

VYSOKÁ ŠKOLA UMĚLECKOPRŮMYSLOVÁ V PRAZE

Bakalářská práce

VYSOKÁ ŠKOLA UMĚLECKOPRŮMYSLOVÁ V PRAZE

Tvorba písma a typografie
Bakalářská práce
Jan Šindler

Písmo pro digitální zobrazování

Praha, 2019
Vedoucí bakalářské práce: Tomáš Brousil
Konzultant bakalářské práce: Lukáš Pilka

ACADEMY OF ARTS, ARCHITECTURE AND DESIGN IN PRAGUE

Typography and typeface designing
Bachelor thesis
Jan Šindler

Typeface for digital displaying

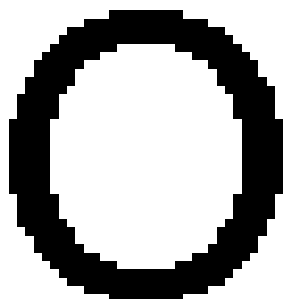
Prague, 2019
Bachelor Thesis Supervisor: Tomáš Brousil
Bachelor Thesis Consultant: Lukáš Pilka

1 úvod

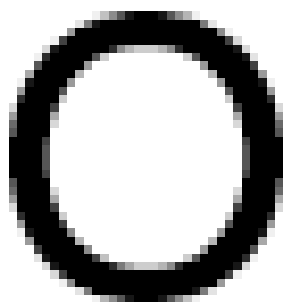
Téma jsem si vybral hlavně kvůli vlivu stáže ve firmě LucasFonts v létě 2018. Při stáži jsem se dozvěděl spoustu nového a zjistil jsem, že naše praxe není pouze o tvarování křivek. LucasFonts se věnují navrhování a produkci těch nejkvalitnějších fontů, které vydrží spoustu let. Rád jsem byl součástí týmu a doufám, že zase někdy budu. Poprvé když jsem od Lucase slyšel magická slova jako VTT a hinting, tak jsem mohl pouze předstírat, že vím o čem je řeč. První pokusy s hintingem proběhly už na stáži, šlo především o pokus-omyl metodu aplikování hintingu, moc jsem si z toho neodnesl. Po stáži jsem se začal věnovat oborům jako je programování a začal jsem se zajímat o strukturu fontového souboru. Hinting pořád neodmyslitelně patří k písmové produkci a je škoda, že o tomto řemeslu už moc lidí neví. Je důležité znát jeho sílu abychom mohli lépe navrhovat a uspokojit nejen svojí neukojitelnou autorskou touhu ale i koncového spotřebitele. O hintingu chci mít hlubší ponětí abych mohl dělat lepší písma, znát okolnosti a získat výhodu na trhu práce. Během studia jsem žil v naivní představě o nadvládě tvarosloví nad vším ostatními odvětvími navrhování písma, už to tak není a svou bakalářskou práci posouvám tvarosloví dané bezierovými křivkami ještě dále. Budu zpracovávat písmo, které bude vycházet z dat, které představuji dále v mé bakalářské práci. Zajímá mě hlavně vztah bitmapové rasterizované kontury a kontury samotné. Kdy se stane to, že naše milované písmo ztratí všechno co jsme mu dali a je z něho pár pixelů? Mohu navrhnout písmo, které bude vycházet z omezení mřížky, do které se navrhuje a nevypadalo jako z 90. let? Chci udělat písmo pro 1bitové displeje, protože právě pro tyto systémy vidím hinting jako obrovský potenciál. Chci navrhnout alternativu k písmu Verdana a Tahoma, která bude nabízet trošku jiný pohled na věc a bude stejně dobře zpracovaná. Samotná teoretická práce se zabývá analýzou situace ve světě digitálních písem, a přehledem nad tzv. big daty, které jsem sesbíral a analyzoval. Výsledky analýz jsou fakta, na kterých zakládám svůj projekt.

2 Co je to vlastně hinting?

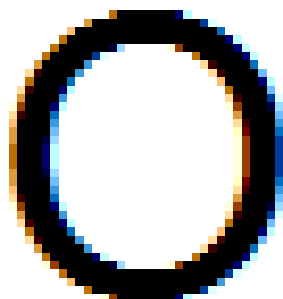
Hinting, nebo optimalizace pro obrazovky, je proces při kterém jsou TrueType or PostScript fonty upraveny k maximální čitelnosti na displejích (Petr Biřák, typotheque.com/articles/hinting)
Hint je v překladu nápořveda, hinting je tedy jakési dáváni rad rasterizéru, který je hloupý a bez rad by písmo dobře nezobrazil. Jsou 4 druhy rastrování křivky (Beat Stamm, rastertragedy.com/RTRCh2.htm)



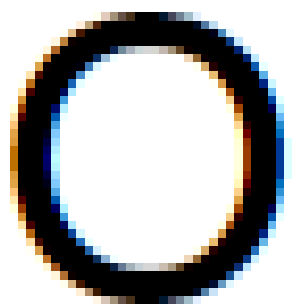
Obr. 1: celopixelové - pixely jsou buď zaplé a nebo vyplé, 1bitový princip.



