TOMÁŠ MIKULA

UŽIVATELSKÁ ROZHRANÍ

KATEDRA INFORMATIKY
PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO

Copyright © 2020 Tomáš Mikula

KATEDRA INFORMATIKY
PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO

Obsah

Stručná historie uživatelského rozhraní 7

Design a psychologie 23

Grafické uživatelské rozhraní 55

Elementy grafického uživatelského rozhraní 93

Literatura 107

Úvod

„The next big thing is the one that makes the last big thing usable.“

- Blake Ross, *Co-creator of Mozilla Firefox*

„If you think good design is expensive, you should look at the cost of bad design.“

- Dr. Ralf Speth, *Chief Executive Officer, Jaguar Land Rover*

Pokud poprvé spustíte textový editor Vim[1], je pochopitelné, že budete mít problémy jej vypnout. Pokud se budete snažit složit nabýtek IKEA bez přiloženého návodu, je pochopitelné, že to nepůjde. Pokud ale používáte program jako je Microsoft Word[2], očekáváte, že nebude problém naformátovat a vytisknout dopis. Opak je však často pravdou, a to nejenon v případě programu Microsoft Word.

Uživatelské rozhraní je všude okolo nás. Toho kvalitně navrženého si většinou ani nevšimneme, v opačném případě mnohdy ztratíme několik hodin času bez jasného výsledku.

Co rozhoduje o kvalitě uživatelského rozhraní? Jak vlastně lidé interagují s objekty? Na tyto otázky se pokusíme odpovědět v následujících kapitolách.

[1] Jeden z nejpopulárnějších editorů fungující v příkazové řádce, odvěký rival editoru Emacs. Známý tím, že jej spousta lidí nedokáže ani vypnout, natož v něm pracovat.

[2] Populární textový procesor od firmy Microsoft, na začátku se jednalo o poměrně intuitivní software, který se stal postupem času pro většinu uživatelů příliš komplikovaným.

Stručná historie uživatelského rozhraní

V této kapitole se budeme věnovat historii uživatelských rozhraní. Za uživatelské rozhraní lze považovat jakýkoli způsob interakce člověka se strojem nebo objektem.

Ke konci tato kapitola obsahuje novodobá zařízení či aplikace. Ačkoli se nejedná o historii v pravém slova smyslu, je přínosné vidět tyto zařízení v kontextu historických.

První počítače

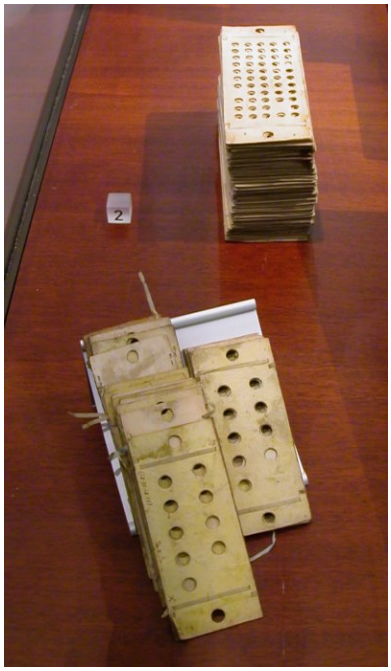
Charles Babbage v roce 1837 přichází s návrhem obecně použitelného mechanického počítače – Analytický stroj. Uživatelský vstup byl zprostředkován pomoci *děrných štítků*. Pro výstup byla použita tiskárna, plotter a zvonek.

V roce 1945 ve svém článku *As We May Think* přichází Vannevar Bush s konceptem zařízení s názvem *Memex*. Zařízení mělo umožnit „zvětšení lidské paměti". Navrh popisoval stroj ve formě pracovního stolu[3] s několika obrazovkami a klávesnicí. Hlavním účelem mělo být ukládaní a čtení informací ve formě asociací – podobně jako lidský mozek.

Koncept Memexu významně ovlivnil vývoj hypertext systému např. World Wide Web (WWW) a systému známých pod názvem Personal knowledge base sloužících k uchování, reprezentaci a následného získání znalostí uživatele.

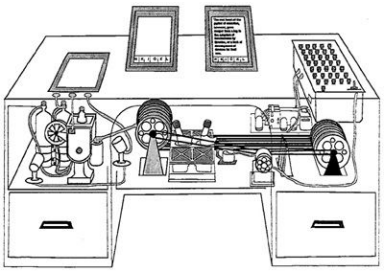
Dávkové zpracování

Okolo roku 1950 se začínají objevovat sálové počítače (také známé pod názvem *mainframe*). Provoz počítačů byl natolik nákladný, že pro jejich efektivní využití bylo potřeba zadávat vstup v takzvaných *dávkách*. Každá dávka se skládala ze sady děrných štítků, případně papírových pásek. Spuštění výpočtu vyžadovalo přípravu programu a vstupních dat na děrných štítcích. K tomuto účelu sloužily speciální stroje – *děrovače* – podobné psacím strojům. Výstup zprostředkovávala řádková



Obrázek 1: Děrné štítky používané analytickým strojem.

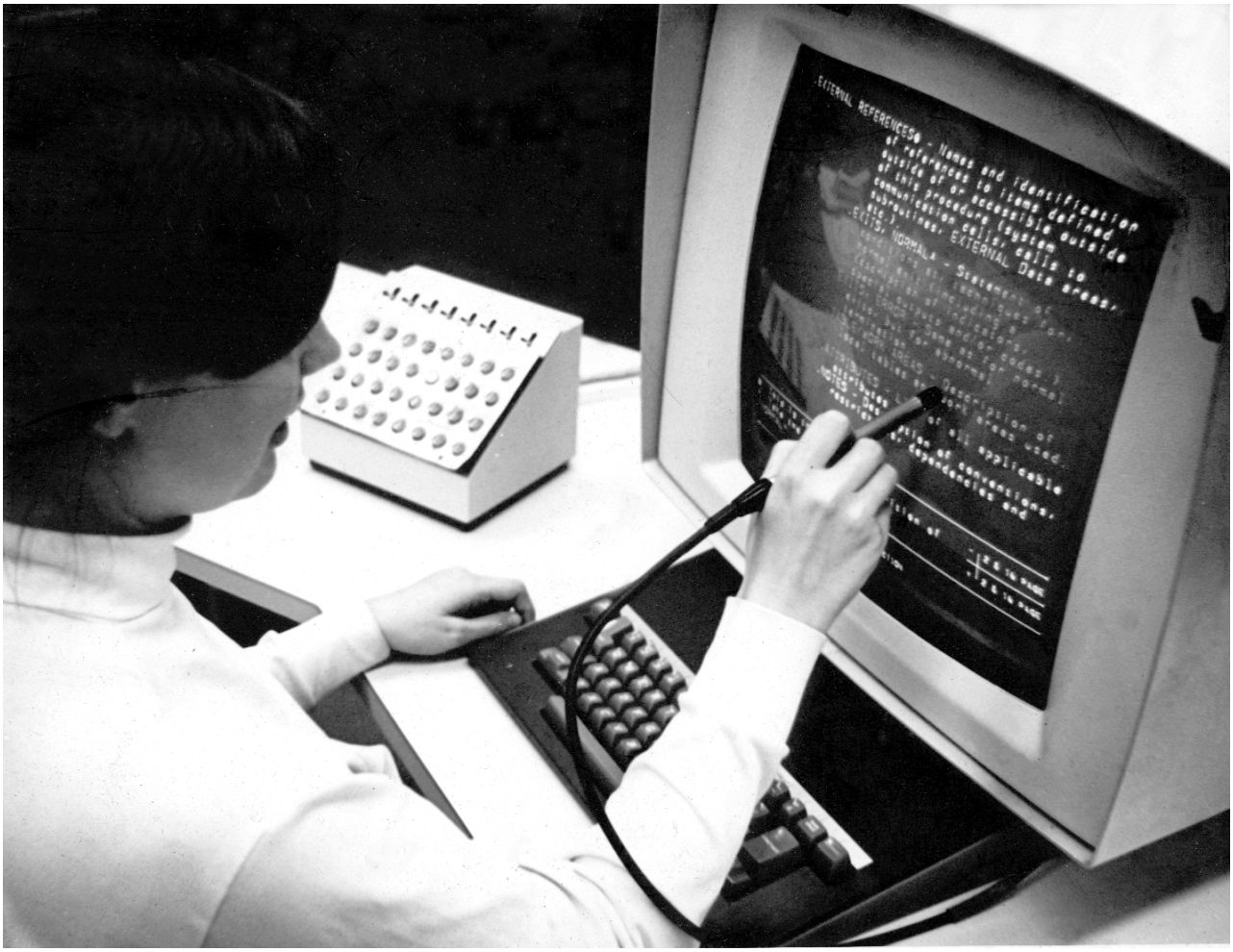
[3] Všimněte si, že ani v teoretickém konceptu si nikdo nedokázal představit zařízení velikosti hodinek.



Obrázek 2: Koncept zařízení Memex.

tiskárna. Mezi člověkem a počítačem neprobíhala žádná interakce v reálném čase. V případě chyby na vstupu nedocházelo k žádné okamžité zpětné vazbě.

Roku 1963 Ivan Sutherland prezentuje v rámci svého PhD. studia revoluční software s názvem *Sketchpad*. Sketchpad považujeme za první graficky orientovaný systém. Sketchpad byl vzdálený předchůdce CAD systémů a první interaktivní systém v době dávkového zpracování. Ivan Sutherland využil *light pen* – speciální světelné pero, které uživateli umožňovalo ukazovat na daný bod na CRT obrazovce.



Obrázek 3: Demonstrace zařízení Lightpen.

Příkazový řádek

Rozhraní příkazové řádky (CLI) bylo vyvinuto na základě monitorů dávkového zpracování, jeho odezva však byla výrazně rychlejší (došlo ke zrychlení z několika dní či hodin na sekundy). Uživatelský vstup

je zadáván formou textových příkazů ze speciálního slovníku (sada příkazů). Výstup je zobrazen ve formě textu. Jedná se tedy o *request-response* komunikaci mezi uživatelem a terminálem. Díky zrychlené komunikaci bylo možné proces v reálném čase ovlivnit a získat lepší zpětnou vazbu v případě chyby. Stále je však potřeba znalost příkazů a interakce s počítačem není intuitivní.

Příkazový řádek jako uživatelské rozhraní je dodnes hojně využíváno. A to především pro svojí rychlost a efektivitu (klávesnice vs. počítačová myš).

Textové uživatelské rozhraní

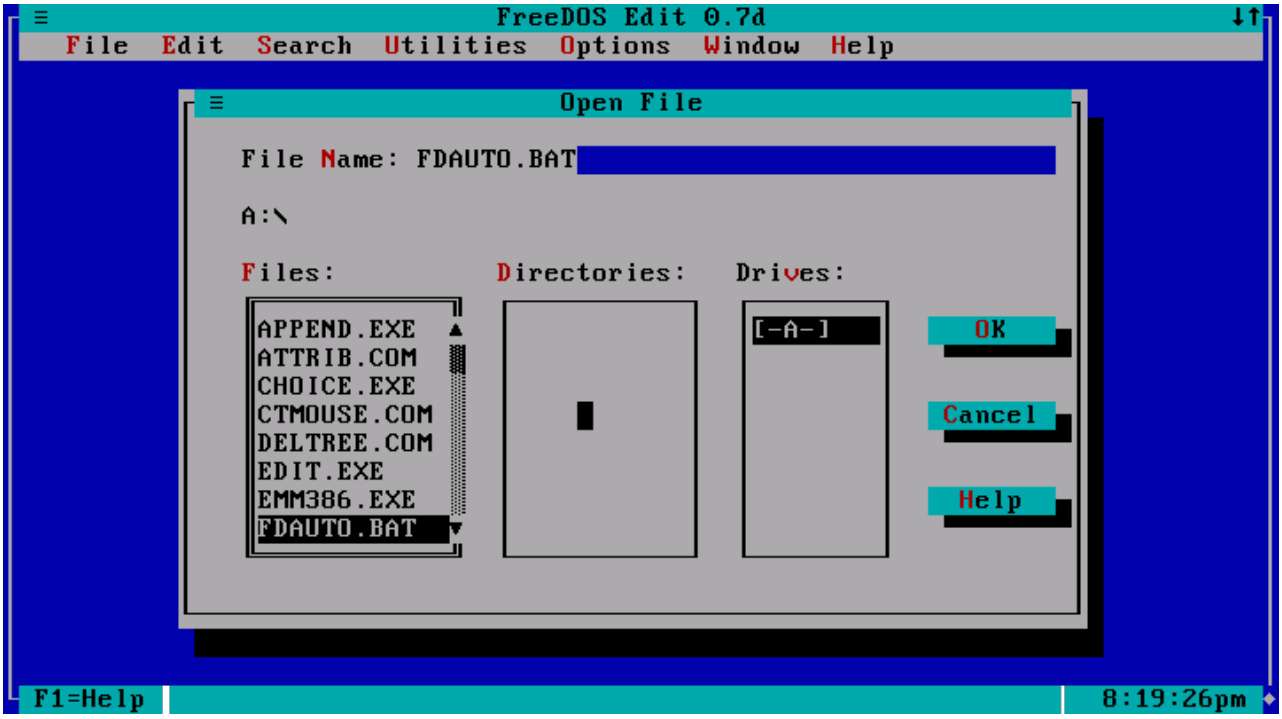
Textové uživatelské rozhraní (TUI) je vývojový mezistupeň mezi rozhraním příkazové řádky a grafickým uživatelským rozhraním (GUI). Textové rozhraní jak už název napovídá pracuje v textovém režimu, při kterém je obrazovka rozdělena na 2D rastr. Každá pozice rastru může zobrazovat pouze jeden znak (nejčastěji z ASCII tabulky).

