

iOS 游戏开发者成长记：从“菜鸟”到“牛人”

这是一个毫无编程经验的“菜鸟”在成长为 iOS 游戏开发者后，总结的成长史。短短 6 个月间，他从 Python 入门，发展到给 Mac 开发小游戏，再到自学 Objective-C 和 Cocos2d。除了你可能已经读过无数次的励志元素，文中的书籍、网站推荐以及开发过程中的经验教训总结很值得打算或刚刚进入苹果开发领域的新人一看。

我是菜鸟

几乎刚好是 6 个月前，我决定开始从事一项新的职业。2010 年 10 月之前，我从未接触过编程。要说有，那也只是在 6 年级时学过了大概一周的非常基本、基本、基本的编程知识。就这些。我不知会学成什么样。2010 年 10 月 9 日我在微博上写道：“我究竟想干吗？我决定学习编程，我没有任何的基础。唉，兄弟们，我有些担心。”这是一段生涯的开始，我很高兴能走上这条路。我不会回头。

开端

我将只快速简单的给你们讲一下我前几个月的学习过程，因为这段时间非常的枯燥。我从 Invent with Python 这本书开始，这是一本对新手来说非常不错的教程。为了让你们知道它是如何的初级，我向你们展示这本书里的一段话：

+ 这个符号是来告诉计算机把 2 和 2 相加。做减法用这个符号 -，做乘法用星号 (*)。

在我学会了一些非常基本的知识后，我随意的做了很多文本形式的小游戏，比如刽子手，jotto，猜数字。我的第一个比较大一点的小玩意叫做“大齿怪”。这个游戏的情景是关于一个大齿怪（就把它想成一个怪物吧），祸害村里。它藏在什么地方，你需要去猜它在哪：

A screenshot of my very first game.
This thing is going to make me a millionaire.

There is a man-rodent in the town and he has stolen money
from the townspeople. You have to find him and get their money back!

When you want to start, press enter

There are a number of places the man-rodent could be hiding.

Where do you think he is hiding?:

1. The bush
2. The broken treehouse
3. Uncle Crotch's Tavern
4. The barn
5. The floorboards

Enter the number of the location where you think the man-rodent is.

2

Nope. He is certainly not in the broken treehouse. Guess again.

3

Nope. He is certainly not in Uncle Crotch's Tavern. Guess again.

4

Wahoooooooo, you got it! The man-rodent is in the barn! Press enter.

www.WhitakerBlackall.com

也就是在此时我收到了我的第一张珍贵的粉丝画的作品。我把这个游戏发布在了 Reddit 上, Michael Hussinger 为它制作了这张令人惊叹的封面画:



接下来，我转移到了我的第一个半图形游戏：“扫雷”。在此时我才开始真正的理解构造一个灵活的游戏引擎有多么的重要（没错，就像我开发的这个游戏引擎）。这个游戏仍然是文本形式的，但已经有相当的灵活度了，在这个游戏里，我可以指定布景的大小和地雷的数量。这样一来游戏就可以从头再玩，你可以把地雷数量设置很小而很快能看到游戏的结果，这样有趣多了。下面就是这个游戏：

欢迎使用苹果 iOS

此时，我已经感觉到应该转向真正的图形游戏了。我感觉陷入了僵局。我根本不知道从哪里入手。我研究了一下 SDL，这个东西超过了我的接受能力，我很快放弃了。因为我知道最终我想做 iPhone 应用程序，我请教了 Matt Rix (Trainyard)，他给了我一些很好的建议。他建议我从学 C 语言开始，然后学

Objective-C，然后 Cocos2d。今天说起这些我很快高兴，这三种语言我都学会了，虽然很基础。

在学习 C 和 Obj-C 时，我遇到的大量的挫折，我没有做出多少游戏。很多次，我真的怀疑是否能渡过难关、找到出路。很多东西当我第一次看到时愁得我挠掉了许多头发：结构体，数组，多维数组，属性，表现层，控制层，协议和代理，内存管理，还有很多。很多东西我目前仍然无法理解领会。但是我坚持继续下去，我对很多人都说：让自己进步的唯一方法就是不断的做游戏。我十分清楚，想成为钢琴家，练习时根本，所以，这也是我需要的（我就持续这样下去）。