Obr. 2: celopixelové s celopixelovým vyhlazováním - Celé pixely doplňují pixely ve stupních šedi, které opticky zaplňují pixelové zuby v distribuci pixelů celopixelového rastrování.



Obr. 3: asymetrické s necelopixelovým vyhlazováním - Pixel je horizontálně složen ze 3 podpixelů, RGB - červené, zelené a modré. Tyto podpixely svítí různou intenzitou dle komplexních algoritmu založených na optice a fyzice. S vzdálenostním odstupem spolu tvoří celek, kde už vnímáme jen černou a bílou.



Obr. 4: Hybridní – V ose X se chová jako předešlý způsob. V ose Y se chová podobně jako druhý způsob rasterizace akorát taky pracuje se samotnými barvami pixelů



Obr. 5: Náhled toho jak ve skutečnosti vypadá hybridní rasterizace na RGB LCD obrazovce. RGB (red, green, blue) je pořadí barev v jednom pixelu, existují i BGR monitory.

2.1 Historie hintingu

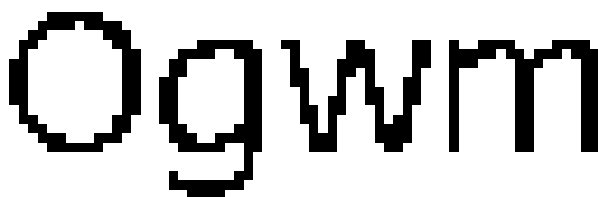
Hinting je samotným programovacím jazykem, před příchodem vizuálního programování hintingu se křivky musely programovat tak jak programování si asi běžně představujeme. S příchodem vizuálního hintování ovšem tento způsob textového programování úplně nevymizel a někteří inženýři jej stále používají. Vizuální způsob ale ulehčil ono řemeslo hlavně nám designérům, kteří nejsme zaměřeni jenom na hintování a věnujeme se i ostatním disciplínám typografie.

2.2 Hinting dnes

Hinting dnes je zřejmě umírajícím řemeslem, designéři o tuto kvalitu fontu nejeví zájem zřejmě protože pracují na zařízení s vysokým rozlišením a samotné nedostatky ani nejsou schopni vidět na svém zařízení. Špatný hinting se ale projeví až u koncového uživatele. Nutno přiznat, že hinting není tak potřeba jak dříve při užití na stolních počítačích, větší část zařízení zobrazí písmo kvalitně i bez něj. Přesto i já s poměrně moderním počítačem si stále všímám špatně vyhintovaných fontů na webu, ty často velmi znepříjemňují čtení na digitálních zařízeních, mluvím to o webech jako je BBC a nebo YouTube, což rozhodně nejsou periferní záležitosti. Jsou profese, u kterých lidé tráví několik hodin denně u počítače, na čitelnosti jim záleží. Příkladem může být například opensourcové písmo Fira, které je velmi rozšířené mezi programátory. Uživatelé si nesprávného hintingu všímají a i když často neví, jak tento problém pojmenovat, tak ho nahlásí tvůrci písma aby jej opravil (github.com/tonsky/FiraCode/issues?utf8=

2.3 Druhy Hintingu

1. **TrueTypový hinting** je velmi silný nástroj, který umožňuje modifikace písmen u každé velikosti a to do těch nejmenších detailů, jako posunutí bodu pro určitou písmovou velikost klidně i o 1/16 pixelu. Nástroj je to silný a jeho ovládání je velmi komplexní. TT hinting jak ho známe dnes si můžeme představit jako vizuální programování, kde nastavujeme pravidla. Je nutno si uvědomit, že je to programovací jazyk a každý znak má program v takovémto jazyku k sobě přiřazen, který komunikuje s rasterizérem a radí mu jak se vykreslit, pak je už na samotném rasterizéru jak s takovýmito radami naloží.
2. **PostScriptový hinting** Postscriptový hinting je modernější forma hintingu, je jednodušší než TrueTypový ale neumožňuje takové modifikace.
3. **Autohinting** Je automatické vytvoření hintingového programu pro znaky fontu. TrueTypový může fungovat buď na základě postscriptového hintingu a dobrého nastavení měř ve fontu samotném (bude fungovat přesněji než kdyby se algoritmus pokoušel o automatizaci hintingu bez nich). TT autohinting může narazit v případech kdy jde o nestandardní písmo, například když konce tahů nekončí přímo na dotažnicích. Autohinting taky nemůžeme aplikovat na písma, kde je důležité vědomí o souvislostech v písmu. To jsem už sám pocítil při práci na variabilním písmu, kde jsem mohl hintovat pouze v ose Y. To by ještě nebyl takový problém, všechny znaky byly ve skupinách zástupných znaků a ty musely být vyhintované úplně stejně jako děti této skupiny.



