ISSN 2053-5104

ICT in Practice

Issue 18 / Autumn Edition October 2017

Editorial

This month started with the celebration of the 5th birthday of Code Week EU. 1000s of people around the EU, young and old celebrated creating and having fun with code. In the UK alone, we had over 200 events organised across the country. You can read about some of the highlights of this event on page 6.

In this issue Maggie Morrissey discussed the use of floor robots in literacy lessons. She specifically focused on Roamer Too. I think that this is really important as integrating computing activities into other subjects can make learning more accessible and meaningful for our diverse learners.

We love hearing from teachers who have tried different tools and approaches in-link to technology in their classroom. This time we have a guest teacher from Israel, Efeat Maatuf. She shared her ideas about how using digital tools for evaluation in the learning process can be effective.

Coding has become the focus of many educators especially in recent years. Many teachers found it challenging to integrate coding activities into their curricula. Please do share your tips and experiences of teaching coding to children with us, so that we can support each other. In this issue Garrett Zimmer and Mark Grundel, co-founder of MinecraftEdu chat, gave us some insights of their approach to coding and computational thinking using examples from their own experiences.

I hope that you will enjoy reading this issue and continue to support our magazine by sharing with your colleagues.

Yasemin Allsop

Editor

Contents

Computational Thinking goes far beyond coding by Garrett Zimmer

Two young boys sit beside their father at the computer desk, one is paying rapt attention while the other stares out of the office window pining for the outdoors where his roller blades and ramp are waiting. The young boy is quickly jolted back to the conversation at hand! "Programming is so important for your future son, so pay attention." Boy number one grew up enjoying these programming lessons, and eventually became a programming engineer, whereas boy number 2 remained active in sports and didn't pick up programming till much later in life. Yet those introductory lessons served him well and taught him to think deeply and problem solve in a logical way.

I was boy number two, and my story illustrates a few of the key challenges that we often face when trying to teach computational thinking and coding. For me it was one of 'Why will I ever have to learn this when I have absolutely no desire to be a programmer?'. The questions and decisions around this affect both parents and teachers alike. How can we get the "actives" interested in computational thinking? How do we impress upon all that computational thinking goes far beyond coding and can develop key career skills for all?

I for one believe that computational thinking is a skill-set and a practice that must be taught early and taught in such a way that allows all kids, regardless of the activity preference to participate and find enjoyment. Again Computational thinking goes far beyond coding and really does have a place for the jocks and the deeply active child.

I'll share my experiences of trying to teach my very active niece and nephew how to code, and more importantly teaching them in a way that supported their motivation and passion and didn't rely on a desire to code or program. I decided to use a tool from Microsoft's Minecraft: Education Edition called Code Builder. Code Builder is a relatively simple coding application that supports block based coding languages like Scratch, Tynker and of course Microsoft's new Make Code, and allows players to program a little robot inside the game called an Agent. By choosing blocks tied to Loops, Variables and Events players can make their robots do all sorts of things within the game.

Now of course when player's program an agent to perform a variety of tasks, there is computational thinking at work, but if we want these skills to be intuitively connected to areas outside of coding or to support the motivations of non-screen-centric families we have to focus less on the screen-time and more on the thought process related to kids passions. In my case

my nieces and nephews love Minecraft, but they were also extremely active kids and I didn't feel that the motivation of simply making an Agent build for them was going to be enough. So, I thought of a solution: Lets combine some physical activity to the mix to develop computational thinking skills. It all started with a question: "How do we get our agent to jump off of a platform and jump back?". Jumping and running of course is an intuitive skill that all kids understand, however the computational thinking process asks us to break these activities down into it's key parts. In essence we need to plot the path that our robot will have to take to make it realistic. In physical sports it's this aspect of plotting the actions that a coach takes when training players to play football.

So, how do we get a Robot to Jump? Well we need to understand what a jump looks like. If you've ever asked a 7 and 9 year old to start jump off a couch, you'll quickly see some dangerous activity happening, and so we took the activity outside and began jumping all around. Not only were we jumping around and having fun, but because our objective was to understand a jump, we had some deeper thinking happening about the entire process of jumping.

The long and short of it is that this activity worked, engaged them in coding, in physical activity, and supported deep meta-cognitive understanding of computational thinking for these kids. An added bonus was the discussion between these two siblings about gravity and weight. But that's for another time.

The next challenge for them when they arrived home was to practice their favorite sports together, and methodically track for each other, the steps involved in each action, and how to improve upon these. I explained it to them like this: If you look at a soccer field at any point in the game you can use the process of computational thinking to find the patterns that will help you win. But only if you practice, learn and can replicate those steps, and that is how computational thinking and coding can help you succeed with your physical activities.

Now I want my nieces and nephews to learn how to code, not because they should be programmers in their careers, and certainly not because I want them in front of screens for much of their lives. My motivation for having them learn computational thinking and coding is simple: Having the skills and becoming adept at seeing big challenges as a series of smaller pieces, understanding 'cause and effect', and being able to quickly solve problems that are going to support them in whatever their career choices are.

Whether you or your child will be a Computer Engineer, a Star Football player or a Car Mechanic, teaching them computational thinking, in line with their motivations and passions, will undoubtedly help them succeed.

Celebrating the 5th Birthday of Codeweek EU in the UK

by Yasemin Allsop

This year's Codeweek EU kick-off event in the UK, took place at an impressive venue called Base KX in Camden, London. It all started with a surprise, and unexpected visit from Georgia Gould, head of Camden Council and Chris Shaw from the Shaw Corporation, the founder of Base KX. I was so happy to see Georgia Gould coming at that early time to open our event. Her energetic and positive attitude set the mark for the rest of the day.

We then had two very interesting talks. Jane Butler, Vice Dean for Enterprise at UCL engineering blew our minds away when she told us that she was part of the team who designed the computer that Prof. Dominic Wyse shared in his presentation. She also shared her story of how she became an engineer. This was interesting, as she explained that her school was very unsupportive of her decision to study Computer Science, however, her family was very supportive of her ideas and decisions.
I think that it is important to engage parents with children's computing activities, so that they can support and encourage their children just like Jane's parents did.

We also had Prof. Dominic Wyse from UCL IOE, who talked about the relationship between writing, music and coding. He shared some images of Egyptian artifacts from his book 'How Writing works' showing how music was presented in Egyptian hierogliphics.

Then the full speed fun began! From robotics to unplugged coding activities, we had many activity stations across Base KX. It was clear that the children were having fun with coding and digital making as 150 children from 15 schools from different parts of the UK filled UCL BASE KX in Camden with joy and laughter! The feedback we received from both the children and the teachers was just amazing. They told us that they couldn't wait for next year's event. I hope that we managed to inspire some young minds and gave ideas to teachers, so that they can continue to create with code in their classrooms.

After lunch we had a Skype chat with the Payas Stem Centre in Payas, Hatay, Turkey. They have done amazing work with both Syrian refugee children and Turkish children coming from disadvantaged backgrounds. We let the children share what they have been doing for EU Code Week. You see, coding is like a common language for kids, they had no problem with communicating.

