Buckingham Park
Church of England Primary School
Excellence, through God who strengthens us

Jubilee Square, Buckingham Park.
Aylesbury, Buckinghamshire. HP19 9DZ

Email: office@buckinghampark.org
Web: www. buckinghampark.org

School Policy Document

Computing Policy

Date Adopted by Full Governing board:	
Last reviewed on:	01/07/2021
Next review due by:	Click here to enter a date.

1. BACKGROUND

1.1. At Buckingham Park Church of England Primary School it is important that every member of the school community feels valued and respected, and that each person is treated fairly and well. We are a caring community, built on a clear Christian foundation and rooted in Christian values. We aim to provide the highest quality all round education, for each and every child, in partnership with parents, within the context of a Christian community. In short, 'Excellence, through God who strengthens us'.

1.2. All school policies are therefore designed to support the way in which all members of the school can live and work together in a supportive way. It aims to promote an environment where everyone feels happy, safe and secure.

1.3. The school has a set of values that are based on Jesus' Sermon on the Mount. These are a means of promoting good relationships, so that people can work together with the common purpose of helping everyone to learn. These values are displayed below and permeate everything we do:

We are kind, helpful and polite
We do our best
We are honest
We share
We are peacemakers
We forgive others
We take care of everything, and everyone

2. INTENT

At Buckingham Park we want our children to be an exceptional computer whizz. Our intent is to facilitate and promote the skills and knowledge children need to reach their potential in school and beyond. We aspire is to create motivated 'life-long' learners with Computing to enhance and extend teaching and learning across the whole curriculum. Therefore, we want to model and teach our pupils on how to use technology positively, responsibly and safely. The overall aim of our computer scheme of work - which is Switched On Computing - is that pupils leave primary school as confident, capable and creative users of digital technology, with a secure understanding of the fundamental principles of computer science and as safe, responsible and discerning digital citizens.

The aims

1. Meeting the requirements of the Foundation Stage Curriculum, KS1 and KS2 National Curriculum.
2. Ensuring all children know how to stay safe online (see separate **Online Safety Policy** for details).
3. Developing children's individual Computing capability and understanding.
4. Developing Computing as a tool for learning and investigation.
5. Innovative use of resources and stimulating interest in new technologies.
6. Enhancing teaching and learning in other areas of the curriculum by cross-curricular use of Computing.
7. Equipping pupils with the confidence and capability to use Computing throughout their education, home and further work life.
8. Ensuring children, parents, staff, governors and the wider community have relevant and meaningful experiences of Computing.

9. Children having a growing awareness of how Computing is used in the world around them and of the benefits that it provides.

Special Educational Needs Disability (SEND) / Pupil Premium / Higher Attainers statement:

At Buckingham Park School, we value, nurture and celebrate the skills and talents of every child.

Our curriculum is aspirational, vibrant, engaging and inclusive. We strive to enable all children to do their best and optimise their potential through quality first teaching, careful planning- in line with developmental stage and interests of cohorts, removal of barriers in accessing the curriculum e.g., writing frames, visual prompts, adapted resources and alternative methods of recording.

The careful planning and learning opportunities are designed to reduce, and ultimately remove, gaps between disadvantaged and vulnerable learners and their peers. Our commitment to engaging, inspiring and equipping all learners is at the heart of our curriculum intent and fosters the implementation of our curriculum and the development of skills, in both academic and non-academic subjects equally, ensuring equality of opportunity and a broad and balanced provision- for all learners- in a holistic and personalised way.

Careful consideration is afforded to the broad and diverse offer interwoven throughout our curriculum to reflect our multi-cultural multi faith school community and so that quality, first-hand experiences are presented in a multitude of ways enabling full participation and maximum engagement. All educational visits are risk assessed and planned, so that every child may access and enjoy these educational opportunities, capitalising on enriching and memorable experiences.

At Buckingham Park School we endeavour to instil a love of learning, for life.

3. IMPLEMENTATION

Our scheme of work recognises that computing has three inter-related aspects, and these are covered in each year:

- Computer Science (the foundations of computing, covering coding and computational thinking)
- Information Technology (the applications of computing, including working with documents, data and digital media)
- Digital Literacy (the implications of computing for individuals and society)

Our computing curriculum also recognises the 'spiral' nature of progression within computing: new knowledge, skills and understanding within each of the strands of the subject build on what's gone before. See curriculum overviews and knowledge organisers below.

