

```

1 ;each tick is representative of 1 hour
2
3 ;6264 ticks = 261 days
4 ;model only models working days Mon-Fri (footfall in the City dramatically reduces
  on weekends)
5
6 ;singular and plural
7 ;TO NOTE: bicycles and bikes are used interchangeably
8 breed [bikes bike]
9 breed [thieves thief]
10 breed [policeofficer police]
11
12 ;global variables that all agents own
13 globals[
14   count-total
15   enforce?
16   hide?
17   police-thief?
18   police-bike?
19   bikemarking?
20   removeparts?
21   recovered
22 ]
23
24 ;variables that all patches own
25 patches-own[
26   accessible?
27   q-val-north
28   q-val-south
29   q-val-east
30   q-val-west
31 ]
32
33 ;variables that thief agents own
34 thieves-own[
35   crime-probability
36   birth-tick
37   hidden?
38 ]
39
40 ;variables that bicycles own
41 bikes-own[
42   desirability
43   security
44   stolen?
45   birth-tick
46   show?
47   marked?
48   advised?
49   remove?
50 ]
51
52 ;creates the map of the City of London
53 to init-map
54   import-pcolors "./images/london.png"
55   ask patches[
56     ;set all patches in the image (grid space) as accessible
57     set accessible? true
58     ;patches that are pinkish in the image are buildings and therefore are
  inaccessible
59     if pcolor = 28.6 [set accessible? false]
60   ]

```

```

61 end
62
63 ;sets bicycles in the model
64 to setup-bikes
65   create-bikes ratio-bikes * population-size [
66     spawn-bike
67   ]
68 ]
69 end
70
71 ;sets thieves in the model
72 to setup-thieves
73   create-thieves ratio-thieves * population-size [
74     spawn-thief
75   ]
76 end
77
78 ;sets police in the model
79 to setup-policeofficer
80   create-policeofficer ratio-policeofficer * population-size / 40[
81     spawn-police
82   ]
83 end
84
85 ;setup procedure that is called once the setup button is clicked
86 to setup
87   __clear-all-and-reset-ticks
88   init-map
89   setup-bikes
90   setup-thieves
91   setup-policeofficer
92   set hide? false
93 end
94
95 ;go procedure that is called once the go button is clicked
96 to go
97   ;thieves are not seen in the model at all times, if hidden is false they are
   seen, if hidden is true they are hidden
98   ask thieves with [hidden? = false][
99     set hide? false
100  ]
101   ask bikes[
102     ;bicycles leave during 5-7ish
103     let leave (8 + random 2)
104     ;bicycles arrive during 7-9ish
105     let arrive (22 + random 2)
106
107     ;ensures that bicycles come and go during the working day
108     ;hides and shows the turtles
109     if (ticks - birth-tick = arrive) or (ticks - birth-tick = arrive + 1)[
110       ;restarts the time when bicycles arrive
111       set birth-tick ticks
112       set show? true
113       st
114     ]
115     if (ticks - birth-tick = leave) or (ticks - birth-tick = leave + 1)[
116       ; or (ticks - birth-tick = leave + 1)[
117         set show? false
118         ht
119       ]
120   ]
121
122   ;thieves move about if they are in the area

```

```

123   if hide? = false[move-thief]
124
125   ;if the bike marking intervention has been clicked, this is run
126   if bikemarking? = true [
127       bike-bikemarking
128       ask bikes with [bikemarking? = true and show? = true][
129           bikes-talk
130       ]
131   ]
132   ;if the advise removal of bicycle parts intervention has been clicked, this is
run
133   if removeparts? = true [
134       remove-parts
135       ask bikes with [remove? = true and show? = true][
136           bikes-talk
137       ]
138   ]
139   ;calls the procedure that moves police agents
140   move-police
141
142   ask thieves[
143       ;like bicycles, thieves do not spend all their time at the location
144       if (ticks - birth-tick) = random 8 + 3 [hide-turtle set hidden? true]
145       if (ticks - birth-tick) = 15 + random 5 [show-turtle set hidden? false set
birth-tick ticks]
146       if random 1000 = 5 [die]
147       if random 1000 = 5 [thief-enter]
148       ;low threshold, once this is reached thief-elsewhere procedure is called
149       if crime-probability < 0.1 [
150           thief-elsewhere
151       ]
152   ]
153   ;new thieves enter the environment
154   if count thieves < ((ratio-thieves * population-size ) - (ratio-thieves *
