



University Learning in Schools

Computing

Game Creation in Scratch

Lesson Plans



Introduction- Please read carefully before commencing

In this module you have the opportunity to use Scratch and flowcharts to build students ability in programming, problem solving and STEM knowledge. Focus has been applied to ways in which a STEM focus can be applied and taking into studies later on in education with Maths and Physics incorporated into the module.

With students and teachers knowledge of programming so varied we felt it was important to create a module that was well differentiated and inclusive for both parties.

Each lesson has a range of PowerPoint's and Lesson Plans that provide detail and concepts that should be covered when moving through the lesson. Worksheets, extension tasks and help sheets have all been provided that can be used and adapted to suit the needs of each user.

Scratch (Programming language)

Scratch is a visual programming language that at its most basic allows users to create programs relatively quickly, with the use of blocks and scripts. Scratch is used in Primary and Secondary education and can be extremely useful when looking to build students programming capabilities leading into a text based programming language.

Scratch can be accessed through the MIT website with the online version available, allowing students to work from home effortlessly. You can also download the previous version to install as a program should internet connectivity be an issue.

Scratch Online (Recommended) - <https://scratch.mit.edu/>

Scratch 1.4 program download - https://scratch.mit.edu/scratch_1.4/

Note – .SB files that are folder in the module folders are Scratch files that will need to be opened with Scratch. Each lesson has a different version with the programming task in lesson incorporated in the version.

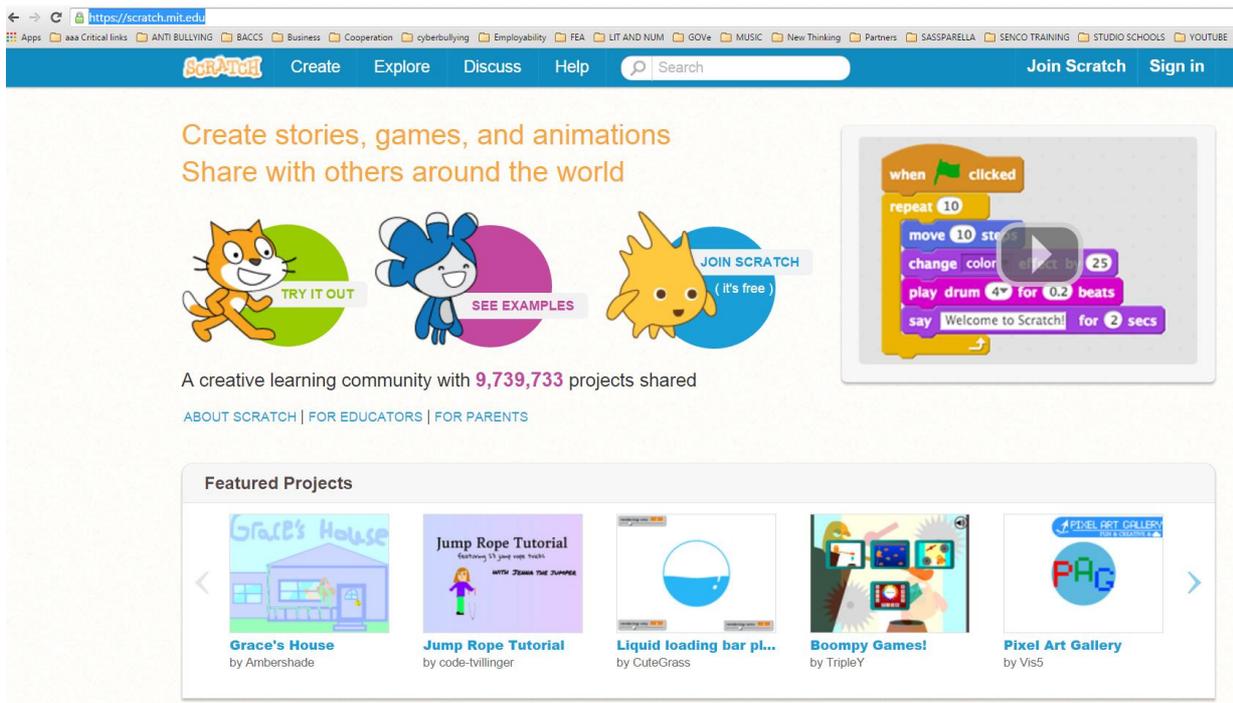
Assessment

With assessment changing and levels being removed/removed from the National Curriculum the assessment document may require change dependent on the school needs.

