Enthusing the next generation: Teaching computing in Primary schools

University of Manchester

School of Computer Science

BSc (Hons) Computer Science and Mathematics

Final year project, May 2016

Author: Aisha Saini

Supervisor: Dr Paul Nutter
Abstract
Title: Enthusing the next generation: Teaching computing in Primary schools
Author: Aisha Saini
Supervisor: Paul Nutter
May 2016

In 2014, a new national curriculum was introduced at Primary school levels Key Stage 1 and 2 that focuses more on computational thinking which aims to teach the fundamental principles that underpin Computer Science. The change led to teachers teaching a subject they are uncomfortable with as they do not hold much Computing knowledge, therefore, lessons may not be as exciting for children as they could be. To address this problem, this report describes the design, implementation and testing of a range of fun activities to help support teachers with the new curriculum and to excite young children in Key Stage 2 (aged 7 – 11) about computing. Through researching and defining successful activities, three activities were developed which were named: Maze Game, Star Switch and Crypto CodeBug. These were tested in schools and evaluated using self-evaluation, the feedback from teachers and judging the children’s engagement to measure the success of each, which led to further improvements in the activities. The evaluation showed that the overall aim to ‘enthuse the next generation’ was met through the development of successful activities which supported the new curriculum. To ensure children are continuously being exposed to these types of activities they need to leave a legacy for teachers to try out themselves. The success of the activities can be highlighted by the positive feedback received from teachers. One particular teacher said ‘Thanks again for a fantastic afternoon. Your sessions looked like great fun and gave the children a really exciting alternative to their usual computing sessions.’
Acknowledgments

I would like to say a big thank you to my supervisor Paul Nutter for his constant support and guidance throughout this project. Thank you to Sarah Zaman and Duncan Hull for helping me gain permission to go into Primary schools. I’d also like to thank David Rydeheard, David Lester and Stephen Rhodes for contributing to my project.

Thank you to all the schools that I attended.

Finally, I would like to thank Alex Glenny who has continuously encouraged me throughout my project as well as my family and friends.
Chapter 1: Introduction

1.1 Why teach computing at schools?

In 2014, a new computing curriculum was introduced which replaced information technology. This was to ensure children had a good understanding of the technologies that surround them whilst providing them with more interesting topics to learn about. Due to this change, teachers were expected to teach these topics without much computing expertise so they might feel uncomfortable using the new technologies invested by schools. Many primary school teachers do not possess much computing knowledge and will stick to their comfort zones so lessons may not be as fun as they could be. This is where this project comes into it. Through the development and teaching of exciting activities children can be inspired by computing and would want to learn more. These activities will support the new curriculum by portraying computing concepts primary school children have been previously introduced to so they can build on their previous knowledge.

It is important for our young generation to understand how technology works as they are surrounded by it and have grown up with it in their lives. Within the new curriculum topics such as E-safety, information technology, digital literacy and computer science are covered. This project focuses on the computer science aspect which includes: algorithms, coding, de-bugging and networks [1].

A conference about what schools are currently doing and can be doing to encourage computing in schools was held at the University of Manchester on 17th October 2015. It was organised by Computing at School [2] and David Rydeheard. At this conference, the new curriculum was covered in depth as well as having workshops to introduce many different technologies that are being used for different age groups such as CodeBug, Arduino and Makey Makey (some of which will be explored later within this chapter). One of the lecturers gave a talk and discussed the approaches teachers take and the amount of guidance they give to their children which were really useful to know for trying out the activities in schools. This highlighted the challenges that may arise during this project which will be discussed later in chapter 3.

1.2 Aims

This project is very different to other computer science 3rd year projects as activities are being developed and not a program/system. This makes it harder to define the functional and non-functional requirements so instead it will have aims, objectives and goals. These will be used as a measure of the success of the activities.

The main aim of this project is to ‘enthuse the next generation about computing’. To achieve this aim, the following objectives will be met:

- To develop activities
- To test out the activities within Primary Schools
- To improve the tested activities
Each activity should meet the following basic requirements which can be treated as goals to fulfil the objectives [3]:

- **Excite young children**: Eliminates the ‘computing is boring’ idea.
- **Encourage creativity**: Allows the children to produce interesting products.
- **Explain how technology and computing concepts work**: Children should be able to understand how things work e.g. an email is not sent by magic.
- **Explain how Computing is a Science**: Shows there is more to computing than ‘cool gadgets’.
- **Introduce the uses and applications of Computing**: Not just limited to internet and phones.
- **Provide a sense of achievement**: Shows how rewarding computer science can be.
- **Introduce career options**: A child should be able to ask themselves ‘Is this for me?’
- **Let the children take something away from the activity**: The children can take what they have produced home to show their family and friends and could even develop the activity further.
- **Use games and competition**: These make the activity fun and engaging.
- **Cater to both genders**: Should change the perception that ‘Computing is for boys’.
- **Leave a legacy**: Be able to educate teachers as well as pupils.
- **Tailor to schools resources**: Some schools may have invested in a range of technology whereas other schools may only have a few computers.
- **Target all capabilities**: Children have different abilities, for example, some may be able to complete tasks quicker and easier than others. Some children may not speak English very well or some may have special educational needs. The activities should cater for every child.
- **Meet the curriculum**: The activities should support the curriculum to help reinforce key concepts.
- **Target Key Stage 2 (KS2) children**: This project is aimed at this target audience which are Years 3, 4, 5 and 6.

### 1.3 Types of activities

Throughout this project, it is important to tailor to schools resources. Some schools have more money than others and therefore may have more technology. When developing activities for these schools it is easier to make the children excited as they have ‘cool’ gadgets to play with whereas with schools with less technology other approaches have to be considered to try and make children engaged.

There are two types of activities: unplugged and plugged [4]. Unplugged is teaching computer science without a computer whereas plugged is with a computer. The former is good when a school is limited in terms of time, hardware and computers as you can teach fun computing concepts by moving around or using the traditional pen and paper. An example of an unplugged activity named Count the dots will be now be explored [5].
The children are given a set of cards which represent the power of twos (see Figure 1). They would turn some over (which represent zero) and keep some facing up (which represents one). Then the children would count the dots on all the up facing cards which would correspond to a number. This is a really good way to teach binary to young children as it makes it fun by introducing gamification.

Another example of an unplugged activity is teaching algorithms through the ‘Jam Sandwich activity’ [6]. This is where the children give the teacher a set of clear instructions to make a jam sandwich and the teacher follows these precisely. For instance ‘take some jam out’ is not a good instruction. ‘Use the knife to pick out some jam’ would be better. This makes the children really think about the order and the precision of instructions expected in an algorithm. It is also a fun way to teach algorithms and requires cheap resources, therefore, this is a really good example of an unplugged activity.

Plugged activities are more applicable when the school has invested into hardware and want to use them. Although it is important to note that if schools have bought new products to try out that does not necessarily imply that teachers know how to use them. This project will support the teacher with computer science knowledge. Plugged activities would integrate one or more hardware/software. For some examples of the hardware used within plugged activities see Figure 2.