Revoluční posun nastal v možnosti využití myši a klávesnice pro ovládání celo-obrazové aplikace. Uživatelé mohli využívat prvky, které známe z novodobých GUI prostředí – tlačítka, menu, posuvníky, rolovací seznamy atd. Programy se tedy staly uživatelsky přívětivější a poskytovaly srozumitelnější zpětnou vazbu.

Obrázek 4: Dostupné znaky na tvorbu uživatelského rozhraní.

Obrázek 5: IBM Common User Access (CUA).

Systems Application Architecture (SAA) bylo uvedeno v roce 1987, jedná se o skupinu standardů počítačového softwaru vytvořených firmou IBM. Hlavním záměrem bylo unifikovat styl fungování počítačového softwaru. Rodina standardů SAA obsahuje standard pro uživatelské rozhraní operačních systému a počítačových programů – Common User Access (CUA).



Obrázek 6: Textové uživatelské rozhraní – FreeDOS.

Tento standard zavedl uspořádání ovládacích prvků tak, jak jej známe dnes[4]. Definoval například pozici status baru (spodní okraj obrazovky), pulldown menu (horní okraj obrazovky) a konzistenci klávesových zkratek napříč softwarem.

[4] Tento standard silně ovlivnil počátky operačního systému Windows, částečně je rovněž použit ve variantách operačního systému Unix.

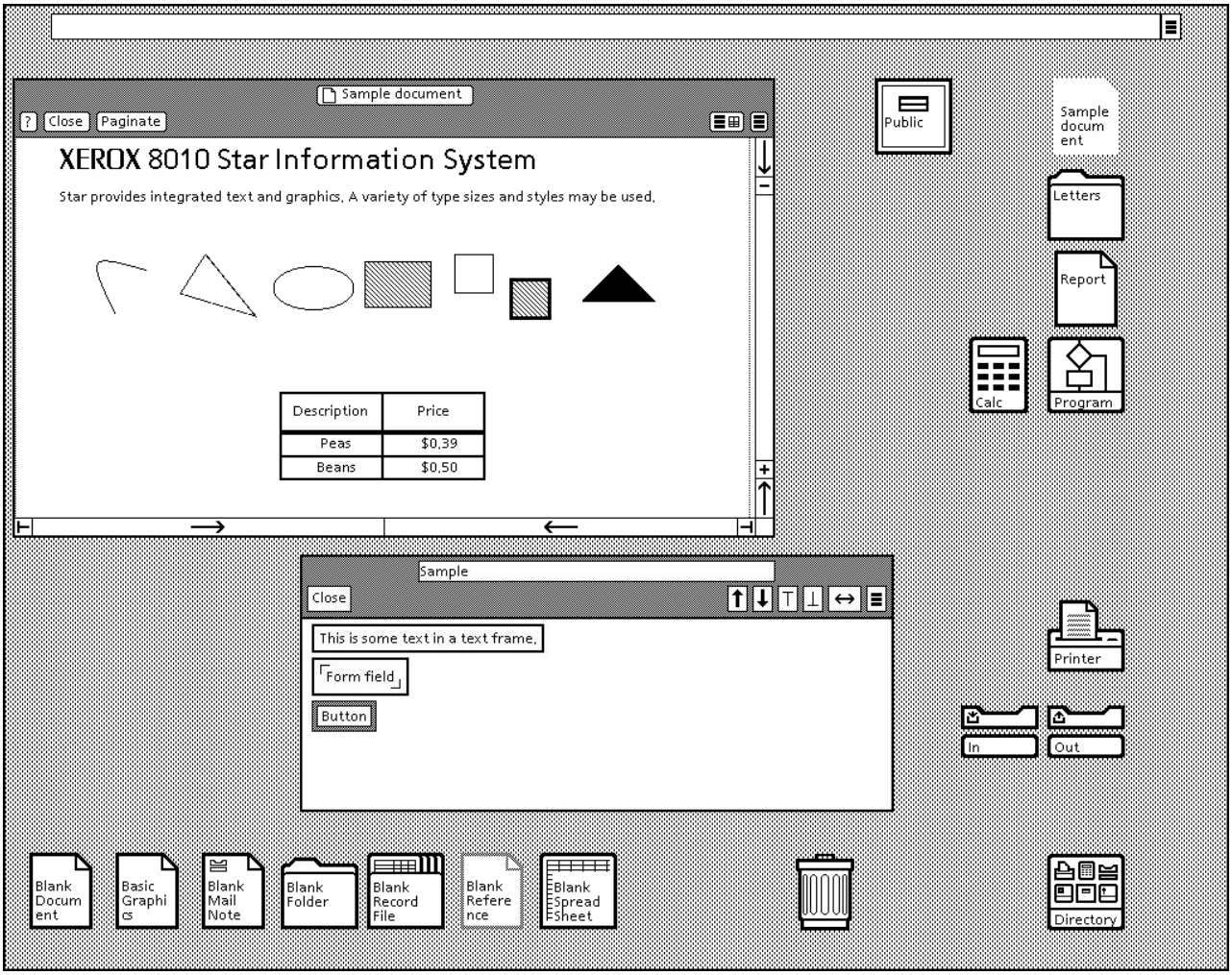
Grafické uživatelské rozhraní – oN-Line System

V roce 1968 přichází Douglas Engelbart s demonstrací systému oN-Line System (NLS), který využíval myši, ukazatele, hypertextu a několika oken (dříve jedno-okení aplikace). Mimo to systém umožňoval video konference, kontrolu revizí a kolaborativní real-time editor.

Obrázek 7: První prototyp počítačové myši.

WIMP paradigma

WIMP (Windows, Icons, Menus, Pointers) paradigma vytvořené výzkumníky z Xerox Palo Alto research center. WIMP je často zaměňován s pojmem GUI. Grafické rozhraní je však součástí WIMP paradigmatu, není ale nutné, aby GUI používalo veškeré prvky WIMP paradigmatu. Z dnešního pohledu je WIMP paradigma stále hojně využíváno, jak ale uvidíme později, je postupně nahrazováno Post-WIMP paradigmatem.



Obrázek 8: WIMP paradigma.

Xerox Alto

V roce 1973 byl představen první počítač navržený s důrazem na GUI[5] – Xerox Alto – jednalo se o počítač určený pro kancelářskou práci. Pro vstup byla použita klávesnice a myš. Jeho rozhraní nebylo zcela jednotné, ale umožňovalo používat více oken najednou. Kvůli jeho vy-

[5] Založeno na principu NLS

soké ceně nebyl komerčně úspěšný[6]. Xerox Alto začal být vnímán jako budoucnost uživatelského rozhraní. Roku 1979 Steve Jobs sjednal pro svůj tým inženýrů ukázku použité technologie v Xeroxu, výměnou za možnost nákupu akcií firmy Apple.

[6] Zde vidíme, že i v případě dobrého designu je často složité udržet produkty komerčně úspěšné. Cena Xerox Alto byla po přepočtu na rok 2017 cca $108 075.

Atari 2600

Nejpopulárnější herní konzole z takzvané druhé generace herních konzolí spatřila světlo světa v roce 1977. Jednalo se o jednu z prvních konzolí, které využívaly výměnné moduly s hrami. Bylo možné používat tří typy ovladačů: 8 směrový joystick s jedním tlačítkem, paddle (analogový ovladač s jednou osou) a číselné klávesnice s 12 klávesami.

Obrázek 9: Atari 2600.

Xerox Star

Xeroxu se povedlo koncept Xerox Alto v roce 1981 komercializovat ve formě produktu s názvem Xerox Star – značně modifikované verzi počítače Xerox Alto. Jednalo se o první komerční počítač, který využíval dodnes používané koncepty – bitmapový display, grafické prostředí založené na oknech, ikony, složky, myš s dvěma tlačítky, ethernet připojení, souborový server, tiskový server a e-mail. Rovněž se poprvé

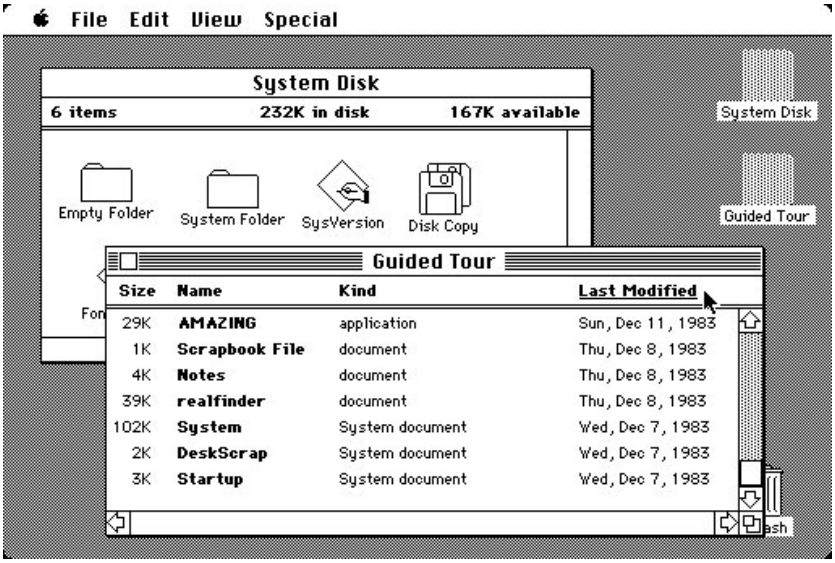
objevuje metafora pracovní plochy a ikon. V tomto přístupu hrálo klíčovou roli napodobení kancelářských předmětů se snahou vytvořit intuitivní uživatelské prostředí. Takový přístup k vývoji uživatelského rozhraní nazýváme *objektový*. Textový procesor obsahoval objekty jako odstavec, stránka, věta atd. Uživateli bylo umožněno s objekty interagovat a provádět různé operace. Koncept *What you see is what you get* (WYSIWYG) byl pro Xerox Alto klíčový.

Obrázek 10: Xerox Alto.

Apple Lisa a Macintosh

Po návštěvě Xerox PARC Apple okamžitě přesměroval projekt Apple Lisa z textového uživatelského prostředí na grafické[7].

Ve stejné době byl spuštěn projekt Macintosh, u kterého byl důraz kladen na nízkou cenu a jednoduchost použití pro běžného uživatele[8]. Originální koncept Macintosh obsahoval textové uživatelské prostředí, které umožňovalo přepínat aplikace pomoci klávesových zkratek. Během vývoje hardware bylo docíleno kombinace nízké ceny počítače Apple II a výpočetního výkonu Apple Lisa. Tato souhra umožnila pozdější nasazení grafického uživatelského rozhraní podobného tomu na Apple Lisa. Tímto si projekt získal pozornost Steva Jobse, který okamžitě uviděl potenciál. V roce 1984 se Macintosh dostává do prodeje.



Obrázek 12: Macintosh UI.

[7] Apple Lisa měl být nástupcem Apple II, který používal textové uživatelské rozhraní. Apple Lisa měl v tomto trendu pokračovat.

[8] Během této doby můžeme pozorovat souboj firem o vytvoření prvního uživatelsky přívětivého počítače

Obrázek 11: První Apple Macintosh 128k.

Microsoft Windows 1.0 až 3.1

V roce 1985 Microsoft začal pracovat na programu Interface Manager, který byl poté oznámen pod názvem Windows.

