Human Technology

An Interdisciplinary Journal on Humans in ICT Environments ISSN: 1795-6889

www.humantechnology.jyu.fi Volume 4 (1), May 2008, 47–61

SPATIAL ABILITY AND LEARNING TO PROGRAM

Sue Jones
School of Computer Science
University of Nottingham
Nottingham, UK

Gary Burnett
School of Computer Science
University of Nottingham
Nottingham, UK

Abstract: *Results in introductory computer programming modules are often disappointing, and various individual differences have been found to be relevant. This paper reviews work in this area, with particular reference to the effect of a student's spatial ability. Data is presented on a cohort of 49 students enrolled on an MSc in Information Technology course at a university in the UK. A measure was taken of their mental rotation ability, and a questionnaire administered that focused on their previous academic experience, and expectations relating to the introductory computer programming module they were studying. The results showed a positive correlation between mental rotation ability and success in the module (*r *= 0.48). Other factors, such as confidence level, expected success, and programming experience, were also found to be important. These results are discussed in relation to the accessibility of programming to learners with low spatial ability.*

Keywords: *spatial skills, programming ability, individual differences.*

INTRODUCTION

Results in introductory computer programming modules are often disappointing (Mancy & Reid, 2004), with reports of up to 30% of students failing to complete them (Guzdial & Soloway, 2002). Students who perform well in other subjects may not achieve equivalent success in programming tasks (Byrne & Lyons, 2001; Fincher et al., 2005), lose confidence, and give up computer science courses. Irrespective of experience, some programmers appear to be more skilled than others. Curtis (1981) found a range of performance scores of 23 to 1 in a debugging exercise, and Shneiderman (1980) reported differences in performance of 100 to 1 among programmers of similar programming experience.

Previous research has focused on why some students underperform in programming. Various individual differences have been implicated in programming success, with debate over the relative importance of each of these factors. This paper will focus on spatial ability,

URN:NBNfi:jyu-200804151352

but also considers how some of the other individual differences are related in their influence on learning to program. The next two sections will focus on a literature review of the work in this area. Then follows an analysis of data collected in a study carried out over the academic year 2005/06 at the University of Nottingham. Finally, the results will be discussed in relation to programming achievement.

SPATIAL ABILITY AND MENTAL MODELS

One individual difference considered to have some relevance to programming aptitude is spatial ability. This is a heterogeneous cluster of skills considered to be a dimension of intelligence distinct from verbal, mathematical, and reasoning skills. Halpern (2000) defines spatial ability as a cognitive characteristic that gives a measure of the ability to conceptualize the spatial relations between objects. As this is a broad cognitive concept, various categorizations have been suggested to help organize our understanding of this area, one of these being mental rotation. Mental rotation is the capacity to accurately picture the rotation of two- or three-dimensional objects in the mind, and some researchers believe that it is a good measure of a general spatial reasoning ability (Halpern, 2000). The Vandenberg and Kuse (1978) Mental Rotation Test is the standard test for this skill.

Spatial ability has been shown to be important for navigation in the real world, and in an abstract information space such as hypertext (Jones & Burnett, 2007b). It is also considered by some to be an important determinant in program comprehension, due in part to source code being likened to a multidimensional virtual space that requires similar skills for navigation as those utilized in a real environment (Cox, Fisher, & O’Brien, 2005). However, there are few studies looking at relations between spatial ability and individual programming performance. Mayer, Dyck, and Vilberg (1986) showed that success in learning Basic was related to spatial ability ($r = 0.31$, $p < 0.05$). Fincher et al. (2005) showed an overall small positive correlation between performance in a spatial visualization test and marks achieved in the introductory programming courses at 11 institutions ($r = 0.17$, $p = 0.047$). Both of these studies used versions of the Paper Folding Test, designed to measure spatial visualization, one of the various types of task included in the broad cognitive category of spatial ability (Halpern, 2000). The Paper Folding Test requires subjects to imagine the result after folding a paper object, but this process does not require mental rotation (Velez, Silver, & Tremaine, 2005).

Webb (1984), studying children between the ages of 11 and 14, found a relationship between spatial ability and various programming components of a short course in Logo programming, with an average correlation of 0.63 ($p < .001$). The only pretest measure to give a stronger correlation was the score on a mathematics reasoning test. Webb utilized three measures of spatial ability, one of which was the Paper Folding Test, but the others requiring mental rotation of figures. Fisher, Cox, and Zhao (2006) used the Vandenberg and Kuse (1978) Mental Rotation Test to study correlations with a software maintenance task for a short Java program. While they found a high correlation for the men ($r = 0.63$, $p = .012$), this was not reflected in the women’s results.

