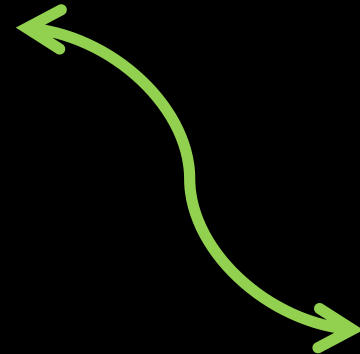


How to catch code! – Taking the pain out of the new computing curriculum



Max Wainwright – ICT Apps
Twitter@ictapps
Dr David Whytey – Whytek Consulting
Twitter@davewhy
www.whytekconsulting.co.uk

We hope to share with you...

- How to cope with the new Computing curriculum
- Share the range of software available
- Look at working away from the computer
- Explore the range of skills that can be learned in an inclusive fun experience
- Discuss how teaching Computing might be organised
- Share classroom approaches and examples
- Look at a simple progression of learning
- Hand on with a new resource



Before we begin - Why bother with computing?

The New Curriculum says so

The Games Industry wants more programmers

Good for Maths

Children need to know how computers work

Will help cover non contact time

Can develop thinking skills

Provide increased job prospects

Computing – Then..... and..... Now

**How many Computer Programmers
were needed In 1946?**



**How many Computer Programmers are
needed now?**



Why do we need to...

www.codestars.org

So who invented computer programming then?

- 1949 – A2 Compiler
- 1957 – FORTRAN
- 1958 – ALGOL
- 1959 - COBOL – Includes Input and Output
- 1964 – BASIC
- 1966 - LOGO
- 1970 – Forth – uses sequences of words
- 1971 - PASCAL – uses blocks of code
- 1972 - Prolog
- 1972 – C – For writing drivers
- 1975 – Microsoft BASIC
- 1982 – Foundation of ADOBE and PostScript
- 1983 C++ - Object oriented language
- 1987 Perl –
- 1991 – Java
- 1991 – Python
- 1991 – Visual Basic
- 1994 - Java Script
- 2001 - C#
- 2001 – Visual Basic.net



Grace Hooper

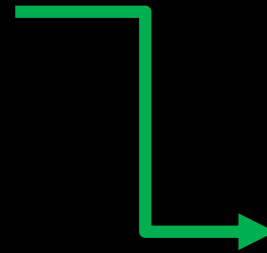
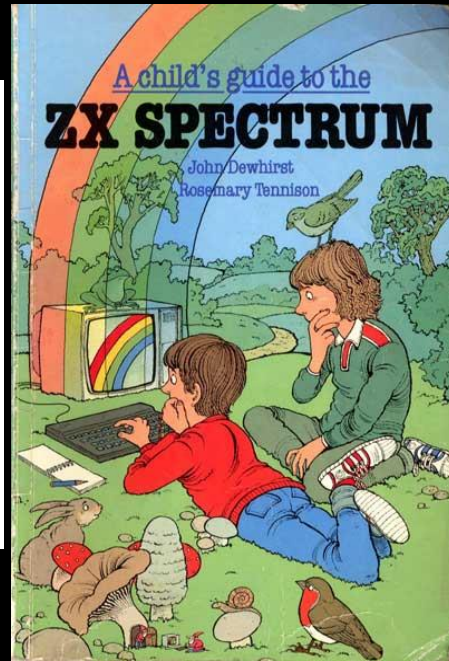
Known for her discovery of a moth in a relay of the Mark I computer in 1945 which lead to the term "computer bug" Invented the "Compiler"

I want children to be able to experience the beauty of coding and joy of creating - (Simon Peyton-Jones Microsoft Research Labs – March 2013)

“A High Quality Computing Education.....develops and requires logical thinking and precision.

