# Educational robots for teaching programming

## Introduction

Robots have been used in order to further education and increase engagement in a range of topics. In this project I will make use of a user friendly programming language and integrate it with a robot in order to increase interest in computer science.

## Aims and Objectives

The aim of this project is to create an encouraging way to learn computer science skills by integrating user friendly programming languages with robots.

**Objectives**

* To understand the Scratch API and the relevant robot API.
* To adapt scratch for use with the Thymio II.
  + To implement custom blocks
* To create software to integrate scratch with the Thymio II.
* To create and run a focus group session.
* To Compile and assess the results of a focus group.

## Background

Programming and computer skills are becoming increasingly important as the influence of the internet and the power of computers grows, programming has even been called the 'second literacy'. Despite this there are still significant barriers to education in this field and few attempts to integrate it in to other computer related topics. This project will use the language Scratch and the Thymio II as they have both been shown to be good at introducing people to the topic of programming and robots as well as maintaining interest and creating enthusiasm.

Scratch is a language developed at MIT which has been used in education with broad success. Scratch allows for the use of most programming concept without requiring the user to be aware of syntax through the use of blocks. As well as being easy to use it is also free to use and has a large community with a wide range of users from 4 year olds to 60 meaning support can be found relating to the most basic of tasks to complex ones. One study found that during a Harvard Summer School for Computer Science course that 76% of students felt that using scratch as an introduction help them when they later moved on to java, students also found it was more rewarding to have visual feedback on what they had programmed than just having a text window (Malan and Leitner, 2007).

Amongst the reasons why teachers don't accept technology in to the class as readily as they do other tool are stress and fear of failure. Trying to teach something with which you have little experience in can be daunting and trying to learn a new topic can cause stress. Introducing people to scratch has both caused people to be more likely to include programming in lessons and to worry about failing less when they consider education in programming. A study of students learning to become preschool teachers were given a introductory lesson in computer programming and found that interest in using technology in the classroom increased from 80% to 92%. As well as this they found that 65% found scratch easy to use and 85% found it simple and understandable (Fesakis and Kiriaki, 2009).

One challenge facing robotics in education is the price of the platforms and how easy they are to use. The Thymio II can be bought for around £100 which is cheaper than alternatives such as the LEGO Mindstorm while still having most of the feature. Besides the LEGO robot there are few available robotics platforms that are simple enough that they can be used for an introduction to the topic while also being in depth enough that they can perform complex programs. The Thymio II is a capable system and with scratch would mean that it can be easily picked up while still being able to perform some complicated programs. Scratch can create object orientate programs and is seen to have one major limitation which is recursion which has been purposely left out so that beginners would not feel threatened (Harvey, B. and Mönig, J. 2010).

Robots have been used before with other aspects of programming to create courses that have proven to create very enthusiastic students. For example, at the University of Lincoln robotics was taught alongside computer vision, this lead to positive results in practical and some students going far beyond the brief of their assignments. (Cielniak, G. and Bellotto, N. et al, 2013), this suggests that enthusiasm can be created with practical assignments using robots.

The Thymio II is a programmable robot with a wide variety of sensors and methods for feedback. It has 2 wheels for movement, a speaker for audio output and several light, some of which are programmable and others which indicate the feedback from the distance sensors. There are 9 distance sensors to prevent it from falling off objects and to detect thing in front or behind it. It also has a 3 axis accelerometer, a microphone and an infrared sensor for remote input. With all these feature the Thymio II is well suited to education as it can be applied to a lot of situations. The Thymio II is the result of testing amongst children with the Thymio I. After running courses with the Thymio 89.2% of parents thought the session was educational and 78.5% thought that it had increased their child's interest in robotics (Riedo, F. and Rétornaz P. et al 2012)

## Methods: development

Waterfall and incremental development will be appropriate for this project. Water fall makes use of a rigid structure in which one step is finished and then you move on to the next and is suitable for individuals to use.. Generally this method doesn't allow for revisiting or revising phases when they are complete, it also requires early identification of requirements and would not handle unexpected development needs well. This would have serious disadvantages for this project. The incremental method is similar to waterfall in that it uses several mini waterfalls. Each part of the project is treated like a mini waterfall so that that part is completed before proceeding to the next step. This approach allows exploitation of knowledge that would be gained during the project as well as allowing the project to be easily monitored for progress. In this project I will use an incremental method as it better fits my needs as the details of creating the software for the project aren't known yet. (cms, 2005)