Obr. 6: ukázka textu bez aplikace žádného hintingu při 1bitovém rasterizování. Maličký rozdíl v bodech vyústí k nechtěnému zaokrouhlování. Písmena mají typicky jináčtí tloušťky tahů



Obr. 7: Autohinting si počínal o trochu lépe. Je vidět že algoritmus nemá pojem o symetrii a souvislostech v typografii. Horizontální tah u 'O' je tlustší protože tah je o pár bodů tlustší, to je potřeba optické kompenzace tištěné typografie s kterou nemůžeme nic udělat. Konkrétně je to 42 bodů u verzálek a 40 u minusek, vzdálenost 42 bodů se při této velikosti už zaokrouhluje na 2 pixely nicméně vzdálenost 40 bodů se ještě zaokrouhluje na 1 pixel.

3 programy okolo hintingu

Hinting se může dělat přímo v písmovém editoru a nebo v externím editoru. Implementovaný editor má nevýhodu, že není tak specializovaný a silný jako ten externí. Na druhou stranu, je produkce takového písma s implementovaným hintingovým editorem jednodušší, protože jsou všechna data na jednom místě a nemůže dojít k problémům například při změně pořadí bodů v kontuře.

3.1 VTT

VTT je nejsilnější a nejprofesionálnější editor na hinting, který existuje. Vyvíjí ho firma Microsoft a produkuje v něm také písma pro systém Windows.

- (a) Link - základní nástroj pro, kterým určujete jaký vztah jeden kotevní bod naplňuje k druhému. Může jít o jednoduchá pravidla - zaokrouhlí-li se jeden doleva, zaokrouhlí se i druhý doleva a nebo že tyto dva 'a' a 'b' body budou od sebe vzdáleny stejný počet pixelů jako druhé dva body 'x' a 'y'.
- (b) Interpolate - interpoluje jeden bod na základě dalších dvou – něco jako když na pružinu umístíme bod a pak s ní natahujeme, bod bude vždy ve stejném vzdálenostním poměru ke krajům pružiny.
- (c) Delta – posouvání buď globálních a nebo lokálních hodnot - nastavíme-li že všechny body na x-výšce jsou označeny proměnnou, nástrojem delta můžeme určit v jakých velikostech se případně výška o bod zvedne, nejsme-li spokojeni s plynulostí gradace velikostí. Lokální deltou pak ovládáme body stejným způsobem a to od nejjemnější nuance 1/16 bodu až po 8 bodů do plusu i záporu, pokud by to mělo nějaký důvod, můžeme nastavit, že písmeno 'o' je širší nebo užší až o 8 bodů ve zvolené velikosti.

X-height je v češtině střední dotažnice, je výška hlavní hmoty minuskových znaků, vede právě tam kde končí výška malého písmene 'x'

3.2 rasterizéry

- (a) ClearType - rasterizer Windowsu, funguje na bázi rozdělení pixelu na sub-pixel. $RGB = 1 \text{ pixel}$, $R = 0.33 \text{ pixelu}$
- (b) FreeType - opensourcový projekt, který najdeme ve velkém množství zařízení. Od Androidu/Linuxu až po telefony společnosti Apple. Jde o opensourcovou variantu ClearType
- (c) Quartz - rasterizer MacOS, ten ignoruje veškeré hintingové instrukce a rastr vytvoří podle vlastního algoritmu, tak aby text vypadal na obrazovce co nejvíce jako jeho tištěná podoba. To má za důsledek, že písma jsou velmi špatně čitelná v malých velikostech.
- (d) CoolType - je metoda rastrování textu, kterou využívá Adobe a jejich řada programů

4 analýzy

Proto abych lépe pochopil data okolo fontů a digitální stopu, kterou ve světě zanechávají, tak jsem použil služby od Googlu se jménem BigQuery. Ta umožňuje prohledávat ohromné množství dat za pár vteřin. BigQuery¹ navíc nabízí svoje vlastní datasety a to zdarma. Já využil hlavně datasetu od githubu, zajímalo mě hlavně v jakém poměru se používají patkové a nepatkové písma na webu a také jaká velikost písma je nejpoužívanější.

