

前端小学期综合实验说明文档

——微信小游戏探格 Tango

刘雍 2017013578 软件 72 liuyong1095556447@gmail.com

刘旭鑫 2017013621 软件 73 liuxuxin@outlook.com

Github: <https://github.com/WenWeiTHU/Tango>



小程序码

1. 项目结构

Assets:

- Prefab (预制资源)

- Resource

 - Anim (动画资源)

 - Audio (音效资源)

 - Frames (动画帧资源)

 - Img (静态图片资源)

 - Ttf (字体资源)

- Scene (场景资源)

- Scripts (脚本资源)

2. 开发环境

操作系统: Windows 10 Pro Education

编程语言: Javascript (ES6)

游戏引擎: Cocos Creator v 2.0.10

版本控制: Github

发布环境: 微信开发者工具 Stable v1.02.1904090

3. 分工情况

我们在 Github 上使用了两个分支

刘雍: (master 分支), 摇杆移动控制, 主角链接实现, 技能; 敌机类设计, 继承, 运动, 攻击; 镜头跟随, 镜头切换, 镜头过渡; 动画效果, 音效; 无限模式, 游戏数值调试, 帮助向导, 剧情设计, 关卡设计 (7-12 关); 分包加载。

刘旭鑫: (map 分支) 碰撞检测及回调, 边界检测; 地图类 (含炮台) 实现, 分辨率适配; UI 设计 (角色图片, 按钮, 加载页和 logo); 物理效果模拟; 关卡设计 (1-6 关), 模式选择界

面；场景切换和本地存档，游戏暂停，排行榜。

4. 游戏概述



游戏图标

名称：Tango（中文名：探格）

说明：

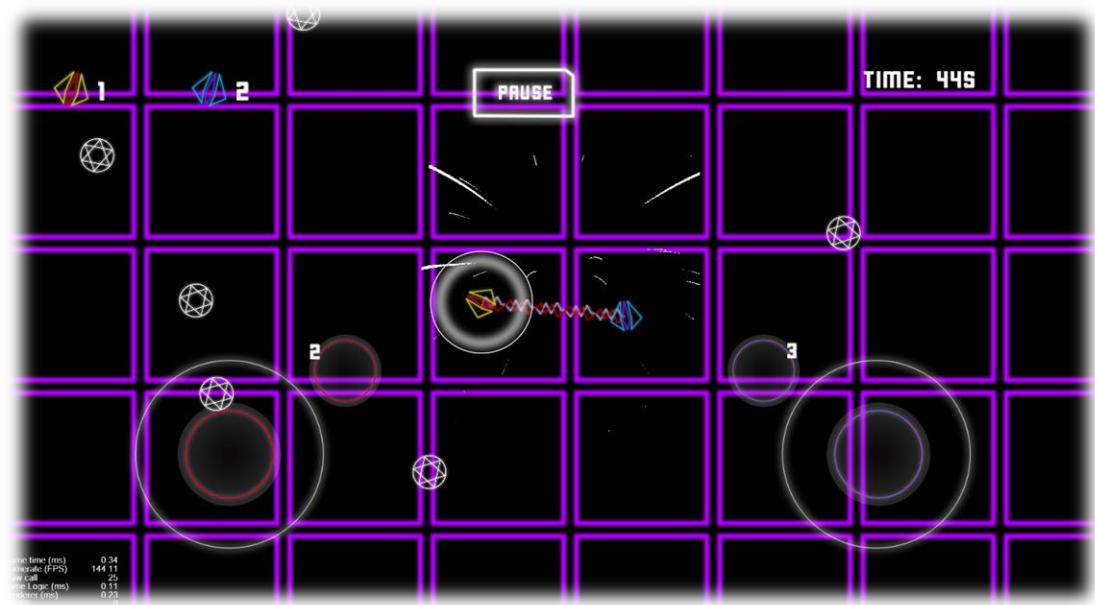
探格 Tango 是一款基于 cocos 引擎开发的、单人离线休闲小游戏。在游戏中，玩家需要操作两个操纵杆，控制两部战机移动，并使用战机中间的闪电链接击杀敌人。

