



创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

章节内容

基础背景	1
创意程设思维(CREATIVE COMPUTING)是什么?	1
Scratch是什么?	2
这份指引手册是什么?	2
这份指引手册是给谁用的?	3
要运用这份指引手册, 我需要准备什么?	3
这份指引手册涵盖了什么?	4
我应该如何使用这份指引手册?	5
这份指引手册从何而来?	5
单元 0 – 开始出发	7
Scratch 简介	10
Scratch 账号	12
设计手札	14
Scratch 小惊奇	16
Scratch 工作室	18
小组讨论	20
单元 1 – 探索发现	23
依照程序来跳舞	26
循序渐进	28
10个程序积木	
30	
我的工作室	
除错(Debug)!	32
关于我	34
	36
单元 2 – 动画	39
演出程序	42
组乐团	44
橘色正方形、紫色圆形	46
动起来了!	48
除错(Debug)!	50
音乐影片	52

单元 3 – 故事	55
人物	58
对话	60
场景	62
除错(Debug)!	64
打造生物	66
故事接龙	68
单元 4 – 创造游戏	71
梦想游戏清单	74
新手村游戏	76
记录得分	80
延伸变化	82
互动课题	84
除错(Debug)!	86
单元 5 – 深入沈潜	89
我知道、我想要、我学到	92
第二回合	94
进阶概念	96
硬件与延伸套件	100
活动设计	102
我的除错(Debug) 谜题	106
单元 6 – 黑客松	109
专案游说	114
项目规划	116
设计冲刺跑	120
项目回馈	122
项目检查哨	124
非焦点团体	126
成果发表预备	128
最终成果发表	130
附录	133
词汇表	135
Standards (编按: 为美国课纲准则, 保留原文未进行翻译)	141
程序设计思维	143
延伸阅读	149
网站连结	151
台湾十二年国民基本教育科技领域课程纲要草案	153



基础背景

为了帮助您尽快融入创意程设思维的世界，我们已经拟定好了针对八个常见问题的答案：

1. 创意程设思维(Creative Computing)是什么？
2. Scratch是什么？
3. 这份指引手册是什么？
4. 这份指引手册是给谁用的？
5. 要运用这份指引手册，我需要准备什么？
6. 这份指引手册涵盖了什么？
7. 我应该如何使用这份指引手册？
8. 这份指引手册从何而来？

歡迎來到
創意程設思維
教學課綱
指引手冊！

创意程设思维 (CREATIVE COMPUTING)是什么



创意程设思维是关于：创意创造。年轻族群早已对于计算机科学与计算相关领域不再陌生。然而，这些初步的认识却与他们的兴趣与价值观没有连结：过度强调技术细节，而非创造的潜力。创意程设思维，可藉着发挥个人的创造力、想象力以及兴趣喜好，协助建立个人与程设思维之间的关联。



创意程设思维是关于：授与能力。年轻族群中的许多人是以消费者的角色来使用计算机，而不是作为设计师或创作者。创意程设思维着重于传达：年轻族群创作动态与互动类计算机程式媒体时，所需的相关知识、实际案例及基本素养。这些类型的作品在日常生活中也同时受到年轻族群的喜爱。



创意程设思维是关于：编写程式。投入创作计算机程序作品可让年轻族群不仅学习到要成为计算机科学家或程序设计师该做的准备，也能协助年轻族群发展为程序设计思考者。程序设计思考者会将程序设计的概念、实际案例以及观点运用到生活中的各种层面——无论哪种领域与情境。

SCRATCH是什么？



有众多的工具可运用来培养创意程设思维。在这份指引手册中，我们选用Scratch。Scratch是一种免费/自由的计算机程序设计语言，可由以下网址取得：

<http://scratch.mit.edu>。运用Scratch，人们能够创作广泛多样的交互式媒体项目，像是：动画、故事、游戏以及更多可能性。此外，还能将程序项目与在线社群的成员一同分享。自从Scratch在2007年五月启动以来，全世界成千上万的人们已经创作并分享了超过六百万个的程序计划项目。

这份指引手册是什么？

这份指引手册搜集了运用Scratch程序语言来介绍创意程设思维的相关教材，涵盖了：构想、策略方法与各种活动。各种活动都设计用来增进程设思维与创造力的流畅度与熟练度。更精确地说，各类型的活动增进了关键程序设计概念(次序—sequence、循环—loops、事件—events、平行—parallelism、条件—conditionals、运算符—operators、数据—data)与实际操作(试验

—experimenting与重复—iterating、测试—testing与除错—debugging、再利用—reusing与融合—remixing、抽象化—abstracting与模块化—modularizing)的尝试与探索。想获知更多关于程序设计思维的细节，例如：什么是程序设计思维？如何评估学习成效？，可见手册附录或前往：

<http://scratched.gse.harvard.edu/ct>

受到建构式学习方法的启发，这份指引手册中的活动皆着重于以下原则：

原则 #1:
创作

提供机会让学习者融入设计与制作，而不是只有聆听、观察与使用。

原则 #2:
个人化

提供机会让学习者融入对个人而言有意义且相关的活动中。

原则 #3:
分享

提供机会让学习者融入与他人(像是：听/观众、指导者与共同创作者)互动的活动之中。

原则 #4:
回想沈思

提供机会让学习者回顾并再次思索他们具创意的程序计划作品与实际演练过程。

这份指引手册是给谁用的？

无论你目前的背景或之前的经验，这本指引手册本质上是设计给广泛各式各样的学习者与教育者。以下是一些可能运用这份手册的角色以及他们可以如何使用：

K-12 教师

Scratch正被全世界数以千计的小学、中学、高中所采用。这份手册可完整用来作为一个学期的程序设计课程，或是选择性的作为其他领域课程的一部份。许多教学者将创意程设思维作为课后或午间课程，并运用手册中的活动作为学生们开放式探索的灵感来源及学习基础。

博物馆或图书馆教育专员

除了像教室一般的正规学习环境，Scratch也已运用于像是博物馆及图书馆等非正规的学习空间。无论是作为结构化的工作坊体验或随时可进入的游玩空间，由于免除了传统场域的限制，这些学习环境非常适合协助探索创意程设思维。

家长

家长可用各式各样的方式来运用这份指引手册。无论是在家自学的活动、创办学校里的创意程序设计社团、在小区活动中心举办工作坊，我们十分鼓励家长思考如何运用这个指引手册来协助年幼学习者学习与体会创意程设思维。

每個人
都可以有
創意程設
思維！

大专院校讲师

在计算机科学课程中，Scratch可以作为基础程序设计概念与实际操作的前导课程，随后再转换至更加传统以文字为基础的程序语言。例如：哈佛大学的CS50课程在转换到C程序语言前，系采用Scratch作为程序设计的前导体验。手册中的活动亦已运用作为大专院校程度的教育、艺术与媒体素养等课程的一部份。

年幼学习者

自从Scratch启动后这过去七年，年幼学习者在许多不同的情境下已成为创意程设思维的热情倡导者。例如：将程序设计介绍给家长与老师、或是为同学们创造学习机会，创意程设思维已经不仅是为他们所设计的事物，而已经成为他们能够主导并完成的事物。

要运用这份指引手册，我需要准备什么？

在准备：时间与开放探险的心态之外，一些重要的资源包含如下：

- + 有音响喇叭的计算机(非必要性的麦克风与网络摄影机)：用于以计算机作为基础的设计活动。
- + 网络联机：用于联机到Scratch网站(若你所处的环境没有网络联机，可取得脱机单机版的Scratch)
- + 投影机或具有喇叭的交互式白板：用于分享进行中的程序计划及用于示范展演。
- + 设计手札(数字或实体)：用来写笔记、素描、激荡构想与执行计划。

这份指引手册涵盖了什么？

这份指引手册由七个单元所组成 — 从最初的预备单元开始，最后以项目型态的单元作为结束 — 每个单元包含了6个活动。每个单元的摘要如下：

单元 透过探索可行性及筹备技术层面(例如：创建Scratch账号、启动设计手札)与人际层面(例如：建立讨论小组)的基础架构，来准备好创意程设思维的氛围。借着Scratch角色做出令人「惊奇」的事情，让学习者置身于最初始的创意体验。

单元 透过一系列提供不同结构强度的活动，让学习者逐步适应关键的程序设计概念：次序(sequence)。活动包含：循序渐进的教学指导、使用限定数量程序积木的创意挑战、让学习者自由探索并制作一个关于自己的程序计划。

单元 Scratch强调媒体运用。学习者在一系列着重于动画、艺术与音乐的活动中，试玩图像与音效。借着这些活动，让学习者探索关键的程式设计概念：循环(loops)、事件(events)与平行(parallelism)。活动包含：建立属于你的乐团、设计会动的生物、为喜爱的歌曲创作音乐影片。

单元 一起说故事来创作新的互动世界。首先，发展人物角色、学习编写对话程序，接着在场景转换中，安排这些角色与对话。将角色、对话、场景融合在一个大型的说故事程序计画。这个计划可在创作者之间流传，进一步成为故事接龙、或是重新构想为一个全新的故事！

单元 结合基本的游戏机制(例如：得分或等级关卡)与关键的程序设计概念：变数(variables)、运算符(operators)、条件(conditionals)。分析你最喜爱的游戏程序、想象新的游戏、并藉由实作(与扩充)经典游戏(例如：乒乓)来练习游戏设计，。

单元 进行最后单元之前，花一些时间来回顾先前单元的程序作品、进一步探索进阶概念、或借着设计新的活动或除错(debug)谜题来协助他人学习创意程设思维。

单元 借着一次又一次的「规划、制作、分享」循环流程，整合全部所学的程序设计概念与案例，来设计与制作你自己的程序项目。

学习成效评估的(策略)方式描述于指引手册之中，此外，手册的附录中包含了数种成效评估的工具。我们成效评估的方法是以流程导向为基础，并且着重于创造机会给学生谈论自己(及他人)的创作与创意案例。教师可采用指引手册中所建议的各种(策略)方式，来搜集许多流程导向形式的数据，例如：

- + 协助学生之间对谈、讨论他们的程序计划，运用录音、影片或文字留下纪录
- + 检视考察学生的程序计划作品集
- + 保留学生们的设计手札

我们以学生们实际完成的作品作为成效评估，并以这项评估来进一步协助学生们了解他们已经学会的东西，以及他们仍然想要学习的东西。成效评估可以纳入各种不同的参与者观点，像是：创作者本身、创作者同侪(学)、教师、家长与其他人。

我应该如何使用这份指南手册？

使用
多或少
随心所欲

设计
全新
活动

混合运用
手册中的
各式活动

选择
你自己的
冒险旅程！

我们鼓励你：随心所欲地大量或少部分使用这份指南手册、设计全新活动、并且混合运用手册中的各式活动。无论你先前的经验或专业是什么，我们认为每一位教育者都是创意程设思维体验的共同设计者。我们热爱获知你正在进行的教學项目，因此我们也鼓励你能提供文件数据，并透过ScratchEd在线社群 (<http://scratched.gse.harvard.edu>) 与我们及其他教育者分享你的经验。

我们采用「创用CC的姓名标示- 相同方式分享授权 (Creative Commons Attribution-ShareAlike license)」来发布这份指南手册。这表示您可以完全自由地使用、改变及分享这份著作，只要您适当的表彰著作人的姓名，并让他人可取得任何的衍生著作。

这份指南手册从何而来？

这份指南手册是由哈佛教育研究所ScratchEd研究团队的成员— Christan Balch、Michelle Chung、Karen Brennan所研发制作。Jeff Hawson 提供了修订上的协助及用之不竭的热忱。

这份手册的内容借鉴了先前版本的创意程设思维指引手册 (Creative Computing Guide, 2011发布) 以及创意程设思维在线工作坊 (Creative Computing Online Workshop, 2013年举办)。来自 National Science Foundation through grant DRL-1019396、Google CS4HS program、Code-to-Learn Foundation的协助让这些事情成为可能。

我们由衷感谢使用了先前版本并且参与工作坊的许多教育者。尤其，我们想要感谢广泛测试了第一版指引手册的教育者 (Russell Clough、Judy Hoffman、Kara Kestner、Alvin Kroon、Melissa Nordmann、Tyson Spraul) 以及广泛

审视目前这个版本的教育者 (Ingrid Gustafson、Megan Haddadi、Keledy Kenkel、Adam Scharfenberger、LeeAnn Wells)。我们也深深感谢我们的协作者。我们想要谢谢来自 Education Development Center's Center for Children & Technology 的 Wendy Martin、Francisco Cervantes、Bill Tally 以及来自 MIT Media Lab 的 Mitch Resnick，我们想要谢谢他们在建构发展程式设计思维框架与资源方面所作的广泛贡献。我们想要谢谢许多令人惊艳的哈佛教育研究所实习生，谢谢他们自从2011年初始版本以来这些年对于发展指引手册的贡献，他们分别是：VanityGee、Vanessa Gennarelli、Mylo Lam、Tomoko Matsukawa、Aaron Morris、Matthew Ong、Roshanak Razavi、Mary Jo Madda、Eric Schilling、Elizabeth Woodbury。

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

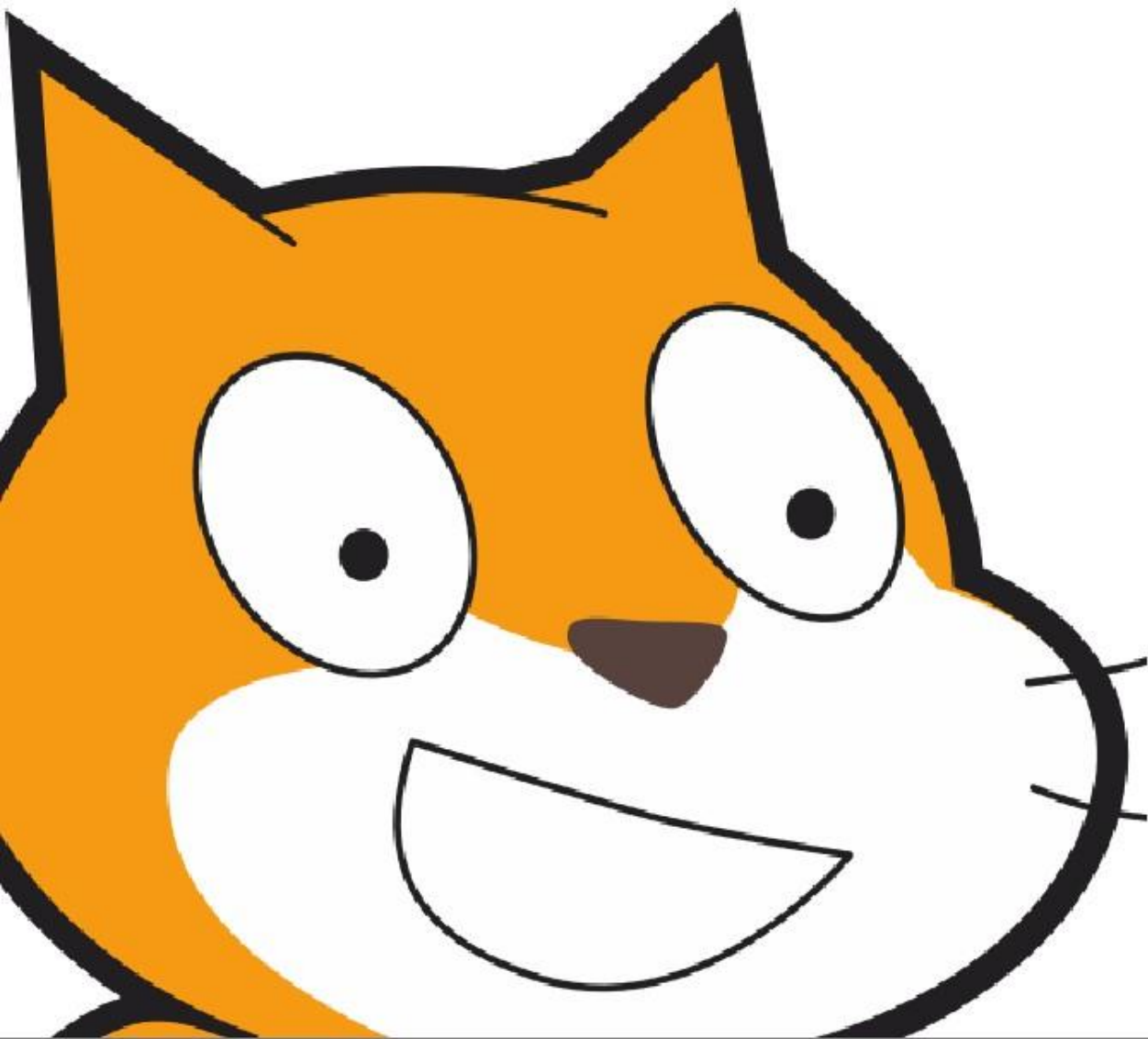
Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 0

开始出发



你的进度

本单元包含



SCRATCH 简介	10
SCRATCH 账号	12
设计手札	14
SCRATCH 小惊奇	16
SCRATCH 工作室	18
小组讨论	20

单元 0

整体概观

「大处」着眼

当作者等人与教师们分享这份教学指引的初稿时，普遍的第一个反应是：「单元0?! 为什么是0?」

我们希望传达的是：这是一个先备单元，透过创作、个人化、共享与反思，这个单元可以帮助你建立一个创意程设思维的文化氛围。我们建立这种学习氛围的目标将贯穿并落实于这份教学指引之中。

创意程设思维集合了智力层面，涵盖了一套程序设计概念与实务。它也包含实体层面，藉由桌椅、计算机的设置，鼓励与他人互动。最重要的是，它包含了情感层面，累积培养自信心与不惧怕的勇气。

若在你的班級中有這樣的氛圍將十分有幫助。從第一天就開始 - 讓孩子們理解他們將會開始犯錯以及我會要求他們作困難的事情。我會一直坦然接受。但剛開始的時候，孩子們並不會，因為他們想要的是成功。連大人們都不喜歡失敗或是犯錯。但我認為這非常重要 - 當你面對困境時，沒有時間讓你放棄或是哭泣。那是思考採取何種策略解決問題或是尋求協助的時候。你必須堅持下去，沒有理由崩潰或放棄。
小學老師 TS

学习目标

学生们将：

- + 在Scratch的情境下，开始认识程设创作的概念。
- + 能够想象他们自己创作Scratch程序的可能性。
- + 开始熟悉可以帮助他们程设创作的资源。
- + 藉由建立Scratch账号、探索Scratch工作室、制作设计手札及小组讨论，准备创作Scratch程序计划。

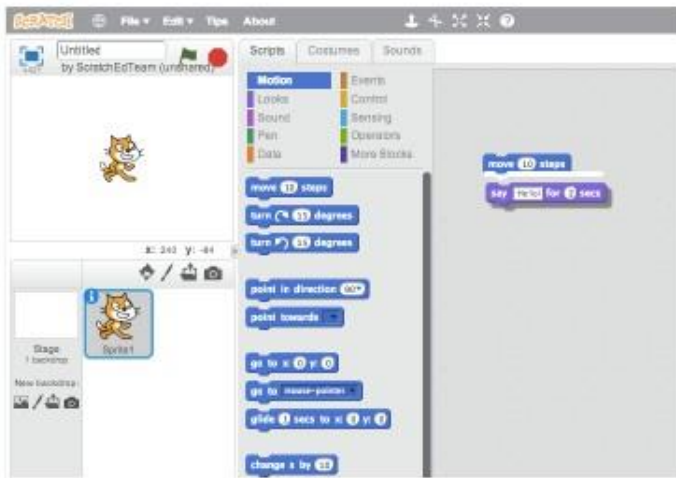
关键词、概念与实务

- + 个人简介编辑器 (profile editor)
- + 计划页面 (project page)
- + 工作室 (studio)
- + 小组讨论 (critique group)
- + 红、黄、绿 (red, yellow, green)

附注

- + 与信息部门（计算机室）协调来确保计算机可以连接到Scratch网站。
- + 没有网络环境？可以下载脱机单机版的Scratch：
<http://scratch.mit.edu/scratch2download>

选择你自己的冒险旅程

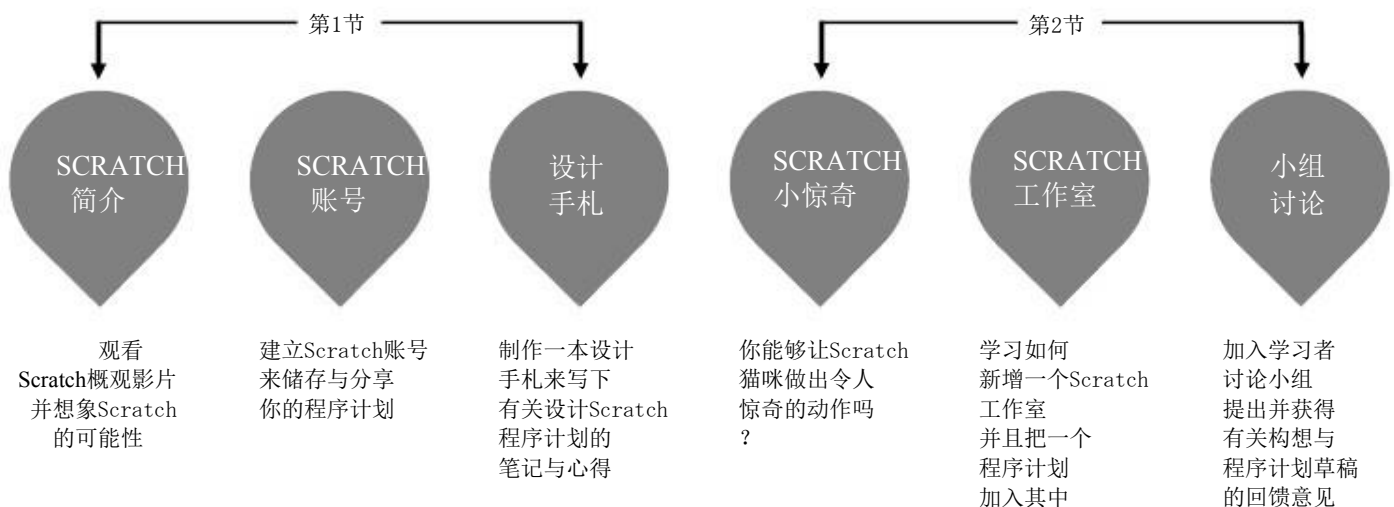


准备好开始了吗？这个单元是为了完全没有接触过Scratch的人所设计的。从「探索新奇程序计画」、「创建Scratch账号」到「尝试使用Scratch程式计划编辑器 (Scratch project editor)」，每一项活动都是设计来引导你与学生透过实际操作来开始了解Scratch。

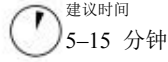
每一个单元会提供一些精选的活动，但我们也建议你调整活动的选择或顺序。不同的情境与参与者会有截然不同的体验。藉由混合与调整最符合你与学习者需求的活动，来选择你自己的冒险旅程吧。

仍不确定该从何处开始？看看下面所提供的建议学习途径与活动，来获得更多帮助。

可行的学习途径



SCRATCH H 简介



建议时间
5-15 分钟

目标

完成这项活动，学生将：

- + 借着观看Scratch概观影片或探索范例程序计划，开始认识如何使用Scratch开发环境来进行程序创作
- + 能够想象他们自己的Scratch程设创作的可能性

活动描述

- ❑ 运用右边的反思问题，询问学生们有关他们使用计算机的经验
- ❑ 以Scratch向学生介绍创意程设思维。藉由播放Scratch概观影片并展示一些会引起并激发学生兴趣的范例程序，向学生介绍创意程设思维以及他们用Scratch可以创作的程序范围。此外，并解说当完成未来的一些课程进度，学生便能够用Scratch创造属于他们自己的互动程序。
- ❑ 你会创作什么？要求学生们想象他们想用Scratch创作什么类型的程序计划。

教材资源

- ❑ 播放Scratch概观影片的投影机 (选择性)
- ❑ Scratch 概观影片
<http://vimeo.com/65583694>
<http://youtu.be/-SjuiaWRMU4>
- ❑ 范例程序计划工作室
<http://scratch.mit.edu/studios/137903>

反思问题

- + 当你使用计算机时，有哪些不同的方式呢？
- + 在这些方式当中，有多少种是创意的计算机互动方式呢？

审视学生成果

- + 学生们是否发想了各式各样广泛的程设计画想法呢？如果没有，尝试展示多样化的程序计划让学生感受到可能性。

附注

- + 如果没有网络，课前下载Scratch概观影片，从：
<http://vimeo.com/65583694>
- + 取代用笔写下反思问题的答案，可鼓励学生用创意的方式画出他们的回答（例：画出你使用计算机时的各种不同方式）

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

clicked

10

10 steps

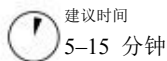
change color effect by 25

drum 4 for 0.2 beats

Welcome to Scratch! for 2 s



SCRATCH 账号



目标

完成这项活动，学生将：

- + 创建一个Scratch账号
- + 探索Scratch在线社群并审视Scratch社群指引

活动描述

- ❑ Scratch 在线账号需要填入一个email地址。如果学生无法提供个人或学校的email地址，可以使用老师或家长/监护人的email。若需要搜集，请事先规划同意书。
- ❑ 协助学生前往Scratch网站 (<http://scratch.mit.edu>) 并点选「加入Scratch」来开始创建Scratch账号。教室可以选择性地使用「Scratch账号讲义」来引导学生。保留一些时间让学生注册、更新他们的Scratch个人简介，并探索Scratch在线社群。鼓励学生练习登入并注销他们的账号。
- ❑ 为了让课程学员能够更简单地找寻并追踪其他人的Scratch账号，可以考虑建立一份Scratch使用者帐户与姓名的清单。
- ❑ 以小组方式检视Scratch社群指引来讨论良好及有建设性的行为。检视如何回报网站上不恰当的发言。

教材资源

- ❑ Scratch账号讲义
- ❑ Scratch社群指引(Scratch Community Guidelines) http://scratch.mit.edu/community_guidelines

反思问题

- + 你的Scratch账号名称是什么呢？
- + 能够帮助你回想密码的提示是什么呢？

审视学生成果

- + 学生们是否能够建立Scratch账号，并且登入及注销Scratch网站？

附注

- + 老师们可能会偏好提供他们的email或是建立一个班级的email信箱用来注册账号，以便收到任何在Scratch网站上不适当行为的通知。
- + 确认是否有学生已经拥有在线账号了。
- + 要记录密码并同时保留隐私？让学生在信封中写下他们的用户账号与密码，黏封起来并放在教室中的安全处。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

SCRATCH 账号

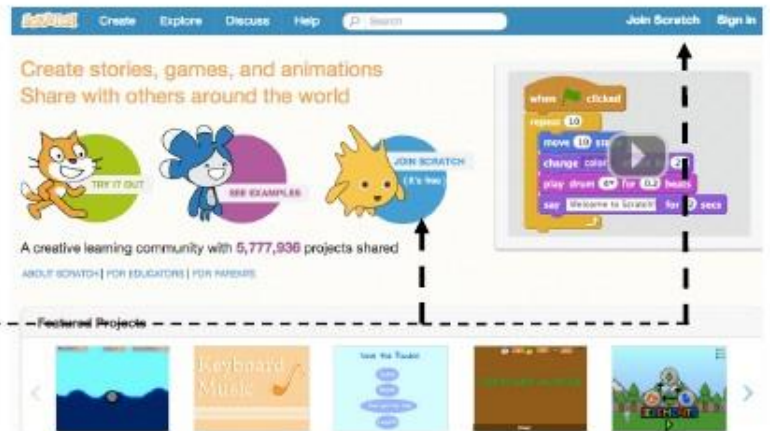
你是Scratch的初学者吗？
来建立自己的Scratch账号吧！

有了Scratch账号，你可以新建、保存并分享你的Scratch程序计划。下面的步骤会引导你建立新账号并设定个人简介。



动手开始吧

- ❑ 启动网络浏览器并输入网址：
<http://scratch.mit.edu>。
- ❑ 在首页中，点选右上方的「加入Scratch」或是蓝色圈圈中的「加入Scratch乐园」。
- ❑ 三个步骤来注册你自己的Scratch账号。



Join Scratch

It's easy (and free!) to sign up for a Scratch account.

Choose a Scratch Username

Don't use your real name

Choose a Password

Confirm Password



1

2

3

Next

设计手札



目标

完成这项活动，学生将：

- + 制作一份个人化的设计手札，开始记录他们的设计过程与反思心得

活动描述

- ❑ 向学生们解说设计手札的概念，它可以是实体或数字的笔记本，类似个人化的日记。学生可以在上面发想点子或分享个人心得。向学生说明在Scratch的冒险旅程中，会要求他们更新设计手札，也鼓励他们在设计程序计划的过程中随时记录点子、灵感、笔记、涂鸦、疑问、挫折、胜利与其他事情到设计手札。
- ❑ 浏览一遍设计手札的模板来思考哪种形式的设计手札(实体或是数字)最适合你的学生。留下时间给学生们开始制作并个性化他们的设计手札。
- ❑ 藉由回答右边的反思问题，要求学生创作第一则设计手札心得。
- ❑ 鼓励学生与他人分享设计手札，并与一位邻座同学进行讨论。

教材资源

- ❑ 设计手札模板
<http://bit.ly/designjournal-paper>
<http://bit.ly/designjournal-digital>
<http://bit.ly/designjournal-blog>
- ❑ 纸张与手工艺材料 (用于制作纸本手札)

反思问题

- + 你会如何介绍Scratch给其他朋友呢？
- + 写下或画出你想要用Scratch创作的三个不同新点子。

审视学生成果

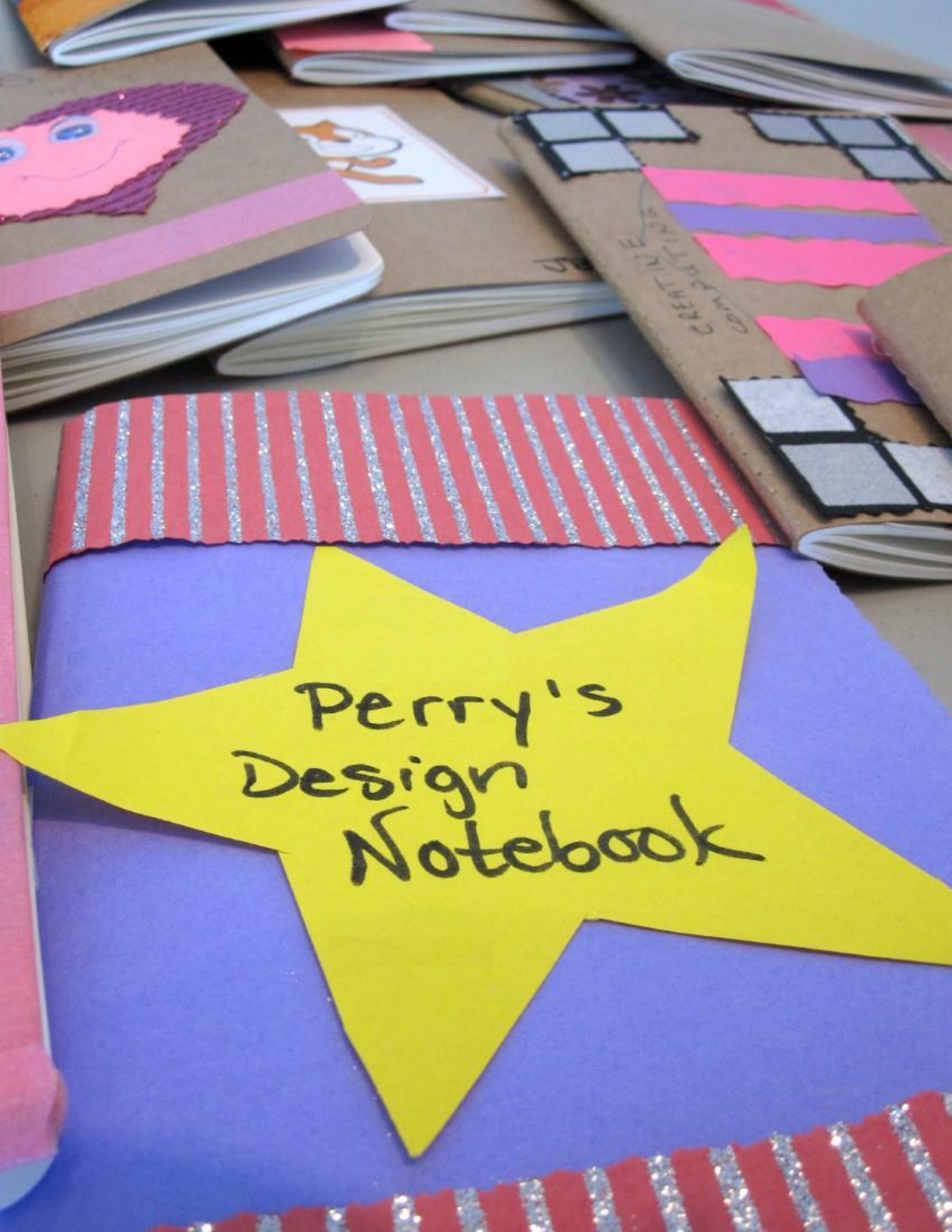
- + 反思问题的答案告诉你，哪些类型的程序计划是学生们有兴趣追寻制作的？
- + 根据学生的回答，这份教学指引中的哪些个单元可能会吸引不同的学生？

附注

- + 在其他的引导活动中，促进小组对于反思问题进行讨论。
- + 决定设计手札要保密或是公开。例如：透过保密的手札，你可以保有一对一与学生的互动，而共享的设计手札则让学生留下给彼此的建议。思考一下每个选择的优缺点。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



