code-it.co.uk

No cost subscription for new ideas

Pathway Planning

Create a simple game where children program a character to meet their friend. If they bump into the obstacles they have to go back to the start. If they reach their friend they are greeted.

Multiple Approaches

In this planning I will outline lots of different approaches to using this with your pupils so you can choose the one that fits best with your learning philosophy.

Approach 1

Just play the game

The simplest way to use this is for the teacher to make their own version without the directional arrows.

Set it up on a tablet computer and encourage pupils to design their own pathway for the friends to meet. When a pupil has solved it they can move the obstacles around so there is a different challenge for their partner

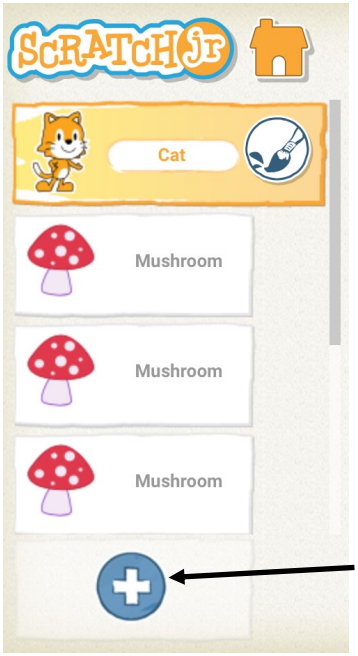
This approach uses simple sequence using only the most basic up down right and left arrows.

You can watch me play this to see how to do this here.

https://youtu.be/Aw9zqIlgrDc

A really good way to introduce this would be to play human robots first. Set up some chairs as obstacles and use cones or bean bags to show the start and finish points. Print out some Scratch Jr direction cards and pupils can design an algorithm from start to finish by putting the cards on the floor.

Encourage pupils to document and reflect on their learning taking photos and talking about their routes and bugs (things that didn't work) using programs like Book Creator or My Story.

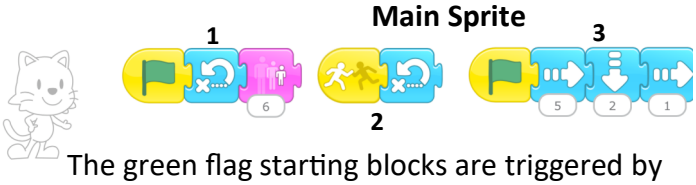


Create the Game

Create two main characters and some obstacle sprites

Click + to add characters

Drag these programming blocks into each character/sprite



The green flag starting blocks are triggered by the green flag at the top of the game.

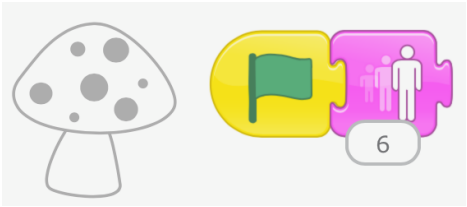
The first sequence sends the cat back to the start and then shrinks the sprite.

The second sequence is only triggered if the sprite touches another sprite.

The third sequence would be left blank so the user can add direction blocks and test their program through the green flag at the top.

End Sprite

The colliding people means the say block will only start if this end sprite is touched by another sprite



