Robot Dance: Edutainment or Engaging Learning

Chris Martin
School of computing
university of Dundee
cjmartin@computing.dundee.ac.uk

Janet Hughes
School of computing
university of Dundee
jhughes@computing.dundee.ac.uk

Keywords: POP-I B. learning in projects,POP-I C educational technology, POP-II A. novices

Abstract

This paper describes an Arduino-based robot workshop (Arduino, 2011) funded by the scottish government that uses robots and open problems to engage a wide range of participants from primary 5 school pupils to undergraduate students. This workshop was developed as an outreach activity which has strong links to active research in the area of teaching and learning of programming. This paper shall describe briefly the structure, aims and objective of the Robot Dance workshop. This is followed by the research questions it seeks to address and a description of the methodology utilised to explore these questions. Result are presented and discussed before purposing future work.

1. Introduction

"How do you teach a Robot to Dance" is a workshop that was developed as an outreach activity to engage and enthuse possible future computing students. The workshop presents the opportunity to measure the change in programming expertise as a result of a 1-2 hour intervention. This is not a novel activity: work with robots as outreach in computer science education is commonplace. In previous school of computing outreach events using lego mindstorms with local school pupils, robots have been very well received with enthusiasm from participants and teachers. This study goes one step further than previous work and assess change in participant expertise as a result of the workshop to discern whether playing with robots is fun or a means to engage students in challenging area of study.

2. Background

Computer programming is a challenging subject to teach and learn. One reason for this is that competence in computer programming is indicative of competence in a diverse array of interrelated skills such as design, problem solving and abstract thinking. In short, many skills and pieces of knowledge need to be in place for simple computer programs to be constructed. There is great interest in exploring how best to support teaching and learning of computer programming. A substantial literature exploring this subject exists. For example Robins et al (2003) offer a good review of the field. This literature has identified and discussed areas of difficulty (Du Boulay, 1989) and purposed a number of tools and techniques to elevate problems common in introductory programming courses. Powers et al (2006) categorised introductory programming support tools as narrative tools, visual programming tools, flow-modelling tools, specialised realisation tools and tiered tools with the implication that the tools tend to have features from a number of categories.

The tools analysed and discussed to form this categorisation are generally composed of one or more categories, as is evident by multiple appearances in the example tool column (table 1). Each category has qualities that can be considered as a response to the difficulties of programming.

Visual programming seeks to alleviate syntax issues by providing ‘drag and drop’ program creation rather than typed text. Each operation leaves the program in an executable state, giving immediate feedback. This may reduce the likelihood of a novice becoming a “stopper”(Perkins, Hancock, Hobbs, Martin, & Simmons, 1989) as a result of persistent syntax errors.

Flow modelling seeks to provide a medium for visual externalisation of mental models of the notional machine (Du Boulay, 1989). This provides a detailed representation of the program's execution path and reinfoces the sequential nature of the program. This supports understanding of the cumulative effect of the program constructs viewed step, by step giving access to intermediary states along the temporal dimension.

CATEGORY	DESCRIPTION	EXAMPLE TOOL
Visual programming	The visual programming paradigm is used in a number of introductory programming tools to abstract above syntax and allow higher level programming concepts to be explored. Visual programming consists of dragging and dropping program components to assemble a program. It is a general premise of visual programming that the program is always left in an executable state (Powers et al, 2006) to facility "tinkerability" (Resnick, 2007) or to reduced premature commitment (Green & Petre, 1996).	JPie ALICE JHAVE RoboLab SCRATCH
Flow model	Flow modelling tools provide visual representation of program flow, often representing the path of execution as a flow chart. Flow modelling tools allow for a external representation of the executing program which can serve as an intermediary between mental models of solution and problem domain.	RAPTOR ALVIS RoboLab Greenfoot SCRATCH
Tiered	Tiered tools offer a gradated approach to programming, offering the opportunity to perform meaningful tasks with a variable sub set of the full language. This approach can support teaching and learning by allowing the gradual introduction of syntax and more complex programming strategies.	DrScheme RoboLab
Narrative	Narrative tools use the creation of interactive games or non interactive movies as a vehicle for programming a story. Cooper states: "The ability to direct your own movie is extraordinarily attractive to a wide range of students" (Powers et al, 2006).	ALICE JEROO
Specialised output	Specialised outputs include programs that are executed on unconventional computing hardware, for example programming an autonomous robot, which can "embody state and behaviour, physically modelling the programming solution" (Powers et al, 2006).	LOGO turtle LEGO mindstorms scribbler handyboard