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>CodeBug</th>
<th>Makey Makey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lego MindStorms</td>
<td>Minecraft</td>
<td>Scratch (software)</td>
</tr>
</tbody>
</table>

Each one has its own advantages and disadvantages which determine which one should be used to teach a particular computing concept in the classroom. For a list of these pros and cons see Table 1.
<table>
<thead>
<tr>
<th>Name of hardware or software</th>
<th>What it does</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi [7]</td>
<td>A small computer that plugs into a computer monitor or TV that allows you to do anything, from browsing the internet to playing games.</td>
<td>Has a range of capabilities</td>
<td>Would require some sound knowledge to teach to a class.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>May not be seen as exciting as may be difficult to use for KS2 children.</td>
</tr>
<tr>
<td>CodeBug [8]</td>
<td>A piece of hardware to display text and graphics.</td>
<td>Simple to use</td>
<td>Limited capabilities as can only use the LED lights to display something.</td>
</tr>
<tr>
<td>Makey Makey [9]</td>
<td>A device that emulates a keyboard. You can connect everyday objects to the Makey Makey to act as keys.</td>
<td>Easy to use and can connect to Scratch to open up possibilities</td>
<td>Limited in use as it is mainly used to make the controls for a game.</td>
</tr>
<tr>
<td>Lego Mindstorms [10]</td>
<td>Program a Lego Mindstorms robot through mazes and obstacles.</td>
<td>Easy to use</td>
<td>Very expensive</td>
</tr>
<tr>
<td>Minecraft [11]</td>
<td>A popular game especially for children where they can create and build their own virtual worlds.</td>
<td>Easy to play</td>
<td>Would need licenses for each child which costs £20 each and would need to pre-install</td>
</tr>
<tr>
<td>Scratch [12]</td>
<td>A drag and drop coding application which allows you to develop programs and animations. Can be installed on desktops or accessed via the web.</td>
<td>Easy to use and develop a range of animations and programs</td>
<td>Children would need some experience in coding to create their own programs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Suitable for Key Stage 2 children</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Free to use</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Taught in the new national curriculum</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Table of advantages and disadvantages of some hardware and software that could be used in a plugged activity
Now the hardware and software examples within this table will be narrowed down to obtain a few to use in plugged activities that are suitable for KS2 children.

From this table, we can see that CodeBug and Scratch are the most accessible as they are the only two in the list that are free to use online. This means that children can go home and add more to their work via the websites. Therefore, these two would be really good to use.

Lego MindStorms can be ruled out as each one is around £200 which is quite expensive seeing as each class has thirty children. The high price would also mean that children would not be able to go home and continue working on their projects.

MineCraft can also be crossed out for use in plugged activities for KS2 children for a similar reason as Lego MindStorms. It would be expensive and would need to be pre-installed onto school computers.

This leaves the Rasberry Pi and Makey Makey. The Rasberry Pi is more suitable for Secondary schools rather than Primary School children so this can be left out too.

Makey Makey can be used in conjunction with Scratch and it’s easy to use design makes it a suitable choice to try out in schools.

Therefore, from the original six examples now three remain (Scratch, CodeBug and Makey Makey). These three would be suitable to use for plugged activities in Primary Schools.

An example of a plugged activity could be coding a game within Scratch and then using Makey Makey to control the game.

1.4 An example of a successful activity
As seen there are many examples of unplugged and plugged activities. The Jam Sandwich activity will now be evaluated to measure its success by comparing against the requirements covered in section 1.2.

It is definitely ‘exciting’ as here the concept of algorithms is taught in a way that would keep the children engaged and enthused. Through the use of making a jam sandwich KS2 children are learning how algorithms work and how instructions should be formed. This aspect satisfies the ‘explains how Computing is a science’ requirement. The activity ‘provides a sense of achievement’ by reaching the end goal of having made a sandwich correctly. Both girls and boys are involved as this topic ‘caters to both genders.’ If other teachers or teaching assistants are in the room whilst this activity is taking place they might learn something about algorithms as well as the children which meets the ‘leaving a legacy’ requirement. It also ‘tailors to schools resources’, ‘targets all capabilities’ and ‘meets the curriculum’ by using cheap materials to portray a concept from the new curriculum to help children of different abilities understand algorithms.

Here eight out of fifteen requirements are satisfied and would be considered a successful activity. It is very difficult to meet all fifteen. It is important to meet as many of these throughout this project in order for Primary school children and teachers to take as much as they can away from the activities.
For completeness, an unsuccessful activity would not satisfy any of these requirements or just one or two. The children would be disengaged and would not learn a computing concept. They would be just ‘doing a task’ with no real purpose, goal or enthusiasm, for example, colouring in a picture of a computer in grey would be classed as an unsuccessful activity.
Chapter 2: Implementation

There is a range of computing topics that can be taught. To see a few of these see Figure 3. This was the original spider diagram that was made to note down a few concepts that could be implemented into activities suitable for KS2 children. Once these were written down the concepts that could be portrayed in an exciting manner to make activities successful were selected. After this, the development of worksheets for each activity could begin and testing in schools followed. This was the general implementation structure.

![Spider diagram of some computing topics](image)

Figure 3: A spider diagram of some computing topics

2.1 Development of the Star Switch activity

In section 1.3, Makey Makey was one of the three devices that was selected as a good choice of hardware to use in Primary schools for plugged activities. This led to the research of what exactly is possible with this hardware. On the Makey Makey website there is a range of lesson plans [13]. Most of these centre around teaching a topic such as Maths or Science. At first, this seemed like a good idea such as creating a Maths Quiz and using the Makey Makey as a control to select certain shapes or numbers. However, adding this extra element of education could make the activity seem boring. It would also steer away from the computing aspect of what is being portrayed therefore making it an unsuccessful activity and so the idea was not carried forward.

From watching YouTube videos [14] on projects that people have created using the Makey Makey, it was noticed that most of them were either a sound keyboard or a control for a game. Some school computers do not provide headphones for children to use within their computer clusters. As a result making a sound keyboard was left alone and making a game controller was looked into.

It is always good to start with the basics when learning something new. This is where a simple on and off switch was created to turn a circle on the screen to red when on and white when off to test out the functionality of the Makey Makey. The code was created within Scratch as previously mentioned in table 1 both are compatible with each other. The switch was created using some tin foil as from the research it was noticed that Makey Makey can be connected to anything conductive so foil seemed an obvious choice as it is cheap and easily accessible. The code was really simple as it consisted of only four lines. The process of making the switch was really enjoyable and very easy. This seemed a really good idea to try out in schools as children would see using foil to control something on a computer screen as exciting and enjoyable.
To make the activity more interesting a star shape was used instead of a circle. Pressing on using the foil switch turns the star to yellow and pressing off turns it to white. To summarise this activity the children make a switch using card and foil, connect this to Makey Makey, write the code within Scratch, connect the Makey Makey to the computer and then use the switch to control the star colour.

Initially, this activity seemed too short so a starter activity was created. During the time of development, Christmas was approaching and most children associate this time with enjoyable experiences, therefore, a Christmas-themed activity would be well received. It seemed a natural choice to make a game for this theme as children play games during this season so this was created within Scratch and to control the game the Makey Makey was used. This game required a foil switch, a present coloured in pencil (as graphite is also conductive) and a star covered in foil. After making all these components the children connect them to the Makey Makey which is then connected to the computer to see how it controls the game.

The Christmas themed game plus the Star Switch activity combine to form a lesson to try out in schools. This activity covers the following objective found in the new national curriculum [15]:

‘work with variables and various forms of input and output’

The children learn how to use Makey Makey (a peripheral) and investigate how it works with some simple code. For those who finish early, they could try changing the colour of the Star within Scratch, change the keys that control the Star or think of their own extras.

### 2.2 Development of the Maze game activity

This activity was inspired by the Jam Sandwich unplugged activity which was about teaching algorithms. As mentioned earlier in section 1.3 not many resources were needed for that activity yet it portrayed the concept really well and the children enjoyed themselves.

Instead of teaching the importance of precise and logical instructions by making a sandwich the Maze Game will teach this concept by going from A to B within a maze.

A paired unplugged activity was created to guide one player from start to finish. The activity will now be described.

![Figure 4: The maze game board required for the activity](image-url)
• Player 1 has a maze board (see Figure 4).
• Player 2 has the same maze board but it is blank (No icons on the board).
• Player 1 and Player 2 sit back to back.
• Player 1 instructs Player 2 where to place the arrow and which way to move the arrow in order to get to the X. (For kids with lower abilities they could have the set of instructions mixed up and they have to rearrange them first. For kids with higher abilities they could use the words ‘move’ ‘clockwise’ ‘anticlockwise’ ‘90 degrees’ etc.)
• Player 2 follows the instructions step by step.
• Player 2 reads out the coordinates of where they have placed the X.
• Check if it is a match.

Here the children would be learning about following clear logical instructions. This activity is really good if a school has limited resources although it was not tested out in schools as its unplugged nature may not be seen as ‘fun’ to young children. Therefore, this activity was left to the side and a plugged alternative was created to engage them.

Based on the pros and cons of plugged technologies covered in section 1.3 Scratch was chosen to help deliver a plugged version of the Maze game activity as it is free and part of the new curriculum.