Windows 1.0 byl navržen s cílem soupeřit s operačním systémem firmy Apple, nezískal však velkou popularitu. Nejednalo se o samostatný operační systém, ale pouze o nástavbu MS-DOS. Windows 1.0 nepodporoval překrývání oken (dostupné byly pouze modální dialogové boxy) a neobsahoval pracovní plochu.

Obrázek 13: Windows 1.0.

Windows 2.0 již podporoval překrývání oken[9].

Windows 3.1 byly první populární operační systém od firmy Microsoft. Stále však neobsahoval pracovní plochu.

[9] Tato změna způsobila žalobu ze strany firmy Apple.

Microsoft Windows 95

V roce 1995 vzniká operační systém Windows 95, který již obsahoval uživatelské rozhraní založené na objektech a metaforách. Byla zavedena ikonická nabídka Start, TaskBar (panel na spodní hraně obrazovky), pracovní plocha a koš. Tyto prvky uživatelského rozhraní

vidíme v operačních systémech Microsoft Windows dodnes.

Obrázek 14: Windows 95 vs. Windows 10.

Nokia 2110

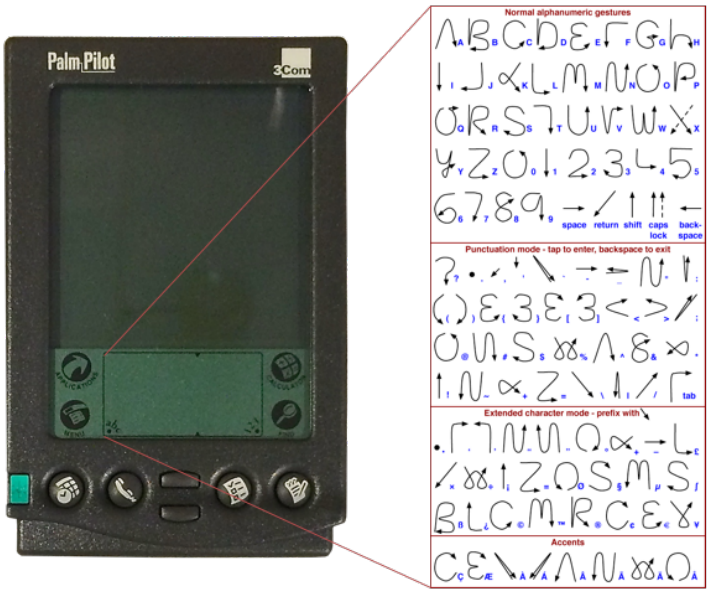
V roce 1994 firma Nokia představila mobilní telefon Nokia 2110. Telefon byl znatelně menší než konkurenční a obsahoval větší display. Díky těmto parametrům se stal jedním z nejoblíbenějších telefonů té doby. Z hlediska uživatelského rozhraní přinesl koncept dvou dynamických tlačítek, kterým se kontextově měnila funkce. Na základě těchto tlačítek přišla později firma s konceptem již dobře známeho Navi-Key u modelu 3110.

Obrázek 15: Nokia 2110.

Palm (PDA)

Zařízení od firmy Palm jsou často vnímány jako první masově rozšířené mobilní počítače. V roce 1996 firma produkuje první PDA PalmPilot 1000 poháněné operačním systémem Palm OS. Zařízení obsahovalo monochromatický display s rozlišením 160x160px, ve spodní části

displaye byla takzvaná „Graffiti input zone“. Graffiti rozpoznával jednotlivé znaky, které uživatel mohl za pomoci speciálních značek do zařízení zapisovat. Značky byly navržené tak aby šly napsat jedním tahem.



Obrázek 16: PalmPilot 1000 a Graffiti input zone.

Post-WIMP paradigma

Postupem času začalo být WIMP paradigma rozšiřováno o další principy. Dělo se tak především z důvodů nástupu většího nasazení CAD systémů, interaktivních video her a práce s větším množstvím dat. WIMP paradigma vyžadovalo použití nemalé části obrazovky pro zobrazení uživatelského rozhraní, v mnoha aplikacích začalo být výhodnější přicházet s vlastním řešením.

Většina běžných počítačů stále funguje na principu WIMP paradigmatu. Velký posun ale nastal u zařízení používajících dotek jako uživatelský vstup. Dále pak s nástupem webových aplikací a virtuální reality.

Apple iPhone

Před vydáním prvního telefonu iPhone na trhu dominovaly firmy jako Nokia a Motorola. Tyto firmy těžily primárně z potřeb korporátních zákazníků. Každá značka prodávala nespočet různých modelů určených různým cílovým skupinám. V roce 2007 Steve Jobs představil

první iPhone. Telefon, který se pokusil sjednotit vše do jednoho zařízení.

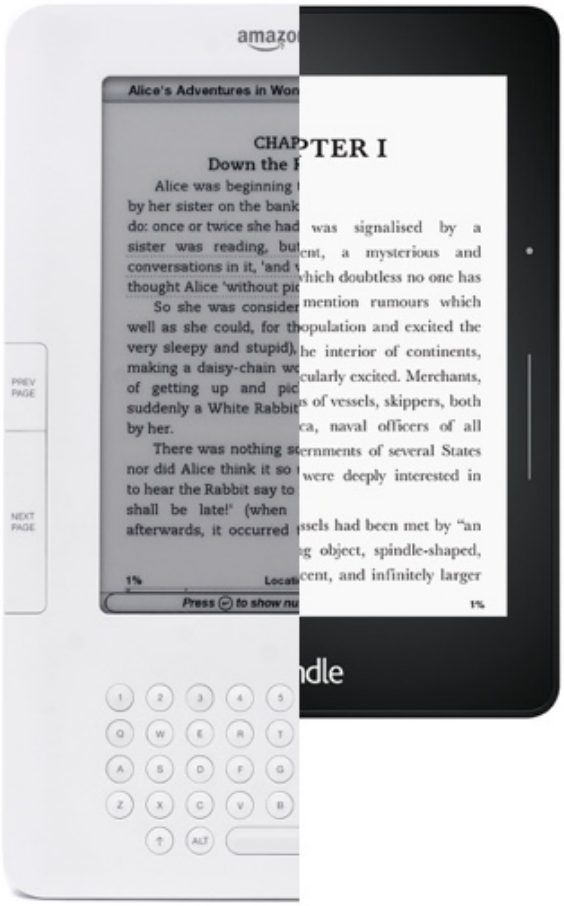
Obrázek 17: Apple iPhone první generace vs. iPhone Xs

Průlomové bylo použití kapacitního dotykového displaye. Pro ovládání telefonu nebylo nutné používat přesný hrot stylusu, stačilo se dotknout obrazovky v místě zobrazeného tlačítka. Telefon však stále obsahoval i hardwarová tlačítka. Hlavní, tzv. *Home button*, sloužilo k rychlému návratu na hlavní obrazovku. Ostatní pak umožnily změnu hlasitosti a uzamčení telefonu. Zbytek funkcí bylo ovládáno dotykem na základě kontextu aktuálně používané aplikace. Tento koncept nám z pohledu dnešní doby připadá běžný, v roce 2007 se však jednalo o nevídaný průlom.

iPhone poháněl operační systém iOS (dříve označován jako OS X – s OS X sdílel UNIXový základ). Tento operační systém v době spuštění nepodporoval aplikace ani funkci „copy and paste“. Tyto funkce byly přidány později.

Amazon Kindle

Myšlenka elektronického papíru se objevila poměrně brzo. V roce 1997 vzniká společnost E Ink Corporation, která si dává za cíl vytvořit zobrazovací display, který nepotřebuje umělé podsvícení.



Obrázek 18: Amazon kindle (2. gen) vs. Amazon Kindle Paperwhite 4

V roce 2007 firma Amazon uvedla na trh první generaci Amazon Kindle – nejpopulárnější čtečky elektronických knih. U prvních generacích Amazon Kindle bylo ovládání zprostředkováno pomoci klávesnice a tlačítek. Později se objevují verze s kapacitní vrstvou, které umožňují podobný styl ovládání jako dotykové telefony.

Elektronický papír však přináší mnohá omezení. Jedná se především o jeho pomalou rychlost překreslení a absence barev. Tyto limitace bylo nutné při vytváření uživatelského rozhraní překonat.

Apple iPad

> *„Strategy is really simple. What we want to do is we want to put an incredibly great computer in a book that you can carry around with you and learn how to use in 20 minutes ... and we really want to do it with a radio link in it so you don't have to hook up to anything and you're in communication with all of these larger databases and other computers."*
>
> - Steve Jobs, *1983*

V roce 2010 přichází Apple s dalším revolučním produktem – Apple iPad. iPad byl na rozdíl od iPhonu větší a výkonnější. Poukázal tím na možnou budoucnost osobních počítačů. Stejně jako iPhone, iPad obsahuje kapacitní dotykovou vrstvu – je tedy možné jej ovládat dotykem prstu. Během následujících let se těšil Apple iPad mimořádné oblibě mezi zákazníky. V roce 2015 Apple vydal iPad Pro umožnující používat nový typ uživatelského vstupu – Apple Pencil. Jedná se o chytrý stylus, který umožňuje ergonomické psaní a kreslení.

Prozatímní vývoj naznačuje, že v blízké budoucnosti právě tablety nahradí osobní počítače pro většinu běžných uživatelů.

V roce 2019 Apple oznámil iPadOS, které na iPadu nahradilo základní iOS. iPadOS přineslo kriticky důležité funkce jako je podpora práce s externím diskem, plnohodnotný prohlížeč internetu, podporu externí myši a mnoho dalšího.

Kinect

Zařízení podobné webové kameře vyvíjené společností Microsoft pro konzole Xbox 360/One. Přineslo ovládání her pohybem celého těla případně hlasovými příkazy.

Natural-language user interface (NLUI)

Využívá rozpoznání přirozené řeči ke komunikaci mezi uživatelem a počítačem. Popularita tohoto uživatelského rozhraní vzrostla s příchodem osobních asistentů Apple Siri (2011), Google Assistant (2016), Amazon Alexa (2014) atd.

Uživatelské rozhraní založené na rozpoznávání přirozeného jazyka má velký potenciál v jednoduchosti ovládání. Zatím je však problém správně a rychle porozumět příkazům od uživatele. Další limitací je absence méně rozšířených jazyků (například český jazyk).

I přes tyto problémy má tento přístup velkou perspektivu, a to především mezi běžnými uživateli osobních počítačů. Především v USA se staly velmi populární.

Obrázek 19: Apple iPad první generace vs. iPad Pro 2018.

Chytré hodinky

Především kvůli jejich malé velikosti bylo nutné věnovat větší pozornost uživatelskému rozhraní. Display je natolik malý, že dotyk prstu většinu plochy obrazovky překryje. Proto je nutné, především pro navigaci, využívat dalších ovládacích prvků.

V případě návrhu uživatelského rozhraní chytrých hodinek bylo nutné pracovat s již existujícím mentálním modelem (co to znamená uvidíme později). Příkladem je hodinová korunka původně sloužící k nastavování času[10]. U chytrých hodinek se s korunkou pracuje podobným způsobem, slouží však k navigaci či jiným účelům.

[10] Dříve navíc k natahování pérka hodinového strojku

Oculus Rift

V roce 2012 vzniká projekt na serveru Kickstarter pro zařízení s názvem Oculus Rift. Brýle a sada ovladačů sloužící ke zprostředkování

virtuální reality (VR). Brýle obsahují pohybové senzory, sluchátka a dvojici displayů. V roce 2014 firmu stojící za Oculus Rift kupuje firma Facebook.

Dnes patří virtuální realita k běžně dostupným rozšířením pro herní konzole. Komerčně dostupnější a úspěšnější se jeví tzv. rozšířená realita, která umožňuje sledovat okolí skrze obrazovku mobilního zařízení, které toto okolí určitým způsobem modifikuje, případně doplňuje.

Playstation 4 - DualShock 4

Sony Interactive Entertainment přichází v roce 2013 se čtvrtou generací herní konzole Playstation. Z uživatelského hlediska nás nejvíce zajímá herní ovladač – DualShock 4. Poprvé v historii Playstation konzole se objevuje dotykový panel, podobný touchpadu z notebooků. Touchpad mohou uživatelé využít například k psaní na virtuální klávesnici. Ovladač rovněž obsahuje 3-osý gyroskop. Ovládání pohybem však již dříve proslavila herní konzole Nintento Wii.

Obrázek 20: Ovladač DualShock 4.

