



一、项目分工

学号	名字	角色	班级	职责	贡献
15331300	王世祺	组长	晚上班	负责意见整合任务分配, 源代码、项目书的编写	25%
15331248	潘承远	组员	早上班	UI 设计与完善、源代码的编写与完善	25%
15331298	王瑞锋	组员	晚上班	源代码的编写与完善、项目计划书编写	25%
15331116	黄敏怡	组员	晚上班	源代码的编写与完善、项目计划书编写	25%

二、开发环境

Windows10, Visual Studio 2015, cocos2d-x3.10

三、项目阐述

名称 StarFish

简介 这是一款在星空下钓鱼的游戏, 游戏的灵感来自于捕鱼达人以及摩尔庄园里面的小游戏, 经过队员们的奇思妙想, 由原来的在海洋里面捕鱼变成了在星空下捕鱼

功能

- 1 由登陆界面和游戏界面两个界面组成
- 2 采用本地服务器, 在登陆界面登陆玩家的名字, 游戏结束后会自动显示 top5
- 3 在游戏界面, 可以使用左右键(AD 键)左右移动“海绵宝宝”, 这样能够尽可能快地钓到鱼
- 4 点击鼠标左键, 上下移动鼠标, 从而鱼线会随着鼠标伸长缩短, 这样能够钓到鱼
- 5 倒计时功能, 每次游戏只有 50 秒
- 6 计分功能, 每条鱼都有对应的分数, 水母触碰到鱼线或鱼钩会减分
- 7 自动提交分数和显示排名, 采用了本地服务器, 游戏结束后自动提交玩家分数, 并显示 top5
- 8 再玩一次功能
- 9 登陆界面和游戏界面采用不同的背景音乐