Instead of guiding a person from A to B using pen and paper it was decided to implement this same concept within Scratch. Here the children would have to guide a cat icon from start to finish by coding the correct instructions needed for the algorithm.

The aim of the game was to move the cat using the keyboard arrow keys to eat all the fish, avoid all the dogs and then go to the water bowl. Notice the maze game board in the unplugged version is used here (see Figure 5). To make it suitable for all abilities obstacles were introduced. Eating a fish increases the score by one and touching a dog decreases the score by one. If the top score was obtained then the water bowl produced a message saying ‘Well done. You have got a top score!’ The cat also drew a path whilst it moved.
It is always good to develop continuations of concepts to help reinforce understanding. Therefore, another plugged activity was created and was inspired by some students in Kilburn playing a game called ‘Keep talking and nobody explodes’ [16] using the Oculus Rift where one player was guiding another player to reach the end goal of the game.

Using this concept the Oculus Rift and Minecraft (together known as Minecrift) [17] would be used to guide a player from a starting point to a goal (i.e. going from A to B). Player 2 would voice clear instructions to guide Player 1 through the virtual game as only they would know how to get to the end (similar to the unplugged back to back Maze game activity).

This would have been a really fun and good way of teaching algorithmic steps. However as previously mentioned in Table 1, there would be an issue with getting licences for the children as each license is £20 and getting these in schools would have been difficult as well as expensive. Not only this, the children would not be able to go home and try this activity themselves as not many children own an Oculus Rift. Therefore, this activity did not go ahead but the plugged Scratch version was a strong activity to try out in schools.

2.3 Development of the Crypto CodeBug activity

The two activities that have been covered so far are mainly targeted at children in Years 3 & 4 so this activity was created to target Years 5 & 6. CodeBug can be quite fiddly when connecting it to the computer and downloading programs onto it so this is why it was more suited to the older children of KS2. Also, the two activities previously discussed both use Scratch so children might be disappointed using Scratch again, therefore, the CodeBug was used to eliminate this disappointment. In section 1.3 it was decided that the CodeBug is a good hardware device to use within plugged activities as it can be accessed online for free. The CodeBug is really cheap to buy as it only costs £15 so it was a good choice to try out in schools.

Again a concept had to be chosen. To arrive at this concept the CodeBug’s capabilities were investigated. It is quite limited as it can only display text or graphics on its LED lights. A popular activity online was displaying names onto the CodeBug [18]. From using the earlier spider diagram, concepts were narrowed down to the ones that rely heavily on text, for example, binary and cryptography.

First, binary was looked into however the only numbers that would be displayed would be a combination of zeros and ones which did not seem too exciting. On the other hand, cryptography allowed the use of text and numbers when encrypting and decrypting. From comparing the two topics cryptography allowed more flexibility so this topic was chosen to teach in schools using the CodeBug.

To teach children about encryption and decryption using the CodeBug the children have to download a pre-made program to their CodeBugs which displays a code. Then they decrypt the code using a table to reveal a hidden message. After this, they encrypt their teacher’s name or their own names and display it on the CodeBug. Finally, the children send secret messages to each other.
2.4 Summary

Notice that all three activities portray different concepts. This is so each activity could be taught in the same school and could target different ages as well as different capabilities to help meet the requirements in section 1.2.

Whilst making the activities it was really difficult to create original, fun and suitable activities which portrayed a strong computing concept. It is important for children of different abilities to enjoy the activities, be able to complete them without too much guidance and learn something about Computing. Combining all these into one was tough.

Now that the concepts and the activities have been established the development of worksheets for the children can begin. This is much easier than proposing concepts and forming the activity ideas as all that has to be done is writing out the ideas down in logical steps for children to follow. Throughout the development of worksheets, the main question asked was ‘are the steps clear for young children?’ (See Appendix A, B and C for the worksheets for the three activities).
Chapter 3: Testing and redevelopment of activities

In section 1.1 it was mentioned that a conference was attended which highlighted the challenges that may arise throughout this project. This chapter will now discuss these as a range of issues were brought to attention whilst testing out the activities in Primary schools. Some were found during self-evaluation and some from feedback from teachers (both verbal and via email). The issues that arose led to the redevelopment of the activities in order to make them more successful. Then they were tested again in other schools.

3.1 Testing in schools

During the testing phase, the different activities were not tested one after the other due to the availability of schools, the schools resources and activity requirements. For example, the Crypto CodeBug activity needs access to certain folders and links. If the school did not have these then a different activity was tested. See Table 2 for the order testing occurred, the activities that were tested out in each school and the year group taught.

<table>
<thead>
<tr>
<th>Date</th>
<th>School name</th>
<th>Activity</th>
<th>Year group</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/12/2015</td>
<td>Thorp Primary</td>
<td>Star Switch (Christmas themed)</td>
<td>4</td>
</tr>
<tr>
<td>26/02/2016</td>
<td>Acacias Primary</td>
<td>Maze Game (debugging)</td>
<td>4</td>
</tr>
<tr>
<td>26/02/2016</td>
<td>Chapel Street Primary</td>
<td>Star Switch (had to write the code themselves)</td>
<td>4</td>
</tr>
<tr>
<td>02/03/2016</td>
<td>Mersey Vale Primary</td>
<td>Star Switch (code provided)</td>
<td>3</td>
</tr>
<tr>
<td>02/03/2016</td>
<td>Mersey Vale Primary</td>
<td>Crypto CodeBug</td>
<td>5</td>
</tr>
<tr>
<td>10/03/2016</td>
<td>St Paul’s CE Primary</td>
<td>Crypto CodeBug</td>
<td>5 &amp; 6</td>
</tr>
<tr>
<td>17/03/2016</td>
<td>Tithe barn primary</td>
<td>Crypto CodeBug</td>
<td>4 - 6</td>
</tr>
</tbody>
</table>

Table 2: Order testing occurred within schools (in descending order) with corresponding activity and year group

Due to nature of this testing method, general teaching techniques were picked up and were applied across all activities, for example, how to gain children’s attention to keep them focused and to address any common issues. The teacher at Acacias Primary stopped the children after certain periods of time asking whether they had finished a specific task and if anyone needed extra time. This helped manage the class much better and the technique was used in following activity sessions. Another general issue that was noticed was that younger children tend not to follow worksheets. This led to more demonstrations at the start of the activity to help the children visualise what they were aiming for and how to do this. The demos worked much better than just giving children sheets to follow themselves.

During each session, an introduction was given before the activity begun. This included what they were going to learn (i.e. the concept that the activity portrayed), an overview of the activity and any demonstrations of tasks that children might find difficult or need to know how to do. Then during the main activity, the children were observed on which particular task they were on, how easy they found that task and how engaged they were. Any questions that the children had were also answered if they were stuck. If common questions arose then it was best to stop the class and address the issue with a demonstration of what needed to be done. Extensions were given to those...
children who had finished the activity much earlier than the majority of the class. Once the entire
class had completed the activity, the class was stopped and a conclusion/wrap up was given. This is
where questions about the concept were asked to help judge if they understood the underlying
computing concept of the activity. Also, some examples of other uses and applications of the
concept were given to help them comprehend the scope of the technology used. In the end, it was
asked if the children enjoyed the activity to gauge how much fun they had. The majority of the class
put their hand up.

3.2 Testing of the Star Switch activity
From the worksheets for this activity in Appendix A it is clear that this activity is quite long. The
activity was tested at Thorp Primary during a two hour slot. However, this time period was not
enough to complete all the worksheets as well as including an introduction and conclusion. Also,
additional time should have been allocated to deal with any questions the children had. From this, it
was obvious that the worksheets would need to be cut down considerably seeing as normal Primary
school computing sessions last one hour. The theme was also Christmas themed and would not have
been applicable after the Christmas holidays so a general theme needed to be made.

Furthermore, at Thorp Primary they only had access to the web-based version of Scratch, although
the worksheets were based on the desktop version of Scratch. There is a slight difference between
the two so there was an issue locating certain components within Scratch. In the end, the children
managed to get the Christmas game working which they really enjoyed but they did not end up
finishing the Star Switch activity as the game took much longer than expected.