A related factor in the role of spatial ability to programming success is the development of mental models. Mental models are variously defined, but in the context of this paper are considered as predictive representations or abstractions of a program. In recent work, Jones and Burnett (2007a) demonstrated differences in the navigation of source code in a code comprehension exercise, with individuals with higher spatial ability jumping between functions more frequently and making more interclass jumps (moving between files). The authors speculate that this style of navigation may allow a better mental model of the program to be formed, thus aiding comprehension. Mental models of spatial information are called cognitive maps (Downs & Stea, 1973), and people build these while familiarizing themselves with an environment. They become disorientated if this internal map does not correspond to the physical representation of the environment (Westerman & Cribbin, 1999). In relation to cognitive maps of program code, Fisher et al. (2006, p. 1) use the term "codespace," and define it as "a programmer's mental model of source code with respect to the perceived spatial attributes of entities identified within the code." Hence the mental model is an abstraction of the program formed from piecing together various kinds of information extracted while navigating the source code. Wiedenbeck, LaBelle, and Kain (2004) stress the importance of a good mental model in program understanding.

It has been argued that because source code is linear, mental rotation may not be as important as other types of spatial ability for navigation in programming environments (Fisher et al., 2006). However, the studies utilizing mental rotation as a measure of spatial ability appear to show stronger relations with measures of programming aptitude than those using the Paper Folding Test as a measure of spatial visualization. Kimura (1999) makes the link between success in mental rotation and the capacity to build a mental model, or cognitive map, of an environment. She suggests that good mental rotation capacity enables us to recognize a scene from different angles, and thus to retrace a route in reverse when returning to a destination, or piece together other bits of information to devise a new route back. There is an increasing recognition of the importance of mental models to learning programming (Wiedenbeck et al., 2004), and perhaps a good mental rotation capacity is allowing formation of more accurate mental models of programs.

One study looked at the map-drawing styles of students to determine if this was a predictor of success in the introductory programming courses they were studying (Tolhurst et al., 2006). Students were required to sketch maps of a given real world environment, and the maps were then grouped according to the landmark, route, survey model for the acquisition of spatial knowledge (Werner et al., 1997). When compared to the marks achieved in the course, they found a trend for the high achievers to draw survey maps, while those who sketched route maps performed less well but better than those who produced landmark maps. The authors speculate that programming ability is related to an ability to navigate through the information space using the same skills as in the real world. Good spatial ability has been related to the development of survey knowledge, equivalent to a well-formed cognitive map of an environment (Cutmore, Hine, Maberly, Langford, & Hawgood, 2000).

RELATION OF SPATIAL ABILITY TO OTHER INDIVIDUAL DIFFERENCES

This section reviews the literature on the impact of various individual differences on programming performance. The main focus will be the interactions with spatial ability.

Gender

Girls are underrepresented in university computer science (CS) courses (Byrne & Lyons, 2001). Suggested reasons for this include sex stereotyping and males' greater exposure to computers and computer games (Mumtaz, 2001; Schumacher & Morahan-Martin, 2001). Boys appear to have a more confident, positive attitude toward computers (Durndell & Haag, 2002). Another reason is that spatial skills are crucial to most video and computer games as well as many computer applications, and repeated practice may actually enhance spatial skills (Subrahmanyam, Greenfield, Kraut, & Gross, 2001). There is a wealth of evidence of gender differences in spatial ability, with females appearing to underperform in certain measures, such as mental rotation (Voyer, Nolan, & Voyer, 2000).

It is difficult to study gender differences in programming style and ability due to the fact that many females, prior to university admission, have already chosen not to take CS courses for the reasons mentioned above (Scragg & Smith, 1998). Byrne and Lyons (2001) found no significant difference in performance between male and female students in a first-year programming module, possibly because the module was part of a Bachelor of Arts honors degree program with a preponderance of female students (61%). However, other studies also have not found the expected gender difference (Bergin & Reilly, 2005; Rountree, Rountree, & Robins 2002), perhaps because group-based differences such as gender have less effect on an individual's performance than individual differences such as spatial ability or cognitive style (Beckwith & Burnett, 2004). Fisher et al. (2006) hypothesize that females prefer a more low-risk, bottom-up approach to program development and comprehension, and males a more high-risk, abstract, top-down approach. Bradley (1985) demonstrated that top-down processing was positively related to Logo programming success.

Self Efficacy/Comfort

Self-efficacy relates to how we estimate our capability to perform well in a certain context (Bandura, 1986). A person with high self-efficacy is more likely to undertake challenging tasks, expend more effort to achieve them, and demonstrate persistence when difficulties arise. Rountree et al. (2002) surveyed students early in a first-year computer science course and found that students' expectation of how they were going to perform was the biggest indicator of success. Surprisingly, students predicted the outcomes very early in the course, and this may contribute to their level of motivation and persistence required to achieve. Closely related to self-efficacy is comfort level, based on our perception of the degree of difficulty of a task, which affects our anxiety levels. Studies have found that comfort level, derived from questionnaires, was strongly predictive of programming performance (Bergin & Reilly, 2005; Wilson & Shrock, 2001). Students with low computer experience are likely to be less confident and more anxious when starting a programming course (Byrne & Lyons, 2001). It has also been suggested that males tend to show greater self-efficacy and lower computer anxiety than females (Beckwith & Burnett, 2004; Durndell & Haag, 2002). Having a good mental model increases self-efficacy by enabling program comprehension (Wiedenbeck et al., 2004).

