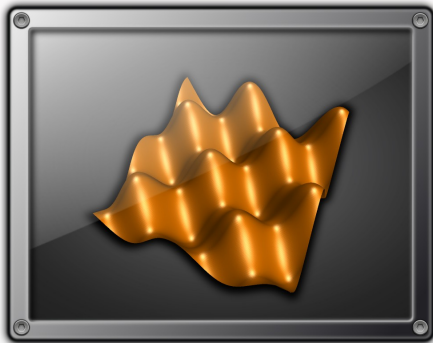


VPlot & VRL-Studio

First OpenJFX 8 App in the Mac App Store



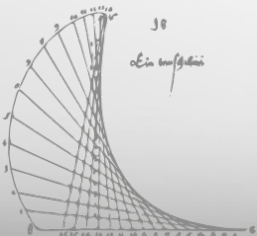
Devoxx

Nov. 12, 2013
Antwerpen

Michael Hoffer

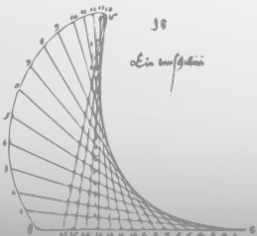


G-CSC, Uni Frankfurt, Germany



Outline

- About Me
- VPlot & VRL-Studio
- Building OpenJDK 8 & OpenJFX 8
- Submitting to the Mac App Store



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



About Me

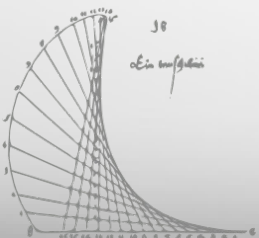
- Doing my PhD at the G-CSC, University of Frankfurt
- Research Interests:
 - Developing Visual Programming Concepts
- Software Projects:
 - VRL-Studio, VPlot, ...
 - Vworkflows
 - JFXtras
 - **DBF:BrainLab**, Event Handler, ...

Twitter: @mihosoft

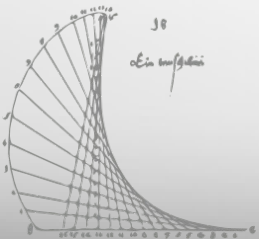
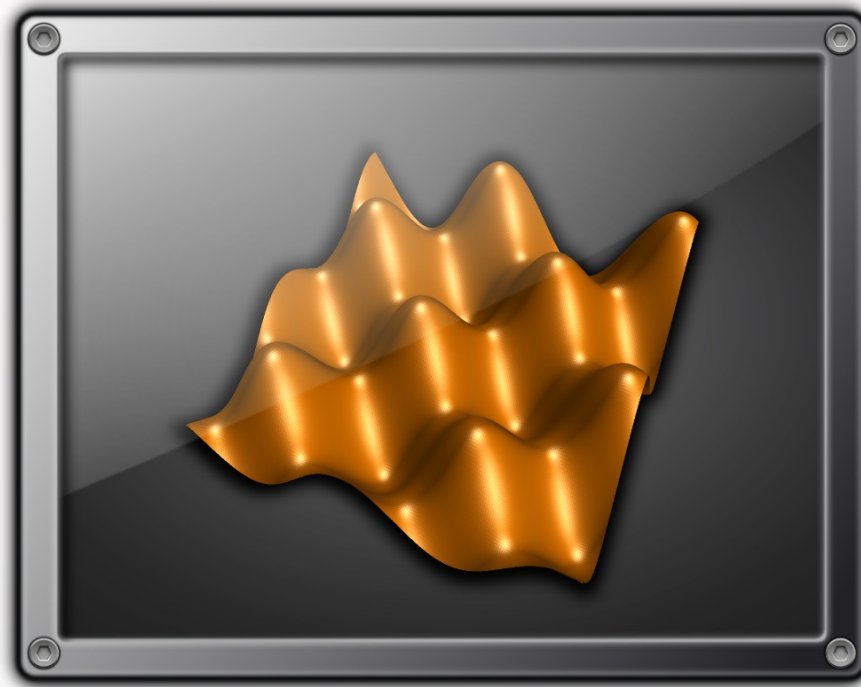
Web: mihosoft.eu

Michael Hoffer

G-CSC, Uni Frankfurt, Germany



VPlot



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



VPlot

By Michael Hoffer

Open the Mac App Store to buy and download apps.



