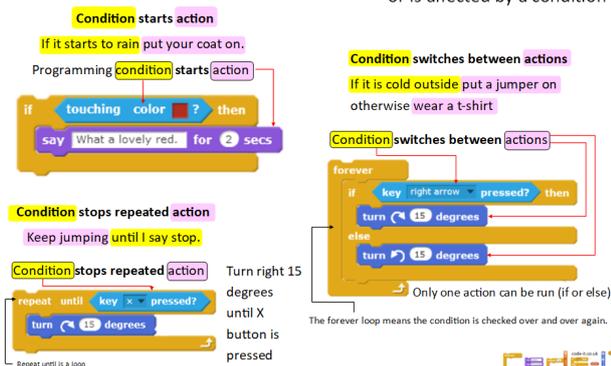# Adventure Game Part 1

Pupils work in small groups to create a linear adventure game. See section 2 for a longer discussion around theme. They splice their code together to form a larger narrative. They use conditional selection that they learnt about in the Maths quiz independently whilst being introduced to making more than two choices as new learning.

1. Revise basic types of conditional selection

2. Introduce the idea of writing a collective adventure game

3. Can pupils plan simple adventure type questions for all three types of conditions?

4. Ask pupils to think if there might be anything else they might like conditional selection to do?

5. Explore how conditional selection can have more than two outcomes

6. Pupils plan an adventure game in pairs or small groups, they choose whether to use conditional selection scaffolds or not

7. Additional design elements they may wish to include

8. Pupils program individually sharing their planning with other group members

9. Pupils test and refine their adventure games

10. Teachers may wish to move onto introducing adventure points or lives to the game using Part 2

Program Aim

Design and Create an Adventure Game. Use conditional selection independently for a purpose.

Computer Science Concepts

-condition starts action
-condition stops repeated action
-Condition switches between multiple actions
-Algorithm to code
-User typed input into program

National Curriculum Programs of Study

Pupils should be taught to:

design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

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

Conditional Selection

Programming that starts or is affected by a condition

Condition starts **action**
If it starts to rain put your coat on.
Programming condition starts action

```
if  touching color [ ] ? then
    say What a lovely red. for 2 secs
```

Condition stops repeated **action**
Keep jumping until I say stop.
Condition stops repeated action

```
repeat until key x pressed?
    turn ↻ 15 degrees
```
Repeat until is a loop

Condition switches between **actions**
If it is cold outside put a jumper on otherwise wear a t-shirt
Condition switches between actions

```
forever
    if key right arrow pressed? then
        turn ↻ 15 degrees
    else
        turn ↺ 15 degrees
```
Only one action can be run (if or else)
The forever loop means the condition is checked over and over again.

Turn right 15 degrees until X button is pressed

Things to do

• Print out a copy of conditional selection fact sheet shown on the left.

• Download the adventure game PowerPoint slides.

• Print out and cut up copies of the scaffold PDF

• Check pupils can save to a common area which they can all access.

Main methods of conditional selection learnt in the Maths quiz

Condition starts action

Condition switches between actions

Condition stops continuous loop.
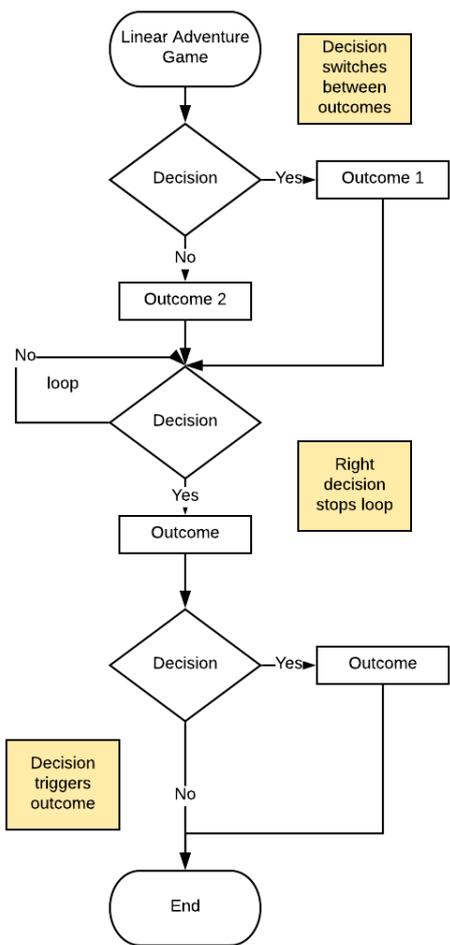
Linear adventure games only have one main path which means that the player gets every decision rather than miss some due to their choices.