## Tools

|  |  |  |
| --- | --- | --- |
| Tool | Pros | Cons |
| Microsoft Word | * Very familiar with its functionality * Widely used * Auto correct and spell check |  |
| Notepad | * Very simple * Universal format | * No spell check * Very few features |
| GitHub | * Useful when using multiple computers * Stores a back up so hardware loss isn't an issue | * Unreliable GUI which can cause freezing |
| Visual studio | * Supports many Languages * Lots of online support | * Not compatible with robotics platforms |
| Aseba | * Allows the Thymio II to be programmed using script | * Not compatible with other languages |
| MonoDevelop | * Can be used on multiple operating systems * Can be used to program in C++ and C# | * No robot support * Poor code assistance and auto complete |

For this project I will use Microsoft Word as it is the word processing tool that I am most familiar with as well as having a wide range of features such as spell check which will improve productivity. I will also use Visual studio for development as it can be used for C++, it also provides good error detection and auto complete. As well as visual studio I will also use Aseba to communicate with the robot

## Methods: evaluation

For this project I will use a focus group to use my software in the form of a 'robo-jam'. After the session is over I will then give them a questionnaire that I will collect in and evaluate the results. A focus group will allow me to find out what a variety of people think about the software I will produce, I will be able to find out people's perceptions and attitude towards using the software. I will get my participants to fill in a standardised questionnaire. By evaluating the results I will be able to look at trends in peoples opinion and use these to make conclusions about the project and how successful it was.

## Risk assessment and contingency plans

During this project there are several risks that could cause setbacks and potentially jeopardise the project, to prevent and prepare for these I will list the most likely to occur, what I will do to prevent them and what I will do to recover from them.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Risk | Severity | Contingency plan/ Mitigation |
| Loss of data | Low | Low | To mitigate the effects I will make use of GitHub to create online backups so I should not lose a large amount of data. I will revert to the last commit on GitHub or local backup if that is more recent. |
| Increased work load from other modules | Medium | Medium | I will manage my other modules to ensure that certain days are dedicated to project and plan. |
| Aseba changes the way it works | Low | High | I will look in to the update notes to see if they would affect my project and not update if it would. If an update does occur then I will try to revert to the previous version |
| Ethical approval rejected | Low | Medium | I will resubmit the form after amending it. |
| Focus group doesn’t attend | Low | Medium | I will rearrange the date and find a new group. |
| Robot develops a fault | Low | Medium | I will store the robot in its box when not using it. I will attempt to replace it if it becomes damaged. |
| Focus group results become lost | Low | High | I will make use of GitHub to store my results so that they will be safe online. If this does occur then I will have to run another focus group in order to gather more results. |
| Robot and Scratch are incompatible | Low | High | I will have to re-evaluate the language that is being used to interface with the robot. |

### Contingency

Contingency time is an important part of the project as it allows time to deal with unexpected problems and to review the work done up to that point. I will assign Contingency for the week at the end of each task on the weekend so I will have 2 full days for catching up or review of my work so far as well as any adjustments to my plan.

## Gantt chart

## A:\Users\Adam\Desktop\gantt1.png

## References

Cms, (2005) *Selecting Developing Approach*. [online] Available from: www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf [Accessed 18 October 2014].

Malan, David J. and Leitner, Henry H. (2007) Scratch for Budding Computer Scientists. *ACM SIGCSE Bulletin* 39(1): 223-227.

Fesakis, G, and Kiriaki, S. (2009) Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, 41 (3) 258-262.

Harvey, B. and Mönig, J. (2010). Bringing “No ceiling” to scratch: can one language serve kids and computer scientists. *Proc. Constructionism*.

Cielniak, G., Bellotto, N. and Duckett, T. (2013) Integrating mobile robotics and vision with undergraduate computer science. *Education, IEEE Transactions on* 56(1) 48-53.

Riedo, F., Rétornaz, P., Bergeron, L., Nyffeler, N., and Mondada, F. (2012). A two years informal learning experience using the thymio robot. *Advances in Autonomous Mini Robots* (pp. 37-48). Springer Berlin Heidelberg.