

Teaching Science Using a Mobile Phone

Dieter W. Heermann

Institut für theoretische Physik

Universität Heidelberg

Philosophenweg 19

D-69120 Heidelberg

and

Interdisziplinäres Zentrum

für Wissenschaftliches Rechnen

der Universität Heidelberg

3rd June 2002

Abstract

Mobile phones are very useful devices. One can phone with them but one can also use them to reach and teach students in high schools as well as in universities. To this end I demonstrate, with an example,

the feasibility of the approach. I show that the phone can be used to do a science simulation using the processing power of the phone.

Keywords: mobile phone, education, J2ME, fractal, Monte Carlo simulation

1 Introduction

There is one thing that you find these days in many pockets of students: a mobile phone. Not much seems to be more interesting than holding a mobile phone and chatting with a friend, sending a short message (SMS) or simply being reachable. What is also striking is that many use their phone to pass time, playing simple games while waiting or riding on the bus. There is a generation that grows up with mobile devices in much the same way that a generation is growing up with computers. Also there playing was a strong driving force for the adoption of the new device. Maturing many use the computer also as a device that helps solve problems posed in class or simply as a better typewriter or calculator.

With the advent of mobile devices such as the new generation mobile phones that carry computing power with it a new corridor opened to reach students. Here we are not concerned with being able to log on to the internet with the mobile phone and accessing information relevant for the particular home work assignment. We propose to use the mobile phone as a smart playing device to bring across scientific ideas. At least for those that are scientifically inclined, the OTA (over the air) download of a scientific game or problem may be more interesting, at least sometimes.

In the following it is demonstrated that it is possible to design software that meets the stringent requirements on the available computing resources of mobile phones.

2 The Framework for Teaching and doing Science

The key to being able to do science and to use the mobile phone as a vehicle for teaching is the Java 2 Platform, Micro Edition (J2ME-environment) [1]. J2ME is a Java runtime environment targeting a wide range of products, including pagers, cellular phones, screenphones, digital set-top boxes and car navigation systems. This standard is now being supported by all manufactures of mobile phones as well as other mobile devices. J2ME provides means for a runtime environment for programs as well as means for the download of programs over the air (OTA). Programs in this context are called MIDlets (MID stands for mobile information device).

For a mobile phone two components are important: The Mobile Information Device Profile and the Connected Limited Device Configuration. The Mobile Information Device Profile (MIDP) is a set of Java APIs which, together with the Connected, Limited Device Configuration (CLDC), provides a complete J2ME application runtime environment targeted at mobile information devices, such as mobile phones. The configuration describes the minimal Java platform for a device class (horizontal group; each group has roughly the same amount of memory, communication bandwidth etc.) The profile enlarges the configuration with an application interface for the specific device.

To develop programs for mobile phones one needs an development platform that emulates the capabilities of the mobile phone. Several such platforms exist. Besides the J2ME Wireless Toolkit (a set of tools that provides developers with the emulation environment needed to develop CLDC/MIDP

compliant applications), which is also is free download at Sun Corp., some commercial Integrated Development Environments exist.

Mobile devices come with a range of restrictions. Because of its mobility the battery weight requirement puts a clear limit on the computing capacity of the processor inside of the mobile phone. However, the processors are fast enough to sustain a good deal of computing much in the range of the very early speed of the desktop computers. Compare this to the computing power of the *Earth Simulator*. This ultra high-speed parallel computing system has a peak of execution performance 35.61 TFlops! [2].

From the point of view of doing science the low memory presents a more challenging problem. There is the possibility to store data into additional storage such as memory sticks of media cards. The writing process to the off-chip persistent memory is usually very slow and hampers the use if not making it impossible to use at least for some scientific problems where larger amount is being used.

Perhaps the most fundamental restriction is that J2ME does not know what real numbers are. That may not be a problem you may say, we could write some routines that would emulate real numbers. Yes, but we also need the implementation of scientific functions like sine, the exponential function etc. Due to the limited memory of about 30K that really is a problem given the above mentioned small bandwidth to the persistent memory.

There is one additional constraint: the limited responsiveness of the displays. Remember, the phone was not designed for impressive graphics. Who needs a fast display if all you want to display are some numbers? A graphic processor may well consume as much power as the computing processor in modern computers!

Having laid out the constraints we can now exploit what we have at hand to design application for educational use. To this end we need to look closer at the environment at hand. The J2ME environment for the mobile phone is defined by the MID Profile which consists of the:

- **User Interface Package**, which provides a set of features for implementation of user interfaces for MIDP applications. Note the above mentioned restrictions on the capabilities.
- **Persistence Package**, which provides a mechanism for MIDlets to persistently store data and later retrieve it.
- **Application Lifecycle Package**, which defines the interactions between the application and the environment in which the application runs.
- **Networking Package**, which includes networking support based on the generic connection framework from the Connected Limited Device Configuration.
- **Core Packages**, which provides for system input and output through data streams, the language classes and the utility classes included from Java 2 Standard Edition.