Computing in EYFS

In Early Years foundation stage pupils build confidence to use technology purposefully to support their learning for all the Early Learning Goals as appropriate. Pupils in the foundation stage classes will have experience using technology indoor, outdoors and through role play in both child-initiated and teach directed

time. Opportunities for technology as a tool to support learning and teaching in all areas are identified in their curriculum planning.

In Early Years, we develop the foundation for computing skills by teaching pupils to complete a simple program on a computer.

Pupils will learn to:

- To control a simple program.
- To program a toy (Bee-Bot) using simple instructions
- To understand that they control the programmable toy
- To use a suitably aged program on a computer effectively

Pupils will also be taught to:

- Use ICT hardware to interact with age appropriate computer software
- To turn the computer on/off
- To use the mouse effectively to achieve a desired outcome
- To can use age appropriate software correctly.
- To be safe online

The most relevant statements for Computing are taken from the following areas of learning:

- Understanding the World
- Personal, Social & Emotional Development

<u>Computing scheme of work units by year group</u>

Year 1

BOOK 1

Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
1.1	We are treasure hunters	Solving problems using programmable toys	Blue-Bots Blue-Bot app	Other programmable toys Scratch Bee-Bot emulator	Computer Science: Coding
1.2	We are TV chefs	Filming the steps of a recipe	iPads Camera app iMovie	Laptop/desktop computers Digital cameras Android tablets WeVideo Microsoft Photos	Computer Science: Computational thinking
1.3	We are digital artists	Creating work inspired by great artists	iPads Brushes Redux Autodesk Sketchbook	Laptop/desktop computers Chromebooks Android tablets Microsoft Paint/Paint 3D PaintZ for Chromebook	Information Technology: Creativity
1.4	We are publishers	Creating a multimedia eBook about our achievements	iPads Book Creator Google Photos	Laptop/desktop computers Chromebooks Google Slides Microsoft PowerPoint	Digital Literacy: Online safety
1.5	We are rhythmic	Creating sound patterns in ScratchJr and GarageBand	iPads GarageBand ScratchJr	Laptop/desktop computers Chromebooks Scratch Audacity, LMMS, Soundtrap	Information Technology: Media
1.6	We are detectives	Using data to solve clues	iPads Popplet Google Forms Google Sheets	Laptop/desktop computers Chromebooks FreeMind, Bubbl.us, MindMeister Microsoft Forms and Excel	Information Technology: Data

Year 2

BOOK 2

Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
2.1	We are astronauts	Programming on screen in ScratchJr	iPads ScratchJr	Laptops/desktops/Chromebooks Android tablets Blue-Bots or Bee-Bots Scratch	Computer Science: Coding
2.2	We are games testers	Working out the rules for games	iPads Scratch Laptops/desktops/Chromebooks FixTheFactory	Android tablets Blockly Games	Computer Science: Computational thinking
2.3	We are photographers	Taking, selecting and editing digital images	iPads Camera app Photos app Snapseed	Android tablets Laptops/desktops/Chromebooks Digital cameras Pixlr, Windows Photos	Information Technology: Media
2.4	We are safe researchers	Researching a topic	iPads Popplet Google Slides Google custom search	Android tablets Laptops/desktops/Chromebooks FreeMind Microsoft PowerPoint, Keynote	Digital Literacy: Online safety
2.5	We are animators	Creating a stop-motion animation	iPads Stop Motion Studio	Android tablets Laptops/desktops/Chromebooks Digital cameras iStopMotion, Zu3D Stop Motion Animator	Information Technology: Media
2.6	We are zoologists	Collecting data about bugs	iPads Google My Maps Google Docs/Sheets/Slides Camera and Photos apps	Laptops/desktops/Chromebooks Digital cameras Windows Maps Microsoft suite	Information Technology: Data

Year 3

BOOK 3

Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
3.1	We are programmers	Programming an animation	Laptops/desktops/Chromebooks Scratch	Android tablets Cameras and microphones ScratchJr	Computer Science: Coding
3.2	We are bug fixers	Finding and correcting bugs	Laptops/desktops/Chromebooks Scratch Screen recorder software	Android tablets Snap!	Computer Science: Computational thinking
3.3	We are presenters	Videoing a presentation against a green screen	iPads Green screen background Tripods and iPad mounts Popplet iMovie	Camera app Microsoft Photos Adobe Premiere Elements	Information Technology: Media
3.4	We are who we are	Creating presentations about ourselves	Laptops/desktops/Chromebooks Google Slides Screen recorder software	iPads or Android tablets Microsoft PowerPoint	Digital Literacy: Online safety
3.5	We are co-authors	Producing a wiki	Laptops/desktops Google Sites Popplet	iPads or Android tablets Chromebooks	Information Technology: Media
3.6	We are opinion pollsters	Collecting and analysing data	Laptops/desktops/Chromebooks Google Forms Google Sheets Google Slides Google Drive	iPads or Android tablets Microsoft equivalents j2vote, j2data and j2office	Information Technology: Data