At the end of the workshops we let each school share their feelings about the day and what they liked the most. We were able to give a Microbit class set to each school that attended our event. For this we are grateful to the Microbit Foundation.

We received support from amazing companies and organizations for this event, without which this event couldn't have taken place. So a big thank you to; UCL Computer Science, UCL Engineering, UCL IOE, Discovery Education, Ohbot, Lego Education, Primo, Roamer, ICT in Practice, Redfern Electronics-creator of Crumble, Mama Codes, Institute of Imagination, Dare Collaborative. Thank you for being part of our exciting Codeweek EU event. Also thank you Base KX for hosting our event, we need more community spaces like this to accommodate more innovative events.

Most importantly thank you Svitlana Yarmolchuk, Dr.Elpida Makrygianni, Katty Potts (Computing and E-Safety Lead for Children's Services Islington Council) and Simon Humphrey's from Computing at School. Without your fantastic work behind the scenes we wouldn't have been able to help 100s of children to have fun with coding.

Finally, we would like to give a special thanks to Google Europe for their generous support towards the organization of our Kick off event.

The Codeweek EU celebration in the UK continues with an amazing competition by the DARE research centre at University College London. We are calling on everyone to make Viking games, using their popular game-authoring tool Missionmaker. This version of Missionmaker was designed for a project on the Anglo-Saxon poem Beowulf, so its environments and characters are perfect for making Viking games.

Please use the following link to register:

http://creativeedutech.com/products/missionmaker-beowulf-codeweek/

Using digital tools for evaluation in the learning process

by Efeat Maatuf

Using digital tools for evaluation in the learning process can be effective in several aspects.

Using various media

Online tests enable the integration of multimedia - videos, animations, simulations, audio files, pictures, maps, articles from the Web,and more.

Since students can find answers online, it is recommended to minimize simple knowledge questions and focus on questions that require high-order thinking.

Using different types of questions

Online tests encourage teachers to use different types of questions - such as open, multiple choice, analysis of various types of texts (articles, videos, caricatures, etc.), word completion, matching, and more.

Fast and convenient data collection

The information regarding each student or topic is received in an organized manner as soon as the activity is completed.

Immediate feedback

The student can receive immediate feedback about their achievements, a great advantage that helps shape the learning process, especially when the assessment is part of the preparation for a test.

Time saving

Many digital tools allow the teacher to choose the option of automatic checking and scoring in Multiple choice questions, thus shortening the testing process.

Paper saving

The use of digital tools saves the use of paper..

Wizer is a wonderful tool that offers everything I have described so far.

Determining learning progress by success
In some tools, such as Versal, a teacher can block further progress in online activity (course / lesson) until the student achieves the pass grade in a specific task.

Peer evaluation
A teacher can enable their students to deepen their understanding of the subject by making them evaluative. Nipagesh enables peer-based indicator assessment.

Unreadable handwriting
Typing makes it easier for students with writing difficulties. And saves time for the teacher during the task Correction

Reading the questions
A teacher can adjust the exam to suit students who need the questions read out loud. This is done by adding audio clips that read the questions and texts. Voocaro and Voki are very suitable for this purpose.

Evaluation of behaviour
Evaluation can also be given through gamification, to encourage positive behaviors. a tool which could be used for this purpose is CLassdojo.

Class mapping
The results of the online assessments are usually displayed in a way that allows the teacher to examine the success of the class in a specific question or subject. The information can help the teacher to evaluate the level of their success in imparting knowledge or specific skill to his students.

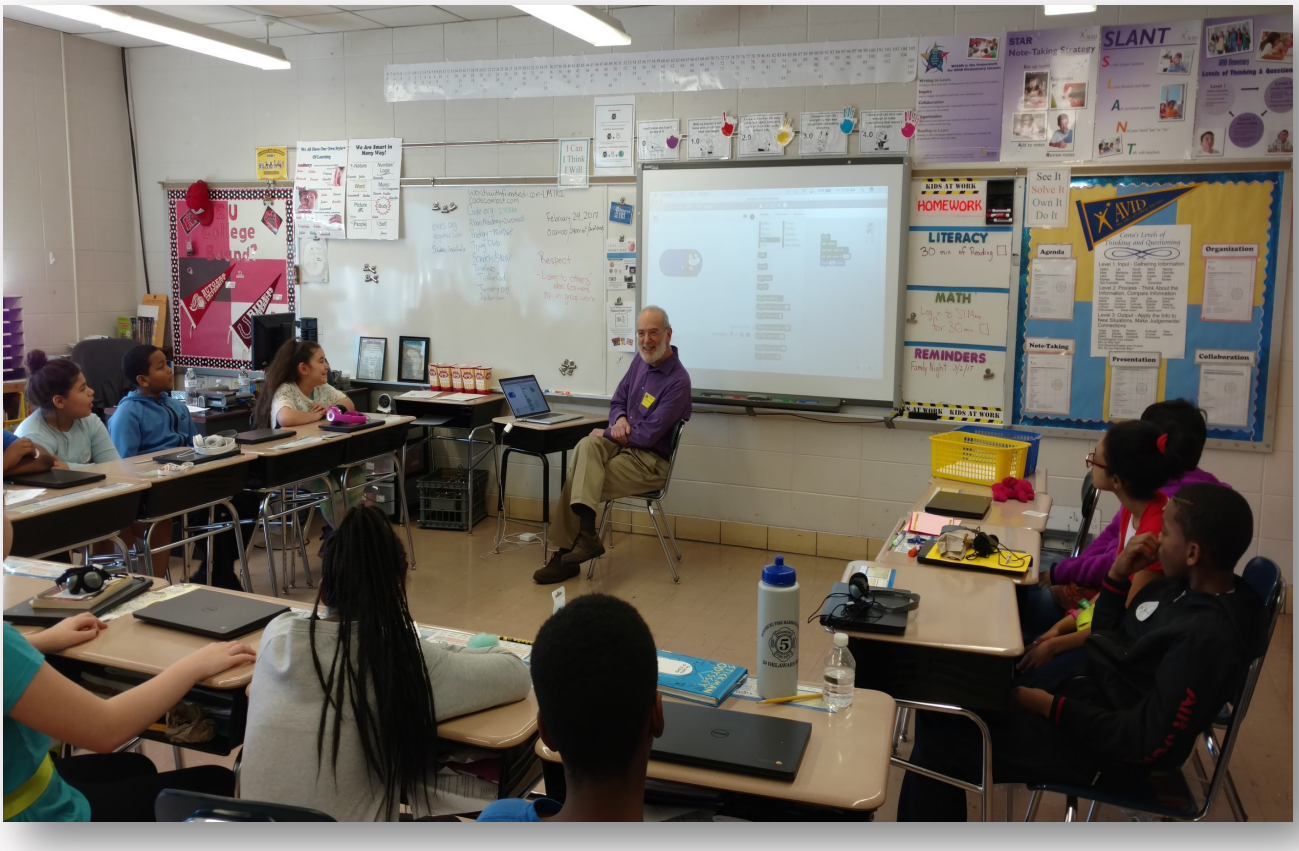
Class mapping can be done also during a competitive game, at the end of learning, with tools such as Triventy or kahoot.

