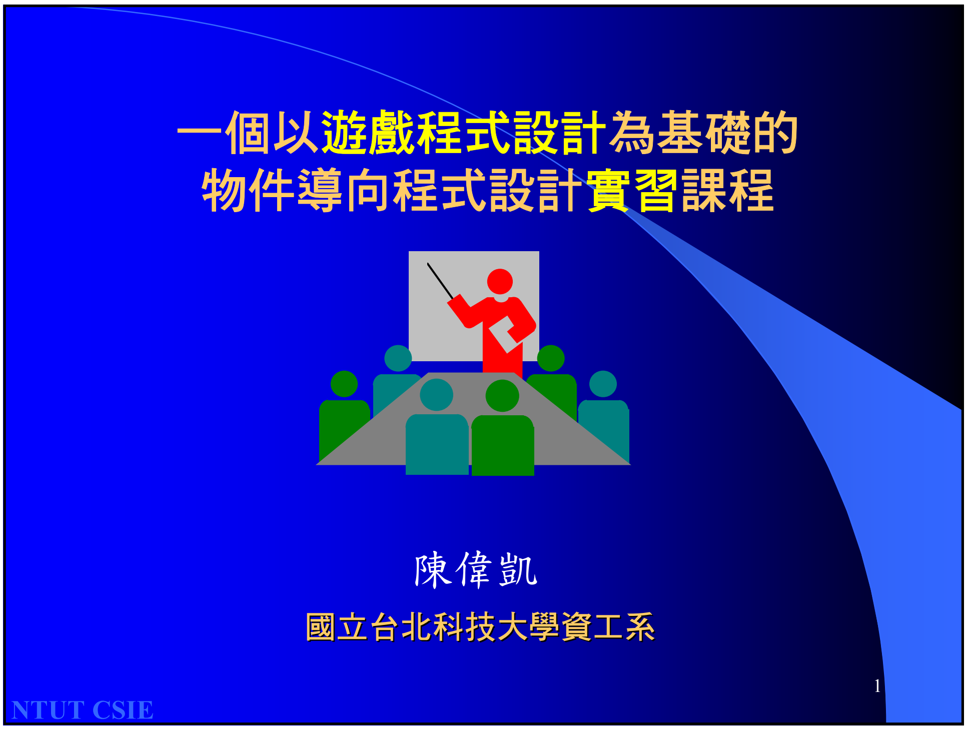


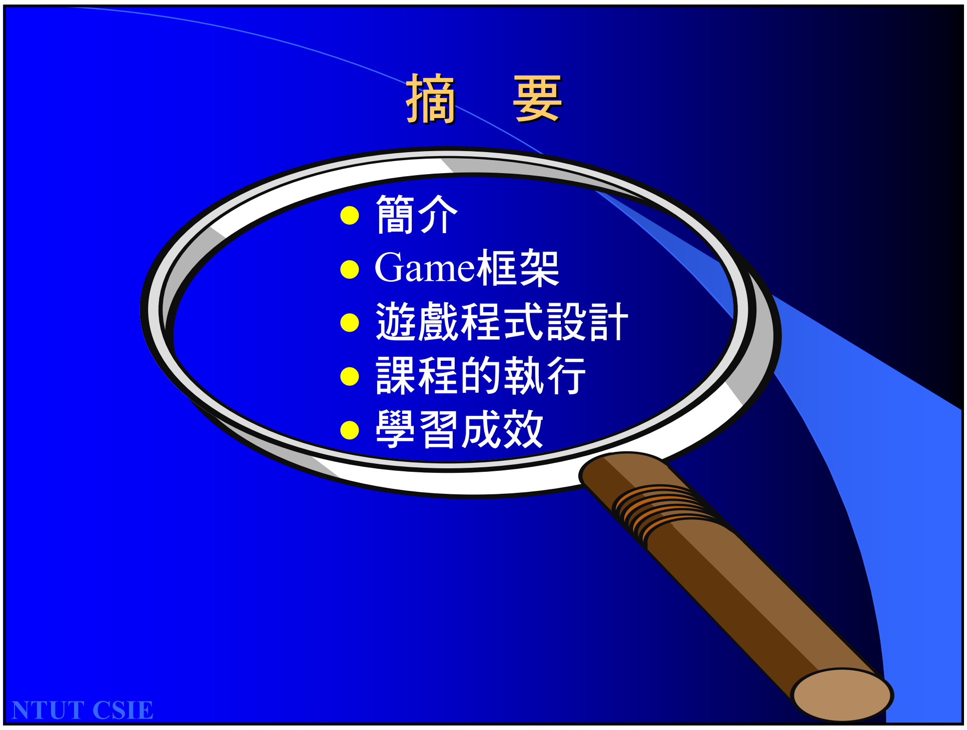
一個以遊戲程式設計為基礎的物件導向程式設計實習課程

陳偉凱

國立台北科技大學資工系

1

NTUT CSIE



摘　要

- 簡介
- Game框架
- 遊戲程式設計
- 課程的執行
- 學習成效

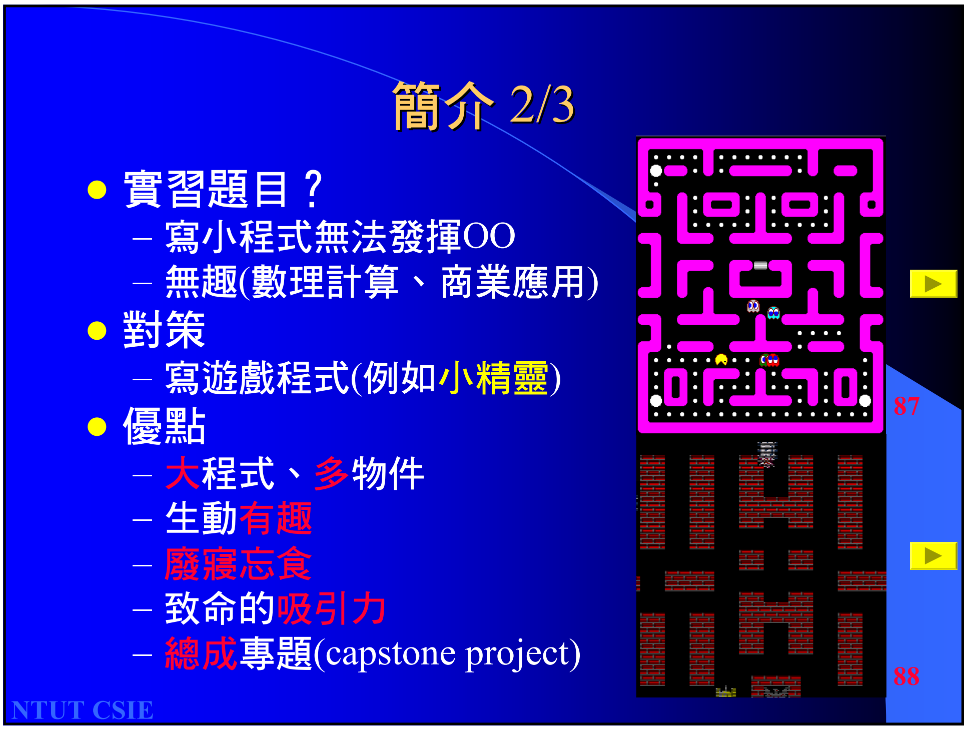
NTUT CSIE

簡介 1/3

- 物件導向程式設計實習(大二)
  - CC 2001
  - 已修完物件導向程式設計
  - 1學分3小時
- 目標
  - C++
  - 熟練OO設計與撰寫的技巧
  - 熟練物件的用法與互動
  - 培養系統的觀念
  - 訓練寫作大型程式的能力

3

NTUT CSIE



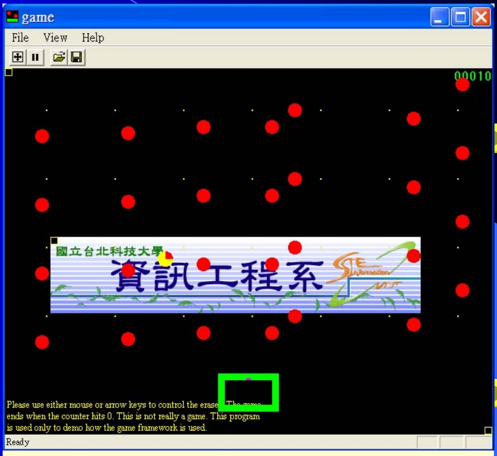
簡介 2/3

- 實習題目？
  - 寫小程式無法發揮OO
  - 無趣(數理計算、商業應用)
- 對策
  - 寫遊戲程式(例如小精靈)
- 優點
  - 大程式、多物件
  - 生動有趣
  - 廢寢忘食
  - 致命的吸引力
  - 總成專題(capstone project)

NTUT CSIE

簡介 3/3

