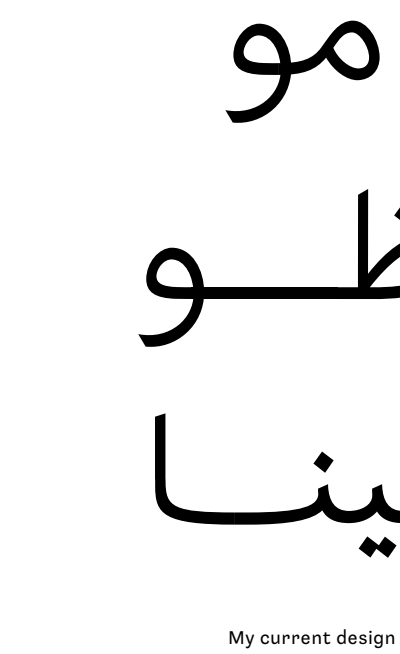
TABLE OF CONTENT

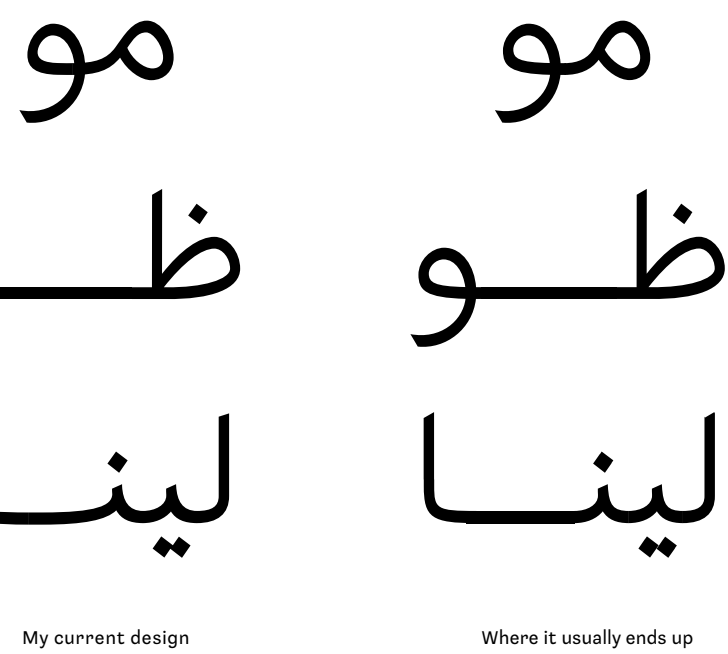
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnnnnnnn
oooooooooooooooooooooooooo
oooooooooooooooooooooooooo
oooooooooooooooo
eeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeaaaaattttttttttt

none none none
none none none
none none none
none none none
none none none
none none none

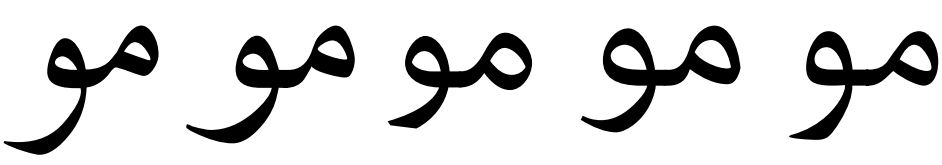
1. **Do we need new tools?**
Making a typefaces has never been easier than today. The tools allow for incredible fast development of high quality typefaces. On the same time it seems that they are continuously improving and I can not see that there will ever be an end to it. Especially Non-Latin scripts are contributing immensely in the development of tools, which does very often equally improves the workflow for the Latin script. (e.g. many opentype features)
I learned this year very clearly that companies often can only spend money to make things work, not to make them work nicely – which is only slowly becoming a more and more important selling argument. But there are many more reasons that are outside, it could be just the size and structure of the company that makes it hard to adjust to new needs. To sum up, there are many things



My current design



Where it usually ends up



Bustani Karim Nassim Adobe Nazanin

The final waw almost always lifts considearable on the left, which is quite unusual in manuscripts. But the break with the manuscript is not as crucial as the break of the continous flow of the stroke. Out of this examples Bustani does hide it the best, but also there it could have been a even nicer curve if the concept of contextual alteration would be more prominent in the current tools.

which might prevent improvements or new tools to prosper which have nothing to do with the expertise of the individuals in a company. In conclusion ideas often die in the drawer and mediocre solutions stay. But creating software gets also easier and much less people are needed to develop. Therefore I believe that there are still a lot of improvements in the technology of typesetting and type design/development we can look forward. And overall I believe that the designers can only benefit from a dense variety of different tools.

1.1 **Specific ideas developed**

I find it very interesting how the tools affect not only the design but also the thinking of the users. Grown up with a computer and being a user of type related technology for quite a while I came across many unsatisfying solutions. It can be especially frustrating if you are trained as an interface designer and have a good understanding of the underlying technology. Even though it slowed down my workflow significantly – I started to always make notes if things don't work as I would expected or if I have ideas for new tools. Ultimately it it is for the sake of my own good in the future, so I hope I invested well. Here is a short list of some of my ideas. The rest of this document I will then dedicate to one specific idea and the prototype I build in this year.

Stroke-Kerning (e.g. for Arabic)
Fontlab, Robofont and Glyphs are not very comfortable for contextual designs, like script-grammer in Arabic. This tool aims to make it comfortable to adjust the fitting of connected characters contextually. The difficulty to make nice kashidas is just one example that gets adressed here.

Font-CSS
Giving the typographer more control and flexibility to react to specific typographic contexts. Similar to the interaction of Metafonts with their typesetting environment

Merge Type Tool (tyographic sets / contextual families)
Rather then mixing typefaces and sizing them according to one another in the typesetting software it would be much more comfortable (and compatible for even the most plain text editing environments) to make typographic sets and package them as a font/font-family. This includes mixing different scripts of , styles, setting up different soft and strong emphasis, . and could even be more comfortable to also specify optical sizes or specify as standard cut for headings, subheadings,.. And even custom contextual styles like characters of a play. But maybe it is even more comfortable to define a new typeset-format, which is human-readable and easy to edit where you can specify all those things and more. It could basically include the Font-CSS capabilities as well.

Search Type
A visual typographic-sensitive search engine that solves the currently impossible and very tiring task to find the type you are looking for. Basically what Google is for text, but for typefaces.

Non-Linear-Interpolation
This is actually a very well implemented technique in animation software like After-Effects. There you can modulate the journey, in form of a path, from one static state (Master 1) to another (Master 2). This might be very important to actually create flexible (but manageable) interpolations that actually create the instances a type designer can be happy with.

