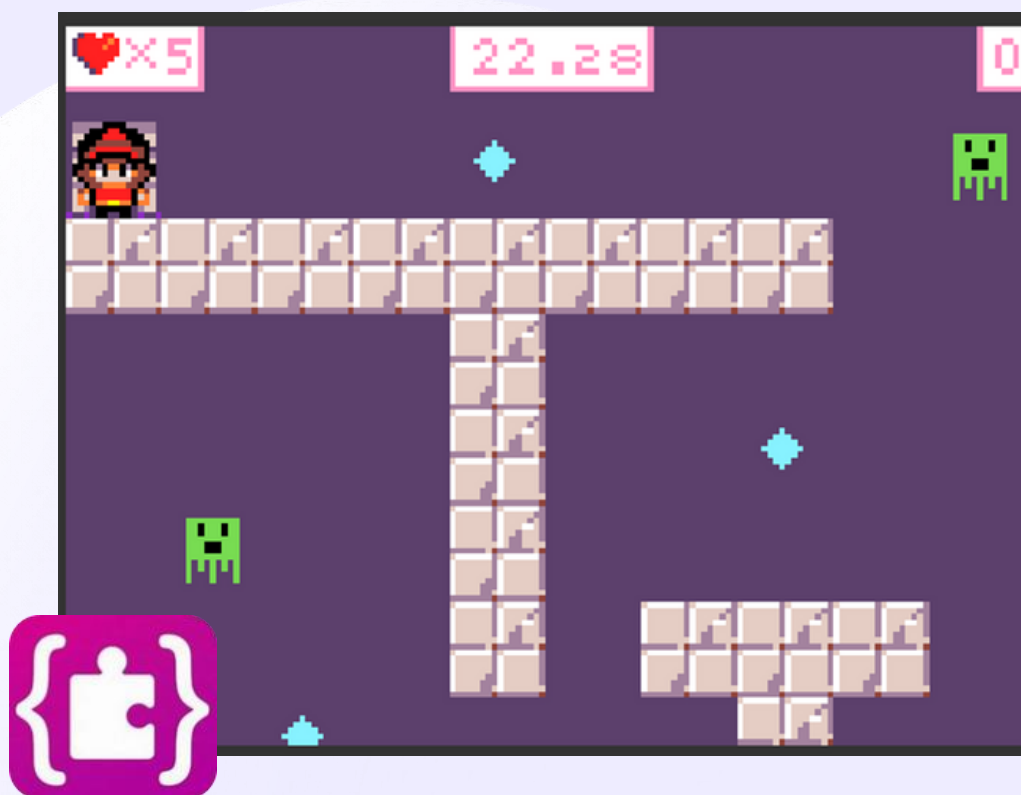


Make Code Arcade Code-Along

Maze Run Second & Third Level



Overview of the team



The Digital School – who are we?

The NL Digital School team currently consists of:

- 1 x Curriculum Manager of Digital
- 1 x Digital Education Officer
- 15 x Digital Pedagogy Practitioners



The NL Digital School team has been established for 4 school sessions and our priorities are to continue to drive forward the Council's priority of Digital First.

The aims of the Digital School have been consolidated into 3 linked areas.

- **Transformational Change** – focus on the transformation of the digital infrastructure and connectivity across the school estate. This will also include project management of devices and experiences such as interactive panels and Immersive Classrooms
- **Building Capacity** – continual development of sustainable CLPL for all staff. This will progress to the development of digital pedagogical approaches across learning, teaching and assessment and empower establishments to lead digital improvement priorities. There will be continued promotion of Digital Schools Award Scotland and Microsoft Educator.
- **Learning Experience** – linking very much to the first area of transformational change, this will focus on the devices and technologies in establishments and the impact these have on individual learners experiences. As well as continuing to develop consistent experiences and access for learners across North Lanarkshire.