Perry's
Design
Notebook

SCRATCH

小惊奇



目标

完成这项活动，学生将：

- + 借着亲自动手，参与探索Scratch的新体验

活动描述

❑ 帮助学生开启Scratch程序计划编辑器，先前往Scratch网站：<http://scratch.mit.edu>，登入个别账号，再点选页面上方的「新建」。老师也可以选择性地提供「Scratch小惊奇讲义」与「Scratch教学图卡」来引导学生。

❑ 留给学生10分钟的时间来自由探索Scratch的操作界面。以几个问题来提示学生，例如：「你们有10分钟的时间让Scratch猫咪做出任何惊奇的动作」或「不用担心弄坏，用10分钟的时间探索Scratch接口，你注意到了什么？」，鼓励学生们一起合作，请求别人的帮助并且分享他们发现、弄清楚了什么。

❑ 寻求3或4名自愿者来对整个班级分享他们的发现。在心得分享后，老师也可以选择性地给予学生一些挑战：

- 有人知道如何增加音效吗？
- 有人知道如何改变背景吗？
- 有人知道如何找到程序积木的说明吗？

教材资源

- ❑ Scratch小惊奇讲义
- ❑ Scratch教学图卡
<http://scratch.mit.edu/help/cards> (英文版)
<https://goo.gl/XRyh3D> (繁体中文版)

反思问题

- + 你解决、弄明白了什么呢？
- + 什么是你想了解更多、更深入的呢？

审视学生成果

- + 学生们明了如何建立一个新的程序计划吗？
- + 学生们了解将Scratch程序积木黏在一起的技巧吗？

附注

- + 这个活动的主要目标在于建立「不惧怕」、「勇于尝试」、「彼此互助」的氛围。这个活动设想的是：学生们(与老师们)事前都不明白任何东西 - 所有人在当下的环境中一起学习。
- + 执行Scratch，需确认你的计算机有最新版的Flash：<http://helpx.adobe.com/flash-player.html>

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

SCRATCH 小惊奇

你可以让Scratch猫咪做出什么惊奇的动作呢？

这个活动中，你会用Scratch创作一个新的程序计划。此外，你也可以体会不同的Scratch积木会让猫咪做出神奇的多样动作。那么，你会创作些什么呢？



动手开始吧

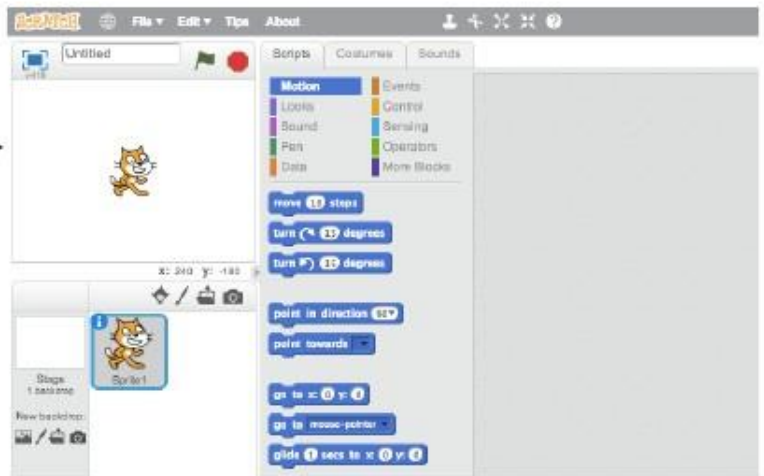
□ 前往Scratch 网站: <http://scratch.mit.edu>

□ 以你的账号登入。

□ 点选在画面左上方的「新建」来产生一个新的程序计划。

□ 开始探索！尝试点击Scratch的不同部分，看看画面有什么变化。

□ 组合不同的Scratch积木！ 拖拉放Scratch积木到「程序」区域。体验看看点击每一个积木时的反应，也可以尝试让积木黏在一起。



SCRATCH

H

工作室

建议时间
5-15 分钟

目标
完成这项活动，学生将：

- + 能够将程序计划加到工作室之中
- + 能够对其他Scratch程序计划进行留言

活动描述

❑ Scratch工作室是搜集与整理Scratch在线程序计划的一种方式。在这个活动当中，帮助学生理解什么是「工作室」并学会如何将一个程序计划加到工作室当中。老师也可以选择性地提供「Scratch工作室讲义」给学生。

❑ 首先，让学生前往Scratch网站并以个别的账号登入。接着，引导学生找到「Scratch惊奇工作室」或是你创立的「班级工作室」。然后，让学生藉由将他们的Scratch小惊奇作品加入工作室来与他人分享。

❑ 鼓励学生调查在工作室之中的其他程序计划。让学生对两个最吸引他们兴趣的两个程序计划进行留言。让班级群体参与讨论，要如何留下适当且有建设性的回馈。

❑ 藉由在设计手札或小组讨论中回答反思问题，要求学生回想一下他们自己的创意作品。

教材资源

- ❑ Scratch工作室讲义
- ❑ Scratch惊奇工作室
<http://scratch.mit.edu/studios/460431>

反思问题

- + Scratch 工作室的用途是什么？
- + 看着别人的程序计划，你发现了什么有趣或新奇的事情了？
- + 你分享了哪两则留言？
- + 什么是「好的」回馈呢？

审视学生成果

- + 学生是否成功地将他们的程序计划加到工作室？
- + 学生是否适当地对其他作品给予留言？

附注

- + 建立你自己的工作室来搜集学生的程序作品。建立一个班级专属的Scratch小惊奇工作室，把链接地址给学生，让他们把作品传进来。建立一个专门的工作室来搜集所有班级的程式作品或以个别的工作室来追踪学生们的进度。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

SCRATCH 工作室

室

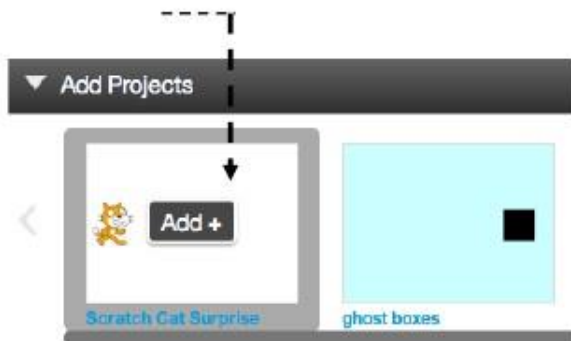
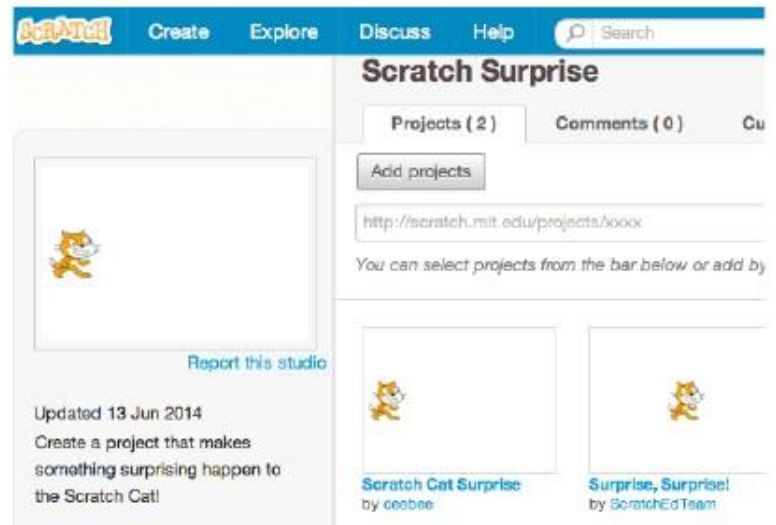
学会如何将你的程序计划加入到一个Scratch在线工作室

工作室可以搜集许多Scratch程序计划。跟着以下的步骤将你的Scratch程序计划加入Scratch网站中的惊奇工作室。



动手开始吧

- ❑ 前往Scratch惊奇工作室：
<http://scratch.mit.edu/studios/460431>
- ❑ 以您的账号登入。
- ❑ 点选「新增计划」，画面下方会出现「我分享出去的计划」、「我的最爱」、「我最近看过的内容」。
- ❑ 用指针找到你的「Scratch小惊奇」程序计划，接着再点选「Add+」把计划加到惊奇工作室。



小组讨论



目标

完成这项活动，学生将：

- + 分发到讨论小组，对于设计构想与进行中的程序计划，给予意见也获得回馈。

活动描述

- ❑ 将讨论小组的概念传达给学生。讨论小组是一个由设计师组成的小型团体，设计师彼此分享点子并测试正在进行的项目，来给予建议并获得回馈，帮助彼此有更进一步的发展。
- ❑ 老师可以选择性地提供「小组讨论讲义」来引导学生进行心得回馈。
- ❑ 将学生3至4人分成一组。在讨论小组中，要求学生轮流分享他们的点子、程序计划草稿或是程序原型，例如：Scratch小惊奇计划。
- ❑ 让学生运用「小组讨论讲义」，藉由小组成员回答「红、黄、绿 (red, yellow green)」的问题来搜集心得回馈。鼓励学生们在设计手札中记录他们的笔记、心得回馈与建议。

教材资源

- ❑ 小组讨论讲义

反思问题

- + [红] 哪个部分无法执行或可以改进？
- + [黄] 哪个部分不懂或可以换个做法？
- + [绿] 哪个部分做得很好或你最喜欢？

审视学生成果

- + 所有学生是否都有机会分享他们的程序作品并且获得回馈？

附注

- + 在一次又一次的程序设计过程中，有一个专属的小组成员给予鼓励与回馈是非常有价值的。在单元1到6中，留给学生们与讨论小组持续互动的机会。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

小组讨论

反馈给： _____

计划名称： _____

反馈者	[黄][绿][红] 哪个部分无法执行或可以改进？ 哪个部分不懂或可以换个做法？ 哪个部分做得很好或你最喜欢？		

可以帮助思考程序计划的线索：

- + 明白清楚：你了解程序计划要做什么吗？
- + 设计特色：这个程序计划有哪些特色？程序执行的与想象中一样吗？
- + 操作呈现：这个程序计划的吸引力如何？它是具有互动性、原创性、精巧的、好笑的或是有趣的吗？当你使用时感觉如何？

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 1

探索发现



YOU ARE
HERE 你的进



WHAT'S
INCLUDED 本单

依照程序来跳舞	26
循序渐进	28
10个程序积木	
30	
我的工作室	32
除错 (DEBUG) !	34
关于我	36

单元 1

整体概观

「大处」着眼

当面临要开始进行创意程设思维的课时，许多与我们多年来一起合作的教育者都努力思考着两个问题：「什么是帮助学习者开始起步的最好方式？」以及「作为老师的我需要知道些什么？」Seymour Papert（一位知名数学家、教育家，并藉由乐高(Lego)程序语言对于Scratch开发具重大影响）的一篇文章可以作为思考这些问题的灵感来源。

针对第一个问题，人们倾向于采取两种极端的立场。一种立场是认为学习者需要被告知该学什么并以高度结构化的方式来学习、体验。另一种则是完全放任学习者依照自己的方向来探索学习。Papert提倡：「学习者应该成为他们自身思考与学习的主导者与探索者」。Papert鼓励老师应该追求教与学之间的平衡。在这个教学手册中，我们以改变着活动中结构化学习的份量来提供教学平衡。

关于第二个问题，教学者有时候会担心他们对于Scratch认识得不够而无法帮助他人。我们建议你采取一种更宽广的角度来看待什么是「认识」Scratch。你并不需要了解Scratch所有的接口，也不需要知道如何解决学习者遇到的每一个问题。但是，就如同Papert所说的，教育者可以成为具有认知力的向导，以有技巧的发问，进一步协助将大问题拆解为能够进一步处理的小片段。

附注

- + 确认学生们都已经拥有了一个Scratch账号可用来在线储存与分享他们的程序计划。
- + 想一想你计划如何来取得学生的作品。例如：你可以创建一个班级专属的工作室来收集程序计划、让学生把程序计划的链接email给你、或是建立一个班级的部落格。

當他們在一起解決謎團時，小朋友得到了一個啟示，他說：「你是說，你真的不知道如何修正它嗎？」雖然小朋友還不知道怎麼表達，但表現在他眼前的情形是：他和老師一起融入了一個研究計劃之中。這是一個慘痛的事實。就如同人們常常提到的，孩子融入了老師設定的「讓我們一起完成」的遊戲之中，事實上大家都知道一起合作只是幻想。發現不是設定好的；發明無法事前安排。

(Papert, 1980, 115頁)

学习目标

学生们将：

- + 藉由创建一个交互式的Scratch程序计划，来探索、体验Scratch学习环境。
- + 认识更广泛的Scratch程序积木。
- + 变得更熟悉次序(sequence)的概念。
- + 在建构程序计划中，动手实际演练：试验(experimenting)与重复(iterating)的概念。

关键词、概念与实作

- | | | |
|---------------------|--------------------|-------------------------------|
| + 试验(experimenting) | + 动作 | + 交互式剪贴簿(interactive collage) |
| + 与重复(iterating) | + 外观 | + 成对分享 |
| + 测试与除错(debugging) | + 音效 | |
| + 次序(sequence) | + 造型 | |
| + 角色 | + 背景 | |
| | + 技巧窗口(tip window) | |
| | + 混合搭配(remix) | |

选择你自己的冒险旅程

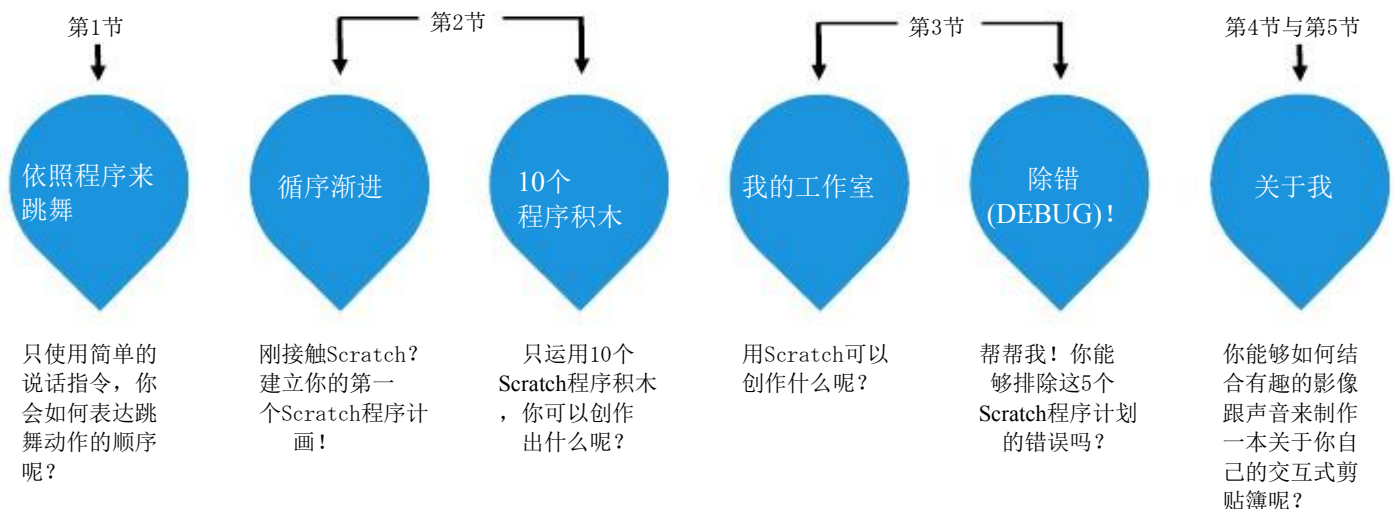
本单元涵盖了一系列结合了结构化与开放式的活动。透过这些活动，让学生融入在探索与体会重要的次序(sequence)概念。次序概念是指：判别并安排一组有先后顺序的指令。对学生而言，这是一个具有强大威力的时刻：借着把他们的想法转换成计算机程序码(Scratch程序积木)，他们可以告诉计算机该怎么做。

从「循序渐进」、「组合有限数量的程序积木」再到「排除错误的挑战」，每一个活动都在帮助学习者准备着建立「关于我-程序计划」必须具备的技巧。在最后的程序计划中，学习者将在Scratch的环境里，尝试及试验各种角色、造型、外观、舞台背景与音效来创作专属个人化的交互式剪贴簿。

由你决定要运用全部或挑选其中一些最适合学生需要和感兴趣的活动。如果不确定如何做，可以参考以下建议的学习途径。



可行的学习途径



依照程序 来跳舞



目标

完成这项活动，学生将：

- + 学习使用有顺序的简单指令来表达复杂的活动

活动描述

- ❑ 征求8名自愿者 — 其中4个人不排斥发号施令，另外4个人不排斥接受指令。安排「发号施令」/「接受指令」成一对，共分四组。老师也可以选择性地准备投影机来播放「依照程序来跳舞」影片。
- ❑ 每一对「发号施令」/「接受指令」：
 1. 「接受指令的人」不要看画面，而「发号施令的人」和剩下的所有人看着画面。
 2. 播放影片给「发号施令的人」和剩下的所有人看，但是不要让「接受指令的人」看见
 3. 要求「发号施令的人」只能用讲话的方式描述给「接受指令的人」听，要如何做出影片中跳舞动作的顺序。
- ❑ 运用这个活动，开始让学生们讨论安排一系列指令顺序的重要性。你可以让学生们个别在自己的设计手札中记录想法，或是安排小组讨论让「发号施令的人」、「接受指令的人」还有观察者来分享他们的想法。

附注

- + 这是好几个不需要计算机的活动其中之一。退离开计算机可以提供对于程序设计概念、实作及观点的崭新角度与理解。
- + 让学生们对于其中一个舞蹈片段写下来一个步骤再一个步骤的指令。在程序设计里，这些指令就称为「虚拟程序代码(pseudocode)」。

教材资源

- ❑ 投影机 (选择性)
- ❑ 依照程序来跳舞 (Programmed to Dance)影片
 - <http://vimeo.com/28612347>
 - <http://vimeo.com/28612585>
 - <http://vimeo.com/28612800>
 - <http://vimeo.com/28612970>

反思问题

- + 当发号施令的人有什么简单/困难的地方呢？
- + 当接收命令的人有什么简单/困难的地方呢？
- + 旁观的人有什么简单/困难的地方呢？
- + 这一个活动与我们使用Scratch有什么关联呢？

审视学生成果

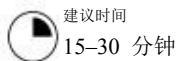
- + 学生们是否能够解释安排指令前后顺序的重要性？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



循序渐进



目标

完成这项活动，学生将：

- + 跟着一步步循序渐进的指引，在scratch创造一只跳舞的猫咪
- + 藉由试验(experimenting)及重复(iterating)的练习，体会编写程序的经验

活动描述

- ❑ 帮助学生他们的Scratch登入网站并且点击网站上方的「新建」来开启Scratch程序计划编辑器。老师也可以选择性地准备「循序渐进讲义」与「Scratch教学图卡」在活动中来引导学生。
- ❑ 让学生开启「技巧」窗口并且按照「Getting Started with Scratch」循序渐进的教学来创作跳舞猫咪的程序计划。鼓励学生们加入其他的程式积木并且实验使用「动作」、「角色」、「造型」、「外观」、「音效」或「背景」来改造他们自己的程序计划。
- ❑ 让学生们与其他人分享他们的第一个Scratch创作！老师也可以选择性地帮助学生把他们的程序计划加入「循序渐进工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 循序渐进讲义
- ❑ 循序渐进工作室 (Step-by-Step studio)
<http://scratch.mit.edu/studios/475476>
- ❑ Scratch教学图卡
<http://scratch.mit.edu/help/cards> (英文版)
<https://goo.gl/XRyh3D> (繁体中文版)

反思问题

- + 这个活动有什么令你感到惊奇的地方呢？
- + 在这个活动的过程中，被带领着一步步循序渐进有什么感觉？
- + 什么时候你觉得最有创造力？

审视学生作品

- + 学生们能够开启Scratch技巧窗口吗？
- + 学生们能够创造跳舞的猫咪吗？
- + 学生们能够储存并分享程序计划吗？

附注

- + 如果学生们还没有Scratch账号，参考单元0的活动来帮助他们创立自己的账号，这样一来学生们便可以储存并分享他们的第一个Scratch程序计划给朋友与家人。
- + 使用单元0「Scratch工作室」活动或讲义来提醒学生如何将程序计划加入工作室。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

循序渐进

刚接触Scratch? 建立你的第一个Scratch程序计划!

在这个活动中, 你会跟着技巧视窗内中循序渐进的介绍, 在Scratch中创造一只跳舞的猫咪。当你完成了这些步骤, 试验着加入其他的Scratch程序积木, 让程序计划融入你自己的想法。



动手开始吧

- ❑ 跟着技巧窗口中一步步的介绍。
- ❑ 加入更多程序积木。
- ❑ 试验加入你自己的想法!



你想试验看看哪些程序积木呢?



尝试看看

- ❑ 尝试录下你自己的音效。
- ❑ 创造不同的背景。
- ❑ 增加更多跳舞的角色, 让你的程式计划变成一场舞会。
- ❑ 尝试为你的角色设计新的造型。

都完成了?

- + 把你的程序计划加入循序渐进工作室:
<http://scratch.mit.edu/studios/475476>
- + 挑战自己做更多! 加入新的程序积木、音效、动作一起玩。
- + 帮助一位邻座的同学!
- + 挑选一些新的程序积木来试验看看!

10个 程序积木

建议时间
15-30 分钟

目标

完成这项活动，学生将：

- + 创作一个限制只使用10个程序积木的程序计划。

活动描述

- ❑ 帮助学生使用他们的账号登入Scratch网站，并点击网站上方的「新建」来创立一个新的Scratch程序计划。老师可以选择性地准备「10个程序积木讲义」在活动中来引导学生。
- ❑ 留给学生时间来使用以下10个积木来创作程式计划：移到(go to)、滑行(glide)、说(say)、显示(show)、隐藏(hide)、set size to、播放音效直到播放完毕(play sound until done)、当角色被点击(when this sprite clicked)、等待(wait)及重复(repeat)。提醒学生们在程序计划中，每个程序积木都至少要使用一次。鼓励学生试验不同的角色、造型或舞台背景。
- ❑ 邀请学生在他们的讨论小组中分享他们的程式计划（参考单元0小组讨论活动）。老师也可以选择性地将他们的程序计划加入「10个程序积木工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

附注

- + 只用10个程序积木可以完成的事情多到令人惊讶！老师可以藉由邀请数字学生在课堂上分享他们的程序计划或是浏览「10个程序积木工作室」中的程序计划，好好把握这个机会来鼓励不同的点子并为创意喝采。

教学资源

- ❑ 10个程序积木讲义
- ❑ 10个程序积木工作室(10 Blocks studio)
<http://scratch.mit.edu/studios/475480>

反思问题

- + 只能使用10个程序积木，有什么是困难的呢？
- + 只能使用10个程序积木，有什么是简单的呢？
- + 这个有限制的程序活动，是如何使你对于事情有了不同的想法？

审视学生作品

- + 程序计划包含了所有10程序积木吗？
- + 不同学生对于有条件限制的创作的反应是如何呢？什么事情可能会告诉你该位学生学得如何呢？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

10个程序积木

只运用10个Scratch程序积木，你可以创作出什么呢？

运用这10个程序积木来新建一个程序计划。学生可以运用这些积木一次、两次或更多次，但每个至少要使用一次。

动手开始吧

- 实验每一个程序积木来测试点子。
- 以各种方式来混合并搭配程序积木。
- 再来一次！

感覺卡住了？

沒關係！試試這些做法...

- 尝试不一样的积木组合来测试点子。混合与搭配程序积木直到你发现有趣的组合！
- 与邻座同学一起进行创意发想！
- 探索其他的程序项目，看看其他人都用Scratch做什么。这是一个发掘灵感的好方法！

移到

go to x: 0 y: 0

滑行

glide 1 secs to x: 0 y: 0

说

say Hello! for 2 secs

显示

show

隐藏

hide

set size to 100 %

播放音效
直到播放完毕

play sound meow until done

等待

wait 1 secs

当角色被点击

when this sprite clicked

重复

repeat 10



都完成了？

- + 将你的程序计划加入「10 Blocks 工作室」：
<http://scratch.mit.edu/studios/475480>
- + 加入不同的角色、造型和背景一起玩！
- + 挑战自己做更多！看看你可以用这10个程序积木可以创作多少个不同的程序计划。
- + 与一个同伴交换程序计划并且混合彼此的程序创作。

我的工作室



目标

完成这项活动，学生将：

- + 借着探索Scratch网站上众多程序计划中的一部份，进而发现创意的广大可能性
- + 在Scratch工作室中，展示他所搜集的3个或更多的程序计划

活动描述

- ❑ 老师可以选择性地示范如何建立一个新的工作室或是提供「我的工作室讲义」来引导学生。
- ❑ 老师可以选择性地使用右边提供的连结来展示案例工作室。留给学生们10分钟来浏览在Scratch首页上的程序计划并且使用「搜寻」页面来找寻有趣的程序。
- ❑ 要求学生找出三个或更多可以用来加强或激发他们创作的Scratch程序计划。帮助学生从「我的东西」页面来建立一个新的工作室并且把能引发灵感的程序计划加入其中。
- ❑ 邀请学生们分享找寻具有启发性的程序计划的方法。我们建议两人一组，让学生们彼此分享工作室并讨论搜寻的方法。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 我的工作室讲义
- ❑ 案例工作室
 - <http://scratch.mit.edu/studios/211580>
 - <http://scratch.mit.edu/studios/138296>
 - <http://scratch.mit.edu/studios/138297>
 - <http://scratch.mit.edu/studios/138298>

反思问题

- + 你用什么样的搜寻方法来找寻有趣的程序计划？
- + 每个范例程序计划可能会如何帮助未来的程序创作呢？
- + 表扬灵感来源的功劳是非重要的！你可以如何表扬灵感来源的那些程序计划呢？

审视学生作品

- + 工作室中是否有三个或更多的程序计划？
- + 程序计划反应了学生的设计兴趣是什么？

附注

- + 如果学生们没有自己的Scratch账号，建立一个班级性的工作室，让学生可以展示。
- + 使用者可以创立各式各样的工作室 — 学生们可以建立工作室来搜集有关于他们创作主题相似的Scratch程序计划、或是含有未来可能会用的技巧或素材的Scratch程序计划。

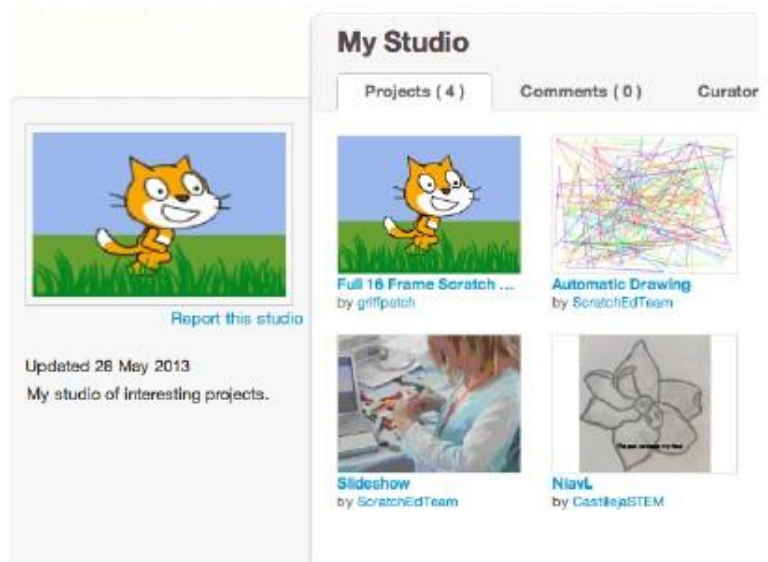
我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

我的工作室

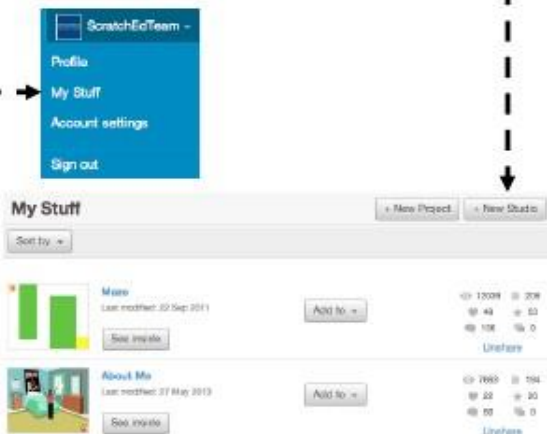
用Scratch可以创作什么呢？

在这活动中，借着探索Scratch网站上众多程序计划的一部份，你将可以调查并研究创意的广大可能性。此外，你也将学会在Scratch工作室中开始搜集我最爱的程序计画。



动手开始吧

- ❑ 查看Scratch首页上的程序计划或点选「浏览」来找寻特定种类的程序计划。
- ❑ 从「我的东西」页面，建立一个新的工作室。
- ❑ 将三个（或更多）的程序计划加入到你的工作室。



尝试看看

- ❑ 使用搜寻列来找寻让你兴趣相关的程序计划。
- ❑ 查看在「浏览」页面中的每个类别，像是：动画、美术作品、游戏、音乐以及故事，或查看：案例工作室。
- ❑ 查看首页上的特色工作室来找寻灵感。

都完成了？

- + 挑战自己做更多！探索浏览越多Scratch程序计划，你将了解越多可以用Scratch完成的事情。
- + 找寻其他Scratch使用者建立的、让你觉得有趣的工作室。
- + 问问旁边的同学，他们是用什么方法找到有趣的程序计划。
- + 与旁边的同学分享你刚刚建立的工作室！

除错 (DEBUG) !



目标

完成这项活动，学生将：

- + 深入调查五个除错挑战的问题并且提出答案
- + 藉由实际的测试与除错练习，探索一系列的概念，包含次序概念(sequence)
- + 建立程序除错方法的列表

活动描述

- ❑ 教师可以选择性地提供除错 (Debug) 讲义在活动中来引导学生。
- ❑ 协助学生从除错工作室中开启除错挑战，或前往讲义中所列出的网址连结。鼓励学生点击「看看里面」来调查这个有问题的程序计画、仔细斟酌有问题的程序代码、并且测试可能的解决方法。
- ❑ 留给学生们时间来测试并排除每个除错挑战的问题。老师也可以选择性地让学生使用 Scratch的融合(remix)功能来除错并保存正确的程序计划。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们测试与除错的体验与感受。
- ❑ 藉由搜集学生的「问题找寻 (problem finding)」与「问题解决 (problem solving)」的方法，产生一份属于班级的除错方法清单。

教学资源

- ❑ 除错 (Debug) 讲义
- ❑ 除错工作室 (Debug It! studio)
<http://scratch.mit.edu/studios/475483>

反思问题

- + 问题是什么？
- + 你是如何辨别、找出这个问题的呢？
- + 你如何修正这个问题？
- + 别人有修正这个问题的其他方法吗？

审视学生作品

- + 学生们能够解决这五个程序错误吗？如果不，你可能会如何说明在这些未解问题中表达的概念呢？
- + 学生们采用了什么不一样的测试与除错方法呢？

附注

- + 这个活动以小组方式进行更好！让学生们2至4人一组共同解决问题并分享除错方法。
- + 测试与除错可能是程序设计师最常进行的活动。很少程序能如同预期中的执行，因此建立一套测试与除错方法对于任何一个程序设计师都是很有帮助的。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

除错 (DEBUG)!

帮帮我！你能够排除这5个Scratch程序计划的错误吗？

在这个活动中，你将研究调查这五个除错挑战中有什么东西出错了，并且为每个挑战找到解决方法。

动手开始吧

- ❑ 前往「单元1的除错工作室」：
<http://scratch.mit.edu/studios/475483>
- ❑ 对于工作室中的五个除错挑战进行测试并排除错误。
- ❑ 写下你的解决办法或用你的方法调整有错的程序。

感覺卡住了？

沒關係！試試這些做法...