Year 4

BOOK 4

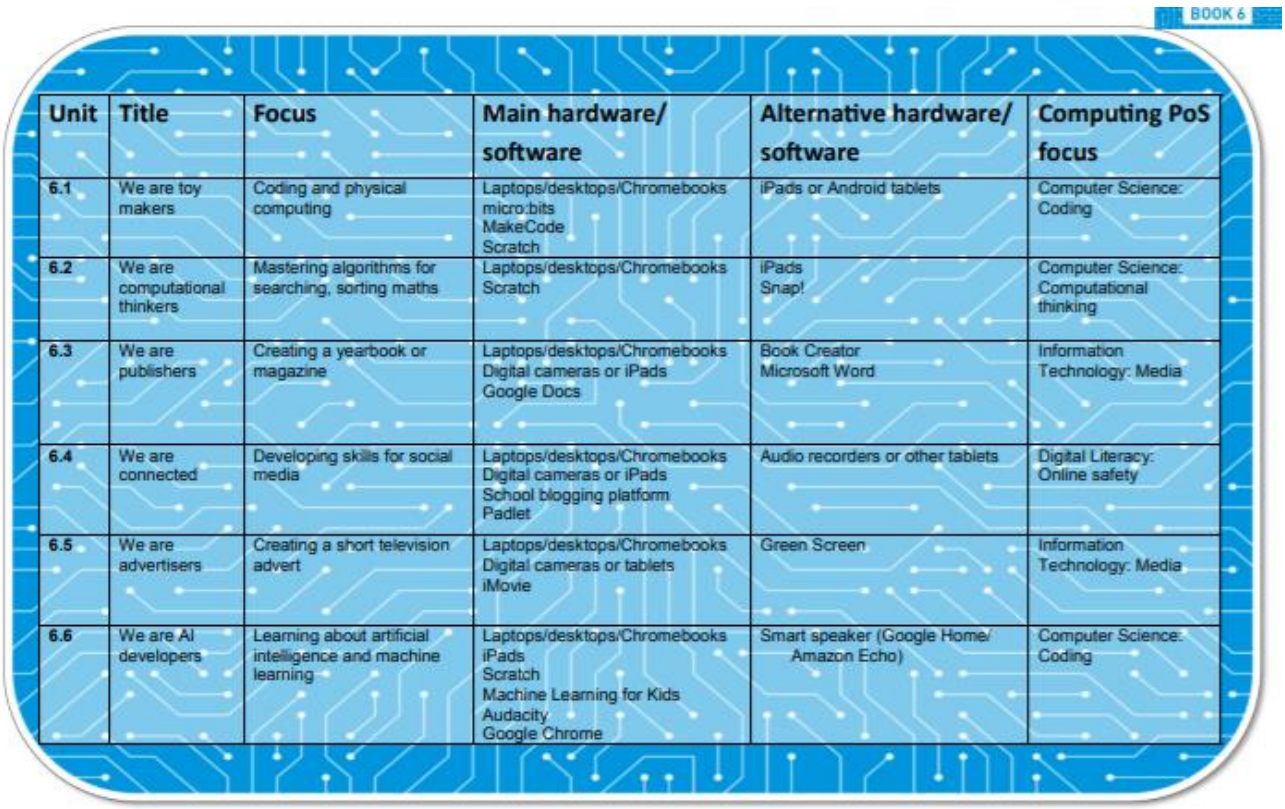
Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
4.1	We are software developers	Developing a simple educational game	Laptop/desktop computer Scratch	Snap! Pyonkee	Computer Science: Coding
4.2	We are makers	Coding for micro:bit	Laptop/desktop computer micro:bit Microsoft MakeCode	Crumble	Computer Science: Coding
4.3	We are musicians	Creating a piece of music in GarageBand	iPad GarageBand	LMMS	Information Technology: Media
4.4	We are bloggers	Sharing experiences and opinions	Laptop/desktop computer Digital camera WordPress or Blogger	Audio recorders or tablets	Digital Literacy: Online safety
4.5	We are artists	Fusing geometry and art	Laptop/desktop computer Scratch Inkscape Terragen	Logo Adobe Ideas Pyonkee Snap!	Computer Science: Coding
4.6	We are meteorologists	Recording and presenting the weather	Equipment for measuring weather Microsoft Excel Microsoft PowerPoint Keynote	Google suite	Information Technology: Data