- Decide on a theme or themes see section 2

1. Revise basic conditional selection

Refer pupils back to the conditional selection information sheets that you used with the maths quiz. Spend five minutes roleplaying conditional selection every day examples. Can pupils identify the three main methods they used in the Maths quiz?

2. Introduce the idea of writing a collective adventure game

In the past adventure games were typically dungeons and dragons type adventures. A modern adventure game need not use this genre. It could be any set of questions around a common theme that engages the adventurer.

- How to survive a school prom

- Choices in a football match

- Fictional choices that might be made by a favourite book character

- Choices presented to a historical character

Traditionally an adventure book/game would create many workflow branches as the reader/player made decisions that affected the outcome of the book/game. To write this type of adventure is possible within Scratch but it would greatly increase the time and complexity of the project. If however, a linear approach is taken where every choice is available one after another then the project is simplified and made manageable.

It is also possible to create project teams of writers who all create a few questions each and combine them into one bigger end projects. As long as the theme and demarcation of questions are agreed in the beginning of the project.

Consider the school prom idea. A team of four pupils might agree to each write two questions each. One writer concentrating on the perils of getting ready for the big event another on arriving and finding their friends whilst another concentrates on dancing and having fun.

Whatever you choose it is going to be up to you to enthuse your pupils about the genre and groupings. Why not create a quick example algorithm yourself using a chosen genre.

Choice is very important to pupils and a range of choices will enable greater participation and interest. Typically I might prepare three possible topics but allow pupils to suggest oth-

er ideas that fit in with topics studied.

3. Can pupils plan simple adventure type questions for all three types of conditions?

Use the Prom Example PowerPoint slides to revise different uses of conditional selection. After each type there is an opportunity for pupils to write their own example. You could use the printed and cut up scaffolds for this, pupils small whiteboards, Word or Publisher or lined paper. This is a great chance for pupils to demonstrate that they understood the three simple conditional types introduced in the Maths quiz.

- Condition starts action

- Condition switches between two actions

- Condition stops a repeated action

Sharing some of these will also give pupils more ideas that they can adapt during the planning phase.

You may wish to rewrite the Prom example PowerPoint to use examples from your chosen genre if you are limiting pupils to a single chosen theme.

4. Ask pupils to think if there might be anything else they might like conditional selection to do?

This allows pupils to think carefully and deeply about the limitations of the three conditions types that you have introduced so far.

Pupils can discuss in pairs before sharing to the class.

Whilst there are lots of additional functions that can be associated with conditional selection such as conditions that respond to greater than or less than rather than equals to or multiple conditions that need to be met. The most likely useful addition would be being able to choose three or more possible outcomes.

5. Explore how conditional selection can have more than two outcomes

If your pupils suggest this or not introduce the multiple outcome condition using the multiple outcome slides on the PowerPoint. As this is very much just an extension of previous work leaving these examples up where they can be observed is normally enough to encourage

Ask Should you get your hair done? Yes or no
If answer = no
          say Everyone notices that you haven't even combed your hair

Ask _____
If answer = _____
          say _____

Write your own

ask something that needs a choice (type choice word or number) and wait
if   answer = choice word or number   then
      say something or do something for 5 secs

NOTE

If using a condition stops repeated action then you must be careful that the number or string that triggers it to stop repeating is not the same as the previous question. If it is then the fixed answer variable will already be set to this value and the loop will be stopped before the question has been asked.

Fixed answer variable ⟶ answer

Ask travel to prom by walking, car, or horse
If answer = walking
          say Everyone will think you are so eco friendly
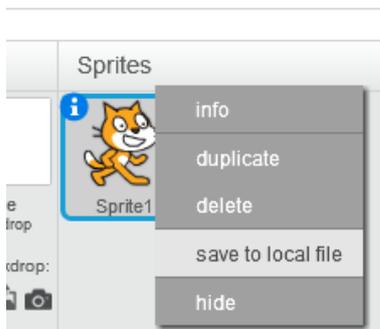If answer = car
          say Cars are so last year
If answer = horse
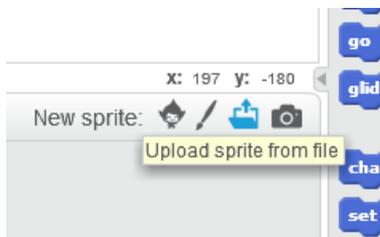   say Everyone who loves horses will be so jealous

