Spatial Ability and Learning to Program

Sue Jones and Gary Burnett

School of Computer Science and IT, University of Nottingham, Nottingham, UK
{sjj, geb}@cs.nott.ac.uk

Abstract. Results in introductory programming modules are often disappointing, and various individual differences have been found to be of relevance. 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 a Masters in IT 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 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. Data from a subgroup of students who had chosen to continue with programming showed a stronger correlation between spatial skills and programming success (r=0.57). These results are discussed in relation to the accessibility of programming to learners with low spatial ability.

1   Introduction

Results in introductory programming modules are often disappointing [1], with reports of up to 30% of students failing to complete [2]. Students that perform well in other subjects may not achieve equivalent success in programming tasks [3, 4], lose confidence, and give up computer science courses. Irrespective of experience, some programmers appear to be more skilled than others. Curtis found a range of 23 to 1 in debugging performance [5], and Shneiderman reported differences in performance of 100 to 1 amongst programmers of similar programming experience [6].

   Previous research has focussed on why some students under-perform 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, but also considers how some of the other individual differences are related in their influence on learning to program. The first section 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.

2 Spatial Ability and Mental Models

One individual difference considered to have some relevance to programming aptitude is spatial ability, a cognitive characteristic that gives a measure of the ability to conceptualise the spatial relations between objects [7]. Spatial ability has been shown to be important for navigation in the real world, and in an abstract information space such as hypertext [8]. 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 utlilised in a real environment [9].

There are few studies looking at relations between spatial ability and individual programming performance. Mayer *et al* [10] showed that success in learning Basic was related to spatial ability (r=0.31, p<0.05). Fincher *et al* [3] showed an overall small positive correlation between performance in a spatial visualisation test and marks achieved in the introductory programming courses at eleven institutions (r=0.17, p=0.047). Both these studies used versions of the Paper Folding Test designed to measure spatial visualisation, one of the different types of task included in the broad cognitive category of spatial ability [7].

Webb [11], 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 score in a mathematics reasoning test. Webb utilised three measures of spatial ability, one of which was the paper folding test, but the others requiring mental rotation of figures. Mental rotation (MR) 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 [7]. Fisher *et al* [12] used the Vandenberg and Kuse MR test (MRT) 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 demonstrated differences in navigation of source code in a code comprehension exercise, with those with higher spatial ability jumping between functions more frequently and making more interclass jumps (ie moving between files) [13]. 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 [14], and people build these while familiarising themselves with an environment. They become disorientated if this internal map does not correspond to the physical representation of the environment [15]. In relation to cognitive maps of program code, Fisher *et al* [12] 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 within the code."* Hence the mental model is an abstraction of the program formed from the piecing together of various kinds of information extracted while navigating the source code. Wiedenbeck *et al* stress the importance of a good mental model in program understanding [16].

Tolhurst *et al* 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 [17]. 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 [18]. 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 [19].

3   Relation of Spatial Ability to Other Individual Differences

3.1 Gender

Girls are under-represented in university computer science (CS) courses [4]. Suggested reasons for this include sex stereotyping and a greater male exposure to computers and computer games [20, 21]. Boys appear to have a more confident, positive attitude to computers [22]. Subrahmanyam *et al* [23] comment 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. There is a wealth of evidence of gender differences in spatial ability, with females appearing to under-perform in certain measures, such as mental rotation (MR) [24].

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 [25]. Byrne and Lyons [4] found no significant difference in performance between male and female students on a first year programming module, possibly because the module was part of a BA honours degree with a preponderance of female students (61%). However, other studies have also not found the expected gender difference [26, 27], 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 [28]. Fisher *et al* [12] hypothesise that females prefer a more low-risk, bottom-up approach and males a more high-risk, abstract, top-down approach to program development and comprehension. Bradley demonstrated that top-down processing was positively related to Logo programming success [29].