Haptic feedback

Technologie, která dokáže skrze přesné vibrace simulovat dotyk. Hojně využíváná nejenom firmou Apple. Apple Macbook obsahuje touchpad, u kterého má uživatel při kliknutí pocit, že dochází k jeho prohnutí, touchpad je však zcela rovný. Dalším příkladem mohou být chytré hodinky Apple Watch 4, které vytváří iluzi mechanické (krokové) korunky hodinek.

Design a psychologie

Proč by nás měl design vůbec zajímat? Jaký dopad může mít špatný design? Tato kapitola je volný přepis základních myšlenek Donalda A. Normana z jeho knihy „Design of everyday things“[11].

[11] Donald A. Norman. *The Design of Everyday Things*. Basic Books, revised and expanded edition edition, 2013

Design se zabývá interakcí mezi lidmi a věcmi, tím jak věci fungují a jak jsou ovládány. Pokud je design uchopen správně, výsledkem je produkt, který je příjemný a intuitivní. V opačném případě se velice často dostavuje zklamání a frustrace.

Veškerá pravidla uvedená v následující kapitola jsou aplikovatelná na jakoukoli formu designu. Ať už se jedná o design uživatelského rozhraní, nebo fyzické věci – priority jsou stejné. Častou příčinou špatného designu jsou lidé z technologických odvětví, kteří se k němu snaží přistoupit intuitivně. Tito lidé jsou však limitování svou omezenou/naivní znalostí lidského chování. Všichni máme pocit, že intuitivně rozumíme tomu, jak se lidé chovají. Lidé z technologických odvětví přemýšlí logicky, to však neznamená, že všichni uživatelé produktu přemýšlí stejným způsobem. Mnohdy stačí sledovat uživatele zápasícího s produktem, kdo je v ten moment na vině? Uživatel? Designer? Zkušenost ukazuje, že je možné design navrhnout takovým způsobem, aby k podobným situacím docházelo minimálně.

> *„Lidská mysl je mimořádně bohatě vybavená k tomu, aby zkoumala svět a objevovala v něm smysl a řád.“*
>
> - Donald A. Norman,

Dobrý design přináší vodítka, špatný opak. Z tohoto důvodu je nutné se nejprve věnovat poznatkům z kognitivní psychologie, které tvoří pevný základ každého kvalitního designu.

Výzva designu v informatice

Správný design vyžaduje spolupráci několika disciplín. Nedostačuje, aby byla aplikace funkční a stabilní, stejně tak nestačí, aby byla pouze

graficky působivá. V komerční sféře častý jev – kvalitní design může být drahý a zdlouhavý, znovu však připomeneme citát:

> *„If you think good design is expensive, you should look at the cost of bad design.“*
>
> - Dr. Ralf Speth, *Chief Executive Officer, Jaguar Land Rover*

V kontextu informatiky je často složité vysvětlit důležitost správného designu. Většina lidi s technickým vzděláním preferuje (technickou) funkčnost a logiku nad estetikou – tento jev je viditelný na většině open-source projektů. Z toho důvodu je důležité mít alespoň okrajové znalosti ze světa designu a psychologie. Nesmíme zapomínat, že cílem mnoha aplikací je lidem ulehčit a zpříjemnit život, mnoho produktů však dělá pravý opak.

Ze studie[12] vyplývá, že uživatelé jsou obecně tolerantnější pokud je design esteticky přívětivý.

[12] Nielsen Norman Group. The aesthetic-usability effect. https://www.nngroup.com/articles/aesthetic-usability-effect/

> *„Users are more tolerant of minor usability issues when they find an interface visually appealing. This aesthetic-usability effect can mask UI problems and can prevent issue discovery during usability testing. Identify instances of the aesthetic-usability effect in your user research by watching what your users do, as well as listening to what they say.“*
>
> - Kate Moran, *Nielsen Norman Group*

Human-centered design (HCD)

Jedná se o přístup, který prioritizuje požadavky a nároky uživatele. Správný design umožňuje komunikaci uživatele s zařízením. Komunikace je důležitá především v momentech, kdy se věci nedaří a uživatel má problémy dosáhnout stanoveného cíle. Z toho důvodu je nutné, aby se designeři soustředili na situace, ve kterých se stalo něco špatného než na situace, kdy jde vše hladce.

Jakmile uživatel začne pracovat s programem, jeho hlavním úkolem je přijít na způsob, jak program použít. To obnáší zjistit, jaké funkce program obsahuje a jak se používají – to je možné díky *viditelnosti*. Viditelnost vzniká kombinací šesti psychologických principů: *afordance*, *signifier*, *mapování*, *zpětná vazba*, *omezení*, a (nejpodstatnější) *mentální model*.

Obrázek 21: Konvička na šálky na čaj.

Afordance

Pojem který zavedl James J. Gibson. Vyjadřuje vztah mezi předmětem a organismem, nikoli charakter samotného objektu. Obsahuje možné způsoby interakce.

Uveďme počítačovou myš jako příklad. Obsahuje dvě tlačítka, které evokují možnost stisku. Při pohnutí uvidíme zpětnou vazbu ve formě pohybujícího se kurzoru. Pochopíme tedy, že je zamýšleno s myší pohybovat. Afordance již neřeší kam a kdy budeme klikat, případně jak rychle budeme s myší pohybovat. K tomuto účelu slouží signifier. Zkuste se výše uvedeným způsobem zamyslet nad objekty na obrázku 21. Jakým způsobem s vámi komunikují?

Signifier

Jedná se o signály – nápisy, značky a ikony umístěné okolo nás. Například nápis „TAM“ na dveřích, nebo ikonka diskety na tlačítku pro uložení souboru. Explicitně vyjadřují jaké akce jsou dostupné a jak mají být využity. V designu uživatelského rozhraní je signifier díky

své přímé komunikaci důležitější než afordance.

Ze studie[13] vyplývá, že v případě *flat designu* uživatelé strávili průměrně o 22% více času studováním zobrazené webové stránky. Flat design odstraňuje počet jednoznačných signifierů, které dříve ulehčovaly identifikaci tlačítek a jiných důležitých ovládacích prvků. Z průzkumu vyplynulo, že pouhé použití stínů, nebo barevného přechodu ve výplni tlačítka tuto situace zlepšuje (obr. 22). V oblastech, kde uživatelé tlačítko již očekávají, nemá rozdílný vzhled takový vliv jaký bychom čekali.

[13] Nielsen Norman Group. Flat ui elements attract less attention and cause uncertainty. https://www.nngroup.com/articles/flat-ui-less-attention-cause-uncertainty/

Obrázek 22: Demonstrace signifieru na případu tlačítka.

Signifier může být účelový ale i neúčelový. V některých případech je to neškodné (ba dokonce prospěšné), například záložka knihy. Její originální účel byl označit, kde čtenář naposled četl. Originálním účelem nebylo poskytnout čtenáři možnost, jak sledovat jeho pokrok v čtení knihy. Čtenáři však tuto neplánovanou funkčnost ocenili. Extrémním případem by mohlo být označit tlačítko „smazat“ zelenou barvou a ikonou diskety. V tomto případě je pro uživatele nemožné odhadnout funkci tlačítka správně a bude nucen funkci nejprve vyzkoušet.

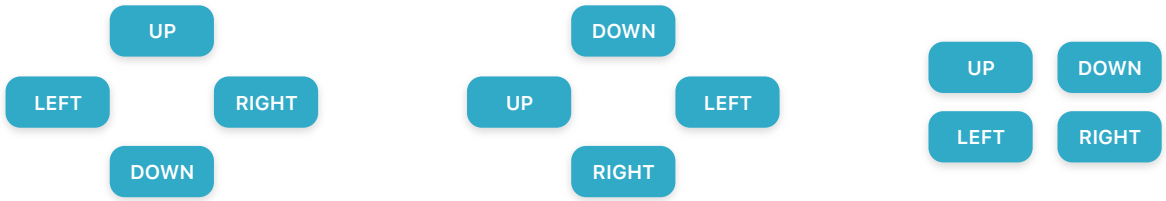
Mapování

Jedná se o vztah mezi objekty dvou množin. Představme si osvětlení na stropě, mapování mezi světly a vypínači určuje, který vypínač ovládá jaké světlo.

Mapovaní je z pohledu designu a UI důležitý koncept. Pojednává především o rozmístění ovládacích prvků, takzvaný *layout*. Pakliže je

rozmístění prvků vhodné, používání zařízení je intuitivní a jednoduché.

Takzvané *přirozené* mapování využívá reálných analogii a kulturních standardů. Příkladem zde může být rozmístění ovládacích prvku v liště nástrojů, od nejdůležitějších po méně důležité, a to zleva doprava. Toto rozmístění nám připadá intuitivní, protože jsme zvyklí číst zleva doprava. Jak víme, tento zvyk pochází ze západní kultury a nejedná se tedy o všeobecně platnou pravdu.



Obrázek 23: Demonstrace správného a špatného mapování.

Dalším důležitým poznatkem je seskupování souvisejících ovládacích prvků. Tento postřeh vychází z Gestaltismusu (tvarová psychologie) – lidé mají tendenci hledat souvislost mezi sousedícími objekty. Příklady správného a špatného mapování jsou demonstrovány na obrázku 23. Poznáte které je nejlepší a proč?

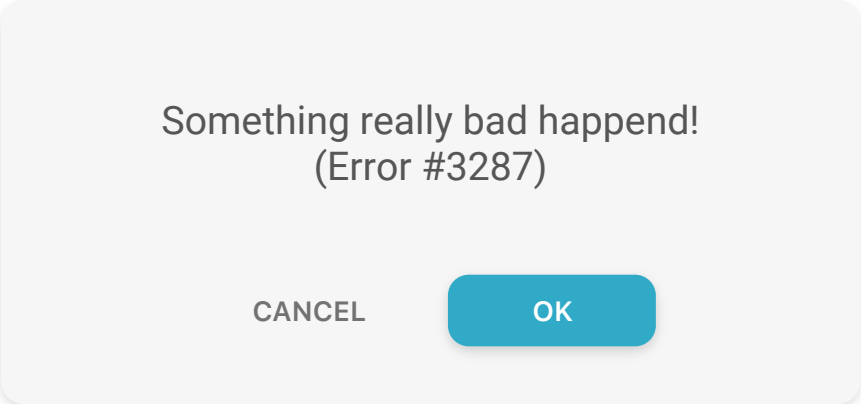
Zpětná vazba

Zpětná vazba je komunikace výsledků akce. Kdykoli se snažíte na dotykovém telefonu stisknout tlačítko a nevíte, zda jste jej úspěšně aktivovali nebo ne, je na vině špatná zpětná vazba. Přitom, stačí aby vás aplikace upozornila, že spuštěná operace probíhá a vy máte čekat na výsledek[14]. Lidský nervový systém je vybaven mnoha mechanizmy poskytující zpětnou vazbu.

[14] V opačném případě víte, že aktivace nebyla úspěšná.

Je důležité, aby byl uživatel informován včas. V případě stisku tlačítka stačí zpomalení zpětné vazby o několik sekund a netrpělivý uživatel již stihl tlačítko stisknout hned několikrát.

Dalším podstatným aspektem zpětné vazby je její informativnost. Změna barvy tlačítka, nebo zobrazení čísla chybového hlášení dá uživateli najevo, že se něco stalo, neví však co a proč. Taková zpětná vazba je lepší než žádná, ale rozhodně se nejedná o dostatečné řešení. Příklad nedostatečné zpětné vazby je možné vidět na obrázku 24.



Obrázek 24: Příklad nedostatečné zpětné vazby.

Zpětná vazba by rovněž neměla být příliš častá. Uživatel poté ztrácí koncentraci na jeho cíl a jeho pokus končí frustrací. V horším případě začne varování ignorovat úplně, což v extrémních případech může vést ke katastrofě. Z toho plyne, že kritické signály musí být prezentovány prioritně a důrazně, aby nedošlo k jejich ignoraci nebo přehlédnutí (např. ukazatel baterie na obrázku 25).

Obrázek 25: Příklad s ukazatelem baterie.

Důležité je vědět, že chybovat je lidské. Je-li někde možné udělat chybu, některý uživatel ji udělá. Designer musí s tímto faktem počítat, a když je potřeba podat uživateli pomocnou ruku, když je potřeba.

Dobrý design se rovněž snaží vyhnout *falešné kauzalitě*. Kauzalita neboli *příčinnost*, znamená, že se nic neděje bez nějaké příčiny. Problém nastává v případě falešné kauzality, kdy má uživatel pocit, že něco způsobil, přitom se jedná o náhodnou reakci systému. Příkladem může být chování starších textových editorů, které trpěly pomalou odezvou. Pokud uživatel opakovaně mačkal klávesu „Delete“ mohlo se stát, že textový editor nereagovat dostatečně rychle a uživatel nechtěně smazal více textu než bylo žádoucí.