Dataset je soubor dat, který je v případě BigQuery zadarmo. Může jít jak

4.1 font-size

Obrazovky dnešních počítačů ve většině světa se zobrazováním písma nemají skoro žádný problém, nesmíme ovšem opomenout oblasti a nebo instituce, kde je velmi náročné přejít na novější systém, buď kvůli kvůli vlastním interním nástrojům, které je velmi náročné aktualizovat a nebo kvůli nedostatku financí. Velké firmy by měly mít písmo, které obstojí v jakékoliv situaci a to i na tom nejslabším zařízení nebo v aplikaci, kde se třeba dlaždičky a nebo dlažební kostky jsou samotnými pixely. Typickým příkladem může být displej pračky, displej bojleru, digitální cenovka v supermarketu, úvodní obrazovka monitoru při zapnutí a nebo při nastavování menu na obrazovce a taky každý další jednoduchý počítačový systém, který nepodporuje nejmodernější zobrazovací technologie. Nesmíme zapomenout i na stále poměrně silnou uživatelskou základnu, která pořád pracuje na Windows XP dle dat z roku 2017 by těchto uživatelů mělo být více než uživatelů Linuxu a nebo MacOS Sierra(businessinsider.com/how-many-people-use-windows-xp-chart-2017-5). Windows Xp umožňuje dodatečnou instalaci rastrového engine ClearType, tím ale velmi pravděpodobně dipsonuje pouze malá část těchto počítačů. Všechna tato zařízení vyžadují buď aplikace firemních písem, log a nebo piktogramů. Zde všude přijde hinting vhod. Písmo jsem navrhoval primárně pro nějakou velikost, která je lidmi nejpoužívanější, svůj požadavek jsem zadal pouze na nastavení webových stránek, hlavně protože je ustálené a programátoři velikost vybírají s vědomím, že neví na jakém zařízení se jejich web zobrazí a předpokládám že zvyky z navrhování pro jednu platformu se mohou projevit i při navrhování na druhou platformu. Jelikož jde o akademický úkol, tak ani já nevím na jakém zařízení bude moje písmo zobrazováno. O to zajímavější to je, neboť mi jde o to aby vypadalo, četlo se a fungovalo dobře na všech zařízeních a mnoha okolnostech. Pro moji analýzu jsem využil služby BigQuery od Googlu, která dovoluje zkoumat velká množství dat pomocí jejich vlastních počítačů. Google nám v této službě už připravil nějaká data, takovému balíčku dat se říká dataset. Takový dataset si můžeme představit jako hrozně velkou excelovou tabulku, která je tak velká, že bychom ji na našem počítači neotevřeli, pravděpodobně by se zasekl a nedošel by ani do půlky. Jeden z datasetů jsou data GitHubu má 2,3TB , já jsem svůj pokus dělal pouze na neplacené části 30GB a i tak jsem dostal uspokojující množství dat, kterou jsou svojí kvantitou dostačující. Z analýzy vyplynulo, že programátoři využívají z velké části jednotky PX oproti PT. Plný dataset má tedy 2,3TB, to je 80krát více než dataset, který jsem použil já. Nechci přepočítávat terabyty na hodiny filmu, ale řekl bych že 2,3TB je kapacita paměti 5-7 notebooků v roce 2019. PT jsou určené spíše pro tištěné výstupy, programátoři jsou si tedy vědomi specifiky digitálního písma. Z dat vyplynulo, že nejpoužívanější velikostí pro weby je 14PX a 10PT což se rovná 13.3PX. Základní velikostí pro navrhování písma určeného k digitálnímu zobrazování je tedy velikost 14PX. To je velikost, kde se právě začínají lámat rozdíly mezi hintingovými unibody písmy a písma se už začínají lišit. Mřížka B/W monitoru je omezena a proto mohou působit nějaká písma dost podobně i když jde původně buď o egyptienkové písmo a knižní serif. V případě malých velikostí ovšem nemůže jít o to se odlišit ale zaručit, že písmo bude fungovat i v malých velikostech. Nemluvě o přizpůsobení piktogramů a log k takovémuto zobrazení.

GitHub je platforma pro vývojáře, kde si lidé mohou ukládat a sdílet právě vyvíjenou aplikaci buď mezi sebou a nebo veřejnosti.