Previous Academic Exposure

Previous programming experience seems to relate to success in introductory programming courses (Rountree et al., 2002; Wilson & Shrock, 2001), with Wiedenbeck et al. (2004) linking this with self-efficacy. Boys are more likely than girls to have previous programming experience (Bruckman, Jensen, & DeBonte, 2002). Good performance in mathematics is also relevant, and is often an entry requirement for computer science, with the belief that the skills required for solving mathematics problems are similar to those needed for programming tasks (Byrne & Lyons, 2001). Various studies have found relations between mathematics results and success in learning programming (Webb, 1984; Werth, 1986; Wilson & Shrock, 2001). Byrnes and Lyons (2001) found a relationship between results in secondary school mathematics examinations and results from a first-year programming course (r = 0.353, p < 0.01). The correlation with science results was even stronger (r = 0.572, p < 0.01). There was no correlation between the English or foreign language results and programming achievement. Others have found a similar relationship with science (Bergin & Reilly, 2005; Werth, 1986).

The ability to succeed in mathematics has been related to spatial ability (Pease & Pease, 2001), and one study showed that males outperformed females in both the Vandenberg and Kuse (1978) test for spatial ability and a mathematics aptitude test, with mental rotation predicting mathematics aptitude for the female samples (Casey, Nuttall, Pezaris, & Benbow, 1995). When mental rotation ability was statistically adjusted for, the gender difference in mathematics achievement was eliminated in most of the groups studied.

Cognitive Style

Cognitive abilities are specific to a particular domain of content or function, such as verbal, numerical, or spatial ability. A measure can be taken of an individual's spatial ability as separate from their verbal reasoning score—one may be high, the other low. In contrast, cognitive styles cut across these domains, and have more to do with organization and control of cognitive processes. Consequently, there appears to be an interaction between cognitive abilities and styles, with field-dependency being the style most associated with spatial ability (Chen, Czerwinski, & Macredie, 2000). McKenna (1984) presents arguments debating whether field dependence, often measured by the Embedded Figures Test (EFT), is a cognitive style or cognitive ability. The EFT requires participants to locate a given simple shape embedded within a larger complex one. Because the EFT is timed, it has been argued that it is more a measure of cognitive ability than style, assessing differences in level of, rather than manner of, performance. McKenna reviews work showing there is a strong relation between the EFT and spatial ability.

Various studies have demonstrated some impact of field dependency on programming achievement. Bishop-Clark (1995) carried out a review of some of the work in this area. She concluded that the results are not consistent (correlations ranging from .08 to .80), but that field independence appeared to be positively related to programming success. Mancy and Reid (2004) compared field dependency and marks in various assessments on an introductory programming course. Field dependency was measured using the EFT, and the results showed positive correlations with marks on the different assessments, with a good correlation (r = 0.40) with the final examination.

In summary, individual differences with varying degrees of impact on programming performance have been discussed in relation to spatial ability (see Figure 1). However, there are only a small number of adult studies looking at mental rotation in this context. The following study aimed to gather extra data to supplement the research knowledge regarding any effect of spatial ability, and other interacting factors, on programming performance.

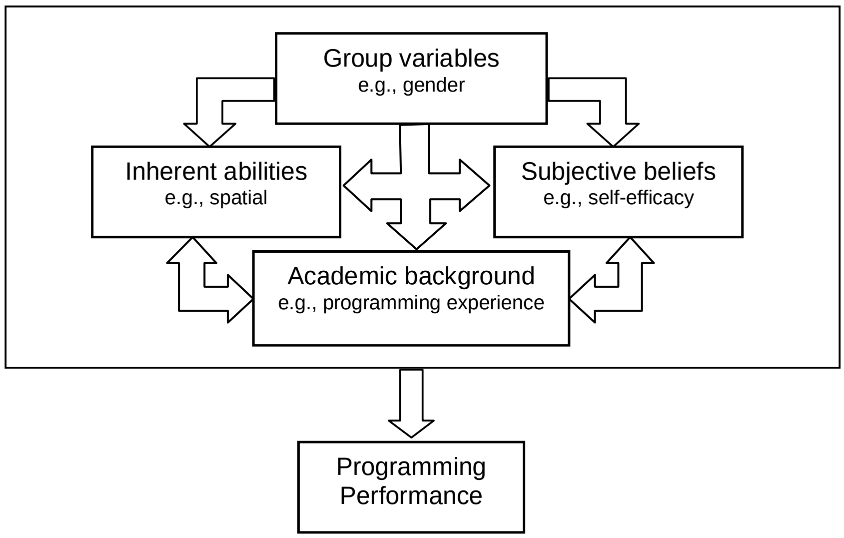

Figure 1. Potential interactions between individual differences and programming performance.