It combines creativity with rigour”

```
310 LET A=USR 18255
320 IF INKEY$="" THEN GOTO 320
330 POKE 17901,INT (RND*128)
335 IF PEEK 21623(>178 THEN POK
E ((PEEK 16396+256*PEEK 16397)+0
0)
340 POKE 16522,CODE INKEY$
350 LET A=USR 18224
360 IF PEEK 16519>=120 THEN GOT
O 3000
370 FOR N=0 TO 5
380 NEXT N
390 IF PEEK (PEEK 16514+17920) <
>45 THEN GOTO 330
400 FOR N=0 TO 30
405 POKE 17901,INT (128*RND)
410 LET A=USR 18224
420 FOR M=0 TO 3
425 NEXT M
430 NEXT N
440 CLS
450 GOTO 2520
370 FOR N=0 TO 0
```



Max on the New Curriculum

Nursery to KS4

Nursery and reception

<http://primo.io> c£170 early 2015



KS4

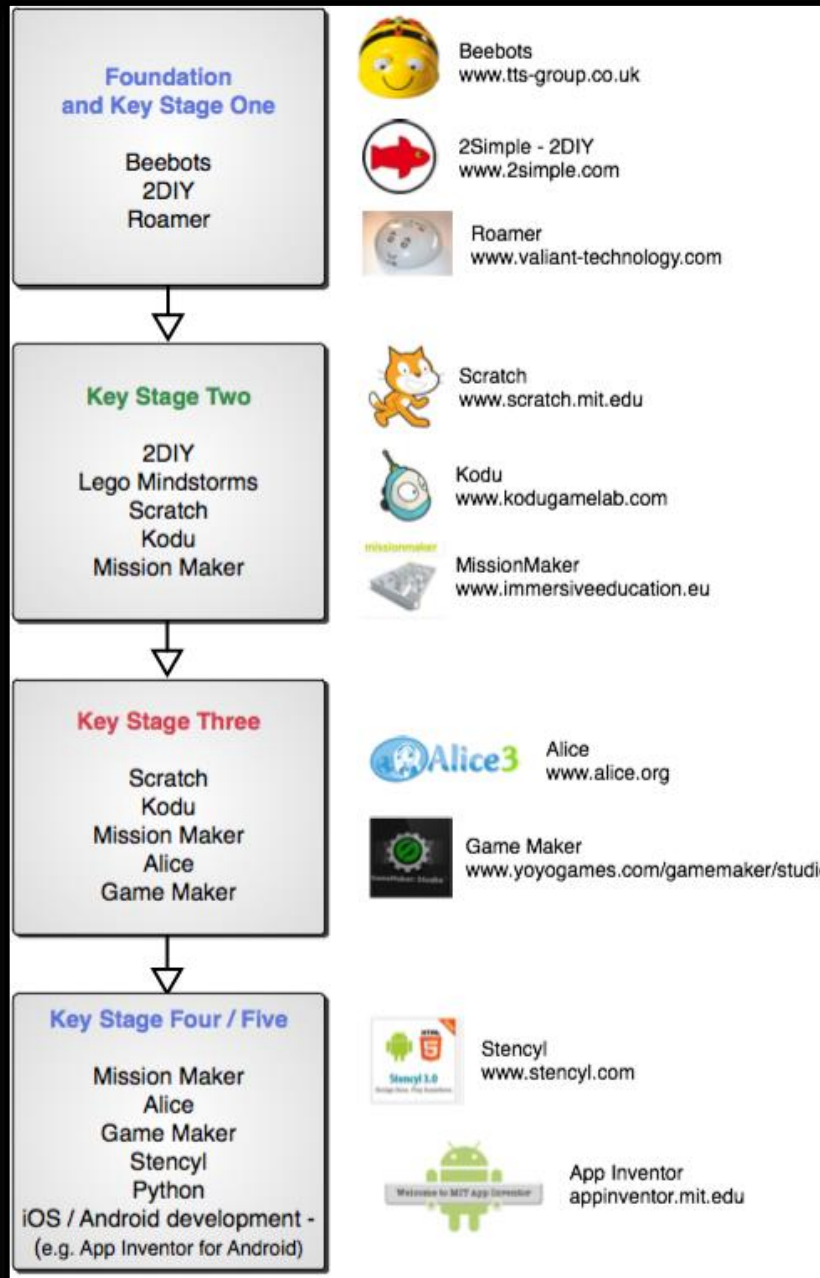
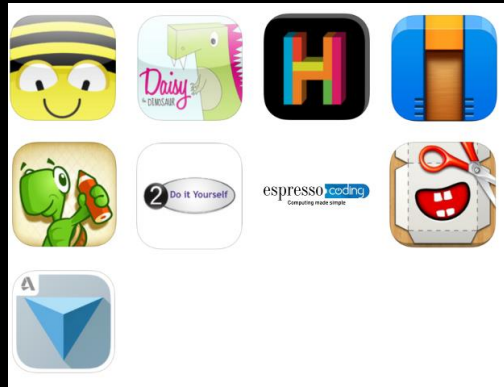
Microsoft Touch Develop