欢迎使用 Cocos2d



一旦我最终把 iPhone 应用的基本知识攻克下，我立即开始研究 Cocos2d。我很紧张，也很兴奋。现在，我不知道自己是太挑剔还是太笨，我感觉 Cocos2d 的文档写的不太好。所有的类，方法，属性都列出来了，但却很难让人读懂，它们解释的不太清楚。一个例子：每个” node “（Cocos2d 的主要元素）都有一个”（BOOL） isRunning “属性。看起来非常的简单，但它似乎能够表示任何意思。所有的文档对这个属性的解释都是：” 标志这个 node 是否在运行。嗯，ok … 不用说，Cocos2d 里面有很多东西我都十分的困惑。

尽管这些的不满，我还是非常喜欢 Cocos2d。一旦掌握了一些基本知识，我就开始研究它如何能够容易的做出一个简单游戏，让它马上能跑起来。我已经知道了如何做音乐和声效，我还知道一些 Photoshop，我就真的开始要让它输出一些很酷的结果了。起初我对 Cocos2d 很困惑，我从 Ray Wenderlich 的一些非常棒的教材开始学习。

我开发的各种暂且算是游戏的游戏

下面的是我的第一个 Cocos2d 游戏，它是我直接按照 Ray 的教材开发出来的（我在里面还加上了我的音乐和当一个敌人被消灭时的很傻的音效！）：

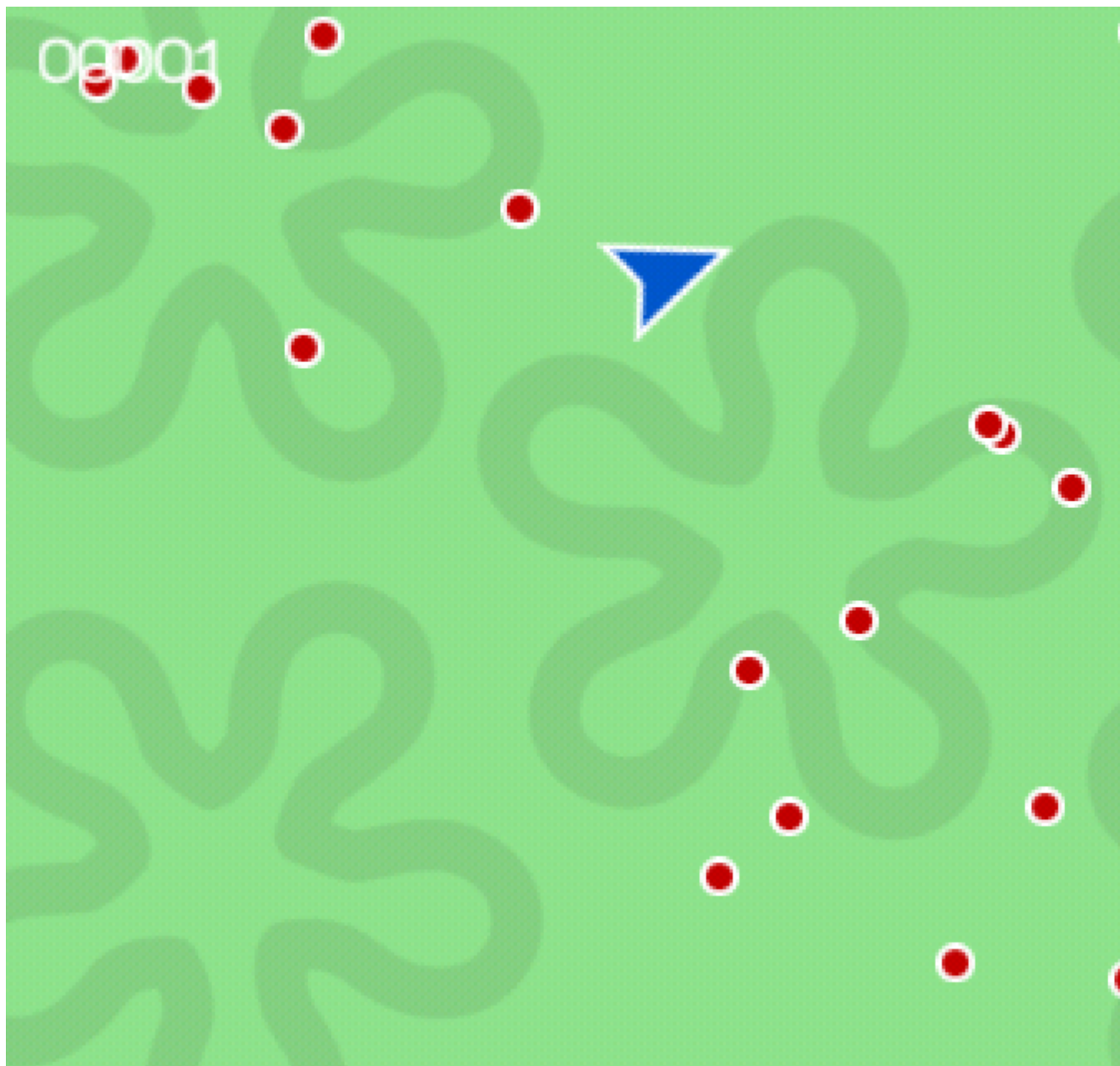


接着，我仿制了一个 Pong。这个游戏还可以多人参与：