THE STUDY

During the academic year 2005/06, a test of mental rotation ability and a questionnaire were administered to a cohort of university students. This section will focus on the results of our analyses to ascertain any relations between the variables studied.

Data Collection

Participants consisted of 49 volunteers (average age 26 years) from students enrolled in a one-year master's conversion course (meaning students did not have computer science as their first degree) at the University of Nottingham, UK. An introductory programming module (ICP) was compulsory for all students in the first semester. The course consisted of two streams. Students in the MSc in Information Technology (IT, n = 28) had their first degree in a science or engineering subject and would continue with further compulsory programming modules in the second semester. Those studying for the MSc in the Management of Information Technology (MIT, n = 21) had a first degree in a wide range of subjects (including arts and humanities), and were not required to take programming modules after the first semester. The cohort consisted of 39 males and 10 females, with only 2 females in the IT stream.

Data on the participants' academic history and perceived programming experience were collected by questionnaire. At the end of Semester 1 (December 2005), and before taking the introductory programming examination, the students were asked how they rated their confidence levels in their last (as yet unmarked) programming coursework. They also rated the ICP in comparison to other nonprogramming modules on the parameters of difficulty, workload, and expected success, similar to Rountree et al. (2002). In addition, the final marks in the following modules were collated:

- Semester 1: compulsory modules
  - Introduction to Computer Programming (ICP). The assessment consisted of 50% coursework (2 programming assignments) and 50% examination. The language taught was Java.
  - Introduction to Human Factors (IHF), a nonprogramming module with assessment based on 25% coursework and 75% final examination.
- Semester 2
  - Object Oriented Systems (OOS), with two programming assignments contributing 50% to the final mark, and a final examination. This module was compulsory for the master's in IT. The language taught was C++.
  - Management of IT (MAN), a nonprogramming module with 25% coursework and 75% examination. This module was compulsory for those taking the master's in MIT.

The programming modules involved considerable practical programming assignments, while the nonprogramming subjects focused on issue-based discussion elements.

Individual differences in the participants' spatial skills were measured using a version of the 3D Mental Rotation Test found at Psychlab OnLine[1]. The test is a modified version of the Vandenberg and Kuse (1978) Mental Rotation (MR) test, and was customized for this study by Professor Hay of the University of Wisconsin, Milwaukee.

In the version used, the participants were asked to determine if one shape could be mentally rotated to match the orientation of a second (see Figure 2 for an example). The students were presented with 30 examples to be completed as quickly as possible, with equal emphasis being given to accuracy and speed. This was an on-line test. Once an answer was submitted, there was no recourse for correcting it, and no feedback was provided on the correctness of the answers. At the completion of the test, a file was generated with the number of correct answers, and the total time taken for completion of the 30 questions (mean time = 215s, range 59 to 452 s). A score for spatial ability was derived from the number of correct answers divided by the total time (in seconds) to complete the exercise. The final number was multiplied by 100 to provide a more usable scale.

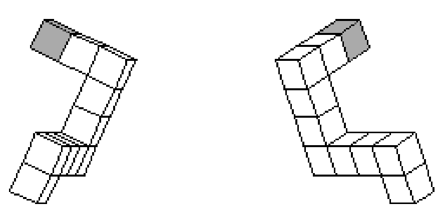

Figure 2. Example of the modified Vandenberg and Kuse (1978) Mental Rotation Test, customized by Professor Hay.

Results

Module marks and questionnaire responses were recorded. Statistical analyses were run to determine any relations between spatial ability and other individual differences.

Module Marks

There was a strong correlation between mental rotation (MR) scores and the participants' grades in the programming modules, but this was not reflected in the nonprogramming modules (see Table 1). There was a stronger correlation for the IT students (those who carried on with programming) in the ICP, and a high correlation between MR scores and results in the more advanced (OOS) programming module for these students. As can be seen in Figure 3, the spread of results in the ICP module was greater for the IT stream (30% to 95%; $M = 66.59$, $SD = 15.43$) than the MIT stream (42% to 78%; $M = 64.90$, $SD = 9.93$), although the difference in means was not statistically significant ($p = 0.66$). Similarly, the MR test scores showed a greater range among the IT students (3.09 to 30.76; $M = 16.47$, $SD = 7.22$) than the MIT students (4.18 to 22.30; $M = 11.73$, $SD = 4.84$), with the IT students scoring significantly higher ($p < 0.05$).

When the results obtained in each of the modules were compared, there was found to be a strong positive correlation between the two programming modules, ICP and OOS, and between the two nonprogramming modules, IHF and MAN. There were no correlations between the programming and nonprogramming modules (see Table 2).

Table 1. Correlation Analysis for Mental Rotation (MR) Scores and Module Marks.

