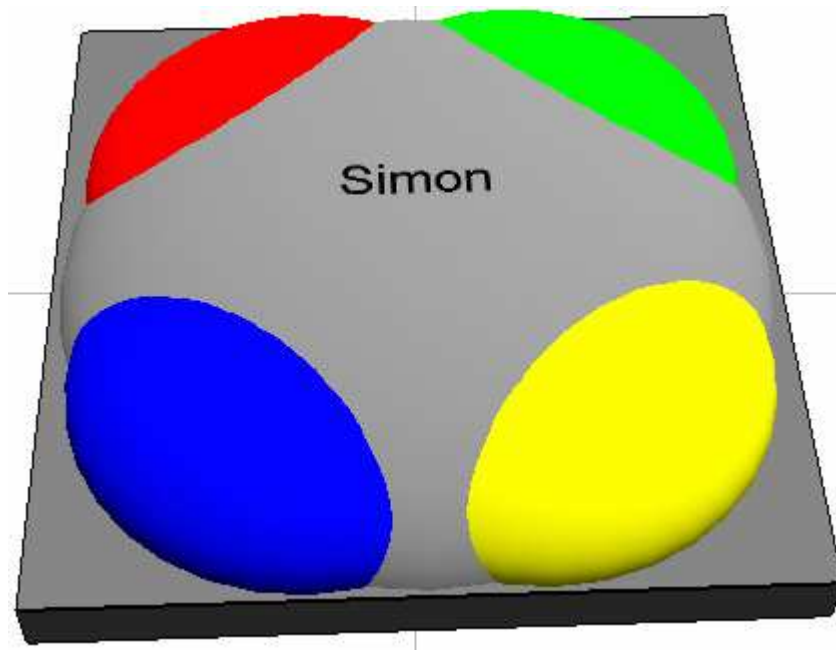


## **Activity: Creating a Simon game**

Create a downloadable program which behaves as a Simon game.

The program should be capable of running as a game in simulation but only use pin connections so it will also download OK.



## **Requirements**

Simon is a game of memory skill for one player.

The game should have four coloured buttons, coloured red, green, blue and yellow. Each turn a sequence of the lights are lit up and the player has to click the same lights that the computer lit up.

If the player fails to get the correct order, the game is over. If the player presses the buttons in the correct order then the sequence increases by one random light and the game repeats.

## ***What you will need to do***

### **Planning out the component: 15 mins**

- You will want to create a rough outline of the project with a top level breakdown
- What are the stages to progress through in your project?
- Do not go into too much detail – you will just end up changing this later on!
- Do you want to sketch a flowchart in rough detail before you attempt the programming?

### **Lay out the components on the panel: 10-15 mins**

- Prioritise! Do the components need to look perfect before you start your program?
- Is the look of the game in simulation important?
- Remember you will be referencing the shapes you add in code, so give them meaningful names.

### **Using pins to create a downloadable components: 10 mins**

- If all your program uses are downloadable parts, the chances are it will also work on download.
- Use events to trigger simulation-side activities to help debug your program.

### **Clicks and animating the component using the button helper: 5 mins**

- Some components are designed to help you in simulation.
- The Button Helper component will make any object behave like a button.
- This helps during simulation without breaking your program on download.

### **The game logic: 50 mins**

- Above all be methodical!
- Take your time, keep calm when things go wrong, and follow the hints to writing a program below.

## ***Hints to writing the program***

However you choose to work, remember that the key to a good program is clarity. If the intention of your program is clear it is easy to understand what is happening and why, and easier to spot bugs.

Here are a few hints on writing the Simon game as your first big Flowcode project. Remember this is a guide, and you should feel free to design your program in the way you feel most comfortable.

- Use an array to remember the colours
  - Keep a counter to remember the number of items in memory
  - Keep a counter to remember the number of items the player has clicked on
- Use events to keep simulation-side activities separate from downloadable ones.
  - Events will help you structure your program so specific activities such as reacting to pin changes are kept together and out of the way.
- Keep connected pieces of code in macros
  - A macro should fulfil a single task – do not try to cram too much into a macro, as this defeats the object of grouping the code together.
- Plan and adapt as you go
  - Planning everything at once is hard – it is easier to get a rough outline of your program then adapt it as you go.
  - Prepare for brick walls – writing a program is like finding your way out of a maze – there will be dead ends, but you can always turn back!
  - Some developers like to write comments as to what to do at each point, then follow these comments one at a time like an instruction book to fill in the program.
- Keep to a standard
  - Whatever your style of programming, be consistent! Do you use capital letters to denote constants? Then make sure this is always the case!

- Work methodically
  - Plan your work out – what should you do first?
  - Do not try to do everything at once! Work in stages – once you are confident part of the program works, move on to the next part.
  - Work in a way that builds your confidence. Writing a program that makes you feel more and more uneasy and out of control as you progress is a sure-fire way to create problems!
- Expect problems!
  - Experienced developers feel uneasy when their programs work first time!
  - Do not panic if things do not work. Instead, find out what does work and break down the program systematically to corner any bugs.