Getting feedback from students
A teacher can also use online tools to receive feedback (anonymous or not) from students about the lesson and teaching methods. You can create a feedback questionnaire, as well as online testing, using Google Forms.

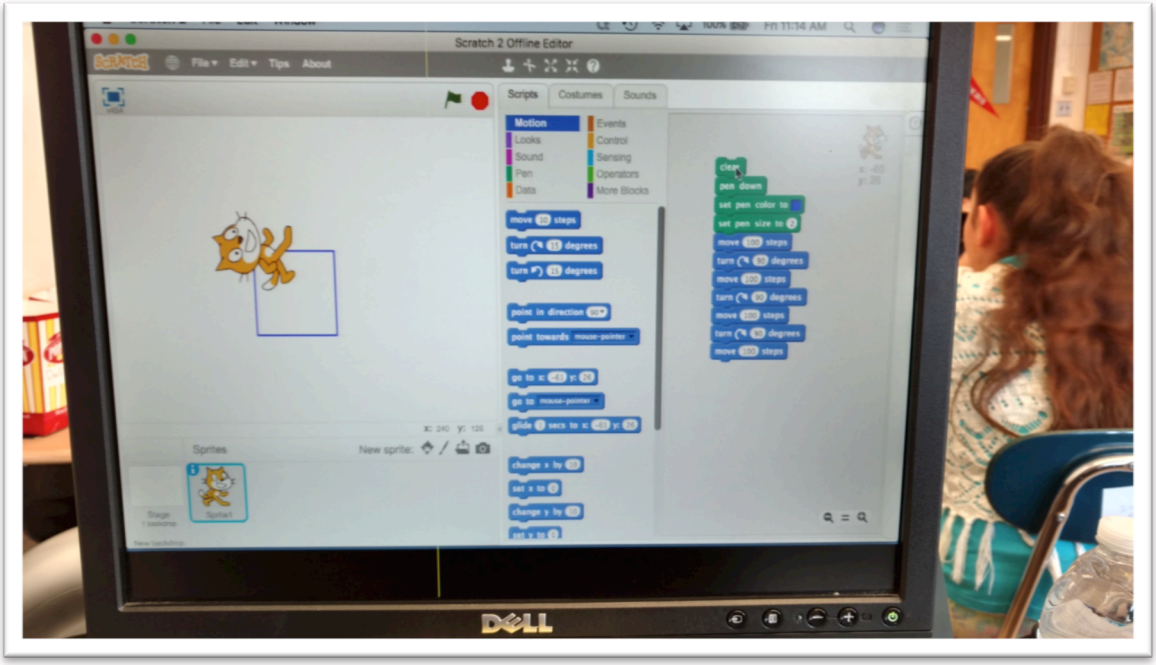
Coding in the Second Dimension – Using Scratch to Learn About Exterior Angles

by Mark Grundel

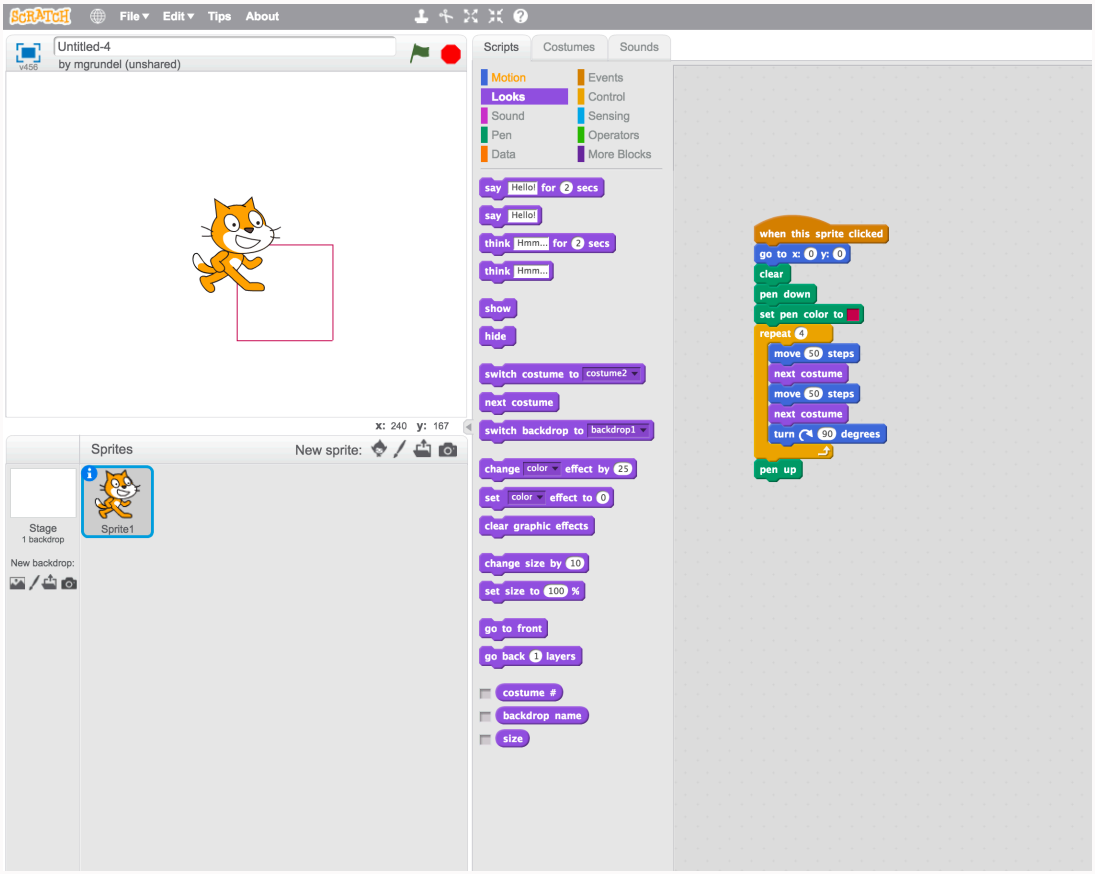
I originally used this lesson in conjunction with the Hour of Code week, where we invited retired Professor, Stephen Garland, from MIT into our classroom to learn about coding and computational thinking. At the time, we were studying decimals in mathematics and wanted to incorporate coding into our lesson design. This lesson achieved our initial goal of meeting the Common Core Standard: 5.NBT.A.2: Explain patterns in the number of zeros of the product when multiplying a number by powers of 10, and explain patterns in the placement of the decimal point when a decimal is multiplied or divided by a power of 10.