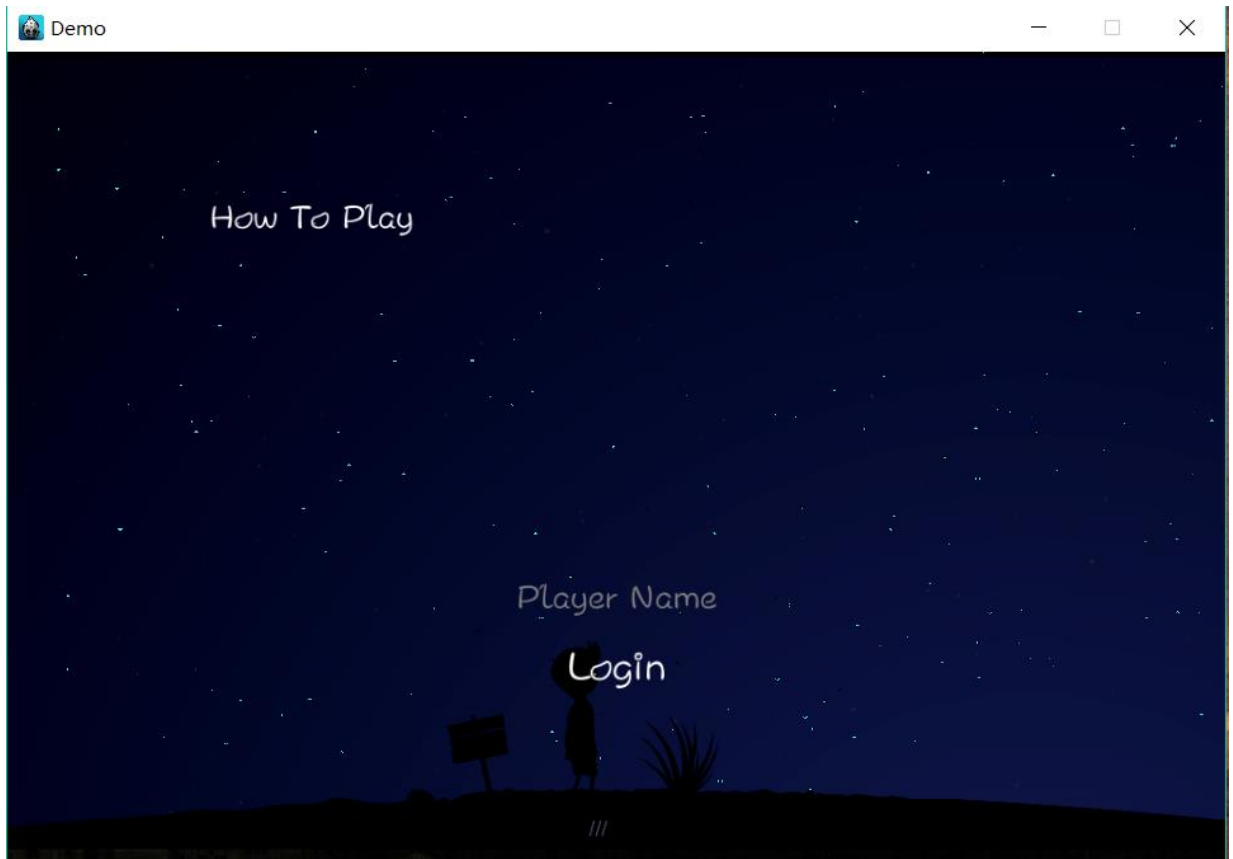
亮点

- 1 登陆界面的粒子系统与背景完美融合
- 2 登陆界面的和游戏界面的背景音乐与游戏主题很好地融合
- 3 鱼线会随着鼠标的上下移动而伸长缩短, 由于鱼钩必须保持原来的形状, 所以鱼线与鱼钩是两个不同的精灵
- 4 钓到鱼时“海绵宝宝”会高兴地感叹, 被水母电击后会发出惊叫
- 5 游戏由很多种“游动”的鱼, 鱼的资源质量很好, 能够吸引玩家
- 6 钓到鱼时“海绵宝宝”展示开心的动画, 被水母电击后会展示伤心的动画
- 7 采用本地服务器, 能够进行网络连接, 自动提交分数和显示 top5
- 8 游戏操作简单, 界面美观, 游戏新颖, 实现起来难度不低

