

3 Event onReceive

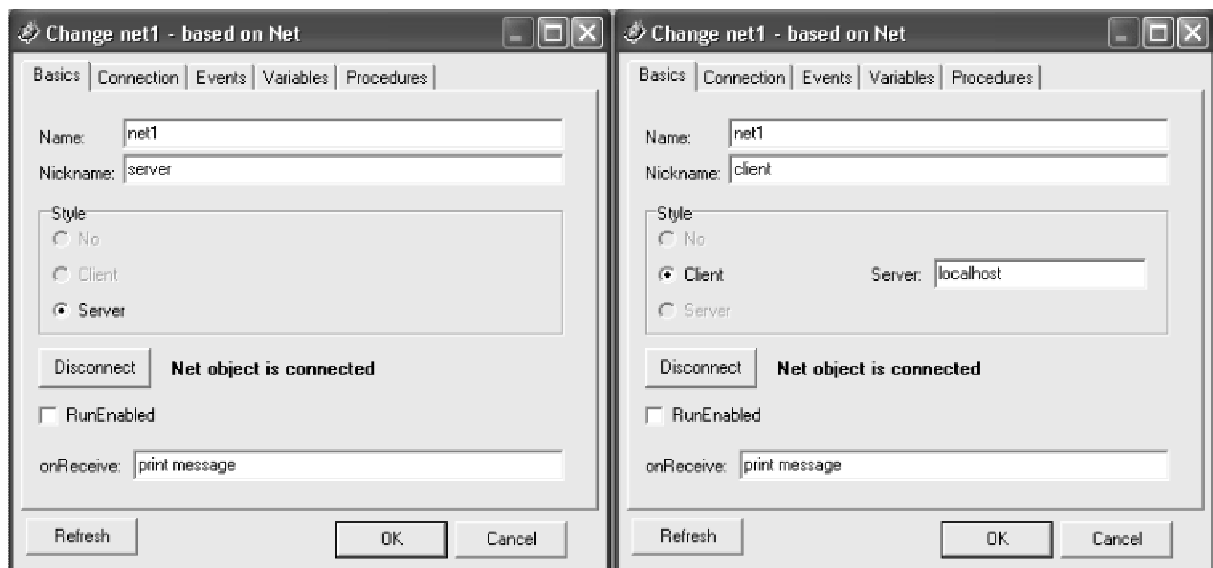
In the previous chapter we developed projects, which sent simple messages. Now, we are going to refine them.

We know how to send and receive text messages. If we want to find out, what message we received, we have to type in the command line the instruction `print net1'message`. This instruction will show the most recently received message. However, how can we easily find out, that the next message has just arrived? The solution is the `onReceive` **event** of the `Net` object.

1. If you open the **Change net1** dialogue box, right there on its **Basic** tab sheet you will see the edit line titled `onReceive`. The connection will react to this event and will print the input message in the Text screen immediately. Type the following instruction into the edit line :

```
$ print message
```

You don't have to type `print net1'message`, because this is the **Change net1** dialogue box, so this instruction naturally belongs to the `net1` object.



Specify the `onReceive` event in this way in both connected Imagines.

2. Now when the left Imagine sends a message, this message will show in the Text screen immediately in the right Imagine (and vice versa).

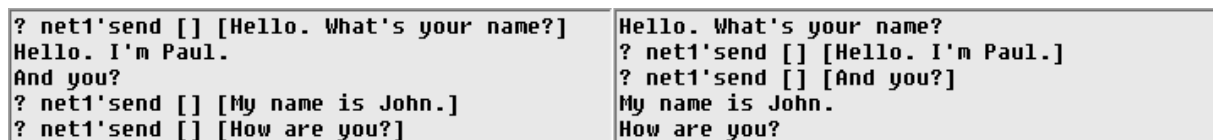


Illustration of the communication between two Imagines – a kind of very simple chat.

3. Let us slightly upgrade our simple communication – we won't display the coming messages in the Text screen, but let the turtle `t1` print them in the page. Change the `onReceive` event for the `net1` object (in both Left and Right Imagines):