		Programming modules				Nonprogramming modules	
		ICP (All)	ICP (MIT)	ICP (IT)	OOS (IT)	IHF (All)	MAN (MIT)
MR Score	*r*	0.48	0.37	0.57	0.68	0.21	0.10
	p	<0.01	<0.05	<0.01	<0.01	0.16	0.65
	n	49	21	28	27	46	21

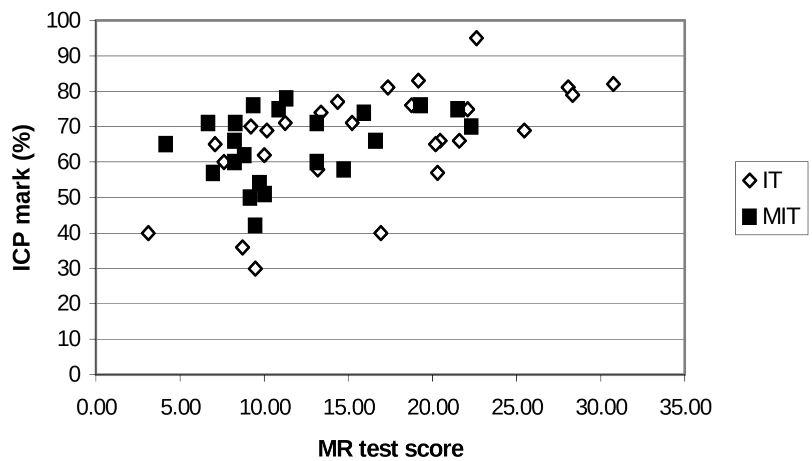

Figure 3. Scatterplot of ICP results as a function of MR test scores for the two master's streams.

Table 2. Pearson Correlation Analysis for Module Marks.

	Programming modules		Nonprogramming modules	
	ICP	OOS	IHF	MAN
ICP				
OOS	0.73 (<0.001)			
IHF	0.28 (0.07)	0.26 (0.26)		
MAN	0.10 (0.76)	N/A	0.69 (<0.005)	

Note: The significance values are in parentheses.

Questionnaire Results

Many of the multiple-choice questions had a choice of 4 or 5 answer categories, and with only 44 students completing the questionnaire, some of the categories had a very small number of respondents. To enable statistical analysis, these categories were collapsed into 2 or 3 items. This allowed *t* tests to be carried out to determine if there were differences in the means for MR test scores and ICP results for the dichotomous measures, and Kruskal-Wallis tests on the trifold measures. The dichotomous variables were also subjected to chi-square analysis to determine if there were any differences in the answers for the two degree streams, and for confidence levels.

Those with greater perceived programming experience had higher spatial scores, and performed significantly better in the ICP results (see Table 3). Similarly, those with higher confidence levels for the ICP coursework performed very well in the ICP module, and there was a trend for them to have higher MR scores (although just failing to reach statistical significance).

When students were asked to compare ICP with other nonprogramming modules, it was found that individuals rating ICP as more difficult tended to have lower spatial scores. Those rating success with ICP (relative to nonprogramming modules) as high were achieving better end results (see Table 4).

There was a nonsignificant trend for those in the IT stream to have more programming experience. Seventy-four percent claimed to be professional/intermediate, compared to only 39% of the MIT students (see Table 5).

Table 3. T-tests for Questionnaire Results.

		MR test score			ICP mark		
Parameter	Categories	Mean	SD	p	Mean	SD	p
First degree	Science	15.71	7.63	0.28	68.00	15.55	0.69
	Non-science	13.29	6.00		66.29	9.76	
Gender	Male	14.67	6.72	0.47	68.51	11.70	0.19
	Female	12.88	5.41		62.22	15.91	
Programming experience	Professional/intermediate	16.65	7.41	<0.05	72.50	9.54	<0.005
	Novice/none	12.31	5.51		60.60	14.34	
Confidence	High/moderate	15.95	7.15	0.09	72.59	8.23	<0.001
	Little/none	12.23	5.85		56.47	14.83	

Table 4. Results of the Kruskall-Wallis Analysis of Student Expectations.

		MR test score			ICP mark		
Parameter	Categories (collapsed)	Mean rank	χ2	p	Mean rank	χ2	p
Workload	Much less/less	33.67	4.20	0.12	26.33	2.46	0.29
	The same	26.09			27.09		
	More/much more	20.07			20.43		
Difficulty	Much less/less	38.60	10.50	<0.01	35.50	5.97	0.05
	The same	26.64			22.18		
	More/much more	18.79			20.30		
Success	Much less/less	14.60	5.01	0.08	14.00	6.10	<0.05
	The same	24.11			23.67		
	More/much more	25.63			26.50		

Table 5. Results of Chi-square on Programming Experience and Confidence.