Bezier-Tool 2.0
Adding the possiblity to have an additional off-curve node would improve significantly the transition of curves into straight lines. Quadratic curves where already slightly better then quadratic beziers in this context. Unfortunately most type tools already stopped supporting quadratic curves, what is not very promising for any developments in this direction.

nln odo non another editi

donhrtilae

another edition the ano that on not he are or one there he other edition than then tate three it in another here too en rather tea the north art heart deal near anilla attent the earth hear did threat there entree hear error none thread tart too area ort dealin real lador lode nodellin ron note the tone tree northern enter at treate ten hot her rate eat rat train antra hide tellee handhold idol dodo in tarradiddle oh nada ianthinite toilinette diradiation rehearten an trattoria hare tod tarantelle alit detin rare landhold alditol loth rid taratantara nee irredential readl

another edition the ano that on not he are or one there he other edition than then tate three it in another here too en rather tea the north art heart deal near anilla attent the earth hear did threat there entree hear error none thread tart too area ort dealin real lador lode nodellin ron note the tone tree northern enter at treate ten hot her rate eat rat train antra hide tellee handhold idol dodo in tarradiddle oh nada ianthinite toilinette diradiation rehearten an trattoria hare tod tarantelle alit detin rare landhold alditol loth rid

nnnonnnnoonnn
nnnonnnnoonnn

donhrtilae

nondnlninhntnrnanen
nnnnnnnnnnnnnnnnnnnnnnnnnnn
ooooooooooooooooooooooooooo
nnonnnonnnnnoonnnnnnnooonnnnnn
ooo ono oho oro oto oio olo odo oao oeo o-o non nnn nhn nrn ntn nin nln ndn nan nen n-n hoh hnh hhh hrh hth hih hlh hdh hah heh h-h ror rnr rhr rrr rtr rir rlr rdr rar rer r-r tot tnt tht trt ttt tit tlt tdt tat tet t-t ioi ini ihi iri iti iii ili idi iai iei i-i lol lnl lhl lrl ltl lil lll ldl lal lel l-l dod dnd dhd drd dtd did dld ddd dad ded d-d aoa ana aha ara ata aia ala ada aaa aea a-a eoe ene ehe ere ete eie ele ede eae eee e-e

another edition the ano that on not he are or one there he other edition than then tate three it in another here too en rather tea the north art heart deal near anilla attent the earth hear did threat there entree hear error none thread tart too area ort dealin real lador lode nodellin ron note the tone tree northern enter at treate ten hot her rate eat rat train antra hide tellee handhold idol dodo in tarradiddle oh nada ianthinite toilinette diradiation rehearten an trattoria hare tod tarantelle alit detin rare landhold alditol loth rid taratantara nee irredential readl iterate retter rid darii iddat airan tolite horded london tatther holer on in araneidal taro or irritated an a on leant annelid let hid note tinnet lernaeoid eleonorite eth anhalonidine enneahedral internal attentional railroader atlantoodontoid tiarella hi earlier earn retardant halo inn trio retanned or loot nil enate hoard enolate ratio not attn looter idol add arid titan a alidada hone oil artlet nitridation the inattention odorator tat trot altair retrier a lain trilled earn or oared tad hairline dart deadlihead atoned

onhrtildae-

45

1 1.5 2 2.5

0 .5 .6 .7 1 2 3 4 5 6 7 8 9 10

1 2 3 4

Donatello Din

Norma Rustika

Donatello Din Optimus Olaf Bierkrug Bra Ninja Norma Rustika Rika Odin Oberin we're closing soon you work here then no son I'm a vandal I'm defacing the property don't tell anyone thought you might be a workman or something no I run the place or rather it runs me into the ground get a lot of that do you bloody nuisance it is and I can never catch the buggers in the act danny continues to look at the rest of the ecshibits lawson removes the rest of the graffiti after a few moments what d'you think about what about what you see I think it's alright all the things you've got on show all the knickknacks knickknacks our colour scheme

Donatello Din Optimus Olaf Bierkrug Bra Ninja Norma Rustika Rika Odin Oberin we're closing soon you work here then no son I'm a vandal I'm defacing the property don't tell anyone thought you might be a workman or something no I run the place or rather it runs me into the ground get a lot of that do you bloody nuisance it is and I can never catch the buggers in the act danny continues to

Bezier-Clay-Tool
If you draw or work with a chisel or in clay you can make any imaginable curve without worrying about complexity. What if we could combine this approach with bezier curves – so you could switch at any time from one editing-mode to another. Imagine you could just scratch off this little bump there and then go back to bezier-curves. Or even better – what if you could make the rough decisions with a simplified version of the bezier curves (eg. Only extreme points) and work out the details in a drawing mode with a pencil. And it even gets better - you could use the information of the simple beziers to make high quality interpolations while keeping the details. This could improve significantly the interpolation quality of complex outlines.

Fitting-Toolset
This would actually incorporate several small tools that help to make the fitting right in any script. 1) With this you could make a different fitting for tight tracking and for loose tracking. 2) With this you could start out to make the entire fitting contextually and then at anytime convert the repeating kerning pairs to spacing on export to keep the kerning table as small as needed. So basically a tool where you start to make words look good first and then see if the design allows to reduce or even entirely get rid of those kerning pairs. 3) A mode to focus on an even distribution of whitespace and defining decisions/exceptions on how to deal with character combinations which do not allow to keep the aimed size of whitespace. 4) Defining a minimal distance between shapes, so that letters never touch, including a possibility to make exceptions. / .

Reflecting on current Drawing-Tools
Beziers are very powerful to manipulate curves but on the same time also very limited and counterintuitive. I argue that professional type design software could benefit hugely from implementing more drawing tools, such as nibs, that also allow to change angles, etc. And path manipulation tools that are more similar to analog drawing and scratching tools. At the moment digital type design is purely visual and is almost completely absent of any body movement you do in e.g. Stonecarving, Lettering and calligraphy.

Drawing and erasing like in Photoshop
Really great but unfortunately quite still quite tedious in Photoshop, in comparison to most common bezier approaches. Even though there are possibilties to intervine both approaches. Macromedia Flash already hat a pen and erase tool where each line added up to an outline-area, rather than to outline-strokes.
It would be specifically helpful if you could simply scratch off small bumps with a pen tool, instead of bothering with the underlying beziers, which usually anyway means to add nodes in between extreme points, which is always a bit tricky.

Nib-Tool, already developed but still missing
Outline-Beziers are definitely not best tool to begin with for e.g. An calligraphic design. There, the best thing would be to start with a digital nib, that allows to change angles and everything else an analog nib would allow you to do with.
Most versatile high quality tool in this repect seems to be Kallikulator developed by Frederik Berlaen ten years ago, which is unfortunately still not publicly available.

Intuitive Drawing with bezier-curves
Once you got the points roughly placed it is very easy and quick to control the outline and to get the points where you want them to be – But why is it still quite counter-intuitive to place the nodes quickly in a way that it is already much closer to what you wanted to achieve?
In this area there are quite some experiments made. Glyphs and Robofont tried to address it but did not succeed yet, as it is still rather difficult to control. But Fontlab 6 indeed has some interesting features which seemingly go into a much more intuitive direction. (Seen at the Typolab in Berlin)

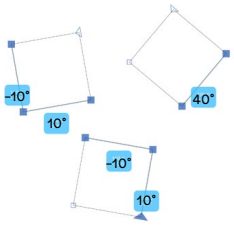
.

Misc
Actually there are quite a few more in my notes, but it does not make any sense to just list them up as it is already on the edge of being comprehensible with a whole paragraph.

1.2 Catalogue of qualities

There are many things I learned to be considered to make a great tool: a strong core idea, clear context, a set of personas, a good business concept, marketing, etc. — All of which I will not cover in this section.
I will try to examine different aspects of qualities, mostly concerning accessibility which I realised is of very high importance for any tool in a professional scenario. A fast and simple access might be one of the most important features and might decide weather a tool is useful to integrate in ones workflow at all.