[View in Mac App Store](#)

Free

Category: [Developer Tools](#)

Released: Nov 09, 2013

Version: 1.0

Size: 66.4 MB

Language: English

Seller: Michael Hoffer

© 2013 Michael Hoffer

Rated 4+

Compatibility: OS X 10.7.4 or later, 64-bit processor

Customer Ratings

We have not received enough ratings to display an average for the current version of this application.

More by Michael Hoffer



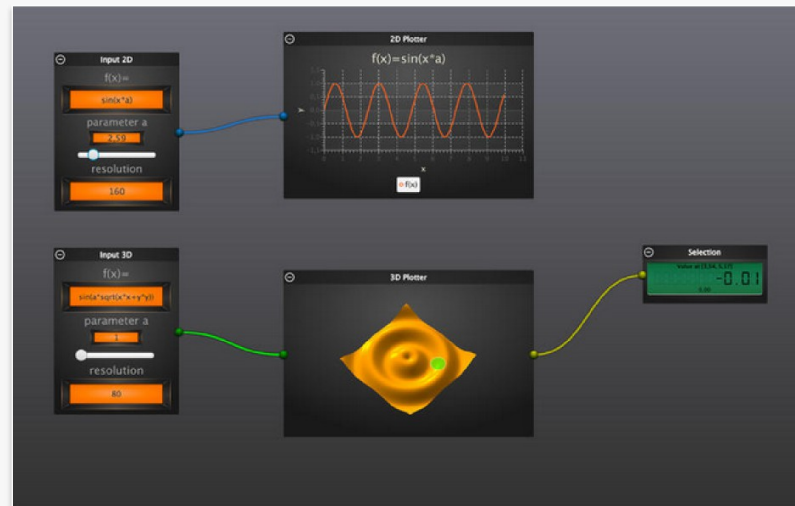
Description

This app provides a 2D and a 3D function plotter. It is ideal for teaching! It enables the user to interactively plot parameterized 2D and 3D functions.

[VPlot Support](#)

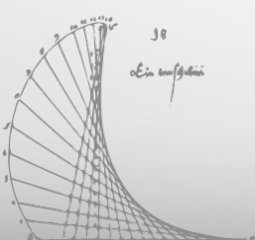
[...More](#)

Screenshot

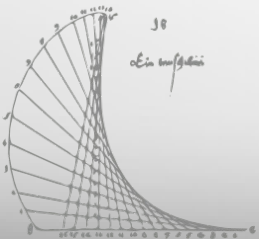


Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Demo

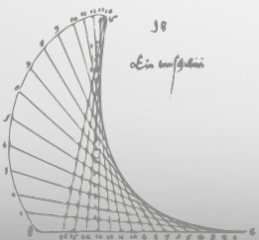


Michael Hoffer

G-CSC, Uni Frankfurt, Germany



VRL-Studio



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



VRL-Studio

Innovative, intuitive and powerful Visual IDE for rapid prototyping, learning, teaching and experimentation.

[Download VRL-Studio](#)

[Public PGP Key](#) [PGP Signature](#)

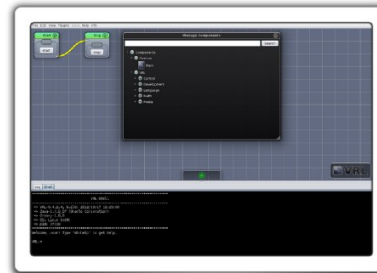
[GitHub project](#) [Examples](#) [All Versions](#)

developed by Michael Hoffer @ G-CSC

Introduction

VRL-Studio combines both visual and text-based programming. In contrast to many other Development Environments VRL-Studio Projects are fully functional programs that are developed at run-time. This makes it an ideal environment for rapid prototyping, teaching and experimentation.

VRL-Studio is based on the Visual Reflection Library (VRL). VRL enables declarative and fully automated creation of graphical user interfaces from Java objects. To accomplish that, VRL uses the information accessible via the Java Reflection API.