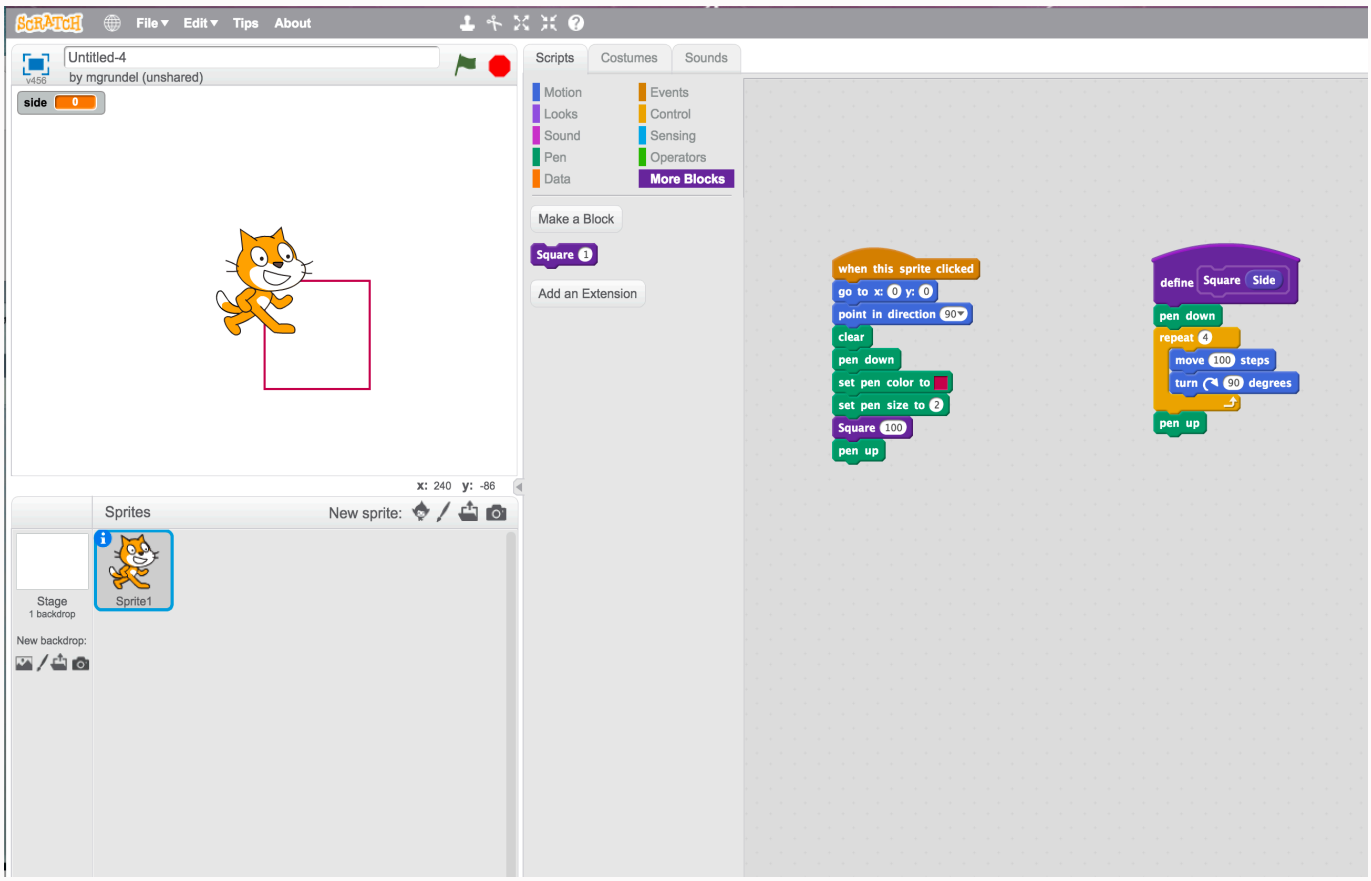
Our first task was to write simple code in Scratch that allowed the cat to draw a square. Through trial and error, we eventually discovered how to move the cat in a square like motion, but could not get the cat to draw, until we discovered the “Pen Down” block.



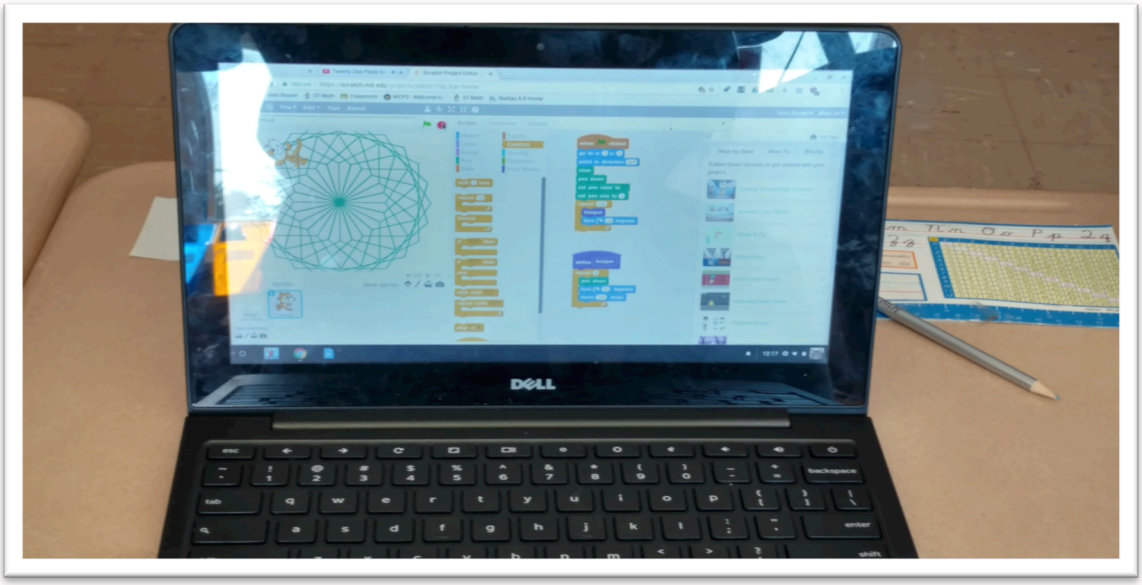
From there, we learned how to write our code more efficiently using loops, such as the repeat block. Some children broke the cat's movements up into smaller components and added a costume change, for the cat to move in a more realistic manner.