3.2 Self Efficacy/Comfort

Self-efficacy relates to how we estimate our capability to perform well in a certain context [30]. 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* [27] surveyed students early on in a first year

computer science course, and found that their expectations of how they were going to perform was the biggest indicator of success. Surprisingly, students predicted the outcomes very early on 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 [26, 31]. Students with low computer experience are likely to be less confident and more anxious when starting a programming course [4]. It has also been suggested that males tend to show greater self-efficacy and lower computer anxiety than females [22, 28]. Having a good mental model increases self-efficacy by enabling program comprehension [16].

3.3 Previous Academic Exposure

Previous programming experience seems to relate to success in introductory programming courses [27, 31], with Wiedenbeck *et al* linking this with self-efficacy [16]. Boys are more likely than girls to have previous programming experience [32]. 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 [4]. Various studies have found relations between mathematics results and success in learning programming [11, 31, 33]. Byrnes and Lyons [4] found a relationship between results in the mathematics section of the Leaving Certificate and results in 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 and foreign language results and programming achievement. Others have found a similar relationship with science [26, 33].

The ability to succeed in mathematics has been related to spatial ability [34], and Casey *et al* [35] show that males outperformed females in both the Vandenberg and Kuse MR test for spatial ability and a mathematics aptitude test, with mental rotation predicting mathematics aptitude for the female samples. When MR ability was statistically adjusted for, the gender difference in mathematics achievement was eliminated in most of the groups studied.

3.4 Cognitive Style

Cognitive *abilities* are specific to a particular domain of content or function, such as verbal, numerical or spatial ability [36]. 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 are more to do with organization and control of cognitive processes. Consequently, there appears to be interaction between cognitive abilities and styles, with field-dependency being the style most associated with spatial ability [37]. Mckenna [38] presents arguments debating whether field dependence, often measured by the embedded figures test (EFT), is a cognitive style or ability. 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 [39] 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 compared field dependency and marks in various assessments on an introductory programming course. Field dependency was measured using the embedded figures test (EFT), and the results showed positive correlations with marks on the different assessments, with a good correlation (r= 0.40) with the final examination [1].

4   Motivation for the study
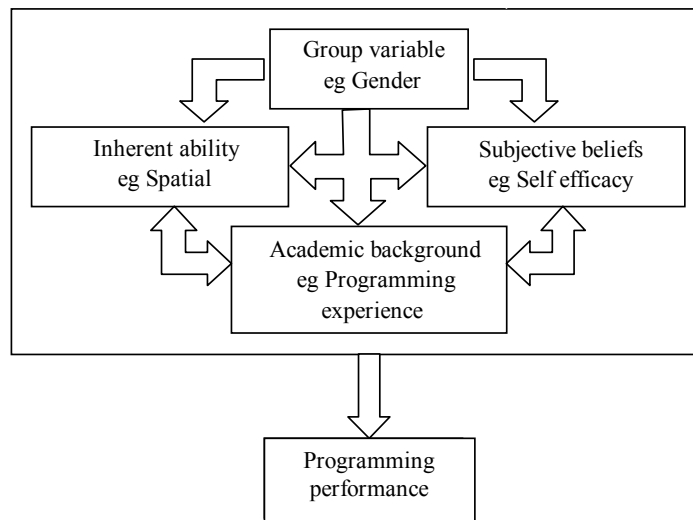
Fig. 1. Potential interactions between individual differences and programming performance

In summary, individual differences with varying degrees of impact on programming performance have been discussed in relation to spatial ability. 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 add to the research knowledge regarding any effect of spatial ability and other interacting factors on programming performance. The individual differences considered are shown in figure 1.

5 The Study

Over the academic year 2005/06, a test of MR 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.

5.1 Data Collection

Participants consisted of 49 volunteers (average age 26 years) from students enrolled on a one-year Masters conversion course at the University of Nottingham, UK (as a conversion course, students did not have a Computer Science first degree). An introductory programming module (ICP) was compulsory for all students in the first semester. The course consisted of two streams. Students on the Masters in IT (IT, n=28) had a first-degree in a science or engineering subject and continued with further compulsory programming modules in the second semester. Those studying for the Masters in the Management of IT (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 on the IT stream.