游戏有两个模式，分别是闯关模式和生存模式。在闯关模式中，玩家消灭场景中所有的敌人并到达目标点即可通关。在生存模式中，地图中会不断产生各种类型的敌人，玩家需要不断击杀敌人并保持自己的生存，最后游戏会以玩家的生存时间作为玩家的战绩。

探格 Tango 取材自 Android 弹幕游戏 PewPew，并在 PewPew 的基础上进行了一定程度的改变，并新增了一些自创特性和玩法。整体设计风格充满科幻与电子气息，图片素材均为自主设计，精心打磨。游戏开始还有帮助场景使玩家快速融入游戏，还在游戏中穿插了一些有趣的彩蛋等待玩家发现。

设计理念：

游戏设计之初，我们计划做一款相对休闲的战机类游戏，考虑到基于射击与弹幕的游戏过于俗套，而且也并不适合微信小游戏开发的风格。我们突发奇想，结合之前玩过的一款名为 Pewpew 的游戏中的一个模式，打算做一款完全不基于射击而是着重于双手协调的休闲游戏。我们给游戏起名为 Tango，即探戈，它是一种源自阿根廷的双人舞蹈，作为双人舞蹈，其讲究男女之间的配合和舞步协调，我们的游戏也是如此，玩家通过左右两个操纵杆同时控制两架战机，通过灵活的走位躲避敌人的攻击，并通过战机之间的牵引电流消灭敌人。



游戏截图

5.游戏元素

游戏元素主要分为四类，主角类，敌机类，地图类，渲染类：

- 角色控制（主角的位置移动，护盾生成等）与主角的显示（尺寸，状态，生命）分离。
- 敌机类继承自 Enemy 的 Prefab，有着丰富的接口用与游戏难度的调整。
- 地图类 MapH, MapV 分别做为水平和垂直墙体元素，通过尺寸调整，构成了丰富的地图。
- 渲染类主要是实现音效，动画和相机的特殊效果。


6.玩法说明


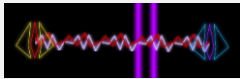
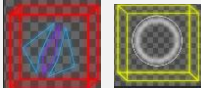
- 开始界面 Help 按钮导向帮助向导界面，建议玩家先从 Help 界面学习基础操作。
- 开始界面 Start 导向模式选择界面：

生存模式(Survive)：玩家需要保护好自己的两架战机，用战机之间的连接消灭敌人，生存时长显示在左上方，可以随时选择暂停或返回主界面，随着时间的推移，敌机的攻击方式越来越多样，难度也会逐渐变高。当任何一架战机血量低于 0 时，游戏结束，以生存的时长作为游戏的最终得分。游戏结束后可在 Ranking 界面看到自己历史最高分。

闯关模式(Battle)：玩家可以进行选关，每通过一关会解锁下一关，即使玩家在中途返回主界面，通关信息也会保存。一共设计了循序渐进的 12 个关卡，通关的条件是消灭场景中存在的所有敌人，随后找到场景中的目标点，到达目标点即可通关。

图例说明：

名称	图标	说明
Enemy		以自转姿态运动，按初始设定的方向移动，碰到墙体反弹，主角碰撞后扣血（可被护盾抵挡），碰到链接后消失，以下敌人攻击方式均包含此类。
Enemy_static		往往不会运动，被链接破坏后按一定速度向周围发射一定数目的 Bullet。
Enemy_spin		以主角为原点做圆周运动，半径恰为主角之间链接长度（即不做出反应便会撞向另一侧的主角）。
Enemy_swing		常以整齐队列的形式移动，周期与位移可调。
Enemy_track		具备追踪能力的敌机，初始速度较慢，会不断加速。
Enemy_Copter		运动方式与 Spin 相同，定期向主角所在位置发射 Bullet。
Battary		向主角定向发射一定数量的 Bullet。属于地图类敌机，不可被破坏，主角碰撞与碰撞墙体类似。