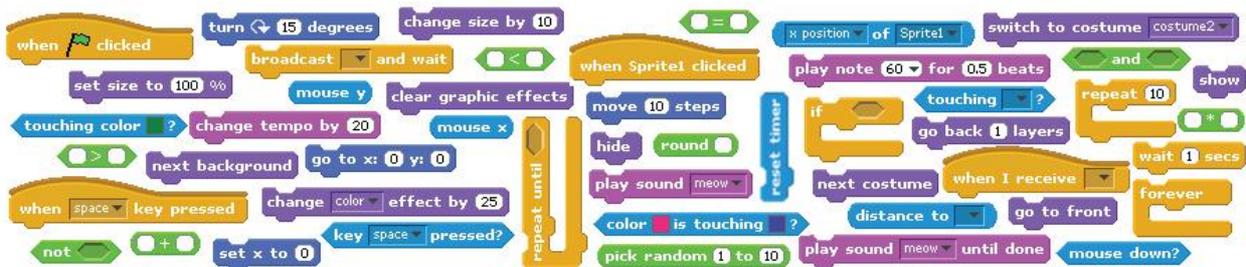
There is also the option to test students before and after the module as part of the assessment, or show game progression and progress over time by getting students to take screenshots of their game, and explain, at the end of each lesson.

About Scratch

<https://scratch.mit.edu/>



The screenshot shows the Scratch website homepage. At the top, there's a navigation bar with 'Create', 'Explore', 'Discuss', and 'Help' buttons, along with a search bar and 'Join Scratch' and 'Sign in' links. The main content area features the Scratch logo and the text 'Create stories, games, and animations. Share with others around the world.' Below this are three circular buttons: 'TRY IT OUT' with the Scratch cat, 'SEE EXAMPLES' with a blue cat, and 'JOIN SCRATCH (It's free)' with a yellow cat. A code block is shown on the right, demonstrating a sequence of actions: 'when green flag clicked', 'repeat 10 times' containing 'move 10 steps', 'change color effect by 25', 'play drum 4 for 0.2 beats', and 'say Welcome to Scratch! for 2 secs'. Below the main content is a 'Featured Projects' section with five project thumbnails: 'Grace's House' by Ambershade, 'Jump Rope Tutorial' by code-ivillinger, 'Liquid loading bar pl...' by CuteGrass, 'Boomy Games!' by TripleY, and 'Pixel Art Gallery' by Vis5.



Extracts from Scratch Curriculum Guide

<http://scratched.gse.harvard.edu/resources/scratch-curriculum-guide-draft>

It is strongly recommended that you become familiar with Scratch.

Scratch is a web-based computer programming environment that introduces children and young people to coding.

Here are some quotes from the guide (p.3).

“What is creative computing?”

Creative computing is about *creativity*. Computer science and computing-related fields have long been perceived as being disconnected from young people’s interests and values. Creative computing supports the development of personal connections to computing, by drawing upon creativity, imagination, and interests.

Creative computing is about *computing*. Many young people with access to computers participate as consumers, rather than designers or creators. Creative computing emphasizes the knowledge and practices that young people need to create the types of dynamic and interactive computational media that they enjoy in their daily lives.

Engaging in the creation of computational artifacts prepares young people for more than careers as computer scientists or as programmers. It supports young people’s development as *computational thinkers* – individuals who can draw on computational concepts, practices, and perspectives in all aspects of their lives, across disciplines and contexts.

The activities in this guide are designed to explore computational thinking *concepts* (sequence, loops, parallelism, events, conditionals, operators, data), *practices* (working iteratively and incrementally, testing and debugging, reusing and remixing, abstracting and modularizing), and *perspectives* (expressing, connecting, questioning).

What is design-based learning?

Design-based learning is an approach that emphasizes *designing* (creating things, not just using or interacting with things), *personalizing* (creating things that are personally meaningful and relevant), *collaborating* (working with others on creations), and *reflecting* (reviewing and rethinking one’s creative practices). As such, a design-based approach to learning is particularly well suited to creative computing, and forms the basis for the design of each session described in this guide.”

- **Every learner will need to set up a Scratch Account (very easy “sign-in” process via the Scratch website)**
- **Your IT system must allow unrestricted access to the Scratch site <https://scratch.mit.edu/>**
- **Or... download standalone application as outlined above**

Lesson 1 Understanding algorithms, Sprite and background creation

Learning Objectives:

- 1) Understand the uses of algorithms (flowcharts) and the shapes that are used.
- 2) Create a flowchart using allocated shapes to perform a task.
- 3) Students create their background and sprite for their game.

Resources:

Flappy bird example – [ulis.it.scratch.Lesson_1.sb](https://ulis.it/scratch/Lesson_1.sb)
Scratch Programming Language (BYOB can be used also)
Creating an algorithm worksheet
Flowchart Symbols and Description sheet
Scratch help sheets
Computer

Activities:

Starter

Students get logged on and then start to go through presentation.
Allow students to work in pairs to outline their sequence of instructions for 'Rice Krispie Cakes', can be done with anything relatable.
Emphasis must be applied to the sequence as order of instructions is important when programming.