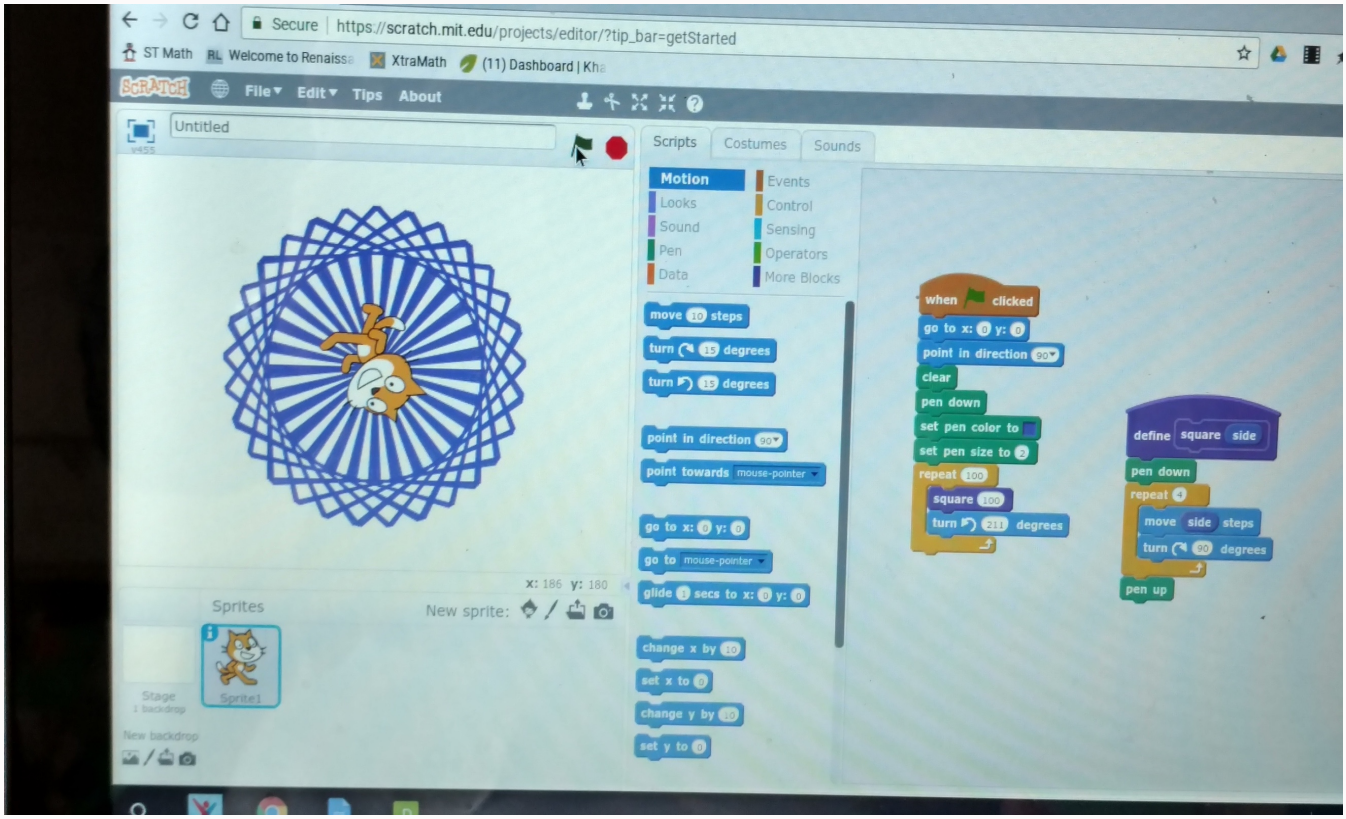


The next task for the students was to determine how to create a block, which would draw a square from a single block. This was accomplished using the “More Blocks” tab under “Scripts.” When making the block, students used the “Options” feature to add a number input. We called this number input “side,” but any name would work for this variable block.



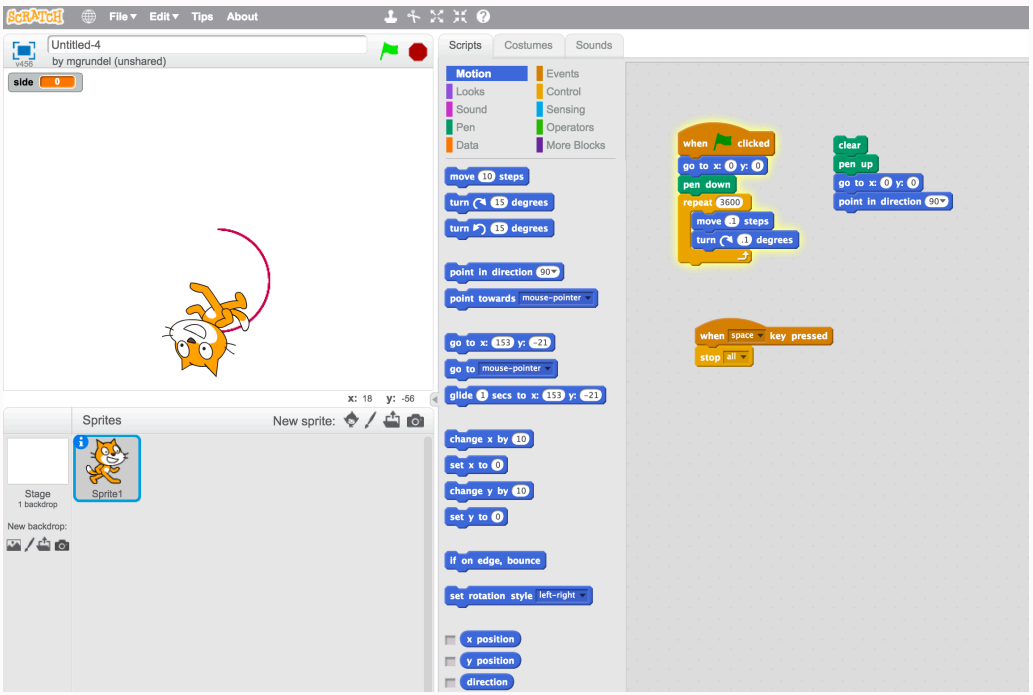
From here, students were tasked with various challenges, such as drawing repeating patterns and asked to discover how to create additional polygons, using what was already learned. The purpose of this challenge was for students to discover, through trial and error, that the sum of the angles must equal 360 degrees, to create a complete, regular polygon. Here are samples of repeating patterns with squares and hexagons.