population-size * 0.4)) [create-thieves random (ratio-thieves * population-size)
[spawn-thief]]
155   ;new people decide to cycle to work
156   if count bikes < ((ratio-bikes * population-size) - (ratio-bikes * population-
size) * 0.15) [create-bikes (ratio-bikes * population-size * 0.1 ) [spawn-bike]]
157
158   tick
159   ;model stops once 261 days has been reached
160   if ticks = 6264 [stop]
161   end
162
163   ;sets out the value of patches for thieves
164   ;as each patch is 1 pixel in the image, a radius of 5 is used to make it as close
to real life as possible
165   to thief-patches
166       ask patches in-radius 5 with [(accessible?) and (count turtles-here > 1)][
167           if (any? thieves in-radius 5 with [accessible?])[
168               set q-val-north 0
169               set q-val-east 0
170               set q-val-south 0
171               set q-val-west 0
172           ]
173           if (any? bikes in-radius 5 with [accessible?])[
174               set q-val-north 10
175               set q-val-east 10
176               set q-val-south 10
177               set q-val-west 10
178           ]
179           if (any? policeofficer in-radius 5 with [accessible?])[

```

```

180         set q-val-north -10
181         set q-val-east -10
182         set q-val-south -10
183         set q-val-west -10
184     ]
185 ]
186 end
187
188 ;sets out the value of patches for police
189 to police-patches
190     ask patches in-radius 5 with [(accessible?) and (count turtles-here > 1)][
191         if (any? thieves in-radius 5 with [accessible?])[
192             set q-val-north 10
193             set q-val-east 10
194             set q-val-south 10
195             set q-val-west 10
196         ]
197         if (any? bikes in-radius 5 with [accessible?])[
198             set q-val-north 2
199             set q-val-east 2
200             set q-val-south 2
201             set q-val-west 2
202         ]
203         if (any? policeofficer in-radius 5 with [accessible?])[
204             set q-val-north -10
205             set q-val-east -10
206             set q-val-south -10
207             set q-val-west -10
208         ]
209     ]
210 end
211
212 ;movement of all agents
213 to move
214     ;gets the current x and y co-ordinates
215     let current-xcor xcor
216     let current-ycor ycor
217     ;25% chance of moving north, east, south, west
218     set heading ((random 4) * 90)
219     let probability random-float 1
220     ifelse (probability < 0.8)[ ;80% chance going in intended direction (MDP)
221         ][
222             ifelse (probability < 0.9)[
223                 set heading (heading + 90)];go left of intended direction
224                 [set heading (heading - 90)];go right of intended direction
225             ]
226         fd 10 ; direction is set, go forwards 10 patches
227         ;accidentally entered an area that is marked as a building, reverse
228         if pcolor = 28.6[
229             bk 10
230         ]
231         ;set the q-values
232         set-qvalue current-xcor current-ycor heading xcor ycor
233     end
234
235 ;movement of thief agents that are active
236 to move-thief
237     ask thieves with [hidden? = false][
238         move
239         if crime-probability > 0.2[
240             steal-bike
241         ]
242         sell-bike

```

```

243     ;]
244     thief-patches
245   ]
246 end
247
248 ;police need to check for stolen bikes with trackers
249 to move-police
250   ;ifelse enforce? = true[
251   ifelse (police-thief? = true) or (police-bike? = true) or (bikemarking? = true)[
252     ask policeofficer[
253       move
254       if police-thief? = true [police-thief]
255       ;policebike recover bicycles
256       if (police-bike? = true) or (bikemarking? = true)[police-bike]
257       ; if bikemarking? = true [police-recover]
258     police-patches
259   ]
260 ]
261 [ask policeofficer[hide-turtle]]
262
263 end
264
265 ;thief steals a bicycle
266 to steal-bike
267   let randomnumber random-float 1
268   let randomnumber2 random-float 1
269   ; ask bikes in-radius 6 with [(hidden? = false) and (shape = "bike") and (color =
green) and (not stolen?)]
270   ask bikes in-radius 2 with [(show? = true) and (shape = "bike") and (color =
green) and (not stolen?)]
