

Learn Computational Thinking through Gamification & Experience the Fun of Game Programming

遊戲化學習運算思維，體驗遊戲程式設計樂趣

2024



陳俊銘 Jimmy

Vice-Chairman of HKACE

Fellow Member of HKACE

學與教 副校長

教授科目：IT/ICT/DT&T/STEAM/Maths/LS

曾連續5年擔任教育局資訊科技教育卓越中心 借調老師



香港電腦教育學會
The Hong Kong Association
for Computer Education



Google for Education
Certified Trainer



Google for Education
Champions



Professional Learning
Specialist



Learning
Coach

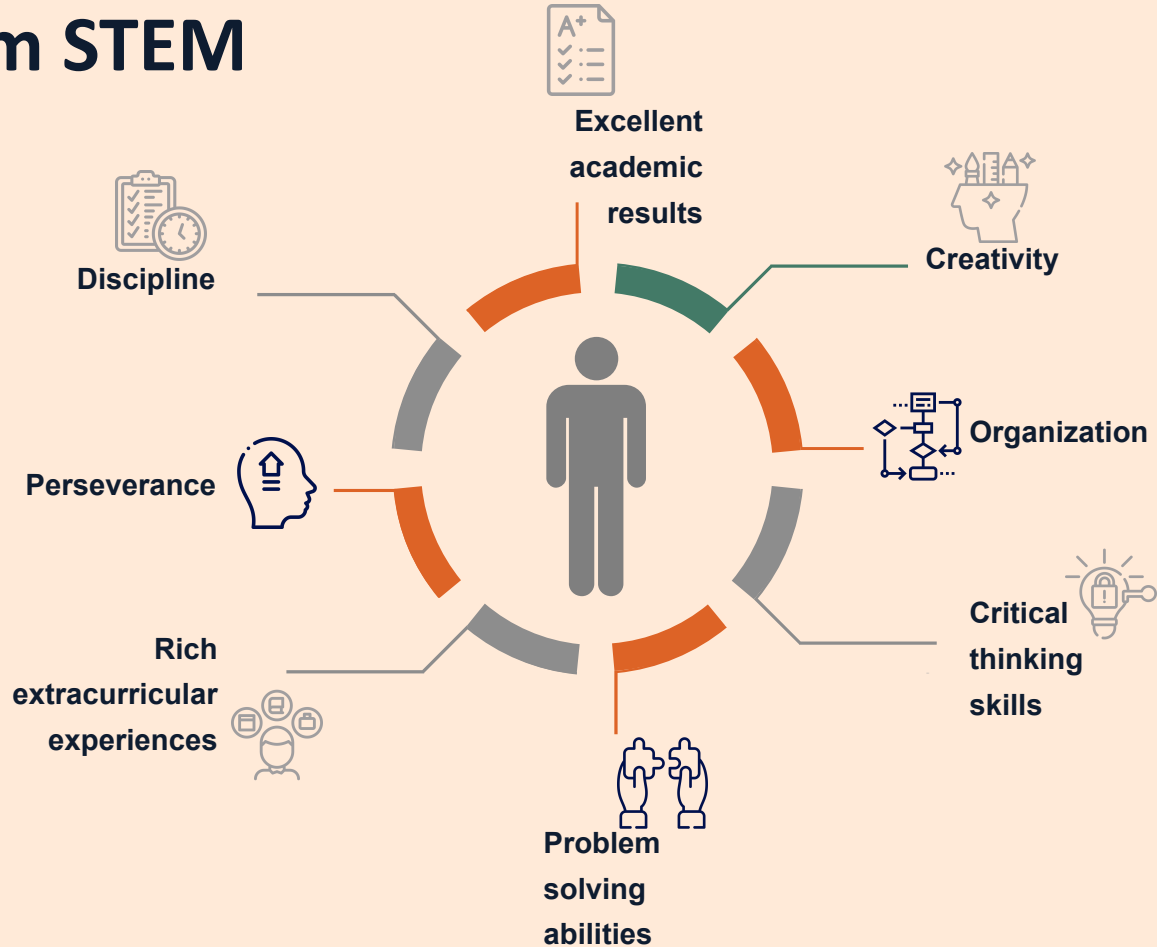


Apple Teacher

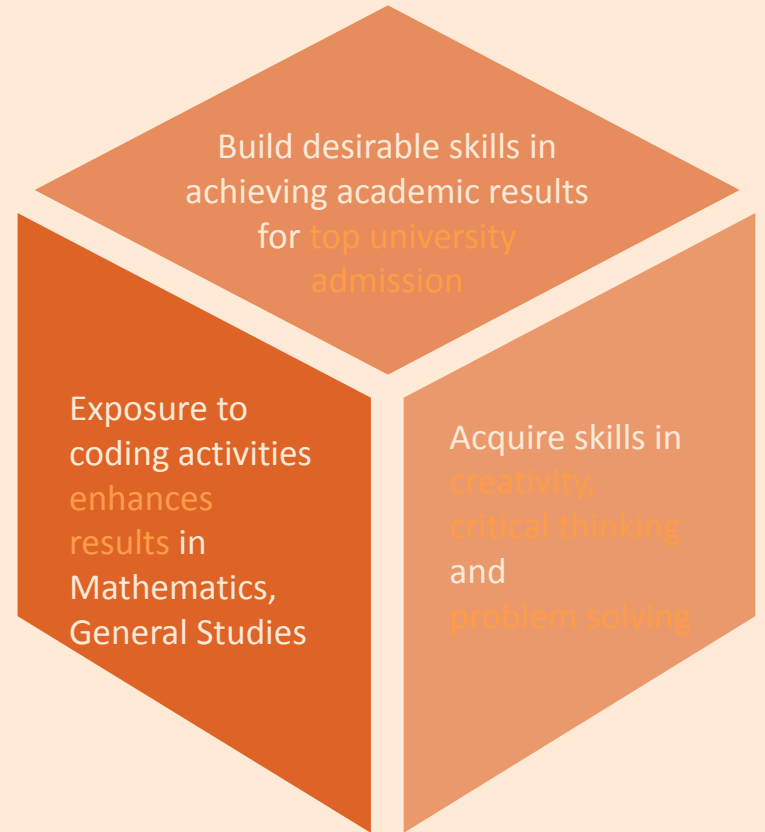
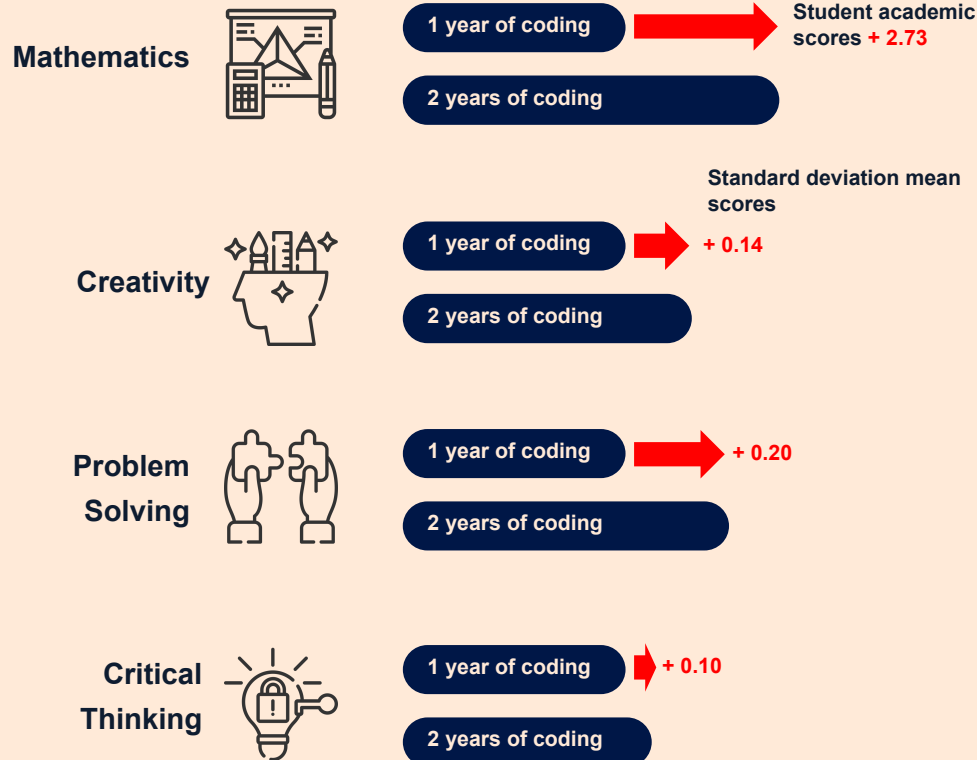


