An Assessment of Visual Representations for the 'Flow of Control'

Jarinee Chattratichart and Jasna Kuljis
Department of Information Systems and Computing
Brunel University, United Kingdom
{Jarinee.Chattratichart, Jasna.Kuljis}@brunel.ac.uk

Keywords: POP-III.D. visualisation

Abstract

Historically, graphical programming and visual programming most commonly used arrows, lines or juxtaposition to represent visually the order or a sequence of actions. The choice of representation can affect the performance of a user of such systems in following a sequence of actions, both in terms of time and of accuracy. However, there is little empirical evidence which would justify the use of one representation over the other. This paper addresses these issues and examines what should govern the choice of representation for the 'flow of control'. An experimental study that compares how the three representations affect users' performance is conducted on several groups of undergraduate students. The result suggests that arrow is the best performer and that juxtaposition is the most error-prone.

1 Introduction

Computer scientists have been using various diagramming techniques as an aid to the analysis of programming problems and in communicating their thinking to others. In graphical programming various diagramming techniques have been used as program specification tools. A flowchart, for example, represents a diagram assembled of graphical objects representing an algorithm of a program. Since its introduction, many other diagramming techniques have evolved. All of them were supposed to aid programming. When computers became more powerful and less expensive the use of graphical objects, such as icons, pictures and images, have become more common. The end of the 1980's marked a sudden disappearance of graphical programming from the literature which evolved into 'visual programming' (VP). Graphical programming and visual programming have one thing in common: programs are specified visually.

Our research focuses on how the visualisation of program specification can aid novice programmers. The question that needs to be answered before any visual representation and program specification technique can be considered is: Do existing visual representations make programming any easier as is currently claimed. Only then we can examine which representations would be most appropriate.

In both visual and graphical programming program constructs are represented with visual/graphical objects. Most existing techniques use lines or arrows (of various thickness) to represent order or sequence. Juxtaposition is used in Nassi Shneiderman's Diagram (Nassi and Shneiderman 1973) and its variants and also in some visual programming systems. The choice of representation is not necessarily governed by consideration of the user. Other considerations may govern the choice, like, for example, saving screen space for choosing the juxtaposition, the line for its bi-directional property, and the arrow because it is commonly used for representation of direction.

This paper discusses what should govern the choice of representation for the 'flow of control' in the visual representation of programming problems. The choice of representation can affect the performance of a user in following a sequence of actions, both in terms of speed and of accuracy. Section 2 gives the research background to the problem. Section 3 describes the first set of experiments carried out to assess the performance of the three visual representations for direction or sequence. In section 4 we discuss the results of the experiments. Section 5 describes the second set of experiments carried out to confirm the results of the first set of experiments. Section 6 considers outcomes from both sets of experiments. In section 7 we draw some conclusions and present the plans for future work.

2 Research Background

Graphical programming and visual programming languages both claim to make programming easier. Myers (1990) stated that a more visual style of programming than flowcharts, which have long been known to be helpful aids in program understanding, could be easier to understand and generate for novice programmers. However, experiments which compared the comprehensibility of textual with visual programs failed to give support to such claims (Green *et al.* 1991; Moher *et al*. 1993). The choice of graphical notations and diagramming techniques is rarely empirically justified. Both graphical programming and visual programming languages use arrows, lines or juxtaposition to express order or sequence. Again, there is not enough empirical evidence to justify the use of one over the others. There are several empirical studies with mixed results on whether flowcharts aid program comprehension and debugging (Shneiderman, *et al*. 1977; Brooke and Duncan 1980a, 1980b; Gilmore and Smith 1984). Cunniff *et al.* (1986) compared the number of bugs and the bug types (see Spohrer *et al.* 1985) made on nine programming problems written in FPL (First Programming Language) by six students to those obtained by the experiments on Pascal programs written by novice programmers at the Yale University. FPL, a graphical programming language developed at Columbia University for novice programmers, allows writing program specification with FPL icons. Each FPL icon represents one programming construct. Cunniff *et al.* (1986) concluded that FPL performed equally well as Pascal in some bug types but much better in others.

