CloneGiz (title of project)

One-sentence description of CloneGiz:

Within the project CloneGiz basic arithmetical calculations are represented via the biological process of cloning and reproduction. CloneGiz is a sample application of WitGiz.

[WitGiz is a collection of experimental interfaces and concrete implementations (also called examples) which provide a more organoleptic oriented representation of mathematical content.]

Author of text and idea for WitGiz and CloneGiz: Nadja Kutz

Development of WitGiz: Nadja Kutz and Tim Hoffmann

Introduction

Since CloneGiz is only a specific application of WitGiz one of the main purposes of CloneGiz is to promote the concept of WitGiz. Therefore we will have to explain what the concept of WitGiz is.

Moreover WitGiz and hence CloneGiz belong to the field of mathematical visualization (MV). They serve as an instrument for exploring the possibilities of MV. In order to understand the impact and concept of WitGiz and CloneGiz we need to say something about the impact and features of mathematical visualization (MV). This will add quite some information to the pure concept of CloneGiz, but we consider this to be an important add-on. In particular it explains the connection to EVIDENCE.

Add-on: About Mathematical Visualization

One can understand the term "mathematical visualization" (MV) in the following way:

Mathematical visualisation is the communication of mathematical concepts by means of visual and sometimes also other organoleptic support.

organoleptic: using the five senses vision, hearing, taste, sense of smell, sense of touch.

So the word "visualization" in "mathematical visualization" or "scientific visualization" should rather be replaced by e.g. "organoleptization" or something similar but since most of the existing "mathematical visualization" is indeed a visualization in the old meaning we keep that term for the time being.

Remark: A lot about what we are going to say about MV is transferable to scientific visualization in general.

On the impact of Mathematical Visualization

With the rise of computer technology the field of "mathematical visualization" (MV) is gaining importance. New forms of representing mathematical content evolved. Computer algebra systems and their graphical possibilities (like e.g. Maple and Mathematica, MuPad, Magma), interactive mathematical applets (like e.g. in 3D-XPlorMath, Cabri, Cinderella), explanatory math videos etc. are extending the former possibilities of mathematical visualization by far. This has consequences on the way mathematics is perceived in **public, science** and even in **math** itself:

On the impact of Mathematical Visualization on society

Up to now mathematics is still regarded as a quite secluded science. This is in strong contrast to the actual importance of mathematics. It may sound exagerated but to a certain amount it is true to say that mathemathical thinking is ruling modern life. Mathematics is interwoven with computer science. Mathematics is the language of natural sciences and also an ordinary non-scientific contemporary human needs to know much more math than 100 years ago. Basic mathematical concepts of information, statistics, optimization, geometry a.s.o. are occurring everywhere. Nevertheless people are often not aware of that fact or repress it. Theories of e.g. climate change or economic development are mostly regarded as being "too complicated", whereas the underlying mathematical principles could be made understandable to (almost) everybody to a great extend. So why are people not "interested" in mathematics? Why do science majors in college find mathematics "hard"?
This has to do with the representation and appearance of mathematics.
With the help from mathematical visualization a lot new representations and sights on mathematics have been opened, nevertheless this field is only at its beginning.

On the impact of Mathematical Visualization on science and math

-Communication within the scientifc community:

The advancement of computer technology over the past 50 years changed the communication within mathematics and science dramatically. The number of publications has increased enormously and it will increase more. This is not only due to an increased interest in mathematical or scientific questions and new and sometimes easier ways to communicate (via e.g. online learning and electronic publications). Also a growing world population is and will enlarge the number of people who are (in principle) able to work in advanced science. Due to the above facts most probably a change in the composition of scientists with respect to gender, ethnic and social origin will take place. All this changes the forms of how and when mathematics is done and communicated. Hence also in mathematics there will be/are new questions concerning the administration of information. Up to now there is still quite an amount of individuals who have a great overview about "what's going on where and how in math", but this will become more and more difficult. Like in other sciences new forms of administrating mathematical knowledge have to and will evolve. Mathematical visualization will play a role here.