Bullet		子弹，主角碰撞会扣血（可被护盾抵挡），只有碰到墙体或主角会消失，碰到链接不会消失。
Player		正确消灭敌机的方法是用主角之间的电流，若敌人直接碰到主角会扣血，随后主角有三秒中的无敌时间，两架战机中任何一架血量低于 0 则游戏结束。
Supply		主角触碰后相应地增加生命或护盾数

7.特色功能

7.1 与众不同的玩法

想到战机游戏，往往会联想到射击和躲避子弹；想到双手摇杆，往往会想到控制移动和技能操作。我们的游戏 Tango 打破了这一常规，用纯粹的摇杆控制移动和战机链接电流消灭敌人的方法，更考验了玩家的双手配合能力，而且入门门槛低，适合各年龄段的玩家。

7.2 相机跟随与地图设计

为了达到更好的用户体验和真实的战机操作感，本游戏在游戏摄像机上做了深入的尝试，不但为战机添加了镜头跟随功能，还扩展了各式各样的全局地图，游戏地图均为自主设计，我们还为每个地图关卡起了符合其特色的名称，保证游戏流畅性的同时，也尽量传达出我们想通关这款游戏给玩家传达的故事。

7.3 过渡界面与相机缩放

如上一条所言。我们在游戏摄像机上做了深入尝试，当然不止于相机跟随，在一个普通场景中，我们设置了 4 台摄像机: UI Camera, Designer Camera, Map Camera, Main Camera，为了解决操纵杆触点的坐标系问题，我们专门使用了分离的两个相机：UI Camera 用于渲染所有 UI 元素，Main Camera 用于渲染游戏元素节点，Designer Camera 是我们在地图设计时可以用方向键控制的移动相机，Map Camera 用于界面切换时的相机缩放，为了达到连贯的游戏效果，我们借鉴了 Pewpew 的关卡过渡方法，在每个场景开始时会给我们设计的地图一个全局特写，然后镜头拉近直到我们的主角位于镜头中心，同样的，场景结束时同样会有此效果，一来让场景切换不显得那么突兀，二来让玩家在场景开始时对地图有个大致的认识。

7.4 动画效果与图片光影

Tango 的静态图片素材均是我们自己手工设计的。游戏整体走科幻风格路线，偏几何简约风格的游戏元素，再搭配上赛博朋克的音乐，使得整个游戏主题一致。观感上，我们采用主色调为黑色的背景，配上打上高斯模糊的主角，营造出发光的效果。游戏中使用了丰富的动画效果，动画素材是在 giphy 这个专业动图网站上精心挑选的，每一帧图像都要先经过透明化处理后才能使用，虽然很费时间，但是为了做出令人满意的游戏，我们认为这是值得的。

8.设计难点

8.1 摇杆操纵

由于 cocos 没有自带的摇杆组件，所以在实现摇杆操作上费了一番功夫，其本质是通过摇杆圈（ring）和摇杆点（stick）的相对位置获得方向和速度大小，然后绑定到对应的主角身上。一开始由于对 cocos 的父子节点的坐标系统不是十分清楚，加之相机跟随导致的触点坐标非场景坐标的问题，所以绕了不少弯路。这一问题直到使用了多组相机分别渲染的方式才得以解决。