Experiences and Outcomes

- I can **create, develop** and **evaluate** computing solutions in response to a design challenge. **TCH 2-15a**

Benchmarks

- **Creates** programs in a visual programming language including variables and conditional repetition.
- **Identifies** patterns in problem solving and **reuses** aspects of previous solutions appropriately for example, **reuse** code for a timer, score counter or controlling arrow keys.
- **Identifies** any mismatches between the task description and the programmed solution, and **indicates** how to fix them.





Experiences and Outcomes

- I can **select** appropriate development tools to **design, build, evaluate** and **refine** computing solutions based on requirements. TCH 3-15a

Benchmarks

- **Designs** and **builds** a program using a visual language combining constructs and using multiple variables.
- **Represents** and **manipulates** structured information in programs, or databases for example, **works** with a list data structure in a visual language, or a flat file database.
- **Interprets** a problem statement and **identifies** processes and information to **create** a physical computing and/or software solution.
- Can **find** and **correct** errors in program logic.
- **Groups** related instructions into named subprograms (in a visual language).
- **Writes** code in which there is communication between parallel processes (in a visual language).
- **Writes** code which receives and responds to real world inputs (in a visual language).
- **Designs** and **builds** web pages using appropriate mark-up languages.



Learning Outcomes



Learning Intention

I can **explore** a visual programming language to **create** an arcade game.

Benchmarks

- **Explore** a visual programming language interface (MakeCode Arcade)
- **Include** multiple variables to keep track of points and time.
- **Explain** how parallel programming is used within the scripts of code.
- **Identifies** any bugs in the blocks of code and **describes** how to **fix** the bug.

Key Learning

Learners will **build** a maze game using MakeCode Arcade within an online web browser. The learners will **explore** aspects of computing science such as **variables, algorithms, parallel programming** and **loops**.



Key Terms

- The **workspace** is where you drop blocks of code to create scripts.
- **Script** is a sequence of blocks joined together.
- **Sprites** are an object or character within your game.
- **Variables** are a changeable value that is recorded in MakeCode's memory.
- **Algorithms** are a process or set of steps that must be followed to execute a task or program.
- **Parallel programs** are when two or more scripts of code run at the same time to create a function or program.
- **Loops** are when a piece of code is repeated either a fixed number of times or forever.
- **Tiles** are part of the map that will create our maze and/or objects.
- The **tilemap** is what can be seen as the background to your game.
- **Scene blocks** allow you to change features of the scenery.
- **Info block** allow you to add in features such as a countdown timer or points system.
- **Controller blocks** allow you move your sprites around the screen.



Key Terms

- **Game blocks** allow you to add in gaming features such as game over screens.
- **Music blocks** allow you to add sounds and music to your program.
- **Loops blocks** allow you to repeat pieces of code when needed.
- The **MakeCode Arcade Simulator** is located on the left of the screen and shows the blocks of code in action.
- **Bugs** are errors within the code that stops the program working.



Important Information



Devices

Learners can use a PC, laptop or tablet to access [MakeCode Arcade](#).

Some of the actions are slightly different but these will be explained during the code-along.

Learners should be encouraged to tinker with [MakeCode Arcade](#) before the code-along to help familiarise themselves with the interface.

However, this is [not essential](#) as the interface will be explained during the code-along.

Stage

This code-along is aimed at [Second](#) and [Third Level](#) to ensure all learners have an opportunity to experience coding with [MakeCode Arcade](#).

There are also tutorials and videos available for all stages and you can create your own to suit the needs of your learners.



Are there any tutorials on MakeCode Arcade?

Yes! MakeCode Arcade has got some excellent tutorials to help you get familiar with block coding or Python and Java Script coding. Most tutorials are step by step and give you hints for what you do next or they will include a video to follow along with. Click this [link](#) to see what they have on offer.

Can I create my own MakeCode tutorials?

Yes! MakeCode provides you with a platform to create your code using blocks and add in text descriptors that explains what code you need and what you should do next. Click on this [link](#) to start creating your own tutorials.

Do learners need a login?

No! Learners do not need a login to take part in this code-along. If they wish to save their projects then, yes, they will need to login using their Glow account and this will save in their own MakeCode Cloud Space.