271   ;if (shape = "bike") and (color = green) and (not stolen?)[
272     if (desirability > randomnumber) and (security < randomnumber2)[
273       set count-total count-total + 1
274       hatch-bikes 1 [set stolen? true set color red]
275       die
276     ]
277   ]
278 ; ]
279 end
280
281 ;bicycles are sold outside the area
282 to sell-bike
283   ask bikes with [color = red and show? = true][
284   ;ask bikes with [stolen? = true][
285     ;if ticks - birth-tick > localvariable2 + random 10 [show-turtle set hidden?
false set birth-tick ticks]
286     ;bike is sold in the area
287     if ticks - birth-tick = random 40 [hatch-bikes 1 die]
288     ;bike is sold outside of the area
289     if ticks - birth-tick > random 50 [die]
290   ]
291 end
292
293 ;thief comes to area
294 to thief-enter
295   hatch-thieves 1 [spawn-thief]
296 end
297
298 ;thieves detered from stealing/go to another area
299 to thief-elsewhere
300   die
301 end
302

```

```

303 ;police interaction with thieves
304 to police-thief
305   ask thieves in-radius 5 with [hidden? = false][
306     if random-float 1 > 0.5 [
307       set crime-probability crime-probability - (crime-probability * 0.25)
308       thieves-talk
309       ;thieves go elsewhere/stop thieving here
310       if crime-probability < 0.2 [die]
311     ]
312   ]
313 end
314
315 ;police have greater 'vision' of the environment
316 to police-bike
317   ask bikes in-radius 50 with [show? = true and advised? = false][
318     ;if the giving advice intervention is active, advise them on safer locking
    through verbally or leaflet
319     if police-bike? = true[
320       let chance random-float 1
321       if chance > 0.5 [set security security + 0.2 set advised? true]
322     ]
323     ;if the bike marking intervention is active and a bicycle is found, take it.
324     if bikemarking? = true[
325       if any? bikes in-radius 3 with [(stolen? = true) and (marked? = true)][
326         if random-float 1 > 0.8[
327           let time ticks
328           set recovered recovered + 1
329           die
330           ;bicycle is returned to the owner
331           if ticks = time + 48 [
332             hatch-bikes 1 [spawn-bike set marked? true set security random-float 1 +
0.2]
333           ]
334         ]
335       ]
336     ]
337   ]
338 end
339
340
341 ;bikes have bikemarking trackers- if recovered, possible to return to owner
342 to bike-bikemarking
343   ask bikes with [(show? = true) and (random-float 1 < 0.02)][
344     if random-float 1 < 0.02 [
345       set marked? true
346       set desirability desirability - 0.1
347       set security security + 0.05
348     ]
349   ]
350 end
351
352 ;bicycle parts are removed if the owner has finally decided upon
353 to remove-parts
354   ask bikes with [(show? = true) and (random-float 1 < 0.02) and (remove? = not
true)][
355     set remove? true
356     set desirability desirability - 0.1
357   ]
358 end
359
360 ;thieves talk to other thieves
361 to thieves-talk
362   ask thieves in-radius 15 with [(hidden? = false) and (random-float 1 > 0.5)][

```

```

363     set crime-probability crime-probability - 0.05
364 ]
365 end
366
367 ;bike owners talk to other bike owners
368 to bikes-talk
369     ask bikes in-radius 5 with [(show? = true) and (random-float 1 > 0.2)][
370         if bikemarking? = true [set marked? true]
371         if removeparts? = true [remove-parts]
372     ]
373 end
374
375 ;open source code that has been adapted from Larry Lin
376 to set-qvalue[current-xcor current-ycor current-heading new-xcor new-ycor]
377     ; Q(s',a') optimal future value
378     let optimal-f-val 0
379
380     ;compute optimal future value
381     ;finds the maximum reward possible (north, east, south, west)
382     ask patch new-xcor new-ycor[
383         set optimal-f-val (max (list q-val-north q-val-east q-val-south q-val-west))
384     ]
385
386     ;computed q-values
387     ask patch current-xcor current-ycor[
388         let alpha 0.8 ;learning rate
389         let gamma 0.8 ;discount factor
390         let reward 10
391         if(current-heading = 0)[
392             ;; north
393             set q-val-north (precision (q-val-north + alpha * (reward + (gamma * optimal-f-val) - q-val-north)) 1)
394         ]
395         if(current-heading = 90)[
396             ;; east
397             set q-val-east (precision (q-val-east + alpha * (reward + (gamma * optimal-f-val) - q-val-east)) 1)
398         ]
399         if(current-heading = 180)[
400             ;; south
401             set q-val-south (precision (q-val-south + alpha * (reward + (gamma * optimal-f-val) - q-val-south)) 1)
402         ]
403         if(current-heading = 270)[
404             ;; west
405             set q-val-west (precision (q-val-west + alpha * (reward + (gamma * optimal-f-val) - q-val-west)) 1)
406         ]
407     ]
408 ]
409 end
410
411 ;sets all characteristics and variables of police
412 to spawn-police
413     set shape "person"
414     set color blue
415     set size 20
416     move-to one-of patches with [accessible?]