摇杆控制的另一个难点在于运动感的模拟。我们一开始的解决方案是在摇杆对象的更新中对主角的位置进行更新，但是这样带来的问题是在主角和地图边界发生碰撞时，难以阻止主角穿越边界。后来，我们在主角对象中加入了加速度和速度属性，在摇杆的更新中，只对主角的速度进行更新。在主角对象的更新中，才根据速度来对主角的位置进行更新。此外，我们还在主角对象中加入了阻力项，在每帧的更新中主角的速度会衰减为原来的 99%。我们在电脑上使用模拟器模拟时，发现这样做的话战机移动非常逼真，我们一度非常满意。但后来在手机上进行调试时，我们发现战机虽然移动的非常真实，但是操纵感非常差。于是我们参考了现有的操纵感比较好的游戏（如PewPew、王者荣耀等），**发现他们的操纵是移动操纵杆即开始移动（即无加速过程），停下操纵杆移动即停止（即无减速过程），且碰到墙以后是只有沿着墙壁方向的速度（即无反弹）。**于是我们设计出了现在这一套速度系统。

8.2 主角链接

Tango 的玩法核心是通过主角间的链接消灭敌人，所以在链接的处理上我们花了较大功夫，一是链接的角度和长度要与两个主角适配，这要求我们对 cocos 的坐标系统有着深入的了解；二是链接的拉伸效果，链接本身具有弹力属性，在玩家不操作时，主角将会会到平衡位置，而且在场景内需要存在一定阻力，否则两主角会不停做简谐运动，这很影响游戏体验，并且链接用到的图像不能简单套用尺寸变化的模板，图像变形严重，所以我们使用 Tiled 和 animation 结合的方法，想出这个方法时，我们都觉得非常巧妙（这在背景图的处理也有所体现），使得链接就像从主角之间自然牵引出来的一样。

8.3 敌机设计

敌机的运动上我们尝试了多种方式，包括直线加速，圆周运动，螺旋线和 cocos 自带的 runAction，在设计时，我们也为敌机运动开放了多个可调的接口，可以设定敌人的各种运动参数（运动半径、运动速度、运动距离等），便于设计者完成游戏场景的设计。一方面所有敌机都是 prefab，充分利用敌机的共同特征，另一方面采用了继承，使得我们的代码复用性强，敌机可调节范围广。

8.4 地图与相机

在这个游戏中，我们使用的地图中边界都只有两个方向：纵向和横向。通过对这两种边界的组合，我们设计出了总共 12 个不同的地图，并分别在 12 个关卡中得到使用。

地图边界为细长的长方体，并且使用了 Cocos 提供的矩形包围盒。当玩家与地图边界发生碰撞时，会调用玩家对象的 onCollisionEnter 函数。在这个函数中，我们让玩家对象向

撞击方向的反方向后退一定距离，防止玩家冲出地图边界。

地图的设计是在 Cocos Creator 中完成的，我们将地图的边界制成 Prefab，并在设计窗口中，将边界拖到场景内进行调整。

如前所述，Tango 中广泛用到了相机，这是得益于之前学过一段时间的 AE。本质上相机的作用是在有限的显示空间中发掘出游戏的更多视角，其实对于一个第三人称射击类游戏来说，相机跟随并不是必要的，但为了满足地图设计的需要和更好的用户体验，我们认为这是不可或缺的。

在机型适配上，由于我们一开始就选择了横屏界面，所以避免了“刘海屏”或相机头遮挡的问题；我们使用全局地图，所以不用考虑不同机型的尺寸的显示问题（主相机拍摄到的完全占满手机显示空间），用 UI 相机专门渲染 UI 界面（摇杆，按钮等），所以我们只需对所有游戏场景重复一套相对布局的 UI 界面即可解决适配问题。

8.5 动画音效

在动画上，我们没有使用 cocos 的粒子系统和拖尾效果，一是资源包大小的限制，二是我们觉得风格不符，我们使用了最基本的 animation clip，把优秀的动画用在最常见的地方。在音效上，我们对背景音乐和游戏音效采取了不同的处理方式，由于我们的游戏节奏较快，采取的是切换场景但背景音乐不重播的方式，并且由于资源大小的限制，我们适当地对背景音乐做了剪辑。