Main

Identifying symbols used in a flowchart.
Paired activity can be used to match up shapes and definition before discussion. Then allow time for students to complete their table of the Flowchart sheet.

Demo with a student to discuss how using a flowchart can be done for a light switch scenario. (Student must be a robot and only follow instructions given)

Students create a flowchart based on a given scenario (Making a cup of tea). Outline the fact that decision will be highlighted with question marks and make sure that they are done early to avoid confusion.

Lesson Slide 13: very important

If you are able to log directly into the Scratch site, and pupils are allowed to open their own Scratch accounts, use the guidelines in slide 13. If you have to download a standalone version of Scratch, you will have to modify slide 13 to show pupils how to access the programme on your own network.

Identifying and creating background and sprites needed for game creation. This activity will provide a basis to identify competency of students using the Scratch environment. Students should look to create a sprite for their bat and select or create a background for their game.

Plenary

Question students on symbols used in a flowchart and the purpose they serve. Identify confidence using Scratch.

Pupil Assessment:

Questioning during the mini activities will be important when underlining the importance of sequencing instructions.

Students assessed on completion of their flowcharts.

Extension task for higher ability involving ways in which multiple drinks can be
By the end of the lesson students will have a scenario flowchart and created their sprite and background.

Lesson 2 Creating movement and using loops

Learning Objectives:

- 4) Identify and complete algorithm needed to construct movement for sprite.
- 5) Understand and apply movement to character to flap and fall.
- 6) Create a game over action/messages when the sprite dies.

Resources:

Flappy Bat creation file: [ulis.it.scratch.Lesson_2.sb](https://ulises.scratch.me/Lesson_2.sb)
Scratch
Sprite Movement Lesson 2 Worksheet
Moving your character Lesson2 Helpsheet
Computer

Lesson 2 HW Sheet (if required)

Activities:

Starter activity

Cover key words and run through the Scratch interface, using pupils to identify particular areas and demonstrate prior knowledge.

Give pupils time to continue creating background and sprites.

(Extension – Allow pupils to create a second stage background or character costume to get them thinking about levels in their game)

Main

Provide a brief outline of how co-ordinates can be used and the way in which positive and negative numbers affect movement.

Identifying the correct flowchart for movement up the screen, allow pupils time to complete their worksheet. Once explained on the board, allow pupils to explain the scripts on the Task 2 slide before giving time to implement in games.(Strict on time)

Try to get pupils to think of the steps needed for the game to stop, allowing them to come up with their own algorithm. Provide hints to start them off with PP. Then allow pupils to implement the game over script.

Plenary

Get pupils to save work.

Think of answers to questions on the board once logged off.

Use targeted questioning for the pupils allowing HA pupils to build on answers.

Pupil Assessment:

Pupils to provide verbal feedback on the choice of the correct flowchart during class.

Pupils will also complete a confidence assessment in terms of their use of flowcharts as well as their understanding of movement in Scratch.

Lesson 3 – Obstacle Movement

Learning Objectives:

- 1) Understand how to create a sprite to be used as an obstacle.
- 2) Use algorithm to add movement script to obstacles.
- 3) Create different obstacle costumes, including movement that move at different speeds.

This lesson may require some run over time as there are a number of tasks.

Resources:

Flappy Bat creation file [ulis.it.scratch.Lesson_3.sb](https://www.ulis.it/scratch/Lesson_3.sb)

Pupils to access their saved game file

Scratch

Variables Homework

Computer

Changing Obstacles, Moving Obstacles and Stopping game if hit Helpsheets

Activities:

Starter activity

Log on and open game.

Identifying what can be added to make the game harder than it currently is.

Pair discussion about how obstacles could be added to the game and what they would look like.

Main

Demo creating obstacles then allow pupils to add to their game. Stress importance of not creating more than one obstacle in sprite. (Strict on time.)

Get ideas from pupils, using what they learned with movement last lesson, how to set a particular position and then move the obstacle from right to left. Could get pupil to drive here. Allow time to implement.

Get pupils to create algorithm for collision with an obstacle, draw attention to the fact that it will be similar to what happened when touching the edge. Before allowing pupils to implement, move on to next PP slide and show the problem with not resetting. Demo creating a reset line sprite that will allow obstacles to go back to start.

Allow pupils time to add script to their obstacles.

Mini Assessment: Could get pupils to use mini whiteboards to explain in pairs what the three different scripts do.

Plenary

Targeted questioning covering content of lesson.

Pupil Assessment:

Pupils questioned during each part of the main activity, highlighting their understanding on their algorithms. Particular focus should be applied to the scripts before they start to add them to their game.

Lesson 4 Movement and conditions

Learning Objectives:

- 1) Identify and be able to define what a variable is.
- 2) Apply knowledge to create algorithms which use variables to keep score in their game.
- 3) Create will be able to use multiple variables to keep an on-going high score in their game.