- ❑ 把程序中可能的错误列出一张清单。
- ❑ 追踪并弄清楚你做过的事情！这样可以提醒你已经做过的尝试并且引导你进行下一个尝试。
- ❑ 与你旁边的同学分享并且比较「发现问题」以及「解决问题」的方法，直到发现对你有帮助的方法！

❑ 除错挑战! 1.1 <http://scratch.mit.edu/projects/10437040>
当点击绿色旗子时，「Gobo」跟「Scratch猫咪」应该一起开始跳舞。但是现在只有「Scratch猫咪」开始跳舞！我们要如何修正这个程序呢？

❑ 除错挑战! 1.2 <http://scratch.mit.edu/projects/10437249>
在这个程序计划中，当绿色旗子按下时，Scratch猫咪应该开始出现在舞台左边，在左边说一些话，滑行到舞台右边，在右边说一些话。第一次点下绿色旗子可以正常执行，但没有办法再执行一次。我们要如何修正这个程序呢？

❑ 除错挑战! 1.3 <http://scratch.mit.edu/projects/10437366>
当空格键按下时，Scratch猫咪应该会蹦跳一次。但是现在按下空格键时，没有反应！我们要如何修正这个程序呢？

❑ 除错挑战! 1.4 <http://scratch.mit.edu/projects/10437439>
在这个程序计划中，当Scratch猫咪被点击时，它应该要在舞台上前后来回走动。但现在，Scratch猫咪却失去了控制 — 它是上下走动的。我们要如何修正这个程序呢？

❑ 除错挑战! 1.5 <http://scratch.mit.edu/projects/10437476>
在这个程序计划中，当点击绿色旗子时，Scratch猫咪应该要在对话泡泡里说「Meow, meow, meow!」并且同时发出声音。但是现在，对话泡泡比声音提早出现，并且Scratch猫咪只发出一次「Meow」的声音。我们要如何修正这个程序呢？

都完成了？

- + 与一位伙伴讨论你们测试与排除错误的实际过程。将你们方法之间的相同与不同的地方，进行纪录。
- + 用右键对你的程序区域中的积木进行点击，可以增加程序批注。这样做可以帮助别人理解你程序中不同的部分！
- + 帮助一位旁边的同学！

关于我

建议时间
45-60 分钟

目标

完成这项活动，学生将：

- + 对于更多的Scratch程序积木感到熟悉
- + 能够创作一个没有标准答案的Scratch程序计划，而这个计划则交互式的呈现了他们个人的喜好与兴趣

活动描述

- ❑ 向学生介绍交互式剪贴簿(interactive collage)的概念。交互式剪贴簿是一个藉由可点击的角色，展示学生本身各种层面的Scratch程序计划。老师可以选择性地从「关于我工作室中」展示交互式程序计划的范例。
- ❑ 让学生以他们的Scratch账号登入网站并新建一个程序计划。老师可以选择性地提供「关于我讲义」与「Scratch教学图卡」来引导学生。留给学生时间来创作关于我的交互式剪贴簿，鼓励他们借着试验(experimenting)与重复(iterating)来建构他们的程序。
- ❑ 允许学生可以将进行中的程序计划分享给别人。我们建议两人一组，让学生能彼此讨论与分享他们的程序计划。老师可以选择性地邀请学生将他们的程序计划加入到「关于我工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 关于我讲义
- ❑ 关于我工作室 (About Me studio)
<http://scratch.mit.edu/studios/475470>
- ❑ Scratch教学图卡
<http://scratch.mit.edu/help/cards> (英文版)
<https://goo.gl/XRyh3D> (繁体中文版)

反思提问

- + 你最引以为傲的事情是什么？为什么？
- + 你在什么事情上困住了呢？你是如何解决、脱困的呢？
- + 接下来，你可能想做些什么？
- + 从查看别人的「关于我」程序计划，你发现了什么？

审视学生作品

- + 程序计划是否创意地运用了角色、造型、外观、背景或音效？
- + 程序计划是交互式的吗？使用者是否能与程序计划中的多种元素互动呢？

附注

- + 范例程序会同时引发灵感与模仿、启发创意也会限制创意。鼓励广泛主题的创作，越多样越好！
- + 学生们可以进一步藉由使用相机或网络摄影机来把影像加入到程序计划中，让程序更具个人化的特色。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

关于我

你能够如何结合有趣的影像跟声音来制作一本关于你自己的交互式剪贴簿呢？

尝试用角色、造型、背景、外观与音效，来创作一个交互式的Scratch程序计划。这个程序计划能够帮助别人更了解「你」还有你关心的想法、活动以及人们。



动手开始吧

- ❑ 创建一个角色。
- ❑ 让他可以互动。
- ❑ 再做一次！



增加程序代码，让你的角色可以用鼠标点击、按下键盘或更多方式，来进行互动！



尝试看看

- ❑ 使用造型来改变角色的外观。
- ❑ 创造不同的背景。
- ❑ 尝试增加音效到你的程序计划。
- ❑ 尝试在你的剪贴簿加入动作变化。

TRICKS 可以运用的程序积木

都完成了？



- + 将你的程序计划加入到「关于我工作室」：
<http://scratch.mit.edu/studios/475470>
- + 挑战自己做更多！尝试新增新的程序积木、音效或是动作！
- + 帮助一位邻座的同学！

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 2

动画

Turn up the music!



你的进度

本单元包含



演出程序	42
组乐团	44
橘色正方形、紫色圆形	46
动起来了!	48
除错 (DEBUG)!	50
音乐影片	52

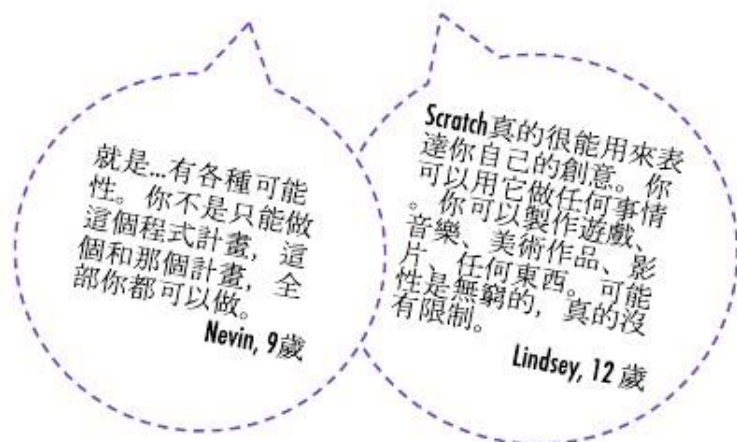
单元 2

整体概观

「大处」着手

孩童们已经在Scratch在线社群中分享超过了6百万个程序项目(动画、故事、游戏与其他)。我们想要这份教学手册达成的目标之一是能够反映创作的无尽多样性。在活动中,我们提供了加入自己想法的机会并且避免设定只有一种「正确」答案的挑战。藉由活动,我们让学习者融入在各种类型的练习之中。在这个单元,我们深入研究动画、美术及音乐,开始来探索创意的多元性。

学习者常常会强调使用Scratch能够发挥创意的多元性。当学习者被问到「如果你要向一位你的朋友解释Scratch是什么,你会如何描述呢?」,以下是一些回答:



学习目标

学生们将:

- + 开始认识程序设计思维的其他概念: 重复(loops)、事件(events)及平行(parallelism)
- + 更为熟悉次序(sequence)的概念
- + 试验在事件、控制、音效、外观类别中的尚未接触的程序积木。
- + 探索许多以艺术为主题的Scratch程序计划
- + 创作一个动画式的音乐影片程序计划

关键词、概念与实作

- + 循环(loops) + 程序(scripts) + 动画(animation)
- + 事件(events) + 放映模式 + 成果展(gallery
- + 平行 (presentation walk)
- + 控制(control) + 点阵(bitmap)
- + 广播(broadcast) + 向量(vector)

附注

- + 本单元许多活动涉及了音效和音乐。我们建议可以为学生们准备耳机。

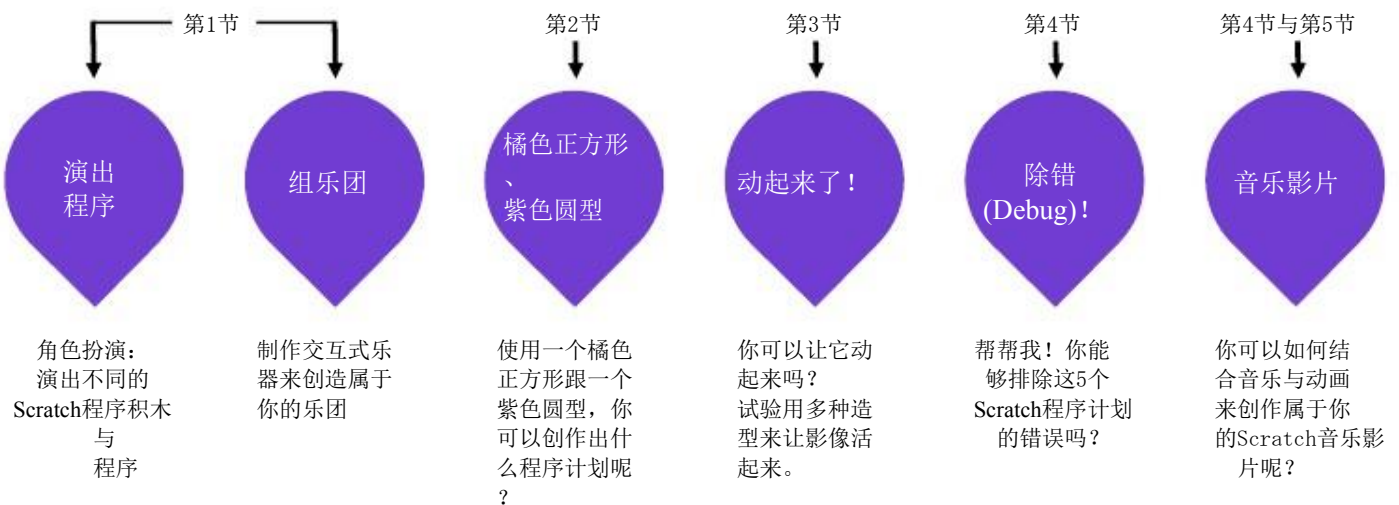
选择你自己的冒险旅程



用Scratch写程序就像导演一出戏。在一出戏剧中，就如同在Scratch一样，有人物（在Scratch中称为角色）、服装、背景、脚本和一个舞台。用Scratch写程序是运用名为「事件(event)」的线索来发出信号，标记什么事情该在什么时候发生：例如：启动程序(当绿色旗子被点击时when green flag clicked)、触发角色动作(当角色被点击时when this sprite clicked)、或是送出暗示给所有的角色或背景(广播broadcast)。

由戏剧意象(theatre metaphor)获得的灵感，本单元设计一系列以艺术为主题的活动来帮助学生探索程序设计概念：循环(loops)、事件(event)及平行(parallelism)，最后综合这些概念来完成个人化音乐故事的程序计划。

可行的学习途径



演出 程序

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 藉由表演，对于事件(event)与平行(parallelism)的概念有初步认识。事件：一件事情触发另一件事情；平行：多个事情在同一时间发生
- + 能够解释什么是事件，并了解事件在Scratch中是如何运作
- + 能够解释什么是平行，并了解平行在Scratch中是如何运作

活动描述

- ❑ 老师可以选择性地准备一台投影机链接计算机，并开启Scratch来显示要「表演」的程序与程序积木。
- ❑ 征求两名志愿者。
- ❑ 提示两名志愿者演出一系列的指令，可运用Scratch接口或印出来的Scratch程序积木来协助这两名志愿者
 - 让一名志愿者做一件事，例如：在房间里走动。
 - 重新设置(reset)该名志愿者 — 回到初始状态。
 - 让该名志愿者同时做两件事情，例如：在房间里走动并说话。
 - 加入第二名志愿者。让他同时做一件事(独立于第一名的动作)，例如：说话。
 - 让第二名志愿者做一件与第一名志愿者有关联性的事，例如：让第二名志愿者被动地回答回答第一名志愿者的话，而不是互相发问。
- ❑ 组成讨论小组回想相关经验并运用右边的反思问题来讨论事件与平行的概念。

教学资源

- ❑ 投影机 (非必备)
- ❑ 实体Scratch程序积木 (非必备)

反思问题

- + 有哪些不同方式，让动作被触发呢？
- + 对于Scratch中的事件(event)有哪些机制呢？
- + 有哪些不同的方式，让事情同时发生呢？
- + 有哪些机制让Scratch中的平行(parallelism)能够运作呢？

审视学生成果

- + 学生们是否能够解释「事件」与「平行」的概念是什么以及在Scratch中是如何运作的吗？

附注

- + 这个活动强调了「重置(reset)」的概念。这是Scratch初学者时常感到困扰的概念。如果学生们想让程序的角色以特定的外观、从特定的位置开始，学生们必须理解到这些设定的步骤都是需要自行处理的。
- + 这个活动可以有效地展现广播(broadcast)与接收指令这样成对的概念。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

SCRATCH'S
CENTRAL
THEATRICAL
METAPHOR

SPRITE



~~STAGE~~



组乐团

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 创作一个结合了交互式角色与有趣音效的程序计划
- + 更加融会贯通以下概念：次序(sequence)、循环(loops)、事件(events)及平行(parallelism)
- + 在建构程序计划中，实际练习试验(experimenting)及重复(iterating)

活动描述

- ❑ 老师可以选择性地，从「组乐团工作室」中展示范例程序计划并提供「组乐团讲义」来引导学生。
- ❑ 留时间给学生们藉由搭配角色与音效来创作交互式乐器。鼓励学生藉由探索在「音效」类别中的其他程序积木，或使用音效编辑器（在音效分页中），试验以不同的方式来表现音效。
- ❑ 让学生们互相展示他们的乐团或让他们四处走动来与乐器互动。我们建议进行「成果展(gallery walk)」：让学生们把自己的程序计划切换到「放映模式(presentation mode)」然后让学生们四处走动来探索别人的程序计划。老师也可以选择性地让学生们把他们的程序作品加入到「组乐团工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

附注

- + 分享成为一体：让学生们一起演奏他们的Scratch乐器来组一个班级乐队！

教学资源

- ❑ 组乐团讲义
- ❑ 组乐团工作室(Build-a-Band studio)
<http://scratch.mit.edu/studios/475523>

反思问题

- + 你最先做什么？
- + 你下一步做什么？
- + 你最后做什么？

审视学生作品

- + 程序计划是否有创意的运用音效？
- + 程序计划中的角色是否是有互动性的？

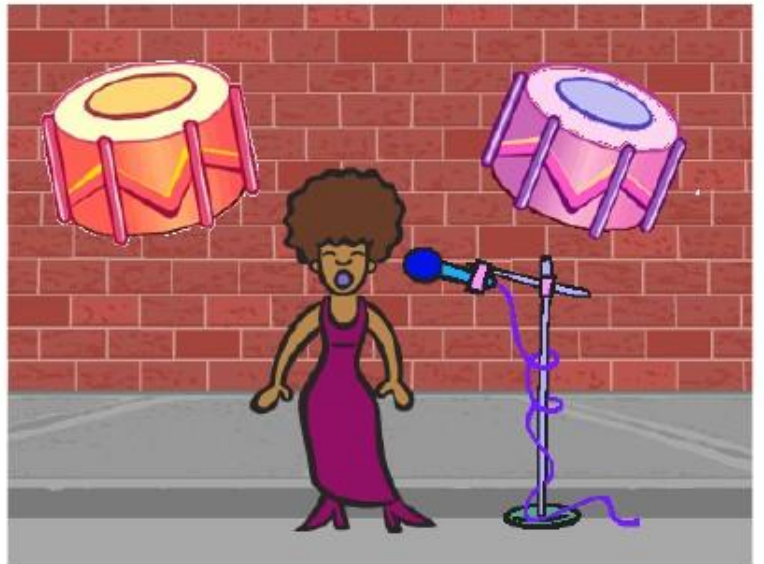
我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

组乐团

你可以如何运用Scratch来创作音效、乐器或是你最喜爱的音乐类型呢？

在这个活动中，你会藉由角色与音效的搭配来设计交互式乐器，并且建立一个音乐性质的Scratch程式计划。



动手开始吧

- 建立一个角色。
- 加入「音效」程序积木。
- 尝试让你的乐器可以产生互动的方式。



从角色范例库中选择一个角色来制作你的乐器



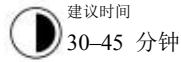
尝试看看

- 运用「重复」程序积木来让音效播放不只一次。
- 输入或录下你自己的音效，或是试验看看音效编辑器(Sounds editor)。
- 试用「节奏(tempo)」程序积木来加速或减慢韵律。

都完成了？

- + 将你的程序计划加入到组乐团工作室(Build-A-Band Studio): <http://scratch.mit.edu/studios/475523>
- + 挑战自己做更多！发明一个新乐器或是录下你自己的声音。
- + 帮助一位邻座的同学！

橘色正方形、 紫色圆形



目标

完成这项活动，学生将：

- + 可以藉由完成这个艺术主题的挑战来表达他们的创造力
- + 对于使用外观程序积木与绘画工具更加流畅顺手

活动描述

- ❑ 老师可以选择性地，从「橘色正方形、紫色圆形工作室」中展示范例程序计划并提供「橘色正方形、紫色圆形讲义」来引导学生。
- ❑ 留时间让学生们创作一个包含了一个橘色正方形及紫色圆形的程序计划。鼓励学生尝试「外观」程序积木与绘画工具来探索他们的艺术天份。
- ❑ 让学生们与他人分享他们的创作。我们建议进行「成果展(gallery walk)」：让学生们把自己的程序计划切换到「放映模式(presentation mode)」然后让学生们四处走动来探索别人的程序计划。老师也可以选择性地让学生们把他们的程序作品加入到「橘色正方形、紫色圆形工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 橘色正方形、紫色圆形讲义
- ❑ 橘色正方形、紫色圆形工作室
<http://scratch.mit.edu/studios/475527>

反思问题

- + 你是如何在程序计划中运用一个橘色正方形和一个紫色圆形呢？这样的点子想法是从哪里来的呢？
- + 这个活动挑战了什么？
- + 这个活动令你惊讶的是什么？

审视学生作品

- + 程序计划内是否包含了一个橘色正方形与紫色圆形吗？

附注

- + 如果学生们遇到困难，提醒他们可以查看「技巧窗口」来进一步学习特定的程序积木与 Scratch 编辑器。
- + Scratch支持位图与向量图形。协助指引学生使用绘图工具中的「向量模式」或「点阵模式」按钮，进一步来设计或修改不同类型的影像和文字。

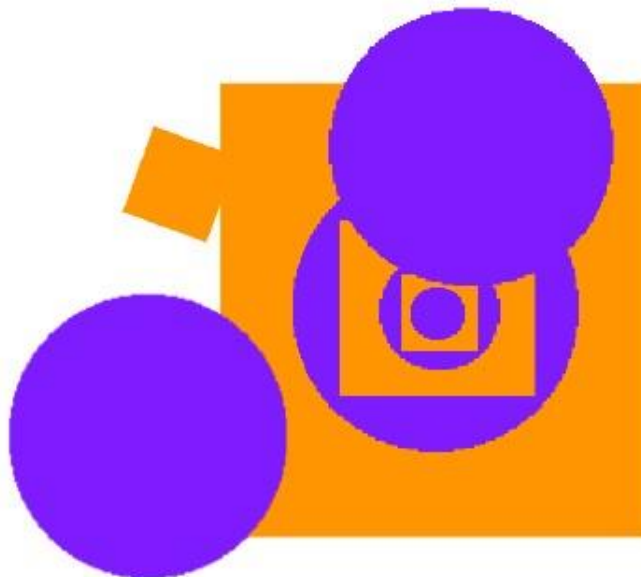
我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

橘色正方形、 紫色圆形

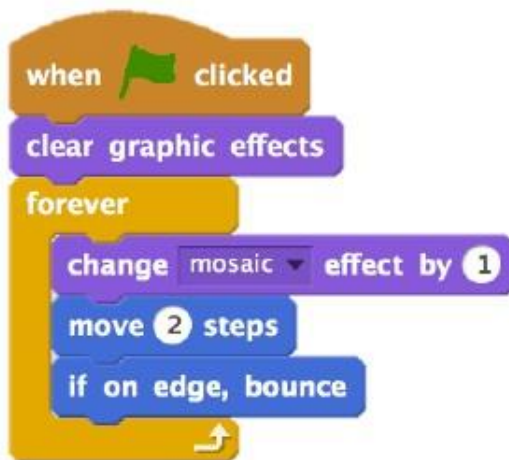
使用一个橘色正方形跟一个紫色圆形，你可以创作出什么程序计划呢？

在这个挑战中，你将会创作一个包含了一个橘色正方形及一个紫色圆形的程序计划。
你会创作什么呢？



动手开始吧

- ❑ 使用「绘图工具 (paint editor)」画出你的角色。
- ❑ 加入不同的外观与造型，让你的角色「活动」起来。
- ❑ 再做一次！



感覺
卡住了？

沒關係！試試這些做法...

都完成了？

- ❑ 试试看与邻座的同学一起脑力激荡！
- ❑ 在你开始用Scratch制作程序计划前，列一张你想尝试的想法清单！
- ❑ 探索看看别人用Scratch做些什么样的程序计划 – 这是一个发现灵感的绝佳方式！

- + 将你的程序计划加入「橘色正方形、紫色圆形工作室」：<http://scratch.mit.edu/studios/475527>
- + 探索位图模式与向量模式的差异 (切换按钮在绘画工具下方)。
- + 挑战自己做更多！加入其他形状与颜色。
- + 与一位伙伴交换程序计划并进行融合 (remix) 创作。

动起来了！



目标

完成这项活动，学生将：

- + 藉由试验「控制」程序积木，更加熟悉程设设计思维的次序(sequence)及循环(loops)的概念
- + 能够解释角色与造型之间的差异
- + 在建构一个动画程序计划中，实际练习试验(experimenting)及重复(iterating)

活动描述

- ❑ 老师可以选择性地，从「动起来了！工作室」中展示范例程序计划并提供「动起来了！讲义」来引导学生。
- ❑ 用重复循环一系列逐渐变化的图片(像是手翻动画书或黏土动画影片)来介绍动画的概念。鼓励学生借着变化造型或背景来创作动画并探索循环(loops)的概念。
- ❑ 让学生们与他人分享他们的创作：让学生们把自己的程序计划切换到「放映模式(presentation mode)」然后让学生们四处走动来探索别人的程序计划。老师也可以选择性地让学生们把他们的程序作品加入到「动起来了！工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 动起来了！讲义
- ❑ 动起来了！工作室(It's Alive! studio)
<http://scratch.mit.edu/studios/475529>

反思问题

- + 角色和造型的差异是什么？
- + 动画是什么？
- + 列出在现实生活中，你体会到循环(Loops)概念的三种方式？(例：每天晚上都要睡觉)。

审视学生作品

- + 学生可以分辨角色跟造型吗？
- + 有些Scratch用户特别喜爱编写动画程序计划并且愿意投入时间来画图设计角色、造型或背景。你如何让学生融入在艺术与科技兼具的程序计划呢？

附注

- + 角色与造型的差异常常困扰着Scratch使用者。演员穿许多戏服的现象可以帮助了解其中的差异。
- + 学生们可以借着相机或网络摄影机拍摄他们自己的照片，再把他们自己的影像做成动画。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

动起来了！

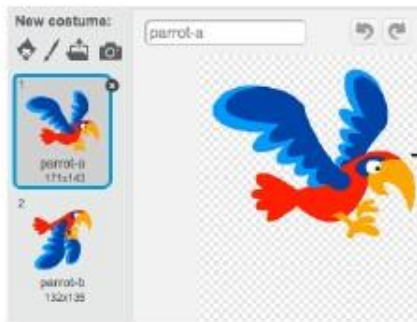
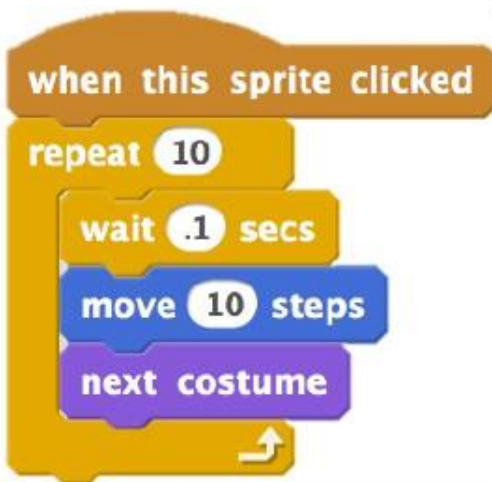
你如何能够拿一张影像或照片并让它动起来呢？

在这活动中，你将会借着编写一连串的造型变化，来探索让角色、影像与想法动起来的方式。



动手开始吧

- ❑ 选择一个角色。
- ❑ 加入一个不同的造型。
- ❑ 加入让影像动起来的程序积木。
- ❑ 再做一次！



尝试看看

- ❑ 尝试先在纸上画出你的动画点子——如同手翻动画书。
- ❑ 尝试不同的程序积木与造型直到你发现有趣的事情。
- ❑ 需要一些灵感？在「浏览」页面中的「动画」类别找找看别人的程序计划。

都完成了？

- + 加你的程序计划加入到「动起来工作室(It's Alive studio)」：<http://scratch.mit.edu/studios/475529>
- + 挑战自己做更多！在你的程序计划加入更多特色，让你的动画看起来更生动。
- + 帮助一位邻座的同学！
- + 与一个同学分享你的程序计划并介绍你的设计过程。
- + 找一个引发你兴趣的动画程序计划并且融合(remix)你自己的想法！

除错 (DEBUG) !

建议时间
15-30 分钟

目标

完成这项活动，学生将：

- + 深入调查五个除错挑战的问题并且提出答案
- + 藉由实际的测试与除错练习，探索一系列的概念，包含次序(sequence)及循环(loops)概念
- + 建立程序除错方法的列表

活动描述

- ❑ 教师可以选择性地提供单元2除错(Debug)讲义在活动中来引导学生。
- ❑ 协助学生从单元2除错工作室中开启除错挑战，或前往讲义中所列出的网址连结。鼓励学生点击「看看里面」来调查这个有问题的程式计划、仔细斟酌有问题的程序代码、并且测试可能的解决方法。
- ❑ 留给学生们时间来测试并排除每个除错挑战的问题。老师也可以选择性地让学生使用Scratch的融合(remix)功能来除错并保存正确的程序计划。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们测试与除错的体验与感受。
- ❑ 藉由搜集学生的「问题找寻(problem finding)」与「问题解决(problem solving)」的方法，产生一份属于班级的除错方法清单。

教学资源

- ❑ 单元2除错(Debug)讲义
- ❑ 单元2除错工作室(Debug It! studio)
<http://scratch.mit.edu/studios/475539>

反思问题

- + 问题是什么？
- + 你是如何辨别、找出这个问题的呢？
- + 你如何修正这个问题？
- + 别人有修正这个问题的其他方法吗？

审视学生作品

- + 学生们能够解决这五个程序错误吗？如果不，你可能会如何说明在这些未解问题中表达的概念呢？
- + 学生们采用了什么不一样的测试与除错方法呢？

附注

- + 可将本活动设置为团体活动，让学们像「演出程序活动」那样将除错挑战演出来，或是将「演出程序」作为一种测试与除错的新方法介绍给学生们。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

除错 (DEBUG)!

救救我！你能够排除这5个Scratch程序计划的错误吗？

在这个活动中，你将研究调查这五个除错挑战中有什么东西出错了，并且为每个挑战找到解决方法。

动手开始吧

- 前往「单元2的除错工作室」：
<http://scratch.mit.edu/studios/475539>
- 对于工作室中的五个除错挑战进行测试并排除错误。
- 写下你的解决办法或用你的方法调整有错的程序。

感覺卡住了？

沒關係！試試這些做法...

- 把程序中可能的错误列出一张清单。
- 追踪并弄清楚你做过的事情！这样可以提醒你你已经做过的尝试并且引导你进行下一个尝试。
- 与你旁边的同学分享并且比较「发现问题」以及「解决问题」的方法，直到发现对你有帮助的方法！

□ 除错挑战! 2.1 <http://scratch.mit.edu/projects/23266426>
在这程序计划中，Scratch猫咪想要跳舞给你看。当你点选他时，他应该开始随着鼓声跳舞。但是，就在猫咪开始跳舞后，它停下来了，但是鼓声还是持续着！我们要如何修正这个程序呢？

□ 除错挑战! 2.2 <http://scratch.mit.edu/projects/24268476>
在这个程序计划中，当按下绿色旗子后，Pico应该朝向Nano移动。当Pico碰到Nano时，Pico应该说：「Tag, you're it!」而且Nano说：「My turn!」但是有地方错了！Pico并没有对Nano说任何话。我们要如何修正这个程序呢？

□ 除错挑战! 2.3 <http://scratch.mit.edu/projects/24268506>
这个程序计划是用来画一个笑脸，但是有地方出错了！画笔从一只眼睛连续画到了微笑，这不应该是这样子的。我们要如何修正这个程序呢？

□ 除错挑战! 2.4 <http://scratch.mit.edu/projects/23267140>
在这个程序计划中，当按下绿色旗子后，一朵花开始绽放，当花全开了便会停止。但是有地方错了！当所有的花瓣都开了后，它并没有停止，动画又重来一次。我们要如何修正这个程序呢？

□ 除错挑战! 2.5 <http://scratch.mit.edu/projects/23267245>
在这个程序计划中，当按下绿色旗子后，生日快乐歌会开始播放。当歌曲结束时，会出现一个提示「click on me to blow out the candles!」但是有地方错了！这个提示在歌曲正在播放时就出现了，而不是等到歌曲结束后。我们要如何修正这个程序呢？

都完成了？

FINISHED!

- + 与一位伙伴讨论你们测试与排除错误的实际过程。将你们方法之间的相同与不同的地方，进行纪录。
- + 用右键对你的程序区域中的积木进行点击，可以增加程序批注。这样做可以帮助别人理解你程序中不同的部分！
- + 帮助一位邻座的同学！

音乐影片



目标

完成这项活动，学生将：

- + 借着制作一个自己当导演的音乐影片，来创作一个结合了动画与音乐的影片程式计划
- + 更为熟悉角色、造型与音效的运用

活动描述

- ❑ 将使用Scratch来结合动画与音乐的构想介绍给学生们。老师可以选择性地从「音乐影片工作室」中挑选一些范例展示给学生们。
- ❑ 留给学生开放式的时间来制作他们的程序计划，并提供「音乐影片讲义」来引导学生并给他们灵感。鼓励学生要在程序计划页面注明他们引用的(属于别人的)构想、音乐或程式码。
- ❑ 协助学生在建构程序计划的过程中，给予他人建议并获得回馈。我们建议每一个学生与一位邻座的同学相互检查：让学生们在中途停下来并对另一位同学或讨论小组(单元0小组讨论活动)分享进行中的作品，来寻求建议。老师可以选择性地邀请学生将作品加入到「音乐影片工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

附注

- + 要让程序计划更具个人特色，协助学生使用「音效」页面中的功能来加入一首最喜爱的歌曲、录下他们自己的声音、或是演奏一个乐器。
- + 有关融合(remix)与抄袭(plagiarism)的疑问可能会在活动中发生。好好把握这个机会，引导学生讨论如何标记与注明出处。可运用Scratch常见问题 — 关于融合：
<http://scratch.mit.edu/help/faq/#remix>

教学资源

- ❑ 音乐影片讲义
- ❑ 音乐影片工作室(Music Video studio)
<http://scratch.mit.edu/studios/475517>

反思问题

- + 你克服的挑战是什么？你是如何克服它的呢？
- + 还有什么事情是你想继续弄清楚的呢？
- + 你是如何注明来表扬你借用在你程序计划中的构想、音乐或程序代码呢？

审视学生作品

- + 这计划是否结合了角色与音效？
- + 学生选择让程序计划的哪一个部分动起来(有动画效果)？
- + 有哪些介绍过的程序积木或概念，到现在学生还是觉得困惑呢？你会如何帮助他们理解呢？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

音乐影片

你可以如何结合音乐与动画来创作属于你的Scratch音乐影片呢？

在这个程序计划中，你将探索与剧院、歌曲、舞蹈、音乐、绘画、素描、摄影与动画有关的构想来创作一个专属于你个性化的音乐影片。



动手开始吧

- ❑ 加入音效。
- ❑ 创作一个角色并让它动起来。
- ❑ 让音效与角色一起互动！



尝试看看



- ❑ 使用造型来帮助你的动画生动起来！
- ❑ 将你的角色加入「点选(click)」、「按下按键(key press)」与更多的程序积木，让它更具有互动性。
- ❑ 在程序计划页面加入说明指引，让其他人了解如何与使用你的程序计划来与它互动。

TIPS & TRICKS 可以运用的程序积木

都完成了？



- + 新增你的程序计划到音乐影片工作室：
<http://scratch.mit.edu/studios/475517>
- + 一定要注明来表扬任何用在你计划中的音乐、程序代码或其他创作。
- + 挑战自己做更多！创作你自己的角色、音效与造型。

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

簡体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新簡体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 3

故事



你的进度

本单元包含



人物	58
对话	60
场景	62
除错 (DEBUG) !	64
打造生物	66
故事接龙	68

单元 3

整体概观

「大处」着眼THE

Andres Monroy-Hernandez (Scratch在线社群初始版本的设计总监)在探讨融合文化的博士论文中,他在简介的部分引用了三段话:

长期以来,在他人的作品上继续创作是程式设计的实际状况。现今的网络科技让我们能够取得各式各样的作品,这个状况受到网络科技的影响更加扩大。程序设计思维的一个重要概念是透过「再利用(reuse)」与「融合(remix)」进而促进学习者之间的连结。Scratch开发环境及在线社群可以在这个关键的实际情况中帮助初学者。藉由帮助使用者找寻构想与程序代码来继续创作,Scratch能够帮助他们创造比起完全自己创作更复杂的程序计划。

本单元中的活动提供了培养再利用及融合氛围的初步构想与方法。你如何能够进一步促进学习者之间的分享与连结呢?