Personal Traits from STEM

Individuals with STEM, Coding and AI experience stand out among thousands of students, as these students usually possess the following desirable qualities and skills, such as:

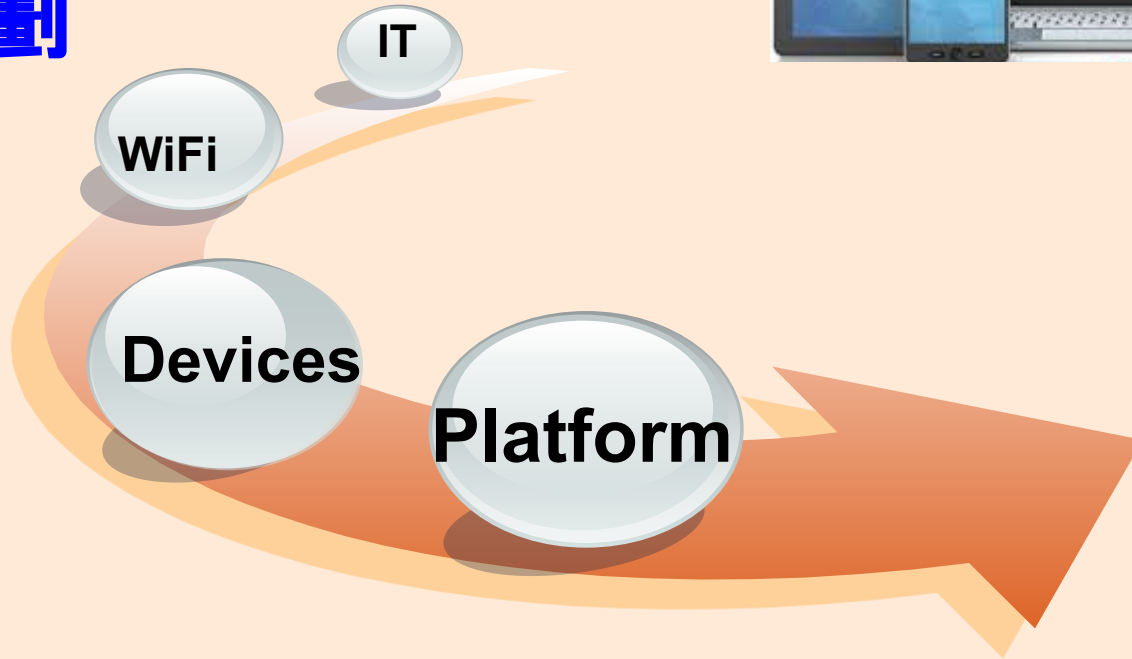


Why Focus on Coding and AI in STEM?



Chong Gene Hong College

學校數碼基建規劃



Personalized Learning

個人化學習的四大元素

Source: Gates Foundation's "Working Definition" of Personalised Learning

A WORKING DEFINITION OF PERSONALIZED LEARNING

Personalized learning seeks to tailor student learning by tailoring the learning environment—what, when, where, and how students learn—to address their individual needs, skills and interests. Students can take ownership of their learning, while also developing personal connections with their teachers and other adults.

GETTING STARTED
This is a working definition of personalized learning that is intended as a tool to help educators design student-centered instructional models. These attributes and tactics were developed from the practices of a number of leading schools. They are grouped together to offer a comprehensive view of the possible. No one school fully employs each of these today. Start where you want and progress from there.

LEARNER PROFILES
Each student has an up-to-date record of his/her individual strengths, needs, motivations and goals.

How might we capture each student's current level of mastery within each of the dimensions that we believe essential for his/her success (e.g. academic skills, social-emotional skills)? In what ways might we highlight a student's attention to their individual needs?

How might we support each student in understanding and articulating his/her interests and aspirations?

How might we support each student in setting personal goals within each dimension that we believe is essential for his/her success? In what ways and how frequently might we ask students to reflect on their progress and adjust their goals accordingly?

COMPETENCY BASED PROGRESSION
Each student's progress toward clearly-defined goals is continuously assessed. A student advances and earns credit for a skill or knowledge only when he/she demonstrates mastery.

How might we assess a student's mastery within the dimensions that are essential for his/her success?

How might we support each individual student to pursue mastery as soon as he/she has mastered a skill or knowledge? How might students attain mastery?

PERSONAL LEARNING PATHS
All students are held to clear, high expectations, but each student follows a customized path that is designed to meet his/her individual strengths, needs, interests, motivations and goals.

How might we ensure that each student has a learning path that is tailored to his/her strengths, needs, interests, motivations and goals? How might we ensure that a student's plan is dynamic and responsive to his/her changing needs?

How might we ensure that students are able to complete complex tasks, projects and assignments that they need to succeed? How might we ensure that students need the ideal modalities and resources to succeed? How might we ensure that students are able to personalize their learning experiences? How might we ensure that students are able to personalize their learning experiences?

FLEXIBLE LEARNING ENVIRONMENTS
Students choose the design of the learning environment. All operational elements—scheduling, space allocation and time allocation—improve and adapt to support students in achieving their goals.

How might we deliver all of the learning experiences that our students need, with the resources we have available? How might we build flexibility into our design to enable us to respond and adapt to our students' changing needs?

OPERATIONAL ALIGNMENT
How might we ensure that all of the learning experiences that our students need, with the resources we have available? How might we build flexibility into our design to enable us to respond and adapt to our students' changing needs?

STAFFING & ROLES
How might we ensure that all of the learning experiences that our students need, with the resources we have available? How might we build flexibility into our design to enable us to respond and adapt to our students' changing needs?

GROUPING & CONNECTIONS
How might we ensure that all of the learning experiences that our students need, with the resources we have available? How might we build flexibility into our design to enable us to respond and adapt to our students' changing needs?

學生學習歷程記錄

學生能力為本的學習進程

個人化的學習途徑

靈活的學習環境

Learning to Learn 2+ — The Hong Kong School Curriculum

A broad and balanced curriculum with diversification and specialisations (choices) for academic, professional and vocational development according to students' needs