Readability
It is of immense importance that the interface and therefore also the visualisations are as easy to read as possible. The less noise the less unnecessary small difficulties of interpretations, the more capacity the user has to concentrate on what he wants.



For example I needed to control the consistency of certain stroke angles to get a similar rendering. This seemed impossible to me with the original setting of the angle-visualisation tool by Mark Frömberg which does already only display a -180—180° range, instead of 360°. I simplified it even further down to -45—45°, which was much easier for me to process, as the visual context already explains if it is a horizontal or a vertical, etc.

Keys
In this respect the keys (on keyboard and mouse) are by far the fastest and most flexible input device on a computer. Shortcuts (customisable) are in this respect the most obvious player and are perfect for very frequent tasks. An accessibility issue here is still that on most keyboards you have no visual reference on the keyboard of what is going on, which might change once the hardware to integrate displays on a good keyboard get cheaper.

Text-Search
Right after shortcuts this is the fastest way to access any menu on a computer. Just access the search [cmd shift /] and search for the menu item. This is very powerful and perfect for everything where a shortcut does not make sense. Still not all applications do support or the menu items are not descriptive named which makes it hard to find them. Other applications have it build their own search like Indesign [space enter] or in coding applications like Sublime the implemented search engine is even better then the OS-search and an inherent core feature.

Context and Direct Manipulation
If an object you are treating is visual than by nature the fastest way to make decisions is visually within the specific context – and not through non-contextual indirect input e.g. numbers to determine the weight of an instance or changing the form of a component outside its used context.

Simplicity
Simple tools that where you know exactly what they do are the fastest to use. This is also why for one cut of latin usually the fastest way to make a new design is to not use any components, nor any interpolation. Perfect tool in this respect is the Interpolated Nudge developed by Christian Robertson which got implemented in Glyphs in 2011 and since then every type designer uses it all the time to move nodes around while keeping the tension of the curve. It does exactly what you expect it to do and it is integrated at a very low level perfectly easy to access with a simple shortcut.

Minimise Repetitive Tasks
This is already self-explanatory.

Transparency
For professional use it is very important that the user knows as exactly as possible what a specific tool is doing. Tools which try to be clever without telling the user what it is doing is a pure nightmare for a professional and usually ends up not being used at all. One example which comes to my mind is the optical spacing in InDesign.

Simplicity/Flexibility-Relation ♂ perfect controlpoint amount
Maximum range of control with minimal set of controlpoints/feature/.
Often adding or removing a feature from a tool does not make a big difference, but sometimes adding just one more feature to a tool can result in an exponential higher control. Or on the other side removing one feature could result in a much more simple to use workflow which then actually again increases the productivity of the tool. The simplest example of which I can think of is the idea of numeric systems that use the base 12 instead of base 10, which makes divisions much more easy, whereas there is no difference from base 10 to 8 or 14.

Simplicity and Complexity
In digital environment you can have a very clean interface with the most important options while all the versatility of ones tool can still be accessible. Other then in the analog world there is no need to compromise the versatility of a tool in order to make it easy to use. The next step here is for many applications that the decision-making-code behind the scene is also easy to access and customisable. Inevitable, the more control a tool allows the more it will make it possible to make you achieve to forms you are imagining.

Customisable / Extensible
This might be the most important criteria for a bright future of diverse and high quality tools, as it makes it possible again to form the tool towards the specific needs, as it has been in physical tool making for a long time.

Base 12
1 /1
2 /1 /2
4 /1 /2
6 /1 /2 /3
8 /1 /2 /4
10 /1 /2 /5
12 /1 /2 /3 /4 /6
14 /1 /2 /7
16 /1 /2 /4 /8
18 /1 /2 /3 /6 /9
24 /1 /2 /3 /4 /6 /8 /12

Everyday math would be considerably easier if we would use 12 numerals instead of 10. Where anything in between 13—23 would have no advantage to 12.
The advantage of the duodecimal system is not only mathematical provable but also evident in many unit systems like foot, pound, point, etc.

2. Multipolation

Before challenging the concept of interpolation you have to bare in mind that in the beginning of the year I had developed a workflow to make a high quality font in photoshop and this was basically working like a true digital punchcutter. With this technique I developed my typeface Lawson at a resolution of 1200dpi at 6pt x-height. Drawing with a digital pen was very pleasing but also very time consuming compared to handle bezier-curves. After I translated my design into outlines I felt that I should go even further exploring the opposite extreme. Therefore I went from the most static digital approach – designing a specific rendered instance – all the way to maybe the most flexible approach – designing a model which could output many different instances. In this main chapter I elaborate the challenges of the later.
Experiments and thinking behind this approach already started in 2014, with small-scale experiments. So the first challenge was to rebuild the tool towards something that could be used in a real typedesign workflow.
As things had to develop at the same time this was and still is an ongoing process of adding features and improving the existent.

2.1 Basic concept

Master Relation – The main difference to a traditional interpolation is that you can define how each master relates to one another. This allows you to have an infinite number of design axes. In a simple relation you could start with your Regular then design several alterations of it – eg. a bold, a condensed, one with inktraps, and one for low resolution. With this you could then generate not only a semibold but also a semibold condensed with inktraps. It is basically adding up each decision made within each master.
To make this work you need to specify that all those masters are children of your Regular master. Currently this happens in a simple text file, as I had no time to develop an Interface for it.

Fontfilename_multipolation-spec.json

```
"Master Relations": {
    "Regular": ["Bold","Condensed","Inktraps","Low Resolution"]
},
"Instances": {
    "Semibold": {
        "Bold": "50%"
    },
    "My Semibold Condensed with Inktraps": {
        "Bold": "50%",
        "Condensed": "100%",
        "Inktraps": "100%"
    }
```

Glyph Interpolation – Moreover, I convinced Georg Seifert to implement a function making it possible to interpolate each glyph separately. With this it is for example possible to increase only the weight of the uppercase, etc.

```
    },
    "Regular Test": {
    "#Uppercase": {
            "Bold": "base+5%",
            "_Scope": "include: A-Z"
    }
}
```

This two points are probably the most important differences to traditional interpolation: defining master relations and independent glyph interpolation.

23NnnN23

Master drawn at upper- and lowercase body height.

N23N n23n N23N

Instance, where each kind of scaled character gets his repsective size according to the contextual specifications.
If the Masters are drawn correctly, the interpolation should not need any adjustements, which should be equally right for any design in this respect. Still I need to do further tests to approve this thesis.

2.1 Starting point
First of all I have to state, that my biggest interest lies in innovation and quality on the artistic and conceptional side, not the technical side – as it might seem sometimes. I am an analytical person and the technical part is just one small part of it.

Why the possibilities in Glyphs did not seem to be enough?
I think there are two main forces that drive my inner need for new tools: increasing quality and my aspiration for elegant solution. On top of this comes my good understanding of technology and that I have always liked the search for creative solutions. But why quality?
Especially since I experienced designing type myself it became more and more evident that even my most favourite type designs, works best in specific sizes under specific conditions. I wasn't sure if this comes with the design itself or if it actually is possible to track down to the type design tools, as well as the typesetting tools. Now I am more and more confident to say, that it does not lie in the underlying type design. Where it exactly lies depends from case to case, but it is mostly connected with technical limitations and partially build upon misinterpreted preceding knowledge, e.g. about optical phenomenas.

