Bells & Whistles: Learning Programming Principles Through Multimedia Authoring

Stephen Lowe

Aoraki Polytechnic
Timaru, New Zealand

stephenl@aoraki.ac.nz

Proceedings of the 15th Annual NACCQ, Hamilton New Zealand July, 2002 www.naccq.ac.nz

ABSTRACT

This paper describes a study carried out at Aoraki Polytechnic. A sub-group of students enrolled on Certificate of Business Computing was identified: those who were afraid of programming.

Members of this sub-group performed poorly on the compulsory modules PD100 (Program Development) and PP100 (Programming Principles) and subsequently avoided programming modules wherever possible. Students belonging to this sub-group could be tracked right through to PJ300 (NDBC Project).

This was of grave concern to the teaching staff because programming skills were seen as essential to satisfactory employment in the IT sector. Gun programmers were seen as rare, and special aptitude was needed to gain excellence. However, it was considered that workaday competence was within reach of the majority of students, given the right training.

Students belonging to the sub-group favoured multimedia modules MA100 (Multimedia Principles) and MA200 (Multimedia Development) along with IN200 (Web Site Development) because they perceived these to be 'soft' subjects dealing in design, layout and communications. When on these modules students encountered JavaScript and Macromedia's ActionScript they found it easy, and performed satisfactorily or even well in assignment work.

It was thought to be the non-threatening, visual and fun environment in which to learn the principles of programming that enabled the students.

This happy accident was developed into a system. The multimedia studies tutor reviewed the prescriptions and course materials for MA100, MA200 and IN200 and ensured that sufficient attention was paid to programming structures, syntax and development.

A system of technical storyboarding was subsequently developed which combined structure, logic depiction and pseudocode. This visual storyboarding system proved to be highly effective as a tool for planning interactivity and functionality in multimedia production and was within the grasp of all the students.

1. INTRODUCTION

Students who avoided programming were found to list a common set of mythical obstacles: right-brain thinker; no good at maths; had chosen the arts, not the sciences; not the logical type; didn't like the geeky image; programming was a boy thing.

Many went further than just a list of obstacles, they expressed a fear of programming. Their fears were founded in visions of: long solitary hours of work; immensely thick books full of arcane texts; numbers bigger than 100; of strange symbols; and of fruitless hours failing to correct syntax.

This group chose every module with the one criteria —that it didn't involve programming.

It was necessary to make some provision for these students, to sneak the business of programming up on them, and later present them with a fait accompli: 'You see, you can program'. It was thought that to gain some mastery of programming, albeit at a modest level, was essential to satisfactory employment prospects in the IT sector.

In multimedia work it was found that students became so determined to make the thing work, they willingly persevered with logical, typographical, and syntax errors. The tutor encouraged group effort to solve problems, and rarely offered code verbatim.

It became apparent therefore that the tutor's role had shifted from technician to motivator, and it was further noted that this was closely aligned with the findings of Jenkins (2002) in his paper "Teaching Programming, A Journey from Teacher to Motivator".

2. METHOD

Working with the teaching of motivational speaker Rakesh Pandey, whose ethos of "easy, quick, and fun" could be used as a diagnostic tool on failing modules, the tutor looked for methods that would motivate the threatened sub-group of students.

The pragmatic, results-oriented, problem-solving, team-effort aspects of multimedia production provided precisely the vehicle the tutor needed to teach, under cover of 'fun', the principles of programming. The criteria of 'quick' was met, because correct code produced immediate positive feedback, and the criteria of 'easy' was met because many hands made light work, and because the mechanical processes of Macromedia Flash were easily accessible to even the simplest thinkers.

To engage the student in meaningful tasks the problems were presented in a contemporary format: drag and drop web media; selecting colours and sizes of product; animating avatars in multi-user environments; hiding purchased items in a shopping cart; and passing parameters from the checkout to a server-side script.

The logic was diagrammed in storyboards and in plain English with the rough visuals and beat outline; the logic inlined with the story. The tutor searched for an established system of logic depiction specifically suited to multimedia, and failed to find one. The tutor's own system, tried and tested in the industry, was developed into a teaching tool similar to the structured diagrams normally associated with PP100 and PD100 (see Figure 1).

3. APPLICATION

Two scenarios were favoured by the tutor: both were popular with students; both were current in the modern idiom; both met the requirements set by the need to teach a complete set of basic programming structures.

3.1 DRAG-N-DROP PIZZA ONLINE ORDER

The project type: Media-rich web application.

The technology: Macromedia Flash 5.