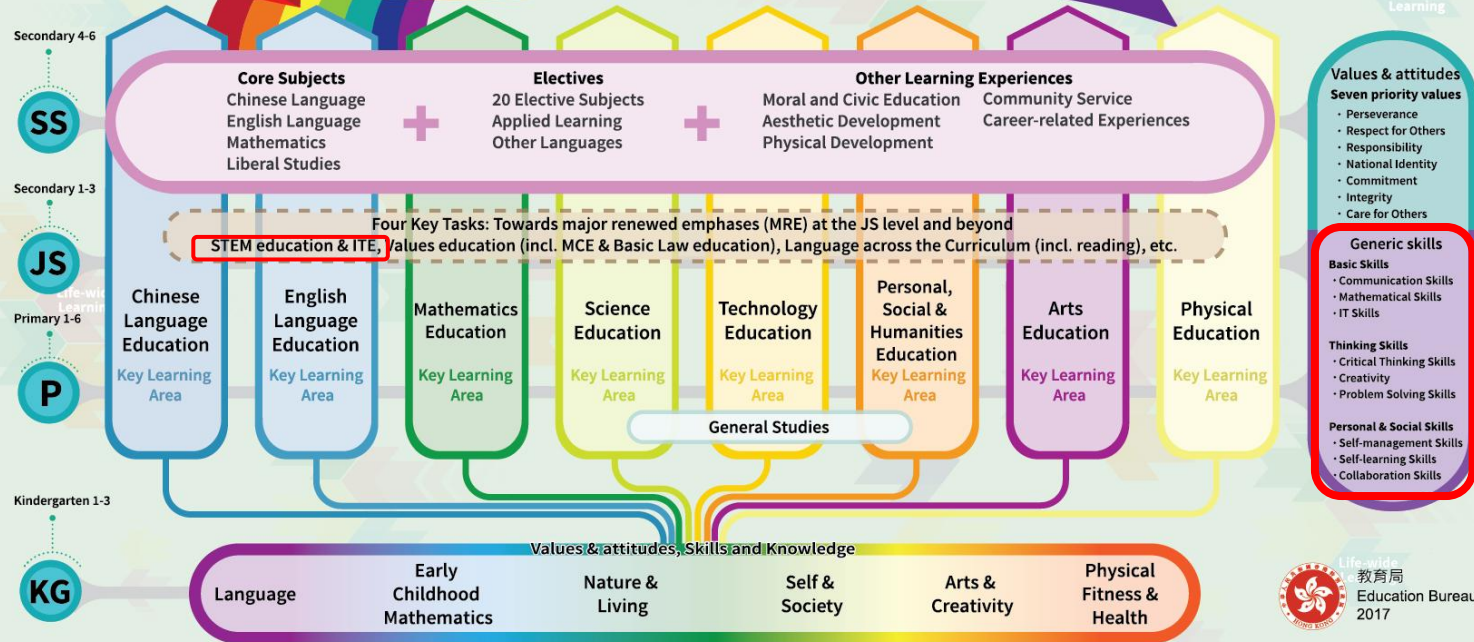
Nurturing
lifelong & self-directed
learning capabilities

Fostering
whole-person development

SEVEN LEARNING GOALS

FIVE ESSENTIAL LEARNING EXPERIENCES

Moral and Civic Education Intellectual Development Community Service Physical and Aesthetic Development Career-related Experiences



編程的重要

生活化

STEM 教育強調以日常生活問題作為學習情景，讓學生從課本走到社會及生活上去，增加學習內容與學生的關聯，亦令學生明白到不同學科並非獨立存在，而是環環緊扣，更與社會有著密切關係，需要綜合運用才能解決實際問題。

問題為本

STEM 綜合活動通常以實際問題為發端，旨在連繫日常生活，令學習聚焦。通過問題為本學習，發展學生解難及自學能力。因應問題的複雜程度，學生需要綜合運用不同學科的知識，從中學會新知識和鞏固原有的知識。通過自主學習，學生還可以體會到學習和解難並非兩種獨立存在的認知活動，又或是有先後次序之分，而是處於互動狀態。

手腦並用

STEM 教育包含手腦並用的活動，這些活動包括科學探究、工程設計和製作、科技器材的運用、電腦程式的編寫，而至視覺藝術和工藝創作等。因此，STEM 教育並不局限於認知能力的發展，也是操作技巧的訓練，工匠精神培養和對美感的欣賞和追求。

科技融入

在活動中融入科技是 STEM 教育的一大特色，也是 STEM 其中兩個主要範疇 T 和 E 的基本內容。科技可以提供探究和製作工具，促進科學探究和工程製作。此外，訊息的搜尋和傳播，也得依賴資訊科技的協助。至於如何融入科技於 STEM 教育，取決於要解決問題的性質和學生的學習階段。

激勵創意

STEM 活動鼓勵創意和創新地應用知識，強調運用分散性思維，與現行課程所重視的聚合性思維有顯著差別。發展創新科技更是香港推行 STEM 教育的長遠目標之一。

學習自主

STEM 教育與傳統學科學習的最大分別是鼓勵自主學習。通過自主學習，營造開放的學習環境，培養學生創意思維、解難能力，以及高階思維如分析、評鑑和創造。學生在學習過程中所經歷的策劃、檢討、反思及改良過程，更有助發展後設認知能力，有利學習遷移。

協作性

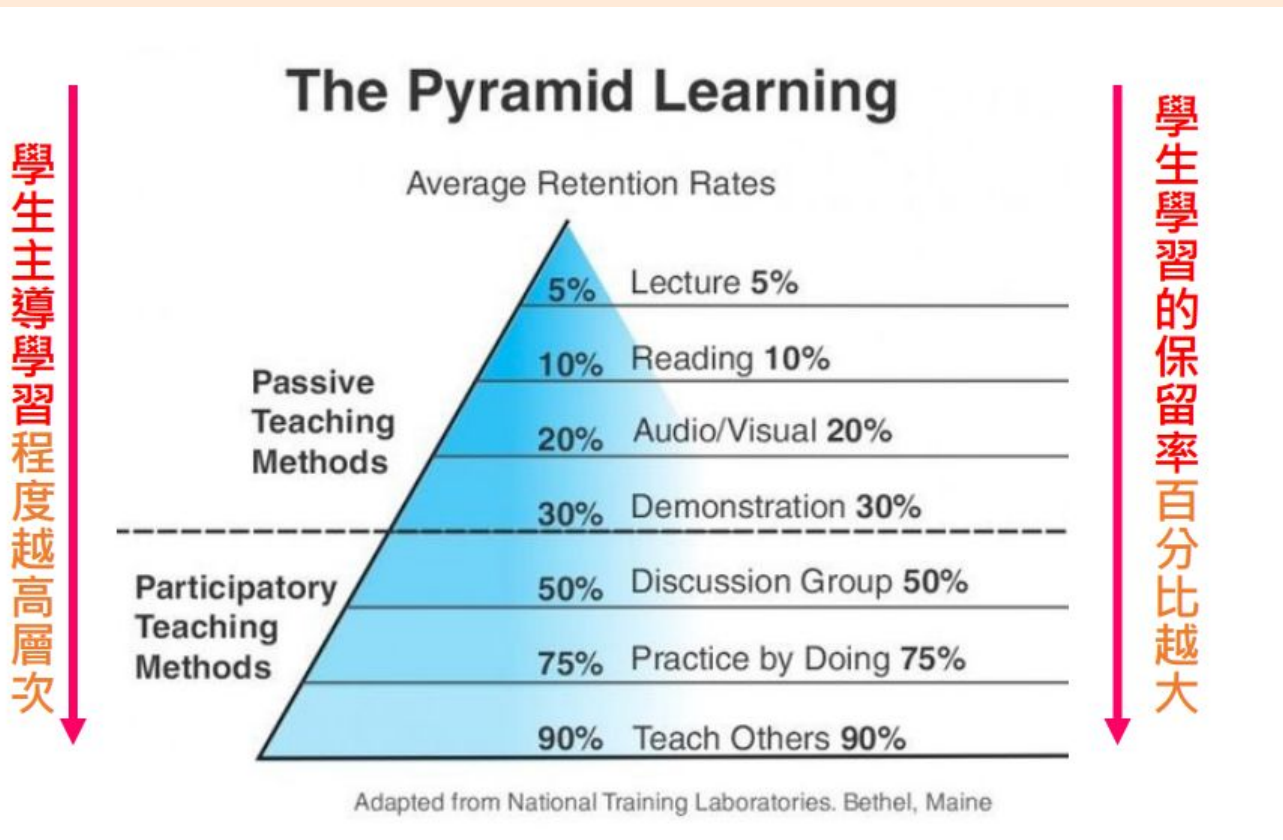
由於 STEM 教育是以解決問題為導向，而在解難過程中需要應用多元知識和技能，所以非常強調群體合作，在互相幫助和共同努力下完成任務。因此，STEM 教育多是以小組協作活動為基礎，著意培養學生的溝通和協作能力。

設計多元化

STEM 課程及教學設計是沒有通則的。基本上，學校可以自行決定 STEM 課程與正規課程的緊扣程度、課時安排、上課地點、活動形式和學生學習的自主程度等，只要符合以上所討論的 STEM 教育特質便可。

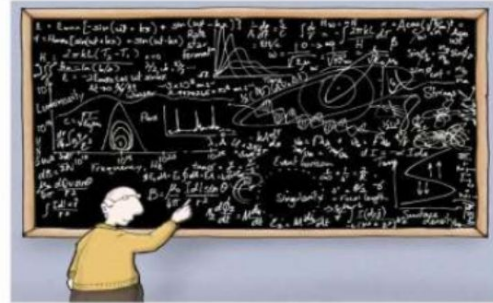


自主學習 (Self Directed Learning)



SDL, SRL or SDL ?!?

