

除了 Scratch 或機器人的[基礎程式設計]課程外，

更進階的程設課程-[物件導向程式設計(OOP)]

OOP-開啟進入[高階程式設計]的能力之鑰!

何謂物件導向程式設計(Object-oriented programming, OOP) ?

OOP 是種具有物件(Object)概念的程式設計方法。OOP 包含資料、屬性、程式碼與方法。物件(Object)指的是類別(Class)的範例(Instance)。OOP 將物件作為程式基本單元，將程式和資料封裝於物件中，以提高軟體的重用性、靈活性和擴充功能性。

為何要學習物件導向程式設計概念?

(1)因為-OOP 是現今程式設計的高階抽象概念與重要設計方法:

OOP 概念能讓程式更易於設計、理解、分析、與維護。而此 OOP 概念更有益於運算思維的系統化思考與培養。

(2)因為-OOP 是現今主流程式語言的重要設計基礎:

許多主流程設語言(如: Python (AI 熱門語言)、C++、Objective-C、Java、C#、PHP、LISP (AI 語言))都是基於 OOP 設計概念，學會 OOP 概念才能有效駕馭主流程設語言，也是進入高階程設與人工智慧(AI)的必備能力。

為何要選擇此課程?

(1)因為-全台首創面向兒童的 OOP 程設概念學習課程:

現今坊間面向兒童開設的程設課程，普遍都是只能學習[[結構化程式設計](#)]的基本程設概念(如: Scratch 課程，機器人課程、甚至 Javascript 課程)，尚未有能夠針對兒(學)童教授高階概念的 OOP 程設課程。

(2)因為-全台首創透過圖形化學習工具來教授與學習難懂的 OOP 程設抽象概念:

目前 OOP 概念的教學仍多採用[文字式開發工具]做教學，但 OOP 概念因抽象所以難懂難學，此課程獨有的[圖形化學習工具]可讓 OOP 的學習更加簡單易學。

(3)因為-易懂易學，即學即用，遊戲創作，樂趣無窮

此課程透過圖形化學習工具來說明如何基於 OOP 概念來設計與製作可遊玩的遊戲應用，讓學生在過程中慢慢學會抽象難懂的 OOP 概念，進而能自行設計與創作屬於自己創意的遊戲應用，能有效提升學習樂趣與動機。

授課現場：

- 豐富有趣教材，學生快樂學習



- 多樣挑戰活動，同儕協力討論



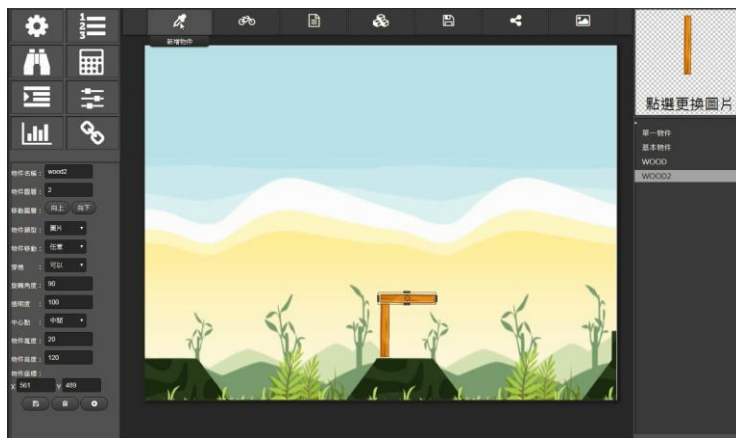
● 積極完成 OOP 遊戲作品，相互切磋競技



● 自製 OOP 遊戲專案，創意思考動手實踐

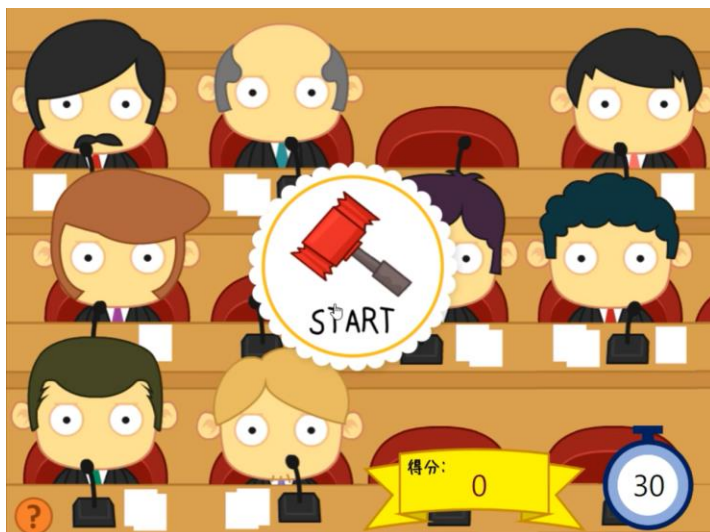


OOP 圖形化學習工具範例：(掃圖或點選 QR Code 觀看展示)