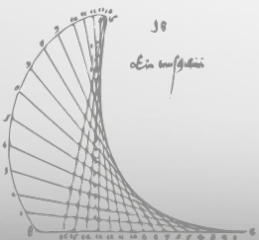


In addition to the automatic generation of user interfaces VRL-Studio supports the definition of data-flow and control-flow and allows for a certain degree of visual programming.

For text-based programming the language Groovy is used. It supports static typing, dynamic typing and meta programming. The syntax of Groovy is based on Java. Almost any Java code will just work without any changes. Even though Groovy is often used as scripting language, it provides a compiler that creates byte code that is fully compatible with the JVM. Therefore, Groovy classes are accessible via the Java Reflection API.

Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Examples

The following example projects give an overview over the possibilities VRL-Studio offers. A lot more is possible! Contact us if you want to contribute your amazing project!



2D Function Plotter

This example demonstrates how to create a 2d function plotter in VRL-Studio.

[Download Example](#)



Sound Generator

Download this exciting example to learn how to generate sound from arbitrary functions.

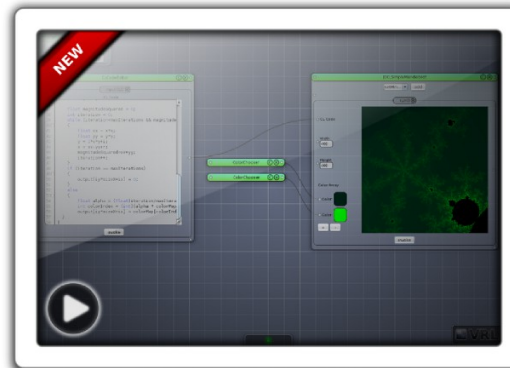
[Download Example](#)



3D Function Plotter

This example teaches you to create a powerful 3D plotter.

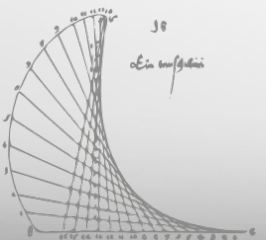
[Download Example](#)



Mandelbrot with OpenCL

VRL-Studio provides access to the GPU via OpenCL! Checkout this example to see how easy it is.

[Download Example](#)



Michael Hoffer

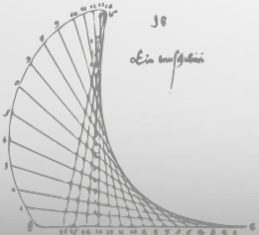
G-CSC, Uni Frankfurt, Germany



The screenshot shows a Java IDE with two windows. The left window, titled "Code of class SampleClass (*)", contains the following code:

```
1 package eu.mihosoft.vrl.user;
2 @ComponentInfo(name="Sample Class", category="Custom")
3 class SampleClass implements Serializable {
4
5     private static final long serialVersionUID=1;
6
7     public String hello(){
8         return "Hello, World!";
9     }
10 }
11
```

Below the code is a "Compile" button. The right window, titled "SampleClass", shows the execution of the `hello()` method. The output is displayed as "String: Hello, World!" with an "invoke" button below it.



Michael Hoffer

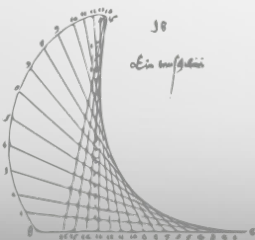
G-CSC, Uni Frankfurt, Germany



The image displays two windows from a Java IDE. The left window, titled "Code of class ColorChooser", shows the following code:

```
1 @ComponentInfo(name="Color Chooser", category="Custom")
2 class ColorChooser implements Serializable {
3
4     private static final long serialVersionUID=1;
5
6     public Color choose(
7         @ParamInfo(name="R", style="slider") Integer r,
8         @ParamInfo(name="G", style="slider") Integer g,
9         @ParamInfo(name="B", style="slider") Integer b) {
10
11         return new Color(r,g,b);
12     }
13 }
14
```

Below the code is a "Create Instance" button. The right window, titled "ColorChooser", shows the graphical user interface. It features three sliders for Red (R), Green (G), and Blue (B) components, each with a range from 0 to 255. A "Color:" label is next to a color preview box, which is currently red. An "invoke" button is located at the bottom of the window.