-Cognitive changes in mathematical thinking

In addition to the problem of information administration there had been a substantial change in the cognitive aspects of mathematics. The importance of this aspect will increase even more with an increase of different cultural and social backgrounds in the (may be newly to be defined) scientific community.
There is a growing number of mathematical disciplines, where mathematical programs and in particular mathematical visualization is used as an **enhancement** for the cognition of mathematics. For example: Geometers who want to understand the surface they are thinking about, visualize this surface on a computer. People in numerics who want to demonstrate the quality of an approximation show sometimes a picture of their data.
This had gone sofar that the most important task of mathematics, namely the task of finding **evidence** for precise statements has partially been taken over by computers and visualization. Before proofing a statement (i.e. which means having the absolute evidence (in mathematical terms)) some mathematicians nowadays crosscheck with a computer wether evidence is possible. This crosscheck comes in two flavors. There is on one hand the possibility that the computer can actually proof a mathematical problem (this is especially the case for algebraic reformulations done with a Computer algebra system). Here the computer is used in substitution to humans who could (in principle !?) perform the same calculations). In this case the mathematical computer code will usually be cited in a corresponding publication. On the other hand the computer is used for plausibility arguments, i.e. aspects of a mathematical statement are crosschecked with the help of computers. This process is called by a lot of mathematicians "the process of looking for numerical evidence". Or in other words, if a crosscheck of a given mathematical problem on a computer yields an unpromising result then some mathematicians don't even start to look for evidence and/or they check back on their mathematical reasoning. And -what is even more interesting- mathematicians start to **trust** their collegues (and themselves) more if those collegues show besides the mathematical proof also some **numerical evidence**. It often takes a lot of effort to check a mathematical proof and hence plausability arguments had always played a role in mathematical argument. New is that plausability arguments are more and more supported by computers and in particular by visualization.

Concept and goals of CloneGiz and WitGiz

Concept and goals of WitGiz

There are a lot of projects, where different representations of mathematical content are investigated with the help of mathematical visualization. Nevertheless these examples are usually all linked to certain applications. There are no works, which investigate the possibilities of mathematical visualisation in a more open but still systematic and at the same time applied way.

What do we mean by that?

As already mentioned in our explanation about the impact of mathematical visualization there exisits nowadays the possibility that a computer actually can proof (find evidence for) certain mathematical assertions. This possibility is given via the use of socalled computer algebra systems (CAS), which allow to **reformulate** mathematical content. The most wellknown computer algebra systems are Mathematica, Maple and MuPAD.

[**Add-on: What is a Computer algebra system (CAS):** Ordinary computer programming languages (like C and Java) "know" some basic mathematical functions which are essential for programming. There exist a lot of libraries which provide more mathematical functions for a socalled numerical treatment (Especially important for engineering). Nevertheless computer languages are not really appropriate for higher integrated and more abstract mathematical tasks. These tasks can be adressed in part by a CAS, which is a programming environment adapted to mathematical reasoning. Within a CAS certain mathematical content is **reformulated** as a kind of mathematical computer programming language so that the computer can actually perform computations, which are more abstract and which in particular may not involve numbers.]

As we described above the mathematical programming language of a CAS is a **reformulation** of certain mathematical code. We would like to distinguish explicitely the term **reformulation** from **newly represented**, these two terms have an essential different meaning within mathematics, this has to do with the semiotics of mathematics. To some extend mathematics can be thought as a kind of "programming language" which casts certain human reflections into a formal framework. A **reformulation** would now mean to apply a kind of one-to-one translator - each bit of mathematical code has to be mapped unambigously onto the code which gives the reformulation. **Newly represented** means that the mathematical code had been translated with ambiguities or insufficiencies - like the ones one acquires when translating normal human languages. Every **new representation** of mathematical code will not give the original mathematics (in contrast to a **reformulation**) - nevertheless it could describe the original mathematics to a certain extend. A representation may be useful and sometimes even be explicitely desired, like e.g. in order to avoid an information overflow.
The above gives also an indication for the limitations of mathematical visualization. In most of the cases mathematical visualization is always a **new representation** of mathematical content. And - depending on the "freedom of translation" - this representation can (in mathematical terms) be more or less "faithful". Artists or other individuals who use mathematical content in an unorthodox way sometimes

disregard the importance of the faithfulness of a mathematical representation. The above should have made clearer now: if you are representing mathematics not faithful enough, then your project has basically not much to do with mathematics. This not bad per se, it just means that you have to be very careful about what you say about your artistic project and the inclusion of scientific content.