TB jako Terabyte, GB jako Gigabyte jsou jednotky pro popis velikosti digitálních dat

¹<https://bigquery.cloud.google.com/queries/fontsize>

jednotka	medián			počet nálezů
		m. průměr		
PX	14.0	26.8330	13352	
PT	10.0(13.3330PX)	12.3086	1231	
EM	1.2	3.0410	7225	
%	100.0	114.9932	6295	

4.2 font-family

Mým dalším cílem zkoumání pomocí služby BigQuery bylo i jaké písma uživatelé používají. Výsledky ukazují, že převažují písma bezserifová a většinou systémová. Z toho vyplývá, že programátoři jsou si vědomi výhod takových písem. Systémová se nemusí zobrazovat a bezserifová jsou lépe čitelná na obrazovkách nižších rozlišení. Těmito dvaceti nepoužívanějšími fonty je pouze jedno serifové, velmi překvapivé je, že Times New Roman se do téhle dvacítky vůbec nedostal. Zda absence patkových písem je následkem slabé nabídky a nebo zkrátka nejsou určena pro digitální formu zobrazování je na diskusi. Fonty se pro weby nastavují buď konkrétně a nebo jako skupina, tedy název fontu – Arial – nebo druh fontu - Sans Serif. Font se nemusí nastavit jenom jeden v případě, že by font na cílové mašině nebyl, nahradil by se nekontrolovatelně na základní font platformy. To nechceme a proto můžeme v pořadí našich preferencí nastavit fonty dle libosti, pokud první v systému není, aplikuje se ten další atd. příklad font-family="Arial, Helvetica, Verdana".

název písma/skupiny	počet nálezů	
		distribuce
sans-serif	25895	Skupina
Arial	20070	WIN
Helvetica	11886	MAC
Helvetica neue	6560	MAC
Verdana	5618	WIN
monospace	4881	Skupina
Courier	4315	WIN
Tahoma	2953	WIN
inherit	2943	Skupina
Lucida console	2239	WIN
Font Awesome	2163	ID
serif	2146	Skupina
Lucida grande	1733	OS X
Monaco	1413	OS X
Lato	1191	G
Open sans	1135	G
Courier new	1119	WIN
Consolas	1099	WIN
Georgia	1038	WIN&MAC
Segoe ui	819	WIN

5 Pixel a pixel mimo digitální svět

rastr vektoru - pixelizaci nesmíme striktně chápat jako něco co se vyskytuje výhradně na obrazovkách, jde o rozložení křivek na pravidelnou mřížku. O pravidelnou a ovladatelnou distribuci bodů rastrové křivky nám jde především když je hustota mřížky omezena. Uvádím zde tři příklady, kdy se taková distribuce hodí i pro aplikace jiné, než je obrazovka. Všechny tyto aplikace jsou velmi pravděpodobné a pokud firma písmo s hintingem má jednou, tak už problém rasterizace kdekoliv nikdy nebude muset řešit.

- (a) **Jehličkové tiskárny** s povinností tisknout všechny účtenky, roste i spotřeba thermopapíru, ten je ovšem nerecyklovatelný a na tento fakt často ochránci životního prostředí poukazují. Obchody, které jsou s touto problematikou obeznámeny používají staré jehličkové tiskárny, protože jde o tisk barvou na recyklovaný a dále recyklovatelný papír. Jde o jakýsi trend, který se šíří napříč bioobchody a kavárnami, řeší problém s nerecyklovatelnými thermopapíry a ukazuje nám, že žádná technologie není mrtvá a je velká šance, že se brzy či později opět vrátí.
- (b) **Práce se samotným pixelem** Digitální rastrový obraz není jenom sled tvarů a bodů ale přesná mřížka s přesnými údaji, to v dnešní době můžeme využít a můžeme pracovat se samotnými pixely. Není vůbec těžké si jako designér vytvořit nástroj, který přemění pixel na jiný tvar. Takový nástroj nám pak bude tvořit například vizuál festivalu pouhým aplikováním nástroje na fotografii. To nám skvěle předvedl Just van Rossum a Hansje van Halem při vizuálech na vizuálních stylech festivalů Lowlands a nebo studio Norm při navrhování nové podoby švýcarských bankovek. Pokud mám pole typu šachovnice o velikosti 2x2 pixelů, data obrázku by vypadala asi takto [[255, 0], [0, 255]]. S daty pak můžeme libovolně pracovat. Můžeme například všechna políčka, kde je bílá (255) spojit čarou a nebo je nahradit nějakým symbolem. Můžeme je nahradit třeba obličejí a nebo srdíčky, síť by pak byla asi v mnohem větší velikosti, než bychom očekávali od obrázku s rozlišením 2x2 pixelů. Z tohoto obrázku se nyní stává síť, kde každý bod může být pozorován samostatně. Dohromady však tvoří celek, kde rovnoměrná distribuce je důležitá.
- (c) **Průmyslová produkce návrhů** jsou technologie, kdy při realizaci návrhu je míra detailu velmi omezena. Typicky může jít o pletení, tkaní, vyšívání a nebo perforace do nejružnějších materiálů a nebo třeba i vyskládání nápisu do dlažby nebo do kachlíček.