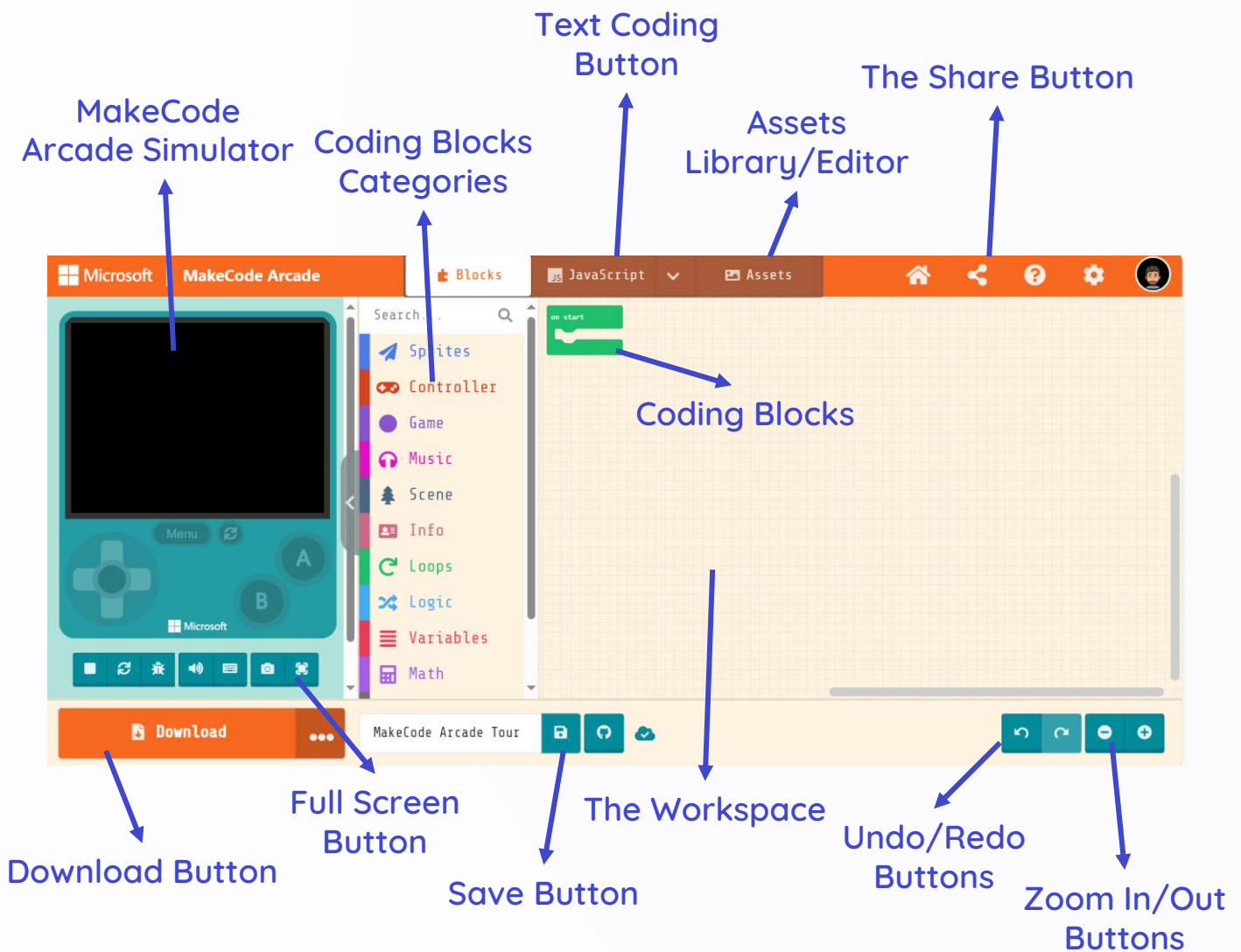
Where can I find more videos on MakeCode Arcade?