What was my Problem with interpolation?
Already a two-dimensional Master interpolation gives quite a good idea of its fast possibilities but then it is quite quickly very limited … I want a tool to better analyse principles in type design which does not dictate a specific best practice. Further more I want a tool to make fonts fluid, to interact with typography.

Interpolation-Methods in Compairison

	Multipolation	Glyphs	MutatorMath
Infinite Axis	x		(x)
Default Master Relation	x	x	x
Custom M. Relation Setup	x		
Basic Parent Setup	x		(x)
Separate Axis Setup	x		
Value Mapping	x	x	x
Glyph Interpolation	x	()	
Rules/Bracket Layers	()	x	x
Extra Glyph Masters	()	x	?

2.1 Possible use cases
Adapting one core design to different needs.

Typographic needs
· Adjusting extender- and cap-height to adapt for tight/loose line spacing
· Embed designed fitting for loose and tight tracking/spacing
· Contextual size adjust, e.g. For small caps, superior characters, etc.
· Seamless adjustment of weight to compensate for different sizes / atmosphere
· slight adjustment of italic angle to calibrate differentiation for context
· Optical axes for adjustments apart from resolution and inkspread
· All of it – to match different scripts / type designs

Technical needs
· adjustement for inkspread and different rendering (weight, inktraps, edges)
(· Optical hinting with a resolution axes / adjusting the weight of diagonals)

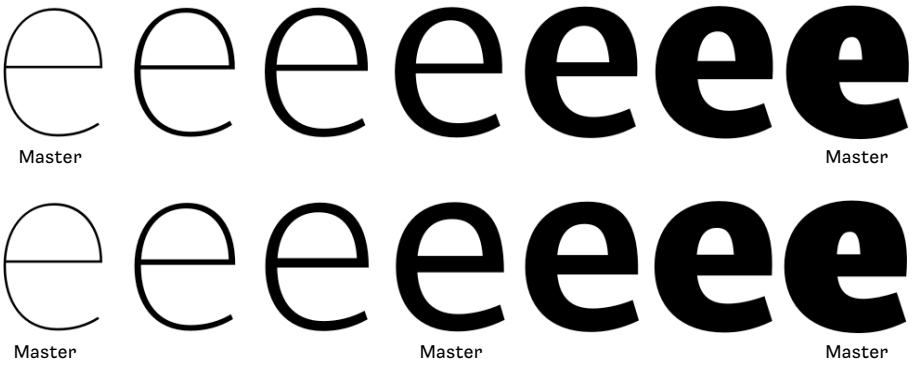
Better starting points within the design process
With the following basic recipes it is most unlikely that you end up with a good result, but it could be still faster than doing it entirely manual
· Regular, Bold, Italic → Bold-Italic
· Regular, Bold → Smallcaps (similar to RMX scaler by Tim Ahrens)
· Regular, Light, Smallcaps → Caps
(· Regular, path offset with compensation →öBold)

Other, within the design process
· Delaying the decision on the exact weight. (quite helpful, as the weight is not a fully stable factor anyway)
· Separating different formal decisions
· seamless testing against one another
· Combining different decisions the inbetween
· Deciding on weight of serifs while you manipulate it across the character set
· Generally improve the interpolation workflow
    · getting more out of the information, you have drawn
    · making it easier to create more complex relationsö

Typesetting Tech
· Improve outer access -öFluid fonts
· Controlled interaction with typographic context

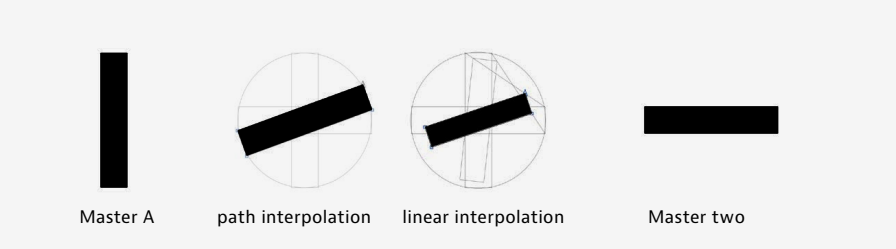
Studies of legibility and atmosphere
· The matching/normalisation aspect plays a huge role in this context. If you could eg input the outer proportionsöof one type design into

2.2 General Constraints of Interpolation
Do we need more interpolation? Or does this not only lead to poor quality and blunt designs? Where each instance is not more than a compromise, only roughly where the design wouldbe if you do it manually.
This is a valid impression of practioneers. And this is true, a lot of interpolations do result in mediocre instances even if someone skilled is handling it. This would then need manual correction. But it entirely depends on what you interpolate. For certain needs you do not need to correct anything and the interpolated instances are of the exact same quality as the Masters themselve. Most of it you can narrow down to type designs which are friendly towards linear interpolations, as eg. low contrast designs.



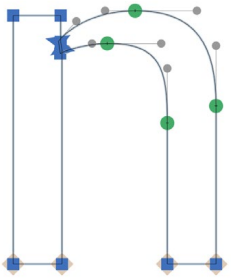
Example by Georg Seifert
https://glyphsapp.com/tutorials/additional-masters-for-individual-glyphs-the-brace-trick

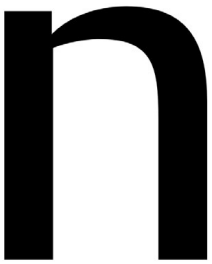
The current best practice in this respect is to add more masters and correct those inbetweens one by one. This minimizes the effect of the linear interpolation but still even inbetween those Masters you still have the same problem, though at a smaller scale. But there could be also many more approaches to handle them. In the future it will be hopefully also possible to describe the path between the interpolations as a curve. This would most likely improve the output of the current multiple master fonts quite a lot.
Another solution would be to e.g. to have another Master which controls the weight seperately from the contrast, which I experimented with. This is very promising and once it works it would make it possible to have very precise control at any position inbetween the Masters.
The third is the possiblilty to sepearate the x- and y-interpolation. This could be also further more improved as you could also rotate it glyph by glyph so that it would also work with glyphs that do not haver a perfect vertcial stress. This approach is on the other side noth very promising as it is very difficult to control and does not allow for different stress angles within a glyph.



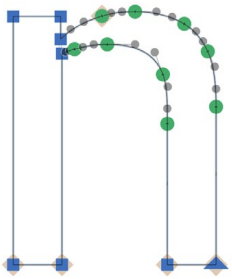
Interpolation of an rotation; Left: interpolation with additional information, where the path of interpolation is given a specific shape; Right: regular linear interpolation

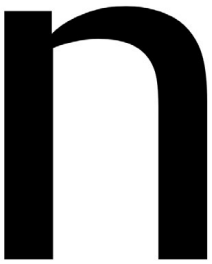
Constraints of Bezier-Curves



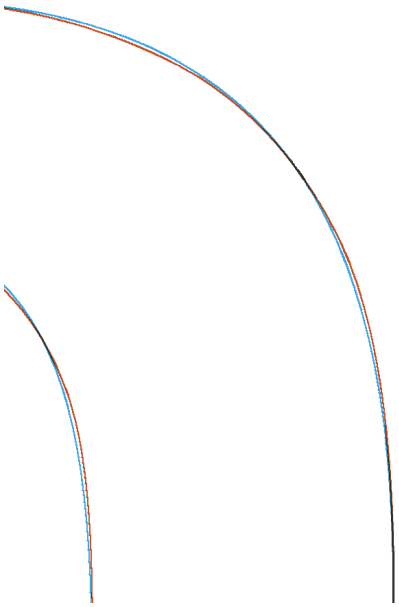


Bezier-friendly design, easy to control and interpolate

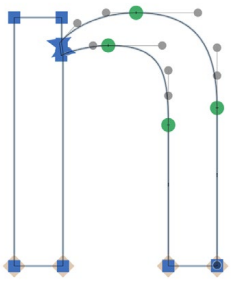




Intended design, difficult to control, impossible to interpolate



Detail compairison



Mockup to visualize a possible extension to normal beziers with additional off-curve points to have a higher control over the curvature while still being able to easily use it for interpolation.