<http://teachwithict.weebly.com/>



Possible software progression

Transitional arrangements?



Richard Anderson
www.lttonline.net



How will you manage the teaching of computing?



Wolverhampton Experience - SCRATCH

PRO's

- Pupils can import their own backgrounds and characters
- Easy to “See” the code
- Algorithms are built up in blocks
- Can be used to program other than games



CON's

- More complex for teachers to become familiar with
- Requires more understanding of coding
- Less professional results are possible

Wolverhampton Experience - KODU

PRO's

- Easy interface with object oriented programming
- Professional looking 3D game
- Easy to add complexity
- Can use games controller
- Kodu Kup



CON's

- Have to use pre set characters and backgrounds
- Limits the type of coding you can do
- Games have yet to be made available to play on XBOX

Wolverhampton Experience – Higher Order Learning

HOTS

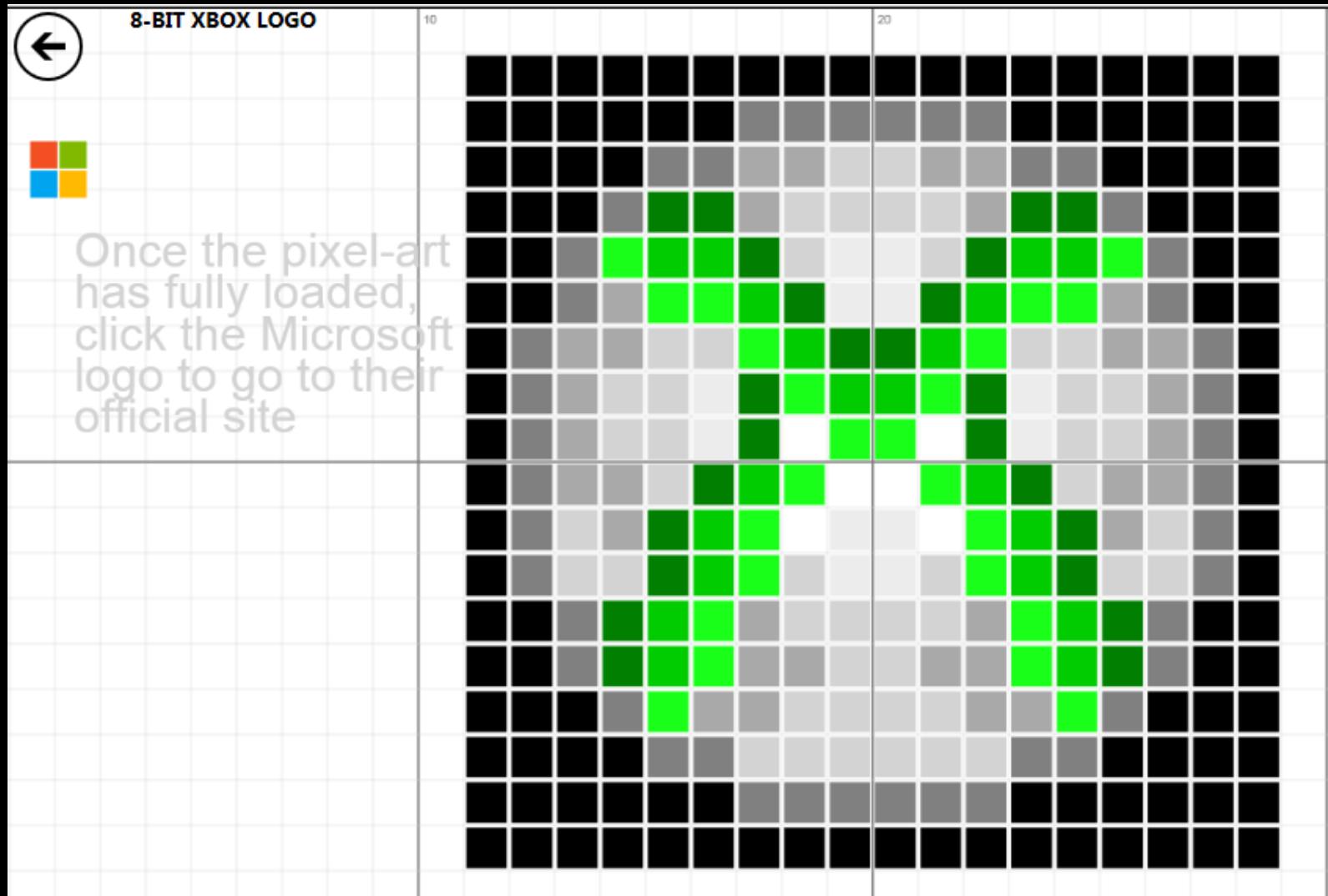
- Creating
- Hypothesise
- Solve
- Logical
- Evaluate
- Analyse
- Explain
- Challenge
- Fairness