Pong 之后，我受到了鼓舞，决定开发 Tetris。这是个巨大的错误。我根本没有准备好。我遇到了各种让人懊恼的错误和问题，当我最终把框架搭起来后，我在微博上写了这样的话：“我很骄傲。我为我的 Tetris 仿品开发了一个真正复杂的类/子类系统。” Noel Llopis 很快的告诉我，说我的决定十分的错误。我搜集了一些教材，试图好好的修改一下这个仿 Tetris 游戏，但太困难了。我决定放弃，但有一天我会打败你的，Tetris!

这次 Tetris 失败的尝试之后，我受 Tilt to Live 的启发做了一个游戏初始原型，我计划在这个游戏里，你需要去收集红点来获取积分，而不是像其它游戏里去避开它们。现在做的这个里面并没有加入真正的游戏里面会有的音乐：



学习了更多的 Cocos2d 之后,我突然来了灵感要做一个循环的褶皱纸样式的背景。于是把一张纸做出褶皱型,以循环方式填充背景。然后我在上面放了一个小圆片,你可以用加速器让它在上面滚动。当我弄明白如何让这个小圆片滚动后,我为自己能做出这个东西很自豪,因为这牵涉的一些数学知识,我并没有用很久的时间就把它能明白了(尽管只是一个很简单的几何问题)。背景很难看——但很好用:

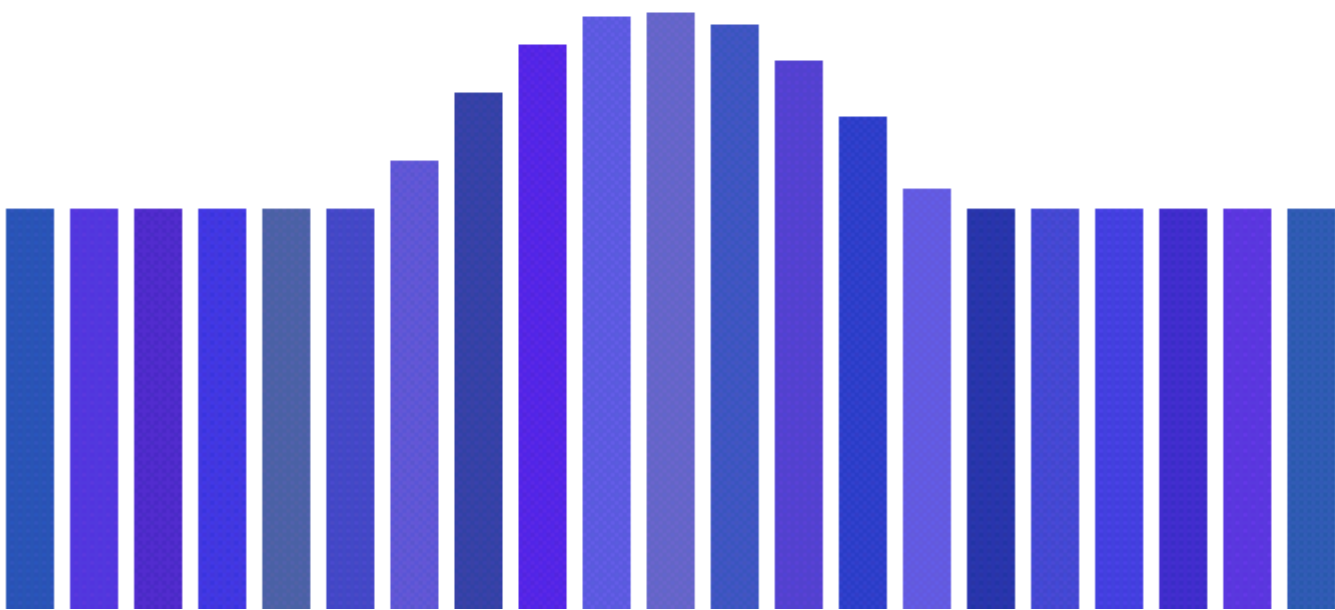


接着，当我看到了即将发布的游戏 Bumpy Road 的令人惊讶的海报后，我受启发，心想是否能够也做一个跟里面的高高低低的路面相似风格的游戏元素。很显然，我的做图水平很有限，你可以比较一下。