Experts and novices use programming plans to accomplish programming tasks (Soloway and Ehrlich 1984; Rist 1986). These plans are '*..program fragments that represent stereotypic action sequences in programming ..*' (Soloway and Ehrlich 1984). Therefore a language such as BridgeTalk (Bonar and Liffick 1990) that allows specifying programs by using programming plans takes a more natural approach and hence imposes less cognitive load on users than conventional languages. Each icon in BridgeTalk looks like a piece in a jigsaw puzzle and it uniquely represents one programming plan. Plans are then connected the same way as in a jigsaw puzzle. Even though BridgeTalk evolved through six generations of evaluation and testing on students the language had not been systematically tested against a standard programming language like Pascal (Bonar and Liffick 1990). FPL uses line and BridgeTalk uses juxtaposition to represent a sequence. Regardless of the attempts by the developers of FPL and BridgeTalk to evaluate their design it is not clear what governed them to choose line and juxtaposition, respectively, over other sequence representations.

One of the objectives of our work is to find a more natural way for novice programmers to specify programs. By 'natural' we mean intuitive, easy to understand and to use, and less error-prone. Invariably all programming problems require indicating the sequence of actions. We investigated which of the three of the most common visual representations, arrow, line and juxtaposition, represents the 'flow' of control most naturally. An experimental study that compares how these three representations affect users' performance in terms of time and of accuracy is described in the section that follows.

3 Experiments

The purpose of experiment was to assess which of the three most common flow representations (arrow, line, and juxtaposition) is the most intuitive. We assume that such representation will cause least difficulties and therefore have the best performance in both the accuracy and in the speed of following the direction. Our intuition was that arrow is the most natural representation. However, we had to prove that. We have conducted two experiments on 84 first year programming students from three universities.

The experiments involved testing the ability of the subjects to follow a direction. We did not use any commonly used graphical programming diagramming technique, such as flowchart or Nassi Shneiderman diagram, since each technique has its own inherent concepts that need to be learned and understood. Our experiments used what we call a ‘maze’ which is really a route map representing all possible routes connecting *n* starting points (names of travellers) to *m* destination points (cities). It required participants only to follow a route/direction. We chose the maze paradigm because this kind of diagram is typically used to represent bus routes, train networks, and underground maps and therefore all our subjects would be familiar in using this type of representation. Our maze consists of only three types of objects: starting points, destination points, and one directional notation for each maze to indicate paths or routes. Our hypothesis was that the speed and the accuracy in following a path in the diagram would be affected by the choice of directional representation.

Methods Participants in the first set of experiments consisted of two separate groups of first year Computer Science undergraduate students. Nineteen students participated in the experiment at the University of Westminster and eighteen participated at Goldsmiths College, University of London. Both groups were doing their first programming language courses, Visual Basic at Westminster and Pascal at Goldsmiths.

The experiments were carried out on computer-based mazes written in Visual Basic. Each group was assembled in a computer laboratory on their respective campuses. Participants performed nine tasks on three mazes that differed only in the types of directional representation: arrow, line or juxtaposition. The maze program also measured and recorded the speed of performance in completing each task and the given answer. Figure 1 illustrates the three mazes. Every route in each maze consists of 28 to 31 steps and 6 to 8 turns including one backward turn.

Procedure. The maze program first introduced participants to the nature of the experiment, emphasising that correctness and reaction time were essential. Description and examples of the three directional representations were then given, followed by a practice test that mimicked the real test using a smaller sample maze. The mazes were described as consisting of routes that take the travellers, whose names were listed on the left-hand side of the maze, to one of the destinations on the right-hand side of the maze. During the practice session participants were free to ask questions and could navigate between practice screens. The maze program allowed them to start the real test whenever they were ready after they had done at least one practice test.