417 end
418
419 ;sets all characteristics and variables of bicycles
420 to spawn-bike
421     set shape "bike"

```

```
422 set color green
423 set size 20
424 set desirability random-float 1
425 set security random-float 1
426 set stolen? false
427 set birth-tick ticks
428 set marked? false
429 set show? true
430 set advised? false
431 move-to one-of patches with [accessible?]
432 end
433
434 ;sets all characteristics and variables of thieves
435 to spawn-thief
436   setxy random-xcor random-ycor
437   set shape "person"
438   set color red
439   set size 20
440   set crime-probability random-float 1
441   move-to one-of patches with [accessible?]
442   set birth-tick ticks
443   set hidden? false
444 end
445 @#$#@#$#@
446 GRAPHICS-WINDOW
447 217
448 21
449 1490
450 459
451 -1
452 -1
453 1.0
454 1
455 10
456 1
457 1
458 1
459 0
460 1
461 1
462 1
463 -632
464 632
465 -214
466 214
467 0
468 0
469 1
470 ticks
471 30.0
472
473 BUTTON
474 34
475 196
476 98
477 229
478 Setup
479 setup
480 NIL
481 1
482 T
483 OBSERVER
484 NIL
```


485 NIL
486 NIL
487 NIL
488 1
489
490 SLIDER
491 14
492 21
493 186
494 54
495 population-size
496 population-size
497 5000
498 30000
499 12000.0
500 1000
501 1
502 NIL
503 HORIZONTAL
504
505 SLIDER
506 15
507 66
508 188
509 99
510 ratio-thieves
511 ratio-thieves
512 0
513 0.01
514 0.01
515 0.001
516 1
517 NIL
518 HORIZONTAL
519
520 SLIDER
521 16
522 108
523 187
524 141
525 ratio-bikes
526 ratio-bikes
527 0
528 0.3
529 0.25
530 0.005
531 1
532 NIL
533 HORIZONTAL
534
535 BUTTON
536 113
537 196
538 176
539 229
540 Go
541 go
542 T
543 1
544 T
545 OBSERVER
546 NIL
547 NIL

548 NIL
549 NIL
550 1
551
552 MONITOR
553 154
554 240
555 211
556 285
557 Police
558 count policeofficer
559 17
560 1
561 11
562
563 SLIDER
564 16
565 153
566 188
567 186
568 ratio-policeofficer
569 ratio-policeofficer
570 0
571 0.1
572 0.1
573 0.01
574 1
575 NIL
576 HORIZONTAL
577
578 MONITOR
579 10
580 299
581 103
582 344
583 bicycles stolen
584 count-total
585 17
586 1
587 11
588
589 BUTTON
590 24
591 391
592 194
593 424
594 Police warn thieves
595 set police-thief? true
596 T
597 1
598 T
599 OBSERVER
600 NIL
601 NIL
602 NIL
603 NIL
604 1
605
606 BUTTON
607 24
608 435
609 196
610 468

```
611 Offer locking advice
612 set police-bike? true
613 T
614 1
615 T
616 OBSERVER
617 NIL
618 NIL
619 NIL
620 NIL
621 1
622
623 PLOT
624 271
625 493
626 1077
627 806
628 Bicycle thefts over time
629 time (in ticks)
630 stolen bicycles
631 0.0
632 2545.0
633 0.0
634 800.0
635 true
636 false
637 "" ""
638 PENS
639 "default" 1.0 0 -16777216 true "" "plot count-total"
640
641 BUTTON
642 23
643 482
644 197
645 515
646 Bike marking
647 set bikemarking? true
648 T
649 1
650 T
651 OBSERVER
652 NIL
653 NIL
654 NIL
655 NIL
656 1
657
658 MONITOR
659 112
660 299
661 182
662 344
663 recovered
664 recovered
665 17
666 1
667 11
668
669 MONITOR
670 84
671 239
672 141
673 284
```

```

674 Bicycles
675 count bikes with [show? = true]
676 17
677 1
678 11
679
680 MONITOR
681 10
682 239
683 72
684 284
685 Thieves
686 count thieves with [hidden? = false]
687 17
688 1
689 11
690
691 TEXTBOX
692 75
693 367
694 225
695 385
696 INTERVENTIONS
697 11
698 0.0
699 1
700
701 BUTTON
702 15
703 529
704 214
705 570
706 Advise removal of expensive parts
707 set removeparts? true
708 T
709 1
710 T
711 OBSERVER
712 NIL
713 NIL
714 NIL
715 NIL
716 1
717
718 @#$#@#$#@
719 ## WHAT IS IT?