Resources:

Pupils to access their saved game file
Scratch
Scoring Helpsheet
High Score Helpsheet
Computer

Activities:

Starter

Identifying and explaining variables that have been used in Maths and Science.
Provide pupils with explanation and definition of a variable in Computing.
Discuss in pairs ways that a variable could be incorporated.

Main

Get pupils to create the algorithm after explaining the term. The algorithm should structure how scoring should be implemented.
Pupils can discuss in pairs the algorithm needed before providing answers with teacher going through each step.

Give pupils time to implement score variable.
Extension – Allow pupils to create and use a high score variable.

Plenary

Question pupils on their understanding of variables using targeted questioning.

Pupil Assessment:

Pupils assessed on their understanding, explaining the scripts they have implemented.

Lesson 5 - Gravity

Learning Objectives:

- 1) Identify and explain what gravity is and how it affects everyday life.
- 2) Apply knowledge of variables to create gravity in Flappy Bats game.
- 3) Create extra difficulty in the game by adding difficulty buttons at start of the game. (Changes in gravity)

Resources:

Flappy Bat Creation File [ulis.it.scratch.Lesson_5.sb](https://ulises.it/scratch/Lesson_5.sb)

Pupils to access their saved game file

Scratch

An old style computer game (eg mega drive game) that shows the difference between a level that is easy and hard (Not included)

A video animal (Not included)

Computer

Activities:

Starter activity “What is gravity” “What would the world look like without it”

Discussion “How do we make the game more difficult”. Draw on image of skydivers, what happens as they are falling? Do they fall at one speed?

This leads into the fact that we need to be able to change the falling speed and that can be done using...a variable.

Main

Pupil drives to show how to create a variable called gravity.

Once done allow pupils to complete and then bring seats or come closer to the screen.

Show pupils the scripts on the board and ask them to discuss in pairs what they think the changes for each script does. Provide explanations for pupils as well as highlighting the fact they will need to edit their scripts rather than create new ones.

Allow pupils to create the scripts, but you may want to provide a short focus session for pupils that do not understand it.

Plenary

Get pupils to analyse and come up with an explanation of the script on the board. Emphasis should be applied on the effect the gravity block has on the game visually but also what it means in terms of the speed of the Sprite. This can be related to a real world example.

Pupil Assessment):

Pupils understanding would be gauged by their explanations of how they are using the variable and the scripts. Pupils would also put this into practice by creating two separate levels that have different gravity settings.

Understanding could also be shown by the inclusion of gravity or resistance variable 'off' the other with 'on'. More advanced pupils would have the variable with varying degrees of resistance or even create their own variables.

Lesson 6- Evaluation and assessment: Game Improvements

Learning Objectives:

- 1) Identify areas of improvement for your game and provide feedback on other games.
- 2) Apply improvements that will increase difficulty or playability.
- 3) Create new ways to use variables in your game.

Resources:

Pupils to access their saved game file
Scratch
Game Improvements Worksheet
Flappy Bats Evaluation Homework
Computer

Activities:

Starter

Identifying criteria already met by the pupils in terms of their targets. This can be done by getting pupils to pre-assess themselves using the mark sheet in the assessment folder with pencil. (Teacher to go over in pen.)

Main

Pupils complete their game improvement sheets, describing their problem and trying to create an algorithm for their solution.

Pupils implement game improvements. Teacher should look to provide assistance and facilitate ideas during this lesson. Get pupils to check their games with the checklists on the board.

Plenary

Pupils given time to test each other's games and give feedback based on WWW's and EBI's. These changes could be implemented before the assessment in the next lesson. Give pupils Homework sheet to complete. (Ensure answers are explained in full sentences.)

Pupil Assessment:

Feedback given on game testing. Pupil self-assessment as the start of the lesson provides pre-assessment knowledge. Pupils will also be assessed on their English skills when they submit their self-evaluations.

Lesson 7- Assessment

Learning Objectives:

- 1) Complete Flappy Bats Assessment.

Resources:

Pupils to access their saved game file
Scratch
Flappy Bats Assessment and Assessment Help sheets
Flappy Bats Assessment example (Included in PP too)
Computer

AssessmentGrid also available for marking and overall self-assessment

Activities:

Starter

Get pupils to open their game and the Assessment sheet on their computer. Run through PP highlighting what they will need to take screenshots of in their assessment work. Pupils must screenshot an image and code in relation to the question, as well as explaining an answer towards it.

Main

Allow pupils to complete assessment work. Example slide could be left on the board.

Plenary

Allow pupils to save work to be printed and handed in, or at the start of next lesson.

Pupil Assessment:

Pupils will be assessed on the content of their game as well as their understanding of the concepts and features they have incorporated.