

Jan Šindler

písmo pro digitální zobrazování, technologie TT hintingu

Tvorba písma a typografie, bakalářská práce, 2019

1 Co je to hinting?

Petr Bilak vic definici Hinting, or screen optimising, is the process by which TrueType or PostScript fonts are adjusted for maximum readability on computer monitors.

1.1 Hinting dnes

je umírajícím řemeslem, pro většinu případů stačí auto-hinting, v dalších případech o tento aspekt fontu designér nejeví zájem protože pracuje na zařízení s vysokým rozlišením. To se projeví až u koncového uživatele. Nutno přiznat, že hinting není tak potřeba jak dříve pro užití na stolních počítačích. Dokonce i já jako středoevropan s poměrně výkonným počítačem si všímám nekvalitně udělaných písem na webu, ty pak často velmi znepríjemňují čtení na digitálních zařízeních.

1.2 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 písmo. TrueTypový může fungovat buď na základě postscriptového hintingu (bude fungovat přesněji než kdyby se algoritmus pokoušel o automatizaci hintingu bez něj). TT autohinting může narazit v případě kdy jde o nestandardní písmo, například když konce tahů nekončí přímo na dotaznicích. 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 nebyl takovým problémem sám o sobě, písmo působí efektem 3D rotace a při pohledu z 90 stupňů se začnou překrývat a kontura se začne zaokrouhlovat na rozdílné strany, to vytváří na kontuře nepříjemné schody. V normální situaci by tohle hinting vyřešil, jelikož ale chceme plynulou animaci tak v ose X žádný hinting aplikovaný být nemůže, kontura se vymění za obdelník stejných rozměru přesně při devadesáti stupních, při

pohledu ze strany tedy na kontuře nevznikají žádné schody. Aby přechod mezi znaky a obdelníky byl plynulý, musí i tyto obdelníky být vyhinitované stejně jako znaky, které zastupují. Stroj tyto souvislosti neví a tak bylo potřeba toto písmo vyhinitovat ručně.

2 programy okolo hintingu

neco napsat o programech obecně

2.1 VTT

Link - základní nástroj pro, kterým určujete jaký vztah jeden kotevní bod naplňuje k druhému. Může jít o jednoduchý pohyb - pohne-li se jeden, pohne se i druhý a nebo že tyto dva body budou od sebe vzdáleny stejný počet pixelů jako druhé dva body. Delta - posouvání buď globálních a nebo lokálních hodnot - nastavíme li že všechny body na x-výšce jsou označeny číslem 6, 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 1/8 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. Interpolate - interpoluje jeden bod na základě dalších dvou - něco jako když na pružinu umístíme bod a pak s jí natahujeme, bod bude vždy ve stejném vzdálenostním poměru ke krajům pružiny.

2.2 rasterizéry

1. ClearType - rasterizer Windowsu, funguje na bázi rozdělení pixelu na sub-pixelu. RGB = 1 pixel, R = 0.33 pixelu
2. FreeType - opensourcový projekt, který najdeme ve velkém množství zařízení. Od Androidu/Linuxu až po telefony společnosti Apple. Jde o opensourcovou variantu ClearType
3. 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.
4. CoolType - je metoda rastrování textu, kterou využívá Adobe a jejich řada programů

3 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. BigQuery 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.

