

# 1 Preface

*Cinderella* is a program for doing geometry on a computer. In its present form it is the product of a sequel of three projects done between 1993 and 1998. It is based on various mathematical theories ranging from the great discoveries of the geometers in the nineteenth century to newly developed methods that find their first applications in this program.

The idea for the first of these projects was born in 1992 during a combinatorics conference at the Mittag-Leffler Institute in Sweden, when Henry Crapo and Jürgen Richter-Gebert were taking a trip on a boat called "Cinderella." At that time Jürgen Richter-Gebert had developed symbolic methods for automatic theorem proving in geometry [RG], and both of them dreamed of a software where one could input geometric configurations with just a few mouse clicks and then ask the computer about properties of these configurations.

Henry and Jürgen started the project on a NeXT platform which at that time was famous for its marvelous software architecture. *Cinderella* became the working title for the project and this title turned out to be unremovable from the project.

A few weeks of development produced the first working prototype. The program was based on principles from projective geometry and invariant theory. It was able to find *readable* algebraic proofs for many theorems of projective geometry about points, lines and conics [CrRG].

However, as a platform NeXT gradually declined in popularity, and with it the initial enthusiasm for *Cinderella*. After the summer of 1995 almost no further progress was made. At a conference on computational geometry in Mt. Holyoke, MA, it was almost impossible to give a software demonstration due to the vanishing of NeXT computers and their operating system.

In August 1996, right after that Mt. Holyoke conference, we (Ulli Kortenkamp and Jürgen Richter-Gebert, at that time working at the Technical University of Berlin in the group of Günter M. Ziegler) decided to start a new project, based entirely on the platform independent language Java<sup>TM</sup>. At that time the language Java was relatively new, and at first both of us were very sceptical about using an interpreted (presumably slow) language as the basis for a program that requires lots of computation in real-time. But we tried anyway.

The goal of this second project was to have the old functionality (that was available in the NeXT version) substantially extended by features of Euclidean and non-euclidean geometry. We also wanted functionality for geometric loci. Moreover, since Java is designed to be "internet-aware," the new program should be able to run inside a web browser. In particular, we wanted to be able to create student exercises for the web. The theorem proving facilities of the program should be used to automatically check the correctness of the student's solution.

Conferences, competitions and their deadlines are often driving forces for rapid development. A first working version was presented at the "CGAL-startup-meeting" in September 1996 at the ETH Zürich. A second version won the "Multimedia Innovation Award" at the Multimedia Transfer of the ASK Karlsruhe in January 1997.

During 1997 Jürgen Richter-Gebert became an assistant professor at the ETH Zürich. This change forced another break in the development. Ulli Kortenkamp moved to Zürich in September.

At the same time we began negotiating about the publication of *Cinderella*. Originally we planned to polish and finish the second project. However, things turned out differently.

The second version, like other computer programs for geometry, suffered from seemingly unavoidable mathematical inconsistencies. These inconsistencies came from ambiguities in operations like "Take the intersection of a circle and a line." There may be two, one or no intersections depending on the position of the circle and the line. While dragging a construction the program has to decide which of them to chose. This seemingly innocuous ambiguity may lead to terrible inconsistencies in the behavior of a construction. It may happen that while you move a point only a little bit whole parts of the construction flip over.

At the beginning of 1998 it turned out that this problem of jumping elements was indeed solvable. However, it was clear that implementing the theory would not be an easy job. Every configuration had to be embedded in a complex vector space. Results of analytic function theory had to be used to avoid "singular situations." If we wanted to use those new insights we had to rewrite the mathematical kernel of the program from scratch. The program had to perform approximately 20-100 times more computations, a challenge for us and for Java.

We decided to do this and ended up with the third project whose outcome you see here. In a period of unbelievably intensive work (that stretched our patience and that of our families to the extreme) we rewrote the whole program again - thereby tuning the program to higher performance at every possible place.

It turned out to be a good idea to undertake this effort. The benefits of the newly developed theory were much greater than we had originally thought. Based on the new methods we were able to do reliable randomized theorem checking. This proved to be much more useful than the old symbolic methods. It was also possible to generate complete geometric loci by generic methods, which is a novelty to the best of our knowledge.

The present program is a mixture of old geometry from the nineteenth century, complex analysis, our new methods and modern software technology. We hope you will enjoy it as much as we do.

Jürgen Richter-Gebert & Ulli Kortenkamp  
Zürich, December 1998