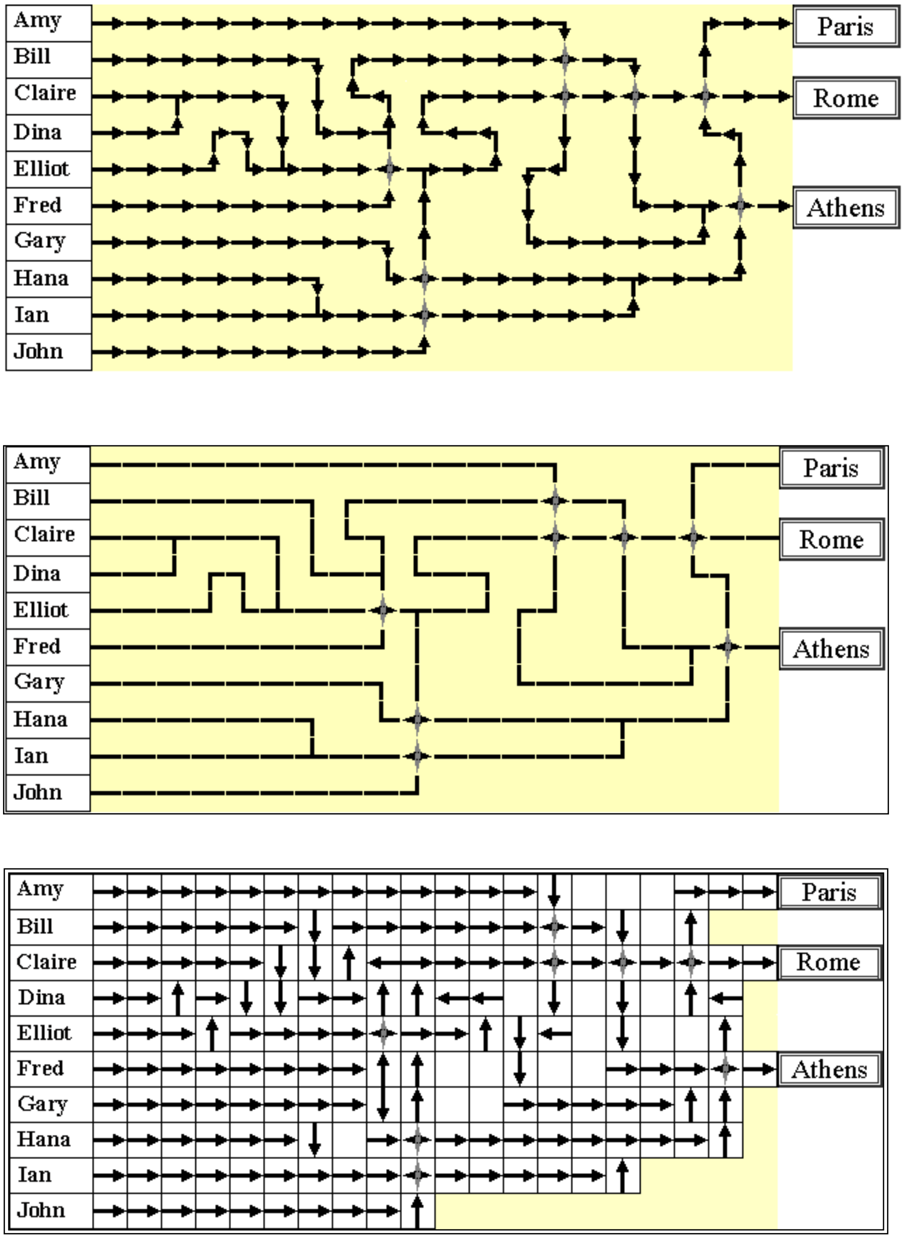

Figure 1: The Arrow, the Line, and the Juxtaposition Maze

In the real test each participant was asked to answer nine questions, three per representation. The routine was as follows: on the first screen the participant was shown for a few seconds only the travellers and the destinations parts of a maze as in figure 2. Then the screen was blanked for two seconds before the incomplete maze was displayed again, but this time with a question asking the participant to give the destination for a specified traveller. When the participant clicked the mouse on the specified traveller's name the missing routes appeared in the maze and the clock started measuring the time taken to answer the question. The participant then mentally followed a route leading from the traveller in question to a destination city which they then pointed and clicked with the mouse. When that happened the clock stopped and the time for that participant and the task was recorded. Before moving to the next question, the participant was asked to confirm the answer and was allowed to change it. The answer was then recorded.

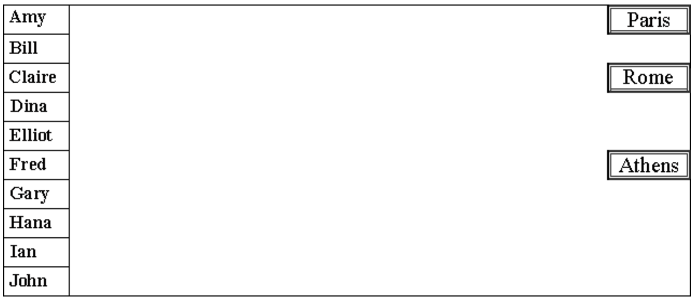

Figure 2: The first maze screen

The order in which each directional representation was used in mazes presented to participants was randomised. However, all three trials for the same representation were completed before another set of trials for a different representation followed. For example, a participant would be given three questions for the *Line* maze followed by three questions for the *Juxtaposition* maze, and finally three questions for the *Arrow* maze. For another participant the order of mazes might be different.