学习目标

学生们将:

- + 将更为熟悉且了解「再利用(reuse)」与「融合(remix)」等概念,在设计程序的过程中带来的助益。
- + 对于程设思维的概念(事件events、平行parallelism)与实际操作(试验experimenting与重复iterating、测试testing与除错debugging、再利用reusing与融合remixing)更加熟练
- + 藉由共同设计与描述情节,探索以故事为主题的程序创作。

关键词、概念与实作

- | | | |
|--------------------------------|------------------------------|----------------------|
| + 再利用与融合(reusing and remixing) | (pass-it-on story) | + 设计展示 (design demo) |
| + 创造一个积木 | + 双人结对程式设计(pair programming) | |
| + 背包 | + 作品放映 (scratch screening) | |
| + 舞台 | | |
| + 故事接龙 | | |

附注

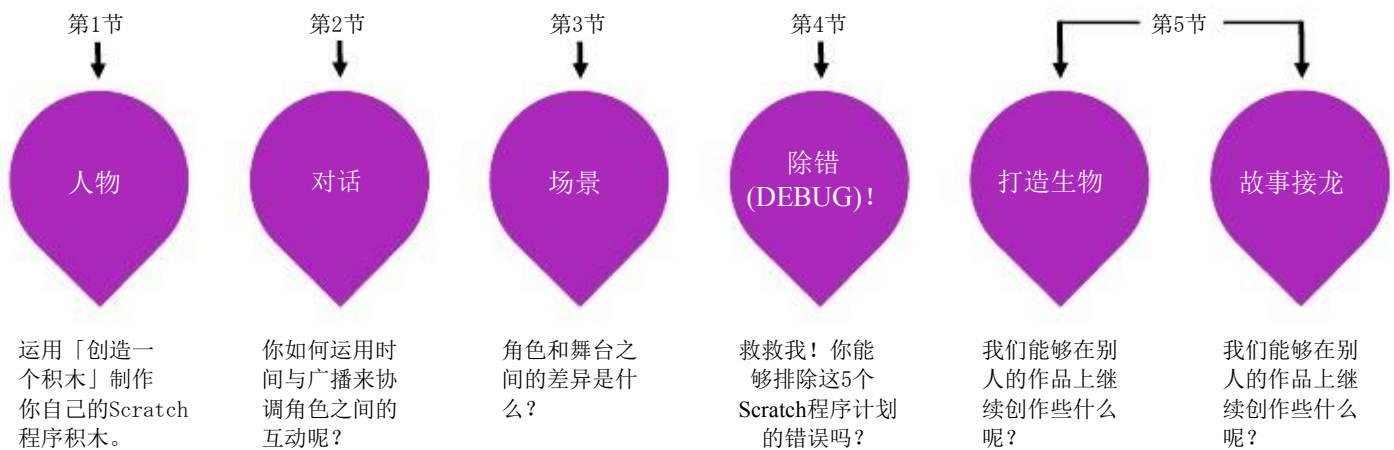
- + 再利用与融合的活动有助于建立阅读程序代码的能力,也可能引发关于所有权及著作权的疑问。对于共同合作的成果要如何:促成、讨论及评估,思考不同的策略方法。

选择你自己的冒险旅程



本单元着重于帮助学生藉由多样化「动手作」与「不用计算机」的活动，建立说故事与融合的能力。同时，本单元也着重于让学生们有机会共同合作，并在他人的作品上继续创作。以单元2的学习经验作基础，这个单元中的活动是设计来帮助学生更加深入熟悉程序设计的概念(事件与平行)与实际操作(试验与重复、再利用与融合)。每一个能力养成的活动都设计来帮助学生创作说故事的程序计划，像是：探索新的程序积木与方法来编写角色与舞台之间的互动，最终结合来完成故事接龙的程序计划。

可行的学习途径



人物

建议时间
30-45 分钟

活动描述

- ❑ 老师可以选择性地，从「人物工作室」中展示范例程序计划并提供「人物讲义」来引导学生。
- ❑ 保留时间给学生使用「更多积木」类别中的「制作一个积木」功能，创作属于他们自己的程序积木。协助他们设计两个人物（或称为角色），每个人物有两种行为。老师可以选择性地带着整个班级，从头到尾示范一次「制作一个积木」的功能。
- ❑ 鼓励学生们与他人分享他们的人物与行为。我们建议举行一个设计展示(design demo)的活动：邀请几位学生对班级发表他们的作品并展示他们是如何实际操作「制作一个积木」的功能。老师也可以选择性地让学生将他的程序计划加入到「人物工作室(Character studio)」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

附注

- + 如果学生们对于「制作一个积木」的功能感到困惑，邀请他们先探索「人物工作室」，看看其他计划的程序代码来了解他人是如何运用的。
- + 更加深入了解「制作一个积木」的功能，可参考这个教学：<http://bit.ly/makeablock>

目标

完成这项活动，学生将：

- + 使用「制作一个积木」的功能，试验替人物定义动作
- + 更为熟悉「事件(event)」与「平行(parallelism)」的程序设计概念，并实际操作「试验(experimenting)」与「重复(iterating)」

教学资源

- ❑ 人物讲义
- ❑ 人物工作室(Character studio)
<http://scratch.mit.edu/studios/475545>

反思问题

- + 你会如何向他人解释「制作一个积木」这个功能呢？
- + 什么时候你可能会使用「制作一个积木」的功能呢？

审视学生作品

- + 程序作品是否包含了两个人物，且每个人物都有两个使用「制作一个积木」功能创作的行为？
- + 学生们是否能够对其他人以及你解释「制作一个积木」的功能该如何使用？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

人物

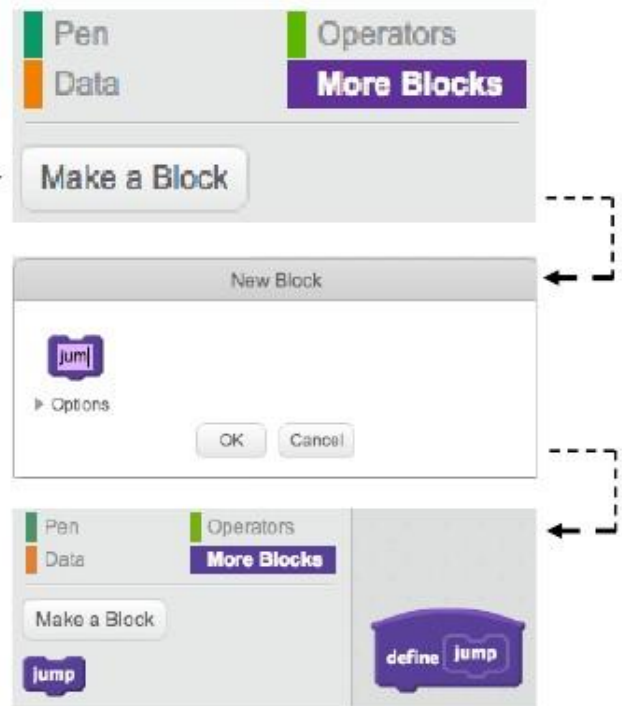
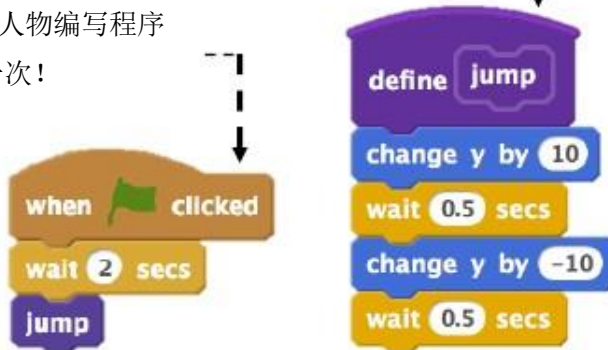
你想创造自己的Scratch程序积木吗？

试验看看Scratch中「制作一个积木」的功能！在这个活动中，你将会制作数个程序积木，这些积木则定义(define)了两个不同角色的两种行为。



动手开始吧

- ❑ 从角色库选择、画出、或上传两个角色人物。
- ❑ 在「更多积木」类别中点选「制作一个积木」来创作并「命名」你的积木。
- ❑ 在「定义(define)」积木下增加其他程序积木，来控制你的积木会做的行为。
- ❑ 试验用你创造的程序积木来替你的人物编写程序
- ❑ 再做一次！



尝试看看

- ❑ 遇到困难了？没关系！先看看这个影片来了解如何开始「制造一个积木」：<http://bit.ly/makeablock>
- ❑ 探索在人物工作室(Character studio)中的其他计划，来看看别人都创作了什么新的程序积木。
- ❑ 有时候，同样的行为可以有多种方法来定义。尝试用不同积木的组合来试用更多选项及结果。

都完成了？

- + 将你的程序计划加入「人物工作室(Character Studio)」：<http://scratch.mit.edu/studios/475545>
- + 挑战自己做更多！使用「制作一个积木」的功能来试验不同的人物与行为。
- + 帮助一位邻座的同学。

对话

建议时间
30-45 分钟

活动描述

- ❑ 老师可以选择性地，以小组的方式来探索「企鹅说笑话」初学者程序计划，并且提供「对话讲义」来引导学生。
- ❑ 邀请学生仔细观察「企鹅说笑话」程序计划是如何运用「等待」程序积木让对话动起来。让学生运用融合的功能并且使用「广播(broadcasting)」、「广播并等待(broadcasting and wait)」、「当我接收到(when I receive)」程序积木来让调整对话，重新设计「企鹅说笑话」程序计划。
- ❑ 鼓励学生们与他人分享他们的「说笑话」计划。我们建议举行一个设计展示(design demo)的活动：邀请几位学生对班级发表他们的作品并展示他们是如何实际运用「广播」积木。老师也可以选择性地让学生将他的程序计划加入到「对话工作室(Conversations studio)」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

NOTES

- + 如果学生对于「成对」使用「广播(broadcasting)」与「当我接受到(when I receive)」程序积木感到困惑，邀请他们探索「广播案例工作室(Broadcast Examples studio)」中的范例程式码：<http://scratch.mit.edu/studios/202853>

目标

完成这项活动，学生将：

- + 藉由融合一个说笑话的程序计划，探索两种不同的方式来让角色间的互动步调一致：时间(timing)及广播(broadcasting)
- + 更为熟悉「事件(event)」与「平行(parallelism)」的程序设计概念，并且实际练习「再利用(reuse)」与「融合(remix)」

教学资源

- ❑ 对话讲义
- ❑ 企鹅说笑话初学者程序计划
<http://scratch.mit.edu/projects/10015800>
- ❑ 对话工作室(Conversations studio)
<http://scratch.mit.edu/studios/475547>

反思问题

- + 你会如何向其他人，描述「广播(broadcast)」的概念呢？
- + 在程序计划中，什么时候你会使用「时间(timing)」？什么时候你会使用「广播(broadcasting)」？

审视学生作品

- + 程序计划是否运用了「广播」与「当我接收到」程序积木？
- + 学生们可以解释如何运用「广播」、「广播并等待」、「当我接收到」等三种程序积木吗？

NOTES TO SELF

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

对话

有哪些不同的方法可用来协调角色之间的互动呢？



在这个活动中，你将会探索不同的程序设计方式来让角色产生对话！藉由融合一个说笑话的程序计划，来试验使用「时间(timing)」并尝试「广播(broadcast)」程序积木。查看[广播案例工作室](#)来获得更多灵感。

动手开始吧

- ❑ 查看企鹅说笑话(Penguin Jokes)程序计划的内容：<http://scratch.mit.edu/projects/10015800>
- ❑ 研究程序代码来了解「等待(wait)」与「说(say)」程序积木是如何协调对话。
- ❑ 用「广播(broadcast)」及「当我接收到(when I receive)」程序积木，来取代「等待(wait)」，融合这个程序计划。

```
when clicked
say Hello! for 2 secs
wait 2 secs
say What do Penguins love to eat? for 3 secs
wait 2 secs
say Nope... for 2 secs
wait 2 secs
say Ice-burgers! for 2 secs
```

```
when I receive message1
broadcast message1
broadcast message1 and wait
```

感覺卡住了？

沒關係！試試這些做法...

都完成了？

- ❑ 与一位邻座同学讨论构想！写下一个可能解答的清单并且一起试验这些解答。
- ❑ 尝试在你的程序计划不同的部分，使用「广播」与「当我收到」程序积木
- ❑ 探索在「对话工作室」中的程序计划来获得灵感，可帮助你运用不同的方式来协调角色之间的对话。

- + 将你的程序计划加入「对话工作室」：<http://scratch.mit.edu/studios/475547>
- + 挑战自己做更多！加入其他人物与对话。
- + 与一个邻座同学分享你的程序计划，并带着他从头到尾，说明一次你尝试与设计的过程。
- + 帮助一位邻座同学！

场景

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 将会借着创造一个新的程序计划来试验变换背景，就像一个具有很多场景的故事或幻灯片
- + 更为熟悉「事件(event)」与「平行(parallelism)」的程序设计概念，并实际操作「试验(experimenting)」与「重复(iterating)」

活动描述

- ❑ 老师可以选择性地，从「场景工作室」中展示范例程序计划并提供「场景讲义」来引导学生。
- ❑ 保留时间给学生来创作一个包含了多个场景变化(就像是幻灯片一样)的程序作品。鼓励学生们去探索与修改舞台中的程序代码来进行背景变化。
- ❑ 鼓励学生们与他人分享他们的程序计划。我们建议举行一个设计展示(design demo)的活动：邀请几位学生对班级发表他们的作品并展示他们是如何实际操作「背景(backdrops)」变化。老师也可以选择性地让学生将他的程序计划加入到「场景工作室(Scene studio)」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 场景讲义
- ❑ 场景工作室(Scenes studio)
<http://scratch.mit.edu/studios/475550>

反思问题

- + 舞台与角色有什么是相同的吗？
- + 舞台是如何与角色不同呢？
- + 你是如何让在舞台中的角色开始动作？
- + 在动画以外，哪些其他类型的程序计划会用到场景变化呢？

审视学生作品

- + 程序计划是否成功的运用背景变换来调整多个场景呢？

附注

- + 如果学生们对于如何「变换背景」感到困惑，邀请他们先探索「人物工作室」，鼓励他们尝试使用在外观(look)分类下的这些程序积木：「切换背景到」、「切换背景到...并等待」、「下一个背景」。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

场景

角色和舞台之间的差异是什么？

在这个活动中，你将会借着创造一个新的程序计划来试验变换背景，就像一个具有很多场景的故事或幻灯片。



动手开始吧

- 从范例库中挑选、自己绘画、或是上传档案，将多个背景加到你的程序计划中。
- 试验使用「外观(looks)」与「事件(events)」类别中的程序积木来变换背景。
- 将程序代码加入到舞台与角色中，以便在背景变换时，来调整及控制他们的反应。



switch backdrop to backdrop1

when backdrop switches to backdrop1

backdrop name

尝试看看

- 找寻角色与舞台之中与背景相关的程序积木，并且试用它们，看看发生什么变化！
- 需要更多灵感吗？浏览Scratch在线社群来找寻使用多重背景的程序计划吧。

都完成了？

- + 将你的程序计划加入背景工作室：
<http://scratch.mit.edu/studios/475550>
- + 挑战自己做更多！在你的程序计划中加入更多背景变化。
- + 帮助一位邻座的同学！
- + 打开一个你过去的计划或是找到一个引发你兴趣的计划，藉由加入背景变化，融合(remix)产生新计划。

除错 (DEBUG) !

建议时间
15-30 分钟

目标

完成这项活动，学生将：

- + 深入调查五个除错挑战的问题并且提出答案
- + 藉由实际的测试与除错练习，探索一系列的概念，包含事件(event)及平行(parallelism)概念

活动描述

- ❑ 教师可以选择性地提供单元3除错(Debug)讲义在活动中来引导学生。
- ❑ 协助学生从单元3除错工作室中开启除错挑战，或前往讲义中所列出的网址连结。鼓励学生点击「看看里面」来调查这个有问题的程式计划、仔细斟酌有问题的程序代码、并且测试可能的解决方法。
- ❑ 留给学生们时间来测试并排除每个除错挑战的问题。老师也可以选择性地让学生使用Scratch的融合(remix)功能来除错并保存正确的程序计划。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们测试与除错的体验与感受。
- ❑ 藉由搜集学生的「问题找寻(problem finding)」与「问题解决(problem solving)」的方法，产生一份属于班级的除错方法清单。

教学资源

- ❑ 单元3除错(Debug)讲义
- ❑ 单元3除错工作室(Debug It! studio)
<http://scratch.mit.edu/studios/475554>

反思问题

- + 问题是什么？
- + 你是如何辨别、找出这个问题的呢？
- + 你如何修正这个问题？
- + 别人有修正这个问题的其他方法吗？

审视学生作品

- + 学生们能够解决这五个程序错误吗？如果不，你可能会如何说明在这些未解问题中表达的概念呢？
- + 学生们采用了什么不一样的测试与除错方法呢？

附注

- + 能够读懂别人的程序代码不但是非常重要的技巧，也是能够融入「再利用(reuse)」与「融合(remix)」实际操作的关键。
- + 这个活动是进行「双人结对程序设计(pair programming)」的好机会。将学生两人分除一组进行除错挑战。
- + 藉由在Scratch程序积木右键单击来加入程序代码注解，学生可以解释程序代码的修订与调整。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

除错 (DEBUG)!

救救我！你能够排除这5个Scratch程序计划的错误吗？

在这个活动中，你将研究调查这五个除错挑战中有什么东西出错了，并且为每个挑战找到解决方法。

动手开始吧

- ❑ 前往「单元3的除错工作室」：
<http://scratch.mit.edu/studios/475554>
- ❑ 对于工作室中的五个除错挑战进行测试并排除错误。
- ❑ 写下你的解决办法或用你的方法调整有错的程序。

感覺卡住了？

沒關係！試試這些做法...

- ❑ 把程序中可能的错误列出一张清单。
- ❑ 追踪并弄清楚你做过的事情！这样可以提醒你你已经做过的尝试并且引导你进行下一个尝试。
- ❑ 与你旁边的同学分享并且比较「发现问题」以及「解决问题」的方法，直到发现对你有帮助的方法！

❑ 除错挑战! 3.1 <http://scratch.mit.edu/projects/24269007>
在这个程序计划中，Scratch猫咪教Gobo叫「meow」。但是轮到Gobo尝试时 — Gobo却是保持安静。我们要如何修正这个程序呢？

❑ 除错挑战! 3.2 <http://scratch.mit.edu/projects/24269046>
在这个程序计划中，Scratch猫咪应该要从1数到用户输入的数字。但是Scratch猫咪总是数到10。我们要如何修正这个程序呢？

❑ 除错挑战! 3.3 <http://scratch.mit.edu/projects/24269070>
在这个程序计划中，Scratch猫咪正在轮流点名Gobo的朋友：Giga、Nano、Pico跟Tera。可是现在却是同时点到所有人的名字。我们要如何修正这个程序呢？

❑ 除错挑战! 3.4 <http://scratch.mit.edu/projects/24269097>
在这个程序计划中，Scratch Cat与Gobo正在练习跳跃。当Scratch Cat说「跳！」，Gobo应该往上跳再下来。但是Gobo没有跳。我们要如何修正这个程序呢？

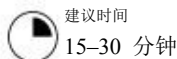
❑ 除错挑战! 3.5 <http://scratch.mit.edu/projects/24269131>
在这个程序计划中，当你按下右箭头键时，场景会变换。这个计划的主角— 恐龙，应该在每个场景都隐藏，除了当场景切换到礼堂时。在礼堂中，恐龙应该要出现并开始跳舞。但现在，恐龙一直出现而且不在正确的时间跳舞。我们要如何修正这个程序呢？

都完成了？

FINISHED!

- + 与一位伙伴讨论你们测试与排除错误的实际过程。将你们方法之间的相同与不同的地方，进行纪录。
- + 用右键对你的程序区域中的积木进行点击，可以增加程序批注。这样做可以帮助别人理解你程序中不同的部分！
- + 帮助一位邻座的同学！

打造生物



目标

完成这项活动，学生将：

- + 藉由参与一次共同绘画创作，开始认识并且实际练习「再利用(reuse)」与「融合(remix)」的程序设计概念

活动描述

- ❑ 在这个活动中，学生们将分三个部分来画出一只「生物」。
- ❑ 发给学生们每人一张折成三分之一的图画纸，并留给他们一分钟的时间来画出生物的「头部」。接着，让学生们把纸张的前三分之一折起来，把生物的头藏起来，只留下一点点记号，方便继续画。把头部藏起来后，让学生们将画图纸交换给另外一个人。然后，再给学生们一分钟的时间，从头部提示的记号，继续来画出生物的「中间」部分，但不可让学生偷看头部。同样的，让学生把中间的部分折好隐藏起来(留下继续画的提示)，再把图画纸传给另外一个人。最后，留给学

们一分钟的时间画出生物的「下半部」。完成后，将图画纸打开来，揭晓分工合作下创造的生物！

- ❑ 将图画贴在墙壁或板子上，让学生可以观看他们的创作与贡献。
- ❑ 以小组方式来讨论：共同著作(co-authorship)、分工合作(collaboration)、与再利用(resue)及融合(remix)的作品。

教学资源

- ❑ 空白纸 (A4即可)，折成三折
- ❑ 画图用具(铅笔、原子笔、签字笔等)

审视学生作品

- + 学生们是否能够解释融合是什么，以及融合带来的帮助呢？

附注

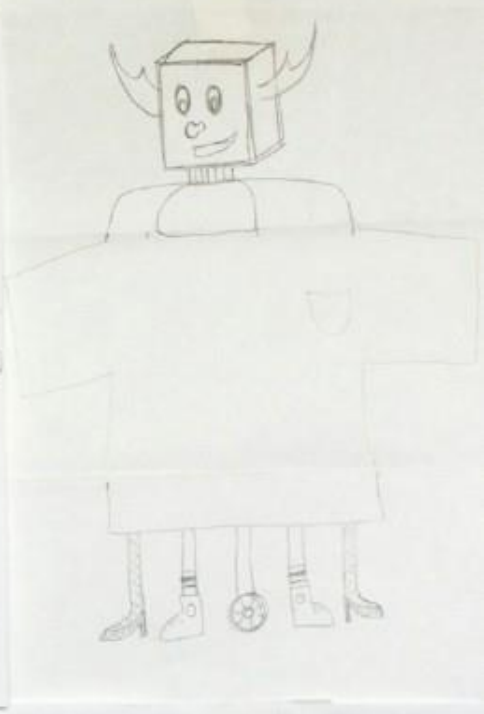
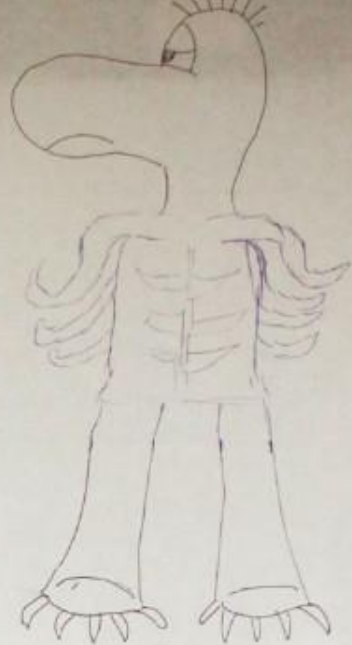
- + 这个活动是「故事接龙」计划的绝佳热身操！我们建议在「故事接龙」前，举行「打造生物」活动。
- + 老师可以选择性地，让学生在每一次绘图创作时，都在右下角写下他们的名字，来辨别每个人的创作贡献。

反思问题

- + 你对于「融合(remix)」的定义是什么？
- + 回想你起头的生物(你画头的部分)，你的构想是如何藉由别人的帮忙来延续及加强？
- + 回想你帮忙延续的生物(你画中间或是下半部的部分)，别人的构想是如何藉由你的帮忙来延续或加强呢？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



故事接龙



目标

完成这项活动，学生将：

- + 能够藉由「再利用」与「融合」别人的作品，来创作一个说故事的Scratch程序计划
- + 体验用「双人结对程序设计 (pair programming)」来开发一个共同参与的说故事程序计划

活动描述

- ❑ 将学生们每两人分成一组。对学生们解说故事接龙的概念：首先，两人小组开始创作，然后把计划传给其他的两人小组，让他们继续延伸创作与重新构想。老师可以选择性地，印出「故事接龙」讲义。
- ❑ 鼓励学生们以他们想要的任何方式开始创作，像是：人物、场景、情节、或是任何他们喜欢的事物。留给每个小组10分钟的时间来创作他们共同合作的故事，接着让他们交换，藉由融合的概念来延伸创作别人的故事。鼓励学生注明他们再利用或融合的内容。
- ❑ 两次交换之后，让学生们回头看看他们曾参与创作的故事计划。我们建议举办一个程序放映会 (Scratch screening)：使用投影机跟屏幕来展示故事计划，并让学生们聚在一起观看。老师可以选择性地，邀请学生们将他们的程式计划加入到「故事接龙」工作室或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 故事接龙讲义
- ❑ 故事接龙工作室 (Pass It On studio)
<http://scratch.mit.edu/studios/475543>
- ❑ 投影机跟屏幕，用来展示学生的作品 (非必备)

反思问题

- + 运用融合 (remix) 并在别人的作品上继续创作，让你感觉如何？当别人融合你的作品时，你感觉如何？
- + 在你的生活中还有什么地方你曾看到或体验到再利用跟融合呢？分享两个例子。
- + 「与别人一起创作」跟「自己一人创作Scratch计划」的经验比较起来，有怎么样的不同呢？

审视学生作品

- + 学生们继续创作程序计划的哪个部分？
- + 学生们能理解并接受「事件 (event)」及「平行 (parallelism)」的概念以及「再利用 (reuse)」与「融合 (remix)」的实际操作吗？如果不能，还有什么方法能够进一步说明清楚呢？

附注

- + 可以考虑把程序放映会 (Scratch screening) 举办成一个活动！邀请其他班级的学生来观看，提供点心跟饮料，或是将活动办在礼堂或是有大墙面跟屏幕的房间来展示作品。
- + 向学生介绍「背包 (backpack)」，它位在Scratch编辑器画面的下方，这是另一种融合程序计划的方式。学习更多背包的使用方法：<http://bit.ly/scratchbackpack>

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

故事接龙

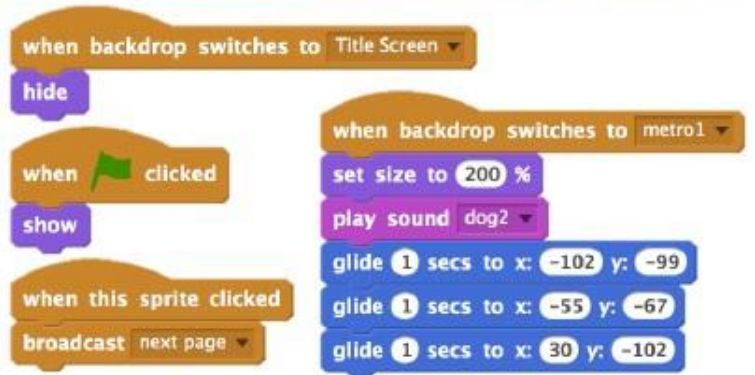
我们能够在别人的作品上继续创作些什么呢？

在这个程序计划中，你将起头一个动画故事的程序计划，然后你将计划传给别人，让别人继续创作，融合、延伸并且重新构想！



动手开始吧

- 开始运用人物、场景、情节或更多你感兴趣的内容，来创作一个说故事的程序计划。
- 10分钟后，储存并在线分享你的计划。
- 交换并且运用融合(remix)的概念，继续延伸创作别人的故事。
- 再做一次！



尝试看看

□ 发想不同的可能性，来融合、延伸或重新构想故事。你想在后面加入一个新的场景吗？你是否可以重新构想故事开始前发生了什么呢？如果增加一个新人物会发生什么事呢？增加一个情节转折看看？还有什么呢...

□ 在你的程序代码中加入批注可以帮助别人理解你程序中不同的部分。要对程序代码增加批注：用右键对你的程式区域中的积木进行点击，并且加入说明。



可以运用的程序积木

都完成了？



- + 新增你的计划到故事接龙工作室：
<http://scratch.mit.edu/studios/475543>
- + 帮助一位邻座的同学！
- + 回头看看你有贡献创作的故事，瞧瞧故事变得如何了！

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

簡体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新簡体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 4

创造游戏



YOU ARE

WHAT'S



梦想游戏清单
新手村游戏
记录得分
延伸变化
互动谜题
除错(DEBUG)!

本单

74
76
80
82
84
86

单元 4

整体概观

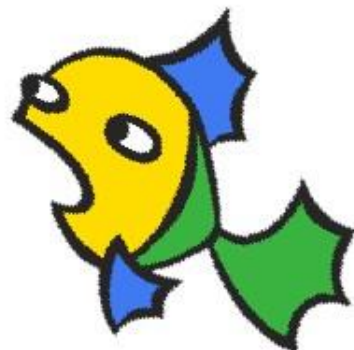
「大处」着眼THE

个人化(Personalization)对于设计创意计算机思维是一个重要的导引原则。我们认为「个人化」的意义有两种：与个人的喜好兴趣密切相关、每个人的喜好兴趣可能是大异其趣。知(knowing)与行(doing)可以有多种方式，而探讨这些不一样的方式有助于维系初学者的学习兴趣、动机与恒心。在这个单元中，学生们将以探索一些进阶的程序设计观念，以及与游戏设计相关的挑战谜题。当活动中的进阶概念或解谜挑战对学习个人非常有(实)用时，这些活动就变得更为适合这位学习者。我们以Mitch Resnick (MIT Scratch计划总监)所分享的故事，作为一个案例来说明学习情境的影响力：

好几年前，那时候的我在学校中的一个计算机俱乐部(教室)，我遇见了一个正在创作个人游戏的13岁男孩。那时候的他已经可以控制一只鱼的角色。他想要做的是游戏可以纪录得分，这样一来你就可以知道大鱼吃了多少只小鱼，但是小男孩并不知道怎么做。

看到这个机会，我向他介绍了变量(variables)的构想。我示范了这个给他看，他马上就知道他可以如何运用这个程序积木来记录游戏中多少条鱼被吃了。他挑出了程序积木并且把它放在大鱼吃小鱼的程序那里。他很快的试验了一下。而理所当然的，每一次大鱼吃掉小鱼时，得分就会往上加一。

我认为小男孩对于变量有了更深刻的理解，因为他那时真的非常想运用变量的概念。这就是Scratch的目标之一。也不仅仅是变量，更是所有类型的概念。我们认为：当孩子以实用且有动机的方式运用特定概念时，他们对于那个概念会有更深入的理解。



学习目标

学生们将：

- + 对于以下的程序设计概念有所认识：条件(conditions)、运算符(operators)与数据(data, 例如：变数variables与清单lists)。
- + 藉由自己设计的迷宫游戏、乒乓游戏或是滚动条游戏，对于程设思维的概念(事件events、平行parallelism)与实际操作(试验experimenting与重复iterating、测试testing与除错debugging、再利用reusing与融合remixing、抽象化abstracting与模块化modularizing)更加熟练
- + 找出并了解游戏共通的结构

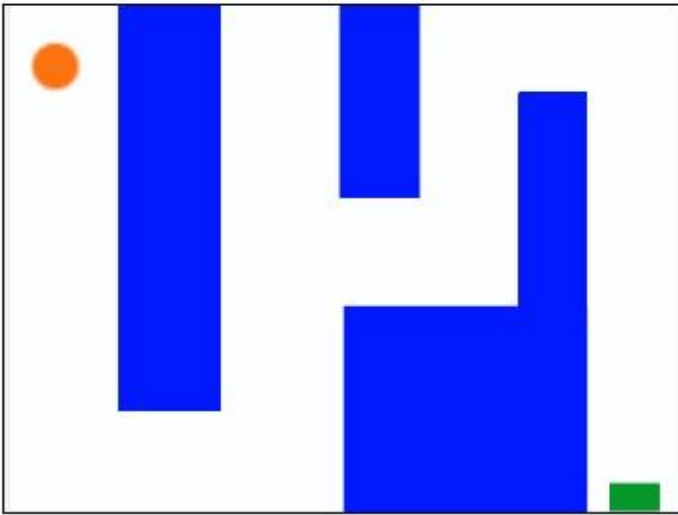
关键词、概念与实作

- + 抽象化与模块(variables and lists) + 谜题罐(puzzle化(abstracting andjar) modularizing)+ 侦测(sensing)+ 倒出构想(brain dump)
- + 条件(conditionals) + 回馈博览会(dump)(feedback fair)+ 运算符(operators)
- + 资料(data)+ 电玩日(arcade day)+ 变量与列表

附注

- + 本单元探索了很多新概念，因此我们增加了额外的补充教材：范例程序计划工作室、额外的新程序谜题与新手村游戏计划，我们鼓励老师需要时进行融合与再利用。

选择你自己的冒险旅程



在这个单元，学习者将变成游戏设计师并实际体验制作自己的游戏程序。藉由本单元许多活动的引导，学生们在认识程设思维概念(条件、运算符、数据)与实际操作(抽象化与模块化)的同时，将学习游戏的机制与开发。

你可以让学生从新手村游戏开始建构他们自己的游戏程序，再以其他活动提供进一步的协助。单元4提供了学生们多样的机会来实际练习游戏的开发，例如：学习游戏都有的得分与横向卷动机制，甚至是多人游戏的设计(例如：乒乓球)。

可行的学习途径



梦想游戏 清单

建议时间
15-30 分钟

目标

完成这项活动，学生将：

- + 能够识别游戏所共有的设计要素

活动描述

- ❑ 将学生以2-3人分成小组。
- ❑ 在小组中，要求学生写下一份他们喜欢玩的游戏清单。学生可以用他们的设计手执或是一张空白纸。我们建议进行「倒出构想 (brain dump)」活动：让学生在很短的时间内(1-2分钟)尽可能写下游戏的名称，越多越好。然后，让学生们开始缩小范围，从清单中挑出他们喜爱的游戏。
- ❑ 几分钟后，询问每个小组的游戏清单中：所有游戏都有的东西是什么？什么样的设计特色让程序变成了游戏？
- ❑ 让班级开始讨论什么特色、特点构成了游戏，并且产生一份列表来记录游戏共有的机制。接着邀请学生们开始想象他梦想中的游戏并且针对梦想游戏写下设计元素的列表。
- ❑ 邀请学生们在分组内分享他们的清单，也可以进行小组讨论(见单元0的小组讨论活动)，让学生们获得建议与回馈。

附注

- + 在进行单元4的其他游戏设计活动时，邀请学生回头看看这份梦想游戏清单。

教学资源

- ❑ 用来写下设计要素的空白纸(非必要)
- ❑ 画图用具(铅笔、原子笔、签字笔等)