Omezení

Spousta věcí na první pohled slouží pouze na omezený počet úkonů. Jestliže má objekt nebo aplikace vhodně zvolená omezení, usnadňuje to uživateli jeho použití. Uvádíme čtyři druhy omezení: *fyzická*, *kulturní*, *sémantická* a *logická*.

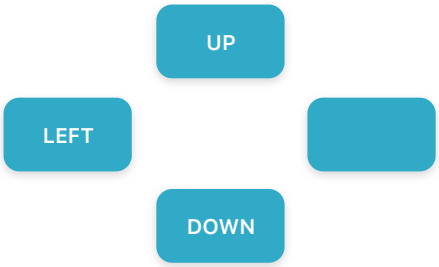
Fyzická omezení Limitují (fyzicky) možné operace. Do menšího otvoru na nůžkách nelze vložit více prstů, počet způsobů jak tedy nůžky používat je limitován. Baterie jsou navrženy tak, aby většina zařízení nešla přepólovat nevhodným vložením baterie. V případě USB kabelu bylo toto omezení spíše obtěžující. V další generaci USB-C kabelu bylo toto omezení odstraněno a kabel lze zapojit v libovolné orientaci.

Kulturní omezení Určují množinu možných akcí v rámci sociální situace. Příkladem může být výtah, pokud si zkusítě stoupnout zády ke dveřím a uvidíte, jaký pocit vyvoláte v sobě a ostatních. Již dříve zmíněný příklad, v naši kultuře jsme zvyklí číst zleva doprava, proto také očekáváme, že aplikace budou tento kulturní zvyk respektovat. Problémem kulturního omezení je jeho nestálost – kultura je neustále ve vývoji.

Sémantická omezení Sledují význam dané situace pro redukování množiny možných akcí. Není nutné zobrazovat ovládací panel pro editaci obrázku, když uživatel pracuje pouze s textem. Stejně jako kulturní omezení i sémantické se neustále vyvíjí. To jak chápeme určité situace se během času mění. Jak například budou vypadat exteriéry automobilu až budou auta zcela samořídící? Budou mít červená brzdová světla nějaký význam? Ostatní samořídící auta pravděpodobně budou komunikovat jinak než světelným signálem. Sémantika červených světel se tedy pozvolna vytratí. V kontextu UI se můžeme zamyslet nad sémantikou barev a tvarů. Není známo, zda budeme tvar tlačítka chápat stejně i v budoucnu. Dříve byla tlačítka plastická, dnes uživatelům nedělá problém považovat za tlačítko pouhý barevný text[15].

[15] Především s příchodem takzvaného „flat designu“

Logická omezení jsou nejlépe pozorovatelná v přirozeném mapování. Mezi vypínači a světly na stropě je jasná logická souvislost. Nejedná se o fyzické ani kulturní omezení. Dalším příkladem může být situace, kdy uživatel zná celou množinu akcí, nezná však celou množinu ovládacích prvků. V tomto případě není pro uživatele těžké odvodit funkce zbylých ovládacích prvků. Tato situace je viditelná na obrázku 26.



Obrázek 26: Demonstrace logického omezení.

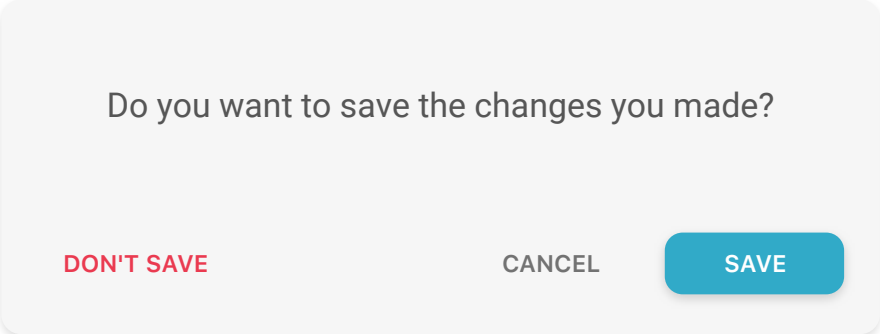
Na omezení se můžeme podívat i z dalšího úhlu pohledu.

Interlocks Vynucení určitého pořadí operací. V kontextu každodenních věcí lze zmínit například mikrovlnou troubu, kterou nelze otevřít pakliže nedošlo k jejímu vypnutí. Z hlediska uživatelského rozhraní se může jednat o vynucení vyplnění určitých údajů ve formuláři před jeho odesláním.

Interlock může mít i další formu – „pojistka mrtvého muže“. Je běžné, že strojvedoucí musí jednou za určitý čas zmáčknout tlačítko, aby bylo jasné, že je při smyslech a vlak plně ovládá. Toto využití omezení se týká převážně bezpečnosti.

Lock-ins Udržují akci v chodu tak aby ji nemohl nikdo přerušit. Příkladem z uživatelského rozhraní je neumožnění vypnout program bez uložení rozpracovaného projektu (obr. 27).

Lockouts Opak lock-ins, zabraňují tomu, aby uživatel vstoupil kam nemá. Převážně používány pro zvýšení bezpečnosti. Příkladem můžou být dětské pojistky, pojistka na hasícím přístroji, vyžadování hesla pro žměnu důležitých nastavení a další.



Obrázek 27: Příklad využití lock-in.

Mentální (konceptuální) model

Je vysvětlení (často velice zjednodušené) toho jak něco funguje. Není nutné aby toto chápání bylo správné nebo přesné, stačí aby bylo funkční. Soubory, složky a ostatní ikony, které vidíme v počítači slouží k vytvoření mentálního modelu. Víme, že se v počítači žádné složky nenacházejí, jedná se pouze o abstrakci ulehčující používání.

Mentální model vzniká zkušeností předávanou mezi lidmi, nebo v literatuře. Zkušeností rovněž vzniká takzvaná *intuitivní expertíza*. Jedná se o situaci, kdy má člověk vnitřní pocit jakým způsobem jednat, ale neví racionálně proč[16]. Intuice selhává v momentě, kdy je problém moc složitý. Lidé začínají jednat na základě emocí nikoli rácia a začínají dělat zdánlivě nepochopitelné rozhodnutí.

[16] K tomuto racionálnímu pochopení dojde později

„Intuition is nothing more and nothing less than recognition."

- Herbert A. Simon, *Držitel Nobelovy ceny*

Příkladem mentálního modelu může být čtení e-mailů. Uživatel vidí e-mailového klienta a veškeré svoje zprávy, může nové e-maily přijímat či psát. Vše funguje, vypadá to jako jednotná věc – jednotný mentální model. Je však pravdou, že jednotlivé části se nachází na různých místech – uživatelské rozhraní je lokální, e-mailové zprávy se mohou nacházet na serverech po celém světě. Vše funguje dokud je uživatel

připojený k internetu. Jakmile dojde k odpojení, mentální model přestává fungovat, e-mail nejde přijímat ani odesílat, uživatel neví co se děje.

Mentální modely umožňují uživatelům předvídat důsledky jejich činů. Nejhorší situace vzniká, pokud si uživatel na základě špatného designu vytvoří špatný mentální model[17]. Přirozeností člověka je snaha o hledání vysvětlení, proto nelze zabránit vzniku mentálního modelu, můžeme se pouze snažit o správnost/vhodnost tohoto modelu.

[17] Klasická věta začínající slovy „To přece musí fungovat takto...“

Příkladem může být termostat topení. Uživatel přijde domů a potřebuje co nejrychleji zatopit. Na základě zjednodušeného mentálního modelu nastaví teplotu na termostatu na maximální. Při bližším zkoumání funkce termostatu však zjistíme, že toto nastavení vytápění nijak neurychlí, naopak může dojít pouze k plýtvání energie. Klasický termostat zná pouze dva stavy: topení zapnuto, topení vypnuto. Nastavení maximální teploty tedy neovlivní rychlost vytápění.

Myšlení rychlé a pomalé

Sekce založená na poznatcích ze stejnojmenné knihy [18]. Ve výše zmíněné knize autor popisuje dva systémy – *Systém 1* a *Systém 2*:

[18] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 1st edition edition, 2013

Systém 1 funguje automaticky a rychle, s malým nebo žádným úsilím a bez pocitu úmyslné kontroly.

Systém 2 přiděluje pozornost vědomým duševním činnostem, které ji vyžadují, včetně složitých výpočtů. Činnost Systému 2 bývá spojena se subjektivním prožitkem jednání, volby a soustředění.

Pokud provádíme již dobře známé akce kontrolu převezme Systém 1 a my ani často nevíme, jak konkrétně úkol provádíme. Proto je možné, že zkušený řidič zvládne řídit automobil a povídat si se spolujezdci – provádí dvě akce, které nejsou nikterak náročné. Množství těchto akcí je ale rovněž omezeno. Pokud by se řidič dostal do krizové situace, přestane okamžitě komunikovat a své plné soustředění věnuje vyřešení krizové situace.

Systém 2 je na druhou stranu kritický pro učení se nových věcí. Když uživatel používá aplikaci poprvé, Systém 2 bude aktivnější. Jak moc tento proces bude pro uživatele náročný, záleží rovněž na designu dané aplikace.

Systém 1 exceluje v rozpoznávání vzorů a aktivaci asociací. Aktivace asociací si většinu času nejsme ani vědomi. Převážně se jedná o provázanou sadu akcí, nikoli jednu samostatnou asociaci. Pokud tedy naše aplikace dodržuje obecné konvence, uživatel má možnost rozpoznat většinu ovládacích prvků intuitivně – za pomoci Systému 1[19]. Systém 1 je rovněž zodpovědný za většinu omylů, lidé mu totiž důvěřují více

[19] Z tohoto důvodu je důležité neměnit význam ovládacích prvků napříč aplikacemi

než Systému 2. Situace je vyhodnocena rychleji než Systém 2 stihne zasáhnout a situaci přehodnotit.

Ačkoli se jedná o popis dvou oddělených systémů, toto rozdělení je pouze ilustrační, reálně nejsou tyto systémy nijak fyziologicky rozděleny.

Kognice (kognitivní procesy – souhrn operací a pochodů, jejichž prostřednictvím si člověk uvědomuje a poznává svět i sebe samého) a emoce rovněž nemohou být rozděleny. Kognice se snaží pochopit svět, emoce tomuto poznání udávají hodnotu.

Konceptuální model lidského chování

V této sekci se budeme zabývat tím, jak lidé interagují s technologiemi. Mnohdy je jednoduché pochopit základy používání některé z technologií, co se ale stane když věci nejdou podle plánu? Jakým způsobem rozpoznáme, že věci nefungují?

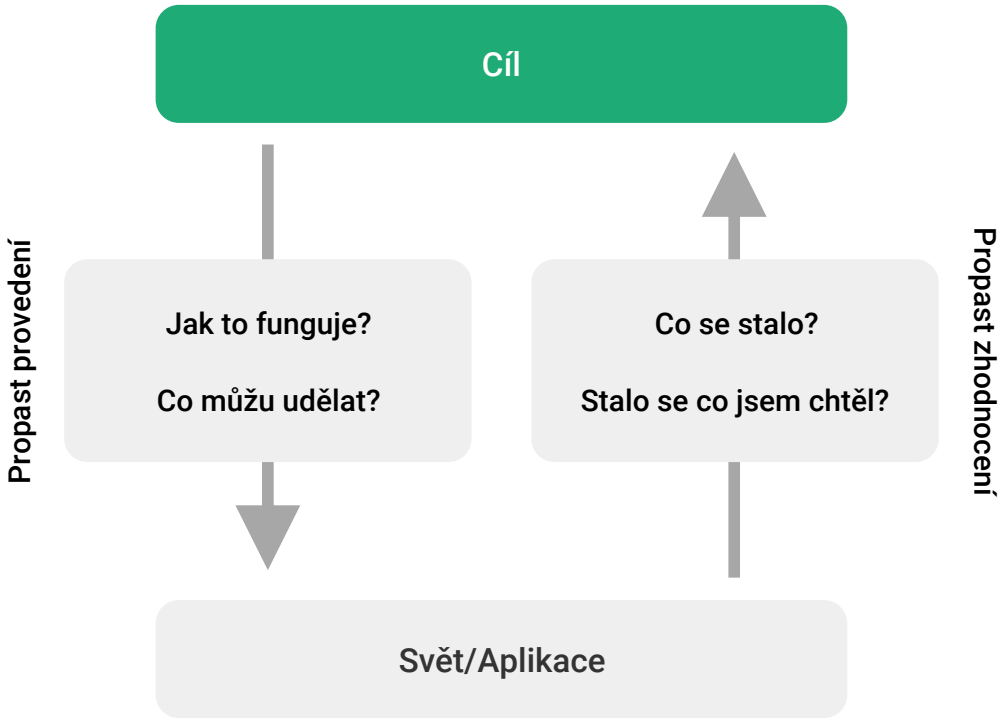
Pro ilustraci tohoto procesu využijeme analogii dvou propastí: *propast provedení* a *propast zhodnocení* (obr. 28). Propast provedení symbolizuje moment, kdy se uživatel snaží zjistit jak s daným objektem či aplikací pracovat. Propast zhodnocení symbolizuje stav, kdy uživatel vyhodnocuje v jakém stavu se věc či aplikace nachází, případně jaké další akce by měl podniknout. Velikost těchto propastí ovlivňuje kvalita designu. Propast provedení je zmenšena vhodným použitím signifierů, omezení, mapování a mentálního modelu. V případě propasti zhodnocení pomáhá kvalitní zpětná vazba a mentální model.