WitGiz is now a project, where the possibilities of a "visual" (organoleptic) **reformulation** of some mathematical concepts are tested. To some extend one could see WitGiz as a more or less systematic way for testing new possibilities for a **visual analogue** of (parts of) a mathematical programming language and such for (parts of) the language of mathematics.

This will be done in the follwing way:

1. by providing new interfaces which are adapted to doing mathematics. So WitGiz includes a collection of new interfaces. For example you can think about the possibilities you have if you want to find a "visual" representation for the act of adding two numbers and what kind of interfaces you could need for implementing your representation.
2. by providing concrete implementations in form of "visual" experiments. These experiments should be done in collaboration with e.g. artists and cognitive scientists.

Remark: With the above structure WitGiz follows the structure of e.g. the projects 3D-Xplor math, Cabri and Cinderella, as these also provide more or less a collection of interfaces and concrete implementations for visualizing mathematics. Nevertheless here the construction of new interfaces and examples is rather restricted - last not least by their technical specifications and mathematical application area (Cabri and Cinderella are made for doing mainly geometry only). In addition the idea of trying to "translate" a mathematical programming language into visual form had in these programms never been the underlying concept. Nevertheless there are certainly features in these programs which point into that direction, as these programs intuitively "translate" certain abstract mathematical content into a visual language like e.g. the feature of a *process* in 3D-Xplor math or e.g the feature of visualizing certain mathematical proofs for 2D geometry in Cinderella.

The construction of the interfaces of WitGiz, as well as the construction of the concrete implementations, is interwoven. That is - often a concrete implementation shows what kind of interfaces are needed, on the other hand a given set of interfaces will allow to construct fast explicit implementations and will trigger ideas for an implementation. Hence the concrete implementations of interfaces serve as a think tank for the design of the interfaces and vice versa.

Moreover the concrete implementations of WitGiz are not only important for experimenting with the functionality of the interfaces but also for demonstrating the powers of WitGiz in finding a visual reformulation for the underlying mathematical language.

By doing these experiments also new forms of **representations** in contrast to **reformulations** evolve with great control about their faithfulness. Not everything will find a reformulation but as indicated above a representation may also be useful and sometimes even be desired.
The project CloneGiz is a concrete implementation of WitGiz.

Impact and goals of WitGiz:

The impact of WitGiz on mathematical visualization and the impact of mathematical visualization was indicated above. For obvious reasons (universality of mathematics) WitGiz will most probably also be useful for the visualization in other disziplines.
In particular questions of complicated interactions of and with physical models and a visual representation of the functionality of a physical model will automatically be adressed by WitGiz (any physical model is just a mathematical program). In this sense WitGiz could in principle provide a new type of physical programming interfaces. This will be also of interest to industrial applications.
We expect that WitGiz will make new contributions to the field of interface design in general.
With WitGiz new implementations of examples for the eduction of science can be constructed.
Last not least WitGiz will enhance the possibilities of artistic expression.

CloneGiz

CloneGiz is a sample implementation of WitGiz so all what was said about audience and impact of WitGiz can be partially taken over. We would call CloneGiz a digital art project. Nevertheless is bears also some educational value.

-General features:

Within CloneGiz basic arithmetical calculations are **represented** via the biological process of cloning and reproduction. I.e. CloneGiz represents expressions like

$$3 \cdot f + 4 \cdot (g - 2 \cdot h)$$

in terms of a reproduction rule. This works in the following way: Let f be a name for a picture of a human or parts of a human creature. Lets assume f is a name of a front view of a specific human face belonging to a human with the name "F". g and h should then also be names for front views of human faces belonging to the humans "G" and "H". Imagine F, G and H are genetical parents of a clone. **The result of an arithmetical calculation will then be represented by a view of the face of a clone of the corresponding parents.**

In particular the expression $2 \cdot f$ will be represented by the view of a face of a cloned child as understood in nowadays world - namely a child $2 \cdot F$ which is an identical copy of its parent F, hence the pictures $2 \cdot f$ and f are identical. In general the pictures $factor \cdot f$ and f will be identical.

The expression $2 \cdot f + g$ will denote a view of the face of a "clone" named $2 \cdot F + G$ whose "genetic material" is a weighted mixture of the genes of the parents F and G. Here F has double influence on the composition of the cloned child (correponding to the factor 2). This means that the cloned child $2 \cdot F + G$ resembles rather F than G.