- 問題
  - 時間夠嗎？
  - 能力夠嗎？
  - 成果好嗎？
  - 繪圖？
  - 音效？
  - 速度？
- Paper (IEEE Transactions on Education; SCI/SSCI)



Please use either mouse or arrow keys to control the erase The game ends when the counter hits 0. This is not really a game. This program is used only to demo how the game framework is used.

Game Framework

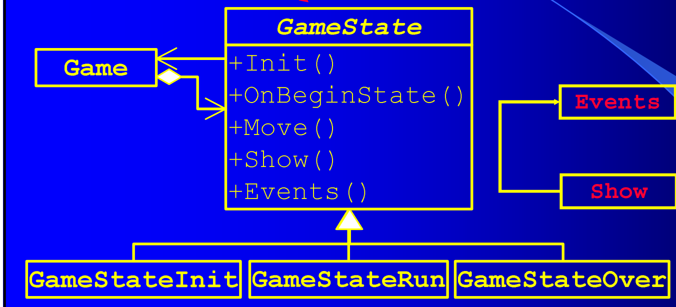
Game框架—簡介 1/5

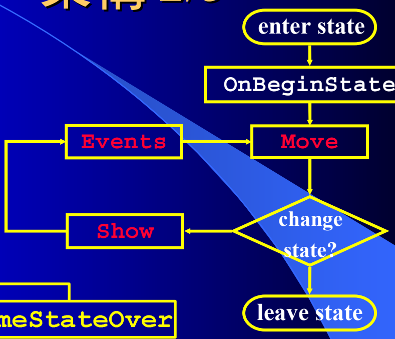
- **Game framework**
  - 1998年開始
  - 2D games
  - GPL
  - VC++/MFC
  - DirectX
  - Library → Framework
- 功能
  - 簡易的(點陣圖)繪圖物件
  - 簡易的音效物件(wave與midi)
  - 全螢幕/視窗
  - 遊戲狀態性的控制
  - 遊戲迴圈速度的控制