This is a very rich framework to work with. It supports computing, displaying information on devices with limited displaying capabilities and the possibility to connect the device for example to the internet.

3 Simulating Fractal Growth on a Mobile Phone

Let us now work out an example application. The generation of fractal objects furnishes us with an example that is on the one hand readily implemented with a modest amount of code and on the other interesting from point of teaching.

Fractals [3] are objects with a size dependence that is not that of the embedding space. Consider a metal ball. The volume V scales with the radius R as

$$V \propto R^3 \tag{1}$$

The volume of a fractal object scales as

$$V \propto R^{d_f} \tag{2}$$

with $d_f < 3$. Whereas the ball would fill all space in the limit of R going to infinity it is clear considering the ratio of space with the volume of the object, that fractal objects do not fill space. Where does one find such fractals? Coast lines are fractal objects for example

Here, consider the well-known diffusion limited aggregate, a fractal that one obtains by particles diffusing on a plane. We start with a seed particle that is fixed at origin. Now we place a new particle onto the plane away from the seed (in principle it should come from infinity). If the particle comes near to the seed particle (nearest neighbour) it sticks to the seed. We continue with placing a new particle on the plane. Eventually the initial seed grows into an object of size N (the number of particles that are attached to the seed) with a radius R .

Simulating the growth of the fractal on the mobile phone the student can now see how the fractal grows (c.f. Figure 1). With the information on the radius and number of particles of the cluster one can also plot these numbers to obtain an estimate for the fractal dimension.

As with all Java application we define a root class *MyDLA*. This is the class that first called upon by the runtime environment. The class will be part of a package *DLA* that we load into the memory of the mobile phone (c.f. Figure 2). The package will be assembled into a *JAR*-File. Here all the resources that the program needs are assembled: The program classes, the possible images as well as an icon that represents that program in the list of executable programs in the mobile phones display. Along with the *JAR*-File comes the *JAD*-File [5]. It describes the resources that the program needs.

The main class of the program implements a *commandListener*. The main class is responsible for the command that a user issues to the program. The class constructor *public DLACluster ()* sets up a *display* i.e. something like a screen on which we can write our results to and constructs a menu for the user. We then create an instance of the class *Canvas* in conjunction with the class *Particle* which is responsible for the actual computing. The diffusion of the particle is simulated in the class *Canvas* using a Monte Carlo algorithm [4]. As a parameter in the class constructor of *Canvas* we submit the *display* so that we write and draw to the display of the mobile phone. The main Monte Carlo loop as well as the drawing is implemented here. The class *Particle* is responsible for the displacement of the particle and also controls whether the particle is still inside of the simulation region. This has the advantage that we could have several particles simultaneously running given that the processor has enough power. The class *canvas* synchronizes

the possible many instances of the class *particle*.

4 Outlook and Discussion

Using the mobile phone as a teaching device is a new ansatz to reach students. Two aspects may be considered in this respect. One is to interest high-school students in science. The other aspect is to use the mobile phone in teaching.

It is well worth an experiment to set up a *mobile class room* where students can study at any time anywhere. With the possibility of the mobile phone to connect to the internet it is also possible that results that the student obtains during the engagement with a problem be transferred to a central server, so desired. This may trigger an automatic response from the server to the student giving advice or storing the result for later evaluation. With the up-coming availability of GPRS it is also possible to assign problems to students over the air and one could also think of collaboration projects between students.

The success of the approach rests to a larger degree on the playfulness of the problems. Perhaps the above example is not a brilliant one in this respect as it involves little or no interaction between the user and the problem. Here only the results from the simulations need to be noted (N and R) and then plotted at home to derive the fractal dimension.

Acknowledgement

It is a pleasure to acknowledge the many fruitful discussions with L. Ackermann and D. Stauffer.

References

- [1] J2ME, Sun Microsystems, java.sun.com/j2me
- [2] Earth Simulator Center, www.es.jamstec.go.jp
- [3] A. Bunde and S. Havlin (Eds.). *Fractals and Disordered Systems*, 2nd ed. New York: Springer-Verlag, 1996
- [4] K. Binder and D.W. Heermann, *Monte Carlo Simulation in Statistical Physics: An Introduction, Fourth Edition*, Springer Verlag, Heidelberg, 2002
- [5] The application can be downloaded at wwwcp.tphys.uni-heidelberg.de

Figure 1: The left figure shows the resulting display when the program is called. The right figure shows the result of a simulation of about 30 minutes length.



Figure 2: Shown is the contents of the package for the midlet and the dependences of the class in the corresponding unified modelling diagram

