

Comment: initialisation of the model

set variable `map` as dictionary

set variable `agents` as list

for 10 times

set variable `agent` as dictionary

set variable `agent.colour` as blue or red at random

set variable `agent.co-ordinate x` as random number between 0 and 10

set variable `agent.co-ordinate y` as random number between 0 and 10

Comment: here we should check that there is space at co-ordinates x, y on the map; if not, we re-randomise them

add `agent` to `agents`

set `agent` to `map` with key `agent.co-ordinate x`, `agent.co-ordinate y`



Comment: running the simulation

for 1000 times

for each `agent` in `agents`

Comment: rules for an individual unit

set variable `own colour` as `agent.colour`

set variable `different colour gatherer` as 0

Comment: check left, right, up, and down from the current agent

set variable `neighbours` as an empty list

add variable `map` from co-ordinates `agent.co-ordinate x` - 1, `agent.co-ordinate y` to `neighbours`

add variable `map` from co-ordinates `agent.co-ordinate x` + 1, `agent.co-ordinate y` to `neighbours`

add variable `map` from co-ordinates `agent.co-ordinate x`, `agent.co-ordinate y` - 1 to `neighbours`

add variable `map` from co-ordinates `agent.co-ordinate x`, `agent.co-ordinate y` + 1 to `neighbours`

for each `neighbour` in `neighbours`

if `neighbour.colour` != `own colour`

set variable `different colour gatherer` as `different colour gatherer` + 1



if `different colour gatherer` > 2

set empty to `map` with key `agent.co-ordinate x`, `agent.co-ordinate y`

set variable `agent.co-ordinate x` as random number between 0 and 10

set variable `agent.co-ordinate y` as random number between 0 and 10

Comment: here we should check that there is space at co-ordinates x, y on the map; if not, we re-randomise them

set `agent` to `map` with key `agent.co-ordinate x`, `agent.co-ordinate y`