Year 5

BOOK 5

Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
5.1	We are game developers	Developing an interactive game	Laptops/desktops/Chromebooks Scratch	Microphones (optional) Snap! Kodu	Computer Science: Coding
5.2	We are cryptographers	Cracking codes	Laptops/desktops/Chromebooks iPads or Android tablets Scratch	Snap!	Computer Science: Computational thinking
5.3	We are architects	Creating a virtual space	Laptops/desktops/Chromebooks iPads or Android tablets Trimble SketchUp Screen recorder	CoSpaces Minecraft Education Edition	Information Technology: Media
5.4	We are web developers	Making sense of the Internet and building a website	Laptops/desktops/Chromebooks iPads or Android tablets Google Chrome Google Sites	N/A	Digital Literacy: Online safety
5.5	We are adventure gamers	Creating an interactive adventure using presentation software	Laptops/desktops/Chromebooks Google Slides Voice recorder	iPads Microsoft PowerPoint	Information Technology: Media
5.6	We are VR designers	Experimenting with virtual and augmented reality	iPads or Android smartphones Google Cardboard Google Street View GarageBand CoSpaces	N/A	Information Technology: Media

Year 6



BOOK 6

Unit	Title	Focus	Main hardware/ software	Alternative hardware/ software	Computing PoS focus
6.1	We are toy makers	Coding and physical computing	Laptops/desktops/Chromebooks micro:bits MakeCode Scratch	iPads or Android tablets	Computer Science: Coding
6.2	We are computational thinkers	Mastering algorithms for searching, sorting maths	Laptops/desktops/Chromebooks Scratch	iPads Snap!	Computer Science: Computational thinking
6.3	We are publishers	Creating a yearbook or magazine	Laptops/desktops/Chromebooks Digital cameras or iPads Google Docs	Book Creator Microsoft Word	Information Technology: Media
6.4	We are connected	Developing skills for social media	Laptops/desktops/Chromebooks Digital cameras or iPads School blogging platform Padlet	Audio recorders or other tablets	Digital Literacy: Online safety
6.5	We are advertisers	Creating a short television advert	Laptops/desktops/Chromebooks Digital cameras or tablets iMovie	Green Screen	Information Technology: Media
6.6	We are AI developers	Learning about artificial intelligence and machine learning	Laptops/desktops/Chromebooks iPads Scratch Machine Learning for Kids Audacity Google Chrome	Smart speaker (Google Home/ Amazon Echo)	Computer Science: Coding

Examples of Knowledge organisers.

Year 1

Knowledge Organiser

Unit 1.1: We are treasure hunters

Solving problems using programmable toys

Buckingham Park
Church of England Primary Schoo
Excellence, through God who strengthens

Software: Blue-Bot app (optional, alternatives: programming interface for alternative toys)
Hardware: Blue-Bot (alternatives: Cubetto, Bee-Bot, Roamer Too, STEM Robot Mouse). If robot toys are not available the Blue-Bot app or the Scratch Bee-Bot emulator can be used instead.

Overview

In this unit, pupils learn basic programming ideas through experimenting and solving problems with simple, programmable robots, such as the Blue-Bot. In:

- **Session 1** they take on the role of a **robot**, responding to instructions they are given
- **Session 2** they take on the role of a robot-pirate to work out a sequence of instructions (an **algorithm**) to find their way to an objective
- **Session 3** they explore the Blue-Bot controls
- **Session 4** they follow, create and test sequences of instructions to solve problems with the Blue-Bot
- **Session 5** they predict what the Blue-Bot will do when given different sequences of instructions
- **Session 6** they correct mistakes in Blue-Bot **programs**.

Key vocabulary

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Computer: a device that accepts input, processes it according to instructions or rules and produces output

Debug: correct mistakes in a computer program or algorithm

Input: data supplied to a computer, in this case, pressing buttons on the robot

Logical reasoning: to be able to give a reason for something which others would have to accept as correct

Output: information produced by a computer – in this case, movements of the robot

Program: a sequence of instructions (or sometimes a set of rules) that can be followed by a computer

Robot: a computer that can move, or that can move part of itself

Assessment – by the end of the unit:

All pupils can:
- follow instructions to move around a large space
- record a set of instructions for a Blue-Bot
- **program** a Blue-Bot to move by giving one instruction at a time
- program a Blue-Bot to move by giving a sequence of instructions.