Table 1: Tools to support introductory programming (Powers et al, 2006)

Tiered tools offer a structured and gradual increase of syntax and program complexity. This has the advantage that the one language or tool can provide a tier of sophistication appropriate to the learner and stage of learning, from introduction to advanced topics. This may reduce confusion when switching from an introductory tool to a more professional tool.

Narrative tools seek to present programming as a creative endeavour and build on existing concepts that are familiar and engaging, as evidenced by Cooper observing enthusiasm from minority student groups with a strong oral tradition (Powers et al, 2006). Good and Robertson (2006) discuss the increased motivation observed in narrative game creation through "Creation of a valued artifact", an idea rooted in constructionism as explored by Papert and Harel (Seymour Papert, 1980; S Papert & Harel, 1991).

Special Realisation tools place programming in a novel context which can improve motivation and lead to an increased time on task (K. Powers, et al., 2006).

Visual programming, flow modelling and tiered tools all seek to alleviate difficulties in programming with relation to syntax and flow of execution by abstracting certain elements to focus on the key learning aims and objectives. Narrative and Special Realisation tools attempt to increase the sense of purpose and value the learner associates with programming tasks, which can improve motivation.

It is necessary to illustrate these categories in the context of the technical tools in which they are employed. Treating this as a purely technical characterisation is however overly simplistic: the tools can be used to support teaching and learning but other contextual material will have a substantial influence on the teaching and learning. Prevalence of supporting materials, integration into curriculum, availability, accessibility, online resources and support will have substantial effect on the teaching and learning benefit. This characterisation may embody a philosophical standpoint which can be adhered to in a number of technologies.

Robots are examples of specialised realisation tools. This is a tool that realises the executing program in novel fashion. A robot executing the instructions forward 10, spin right 2, forward 5 offers a concrete representation of the program in execution. Since their introduction in Mindstorms (Papert, 1980) as a constructionist tool for engaging maths education. Robots have been used in computer science education in degrees of sophistication ranging form outreach (Osborne, Thomas, & Forbes, 2010) to fundamental and advanced artificial intelligence (Kumar & Meeden, 1998).

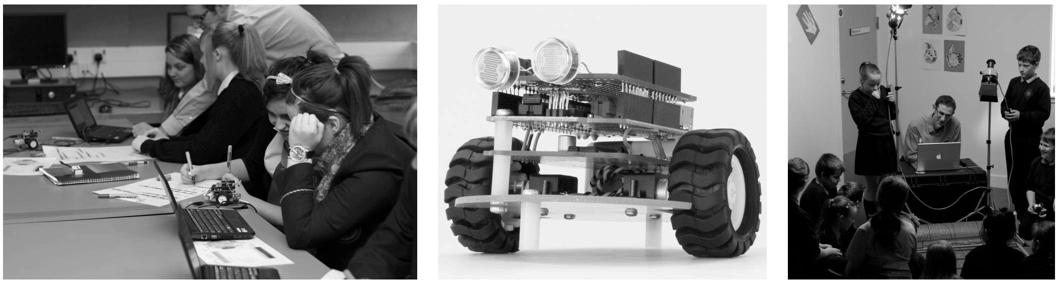


figure 1. (left) developing dances, (centre) Arduino based robot, (right) dance performances

3. The Robot Dance workshop

The Robot Dance workshop has been developed as a outreach activity that can engage an audience from middle primary (P5) through secondary school and beyond (figure 1). It has been developed with flexibility at its centre though generally runs as a taught classroom session and a drop-in format has been used at science fairs.

One of the main aims of the funder was that this project must have genuine links to active research. This was achieved by seconding a PhD student that was in a position to develop a workshop around their research. In the case of Robot Dance, the area of inquiry is in teaching and learning of computer programming to the novice.

