Key Stage 3 Curriculum Map 2019 - 2020

Term 2

Year Group: Year 7	Subject: Computer Science		
Focus/Topic	**Objectives**	**Key Skills**	**Home Learning/Recommended Reading**
• Kodu – How programs work	• Learn what the terms **program**, **navigate**, **object** and **world** mean in computer games design • Understand that a computer program requires a precise series of instructions to operate correctly	• Kodu • Object • World • Sequencing • instructions • Event driven programming	• Kodu Homework 1 – explain the code
• Kodu – creating landscapes	• Create and alter basic landscape features in Kodu • Learn a range of techniques for creating a landscape which is suitable for a chosen game	• Landscaping • Suitability • Forward planning	• Kodu Homework 2 – What makes a good computer game?

• Kodu – Advanced navigation and pathing	• Learn about a range of game techniques such as pathing
	• Learn the steps involved in programming at least two different methods, one manual and one automatic, to make a Kodu move
	• Apply a range of skills to modify and create a simple Kodu game world which interacts with objects
		• Automatic pathing
		  ○ When see…
		  ○ …Do move…
		  ○ paths
		• Manual pathing
			• Install Kodu and continue development

• Kodu – Clones and Creatables	• Learn how to create clones and creatables • Understand the difference between clones and creatables • Be able to explain the advantages of each in terms of ease of program maintenance	• Effective cloning • Editing clones	• continue development
• Kodu – Pages and Selection	• Understand what is meant in programming by the term **selection** • Learn how the selection concept of pages in Kodu can be used in order to code different behaviours • Modify a game to make a Kodu move in response to behaviours	• Sequence • Selection • Pages	• Kodu Homework 3 – Interpreting a Kodu Code listing
• Kodu – Game depth and Complexity	• Learn how to use a range of more advanced game techniques such as **power ups**, **timers**, **health** and **sound** • Explain how to use scoring and methods such as **colour winning** to add additional depth to a game	• Advanced functions	• Complete development of game
Mid Term Break			
• **Kodu - Assessment**	• **Kodu - Assessment**	• **Kodu - Assessment**	• **Kodu - Assessment**
• Scratch - Movement	• Understand that Scratch is a programming environment that allows you to create games, animations and other simulations • Understand what is meant by an algorithm • Create a sprite and write code to make it move and bounce • Load and use an existing Scratch file • Produce design ideas for a Scratch project	• Algorithm • Sprite • Scratch project	• MS Teams

• Scratch – Lives and scoring	• Define a variable • Write algorithms which use variables to hold values such as **Number of Lives Left** or **Score** in a computer game • Understand the purpose of comments in a program • Annotate a program with comments •	• Selection • Iteration • Annotations • Variable declaration • Comments/annotation	• MS Teams - Find a simple game on the Scratch Community site and print the code blocks. Annotate the code to explain what various blocks of code do.
• Scratch – adding a new level	• Understand the purpose of repeat loops and procedures (“broadcasts”) • Use a broadcast in your own Scratch program •	• Selection • Iteration • Broadcast	• MS Teams
• Scratch – Randomising behaviour	• Learn what each of the operators in the Scratch Green block menu does • Use the **Pick Random** block to position objects randomly on the screen • Understand the use of the operators **<, =, >, and, or, not**. • Use some of these in a Scratch game	• Operators (green blocks)	• MS Teams - Continue with game development. Research and add one new game feature.
UAE Links across the term			