The expression $3 \cdot f + g$ would correspondingly denote a view of a face of a clone where the parent F has triple influence. $f + g + h$ shows a clone were every genetic parent has the same impact on their cloned child. $f - g$ shows a cloned child which has no resemblance to its parents, but where the genetic features of G "revert" the features of F a. s. o. Of course there exists in principle further possibilities of including more mathematical and reproductional features.

- Implementation of CloneGiz:

CloneGiz has sofar been realized as a Java applet. The applet encompasses a drawing window with a front view of a default face drawn by a fixed number of lines, a main window, a socalled parent bank window with front views of other default faces together with their corresponding names and a command line.
The face lines in the drawing window can be altered by dragging interpolation points. Hence starting from a default face (default configuration) one can "draw" (or rather "drag") a new face by dragging the interpolation points in the desired way. The new configuration can be saved and loaded into the parent bank, where a name will be assigned to it (like f). One can now create the face of a clone by typing an arithmetic expression like $2 \cdot f + g$ into the command line. The result of the expression (i.e. the face of the clone, which is a "weighted mixture" of the faces of f and g) will appear in the main window.

Remark: In addition to the mathematical features one of the intentions of the applet is to show the absurdity and ridiculousness, which lies behind the idea of trying to "form" a specific type of human. In particular we are **not modelling** realistic genetic manipulations - these are much more complicate and sofar more or less uncontrollable. As a consequence our applet can't be abused by racist groups - the cloning results will by principle not conform to their ridiculous theories.

Collaboration with Vectors:

In designing our default faces for the parent bank, we had to rely on our limited abilities to draw faces (or bodies or body parts). Since we submit CloneGiz also to other occasions we implemented a default parent bank, i.e. a collection of front faces, which were drawn by Nadja Kutz. Nevertheless for the emotional component of CloneGiz it would be important to have faces which resemble human faces in a catching way- especially if they are drawn with only a few lines. That means CloneGiz, as it is now, is a more or less "finished project" which needs improvement. Thus we would like to work together with artists mainly from the area of comics or animation. The artists task should be to find expressive default faces for which we could include more features (like shading etc.). We also would like to discuss our "window-commandline interface" with experts. So the timeline for CloneGiz is depending on the artists and the features they want to have included. The above decribed appearance of CloneGiz is only the beginning - we imagine to have may be 3-D virtual clones in a virtual environment, more mathematical features such as abstract functions (represented e.g. by "surgery") and also other interfaces as e.g. for manipulating various body parts. We could imagine that the collaboration could be extended to that future part and also to other sample implementations.

Add-on: Technical details: WitGiz, JSYM and jReality

The main purpose of WitGiz was to find a "visual" representation (more or less faithful) of parts of a mathematical programming language. This mathematical programming language is not one of the major existing mathematical programming languages like e.g. the one of Mathematica. This is mainly because the syntax of e.g. Mathematica programming language is not appropriate for such a task. One needs a mathematical programming language which is more addapted to a visual representation and various forms of interaction. In particular this language needs to be more object oriented than the existing mathematical programming languages. The mathematical programming language underlying the construction of WitGiz is JSYM - a java based object oriented symbolic programming language.
JSYM (development sofar: Tim Hoffmann) is developped parallel to WitGiz, so there is a mutual influence onto each other. This allows to design a symbolic language, which is adapted to "visual" mathematical interfaces. This is in contrast to other object oriented symbolic languages. It clearly enhances the possibilities of WitGiz. In addition it means to start with very basic mathematical functionalities while having the possibility to adress their integration in terms of visual and interactive representations. That this is possible in a very structured way is of course also due to the object oriented form of the program.
jReality (development Tim Hoffmann, Charles Gunn, Holger Pietsch, Markus Schmies, Ulrich Pinkall) is a platform independent 3d viewer in java. Platform independent means that graphical objects can interactively be viewed in such different environments as the internet or a virtual reality theater (at the math department of the TU Berlin we have a socalled PORTAL virtual reality theater). WitGiz uses jReality for its visual output, so WitGiz will be more or less platform independent.
WitGiz, JSYM and jReality will be licenced under the GNU public licence.
WitGiz and JSYM are developped in close contact to the research project F5 at the centre for mathematics in key technologies at the technical university of Berlin. jReality is part of F5.