Data on the participants' academic history and perceived programming experience were collected by questionnaire. At the end of semester one (December 2005), and before taking the introductory programming examination, the students were asked how they rated their confidence levels on their last (as yet unmarked) programming coursework. They also rated the introductory programming module (ICP) in comparison to other non-programming modules on the parameters of difficulty, workload and expected success, similar to Rountree *et al* [27]. The results of the final marks in the following modules were collated:

- Semester one – compulsory modules
    - o Introduction to Computer Programming (ICP) – with 50% coursework (2 programming assignments) and 50% examination. The language taught was Java.
    - o Introduction to Human Factors (IHF) – a non-programming module with 25% coursework and 75% final examination.
- Semester two
    - o Object Oriented Systems (OOS) – with two programming assignments contributing 50% to the final mark, and a final examination. This module was compulsory for the Masters in IT. The language taught was C++.
    - o Management of IT (MAN) – a non-programming module with 25% coursework and 75% examination. This module was compulsory for those taking the Masters in MIT.

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

Individual differences in spatial skills were measured using a version of the 3D mental rotation test found at Psychlab OnLine (available at http://www.uwm.edu/~johnchay/). The test is a modified version of the Vandenberg

and Kuse mental rotation (MR) test [40], and was customised for this study by Professor Hay (the University of Wisconsin, Milwaukee). In the version used, the students were asked to complete all the 30 MR examples as quickly as possible, with equal emphasis being given to accuracy and speed. This was an online test, and once an answer was submitted, there was no recourse to correcting it, and no feedback was provided on the correctness of their answers. On 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. A value 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.

5.2 Results

5.2.1 Module Marks

Table 1. Correlation analysis for mental rotation (MR) scores and module marks

		Programming modules			Non-programming 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
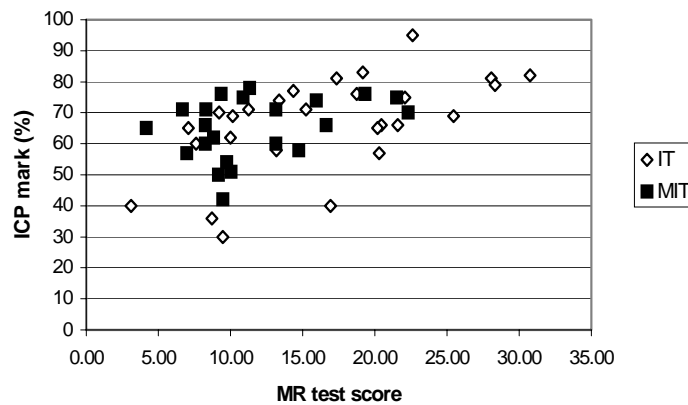
Fig. 2. Scatterplot of ICP results as a function of MR test scores for the two Masters streams

As can be seen in Table 1, there was a strong correlation between MR scores and results in the programming modules, but this was not reflected in the non-programming modules. There was a stronger correlation for the IT students (those who carried on with programming) in the introductory module (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 2, the spread of results in the ICP module was greater for the IT stream (30% to 95%; mean=66.59, SD=15.43) than the MIT stream (42% to 78%; mean=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 amongst the IT students (3.09 to 30.76; mean=16.47, SD=7.22) than the MIT students (4.18 to 22.30; mean=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 non-programming modules, IHF and MAN. There were no correlations between the programming and non-programming modules (see Table 2).

Table 2. Pearson correlation analysis for module marks (significance values in brackets)

	Programming modules		Non-programming 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)	

5.2.2 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.

Table 3. T-tests for questionnaire results

		MR test score			ICP mark		
		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	Prof/intermed	16.65	7.41	<0.05	72.50	9.54	<0.005
	Novice/none	12.31	5.51		60.60	14.34	
Confidence	High/mod	15.95	7.15	0.09	72.59	8.23	<0.001
	Little/none	12.23	5.85		56.47	14.83	

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 (just failed to reach statistical significance).