720
721 This model is a 2D agent-based model of bicycle theft within a small area in the
City of London specifically focussing upon interventions that may be implemented
to deter bicycle thefts.
722
723 ## HOW IT WORKS
724
725 Agents are broken down into 3 types: bicycles, thieves and police. It is the sole
aim for thieves to steal bicycles within the environment, bicycles are naturally
in the environment and police are there to try and limit thefts.
726
727 ## HOW TO USE IT
728
729 Click on the SETUP button to set up the environment. Set the NUMBER slider to
change the number of agents within the environment. Click on GO to start the
agents moving. Interventions can be activated by clicking on the button of the
intervention, remember to setup the model again with the new intervention.

```

```
730
731
732
733 ## THINGS TO TRY
734
735 Change the number of agents within the environment and experiment with the
interventions provided. To note, multiple interventions are allowed in the model.
736
737 ## EXTENDING THE MODEL
738
739 The inclusion of environmental features for example street lighting, CCTV and bus
stops.
740 Greater interventions.
741
742
743
744 ## CREDITS AND REFERENCES
745
746 Emily Liu
747
748 Q-learning: Larry Lin, Modelling Commons.
749 @#$#@#$#@
750 default
751 true
752 0
753 Polygon -7500403 true true 150 5 40 250 150 205 260 250
754
755 airplane
756 true
757 0
758 Polygon -7500403 true true 150 0 135 15 120 60 120 105 15 165 15 195 120 180 135
240 105 270 120 285 150 270 180 285 210 270 165 240 180 180 285 195 285 165 180
105 180 60 165 15
759
760 arrow
761 true
762 0
763 Polygon -7500403 true true 150 0 0 150 105 150 105 293 195 293 195 150 300 150
764
765 bike
766 false
767 1
768 Line -7500403 false 163 183 228 184
769 Circle -7500403 false false 213 184 22
770 Circle -7500403 false false 156 187 16
771 Circle -16777216 false false 28 148 95
772 Circle -2674135 false true 24 144 102
773 Circle -16777216 false false 174 144 102
774 Circle -2674135 false true 177 148 95
775 Polygon -2674135 true true 75 195 90 90 98 92 97 107 192 122 207 83 215 85 202 123
211 133 225 195 165 195 164 188 214 188 202 133 94 116 82 195
776 Polygon -2674135 true true 208 83 164 193 171 196 217 85
777 Polygon -2674135 true true 165 188 91 120 90 131 164 196
778 Line -7500403 false 159 173 170 219
779 Line -7500403 false 155 172 166 172
780 Line -7500403 false 166 219 177 219
781 Polygon -2674135 true true 187 92 198 92 208 97 217 100 231 93 231 84 216 82 201
83 184 85
782 Polygon -2674135 true true 71 86 98 93 101 85 74 81
783 Rectangle -16777216 true false 75 75 75 90
784 Polygon -2674135 true true 70 87 70 72 78 71 78 89
785 Circle -7500403 false false 153 184 22
786 Line -7500403 false 159 206 228 205
```

```
787
788 box
789 false
790 0
791 Polygon -7500403 true true 150 285 285 225 285 75 150 135
792 Polygon -7500403 true true 150 135 15 75 150 15 285 75
793 Polygon -7500403 true true 15 75 15 225 150 285 150 135
794 Line -16777216 false 150 285 150 135
795 Line -16777216 false 150 135 15 75
796 Line -16777216 false 150 135 285 75
797
798 bug
799 true
800 0
801 Circle -7500403 true true 96 182 108
802 Circle -7500403 true true 110 127 80
803 Circle -7500403 true true 110 75 80
804 Line -7500403 true 150 100 80 30
805 Line -7500403 true 150 100 220 30
806
807 butterfly
808 true
809 0