The purpose: To provide a pizza house operator with a reliable self-ordering system for take-away pizza buyers. To provide the pizza buyers with a compelling, easy-to-use, reliable way of ordering.

The scope: To create a graphical drag-n-drop interface to pizza ordering where the confirmed order was emailed to the kitchen. Outside the scope was bank card processing.

The plan: To create a pizza base, scalable to 8-inch, 10-inch, or 12-inch. To create graphical representations of various common toppings. To make these toppings draggable to the pizza base, so customers could create their dream pizza online. Name-and-value pairs were passed with each dragged-on topping to a form, city zone factors were applied to cover delivery costs, voucher discounts were applied, and the form, confirmed and validated, submitted url-encoded data to a server-side cgi script which output an email to the kitchen.

The schedule: To deliver the teaching and facilitate the necessary group work over 32 hours in the classroom, plus additional tutorial time, attached to the module MA200 Multimedia Development.

The desired learning outcomes: In line with the prescription for MA200, 'implement the project using appropriate technology', it was desired that students would become thoroughly familiar with the primary

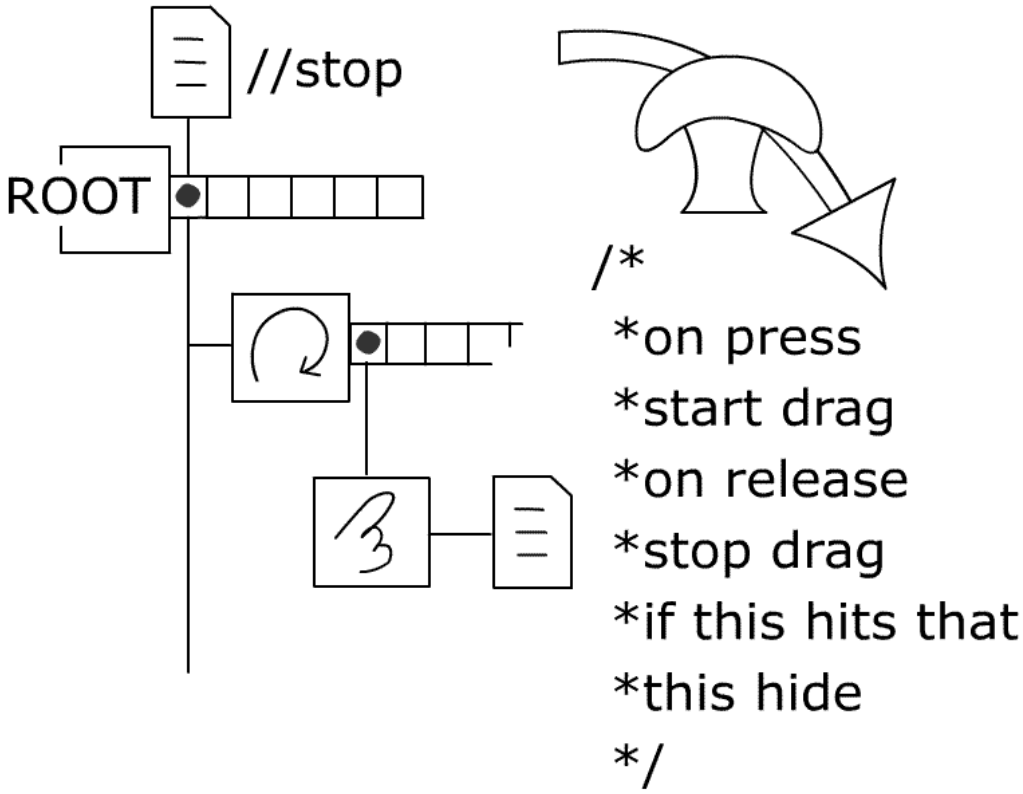

Figure 1. The hierarchy of nested instances, and the scripting, is shown in this technical storyboard of a drag and drop object

structures, objects, methods and properties of Flash ActionScript, a language akin to JavaScript and based on the ECMA-262 standard.

3.2 DRAG-N-DROP FASHION BOUTIQUE

The project type: Media-rich web application.

The technology: Macromedia Flash 5.

The purpose: To provide a boutique operator with a reliable online catalogue ordering system for fashion label buyers. To provide apparel and accessory buyers with an easy-to-use, fun, reliable way of ordering.

The scope: To create a graphical drag-n-drop interface to apparel and accessory ordering where the confirmed order was emailed to the warehouse. Outside the scope was bank card processing.

The plan: To create a virtual boutique with ambience, a compelling online shopping experience. To create graphical representations of various fashion lines. To make these clothes and accessories draggable to an animated mannequin (or avatar), so customers could try various outfits. Name-and-value pairs were passed with each combination to a form, which, when confirmed, could submit to a server-side cgi script which output an email to the warehouse.

