CDI020080299

Using Software Packages to Teach Programming in Computer Subjects

在電腦科中使用套裝軟件教授程式編寫

13 May 2008

Program Rundown

1400 – 1415	Introduction
1415 – 1500	Flowcharting
1500 – 1545	Game Programming
1545 – 1600	Break
1600 – 1645	Robot Programming
1645 – 1700	Q & A

Objectives

To raise computer teachers’ awareness on using software packages to support the learning and teaching of programming

1. Introduction of different types of software packages:
   - Flowcharting
   - Robot programming
   - Game programming
2. Demonstration of the software packages
3. Examples on using the software packages to teach programming in computer subjects

Programming in Computer Subjects

Curriculum		Time Allocation
Computer Literacy (S1 – S3) *School may opt either Logo or any other computer language for teaching programming*		17%
CIT (S4 – S5)	Core Module 4. Basic Programming Concepts	18 hours
	Elective Module (A) Algorithm & Programming	30 hours
NSS ICT (S4 – S6)	Compulsory Part D. Basic Programming Concepts	20 hours
	Elective Part D. Software Development	75 hours

Programming Software to be introduced

- Coding is NOT required
- Both commercial and freeware
- Targeted curriculum
  - CL
  - Core module of CIT
  - Compulsory part of ICT

Flowcharting

Challenges in
Teaching CIT Core Module
4. Basic Programming Concepts

- How to teach flowchart?
- Should I use programming languages to teaching this topic?
- If not, what should I use?

Challenges in
Teaching CIT Core Module
4. Basic Programming Concepts

- Question: What should be taught in this topic?
  - Syntax of the programming language?
  - Problem solving / logical thinking skills?

CROCODILE ICT

Crocodile ICT

- Flowcharting Software
- Commercial Software
- http:// www.crocodile-clips.com/

Demonstrations / L & T Examples

Built-in Examples

- Includes:
  - Clock
  - Burglar Alarm
  - Heating System
  - Greenhouse
  - Fire Alarm
  - Elevator
  - Driving Game
  - Maze
  - …………..

Built-in Examples

- Covers:
  - Flowchart notations
  - Input/Output/Variable value assignment
  - Branch structure
  - Iteration structure
  - Infinite loop
  - Tracing program flow
  - Modularity
  - Debugging of logical error

Crocodile ICT

- Flowcharting Software
- Commercial Software
- http://www.crocodile-clips.com/

Sub-topics Covered in Basic Programming Concepts

- **Problem solving procedures (CIT C&A Guide P.19)**
  - **Real life application of the problem solving procedures**
  - **Using a program flowchart to realize the algorithm**

Sub-topics Covered in Basic Programming Concepts

- **Programming concepts in using a high level programming language (CIT C&A Guide P.19)**
  - **Input, output, assignment of values to variables**
  - **Using constructs such as branching and iteration structures**
  - **Tracing program flow: Identifying values of variables and debugging logical errors**

ICT Module Covered

Compulsory Part:

D. Basic Programming Concepts

Topics Covered in Basic Programming Concepts

- a. Problem-Solving Procedures (ICT C&A Guide P.28)
  - Solve a problem by breaking it down into sub-problems or modules
- b. Algorithm Design (ICT C&A Guide P.28)
  - Design an appropriate user interface
  - Data types and data structures (integer, real, character, Boolean, string, 1D array)
  - Control structure (Sequence, selection, iteration)
  - Modularity
- c. Algorithm Testing (ICT C&A Guide P.29)
  - Trace & test algorithm

Crocodile ICT- Features

- Easy to use
- Adequate examples, tutorial and teaching resources that vary from different levels of difficulty
- Some built-in examples are presented in a step-by-step manner
- Resources are highly correlated with CIT, ICT and CL curriculum
- Can carry out dry run with adjustable speed

Crocodile ICT-Limitations

- Not free
- English interface only
- Cannot export/compile
- Some dance motions are too fancy to be used as demonstrations
- The interface shown in the tutorial is a bit different from the actual software
- Limited support on the print function
- Most iteration examples are of infinite loop

Crocodile ICT Player

- Flowcharting software
- Trimmed down version of Crocodile ICT
- Free
- Provides very limited functions
- Download:
  - http://www.crocodile-clips.com/

RAPTOR

Raptor

- Rapid Algorithmic Programming Tool for Ordered Reasoning (RAPTOR)
- Flowcharting Software
- Free
- http://raptor.martincarlisle.com/

Demonstrations / L & T Examples

CIT Topic Covered

Core Module:

4. Basic Programming Concepts

Sub-topics Covered in Basic Programming Concepts

- **Problem solving procedures (CIT C&A Guide P.19)**
  - Real life application of the problem solving procedures
  - Using a program flowchart to realize the algorithm
  - Different ways to solve the same problem and their brief comparison

Sub-topics Covered in Basic Programming Concepts

- **Programming concepts in using a high level programming language**
  (CIT C&A Guide P.19)
  - **Input, output, assignment of values to variables**
  - **Using constructs such as branching and iteration structures**
  - **Tracing program flow: Identifying values of variables and debugging logical errors**

ICT Module Covered

Compulsory Part:

D. Basic Programming Concepts

Topics Covered in Basic Programming Concepts

- a. Problem-Solving Procedures (ICT C&A Guide P.28)
  - Solve a problem by breaking it down into sub-problems or modules
- b. Algorithm Design (ICT C&A Guide P.28)
  - Design an appropriate user interface
  - Data types and data structures (integer, real, character, Boolean, string, 1D array)
  - Control structure (Sequence, selection, iteration)
  - Modularity
- c. Algorithm Testing (ICT C&A Guide P.29)
  - Trace & test algorithm

Raptor - Features

- Can carry out dry run with adjustable speed
- Can export to “pseudocode” with the following syntax: Ada, C#, C, Java
- Advanced features: subchart, drawing graphs, sound, array, file, event (mouse, keyboard) etc
- Can find most reference in Help (Especially Reference > Reference)
- Support sound, vector graphic, file operation

Raptor - Limitations

- English Interface only
- Limited L&T resources
- Flowchart symbols are different from commonly used ones

Comparison

	Crocodile ICT	Raptor
Cost	Not Free	Free
Language	English	English
Built-in examples	Adequate	Limited
Help	Clear	Clear

Comparison

	Crocodile ICT	Raptor
GUI	Fancy	Limited
Pseudocode Translation	Nil	Ada, C#, C, Java
Variable Types	Number, String, Boolean, Map (1D Array), List	Number, String, 1D Array, 2D Array, Boolean
User Friendliness	Good	Fair
Target Curriculum	CIT, ICT	CIT, ICT

Game Programming

WHY Teaching Game Design?

- Great interest and motivation of students
- Problem-solving skills
- Creativity
- Project learning
- Cross-curriculum learning

THE GAMES FACTORY 2

The Games Factory 2

- Game Design Software (can also develop screensaver and application)
- Commercial Software
- http://www.clickteam.com/eng/index.php

Demonstrations / L & T Examples

The Games Factory 2 - Features

- – Sufficient built-in objects and functions
- – Drag-and-drop, point-and-click
- – Libraries with rich resources
- – Picture and animation editor
- – Useful tutorials provided in web site
- – Sufficient examples
    - Step-by-step tutorial : 1
    - Sample Objects: 23
    - Games: 16
    - Application: 1

The Games Factory 2 - Limitations

- English Version Only
- No scripting
- Examples are too complicated for beginners

GAME MAKER

Game Maker

- Game Design Software
- Freeware
- http://www.yoyogames.com/

Demonstrations / L & T Examples

Game Maker - Features

- Sufficient built-in functions
- Drag-and-drop
- Allow coding for advanced users
- Picture and animation editor
- Tutorials with “Game Development” approach
  - Game design factors
  - Design document
  - Semi-finished products

Game Maker - Limitations

- English Version Only
- Examples are too complicated for beginners
- Tutorial documents skip minor steps

Comparison

	The Games Factory 2	Game Maker
Cost	Not Free	Free
Language	English	English
User Friendliness	Good	Fair
Product	Game, application	Game
Coding	Not Allowed	Allowed
Tutorial/ Help	Step-by-step tutorial	Game development approach

Game Design Factors

- Goals & sub-goals
- Decision making
- Balance (between different features, between players, ability vs challenge)
- Rewards
- Immersion (story, main characters, music, special effects, etc.)

Teaching Game Design

- Design document (description, game objects, sound, control, game flow, levels, etc.)
- Create objects & properties
- Define event and action
- Add music and sound effects
- Calculate score, lives
- Modify with challenges and levels

Teaching Game Design

- Object oriented (non procedural) programming
- Event driven approach
- Game making skills:
  - concept of frame/room, event/condition, action
  - object properties, object controller
  - list of events & actions *(this part can let students explore and discover by themselves more)*

Teaching Game Design

Typical events/conditions:

- Collision between objects
- Object leaves or enters the frame/room
- User clicks an object, or a specific zone
- User presses a key
- "Start of application" conditions occur
- The timer reaches a specific value
- Comparison conditions
- Object movement
- Counter object conditions

Teaching Game Design

Typical actions:

- Object movement, creation or destruction
- Sound playing or stopping
- Set alarm/timer
- Lives or score changing
- Jump to the next frame, the previous frame, restart the current frame, and end the application

Implementation

- Computer Literacy Curriculum
- Game design course in CCA activity
- Elite training for competition
- Cross curriculum project learning

CL Topics Covered

- Introducing the working environment (PA01)
- Executing, saving and retrieving programs (PA01)
- Ideas of programming (PA01)
- Basic input/output techniques (PA02)
- Arithmetic operations (PA02)
- Manipulating text (PA03)
- Screen control and sound (PA03)
- Ideas of flow control (PA11)
- Uses of flow control (PA11)
- Random number (PA12)
- Designing a programming project (PA13)
- Project assignment (PA13)

CL Topics Not Covered

- Looping technique (PA12)

Cross Curriculum Project Learning

Cross curriculum project learning elements such as:

- – Game Design (CL)
- – Graphic Design (Visual Art)
- – Music & Sound Effect Design (Music)
- – Game Promoting and Marketing (Business)
- – Game Packaging (D&T)
- – Project Management (CL, Business)
- – Issues of addiction, health, violence, intellectual property (CL, Liberal Studies)
- – *Cultivation of generic skills (problem solving, creativity, IT, communication, collaboration, critical thinking, study)*

Break

Robot Programming

IQ-Bug

IQ-Bug

- Robot Hardware (JMC-BG-2000) + Software
- Commercial Software
- http://www.robotplayer.com/

Demonstrations / L & T Examples

CL Topics Covered

- Introducing the working environment (PA01)
- Executing, saving and retrieving programs (PA01)
- Ideas of programming (PA01)
- Basic input/output techniques (PA02)
- Screen control and sound (PA03)
- Ideas of flow control (PA11)
- Uses of flow control (PA11)
- Looping technique (PA12)
- Designing a programming project (PA13)
- Project assignment (PA13)
- Introducing control technology (IT11)
- Features of a control system (IT11)
- Project with control technology (IT11)

CL Topics Not Covered

- Arithmetic operations (PA02)
- Manipulating text (PA03)
- Random number (PA12)

IQ-Bug - Features

- Can install both the English and Chinese versions on the same machine
- Attractive to students
- Good for performing programming projects for CL (PA13)
- Students can VIEW the C source code

IQ-Bug - Limitations

- Poor user interface. E.g.
  - *Do not support undo*
  - *When moving boxes, sometimes arrows will tangle together*
- Limited L&T resources
- A number of typos and grammatical mistakes can be found in the English user menu
- The “Help” section in the English software is written in Chinese
- Limited built-in examples
- The “Program Guide Book” does not provide detailed examples in advanced topics
- Symbols are different from traditional flowchart

IQ-Bug Explorer

- Robot controlling software
- Comes with IQ-Bug
- Provide a GUI for IQ-Bug control
- No programming is needed

ROBOTPROG

Robotprog

- Robot Programming + Flowcharting Software
- Free
- http://www.physicsbox.com/supportrobotprogen.html

Demonstrations / L & T Examples

CL Topics Covered

- Introducing the working environment (PA01)
- Executing, saving and retrieving programs (PA01)
- Ideas of programming (PA01)
- Basic input/output techniques (PA02)
- Arithmetic operations (PA02)
- Ideas of flow control (PA11)
- Uses of flow control (PA11)
- Looping technique (PA12)
- Designing a programming project (PA13)
- Project assignment (PA13)
- Introducing control technology (IT11)
- Features of a control system (IT11)
- Project with control technology (IT11)

CL Topics Not Covered

- Manipulating text (PA03)
- Screen control and sound (PA03)
- Random number (PA12)

Robotprog - Features

- Focus more on programming instead of mechanics
- The “Help” section contains a list of built-in functions
- The online tutorial contains materials for self learning
- Can carry out dry run with adjustable speed

Robotprog - Limitations

- English Interface only
- Limited L&T resources
- Flowchart symbols are different from commonly used ones
- Data type supported: Integer only
- Do not support undo

Comparison

	IQ-Bug	Robotprog
Cost	Not Free	Free
Language	English+Chinese	English
Built-in examples	Limited	Limited
Help	Fair	Clear
User Friendliness	Poor	Fair

Comparison

	IQ-Bug	Robotprog
Code Translation	C	Nil
Variable Types	Int, char, long, unsigned int, unsigned char, unsigned long, double	Integer
Target Curriculum	CL	CL

Concluding Remarks

- Learning programming ≠ Learning syntax
- Keyword: Problem Solving
- Many alternatives in teaching programming
- Opt the most suitable one for your students

Learning programming can be fun!!!

Question
&
Answer

Please help by filling in the evaluation form