Three concepts, lots of overlap: learner-centered, autonomous learning, motivation, skills, collaboration



Astrophysics made simple

Differences?

- Self-Directed Learning (Knowles/Merriam, 2001) - Andragogy
- Self-Regulated Learning (Zimmerman & Schunk, 1989; LittleJohn) – Classroom learning
- Self-Determined Learning (Hase & Kenyon, 2000; Blaschke (yeah!)) – Heutagogy

Cartoon: fabulous Nick D. Kim - the <http://www.lab-initio.com/> site

Sources from:

<https://www.slideshare.net/ignatia/investigating-selfdirected-learning-dimensions-adapting-the-bouchard-framework>

Self-Regulated Learning VS Self-Directed Learning

- Self-Regulated Learning: 既定課題及框加內, 學生自律地學習;
- Self-Directed Learning: 學生自主地選擇學習材料同課題。
- 其實中文翻譯兩者都是自主學習:
- Self-Regulated Learning 較適用於中小學生/班本模式教育
- Self-Directed Learning 較適用於大專學生/小組抽離模式資優教育

教學新常態 學與教 (eLearning Devices)



流動電腦裝置 + 無線網絡環境 = 提升學習效能?

 Microsoft Surface

 chromebook

 iPad Pro

 ANDROID

編程新世代嘉年華 Programming New Generation Carnival

HOURLY OF CODE

活動 活動指南 推薦 常見問題與解答 關鍵字



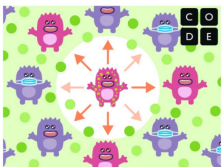
一小時玩程式

體驗為所有年齡層設計的一小時玩程式教程 (包括超過 45 種語言)。參加超過 180 個國家、數以萬計的學生。教師新增的一小時玩程式活動。

想繼續學習? 超越一小時

教師: [主話一小時](#) 或 [閱讀指南](#)

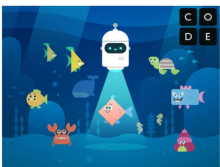
中文活動



Outbreak Simulator
年級 2+ | 橫式



舞動派對 2019
年級 2+ | 橫式



保護海洋的人工智慧 (AI for Oceans)
3 年級以上 | AI 和機器學習



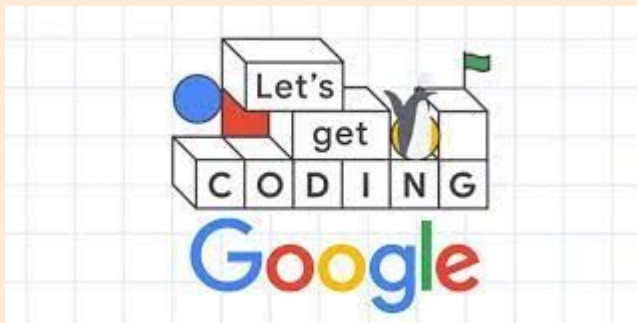
一小時玩 Minecraft
年級 2+ | 橫式



AI 棋盤之路
年級 6-8 | JavaScript, Python, Lua, CoffeeScript

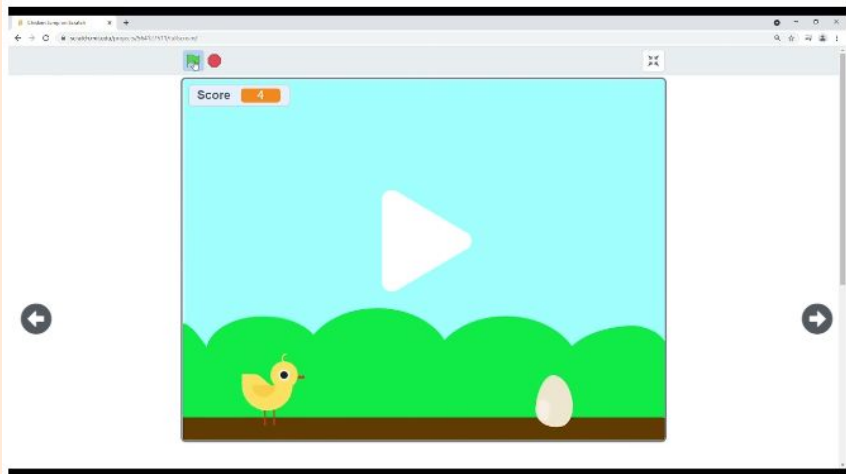


Kodable (適合學齡)
學齡 - 年級 5 | 橫式 | 所有教師的瀏覽器, iPad app

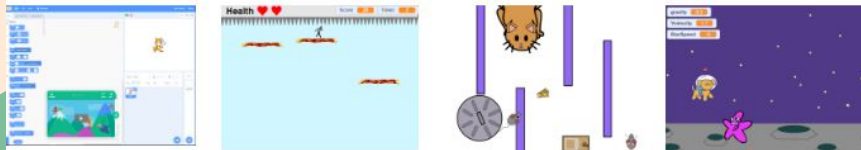


免費遊戲引擎 (Free Game Engine)

名稱	主要開發	難易度	功能	寫程式
Unity	2D & 3D	★★★★★	★★★★★★	需要
Unreal	2D & 3D	★★★★★★	★★★★★★	需要
Godot	2D & 3D	★★★★	★★★★	需要
Scratch	2D	★	★★	不需要
GDevelop	2D	★★	★★★	不需要



.....



Scratch 課程

以有趣的Scratch程式設計世界，引導學生學習以積木式程式設計製作動畫、遊戲，甚至是聊天機器人。

MakeCode Arcade



平面遊戲設計課程

MakeCode Arcade是一個開源的遊戲開發平台，它提供了豐富的遊戲開發工具和資源，讓學生能夠輕鬆地創建自己的遊戲作品。

Nintendo Switch 遊戲程式設計課程



通過有趣的遊戲主機Nintendo Switch和Game Builder Garage遊戲介紹遊戲設計和開發的基礎知識。學生將深入了解遊戲設計的方法，並運用他們的創造力來創作自己獨特並涵蓋各種類型的遊戲。學生將透過實作設計他們的第一款遊戲，成為年輕遊戲設計師。

學習目標：

- 掌握使用 Nintendo Switch 上的 Game Builder Garage 遊戲，學習創建遊戲的基礎知識，包括基本控件和遊戲設計原則。
- 開發和設計各種遊戲，從雙人追逐遊戲到 3D parkour 謎題，增強角色控制、關卡設計和互動敘事方面的技能。
- 通過構建和展示原創遊戲，應用學到的技術和創造力來製作吸引且好玩的遊戲。



Nintendo

附帶
導航!

一做就上手

第一次的

遊戲
程式設計



連線式編程

傳統的編程中，我們需要像書寫般把程式碼寫成一行又一行的句子。在《第一次的遊戲程式設計》中，擺脫舊有經驗，簡單連線去完成編程。隨時切換到遊戲中查看進度，一邊遊玩一邊學習吧！



遊戲內住著不思議的生物

透過使用線條去連接不同的「小節點」



搖動操控搖桿就可以走動!