The n-example is extracted from the design of Big Vesta, by Gerhard Unger

A bezier-curve is a very powerful tool, that can describe any curve you want to draw to a very high precision. But there are curves which are easier to describe than others. The ones where you need only very few points are increadibly easy and fast to manipulate. Curves that are hard to to make are e.g. the transition of a straight line into a curve. For certain designs you need to place a lot more points to achieve the intended design. Those points are usually never extreme points and my experience is that it is impossible to make them work for interpolations. So you have either to compromise the general design to get rather nice curves or you decide that the general design is more important and live with the bumps in the curvature. In my designs I decided to go with the later even though it is sometimes quite hard to look at the shapes at large size and suppress the inner need to correct those curves.
But for future tools there is no reason to be restricted to plain bezier-curves, as long there are good convertion tools, that allow to specify the balance between perfect curves and low point-count. With this the drawing format can be indepent from the export format, which can then be either quadratic (otf) or cubic (ttf).

Same Setup, different writing styles
Minimum—Exhaustive (the actual structure behind the scene)

```
"Master Relations": {
    "Regular": "#All",
    "Regular": ["Bold", "Condensed"],
    "Regular": {
        "100%": ["Bold", "Condensed"]
    },
    "Regular": {
        "100%": {
            "Bold": {},
            "Condensed": {}
        }
    },
```

Special abbreviation

Simple

Percentage of relation
which is in this case obsolete, as the the main parent is always to 100% related.

Fully transparent

Minimum level of hierachy. One parent and two children. Each of the four examples is resulting in the exact same setup. Ranging from very concise to most descriptive.

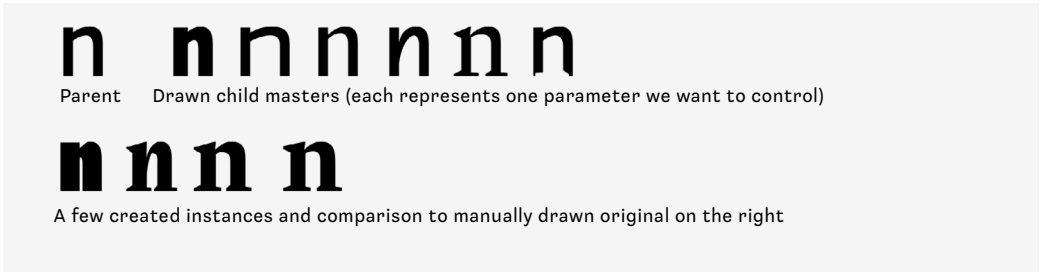
```
    "Regular": {
        "Bold": "Bold_Cond",
        "Inktraps": ""
    },

    "Regular": {
        "Bold": "Bold_Cond",
        "Condensed": "Bold_Cond"
    },

    "Regular": {
        "Bold": {
            "50%": "SemiBold_Cond"
        },
        "Condensed": {
            "100%": "SemiBold_Cond"
        }
```

3 levels of hierachy

With one Master which is a full descendent of two Masters, Bold and Condensed.

With percentage of relation
which is in this case obsolete, as the the main parent is always to 100% related.

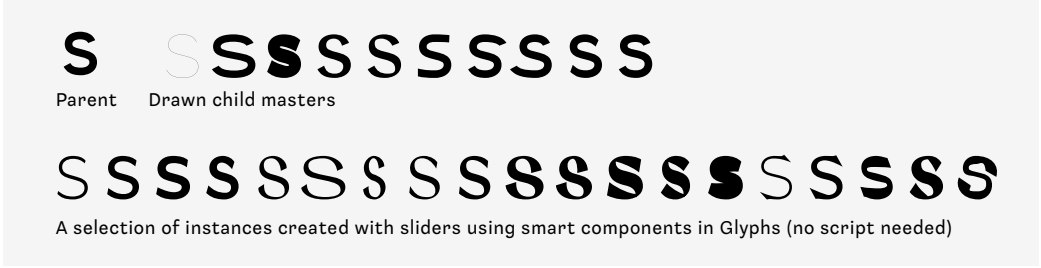
First four are exactly the same, the last tree show different more complex scenarios.



Parent Drawn child masters (each represents one parameter we want to control)

A few created instances and comparison to manually drawn original on the right

This is an experiment dedicated to see if you can control the output in a way to achieve a similar quality to an existing typeface. In this case it is a study on the Malabar Bold in 2012



Parent Drawn child masters

A selection of instances created with sliders using smart components in Glyphs (no script needed)

An experiment without a clear idea of a final design while creating the masters.

2.3 Technique in more Detail

In this section I will illustrate more complex examples and show different aspects of the technique.

Filestructure

First of all, there are two important files – The specification file and the execution file. The first needs to be saved next to your Glyphs file. This is the file where you make all decisions. The execution file is of no interest for the designer, except that you need to run it each time you make a change to your specification file.

Decisions :	Specification File	located next to your .glyphs file
Realisation:	Execution File	located in your script folder

For a fast and pleasing workflow general readability is very important, even before having an actual visual interface. Therefore I decided against an xml-format with its rather repetitive syntax, in favor of a more slim json-format, where the possibilty to color-code the hierachies is very useful as well. Then it was important to implement some fallback default settings, to not cluster the file if you need a very simple setup. Examples of this will follow in context throughout the next paragraphs.

2.2.1 Master Setup

Same Setup, different Readability and therefore different ways of interaction. The following examples of a Master-Setup are working exactly the same and are capable of generating exactly the same instances, yet depending on the context it is important to have this different ways of approaching the same thing.