8.6 场景切换

UI 系统主要包括三个场景，即开始菜单（startMenu）、模式选择（modeSelect）、关卡选择（stageSelect）。每个场景中的对象都只有按钮（Button），我们使用了 Cocos 提供的 Button 类，所以对 Button 的底层实现不需要进行关注。我们在每个场景的总控制器中（和 Canvas 相关联），对每个 Button 添加了监听事件。当具体的 Button 被按下时，我们会触发相应的事件，一般为场景切换。

我们还在场景切换的过渡中选择了镜头拉伸的过渡方式，看上去简单，实际上需要很多次的调试和精确的计算才能达到此效果，由于地图的不同，我们需要为各个场景调试不同的过渡时间的过渡初末方位。在闯关模式里我们还采用了一层层解锁的方式鼓励玩家探索，我们希望用这种循序渐进的方式让玩家一边熟悉操作，一边保证玩家的兴趣。

8.7 本地存档

谈到场景切换，不得不提到本地存档，因为游戏中需要记录玩家的通关情况，即使玩家中途退出，这些信息也应该保留。在游戏关卡选择中，以及游戏场景的结束后的过渡场景中，我们需要知道关卡信息（上一关的关卡信息、某一关是否通关等等）。原本我们想要通过在本地储存一个 log.json 文件，使用这一文件进行信息储存。但是我们后来发现，使用浏览器端的 JavaScript 对本地文件进行读写较为复杂，并且发现 Cocos 提供了一个 cc.sys.localStroage 的 API，通过类似哈希表的方式可以实现本地存储，因此我们使用了这一个 API 作为我们本地信息存储的方式。

我们还用同样的方法对玩家的历史最高分进行了记录，在无尽模式结束后，玩家点击下方的 Ranking 按钮即可看到自己排名前三的得分。该内容存储在用户本地，即使退出游戏，再次上线时，记录依然会保留。

9. 游戏测试

我们在设计通关模式之初就对游戏的难度和可玩性进行大量的测试, 对于游戏的难度我们请了自己的室友进行试玩, 在确认关卡设计的合理性之后 (主要是保证游戏能够全关通过)。我们对请了很多不同机型用户进行试玩, 包括在安卓机和苹果机, ipad 上的运行。由于在设计之初就考虑了适配问题, 并选择合适的方法进行解决, 在机型适配上没有遇到太大的问题。

之后我们将游戏交给体验员进行测试, 一天之后我们得到很多重要的反馈, 包括地图的卡墙问题, 操纵杆的颜色对应, 操纵杆在相机的呈现层级等, 这些都是我们一开始没有想到的问题。感谢体验员的耐心试玩, 让我们可以把 Tango 做的更优秀。

10. 总结

这一次作业是我们进入大学以来第三次制作游戏。和前两次制作游戏不同的是, 我们使用了成熟的游戏框架, 并且使用 github 进行合作和版本控制。在有了游戏框架的帮助以后, 我们发现很多基础的工作, 例如碰撞检测、素材绑定、场景和各个素材的关系等, 已经不需要自己操心, 而是有现成的 API 可以进行调用。并且, 有了框架的帮助以后, 我们在开发中遇到的很多问题都可以在网络上和使用同样框架的程序员们进行探讨, 同时也有很多前人的技术博客可供参考 (这也显示了写技术博客的重要性)。

总的来说, 通过这一次作业, 我们对 JavaScript 的很多特性有了更加深入的理解, 对 Cocos 框架有所了解。同时, 我们也对游戏制作有了更加丰富的认识。如何制作游戏, 使得游戏更具趣味性和可玩性是我们一直在思考和讨论的问题。在近两周的开发过程中, 我们深感游戏开发的不易和困难, 在这里也向那些优秀的游戏的设计师和开发者表示敬意。

感谢助教和老师的帮助, 也感谢奇舞团的前辈们。