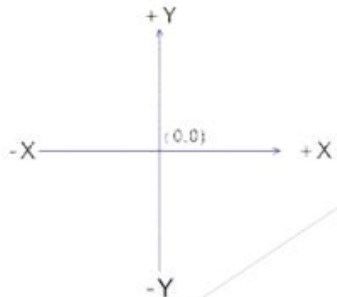
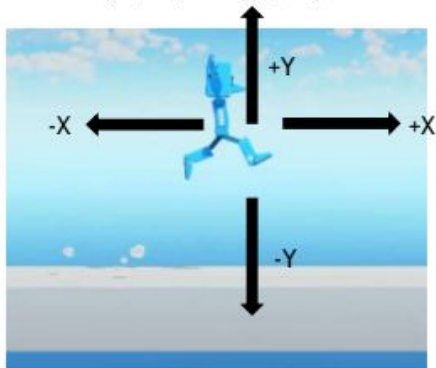
坐標

編程補充站

X-Y坐標系統

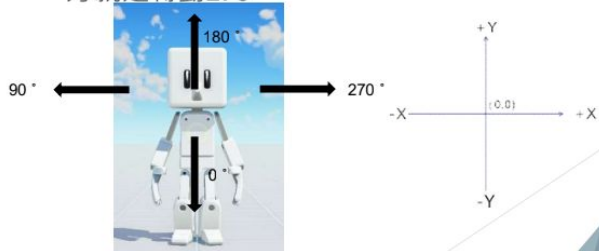


- ▶ 要形容物件在平面上的位置，我們一般使用有X/Y 兩個數值組成的系統
- ▶ 如圖中所示，中間相交位置我們稱為原點(0,0)；往右邊方向X值會相應增加，如(1,0)、(2,0)，反之減少，如(-1,0)、(-2,0)；向上Y值會相應增加，如(0,1)、(0,2)，反之減少，如(0,-1)、(0,-2)



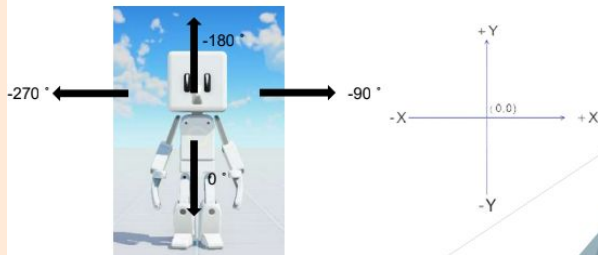
Y軸的轉動

- ▶ 除了先前介紹的X/Y值隨物件的位置而改變外，我們還可以使物件以X軸/Y軸為軸心轉動
- ▶ 以人小節點為例，它一開始面向螢幕(Y軸轉動 0°)，我們以順時針方向轉動，轉到面向左方就是轉動 90° ，面向背面就是轉動 180° ，面向右方就是轉動 270°



Y軸的轉動

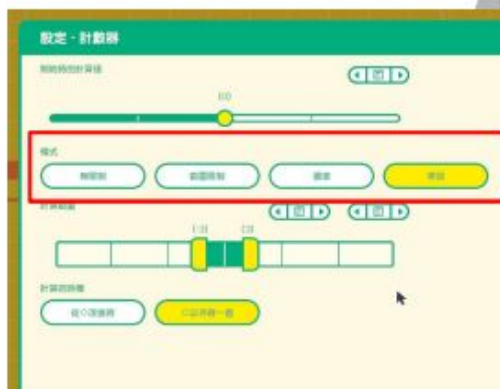
- ▶ 反之，我們亦可以逆時針方向轉動，轉到面向右方就是轉動 -90° ，面向背面就是轉動 -180° ，面向左方就是轉動 -270°



Counter

計數器的應用

- ▶ **模式**共有4種
 - ▶ **範圍限制**：到達**計算範圍**後不再改變數值
 - ▶ **循環**：到達**計算範圍**後重新由**開始時的計算值**中設定的數值開始計算
 - ▶ **來回**：到達**計算範圍**後倒轉增加/減少，直至到達另一端的**計算範圍**



Logic

叫出以下的小節點並把它們按下圖連接

- ▶ 中間-程式整理-無線傳送：出口(A · B)
- ▶ 中間-旗標、計數器、隨機-旗標
- ▶ 輸入-常數
- ▶ 中間-旗標、計數器、隨機-計數器
- ▶ 中間-比較- =
- ▶ 中間-轉換-定位
- ▶ 中間-邏輯-AND

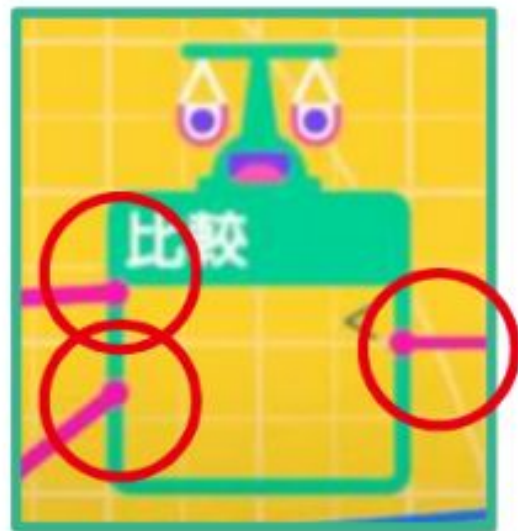


Logic - Boolean

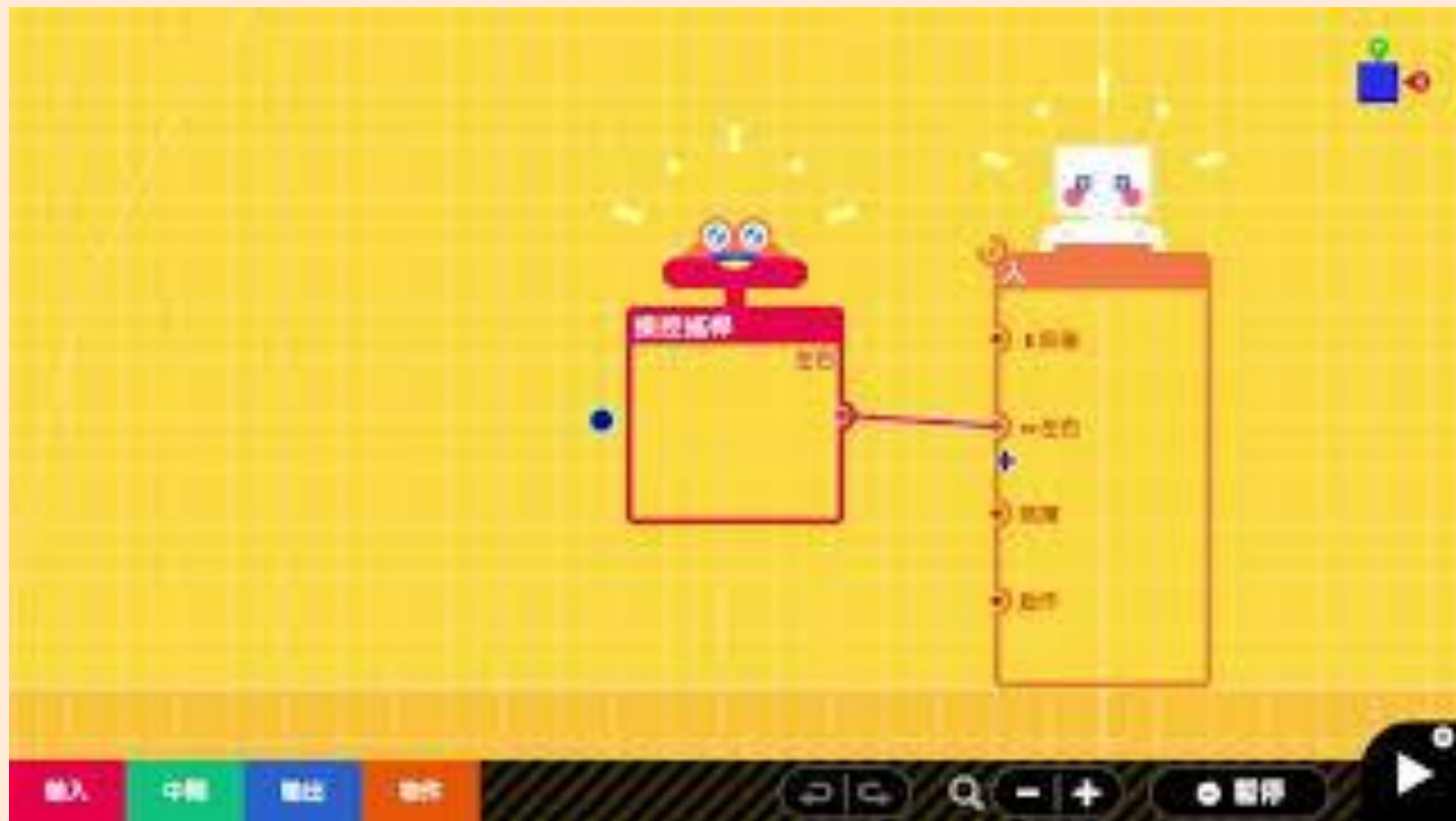
編程中的比較

比較是在編程中其中一樣最重要的概念，用作比較兩個輸入數值的關係並輸出布林值(boolean)