4 Results of Experiments

Performance was considered in terms of the response time (time needed to provide an answer) and the accuracy of answers. Two t-tests for unrelated data of the two groups of participants are presented first, followed by ANOVA analyses.

A t-test of the two groups on total time taken by each participant revealed no significant difference between the two groups, t(33) = 1.267, ns. A t-test of the two groups on total score (number of correct answers) revealed no significant difference between the two groups, t(33) = 0.144, ns. The means of total score were 7.00 for the Westminster group and 7.12 for the Goldsmiths group. However, participants' total score varied from 2 to 9 (the maximum). Those who scored poorly may not have tried their best or did not spend enough time in the practice session to understand the rules and notations used. Frequency statistics shows that only 42.9% of all participants achieved the full score. Therefore, we have lowered our criterion for 'correct responses' to those whose scores are the mean score or above. The mean score was 7.06 and hence the following analysis is based on data from 24 participants whose total scores were 7 or above.

	N	Min	Max	Mean	Std. Deviation
Arrow	24	4	19	6.67	3.05
Line	24	4	24	9.63	4.72
Juxtaposition	24	5	25	10.08	4.84

Table 1 – Descriptive statistics of average response time.

Of the participants whose scores are between 7 and 9, the means of response time in seconds taken over the three trials are 6.67 for arrow, 9.63 for line and 10.08 for juxtaposition (see Table 1).

Response Time

A 2x3x3 (group x representation x trial) mixed ANOVA was performed for response time. The within-subjects factors were type of representation and the trial and the between-subjects factor is group. The ANOVA results show the following:

a. Significant main effect of type of representation, F (1.479, 32.55) = 8.175, $p < 0.003$.

b. Significant main effect of trial, F (1.348,29.65) = 5.516, $p < 0.018$.

c. No significant interaction between trial and representation, F (1.972, 43.38) = 1.378, ns.

d. No significant between-subjects effect, F (1,22) = 0.418, ns.

Pairwise comparison of average response time taken over the three trials for different types of representation was then made. The Paired-Sample T test results revealed a significant difference between 'arrow and line' (t = -3.008, df = 23, p < 0.006) and 'arrow and juxtaposition' (t = -5.796, df = 23, p < 0.0005) but that the difference between 'line and juxtaposition' was insignificant (t = -0.426, df = 23, ns).

Accuracy

A one-factor ANOVA analysis for sum of the scores over the three trials on each representation as the dependent variable and the type of representation as within-subjects factor revealed no significant main effect of representation (F(2,46) = 1.353; ns). Mean scores are 2.96 for the arrow and 2.79 both for the line and the juxtaposition.

Cochran's Q-test for dichotomous nominal data analysis for score obtained in each trial was carried out for each representation separately. Statistics shows the accuracy was not significantly affected by trial number in any of the three representations. For arrow, Cochran Q = 2.0; df = 2; ns. For line, Cochran Q = 3.500; df = 2; ns. For juxtaposition, Cochran Q = 5.200; df = 2; ns.

Discussion

The results support the hypothesis that the type of representation affects the speed of performance. The arrow outperforms both the line and the juxtaposition. The mean response times obtained in seconds were 6.67 for arrow, 9.63 for line and 10.08 for juxtaposition. The t-test revealed that average response time for arrow was significantly faster than the other two representations. However, we also found that the response time was significantly affected by the trial number.

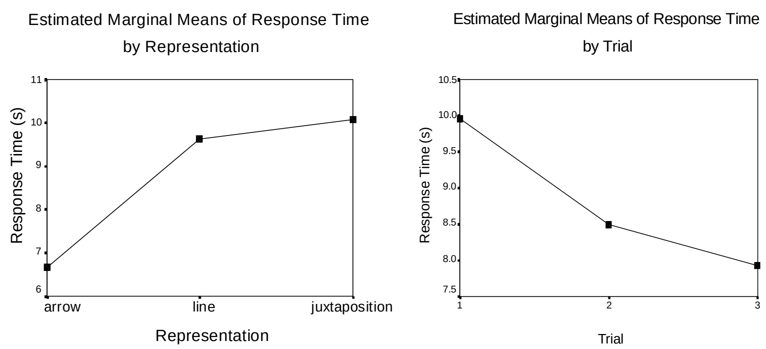

Figure3 – Variation of response time with type of representation

In terms of accuracy since there was neither main effect of representation nor trial it is statistically inconclusive whether or not one representation is better or worse than another. However, the mean scores for the arrow is the highest suggesting that arrow may be better than line and juxtaposition. The arrow has the best performance for the following reasons:

a. Response time. There is statistical evidence that the arrow has the best (the fastest) response time.