Most pupils can:
- give other pupils instructions to move around a large space
- understand **input**, **program** and **output** in the context of a Blue-Bot
- create a program to move a Blue-Bot to a particular location
- **debug** a Blue-Bot program.

Some pupils can:
- predict where a set of instructions will take someone moving in a large space
- predict where a Blue-Bot will end up from a set of instructions
- understand input, program and output in more general contexts
- look for ways in which a Blue-Bot program could be made more efficient.

Cross-curricular opportunities

English: Pupils provide clear instructions for moving around a map. Pupils could read stories about pirates, or have some pirate stories shared as class texts.

Geography: Pupils use geographical language and could use and make their own maps. They could learn about places where pirates would have been found.

History: Pupils could learn about piracy in the past.

Maths: Pupils recognise movements in a straight line (translations) and rotations, and combine them in simple ways. They start to recognise and make whole, half and quarter turns, and learn to recognise a right angle.

PSHE and citizenship: Pupils could consider the effect that pirates' actions had on others.

Year 2

Knowledge Organiser

Unit 2.1: We are astronauts

Programming on screen in ScratchJr

Buckingham Park
Church of England Primary Schoo
Excellence, through God who strengthens

Software: ScratchJr (alternative: Scratch)
Hardware: iPads (alternatives: Android tablets, laptop/desktop/Chromebook computers, Bee-Bots, Blue-Bots)

Overview

In this unit, pupils program a **sprite** (such as a spaceship) to move around the screen. In:

- **Session 1** they take part in playground activities, planning movement between 'planets'
- **Session 2** they are introduced to **ScratchJr** and **program** sprite movement
- **Session 3** they are introduced to **output** and use multiple sprites
- **Session 4** they are introduced to message passing and **input**
- **Session 5** they are introduced to **repetition**
- **Session 6** they create new "costumes" for their

Key vocabulary

Abstraction: computational thinking approach to managing complexity by simplifying things through identifying what is important, and what detail can be hidden

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Code: instructions (or sometimes rules) that can be understood by a computer

Debug: correct mistakes in a program or algorithm

Event: something that happens within a computer program to cause some particular code to be run, such as an internal message being received or a sprite being tapped by the user

Input: data supplied to a computer, in this case, tapping on the screen of a tablet

Output: information produced by a computer – in this case, moving sprites on a screen, text and audio

Parallel processing: when programs run (or appear to run) simultaneously

Program: sequence of instructions (or sometimes a set of rules) that can be followed by a computer

Repetition: programming construct which allows a group of instructions to be repeated a number of times, or until a certain condition is met

Scratch: simple, block-based programming language in which programs for characters are built by snapping together code blocks

Sprite: a graphical character in a program that can be given its own sequence of instructions

Assessment – by the end of the unit:

All pupils can:
- plan a route from one hoop to another in the playground
- create a sequence of move instructions on screen
- record audio and add an instruction to play audio
- create a costume for a **sprite**.

Most pupils can:
- plan a return route in the playground
- create multiple sequences of move instructions
- add instructions to display a sequence of texts
- use different events to launch **code**
- create costumes for multiple sprites.

Some pupils can:
- plan a route visiting multiple hoops in the playground
- predict correctly what a sequence of instructions will do
- have events launch multiple programs in parallel
- use internal messages to control the behaviour of sprites
- use repetition in their programs
- create a background scene.

Cross-curricular opportunities

English: Pupils could explore some aspects of space travel through reading and creative writing.

History: Includes references to early space travel.

Maths: Pupils use simple arithmetic to work out how far the sprites must move, and use the language of position, movement and estimating distances. They think about how to approximate circular movement on a coordinate grid.

Science: The unit could be extended as part of a wider, cross-curricular topic exploring space.

Year 3

Knowledge Organiser

Unit 3.1: We are programmers

Programming an animation

Buckingham Park
Church of England Primary School
Excellence, through God who strengthens us

Software: Scratch (alternative: ScratchJr)
Hardware: Laptop/desktop/Chromebook computers or tablets, cameras and microphones (if needed)

Overview

In this unit, pupils create their own animation in **Scratch**. In:

- **Session 1** they are introduced to Scratch, and explore its tools
- **Session 2** they determine key features of a good animation by looking at examples, and create a **storyboard** for their own
- **Session 3** they plan and **program** character/s and dialogue for their animation
- **Session 4** they begin to animate their character/s by planning and programming movement
- **Session 5** they plan and program switching costumes and backdrops for their animation
- **Session 6** they add sound to their animation before reviewing, **debugging** and improving it.