The schedule: As above.

The desired learning outcomes: As above, and, because these students worked in small disparate groups, a secondary outcome was the discovery of the value of working to a comprehensive specification.

3.3 PROGRAMMING ELEMENTS

A drag and drop object, representing a purchasable item such as a portion of mushrooms, taught: parent and child relationships; classes (library objects in Flash), and instancing; nesting; dot syntax; verbose and terse constructions; commenting; targeting, and many other core programming skills.

A shopping cart taught: conditional statements; the programmatic manipulation of properties; the

passing of parameters; and the formatting of strings and floating point numbers.

An email order system taught: url-encoding, and the passing of name and value pairs; interfacing with server-side technologies other than the primary Flash technology; and usability issues.

Creating a media-rich web application, as a group project, also gave the student an opportunity to learn: working to a user requirement; working to a specification; adhering to naming conventions; prototyping, testing and debugging procedures; and quality assurance.

3.4 RESULTS

The students researched real-world pizza outlets and boutiques to ascertain the user requirement. They created storyboards which illustrated the environment, the user (or avatar), and the user's interaction with the environment. Spurred on by a desire to implement the meaningful and engaging application, they quickly learned the necessary procedures to give the application its back-end functionality. Students with drawing talent worked hard to create original and attractive front-ends. Testing was carried out with rigour, as the pressure came on to deliver a reliable build.

A period of post-production reflection revealed a group that felt satisfied with their achievement, and were keen to explore multimedia scripting to a level of greater complexity. More important were a number of students who now wanted to sign on mainstream programming modules, an interest in programming awakened.

Samples were too small to be processed into statistical evidence, and this study was treated as action research to be iterated each year, involving all the stakeholders; the practice of continuous process improvement.

4. TEACHING ISSUES

Three main issues arose: assessment, portfolios, and learning styles. Skepticism was anticipated, and the tutor was keen to prepare early for moderation or observation.

4.1 ASSESSMENT

Students were required to demonstrate core procedures in a practical test. With that assurance of individual learning in place it was acceptable for students to contribute only their specialist skills to the group work. This raised the standard of the product of that group work, and in turn raised morale and motivation in the group.

4.2 PORTFOLIO

The tutor was concerned not only with the students' attainment of the qualification, but also with the production of meaningful material for display. In a paper by Plimmer (2000) the value of portfolios in programming courses was emphasised, and reflection and creativity resulted. Employers in the media sector, the tutor noted, were more interested in an applicant's portfolio than in their qualification — a good test reel, and a timely arrival on the doorstep of a busy studio, were the surest indicators of success for the job-seeker.

4.3 LEARNING STYLES

Meeting different learning styles was a concern of the tutor, and it was observed that the formats common to multimedia production also met the requirements of the educationalist. In a multimedia studio storylines are pitched to the team: a series of communicative sketches are shown, accompanied by a stagecrafted presentation. If the initial pitch is successful, time and money are made available for the creation of a test reel which validates the proposed sequences. The needs of the visual learner, the kinetic learner, and the theorist are thus met.

5. HISTORICAL ASPECTS

A thorough search of the web, using the keywords 'teaching', 'programming' and 'multimedia', revealed little. However, it was observed that the turtle graphics of LOGO and sound generation in QBasic had been popular aids in schools for the teaching of programming.

Teaching programming through multimedia was found to be not new, but had declined, perhaps because of the considerable complexity and cost of a package like Macromedia Flash. Various freeware packages were used to introduce students to multimedia concepts, but none had the equivalent sophistication of Flash ActionScript.

6. CONCLUSION

There should be equal opportunity for wannabe computer programmers, and a choice made early in

life to follow the arts or the humanities should not preclude anyone from having a go.

Obstacles and fears were largely unfounded, could not be discounted, and could be overcome.

Multimedia authoring (as implemented in Macromedia Flash 5) was a fit, comprehensive, and appropriate vehicle for the teaching of programming principles.

REFERENCES

Jenkins, T. (2002). "Teaching Programming, A Journey from Teacher to Motivator". Accessed May 2, 2002. http://www.ics.ltsn.ac.uk/pub/conf2001/papers/Jenkins.htm University of Leeds, UK.

Plimmer, B. (2000). "A Case Study of Portfolio Assessment in a Computer Programming Course". Proceedings of the NACCQ, Wellington, 2000, pp 279-283.

Moock, C. (2001). "ActionScript, The Definitive Guide". Sebastopol: O'Reilly.