Provedení se skládá ze tří částí: plánování, specifikace a provedení. Zhodnocení pak z: pozorování, vyhodnocení a porovnání.

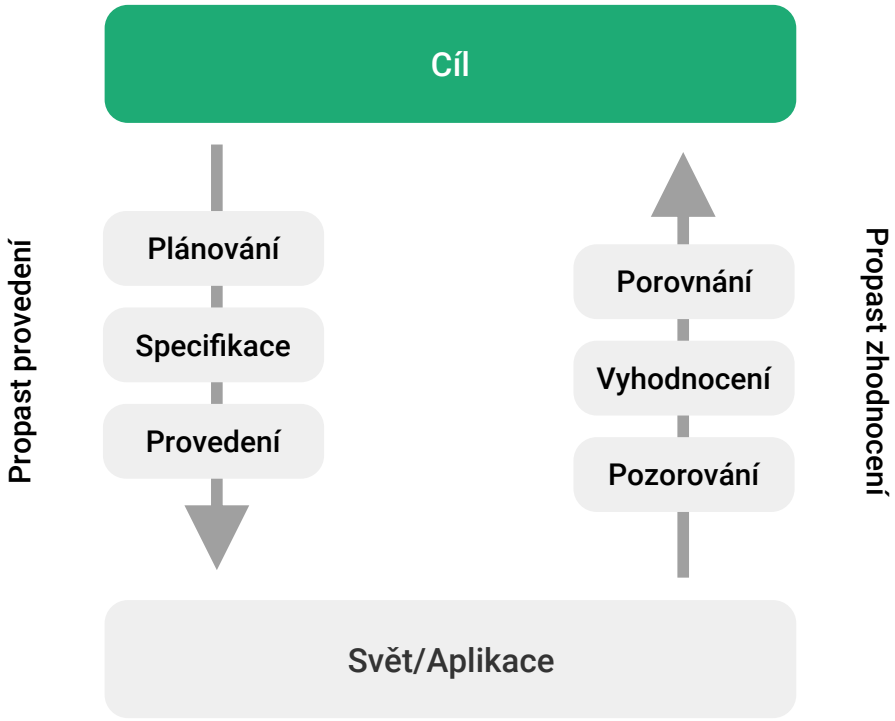
Sedm fází akce

Z výše uvedených poznatků získáváme cyklus *Sedm fází akce* (obr. 29):

1. **Cíl** (formování cíle)
2. **Plánování** (akcí)
3. **Specifikace** (sekvence akcí)
4. **Provedení** (sekvence akcí)
5. **Pozorování** (stavu světa/aplikace)
6. **Vyhodnocení** (vnímání)
7. **Porovnání** (výstupu s cílem)



Obrázek 28: Analogie dvou propastí.



Obrázek 29: Sedm fází akce.

Cyklus sedmi fází akce může začínat stanovením nového cíle, nebo může začít na základě externího podnětu. Uživatel je pak nucen podnět vyhodnotit a poté stanovit další cíl.

Definicí cíle se zabývá takzvaná *root cause analysis*. Definice cíle zkoumá pomoci opakovaného kladení otázky „Proč?“.

„People don't want to buy a quarter-inch drill, they want a quarter-inch hole.“

- Theodore Levitt, *Ekonom*

Výše zmíněný citát ilustruje často zapomínaný fakt. Uživatelé používají aplikace z nějakého důvodu – chtějí dosáhnout nějakého cíle – proto je důležité, aby aplikace uměly těmto poměrně vzdáleným požadavkům vyhovět. Z pozice vývojáře je nutné nezpomínat co jsou hlavní požadavky na vytvářenou aplikaci.

Úrovně zpracování

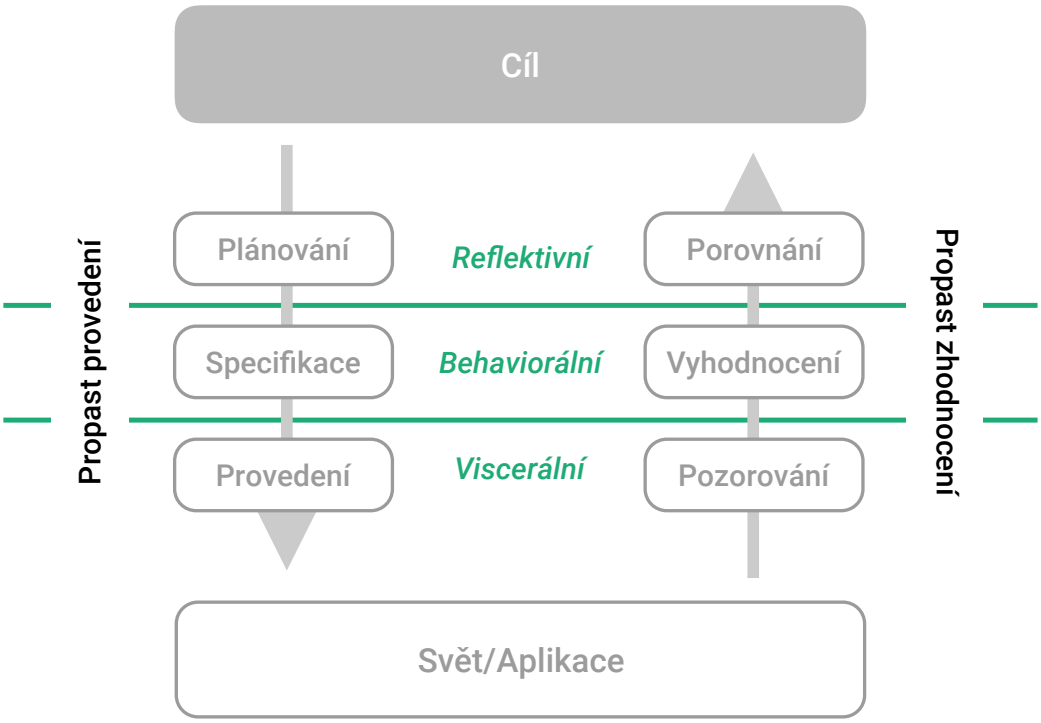
Aproximaci lidské kognice a emocí reprezentuje model úrovní zpracování: *viscerální*, *behaviorální*, *reflektivní*.

Viscerální úroveň Je úrovní základní, někdy označována jako „ještěrčí mozek“. Pro lidi je tato úroveň jednotná. Obsahuje základní instinkty a ochranné mechanismy. Odezva je rychlá a automatická.

Z pohledu designu tuto úroveň můžeme vnímat jako okamžitou odezvu uživatele, převážně pak ve formě kladného nebo negativního pocitu. Na této úrovni rozhoduje vzhled, dotek, přívětivost zvuku. Nejedná se o hodnocení použitelnosti, efektivity nebo pochopení toho k čemu aplikace slouží. Technicky smýšlející lidé mají tendenci touto úrovní opovrhovat a považovat ji za nedůležitou.

Behaviorální úroveň Naučené dovednosti, spuštěné na základě reakce na známou situaci. Akce a analýza je na této úrovní převážně podvědomá. V situaci, kdy provádíme dobře známou akci, se stačí soustředit na cíl a behaviorální úroveň se postará o zbytek.

Z pohledu designu je nejdůležitější aspekt behaviorální úrovně provázanost akcí s očekáváním. Informace v cyklu zpětné vazby hodnocení potvrzují nebo vyvracení očekávání, což vede ke spokojenosti nebo zklamání. Zpětná vazba je kritická pro práci s očekáváním uživatele.



Obrázek 30: Úrovně zpracování v kontextu semi fází akce.

Reflektivní úroveň Vědomé uvažování. Jedná se o úroveň, kde vzniká hluboké porozumění, uvažování a vědomé rozhodování. Reflektivní úroveň je v porovnání s viscerální a behaviorální pomalá a hluboká. Na této úrovni dochází k predikci výsledků budoucích akcí.

Z pohledu designu se jedná o nejdůležitější úroveň. Reflektivní reakce jsou součástí našich vzpomínek a uchováváme si je delší dobu než jednorázové reakce. Je to právě reflexe, která vede k doporučení nebo nedoporučení daného produktu. Reflektivní vzpomínky jsou často důležitější než realita.

Design se musí odehrávat na všech výše zmíněných vrstvách. Pokud schéma dvou propastí doplníme o tyto vrstvy získáme obrázek 30.

Lidská paměť

Pro pochopení určitých designových pravidel je vhodné porozumět tomu jak funguje lidská paměť. Po vzoru knížky[20] rozdělíme lidskou paměť na kontext *znalostí v hlavě* a *znalostí ve světě*.

[20] Donald A. Norman. *The Design of Everyday Things*. Basic Books, revised and expanded edition edition, 2013

V kontextu lidské paměti si lidé nejčastěji vybaví právě statické informace jako hesla, telefonní čísla, data narození a další. Tuto kategorii můžeme nazvat *znalostmi v hlavě*. V dnešní době není již nutné zapamatovat si všechny tyto informace – právě technologie nám přinesly možnosti bezpečného uložení těchto informací, které na první pohled ulehčují situaci strojům a komplikují situaci lidem.

Typy lidské paměti

Paměť rozdělujeme na paměť *senzorickou*, *krátkodobou* a *dlouhodobou*[21].

[21] Dennis L. Byrnes Wingfield, Arthur. *The psychology of human memory*. New York: Academic Press, 1981

Senzorická paměť Slouží k okamžitému zachycení informace z některého z lidských smyslů. Díky této informace získáme krátký čas navíc ke zpracování této informace. Při pozorování objektu jsme schopni i po jeho zmizení krátkou dobu objekt „vidět“, jedná se pocit kdy nám něco „zůstalo před očima“. Senzorická paměť se vyznačuje schopností uchovat velký objem informací na krátkou dobu.

Krátkodobá paměť Obsah krátkodobé paměti je uchován na delší dobu (cca 15-20s) než v případě senzorické paměti. Příkladem je schopnost zapamatovat si telefonní číslo po jeho detailním přečtení. Dobu, kterou jsme schopni telefonní číslo „udržet v hlavě“ můžeme prodloužit opakováním. Zajímavým poznatkem je, v jaké formě je informace uložena. Krátkodobá paměť ukládá informace ve formě přirozeného jazyka. To znamená že telefonní číslo v krátkodobé paměti uložíme ve formě číslic, které jsme se (nejenom) pro tento účel kódování naučili.

Dlouhodobá paměť Slouží k finálnímu uložení informací v paměti. Dlouhodobé uložení vyžaduje zakódování pravého významu události. Často není uložený kompletní význam události, ale pouze střípky informace. Kdykoli je vzpomínka vyvolána dojde k sestavení významu z jednotlivých střípků[22]. Pokud si událost zapamatujeme v kontextu ostatních, dokážeme si ji zapamatovat déle. Samotné uložení a získání informací trvá déle (uložení cca 30 min). Ukládání probíhá jak záměrně (učení), tak samovolně (prožitek). Z praktického hlediska má dlouhodobá paměť neomezenou kapacitu.

[22] Z toho důvodu máme často zkreslené vzpomínky na události.

Hlavním problémem dlouhodobé paměti je její organizace. Jakým způsobem si můžeme cíleně vzpomenout[23]. Stále nevíme jak přesně je dlouhodobá paměť organizována. Získání uložené informace prav-

[23] Tento pocit často popisuje věta „Mám to na jazyku“.

děpodobně obsahuje systém porovnání vzorů a kontrolu konzistence dat[24].