Bumpy Road:



我的版本:



这之后，我想在真正的动画上试试身手。以前从没有做过动画，先在 Photoshop 试试。人物的制作很乏味冗长。即使是画一条很傻很短的线条也很费劲。但我还是完成了，我把他称作“狂欢者”。他能做的就是一遍遍的张开和闭上他的大嘴，每次都随机的发出一些哼哼声。你还可以在屏幕上移动他。背景闪动不同的颜色。当你触碰屏幕时，他只能哼哼和张嘴：



然而，“狂欢者”有了一小队粉丝。这些粉丝的头像图标里包括 Mario, Link, Samus, 等等。事实上我把这个游戏发给了 Touch Arcade 的 Jared (嘿, 是他向我要的)。之后, 让我大吃一惊和无比高兴的是, Retro Dreamer 的 Craig Sharpe 创作了我的第二张正规的粉丝的艺术作品! 下面就是 Craig 为“狂欢者”创作的杰作:



他的作品是如此的优秀，我甚至把它放入了我的下一个游戏里。但起初，我突发奇想，要做一个跳杆子的游戏。起初的出发点只是为了使用一个简单的游戏技术测试一下视差背景的效果，但做下来就做成了这个游戏。最终的游戏有一个小屏幕，有记分功能，有重玩功能，有音乐、音效，各种跳跃高杆，有一个高度指示器，甚至还有一个鸣谢页面！它是我的功能最全的一个游戏。我在游戏里画了一些非常丑陋的精灵，就像你下面看到的：

