Year / Topic	Term	National Curriculum Links	Length of Topic
6.2 Scratch	Autumn 2	CS2.1, CS2.2, CS2.3	7 Weeks

Resources

Student Booklets
PowerPoint
Scratch
Kinect

Target Skills

Programming
Planning
Design
Evaluation

Curriculum Links

Maths - *movement, direction, rotation and angles.*

SMSC – *recapping e-safety.*

E-Safety Coverage

Students will need to consider copyright when sourcing images or media for their games. The students will revisit e-safety rules to create an e-safety animation as part of their assessment work.

Key Classroom ICT Activity

Students will revisit Scratch to create a series of basic animations to improve their programming skills. They will learn the importance of a correctly ordered algorithm and how to implement loops, if and while statements. At the end of the unit students will use their learnt skills to create an e-safety animation and a Kinect game in groups.

By the end of the topic students will be able to:

- Evaluate and review their own and each other's work.
- Create their own working animation using algorithms.
- Understand the importance of initialising at the start of an animation.
- Be able to explain and justify choices of code used.

Assessment - Progression Pathways

All children should **–** *CS, Understand what an algorithm is and is able to express similar linear algorithms symbolically. Understands that computers nee precise instructions. Demonstrates care and precision to avoid errors.*

CS, Understands that computers have no intelligence and that computers can do nothing unless a program is executed. Recognises that all software executed on digital devices is programmed.

Most children should – CS, Uses arithmetic operators, if statements, and loops within programs. Uses logical reasoning to predict the behaviour of programs. Detects and corrects simple semantic errors i.e. debugging in programs.

CS, Recognises that a range of digital devices can be considered a computer. Recognises and can use a range of input and output devices.

Some children should – CS, Creates programs that implement algorithms to achieve given goals. Declares and assigns variables. Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including if, then else statement.

CS, Shows an awareness of tasks best completed by humans or computers. Designs solutions by decomposing a problem and creates a sub-solution for each of these parts (decomposition). Recognises that different solutions exist for the same problem.

Assessment Criteria	6.2 Scratch
Emerging	✓ I can understand what an algorithm is. ✓ I understand that computers see precise instructions. ✓ I can demonstrate care and precision to avoid errors.
Developing	✓ I can use logical reasoning to predict the behaviour of programs. ✓ I can detect and correct simple errors in my code i.e. debugging in programs. ✓ I can use the repeat and forever loop. ✓ I can recognise and can use a range of input and output devices.
Secure	✓ I can use algorithms to achieve given goals. ✓ I can declare and assign variables. ✓ I can use is and if, then else statements
Mastered	✓ I can show an awareness of tasks best completed by humans or computers. ✓ I can design solutions by decomposing a problem and creates a sub-solution for each of these parts (decomposition). ✓ I can recognises that different solutions exist for the same problem.