810 Polygon -7500403 true true 150 165 209 199 225 225 225 255 195 270 165 255 150 240
811 Polygon -7500403 true true 150 165 89 198 75 225 75 255 105 270 135 255 150 240
812 Polygon -7500403 true true 139 148 100 105 55 90 25 90 10 105 10 135 25 180 40 195
813 Polygon -7500403 true true 162 150 200 105 245 90 275 90 290 105 290 135 275 180
814 Polygon -16777216 true false 150 255 135 225 120 150 135 120 150 105 165 120 180
815 Circle -16777216 true false 135 90 30
816 Line -16777216 false 150 105 195 60
817 Line -16777216 false 150 105 105 60
818
819 car
820 false
821 0
822 Polygon -7500403 true true 300 180 279 164 261 144 240 135 226 132 213 106 203 84
823 Circle -16777216 true false 180 180 90
824 Circle -16777216 true false 30 180 90
825 Polygon -16777216 true false 162 80 132 78 134 135 209 135 194 105 189 96 180 89
826 Circle -7500403 true true 47 195 58
827 Circle -7500403 true true 195 195 58
828
829 circle
830 false
831 0
832 Circle -7500403 true true 0 0 300
833
834 circle 2
835 false
836 0
837 Circle -7500403 true true 0 0 300
838 Circle -16777216 true false 30 30 240
839
840 cow
841 false
842 0
843 Polygon -7500403 true true 200 193 197 249 179 249 177 196 166 187 140 189 93 191
78 179 72 211 49 209 48 181 37 149 25 120 25 89 45 72 103 84 179 75 198 76 252 64
272 81 293 103 285 121 255 121 242 118 224 167
```

```
844 Polygon -7500403 true true 73 210 86 251 62 249 48 208
845 Polygon -7500403 true true 25 114 16 195 9 204 23 213 25 200 39 123
846
847 cylinder
848 false
849 0
850 Circle -7500403 true true 0 0 300
851
852 dot
853 false
854 0
855 Circle -7500403 true true 90 90 120
856
857 face happy
858 false
859 0
860 Circle -7500403 true true 8 8 285
861 Circle -16777216 true false 60 75 60
862 Circle -16777216 true false 180 75 60
863 Polygon -16777216 true false 150 255 90 239 62 213 47 191 67 179 90 203 109 218
150 225 192 218 210 203 227 181 251 194 236 217 212 240
864
865 face neutral
866 false
867 0
868 Circle -7500403 true true 8 7 285
869 Circle -16777216 true false 60 75 60
870 Circle -16777216 true false 180 75 60
871 Rectangle -16777216 true false 60 195 240 225
872
873 face sad
874 false
875 0
876 Circle -7500403 true true 8 8 285
877 Circle -16777216 true false 60 75 60
878 Circle -16777216 true false 180 75 60
879 Polygon -16777216 true false 150 168 90 184 62 210 47 232 67 244 90 220 109 205
150 198 192 205 210 220 227 242 251 229 236 206 212 183
880
881 fish
882 false
883 0
884 Polygon -1 true false 44 131 21 87 15 86 0 120 15 150 0 180 13 214 20 212 45 166
885 Polygon -1 true false 135 195 119 235 95 218 76 210 46 204 60 165
886 Polygon -1 true false 75 45 83 77 71 103 86 114 166 78 135 60
887 Polygon -7500403 true true 30 136 151 77 226 81 280 119 292 146 292 160 287 170
270 195 195 210 151 212 30 166
888 Circle -16777216 true false 215 106 30
889
890 flag
891 false
892 0
893 Rectangle -7500403 true true 60 15 75 300
894 Polygon -7500403 true true 90 150 270 90 90 30
895 Line -7500403 true 75 135 90 135
896 Line -7500403 true 75 45 90 45
897
898 flower
899 false
900 0
901 Polygon -10899396 true false 135 120 165 165 180 210 180 240 150 300 165 300 195
240 195 195 165 135
902 Circle -7500403 true true 85 132 38
```

```
903 Circle -7500403 true true 130 147 38
904 Circle -7500403 true true 192 85 38
905 Circle -7500403 true true 85 40 38
906 Circle -7500403 true true 177 40 38