		Masters				Confidence			
Parameter	Categories (collapsed)	IT	MIT	χ2	p	High/ moderate	Little/ none	χ2	p
Programming experience	Prof/intermediate	21	8	3.25	0.07	24	2	13.17	<0.001
	Novice/none	7	13			8	15		
Confidence	High/moderate	22	12	1.36	0.24				
	Little/none	6	9						

The impact of programming experience on confidence levels was obvious, with 92% of the more experienced students rating their confidence levels as high or moderate, and only 35% of the less experienced. Although failing to reach statistical significance, a larger number of the IT students (79%) rated their confidence levels for the ICP coursework as high/moderate, compared to only 56% of the MIT students.

DISCUSSION

It is known that students who perform well in other subjects may produce disappointing results in programming. In the current study, there were correlations between the results gained in programming modules and spatial scores, with no correlations being found with the nonprogramming modules. These results suggest that spatial ability, as measured by a mental rotation test, is related to success in the programming subjects, while not appearing to be of relevance in the nonprogramming modules investigated.

When the relationship between spatial ability and ICP results were viewed for the separate master's streams, the correlation was found to be higher for the IT cohort. Some of this variation between the two groups may be accounted for by the larger range of mental rotation scores and ICP results in the IT stream. Additionally, this was a self-selected group of students who have chosen to continue with programming. Although not reaching significance, there was a trend for them to have greater programming experience, so this variable would have had less of an impact than in the wider master's cohort. Hassell (1982) showed a similar result for second- and final-year students. She looked at correlations between the EFT and measures of programming ability, and found a correlation ($r = 0.5$) for seniors, but a nonsignificant correlation for the second-year students. In the current study, there was a very strong correlation between the results for the IT students in the two programming modules, even though the courses were taught by two different lecturers. There was also a strong correlation between results in the two nonprogramming modules for the MIT students. However, there was no relationship between the programming module (ICP) and the nonprogramming module (IHF) for the group as a whole, nor between ICP and the other nonprogramming module (MAN) for the MIT students. This demonstrates that those who perform well in other subjects may underperform in programming.

As expected, these results suggest that other individual differences may have an impact on the results. Those who considered themselves to be more experienced in programming performed better in the introductory programming module. With this being a master's course, the students were generally older than undergraduate students, and may have had more opportunity for exposure to programming, either within a first-degree course, or from a previous work environment. The more experienced programmers tended to have higher spatial ability and, as expected, admitted to having greater confidence about the coursework. This confidence translated into better performance in the whole ICP module. This is confirmed by the fact that those who expected themselves to be more successful in ICP than nonprogramming modules generally performed better in the final mark. Rountree et al. (2002) found that expecting an A grade was the strongest indicator of success, with expected difficulty and workload making smaller but relevant contributions, trends reflected in the current study. Thus it would appear that self-efficacy is an important contributor to achievement. The data also show that those with low spatial ability were experiencing greater difficulty with ICP compared to the non-programming modules. There was no significant impact of a science background on the results, even though the majority of science graduates were enrolled in the IT stream. There were also no significant differences between males and females in mental rotation test score or ICP mark. This may have occurred because the sample size for the females was too low (20% of the group), a situation reflected in many computer science courses.

One other variable that needs to be considered is the programming activity itself. As shown in this study, the more experienced students had a higher spatial ability, but it is difficult to be sure if this is cause or effect. It is known that high spatial ability predisposes individuals to a choice of spatial subjects (such as engineering) and careers (such as architecture; Quaiser-Pohl & Lehmann, 2002; Smith, 1992). Consequently, it is possible that students with high spatial ability are choosing programming as an option because this skill allows them to excel. Alternatively, the very act of practicing programming may cause an increase in spatial ability. The task of learning to program has been shown to cause students to become more field independent (Cathcart, 1990), and improve their mental rotation ability (Miller, Kelly, & Kelly, 1998). However, these results were found when teaching Logo to schoolchildren; Logo programming requires children to imagine orientating with the turtle, a form of mental rotation (Miller et al., 1998). It would be interesting to give university students a pre- and posttest to see if an intensive programming course resulted in any improvement in mental rotation ability. Additionally, there is a wealth of evidence that training in the use of spatial tasks can improve scores in spatial tests, and this could be an important exercise for students wishing to improve their inherent programming ability. The authors believe that this training should be incorporated into the school curriculum, perhaps as early as 6 years of age (Jones & Burnett, 2006).

CONCLUSION

From the results of this analysis, there is evidence that spatial ability is important when learning to program. There are also interactions with other factors such as confidence levels, expected success, and programming experience. When the impact of these factors was reduced by focusing on a more advanced group of students, spatial ability was observed to have a stronger effect. Future studies need to be carried out on a larger cohort of students to allow statistical analysis of the relative contribution of each of the variables. It would also be beneficial to have a larger ratio of females in the student group to enable a study of gender effects.

This study provides an important contribution to knowledge about why some students struggle to achieve in introductory computer science courses, resulting in high attrition and failure rates. While spatial ability has been shown to be relevant, we do not feel that mental rotation capacity should be used as a means of predetermining programming aptitude, but should be considered while devising pedagogical interventions. Thought needs to be given to teaching methods and software visualizations that help students with low spatial ability to envisage abstract concepts and build better mental models (Wiedenbeck et al., 2004). The benefits of spatial training intervention also need to be assessed.