Overall, the children really enjoyed making the foil switch as well as the other components for the
Makey Makey and seeing how it worked with Scratch. Therefore, the underlying concept did not
need to be changed.

From this, it was decided to get rid of the Christmas themed game and just have the Star Switch
activity as it was shorter to complete. It was also adapted to be compatible with the web-based
version of Scratch. (See Appendix D for worksheet).

After the redevelopment of this activity, it was tested out at Chapel Street Primary. Again the
children really enjoyed the concept and making the components. However, the children had little
experience of Scratch so they struggled to build the four lines of code required for the Scratch
program needed for the Makey Makey to work. (See Figure 6 for the code required).

Also, the worksheets were based on the web-based version as an improvement from Thorp Primary
however Chapel Street did not have access to the online version of Scratch. Despite having this issue,
the children managed to complete the worksheet within the time frame and really enjoyed it. It was
decided to have two worksheets for this Star Switch activity: a desktop and web based version. (For
the desktop version worksheet see Appendix E).
The final test of this activity was done at Mersey Vale Primary with Year 3s. It was discovered that younger children do not like following worksheets so may not understand what to do next. From this more demonstrations were done during this session. Prior the session, it was confirmed that the school has access to the web version of Scratch via email with the teacher. Within this email, it was mentioned that the children had no experience of Scratch and coding. From the session at Chapel Street, it was evident that writing the four lines of code proved to be difficult when children had little experience of Scratch. Therefore, children who have no experience would have really struggled so the code was given to the Year 3s. (See Appendix F for the web based version of this activity with the code provided). These alterations worked very well, the children had no major issues and enjoyed the session.

The children were also able to fit in some extensions such as changing the colour of the star or using different keys on the Makey Makey.

3.3 Testing of the Maze Game activity
Due to the lack of experience of Scratch within schools (despite it being a part of the new curriculum at Thorp Primary) [19] meant that the Maze Game activity would be really difficult to teach within some schools and it would take the children a very long time to complete. Using this knowledge an adapted version of the Maze Game was created. Instead of getting the children to write out the entire code for the game themselves the game was provided to them with a few bugs in which they have to identify and fix. This introduced the concept of debugging. It also made the activity much shorter and suitable for the children. (See Appendix D for the worksheet).

This activity was tried out once at Acacias Primary with a group of thirty Year 4 children. Overall, the activity worked well as the children were able to follow the steps on the worksheets with no major issues. Although, the children with higher abilities finished much quicker than expected and unfortunately there were only a few extensions which they had completed so the teacher had to step in and give them a debugging activity available on the Scratch website.

3.4 Testing of the Crypto CodeBug activity
This activity was tried out three times in three different schools. The first two sessions went really well as the children had fun and the activity fitted within the hour session. There were no major issues as the activity requirements and the schools resources were checked prior to going into the schools via emails with the teachers.
In the final session, the activity was done with a group of seven of mixed ages as part of their after-school computing club. These children were of a very high ability and had a passion for coding as the session was optional. The children enjoyed working with the CodeBug and finished the activity much more quickly in comparison to the Crypto CodeBug sessions done in the other two schools. This meant more extensions were needed for these children.
Chapter 4: Conclusion

The redevelopment and testing of the three activities produced many versions. In this chapter, the final testing of each activity will be evaluated to determine its success by comparing them to the requirements in section 1.2.

All three activities ‘cater to both genders’, ‘target all capabilities’ and ‘target KS2 children’ as every boy and girl (aged 7 – 11) of mixed abilities participated in the sessions. Based on school limitations, activities were redeveloped to make sure that they run smoothly in schools. Therefore, the requirement of ‘tailoring to schools resources’ was met. The sessions were intended to inspire young children about computing so they could ask themselves if computing is for them which ‘introduces career options.’ ‘Meeting the curriculum’ was achieved by developing activities based on computing concepts currently being taught in schools.

The remaining requirements will now be evaluated as these are more specific to each activity.

4.1 Evaluation of the Star Switch activity

For reference, this will evaluate the activity worksheet in Appendix F. Overall the children really enjoyed this activity as they could make components using foil and card and then see how it worked with Makey Makey and Scratch. From Appendix H, it is obvious that the children are having fun and are excited about this activity by looking at the children’s faces. Therefore, the first requirement to ‘excite young children’ was met.

Through making a foil switch the children are able to see that this controls the colour of a star image on the computer screen. By explaining within the introduction that this is not done by magic but by using conductive materials and inputs the children can have a deeper understanding of how it all works. Therefore, this creativity aspect and level of understanding meet the requirements to ‘encourage creativity’ and to ‘explain how technology and computing concepts work’.

One of the main issues of the new curriculum that was discussed in the first chapter was that teachers do not have much computing knowledge so they may feel uncomfortable using new technologies. This was highlighted at a session at Chapel Street Primary. The school had invested into Makey Makey but had never used it. The session really taught the teacher more about the hardware device. This can be shown by her feedback received via email:

‘I found it very useful as Makey Makey is a new resource for me with computing and it was good to see how it can be used with a class of children.’

From this, it can be said that the Star Switch activity ‘left a legacy’ as now the teacher at Chapel Street feels more comfortable with the device and would be more likely to embed it in her activities to support the new computing curriculum.

From having met ten out of fifteen requirements this activity would be considered a successful activity. To improve, it could incorporate some sort of game in a similar way to the Christmas themed game to enthuse the children more.
4.2 Evaluation of the Maze Game activity
This activity incorporated a game and children really enjoy this type of activity. This can be shown by the feedback received from the teacher as she said:

‘I really enjoyed the activity and the children did too! … Please let me know if you would like to visit again.’

This clearly shows the requirement to ‘excite young children’ was achieved. It also allowed the children to be creative as can be seen from Figure 7.

This child added cars which would ‘freeze’ the score if touched. For example, if the cat hits a car the cat can no longer obtain a score for eating a fish. This was his idea which shows this activity really ‘encourages creativity’. For more examples of children’s work see Appendix I.

‘To explain how Computing is a Science’ was met through debugging within Scratch as the children could see that there is more to computing than just gadgets. The activity also showcased ‘the uses and applications of Computing’ which is another requirement. They could also add more to their work at home and show their parents as Scratch can be accessed online for free. By showing friends, family and teachers they can obtain a ‘sense of achievement’ through being creative within their work.

The feedback received from the teacher at Acacias Primary was really useful. She said:

‘The only thing I can think of is that the information sheet could be simplified slightly - maybe split playing / debugging prompt sheets.

In order to get children's feedback, you could ask questions on the sheet such as which keys did what?

Additionally, as a further challenge you could ask children to add their own bugs and ask another child to spot them.’
After this feedback was received, Code Club worksheets were researched as these were recommended by the teacher during the session [20]. They are suitable for KS2 children as they split up tasks clearly. In the future, the worksheet for this activity (in Appendix G) would be adapted to a similar style of the Code Club worksheets to help children follow the steps better as the teacher said the worksheets ‘could be simplified.’

Questions would be added to the worksheets to ask children which lines of code in Scratch performed what. For example, what does the statement ‘change score by 3’ do? The children, in this case, would need to recognise ‘score’ is a variable and it would increase the score by three.

For the higher ability children, the teacher’s suggestion of adding their own bugs and asking other children to spot them would be a good extension which would also be included.

Overall, based on all the feedback, pictures and comparison against requirements this activity went well and would be considered a successful activity. However, it would be modified using the suggestions discussed to make it even better. To increase its validity it would be tested again in a few more schools.

4.3 Evaluation of the Crypto CodeBug activity
This activity went well as two children from different schools asked how much a CodeBug is. This shows that children really enjoyed themselves and want to learn more. One pair displayed their enthusiasm for the activity using the CodeBug code which can be shown in Figure 8. This pair finished early so wrote ‘Daisy and Mariam are having fun’ which highlights that this activity was enjoyed by the children. This can also be demonstrated by another example of a pair who wrote ‘Thank you’ on their CodeBug. Therefore, the requirement to ‘excite young children’ was definitely met. For pictures which display the children’s enthusiasm and work see Appendix J.