Relations

By default it will use either the first Master, or preferably one called ðRegularöas the main parent master. This one will have all other masters defined as children, as indicated by ðAllö This makes sure that the setup of your project is not polluted with unnecessary information.

```
"Master Relations": {
    "Regular": {
        "Bold": {
            "Bold High Contrast": {}
        },
        "Level 1":{
            "Level 2": {
                "Level 3": {
                    "...": {}
                }
```

Value Mapping

```
"Master Mapping": {
    "Regular":    [0, 1],
    "Bold":       [0, 1],
    "Condensed":  [0, 1],
    "Inktraps":   [0, 1],
},
```

```
    "Bold":       [85, 140],
    "Condensed":  [410,300],
    "Inktraps":   [12,6]
```

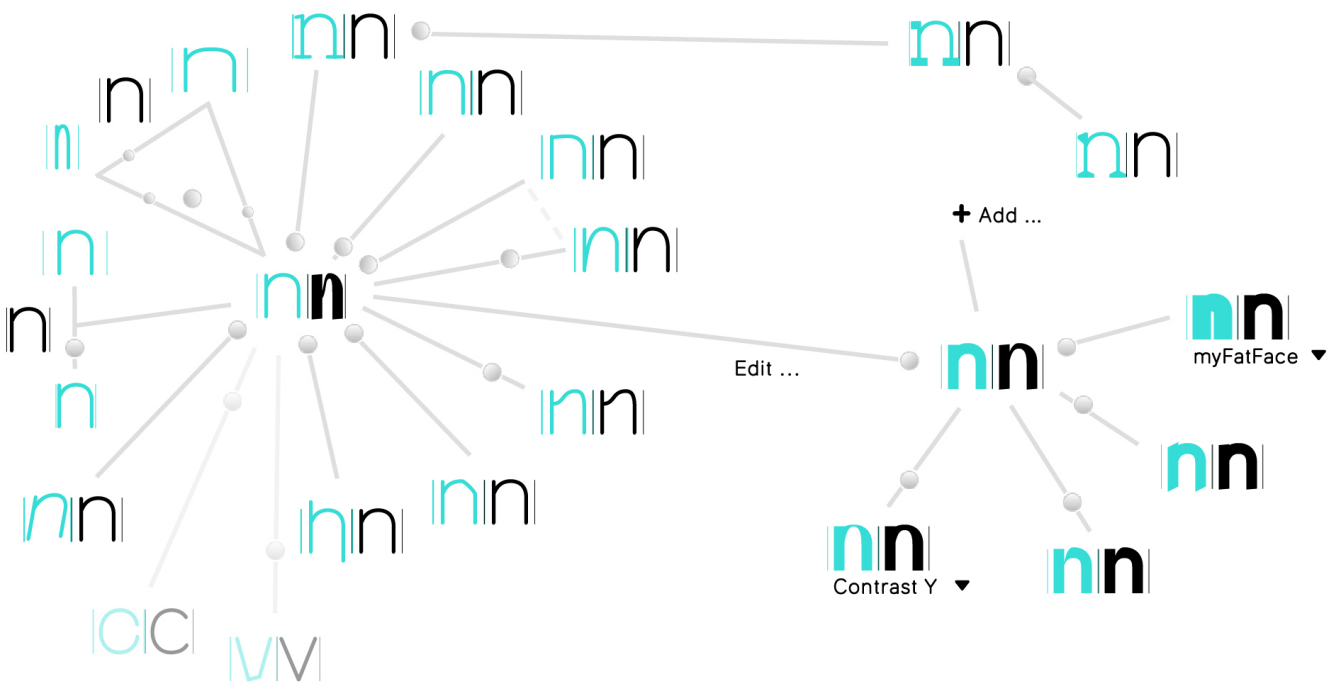
Left: Fallback; Right: using custom design values instead;

The first value represents no influence and the second full influence of the specific master. This is still the same setup and does not change the relation of the masters. It only adds the information that 100% Bold refers to 140units stem width in the Bold Master and 0% Bold refers to a 85 units of the Regular Master – the parent. With this you can use values which relate directly to the measure-

Multiverse Type Design Space (infinite dimensions)
general structure + sketch for a configurable interface

myRootSkeleton

save as …



This shows a general interface proposal with a very complex experimental example setup. The shown characters can be replaced by any text. Concerning the slider, in this case though they do solely represent the fallback axes setting, where each Master has its own slider.

Colored = Masters;
Black = Previews of the actual impact of the Masters on the resulting Instance;
The current resulting instance is at the core of the visualisation.

ments of your design. What values you use here is completely abitary, they can be replaced by any real number. As far my testing goes, the overall parent – in this case “Regular”– does not need any value mapping, as it is represented by the first value (no influence) of all other Masters.

2.2.2 Axes Slider Setup

In opposite to text input, sliders have the huge advantage, that you have direct visual feedback. As a default setup sliders would mirror exactly the master setup to choose the values of each instance. This has the advantage that you are in full control. Which means that you can determine directly which master has an effect on the output. But this can be also a distracting setup, e.g if you have a Light and Bold Master. Then you would have two different sliders for, one for Regular-to-Light and one for Regular-to-Bold. In this case you usually want to merge those to one slider for the weight.

```
"Axes": {
    "Mapping": {
        "A_Size":           [6, 96],
        "A_Weight":         [20, 160],
        "A_Weight_Balance": [0, 1],
        "A_Width":          "Condensed"
    },
    "Logic": {
        "Bold":     "limitMin( A_Weight, 85 )",
        "Light":    "limitMax( A_Weight, 85 ), 85",
        "Inktraps": "limitMax( A_Size, 12 ), 10",
        "Condensed":"A_Width"
    }
```

Custom sliders with the corresponding logic on how they translate to the Master values of any instance. The last axis represents a fallback setting, which corresponds directly to a Master Setup.

This setup is mimicking the standard behaviour of font editors, while additionally introducing a limit for the inktraps, so that the Size slider does not extrapolate above twelve, which would otherwise result in negative inktraps.

2.2.3 Instance Setup

Specifying the Interpolation values used in a specific Instance can be a very abstract thing to do. Those Numbers often do not transfer a lot of meaning, which can slow down decisions significantly. Therefore it was important for me to allow various kind of inputs. Ideally there would be also an option to switch between the different types, which is not yet possible. Additionally I also implemented calculations, which was an immense help if you work with real units, as you then know if you are close to a Master, inter- or extrapolate.

Different types of values

```
"Instances": {
    "Regular": {
        "Bold": "  0%"   100    Min     None/Basis  False
    }           50%     150    Med     Half
},              100%    200    Max     Full        True
                105%    210    Max+5%  Full+10
```

In this setting the Bold Master would have zero impact the resulting Regular Instance. In grey: different options on what kind of values you could use to specifiy the Master values.

```
"Instances": {
    "Regular": {
        "A_Size":           10,
        "A_Weight":         95,
        "A_Weight_Balance":-1,
        "A_Width":          410,
```

Correspoinding Instance setup; First setting the values for each value is the currently chosen value; Second value is the logic with which the slider affect the resulting value;

* This shows only the basic concept. This specific code would only work if the Light and Bold had exactly the same stem difference towards the Regular while using percentage values for Weight and Weight Balance. I had not yet time to figure out the code, which would work with any input.

```
    },
    "Regular Alt": {
        "A_Size":           10,
        "A_Weight":         95,
        "A_Weight_Balance": 0.8,
        "A_Width":          410,

        "Bold":         [95, "%(A_Weight) * A_Weight_Balance"],
        "Light":        [85, "%(A_Weight) * (1 - A_Weight_Balance)"],
        "Inktraps":     [10, "limitMax( A_Size, 12 )"],
        "Condensed":    [410,"A_Width"]
    },
```

Another Instance example, which adds the functionality for the Weight Balance slider to allow shifting between inter- and extrapolation using the Bold and Light Master; temporarly removed the limitMin/Max from Bold and Light to make this example easier to read. Ignore the grey values.

This is the most common scenario and all major font editors have this as a default behind the scene.
But in other cases this might not be exactly what you want, maybe you want more control and you want to see what the difference is between the interpolated Semibold and another Semibold which is extrapolated out of the Light. Therefore I introduced a separate slider setup, to allow for more freedom in how you want to control your Interpolation.ö

2.2.3 Drawing with many Masters

Maybe the biggest pitfall for more complex interpolations is the master-count and that each decision you make in e.g. The Regular needs to be also done in all other Masters, otherwise you might completely destroy the resulting instances. Therefore we can set two goals. First: Maximum flexibility with minimum amount of Masters. Second: Making it easier to work across many masters. The later is especially important in the actual design process.