5.1 O u Timesu je stejné jako u Arialu

Unikátní tvar písmene, který jsme zvyklí posuzovat z papíru a jsme na něj jako typografové tak pyšní, se na obrazovkách promění a je z něho něco o čem jsme vůbec netoužili. V těch nejmenších velikostech naše výmysly zanikají, písmo je nečitelné a my jsme naštvaní. Hinting zaručí čitelnost i v těch nejmenších velikostech. Při mřížce, která může být vysoká i třeba jen 20 bodů zanikají ty nejmenší detaily a jedno písmo začíná připomínat to druhé. Jestli si někdo myslí, že tvorba písma je o tom dělat písma jinak, než ty před námi, tak se mýlí. Jestli jsou písma opravdu identická zkoumám v následujícím pokusu. Porovnávám bitmapy u písmen arial.ttf, georgia.ttf, tahoma.ttf, verdana.ttf, consola.ttf, lucon.ttf a times.ttf ve velikostech 8, 9, 10, 11 a 12 pixelů v jednobitovém režimu zobrazování.

počet shod	shodné písmeno		písmeno a velikost							
5	o	A 9	o	G 9	o	Ta 9	o	Ti 9	o	Ti 10
4	O	A 9	O	Ti 9	O	Ti 10	0	Lu 11		
4	8	A 9	8	G 9	8	Ta 9	8	Ti 10		
3	i	A 8	l	A 8	I	A 8				
3	D	A 8	D	Ta 8	D	Co 10				
3	e	A 9	e	G 9	e	Ta 9				
3	t	A 9	t	Ti 9	t	Ti 10				
3	j	G 9	j	Ti 9	j	Ti 10				
3	l	G 9	l	Ti 9	l	Ti 10				
3	u	G 9	u	Ti 9	u	Ti 10				
3	0	Ta 9	0	Ti 9	0	Ti 10				
3	e	G 10	e	Ta 10	e	Ti 11				
3	0	G 10	o	Ta 10	o	Ti 11				
3	o	G 11	o	Ta 11	o	Ti 12				

A=Arial, G=Georgia, Ta=Tahoma, Ti=Times, Lu=Lucida Console, Co=Consolas

6 návrh písma Delta

Po všech těchto průzkumech a pochopení problematiky jsem teprve mohl začít navrhovat konturu. Potenciál pro využití této technologie vidím hlavně u velkých firem, kde není známo kde všude se jejich písmo může objevit. Chci nabídnout alternativu k písmu Verdana, která bude odrážet vizuální kulturu dnešní doby a bude stejně kvalitně přizpůsobena jednobitovým displejům. Při hledání písma pro takové displeje jsme velmi omezeni a pravděpodobně nejčastěji stejně skončíme u Verdany. Do tohoto boje jsem šel s optimistickými představami, že objevím něco neobjeveného. Kvalitu Verdany jsem si ale krátce na to uvědomil a došlo mi, že tvarosloví, které má, nepůjde úplně tak opustit. Verdana je klasické písmo a nemůže si dovolit být trochu expresivnější protože jde o celosvětový projekt. Proces mé tvorby popisují chronologicky.

6.1 Konzultace

Na fóru TypeDrawers jsem narazil na písmového inženýra – Mika Duggana, byl velmi ochotný se mnou konzultovat a vysvětlil mi nejzákladnější principy, které člověk aplikuje při hintování písma. Mike pracuje v Microsoftu a má za sebou hintování například písma Times New Roman. Dal mi tedy dostatečně velký vhled do problematiky hintování a vysvětlil mi s jakými problémy se potýkají inženýři v praxi.

typedrawers.com, je forum se silnou komunitou, kde si lidé radí ohledně problémů týkajících se snad všech odvětví tvorby písma