[24] Kontrolu konzistence v praxi zažíváme, když si snažíme usilovně vzpomenout a stále se nám vyplavuje ta stejná myšlenka a mi víme, že je špatně

Typy uchovávaných informací

Druhým možným pohledem na znalosti v hlavě je rozdělení na dvě kategorie *svévolných* a *smysluplných* informací.

Svévolné informace Jedná se o mechanicky naučené věci bez hlubšího významu a vztahu k ostatním znalostem. Příkladem může být abeceda, jména lidí, klávesové zkratky[25], příkazy v příkazové řádce. Takové informace se nejlépe učíme opakováním tzv. *rote learning*. Jedná se však stále o učení velice pomalé a nespolehlivé. Především pak v situaci, kdy se věci nedaří podle plánu jsou takto naučené informace k ničemu.

[25] Zde místy smysl nalézt můžeme.

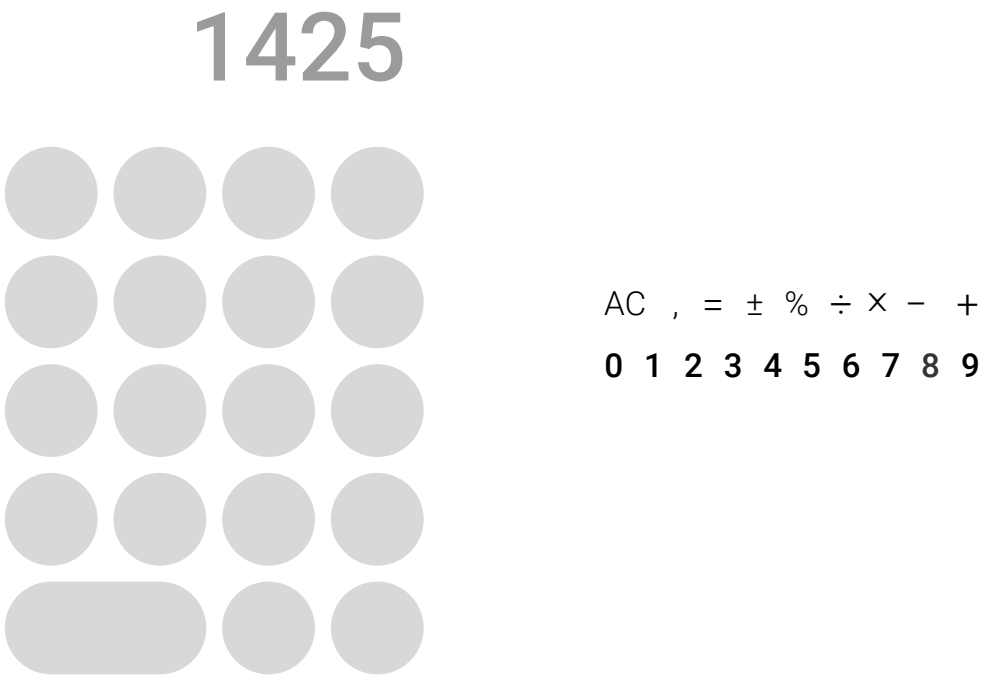
Smysluplné informace Informace založené na smysluplných souvislostech. Většina věcí ve světě má smysl a strukturu. Tento fakt významně usnadňuje učení. Z toho důvodu je dříve zmíněný konceptuální model důležitý – dává věcem význam, a tento význam si lidé jednoduše zapamatují.

Z pohledu designu je tedy nutné neklást velké nároky na uživatelovu paměť. Pokud uživatelské rozhraní bude používat smysluplnou strukturu, bude pro uživatele jednodušeji pochopitelné a zapamatovatelné. V ideálním případě by neměla být uživatelova paměť z hlediska UI zapotřebí. V tomto aspektu excelují GUI aplikace nad aplikacemi v příkazové řádce.

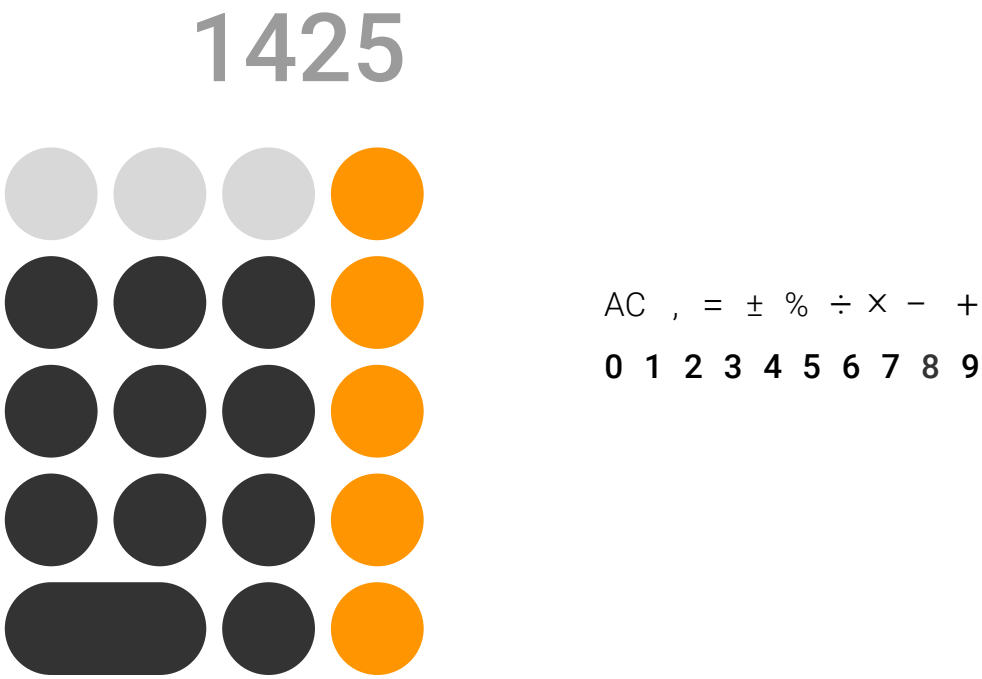
Znalosti ve světě

Častěji než si uvědomujeme používáme naše okolí jako prostředek paměti. Pokud je znalost potřebná k provedení operace snadno dostupná z našeho okolí, není nutné si informaci do paměti ukládat. Z pohledu znalostí ve světě se nejvíce projevují signifiers, omezení a přirozené mapování.

Situaci demonstrujeme na příkladu s kalkulačkou. Na obrázku 31 je zobrazené základní rozvržení kláves kalkulačky. Zkuste z paměti rozmístit operace na kalkulačku. Situace je poměrně ulehčena zobrazením rozložení a barev (obr. 32). Rozložení a barvy slouží rovněž jako uložená informace ve světě. Také nám v tomto úkolu pomůže výčet operací, které máme na kalkulačku umístit. Běžně tyto informace v paměti uložené nemáme, stačí nám pouze rámcová znalost.



Obrázek 31: Základní rozvržení kalkulačky na telefonu.

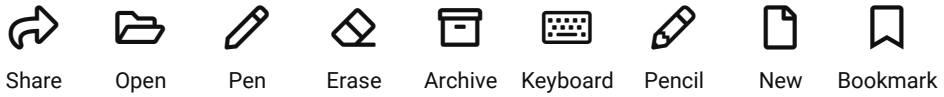


Obrázek 32: Základní barevné rozvržení kalkulačky.

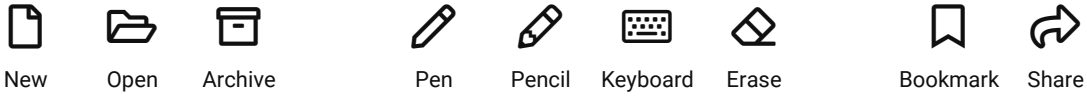
AC , = ± % ÷ × − +

0 1 2 3 4 5 6 7 8 9

Obrázek 33: Správné řešení úlohy s rozvržením kalkulačky.



Obrázek 34: Základní rozvržení kalkulačky na telefonu.



Obrázek 35: Správné řešení úlohy s rozvržením kalkulačky.

Millerovo magické číslo

V roce 1994 George A. Miller publikuje v článku[26] poznatek o „magickém“ čísle sedm plus minus dva. Pojednává o schopnosti krátkodobé paměti udržet na dobu cca 20 sekund cca 7 (plus minus 2 pojmy). V praxi se ukazuje nejvhodnější rozmezí od tří do pěti pojmů. Dobu uchování těchto pojmů lze mnemotechnickými pomůckami prodloužit.

Z pohledu designu uživatelského rozhraní nemůžeme tedy očekávat, že si uživatel zapamatuje více než 5 pojmů. Vhodné je rovněž organizovat objekty do skupin po třech až pěti objektech. Poté je větší šance, že si uživatelé zapamatují, kde se ovládací prvky nacházejí.

Pravidlo lze ilustrovat na příkladu v obrázku 96, kde jsou ovládací rozmístěny a v jedné velké skupině v náhodném pořadí. Pro uživatele je velice těžké si zapamatovat, jaké ovládací prvky jsou k dispozici a kde přesně je najde. Na obrázku 35 jsou ovládací prvky rozděleny již správně – do menších logických skupin po 3-5 prvcích.

[26] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1994

Zákony uživatelského rozhraní

Následující sekce je volným překladem a doplněním skvělého webu Laws of UX[27]. Některé z pravidel již byla zmíněna. Tato sekce by měla sloužit jako zakončení kapitoly a jednoduché připomenutí klíčových poznatků z psychologie.

[27] Jon Yablonski. Laws of ux. https://lawsofux.com

Efekt estetické použitelnosti

Klíčová myšlenka Esteticky příjemný design může učinit uživatele více tolerantním k drobným nedostatkům v použitelnosti. Esteticky příjemný design tedy může maskovat problémy s použitelností a brání projevu těchto problému během testování.

Původ Tento jev byl poprvé zkoumán v odvětví human-computer interaction v roce 1995. Masaaki Kurosu a Kaori Kashimura z Hitachi Design Center testovali 26 variant UI bankomatů. Během testů objevili silnou korelaci mezi hodnocením estetického designu s dojmem uživatelsky přívětivého designu.

Doherty Threshold

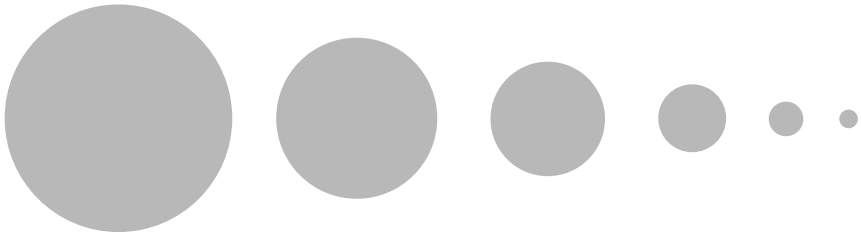
Klíčová myšlenka Produktivita se zvětšuje, když počítač a jeho uživatel reagují rychle (v kratší době než 400 ms). Pokud tedy zajistíme rychlou odezvu systému, dokážeme udržet pozornost uživatele a zvýšit jeho produktivitu.

Původ V roce 1982 Walter J. Doherty a Ahrvind J. Thadani publikovali v IBM Systems Journal vědecký článek, kde tvrdil, že ideální doba odezvy PC je méně než 400 ms. Předchozí standard činil 2 000 ms. Pokud počítač zvládal reagovat na příkazy od uživatele do 400 ms, byl pro uživatele atraktivnější.

Fittsův zákon

Klíčová myšlenka Čas potřebný k zaměření cíle je funkce vzdálenosti a velikosti cíle. Objekty, které mají být jednoduše zaměřitelné by měly být věší a blíže uživateli. Toto pravidlo se vztahuje především na tlačítka a elementy a ostatní interakční prvky.

Původ V roce 1954 psycholog Paul Fitts zkoumal lidský motorický systém. Zjistil, že čas potřebný k přesunutí k cíli je závislý na vzdálenosti a velikosti objektu. Na základě tohoto pravidla rychlé pohyby a malé objekty jako cíle způsobují větší chybovost. Na základě tohoto



Obrázek 36: Fittsův zákon.

pravidla jsou hlavní tlačítka větší (především na dotykových zařízeních).

Hickův zákon

Klíčová myšlenka Čas potřebný k učinění rozhodnutí se zvětšuje s počtem a komplexností rozhodnutí. Více možností tedy vede k delšímu přemýšlení a pomalejšímu rozhodování. Komplexnější operace bychom měli uživateli rozdělit do menších kroků. Situaci můžeme zjednodušit zvýrazněním doporučených možností.