ask something that needs a choice (type choice 1, 2 or 3) and wait
if   answer = 1   then
      say something or do something for 5 secs

if   answer = 2   then
      say something or do something different for 5 secs

if   answer = 3   then
      say something or do something totally different for 5 secs
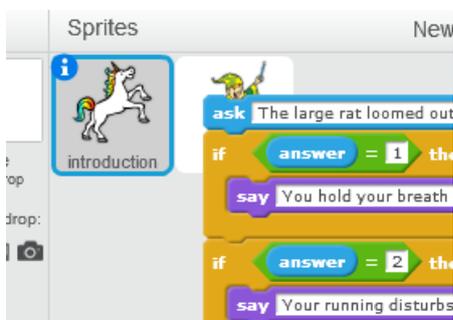
some pupils to use this within their project.

6, Pupils plan an adventure game in pairs or small groups, they choose whether to use conditional selection scaffolds or not

Pupils then need time to divide up parts of their adventure if they have chosen to create a group adventure and plan. Every pupil must create an algorithm for their questions. They may choose to use the algorithmic scaffolds or write these without. The algorithm is a plan and not the final product so speed and abbreviations are more useful that perfect handwriting and meticulous indentations.

The support cards that include algorithmic examples are a useful independent support.

7, Additional design elements they may wish to include

Alongside algorithmic design it can be very useful to encourage a column to show what the sprite will look like or stage background will show. A single ruled line demarcating a quarter of the page helps to encourage pupils to think about this aspect of appearance design. If these aspects change, pupils will need to include how they change in the algorithmic design space.

8, Pupils program individually sharing their planning with other group members

Once pupils have turned their algorithm designs into Scratch programming for their part of the group project they will need to assemble these and the questions from the rest of their group in one Scratch program. Fortunately Scratch characters can be exported by right mouse clicking on the sprite with the code and saving to local file as shown on the left.

Most school networks have a collaborative drives where pupils can exchange files with other pupils and if yours doesn't then your network provider has advised you badly. Fortunately it is not a difficult job to remedy. Pupils need to save their sprites to a collaborative folder so that they can all use them. It will help if they have named their sprite after the section they were working on such as introduction or end.

Each pupil can then import the different sections of code in separate sprites and drag the code sections on top of their chosen sprite before combining them to create their own version of the adventure.

8, Sharing their planning with other group members

After creating their code section each pupil saves their sprite code into a shared folder

Each pupil then uploads the sprites from other members of the group

Physically drag code you want to use from the sprites that you have imported onto the sprite you are using as narrator. You can then assemble this as needed.

9, Pupils test and refine their adventure game

If pupils are working in groups to create versions of a larger combined project then time to test each others versions and feed back on ways these can be improved is time well spent.

Primary teachers should understand this as similar to the drafting process in literacy.

WAGGOL What does a good one of these look like, can be a great question to ask before pupils embark on this phase.

Don't forget to give pupils plenty of time to make changes after peer and teacher feedback.

10, Teachers may wish to move onto introducing adventure points or lives to the game using the Adventure Game Part 2

See the adventure game planning part 2 for this aspect.

This planning is Draft 1

I welcome feedback on the usefulness of spending more time teaching pupils about conditional selection before using it within a project such as this.

Any errors or mistakes

Any ways you adapted this to fit your curriculum

You are welcome to adapt anything for your school but do contact me if you wish to republish in any form as this publishing is protected by a creative commons licence. I only do this as so much of my work has been copied and republished without permission or recognition. http://code-it.co.uk/contact/

I recognise there is more than one way to solve/describe a problem

I don't just accept the first solution

I look for a range of solution to the same problem

I can evaluate my solutions against a set criteria

Handles Ambiguity

Open Ended Problem Solver

I look for how a project can be extended

I can break complex problems into parts

I can design criteria to evaluate my creations

Evaluates

Insert picture of your students here

Copes with Complexity

I can discover / concentrate on the most important part of a problem

I can contribute useful ideas to a partner or group

I can identify patterns in problems & solutions

I can encourage others to share their ideas

Communicates

Adapts

I can adapt existing ideas to solve new problems

I lead using all the people talent in my group

Investigates

I can develop, test and debug until a product is refined

I learn from setbacks and don't let them put me off

Perseveres

I make predictions about what will happen

I can persevere even if the solution is not obvious

I repeatedly experiment through predicting, making, testing & debugging

@baggiepr

Code-IT

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

Problem Solving Attitudes and Skills used in this project