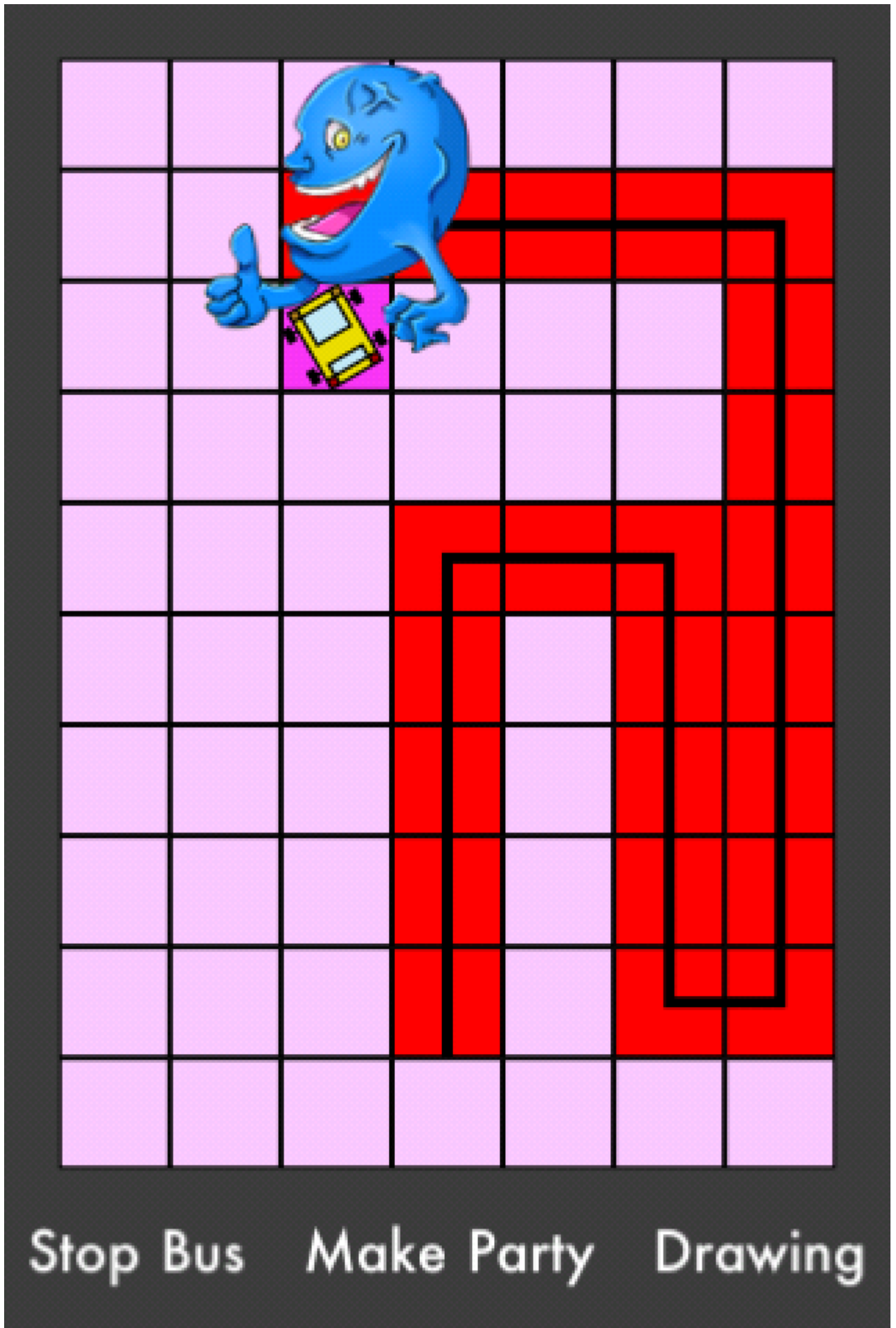


因为画的太差,我询问 Michael Hussinger 是否愿意为我做一些简单的修改,他同意了!他在二十分钟内就画出来一个让人惊叹的匹萨伙计。我估计花几个小时也画不出这么好的作品来。他还为游戏中的视差效果画了一个漂亮的山作为背景。下面的就是在他的帮助下游戏的样子,我加上了标题和鸣谢页:



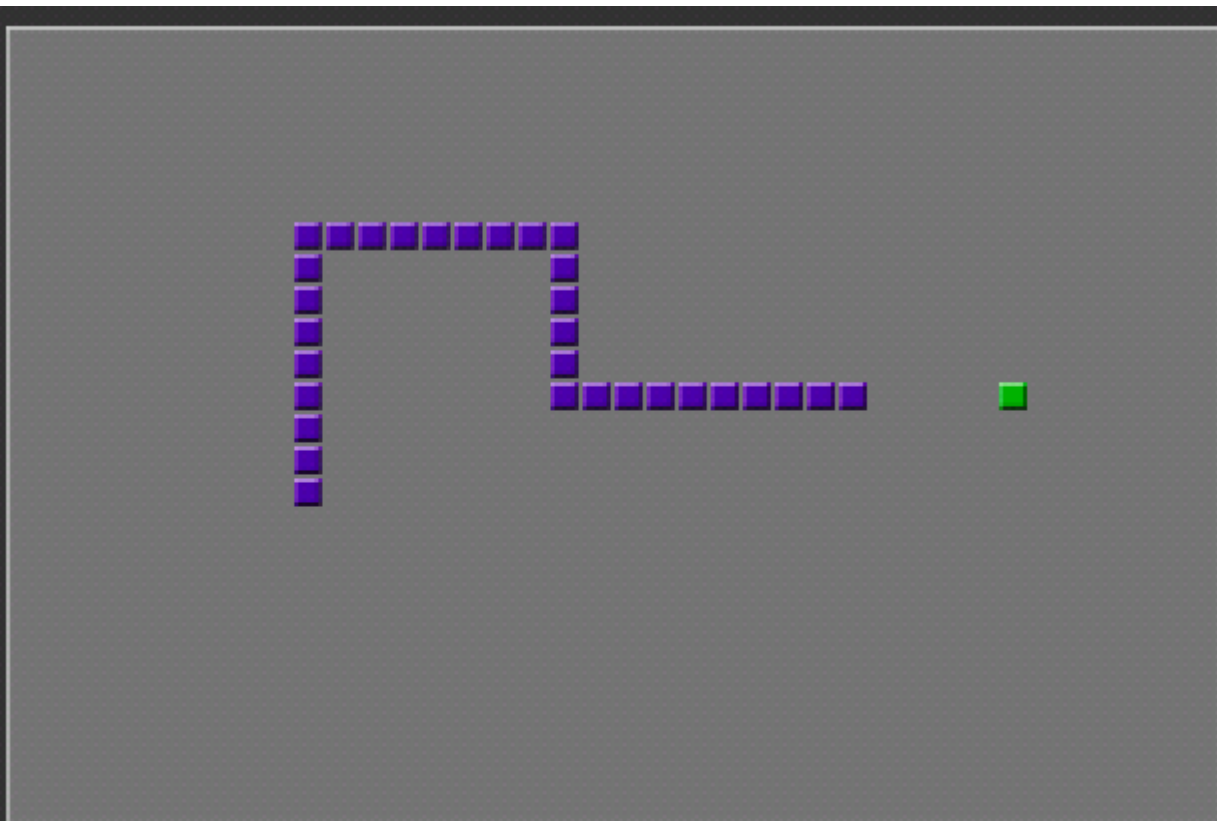
游戏进行中

我的下一个游戏受 Trainyard 的启发。我想看看是否能做出一个地图，上面画些网格道路，卡车能跑在上面。看看，制作“狂欢者”有了很好的回报。所以我称这个为“狂欢大巴”。游戏的玩法就是点击地图上的一个地方就设定一个聚会点（一个闪烁的，颜色变换的点，伴有音乐），然后图上会画出一条路，通向这个点，“狂欢者”会把他的“狂欢大巴”开到那里。不用说，这是一个非常简单的游戏。但里面很酷的东西是，当他到达目标地点后，音乐声会变大，卡车会开始跳舞，Craig 画的作品会出现在开车上面左右摇摆！



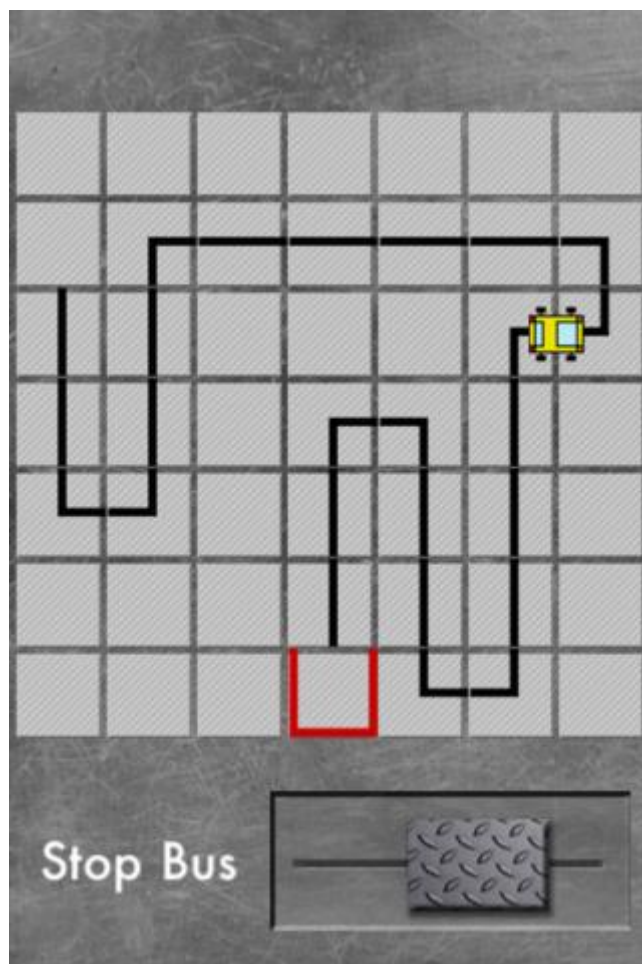
代码写的很乱，有很多问题。我让弟弟试玩了一下，他试了三次，什么都没干成，他总是每次都是在错误的时间点击到错误的地方，这样游戏根本就不能正常玩下去。每次我都要从头开始玩这个游戏，因为我没有给它加入“重新开始”的功能。但我还是挺为这个 Trainyard 的复制品感到自豪的。Matt 应该为此感谢我。