Table 4. Results of the Kruskall-Wallis analysis of student expectations

Parameter	Categories (collapsed)	MR test score			ICP mark		
		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		

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

Table 5. Results of chi-square on programming experience and confidence

Parameter	Categories (collapsed)	Masters		Confidence	
		χ2	p	χ2	p
Programming experience	Professional/ intermediate Novice/none	3.25	0.07	13.17	<0.001
Confidence	High/ moderate Little/none	1.36	0.24		

There was a non-significant trend for those on the IT stream to have more programming experience, with 74% claiming to be professional/intermediate, compared to only 39% of the MIT students. There was a very obvious impact of programming experience on confidence levels (see Table 5), 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.

6   Discussion

It is known that students that 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 non-programming 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 non-programming modules investigated.

When the relationship between spatial ability and ICP results were viewed for the separate Masters 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 MRT 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 Masters cohort. Hassell 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 non-significant correlation for the second year students [41]. 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 non-programming modules for the MIT students. However, there was no relationship between the programming module (ICP) and the non-programming module (IHF) for the group as a whole, nor between ICP and the other non-programming module (MAN) for the MIT students. This demonstrates that those that perform well in other subjects may under-perform in programming.

Fisher *et al* [12] argue that because source code is linear, MR will be less relevant for men than other cognitive attributes such as object location memory will be for women. However, the studies utilising 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 visualisation. The Paper Folding Test requires subjects to imagine the result after folding an object [42], but this process does not require mental rotation [43]. The spatial ability test used in the current study was a variation of the Vandenberg and Kuse mental rotation test [40]. Kimura makes the link between success in this task and the capacity to build a mental

model, or cognitive map, of an environment [42]. Kimura suggests that good MR capacity enables us to recognise 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 [16], and perhaps a good MR capacity is allowing formation of more accurate mental models of programs, which has in turn been linked to increasing beliefs of self-efficacy.

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 Masters 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 to 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 non-programming modules generally performed better in the final mark. Rountree *et al* [27] 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 science background on the results, even though the majority of science graduates were enrolled on the IT stream. There were also no significant differences between males and females in MR 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 on 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 pre-disposes individuals to choice of spatial subjects (such as engineering) and careers (such as architecture) [44, 45]. 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 [46], and improve their mental rotation ability [47]. However, these results were found when teaching Logo to school children; Logo programming requires children to imagine orientating with the turtle, a form of MR [47]. It would be interesting to give University students a pre-and post- test and see if an intensive programming course resulted in any improvement in MR 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 [48].

7 Conclusion

If a student struggles to achieve in an introductory module, they are unlikely to enroll on a more advanced programming module [16], and may give up the course altogether. The reasons for this discrepancy need to be investigated and, ideally, addressed. From the results of this analysis, there is evidence that spatial ability is important when learning to program. However, there are 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. These analyses 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.

The authors do not feel that MR capacity should be used as a means of pre-determining programming aptitude, but should be considered while devising pedagogical interventions. Thought needs to be given to teaching methods and software visualisations that help students with low spatial ability to envisage abstract concepts and build better mental models [16]. The benefits of spatial training intervention also need to be assessed.