6.2 Mřížka a kontura v ní

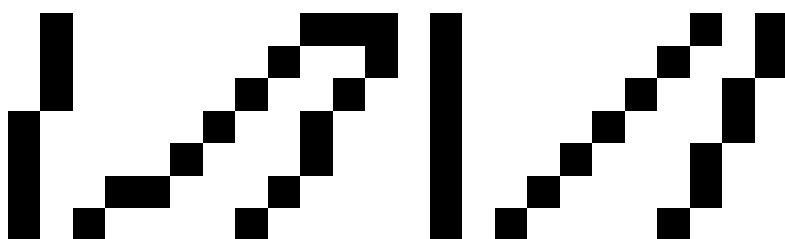
Při navrhování samotné kontury jsem chtěl vycházet z omezení daných touto nejpoužívanější mřížkou, tedy jak data ukazují, mřížkou o velikosti 14px. Právě velikosti okolo této, by měly být ty nejpoužívanější. Omezení, která určuje, by se měly promítnout do finální podoby kontury, aby originální kontura neztratila svou originalitu ve své bitmapové verzi. Skicoval jsem si do čtverečkováného bloku a nebo kreslil rovnou v počítači v jednobitovém režimu. Doufal jsem, že najdu tímto způsobem něco unikátního a písmo v jednobitové aplikaci bude velmi připomínat to, které známe v tištěné podobě. Pro výšku x jsem vymezil 8 bodů, verzálky jsou o 2 body vyšší a spodní dotaznice o 2 body nižší, horní dotaznice je pak o jeden bod vyšší než verzálky. Rozdíl mezi výškou x a verzálkami o právě 2 body je důležitý pro primární velikost 14px hlavně kvůli identičnosti kontur napříč verzálkami a mínuskami v této velikosti. Rozlišení mřížky přenesl do mřížky fontu, tak že jeden čtverec mřížky má rozměry 44*40. Vertikální drík má tedy tloušťku 44 bodu a horizontální tah 40. Písmo je navrhováno do několika kuželek, nejčastěji o šířce 4-6 bodů. Což vychází z faktu, že písmena v těchto kuželkách při 1bitovém zobrazování skončí, tak proč je tak nenavrhnout rovnou?

6.3 Bezpečné úhly

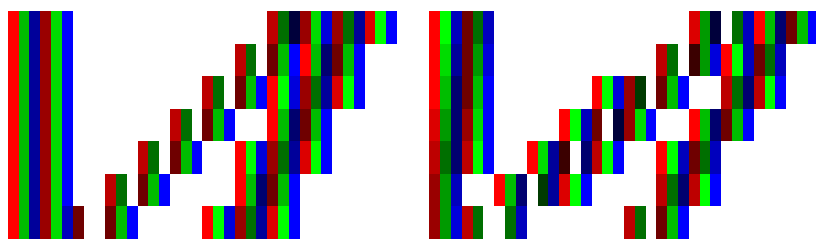
pro rovnoměrnou distribuci pixelů na diagonálách je velmi důležitý úhel ve kterém diagonály jsou. Úhel diagonály v písmu je jedním z úhlů diagonály obdelníku s poměrem stran 1:celé číslo. U 1:1 to je 45 stupňů, u 1:2 to je 26.5 atd. Tento tvar pak vyplňuje konturu, buď stoupá po jednom bodu doprava a nahoru a nebo o jeden doprava a dva nahoru. 90, 45, 26.5 a 18.5 stupňů jsou ty nejužitečnější bezpečné úhly pro obrazovku. V mém písmu se snažím všechny diagonály přizpůsobit těmto úhlům. Mezi 45 a 26.5 je poměrně velký rozdíl a když diagonála nespádá ani k jednomu z těchto úhlů, tak používám umělý bezpečný úhel, tzn. například úhel obdelníku 1:x.5. Jen ale ve velmi ojedinělých případech jsem byl přinucen bezpečný úhel vypustit.



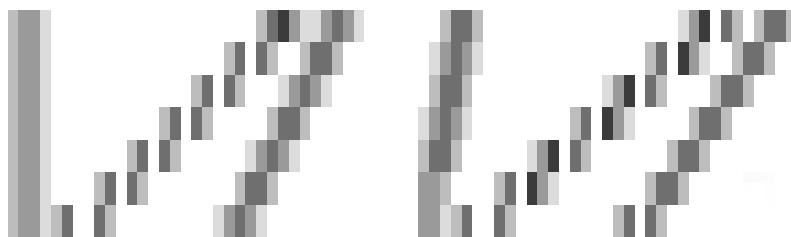
Obr. 8: vlevo tři diagonály pod úhlem 90, 26.5 a 18.5 stupňů, vpravo totéž ale vše o 10 stupňů zkoseno doprava. Jednobitové renderování



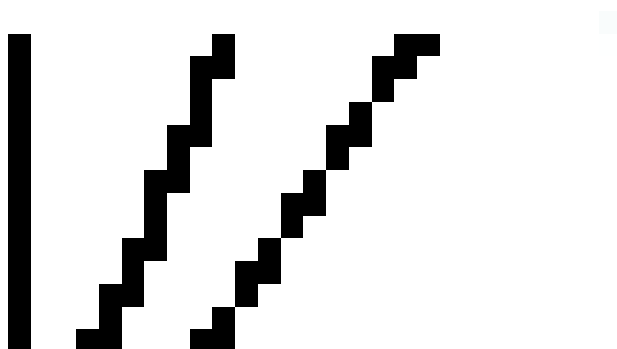
Obr. 9: stejné diagonály. Jednobitové renderování, velikost 14px (Aplikované delty na levý tah)



Obr. 10: stejné diagonály, TrueType render



Obr. 11: náhled intenzity svítivosti pixelů při rendrování TrueType



Obr. 12: tahy pod pseudo-bezpečnými úhly 33.7 a 21.8 stupňů. Distribuce pixelů není úplně rovnoměrná jako u bezpečných úhlů, určitý a stále čitelný vzor se zde ale nachází.