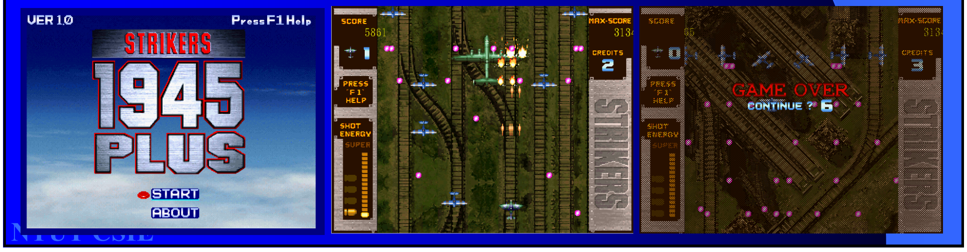
Please use either mouse or arrow keys to control the erase The game ends when the counter hits 0. This is not really a game. This program is used only to demo how the game framework is used.

Game框架—架構 2/5

State Pattern







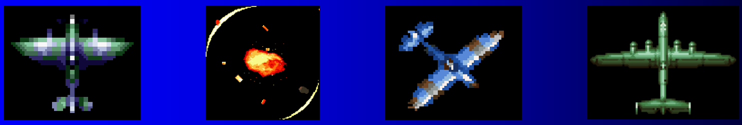
Game框架—圖形、聲音 3/5

CMovingBitmap

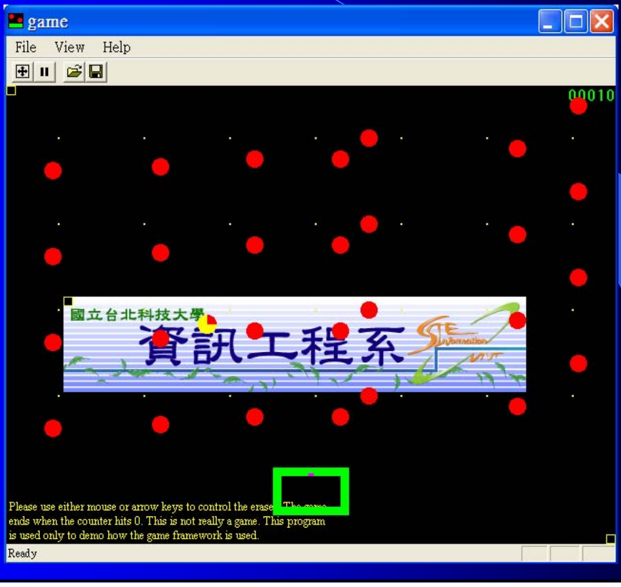
+LoadBitmap()
+SetTopLeft()
+ShowBitmap()
+...()

CAudio
+Instance()

+Load()
+Play()
+Pause()
+Resume()
+Stop()



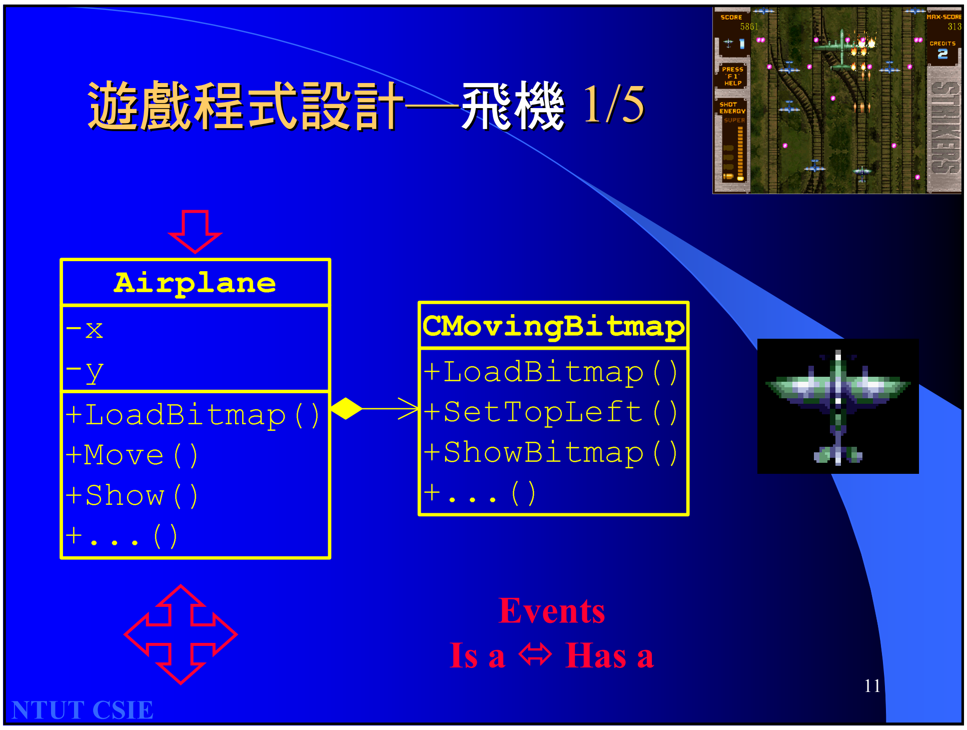
Game框架 4/5



Please use either mouse or arrow keys to control the erase The game ends when the counter hits 0. This is not really a game. This program is used only to demo how the game framework is used.