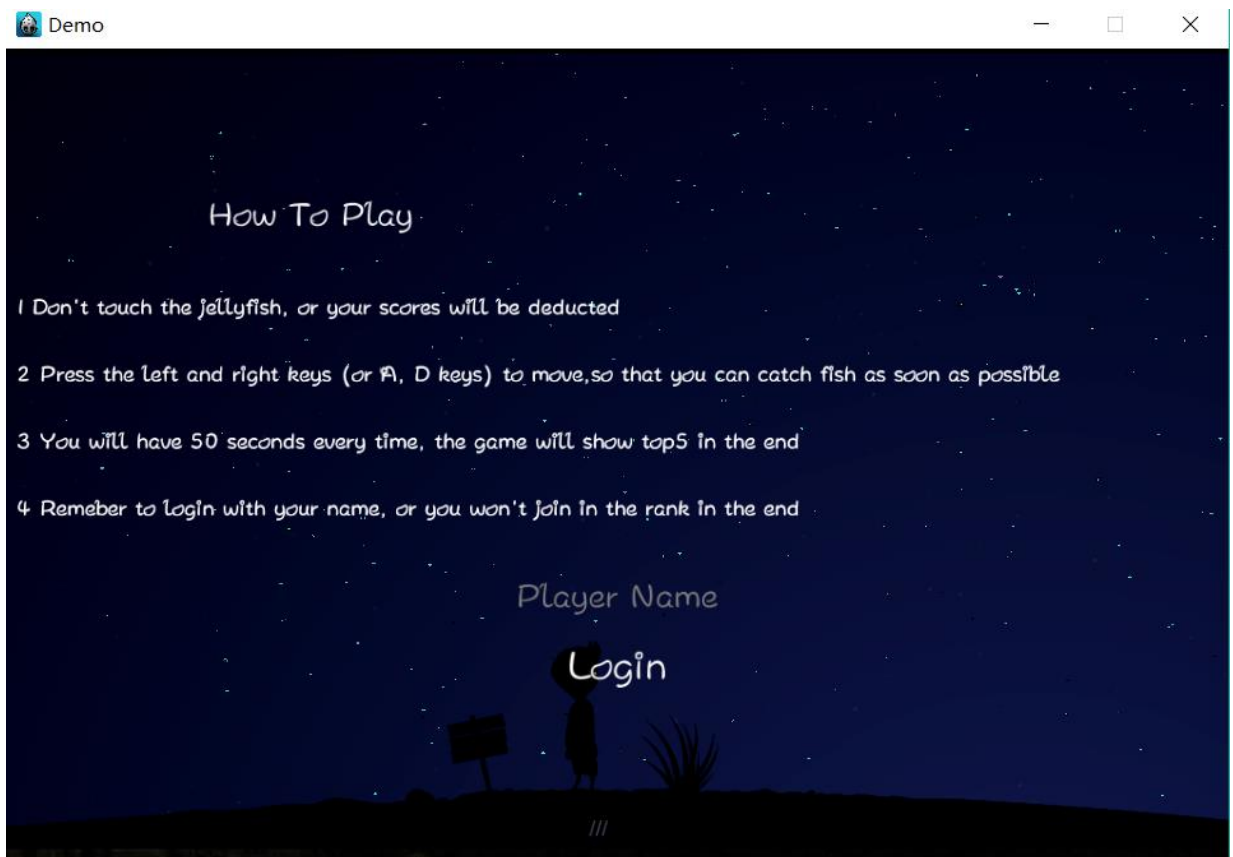


四、项目展示

游戏的登陆界面，粒子系统与背景营造出星空的画面

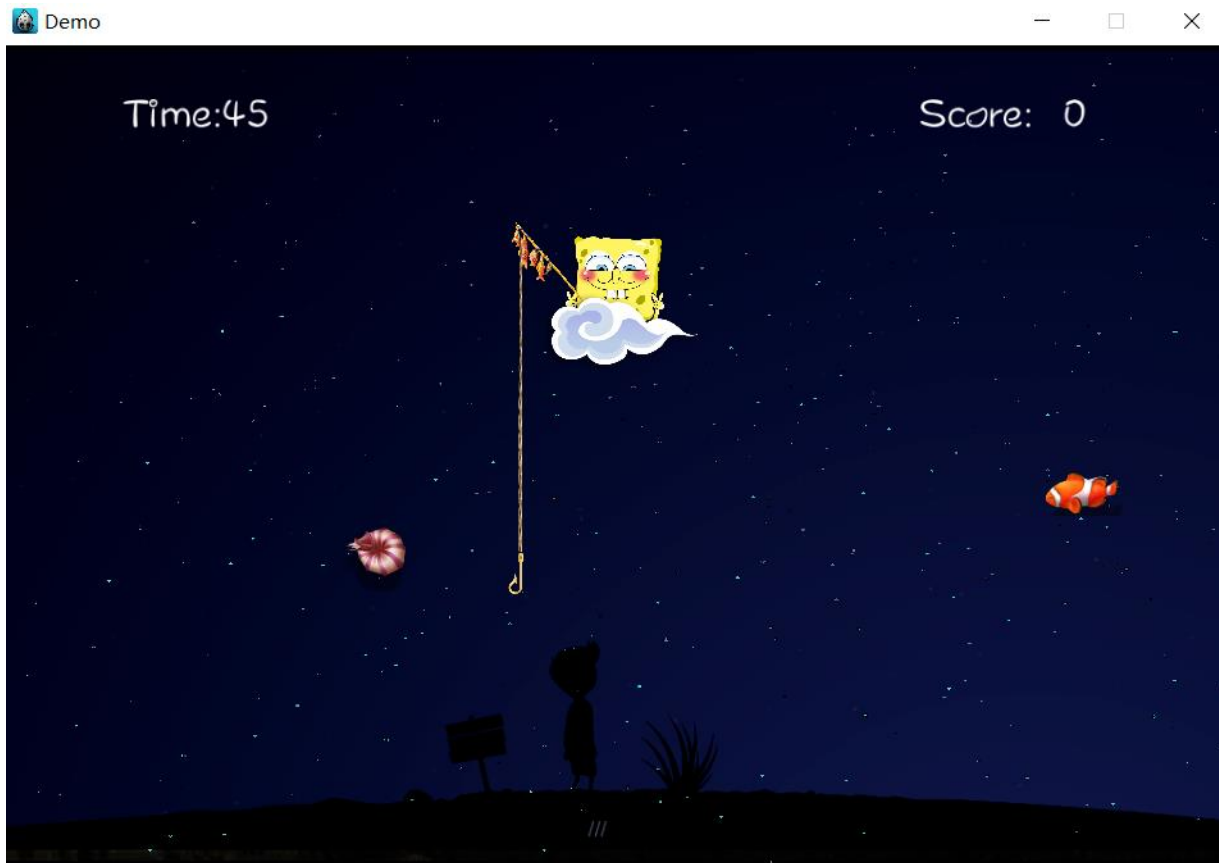


“How To Play” 说明游戏的规则与注意事项

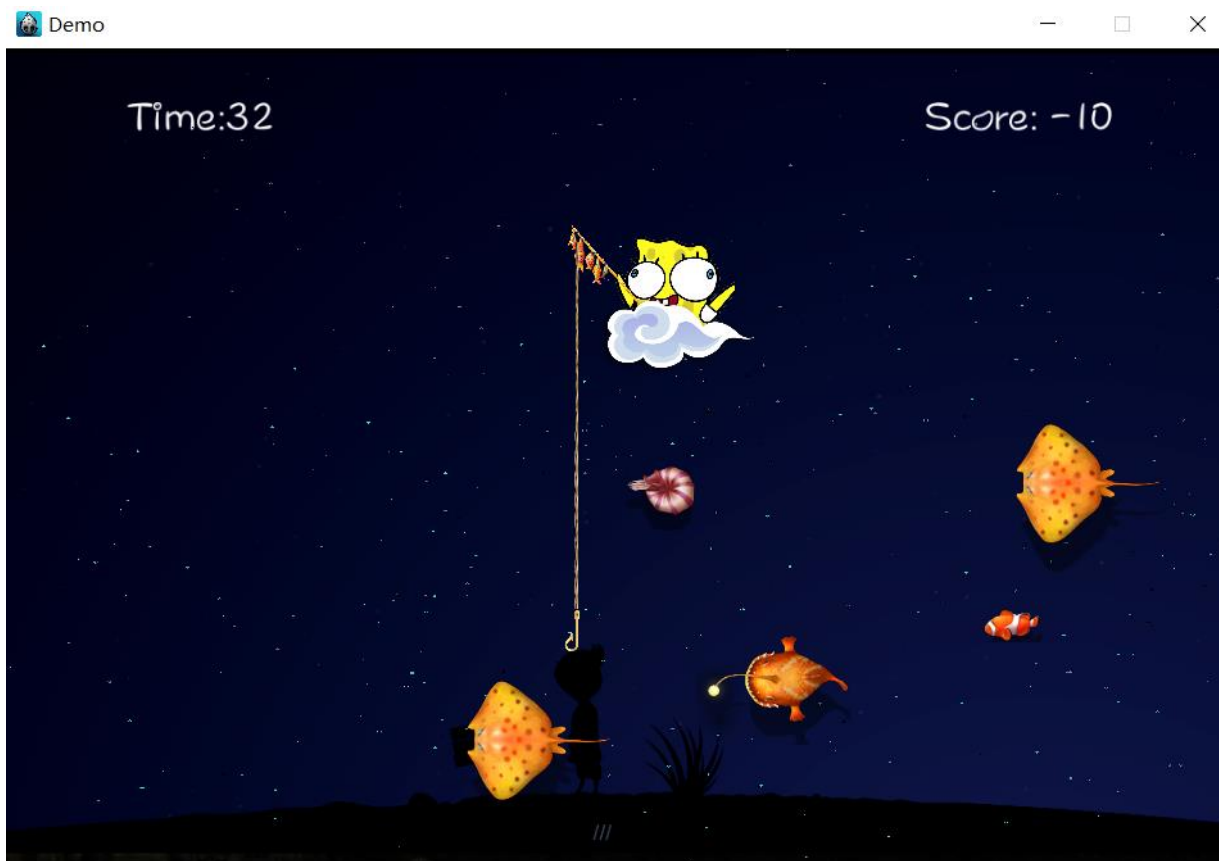




