

# Space Adventures Computing Unit 2

## Lesson 6 – Asteroid Cloud

### Curriculum Mapping (Computing KS2)

- ◆ use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- ◆ use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

### Learning Objective

Code a game flying Tazz's spaceship through an asteroid cloud.

### Prior Learning

Good knowledge of most KS2 coding techniques, familiarity with Scratch.

### Introduction

Show pupils the ***U2L6 introduction.mp4 video***. Using the prompts in the video, ask pupils to explain the rules that make the spaceship and asteroids move. (This can be done orally or written in rough, and is known as the algorithm).

### Main Activity

Pupils use Scratch to create their own asteroid cloud game. They will use variables to store the speed of each asteroid.

Show the class the ***U2L6 demonstration.mp4 video*** or how to access it on their own computers.

Hand out the ***U2L6 step by step.pdf*** guide or show pupils how to access it on their computers.

(Opening a second tab in the browser will allow pupils to switch between the help guide/video and their own work).

### Extension Activity

Show pupils the activities on the ***U2L6 going further.pdf*** document. These include experimenting with aspects of the code and changing the movement of the spaceship.

### Plenary

Remind the class what happens to Tazz when her spaceship gets hit by an asteroid. Brainstorm ideas to defend the spaceship from asteroids. Ask pupils to plan how they might implement some of these ideas with code.

### Notes

This program has two main parts - the code for the spaceship, and the code for the asteroids. The rocket code starts by resetting the timer and setting the size of the spaceship. It then uses a loop to check if any keys have been pressed. This is faster than using **when key pressed** event blocks.

The asteroid sprite code works in two parts. The first short loop creates a new asteroid sprite every two seconds using the create clone command.

The next part of the smoke code runs each time a smoke sprite is cloned. It starts by initialising the sprite (setting its size, position and speed). The speed variable is different for each asteroid, as we chose 'for this sprite only' when creating the variable. A loop then moves each asteroid. An if statement checks to see if it has reached the side of the screen by checking its x coordinate.