T&L

- Collaboration
- Language development
- Extended learning
- Authentic Activities
- Easy to Differentiate
- Engages pupils with challenging behaviour

Key Stage 3 – 4 (Simon Johnson www.teachwithict.weebly.com)



Issues

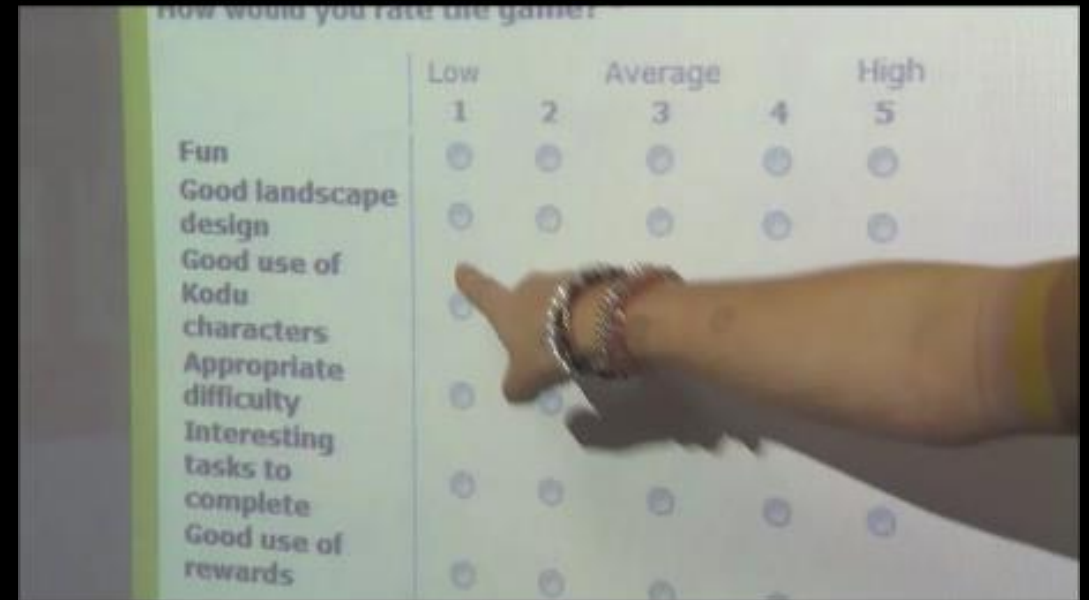
- Very few schemes of work for teaching Computing at KS3 at the moment
- Primary children entering Y7 with little experience
- Good resources from CAS
- Exam boards updating

Enhance by linking to other technologies

**Lesson management and task direction
by IWB**



**Extend learning by linking to learning
platform (Blogging and Survey tools)**



Lesson Structure?