8 Acknowledgments

References

1. Mancy, R., Reid, N.: Aspects of cognitive style and programming. In: Proceedings of the 16th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2004). Institute of Technology, Carlow, Ireland (2004) 1-9
2. Guzdial, M., Soloway, E.: Teaching the Nintendo generation to program. Commun. ACM **45** (2002) 17-21
3. Fincher, S., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M.: Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses. University of Kent, Canterbury, Kent, UK. (2005)
4. Byrne, P., Lyons, G.: The effect of student attributes on success in programming. In: Proceedings of the 6th Annual Conference on innovation and Technology in Computer Science Education. Canterbury, UK: ACM Press, New York, NY (2001) 49 - 52
5. Curtis, B.: Substantiating programmer variability. Proceedings of IEEE **69** (1981) 846-846
6. Shneiderman, B.: Software Psychology - Human Factors in Computer and Information Systems. Winthrop Publishers Inc, Cambridge, Massachusetts (1980)
7. Halpern, D.F.: Sex differences in cognitive abilities. 3rd edn. Lawrence Erlbaum Associates, Mahwah, New Jersey (2000)

8. Jones, S., Burnett, G.E.: Children's Navigation of Hyperspace – Are Spatial Skills Important? In: Proceedings of Web-based Education - 2007. Chamonix, France: Acta Press (2007)
9. Cox, A., Fisher, M., O'Brien, P.: Theoretical considerations on navigating codespace with spatial cognition. In: Proceedings of the 17th Workshop of the Psychology of Programming Interest Group (PPIG 2005). Sussex University, UK (2005)
10. Mayer, R.E., Dyck, J.L., Vilberg, W.: Learning to program and learning to think: what's the connection? Commun. ACM **29** (1986) 605-610
11. Webb, N.M.: Microcomputer Learning in Small Groups: Cognitive Requirements and Group Processes. Journal of Educational Psychology **76** (1984) 1076-88
12. Fisher, M., Cox, A., Zhao, L.: Using Sex Differences to Link Spatial Cognition and Program Comprehension. In: Proceedings of the 22nd IEEE international Conference on Software Maintenance. Washington, DC (2006) 289-298
13. Jones, S., Burnett, G.: Spatial skills and navigation of source code. Paper to be presented at the 12th Annual Conference on Innovation and Technology in Computer Science Education.
14. Downs, R.M., Stea, D.: Cognitive maps and spatial behavior: process and products. In Downs, R.M. and Stea, D., Eds: Image and Environments, Edward Arnold, London, UK (1973) 8-26.
15. Westerman, S.J., Cribbin, T.: Navigating virtual information spaces: Individual differences in cognitive maps. In: Proceedings of the UK Virtual Reality Special Interest Group. Salford, UK (1999)
16. Wiedenbeck, S., LaBelle, D., Kain, V.N.R.: Factors Affecting Course Outcomes in Introductory Programming. In: Proceedings of the Psychology of Programming Interest Group (PPIG 2004). Carlow, Ireland (2004)
17. 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.: Do map drawing styles of novice programmers predict success in programming? A multi-national, multi-institutional study. In: Proceedings of the 8th Australian conference on Computing education. Hobart, Australia (2006) 213 - 222
18. Werner, S., Krieg-Brückner, B., Mallot, H.A., Schweizer, K., Freksa, C., Jahrestagung, G.: Spatial Cognition: the role of landmark, route and survey knowledge in human and robot navigation. (1997)
19. Cutmore, T.R.H., Hine, T.J., Maberly, K.J., Langford, N.M., Hawgood, G.: Cognitive and gender factors influencing navigation in a virtual environment. International Journal of Human-Computer Studies **53** (2000) 223-249
20. Mumtaz, S.: Children's enjoyment and perception of computer use in the home and the school. Computers & Education **36** (2001) 347-362
21. Schumacher, P., Morahan-Martin, J.: Gender, Internet and computer attitudes and experiences. Computers in Human Behavior **17** (2001) 95-110
22. Durndell, A., Haag, Z.: 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** (2002) 521-535
23. Subrahmanyam, K., Greenfield, P.M., Kraut, R., Gross, E.: The impact of computer use on children's and adolescent's development. Applied developmental psychology **22** (2001) 7-30
24. Voyer, D., Nolan, C., Voyer, S.: The relation between experience and spatial performance in men and women. Sex Roles - A Journal of Research **43** (2000) 891-915
25. Scragg, G., Smith, J.: A study of barriers to women in undergraduate computer science. SIGCSE Bull. **30** (1998) 82-86
26. Bergin, S., Reilly, R.: Programming: factors that influence success. ACM SIGCSE Bulletin, Proceedings of the 36th SIGCSE technical symposium on Computer science education **37** (2005) 411-415