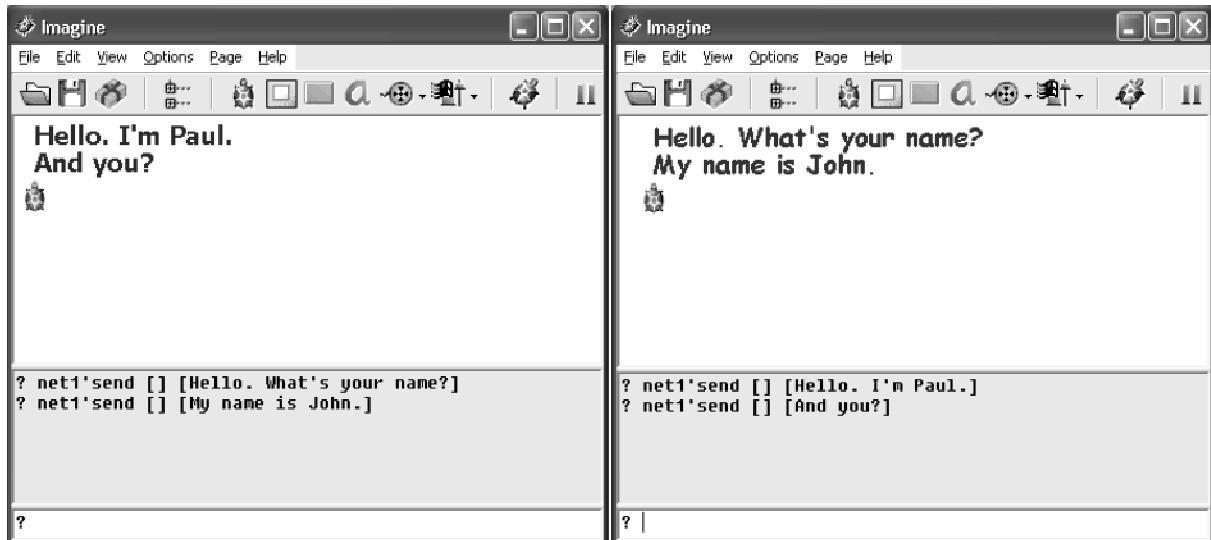
```
$ t1'label message t1'back 20
the instructions label and back belong to the turtle t1, therefore we must write
t1'label ... and t1'back ...
```

```
onReceive: t1'label message t1'back 20
```

4. Next step: set the turtle's pen to penUp, set its colour and font to whatever you like (in both Left and Right Imagines). Now, send the message from the Left Imagine:

```
$ net1'send [] [Hello. What's your name?]
```

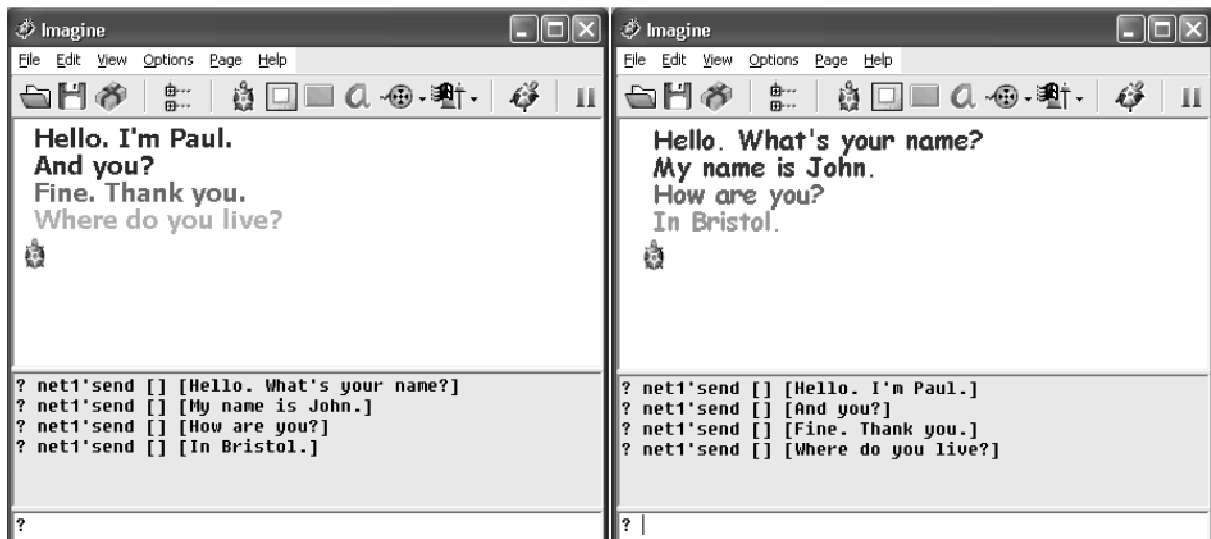
This message won't appear in the Text screen, but will be printed by the turtle to the page of the Right Imagine.



If we want to write each message in different colour, we must modify the onReceive event as follows:

```
$ t1'setPC any t1'label message t1'back 20
```

```
onReceive: t1'setPC any t1'label message t1'back 20
```



Conclusion

The onReceive event for the net object helps us react properly to the incoming messages automatically. We can define it in the **Change net1** dialogue box for the net object.