YouTube! MakeCode Arcade has their very own channel dedicated to the editor and full of useful on how to create games or even just an overview of what you can do with MakeCode Arcade. Click this [link](#) to explore their content!



Learning More About MakeCode Arcade

Below is an annotated screenshot from the MakeCode Arcade Editor that you will use in this code-along. The image highlights the main areas of the platform that you may use as part of the code-along.



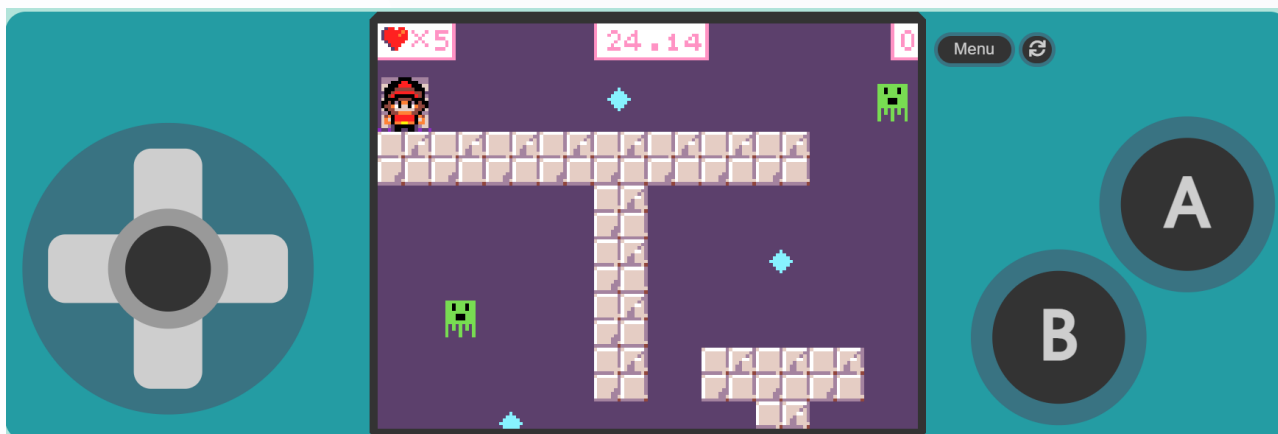
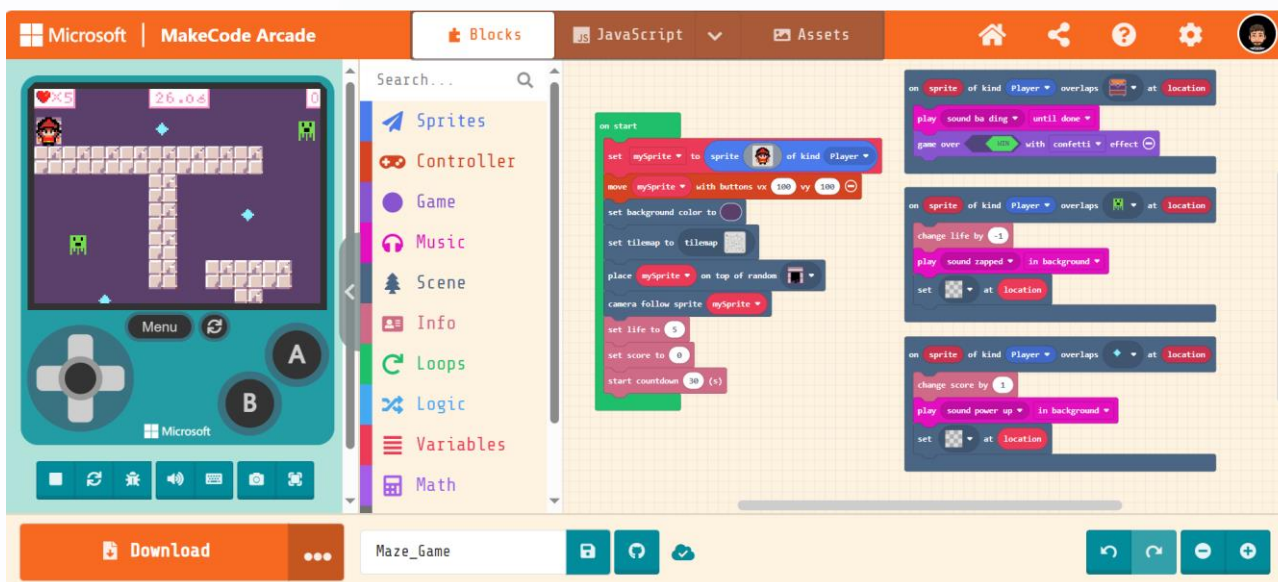