To maximise flexibility, the workshop was developed with a central theme of making a robot dance. This was inspired by Robo Cup Junior (RCJ, 2011), an international event open to school pupils around the world. Robo Cup Junior offers three challenges: robot rescue, robot football and robot dance. Each event places programming and engineering challenges in a different context that will motivate different participants. Robo Cup Junior teams generally develop their robots using lego Mindstorms at after school clubs for some time in the run up to the event. Petre and Blaine (2004) observed evidence of 'back door learning' in students taking part in Robo Cup Junior. Students were observed pooling programming knowledge and concepts as circumstance arose e.g. repetition of code block equaling desire for subroutines or functions.

The Dance element of this competition offers an open problem which is invaluable in developing content that can be presented to a wide audience. Throughout the workshop, primary five students have been observed just as engaged in the task as university lecturers. Prior knowledge is largely irrelevant as the task is creative, subjective and without endpoint. Often there is little difference in the output from an expert or novice, with definite scope to over-think the problem. This has a pitch-levelling effect, a number of teachers commenting that students previously disinterested or failing to show aptitude for programming had engaged in the workshop successfully. Dance was also chosen as a context that may make computer programming more appealing to female participants.This is a strategy seen in *narrative tools* as powerful, the tasking being placed in a context that has cultural relevance or 'coolness'.

Turning the workshop around in one hour means there is no ‘build time’. For this reason, prefabricated robots are used. Past experience with Mindstorm kits have been polarising with some students revelling in the mechanical challenge of constructing their robot and others being disinterested. To allow for a great deal of flexibility in the future tasks, Arduino-based differential drive robots have been constructed. These are compact and robust: the Arduino board has emerged in response to the needs of artists and installation designers. As a result, programming an Arduino in C from the Arduino IDE presents a very low barrier to textual programming of a microcontroller. A *tiered* approach to introductory programming has been taken with much of the low level programming abstracted away to the robot library.

Reproducing textual syntax precisely is a common area of difficulty in introductory programming (Du Boulay, 1989). *Visual programming tools* seek to alleviate syntax errors by removing the opportunity to produce typed errors, rather than highlighting that an error has occurred via a compiler error. In the case of robot dance this is balanced against a desire to expose students to a language and syntax they are likely see if they pursue computing. To tackle this, students are given interactive documentation that allows them to copy and paste functions as they construct their programs.This also has the advantage that it encourages an expert skill of browsing documentation to discover new things. e.g. I know that a function can control the motors, let's see if there is one to control the lights.

The key learning *aim* for the workshop is to offer participants a pleasant programming experience in a real world language (with syntax they are likely to see in future programming courses). Secondary to this is the objective of portraying the task of programming, which is often regarded as rigid and scientific, as a creative endeavour where imagination and creativity are as important as, if not more so, than knowledge. Specifically the workshop aims to address the following three learning *objectives*:

1) Participant should learn about flow of execution by producing a list of dance instructions and observing resulting robot movements.

2) Participants should learn about syntax in programming and the degree of precision required when programming as they produce their programs.

3) Participants in one of the more advanced sessions should gain an understanding of variables and their role in programming. This more advanced session also introduces decision and iteration through development of Braitenberg-style robots to solve mazes and follow lights (Braitenberg, 1986).

4. Research Question

The robot dance workshop is a vehicle to explore the impact of robot programming on aspiring novice programmers. The primary research question is whether learning pertinent to syntax, sequence and variables is taking place as a result of the robot dance workshop. Secondary questions include looking for gender differences and effect of different interventions.

5. Methodologies

The study design had to fit within the overreaching theme of the project: to provide an engaging outreach activity for a large diverse audience of school pupils. This produces some constraints, namely the project must engage with a large audience of at least 2000 people. To reach these numbers, a combination of science fairs and school visits was necessary. The science fair format is that of a 'drop in' session which proved too uncontrolled to gain data from. The classroom setting provided a greater degree of control, with students having an equal amount of time on task and receiving the same introduction and support.

When working in a school setting it is necessary to fit within the school day. Throughout the workshop teachers were likely to offer only a single period to an outreach activity (in some case a double period). To accommodate this, the workshop was developed to be deliverable in under an hour, with the flexibility to make use of any extra time. The key impact this had on study design was that any research activities had to take minimal time out of the activity.