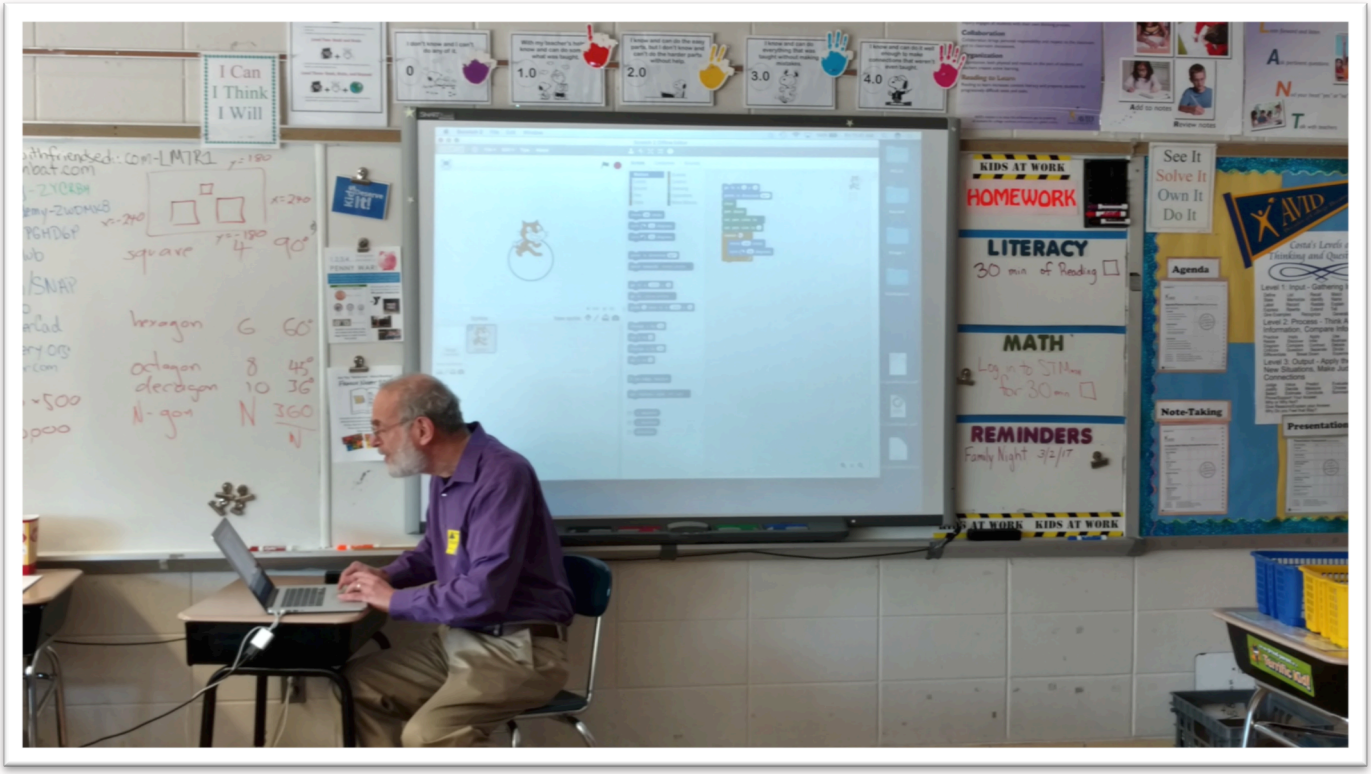




Children enjoyed the drawing aspect of the class, and without explicit instruction, could determine that then sum of the exterior angles must equal 360 degrees, to create a complete regular polygon.

Our final task was to challenge students to create a circle using similar code in Scratch. Students worked collaboratively and discussed ideas excitedly. My ultimate goal was to have students use decimals and man steps to achieve a relatively accurate looking circle. Some early results created near circles, but there were also some misses. This proved very time consuming, but the children eventually refined their circles and were able to discover, on their own, the decimal relationship of powers of 10.





In the end, through hands on coding, I was able to assess learning and understanding of the standard through the completion of the students' workspace in Scratch. However, the tasks and learning that took place beyond the standard, far exceeded 5.NBT.A.2; students worked collaboratively, hypothesized, tested ideas and iterated throughout the entire process. Students had to build their foundational knowledge of exterior angles to understand why they must rotate the cat a total of 360 degrees before they could create a complete circle. Students also gained a deeper knowledge of regular polygons, such as squares, hexagons, and many other shapes.

Finally, as the lesson ended, students added personal touches to their drawings, adding backgrounds and music, taking ownership of the projects. We said goodbye to Mr. Garland and hoped to invite him back to our 5th grade classroom shortly. Not only did we learn about angles, decimals, and block-based coding, but, we added a friend to our classroom community.

Maggie's Corner

In Maggie's corner this time I will be writing about using Roamer in literacy lessons. When I first became an ICT coordinator I purchased three Roamers as part of our school resources. These were of course the classic Roamer and the kits came with activities and mats for the robot to move around on. Typically at that time, they were being used in year two, but I decided that I wanted to use one of them in my year four class during a Literacy lesson. In particularly, teaching instruction writing.

Most teachers are familiar with the frustration of getting children to write instructions without missing at least one or more steps in the process. As we all know if a step is missed out when programming a robot, then the action will not be carried out. I wanted to try out the Roamer to see if this would encourage the children to write instructions correctly. I also wanted to use the robot because the class that I was teaching at the time had a wide range of abilities along with many children who had English as a second language. Using signs and symbols instead of long sentences was a chance for these children to participate fully within the lesson. The aim being, that they would then learn to build up sentences, once they had written the instructions correctly. Well I am pleased to say that using the Roamer worked really well, with many children realising how important it is not to miss out a step. Some more confident children also went on the use the repeat function. For example when making an L shape one child noticed that if this was repeated, then the shape made would be a square.

Following the success of this work in a British school I wanted to see how this would work with children whose first language is not English and who are based in a school out of the UK.

During the academic year 2014/2015, Dave Catlin of Valiant Technology gave me a robot to use in a school I was teaching in in Moscow. This time I took the new ROAMER TOO; this robot is smaller and has additional features.

If you would like to find out more about the functions of this robot then check out Valiant Technologies Website http://www.valiant-technology.com/uk/pages/roamertoohome.php?cat=8&8

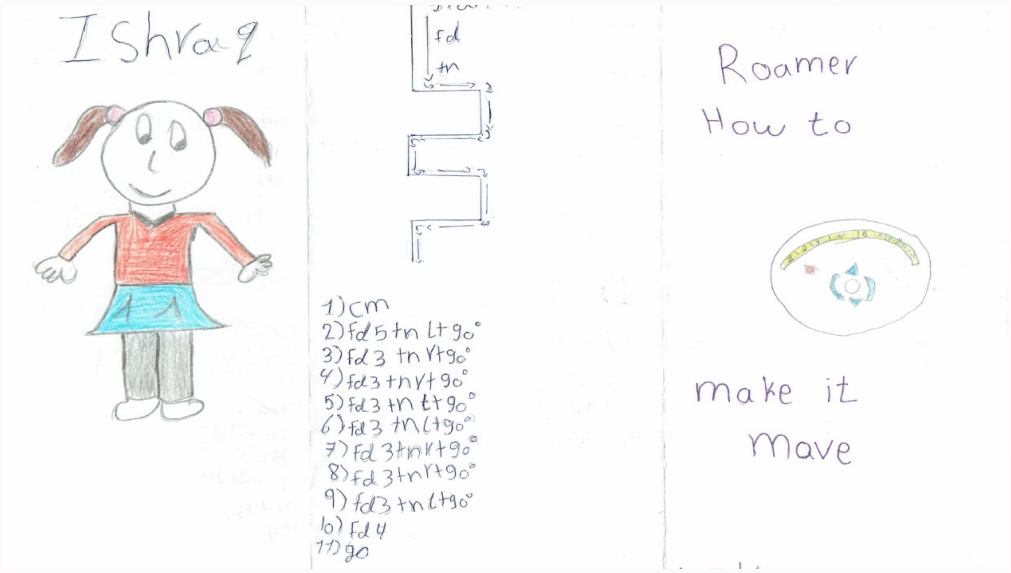
At the time, class teachers were being asked to cover ESL lessons. Most of the resources for these lessons were workbooks and I did not feel comfortable using these. Learning a new language should be engaging and workbooks like this didn't inspire me and I should imagine they were unlikely to inspire the children. Therefore, I set about organising a short scheme of work for the children.