Alternatives

The unit sessions give step-by-step guidance on using Scratch. However, this unit could also be carried out using ScratchJr. A downloadable version of Scratch is also available – 'Scratch desktop' (see *Useful links* on page 12). You can also download this if you prefer to work offline.

Key vocabulary

Abstraction: computational thinking approach to managing complexity by simplifying things, through identifying what is important and what detail can be hidden or ignored

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Code: instructions (or sometimes rules) that can be understood by a computer

Debug: correct mistakes in a computer program or algorithm

Decomposition: breaking a problem down into smaller parts

Event: something that happens within a computer program to cause some particular code to be run, such as an internal message being received, or a sprite being tapped by the user

Iterative development: A trial and improvement approach to programming or other work, in which each successive version builds on the previous one by the fixing of mistakes or the adding of features

Output: information produced by a computer – in this case, an animation

Parallel processing: when programs run (or appear to run) simultaneously

Program: a sequence of instructions (or sometimes a set of rules) that can be followed by a computer

Repetition: programming construct which allows a group of instructions to be repeated a number of times, or until a certain condition is met

Scratch: simple, block-based programming language in which programs for characters are built by snapping together code blocks

Sequence: placing programming instructions in order, so that each happens one after the other

Sprite: a graphical character in a program that can be given its own sequence of instructions

Storyboard: a visual representation of the key scenes or frames in a video; one way of recording an algorithm visually

Knowledge Organiser

Buckingham Park
Church of England Primary School
Excellence, through God who strengthens us

Unit 4.1: We are software developers
Developing a simple educational game

Software: Scratch (alternative: Snap!)
Hardware: Laptop/desktop/Chromebook computers or tablets, microphones (optional)

Overview

In this unit, pupils plan, create, develop and test their own educational game for a target audience. In:

- **Session 1** they analyse existing games and identify what makes them effective
- **Session 2** they create a working prototype of their game
- **Session 3** they develop the functionality of their game
- **Session 4** they improve the interface of their game
- **Session 5** they develop progression within their game
- **Session 6** they test and improve their game.

Cross-curricular opportunities

Maths: Pupils can practise recall of multiplication and/or division facts, rounding decimals with one decimal place to the nearest whole number or converting between different units of measure in their games.

English: Pupils can create their own spelling tests using audio recording and playback.

Languages: Pupils can practise vocabulary in foreign languages in their games.

Other: Any subject where there are facts to learn can also provide a useful context, such as dates in history or capital cities in geography.

Assessment – by the end of the unit:

All pupils can:

- design an interactive educational game
- develop an interactive educational game
- put **Scratch** blocks into the right order for their game
- use the *if/then/else* block correctly
- use the keyboard for **input** and the screen for **output**.

Most pupils can:

- use a repeat block correctly
- keep track of random numbers and the score
- integrate sound into their game
- correct mistakes in their game.

Some pupils can:

- plan their own approach to developing their game
- use a countdown timer
- use the mouse to control the game
- explain how the **algorithm** that underlies their game works
- use logical reasoning to detect and correct **bugs** in their game.

Key vocabulary

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Debug: correct mistakes in a computer program or algorithm

Input: data supplied to a computer, in this case the algorithm taken from the storyboard for the animation

Output: information produced by a computer, in this case an animation

Program: a sequence of instructions (or sometimes a set of rules) that can be followed by a computer

Repeat loop: a sequence of instructions executed a fixed number of times or until some condition is met, or possibly forever

Repetition: programming construct which allows a group of instructions to be repeated a number of times, or until a certain condition is met

Scratch: simple, block-based programming language in which programs for characters are built by snapping together code blocks

Sequence: placing programming instructions in order, so each one happens one after the other

Sprite: a graphical character in a program that can be given its own sequence of instructions

Variable: lets computer programs store, retrieve or change simple data. Typically thought of as a particular location in the computer's memory that holds a specific item of data

Knowledge Organiser

Unit 5.1: We are game developers

Developing an interactive game

Buckingham Park
Church of England Primary School
Excellence, through God who strengthens us

Software: Scratch (alternatives: Snap! or Kodu)

Hardware: Laptop/desktop/Chromebook computers or tablets, microphones (optional)

Overview

In this unit, pupils plan their own simple computer game. They design characters and **backgrounds**, and create a working prototype, which they develop further based on feedback they receive. In:

- **Session 1** they analyse games and plan their own
- **Session 2** they create and source assets
- **Session 3** they create a prototype of a **Scratch** game
- **Session 4** they **debug** the game script
- **Session 5** they test and improve their game
- **Session 6** they write game instructions and publish their games.