Game框架— 輔助教材5/5

- Introduction to Game Framework
- Tutorial #1: Prepare a Bitmap
- Tutorial #2: Display a Bitmap
- Tutorial #3: Move a Bitmap
- Tutorial #4: Make it an object
- Tutorial #5: A sample map object
- Game deployment (create setup file)

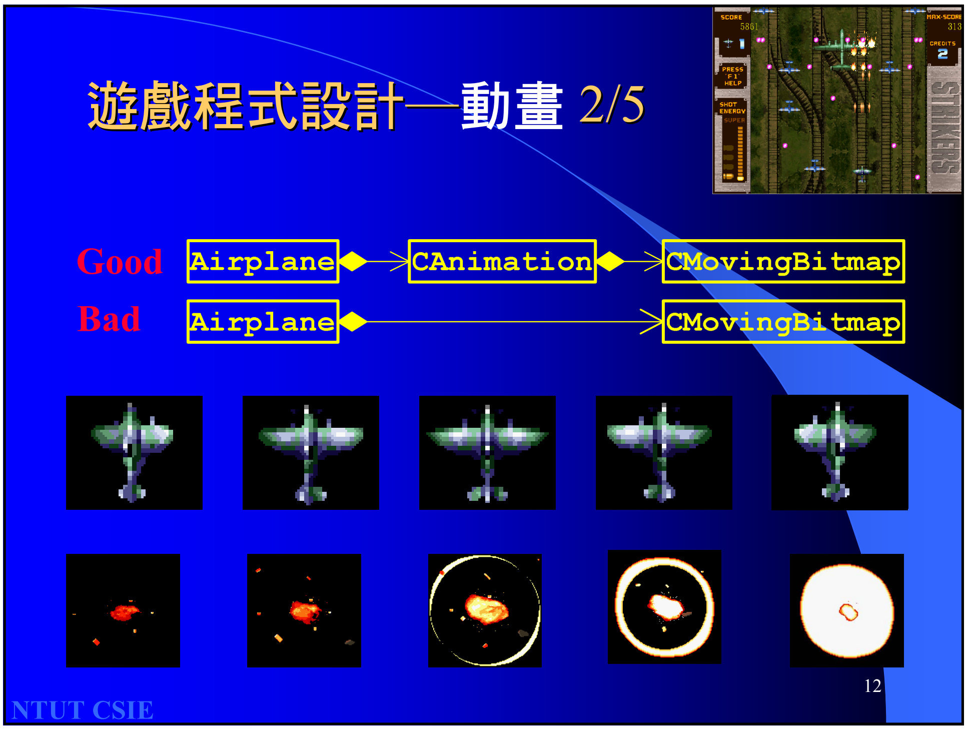


遊戲程式設計—飛機 1/5

Airplane
-x
-y
+LoadBitmap()
+Move()
+Show()
+...()

CMovingBitmap
+LoadBitmap()
+SetTopLeft()
+ShowBitmap()
+...()