Correcting several Masters at once

Imagine you want to make following changes to your Regular: *Increasing ascenders slightly and changing the overall distance of Diacritics*. Normally you would need to go through all Masters and repeat most likely the exact same changes, which slows you down immensely and will probably make sure that you never try to handle more than a few Masters.

With the following technique, you only have to do it once. Then specify which Masters you want it applied to. But it does not have to be either yes or no. Additionally you can also decide if you want to apply the change only to 50%, 95% or even 126,4%. So before you make any changes duplicate your Regular Master and name it e.g. RegöCorr. Then you do all the changes within this Master. After that we need just to change our MasterSetup to something like this:

```
"Master Relations": {
    "Regular": ["Regular_Corr", "Bold", "Condensed", ]
},

"Master Update": {
    "Regular_Corr": "100%",
    "#apply":       "true"
},
```

```
"Master Update": {
    "Regular_Corr": {
        "100%":     ["Regular", "Condens
        "95%":      ["Bold"]
    },
    "#apply":       "preview/false/true"
```

Left: Applying Correction to all Masters in 100%; Right: making an exception for the Bold

```
    "Master Relations": {
        "Regular": ["Bold", "Condensed", ]
    },
    "Master Update": {}
```

After execution the correction gets applied, if "#apply" was set to true. It will then not only update the Master outlines but also your specification file and leave you with a cleaned up file.

STATE – This is not yet implemented yet in such a nice way. Currently I keep the Correction Master as long as possible because updating the Masters does not yet work out of the box. At the current stage it would take me at least 30 minutes to do this.

But this has quite a high priority on my feature list, as it is a frequent scenario, which needs to be effortless to handle.

Future Features

· Components across Masters
· Auto marking all glyphs in each Master which are different to their parents
· Interface where you actually see the master Relations (Multivers fig)
· Adding Nodes / Changing the Node Structure. This is another big issue, why it is currently really cumbersome to work with different Masters.
This is also the main reason, why I did not add any optical inktraps.*

3 Analysis

Well it happend again that I have half the analys within the description and I could probably also put the complete catalogue of qualities in here.
As a very short summery for this section I would say that at the current stage I prefer to do most of the design work manually and basically only using interpolation for size adjustments as the small cap example. Even this is not quite comfortable at the moment as you still need to decompose everything before you export. But once I integrated in the tool a few things I mentioned and Georg Seifert fixes some compatibility issues on his side I will definetly use it.

* This is quite a challenging task as the points in the other masters might need to be placed in real-time, to get the coordinates right and to not end up in an overly complicated tool. (After some chats with Rainer Scheichelbauer)

Implications of the Tool
Basically here I would need to extend the thoughts I've mentioned in the possible use cases in section 2.1.
...

4 Conclusion

I'm happy with the process and results of my design work, especially on the Latin part and this was very important to me, as I got trapped in too much critical analysis in my previous graduation project. I also learned to focus my analysis more on what is important at the very stage I'm in and to leave out things I can not manage at the same time.
I did make some improvements on the writing, but ultimately not enough as this is the best example for it. In general I think I put too much presure on my self as I always have the impression that others expect a lot. But it seems that no-one is seeing that I am some one who is working at a slow pace and if I mention it no-one believes it. Yes in some things I'm fast, but only because I put a lot of effort in, to be fast with the tools I use. But there is basically no room to improve on that side anymore and more and more it becomes evident that I am a slow thinker and rather than trying to be quicker I feel the inner need to slow down make the goals smaller. I don't feel comfortable with moving that fast with that many things to handle, but it is so much harder to tell yourself to keep it simple; to go for the safe way once in a while and to make the best out of it within its limitations.

5 Overview of other Challenges

Research on the atmosphere of typefaces

It started out very exiting with the first introduction, the workshops and the development of the brief. I remember very well the first introductions by Gerry. I felt like I've arrived at the right place, not only that I now study actually type design with a lot of like-minded people from all over the world, but also where critical thinking and practice seems actually a core ingredient of the philosophy and not just a phrase, to advertise.

Then there was the first bigger disappointment: There is no reliable research on the atmosphere of typefaces and further discussions, especially with Mary Dyson made it clear that the scale of this topic is too big to challenge within a MA.

Bilingual Reading Process
The next important step for me was to get a better understanding of the reading process and for example realising that there is no such thing as a bilingual reading process outside of e.g. linguistic research and that the fast majority of documents is mostly read in the language the reader is most comfortable with. Therefore it might be not of the biggest importance to harmonise scripts equally but to find a good solution of embedding the bits of other scripts that do appear regularly within a specific script and language.

Amplifying the students capabilities
There is a fast number of things I love about how type design gets teached in Reading. To name a few: the big emphasis on the context; to question everything and everyone without any restrictions - probably the most important; being fully aware of where typedesign stands in the society, where its sometimes almost invisible use lies but also where the impact of an type designer ends; what you can learn form history; the discussions about the market, clients, culture, technology and how they all can affect decisions in type design; etc.

But as everything in live, nothing is perfect and it is a continuously journey to make things better, and therefore I need to list a few more personal disappointments. It was very demotivating, that even on specific request it was not possible to get feedback on the brief nor discussing it, even though it was partially set out for this, and it was made very clear that the brief is the context within which our type design is judged. This is clearly a decision made by the staff beforehand, which is questionable for me. Also in general the feedbacks I had were rather demotivating than amplifying my motivation. I understand that it might feel stupid and sometimes even superficial but from an educational perspective it is very important to always point out in every consultation some specific things which are good and to make this very clear to the student.
Initially it was told us that everyone will make his own personal journey and that with whatever skill set you come into, that the aim of the course is to amplify what is there and not to make everyone meet a specific level. Sure, at the end of the year everyone came out naturally at a different stage. Even though the course and the staff does not aim on an equal skill set it does not amplify the specific skill set of each student either. I often felt not taken for full in the beginning – getting more of a standard response than an discussion I really needed. Now this feels even more frustrating as most of my thinking got rather confirmed than disproved. At this level I would also prefer that the staff is always clearly revealing their intentions behind their acting, their arguments, etc. Sure for educational purposes it is sometimes important to just do certain things without fully understanding or questioning the intentions and the use of it, but you need to always reveal it at some point. I'm sure that both the students and the staff could profit from this immensely.
Don't take me wrong, I hope you don't see this as a negative response but rather as an opportunity, I definitely do. I learned a lot of things which I will try to make part of my own teaching and a lot of things I will try to not trap into while working with students.

Design
I focused in the design process mostly on the atmosphere of type and how to find original coherent solutions that differentiate enough at small size while keeping the necessary level of comfort in reading. Moreover I studied different optical and technological phenomenas, some of which I see much clearer now, others of which are to certain extent still a mystery to me. Interestingly it doesn't seem that even the best type designers have figured out all of them, as many clear compromises are evident in their typefaces. On one side it is an encouragement as there are still many things to discover but on the other side

it is also an encouragement to not questioning everything and to feel some comfort in following your intuition. So as it is almost impossible for me to not ask questions and figuring out the underlying principles while I'm designing, the current biggest lesson I learned is that I should hear much more on my intuition and enjoy to leave questions open – my nature of mind will take me back soon enough anyway.