反思问题

- + 把你喜爱的游戏写成一张清单。
- + 这些游戏都有的东西是什么？
- + 什么样的设计特色让程序变成了游戏？
- + 为你梦想中的游戏制作一份设计要素的清单。

审视学生作品

- + 梦想游戏清单包含了游戏的特色？
- + 从班级小组制作的清单中比较，哪些设计元素是相似的？哪些是不同的？
- + 从这些清单中，你得知学生喜爱的游戏类型与游戏进行的方式是哪些呢？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

Chess

Monopoly

Mario

Clue

Football

Candyland

Pac
Man

Jump
Rope

Baseball

Tennis

Flappy
Bird

Wheel of
Fortune

Four
Square

新手村 游戏

建议时间
45-60 分钟

活动描述

- ❑ 在这个活动中，学生们将会创建一个新手村游戏程序计划，这个计划在后续的「记录得分」、「延伸变化」、「互动谜题」活动中还可以进一步修改与扩充。老师可以选择性地，向学生们展示「迷宫」、「乒乓」、「滚动条」的新手村游戏计划，并且提供「迷宫讲义」、「乒乓讲义」、「滚动条游戏讲义」来引导学生。
- ❑ 选择一个游戏让整个班级一起来制作，或是让学生在以下游戏中挑选一个他想要制作的游戏：迷宫、乒乓、滚动条。留给学生时间来制作游戏程序或让他们融合其中一个新手村游戏计划。
- ❑ 鼓励学生对于还在制作中的游戏获得意见反馈。我们建议举办「回馈博览会(feedback fair)」：先让一半的学生坐在位置上并将他们的游戏计划开启。另一半的学生则四处走动并且探索游戏程序、提出问题、给予建议回馈。一段时间后，两方交换。老师可以选择性地邀请学生将他们的游戏加入到游戏工作室或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____

目标

完成这项活动，学生将：

- + 藉由着手进行自己设计的游戏计划，更加熟悉程序设计概念(条件conditions、运算符operators、数据data)与实际操作(试验experimenting与重复iterating、测试testing与除错debugging、再利用reusing与融合remixing、抽象化abstracting与模块化modularizing)。

教学资源

- ❑ 迷宫讲义
- ❑ 迷宫游戏初学者范例
<http://scratch.mit.edu/projects/11414041>
- ❑ 乒乓讲义
- ❑ 乒乓游戏初学者范例
<http://scratch.mit.edu/projects/10128515>
- ❑ 滚动条游戏讲义
- ❑ 滚动条游戏初学者范例
<http://scratch.mit.edu/projects/22162012>
- ❑ 游戏工作室(Games studio)
<http://scratch.mit.edu/studios/487504>

反思问题

- + 设计自己的游戏，有什么样的挑战？
- + 在这个活动中，你最引以为傲的事情是什么？

审视学生作品

- + 游戏中包含了条件、运算符、数据等程序设计概念吗？

附注

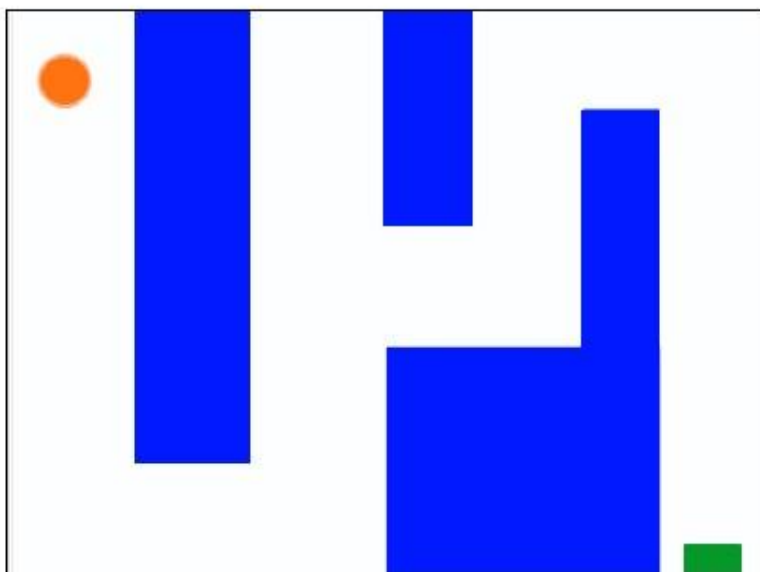
- + 我们建议举办电玩日(arcade day)来庆贺与分享游戏作品。每个游戏计划都切换到放映模式(presentation mode)，让每个学生可以四处走动并且试玩每一个别人的游戏。
- + 滚动条游戏包含了「分身(cloning)」的概念。可运用单元5的分身讲义来帮助学生使用「分身(cloning)」程序积木。

迷宫

你要如何运用Scratch创作一个互动性的游戏呢？

在这个程序计划中，你将会创作一个游戏。这个游戏包含了「角色之间的互动」、「记录得分」还有「等级关卡」。游戏目标：你从迷宫的起点开始移动角色，途中避免接触到墙壁，直到终点。

□ 迷宫游戏初学者范例
<http://scratch.mit.edu/projects/11414041>



动手开始吧

- 画出一个迷宫图样的背景，并且使用不同颜色来标示墙壁与迷宫终点。
- 加入一个角色。
- 让你的游戏变得有互动性！

```

when right arrow key pressed
  point in direction 90
  move 10 steps

when down arrow key pressed
  point in direction 180
  move 10 steps

when left arrow key pressed
  point in direction -90
  move 10 steps

when up arrow key pressed
  point in direction 0
  move 10 steps
  
```

上面这些程序代码，可以让玩家控制角色在迷宫的移动。

尝试看看

- 加入多重等级关卡到你的游戏！这个构想可以运用「不同的背景」与「广播(broadcast)积木」来触发下个等级关卡。
- 使用「变量(variable)积木」来记录分数！
- 实验使用「计数器(timer)」积木来为你的迷宫游戏加入新的挑战！

```

when green flag clicked
  go to x: -205 y: 147
  
```

这段程序代码会设定你的角色回到迷宫的起点。

```

when green flag clicked
  forever
    if touching color ?
      move -10 steps
  
```

这段程序代码会让你的角色碰到墙壁时会反弹。

```

when green flag clicked
  forever
    if touching Ball ?
      say You win!
  
```

这段程序代码用来设定「迷宫终点角色」：当玩家操纵的角色(球)接触到迷宫终点时，显示玩家胜利。

可运用的程序积木

都完成了？

<pre> when space key pressed 当按下空格键 when up arrow key pressed 当按下向上键 when m key pressed 当按下m键 when I receive message 1 当我接收到讯息1 </pre>	<pre> score set score to 0 设定变量score为0 change score by 1 设定变量score改变1 show variable score 显示变量score hide variable score 隐藏变数score </pre>	<pre> - 减 + 加 < 小于 = 等于 > 大于 not 不成立 and 和 or 或 </pre>	<pre> pick random 1 to 10 在1到10间随机选一个数 touching ? touching color ? color is touching ? timer reset timer 定时器 重设定计时器 </pre>
---	---	--	--

- + 将你的程序计划加入到游戏工作室(Games Studio)
：
<http://scratch.mit.edu/studios/487504>
- + 与一位同学交换程序，互相从头到尾解说一次你的程序

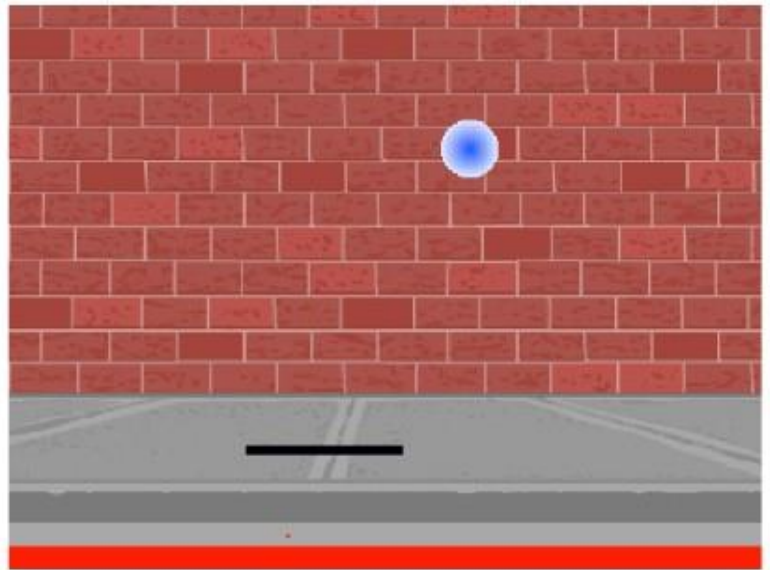
乒乓

你要如何运用Scratch创作一个互动性的游戏呢？

在这个程序计划中，你将会创作一个游戏。这个游戏包含了「角色之间的互动」、「记录得分」还有「等级关卡」。这个游戏类似经典的PONG。游戏目标：保持不要让角色从你旁边穿过。

□ 乒乓游戏初学者范例

<http://scratch.mit.edu/projects/10128515>



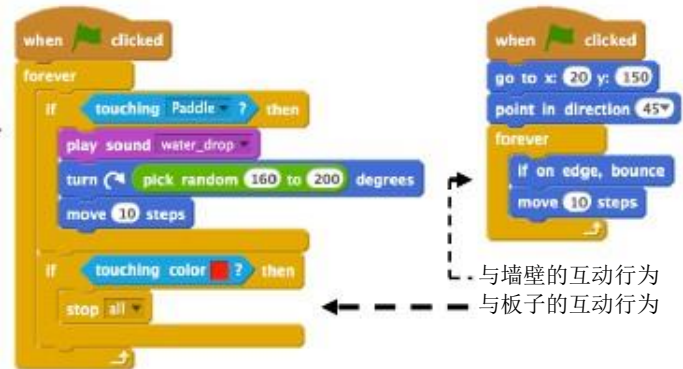
动手开始吧

- 创造两个角色：一个让玩家控制的板子、一个与玩家互动的球。
- 让你的板子角色变得有互动性。
- 让你的游戏动起来！



尝试看看

- 如何增加游戏的难度呢？创造不同的等级关卡、使用定时器、或记录分数，是一些你可以做的范例。
- 试验借着编修舞台背景来改变游戏的外观！
- 试验使用不同的按键来控制你的角色！



这段程序代码用来控制球的行为：如果球碰到了板子或墙壁，球会继续移动。如果球碰到了红色区域，游戏就结束了（代表球穿过了板子）。

可运用的程序积木

都完成了？



- + 将你的程序计划加入到游戏工作室 (Games Studio)
：
<http://scratch.mit.edu/studios/487504>
- + 与一位同学交换程序，互相从头到尾解说一次你的程序

滚动条

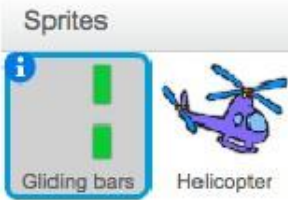
你要如何运用Scratch创作一个互动性的游戏呢？

在这个程序计划中，你将会创作一个游戏。这个游戏包含了「角色之间的互动」、「记录得分」还有「等级关卡」。这个游戏类似「Flappy Bird」，游戏玩法：让你的角色不要坠落到地面、也不要碰到某些物体。



动手开始吧

- ❑ 创造两个角色：一个让玩家控制的角色（直升机），另外一个是要避免的角色（滑动栅栏）。



```
when space key pressed
  change y by 20
```

控制角色移动

- ❑ 让直升机变得有互动性。
- ❑ 让你的游戏动起来：增加程序代码让滑动栅栏卷动横向通过舞台！

```
when green flag clicked
  hide
  forever
    wait 5 secs
    create clone of myself
```

```
when green flag clicked
  go to x: 0 y: 0
  set size to 30 %
  wait 2 secs
  forever
    change y by -2
```

让角色持续往下掉

尝试看看

- ❑ 如何增加游戏的难度呢？创造不同的等级关卡、使用定时器、或记录分数，是一些你可以做的范例。
- ❑ 试验借着编修舞台背景来改变游戏的外观！
- ❑ 试验使用不同的按键来控制你的角色！

这段程序代码会制造分身 (clones)，分身会用于下面的程序代码制造横着滑过画面的栅栏。

```
when I start as a clone
  switch costume to pick random 1 to 3
  go to x: 240 y: 0
  show
  glide 8 secs to x: -240 y: 0
  delete this clone
```

```
when green flag clicked
  forever
    if touching color ? then
      stop all
```

设定游戏结束的时机

可运用的程序积木

都完成了？

可运用的程序积木

<pre>when space key pressed 当按下空格键</pre> <pre>when up arrow key pressed 当按下向上键</pre> <pre>when m key pressed 当按下m键</pre> <pre>when I receive message 1 当我接收到讯息1</pre>	<pre>score</pre> <pre>set score to 0 设定变量score为0</pre> <pre>change score by 1 设定变量score改变1</pre> <pre>show variable score 显示变量score</pre> <pre>hide variable score 隐藏变数score</pre>	<pre>-</pre> 减 <pre><</pre> 小于 <pre>></pre> 大于 <pre>and</pre> 和 <pre>or</pre> 或	<pre>+</pre> 加 <pre>=</pre> 等于 <pre>not</pre> 不成立 <pre>and</pre> 和 <pre>or</pre> 或	<pre>pick random 1 to 10 在1到10间随机选一个数</pre> <pre>touching ?</pre> <pre>touching color ?</pre> <pre>color is touching ?</pre> <pre>timer</pre> 定时器 <pre>reset timer</pre> 重设定定时器
---	--	---	--	--

- + 将你的程序计划加入到游戏工作室 (Games Studio)
<http://scratch.mit.edu/studios/487504>
- + 与一位同学交换程序，互相从头到尾解说一次你的程序

记录得分



目标

完成这项活动，学生将：

- + 有能力描述什么是变量(variables)以及为什么变数很有用
- + 对于程序设计中的数据(data)概念有所认识
- + 体会融合(remix)与再利用(reuse)程序计划或其中一部份的技巧

活动描述

- ❑ 老师可以选择性地，以分组的分式来探索大鱼吃小鱼初学者计划，并且提供「记录得分讲义」给学生们。
- ❑ 协助学生们开启「大鱼吃小鱼初学者计划」。留给学生们时间，藉由融合大鱼吃小鱼初学者计画来探索变量的概念并且将得分机制加入游戏之中。老师可以选择性地，保留时间给学生，让他们将得分的概念加入到之前的「迷宫」、「乒乓」、「卷轴」游戏计划之中。
- ❑ 让学生们分享「大鱼吃小鱼融合计划」或「加入游戏得分的计划」。我们建议举行「设计展示(design demo)」活动：邀请几位学生对小组发表他们的作品并示范他们是如何运用「变量(variables)」来记录游戏得分。老师可以选择性地，让学生们把他们的游戏计划加入到「大鱼吃小鱼融合工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 记录得分讲义
- ❑ 记录得分范例工作室
<http://scratch.mit.edu/studios/218313>
- ❑ 大鱼吃小鱼初学者计划(Fish Chomp starter project)
<http://scratch.mit.edu/projects/10859244>
- ❑ 大鱼吃小鱼融合工作室(Fish Chomp remix studio)
<http://scratch.mit.edu/studios/475615>

反思问题

- + 你会如何跟其他人解释变量呢？
- + 变量有用的地方是什么？

审视学生作品

- + 学生们是否能够解释「什么是变量」与「什么是变量有用的地方」呢？

附注

- + 鼓励学生藉由探索「得分范例工作室」中的案例与程序代码，来厘清他们对于变量的认识。
- + 变量是一项重要的数学与程序设计概念。在数学或自然课，课程会教导学生们变量的概念，但是很多学生学习变量时会感到困难。游戏是让学生更明白变量用途的一种方式。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

记录得分

你要如何在Scratch程序计划中记录分数呢？

大鱼吃小鱼 (Fish Chomp) 是一个玩家使用鼠标控制角色，让角色尽可能多抓鱼的游戏。在这个活动中，你会借着运用变量来加入游戏得分，融合出一个新的大鱼吃小鱼游戏。



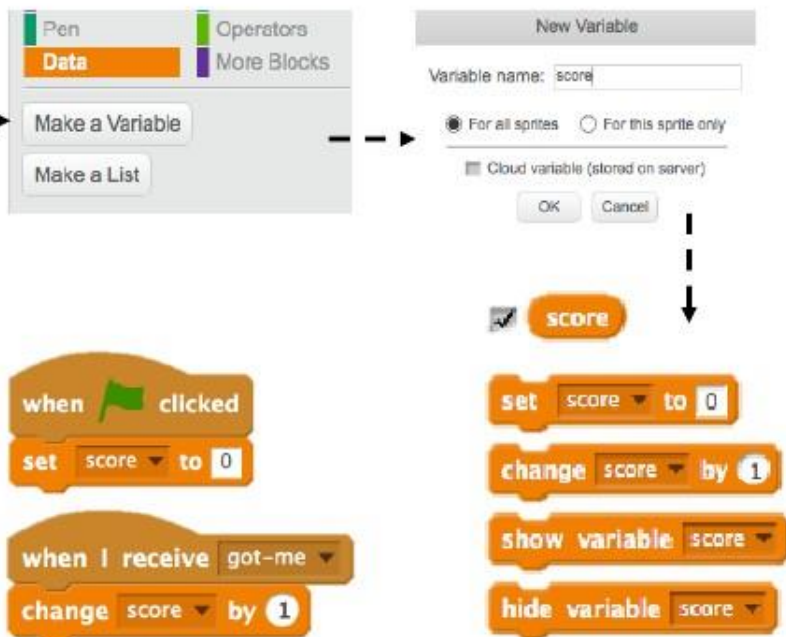
动手开始吧

- 前往大鱼吃小鱼游戏计划：

<http://scratch.mit.edu/projects/10859244>

- 点选数据 (Data) 类别中的「制作一个变量」，来创造并且命名一个用来记录游戏得分的变数。

- 试验使用你的新变量程序积木来加入游戏得分到你的程序计划中！



感覺卡住了？

沒關係！試試這些做法...

都完成了？

- 不确定怎么运用变量？瞧瞧这个程序计划来获得更多信息：<http://scratch.mit.edu/projects/2042755>
- 或是看这个影片：<http://youtu.be/uXq379XkhVw>
- 查看得分案例工作室，探索并探讨游戏中记录得分的程序代码，来进一步学习如何创造变量以及如何将记录得分运用到程序计划中。

- + 将你的游戏计划加入到大鱼吃小鱼融合工作室：<http://scratch.mit.edu/studios/475615>
- + 挑战自己做更多！你会如何运用得分来增加游戏设计的困难度？
- + 找到一个你感兴趣的并且融合出一个新游戏。

延伸变化



目标

完成这项活动，学生将：

- + 藉由探索游戏共通机制的程序计划，更加熟悉以下的概念：条件(conditionals)、运算符(operators)和资料(data)

活动描述

- ❑ 老师可以选择性地，展示延伸变化工作室中的范例程序，并且提供延伸变化讲义来引导学生。
- ❑ 留给学生时间来探索「延伸变化工作室」中的程序代码，调查可以用来增加游戏难度或延伸游戏的不同方法。要求学生们挑选一个或更多的延伸变化加入到之前的「迷宫」、「乒乓」或「卷轴」游戏计划之中。留给学生时间去试验和整合延伸变化到他们的游戏之中。
- ❑ 鼓励学生与其他人分享他们扩充了的游戏程式。我们建议举办「双人结对分享(pair-share)」或「设计展示(design demo)」活动来让学生分享他们的游戏并且展现他们所学。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 延伸变化讲义
- ❑ 延伸变化工作室(Extensions studio)
<http://scratch.mit.edu/studios/475619>

反思问题

- + 哪些不同的方式可以增加游戏的难度度？
- + 你在你的游戏计划中加入了哪些延伸变化？
- + 请叙述一下：你将延伸变化加入到你的游戏之中的处理过程？

审视学生作品

- + 学生们是否能够将延伸变化加入到他们原本的游戏计划中呢？

附注

- + 为了提供更多的「支架式教学(scaffolding, 鹰架理论)」给需要额外帮助的学生，我们建议带着整个班级从头到尾读过一个延伸变化的范例程序(例：等级关卡)，并且协助学生将那个延伸变化加入到他们的游戏计划。
- + 「背包(backpack)」工具：学生们可以用它来将「延伸变化程序」中的某些部份整合到他们自己的新手村游戏。更多背包的用法：
<http://bit.ly/scratchbackpack>

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

延伸变化

你会如何扩充并重新构想以Scratch制作的游戏呢？

借着加入延伸变化的特色到你的Scratch计划中来设计游戏！选择至少一个(或更多)右边的延伸变化，并将它加入到你之前的「迷宫」、「乒乓」、「卷轴」游戏计划之中。

动手开始吧

- 前往延伸变化工作室：
<http://scratch.mit.edu/studios/475619>
- 挑选一个(或更多)延伸变化来深入探索。
- 将你选择的延伸变化整合到你之前的新手村游戏计划中！

- + 游戏得分(SCORE) <http://scratch.mit.edu/projects/1940443>
示范如何设定并改变游戏得分。每一次点击Scratch猫咪就会获得10分。
- + 等级关卡(LEVELS) <http://scratch.mit.edu/projects/1940453>
示范如何改变等级关卡。每一次按压空格键就会获得1分。每10分增加1个等级。
- + 定时器(TIMER) <http://scratch.mit.edu/projects/1940445>
示范如何使用定时器。使用鼠标来引导Scratch猫咪找到Gobo。
- + 敌人(ENEMIES) <http://scratch.mit.edu/projects/1940450>
示范如何加入敌人。使用向上键与向下键来躲开网球。
- + 收集奖品(REWARDS) <http://scratch.mit.edu/projects/1940456>
示范如何收集物品。使用箭头键移动Scratch猫咪来收集游戏物品。
- + 鼠标控制(MOUSE) <http://scratch.mit.edu/projects/25192659>
示范如何设定用鼠标来控制游戏。移动鼠标来移动板子。
- + 重新开始(RESTART) <http://scratch.mit.edu/projects/25192935>
示范如何设定一个案件来让游戏重新开始。点击RESTART按钮来重新开始游戏。
- + 选单(MENU) <http://scratch.mit.edu/projects/25192991>
示范如何在游戏开始时显示一个选单画面。点击选单画面中的START或DIRECTIONS。
- + 多人游戏(MULTIPLAYER)
<http://scratch.mit.edu/projects/25192711>
示范如何在游戏中多增加另一个玩家。玩家1使用箭头键来引导Pico穿越迷宫。玩家2使用W, A, S, D来引导nano穿越迷宫。

尝试看看

- + 用Scratch设计程序时，「背包(backpack)」是一个超级有用的工具。他可以装入任何东西：程序代码、音乐档案、角色和其他，尝试使用背包来整合延伸变化到你的游戏计划中。
- + 也可以这样做：在你的设计手札上画出构想和一些程序代码，这是另外一种用来规划如何整合延伸变化到你的程序的好方法。

都完成了？

- + 在你的「迷宫」、「乒乓」、「卷轴」游戏计划之中，加入另一个延伸变化。
- + 挑战自己做更多！继续将每一个延伸变化都加入到你的游戏中。
- + 帮助一位邻座的同学！
- + 跟一个同学分享你的程序计划，并且互相给予游戏一些建议。

互动谜题



目标

完成这项活动，学生将：

- + 借着解决一系列的9个程序谜题，来探索让程序具有互动性的不同方法
- + 更熟悉以下的概念：条件(conditionals)、运算符(operators)和数据(data)，与实际练习：测试(testing)和除错(debugging)

活动描述

- ❑ 让学生们用Scratch写程序来解开这9个互动谜题，可以单独一人或以2至3人的小组进行挑战。这些互动谜题探索了「侦测(sensing)」程式积木，并融合了关于Scratch程序互动性的进阶概念。老师可以选择性地，在活动中提供互动谜题讲义给学生。
- ❑ 每一个谜题都有数种可能的解决方法。邀请学生或小组来分享不同的解法和构想。我们建议举办「双人结对分享(pair-share)」或「设计展示(design demo)」活动来让学生分享他们的计划并描述解谜的过程。老师可以选择性地，让学生将他们的程序计划加入到「互动解谜工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，回想他们解谜时的挑战。

教学资源

- ❑ 互动谜题讲义
- ❑ 互动谜题工作室(Interactions studio)
<http://scratch.mit.edu/studios/487213>

反思问题

- + 你着手解开哪几个谜题呢？
- + 你解开这些谜题的概念与方法是什么？
- + 哪几个谜题帮助你设想你自己的游戏计画呢？

审视学生作品

- + 谜题都解开了吗？
- + 学生们是否尝试探索解谜的其他方法？
- + 是否有哪些程序积木或概念，学生还是觉得困惑呢？你会如何帮助他们理解呢？

附注

- + 可挑选凸显了新程序积木的特定谜题，来让学生们去探索。或者让学生们发明(创作)程式谜题。
- + 重新定位这些程序谜题做为热身活动或是让某些提早完成其他活动的学生的额外补充。也制作一个谜题罐：将每个谜题在纸上印出来、剪下、对折，最后放入罐子里面。然后让学生从里面抽出要解谜的题目。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

互动谜题

Scratch程序计划的静态影像与影片有什么差异呢？

解开这9个融合了进阶Scratch互动性概念的谜题。每一个挑战都有几种可能的解决方案。

动手开始吧

- ❑ 为这9个互动谜题，每一个都新建一个Scratch程序计划。

感覺
卡住了？

沒關係！試試這些做法...

❑ 谜题 1: 无论什么时候按下B键，角色都会放大一点。无论什么时候按下S键，角色都会缩小一点。

❑ 谜题 2: 无论什么时候，当角色听到一个大的声响，它就改变颜色。

❑ 谜题 3: 无论什么时候，当角色位在屏幕上方的百分之25内时，它会说「I like it up here」。

❑ 谜题 4: 当角色碰触到蓝色区域，它会发出高的音色。当角色碰触到红色区域，它会发出低的音色。

❑ 谜题 5: 无论什么时候，当两个角色碰在一起时，其中一个会说「不好意思(Excuse me)」。

❑ 谜题 6: 无论什么时候，当小猫的角色靠近小狗的角色时，小狗会转过来追逐小猫。

❑ 谜题 7: 无论什么时候，当你点击背景时，在那个地方就会开出一朵花。

❑ 谜题 8: 无论什么时候，当你点击一个角色时，所有其他的角色都会跳一段舞。

❑ 谜题 9: 无论什么时候，当你移动鼠标光标时，角色就会跟着跑，但是角色不会接触到鼠标光标。

都完成了？

- ❑ 在开始用Scratch写程序前，先在你的设计手札上写下你的构想：要如何解开每一个互动谜题的可能作法。
- ❑ 与一位邻座同学一起解谜。与一位伙伴合作是一个解决问题的好方法，不只如此，还会获得使用Scratch写程序的新观念与想法。

- + 将每一个解谜的程序计划都加入到互动谜题工作室：<http://scratch.mit.edu/studios/487213>
- + 帮助一位邻座的同学！
- + 与一位伙伴讨论你解谜的观念与方法，并且将其中的相同与不同的地方写成笔记。

除错 (DEBUG) !



目标

完成这项活动，学生将：

- + 深入调查五个除错挑战的问题并且提出答案
- + 藉由实际的测试与除错练习，探索一系列的概念，包含条件(conditionals)、运算符(operators)及资料(data)

活动描述

- ❑ 教师可以选择性地提供单元4除错(Debug)讲义在活动中来引导学生。
- ❑ 协助学生从单元4除错工作室中开启除错挑战，或前往讲义中所列出的网址连结。鼓励学生点击「看看里面」来调查这个有问题的程式计划、仔细斟酌有问题的程序代码、并且测试可能的解决方法。
- ❑ 留给学生们时间来测试并排除每个除错挑战的问题。老师也可以选择性地让学生使用Scratch的融合(remix)功能来除错并保存正确的程序计划。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们测试与除错的体验与感受。
- ❑ 藉由搜集学生的「问题找寻(problem finding)」与「问题解决(problem solving)」的方法，产生一份属于班级的除错方法清单。

教学资源

- ❑ 单元4除错(Debug)讲义
- ❑ 单元4除错工作室(Debug It! studio)
<http://scratch.mit.edu/studios/475634>

反思问题

- + 问题是什么？
- + 你是如何辨别、找出这个问题的呢？
- + 你如何修正这个问题？
- + 别人有修正这个问题的其他方法吗？

审视学生作品

- + 学生们能够解决这五个程序错误吗？如果不，你可能会如何说明在这些未解问题中表达的概念呢？
- + 学生们采用了什么不一样的测试与除错方法呢？

附注

- + 这个活动提供了一个机会来确认可能需要一些额外注意或协助的学生。特别是关于以下的程序设计概念的运用：
 - + 条件(conditionals)，例如：如果... (if)
 - + 运算符(operators)，例如：运算类(arithmetic)、逻辑类(logical)
 - + 数据(data)，例如：变量(variable)、列表(list)

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

除错 (DEBUG)!

救救我！你能够排除这5个Scratch程序计划的错误吗？

在这个活动中，你将研究调查这五个除错挑战中有什么东西出错了，并且为每个挑战找到解决方法。

动手开始吧

- ❑ 前往「单元4的除错工作室」：
<http://scratch.mit.edu/studios/475634>
- ❑ 对于工作室中的五个除错挑战进行测试并排除错误。
- ❑ 写下你的解决办法或用你的方法调整有错的程序。

感覺卡住了？

沒關係！試試這些做法...