A pre/post paper based test was developed to gain a scalable quick assessment of the students' change in expertise as a result of the intervention. Impact was categorised by assessing the effect of the robot dance intervention on programming expertise. The participants were asked; to answer eight questions as true, false or don't know. The questions relate to the key learning objectives (sequence, syntax and variables). When scores were captured the participant gained a point for a correct answer, lost a point for an incorrect answer and gained or lost nothing for selecting don't know.

A pre/post test design has a number of limitations. The only indication of the participants' prior experience is limited to their pre test score. This is arguable a shallow representation of programming experience or competence. Various options for gaining a richer picture of prior experience were considered, including listing programming languages the participant is familiar with or a brief interview to determine past experience. It was unfortunately not feasible within the constraints of the project to implement these options. For this study, therefore, the pre-test score forms the baseline of expertise to measure change from.

An important implication of this study design is an awareness of the potential for change. If a participant has a high degree of prior competence and performs well in the pre-test there is little opportunity for a large improvement. For this reason mean change in performance may be skewed by participants that do not have anything new to learn form the workshop.

The primary analysis of test score data involves detecting significant change between pre and post test score for each participant. A positive change indicates an increase in performance in one of the learning objectives and respectively a negative change a decrease in performance. Descriptive statistics are used to describe the distribution's limits, mean and standard deviation. To highlight the distribution of change in performance, the percentage of participants displaying a positive, negative and no change in performance shall also be noted. Significance of difference between pre and post-test distributions is tested with a two tailed paired t-test.

Test score data is annotated with gender, group and participant identity. This allows the scores to be analysed with respect to the whole sample's pre and post score, a comparison of male versus female performance a performance of robot dance versus follow the light groups.

As with any field work, there is an element of subjective observation that takes place. It is important to ensure this is reported as just that and presented as grounds for further inquiry rather than evidence for an assertion. The tests were given at the very beginning and end of the sessions.

6. interventions

The workshop consists of two possible activities, Robot Dance and Follow the Light. Both sessions open with a one slide introduction to robots using a popular age appropriate film character to ground a conversation. In primary schools, Wall-e serves as a hook to discuss the similarities between Wall-e, us and the Arduino robots. This usually highlights the fact we can all move, see or sense and make decisions or think. The moving and sensing is categorised as engineering and, as the robots are already constructed, the challenge is a computing one, involving giving the robots a sequence of instructions to make sense of its environment.

6.1.Robot Dance

With the activity introduced, the next task is to describe the steps required to program the robot. This takes three slides. An introduction to the capabilities of the robot is followed by the instructions or language the robot understands. The final component is the IDE or 'fancy word processor' used to group together the instructions and send them to the robot. In total this takes under ten minutes and is all that precedes the first challenge: to make the robot move across the desk. The quick time to task is key to retaining the attention of the students. More than this, the heart of this approach to teaching and learning is that practical activity creates a fruitful learning environment.

This task introduces a key concept that instructions used to initiate motion operate like light switches. forward() changes the state of the robot from whatever it was to both wheels moving forward. Timing is also highlighted as a means to control motion, in this case distance travelled. If the program consists of forward(); immediately succeeded by stopMotion(); the robot will do nothing, in the same way that if you switch a light switch on and off really quickly, you can barely notice the light go on. What is required is a pause before the next change of state and in turn this pause affects how long an action or dance move lasts. Thus to move 50 centimetres, the program must change the state of the motor to forward, wait for however many seconds and then change the state of the motors to stationary. This enables students to explore the examples described and removes the likelihood of the student's knowledge being inert (Perkins & Martin, 1985).

The second challenge builds on challenge one to add an additional action. The robot must move across the desk, spin 180 degrees and return to the original position. This reinforces the concepts from the initial task and extends distance controlled with time, to degrees of rotation controlled by time between the spin command and the next state change. The term forward is also given a new relative perspective as the same instructions to move the robot away from the start point will move it back to its original position. The 'forward' instruction is affected by the orientation of the robot. This begins to give a concrete example of the fact that "...each instruction operates in the environment created by the previous instruction." (Du Boulay, 1989). The effect of a single instruction on the position of the robot is the cumulative effect of all previous instruction in the program.