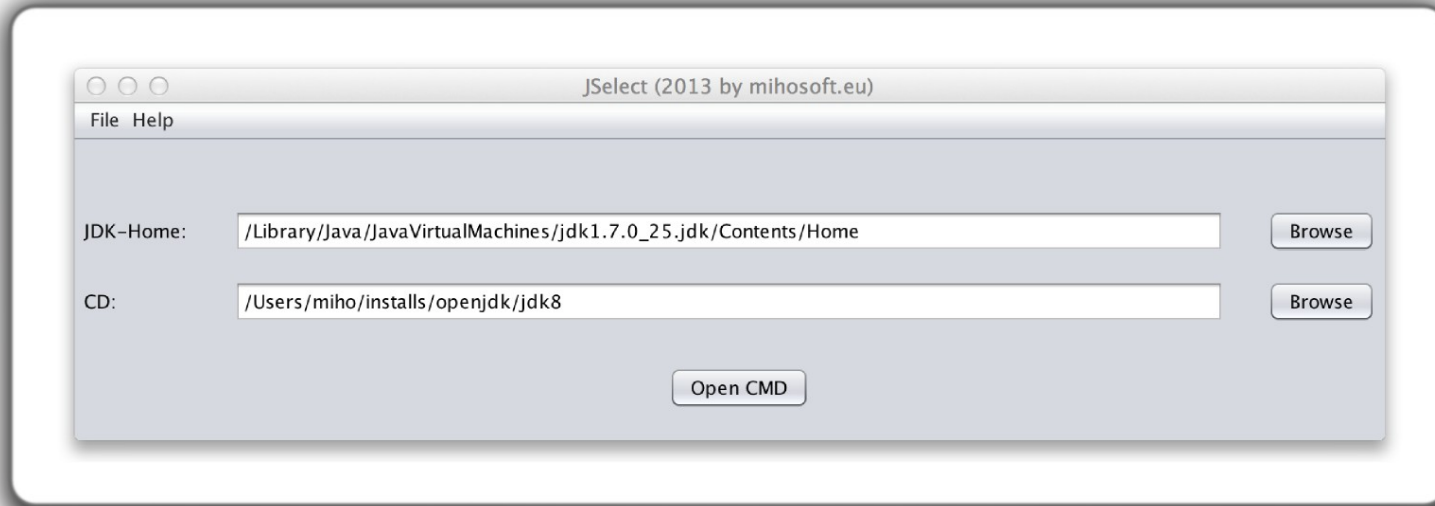


Michael Hoffer

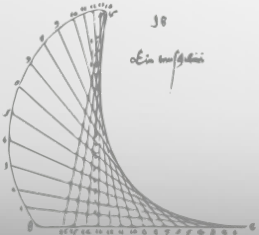
G-CSC, Uni Frankfurt, Germany



Building OpenJDK 8



```
java version "1.7.0_25"  
Java(TM) SE Runtime Environment (build 1.7.0_25-b15)  
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)  
  
#>_
```



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



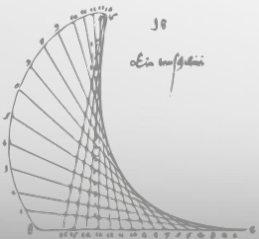
Building OpenJDK 8

```
#> java -version
java version "1.7.0_25"

#> ./configure \

--with-debug-level=release \
--with-milestone="1" \
--with-boot-jdk=/Library/.../jdk1.7.0_25.jdk/Contents/Home/ \
--with-build-number=109

#> make clean images
```



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Building OpenJDK 8

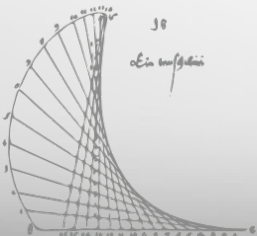
1.) select correct xcode version:

```
#> sudo xcode-select --switch ../../Xcode4.6.3.app/../../Developer
```