Learning Outcomes



What will learners make?

Below is a screenshot from MakeCode Arcade to show an example of what the learners will be making. Learners will **build** a maze game and will **explore** aspects of computing science such as **variables**, **algorithms**, **parallel programming** and **loops**.






SCRIPT 1: Creating the Sprite


Welcome to the first part of today's code-along! This script of code will allow learners to select a sprite then code it to move when the game begins.

Insert the exact code that you see below to the MakeCode Arcade workspace. You can decide what kind of sprite you'd like to choose from the gallery.

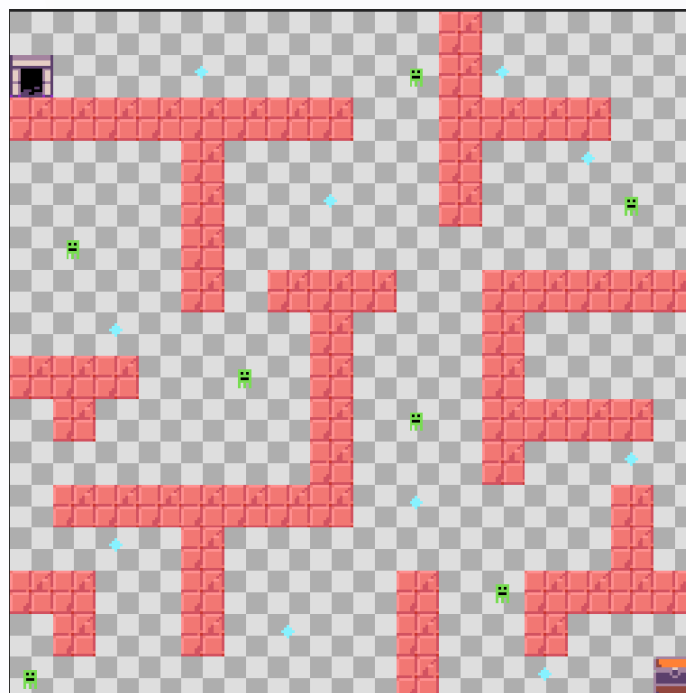
```
on start
  set mySprite to sprite  of kind Player
  move mySprite with buttons vx 100 vy 100
```



SCRIPT 2: Creating the Maze TileMap

Click on the 'Assets' button at the top of the workspace. Click the plus button and choose 'Tilemap'. Now we will design the map for our maze. Once you have added the tiles to create the shape of your maze you need to make sure your sprite can't walk through the tiles and turn them in to walls using the  button.

You can decide what shape your maze is and the looks of your tiles. In this example I have used the dungeon tiles category and chose a tile that looks like a wall. Remember to add in a 'start' and 'end' tile to the maze.

