都可以发生爆炸，爆炸时依然播放爆炸动画，并回调函数。这里回调函数的目的是为了钩子的速度在动画执行完后再恢复较快速度，而不是一碰撞就恢复，这样会导致蘑菇云和钩子分离，视觉效果不好。

3. 游戏内动画的实现

本实验中，设计自然，生动的游戏动画效果占据了很大一部分的工作量，经过统计，在本实验中我一共定义了20个anim动画脚本。其中根据功能主要分为以下几种：

- 小猪的跑动动画，根据方向和种类一共4种，采用 `spriteFrame` 属性循环切换来实现双腿跑动的效果
- 钩子的转动动画，根据转动起点不同一共分为3种，设置角度属性
- 载入动画，共1个，通过循环切换标签文本实现载入效果
- 爆炸动画，共10个，其中以最复杂的TNT爆炸效果为例：为了生成逼真的爆炸动画，我对爆炸的过程进行了分析：发生爆炸时，桶状物体首先变成蘑菇云，随后一段时间内蘑菇云体积变大，最后蘑菇云透明度逐渐降低至0淡出。在透明度为0的那一帧，我调用了回调函数 `onBombOver`，用于判断TNT周围物品并destroy相关节点，完成爆炸——消失的全过程。最后封装成一个动画，配合音效即可实现较为逼真的爆炸效果。



- 金钱/鞭炮获取提示动画，共2个，该动画效果是当获取金钱或者鞭炮时，在计数器旁显示“+xx”字样并淡出以提示玩家。

4. 排行榜——微信接口的调用

微信小游戏为了保护其社交关系链数据，增加了子域的概念，子域又叫**开放数据域**，是一个单独的游戏执行环境。子域中的资源、引擎、程序，都和主游戏完全隔离.....由于子域只能在离屏画布 `sharedCanvas` 上渲染，因此需要我们把 `sharedCanvas` 绘制到主域上。——Cocos Creator文档《接入微信小游戏的子域》

为了增加游戏的社交属性，本游戏加入了微信好友排行榜的功能。经过资料查阅，微信小游戏为了保护用户隐私，增加了子域的概念，即**开放数据域**，开放数据域是一个独立的程序，其中的资源数据等都和主游戏完全隔离，想要在主程序上完成排行榜的功能就要在发布到微信平台时正确配置发布信息，而将开放数据域的界面作为一个画布在主域中渲染出来，很多获取社交数据的api只能在子数据域内调用。

在本次实验中，我使用 `wxSubContextView` 控件作为子域的画布，在用户选中rank界面后，通过显示 `rank` 场景，而子域中通过 `wx.onMessage` 接受主域发送的信息，通过不同的信息状态完成显示排行榜，读取分数，设置分数等功能。本游戏中，我获取了好友的名称与头像，并对游戏的得分做了排序，显示了好友中的TOP5排行榜。

5. 疑难杂症集中地——微信分包

由于微信小游戏突出一个小字，腾讯将上传代码大小限制在了4M，由于使用了比较多的图片音乐文件资源，因此需要采取分包的方法将音乐文件和图片资源分包上传。

这方面cocos的文档写的并不很详细，最开始一直在微信开发者工具上报错。后来经过群内高人指点，发现在cocos生成项目文件后需要在载入脚本中增加loadPackage的语句来载入分包资源。经过这样的方法后，游戏终于可以顺利在手机上运行

但是经过一系列测试，我发现了另一个问题：手机上的游戏在运行时，会时不时出现卡在100%的加载页面的情况。由于这种情况时不时发生，因此我推测和分包的资源加载有关。经过和助教以及群内大神们的探讨与学习后，我找到了真正的原因：由于js的异步执行特性，可能会出现资源还没加载好就开始执行场景的状况，这就导致了资源错误。因此解决方法就是在游戏的开始界面之前，加入一个不需要任何分包资源的loading界面，并且通过回调函数的形式，在资源加载成功后才允许load开始场景。同样为了美观起见，我在载入界面增加了加载动画以提醒用户。