![Figure 8: The code the children had to download to their CodeBugs to display a message. This pair wrote 'Daisy and Mariam are having fun'](image)

Within this activity, children sent encrypted messages to each other which allowed children to be creative with the messages they sent. This shows that computing is more than just the internet and phones through the exploration of cryptography. From this, it can be said that the requirements to ‘encourage creativity’ and ‘introduce the uses and applications of Computing’ were achieved.
This activity went well during the first two sessions however when tested with a group of higher ability children at an after-school coding club the activity needed more extensions. This shows that the success of the activity is also determined by the age group and abilities of the children. To cater for the higher ability children, the worksheets for this activity (Appendix C) would have extras such as including a shift key when encrypting and decrypting so instead of A = 1, A = 3. This would show the children that messages can be made more secure. Then they could invent their own code by using symbols instead of using numbers.

In conclusion, this activity was successful when tested with classes of mixed abilities but not as well with children of higher abilities. To make it more successful these extras would be applied and tested out again in schools.

4.4 Future work

Each evaluation section within this chapter highlighted the further improvements that would be made for each activity. These would be applied and then tested out in schools. Based on feedback and self-evaluation the activities would be redeveloped to make them more successful.

Another thing that could be done is developing a continuation of the activities, for example, it would be nice to see how cryptography (that was used in the Crypto CodeBug activity) can be integrated with other technologies such as the Makey Makey or other devices to help children understand that concepts can be applied in different scenarios.

As well as improving these three activities, it would be good to develop a range of completely new activities that portray different computing concepts. One idea could be creating an activity that helps children understand sorting. This could be done through an unplugged activity. Some children would stand in a line whilst holding up a number. Then the other children would give instructions to help sort the children in order who would move around based on these instructions. Then as a continuation, the children could create the sorting algorithm within Scratch as a plugged activity to help reinforce their understanding. Writing a sorting algorithm at aged 7 – 11 would be quite difficult, therefore, some guidance would need to be provided such as giving the children sections of the code for them to fill in.

For those who are looking at developing activities aimed at KS2 children, it would be beneficial to explore different technologies that have not been mentioned within this report such as Lego WeDo [21]. Chapel Street Primary had bought this product to use within their computing classes but they had not used it yet. This suggests that there must be a range of devices that schools have invested in but have not used. It might be a good idea to identify these devices and develop activities based on them to try out in schools. Another activity could use iPads. They are easy to use and many schools have invested into them as well. This would help teachers become more comfortable with using such devices within their computing sessions.
4.5 Summary

With this project, it is quite difficult to measure the success of the activities as it is not as simple as running some code and seeing if it works. The measurement of success comes from testing the activities within schools, self-evaluation and feedback from teachers. For the full feedback received via email by teachers see Appendix K.

From asking questions in the ‘wrap-up’ part of the sessions children really understood the underlying concepts being portrayed in all three activities which shows the overall success. Activities are always continuously improving. One activity may work in one school but the same one may not work in a different school due to the ability of the children and the resources schools have. This emphasises that there is always room for development to tailor to specific needs.

Through meeting the majority of the requirements for each activity the objectives in section 1.2 were satisfied. Therefore, through developing, testing and improving fun activities the project supported the main aim which was to ‘enthuse the next generation’ whilst targeting the new curriculum. This is evident from all the feedback received by the teachers in schools.
Appendix A

Objective: To gain a basic understanding of Scratch and Makey Makey.

Task 1: Setting up

You need to allocate tasks so we need:

- One person to make the switch board
- One person to make the switches
- One person to make the star
- One person to make the present
- One person to be the collector. If you only have 4 people in your group someone will have to do this role as well.

Follow the steps on the separate piece of paper to make everything.

Task 2: Connecting to Makey Makey

1) Connect the red USB cable to the Makey Makey board as shown in the picture in step 2.
2) Take an alligator clip and attach this to Earth like this:
3) Take the switch you made and turn it over.
4) Attach an alligator clip to the top and one to the bottom, for example:

5) Flip the switch back over.
6) Take the alligator clip you attached to the on switch and connect it to the up key.
7) Now do the same with the off switch by attaching it to the down key. You should have something like this:

8) Take the star and attach an alligator clip to it. Attach the other end to the right arrow key.
9) Take the present and attach an alligator clip to it. Attach the other end to space. You should have something that looks like this:
Task 3: Using Makey Makey with Scratch

1) **Open** Scratch
2) Click file -> Open and open the file **Christmas.sb**
3) **Connect** the other end of the USB cable for Makey Makey into the computer. You should see a light turn on the Makey Makey board.
4) Test your switch out by holding the **yellow** earth alligator clip and **pressing** the on switch. Rudolph’s nose should turn red. Test out the off switch and notice what happens.
5) Click on the **green flag** to start the program.
6) **Follow** the instructions on the screen and have fun!
7) **Take it in turns** to play the game.

When you have finished put your hand up and let the teacher know.
(Separate sheet)

**SWITCH BOARD**

1) Cut out a rectangle from the orange card.
2) Fold the rectangle in half so you have something that looks like this:

```
  __________
  |        |
  |        |
  |________|
```

3) Cut along the dotted lines. Make sure you don’t cut all the way.
4) Unfold the card. You should have two slits in your rectangle. You should have something that looks like this:

![Image of a switch with ON and OFF labels]

5) Give this to your collector.

**SWITCHES**

To do this step you need to make sure your switch has been made.

1) Cut out two smaller rectangles from the pink card. Make sure they fit into the slits of your switch.
2) Cover these small rectangles using foil.
3) Slot them into your switch. You might need to bend them at the back so they stay on the table. You should have something that looks like this:

![Image of a switch with foil-covered rectangles]

4) We now have a finished switch which we will use later. Give this to your collector.
STAR

1) Cut out a star from the yellow card.
2) Cover this all in foil. You should have something like this:

![Star in Foil]

3) Give the star to your collector.

PRESENT

1) Pick up a blue present and colour it in using pencil.
2) Cut the present out and leave to the side. You should have something like this:

![Present Sketch]

3) Give to your collector.

COLLECTOR

On the switch, star and present stick some blu tack onto the back and stick them to the table.
Task 4: Create your own Scratch program

Now you are going to make your first Scratch program on your own.

We are going to make a star turn on and off like we did for Rudolph’s nose. First we need to find a star and upload it to Scratch.

1) Go to Google Images on your browser
2) Search for a star image
3) Find one that is white with a black outline
4) Click on the image
5) Right click on the image
6) Click save image as
7) Save this into your work folder
8) Open Scratch (if not already open)
9) Click File -> New
10) Delete the cat sprite. By right clicking on the cat in the box and clicking delete as is shown by:

![Image of Scratch interface with the cat sprite deleted]

11) Click on the paint brush next to New sprite
12) Click import
13) Find the star image you saved before
14) Click OK and then OK again

You should now see a big white star on your screen. Now we are going to make a yellow star.
15) Click on the **Costumes tab**

16) Change **costume1 to white star**

17) Click **copy**

18) Change **white star1 to yellow star**

19) Click **edit**

20) Click on the bucket and select a yellow colour. Then click into the star.

21) Click OK

We now we have a yellow star. We need to write some code that changes the star to white and yellow when we press some keys. Before we do please make sure you **save your work Star**!

22) After you have saved your work. Click on the **Scripts tab**

23) We will now be using the following tabs:

24) Click on **Control**

25) Click and drag the **flag** statement to the middle.

26) Click on the **Looks tab**

27) Drag and drop the following the **switch to costume** statement so you have this:

28) We need to end the script so click on the **Controls tab** and then **stop scripts**. You should have:
Test this out by clicking on the green flag at the top and notice what happens!

Now we need to write some code to make the star switch to yellow.

29) Drag and drop the following statements:

Press the **up key** on your keyboard and see what happens. Now let’s make the down key.

30) Drag and drop the following statements.

Press the **down key** and see what happens.

Now let’s run this all.

31) Click the green flag. Press up and then down and notice the star turning on and off! You have made your very own Scratch program. Remember to save you work! Why don’t you add some extras?