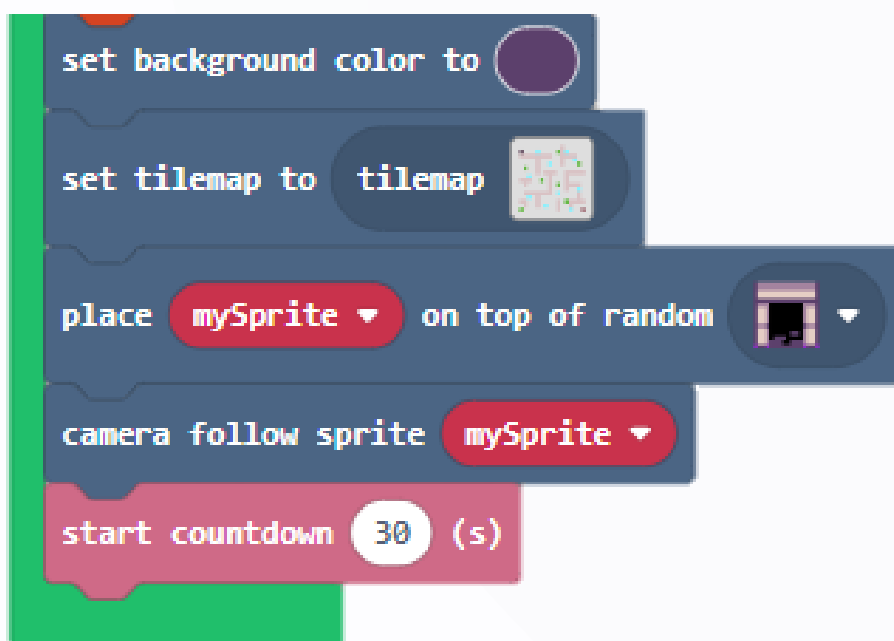




SCRIPT 2B: Coding the Maze Tilemap

In this second piece of script we will code the background and scenery of our maze game. We will begin by adding in the background colour and set the tilemap to the maze we have just created. Then we will tell our sprite to start on the door tile and ensure the birds eye view camera is following our sprite as it travels through the maze. Finally, insert the start countdown block from the information category and change the value to 30 seconds.

Insert the exact code that you see below to the MakeCode Arcade workspace. You can decide what kind of door you would like to use from the tile gallery.





SCRIPT 3: Parallel Program and Sensing - Part 1

In this script we will tell the program what it should do when the player sprite touches or senses specific tiles on the maze.

Add the 'sprite overlaps' block from the scene category and change the tile image dropdown to the open chest tile. Once the sprite touches the chest, the program should play the sound 'ba ding' and show that the game is over with a confetti effect in the background.

Insert the exact code that you see below to the MakeCode Arcade workspace. You can decide what kind of sprite you'd like to choose from the gallery and what sound you would like it to make.

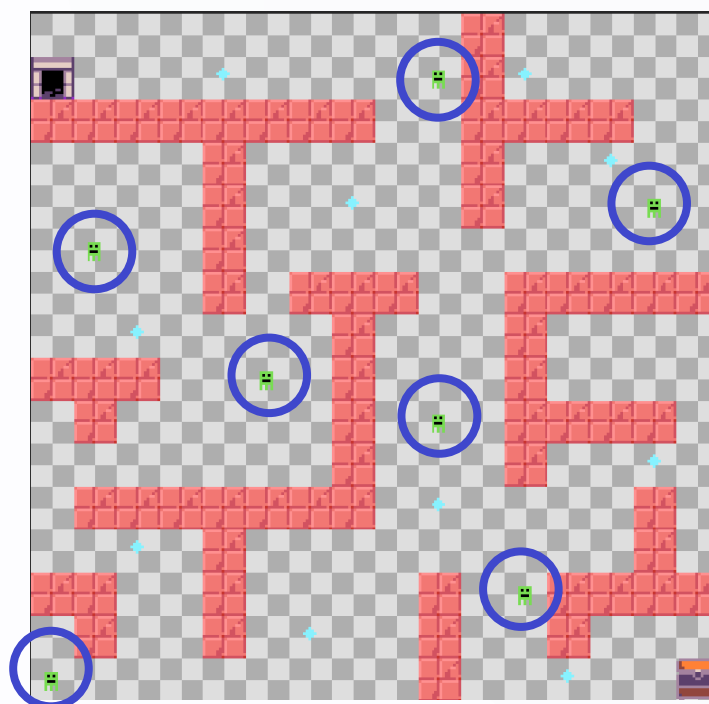




SCRIPT 4A: Creating and placing Monster tiles

Click on the 'Assets' button at the top of the workspace. Click the plus button and choose 'Tile'. Now we will design the Monster for our maze. Once you have designed your Monster you can now add the tiles to your tilemap. Go to Assets and edit the tilemap. Click on the 'My Tiles' section and select the Monster tile. Then you can place the Monsters anywhere on your maze except on the walls.