- ❑ 把程序中可能的错误列出一张清单。
- ❑ 追踪并弄清楚你做过的事情！这样可以提醒你你已经做过的尝试并且引导你进行下一个尝试。
- ❑ 与你旁边的同学分享并且比较「发现问题」以及「解决问题」的方法，直到发现对你有帮助的方法！

❑ 除错挑战! 4.1 <http://scratch.mit.edu/projects/24271192>
在这个程序计划中，每当Scratch猫咪捡起一个新物品的时候，「仓库(inventory)」清单都应该更新。但现在，Scratch猫咪只能捡起笔记本电脑。我们要如何修正这个程序呢？

❑ 除错挑战! 4.2 <http://scratch.mit.edu/projects/24271303>
在这个程序计划中，Scratch猫咪收集到黄色Gobo时会得到10分，撞到粉红色Gobo时会减10分。但是有地方错了！我们要如何修正这个程序呢？

❑ 除错挑战! 4.3 <http://scratch.mit.edu/projects/24271446>
在这个程序计划中，Scratch猫咪会想着1到10之间的一个数字。但是在检查猜测的数字时却发生了错误——回复的答案有矛盾。我们要如何修正这个程序呢？

❑ 除错挑战! 4.4 <http://scratch.mit.edu/projects/24271475>
在这个程序计划中，每当Scratch猫咪被网球碰到的时候，「# of hits」应该会显示增加1次。但现在，当Scratch猫咪被碰到时，「# of hits」却增加超过1次。我们要如何修正这个程序呢？

❑ 除错挑战! 4.5 <http://scratch.mit.edu/projects/24271560>
在这个程序计划中，Scratch猫咪要能在迷宫中找寻黄色长方形。但现在，Scratch猫咪却可以穿过墙壁。我们要如何修正这个程序呢？

都完成了？

- + 与一位伙伴讨论你们测试与排除错误的实际过程。将你们方法之间的相同与不同的地方，进行纪录。
- + 用右键对你的程序区域中的积木进行点击，可以增加程序批注。这样做可以帮助别人理解你程序中不同的部分！
- + 帮助一位邻座的同学！

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 5

深入沉潜



你的进度

本单元包含



我知道、我想要、我学到	92
第二回合	94
进阶概念	96
硬件与延伸套件	100
活动设计	102
我的除错 (DEBUG) 谜题	106

单元 5

整体概观

「大处」着眼THE

释出了前一版的指引手册后，我们从教师们得到一个普遍的回馈是：老师们（以及他们所协助的学生）想要有一段「迎头赶上（catch-up）」的时间，利用这段时间来回头看、再次回顾与延伸在之前单元所创造的构想与程序项目。作为响应，我们加入了「深入沈潜」这个单元。

这单元是一个让学习者融入于沉思与自省的好时机，无论是用来更深入进阶概念与练习、或是回顾过去所学。有什么是能够弄清楚却还没有的？关于Scratch，学习者仍然想知道些什么？能如何帮助学习者？——而学习者又能如何帮助他人？

这也是个让身为教学者的你融入于相似的沉思与反省之中。什么事情令你感到惊讶？什么事情让你感到坐立难安？什么事情你在下一次会采取不一样的作法？为什么？



学习目标

学生们将：

- + 根据过去的学习经验，自我评估现在当下的学习目标与需要
- + 借着延伸扩充先前的程序计划，创作一个自我融合（self-remix）的新专案
- + 对于能将Scratch链接到现实世界的硬件套件有所认识
- + 借着探索Scratch最新的特色（影像侦测、分身），更加熟悉程序设计概念与实际操作
- + 试验替别人设计学习体验

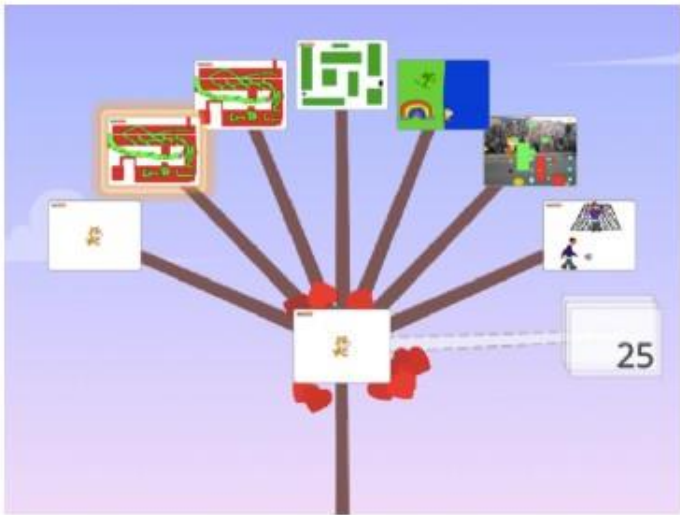
关键词、概念与实作

- + 影像侦测 (video sensing)
- + 同侪访谈 (peer interviews)(extensions)
- + 分身 (cloning)
- + 硬件 (hardware)

附注

- + 找不到你要的？你可以自由地融合（remix）、再利用（reuse）及重新构想（reimagine）教学指引中的任何一个活动，让它最适合你和学生所用。
- + 到ScratchEd网站上寻找为特定课程所设计的教学计画、活动与资源：<http://scratched.gse.harvard.edu>。

选择你的冒险旅程

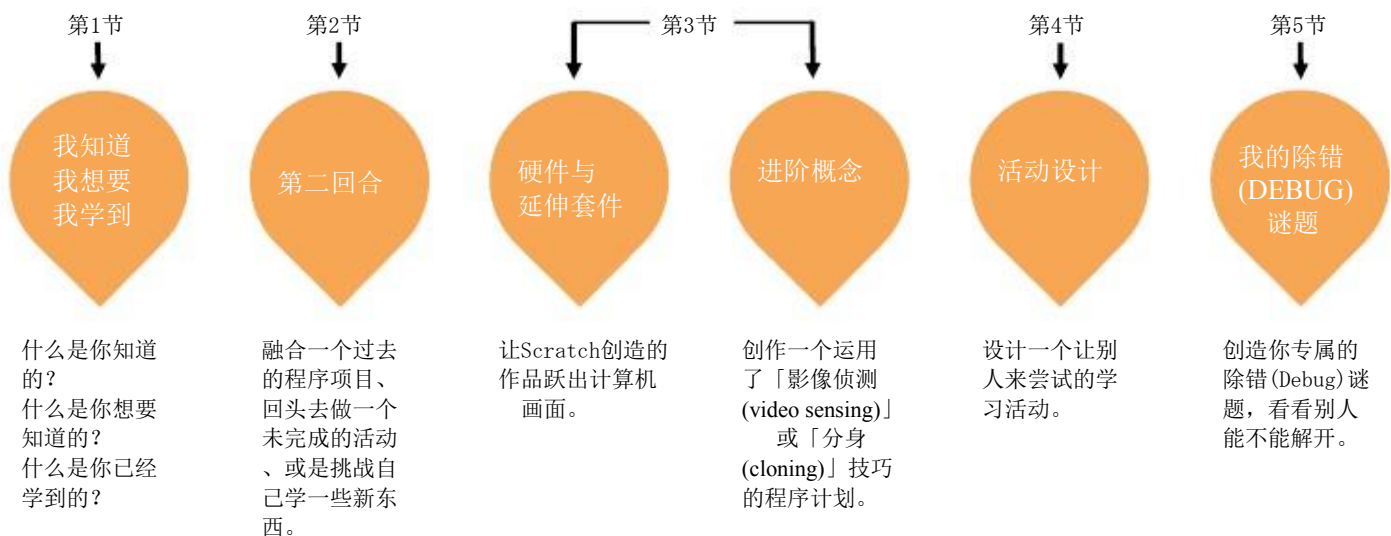


不同于先前三个单元着重在特定的主题或类型，这个单元特意地提供了回顾与回想先前作品的空间。本单元活动特别具有弹性，借着重新回顾挑战、延伸技巧或加强练习等活动来深入探索创意程设思维。

首先，邀请学生回顾他们先前的作品，并在「我知道、我想要、我学到」活动中，投入于自我评估的学习目标。

接着，鼓励学生借着挑选想学习的后续活动，更深入探索Scratch。

可行的学习途径



我知道、我想要、我学到

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 省思先前的程序计划与学习经验
- + 自我评估现在当下的知识与学习目标
- + 在一个由个人主导的研究活动中，追寻个人感兴趣的学习目标

活动描述

- ❑ 在这一个由个人主导的学习活动中，学生们将回顾省思目前理解的事物，并且根据个人的兴趣构筑新知识。老师可以选择性地，提供「我知道、我想要、我学到」作业表格来引导学生。
- ❑ 要求学生回想有关于Scratch以及创意程设思维：什么是他们已经知道的？什么是他们接下来想要知道的？引导学生在设计手札或「我知道、我想要、我学到」作业表格上来回答前两个反思问题。接下来，留给学生们时间，从「什么是你想要知道的？」的回答中，去追寻感兴趣的学习目标。最后，让学生在设计手札或「我知道、我想要、我学到」作业表格上，回答第三和第四个反思问题。
- ❑ 协助学生们彼此分享他们回想的内容以及感兴趣的学习目标。我们建议举行同侪访谈(peer interviews)活动：将学生们两人分成一组，并让他们轮流询问对方有关于：回顾省思、自我评估以及自主研究的过程。

附注

- + 协助学生们在他们的研究过程中，找出并使用其他资源，像是：请教懂得较多的同学、向家人或朋友们提问、或是在Scratch讨论区上进行提问。

教学资源

- ❑ 我知道、我想要、我学到 作业表格
- ❑ Scratch 维基百科(Wiki)
<http://wiki.scratch.mit.edu>
- ❑ Scratch 讨论区
<http://scratch.mit.edu/discuss>
- ❑ Scratch 常见问题(FAQ)
<http://scratch.mit.edu/help/faq>

反思问题

- + 什么是你(已经)知道的？
- + 什么是你想要知道的？
- + 你学到了什么？
- + 你用什么方法来调查你想要知道的事物？

审视学生作品

- + 学生们是否可以学到他们想要知道的事物？
- + 学生们使用了什么样的方法与资源？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

我知道 我想要 我学到

姓名: _____

关于创意程设思维及Scratch, 你已经知道了什么? 什么是你接下来想要知道的? 这个活动是一个让你回想在Scratch中哪些地方是你所熟悉适应的(我已经知道的)? 哪些地方是你想要更深入探索的(我想知道的)? 运用你身边各种的资源来调查你像要知道的事物, 并且分享你的发现(我学到的)。

什么是我(已经)知道的?

回顾省思, 你到目前为止的程序设计经验, 写下有关Scratch及创意程设思维你所知道的事物。

什么是我想要知道的?

根据你自己的兴趣, 写下一份清单, 列出: 你接下来想要「知道更多」或是「探索追寻」的事情。

我学到了什么? TIPS &

搜集资源来调查你上面所列出的事物, 并且分享在你的研究过程中学到的事物。

第二回合

建议时间
45-60 MINUTES

目标

完成这项活动，学生将：

- + 有机会新建一个融合自己先前作品(self-remix)的项目或是将活动时间用于一个先前跳过或未完成的单元活动

活动描述

- ❑ 老师可以选择性地，提供从单元0到单元5的讲义来引导学生。
- ❑ 留给学生们自己主导的时间来：
 1. 自我融合：借着融合自己先前的一个程序计画，来重新构想或延伸出一个新建的程序计画。
 2. 重新回顾并着手进行一个先前跳过或未完成的单元活动。
- ❑ 鼓励学生们与他人分享自我融合或单元活动的成果。我们建议举办「双人结对分享(pair-share)」或「设计展示(design demo)」活动。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 单元 0-5 的讲义

反思问题

- + 为什么你挑选那个项目或单元活动来做呢？
- + 如果有更多时间，你会做什么？

审视学生作品

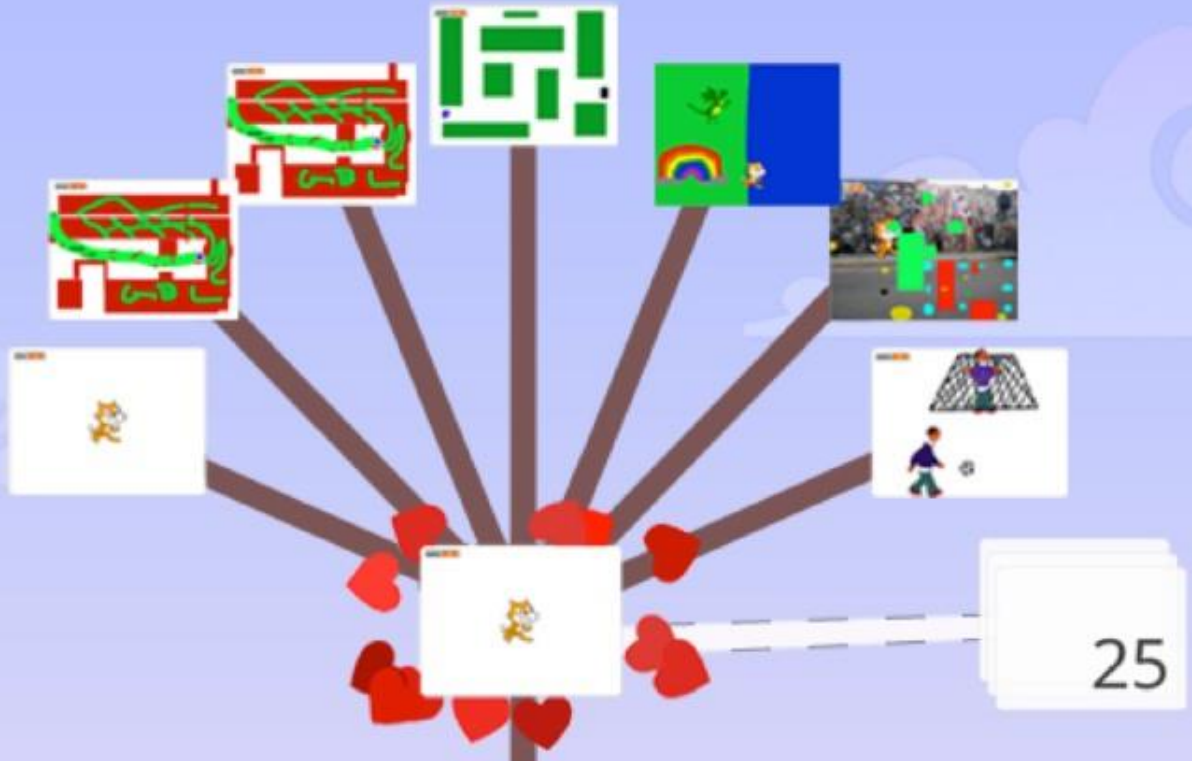
- + 学生们是否新建一个融合自己先前作品(self-remix)的项目，或者着手进行一些单元活动呢？
- + 有关于你的学生感兴趣的事物，你学到了什么？
- + 你的学生们可能需要哪些更多的协助呢？

附注

- + 邀请学生回顾他们的设计手札以及Scratch个人数据来回想先前的作品与单元活动。
- + 鼓励学生们回顾他们在单元1我的工作室中搜集的灵感来激发构想。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



25

进阶概念

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 借着新建一个运用影像侦测或分身技巧的程序计划，更加熟悉程序设计概念(条件 conditionals、运算符 operators、资料 data)与实际操作(试验 experimenting 与重复 iterating、测试 testing 与除错 debugging、再利用 reusing 与融合 remixing、抽象化 abstracting 与模块化 modularizing)。

活动描述

- ❑ 使用进阶概念工作室、影像侦测范例工作室、分身范例工作室来展示相关范例，并且协助学生熟悉控制影像侦测(video sensing)和分身(cloning)的程序积木。老师可以选择性地，提供进阶概念讲义来引导学生。
- ❑ 留下时间给学生们探索范例程序的程序代码来新建程序项目，并试用一个或更多的进阶概念(影像侦测、分身技巧)。
- ❑ 鼓励学生与他人分享他探索的心得。我们建议举办「设计展示(design demo)」活动。老师可以选择性地让学生将他的程序计划加入到进阶概念工作室或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 进阶概念工作室(Advanced Concepts studio)
<http://scratch.mit.edu/studios/221311>
- ❑ 影像侦测(video sensing)讲义
- ❑ 影像侦测范例工作室
<http://scratch.mit.edu/studios/201435>
- ❑ 分身(cloning)讲义
- ❑ 分身范例工作室
<http://scratch.mit.edu/studios/201437>

反思问题

- + 你选择要探索的进阶概念是什么？
- + 对于你所挑选的概念，深入学习更多的方法是什么？

审视学生作品

- + 程序计划探索运用了一个或更多的进阶概念吗？

附注

- + 选择探索影像侦测功能的学生将会需要配有网络摄影机的计算机。
- + 提醒学生可以使用背包(backpack)工具来引用和融合范例程序中的程序代码。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



影像侦测

你可以如何将影像侦测的技巧运用在你的Scratch程序计划呢？

你知道你可以让Scratch程序透过网络摄影机产生互动性吗？借着新建一个运用了影像侦测特色的程式计划来探索这个进阶概念。



动手开始吧

- 开启或新建一个Scratch计划来加入影像侦测功能。
- 试用看看在「侦测(sensing)」分类中用来侦测影像的程序积木。
- 试用「video on」、「将视讯设定为开启(turn video on)」、「设定视讯透明度为(set video transparency to)」程序积木来让你的Scratch程序能够侦测影像移动。



尝试看看

- 确认你的网络摄影机连上了！使用「将视讯设定为开启」积木来测试。
- 如果你感觉有一些困惑，没关系！探索一些在影像侦测案例工作室中的程序计划来看看影像积木是如何运用的，或使用技巧窗口来学习更多有关影像侦测的功能。

都完成了？

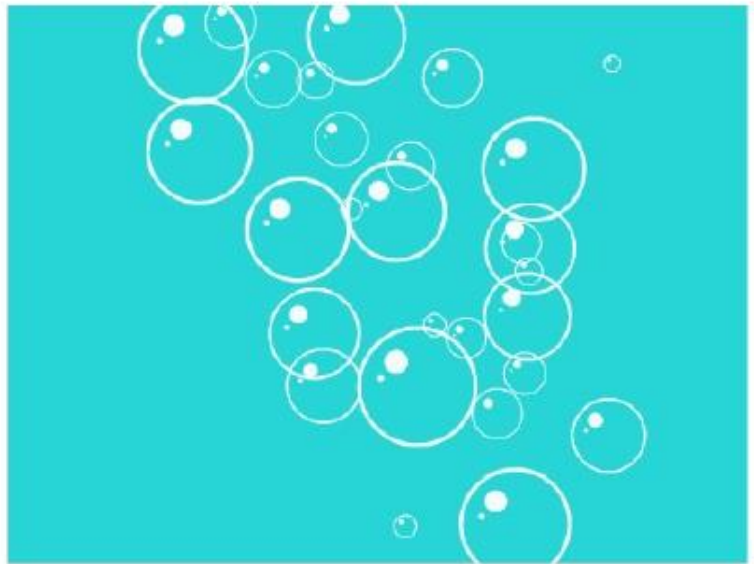
- + 将你的程序计划加到进阶概念工作室：<http://scratch.mit.edu/studios/221311>
- + 将影像侦测技巧加到一个你之前的程序计划！
- + 帮助一位邻座的同学！
- + 融合一个在影像侦测工作室中的程式计划。

分身

你可以如何将分身的技巧运用在你的Scratch程序计划呢？

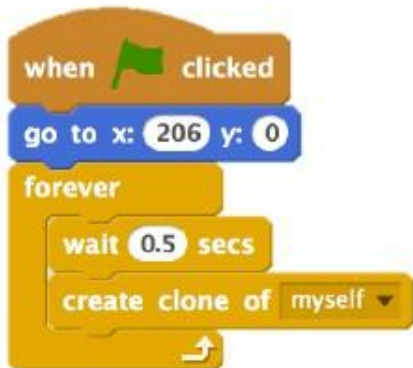
分身技巧是一种简易的方式来创造多个同样的角色。你可以运用分身技巧在程序计划中，创造许多对象以及酷炫的效果。

借着新建一个运用分身特色的程式计划来探索这个Scratch进阶概念。



动手开始吧

- 开启或新建一个Scratch计划来试验分身功能。
- 试用看看在「控制(control)」分类中用来制作分身的程序积木。
- 试用这些积木来制造你的角色的分身。设定角色分身的行为。



THINGS TO TRY 尝试看看

- 如果你没有看见角色的分身，检查看看原本的角色是不是在相同的位置 — 原本的角色可能挡住了分身！设定你原本的角色或分身移动或滑行到不同位置，这样你就可以看见他们。
- 如果你感觉有一些困惑，没关系！探索一些在分身工作室中的程式计划来看看如何运用分身功能，或使用技巧窗口来学习更多有关「(create clone of)」和「当分身产生(when I start as a clone)」积木的使用方法。

都完成了？ FINISHED?

- + 将你的程序计划加入到分身范例工作室：
<http://scratch.mit.edu/studios/201437>
- + 将分身技巧加到一个你之前的程序计划！
- + 帮助一位邻座的同学！
- + 融合一个在分身工作室中的程序计划。

硬件 与延伸套件



目标

完成这项活动，学生将：

- + 对于能将Scratch的数字世界链接到现实世界的硬件套件有所认识

活动描述

- ❑ (重要事项：这个活动需要一种或更多的硬件套件产品。) 向学生介绍Scratch可以连结到其他技术与硬件扩充套件，包含有：LEGO WeDo、MaKey MaKey、PicoBoard。老师可以选择性地从「How can I connect Scratch with other technologies?」影片清单中播放一些范例。
- ❑ 将学生2至4人分成一组。留给每个小组时间运用一种或多种可用的硬件扩充套件，来结合新建的Scratch程序计划与实体世界。
- ❑ 鼓励每个小组与他人分享他们的作品。我们建议举行「成果展(gallery walk)」或回馈博览会(feedback fair)」。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ LEGO WeDo construction set
<http://bit.ly/LEGOWeDo>
- ❑ MaKey MaKey
<http://makeymakey.com>
- ❑ PicoBoard
<https://www.sparkfun.com/products/10311>
- ❑ How can I connect Scratch with other technologies? videos
<http://bit.ly/hardwareandextensions>

反思问题

- + 你探索试验的硬件套件是什么？
- + 你是如何融合数字和实体世界？
- + 有什么困难呢？
- + 有什么令你惊奇的事情吗？

审视学生作品

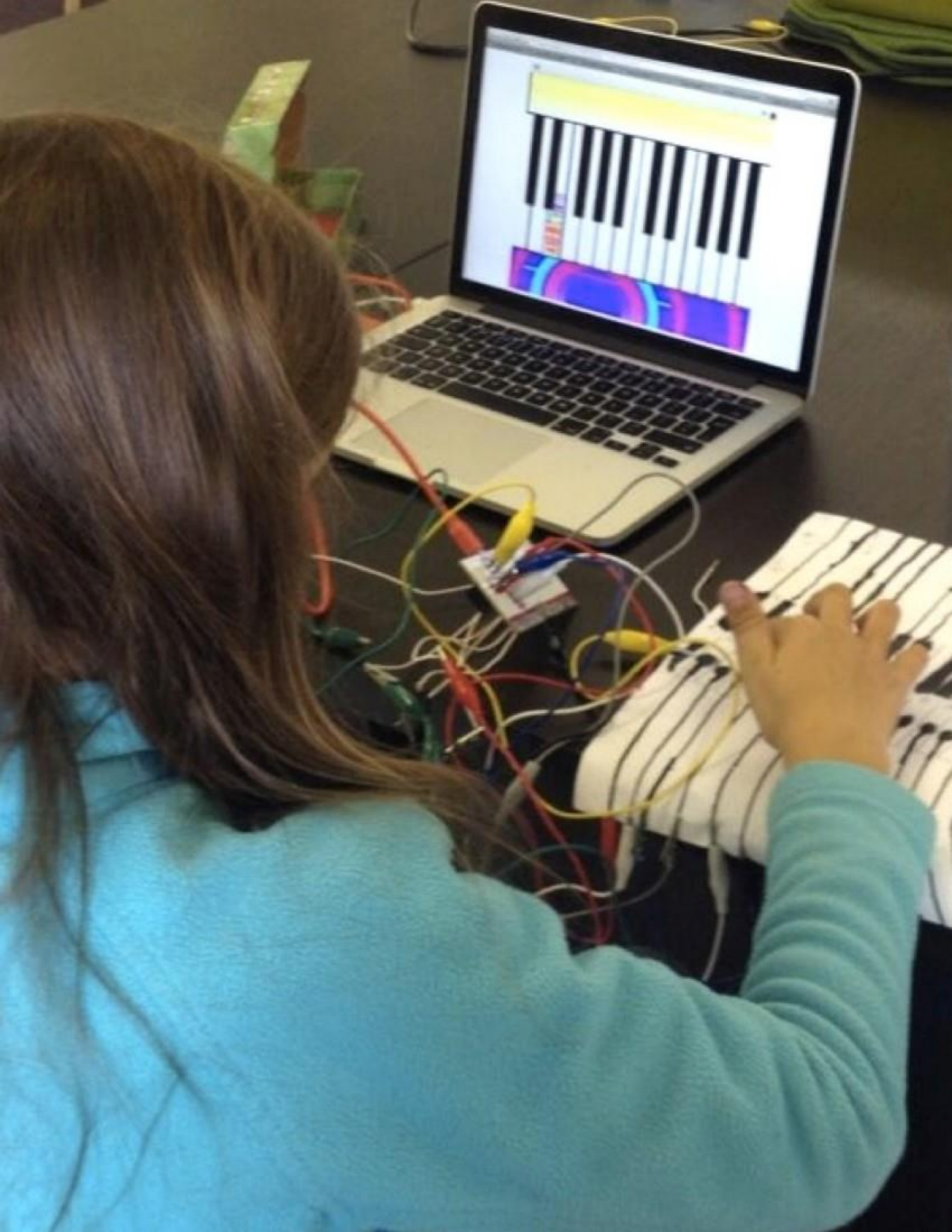
- + 学生的作品是否运用了数字概念和实体套件呢？

附注

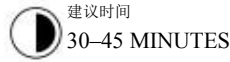
- + 将所有人集合起来，一起进行这个活动！挑战学生们运用 LEGO WeDo 套件及Scratch将他们的程序计划链接在一起创造出连锁(骨牌)效应，就如同一部「鲁布歌德堡机械(Rube Goldberg machine)」一样。观看影片范例：
<http://bit.ly/ScratchChainReaction>。
- + 在Scratch编辑画面中，点击「更多积木」分类下的「加入扩充功能」，进一步来启用可以控制硬件套件的Scratch程序积木。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



活动设计



目标

完成这项活动，学生将：

- + 设计一项活动或教学资源来协助他人学习更多有关Scratch的知识与程序设计的创意

活动描述

- ❑ 让学生体验当老师教学是什么样的感觉！挑战学生去创造、融合或重新构想一套用来协助他人探索创意程设思维的教学活动或资源。老师可以选择性地，提供活动设计讲义作为额外的辅助材料。
- ❑ 协助学生想象与构想不同种类的创意学习体验。老师可以选择性地，回顾这本引导手册中的范例程序与活动，或者，鼓励学生探索Scratch教学图卡与Scratch设计工作室名单来找寻灵感。接着，留给学生们时间来设计他们自己的学习活动或教学资源。
- ❑ 提供机会给学生与学习者一起测试他们的教学活动与资源。鼓励他们与家人或朋友分享他们设计的教学活动或资源，或是邀请学生们成为其他班级、社团或活动的小老师(peer mentor)。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作设计的过程。

教学资源

- ❑ 活动设计讲义
- ❑ Scratch教学图卡
<http://scratch.mit.edu/help/cards> (英文版)
<https://goo.gl/XRyh3D> (繁体中文版)
- ❑ Scratch设计工作室名单(Scratch Design Studio list)
<http://scratch.mit.edu/users/ScratchDesignStudio/>

反思问题

- + 你设想让谁(何种对象)来使用你的活动或教学资源？
- + 你希望人们可以藉由使用你的活动或教学资源学到些什么？
- + 学习者在进行活动或使用教学资源的过程中可能会面临哪些的挑战？你可能会如何进一步协助他们来面对这些挑战呢？

审视学生作品

- + 活动或教学资源是否建构了切入创意程设思维的一种导引或探索模式呢？你能够提供什么样的回馈建议给学生呢？

附注

- + 对于协助他人学习特别感兴趣的学生，可以成为小老师的最佳候选人，鼓励他们在课堂中或是课后及午休社团中担任小老师(peer mentor)。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

活动设计

姓名： _____

你能够如何协助他人学习更多有关Scratch及创意程设思维的知识呢？设计一个能够帮助其他人学习Scratch的活动吧。这个活动可以是不需计算机操作的(像是：打造生物)、程序计划构想(像是：组乐团)或是挑战活动(像是：除错Debug!)。你也可以规划全新类型的活动或讲义！运用下面的问题来进行脑力激荡，在「活动与讲义规划表」来制定细节。

活动是为谁设计的？

谁是你的学习对象呢？你想要帮助哪些人学习更多关于Scratch及创意程设思维的知识呢？

他们将会学到什么？

学习目标有哪些？你希望别人从使用你的学习活动中可以学到什么新的东西？

他们需要些什么？

TIPS & TRICKS

学习者将会需要哪些补充物品(教材、教具等)？什么样类型的协助将可以帮助学习者成功地融入你的教学活动之中？

我的活动

(标题)

建议时间
__-__ MINUTES

学习目标

(2 个学习目标)

完成这项活动，学生将：

+

+

活动描述

(活动计划指引说明)

□ 学习者将会创作些什么？他们要如何达成这些目标呢？

□ 学习者将如何与他人分享他的作品？

□ 学习者将会如何反思他们的设计呢？

教学资源

(2 个计划资源 - 工作室、讲义等)

□

□

反思问题

(3 个反思问题)

+

+

+

审视学生作品

(2 种确认学习者是否完成活动的方法)

+

+

附注

(技巧与提示)

+

+

+

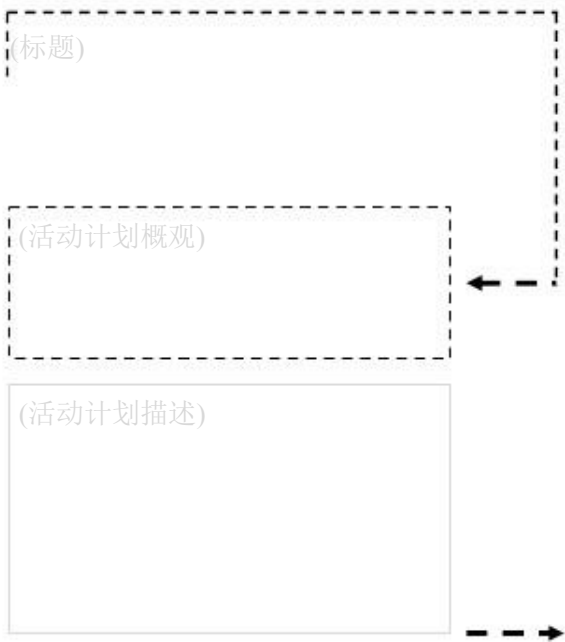
我的笔记

□

□

□

□



动手开始吧

(活动计划指引说明)

-
-
-
-

(活动计划指引说明的插图)

尝试看看 都完成了？

(如果遇到困难，可以做的3件事情)

-
-
-

(如果有多余的时间，可以进行的3件补充事项)

- +
- +
- +

我的除错 (Debug)谜题

建议时间
15-30 分钟

目标

完成这项活动，学生将：

- + 借着设计一个除错谜题的挑战，更加熟悉程序设计的实际操作(试验experimenting与重复iterating、测试 testing与除错debugging、再利用reusing与融合remixing、抽象化abstracting 与模块化 modularizing)。

活动描述

- ❑ 老师可以选择性地，提供「我的除错谜题讲义」在活动过程中来引导学生。
- ❑ 给予学生机会来创作给别人来解决的除错谜题程序。程序错误可以锁定在一个特定的概念、Scratch程序积木、互动行为或是某种其他的程序设计挑战。鼓励学生从过去自己的经验中获得灵感，特别是在创作Scratch程序计划时卡住与脱困的经验。
- ❑ 让学生们交换并且测试及解开别人的除错谜题。老师可以选择性地，邀请学生将他们的除错谜题加入到「我的除错谜题工作室」或是班级专属的工作室。
- ❑ 要求学生借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们创作错误问题的过程。

教学资源

- ❑ 我的除错谜题讲义
- ❑ 我的除错谜题工作室(My Debug It! studio)
<http://scratch.mit.edu/studios/475637>

反思问题

- + 程序的错误(毛病)是什么？
- + 你的灵感从何而来？
- + 你想象中别人会如何来调查并解开这个挑战呢？
- + 别人是否有与你预期中不同的方法来找出并修正那个有错的程序吗？他们的方法是什么？

审视学生作品

- + 程序计划中包含了一个有待解决的除错挑战吗？
- + 学生们采用了什么不同的测试(testing)与除错(debugging)方法呢？

附注

- + 提醒学生要在Scratch网站上的程序计划页面中，加入有关除错谜题挑战的描述。
- + 有额外的时间或是需要热身活动？让学生在「我的除错谜题工作室」中，挑选一些别人提供的除错谜题来练习他们发现问题以及解决问题的技巧。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

我的除错 (Debug) 谜题

现在，你可以设计自己的除错程序谜题了。你会创作些什么？

在这个活动中，你将会创作你自己的除错谜题，让别人来研究、解谜与融合。



动手开始吧

- 回想你曾经在创作或除错你的程序计划时，遇到的各种错误。
- 为你可以创造的除错挑战列出一份清单。
 - 。一个除错谜题可以锁定在一个特定的概念、程序积木、互动行为或是某种其他的程序设计挑战。
- 编写你的除错谜题程序项目

我的除错谜题计划

我的
筆記

都完成了？

- _____
- _____
- _____
- _____
- _____

- + 将你的除错谜题加入到「我的除错谜题工作室」：<http://scratch.mit.edu/studios/475637>
- + 与一个邻座同学交换除错谜题，并且尝试解决对方的除错谜题。
- + 帮助一位邻座的同学。
- + 尝试解决在「我的除错谜题工作室」之中的其他除错谜题。

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

单元 6

黑客松

你的进度

本单元包含



专案游说	114
项目规划	116
设计冲刺跑	120
项目回馈	122
项目检查哨	124
非焦点团体	126
成果发表预备	128
最终成果发表	130

单元 6

整体概观

「大处」着眼

在最后这个单元，学生们将藉由融入于设计、打造他们自己所选择的开放式(open-ended)程序项目，进一步建构创意程设思维的体验。为了协助您与学生获得这种开放式的设计体验，我们受到了启发而将本单元塑造成一场黑客松(hackathon)。黑客松的精神：体现实时的(just-in-time)学习与解决问题的过程、鼓励一次又一次的规划-制作-分享、倡议相互连结与齐心协力的氛围，让它成为最適切创意程设思维的终极体验。

已經下課了，但沒著
有些學生似乎仍忙著
有些察覺。仍忙著
為他們的#scratch
遊戲除錯。團隊
努力的成果。
@Sheena1010



学习目标

学生们将：

- + 对于黑客松(hackathon)活动的形式有所认识
- + 藉由定义、开发并简报一个自我主导、有个人意义的程序计划，来展现程序设计概念(次序—sequence, 循环—loops, 事件—events, 平行—parallelism, 条件—conditionals, 运算符—operators, 数据—data)与实际操作(试验experimenting与重复iterating、测试 testing与除错debugging、再利用reusing与融合remixing、抽象化abstracting与模块化modularizing)的相关知识。
- + 有许多不一样的机会来：与团队协力合作、训练分享(表达)技巧、获得并给予多次的回馈意见。

关键词、概念与实作

- + 黑客松(hackathon)
- + 项目游说+ 成果发表(project pitch)(showcase)
- + 设计冲刺跑(design sprint)
- + 非焦点团体(unfocus group)

附注

- + 本单元可运用于「个人独立」或「小组合作」的程序计划。老师可选择其中之一或让学生们选择。

黑客松是什么？

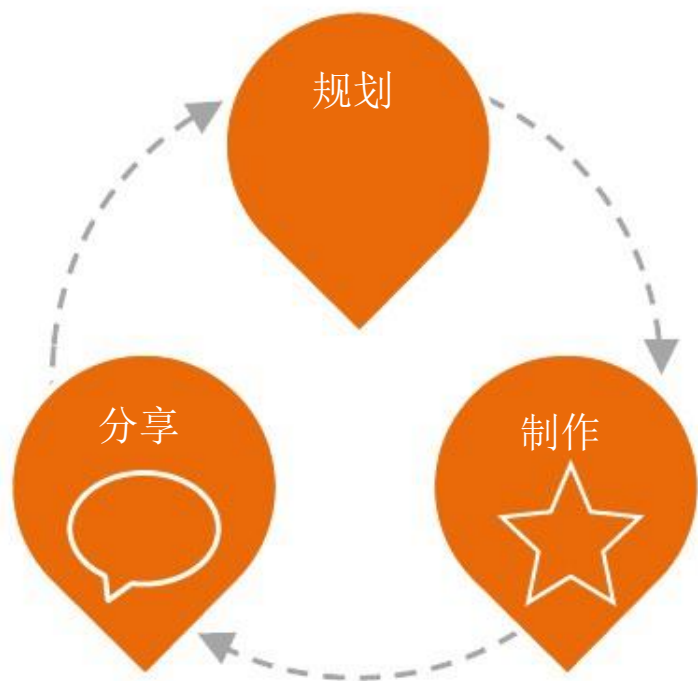
黑客(hack)对于有些人而言是个负面的词汇，然而长期以来它也代表了：趣味性(playfulness)、好奇心(curiosity)、持之以恒(persistence)以及创造力(creativity)。我们最喜爱的一种解释将黑客定义为：「原创力的合宜运用」。根据这个定义，对于年轻的学习者来说，比起如何「成为一个黑客」，有什么更好的意涵呢？