ENDNOTE

1. Available at http://www.uwm.edu/~johnchay/

REFERENCES

Bandura, A. (1986). *Social foundations of thought and action.* Englewood Cliffs, NJ, USA: Prentice Hall.

Beckwith, L., & Burnett, M. (2004). Gender: An important factor in end-user programming environments? In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing Languages and Environments* (pp. 107–114). Washington, DC: IEEE Computer Society.

Bergin, S., & Reilly, R. (2005). Programming: Factors that influence success. *SIGCSE Bulletin*, *37*(1), 411–415.

Bishop-Clark, C. (1995). Cognitive style, personality, and computer programming. *Computers in Human Behavior, 11,* 241–260.

Bradley, C. A. (1985). The relationship between students' information-processing styles and Logo programming. *Journal of Educational Computing Research, 1,* 427–433.

Bruckman, A., Jensen, C., & DeBonte, A. (2002). Gender and programming achievement in a CSCL environment. In G. Stahl (Ed.), *Proceedings of the Computer Supported Collaborative Learning* (CSCL) *2002 Conference* (pp. 119–127). Hillsdale, NJ, USA: Erlbaum.

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. In *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 49–52). New York: ACM Press.

Casey, M. B., Nuttall, R. L., Pezaris, E., & Benbow, C. P. (1995). The influence of spatial ability on gender differences in mathematics college entrance test scores across diverse samples. *Developmental Psychology*, *31*, 697–705.

Cathcart, W. (1990). Effects of Logo instruction on cognitive style. *Journal of Educational Computing Research*, *6*, 231–242.

Chen, C., Czerwinski, M., & Macredie, R. (2000). Individual differences in virtual environments: Introduction and overview. *Journal of the American Society for Information Science*, *51*, 499–507.

Cox, A., Fisher, M., & O'Brien, P. (2005). Theoretical considerations on navigating codespace with spatial cognition. In P. Romero, J. Good, E. Acosta Chaparro, & S. Bryant (Eds.), *Proceedings of the 17th Workshop of the Psychology of Programming Interest Group* (pp. 92–105). Retrieved May 21, 2006, from http://www.ppig.org/workshops/17th-programme.html

Curtis, B. (1981). Substantiating programmer variability. *Proceedings of the IEEE*, *69*, 846.

Cutmore, T. R. H., Hine, T. J., Maberly, K. J., Langford, N. M., & Hawgood, G. (2000). Cognitive and gender factors influencing navigation in a virtual environment. *International Journal of Human-Computer Studies*, *53*, 223–249.

Downs, R. M., & Stea, D. (1973). Cognitive maps and spatial behavior: Process and products. In R. M. Downs & D. Stea (Eds.), *Image and environments* (pp**.** 8–26). London: Edward Arnold.

Durndell, A., & Haag, Z. (2002). Computer self efficacy, computer anxiety, attitudes towards the Internet and reported experience with the Internet, by gender, in an East European sample. *Computers in Human Behavior*, *18*, 521–535.

Fincher, S., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., & Petre, M. (2005). *Programmed to succeed? A multi-national, multi-institutional study of introductory programming courses* (University of Kent, Computing Laboratory Technical Report 1-05). Retrieved May 21, 2006, from http://www.cs.kent.ac.uk/pubs/2005/2157/

Fisher, M., Cox, A., & Zhao, L. (2006). Using sex differences to link spatial cognition and program comprehension. In *Proceedings of the 22nd IEEE International Conference on Software Maintenance* (ICSM; pp. 289–298). Washington, DC: IEEE Computer Society.

Guzdial, M., & Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM*, *45*, 17–21.

Halpern, D. F. (2000). *Sex differences in cognitive abilities*. Mahwah, NJ, USA: Lawrence Erlbaum Associates.

Hassell, J. (1982). Cognitive style and a first course in computer science: A success story. *AEDS Monitor*, *21*, 33–35.

Jones, S., & Burnett, G. (2006). Give the girls a chance: Should spatial skills training be incorporated into the curriculum? In K. Morgan, C. A. Brebbia, & J. M. Spector. (Eds.), *The Internet Society II: Advances in education, commerce & governance* (pp. 105–114). Southampton, UK: WITpress.

Jones, S., & Burnett, G. (2007a). Spatial skills and navigation of source code. In *Proceedings of the 12th Annual Conference on Innovation and Technology in Computer Science Education* (ITiCSE '07; pp. 231–235). New York: ACM.

Jones, S., & Burnett, G. E. (2007b). Children's navigation of hyperspace: Are spatial skills important? In *Proceedings of the Sixth IASTED International Conference on Web-Based Education* (pp. 643–648). Anaheim, CA, USA: ACTA Press.

Kimura, D. (1999). *Sex and cognition*. Cambridge, MA, USA: MIT Press.