2.) maybe force xcode to reinstall CLI:

```
#> sudo pkgutil --forget com.apple.pkg.DeveloperToolsCLI
```

3.) reinstall CLI



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Building OpenJDK 8

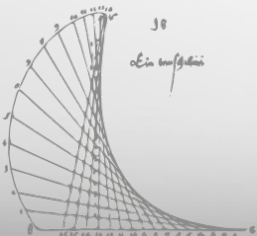
1.) select correct xcode version:

```
#> sudo xcode-select --switch ../../Xcode4.6.3.app/../../Developer
```

2.) maybe force xcode to reinstall CLI:

```
#> sudo pkgutil --forget com.apple.pkg.DeveloperToolsCLI
```

3.) reinstall CLI

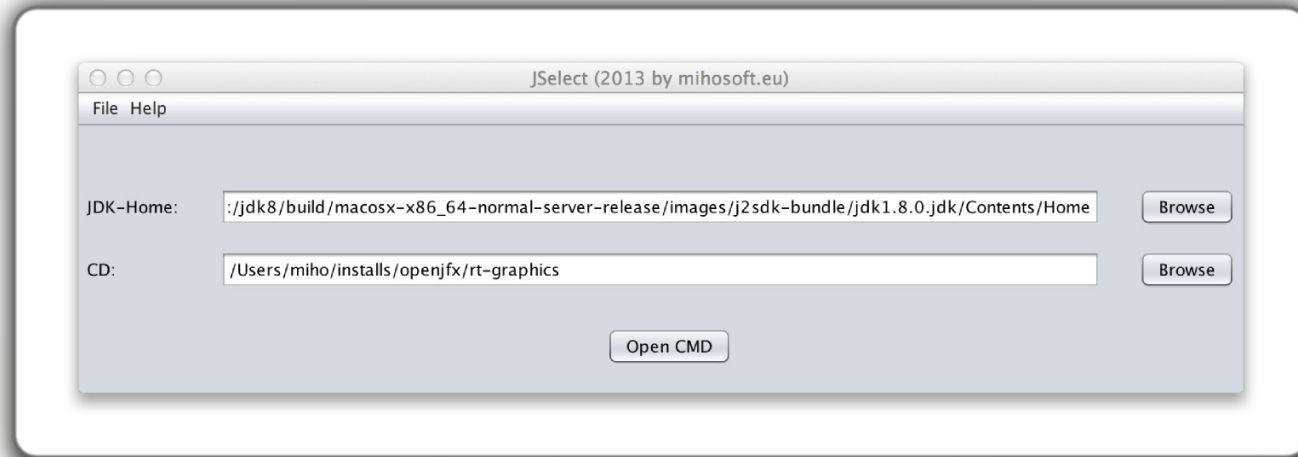


Michael Hoffer

G-CSC, Uni Frankfurt, Germany



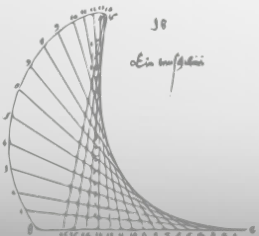
Building OpenJFX 8



```
openjdk version "1.8.0-internal"  
OpenJDK Runtime Environment (build 1.8.0-internal-miho_2013...)  
OpenJDK 64-Bit Server VM (build 25.0-b54, mixed mode)  
  
#>_
```

Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Building OpenJFX 8

1.) `gradle.properties` (specify correct sdk version):

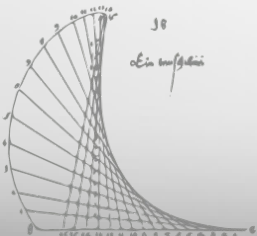
```
#> echo "MACOSX_SDK_PATH = /Applications/Xcode4.6.3.app/ \
Contents/Developer/Platforms/MacOSX.platform/Developer/ \
SDKs/MacOSX10.7.sdk" >> ./gradle.properties
```

2.) `build.properties`

```
#> echo "BINARY_STUB=./stubs/b109/jfxrt.jar \
jfx.build.jdk.version=1.8.0 \
jfx.build.jdk.buildnum=110 \
jfx.build.jdk.buildnum.min=1" >> build.properties
```

3.) finally, build openjfx:

```
#> gradle -PBUILD_NATIVES=true -PBUILD_WEBKIT=true clean sdk
```



Michael Hoffer

G-CSC, Uni Frankfurt, Germany

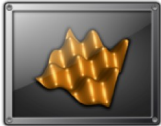


Submitting To The Mac App Store

VPlot

By Michael Hoffer

Open the Mac App Store to buy and download apps.