黑客松活动本身是设定在密集专注(编按：专心做好几件事情，而非很多事情)与有限时间的情境之中，在这样的条件下，黑客松(hackathon)萃取了黑客的趣味与原创性。在本单元，学习者将运用一次又一次「规划-制作-分享」的循环，来激荡构想、开发程序项目、并且发表最终成果雏型。

黑客松为学习者提供了一个绝佳的机会来着手进行与他们自身相关且有意义的程序项目。这个程序专案可以是一个人的独立创作也可以是团队合作的成果。黑客松是一个机会让学生们在合作、创意、弹性、有趣的环境下：展现Scratch相关的知识、扩展目前的程序设计技巧、开发并测试自己的构想。

黑客松如何进行呢？

在黑客松的过程中，学生们将融入在一次又一次的循环：规划(PLAN)—制作(MAKE)—分享(SHARE)。这样的循环能够有意义地激励学生们融入于：创意发想、动手创作、回想沈思等活动。



规划

你想要做些什么？
激荡构想并且准备一个行动计划！



制作

运用资源与他人的协助，
来设计与开发程序作品。



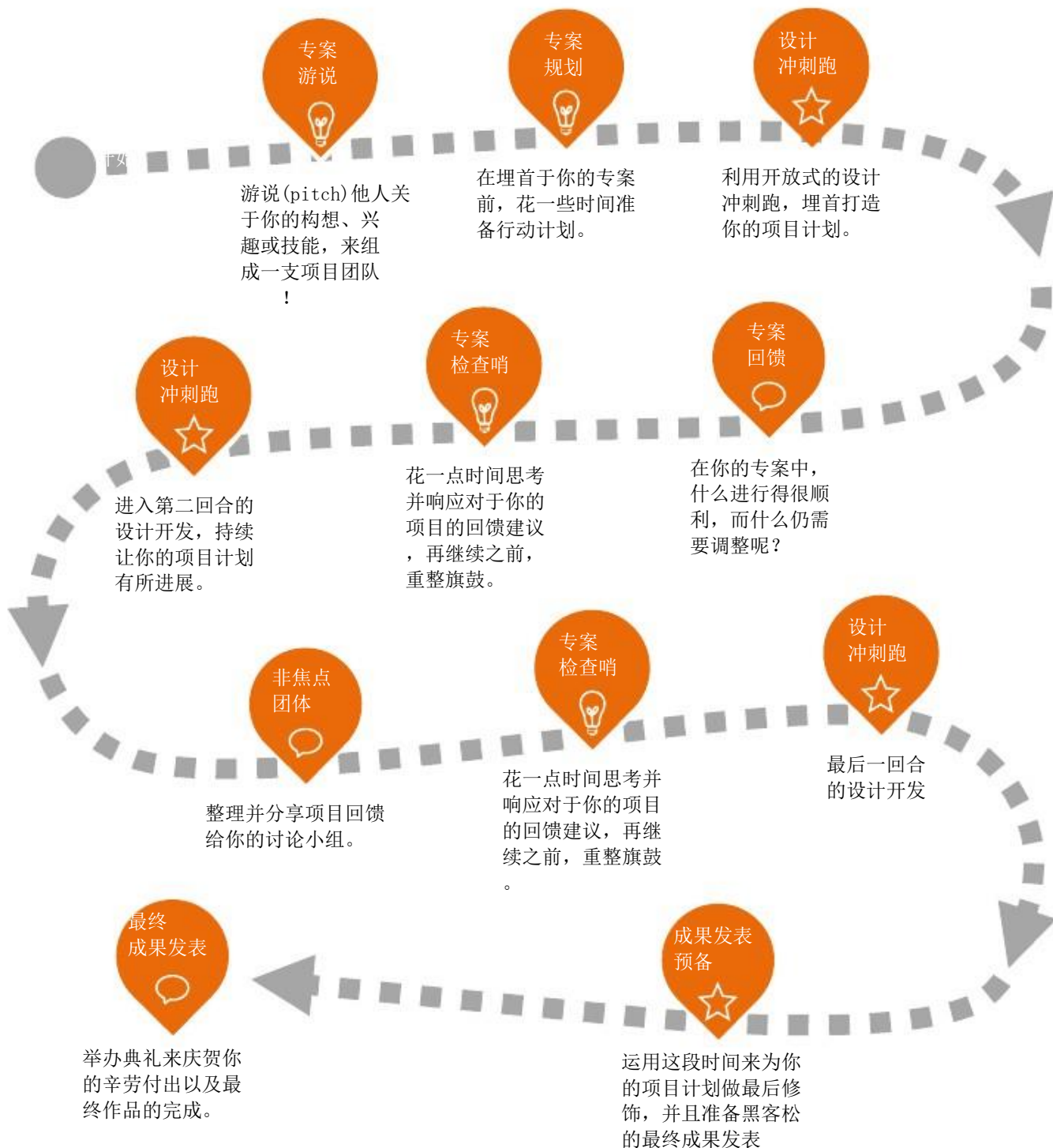
分享

与他人分享你的程序作品
并且收集回馈来进行下一步！

可行的途径

受黑客松启发的单元活动是设计来：挑战学生们在开放式与协力合作的学习场域中，打造更复杂的程式专案。所有我们一直建构的重要精神：「鼓励承担风险(risk-taking)与坚持不懈」、「把失败当做学习良机」、「重视过程而非成果」、「养成愉快合作的文化」，都将在这个单元中极致展现。

为了协助你开始着手进行，根据「规划- 制作- 分享」的循环，我们建议以下的活动次序：





专案 游说 (PITCH)

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 根据个人化的兴趣，激荡项目构想
- + 游说 (pitch) 构想、兴趣、或技能，来组成专案团队。

活动描述

- ❑ 向学生们简介游说 (pitch) 的概念。借着游说的进行，学生们可以发表项目程序的构想来招募团队中的其他成员，或者，学生们可以表现他们的兴趣、技能或天份，好让其他团队来招募他们。
- ❑ 将游说讲义分给学生，保留时间让学生激荡想法并在讲义上回答问题。有些学生可能早已有项目的构想，也可能对于特定的兴趣或技能想要分享或是进一步探索。请让学生们了解，如果他们没有特定的兴趣或构想，没有关系，他们可以加入别人的团队。老师可以选择性地，让学生们回顾在「单元1活动- 我的工作室」所搜集的感兴趣的程序计划来激发灵感。
- ❑ 让每位学生都有机会向团体中的其他人进行游说。要求想要进行游说的人排队，给每人30秒的时间来描述项目构想、兴趣或是技能。
- ❑ 提供时间让学生们3或4人组成一个项目团队。老师可以选择性地，让学生在便利贴上写下姓名与感兴趣的项目，这样就可以在墙壁上进行排序与分配来建构团队。

教学资源

- ❑ 游说 (Pitch) 讲义
- ❑ 便利贴 (选择性，非必要)

反思问题

- + 到目前为止，你最喜爱着手进行的程序计划是什么？
- + 什么类型的程序计划是你接下来有兴趣创作的呢？
- + 什么知识、技能或是天份是你贡献到程序计划的呢？

审视学生作品

- + 是否每位学生都有机会游说他们的构想或是兴趣？
- + 是否每位学生都找到一个项目团队加入？

附注

- + 在所有的Scratch活动中，特别是在黑客松单元里，学生们对彼此提供协助与引导是非常具有价值且重要的。鼓励学生与他人分享他们的知识与技能，不但能减低引导者 (facilitator, 如：老师) 的负担，也能明显地深入强化创作者 (如：学生) 的学习与理解。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

专案 游说

进行项目游说的是： _____

运用以下的问题来激荡你想要在黑客松活动中动手进行打造的构想。
你将有30秒的时间来向其他人游说你的构想、兴趣与技能。

我喜爱的专案

到目前为止，你最喜爱着手进行的程序计划是什么？ 什么原因让这个程序计划特别受到你的注目？

我的黑客松项目构想

什么类型的程序计划是你接下来有兴趣创作的呢？

我的技能与兴趣

什么知识、技能或是天份是你贡献到程序计划的呢？

专案规划

建议时间
30-45 分钟

目标
完成这项活动，学生将：

- + 划定一个具有适当规模且具可行性的专案
- + 拟定完成项目所需活动或任务的要点
- + 产生完成项目所需资源的一份初步清单

活动描述

- ❑ 在项目启动前花一些时间来：探索构想、厘清完成项目所涉及的任务事项、列出什么是已知能(和不能)让项目顺利完成的要素。
- ❑ 将团体以项目团队进行分割。老师可以选择性地，将项目规划讲义和项目素描讲义发给每个团队或每个人。
- ❑ 检视用来规划项目的不同元素：专案素描、任务要点、资源列表、故事板(storyboard)/框架图(wireframes)。给项目团队15分钟来激荡进行项目所需的构想、计划与资源。已经有明确概念与计划的学生们可以开始进行项目设计。
- ❑ 老师可以选择性地，在这个活动结束后回收已经完成的「项目规划」与「项目素描」讲义，并在「设计冲刺跑(design sprint)」阶段开始时，发还给学生们。

教学资源

- ❑ 项目规划讲义
- ❑ 专案素描(sketches)讲义

反思问题

- + 我想要创作的项目是什么？
- + 要打造我的项目，需要准备哪些步骤？
- + 要打造我的项目，哪些资源是我已经拥有的(例如：人、范例程序计划)？
- + 要打造我的项目，哪些资源是我可能需要的(例如：人、范例程序计划)？

审视学生作品

- + 这个项目的规模是否合适这次黑客松活动的可运用时间与可取得资源？
- + 你会如何帮助有需要的学生取得特定资源呢？

附注

- + 虽然规划是有帮助的，但它不是做事情的唯一方式，也不应该花费所有时间与心力在规划上面。不同的学生将会想要(和需要)程度不一的事前规划与直接尝试。此外，在项目的不同阶段也会需要不同的应对方式。应该鼓励与接纳多元不同的设计与开发风格(style)。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



Music

Class Plan
Add
Sounds

Art
Sculpture

Sports

Roller
skating
in
the
park

Garden

Magic

Witch
the
spell

Two player
games

Art

Roller
skating
in
the
park

专案 规划

这个项目计划属于：_____

运用以下的问题，开始来设想：打造你的专案所需的要素。

我的专案

描述你想要创作的程序项目。

列出打造你的程序项目所需的步骤。

我的资源

要打造我的项目，哪些资源是我已经拥有的(例如：人、范例程序计划)？

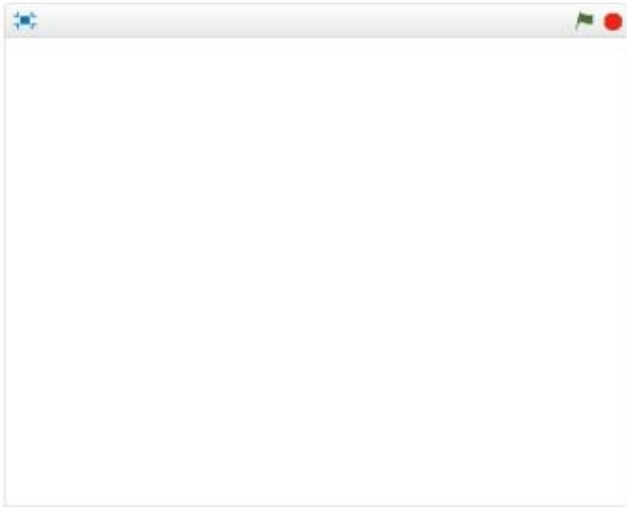
要打造我的项目，哪些资源是我可能需要的(例如：人、范例程序计划)？

专案素描

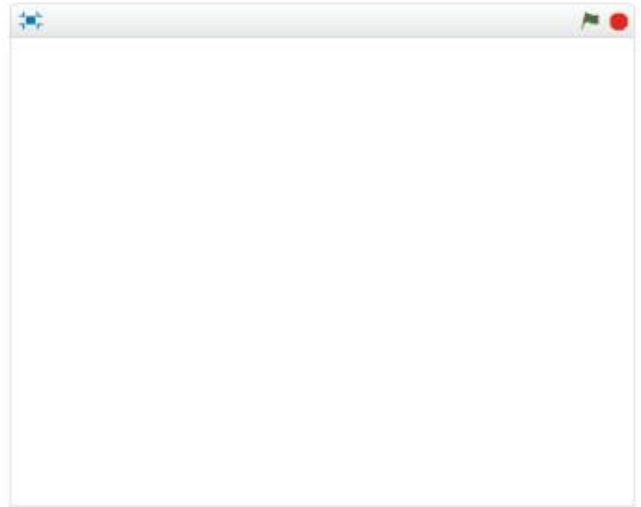
这个项目计划属于： _____

运用下面的空白处画出程序项目外观的素描！

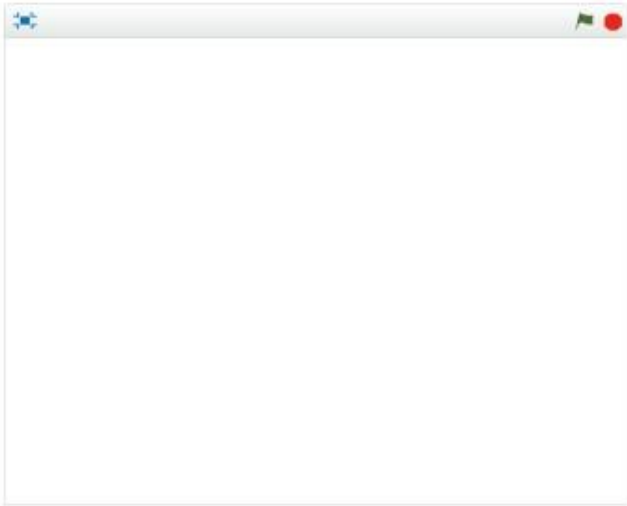
我的专案素描



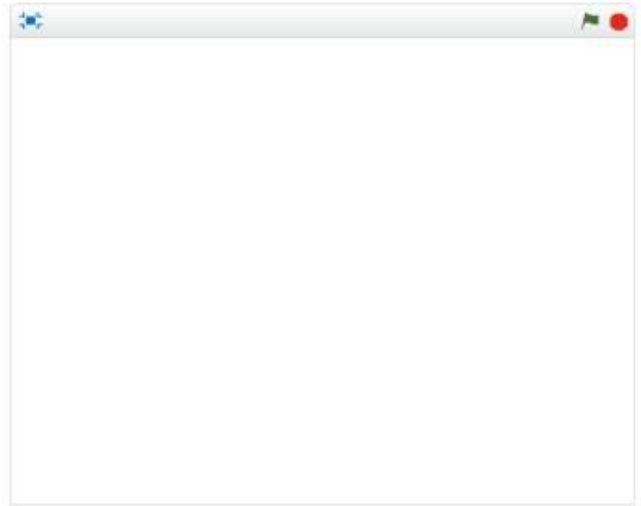
发生了什么事？重要的元素是什么？



发生了什么事？重要的元素是什么？



发生了什么事？重要的元素是什么？



发生了什么事？重要的元素是什么？

设计 冲刺跑



目标

完成这项活动，学生将：

- + 运用程序设计概念与实际操作来进一步打造他们所选择的Scratch项目

活动描述

- ❑ 向学生简介「设计冲刺跑(design sprint)」的概念。设计冲刺跑是指：在一段特定时间内，心无旁骛并集中精神进行项目开发。
- ❑ 要求学生为「设计冲刺跑」订下目标。让学生以项目小组的形式讨论并回答右边的反思问题(并将答案记录在设计手札上)来确立目标。此外，学生也可以运用「项目检查哨(Project Check-In)」活动中的反思问题来确立目标。发给学生们所有已完成的「项目规划讲义」、「项目回馈讲义」、「非焦点团体讲义」来引导学生回头检视最初项目目标，并且鼓励学生们根据建议来规划改善计划。
- ❑ 留给学生们自我主导的时间来着手打造程序计画专案。在需要时，简介与提供额外的辅助资源。在学生们互相帮助之外，汇集一些马上可取用的辅助资源能够协助学生们持续(在项目上)取得进展。在Scratch网站(<http://scratch.mit.edu>)上的程序案例可用来提供灵感，额外的资源可以在ScratchEd网站(<http://scratched.gse.harvard.edu>)找到。
- ❑ 老师可以选择性地，要求学生将项目草稿发布在班级专属的工作室中。

教学资源

- ❑ 补充资源 (例如：范例程序、讲义、Scratch教学图卡、手作材料)

反思问题

- + 你今天会着手开发程序项目的哪个部份呢？
- + 要让项目有所进展，你可能会需要什么协助呢？

审视学生作品

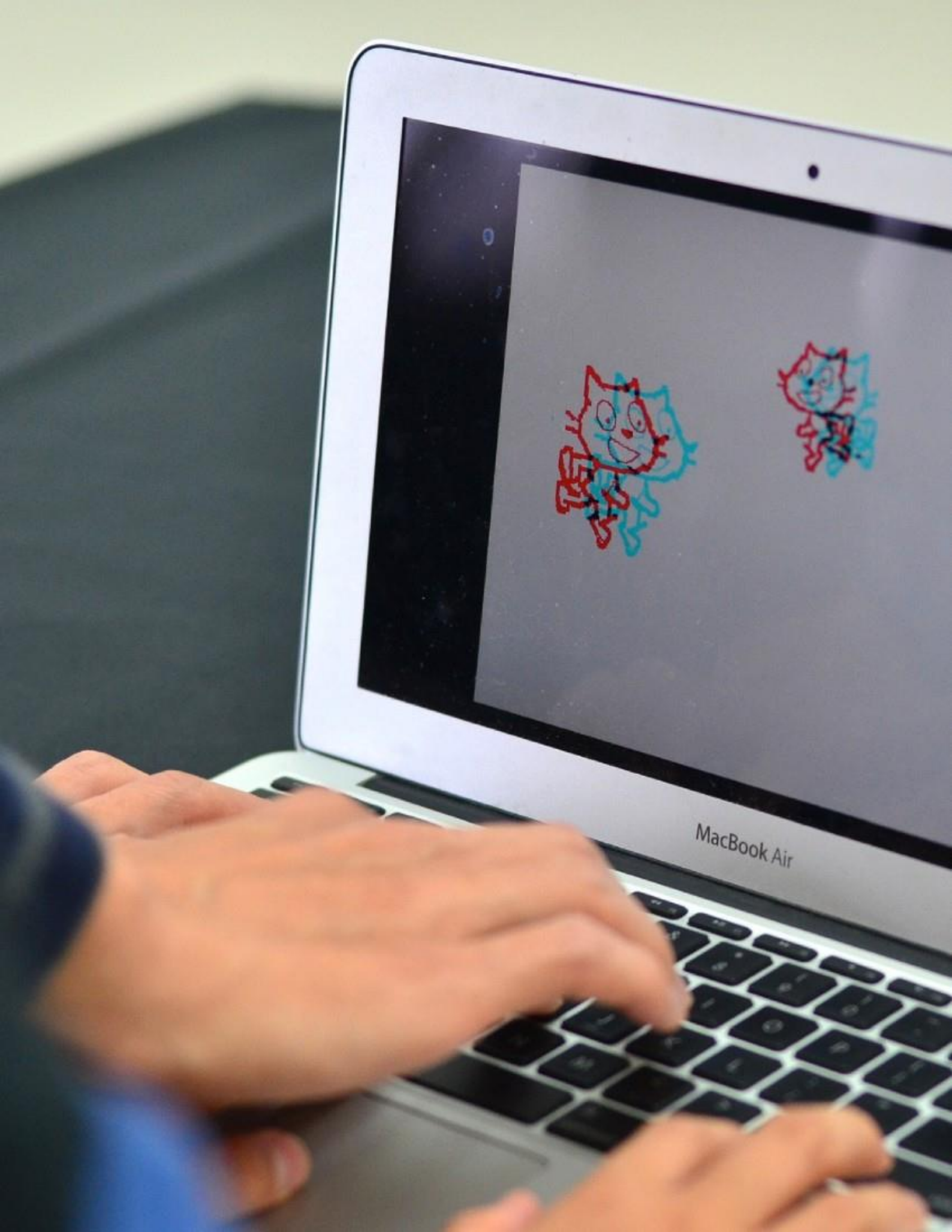
- + 学生个人或团体在程序项目上是否有合理的进度呢？
- + 对于学生的程序项目，你有什么回馈建议呢？

附注

- + 所有的设计活动都是有限制的，像是：给定期间的可用时间、可用资源、自身的能力，因此可能需要作出妥协。这个开放式的活动是一个绝佳的机会来与学生们对谈关于他们项目的关键部份。这个项目最重要的层面是什么？哪些部份在剩下的时间内是可以合理完成的？

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____



MacBook Air

专案 回馈



目标

完成这项活动，学生将：

- + 以小型讨论小组的形式，来给予有关于彼此项目的初步回馈
- + 测试开发中的程序项目
- + 条理化(formulate)并且分享回馈给其他人

活动描述

- ❑ 重新将学生3至4人分成一个回馈小组，让回馈小组的成员与他们所属项目团队的成员不同。老师可以选择性地，让学生回到他们在单元0讨论小组活动时所属的分组。
- ❑ 将项目回馈讲义发给每个人，并回顾讲义上的不同的部分。要求学生在讲义的上方写下自己的姓名以及项目名称。
- ❑ 要求学生花10分钟来：检视回馈小组每一个成员所进行的程序项目，并使用「红、黄、绿(red, yellow, green)」回馈问题来评论每一个项目草案(draft)。当回顾的过程完成时，每位学生都可获得来自每位讨论小组成员对于他(她)所参与程序项目的回馈意见。
- ❑ 在所有回馈回合都结束了之后，保留时间让学生回到各自所属的项目团队，与团队成员共同检视相关的回馈建议，并且讨论哪些建议他们想要纳入项目下一轮的「设计冲刺跑」之中。老师可以选择性地，在这个活动结束的时候回收已经写好的项目回馈讲义，并在下一次「项目检查哨」或「设计冲刺跑」开始时，再发还给学生。

教学资源

- ❑ 项目回馈讲义

反思问题

- + 程序项目的哪些部份是别人可以给予回馈与建议的呢？
- + 哪些建议(如果有)是你下一步规划要加入到程序项目的呢？

审视学生作品

- + 是否每位学生都有机会来给予并获得来自不同来源的回馈呢？
- + 是否每位学生都完成了项目回馈讲义呢？

附注

- + 不同的人可对于开发中的项目(project-in-progress)提供不同的观点。为学习者创造机会，让他们可以获得来自各式各样来源(也包含他们自身)的建议与回馈。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

专案 回馈

这份项目回馈讲义属于： _____

专案名称： _____

红、黄、绿

回馈者	[黄][绿][红] 哪个部分无法执行或可以改进？ 哪个部分不懂或可以换个做法？ 哪个部分做得很好或你最喜欢？		

可以帮助思考程序计划的线索：

- + 明白清楚：你了解程序计划要做什么吗？
- + 设计特色：这个程序计划有哪些特色？程序执行的与想象中一样吗？
- + 操作呈现：这个程序计划的吸引力如何？它是具有互动性、原创性、精巧的、好笑的或是有趣的吗？当你使用时感觉如何？

专案 检查哨



目标

完成这项活动，学生将：

- + 回顾项目进度及意见回馈
- + 拟定完成项目所需活动或任务的要点
- + 产生完成项目所需资源的一份清单

活动描述

- ❑ 在这个活动中，学生们将会进行一次项目的检核。学生们将跟团队伙伴们更新目前的专案进度，并且根据回馈建议，为了即将进行的「设计冲刺跑」列出执行计划的纲要。老师可以选择性地，提供学生或团队「项目检查哨」讲义，以便在活动过程中引导学生。
- ❑ 将群体分为项目团队。老师可以选择性地，发还学生他们已经完成的「项目规划讲义」、「项目回顾讲义」和「非焦点团体讲义」。
- ❑ 保留时间给团队来响应最初的项目目标及已取得的回馈建议。邀请学生为即将进行的「设计冲刺跑」，概要地列出接下来的步骤与改进项目的计划。

教学资源

- ❑ 项目检查哨(Project Check-In)讲义

反思问题

- + 到目前为止，你最爱的整个过程的哪一部分？
- + 你的程序项目的什么部份，仍需要制作？
- + 你的程序项目的什么部份，是你接下来将会着手制作的？
- + 为了让项目有所进展，你可能会需要什么协助？

审视学生作品

- + 团队是否有适宜的进展与计划？
- + 当讨论与分担项目的责任时，团队成员是否同心协力、合作无间？

附注

- + 项目检查哨是一个简短的规划活动。我们建议将它作为每一次「设计冲刺跑」开始(前)时的热身活动。

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

专案 检查哨

这份检核表属于： _____

与你的团队成员讨论你的设计进度，
并且根据回馈建议，拟定下一步计划的执行纲要。

项目进度

到目前为止，你最爱的整个过程的哪一个部分？

你的程序项目的什么部份，仍需要制作？

下一步

你的程序项目的什么部份，是你接下来将会着手制作的？

为了让项目有所进展，你可能会需要什么协助？

非焦点 团体

建议时间
30-45 分钟

目标

完成这项活动，学生将：

- + 访谈、观察并且征询他人来搜集对于开发中项目的反馈建议

活动描述

- 举办非焦点团体 (unfocus group) 是我们引用自IDEO的构想。向学生们介绍非焦点团体的概念：藉由非焦点团体，学生们将向一群多样化的人群组合介绍他们正在开发的项目，并请求这群人给予反馈建议。
- 老师可以选择性地，发给每位学生「非焦点团体」讲义。
- 协助学生激荡想法来找出可能的非焦点团体参与者。鼓励学生们除了考虑他们的目标客群 (target audience)，也同时考虑可以提供独特观点或有趣建议的非寻常或非预期 (unusual, unexpected) 使用者 (例如：家长、老师、兄弟姐妹、其他学生、小区成员)。
- 保留时间让学生去认定 (identify)、访谈 (interview)、观察 (observe) 并且录下 (record) 来自两位非焦点团体成员的反馈建议。
- 给予学生们时间与他们项目团队的成员分享从不同非焦点团体成员所搜集的建议。老师可以选择性地，在这个活动结束后，收集已经写好的「非焦点团体」讲义，并在「项目检查哨」或「设计冲刺跑」开始时再发还给学生。

教学资源

- 非焦点团体讲义

反思问题

- + 描述你的非焦点团体中的参与者，以及为什么你选择他们。
- + 他们的想法可能会如何影响你的项目呢？

审视学生作品

- + 学生们是否认定 (identify) 并访谈两位非焦点团体中的参与者呢？

附注

- + 协助学生发挥创意去研究与找寻回馈来源。是否有间当地的游戏设计公司可能会有意愿协助呢？项目能够跟来自其他学校的学生分享吗？
- + 如果非焦点团体成员无法在活动时间内进行访谈 (例如：家长、老师、兄弟姐妹、其他学生、小区成员)，可以将这个活动在课外时间举行，或是作为回家作业。

我的笔记

- _____
- _____
- _____
- _____

非焦点 团体

专案名称： _____
访谈者是： _____

在这个活动中，你将会访谈与观察他人来获得对于开发中项目的回馈建议。

认定、指认

- + 什么样的人或许可以给你的项目提供独特的观点呢？
- + 你计划跟哪两位非焦点团体的成员分享你的专案草案呢？他们是谁？

观察

与非焦点团体分享你的专案并且观察他们的反应。

- + 让他们卡住、不懂的是什么？
- + 他们跟项目的互动，是否跟你想象一样呢？
- + 他们是否作出了任何令你惊讶的举动？

访谈SKILLS IN

在观察之后，与你的对象进行访谈，询问他们的相关体验。

- + 从访谈中你获得什么回馈建议？
- + 哪些建议(如果有)是你下一步规划要加入到程式专案的呢？

成果发表 预备



目标

完成这项活动，学生将：

- + 着手制作程序项目的最终草案(drafts)，并准备最终成果发表

活动描述

- ❑ 提醒学生们，他们即将与彼此(及可能的来宾)分享程序计划项目。成果发表不但是一种表扬学生们辛苦工作的方式，也是学生们程序设计经验的体现。向学生们解释这节的活动目的是：给学生们一个时机来将他们开发中的程序项目告一段落(finalize)，并且开始准备向别人发表程序项目的策略方法。
- ❑ 保留时间给学生们制作程序项目，以及准备最终成果发表时要跟大家分享的最终作品。老师可以选择性地，将所有最终作品搜集到一个班级专属的工作以便发表。老师可以选择性地，邀请学生将他们的项目加入到「黑客松」工作室。
- ❑ 将项目反思讲义发给学生并讨论「这是什么？过程如何呢？现在如何呢？」的问题架构作为他们向别人表达项目经历的方式。

教学资源

- ❑ 专案反思(Reflections)讲义
- ❑ 黑客松工作室(Hackathon studio)
<http://scratch.mit.edu/studios/488267>

反思问题

- + 你的程序项目是什么呢？
- + 你用来打造这个程序项目的流程是什么？
- + 接下来，你想创作什么？

审视学生作品

- + 每一个团队或学生个人是否写好了项目反思讲义？

附注

- + 对于要完成程序项目，学生们可能会感到焦虑或有压力。这是一个机会来提醒学生：(1) 这次的活动经历，只是他们身为程序设计创作者旅程上的一站；(2) 某些形式的压力是好的，它可以帮助我们专注于目标并且把任务完成！

我的笔记

- ❑ _____
- ❑ _____
- ❑ _____
- ❑ _____

专案 反思问题

这份项目的反思问题属于： _____

运用下面的问题来回想你的设计过程。

这是什么？

你的程序项目是什么呢？
项目是如何运作(执行)的呢？你是如何想到这个构想的呢？

过程如何呢？

你用来打造这个程序项目的流程是什么？
项目过程中，有什么有趣、有挑战性又令人惊奇的事情吗？为什么？
你学到了什么？

现在如何呢？

有关你的项目程序计划，你最引以为傲的是什么？
你会改变些什么？

接下來，
你想創作
什麼？



最终 成果发表

建议时间
45-60 分钟

目标

完成这项活动，学生将：

- + 与他人分享他们项目的最终成果，并反思回想整体的设计过程，以及程序设计创作的体验。

活动描述

- 在活动空间中营造庆典的气氛：邀请来宾、播放音乐、悬挂装饰品、以及/或提供点心。
- 可以选择性地运用投影机与屏幕来显示学生们的作品。
- 邀请学生们与他人分享他们的程序项目并且讨论他们的设计创作过程。老师可以选择性地，提供学生们的设计笔记或先前的程序图画，来展现学生们的进步。
- 保留时间给学生们，借着回答设计手札中的反思问题或在小组讨论中，进一步回想他们所有的程序创作设计的经验。

教学资源

- 用来简报的投影机与屏幕(选择性，非必要)

反思问题

- + 从头到尾检视一次你的设计手札。你都是记录哪一种类型的笔记呢？
- + 哪些笔记是最有帮助的呢？
- + 到目前为止，你最喜爱动手创作的Scratch专案是什么呢？为什么它是你的最爱？
- + 接下来，你想创作什么？

审视学生作品

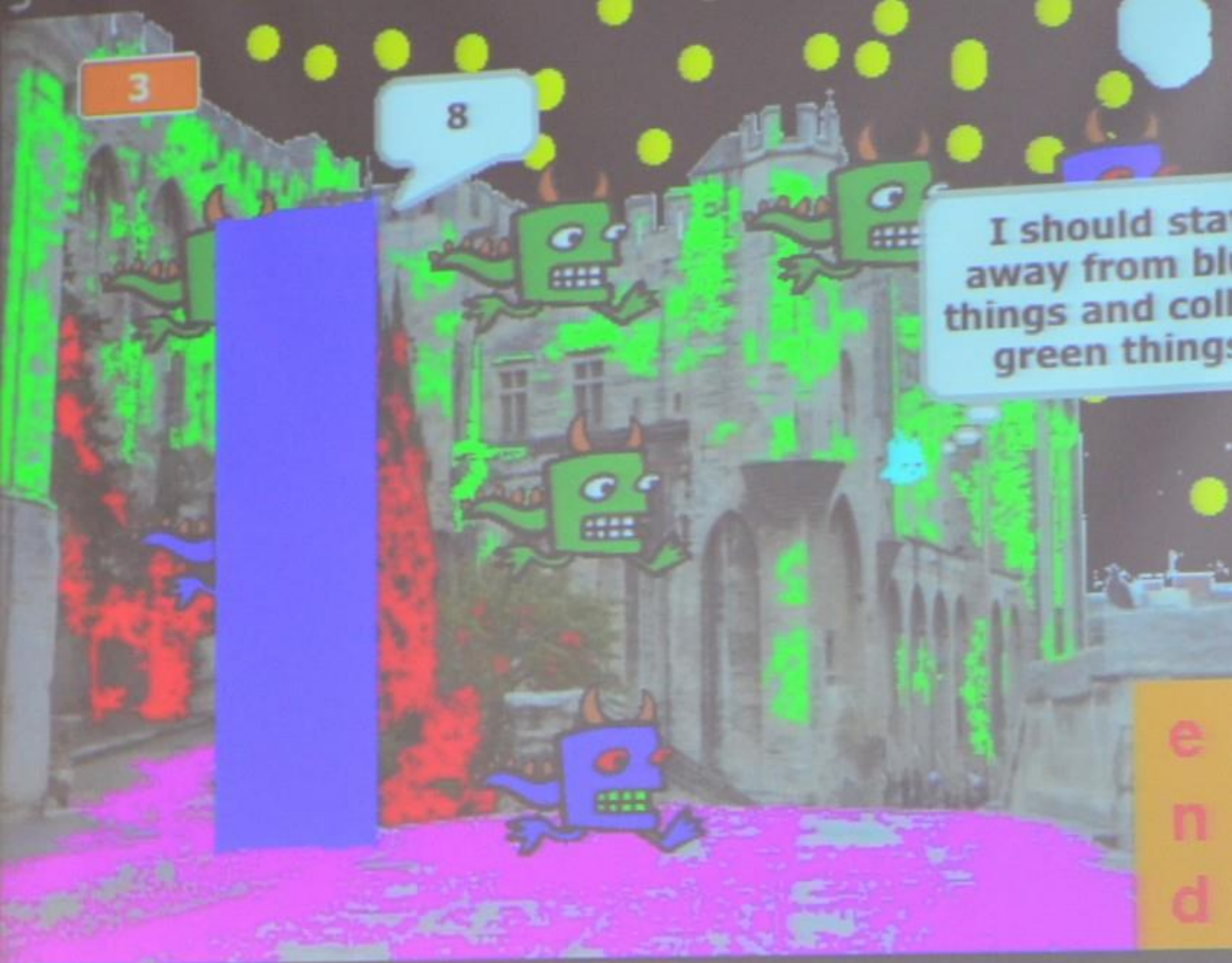
- + 每个学生团队或个人是否都有机会分享他的作品并受到庆贺鼓励呢？

附注

- + 分享发表可用许多形式进行：个人向整个群众简报、将学生分成几群同时进行简报、即时现场展示(live demos)、从网站发表项目等。
- + 学生们在学习过程中产生的：项目作品集(project portfolios)、设计手札、最终项目回馈讲义、项目反思讲义等文件，是许多可用来评估学习成效的文件类型中的一小部分(编按：参阅附录的144页、145页)。

我的笔记

- _____
- _____
- _____
- _____



Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.