游戏界面，左上角为游戏倒计时，右上角为分数记录

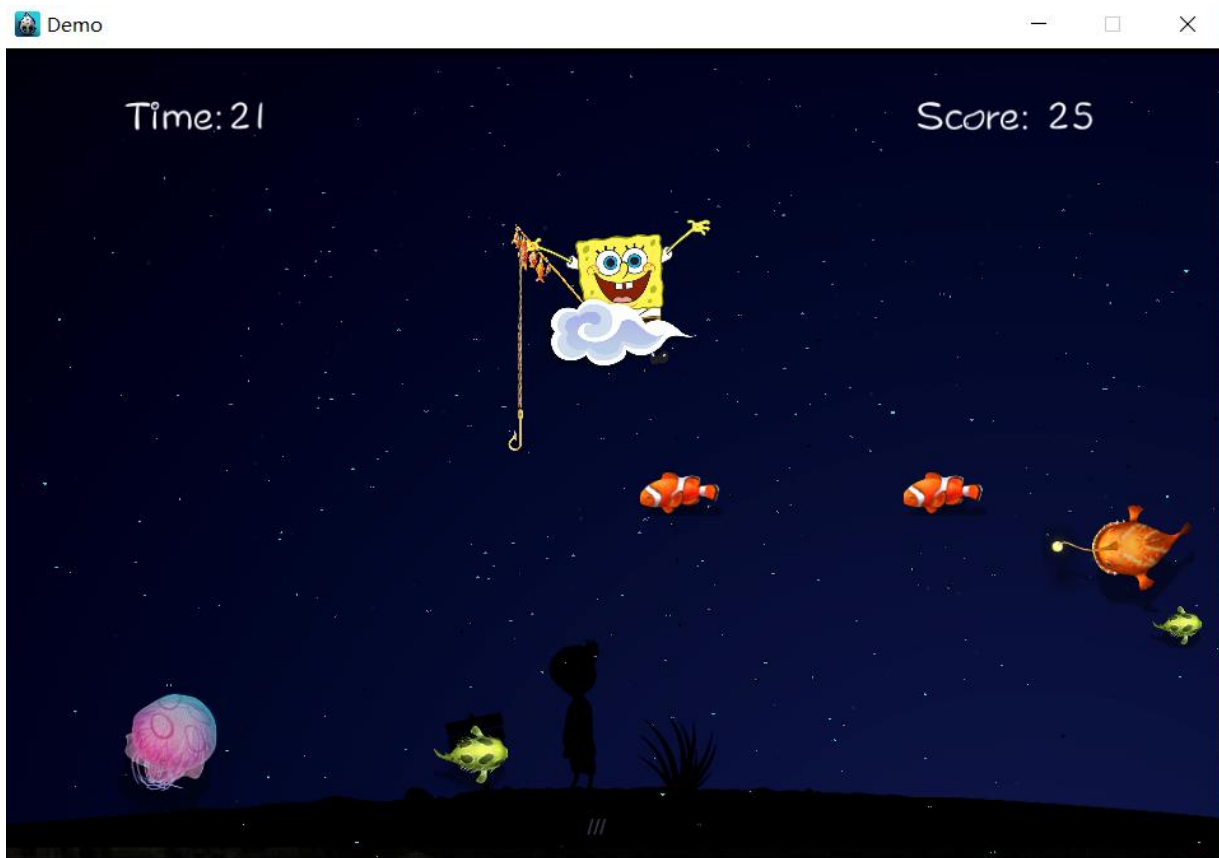


被水母“点击”后的悲伤动画，分数减为-10





钓到鱼后的开心动画，每一种鱼类都有不同的分数



游戏结束，显示 top5，此时可以再玩一次” Play again” 或者退出” Exit”





五、项目难点及解决方案

1、制作鱼的时候，由于鱼的种类、速度、大小等可能不同而分数不同，因此不知道要不要做一个鱼的类来定义相关参数。后来使用了 `setTag` 函数来根据 `addfish` 函数一开始定义的鱼的速度、大小等参数就可以直接设置分数，在钓到鱼后直接通过 `getTag` 即可获取分数

