

# Dino:bit 诞生大事记

本文将讲述 Dino:bit 是如何诞生的！共分为一下七个部分。

注：最终呈现效果均在本目录下的\1.2 里面，具体操作流程请参考 \dino:bit 使用说明书.docx。

以下某人以\1.2 为当前目录。

## 零、硬件准备

3 到 4 个 USB 接口的电池盒（学校的 micro:bit 板的 battery 接口似乎坏掉了……）

1 块 micro:bit 板作为 radio 的接收器（Receiver）、3 到 4 块 micro:bit 板作为玩家传输数据的传感器

电脑（笔记本或者台式均可），配有 vscode 或者 pycharm，有 pip（能在 vscode 终端使用 pip 或者 pycharm 库里有 pip）

其余：没了！我们的硬件就是这么省！

## 一、idea 的诞生

此 idea 由陈奕诺想出并加以完善弯腰、大招等构想。

信息系统任务布置当天，彩笔陈奕诺想信息教室到被 ban 掉的 chrome://dino，突发奇想：为什么不自己搭建一个小恐龙呢？

1. 小恐龙要本地能运行，不会被 ban；
2. 小恐龙更有趣味一点：跳起来！弯腰！可以放大招！

于是，Dino:bit 就诞生了！

## 二、单人小恐龙的按键及数据传输的实现

首先单人小恐龙只需要完成 Up、Down 与“face down”“down”&“freefall”的映射即可，具体参考【五、体感】部分。

数据传输最开始是由张朝单点传输，即一个 group 里面实现“Receiver”和“Player”的数据交互，不过后来被陈奕诺发现所有“Player”共用一个公共信息区域 radio 这个 bug 而 failed 了。

## 三、多人联机小恐龙的按键

该部分主要由王翰宁完成。

因为要实现同一台电脑上控制多个小恐龙，所以会出现键位冲突的情况，所以王翰宁在小恐龙源代码程序的基础上，写了\templates 里的 index.js 和\templates\bins 里的 gameLib.js，来操控每个小恐龙的键位。

具体而言，index.js 是用来渲染 index.html 中的内容的，它实现了引入游戏库 gameLib，和获取一个 id 为“bl-script”的元素中名为“iframeCount”的属性值。

这里王翰宁使用游戏库 gameLib 中的函数创建 iframe 元素，并将它们添加到网页中。对用户按键事件进行监听，根据用户按下的键的不同，调用不同的游戏库函数来控制游戏的行为，例如跳跃、下蹲、加速等。

对用户松开按键事件进行监听，根据用户松开的键的不同，调用不同的游戏库函数来控制游戏的行为，例如停止下蹲。

每隔 100ms 检查所有游戏是否结束，如果全部结束则在页面中显示得分，并重新开始所有游戏。

gameLib.js 定义了一个变量 gameLib，并且声明了一些函数。这些函数都是控制一个或多个小恐龙游戏的函数。其中，gameLib.createIframe 函数用于创建一个或多个小恐龙游戏的 iframe（就相当于在网页中显示网页）。

gameLib.eachIframe、gameLib.specificIframe 和 gameLib.exceptspecificIframe 函数用于操纵每一个小恐龙游戏的 iframe，或者操纵特定的 iframe，或者操纵除了特定的 iframe。其他函数例如 gameLib.pressJump、gameLib.downDuck 和 gameLib.upDuck 用于控制小恐龙游戏中的小恐龙跳起来或者蹲下来，以及加速和停止加速。

这个 gameLib 对象还提供了其他的函数，例如获取游戏分数和获取障碍物的属性。最后，这个 gameLib 对象被导出为一个模块，使得其他代码可以使用这些函数来控制小恐龙游戏。

因为要实现小恐龙的大招功能，所以王翰宁在原本小恐龙源码的基础上添加了 206-207，693-700 行代码的内容，以实现放大招的功能，即控制其它小恐龙让其加速度变为原来的两倍。

## 四、多人信号的无线传输

最开始我们的想法是通过蓝牙低功耗实现快速传输，本来由张朝完成。但是最后发现学校 win7 的电脑上没有蓝牙……所以王翰宁和张朝采用了另外一种方案，就是将一块 microbit 板子接入电脑做监听，另几块 microbit 板子做发射。

