Year 5 level

Crusher tasks
- see manuals, begins with a blank program and introduces *actor-lab*
- very detailed/prescriptive to begin with

Crusher- Task 5
- uses optical sensor to make the crusher safer

three lights
- demo of how **wait for** is used to serialise in *actor-lab*
- intuitive (procedural) reading of the syntax is that the waits are cumulative
- ie that in the 'no delay' actor that the third light will go on 4 secs after the second
- but wrong, each wait is timed from the actor's **begin** event, they go on together

lighthouse
- more complex demo of the use of **wait for**
- the idea of modular programs, so that you can change the flashing pattern easily

wake-sleep
- simple demo of task prioritisation using **wake** and **sleep** messages

Crusher- Task 6
- uses **wake** and **sleep** to make the machine tamper proof

Year 6 level

Car Park- part 1
- simple use of sequencing to control barrier arm and light

Car Park- part 3
- using the sensors 'going high' to detect the car having passed

Car Park- part 4
- using the counter (variable) to limit access

maze buggy drive
- drive the buggy through the maze to understand how it works

maze buggy program
- program the buggy to pass through the maze

Year 7 level

line buggy

- simple program to allow the buggy to follow the line

line-buggy version 2

- using the sensors to send both 'on line' and 'off line' messages

line buggy-variable
- example of the use of counters (variables) to change the speed of motors
- and change the turning angle

line buggy foil
- buggy turns round when it crosses the foil
- example of task prioritising

counter
- demo example of actors sending messages to themselves, and to other actors
- useful to show before attempting stopwatch programs

timer
- a complex stopwatch with single stop/start button and trip

Technical

sensor AND'ing
- using switches connected to analogue inputs and streaming to a counter
- useful example of how with slow buggies an 'instant' ≈ 1 second

cascade
- concurrency demo

cascade-depth
- concurrency demo

repeat loop- variable
- dynamically changing the loop speed and length
- the sort of thing you might attempt in a time-based language