Assessment – by the end of the unit:

All pupils can:

- create an **algorithm** for a game
- create images and sounds for their games
- use sequences of instructions
- detect errors in their game.

Most pupils can:

- create music for use in their game
- use selection and repetition in their game
- correct errors in their game
- improve their game on the basis of feedback
- add instructions to their game.

Some pupils can:

- break their game into its component parts and develop them separately
- create multiple images for characters and use them for animation
- use variables in their game
- explain how their game works
- include comments in the **code** for their game.

Cross-curricular opportunities

Art and design: Pupils can improve their art and design skills by creating artwork for their games.

History/geography: Pupils could create a game that links to a topic being studied.

Maths: Pupils could create a maths or times table game.

English: Pupils could create a game that tests subject knowledge of grammar, for example.

Languages: Pupils could create a game that ties in with a modern foreign language that they are learning.

Music: Pupils can record sound or compose music for their games.

Key vocabulary

Algorithm: a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective

Background: scenery and other unchanging elements in a game

Bug: an error or mistake in a program or algorithm, causing the computer or robot to behave in a manner that was not originally intended

Code: instructions (or sometimes rules) that can be understood by a computer

Debug: correct mistakes in a computer program or algorithm

Iterative development: a trial and improvement approach in which each successive version builds on the previous one by the fixing of mistakes or the adding of features

Logical reasoning: a systematic approach to solving problems or deducing information using a set of universally applicable and totally reliable rules

Program: an automated solution to a problem

Scratch: simple, block-based programming language in which programs for characters are built by snapping together code blocks

Sprite: a graphical character in a program that can be given its own sequence of instructions

Knowledge Organiser

Buckingham Park
Church of England Primary School
Excellence, through God who strengthens us

Unit 6.1: We are toy makers

Coding and physical computing

Software: MakeCode (alternative: Scratch)
Hardware: Laptop/desktop/Chromebook computers or tablets, BBC micro:bits

Overview

In this unit, pupils design and develop a BBC micro:bit powered modification to a soft toy to make the toy interactive. In:

- **Session 1** they revisit **micro:bit** and **MakeCode**
- **Session 2** they research electronic toys
- **Session 3** they design their toy
- **Session 4** they program the micro:bit
- **Session 5** they prepare their toy for adding interactive components
- **Session 6** they connect the micro:bit to the toy.

Alternatives

Crumble **controllers** or Raspberry Pi computers could be used in place of the micro:bit. The activities here could be completed using LEGO WeDo or Mindstorms controllers, or the Makey Makey board as **input** to Scratch.

Key vocabulary

Accelerometer: hardware component providing data on changes in motion, typically in three directions

Bluetooth: wireless digital communication protocol using low energy signals over short distances

Controller: programmable device that determines electronic output based on electronic input

Decomposition: breaking a problem down into smaller parts

Edge connector: part of a circuit board that allows input and output components to be directly connected

Embedded system: computer hardware and software that forms part of a device or product

Input: data supplied to a computer

Interactive: system whose output is determined by the input provided

Light-emitting diode (LED): an electronic component that lights up when current flows in one direction

MakeCode: block- and text-based editor from Microsoft, supporting a variety of hardware platforms including the micro:bit

micro:bit: simple, single board programmable computer with integrated input, output and network capabilities

Microprocessor: single silicon chip that performs all the functions of a computer's central processing unit

Output: information produced by a computer

Simulator: software that allows one computer system to behave as another; in this case, the MakeCode editor includes an on-screen simulator of a micro:bit so that programs can be tested

System: a set of components (perhaps of different types, such as hardware and software) working together

Assessment – by the end of the unit:

All pupils can:

- identify **inputs** and **outputs** for their toy
- name inputs and outputs for the **micro:bit**
- design an **interactive** toy
- add interactivity to a toy
- design a program to control the toy
- connect the micro:bit to their interactive toy.

Most pupils can:

- compare possible toy designs
- program the micro:bit to control their toy
- **decompose** the toy project into a number of stages
- identify problems with their toy
- identify bugs in their program.

