Co je to hinting?

Petr Bilak

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

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 znepříjemňují čtení na digitálních zařízení.

Druhy Hintingu

TrueTypový hinting je velmi silný nástroj, který umožňuje modifikace písmen každé velikosti a 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.

PostScriptový hinting

Postcriptový hinting je modernější forma hintingu, je jednodušší než TrueTypový ale neumožňuje takové modifikace.

Autohinting

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 nestandartní 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 nebyl takovým problém sám o sobě, písmo působí efektem 3D rotace a při pohledu z 90° se začnou překrývat a kontura se začne zaokrouhlovat na rozdílné strany, to vytváří na kontuře nepříjemné schody. V normálni 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 rozměnu 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 vyhintovat ručně.

VTT

Visual TrueType (VTT), je nástroj pro platformu Windows, který usnadňuje zpracování hintingu pro formát .ttf. TrueType je programovací jazyk sám o sobě, VTT je tedy vizuální programování, kde se dá písmo vyladit do těch nejmenších detailů. Kontuře písma se udávají pravidla podle kterých má svůj tvar měnit podle velikosti fontu. Člověku to nedojde, ale určitě bude velice překvapen když zjistí jak vypadá skutečná kontura písmena 's' v písmu Verdana pří velikosti 10 a méně. TrueType umožňuje do velké míry vyladit písmo pro B/W displeje a zároveň nechat tyto instrukce vypustit u displejů, které je nepotřebují nebo může být absolutně explicitní a pro každý rasterizer udělat hinting zvlášť. Jednoduše řečeno, jde o určování pravidel jak se má písmo přizpůsobovat aby padlo do mřížky pixelů, kde pravidla nepomohou, tak nastupuje ruční dolazení pro každou velikost zvlášť. Většinou se takto ladí jen ty nejmenší velikosti ale ve velkých projektech jakým byla i již jmenovaná Verdana je takto vylazena až do velikosti 60px, poté už člověk ztráci přehled o pixelech samotných a začíná je vnímat jako hmotu. Není teda tak důležitá jejich symetrická a oku přívětivá distribuce.

Základní nástroje 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 natahujem, bod bude vždy ve stejném vzdálenostním poměru ke krajům pružiny.

Písmové rasterizéry

Rasterizery

ClearType - rasterizer Windowsu, funguje na bázi rozdělení pixelu na sub-pixely. RGB = 1 pixel, R = 0.33 pixelu

FreeType - opensourcový projekt, který najdeme ve velkém množství zařízení. Od Androidu/Linuxu až po telefony společnosti Apple. Jde o opensourcovou variantu ClearTypu

Quartz - rasterizer MacOs, ten ignoruje veškteré 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.

CoolType - je metoda rastrování textu, kterou využívá Adobe a jejich řada programů

Analýza velikostí písem na GitHubu pomocí služby Google BigQuery

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 určené 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 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 takovýmto zobrazení.

PX

Medián: 14.0   
Matematický průměr: 26.83304934092271  
Vstupů: 13352

PT

Medián: 10.0 (=13.332999999999998PX)  
Matematický průměr: 12.308570268074735   
Vstupů: 1231

EM

Medián: 1.2   
Mathematical Average: 3.0409005795574626   
Vstupů: 7225

%

Medián: 100.0   
Mathematical Average: 114.99321181727609   
Inputs: 6295

Mým dalším cílem zkoumání pomocí služby BigQuery bylo i jaké písma uživatelé používají. Vysledky 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 naších preferencí nastavit fonty dle libosti, pokud první v systému není, aplikuje se ten další atd. příklad font-family="Arial, Helvetica, Verdana".

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

Body

Arial 1510, 24.35483870967742

Verdana 1609, 25.951612903225808

Times New Roman 2684, 43.29032258064516

identičnost distribuce pixelů napříč písmy  
počet shod  
 písmeno  
 písmo\_velikost písmena  
5 o arial\_9 o georgia\_9 o tahoma\_9 o times\_9 o times\_10  
4 O arial\_9 O times\_9 O times\_10 0 lucon\_11  
4 8 arial\_9 8 georgia\_9 8 tahoma\_9 8 times\_10  
3 i arial\_8 l arial\_8 I arial\_8  
3 D arial\_8 D tahoma\_8 D consola\_10  
3 e arial\_9 e georgia\_9 e tahoma\_9  
3 t arial\_9 t times\_9 t times\_10  
3 j georgia\_9 j times\_9 j times\_10  
3 l georgia\_9 l times\_9 l times\_10  
3 u georgia\_9 u times\_9 u times\_10  
3 0 tahoma\_9 0 times\_9 0 times\_10  
3 e georgia\_10 e tahoma\_10 e times\_11  
3 0 georgia\_10 o tahoma\_10 o times\_11  
3 o georgia\_11 o tahoma\_11 o times\_12

Pixely mimo obrazovky

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.

**• 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.

**• Produkce modních návrhů**jsou technologie, kdy při realizaci návrhu je míra detailu velmi omezena. Typicky může jít o pletení a tkaní.

**• 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 trošku 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 veliksoti 2x2 pixelů, data obrázku by vypadala asi takto [[255, 0], [0, 255]]. S daty pak můžeme libovolně pracovat. Můžeme například všechna políčka, kde je bíla (255) spojit čárou a nebo je nahradit nějakým symbolem. Můžeme je nahradit třeba obličeji 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.