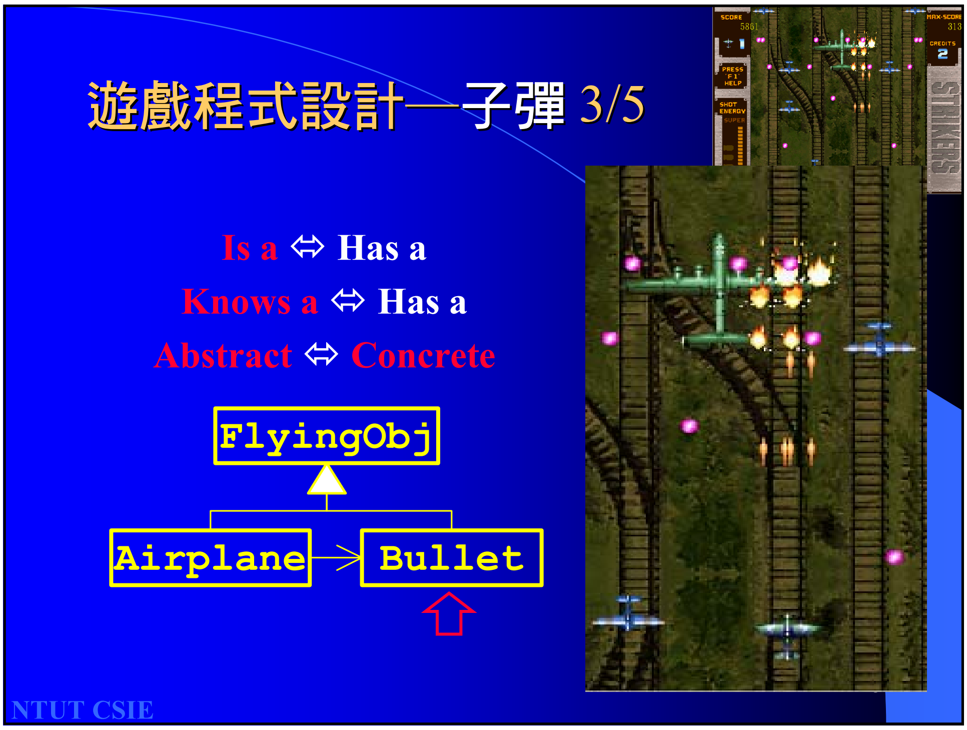
Events
Is a ⇔ Has a



遊戲程式設計—動畫 2/5

Good Airplane → CAnimation → CMovingBitmap

Bad Airplane → CMovingBitmap



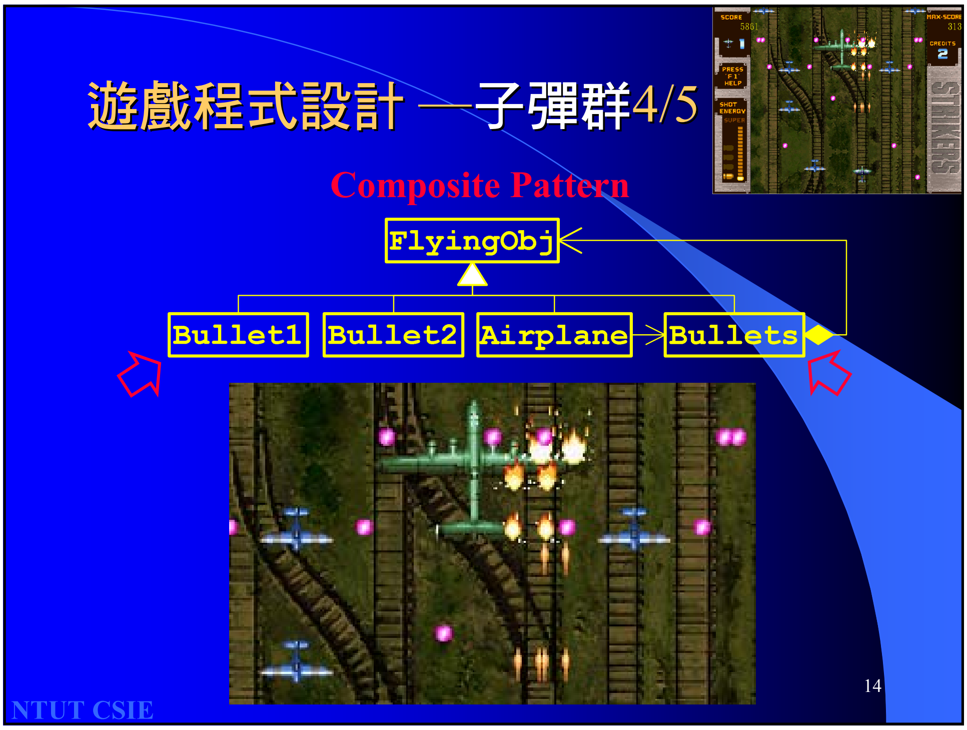
遊戲程式設計—子彈 3/5

Is a ⇔ Has a

Knows a ⇔ Has a

Abstract ⇔ Concrete

NTUT CSIE

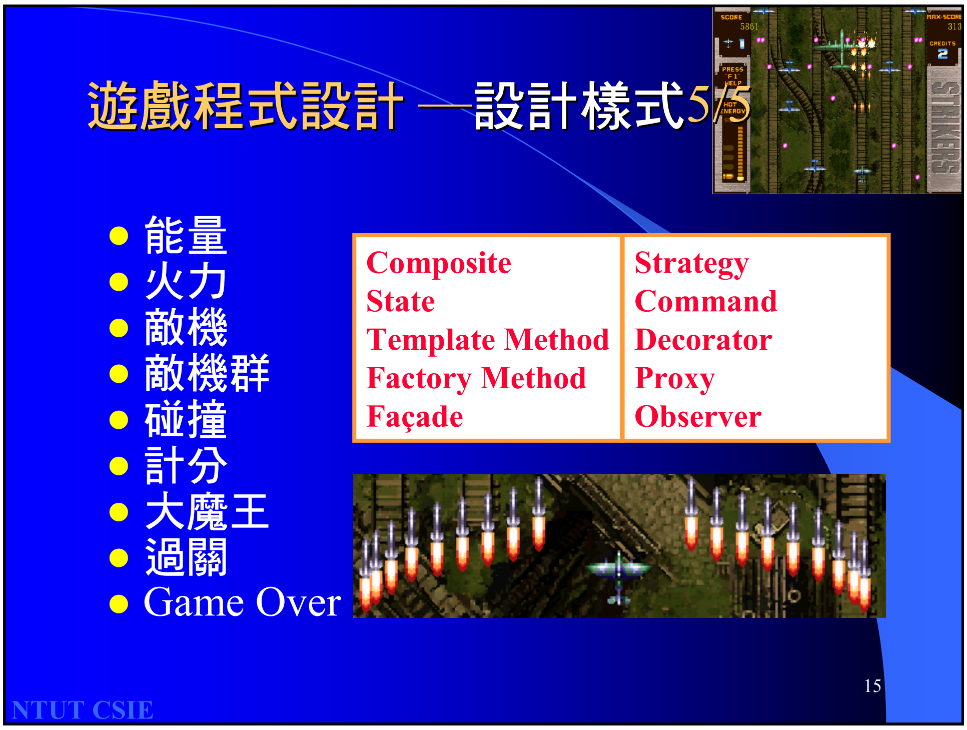


遊戲程式設計 —子彈群4/5

Composite Pattern

14

NTUT CSIE



遊戲程式設計 —設計樣式5/5

- 能量
- 火力
- 敵機
- 敵機群
- 碰撞
- 計分
- 大魔王
- 過關
- Game Over

Composite State Template Method Factory Method Façade	Strategy Command Decorator Proxy Observer