6.4 Kontura

Kontura je výsledkem nekonečného překreslování a zkoumání chování kontur při hintingu. Jak jsem již zmínil, důležitým faktorem pro písma v 1bitovém režimu je rovnoměrná distribuce pixelů, ideálně pod úhlem 45 stupňů a nebo pod jiným z bezpečných úhlů. Diagonály písmen se snaží držet úhel 45 stupňů. Kontura nepotřebuje striktně držet mřížku ze které vychází, to by škodilo na druhou stranu užití tištěnému. V mém písmu je pár momentu, kterými si usnažuji hintování. Může jít i o navíc umístěný bod, který by byl běžně považován za chybu ale pro hinting se nám hodí, protože od něj můžeme například interpolovat. Tento přístup je více viditelný například u písmene 'x', kde zdůrazňuji vertikální dírk, který se v malých velikostech bitmapových písem často objevuje. Tento element taky hodně pomáhá hintování samotnému. Potencionální digitální užití písma může být v případě mé práce primární ale určitě ne jediné, je důležité tedy myslet na to, aby písmo bylo kvalitní a originální v každé aplikaci. Písmo tedy musí splňovat i kvality které vyžadují tištěná média i digitální užití.

6.5 Hinting

Udržení rovnoměrné distribuce pixelů je velmi náročné při 1bitovém renderování, můžeme sami porovnat obr. 2 & 3. Osa X disponuje v případě TrueTypeu třikrát větším rozlišením, což nejsme schopni lehce na zařízeních pozorovat. Je důležité si vzpomenout, že pixel může být jakkoliv veliký a pozorován z jakékoliv vzdálenosti jak v kontextu tak bez kontextu. Subpixelu na monitoru počítače si nevšimneme, všimneme si ale subpixelu na velice

LED instalaci, ke které můžeme přistoupit z jakékoliv vzdálenosti? Estetika písma vychází z části z nástrojů, kterým disponuje hintovací program VTT. Nastavit zde úhel nejde přímo, se vzdáleností se ovšem dá pracovat několika způsoby. Známe-li vzdálenost dvou stran trojúhelníku, můžeme nastavit úhel diagonály pomocí nich. Tenhle způsob je limitován pouze dvěma věcmi. 1. je přehlednost pracovního souboru, pro každou vzdálenost si musíme do šuplíku vzdáleností uložit jednu proměnnou. Jestli je takových proměnných spousta, je pravděpodobné, že nad proměnnými ztratíme přehled. 2. výška trojúhelníku je omezena ještě neznámou proměnnou - výškou dotažnic. Pro vertikální pohyb je tedy zapotřebí volný prostor pro natahování vertikální strany trojúhelníku. Jednoduše řečeno - výška diagonály natažené od účaří k výšce x je dána výškou x , nemůžeme jí tedy nastavit absolutní hodnotou.

6.6 Prezentace a testování na displayích Nokia

Nokia display 5110 se nyní prodává jako otevřený modul a je připravený na to abych se do něj dostal a zároveň je zástupcem jednobitových displejů na které se zaměřuji a proto to byla jasná volba pro prezentaci a testování mého písma Delta. Jako počítač pro vykreslování písmen používám Raspberry Pi Zero WH, na kterém běží program napsaný v Pythonu 3, který jako rasterizér používá knihovnu FreeType, výsledek je tedy totožný s tím jak by se písmo chovalo na zařízení které je postaveno na Linuxu. Testovat písmo právě na tomto rasterizéru je dobrou myslím dobrou volbou hlavně proto, že je velká možnost, že koncové užití písma bude právě na zařízení s tímto rasterizérem.

6.7 Budoucnost mé bakalářské práce

Projekt bych budu sdílet na Githubu a začnu postupně přidávat podporu dalších jazyků. Zkušenosti, které jsem nabral na latině bych rád aplikoval i do jiných písmových systémů, kde technologie nemusí být tak vyvinuta a potřeba dalšího současného písma s hintingem je potřeba.