b. Accuracy. There is *no* statistical evidence that arrow performed poorer than line or juxtaposition. Arrow yielded the highest score among the three representations.

However, we also found the effect of trial on response time. Our hypothesis is that this is due to the practice effect, i.e., participants were asked to do three trials for the same directional representation before proceeding to the next type of representation. Figure 3 shows how response time varies with representation and with trial. In the next set of experiments we wanted to find out whether in a completely randomised set of trials such an effect would still uphold.

5 Second Set of Experiments

The second set of experiments differed from the first set only by order of trials, i.e. three trials for each directional representation were not grouped any more. The type of representation was completely randomised over all nine trials.

The experiment was conducted with first year Computer Science undergraduate students, twenty-two at the University of Westminster (different group of students from those that participated in the first experiment) and twenty-six at the Brunel University. The Westminster group was using Visual Basic in their programming course whereas the Brunel group was using JAVA. The procedure was the same as in the first experiment but, at the end of the experiment, participants were asked to fill in a questionnaire. The questionnaire asked them which one of the three representations they thought was the easiest and which the hardest. The outcome of the questionnaire based on 48 replies is shown in table 2.

	Arrow	Line	Juxtaposition
Easiest	60.4	20.8	18.8
Hardest	10.4	47.9	37.5

Table 2 – The questionnaire summary – in percentages

A t-test of the two groups on total time taken by each participant revealed no significant difference between the two groups, $t(46) = -0.213$, ns.

A t-test of the two groups on total score revealed no significant difference between the two groups, $t(32.498) = 1.905$, ns. The means of total score are 6.55 for the Westminster group and 7.85 for the Brunel group. Total scores varied from 2 to 9 and 43.8% of all participants received the maximum score (9). The mean score was 7.25 and hence the criterion for ‘correct responses’ is 7, the same as in the first experiment. The following analysis uses data obtained from 36 participants who scored 7 or above.

Of the participants whose scores are between 7 and 9, the mean of response time in seconds taken over the three trials is 8.39 for arrow, 11.61 for line and 13.09 for juxtaposition (see table 3).

	N	Min	Max	Mean	Std. Deviation
Arrow	36	5	19	8.39	2.75
Line	36	5	27	11.61	6.20
Juxtaposition	36	7	37	13.09	5.94

Table 3 – Descriptive statistics of average response time in seconds.

The same statistical analysis was performed for the second set of experiments as for the first set. Therefore, we do not discuss in detail how it was conducted. Instead, we compare results of these two experiments. A 2x3x3 (group x representation x trial) mixed ANOVA was performed for response time. The comparison between results in the first and the second set of experiments are given in table 4.

ANOVA results for the response time	Experiment	
	First	Second
Main effect of type of representation	F (1.479, 32.55) = 8.175, $p < 0.003$	F (2, 68) = 15.97, $p < 0.0005$
Main effect of trial	F (1.348,29.65) = 5.516, $p < 0.018$	F (2,68) = 2.151, ns
Interaction between trial and representation	F (1.972, 43.38) = 1.378, ns	F (2.345,79.733) = 1.838, ns
Between-subjects effect	F (1,22) = 0.418, ns	F (1,34) = 0.069, ns

Table 4 – 2x3x3 (group x representation x trial) mixed ANOVA results for response time

Pairwise comparison of average response time taken for different types of representation was then made. The Paired-Sample T test results for both sets of experiments are given in table 5.

A one-factor ANOVA analysis for the sum of the score over three trials on each representation as dependent variable, and type of representation as within-subjects factor revealed significant main effect of representation (F(1.429, 50.019) = 10.077, $p < 0.001$) on accuracy. Mean scores are 2.94 for the arrow and the line, and 2.58 for the juxtaposition.