如右圖中，比較小節點會判斷**輸入端1**的數值是否小於**輸入端2**，並輸出**1(True)**或**0(False)**
(基乎所有有兩個**輸入端**的小節點都是同樣做法)



連接小節點



呼叫小節點

The image shows a programming environment with a yellow grid background. On the right, a white robot character is positioned on a red rectangular block. A vertical script area on the right contains two blocks: a 'Person' block (人) and a 'Say' block (說話). A speech bubble above the script area contains the text '呼叫小節點' (Call sub-block).

On the left, a block palette is visible with several categories: 'Media' (媒體), 'Motion' (動作), 'Looks' (外觀), 'Sound' (聲音), 'Control' (控制), 'Cycles' (循環), 'Logic' (邏輯), and 'Variables' (變數). The 'Control' category is expanded, showing a 'Call sub-block' (呼叫小節點) block highlighted in yellow. This block has a plus sign (+) and the text '呼叫小節點'.

At the bottom, there is a toolbar with icons for 'Insert' (插入), 'Undo' (撤銷), 'Redo' (重做), 'Search' (搜尋), 'Zoom In' (+), 'Zoom Out' (-), and 'Run' (執行).

改變狀態

設定 - 物件

物件顏色

白色 藍色 綠色

透明度

不透明 半透明 透明

顯示

顯示 隱藏 關閉

旋轉

旋轉 一般

旋轉速度

一般 快

旋轉角度

一般 90度

X軸

X1(0.00) X2(0.00)
X3(0.00) X4(0.00)

Y軸

Y1(0.00) Y2(0.00)
Y3(0.00) Y4(0.00)

請把旋轉、透明度、旋轉速度
設定為關閉。

改變大小





傳統編程

我們需要像書寫般把程式碼寫成一行
又一行的句子

這種方式大大加深了學習編程的
難度，發現錯誤時也較難修改。

我們平時在遊戲機上的遊戲，背
後都是遊戲設計師透過編程製作



遊戲界面

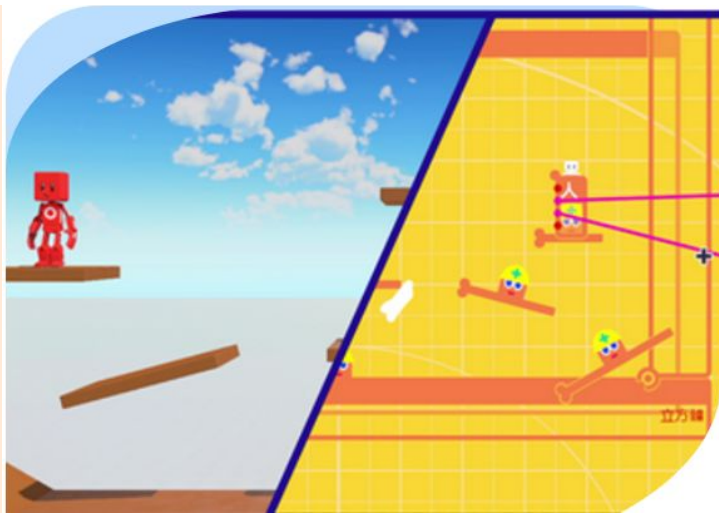


編程邏輯

材質設計

#創作屬於你的角色

描繪的畫可以用來當角色，也可以用來作背景。把腦海中想像的描繪出來，設計屬於自己的遊戲！

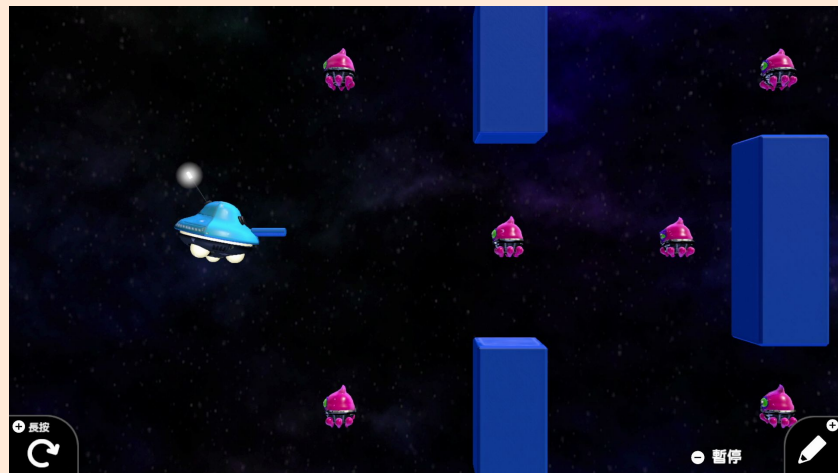
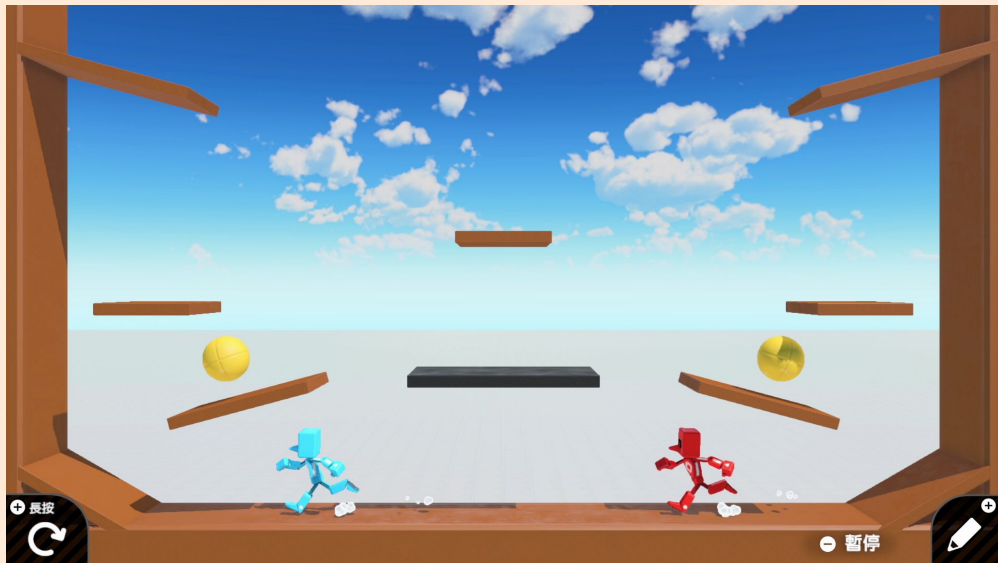