Some pupils can:

- use criteria to evaluate possible toy ideas
- identify risks in the project and suggest ways to mitigate these
- take a lead in managing the project with a partner or in their group
- fix problems they encounter
- debug mistakes in their program
- provide constructive feedback to others using agreed criteria.

Cross-curricular opportunities

Design and technology: Pupils should be taught to apply their understanding of computing to program, monitor and control their products. Pupils are likely to make use of design and technology tools, resources and skills as they work on their toys, and an integrated approach to planning and delivering the unit is recommended.

Art, science and maths: Pupils are also likely to draw on skills, knowledge and understanding from art, science and maths as they work on their toys.

Planning

By having access to the Switched On Computing scheme of work the staff ensure all National Curriculum objectives are met. Teachers employ a range of readily available or free software in their lessons from Paint and iMovie, to Scratch and Kodu. Use is also made of hardware such as digital cameras, video cameras, learn pads, bee bops and web servers. Staff are also able to chose the software and websites to use which we already have access too. Working along side our IT company, staff are able to request applications they feel would be beneficial for the children's learning to meet the required objectives.

For each computing unit the teacher will use:

- Teaching slides: editable presentations that can be used as a front-of-class teaching tool to go through the steps in each session
- Video walkthroughs: show the steps in a session when an application is used – can be used as a front of-class teaching tool to model the steps or allocated to pupils to work through
- Pupil worksheets: there are a range of different types including:
- handouts that take pupils through the activity steps
- supporting worksheets for appropriate activities
- self-assessments at the end of each unit for children
- Key vocabulary used in each unit for children to understand.
- End-of-unit knowledge quiz: questions that test pupils' knowledge and understanding of the key concepts in the unit
- CPD videos: include guidance on computing concepts and subject knowledge to support staff.

Assessment

Formative assessment is undertaken by teachers throughout the lessons, they assess children's knowledge, understanding and skills in Computing by making observations, through conversations with the children during lessons, and the quality of the digital content they create. Children's work is saved on the school server for teachers to assess. Our scheme of work has built in the assessing activities during a lesson to allow the teacher to take stock of the children's progress and then provide feedback. Summative assessment is made at the end of each unit to identify children who have excelled, and those who require extra support.

Teaching Safe Use of the Internet and ICT

We believe that it is crucial to teach pupils how to use the Internet safely, both at home and at school, and we use the Kidsmart safety code to support our teaching in this area:

Kidsmart has been developed by the Childnet charity and is endorsed by the DfES

http://www.kidsmart.org.uk

The main aspects of this approach include the following five SMART tips:

Safe - Staying safe involves being careful and not giving out your name, address, mobile phone no., school name or password to people online.

Meeting someone you meet in cyberspace can be dangerous. Only do so with your parents'/carers' permission and then when they are present.

Accepting e-mails or opening files from people you don't really know or trust can get you into trouble - they may contain viruses or nasty messages.

Reliable - someone online may be lying and not be who they say they are. If you feel uncomfortable when chatting or messaging end the conversation.

Tell your parent or carer if someone or something makes you feel uncomfortable or worried.
Each teacher will spend part of a lesson each term to educate and discuss online safety and revisit this regularly throughout computing lessons. PSHE lessons also focus on this too. On our school website there is some more information for parents "Keeping your children safe online"
http://www.buckinghampark.bucks.sch.uk/online-safety.html

4. IMPACT

At Buckingham Park we measure the impact of our curriculum through the following methods:
• Learning Works
• Scrutiny of work
• Pupil discussions about their learning; which includes discussion of their thoughts, ideas, processing and evaluations of work.
• Staff Questionnaires.

The impact of Buckingham Parks' computing curriculum can not only be seen in displays around school and on the children's individual computer accounts, but also can be measured by speaking to the children themselves. The teaching of our computing curriculum enables our children to use a computer with confidence. Our Computing Curriculum has been structured to demonstrate a progression of skills and ensures that children can build on their understanding, as each new concept and skill is taught with opportunities for children to revisit skills and knowledge as they progress through school.

After the implementation of this robust computing curriculum, children at Buckingham Park School will be digitally literate and able to join the rest of the world on its digital platform. They will be equipped, not only with the skills and knowledge to use technology effectively and for their own benefit, but more importantly – safely. As children become more confident in their abilities in Computing, they will become more independent and key life skills such as problem-solving, logical thinking and self-evaluation become second nature.