907 Circle -7500403 true true 177 132 38
908 Circle -7500403 true true 70 85 38
909 Circle -7500403 true true 130 25 38
910 Circle -7500403 true true 96 51 108
911 Circle -16777216 true false 113 68 74
912 Polygon -10899396 true false 189 233 219 188 249 173 279 188 234 218
913 Polygon -10899396 true false 180 255 150 210 105 210 75 240 135 240
914
915 house
916 false
917 0
918 Rectangle -7500403 true true 45 120 255 285
919 Rectangle -16777216 true false 120 210 180 285
920 Polygon -7500403 true true 15 120 150 15 285 120
921 Line -16777216 false 30 120 270 120
922
923 leaf
924 false
925 0
926 Polygon -7500403 true true 150 210 135 195 120 210 60 210 30 195 60 180 60 165 15
    135 30 120 15 105 40 104 45 90 60 90 90 105 105 120 120 120 105 60 120 60 135 30
    150 15 165 30 180 60 195 60 180 120 195 120 210 105 240 90 255 90 263 104 285 105
    270 120 285 135 240 165 240 180 270 195 240 210 180 210 165 195
927 Polygon -7500403 true true 135 195 135 240 120 255 105 255 105 285 135 285 165 240
    165 195
928
929 line
930 true
931 0
932 Line -7500403 true 150 0 150 300
933
934 line half
935 true
936 0
937 Line -7500403 true 150 0 150 150
938
939 pentagon
940 false
941 0
942 Polygon -7500403 true true 150 15 15 120 60 285 240 285 285 120
943
944 person
945 false
946 0
947 Circle -7500403 true true 110 5 80
948 Polygon -7500403 true true 105 90 120 195 90 285 105 300 135 300 150 225 165 300
    195 300 210 285 180 195 195 90
949 Rectangle -7500403 true true 127 79 172 94
950 Polygon -7500403 true true 195 90 240 150 225 180 165 105
951 Polygon -7500403 true true 105 90 60 150 75 180 135 105
952
953 plant
954 false
955 0
956 Rectangle -7500403 true true 135 90 165 300
957 Polygon -7500403 true true 135 255 90 210 45 195 75 255 135 285
958 Polygon -7500403 true true 165 255 210 210 255 195 225 255 165 285
959 Polygon -7500403 true true 135 180 90 135 45 120 75 180 135 210
960 Polygon -7500403 true true 165 180 165 210 225 180 255 120 210 135
```



```
961 Polygon -7500403 true true 135 105 90 60 45 45 75 105 135 135
962 Polygon -7500403 true true 165 105 165 135 225 105 255 45 210 60
963 Polygon -7500403 true true 135 90 120 45 150 15 180 45 165 90
964
965 sheep
966 false
967 15
968 Circle -1 true true 203 65 88
969 Circle -1 true true 70 65 162
970 Circle -1 true true 150 105 120
971 Polygon -7500403 true false 218 120 240 165 255 165 278 120
972 Circle -7500403 true false 214 72 67
973 Rectangle -1 true true 164 223 179 298
974 Polygon -1 true true 45 285 30 285 30 240 15 195 45 210
975 Circle -1 true true 3 83 150
976 Rectangle -1 true true 65 221 80 296
977 Polygon -1 true true 195 285 210 285 210 240 240 210 195 210
978 Polygon -7500403 true false 276 85 285 105 302 99 294 83
979 Polygon -7500403 true false 219 85 210 105 193 99 201 83
980
981 square
982 false
983 0
984 Rectangle -7500403 true true 30 30 270 270
985
986 square 2
987 false
988 0
989 Rectangle -7500403 true true 30 30 270 270
990 Rectangle -16777216 true false 60 60 240 240
991
992 star
993 false
994 0
995 Polygon -7500403 true true 151 1 185 108 298 108 207 175 242 282 151 216 59 282 94
    175 3 108 116 108
996
997 target
998 false
999 0
1000 Circle -7500403 true true 0 0 300
1001 Circle -16777216 true false 30 30 240
1002 Circle -7500403 true true 60 60 180
1003 Circle -16777216 true false 90 90 120