即時測試

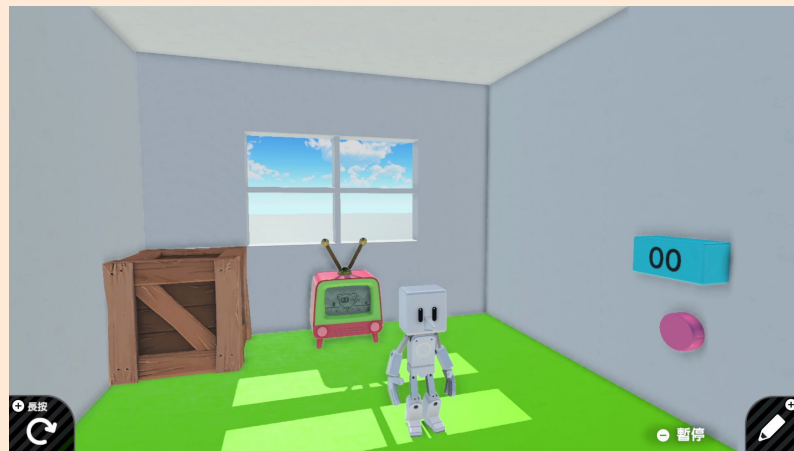
#一鍵切換 #直觀學習

《第一次的遊戲程式設計》擁有獨特的即時查看功能，只需按下一個按鈕，就能立即切換遊戲畫面和程式碼，讓你能直觀地觀察程式碼修改後的結果，提升學習效率，更輕鬆掌握遊戲開發技巧。

課堂上製作的遊戲共7款



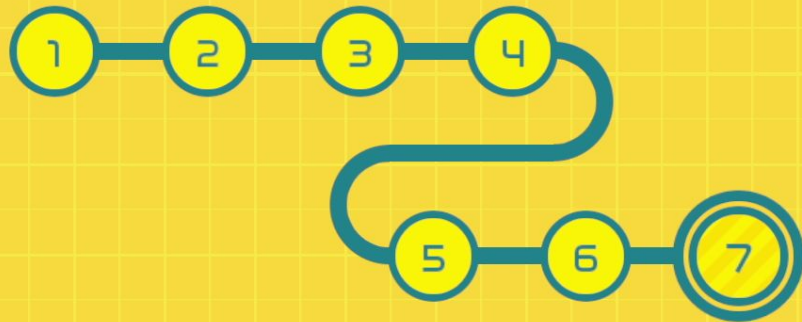
課堂上製作的遊戲共7款



二人對戰! 捉迷藏 大戰

步驟7

最後一步



小節點指南

Y

B 返回 A 決定

「附帶導航的教室」按鈕提供了共7個不同的編程課程，遊戲規定我們必須先完成第一課，其餘課程同學可在課後自行遊玩。

附帶
導航!

第一次的

一做就上手

遊戲
程式設計

◀ 附帶導航的課堂

💡 自由設計程式

之後的課節，我們會直接按下右方的「自由設計程式」按鈕。

開始製作吧！

熱身活動 取得蘋果吧!

移動
跳躍

去取得蘋果吧!

未有編程前，遊戲便不能正常運行

按下 + 按鈕後，我們會進到「小節點車庫」，亦即是遊戲的「後台」。這就是我們編程的地方。

工程思

小節點車庫

左下方的4個
按鈕存放著不
同的小節點



同學亦可隨時
再按下 + 按鈕
回到遊戲畫面
觀看成果



從 **物件-一般物件** 中叫出兩個 **立方體** 物件小節點，調整大小讓它們剛好組成一面牆

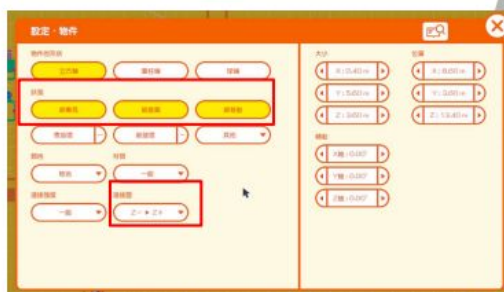
從 **物件-連接零件** 中叫出 **滑動連接** 物件小節點，並按照下圖所示連線



開啟兩個物件小節點的設定

把第一個 **立方體** 小節點的 **狀態** 調成僅有 **能看見** 和 **能接觸** 是開啟

把第二個 **立方體** 小節點的 **狀態** 調成僅有 **能看見**、**能接觸** 和 **能移動** 是開啟，**連接面** 改成 **Z-/Z+**



- ▶ 完成課程後，同學會進入到遊戲庫。
- ▶ 頁面右邊會顯示同學曾製作的遊戲，又或是從網絡中下載的遊戲。
- ▶ 左邊的「分享」及「領取」按鈕可跟身邊的人/網絡上的玩家收/發遊戲。
- ▶ 最後，按下「建立新檔案」開始製作新的遊戲吧。



- ▶ 如果想加入更多自己的元素，可在「自由設計程式」進行
- ▶ 要先完成該關卡，再在以下頁面複製一個，複製出來的那一個就可以進行大改動





遊戲製作工作坊 (Game Develop Workshop)



- GDevelop 遊戲編輯器
 - 與 Construct 非常相似, 但是免費及 Open Source
 - <https://editor.gdevelop.io/>
 - 多平台: PC, Web, Android, IOS
 - 電腦設備要求低
 - 即時預覽
 - 介面簡潔
 - 不用寫程式
 - 功能強勁
 - 對應不同輸入: 鍵盤、滑鼠、手機觸屏

<https://gdevelop.io/>

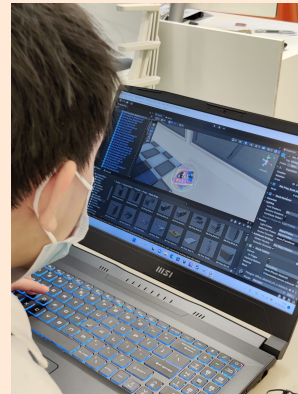
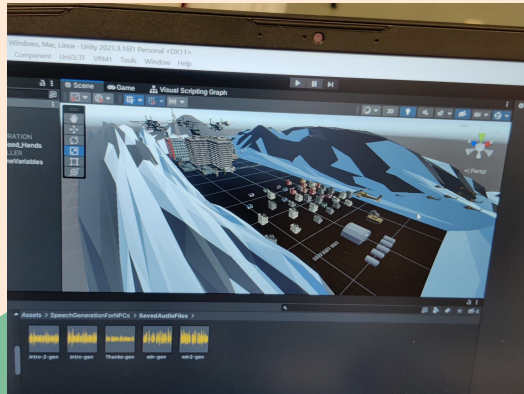
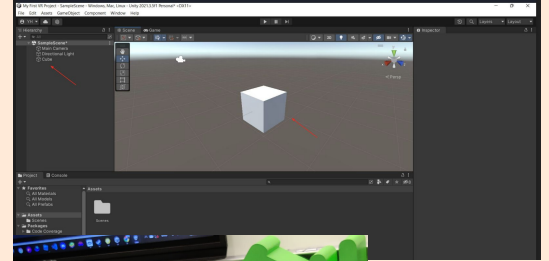
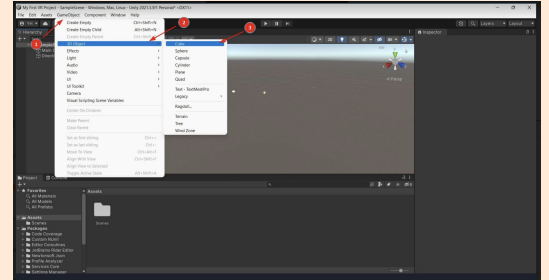
<https://gamecreator.io/>

原宇宙體驗及Unity創作設計

Metaverse in 2020s



UNREAL
ENGINE



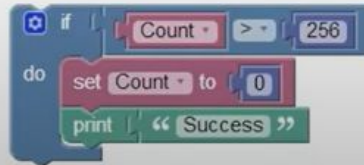


From Blockly programming to Coding

Blockly Vs. Python

Blockly

- Fixed programming environment, unable to add outside libraries easily
- Learning to read before learning to write
- Application is limited



Python

- Dynamic environment allows the addition of outside libraries of code easily
- Fully customizable, easily adapted to many applications





How To Make The Transition

Bridge the gap around key concepts for object-oriented programming

- Variables
- Conditional Statements
- Loops
- Functions
- Lists

```
set Count to 1
repeat while Count <= 3
do
  print Count
```

```
Python
English

Count = None

Count = 1
while Count <= 3:
  print(Count)
```