我的下一个项目是要做一条蛇。我发现了这个如何用 flash 做出一条蛇，我只需要把它移植到 iPhone 上。这个教材让我明白了很多如何做这种游戏的内幕。之前，我计划让蛇每一帧移动一次，但在如何让蛇尾和蛇身保持一致的问题上遇到了很多的麻烦。我本打算建一个数组存所有历史动作，蛇尾将按照这个数组里的信息做它的每个动作，但这样一来它动的太快，让人很不舒服。这个教程整个颠倒了我的思维，我意识到蛇其实根本没有动。这种看起来在移动的视觉效果其实是简单的在帧上添加和移去一小段蛇身产生的。这样，当你捡起一个苹果，所需要做的只是告诉程序不要删除任何片段。事实上我需要重新写这个程序，因为界面太难看了，控制的也不好。因为我是移植过来的，代码写的很乱。但事实上作为一个基本的游戏来，它还是很不错的！

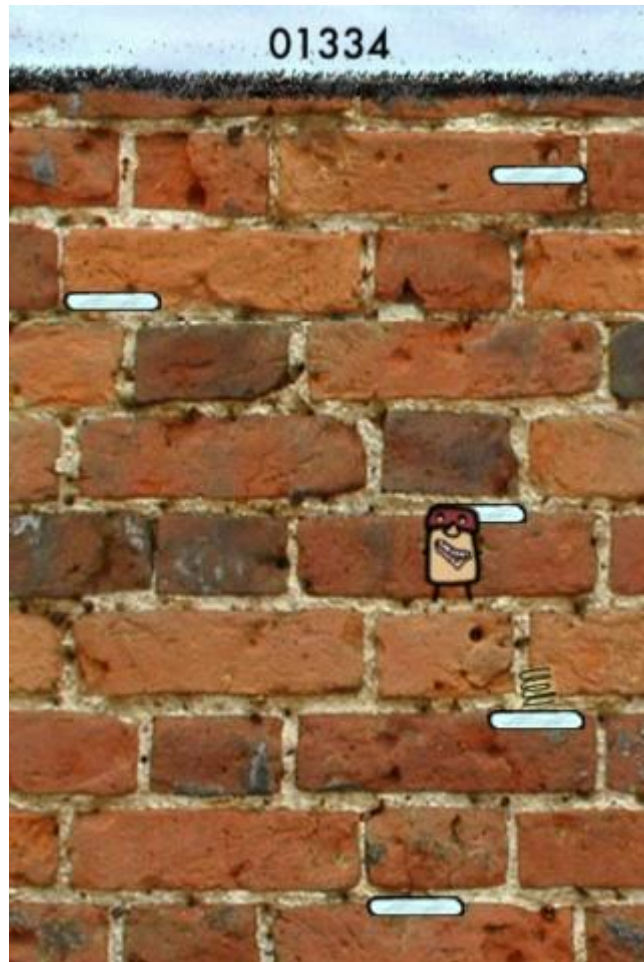


SNAKE

在完成这条蛇之后，我打算制作“狂欢大巴 2”：“狂欢者”的周末狂欢。我把上一版升级了一些。我在里面加入了一些 Trainyard 里面的速度条，一个车站，车可以从里面开车，一些声音特效（当他被碰着时会出现很大的汽车相撞的声音，而且会喊出“今天没有活动！”），程序也整理的更整洁了。但没多久，程序又开始变乱，当我要纠正一个问题时，另外一个东西又出现了问题。最大的问题是当我把速度设成最大时，卡车会在路上某个点崩溃。这是因为更新调用的太频繁，程序来不及在下一次调用前更新完。尽管我知道这个问题的原因，如想修复这个问题需要重做大量的东西。我想我需要升级到“狂欢大巴 3”：最后的狂欢。



我最近的目标是重做 Doodle Jump。目前为止一切进展顺利。里面很多小平台，移动速度会约来越快，一个会跳的小人，一些其它物品（目前为止只有一个弹簧），自定义音乐和声效，以及完全从原作里拷贝的背景。目前为止我对做出的东西很高兴。我会继续往里面增加很多的东西。



结论

练习给人带来的进步是惊人的。每次当写出一个新程序，我都会遇到大量的无法预知的问题和 bug。当遇到挫折时，我一般是将问题移到下一次升级时解决。这让我感觉自己就在走一个 while 循环：

```
while (编程技术仍然很糟) { 练习时间++; 程序 * 新程序 = [程序 程序类型: 游戏]; if (新程序.做的很好吗 == 是的 && 练习时间 >= 大量) { 编程技术仍然很糟 = 否; } }
```

我会一直这样制作下去，一转眼就到现在了，我几乎等不及想知道再 6 个月后会是什么样。请下次当我来述说如何给 Casey' s Contraptions 制作音乐（和一些声音）时，请记住一定来看看！