**Task 5: Something to think about**

1) How would you use the Makey Makey to switch the star on and off?

……………………………………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………………………………
……………………………………………………………………………………………………………………………………………………………………
Appendix B

Objective: To solve problems such as a maze game using a Scratch.

In this activity you will make the cat sprite eat all the fish, avoid the dogs and then reach its water bowl!

Task 1

On the maze below draw a path from the cat to the fish. You can’t go diagonal!

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Task 2

Is this path the quickest path? Does it take the least amount of steps? If this isn’t the shortest path draw the shortest path from the cat to the fish in a different colour on the maze above.

Task 3

There are now some dogs in the way which we want to avoid. Draw a path from the cat to the fish and think about the quickest way to get there.
Task 4: Try it in Scratch!

Setting up the maze game

1) Upload the background file by clicking on: Stage -> Backgrounds -> Import -> Find the maze.jpg file -> Click open
2) Place the cat sprite in A1.
3) Resize the cat so the cat fits in A1 properly using the grow and shrink button at the top.

4) We want to call this sprite ‘Cat’. So click on the cat sprite in the bottom right corner. You should see a box around the cat:

At the top where it says Sprite1 change it to Cat so it looks like this:

Notice the name of the sprite has also changed in the bottom left under the cat sprite. Click on the Scripts tab.

5) Draw a fish for the cat sprite. Use the paint brush button to ‘Paint your own’ sprite to do this.

6) Move this new sprite you’ve created into F4. Make sure it fits by using the grow and shrink buttons.
7) Rename this fish to Fish1 like we did for the cat.
8) Create another sprite of a bowl of water and place this in G5. Similarly, make sure it fits into the box and name it Home. You should have something that looks like this:
Well done! You have created the maze!

Let’s save this! File -> Save as -> Type in Maze game -> Save

**Making the cat sprite move**

9) Click on the cat sprite. You should see a box around the cat.

We need to make the controls to make the cat move. So let’s make the cat move forward. Drag and drop the following statements and fill in the missing gaps.

10) Test your keys out and see what happens. The cat sprite isn’t moving as we want it because it isn’t facing the right direction. Let’s make the cat sprite rotate.

11) Place the cat sprite back at A1.

12) Drag and drop the following statements and test the keys again.

Let’s save what we’ve done so far: File -> Save

**Resetting the cat to the start**

We want to make the cat to go to box A1 every time we start the game. This is called *initialisation*.

13) Drag and drop the statements below and fill in the missing gaps. You need to make sure the coordinates are for the A1 box.
**Hint:** To get the x and y coordinate values, move the mouse into the A1 square and look at the values in the bottom right hand corner.

![Coordinate Diagram](image)

Remember to save your work!

14) Press the space bar and notice what happens. If your cat is not in the A1 box change the x and y values in step 13 again.

**Tracking movement**

We want the cat to draw a path every time it moves so we can track where it is going.

15) Insert the green statements to the space key script.

![Script Diagram](image)

When we run the program we want to make the cat draw and clear any old paths.

**Before we finish the game**

Let’s make sure the cat is always in front of the fish all the time.

16) Insert the **go to front** statement under the **Looks** tab. Place this in between **when space key pressed** and **pen down**.

**Trying it all**

Press space to start the program. Use the keys you’ve created to make the cat go to the fish and then home. Save your work!

Well done! You’ve have created the maze game!
Extras

Why don’t you add more fish?

Just create new sprites like we did in step 5. Or you can duplicate a fish by right clicking on it. Remember to change the name of the fish to Fish2, Fish3 etc.

Making the fish disappear

We can make the fish disappear every time the cat ‘eats’ it. We will use Sensing.

1) Click on the fish.
2) Drag and drop the following statements.
3) Do this for all your fish.
4) Press space to start the program and move the cat to the fish and see what happens.

Introducing obstacles

Let’s draw some dogs that we want to avoid.

1) Using the Paint new sprite button draw a dog.
2) Place it in a box.
3) Resize the dog so it fits in the box
4) Change the name of the dog from Sprite 1 to Dog1.
5) Right click the dog and click duplicate and place in a different box.
6) Rename this dog sprite to Dog2.

Making the dog bark

We are now going to introduce some sound. When the cat goes to the dog the dog will bark.

1) Click on a dog sprite. Drag and drop the following statements.
2) You won’t be able to select Dog1 until you import the sound. Click on the Sound tab -> Import -> Animal -> Dog1 -> OK.
3) Click back on the **Scripts** tab and for the **play sound statement** click **Dog1**.
4) Do these steps for all your dogs.

**Keeping score**

We want to keep a score. If the cat eats a fish we increase the score by 1. If the cat hits a dog we decrease the score by 1.

First we need to set the score to 0 and we will use a **variable** called **Score**.

1) Click on the cat sprite.
2) Click on the **Variables** tab.
3) Click on **Make a variable**
4) Type in **Score**.
5) Make sure the **For all sprites** box is ticked

6) Press OK.
7) Change your code so it looks like this using your x and y coordinate values from earlier.

8) Click on the fish and do this for all your fish:
9) We want to lose a point if you reach a dog. Click on the dog and do this for all your dogs:

![Diagram showing when space key pressed, wait until touching Cat ?, play sound Meow, if Score = 2, say Well done! You've got a top score, wait 3 secs, stop script]

**Final steps**

1) Click on the water bowl (home sprite)
2) Drag and drop the following statements. (Remember to import sound like we did the dog.

![Diagram showing when space key pressed, wait until touching Cat ?, play sound Meow, if Score = 2, say Well done! You've got a top score, wait 3 secs, stop script]

3) Here I have 2 fish so this is why I asked if Score is equal to 2. If you have 5 but 5 in the if statement instead.

**Add your own extra features!**

**Task 5**

**Questions**

a) What does the ‘point in direction’ do in step 15?

How do you change the colour of the pen? Try it out and write down how you did it.

Which statement do we use if we want to turn a sprite?

b) What does the statement clear do?

c) If the cat is in A1 and the fish is in F4, what is the shortest path from the cat to the fish?

d) What does the ‘wait 3 secs’ statement do in final steps?
Appendix C

Objective: To use CodeBug to understand encryption and decryption.

Task 1: Setting up

1. Open the following link on your browser:
   http://www.codebug.org.uk/explore/codebug/6118/encryption/
2. Have a look at the code and try to understand what is happening. (Don’t worry if it doesn’t make sense).
3. Press the green play button and then click the B button. Notice what happens.

Task 2: Using CodeBug (Decryption)

4. Take the black USB cable and plug it into the computer.
5. Now hold down button A on the CodeBug whilst plugging the other end of the USB cable into the CodeBug. You should see a red light flashing (you can let go of A now).
6. Click the big Download button from the CodeBug website and the program should start downloading.
7. When it has finished locate the file that has been downloaded. It should be in your download folder on your computer.
8. Drag and drop this file into the CodeBug drive.
9. Now press button B on your CodeBug to see it running.
10. The CodeBug is now displaying a code. Write out the code here:

11. The code is a secret message. Work out the message using the table below. This is called decryption. What does the message say?
    Example:
    8 1 20 -> Hat (decryption)

    1  2  3  4  5  6  7  8  9  10
   |---|---|---|---|---|---|---|---|---|---|
  A | B | C | D | E | F | G | H | I | J |
  ---|---|---|---|---|---|---|---|---|---|
  11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
  K | L | M | N | O | P | Q | R | S | T |
  ---|---|---|---|---|---|---|---|---|---|
  21 | 22 | 23 | 24 | 25 | 26 |
  U | V | W | X | Y | Z |
Task 3: Using CodeBug (Encryption)

Encryption is the opposite of decryption. Example: Hat -> 8 1 20 (encryption)

12. Now use the table again to write your teacher’s name in numbers. Write the output here:

........................................................................................................................................................................

13. Go back to the website and click on the button Remix
14. Add your name to the code where it has a list of numbers.
15. On the website press the green play button and then click the B button. You should see your new code.
16. Unplug your CodeBug from the USB cable.
17. Go to step 5 and follow through to 9 to redownload the program.