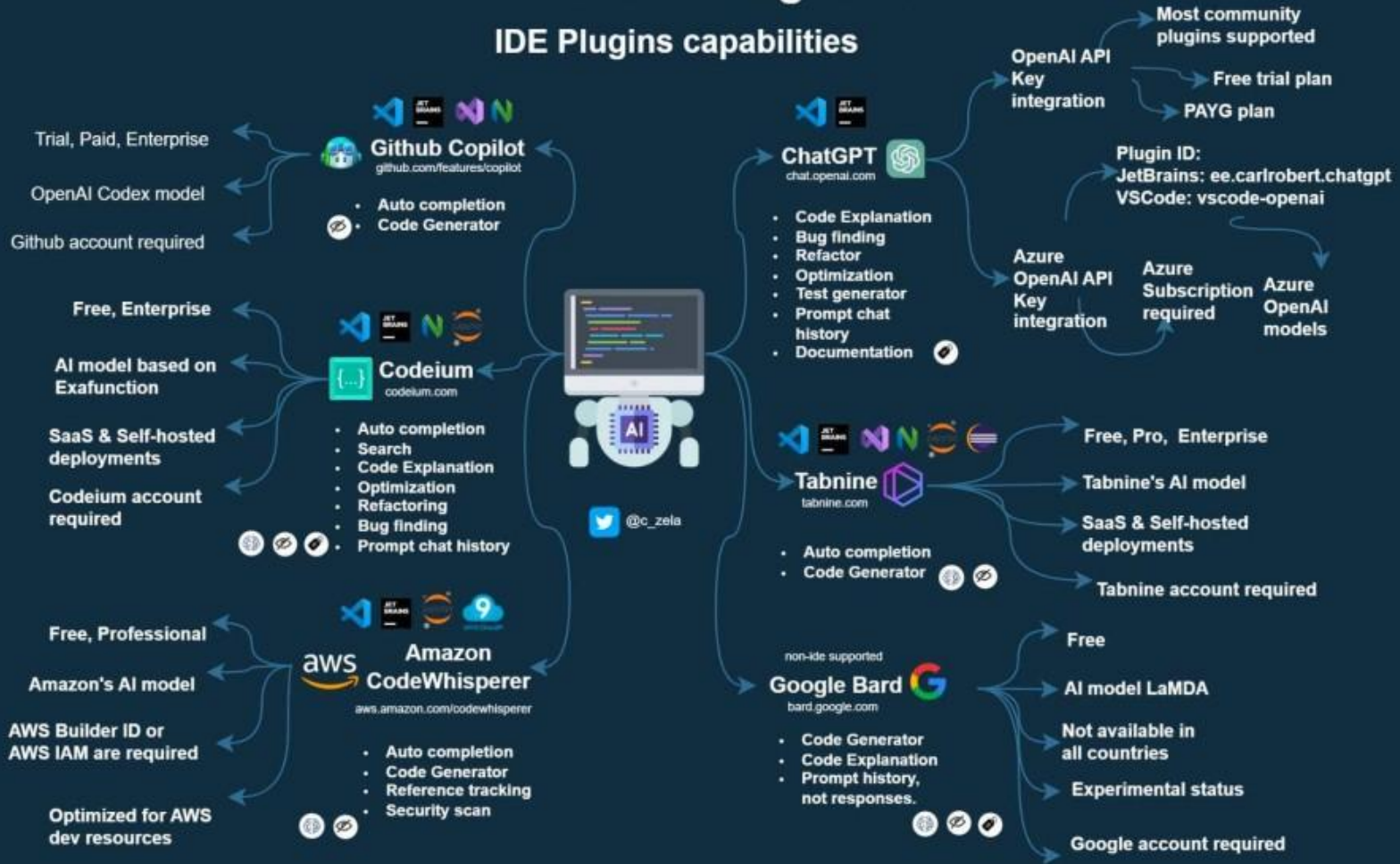
Have students compare Python and Blockly code to determine if they will result in equivalent outputs

```
set Count to 1
if Count < 10
do
  change Count by 1
print "Count Too High"
```

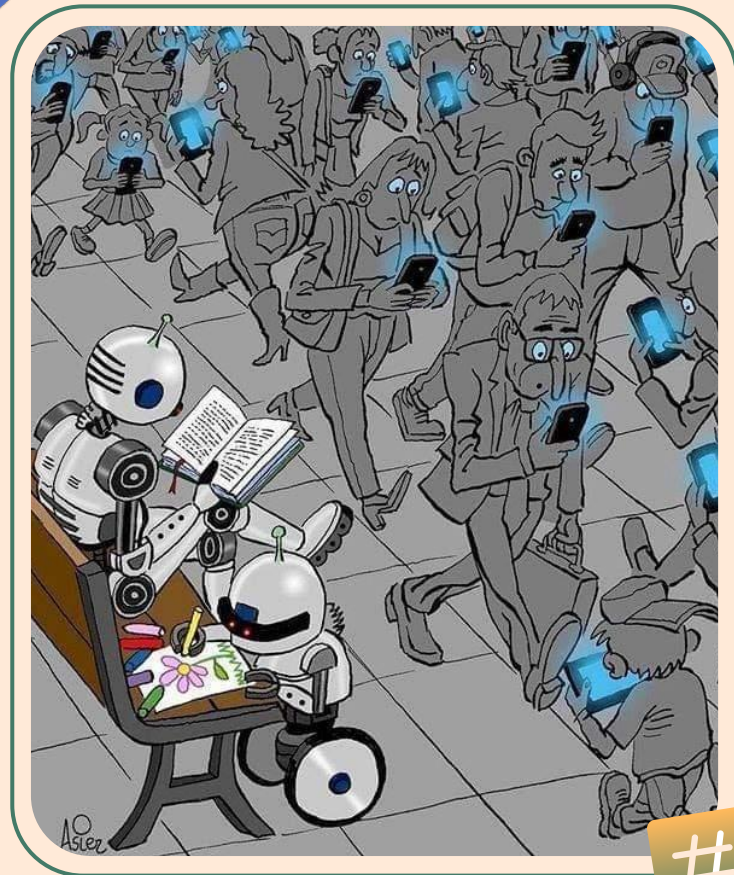
```
if Count < 10:
  Count = (Count + 1)
Else:
  print('Count Too High')
```

AI-assisted coding tools

IDE Plugins capabilities



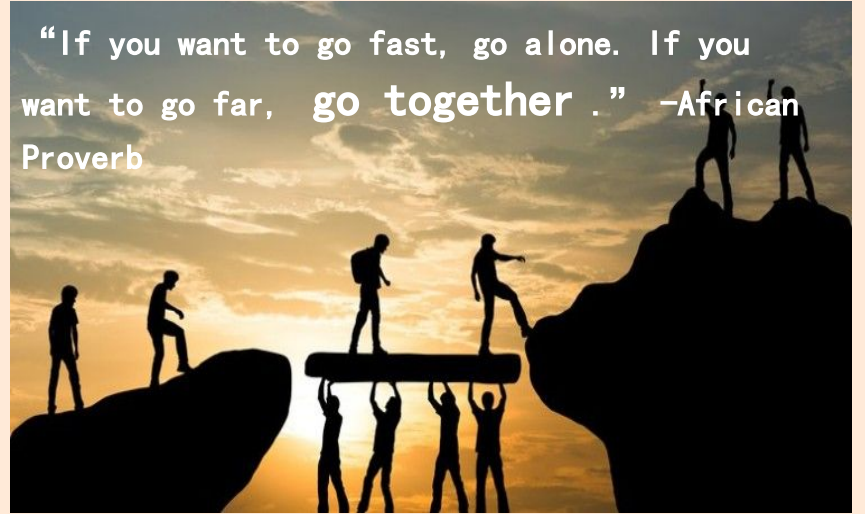
誰在學習？



#



“If you want to go fast, go alone. If you want to go far, **go together** .” -African Proverb



Thank you! Q&A