创意程设思维

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

声明

原始著作作为 ScratchEd team at the Harvard Graduate School of Education 所拥有，该著作系采用“创用 CC 姓名标示-相同方式分享 4.0 国际授权条款”授权。

简体中文版由以下贡献者进行编辑与翻译改作：

方毓贤 (ys.fang@thinkinviz.com)

黄俊宇 (phkid.co@gmail.com)

John Fang (johnth.fang@thinkinviz.com)

目前版本：rev.2016-07-28.01

最新简体中文版可由www.scratch-tw.strikingly.com下载。

ACKNOWLEDGEMENT

The original work is owned by ScratchEd team at the Harvard Graduate School of Education. The work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

This Chinese derivative of the original work is edited and translated by:

Yu-Hsien Fang (ys.fang@thinkinviz.com)

Fred Wong (phkid.co@gmail.com)

John Fang (johnth.fang@gmail.com)

Current reversion: rev.2016-07-28.01

Latest Chinese derivative is downloadable from www.scratch-tw.strikingly.com.



<http://scratch-tw.strikingly.com/>

附录



词汇对照

教学指引手册中的关键词、概念与实际操作之延伸解释：

可前往：Scratchhelppages(<http://scratch.mit.edu/help>)或社群共创的Scratch Wiki (<http://wiki.scratch.mit.edu>) 来取得更多Scratch特定使用的名词解释。

abstracting and modularizing: The computational practice of exploring connections between the whole and the parts. 抽象化与模块化：程序设计实际操作的一种，意指：探索事物整体与构成组件之间的关联(关系)。

animation: An illusion of continuous motion created by the rapid display of a sequence of still images with incremental differences.

动画：静态影像依照顺序快速播放时，静态影像间微小差异造成了连续动作的错觉。

arcade day: A strategy for sharing student work and whole group activity. Students place their finished projects in Presentation Mode

and then walk around and engage with each other's work. 电玩日：分享学生作品的一种(策略)方法，这也是一种团体活动。学生们将他们完成的程序计划切换至放映模式，然后自由四处走动并且查看试玩他人的作品。

backdrop: One out of possibly many frames, or backgrounds, of the Stage. 背景：舞台众多可能的帧、背景的其中之一。

backpack: A Scratch feature that can be used to conveniently transfer media and/or scripts between projects. 背包：Scratch的特色

功能之一，让用户可以便利地在程序计划之间移转媒体与/或程序代码。

bitmap: An image that is defined by a two-dimensional array (grid) of discrete color values (a.k.a. "pixels"). Contrast with vector

graphics. 位图：由离散色彩值(又称为：像素)组成的二维数组(网格)所定义的影像。与向量图有所对比。

broadcast: A message that is sent through the Scratch program, activating receiving scripts. 广播：可传送到Scratch程序内部所有地方的讯息，可以启动接收端的程序代码。

cloning: A Scratch feature that allows a sprite to create duplicates of itself while the project is running. 制作分身：Scratch的特色

功能之一，当程序正在执行时，允许角色可以创造自己本身的复制品。

computational concepts: The concepts designers engage with as they program, such as sequence, loops, conditionals, events, parallelism, operators, and data. 程序设计概念：程序设计者融入在编写程序时的概念，比如像是：次序

、循环、条件、事件、平行、运算符与资料。

computational perspectives: The broader perspectives that designers may form about world around them through computing – such as expressing themselves, connecting with others, and posing questions about technology's role in the world. 程序设计观

点：藉由程序设计，程序设计者对于他们周遭真实世界可能会形成更广阔的观点，比如像是：表达他们自己、与他人产生连结、并对于科技在世界上扮演的角色提出疑问。

computational practices: The distinctive habits of mind that programmers develop as they work, such as experimenting and iterating, testing and debugging, remixing and reusing work, and abstracting and modularizing. 程序设计实际操作：当程式设计者实际工作时呈现的独特心智习惯，比如像是：试验与重复、测试与除错、融合与再利用作品、抽象化与模块化。

conditionals: The computational concept of making decisions based on conditions (e.g., current variable values). **条件:** 根据状况，像是目前的变量值，做决定的程序设计概念。

control: One of the ten categories of Scratch blocks. They are color-coded gold, and are used to control scripts. **控制:** 十种Scratch

程序积木中的一种。它们的颜色编码是金黄色，用来控制程序码。

costume: One out of possibly many "frames" or alternate appearances of a sprite. A sprite can change its look to any of its costumes.

造型: 角色众多可能的帧或其他外观的其中之一。角色可以改变它的外貌成为它的任何一种造型

critique group: A group of designers who share ideas and test projects-in-progress with one another in order to get feedback on how

to further develop their projects. **讨论小组:** 一个由设计师组成的小型团体，设计师彼此分享点子并测试正在进行的项目，来给予建议并获得回馈，帮助彼此有更进一步的发展。

data: The computational concept of storing, retrieving, and updating values. **资料:** 程序设计概念的一种，用来：储存、读取并更新数值。

design demo: An activity in which students are invited to present their work to the class and demonstrate how they implemented a

particular block, skill, or design strategy within their project. **设计展示:** 一种活动。邀请学生对班级发表他们的作品并展示他们是如何在程序计划中，实际操作特定的程序积木、技巧或设计方法。

design sprint: A specified amount of time dedicated to working intensely on developing projects. **设计冲刺跑:** 在一段特定时间内，心无旁骛并集中精神进行项目开发。

events: The computational concept of one thing causing another thing to happen. **事件:** 程序设计概念的一种，意指：引发另一件事情发生的一件事情。

experimenting and iterating: The computational practice of developing a little bit, then trying it out, then developing some more. **试验与重复:** 程序设计实际操作的一种，意指：编写一小段程序代码，然后试试看是否成功，然后再编写更多其他程序代码。

feedback fair: A sharing activity in which half of your students stay in their seats with their projects open while the other half walks

around exploring projects, asking questions, and giving feedback. Once complete, the students then switch sides and start the process

over. **回馈博览会:** 分享活动的一种。先让一半的学生坐在位置上并将他们的程序计划开启。另一半的学生则四处走动并且探索程序计划、提出问题、给予建议回馈。完成一段时间后，两方交换并再次进行上述程序

gallery walk: A sharing activity in which students put their projects in presentation mode and then walk around and explore each

other's projects. **成果展:** 分享活动的一种。让学生们把自己的程序计划切换到「放映模式(presentation mode)」然后让学生们四处走动来探索别人的程序计划。

hardware and extensions: Supplemental materials that connect the digital world of Scratch with the physical world. Examples of hardware extensions include: LEGO WeDo, PicoBoard, and MaKey MaKey. **硬件与延伸套件:** 可以结合Scratch数字世界与真实世界的补充教材。硬件扩充套件的范例有：LEGO WeDo、PicoBoard、MaKey MaKey。

interactive collage: A Scratch project that incorporates a variety of clickable sprites. **交互式剪贴簿:** 整合了多种可点击角色的Scratch程序计划。

looks: One of the ten categories of Scratch blocks. They are color-coded purple, and are used to control a sprite's appearance. **外观:**

十种Scratch程序积木中的一种。它们的颜色编码是紫色，用来控制角色的样貌。

loops: The computational concept of running the same sequence multiple times. 循环：程序设计概念的一种，意指：多次执行相同的一段次序。

make a block: A feature found within the More Blocks category that allows students to create and define their own custom block or

procedure. 制作一个积木：在更多积木分类中可以找到的一个特色功能，它允许学生能够创造并定义他们自定义的程序积木或程序(编按：函式—function)。

motion: One of the ten categories of Scratch blocks. They are color-coded medium-blue, and are used to control a sprite's movement.

动作：十种Scratch程序积木中的一种。它们的颜色编码是蓝色，用来控制角色的移动。

operators: The computational concept of supporting mathematical and logical expressions. 运算符：程序设计概念的一种，用来：帮助数学运算与逻辑表达。

paint editor: Scratch's built-in image editor. Many Scratchers create their own sprites, costumes, and backdrops using it. 绘图工具

：Scratch内建的影像编辑器。很多Scratch使用者运用它来创作他们专属的角色、造型与背景。
pair programming: A programming methodology in which developers pair up and work side-by-side on a project. 双人结对程序设计：程序设计方法论的一种。意指：开发人员两人一组、并协力制作同一个项目。

parallelism: The computational concept of making things happen at the same time. 平行：程序设计概念的一种，意指：让多个事情在同一时间进行。

pass-it-on story: A Scratch project that is started by a pair of people, and then passed on to two other pairs to extend and reimagine. 故事接龙：由两人一组开始起头的Scratch程序计划，然后将程序传给另外的两人小组，让他们延伸创作并重新构想。

peer interviews: A sharing activity in which students take turns interviewing one another about their processes of reflection, self-

assessment, and research. 同侪访谈：一种分享活动。学生们轮流询问对方有关于：回顾省思、自我评估以及自主研究的过程。

pitch: An activity in which students either announce a project idea in order to recruit other team members, or promote their interests,

skills, and talents in order to be recruited by other teams. 游说：一种活动。在活动中，学生们可以发表项目程式的构想来招募团队中的其他成员，或者，学生们可以表现他们的兴趣、技能或天份，好让其他团队来招募他们。

presentation mode: A display mode in Scratch that allows projects to be viewed at an enlarged size. It is accessed by pressing the button on the top left of the Scratch program. This mode is also called full screen mode or enlarged screen. 放映模式：Scratch的

一种呈现模式。这种模式可以让程序项目以放大的尺寸来检视。按下Scratch主画面左上方的按钮来启动。这个模式也称为全屏幕模式或放大屏幕。

profile page: A page on the Scratch online community dedicated to displaying information about a Scratch user, such as projects they

have created or bookmarked (a.k.a. "favorited"). 个人简介页面：Scratch在线社群中的一个页面，专门用来显示Scratch用户的相关信息，像是：他们自身制作的计划项目或存入我的最爱的计划项目。

project editor: A feature of the Scratch online community that allows projects to be modified. This includes the script area (where

scripts are assembled), the sprite area (where sprites can be manipulated), and the stage area (where sprites are positioned and

where backgrounds can be accessed). 程序计划编辑器：Scratch在线社群中的一个特色功能，他让程序项目可以进行修改。这个编辑器包含了：程序代码/脚本区(组合撰写程序的地方)、角色区(调整角色的地方)、与舞台区(角色就定位的地方，以及可以存取背景的地方)。

red, yellow, green: A reflection and sharing activity in which individuals identify aspects of their projects as not going well or still

needing work (“red”), confusing or contentious (“yellow”), or working well (“green”). 红、黄、绿：一种反思与分享的活动。在活动中每个人判断他们的程序计划的内容细节是属于：做的不好或仍需要改进(红色)、令人困惑或有争议的(黄色)、做的很好(绿色)。

remix: A creative work that is derived from an original work (or from another remix). A remix typically introduces new content or

stylistic elements, while retaining a degree of similarity to the original work. 融合：从某个原始作品(或另一个融合作品)所衍生的创作。融合作品通常会加入新的内容或样式元素，同时保留与原始作品一定的相似程度。

reusing and remixing: The computational practice of making something by building on existing projects or ideas. 再利用与融合：程序设计实际操作的一种，意指：以现有的程序项目或构想为基础来制作其他作品。

Scratch screening: A sharing activity in which students gather around to observe each other’s Scratch projects. 程序放映会：一种分享活动。活动中，学生们会聚在一起观察彼此的Scratch程序项目。

scripts: One or more Scratch blocks connected together to form a sequence. Scripts begin with an event block that responds to input

(e.g., mouse click, broadcast). When triggered, additional blocks connected to the event block are executed one at a time. 程序代码/

脚本：一个或多个Scratch程序积木链接在一起会形成一个次序。程序代码/脚本由一个会响应输入(如：鼠标点击、广播)的事件来启动。当触发时，其他链接到「事件程序积木」的程序积木会一次

sensing: One of ten categories of Scratch blocks. They are color-coded light-blue, and are used to detect different forms of input

(e.g., mouse position) or program state (e.g., sprite position). 侦测：十种Scratch程序积木中的一种。它们的颜色编码是浅蓝色，用来探测不同形式的输入(如：鼠标位置)或程序状态(如：角色位置)。

sequence: The computational concept of identifying a series of steps for a task. 次序：程序设计概念的一种，意指：分析并识别要完成一项任务的一系列先后步骤。

showcase: A strategy for sharing in which students present their final projects to others and reflect on their design processes and

computational creation experiences. 成果发表：一种分享的(策略)方法。学生们向他人展示他们的最终专案成果，并反思回想整体的设计过程以及程序设计创作的体验。

sound: An audio file that can be played in a Scratch project, available by importing from Scratch’s built-in sound library, or creating a

new recording. Sounds are played by using sound blocks, which control a sound’s volume, tempo, and more. 音效：能够在Scratch程序计划中播放的声音档案，可以从Scratch内建的音效范例库或由新的录音取得。音效使用Scratch的音效程序积木播放，音效程序积木可以用来控制音效的音量、节奏等等。

sprite: A media object that performs actions on the stage in a Scratch project. 角色：在Scratch程序计划中的舞台上可以执行动作的媒体对象。

stage: The background of a Scratch project. The stage can have scripts, backdrops (costumes), and sounds, similar to a sprite. 舞台：

Scratch程序计划的背景。舞台可以拥有程序代码/脚本、背景(造型)与音效，与「角色」相似。

studio: A user-created gallery in the Scratch online community that can be used to highlight projects contributed by one or many users.

工作室：在Scratch在线社群中使用者创造的展示空间，可以用来突显由一个或多个使用者贡献的计划程序。

testing and debugging: The computational practice of making sure things work – and finding and solving problems when they arise. **测试与除错:** 程序设计实际操作的一种, 意指: 确认程序可以正确执行, 并且在问题发生时可以找出问题并解决问题。

theatre metaphor: A way of describing the design of Scratch that emphasizes its intentional similarity to theatre, with actors (sprites), costumes, backdrops, scripts, and a stage. **戏剧意象:** 描述Scratch设计理念的一种比喻。Scratch的设计理念有意地与「一出戏剧」相似, 包含了: 演员(角色)、造型、背景、脚本及一座舞台。

tips window: Built directly into the Project Editor, the Tips Window is a form of getting help in Scratch. **技巧窗口:** 技巧窗口是在Scratch寻求帮助的一种形式, 它直接内建在程序计划编辑器中。

unfocus group: An activity in which students share their projects-in-progress and request feedback from a diverse collection of people. **非焦点团体:** 一种活动。活动中, 学生们向一群多样化的人群组合介绍他们正在开发的专案, 并请求这群人给予回馈建议。

variables and lists: A changeable value or collection of values recorded in Scratch's memory. Variables can store one value at a time, while lists can store multiple values. **变量与列表:** 一个或一群记录在Scratch记忆空间中可以进行改变的数值。变量一次可以储存一个数值, 而清单可以储存多个数值。

vector graphic: An image that is defined by a collection of geometric shapes (e.g., circles, rectangles) and colors. Contrast with

bitmap. **向量图:** 由一组几何图形(例如: 圆形、三角形)与颜色所定义的影像。与位图有所对比

video sensing: A Scratch feature that makes use of video from a webcam to detect motion or display video input on the stage. **影像侦测:** Scratch的特色功能之一, 它使用从网络摄影机拍摄到的影片, 进一步来侦测动作或是将影片显示到舞台中。

STANDAR

DS The activities in this guide make connections to several different K-12 curriculum standards, including the Common Core

State Standards, the CSTA K-12 Computer Science Standards, and ISTE NETS. We have included connections to Common

Core Standards as an example.

For more connections, please visit the guide site at

<http://scratched.gse.harvard.edu/guide>

Common Core State Standards for Mathematics 2010

http://www.corestandards.org/wp-content/uploads/Math_Standards.pdf

+ Make sense of problems and persevere in solving them – Many guide activities engage students in solving debugging challenges,

which encourage students to discover different ways of finding and solving problems. Example activity: Unit 1 - 4 Debug It!

+ Reason Abstractly and Quantitatively – Students can express abstract concepts and demonstrate their understandings of quantitative

relationships such as variables through visual representations designed in Scratch. Example activity: Unit 4 Score

+ Model with Mathematics – Certain activities in the guide challenge students to represent previously learned equations, data comparisons, or other mathematical relationships as Scratch programs. Example activity: Unit 4 Interactions

+ Attend to precision – On- and off-screen activities help students recognize the importance of attending to detail when specifying

instructions or a sequence of code intended to elicit a particular outcome. Example activity: Unit 1 Programmed to Dance

+ Look for and Make Use of Structure – Looking through scripts during a debugging challenge, reading through someone else's project

code while remixing a project, or reviewing work to build up more complex programs can engage students in looking closely to discern

Common Core State Standards for English Language Arts/Literacy 2010
repeated patterns or structure within their own or others' Scratch programs. Example activity: Unit 3 Conversations

http://www.corestandards.org/wp-content/uploads/ELA_Standards.pdf
+ They demonstrate independence. – Most activities and projects in the guide are designed to be self-directed or can be easily adjusted to

accommodate independent work, although collaborative projects and group work are encouraged. Example activity: Unit 1 About Me

+ They respond to the varying demands of audience, task, purpose, and discipline. – Students are made aware of varying types of audience, task, purpose, and discipline when sharing projects to the worldwide Scratch online community or designing

projects and activities for others. Example activity: Unit 5 Activity Design

+ They comprehend as well as critique. – A variety of feedback exercises and collaborative projects engage students in sharing works-in-

progress, asking questions, and exchanging constructive critique. Example activity: Unit 0 Critique Group

+ They use technology and digital media strategically and capably. – During self-directed activities, students learn to navigate to different parts of the Scratch website to develop projects, search for inspiration, connect with others, and pursue personal

learning goals. Example activity: Unit 5 Know Want Learn

+ They come to understand other perspectives and cultures. – In remixing others' projects, students need to read, understand, and interpret the code and intention of work that is not their own. When building up collaborative projects, students learn to

cooperate, compromise, and share work with others. Example Activity: Unit 3 Pass It On

程序设计思维 COMPUTATIONAL

THINKING

在过去这几年来中，我们相当着迷使用「程序设计思维」来描述发生在Scratch中的学习与过程。在这个小节中，我们分享：(1) 我们把程序设计思维定义为一组概念、实际操作与观点，(2) 一份问卷，来衡量学生们关于程序设计实际操作的熟练程度，(3) 一份自我省思的问卷，来帮助教师衡量他们自身在课室中协助学生进行程序设计实际练习的表现。

这些定义与问卷工具的发展与构成，有赖于以下等人的同心协力：Wendy Martin、Francisco Cervantes、Bill Tally (以上来自Education Development Center's Center for Children & Technology)与 Mitch Resnick (来自MIT Media Lab)。额外的程序

设计思维资源可由以下网址取得：<http://scratched.gse.harvard.edu/ct>。

程序设计概念 — COMPUTATIONAL CONCEPTS

概念	叙述简介
次序 sequence	分析并识别要完成一项任务的一系列先后步骤
循环 loops	多次执行相同的一段次序
平行 parallelism	让多个事情在同一时间进行
事件 events	引发另一件事情发生的一件事情
条件 conditionals	根据状况做决定
运算符 operators	帮助数学运算与逻辑表达
资料 data	储存、读取并更新数值

程序设计实际操作 — COMPUTATIONAL PRACTICES

实际操作	叙述简介
试验与重复 experimenting and iterating	编写一小段程序代码，然后试试看是否成功，然后再编写更多其他程序代码
测试与除错 testing and debugging	确认程序可以正确执行，并且在问题发生时找出问题并解决问题
再利用与融合 reusing and remixing	以现有的程序项目或构想为基础来制作其他作品
抽象化与模块化 abstracting and modularizing	探索事物整体与构成组件之间的关联(关系)

程序设计观点 — COMPUTATIONAL PERSPECTIVES

观点	叙述简介
expressing 表达	明白程序设计是创作的一种方法(媒介) 「我能够创作。」
connecting 连结	明了与他一起创作和为他人创作的力量 「当我与他人接触时，我能够做到不一样的事情。」
questioning 提问	觉得有能力提问关于世界的问题 「我能够(运用程序设计来)提出问题，来了解这个世界(中关于程序设计的事情)。」

评估程序设计实际练习的发展程度

以下的问卷可用来评估学生们对于程序设计思维相关实际操作的熟练程度。程序设计实际操作包含了：试验与重复、测试与除错、再利用与融合、抽象化与模块化。第一栏标注着给学生的提问(例如：可用于访谈过程或是作设计手札一部份的反思问题)。第二、三、四栏则标注着可能会表现出来的不同的熟练程度，分别有低、中、高三种。

试验与重复	低	中	高
描述你是如何一步一步制作出你的程序计划	学生提供了制作程序计划的基本叙述，但没有关于某一个特定程序计划的细节。	学生以大概的例子来描述特定的程序计划的制作过程与某种先后顺序。	学生明确地描述了特定程序计划中不同部分的细节，并且描述制作那些部分的先后顺序。
当你制作你的程序计划时，尝试了哪些不同的事情？	学生没有说明他/她进行尝试的明确例子。	学生以大概的例子来说明他/她在程序计划中所进行的某些尝试。	学生以明确的例子来说明他/她在程序计划中所进行的不同尝试。
你制作了哪些再次修订的程序版本？为什么要这样做？	学生说明他/她没有制作再次修订的版本，或说有却无法举例。	学生描述一个他所制作的修订版程序计划。	学生描述他/她增修到程序计划的内容并说原
描述你在程序计划中尝试的不同方法，或什么时候做了新尝试。	学生没有提出做新尝试的例子。述在程序计划中所做的	学生以大概的例子来描述程序计划中所做的	学生明确描述他/她在
测试与除错	低	中	高
当你的程序执行的结果与你想要的不同时，描述发生了什么事情。	学生无法描述程序执行的结果与想要的不同时，所发生的差异。想要程序执行的结果。	学生描述了程序哪边错了，却没有描述他/她	学生以明确的例子来说明了什么事，以及他/她想要程序执行的
描述你是如何将程序代码读过，来调查造成问题的原因。	学生无法描述问题是什么。式码的过程，但是无法以特定的例子来说明在程序代码中发现的问题。	学生描述他/她阅读程式码的过程，但是无法	学生描述他/她阅读程式码的过程，并且以特定的例子来说明在程序码中发现的问题。
描述你做了哪些改变并测试看看发生了什么事。	学生无法描述他/她遇到的问题是什么，或无法提出解答。测试看看可不可行。	学生以大概的例子来描述他/她所做的改变并	学生以明确的例子来说明他/她所做的改变并测
描述你是如何想到用其他方式来解决问题。	学生无法用例子说明问题的解决方法。述问题的解决方法。	学生以大概的例子来描述问题的解决方法。述问题的解决方法。	学生以明确的例子来描

再利用与融合	低	中	高
描述你是否/如何借着试用他人的程序并阅读他们的程序代码来找到灵感与启发。得构想或灵感。如何让他/她获得灵感。	学生无法描述他/她如何从他人的程序计划获得的程序计划，并且说明。	学生大概的描述了一个他/她灵感的程	学生以明确的例子来说明一个启发他/她灵感
你如何从别的程序计划中挑选出一部份，并改编它来用到你自己的程序计划？	学生无法描述他/她如何改编从其他的程序计划画取出的程序代码、构想或是其它素材。	学生指出他/她改编自其他的程序计划取出的程序代码、构想或是其它素材。	学生以明确的例子来说明他/她改编自其他的程序计划取出的程序代码、构想或是其它素材，并且说明过程。
你如何修改一个现有的程序计划来让它变得更好、功能更强大？	学生无法描述他/她修改了其他的程序。	学生大概的描述了他/她对于某个程序所进行的调整。所做的调整并说明。	学生以明确的例子来说明他/她对于某个程序的原因
你如何将你获得启发的功劳归功于他人(的作品)？	学生没有归功于他人。启发的作品创作者。	学生说出让他/她获得	学生在程序或Scratch网站上，写下并注明让他/她获得启发的作品创作者。
抽象化与模块化	低	中	高
你是如何决定你的程序计划所需要的角色？以及角色们该摆放在哪里？	学生无法描述他/她如何挑选角色。	学生大概的描述了他/她选择某些角色的决定过程。目标来决定所需的角色。	学生明确的说明他/她是如何根据程序计划的角色
你是如何决定你的程序计划所需要的程序代码？以及程序代码该做些什么？	学生无法描述他/她如何创作角色。	学生大概的描述了他/她为了角色们撰写了哪些程序代码。标来撰写角色们的程序码。	学生明确的说明他/她如何根据程序计划的目的来撰写角色们的程序
你是如何组织程序代码让它对你自己或其他人来说是合乎逻辑的(合理的)？	学生无法描述他/她如何组织程序代码。	学生大概的描述了他/她如何组织程序代码。	学生明确的说明他/她如何组织程序代码并说明原因。

绩效评估问卷：在课室中协助学生进行程序设计实际练习的程度

以下的问卷工具可用来帮助你回想沈思：你是如何在学习环境中协助学生进行程序设计实际练习，学习环境可能是：教室、图书馆或是其它学习环境。问卷的目的在于帮助你注意到你设计/协助的学习机会会有哪些的类型。

试验与重复：

编写一小段程序代码，然后试试看是否成功，然后再编写更多其他程序代码

课堂活动提供多少机会让学生们...	没有	一些	很多
一步骤一步骤地制作程序计划			
随心所欲的尝试			
根据发生的状况进行(程序)修改			
尝试不同的制作方式，或尝试新事物			
<p>改善笔记： 如果是「没有」，我能够如何创造更多机会，或保留更多时间？ 如果是「一些」，我能够如何深化、加强这些活动？ 如果是「很多」，我注意到什么？或学到什么？</p>			

测试与执行：

确认程序可以正确执行，并且在问题发生时可以找出问题并解决问题

课堂活动提供多少机会让学生们...	没有	一些	很多
观察当程序执行时，发生了什么结果			
描述与你想要的结果有什么样的差异			
将程序代码读过，来调查造成问题的原因			
做些调整，并测试看看发生了什么变化			
思考解决问题的其他方法			
<p>改善笔记： 如果是「没有」，我能够如何创造更多机会，或保留更多时间？ 如果是「一些」，我能够如何深化、加强这些活动？ 如果是「很多」，我注意到什么？或学到什么？</p>			

再利用与融合：

以现有的程序项目或构想为基础来制作其他作品

课堂活动提供多少机会让学生们...	NONE	SOME	LOTS
藉由试用他人的程序计划并阅读程序代码，来找寻构想与灵感			
从别的程序计划中挑选出一部份，并改编它来用到他们的程序计划			
修改一个现有的程序计划来让它变得更好、功能更强大			
将功劳归属于某些程序计划创作者，这些创作者的程序作品让学生们有所启发或让学生们用来融合出新作品			
<p>改善笔记： 如果是「没有」，我能够如何创造更多机会，或保留更多时间？ 如果是「一些」，我能够如何深化、加强这些活动？ 如果是「很多」，我注意到什么？或学到什么？</p>			

抽象化与模块化：

探索事物整体与构成组件之间的关联(关系)

课堂活动提供多少机会让学生们...	NONE	SOME	LOTS
决定程序计划所需要的角色，以及角色们该摆放在哪里			
决定程序计划所需要的程序代码，以及程序代码该做些什么			
组织程序代码让它对你自己或其他人来说是合乎逻辑的(合理的)			
<p>改善笔记： 如果是「没有」，我能够如何创造更多机会，或保留更多时间？ 如果是「一些」，我能够如何深化、加强这些活动？ 如果是「很多」，我注意到什么？或学到什么？</p>			

延伸阅读

能够进一步帮助你探索创意程设思维的阅读文本集锦

书籍

- + Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- + Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York, NY: Basic Books.
- + Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Mahwah, NJ: Lawrence Erlbaum.
Available at <http://www.yasminkafai.com/minds-in-play/>
- + Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT Press.
- + Margolis, J., Estrella, R., Goode, J., Holme, J.J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.
- + Kafai, Y. B., Peppler, K. A., & Chapman, R. N. (2009). *The computer clubhouse: Constructionism and creativity in youth communities*. New York: Teachers College Press.
- + Rushkoff, D. (2010). *Program or be programmed: Ten commands for a digital age*. New York, NY: OR Books.
- + Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.

学位论文

- + Monroy-Hernandez, A. (2012). *Designing for remixing: Supporting an online community of amateur creators*.
Doctoral dissertation, Massachusetts Institute of Technology.
- + Brennan, K. (2013). *Best of both worlds: Issues of structure and agency in computational creation, in and out of schools*.
Doctoral dissertation, Massachusetts Institute of Technology.

期刊文章

- + Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*.
American Educational Research Association meeting, Vancouver, BC, Canada.
- + Brennan, K. (2013). *Learning computing through creating and connecting*. *IEEE Computer*, Special Issue: Computing in Education.
doi:10.1109/MC.2013.229

网站连结

创意程设思维相关辅助资源：

类型	描述	网站连结
网站	Scratch	http://scratch.mit.edu http://scratch-tw.strikingly.com/ (中文资源)
网站	ScratchEd	http://scratched.gse.harvard.edu
网站	Flash	http://helpx.adobe.com/flash-player.html
在线资源	Scratch脱机版 (Offline Version of Scratch)	http://scratch.mit.edu/scratch2download
在线资源	Scratch教学图卡 (Scratch Cards)	http://scratch.mit.edu/help/cards http://bit.ly/29RKFKQ (中文版)
在线资源	Scratch社群指引 (Scratch Community Guidelines)	http://scratch.mit.edu/community_guidelines
在线资源	Scratch Remix FAQ	http://scratch.mit.edu/help/faq/#remix
在线资源	Scratch Wiki	http://wiki.scratch.mit.edu
在线资源	Scratch Discussion Forums	http://scratch.mit.edu/discuss
在线资源	Scratch FAQ	http://scratch.mit.edu/help/faq
在线资源	LEGO WeDo Construction Set	http://bit.ly/LEGOWeDo
在线资源	MaKey MaKey	http://makeymakey.com
在线资源	PicoBoard	https://www.sparkfun.com/products/10311
在线资源	Scratch设计工作室名单 (Scratch Design Studio list) http://scratch.mit.edu/users/ScratchDesignStudio	
影片	Scratch概观介绍影片 (Scratch Overview Video)	http://vimeo.com/65583694 http://youtu.be/-SjuiawRMU4 https://youtu.be/WUFiy8HJg2o (中文字幕) https://youtu.be/-u-q3eGgEUU (中文字幕)
影片	单元1 依照程序来跳舞影片 (Unit 1 Programmed to Dance Videos)	http://vimeo.com/28612347 http://vimeo.com/28612585 http://vimeo.com/28612800 http://vimeo.com/28612970
影片	「背包」教学影片 (Backpack Video Tutorial)	http://bit.ly/scratchbackpack
影片	「制作一个程序积木」教学影片 (Make a Block Video Tutorial)	http://bit.ly/makeablock
影片	「变数」教学影片 (Variables Video Tutorial)	http://bit.ly/scratchvariables
影片	How can I connect Scratch with other technologies? Video Playlist	http://bit.ly/hardwareandextensions
影片	Scratch Chain Reaction Video	http://bit.ly/ScratchChainReaction

台湾十二年国民基本教育科技领域 课程纲要草案

类型	描述	网站连结	QRcode
网站	国家教育研究院第二波公布之课纲草案、说明及Q&A 下载	http://high.ylsh.chc.edu.tw/files/14-1002-8749,r426-1.php?Lang=zh-tw	
文件	国民中小学及普通型高中科技领域课纲草案	http://12basic-forum.naer.edu.tw/sites/default/files/国民中小学及普通型高中-科技领域课程纲要草案.pdf	

Developed by the ScratchEd team at the Harvard Graduate School of Education and released under a Creative Commons license.