Task 4: Create your own

Now in your pairs, one person will write a secret message and send it to the other person. When the other person receives the coded message they will decrypt the message to read what it says. Make sure the other person can’t see the original message.

18. Write out a message in words that you want to send to your friend (don’t let them see this):

........................................................................................................................................................................

19. Now encrypt the message using the table above by writing the message down in numbers:

........................................................................................................................................................................

20. Make CodeBug display this secret code of numbers.
21. Give the CodeBug to your friend and get them to decrypt the code. Did they figure out the original message?
22. Ask them to reply back on CodeBug using encryption.
Appendix D
(Version 2 of the Star Activity – web based)

**Objective:** To gain a basic understanding of Scratch and Makey Makey.

**Task 1: Setting up**

You need to allocate tasks so we need:

- One person to make the **on switch**
- One person to make the **off switch**
- One person to make the **stickers**

**PERSON 1: ON SWITCH**

1) Cut out a rectangle from the card.
2) Cover it using the foil.
3) Give this to the person making the stickers.

**PERSON 2: OFF SWITCH**

1) Cut out a rectangle from the card.
2) Cover it using the foil.
3) Give this to the person making the stickers.

**PERSON 3: STICKERS**

1) On one sticker write the word ‘ON’
2) On another sticker write the word ‘OFF’
3) Wait to get 2 rectangles covered in foil.
4) Stick the ‘ON’ sticker on **one** of the rectangles at the **top**.
5) Stick ‘OFF’ on the other rectangle at the **top**.

You should now have two switches that look like this:
Task 2: Connecting to Makey Makey

1) Connect the red USB cable to the Makey Makey board as shown in the picture in step 2.
2) Take an alligator clip and attach this to Earth like this:
3) Take your on switch and attach an alligator clip to the bottom, for example:

![Image of on switch with alligator clip](image1.png)

4) Now take the other end of the alligator clip you attached to the on switch and attach it to the up key on the Makey Makey, for example:

![Image of up key with alligator clip](image2.png)

5) Now take the off switch and add an alligator clip to it like this:

![Image of off switch with alligator clip](image3.png)
6) Now take the other end of the alligator clip you attached to the off switch and attach it to the **down key** on the Makey Makey, for example:

![Makey Makey setup](image1)

We have finished setting up the Makey Makey. You should have something like this:

![Makey Makey setup](image2)

When you have finished put your hand up and let the teacher know.
Task 3: Create your own Scratch program

Now you are going to make your first Scratch program on your own.

1) Open your web browser and go to the Scratch website. Log into your Scratch account. If you don’t have an account go to step 2.
2) Click on Create at the top
3) Delete the cat sprite. By right clicking on the cat in the box and clicking delete as is shown by:

4) Click on the folder

5) Find the star.png image from your shared drive and press Open

You should now see a big white star on your screen. Now we are going to make a yellow star.

6) Click on the Costumes tab

7) Change the name of the image from star to white star so you have:
8) Right click the white star and click **duplicate**, as is shown by:

![Duplicate Star](image1.png)

9) Change the name from **white star2** to **yellow star**, so you have this:

![Costumes](image2.png)

10) Click on the **bucket** button:

![Bucket Button](image3.png)

11) Click on the colour **yellow** and then **click** into the star. We have now created a yellow star.

We need to write some code that changes the star to white and yellow when we press some keys.
12) Click on the **Scripts tab**

![Scripts tab image]

We will now be using the following tabs:

![Evento and Control blocks]

13) Find this statement in **Events** and drag it to the middle of the grey space:

![when space key pressed]

14) Change **space** to **up arrow** in the drop down.

15) Now from the **Looks tab** add the **switch costume** statement. Make sure it says **yellow star**. You should have something like this:

![when up arrow key pressed switch costume to yellow star]

Now let’s make the down key.

16) Drag and drop the following statements using the **Events** and **Looks** tab. Here make sure it says **down key** and **white star**.

![when down arrow key pressed switch costume to white star]

17) Press the **down key** and see what happens. It should turn white.

Now let’s run this all.

18) Press up and then down and notice the star turning on and off! You have made your very own Scratch program.
Task 4: Using Makey Makey with Scratch

1) Get your Makey Makey switch that you made.
2) **Connect** the other end of the USB cable for Makey Makey into the computer. You should see a light turn on the Makey Makey board.
3) **Remember if you're holding the earth metal clip only you can control what happens on the screen.** Test your switch out by holding the **earth** alligator clip and press the **on** switch. The star should turn yellow. Test out the off switch and notice what happens.
4) Let the other people in your group have a go with the Makey Makey.
Appendix E

(Desktop version of the Star Switch activity. Note only tasks 3 and 4 are different from the web based version in Appendix D).

Task 3: Create your own Scratch program

We are going to make a star turn on and off. First we need to find the star and upload it to Scratch.

1) Open the Scratch program from your desktop
2) Delete the cat sprite. By right clicking on the cat in the box and clicking delete as is shown by:

3) Click on the paint brush next to New sprite
4) Click import
5) Find the star image from your class shared folder
6) Click OK and then OK again

You should now see a big white star on your screen. Now we are going to make a yellow star.

7) Click on the Costumes tab
8) Change costume1 to white star
9) Click copy
10) Change white star1 to yellow star
11) Click edit on the yellow star costume
12) Click on the bucket found here:

13) Select a yellow colour. Then click into the star. You should now have a yellow star.
14) Click OK

We need to write some code that changes the star to white and yellow when we press some keys. Before we do please make sure you save your work as Star!

15) After you have saved your work. Click on the Scripts tab

16) We will now be using the following tabs:

Now we need to write some code to make the star switch to yellow.

17) Find this statement in Control and drag it to the middle of the grey space:

18) Change space to up arrow in the drop down.
19) Now from the Looks tab add the switch costume statement. Make sure it says yellow star. You should have something like this:
Now let’s make the down key.

20) Drag and drop the following statements using the Control and Looks tab. Here make sure it says down key and white star.

![](image.png)

21) Press the down key and see what happens. It should turn white.

Now let’s run this all.

22) Press up and then down and notice the star turning on and off! You have made your very own Scratch program. Remember to save your work!

**Task 4: Using Makey Makey with Scratch**

1) Get your Makey Makey switch that you made.
2) Connect the other end of the USB cable for Makey Makey into the computer. You should see a light turn on the Makey Makey board.
3) Remember if you’re holding the earth metal clip only you can control what happens on the screen. Test your switch out by holding the earth alligator clip and press the on switch. The star should turn yellow. Test out the off switch and notice what happens.
4) Let the other people in your group have a go with the Makey Makey.
Appendix F

Objective: To gain a basic understanding of Scratch and Makey Makey.

Task 1: Setting up

You need to allocate tasks so we need:

- One person to make the on switch
- One person to make the off switch
- One person to make the stickers
- The rest of the team will connect the switches to the Makey Makey

PERSON 1: ON SWITCH

1) Cut out a rectangle from the card.
2) Cover it using the foil.
3) Give this to the person making the stickers.

PERSON 2: OFF SWITCH

1) Cut out a rectangle from the card.
2) Cover it using the foil.
3) Give this to the person making the stickers.

PERSON 3: STICKERS

1) On one sticker write the word ‘ON’
2) On another sticker write the word ‘OFF’
3) Wait to get 2 rectangles covered in foil.
4) Stick the ‘ON’ sticker on one of the rectangles at the top.
5) Stick ‘OFF’ on the other rectangle at the top.

You should now have two switches that look like this:
REST OF GROUP: Makey Makey team

1) Take out the red USB cable, the Makey Makey board and 3 alligator clips
2) When your team has finished making the switches put your hand up

Task 2: Using Scratch

1) Open up the link: https://scratch.mit.edu/users/sainia2/
2) Under shared projects click view all
3) Click on Star
4) Click see inside
5) Have a look at the code to try to understand what will happen when you press the up and down key.