解决了以上两个问题后，分包导致的问题就再也没有出现过。深刻感觉到，学习前端的时候对后端也有了解是多么重要的事情，这里也感谢提供帮助的助教和同学们，在我成功后也成功帮助了几个同学避免踩雷，感觉十分荣幸。

6. 用户交互的细节设计

在第一节课程上，老师就说过，前端的一大特性就是需要和用户直接打交道，因此一个优秀的前端设计不光需要好用，更要给用户良好的交互体验，作为资深游戏玩家和菜鸟游戏程序员，我对此深有感触。

在本次实验中，我采取了以下几个方面增加用户的交互体验友好度：

- 载入界面使用动态动画，让用户了解现在正在加载而不是卡住了
- 开始界面使用动态钩子的形式选择条目，让新用户提前熟悉游戏模式，并且不至于觉得界面过于简单单调。
- 增加音效，例如在游戏中，放下绳子，碰到不同物品，爆炸等都有自己的音效，游戏成功，失败界面也有自己的音乐。
- 增加按钮等可交互元素的辨识度。因为本游戏整体上采用了二维手绘画风，整体风格扁平化，因此没有采用拟物化的按钮控件，而是使用平面的label，这样就可能使得用户觉得按钮的辨识度不够，因此我对可点击的label文本加上了“>>”的符号，例如退出按钮“>>EXIT”，这样更加醒目，也更动感，对用户的指示性更强。
- 在游戏界面内增加玩家的反馈，例如在得到金钱或者炸药的时候，在计数器旁提示增加个数并逐渐消失的动画，能够引起玩家注意力。

- 使用了和手绘画风搭配字体 `xkcd`，整体上更加协调，画风比较统一。
- 经过测试组测评反馈，在游戏内部加上了返回按钮
- 在显示微信好友的昵称是，我发现如果全部显示的话微信的上限名称过长，因此需要截断名称。但是由于emoji的存在，简单的截断字节可能会出现错误，因此我写了一个函数对上述情况做处理。
- 关卡的设定从难到易，经过二十多位朋友的测试，感觉难度比较合适，最后几个关卡比较考验操作，可以通过重新关卡

7. 一些bug的记录与修改

- 一开始没有写返回时钩子不能进行碰撞判定，导致可能碰到小猪，不够合理
- 在通关界面的重新进行关卡过程中，一开始没有将数据重置，导致一个同学用这个bug刷出了理论最高分的3倍分数
- 经助教测试组反馈，如果点击较快，可能在物品发生爆炸的过程中再使用一次爆竹，不够合理。因此我修改了逻辑，增加了一个状态表示物品正在爆炸中，以防止爆炸中再次爆炸。
- 经测试组反馈，在游戏还有1s中的时候就结束。解决方法是计数器结束条件-1。

H2 五、游戏测试

经过实际测试，在运行体验版的小游戏时，除了排行榜唤出时获取微信云端数据时出现延迟外，其他交互无明显延迟。关卡之中的操作也较为顺滑流畅，没有明显的卡顿。在游戏开发完成后我也邀请了一些同学、朋友做我们游戏的体验者，下面是一些测试感受：

名字	手机型号	测试感受
S同学	Vivo xplay5	“画风不错”
W同学	Vivo x7	“游戏挺复古，关卡有点少”
L同学	iPhone X	“美工不错”
L同学	华为mate 20Pro	“后面几关有点难”
H同学	华为mate 20Pro	“音乐稍有延迟，其余不错”
W同学	小米9	“可以考虑加一些道具”

H2 六、参考资料

https://github.com/fool-wang/WeChatGame_Dudo

<https://docs.cocos.com/creator/manual/zh/>

<https://developers.weixin.qq.com/miniprogram/dev/api/>

https://blog.csdn.net/qq_36720848/article/details/89464085

特别感谢微信群中热心讨论回答问题的同学们，帮我们测试找bug的测试组成员和我的朋友们，以及给予我们指导的老师 and 助教大大们~