Acle Academy

YEAR 7, Coding (Scratch) KS3 Computer Science (Creative Industries) Second unit of three in rotation.

Rationale and Context of Unit:	Core curriculum content:	Tier 2 & Tier 3 vocabulary explicitly taught:
The KS2 curriculum is heavily focused on coding but students rarely come to us knowing the proper terminology of what they are doing, if they have even covered coding at their primary school. Schools in the area tend to use coding apps such as Scratch. Students can often complete simple block coding but cannot explain what they are doing with the correct terminology. Therefore they do not have an in depth understanding of what coding entails. We use Scratch in this unit to offer continuity with the primaries but build in lessons which explicitly teach the skills and vocabulary associated with coding. Microsoft office and the basics of computer systems was taught in the first section of lessons and so moving onto the basics of coding is a logical step regarding what must be covered on the KS3 curriculum. This unit will underpin further units where we encounter coding. The knowledge in this unit will be built upon when students learn how to code using python rather than pseudocode.	**Key knowledge taught:** The end goal of this unit is to be able to code a game using subroutines, decomposition, condition-controlled loops, iteration and variables while being able to use the terminology to explain what they are doing. Students will also think about how this knowledge applies in the real world by knowing your target market, surveying the class using google forms and doing a data analysis. Students will then present their game idea to the class promoting oracy skills. These skills will be used in the first unit of year 8 which focusses on coding: Python.	Duplicate (2) Algorithm (3) NB. Each lesson has a key words list to accompany the students' learning and more words may be explicitly taught than the above but these are obligatory.

Challenge and Support:	World wide learning/ links to 21st century:	Cultural capital/ Industry/ Enrichment:
Each lesson has EDSM descriptors and there are tasks in each lesson which target HAPs. This scheme not only gives students a good understanding of how coding works but also challenges them in terms of presentation and oracy skills. Students have to work in a team to produce a prototype game using scratch and present this to the class with a presentation. This encourages team work and oracy skills. Students will be taught how to change the colours of documents. A list of key words/ word bank is available for every lesson with definitions. Tasks are chunked with step by step instructions and the lessons powerpoints are saved on their google classroom. There is an element of challenge attached to this unit as there is a project which promotes independence and organisation in teams. Extra help guides are also available in both electronic and printed out formats for various pieces of software. Students are given plenty of time (at least a week) to complete any homework tasks. They are encouraged to complete this at lunchtime or at homework club, giving them access to computers, if they do not have IT access at home.	Each lesson has either a 'real life link' or a 'link to careers' section, depending on which one is relevant to the lesson. Nearly every student of this age plays video games of some sort so it is helpful for them to see how much work and coding a game takes. If there are any budding games designers or programmers in the group, then this will be an excellent introduction for them. Hopefully this unit will inspire students to think about a career in this field.	See previous section for links to careers. As students have to work in a team to produce their game / presentation, teamwork and organisational skills will be paramount in producing an effective game / presentation. These soft skills will be useful in the workplace. Students also have to think about target markets and how companies will try and create games to target different markets. It is useful for them to see companies achieve this.

Lessons will be further differentiated in accordance with SEND and PP passports. Seating plans will be annotated based on passports. To support SEND students further, scaffolding, cognitive and metacognitive strategies, explicit Instruction and flexible grouping, memory retrieval tasks are used, along with the aid of technology.		
Historical, Social, Moral, Spiritual, Cultural context:	**Cross curricular links/ literacy/numeracy:**	**Common misconceptions:**
Personal development is promoted in this unit by working as a team. This also promotes social skills.	The presentation promotes oracy skills. The data analysis of the survey is a chance to look at how charts and graphs are used / presented. There are plenty of opportunities to read aloud in class and give information from the lesson powerpoint. There is an opportunity for extended writing when writing up the conclusions from the survey conducted in class.	*"You need to be an expert at maths to code".* *"Coding is dull / not very creative"* Maths and coding are not tightly coupled. There is a lot of creative thinking in creating a program and although there are programming roles which may involve complex maths, there are others which will involve next to no maths at all. Using Scratch to code is an effective way to prove to students that coding can be very creative and fun!
Assessment timeline:		
Game creation and presentation focusing on oracy skills – lesson 7 'Written' assessment – lesson 10		
Home learning		
HMK L4: Overlearning on loops HMK L6: Overlearning on scripts HMK L9: Revision on Scratch for test		

Further reading: https://scratch.mit.edu/ - Lots of extra projects and ideas on Scratch site
Feedback
Feedback given to teams on google classroom assignment Feedback / score given to individual students on google classroom assignment Class discussions used regularly. Online Quizzes. Test buddy feedback (peer assessment) used in class with criteria. Students can self-mark against criteria given.

Length of unit (duration indicated in lessons)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Unit: Coding (Second section of lessons of CS rotation)																													