1004 Circle -7500403 true true 120 120 60
1005
1006 tree
1007 false
1008 0
1009 Circle -7500403 true true 118 3 94
1010 Rectangle -6459832 true false 120 195 180 300
1011 Circle -7500403 true true 65 21 108
1012 Circle -7500403 true true 116 41 127
1013 Circle -7500403 true true 45 90 120
1014 Circle -7500403 true true 104 74 152
1015
1016 triangle
1017 false
1018 0
1019 Polygon -7500403 true true 150 30 15 255 285 255
1020
1021 triangle 2
1022 false
```

```
1023 0
1024 Polygon -7500403 true true 150 30 15 255 285 255
1025 Polygon -16777216 true false 151 99 225 223 75 224
1026
1027 truck
1028 false
1029 0
1030 Rectangle -7500403 true true 4 45 195 187
1031 Polygon -7500403 true true 296 193 296 150 259 134 244 104 208 104 207 194
1032 Rectangle -1 true false 195 60 195 105
1033 Polygon -16777216 true false 238 112 252 141 219 141 218 112
1034 Circle -16777216 true false 234 174 42
1035 Rectangle -7500403 true true 181 185 214 194
1036 Circle -16777216 true false 144 174 42
1037 Circle -16777216 true false 24 174 42
1038 Circle -7500403 false true 24 174 42
1039 Circle -7500403 false true 144 174 42
1040 Circle -7500403 false true 234 174 42
1041
1042 turtle
1043 true
1044 0
1045 Polygon -10899396 true false 215 204 240 233 246 254 228 266 215 252 193 210
1046 Polygon -10899396 true false 195 90 225 75 245 75 260 89 269 108 261 124 240 105
225 105 210 105
1047 Polygon -10899396 true false 105 90 75 75 55 75 40 89 31 108 39 124 60 105 75 105
90 105
1048 Polygon -10899396 true false 132 85 134 64 107 51 108 17 150 2 192 18 192 52 169
65 172 87
1049 Polygon -10899396 true false 85 204 60 233 54 254 72 266 85 252 107 210
1050 Polygon -7500403 true true 119 75 179 75 209 101 224 135 220 225 175 261 128 261
81 224 74 135 88 99
1051
1052 wheel
1053 false
1054 0
1055 Circle -7500403 true true 3 3 294
1056 Circle -16777216 true false 30 30 240
1057 Line -7500403 true 150 285 150 15
1058 Line -7500403 true 15 150 285 150
1059 Circle -7500403 true true 120 120 60
1060 Line -7500403 true 216 40 79 269
1061 Line -7500403 true 40 84 269 221
1062 Line -7500403 true 40 216 269 79
1063 Line -7500403 true 84 40 221 269
1064
1065 wolf
1066 false
1067 0
1068 Polygon -16777216 true false 253 133 245 131 245 133
1069 Polygon -7500403 true true 2 194 13 197 30 191 38 193 38 205 20 226 20 257 27 265
38 266 40 260 31 253 31 230 60 206 68 198 75 209 66 228 65 243 82 261 84 268 100
267 103 261 77 239 79 231 100 207 98 196 119 201 143 202 160 195 166 210 172 213
173 238 167 251 160 248 154 265 169 264 178 247 186 240 198 260 200 271 217 271
219 262 207 258 195 230 192 198 210 184 227 164 242 144 259 145 284 151 277 141
293 140 299 134 297 127 273 119 270 105
1070 Polygon -7500403 true true -1 195 14 180 36 166 40 153 53 140 82 131 134 133 159
126 188 115 227 108 236 102 238 98 268 86 269 92 281 87 269 103 269 113
1071
1072 x
1073 false
1074 0
1075 Polygon -7500403 true true 270 75 225 30 30 225 75 270
```

```
1076 Polygon -7500403 true true 30 75 75 30 270 225 225 270
1077 @#$#@#$#@
1078 NetLogo 6.1.1
1079 @#$#@#$#@
1080 @#$#@#$#@
1081 @#$#@#$#@
1082 @#$#@#$#@
1083 @#$#@#$#@
1084 default
1085 0.0
1086 -0.2 0 0.0 1.0
1087 0.0 1 1.0 0.0
1088 0.2 0 0.0 1.0
1089 link direction
1090 true
1091 0
1092 Line -7500403 true 150 150 90 180
1093 Line -7500403 true 150 150 210 180
1094 @#$#@#$#@
1095 0
1096 @#$#@#$#@
1097
```