That's Amazing (i-Pad) Outline Planner

This document outlines the "Storyboard" to teaching and learning associated with the module "That's Amazing".



- The concept of "Assigning characteristics to objects" How do we make computer objects behave in a certain way which is different to others?
- Examples from various I-Pad, Xbox and Wii games demonstrating actions and responses from objects in games. Show maze games and Accelerometer.
- Teacher goes over the key words (Characteristics, value, tilt, slide) Demonstrates and explains giving an object multiple characteristics via lines of code

Process Model

- Why
- Make
- Play
- Evaluate
- Share

Lesson Structure?

Make

- Teacher demonstrates the association of symbols on the coding interface with the tilt actions of the accelerometer. NB Important to "Lock Screen"
- Children complete the first three activities in the unit
- Teacher uses screen share software e.g. Air Server to demonstrate examples from class to illicit explanation and understanding from the children. What would happen if the left tilt made the object go right!!! Would it increase the difficulty?

Process Model

- Why
- **Make**
- Play
- Evaluate
- Share

Lesson Structure?


Play

- Teacher now demonstrates the final activity where the children build a game by making the ball move around the maze. Why doesn't the ball stop when it hits a wall
- Teacher introduces the concept of assigning a value to an immovable object
- Children have to create a "Game" and ask a partner to share at this stage.
- Partner to give verbal feedback and changes are made (Introduction to Evaluate)

Process Model

- Why
- Make
- Play
- Evaluate
- Share

Lesson Structure?



•Teacher introduces the concept of evaluation on IWB and asks for ideas on how to assess the game. Explains that evaluation is important to "Real" games developers.

•What makes a game difficult? Challenging?

•How could the game be made more complex. What happens if the colour of the wall is changed?

•Children evaluate their partners game using the feedback sheet.

Process Model

- Why
- Make
- Play
- Evaluate
- Share

Lesson Structure?

Process Model



Share

- Children are shown the "Share" facility on the software . Teacher demonstrates how to access games shares in the "Private" area.
- Children are asked to devise a naming convention
- Children are asked to access the game and complete an evaluation and rating
- **Teacher revises the key concepts of characteristics, assigning characteristics to an immovable object, accelerometer and logical controls .**

- Why
- Make
- Play
- Evaluate
- **Share**

Considerations

- Which Software will I use?
- Which approach to managing computing will I choose?
- How will I train my Staff?
- How will I introduce this to the school (Club First?)
- How will I ensure progression?
- How will I keep up to date with resources and developments?
- How will I extend learning to enable pupils to complete activities?
- How can I include the pupils themselves as champions?



Key Resources - Links

Main Program sites

Kodu <http://www.kodugamelab.com/>

Scratch <http://scratch.mit.edu/>

Espresso Coding <http://www.espressocoding.co.uk>

General site good links

Main computing discussion site <https://sites.google.com/site/primaryictitt/home>

Mark Dorling <http://www.digitalschoolhouse.org.uk/>

Richard Anderson www.lttonline.net

KS1/2

James Abela - Teaching Computational Thinking to Primary: <http://www.youtube.com/watch?v=bIUf3Cav>

KS3

Simon Johnson - TouchDevelop Challenge: SoW & resources for teaching students to create apps for Windows 8 - <http://touchdevelop.weebly.com>

Scratch - This guide, provides an introduction to creative computing with Scratch, using a design-based learning approach. It is organized as a series of twenty 60-minute sessions, and includes session plans, handouts, projects, and videos: <http://scratched.media.mit.edu/resources/scratch-curriculum-guide-draft>

KS4

Simon Johnson - teachwithict Computing - SoW & resources for delivering OCR GCSE Computer