The lessons were approximately 40 minutes long, three times a week. The first few sessions involved getting to know how the robot worked, they were to draw out simple shapes starting with a straight line and moving on to an L shape with more confident children trying more challenging moves.

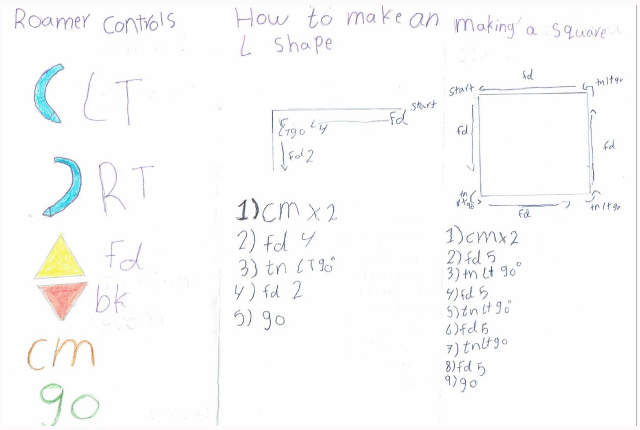
It is worth pointing out that I didn't expect them to use any of the more complex features of the robot, the main aim was to motivate them to draw and write clear instructions.

Once they had developed their confidence with the robot, I then moved on to introducing them how to write instructions, what they were for and gave them examples of different types of instruction writing. They had plenty of chances to redraft and refine their work and they eventually produced some very nice instruction booklets.

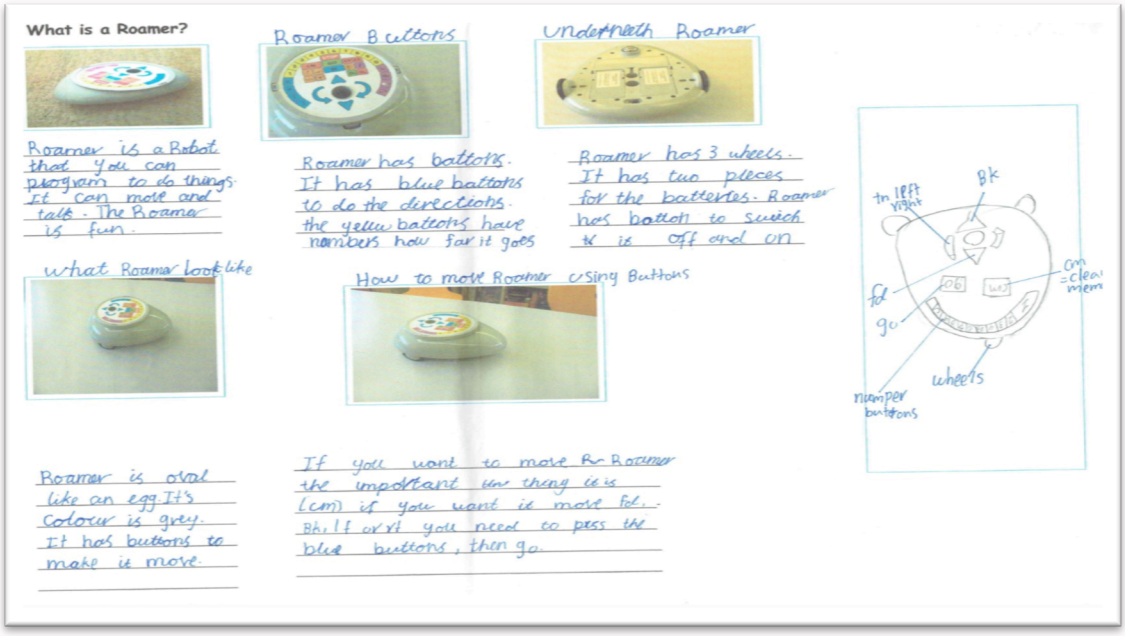
The young girl who did this work was not only new to English but also the Russian language and she had only arrived into the country in September 2014.

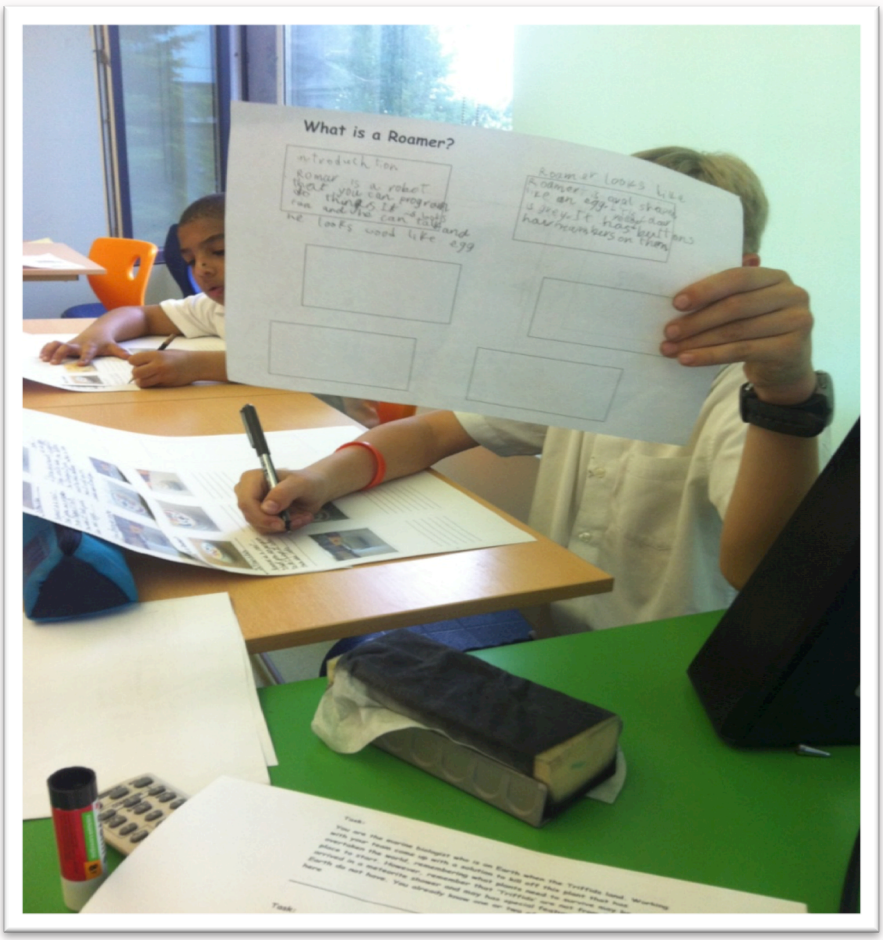


Following on from this work, I decided to introduce the children to explanations. Once again they were going to write about the robot and this time put some more detailed writing into their reports. I designed a short scheme of work to cover examples of explanations, the differences between them and instructions, as well as how to label diagrams.



Both units of work followed the style of the Literacy strategies with amendments for children whose first language is not English. Sessions took slightly longer; vocabulary needed was available on cards along with pictures to support the children. Having the Roamer to hand was also essential as they could then use this to go back and remind themselves about the key features of the Robot.