Students are then given the final challenge: to choreograph a robot dance. There are two constraints: the dance must last for exactly 20 seconds and the robot must not fall off the stage (a 1 square meter mat or equivalent table top for added suspense). Timing is straightforward and can be achieved by keeping note of the wait times used in the program. Not falling off the stage requires a more heuristic approach, of not moving for long time periods, as the robots do not offer enough precision for a more sophisticated approach. To this point the robots have been started via a start button; this also has to be modified to allow the robot to start the program when the stage lights come up. An instruction for this is contained in the interactive documentation.

After the students have been given a relatively short development time of 10 - 15 minutes, they upload their final program and gather round the dance floor to watch each of the performances. Blank and Kumar (2110) describe that performance has wider reaching motivational effect than competition in this area. As a result, we also favour performance. In larger settings this has been softened to a series of dance-offs, with up to ten teams taking part. With two robots performing side by side on stages at the same time.

6.2.Follow the Light: Braitenberg vehicle

Valentino Braitenberg is an Italian cyberneticist who introduced through his book, "Vehicles: Experiments in Synthetic Psychology", that incredibly simple robots with different linkages between sensor and actuators can exhibit behaviours that may be perceived as human-like. The central premise of this style of robot control involves iterating infinitely the actions of sense environment and react as action. This allows what are incredibly small simple programs or decisions, at a local level, to exhibit complex behaviours at a more global scale. Papert (1980) describes an example of this in illustrating how to draw a circle with logo. Mathematicians will immediately reach for complex equations which involve cartesian coordinate systems and a global perspective; in logo however, drawing a circle can be achieved by effectively: move forward a little, turn a little, repeat. This principle of small simple computation producing globally complex behaviour is an interesting hook for students being introduced to programming. Due to the increased complexity of this workshop, it was delivered to senior school pupils exclusively.

'Follow the Light' uses the same introduction and first two challenges as 'Robot Dance'. From basic sequence and state, the concept of a variable is introduced via a blinking light. In Arduino circles, 'blink' is the equivalent of hello world. By turning an LED on, waiting for a period of time, turning it off, waiting for a period of time and repeating, an LED will flash. To extend this we can use a variable to store the duration of the delay, introducing the potential advantages in code readability and the ability to edit the code in one place. This variable is then used to store the value of a potentiometer (similar to a radio volume control). This gives the ability to change the rate of flashing when the code is running. Essential a potentiometer is a tangible object that can give the program a range of numbers (0-1023). This offers a concrete tangible representation of a possible role of a program variable.

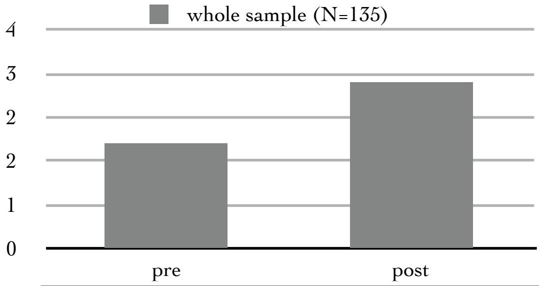
The next skill introduced is decision: this is introduced via an (almost) natural language statement about instructions to maintain a constant temperature in a room: "if the room is too hot open a window or else close the window". Once the basic structure is highlighted, we have some sensed information "room is too hot" and if this is true we shall take one course of action; if it is false we shall take a different course of action. At this stage the syntax of the 'if' block can be mapped out. To get to the point of being able to implement this, it is necessary to unpack the notion of 'too hot'; this is a boolean operation involving reading the room temperature, storing this in a variable t and making a comparison: is t greater than threshold. This forms the basis of the third challenge: when a bright light is shone on the robot, move forward or else stop. In other words, in a one-dimensional land, a light following robot. This again gives the students an opportunity to enact the concept discussed. The final challenge involves extending this program to a two-dimensional land. This involves establishing connection between the two light sensors and independently controlling the motors to achieve the desired behaviour.