3.1 font-size

Obrazovky se dnešních počítačů v Evropě a dalších rozvinutých zemích problémy se zobrazováním písma skoro nemají, nesmíme ovšem opomíjet země, kam naše technologie ještě nedorazila a navrhovat písmo i pro ně. Velké firmy by měly mít písmo, které obstojí v jakékoliv situaci a to i na tom nejslabším zařízení. Typickým příkladem může být displej pračky, digitální cenovka v supermarketu a nebo jednoduchý počítačový systém, který nepodporuje nejmodernější zobrazovací technologie, příkladem může být platební terminál. Všechny tyto zařízení vyžadují buď aplikace firemních písem, log a nebo piktogramů. Zde všude může hinting velmi pomoci. Písmo jsem navrhoval primárně pro nějakou velikost, která je lidmi nejpoužívanější, svůj požadavek jsem zadal pouze na webové nastavení, hlavně protože je ustálené a programátoři velikost vybírají s vědomím, že neví na jakém zařízení se jejich web zobrazí. 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. Pro mojí analýzu jsem využil služby BigQuery od Googlu, která dovoluje zkoumat velká množství dat pomocí jejich počítačů. Data jsou buď externí a nebo přímo od Googlu, takový soubor dat se nazývá dataset. GitHub je platforma pro vývojáře, kde si lidé mohou ukládat a sdílet právě vyvíjenou aplikaci buď mezi sebou a nebo mezi veřejnost. Jeden z datasetů githubu má 2,3TB, já jsem svůj pokus dělal pouze na neplacené části 30GB a i tak jsem dostal velké množství dat, kterou jsou svojí kvantitou dostačující. Programátoři využívají z velké části jednotky PX oproti PT, které jsou určeny spíše pro tištěné výstupy. 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 písma 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 takovému zobrazení.

jednotka	medián	m. průměr	počet nálezů
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

3.2 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 nejpoužívanějšími fonty je pouze jedno serifové, velmi překvapivé je, že Times New Roman se do téhle dvacítky vůbec nedostal. Zdali 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

4 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 křivky na tom rastru nám jde především když je velikost mřížky omezena. Uvádím zde tři příklady, kdy se taková distribuce hodí i pro aplikace jiné, než je obrazovka.

1. **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 jednou či později opět vrátí.
2. **Práce se samotným pixelem** obrázek není jenom sled obrázku a bodů ale přesná mřížka s přesnými údaji, to v dnešní době můžeme využít a můžeme pracovat se samotnými pixely. I trochu zruční designéři jsou už schopní si vyrobit vlastní rasterizer, který bude tvořit například vizuál pouhým festivalu aplikováním rasterizeru na fotografii. To nám skvěle předvedl například 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 obličejem politiků, 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.
3. **Průmyslová produkce návrhů** jsou technologie, kdy při realizaci návrhu je míra detailu velmi omezena. Typicky může jít o pletení, tkání, vyšívání a nebo perforace do nejrůznějších materiálů.

4.1 O u Timesu je stejné jako u Arialu

Unikátní tvar písmene, který jsme zvyklí posuzovat z papíru a jsme na něj 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

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 I 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

5 návrh písma Delta

Po všech těchto průzkumech a pochopení problematiky jsem teprve mohl začít navrhovat konturu. Proces popisuji chronologicky

5.1 Mike Duggan

Na fóru TypeDrawers jsem narazil na písmového inženýra - Mika Duggana, byl velmi ochotný se mnou konzultovat a vysvětlil mi ne-
základnější principy, které člověk aplikuje při hintování písma. Mike pracuje v Microsoftu a má za sebou hintování například písma Times New Roman.

5.2 Mřížka a kontura v ní

Při navrhování samotné kontury jsem chtěl vycházet z omezení daných tou nejpoužívanější mřížkou, tedy mřížkou při velikosti 14px. Omezení, která určuje by se měly promítnout do finální podoby kontury. Pro výšku x jsem vymezil 8 bodů, verzálky jsou o 2 body vyšší a spodní dotažnice o 2 body nižší, horní dotažnice 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.

5.3 kontura

Střed písmen prochází hned o bod výš nad matematickým středem písma, písmu tak spolu s netradičně seřezanými diagonály dodává dynamiku. Diagonály písmen se snaží držet úhel 45 stupňů, horizontální-vertikální seřezávání diagonál usnadňuje práci s tímto úhlem a zároveň písma nepřesahují svojí optickou šířku.

5.4 prezentace a testování na Nokia displayích

Nokia display 5110 je dostatečně otevřený 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.