

Adaptive Systems

Lab 6: Breeds and Shapes

Practical 6: Cars and People

This practical will allow you to learn:

- How to create different families of moving agents (turtles)
- How to control each family
- How to associate different shapes to those moving agents

System to build

The system consists of two families of “turtles”: cars and people. In the system, people move looking for cars. Once a person has found a car, it enters the car (it disappears), and the car starts moving for a random distance, and then stops.

- Create a new project
- Define two new family names, called **breeds** in StarLogo, in the Observer procedure window

```
breeds [cars people]
```

- Create an Observer procedure setup in the Observer procedure window that creates a family of 20 cars, and a family of 50 people.

Create the corresponding button in the interface.

- Modify procedure setup so that:
 - Cars have a shape of cars and are paint in grey
 - People have a shape of people and are paint in red
 - Cars and people are disposed randomly in the screen
 - The command for the Observer to ask a family of agents to perform some behaviour is (similar to ask-turtles):

```
ask-breedname [list of commands]
```

for instance:

- ask-cars [...]
- ask-people [...]

- The Turtle command for changing shapes is:

```
setshape name-of-shape
```

You can find the names of pre-defined shapes in the Shape Chooser window
Control center > Windows > Shape Chooser

Click on the shapes to see their names.

- The Turtle command for the cars or people family to change their colour is (similar to setc for turtles):

`setc colour-name`

- The Turtle command for randomly changing the x and y coordinates of a turtle is:

`setx random 200 sety random 200`

- Create a procedure look-for-car in the Turtle procedure window; such that people look randomly for a car, once they have found one, they enter the car (i.e. they disappear).

The command to test if there is a car at the current position is:
`count-cars-here`

The command for turtles to disappear is `die`.

- Create a procedure move-around in the Turtle procedure window. A car moves randomly for a finite number of steps and then stops. Change the colour of the car when it moves.

Define and use procedure wiggle defined in previous labs.

- Create a procedure go_people in the Observer procedure window. This procedure simply starts the people (starts look-for-car).

Create the corresponding forever button in the interface.

- Create a procedure go_cars in the Observer procedure window. This procedure simply starts the cars (starts move-around).

Create the corresponding forever button in the interface.

- Modify procedures look-for-car and move-around in order to link the people entering the car with the movement of the cars. Use a variable that you associate to a patch.

- To define a patch variable, write

`patches-own [variablename]`

in the Turtle procedure window.

- To change a variable value:

`setvariablename number`

- Create a bar chart to see the progression of the number of people
- Do not forget to save your project!