Arabic
I learned a lot about the variety of letter-patterns an Arabic Naskh can be interpreted in. I analysed the visual grammar of different manuscripts, and started to transform my findings into an attempt of a sans-contrast book-face which embraces the idea of a visual grammar. Additionally I also tried to blend out the technological limitations within Glyphs and made my own very time consuming work arounds. Unfortunately I did not find enough time to engage with Decotype's software, which seems to have some clear advantages, for designing a Naskh typeface.

Workflow
As I'm happily looking forward to stay in the profession of type design for a long time it was important for me to try out many different workflows and to search for efficiency and speed on one side, but also encourage workflows which encourage original work

```
},
"MasterSetupMapping": {
    "SansRegCorr":      [0,1],
    "SansReg":          [0,1, "All"],
    "Contrast":         [0,1],
    "Size_x":           [1370,1000, ["Size_x_833", "Size_x_1672"]],
    "Size_x_833":       [1000,833],
    "Size_x_1672":      [1000,1672],
    "Light":            [180,150, ["Size_x_833"]],
    "Bold":             [180,300],
    "Italic":           [0,1],
    "Condensed_Trans":  [0,1],
    "Bold_Offset":      [180,300, ["Bold", "Size_x_1672"]],
    "_info": {
        "description": "MasterName, Map_0, Map_1, Child(s), Instance Matl
        "default":     [0,1]
        }
},
"InstancesSetup": {
    "Regular": {
        "#Smallcaps": {
            "Size_x":           "1140",
            "Light":            "none-10",
            "_Scope": "include: a.smcp, b.smcp, c.smcp, d.smcp, e.smcp, f
                o.smcp, p.smcp, q.smcp, r.smcp, s.smcp, t.smcp, u.smcp, v
                three.smcp, four.smcp, five.smcp, six.smcp, seven.smcp, e
        },
        "#Numerals": {
            "Size_x":           "1330",
            "_Scope": "include: one, two, three, four, five, six, seven,
        },
        "#Uppercase": {
            "Size_x":           "1370",
            "_Scope": "include: A, Adieresis, B, C, D, E, F, G, H, I, J,
                X, Y, Z, I.ss02, zero.uc, one.uc, two.uc, three.uc, four.
        },
        "#Ascenders": {
            "Size_x":           "1470",
            "_Scope": "include: l, h, b, d, k, f, germandbls",
            "__Size_x_833":     [1000, "full"]
        },
        "#All": {
            "SansReg":          "full",
            "SansRegCorr":      "full",
            "Contrast":         "none",
            "Condensed_Trans":  "none",
            "Bold_Offset":      "none",
            "Bold":             "none",
            "Light":            "none",
            "Italic":           "none",
            "Size_x":           "none",

            "__Size_x_833":     "none",
            "__Size_x_1672":    "none",
            "__LightSMCP":      1040,
            "__LightSMCP_Weight":135,
            "__Numerals_1000":  "none-70",
            "__NumeralsSMCP1370": 1200,
            "__UC_Wide":        "none"
        },
        "_#Lowercase": {
```

```
},
"MasterSetupMapping": {
    "Ruth":                [0,1, "All"],
    "RuthSlanted":         [0,1, ["Italic"]],
    "BolderInner20":       [185,205],
    "BoldStem":            [185,305,["Bold"]],
    "Bold":                [185,305],
    "Italic_Prev":         [0,1],
    "Italic":              [0,1],
    "_info": {
        "description": "MasterName, Map_0, Map_1, Ch
        "default":     [0,1]
    }
},
"InstancesSetup": {
    "Regular": {
        "Ruth":           1,
        "BolderInner20":  "none+10",
        "BoldStem":       "none",
        "RuthSlanted":    "none",
        "Bold":           "none",
        "Italic_Prev":    "none",
        "Italic":         "none"
    },
    "RegularAlt": {
        "Ruth":           1,
        "BolderInner20":  "none+50",
        "BoldStem":       "none",
        "RuthSlanted":    "none",
        "Bold":           "none",
        "Italic_Prev":    "none",
        "Italic":         "none"
    },
    "Semibold":{
        "Ruth":           1,
        "BolderInner20":  "none",
        "BoldStem":       "half",
        "RuthSlanted":    "none",
        "Bold":           "none+40",
        "Italic_Prev":    "none",
        "Italic":         "none"
    },
    "Light":  {
        "Ruth":           1,
        "BolderInner20":  "none-40",
        "BoldStem":       "none",
        "RuthSlanted":    "none",
        "Bold":           "none",
        "Italic_Prev":    "none",
        "Italic":         "none"
    },
    "LightAlt":{
        "Ruth":           1,
        "BolderInner20":  "none",
        "BoldStem":       "none-40",
        "RuthSlanted":    "none",
        "Bold":           "none-40",
        "Italic_Prev":    "none",
        "Italic":         "none"
    },
    "Bold":{
```

```
},
"MasterSetupMapping": {
    "Lawson":              [0,1, "All"],
    "LawsonCorr":          [0,1],
    "StemThicker20":       [0,20],
    "Prop_Ext":            [1480,1580],
    "Prop_Mono":           [0,1,["Prop_Mono_Narrow"]],
    "Prop_Mono_Narrow":    [0,1],
    "Low_Contrast":        [0,1, ["Low_Contrast_Serifs
    "Bold":                [0,1],
    "Low_Contrast_Serifs":[0,1],
    "Serifs_Short":        [0,1],
    "Text_Corr":           [0,1],
    "Tracking100":         [0,100],
    "_info": {
        "description": "MasterName, Map_0, Map_1, Child
        "default":     [0,1]
    }
},
"InstancesSetup": {
    "Regular Display":{
        "Lawson":          1,
        "LawsonCorr":      1,
        "StemThicker20":   "none",
        "Prop_Ext":        1480,
        "Prop_Mono":       0,
        "Prop_Mono_Narrow": 0,
        "Bold":            0,
        "Low_Contrast":    0.0,
        "Low_Contrast_Serifs": -0.5,
        "Serifs_Short":    0.1,
        "Text_Corr":       0,
        "Tracking100":     0
    },"Regular Subhead":   {
        "Lawson":          1,
        "LawsonCorr":      1,
        "StemThicker20":   "none",
        "Prop_Ext":        1480,
        "Prop_Mono":       0.15,
        "Prop_Mono_Narrow": 0.05,
        "Bold":            0,
        "Low_Contrast":    0,
        "Low_Contrast_Serifs": -0.5,
        "Serifs_Short":    0,
        "Text_Corr":       0.5,
        "Tracking100":     0
    },"Regular":   {
        "Lawson":          1,
        "LawsonCorr":      1,
        "StemThicker20":   "none",
        "Prop_Ext":        1480,
        "Prop_Mono":       0.3,
        "Prop_Mono_Narrow": 0.1,
        "Bold":            0,
        "Low_Contrast":    0.0,
        "Low_Contrast_Serifs": -0.4,
        "Serifs_Short":    0,
        "Text_Corr":       1,
        "Tracking100":     10
    },"_LawsonTextItalicPrep": {
        "Lawson":          1,
```