2、在制作鱼线的时候，打算使用鼠标的触摸事件，让鱼线随着鼠标的上下移动而伸长变短。首先对鱼线使用的是 `setScaleY`，鱼钩和鱼线是两个不同的精灵，鱼钩随着鼠标的位置上下移动，这样鱼线和鱼钩就显得不同步（一个使用 `setScaleY` 并且参数是通过鼠标位置换算过来的，一个则直接使用鼠标位置没有任何额外计算），最后的解决方案是通过一系列计算，使得鱼钩随着鱼线的底部上下移动从而消除了鱼钩和鱼线之间的不同步。

```
auto loc = touch->getLocation();
auto pos = yugou->getPosition();
auto strin = fish_string->getPosition();
auto x = visibleSize.height - loc.y;
float num = x / 800;
fish_string->setAnchorPoint(Point(0.5, 1.0));
fish_string->setScaleY(num);
auto yscale = fish_string->getScaleY();
int y = strin.y - fish_string->getContentSize().height * yscale;
yugou->setPosition(Vec2(strin.x - 2, y + 1));
```

3、鱼的来回移动，我们首先想到的是像第 13 周的作业处理子弹那样将离开屏幕视野的鱼儿消除。但是发现这样处理比较繁琐，还需要添加很多代码，为了简化工作，我们直接添加 `if` 判断条件，将离开视野的鱼使用 `setFlippedX` 转向即可。

4、倒计时和分数统计。因为在前几周的实验中遇到了类似的问题，所以本次实现起来的时候就显得比较轻松。

5、记录不同玩家分数的功能。首先想到的是使用本地的数据库，但是后来经过组员的商量，大家觉得使用第 15 周的本地服务器更加简单，并且这样可以和不同的用户比较分数，而不仅仅是使用本电脑玩游戏的本地玩家。

6、游戏资源的寻找。我们组的人员认为一个游戏不光要好玩，更加重要的是它的 UI 界面。美观的 UI 会在第一时间吸引玩家，之后玩家考虑的才是游戏到底好不好玩。于是我组人员对游戏素材的选择与 UI 界面的设计格外注重。花了较多的时间寻找 UI 素材。

六、项目总结

游戏项目完成后，找来了几个高中生和以前的同学试玩，他们给出的建议观点统计下来有以下几点：

- 1 登陆界面好看
- 2 背景音乐好听
- 3 背景图片太黑
- 4 游戏玩起来感觉很简单，难度不够高
- 5 登陆界面的背景和音乐搭配起来很舒适，以致于对游戏抱有比较高的期望，但是进入游戏界面后顿时觉得没有满足期望

参加试玩的同学对游戏 UI 的评价都比较高，但是都觉得游戏很简单。参加试玩的同学说可能是“王者荣耀”这类的大型手游玩多了，觉得这样的游戏过于简单。这确实值得我们思考，如



果想要使用较少的人力物力做出一款能够吸引广大用户的游戏确实是一件不容易的事情。这次项目的代码总共有一千多行，虽然最终做出来的效果和本组人员想象中的游戏还是有一些差距，但是大家对最终的成品都是很满意的，毕竟都是大家花了那么多时间与精力做出来的。

经过半学期的 cocos 学习，最终四个人花了两周的时间将游戏设想实现为一行行的代码，最终游戏出来的时候都感觉很欣慰。制作过程中遇到了很多问题，例如鱼线伸长的问题，从使用 `setScale` 到将一小段一小段的鱼线使用动画伸长，到最终使用 `setScale`，中间经历了很多过程，最终成功将鱼线这个问题解决。

在其他行业中，人与人的之间的沟通是很重要的。项目开发也不例外，很好的沟通能够加快项目的进度，这就要求我们每一个开发人员要学会和善于沟通。在一个项目的开发过程中，队员之间的沟通是一个不断交流和完善的过程。在开发到一定的阶段，我们就需要大家一起沟通已有功能，尽可能的去避免一些隐藏的问题，及时的发现问题，解决问题，从而按时或者提前完成项目的开发。

虽然我们现在的版本没有达到我们预期中的高度，也许是时间和人力问题。一个看似小小的游戏，后面的付出并不小。但是大家相信通过以后的学习会把游戏提升到更高的层次。

最后，感谢我们组每一位的努力和付出，能够在期末这段时间里将游戏完成。此外还要感谢 TA 们和老师的工作。