課堂實務 OOP 遊戲創作範例:

(1) 打擊瞌睡蟲遊戲-專案遊戲: (手機/平板遊玩效果)



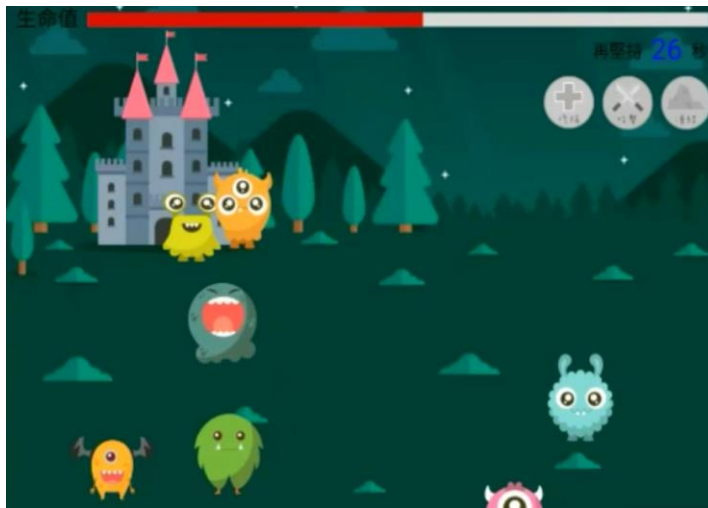
(2) 電子琴-遊戲專案: (手機/平板遊玩效果)



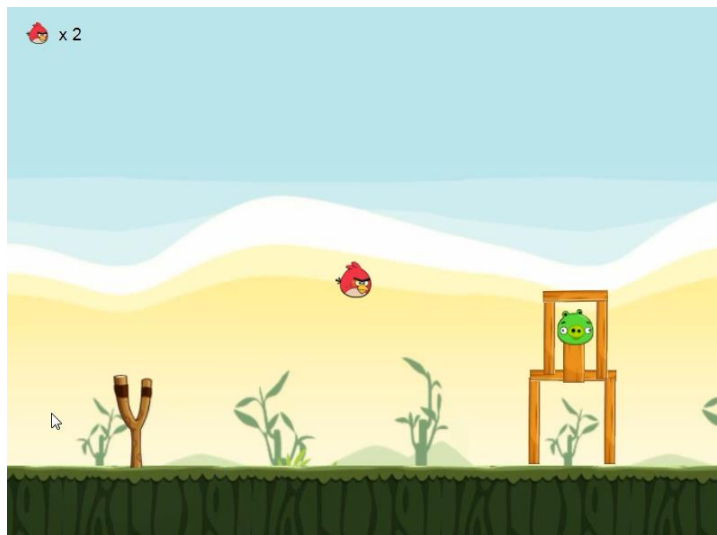
(3) 賭注-遊戲專案: (手機/平板遊玩效果)



(4) 守城-遊戲專案: (手機/平板遊玩效果)



(5) 進階遊戲(參考-憤怒鳥, [Angry Bird](#))



(6) 進階遊戲(參考-馬力歐, Mario)



課程目標：

[全台首創]的面向兒童教授高階程式設計概念:[物件導向程式設計(Object-Oriented Programming, OOP)]的學習課程，透過此課程可完成以下課程目標:

- (1)學習物件導向程式設計(OOP)概念－學會高階的程式設計能力、
- (2)學習 OOP 學習工具與系統操作能力－學會 OOP 工具的使用、
- (3)訓練問題解決與建構遊戲應用能力－運用 OOP 技巧開發遊戲、
- (4)訓練自我創意思考與想法表達能力－學習如何自己介紹作品。

課程特色:

1. **物件導向程設學習-物件導向、高階概念、程設必備:** 有別於現今對於兒童只能開設基本程設概念的學習課程(如: Scratch)，此課程首創開設面向兒童教學的程設必備進階概念:物件導向設計(Object-Oriented Programming, OOP)概念學習課程。
2. **創新圖形學習工具-圖形介面、直覺易用、簡單易懂:** 有別於傳統物件導向課程只能採用文字式程設工具進行教學，此課程首創採用獨有的圖形化學習工具，透過直覺易懂得介面來讓兒童快速理解 OOP 概念與進行實務遊戲製作。
3. **遊戲範例教學實務-遊戲專案、生動有趣、動機加倍:** 有別於傳統物件導向課程教授的枯燥乏味，此課程以遊戲範例作為教學說明，透過圖形化學習工具帶領學生慢慢學會如何建構遊戲範例，進而能自行創作互動遊戲，過程生動有趣，提升學習動機與樂趣。
4. **自創作品訓練表達-融合所學、創意思考、成果發表:** 學員需融合課程技能，動腦思考，運用所學，動手實踐，並勇於將成果作發表，學習與訓練自我表達能力。

開課資訊:

1. **課程地點:** 國立臺南大學、電腦教室 (臺南市中西區樹林街二段 33 號)
2. **招生對象:** 曾經學習過 Scratch 或程式設計相關課程之國中小學生。
3. **招生名額:** 1 班，10-20 人 (10 人以上開班)。
4. **課程時數:** 8 周 (每周 3 小時)，共計 24 小時
5. **上課時間:** 108 年 4 月 13 日(週六)至 108 年 6 月 1 日(週六)，13:30-16:30 上課
6. **上課費用:** 新台幣 4000 元整 (由本校提供講義，“不含”課本及材料，)
7. **網路報名:** <http://academics.nutn.edu.tw/sce/>

課程規劃:

(課程將會不定期更新已達到最佳的學習效果)

週次	課程名稱	課程內容	運算思維能力	OOP 程設概念
第一周	進入 OOP 程式設計的異想世界	1. 相見歡 2. 課程說明(OOP 程式設計) 3. 進行能力測驗 4. 體驗文字式 OOP 語言設計方式 5. OOP 程設初階概念學習：物件(Object) 6. OOP 與運算思維(問題分解)、程式技能(事件、序列)的概念整合 7. 認識 OOP 視覺化學習工具(DiVE) 8. 製作 OOP 遊戲應用專案① 9. 課後測驗遊戲活動(多人對戰遊戲) 10. 問題說明與討論	問題分解 (Decomposition) -問題分解 -問題解析	物件(Object): -屬性(Property) -方法(Method)
第二周	OOP 基礎概念學習與實務製作	1. 課前統整復習活動 2. OOP 程設初階概念學習:類別(Class) 3. OOP 與運算思維(問題分解)、程式技能(運算子)的概念整合 4. 製作 OOP 遊戲應用專案② 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論	問題分解 (Decomposition) -問題分解 -問題解析	類別(Class): -實例化(Instantiation)
第三周	OOP 基礎概念學習與實務製作	1. 課前統整復習活動 2. OOP 程設中階概念學習：類別(Class) 3. OOP 與運算思維(模式辨別)、程式技能(條件)的概念整合 4. 製作 OOP 遊戲應用專案③ 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論	模式辨別 (Pattern Recognition): -模式辨別 -模式一般化	類別(Class): -合成關係(Composition) -繼承關係(Inheritance)
第四周	OOP 中階概念學習與實務製作	1. 課前統整復習活動 2. OOP 程式中階概念學習：封裝(Encapsulation) 3. OOP 與運算思維(模式辨別)、程式技能(迴圈)的概念整合 4. 製作 OOP 遊戲應用專案④ 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論	模式辨別 (Pattern Recognition): -模式辨別 -模式一般化	封裝(Encapsulation): -存取權限(Accessor) -資訊隱藏(Information hiding)

第五周	OOP 中階概念學習與實務製作	<ol style="list-style-type: none"> 1. 課前統整復習活動 2. OOP 程式中階概念學習：繼承 (Inheritance) 3. OOP 與運算思維(抽象化)、程式技能(代碼復用)的概念整合 4. 製作 OOP 遊戲應用專案⑤ 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論 	<p>抽 象 化 (Abstraction):</p> <ul style="list-style-type: none"> -抽象化 -資料蒐集 -資料分析 -資料表示 	<p>繼承(Inheritance)</p> <ul style="list-style-type: none"> - 多重繼承 (Multiple inheritance) - 多層繼承 (Multilevel inheritance)
第六周	OOP 中階概念學習與實務製作	<ol style="list-style-type: none"> 1. 課前統整復習活動 2. OOP 程式進階概念學習：多型 (Polymorphism) 3. OOP 與運算思維(抽象化)、程式技能(資料)的概念整合 4. 製作 OOP 遊戲應用專案⑥ 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論 	<p>抽 象 化 (Abstraction):</p> <ul style="list-style-type: none"> -抽象化 -資料蒐集 -資料分析 -資料表示 	<p>多型(Polymorphism)</p> <ul style="list-style-type: none"> -多載(Overload)
第七周	OOP 進階概念學習與實務製作	<ol style="list-style-type: none"> 1. 課前統整復習活動 2. OOP 程式進階概念學習：多型 (Polymorphism) 3. OOP 與運算思維(演算法設計)、程式技能(平行)的概念整合 4. 製作 OOP 遊戲應用專案⑦ 5. 課後測驗遊戲活動(多人對戰遊戲) 6. 問題說明與討論 	<p>演 算 法 設 計 (Algorithm Design):</p> <ul style="list-style-type: none"> -演算法設計 -模擬 	<p>多型(Polymorphism)</p> <ul style="list-style-type: none"> -覆寫(Override)
第八周	OOP 進階概念學習與實務製作	<ol style="list-style-type: none"> 1. 課前統整總復習活動 2. 製作 OOP 遊戲應用專案⑧ 3. 課後測驗遊戲活動(多人對戰遊戲) 4. 問題說明與討論 	<p>涵蓋所學運算 思維技能</p>	<p>涵蓋所學 OOP 程設概念</p>