	Experiment	
	First	Second
Arrow and Line	t = -3.008, df = 23, p < 0.006	t = -3.705, df = 35, p < 0.001
Arrow and Juxtaposition	t = -5.796, df = 23, p < 0.0005	t = -5.790, df = 35, p < 0.0005
Line and Juxtaposition	t = -0.426, df = 23, ns	t = -2.118, df = 35, ns

Table 5 – Paired-Sample T Test for average response time

Pairwise comparison of scores obtained over the three trials for each type of representation was then made. The Paired-Sample T test results revealed that there was significant difference both between ‘arrow and juxtaposition’ and between ‘line and juxtaposition’, (t = 3.389, df = 35, $p < 0.002$).

Cochran’s Q-test for dichotomous nominal data analysis for score obtained in each trial was carried out for each representation separately. Statistics shows no main effect of trial in any of the three representations. For arrow and line, Cochran Q = 1.0; df = 2; ns. For juxtaposition, Cochran Q = 0.462; df = 2; ns.

	Experiment	
	First	Second
Arrow	6.67	8.39
Line	9.63	11.61
Juxtaposition	10.08	13.09

Table 6 – Mean response time in seconds

The results confirm the conclusion drawn from the first experiment, i.e. that the type of representation affects the speed of following direction. The t-test revealed that average response time for arrow was significantly faster than the other two representations. The mean response times for all representations are higher in this experiment than in the first experiment as a result of the absence of practice effect (see table 6).

The t-test revealed that scores obtained for the arrow and the line were significantly higher than for the juxtaposition. The mean score obtained by the juxtaposition in both experiments was lower than that obtained by the arrow (see table 7).

Figure 4 shows how the three representations compare in terms of response time and accuracy.

	Experiment	
	First	Second
Arrow	2.96	2.94
Line	2.79	2.94
Juxtaposition	2.79	2.58

Table7 – Mean score over three trials

The second set of experiments confirmed that the arrow seems to produce the best performance and that the juxtaposition is the most error-prone. There was no main effect of the trial in terms of response time in the second experiment.

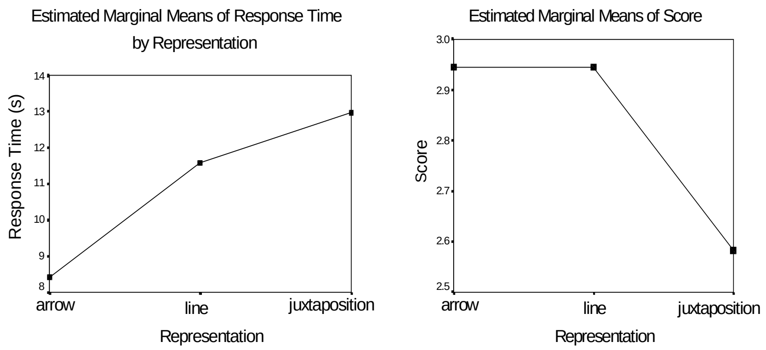

Figure 4 – Response time and accuracy performance vs representation.

6 Discussion

Our experiments suggest that the choice of directional representation may affect the speed of performance. The means of response time taken for each representation in the first experiment is consistently lower than those in the second experiment. Our interpretation is that this is due to the practice effect experienced in the first experiment. The type of representation did not have a significant affect on the accuracy in the first experiment. However, the second experiment revealed the main effect of the type of representation in terms of accuracy, i.e. that juxtaposition was the most error-prone.

There was no significant difference in performance between groups of participants in both sets of experiments regardless that the groups were given different incentives. In the first set of experiments, the Westminster group received no incentive while the Goldsmiths group received money. In the second experiment the Westminster group received a gift incentive while the Brunel group received both gift incentive and the experiment was competitive. Each student in the Brunel group was informed of the total score at the end of the experiment.

In the first experiment 63% of participants from the Westminster group met the 'correct responses' criterion compared to 67% of the Goldsmiths group. In the second experiment 64% of participants in the Westminster group met the criterion compared to 85% of participants in the Brunel group. These figures suggest that our participants were most co-operative when being challenged. This suggests that the program, which will aid students in learning programming, should include some means of intellectual challenge and award. This principle is not new to computer game designers. Computer games players are constantly competing for a better score, either somebody else's or theirs.