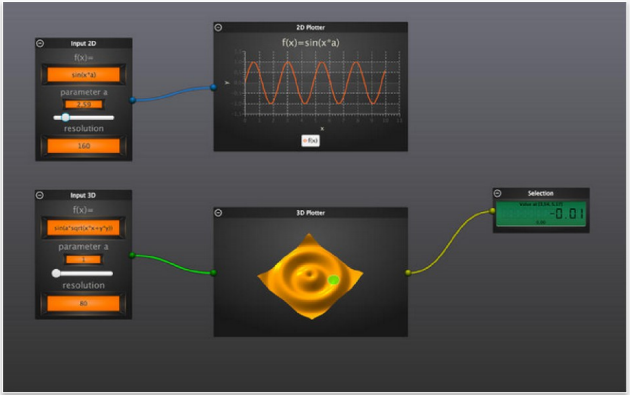


Description

This app provides a 2D and a 3D function plotter. It is ideal for teaching! It enables the user to interactively plot parameterized 2D and 3D functions.

[VPlot Support](#) [...More](#)

Screenshot




Free
Category: Developer Tools
Released: Nov 09, 2013
Version: 1.0
Size: 66.4 MB
Language: English
Seller: Michael Hoffer
© 2013 Michael Hoffer
Rated 4+

Compatibility: OS X 10.7.4 or later, 64-bit processor

Customer Ratings

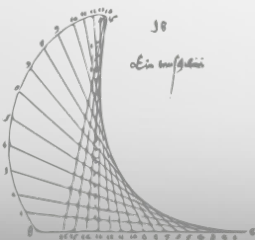
We have not received enough ratings to display an average for the current version of this application.

More by Michael Hoffer



Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Submitting To The Mac App Store

1.) signing the bundle:

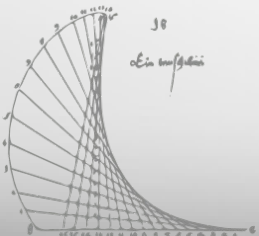
```
#> codesign -f -v -s \  
"3rd Party Mac Developer Application: NAME" \  
VPlot.app --entitlement VPlot.entitlements
```

2.) creating the pkg file:

```
#> productbuild --component VPlot.app /Applications \  
--sign "3rd Party Mac Developer Installer: NAME" \  
--product VPlot.app/Contents/Info.plist VPlot.pkg
```

3.) testing the pkg file:

```
#> sudo installer -store -pkg VPlot.pkg -target /
```



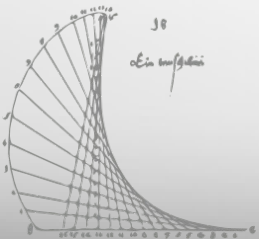
Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Tips for App Store Approval

- Be careful with the description!
- Never use the word „demo“!
- Watch menu names
- Implement „correct“ window behavior



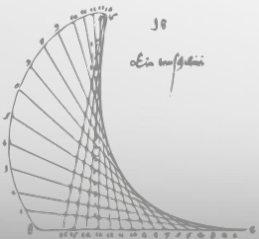
Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Links

<http://devoxx2013.mihosoft.eu>

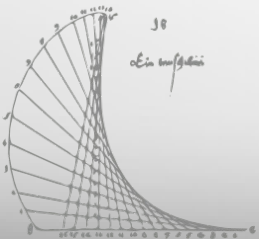


Michael Hoffer

G-CSC, Uni Frankfurt, Germany



Thanks!



Michael Hoffer

G-CSC, Uni Frankfurt, Germany