然后，我们需要解决一个问题：多个 radio 在一个共同的 group 里，消息是互通的，因而 radio.receive() 是某一块 micro:bit 顺次查询所有 radio 组内的消息。于是陈奕诺想出，需要在最初连线、确定每块 micro:bit 的编号时取用最后一组数据，并通过玩家双按按钮、充当接收板的 micro:bit 最终确认玩家人数的方式实现初始化。

其次，keyboardcontrol 文件是接收插在电脑上的 microbit 发来的串口消息，这段代码的作用是从串口读取消息，并将其转换为键盘输入。首先创建了一个串口连接，用于与另一个设备进行通信。然后定义了一个字典，将不同的消息映射为对应的键盘按键。接下来进入一个无限循环，从串口读取消息并进行处理。如果消息是 Jump（跳跃），则发送对应的键盘按键；如果消息是 Low（低姿态），则先按下对应的键盘按键，再释放；如果消息是 Acc（加速），则发送对应的键盘按键。

具体 player 和 receiver 的代码在 \microbit-python\ 里

## 五、体感：Jump、Bend down & Fun Accelerate

总体采用 microbit 的 accelerometer.current\_gesture() 判断 microbit 板朝向。

1. Jump: 模拟跳跃过程中的自由落体，即 “freefall”；
2. Bend down: 即 “face down” 或者 “down”；
3. Fun Accelerate: 给别人加速！让别人撞上仙人掌！“left” 或者 “right”，对应左侧弯腰和右侧弯腰。

## 六、优化和补锅

后期的优化和补锅由陈奕诺和王翰宁共同完成。调试工作由陈奕诺完成。

首先我们（王翰宁和陈奕诺）需要对小恐龙进行倒计时，其次小恐龙需要自动跳起，以优化游戏体验。

然后，在 google 小恐龙框架上，我们将障碍物间距调大、恐龙速度调慢，降低游戏难度。

倒计时部分做了个/countdown 然后重定向了两次。

多人选择部分，原先有一个点击取消的 bug，被王翰宁修复了。

最终最高分数显示部分，陈奕诺调整了输出，最终分数以两位小数呈现。

## 七、实际问题和优化措施

在后期的实际调试中还遇到了诸多新的问题和新的挑战（需要伟夫斗争），这里大概分为以下几个主要问题：

1. 在最开始我们并没有怎么考虑过加速后再减速的问题，然而没有减速的游戏体验感极差：即，一个玩家直接左弯腰……
2. 由于后期所有代码工程由王翰宁和陈奕诺两个 Coder 完成，遇到了变量名不统一的问题（以及王翰宁自己变量名也没有统一），比如 exceptspecific 和 specific 搞混了（可以参考 index.js 里的内容），以及 index.js 里王翰宁将 stopaccelerate 打成了 stopAccelerate（花絮：因为这个，陈奕诺查了两天的代码）
3. Python 实际运行速度远不如 c++ 和 javascript，因而会出现 python sleep(20) 太长，陈奕诺在这里调整了读取 radio 信息的方式，具体参考 \microbit-python\1.1receiver.py，这里经历了 1.0 到 1.1 的迭代更新。都被解决了，完美！

此外，还添加了单人一分钟计时赛。

## 八、其他

实际上，我们经历了从 Dino:bit 版本 1.0 $\beta$  到 1.0 再到 1.1 再到 1.2 的过程。

关于 microbit 离线改如何续电：竟然没！有！一！个！人！知！道！充！电！宝！不！能！持！续！给！microbit！充！电！

于是陈奕诺自备了电池盒子。

以及整个 dino:bit 主要框架是 javascript 和 python。

## 九、鸣谢

特别感谢主力 Coder 王翰宁的所有努力！Coder 张朝的所有努力！

特别感谢陈奕诺的 idea、后期代码完善、软&硬件补锅调试和自备电池盒！

特别感谢 google 的 dino 源码！

感谢张焯的内测和体验！

感谢沈老师、王老师的帮助，学校信息组的硬件支持！

## 十、最后的话

制作团队：TheUnknownTeam

组成人员（名字不分先后）：

201 王翰宁（Main Coder）、

202 张朝（蓝牙连接->micro:bit 组内传输 Coder）、

201 陈奕诺（Idea&后期 main Coder 兼软硬件补锅人员）、

201 张焯（体验&内测人员）。

Dino:bit 除 chrome-dino 框架外均为自主创作，开源但是版权归 TheUnknownTeam 所有。如有问题，请联系王翰宁 qq2634191898 或者陈奕诺 qq2720262181。