NTUT CSIE

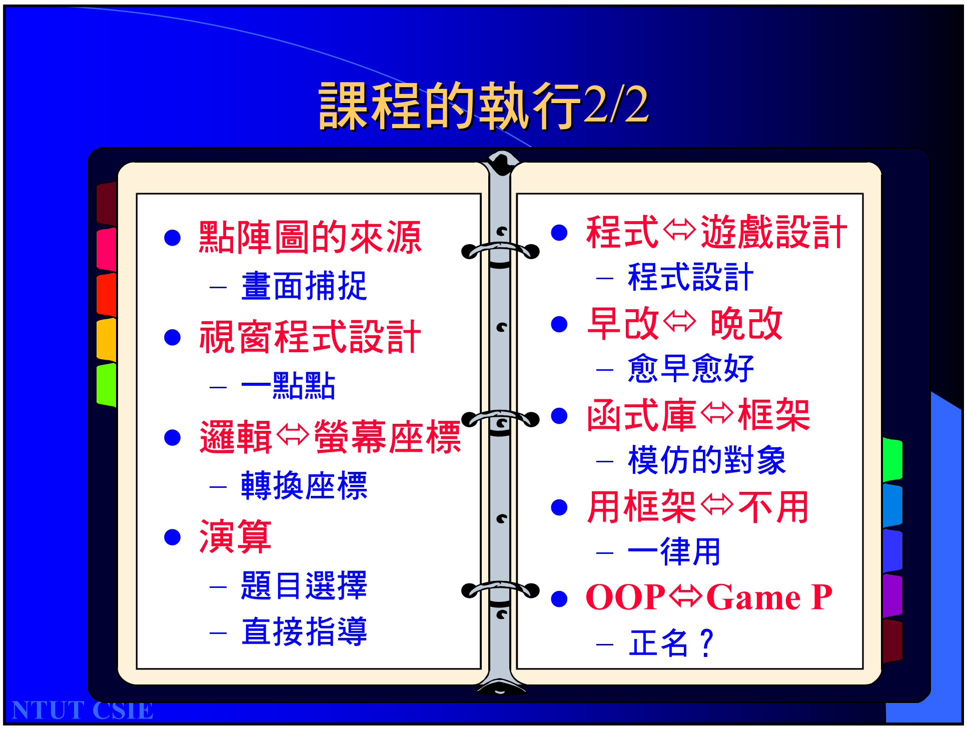
15

課程的執行1/2

- 分組
  - 2人
- 目標訂定
  - 2週
- 目標確認
  - 第3週前
- 一對一指導
  - 每隔週
- 進度的追蹤

- 助教
  - 學長
- 期中考核
  - 1或2次
- 評分
  - 報告
  - 平時
  - 成果
- 比賽

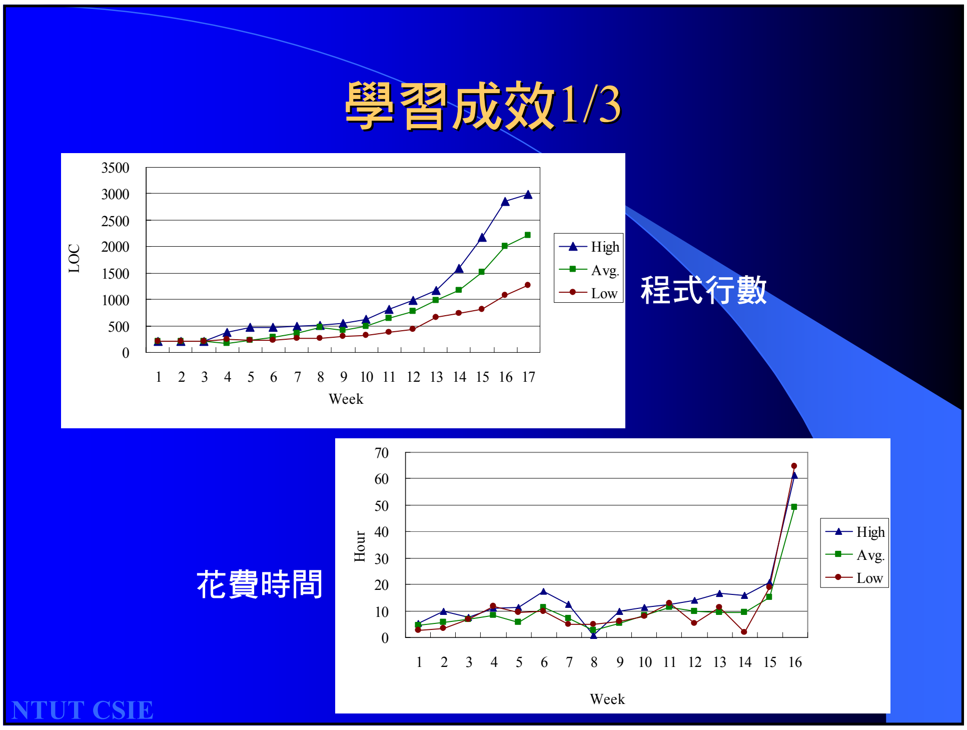
NTUT CSIE



課程的執行2/2

- 點陣圖的來源
  - 畫面捕捉
- 視窗程式設計
  - 一點點
- 邏輯⇔螢幕座標
  - 轉換座標
- 演算
  - 題目選擇
  - 直接指導
- 程式⇔遊戲設計
  - 程式設計
- 早改⇔ 晚改
  - 愈早愈好
- 函式庫⇔框架
  - 模仿的對象
- 用框架⇔不用
  - 一律用
- OOP⇔Game P
  - 正名？