7. Results

The robot dance workshop has had measures taken from 12 sessions, with participants from 4 different populations, ranging from local secondary school (2 sessions), visiting youth group (1 session), pre-application visit days (5 sessions) and sessions at a large science festival (4 sessions). The total population comprises 135 paired pre and post tests with pupils of an age range of 12-16. This population comprised 69 female participants and 66 male participants. 3 of the groups received the light-following intervention with the remaining 9 receiving the Robot Dance intervention. The following section looks at the performance of the population as a whole in relation to test scores for sequence and syntax knowledge. Following this, scores with respect to gender are presented. Finally the differences between the Light Following group vs the Robot Dance Group are presented.

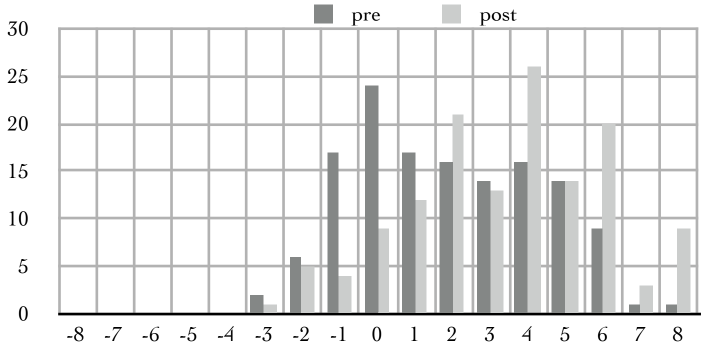
7.1.Whole sample: Sequence and Syntax

The whole population showed a mean pre-test score of 1.9 and a mean post-test score of 3.0 with a significant (p<0.01) difference of 1.1 (graph 1). The minimum difference was -5 and the maximum difference was 10. The whole sample's change in performance was distributed as follows: 24% showed a decrease in performance, 20% had no change and the remaining 57% showed an improvement in performance. Graph two show a frequency distribution of the pre and post test scores which indicates a decrease in frequency of low scores and an increase in frequency of higher scores.



	pre	post	diff
wp	1.9 [SD 1.89]	3.0 [SD 2.05]	1.1 p< 0.01

graph 1: sequence(3) and syntax(3)

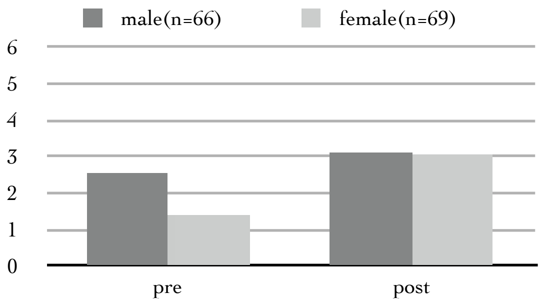


graph 2:pre vs post score frequency distribution

7.2.Male Female: Sequence and syntax

The gender split was 66 male to 69 female, offering comparative group sizes. The male and female groups were distributed evenly throughout the 12 sessions ran. The male group had a mean pre-test score of 2.5 and a mean post-test score of 3.1 with a difference of 0.6 . The minimum difference was -5 and the maximum difference was 5. The male group's change in performance was distributed as follows: 27% showed a decrease in performance, 27% had no change and 46% showed an improvement in performance.

The female group showed a mean pre-test score of 1.4 and a mean post-test score of 3.06 with a difference of 1.8. The minimum difference was -3 and the maximum difference was 7. The female group's change in performance was distributed as follows: 19% had a decrease in performance, 10% had no change and 71% showed and improvement in performance.



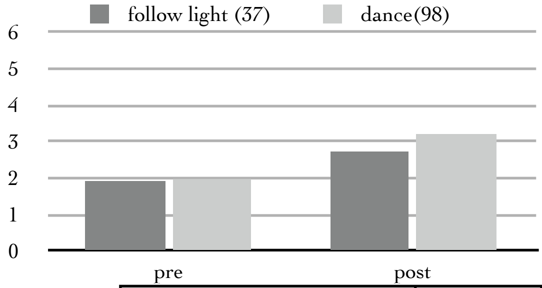
	pre	**post**	**diff**
m	2.5 [SD 1.68]	3.1 [SD 2.07]	0.6
f	1.4 [SD 1.91]	3.1 [SD 2.01]	1.7
			1.11 p< 0.01

graph 3: sequence(3) and syntax(3)

7.3. Follow the light vs Dance: syntax and sequence

The dance group comprised 98 participants and the follow the light group 37 participants. The dance group showed a mean pre-test score of 2.0 and a mean post-test score of 3.2 with a difference of 1.2. The minimum difference was -3 and the maximum difference was 7. The dance group's change in performance was distributed as follows: 18% showed a decrease in performance, 20% had no change and 62% showed an improvement in performance.