You can add as many or as little monsters as you want to make your game easier or more challenging. Make sure to add the same or more monsters than you have given lives e.g. If you have given your sprite 3 lives you must include at least 3 monsters.



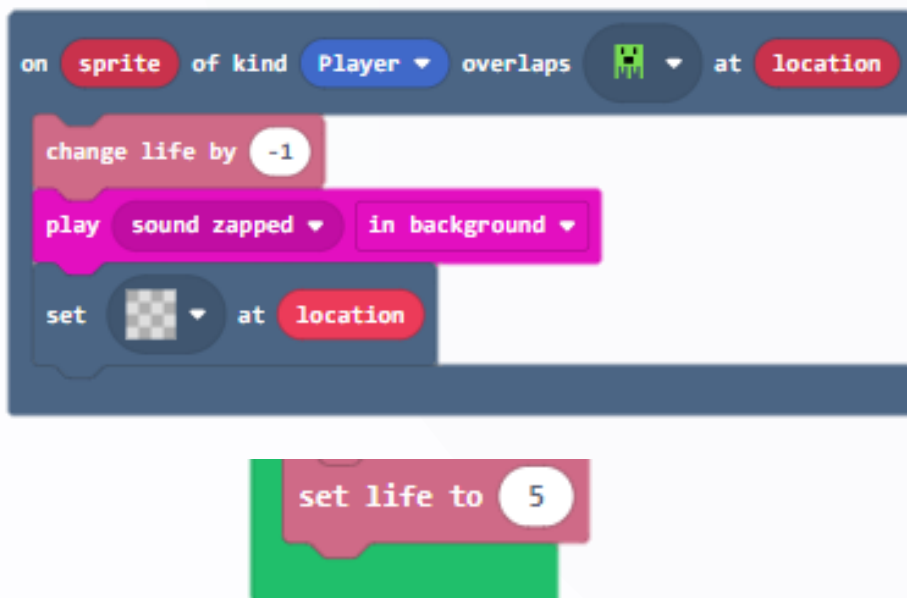


SCRIPT 4B: Create Monster, Parallel Program and Sensing

This script is similar to the one before, as it uses the same block to start the piece of code.

Add the 'sprite overlaps block' from the scene category and change the tile image dropdown to the Monster tile. Once the sprite touches the Monster the program should take away a life, play the sound 'zapped' and the location where the Monster was placed should go blank. Finally, add the set life to block from the information category and place it at the bottom of the green 'On Start' script and change the lives to 5.

Insert the exact code that you see below to the MakeCode Arcade workspace. You can decide what kind of sprite you want to create and what sound you would like it to make.