It was a lovely chance to show that it is possible to integrate computing into literacy lessons and I am sure similar types of robots and technology equipment could be used to help stimulate children's writing. In addition to this, it was also good to see how a good stimulus with both language support and clear concrete objects could help children who are new to English, join in with the same lessons as their peers.

An outline of the schemes of work will be available over the summer from www.technologytoteach.co.uk and http://www.valiant-technology.com/uk/pages/roamertoohome.php?cat=8&8

I also believe that it is possible to use the ROAMER TOO with other writing genres. If we had had more time, I was going to get the children to write an adventure story with the robot as a main character. We have a lot of creative teachers out there and I am sure many could think of other types of writing that could be developed using a programmable robot as the main focus.

If anyone does try to do this, or any other work, it would be lovely to hear how you got on.

Power up STEM learning in Minecraft Edu

by Yasemin Allsop

This week let's have a look at how we can use Minecraft Edu to help children to develop STEM skills (Science, Technology, Engineering and Mathematics). I think that it is important to first explain briefly what STEM learning is and why it is important.

What is STEM Learning?

In a recent chapter that I wrote, I explained what STEM learning is and how it relates to Computer Science in detail. I would like to share a paragraph from this chapter as I think that it clearly explains the main elements of STEM education.

"STEM stands for science, technology, engineering and mathematics. STEM, STEM education and STEM learning are terms that have been used in an interchangeable manner. STEM education aims to blend scientific inquiry and technological design processes through project based learning that focuses on developing students' critical thinking, problem solving, logical reasoning, technical, communication, collaboration, self-directing and creativity skills. STEM learning gives children opportunities to investigate an idea in different contexts and connect the learning across disciplines. Learning in this way becomes more relevant to students as they can draw learning points from their activities in different disciplines to construct meaning. This purposeful integration of learning cannot be merely seen as cross-curricular learning, as it requires learners to use higher order strategies to facilitate creative and critical thinking for solving real-life problems. They need to be able to deploy their cognitive resources to organize, transfer, apply and evaluate their knowledge and skills in different disciplines through integrated activities. Additionally they need to have the ability to direct their self-learning process, which can be seen as metacognitive awareness." (Allsop, 2017).

Developing STEM Skills: Ideas in Minecraft Edu

There are 100s of fantastic Minecraft Edu lesson plans available on https://education.minecraft.net/class-resources/lessons/ for you to use in the classroom to support children developing their STEM skills. The aim is to provide children with opportunities to learn the same material in different contextual settings so that the students can draw learning points from their activities in different disciplines to construct their own understanding and learning.

Tackling Real-life problems

In Minecraft Edu, students can design solutions for real-life problems. For example children can design a dam to solve a water problem for a city or a car to reduce the pollution in their local area. They blend the concepts from Science and engineering through design, tinkering and critical thinking. They might;

- work with different materials
- experiment with design ideas
- use knowledge and skills from different disciplines such as using mathematics skills to decide the size of the dam for the population or science & maths skills to think about wind power for their car design.

Image by Stephen Reid

Stephen Reid (@ImmersiveMind) has really nice projects on his website exploring solutions to many real-life problems using Minecraft Edu.

https://www.immersiveminds.com

Computational thinking

It is often said that computer science is the silent "C" in STEM as it has very strong links with mathematics and science as well as design and technology. Some of the STEM skills that we share can also been seen as Computational thinking skills. In my opinion some of these skills, which have been, defined as transferrable skills for years also fit into other areas under different terminology. Minecraft Edu doesn't only provide opportunities for children to develop their CT approaches such as tinkering with ideas, persevering, creative thinking, working collaboratively, problem solving through design and creating; but also concepts through Code Builder for Minecraft Edu.

Image by Simon Johnson.
http://teachwithict.weebly.com/minecraft-code-skyscraper.html

Simon Johnson (clcsimon) has some really good activities for developing computational thinking skills using Code Builder.
http://teachwithict.weebly.com

Engineering with bricks

In Minecraft Edu children can experiment with

levers, switches and electrical circuits just as in real life using Redstone. They can create farming systems that harvest themselves. They can explore water cycles and renewable energy. They can even design an eco system and explore ways of storing energy.

There are some brilliant engineering projects on the Minecraft Edu website: https://education.minecraft.net/class-resources/lessons/

Mathematical thinking

Minecraft Edu provides a space for learners to visualize and investigate different objects and patterns from real life, which helps them to develop spatial thinking. Manipulating objects in a virtual world also enable learners to test their solutions for problems that they wouldn't be able to evaluate in real world. This process requires not only technical skills for using Minecraft but also critical thinking and logical reasoning for visualizing and predicting the outcomes of their solutions. It is important to give children the time and opportunities to discuss and explain their solutions, as this would allow them to make meaningful connections that would help them to construct their understanding. There are many activities for developing mathematical thinking on the Minecraft Edu website.

Finally, one cannot expect children to develop their STEM skills without facilitating children's learning using appropriate teaching methods and tools. Teachers should provide opportunities for learners to work collaboratively, time to talk about their work and space to express their ideas. Most importantly teachers should allow learners to manage and regulate their own learning process. Minecraft Edu enables learners to build together and experiment with their ideas constantly. The interesting point is that when children break the bricks, they don't feel as though they were destroying something or that they had failed, because they constantly create something new using those broken bricks. In this learning scene, students make decisions and monitor their own activities, which is the vital ingredient for learning to occur!

Reference

Allsop, Y (2017) Computer Science: Silent C in STEM. In: Humble, S. Creating the Coding Generation in Primary Schools. London: Routledge.

Another version of this article was first published on the following website: https://education.minecraft.net/powering-up-stem-with-minecraft/

Call for Papers – Autumn/Winter 2017

The International Journal of Computer Science Education in Schools (IJCSES) is committed to increasing the understanding of computer science education in schools by publishing theoretical manuscripts, empirical studies and literature reviews. The journal focuses on exploring computer science education in schools through pedagogical, cognitive and psychological perspectives.

IJCSES welcomes high-quality research articles from academics, educators, teachers, trainers and other practitioners on all aspects of computer science education in schools. Papers for publication in the International Journal of Computer Science Education in Schools are selected through peer review to ensure quality, originality, relevance, significance and readability.

Authors are invited to submit papers to this journal through the online submission system. Submissions must be original and should not have been published previously or be under consideration for publication while being evaluated by IJCSES.

Call for Papers for January the 2017 Issue:
Last date for submission: 15 December 2017
Notification deadline: 15 January 2018
Camera-ready ready: 31 January 2018

Call for Papers for the April 2018 Issue:
Last date for submission: 15 May 2017
Notification deadline: 15 April 2018
Camera-ready ready: 30 April 2018

You can view our current and previous issues at http://www.ijcses.org

All our papers are indexed at Google Scholar and ERIC.

Editors: Filiz Kalelioğlu and Yasemin Allsop

ISSN: 2513-8359 (Online)
Website: http://www.ijcses.org

Email: info@ijcses.org

Published by
ictinpractice.com
London, UK