The follow the light group showed a mean pre-test score of 1.9 and a mean post-test score of 2.7 with a difference of 0.8. The minimum difference was -5 and the maximum



	pre	**post**	**diff**
follow light	1.9 [SD 1.80]	2.7 [SD 2.24]	0.8
dance	2.0 [SD 1.93]	3.2 [SD 1.94]	1.2
			0.4 p= 0.39

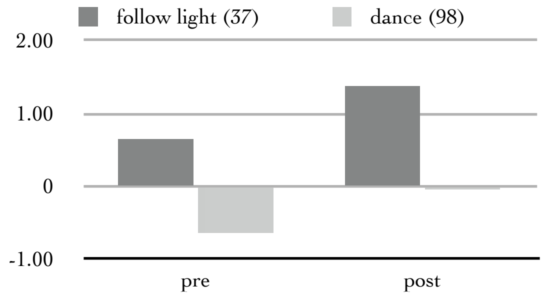
graph 4: sequence(3) and syntax(3)

difference was 7. The follow the light group's change in performance was distributed as follows: 35% showed a decrease in performance, 14% remained the same and 51% showed an improvement in performance. The difference between the dance and follow the light groups performance was 0.43 (p=0.39).

7.4.Follow the light vs Dance: variables

The dance group showed a mean pre-test score of -0.6 and a mean post-test score of 0.0 with a change in performance of 0.6. The minimum difference was -2 and the maximum difference 4. The dance group's difference in performance was distributed as follows: 10% decreased, 46% demonstrated no change and 44% improvement in performance.

The follow the light group showed a mean pre-test score of 0.6 and a mean post-test score 1.4 with a difference of 0.7. The minimum difference was -2 and the maximum difference 4. The follow the light group's change in performance was distributed as follows: 14% showed a decrease in performance, 43% remained the same and 43% improvement in performance.



	pre	post	diff
follow light	0.6 [SD 1.40]	1.4 [SD 0.83]	0.7
dance	-0.6 [SD 0.89]	0.0 [SD 1.16]	0.6
			0.1 p= 0.60

graph 5: variable(2)

8. Discussion of Results

Looking at the mean difference of the whole sample (*graph 1*) these results are more or less what would be expected: there were small improvements as a result of 1-2 hours of tuition. Looking at the detail is a little more concerning: with the session having a negative impact on just less than a quarter of participants. It is probable that for some students this was not an optimal mode of learning. Kinaesthetic Activist learners are likely to react positively to the hands-on nature of the session while others may find it more distracting. It is likely that the research design has impacted on the results. In a number of cases, the tight time for delivery has resulted in the post tests being completed in a rush. To confound this, the energy level of a group of participants is likely to impact this also, as on arrival at the session the students are fresh and after a fairly intensive session they are somewhat fatigued.

Further inquiry is required to fully explain the negative impact. Offering more time for the post test would be a starting point. Performing an assessment of each participant's learning style would also give valuable insight, thought this would likely require research design that is more focused and less scalable.

The girls' improvement as a result of the workshop was significantly ($p<0.01$) greater than that of the boys (*graph 3*). However, looking purely at change data may be insufficient. In a pre/post test design such as this, a high pre test score leaves less room to improve and produce a large difference. The male and female groups showed a significantly different pre-test scores ($p<0.01$) and increased to almost the same level of expertise. This indicates that the female portion of the cohort had less programming expertise than the males at the outset. Robotics tasks are often seen to be more 'geeky' and as a motivation for a male cohort. As mentioned in section 2, one of the reasons to choose the dance task was to soften this and produce a context that female students may engage with. The results support that the female participants engaged and performed well in the workshop. It is important to be mindful of the dangers of designing for stereotypes as you risk alienating users. The combination of dance and robotics appeared to engage the whole audience.

With respect to the robot dance vs follow the light (*graph 3*) there appears to be a small statistically insignificant ($p=0.39$) change in performance of syntax and sequence expertise. The dance group improved slightly more than the follow the light group. This may be a result of the dance group receiving a more focused session with less content to cover.