Task 3: Using Makey Makey with Scratch

1) Get your Makey Makey switch that you made.
2) Connect the other end of the USB cable for Makey Makey into the computer. You should see a light turn on the Makey Makey board.
3) Remember if you’re holding the earth metal clip only you can control what happens on the screen. Test your switch out by holding the earth alligator clip and press the on switch. The star should turn yellow. Test out the off switch and notice what happens.
4) Let the other people in your group have a go with the Makey Makey.
Appendix G

Objective: To debug a maze game using Scratch.

In this activity you will make the cat sprite eat all the fish, avoid the dogs and then reach its water bowl! Then we will fix a broken version of the game and you can add some of your own extras.

Task 1: Play the game

1. Open up this link: https://scratch.mit.edu/users/sainia2/
2. Under shared projects click view all
3. Click on Maze Game
4. Click see inside
5. Have a look at the code (don’t worry if you don’t understand it yet). Click on the different sprites to can see more code. The sprites are:

   ![Sprites](Sprites.png)

6. Click on the green flag to start the game
7. Use the up, down, left and right keys on your keyboard to move the cat sprite. Use space bar to rotate the cat. Try to get all the fish and avoid the dogs. When you have done this make the cat go to its water bowl.
8. Press the green flag again to reset the program.
9. Play the game again and go to the dog only and notice what happens to the score in the top left.
10. Have a play with the game to see how it works.
11. Have another look at the code. Do you understand the code a little bit better now?

Task 2: Debugging

12. Go back to the link: https://scratch.mit.edu/users/sainia2/
13. Click on Maze Game Debugging. (Debugging means fixing errors).
14. Click see inside.

Whilst someone was writing the code they made a few mistakes and they want you to find and fix them. Remember more code is shown if you click on the different sprites.

Remember: To reset the program click the green flag button.

15. Click on the green flag and notice you can’t move. Only space works. Try to fix this by adding suitable numbers to the lines of code that make the cat sprite move. **Hint:** Try using 200 for the right key and -200 for the left key. Are these numbers good to use? Pick better ones if there are not good numbers to use.
16. **Reset** the program. Notice the score at the top left is set to 10. Can you find the line of code where score is set? Fix this line of code so it is *set to 0* every time we start the game.

17. **Reset** the program. (The score should now say 0).

18. Make your cat sprite eat only **one fish** and look at the score.

19. Change the code so when a fish is eaten the score **changes by 1** not 3. (Do this for all the fish sprites).

20. Do you remember when the player had eaten both fishes and avoided the dogs the water bowl displayed a message? Make the water bowl say the message: Well done you have a top score.

21. Test the game out to make sure there are no more errors.

   You now have a complete working program! Well done.

**Task 3: Add some extras?**

Try adding some extras:

- Add more fish to the game
- Add more dogs
- Change the pen colour to your favourite colour
- Change the pen size to make the line thicker
- Think of your own
Appendix H

Christmas themed Star switch activity

(Thorp Primary – Year 4 – aged 8 & 9)
Final Star switch version:

(Mersey Vale Primary - Year 3 – aged 7 & 8)
Appendix I

Maze Game activity

(Acacias Primary – Year 4 – aged 8 & 9)
Appendix J

Crypto CodeBug activity

Task 1: Setting up
1. Open the following link on your browser:
   [https://www.codebug.co.uk/hack/codecbug218/encryption/](https://www.codebug.co.uk/hack/codecbug218/encryption/)
2. Have a look at the code and try to understand what is happening. (Don’t worry if it doesn’t make sense)
3. Press the green play button and then click the B button. Notice what happens.

Task 2: Using CodeBug (Encryption)
4. Take the black USB cable and plug it into the computer.
5. Now hold down button A on the CodeBug whilst plugging the other end of the USB cable into the CodeBug. You should see a red light flashing (you can let go of A now).
6. Click the big Download button from the CodeBug website and the program should start downloading.
7. When it has finished locate the file that has been downloaded. It should be in your downloaded folder on your computer.
8. Drag and drop this file onto the CodeBug drive.
9. Now press button B on your CodeBug to see it running.
10. The CodeBug is now displaying a code. Write out the code here:

   ![CodeBug Display](image1)

   (St Paul’s CE Primary – Mixed class of Year 5 & 6 – aged 9, 10 & 11)
   (Tithe Barn School – Coding club – aged 7 -11)
   (Mersey Vale – Year 5 – aged 9 & 10)
Appendix K

Star Switch activity feedback:

‘12 Children went away and had a play on Scratch last night. They have been inspired by you.’ – Paul Bottemley, Thorp primary.

‘Thank you again for today. The children thoroughly enjoyed the session, as did all of the adults. I spoke to the children after your visit for their thoughts on what they had done and these are a few of the things that they said:

'It was very, very fun.'

'It was exciting and creative.'

'It was difficult, but with the help of Miss Saini I was able to do it and really enjoyed it.'

I also spoke to them about whether they would have a look at Scratch themselves now that they have done some work with it and the majority of the class said that they would.

I found it very useful as Makey Makey is a new resource for me with computing and it was good to see how it can be used with a class of children. I would love you to come in and do some more work with Makey Makey with the children but understand that this is probably not possible. However, any suggested activities that I could attempt with my class would be gratefully received.

Many thanks again for a great session. Thank you for your hard work. The children asked if I would say a big thank you too.’ - Claire Atherton, Chapel Street Primary

Maze Game feedback:

'I really enjoyed the activity and the children did too! I think the information sheets you provided were great.

The only thing I can think of is that the information sheet could be simplified slightly - maybe split playing / debugging prompt sheets.

In order to get children’s feedback, you could ask questions on the sheet such as which keys did what?

Additionally, as a further challenge you could ask children to add their own bugs and ask another child to spot them.

Please let me know if you would like to visit again.’ – Sally Jordan, Acacias Primary.
Crypto CodeBug feedback:

‘Thanks again for a fantastic afternoon. Your sessions looked/sounded like great fun and gave the children a really exciting alternative to their usual computing sessions. I’d be more than happy to get some feedback from pupils and teachers for you, but as far as I’m concerned I’ve been really impressed by how professional and well-organised you’ve been, and by how relevant and well-pitched your sessions seemed to be. We’re really grateful to you for coming and sharing your skills!

A few photos have been posted to our school’s Twitter page which you can find here: @MerseyVale

Thanks again’ – Matt Ellis, Mersey Vale Primary
References

(Accessed: 19th April 2016)

[2] Computing at School - Homepage
https://www.computingatschool.org.uk/
(Accessed: 19th April 2016)

(Accessed: 19th April 2016)


Available at: http://csunplugged.org/binary-numbers/
(Accessed: 19th April 2016)

Available at: https://www.youtube.com/watch?v=leBEFaVHlIE
(Accessed: 19th April 2016)

Available at: https://www.raspberrypi.org/
(Accessed: 19th April 2016)

Available at: http://www.codebug.org.uk/
(Accessed: 19th April 2016)

Available at: http://makeymakey.com/
(Accessed: 19th April 2016)

Available at: http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com
(Accessed: 19th April 2016)

Available at: https://minecraft.net/
(Accessed: 19th April 2016)

Available at: https://scratch.mit.edu/
(Accessed: 19th April 2016)

Available at: http://makeymakey.com/lessons/
(Accessed: 19th April 2016)

[14] YouTube - MaKey MaKey - An Invention Kit for Everyone.
Available at: https://www.youtube.com/watch?v=rfQqh7ICcOU
(Accessed: 19th April 2016)
   (Accessed: 19th April 2016)

   Available at: http://www.keeptalkinggame.com/
   (Accessed: 19th April 2016)

[17] Minecrift - Apps - Oculus Share
   Available at: https://share.oculus.com/app/minecrift
   (Accessed: 19th April 2016)

[18] CodeBug – Activity Scrolling Name Badge
   Available at: http://www.codebug.org.uk/learn/activity/57/scrolling-name-badge/
   (Accessed: 19th April 2016)

   (Accessed: 19th April 2016)

[20] Code Club Projects Site - Scratch projects
   Available at: https://www.codeclubprojects.org/en-GB/scratch/
   (Accessed: 19th April 2016)

[21] Lego WeDo - Homepage
   Available at: https://education.lego.com/en-gb/lesi/elementary/lego-education-wedo
   (Accessed: 19th April 2016)