Science: <http://teachwithict.weebly.com/computing.html>

Computing at School - Online community dedicated to teaching Programming / Computer Science in school. Contains a wealth of resources including: Schemes of Work, Tutorials and discussions on best

practice: <http://community.computingatschool.org.uk>

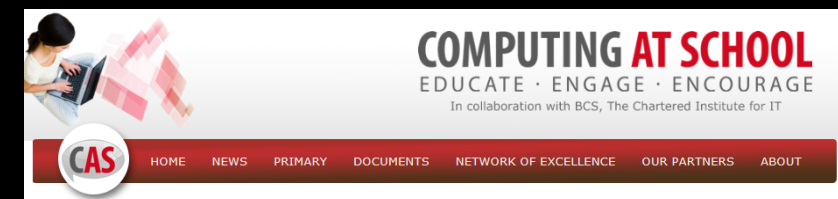
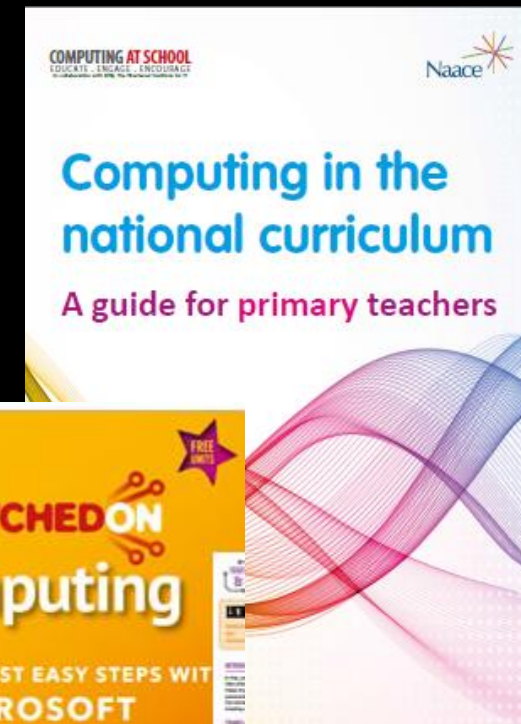
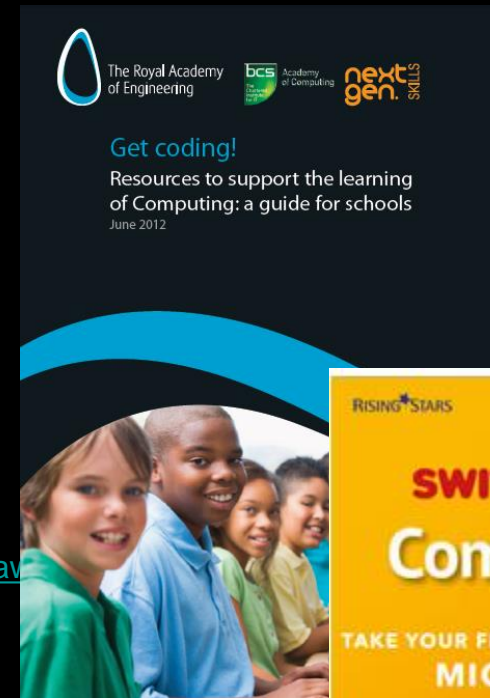
Mark Clarkson - SoW for OCR GCSE Computer

Science: <http://voyager.egglescliffe.org.uk/mwc/mukoku/course/view.php?id=15>

Computer Science Unplugged - CS Unplugged is a collection of free learning activities that teach Computer

Science through engaging games and puzzles that use cards, string, crayons and lots of running

around: <http://csunplugged.com>



What has worked – Golden Nuggets!

- As much work off the computer as on it!
- Computing is equitable some surprising learners shine!
- There are more resources available to help that you first realise
- It teaches more than just how to control a computer
- Pupils become creators using technology not just consumers
- Leads to deeper use of other existing technologies
- Has opportunities for developmental and iterative learning
- Training and support for staff is vital

Remember...

We can make a difference to students lives!



Chris Carter
Managing Director at Void Games
Wolverhampton, United Kingdom | Computer Games

Current	VOiD Games Limited
Previous	Volt Europe/ Rare Ltd, Moseley park
Education	The University of Wolverhampton



Max Wainwright – max@clara.net
Twitter@ictapps

David Whyley – Dave.Whyley@whytekconsulting.co.uk
Twitter @davewhy
www.whytekconsulting.co.uk