7 Conclusion and Future Work

An experimental study that compares how the three most common directional representations affect users' performance was conducted on several groups of undergraduate students. Experiments assessed the performance of students when using arrow, line, and juxtaposition as a directional representation by measuring the speed and the accuracy when they followed a path in a maze. The results of our experiments give support to our initial hypothesis that the type of directional representation affects the speed and the accuracy of following the route.

The arrow appeared to be the most intuitive directional representation having performed best both in terms of speed and in terms of accuracy. Furthermore, participants indicated in questionnaires an overwhelming preference for the arrow over the line and the juxtaposition.

The juxtaposition was the most error-prone representation with the highest mean response time and it therefore appears to be the least intuitive. We will therefore consider the arrow as a priori visual representation to indicate direction. However, the decision as whether to use arrows or not may be governed by other considerations such as, how it combines with various diagramming techniques. Our next project is to investigate how program specifications employing different styles of diagramming techniques using arrow, line, juxtaposition and plain text compare regarding comprehension and debugging.

Acknowledgement

This research is supported by the EPSRC grant number GR/M08387.

References

Bonar, J. and Liffick, B. W. (1990), A visual programming language for novices. In S.K. Chang (Ed.), *Principles of Visual Programming Systems*, Prentice-Hall (NJ).

Brooke, J.B. and Duncan, K.D. (1980a), An experimental study of flowcharts as an aid to identification of procedural faults. *Ergonomics*, 23(4), 387-399.

Brooke, J.B. and Duncan, K.D. (1980b), Experimental studies of flowchart use at different stages of program debugging. *Ergonomics*, 23(11), 1057-1091.

Cunniff, N., Taylor, R. P., and Black, J. B. (1986), Does programming language affect the type of conceptual bugs in beginner's program?: A comparison of FPL and Pascal. In M. Mantei and P. Orbeton (Eds.), *Human Factor in Computing Systems*, CHI '86 Conference Proceedings, 175-182.

Eisenstadt, M. and Breuker, J. (1992), Naïve iteration: An account of the conceptualizations underlying buggy looping programs. In M. Eidenstadt, M.T. Keane and T. Rajan (Eds.), *Novice Programming Environments: Explorations in Human-Computer Interaction and Artificial Intelligence*, Lawrence Erlbaum Associates (UK), 173-188.

Gilmore, D.J. and Smith, H.Y. (1984), An investigation of the utility of flowcharts during computer program debugging. International *Journal of Man Machine Studies*, 20, 357-372.

Green, T.R.G., Petre, M., and Bellamy, R.K.E. (1991), Comprehensibility of visual and textual programs: A test of superlativism against the ‘match-mismatch’ conjecture. In J. Koenemann-Belliveau, T.G. Moher, and S.P. Robertson (Eds.), *Empirical Studies of Programmers: Fourth Workshop*, Ablex Publishing Corp.

Myers, B.A. (1990), Taxonomies of visual programming and program visualization, *Journal of Visual Languages and Computing*, 1(1), 97-123.

Moher, T.G., Mak, D.C., Blumenthal, B., and Leventhal, L.M. (1993), Comparing the comprehensibility of textual and graphical programs: The case of Petri Nets. In C.R.Cook, J.C. Scholtz, and J.C. Spohrer (Eds.), *Empirical Studies of Programmers: Fifth Workshop*, Ablex Publishing Corp.

Nassi, I. and Shneiderman, B. (1973), Flowchart techniques for structured Programming. *Sigplan Notices*, ACM, 8 (8), 12-26.

Rist, R.S. (1986), Plans in programming: definition, demonstration, and development. In E. Soloway and S. Iyenger(Eds.), *Empirical Studies of Programmers*, Ablex.

Shneiderman, B., Mayer R., McKay D. and Heller, P. (1977), Experimental investigations of the utility of detailed flowcharts in programming. *Communications of the ACM*, 20, 6(June 1977), 373-381.

Soloway, E., Bonar, J. and Ehrlich, K. (1983), Cognitive strategies and looping constructs: An empirical study. *Communications of the ACM*, 26(11), 853-860.

Soloway, E. and Ehrlich, K. (1984), Empirical studies of programming knowledge. *IEEE Transactions on Software Engineering*, 10(5), 595-609.

Spohrer, J. C., Soloway, E. and Pope, E. (1985), A goal/plan analysis of buggy Pascal programs. Human-Computer Interaction, 1, 163-207.