Mancy, R., & Reid, N. (2004). Aspects of cognitive style and programming. In E. Dunican & T. Green (Eds.), *Proceedings of the Sixteenth Annual Workshop of the Psychology of Programming Interest Group* (PPIG '04; pp. 1–9). Retrieved January 3, 2005, from http://www.ppig.org/workshops/16th-programme.html

Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: What's the connection? *Communications of the ACM*, *29*, 605–610.

McKenna, F. P. (1984). Measures of field dependence: Cognitive style or cognitive ability? *Journal of Personality and Social Psychology*, *47*, 593–603.

Miller, R. B., Kelly, G. N., & Kelly, J. T. (1998). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, *13*, 348–357.

Mumtaz, S. (2001). Children's enjoyment and perception of computer use in the home and the school. *Computers & Education*, *36*, 347–362.

Pease, A., & Pease, B. (2001). *Why men don't listen and women can't read maps*. London: Orion.

Quaiser-Pohl, C., & Lehmann, W. (2002). Girls' spatial abilities: Charting the contributions of experiences and attitudes in different academic groups. *British Journal of Educational Psychology*, *72*, 245–260.

Rountree, N., Rountree, J., & Robins, A. (2002). Predictors of success and failure in a CS1 course. *SIGCSE Bulletin*, *34*, 121–124.

Schumacher, P., & Morahan-Martin, J. (2001). Gender, Internet and computer attitudes and experiences. *Computers in Human Behavior*, *17*, 95–110.

Scragg, G., & Smith, J. (1998). A study of barriers to women in undergraduate computer science. *SIGCSE Bulletin*, *30*, 82–86.

Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems*. Cambridge, MA, USA: Winthrop Publishers Inc.

Smith, P. (1992). Spatial ability and its role in United Kingdom education. *Vocational Aspect of Education*, *44*, 103–106.

Subrahmanyam, K., Greenfield, P. M., Kraut, R., & Gross, E. (2001). The impact of computer use on children's and adolescents' development. *Applied Developmental Psychology*, *22*, 7–30.

Tolhurst, D., Baker, B., Hamer, J., Box, I., Lister, R., Cutts, Q., Petre, M., de Raadt, M., Robins, A., Fincher, S., Simon, S., Haden, P., Sutton, K., Hamilton, M., & Tutty, J. (2006). Do map drawing styles of novice programmers predict success in programming? A multi-national, multi-institutional study. In D. Tolhurst & S. Mann (Eds.), *Proceedings of the 8th Australian Conference on Computing Education* (pp. 213–222). Hobart, Australia: Australian Computer Society, Inc.

Vandenberg, S. G., & Kuse, A. R. (1978). Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and Motor Skills*, *47*, 599–604.

Velez, M., Silver, D., & Tremaine, M. (2005). Understanding visualization through spatial ability differences. In *Proceedings of the IEEE Visualization 2005 Conference* (VIS 2005; pp. 511–518). New York: IEEE.

Voyer, D., Nolan, C., & Voyer, S. (2000). The relation between experience and spatial performance in men and women. *Sex Roles: A Journal of Research*, *43*, 891–915.

Webb, N. M. (1984). Microcomputer learning in small groups: Cognitive requirements and group processes. *Journal of Educational Psychology, 76*, 1076–1088.

Werner, S. Krieg-Bruckner, B., Mallot, H. A., Schweizer, K., Freksa, C., & Jahrestagung, G. (1997). Spatial cognition: The role of landmark, route and survey knowledge in human and robot navigation. In M. Jarke, K. Pasedach, & K. Pohl (Eds.), *Informatik '97* (pp. 41–50). New York: Springer.

Werth, L. H. (1986). Predicting student performance in a beginning computer science class. In J. C. Little & L. N. Cassel (Eds.), *Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education* (SIGCSE '86; pp. 138–143). New York: ACM.

Westerman, S. J., & Cribbin, T. (1999). Navigating virtual information spaces: Individual differences in cognitive maps. In *Proceedings of the UK Virtual Reality Special Interest Group* (UKVRSIG '99), Salford University, UK.

Wiedenbeck, S., LaBelle, D., & Kain, V. N. R. (2004). Factors affecting course outcomes in introductory programming. In *Proceedings of the Sixteenth Annual Workshop of the Psychology of Programming Interest Group* (PPIG ’04) Retrieved May 15th 2005 from http://www.ppig.org/workshops/16th-programme.html

Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: A study of twelve factors. *INROADS of SIGCSE, 33*, 184–188.

Authors’ Note

The authors wish to thank Professor Emeritus John C. Hay, the University of Wisconsin, Milwaukee, for permitting use of, and customizing, his version of the 3-D Mental Rotation Test.

All correspondence should be addressed to:
Sue Jones
School of Computer Science
University of Nottingham
Nottingham
NG8 1BB
United Kingdom
s.jones@nottingham.ac.uk

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi