

## Preface to Cinderella 2.6

*The best thing about the future is that it comes only one day at a time.*

*Abraham Lincoln*

Another five years passed. Five years full of work, projects, ideas, collaboration, teaching, research, . . . During this time Cinderella got involved in many different projects. We took most of them as opportunities to enhance the capabilities of Cinderella in several (sometimes unexpected) directions. Some of these features are still experimental, but many of them are included in the current release of Cinderella.2 and in this documentation.

A course on the interrelation of mathematics and music, taught in 2008 in collaboration with the Deutsches Museum in Munich, was the starting point for a variety of ways to create audio output with Cinderella. The internet portal [www.mathe-vital.de](http://www.mathe-vital.de), initiated in 2007, was another seed for many new developments. Among them support for  $\text{\TeX}$ -like typesetting of formulas, advanced operators for linear algebra and calculus, and advanced drawing of functions. An exhibition together with the Deutsche Museum triggered the development of new ways to deal with images and their transformations. The growing desire to handle 3D objects was the reason to create a plugin structure.

There are still many half official features that are not covered in full detail or at all by this documentation. Among these are support for multitouch devices, graph algorithms, background jobs, and many more. We are quite sure that many of these features will be made officially available later as well.

In a sense between Cinderella 2.0 and the current release 2.6 the project grew up and by now is in a state in which it could be called a device for mathematical visualization that goes far beyond geometry. We hope that you will enjoy creating such in Cinderella, or just *doing maths*, as much as we and our students do.

Another main feature of version 2.6 is the documentation you are just reading. For the first time since a couple of years the documentation again covers the complete official functionality that is available by the current version of Cinderella. The printed version of the manual has almost 500 pages, and you may wonder why we need them to describe the functionality of just one program. There are many answers to this and we will here mention at two of them: The first one is the obvious one – we added so much to Cinderella. The second answer is related to you, dear reader: We tried to produce a manual that is as readable and as accessible as possible. We included about 600 screenshots, many code examples for the scripting language, step-by-step tutorials and also some technical and scientific background information. Although Cinderella is a very mighty tool, many tasks can be performed in amazingly simple manners. After mastering the basics you can browse through this manual and just pick the aspects that are most interesting for you: geometry, physics, mathematical programming, music, function plotting, fractals, etc.

While we prepared the final version of the documentation for Version 2.6 again many people were very helpful. In particular we here want to thank Elena Kohler and Stefan Kranich for their extensive help in the final stages of the manuscript, and Stephan Berndts for his software to translate our documentation Wiki into  $\text{\LaTeX}$ .

A great and very special thanks also goes to Ralph Möllers, the owner of Terzio Verlag. He gave us the permission to use the beautiful pictures of *Ritter Rost* for explaining the image operations of Cinderella. The copyright of these Images is held by [Terzio Verlag](#) and we are really proud to be allowed to use them.

Cinderella is a constant work-in-progress. We are sure that there are still many features that might be a valuable completion of the software. Our documentation might at some places be misleading or even wrong, there may be several bugs or inaccuracies. We are listening: Please just visit our website at <http://cinderella.de> for updates and errata. If you want to report a bug you can easily do so by sending a mail to [bugs@cinderella.de](mailto:bugs@cinderella.de). And if you just want to talk to us: you can contact us at [authors@cinderella.de](mailto:authors@cinderella.de).

*September 2011*

*Jürgen Richter-Gebert, Munich*

*Ulrich Kortenkamp, Karlsruhe*

## Preface to Cinderella 2.0

*I may not always be perfect, but I am always me!*

*Found on a T-shirt*

Seven and a half years have passed since the first release of Cinderella in 1998, and the project went through several metamorphoses that we did not foresee ourselves. Now, we feel that enough new aspects and features were added to justify a new major release: *Cinderella.2*.

In a sense programming Cinderella turned out to be a kind of “never ending story” and there is always one more feature that should be added or another one that could be improved (or, not infrequently, debugged). So, the current version may not be perfect in all aspects, but it has so many substantially new features that it would be a pity not to release it and see what people will do with it.

So, what do you have to expect from the new version? First of all, the most obvious change is that Cinderella is no longer “only” a geometry program (nevertheless the geometry part has been improved significantly). The present release consists of three major parts: *Cinderella* – the geometry engine, *CindyScript* – a functional programming language and *CindyLab* – a physics simulation engine. At first sight, these three parts can be used almost independently from each other. However, *Cinderella* (p. 67), *CindyScript* (p. 219) and *CindyLab* (p. 167) are designed to work hand in hand and to take as much advantage from each other as possible – if you look at our new logo, this logo symbolizes the three parts interacting with each other in various ways. Although it would be an exciting story, we will not explain here how we ended up with this final design, because this would fill too many pages. In short, it was a long process, driven by demands and requests of our users, our own desire for cool software scenarios, several conferences on scientific visualization and multimedia

and last but not least several days and nights in which we were following fruitful paths (or sometimes dead ends).

In the geometry part of *Cinderella* there are many substantial improvements. Transformations and Transformation groups have been added, there are many more tools for constructing conics, it is even possible to construct fractal objects. Also, the direct construction of regular polygons has been made possible. Transformations and transformation groups turned out to be a great help to make more advanced constructions and we encourage the users to really take advantage of those concepts. One of the mostly requested features were “macros”. Cinderella.2 now comes with a copy/paste/redefine concept that facilitates the re-usage of already available construction parts. It is also possible to encapsulate parts of a construction into a toolbar button that can be reused in other constructions as well.

CindyLab provides an environment in which the points of Cinderella constructions may become masses, and in which segments may act like springs or other force-generating objects. Although CindyLab only provides a particle/mass/force simulation paradigm, it is a quite powerful tool. We already had a lot of fun experimenting with solar systems with several suns, strange mechanical devices and simulated billiard tables.

Finally, CindyScript is a functional programming language that was designed with applications in geometry but also other parts of mathematics in mind. CindyScript grew out of the desire to have a kind of function plotter available in Cinderella. Well, as things sometimes happen in computer science one starts with a simple solution for a demand and observes that by similar techniques much more could be achieved. Roughly the history was like: We want to calculate formulas - why don't we use this for manipulating the position of points? - why don't we add control structures? - why don't we add high level matrix operations? - ... and list operations ... and recursion structures ... and even more powerful function plotting? This is how we finally ended up with a full-featured, mathematically oriented functional, real-time, high level programming language. We are quite sure that, so far, we do not even imagine what is possible by fully exploiting the advantages of a dynamic geometry environment in connection to a programming language. At least we are very surprised how our students who already work with CindyScript use it! We encourage all readers to build really cool and unexpected applications with this tool.

Still there are several bells and whistles that do not fall under the above three program parts. For instance Cinderella.2 supports the recognition of hand sketches that makes it possible to use Cinderella with a pen tablet an interactive whiteboard or a PDA.

There are several features that did not make their way to the final release. The decision whether a feature entered the release was mainly made by stability considerations. We will add many of these features one by one over the next few months whenever we think that they work reasonably well. Among the things that will come are: native support for geometric bases, a recording tool (*CINERella*) for geometric tutorial films, a hardware simulator, and many more. So we recommend updating frequently.

It is almost impossible to mention all the people that have been helpful in finishing Cinderella.2 by comments, user feedbacks, beta-testing, etc. Still first of all we want to give a great “thank you” and a big excuse to our families. Finishing Cinderella

took a lot of our spare time and our families very often missed us as fathers and husbands. Uncountable many weekends and nights were sacrificed for finishing yet another feature or chasing yet another bug. We both hope that in the future there will be more time for all the other things that are also important in live.

We also want to thank Dirk Materlik who got involved into the team during his diploma thesis about sketch recognition and later in the [Matheon Visage project](#), and besides his work there helped to resolve many critical design issues (the Inspector (p. 153) wouldn't be what it is without him). Also a great thank you to Gunter Gemmel who in an overnight-hack gave us the gift of an implementation of the PSLQ algorithm (p. 355). Many people have contributed by actively using and commenting several beta and pre-beta versions. Here are some of them, in no special order (and a big excuse to all those who we forgot in this list): Hermann Vogel, Gunter Gemmel, Martin von Gagern, Peter Lebmeir, Vanessa Krummeck, Thorsten Orendt, Andreas Fest, Carola Dietrich, Wolf Dieter Heker, David Bakin, Christof Boeckler, Gerhard Bischoff, Alexander Elkins, Dan Beaton, Camille Wormser, Franz Klement.

Finally there are two people who definitely deserve a special mention. One of them is David Kramer, our copy editor from Harvard. He carefully read every single line we wrote for the documentation and helped us as non-native speakers to end up with an at least linguistically understandable documentation. Thank you David! The second person is, in a sense, almost a member of the Cinderella team. It is Martin Peters, our responsible editor at Springer Verlag. He always had an open ear for our new problems, always understood that we need even more time to end-up with a release version, was always helpful in finding solutions to publishing issues and was extremely active in making several important contacts. And, the most important thing, always gave us the freedom to make our own decisions while trusting that this is the best way to end up with the best possible outcome. Martin, thanks a lot for your patience and confidence!

*May 2006*

*Jürgen Richter-Gebert, Munich*

*Ulrich Kortenkamp, Schwäbisch Gmünd*

## **Preface to Cinderella 1.2**

*Cinderella* is a program for doing geometry on a computer. In its present form it is the product of a sequence of three projects carried out between 1993 and 1998. It is based on various mathematical theories ranging from the great discoveries of the geometers of 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 [25], and both of them dreamed of computer software that would allow one to 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.

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 [5].

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 at Mt. Holyoke College, in South Hadley, Massachusetts, 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. At that time, the language Java was relatively new, and at first both of us were very skeptical about using an interpreted (presumably slow) language as the basis for a program that requires a large amount 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 Technical University (ETH) Zurich. 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 Zurich. This change forced another break in the development. Ulli Kortenkamp moved to Zurich in September.

At the same time, we began negotiating for 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 intersection point to choose. 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 twenty to one hundred times as many computations as previously, 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, tuning the program to higher performance at every opportunity.

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*

*Zurich, December 1998*

<http://www.springer.com/978-3-540-34924-2>

The Cinderella.2 Manual

Working with The Interactive Geometry Software

Richter-Gebert, J.; Kortenkamp, U.H.

2012, XIV, 458 p. 377 illus., 300 illus. in color.,

Hardcover

ISBN: 978-3-540-34924-2