27. Rountree, N., Rountree, J., Robins, A.: Predictors of success and failure in a CS1 course. SIGCSE Bulletin **34** (2002) 121-124
28. Beckwith, L., Burnett, M.: 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. Rome, Italy (2004) 107-114
29. Bradley, C.A.: The Relationship between Students' Information-Processing Styles and Logo Programming. Journal of Educational Computing Research **1** (1985) 427-33
30. Bandura, A.: Social foundations of thought and action. Prentice Hall, Englewood Cliffs, NJ (1986)
31. Wilson, B.C., Shrock, S.: Contributing to success in an introductory computer science course: a study of twelve factors. Inroads of SIGCSE **33** (2001) 184-188
32. Bruckman, A., Jensen, C., DeBonte, A.: Gender and programming achievement in a CSCL environment. In: Proceedings of the CSCL 2002. Boulder, CO (2002)
33. Werth, L.H.: Predicting student performance in a beginning computer science class. In: Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education. Cincinnati, Ohio, United States: ACM Press, New York, NY (1986) 138-143
34. Pease, A., Pease, B.: Why men don't listen and women can't read maps. Orion, London (2001)
35. Casey, M.B., Nuttall, R.L., Pezaris, E., Benbow, C.P.: The influence of spatial ability on gender differences in mathematics college entrance test scores across diverse samples. Developmental Psychology **31** (1995) 497-705
36. Sjölinder, M.: Individual differences in spatial cognition and hypermedia navigation. In Dahlbäck, N., Ed: Exploring navigation: Towards a framework for design and evaluation of navigation in electronic spaces, Swedish Institute of Computer Science, Stockholm (1998) 61-72.
37. Chen, C., Czerwinski, M., Macredie, R.: Individual differences in virtual environments - Introduction and overview. Journal of the American Society for Information Science **51** (2000) 499-507
38. McKenna, F.P.: Measures of field dependence: Cognitive style or cognitive ability? Journal of Personality and Social Psychology **47** (1984) 593-603
39. Bishop-Clark, C.: Cognitive style, personality, and computer programming. Computers in Human Behavior **11** (1995) 241-260
40. Vandenberg, S., G; Kuse, Allan R.: Mental rotations, a group test of three-dimensional spatial visualization. Perceptual and Motor Skills **47** (1978) 599-604
41. Hassell, J.: Cognitive Style and a First Course in Computer Science: A Success Story. AEDS Monitor **21** (1982)
42. Velez, M., Silver, D., Tremaine, M.: Understanding Visualization through Spatial Ability Differences. In: Proceedings of Visualization 2005. IEEE. Minneapolis, Min (2005) 511-518
43. Kimura, D.: Sex and Cognition. MIT Press, Cambridge, Massachusetts (2000)
44. Smith, P.: Spatial ability and its role in United Kingdom education. Vocational Aspect of Education **44** (1992) 103-106
45. Quaiser-Pohl, C., Lehmann, W.: Girls' spatial abilities: Charting the contributions of experiences and attitudes in different academic groups. British Journal of Educational Psychology **72** (2002) 245 - 260
46. Cathcart, W.: Effects of Logo instruction on cognitive style. Journal of Educational Computing Research **6** (1990) 231-242
47. Miller, R.B., Kelly, G.N., Kelly, J.T.: Effects of Logo computer programming experience on problem solving and spatial relations ability. Contemporary Educational Psychology **13** (1998) 348-357
48. Jones, S., Burnett, G.: Give the girls a chance: should spatial skills training be incorporated into the curriculum? In K. Morgan, C.A. Brebbia, and Spector., J.M., Eds: The Internet Society II: Advances in Education, Commerce & Governance, WITpress (2006).