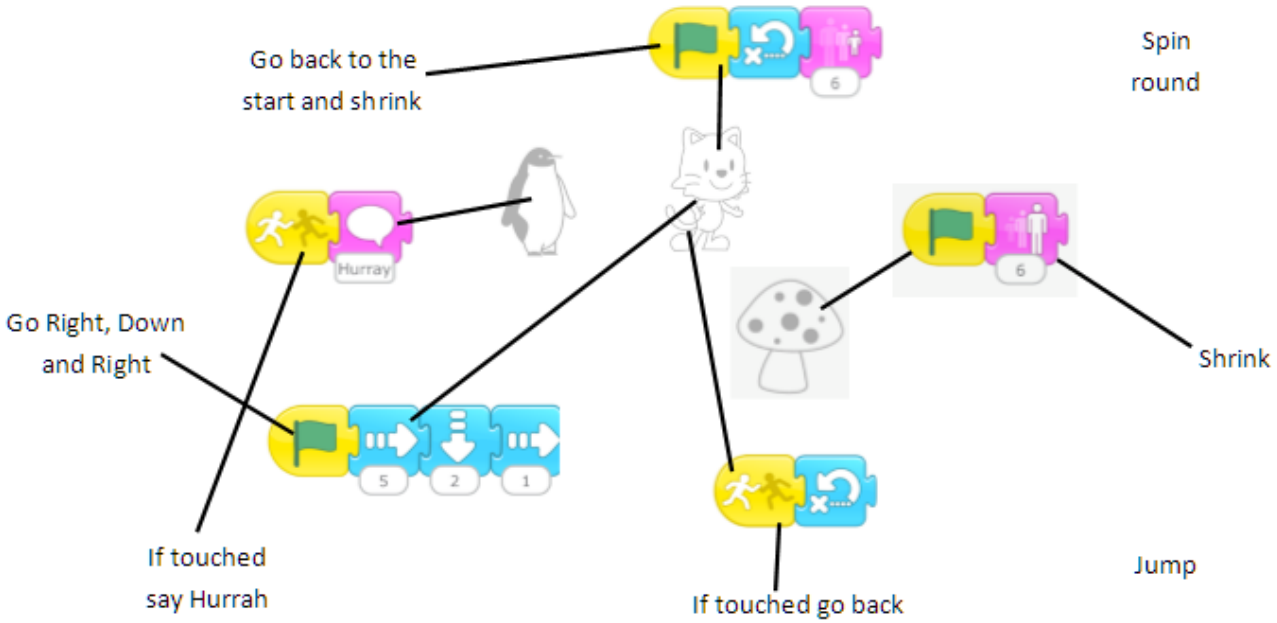
Obstacle Sprite

The sprite is shrunk when the game is started through the green flag.

Approach 2

Show pupils a working copy of the game and let them puzzle out what each set of code instructions does before making their own version of the game.

They can use the decomposed planner to record what they think the game does. Drawing lines from the code blocks to the task they think it does. The teacher can check their understanding first (marking the planner) before allowing them to make their own version.



Decomposed Planner Answers

Approach 3

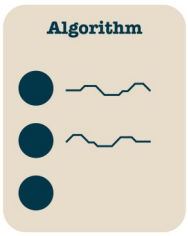
Play the game on your interactive whiteboard or touch screen TV with your class. List the three main sprites. Ask pupils to explain what each sprite needs to do and list their answers as bullet points next to each sprite. Show them some of the code blocks that achieve this. Allow them to go off and make their own version. Will they use the same sprites? Encourage them to come back and check your version for a solution if they are stuck or use the hint card.

code-it.co.uk

No cost subscription for new ideas

Computational Thinking Skills used

Decomposition to break up the project into parts



Algorithm the steps or rules to accomplish a goal

Generalisation to adapt and idea and use it to create a modified project

Problem Solving Skills your pupils are using

Complexity	I can break complex problems into parts	I can discover / concentrate on the most important part of a problem	I can explain how I used decomposition & abstraction
Ambiguity	I recognise there is more than one way to solve a problem	I recognise there is more than one way to describe a problem	I can explain how I managed ambiguity
Open Ended	I look for a range of solution to the same problem	I don't just accept the first solution	I can describe how a project can be extended
Adapt	I can adapt existing ideas to solve new problems	I can identify patterns in problems & solutions	I can explain how I adapted a solution to solve a new problem
Evaluate	I can evaluate my solutions against a set criteria	I can design criteria to evaluate my creations	I can explain how evaluation helped me improve a project
Experiment & Debug	I can develop, test and debug until a product is refined	I repeatedly experiment through making, testing & debugging	I can explain how using the iterative cycle improves my work
Persist	I can persevere even if the solution is not obvious	I learn from setbacks and don't let them put me off	I can describe how I overcame problems
Communicate	I can contribute useful ideas to a partner or group	I can encourage others to share their ideas	I can lead using all the people talent in my group

Decomposition Abstraction Generalisation Algorithmic Evaluation Algorithm

Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/ mentioned with permission