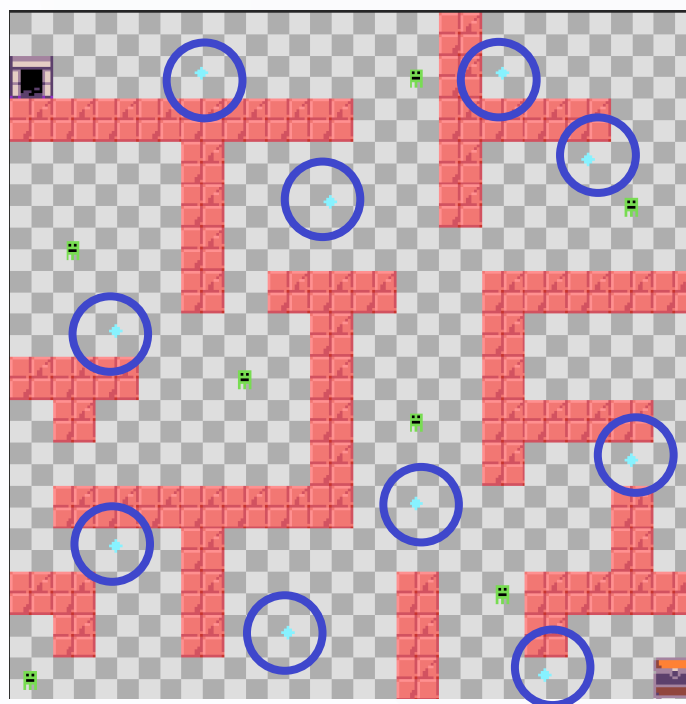


SCRIPT 5A: Create and place Crystal tiles

Click on the 'Assets' button at the top of the workspace. Click the plus button and choose 'Tile'. Now we will design the Crystal for our maze. Once you have designed your Crystal you can now add the tiles to your tilemap.

Go to assets and edit the tilemap. Click on the 'My Tiles' section and select the Crystal tile. Then you can place the Crystals anywhere on your maze except on the walls.

You can add as many or as little crystals as you want to your maze game. Challenge your peers to gain as many points and escape the maze in thirty seconds.



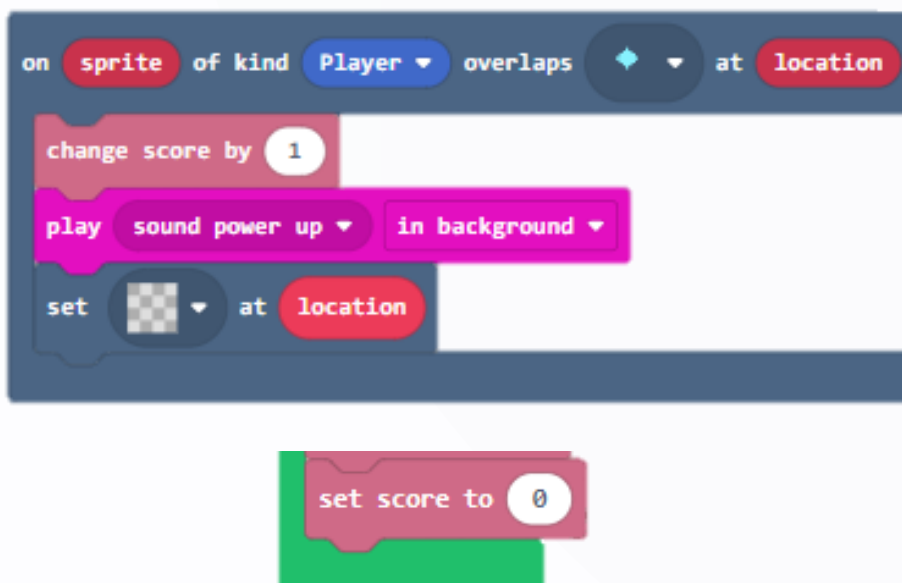


SCRIPT 5B: Create Crystals, Parallel Program and Sensing

This script is similar to the one before as it uses the same block to start the piece of code.

Add the 'sprite overlaps block' from the scene category and change the tile image dropdown to the Crystal tile. When the sprite touches the Crystal, the program should add a point, play the sound 'power up' and the location where the Enemy was placed should go blank. Finally, add the set score to block from the information category and place it at the bottom of the green 'On Start' script and make sure the score is set to zero.

Insert the exact code that you see below to the MakeCode Arcade workspace. You can decide what kind of sprite you want to create and what sound you would like it to make.





Finished!



Congratulations!

You have now completed one of the Education Scotland Computing Science Week Create-alongs!

We hope you had a brilliant time and are inspired to keep coding and creating some fun games.

Tech Issues?

You can follow along by coding along with the step by step tutorial created for this project.

Step-by-step Video Tutorial: [Maze Run Create-Along Recorded Video.mp4](#)

Remember to share how your class get on with the code-along and share their success by posting online and tag us [@NLDigitalSchool](#) and [@digilearnscot](#).