Chapter ??

Saving an image is a really practical way to provide customized environments in which lot of settings can be changed as shown in Chapter **??**. Once the image is saved, you just need to give the image *and* changes files and the students get everything you prepared.

Chapter ??

Adapting the language. You can easily adapt to your own language the names of the messages that are sent to a turtle. For example, if you can define messages that do not require arguments by defining method invoking the method go: and jump: with a predefined length. As we could not guess all your needs we choose the minimal ones but we explain in Chapter **??** how you can define your own set of messages, change the shape of the turtle and even draw the way the turtle should look like.

About the cascade. The cascade is interesting because it shortens the script. However pay attention that your audience understands well that all the messages are sent to the same receiver. In a simple lecture we avoid completely the introduction of the notion of cascade.

About OOP. We took the decision to introduce object-oriented and especially Smalltalk terminology for example by using the term "method", even if the learner would not define its own objects in the context of this book.

We took this decision for the following reason: Firstly the metaphor of object-oriented programming is adapted to teach even simple programming concepts such as loops, or abstraction. Secondly, this choice allows the use of a limited number of terms. As a consequence, some terms represent slightly different concepts. In this book, the methods represent procedures and to not include the idea that a method is chosen at runtime. The advantage of this choice linked with the fact that Smalltalk is dynamically type is that we can use + between integers and floats without even noticing it and without having to teach types and the difficulties they bring into play.

If your intention is to teach object-oriented programming or to explain it to novices, pay attention to use the term *a turtle* and not *the* turtle. In object-oriented programming, several turtles can be created each described by the same class but being able to have a specific state.

Chapter ??

Depending of the age of your audience you may want to create your own methods up, down, left, and right , if this is the case read the chapters **??**, **??**, and **??**. @@ TO CHECK @@

Chapter ??

In Squeak, you can also use the arrow ← by typing the underscore _ key. However, it is better to use := because this is standard in all the other Smalltalk and the arrow is not defined in all the font sizes. When the arrow character is not available in the font, the arrow is displayed as the underscore character: _.

Chapter ??

Our experience showed that students literally loved to produce mazes, spirals and other crazy drawings. However, in the first version of this book we introduced this chapter just after the variables and realized that this was a mistake. In fact the students where losing the notion of finite loops — this was especially true after playing too much with the last exercises. For example one student wrote a method to draw a square by looping far too much time. The problem of such exercises is that there is no simple stopping condition, so the students pick up randomly numbers whithout having any support to reason and understand the result they get. The second drawback of these kinds of exercises is that students associate variables only with loops. This is for these reasons that this chapter arrives so late in the book.

Chapter ??

Saving and loading scripts is not really well done and quite minimal. We did that on purpose because we want to force the students to define methods and to save them and not scripts.