There is a tiny insignificant difference in the robot dance vs follow the light group's improvement in performance with regard to variable expertise (*graph 4*). What is more stark is the initial pre and post test for the groups with the light following group over a point a head of the dance group. This is most likely explained by the groups that were offered the sessions. The robot dance session has been run with the full age (12-16) range however the follow the light session has only been ran with more senior students which may be the reason for this difference. The very similar slight improvement in performance indicates that the choice of intervention has had little to do with this change, though a more focused study may be required.

9. Conclusions

In conclusion through the design and execution the Robot Dance workshop it has been possible to ascertain whether this style of learning opportunity can produce measurable changes in participant expertise. A small improvement in knowledge of syntax and sequence of computer programs has been observed with an interesting gender effect of female participants showing a greater improvement in performance than male participants. There were no significant differences between the groups that participated in Robot Dance and the light following interventions. The results indicate that the workshop has not been uniformly positive, with a number of participants experiencing a negative change in test score. This may be accounted for by the study design and rushed completion of post-tests; further work is required to explore this. The study could look and emotional factors such as motivation and satisfaction. Extending the design to include knowledge of participant's learning style would also be an interesting.

This study represent broad indication that on average a positive change in expertise takes place as a result of a short robot workshop, based on applied open creative problems. The data presented supports the assertion that Robot Dance is more than edutainment and represents a engaging learning experience. The study to follow Robot Dance shall look in detail at pre/post test scores but shall complement this with empirical data, in an effort to support understanding of why this type of activity provided an engaging experience. Looking into emotional factors such as motivation, and satisfaction as well as the interactional opportunities offered by working with physical artefacts.

10. Acknowledgements

This research would not have been possible without the cooperation of many enthusiastic teachers and pupils from around Scotland. This project was supported by a grant from the Scottish Government, match funded by the University of Dundee. The project was overseen by Dr Jon Urch of the Revealing Research offices in the University of Dundee. Consultation with Dundee Science Centre Sensation helped with early shaping of the workshop.

11. References

Aruduino. (2011). Arduino. Retrieved 2010, from http://www.arduino.cc/

Blank, D., & Kumar, D. (2110). Assessing the Impact of using Robots in Education, or: How we learned to Stop Worrying and Love the Chaos. Paper presented at the Educational Robotics and Beyond: Design and Evaluation.

Du Boulay, B. (1989). Some difficulties of learning to program. Journal of Educational Computing Research, 2(1), 57-73.

Good, J., & Robertson, J. (2006). Learning and motivational affordances in narrative-based game authoring. Paper presented at the 4th International Conference for Narrative and Interactive Learning Environments (NILE).

Green, T. R. G., & Petre, M. (1996). Usability Analysis of Visual Programming Environments a 'cognitive dimensions' framework.

Kumar, D., & Meeden, L. (1998). A robot laboratory for teaching artificial intelligence. SIGCSE Bull., 30(1), 341-344.

Osborne, B., Thomas, A. J., & Forbes. (2010). RoboCupJunior Primer: Expanding educational robotics. Paper presented at the Educational Robotics and Beyond: Design and Evaluation.

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas: Basic Books, Inc.

Papert, S., & Harel, I. (1991). Situating constructionism: Ablex Publishing Corporation.

Perkins, D. N., Hancock, D., Hobbs, R., Martin, F., & Simmons, R. (1989). Conditions of Learning in Novice Programmers.

Perkins, D. N., & Martin, F. (1985). Fragile Knowledge and Negelected Strategies: Harvard Graduate School of Education.

Petre, M., & Blaine, P. (2004). Using Robotics to Motivate 'Back Door' Learning. Education and Information Technologies, 9(2), 147-158.

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K. J., Proulx, V., Carlisle, M. (2006). Tools for teaching introductory programming: what works? SIGCSE Bull., 38(1), 560-561.

RCJ. (2011). Robo Cup Junior. 2010, from http://robocup.web.officelive.com/default.aspx

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. Paper presented at the Proceedings of the 6th ACM SIGCHI conference on Creativity cognition.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. Computer Science Education, 13(2), 137-172.