NTUT CSIE

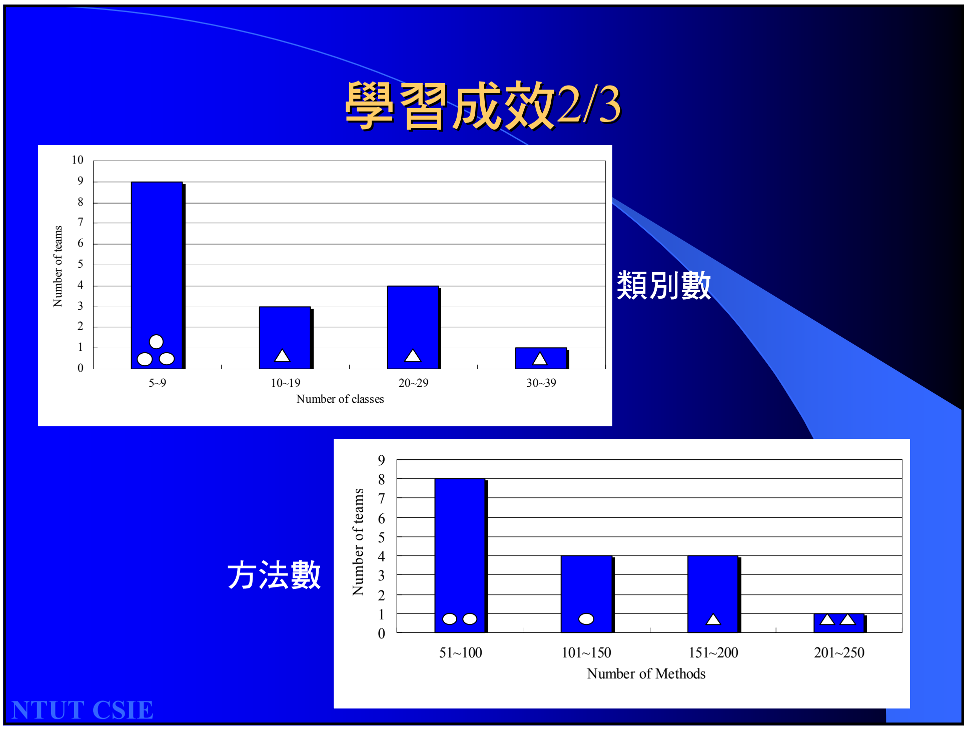


學習成效1/3

程式行數

花費時間

NTUT CSIE

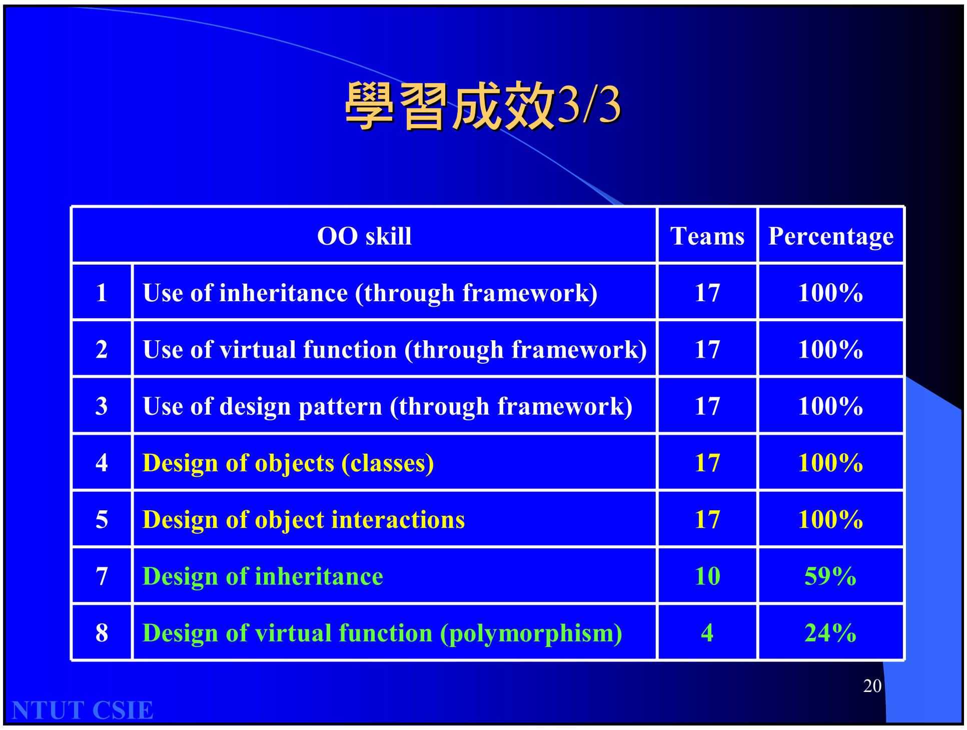


學習成效2/3

類別數

方法數

NTUT CSIE



學習成效3/3

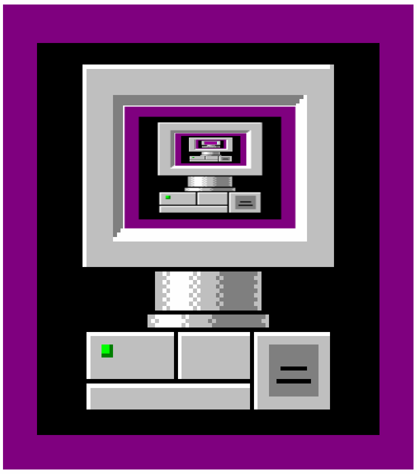
OO skill		Teams	Percentage
1	Use of inheritance (through framework)	17	100%
2	Use of virtual function (through framework)	17	100%
3	Use of design pattern (through framework)	17	100%
4	Design of objects (classes)	17	100%
5	Design of object interactions	17	100%
7	Design of inheritance	10	59%
8	Design of virtual function (polymorphism)	4	24%

20

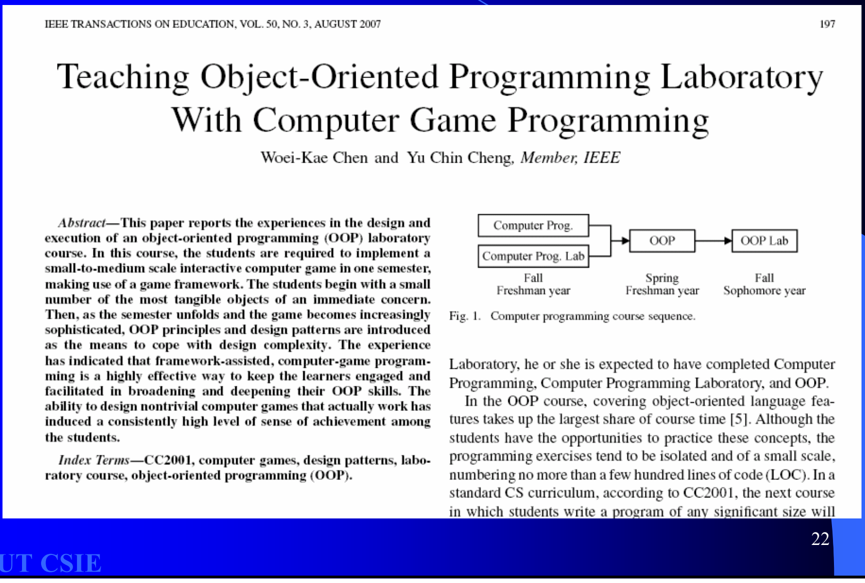
NTUT CSIE

結論

- 教學
  - 激發原動力
  - 深入的教學內涵
  - 滿足個別需要
- OOP Lab with Game
  - 動力與成就感
  - 強化物件教學的深度
  - 非齊頭式的教學



IEEE Transactions on Education 2007



IEEE TRANSACTIONS ON EDUCATION, VOL. 50, NO. 3, AUGUST 2007 197

Teaching Object-Oriented Programming Laboratory With Computer Game Programming

Woei-Kae Chen and Yu Chin Cheng, *Member, IEEE*

***Abstract*—This paper reports the experiences in the design and execution of an object-oriented programming (OOP) laboratory course. In this course, the students are required to implement a small-to-medium scale interactive computer game in one semester, making use of a game framework. The students begin with a small number of the most tangible objects of an immediate concern. Then, as the semester unfolds and the game becomes increasingly sophisticated, OOP principles and design patterns are introduced as the means to cope with design complexity. The experience has indicated that framework-assisted, computer-game programming is a highly effective way to keep the learners engaged and facilitated in broadening and deepening their OOP skills. The ability to design nontrivial computer games that actually work has induced a consistently high level of sense of achievement among the students.**

***Index Terms*—CC2001, computer games, design patterns, laboratory course, object-oriented programming (OOP).**

Fig. 1. Computer programming course sequence.

Laboratory, he or she is expected to have completed Computer Programming, Computer Programming Laboratory, and OOP.

In the OOP course, covering object-oriented language features takes up the largest share of course time [5]. Although the students have the opportunities to practice these concepts, the programming exercises tend to be isolated and of a small scale, numbering no more than a few hundred lines of code (LOC). In a standard CS curriculum, according to CC2001, the next course in which students write a program of any significant size will