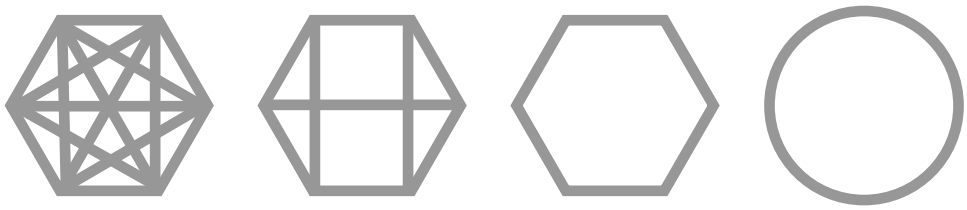
Původ Zákon je pojmenován po dvojci psychologů William Edmund Hick a Ray Hyman. V roce 1952 se rozhodli zkoumat vztah mezi počtem stimulů a délkou reakční doby. Jak můžeme očekávat, více stimulů vedlo k prodloužení reakční doby. Uživatelé, kteří jsou zahlceni volbami musí vynaložit více energie na jejich vyhodnocení a rozhodnutí.

Jakobův zákon

Klíčová myšlenka Uživatelé tráví většinu času na ostatních webových stránkách/programech. Nový produkt tedy bude srozumitelnější pokud bude využívat stejné designové vzory. Jinými slovy, jsme schopni ulehčit proces učení uživatele tím, že využijeme již dobře známé ovládací prvky a designové vzory.

Původ Autorem zákonu je Jakob Nielsen, který spolu s Dr. Donald A. Norman založili Nielsen Norman Group.

Obrázek 37: Hickův zákon.

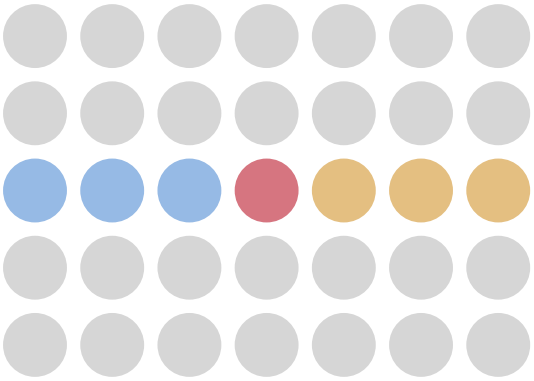


Obrázek 38: Zákon výstižnosti.

Zákon výstižnosti

Klíčová myšlenka Lidé mají tendenci vidět a interpretovat složitější tvary tím nejjednodušším možným způsobem. Je tomu tak, protože tento přístup vyžaduje nejméně kognitivní námahy. Lidské oko tedy rádo hledá jednoduchost a řád v komplexních tvarech, předchází tak zahlcení informacemi.

Původ V roce 1910 psycholog Max Wertheimer pozoroval řadu světel, které se rozsvítily na železničním přejezdu. Bylo to podobné světlům obklopující divadlo. Tím, že se světla postupně rozžínají a zhasínají pro pozorovatele vytváří efekt pohybujícího se světla.



Obrázek 39: Millerův zákon.

Millerův zákon

Klíčová myšlenka Průměrný člověk je schopný zapamatovat si pouze 7 (plus minus 2) pojmů v jeho pracovní paměti. Shlukování je tedy efektivní metoda k prezentování skupin objektů. Je dobré organizovat obsah do skupin po pěti až devíti objektech.

Původ V roce 1994 George A. Miller publikuje v článku[28] poznatek o „magickém" čísle sedm plus minus dva. Pojednává o schopnosti krátkodobé paměti udržet na dobu cca 20 sekund 7 (plus minus 2 pojmy). V praxi se ukazuje nejvhodnější rozmezí od tří do pěti pojmů. Dobu uchování těchto pojmů lze mnemotechnickými pomůckami prodloužit.

[28] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1994

Occamova břitva

Klíčová myšlenka Pokud pro nějaký jev existuje vícero vysvětlení, je dobré upřednostňovat to nejméně komplikované. V designu tedy analyzujte každý element a odeberte co nejvíce je jen možné bez nutnosti vytvářet kompromis na celkové funkcionalitě.

Původ Occamova břitva je princip řešení problémů, který říká, že v případě, že existuje více hypotetických řešení problému, mělo by být vybráno to, které vyžaduje nejméně předpokladů.

Princip Pareta

Klíčová myšlenka Princip Pareta tvrdí, že u mnoha událostí přibližně 80% efektu přichází z 20% příčin. Je tedy vhodné věnovat největší úsilí oblastem, které přinesou největší benefit pro většinu uživatelů.

Původ Vilfredo Pareto ekonom, který vypozoroval, že 80% italské půdy je vlastněno 20% italské populace. I když to působí nejasně, princip 80/20 může poskytnout věrohodnou analýzu nesourodých systémů, včetně strategie pro tvorbu UI.

Parkinsonův zákon

Klíčová myšlenka Jakýkoli úkol bude prováděn tak dlouho, dokud nebude vyčerpán veškerý dostupný čas.

Původ Zákon vyřčen Cyril Northcote Parkinsonem jako část první věty v humorné eseji publikované v The Economist roku 1955. Později byla znovu otištěna v rámci knihy Parkinson's Law: The Pursuit of Progress (London, John Murray, 1958). Tento zákon odvodil z bohatých zkušeností v britské státní službě.

Postelův zákon

Klíčová myšlenka Buďte liberální v tom, co přijímáte, a konzervativní v tom, co posíláte. Je důležité být empatický, flexibilní a tolerantní vůči jakémukoli množství akci, které může uživatel udělat. To obnáší práce se vstupem od uživatele, jeho validaci, definování omezení a poskytování zpětné vazby.

Původ Postelův zákon formuloval Jon Postel, časný průkopník Internetu. Zákon je návrhovou směrnicí pro software, specificky pak týkající se TCP a sítí.

Efekt pozice v sekvenci

Klíčová myšlenka Uživatelé mají tendenci si nejlépe pamatovat první a poslední položky v sekvenci. Z toho plyne, že je dobré umisťovat nejméně podstatné prvky do středu sekvence. Umístění klíčových prvků navigace na levou/pravou stranu může zvýšit pozornost uživatele.

Obrázek 40: Efekt pozice v sekvenci.

Původ Pojem který zavedl Herman Ebbinghaus popisuje jak pozice objektu v sekvenci ovlivňuje přesnost vybavení si objektu. Tento princip byl úspěšně adaptován firmami jako Nike, Apple a Electronic Arts.

Teslerův zákon

Klíčová myšlenka Teslerův zákon, známý také jako zákon o zachování složitosti, uvádí, že u každého systému existuje určitá složitost, kterou nelze redukovat.

Původ Během práce pro Xerox PARC okolo roku 1980 si Larry Tesler uvědomil, že způsob, kterým uživatelé interagují s aplikacemi je stejně důležitý jako aplikace sama. Larry tvrdí, že ve většině případů by měl inženýr trávit týden navíc se snahou redukovat komplexnost aplikace, než aby milióny uživatelů museli trávit o minutu déle času používáním aplikace.

Von Restorff efekt

Klíčová myšlenka Efekt rovněž známy pod názvem izolační efekt, predikuje, že když se vyskytuje několik podobných objektů, tak ten, který se nejvíce liší od ostatních bude zapamatován. Z tohoto důvodu je dobré odlišovat klíčové akce od ostatních.

Původ Teorii vytvořila psychiatrička Hedwig von Restorff, která pozorovala, že lidé, kterým byl prezentován seznam kategoricky podobných objektu s jedním rozlišným izolovaným objektem si tento objekt zapamatovali znatelně lépe.

Obrázek 41: Von Restorff efekt.

Zeigarnik efekt

Klíčová myšlenka Lidé si lépe pamatují nedokončený nebo přerušený úkol, než úkol dokončený. Z toho plyne využití *progress baru* pro komplexnější operace. Je důležité vizualizovat v jaké části úlohy se uživatel nachází.

Původ Bluma Wulfovna Zeigarnik byla sovětská psycholožka a psychiatrička. Objevila Zeigarnik efekt a přispěla k založení experimentální psychopatologie jako separátní disciplíny v sovětské unii po druhé světové válce.

Gestaltův zákon seskupení

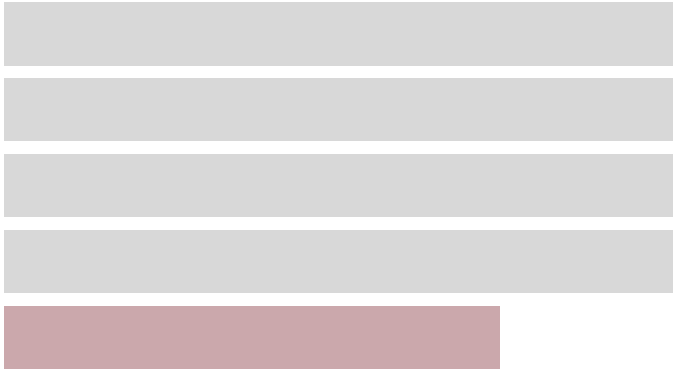
Klíčová myšlenka Objekty jsou vnímány jako jedna skupina, pokud se nacházejí v oblasti s jasně definovaným okrajem. Přidání rámečku (nebo viditelné mezery) okolo objektu nebo skupiny objektů je jednoduchý způsob jak separovat okolní objekty.

Zákon blízkosti

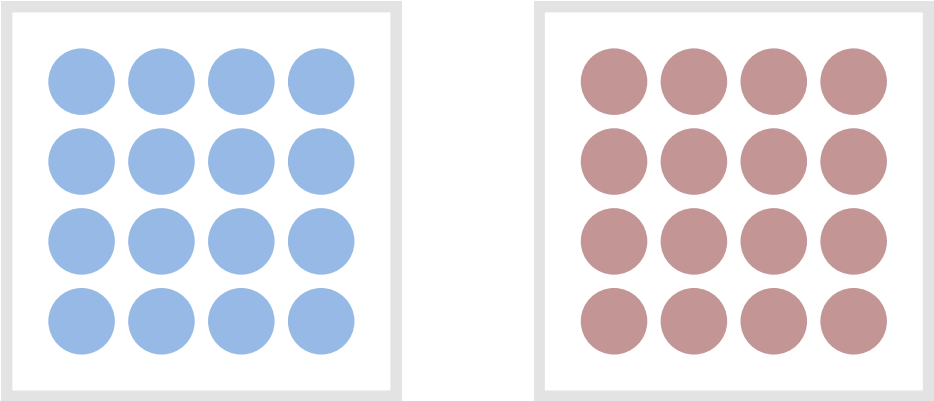
Klíčová myšlenka Objekty které jsou blízko sebe jsou vnímány jako jedna skupina. Toto pravidlo umožňuje uživatelům seskupovat objekty do skupin na základě jejich vzájemné pozice.

Zákon podobnosti

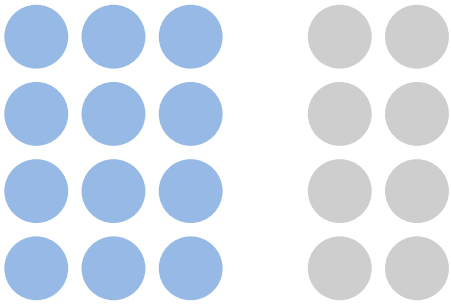
Klíčová myšlenka Lidské oko má tendenci pozorovat podobné elementy v designu jako komplexní obraz, tvar nebo skupinu. Děje se tak i přes to, že jsou elementy odděleny mezerou. Z toho důvodu je nutné



Obrázek 42: Zeigarnik efekt.



Obrázek 43: Gestaltův zákon seskupení.



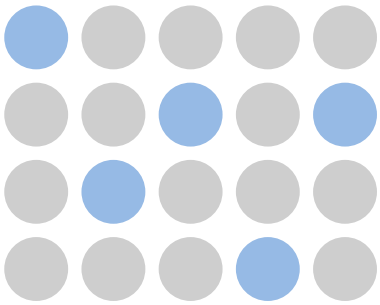
Obrázek 44: Zákon blízkosti.

vizuálně oddělit elementy, které mají uživatelé považovat za rozdílné. Například tedy navigaci od běžného textu.

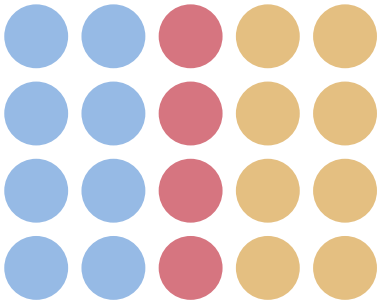
Zákon o vizuální spojitosti

Klíčová myšlenka Elementy, které jsou vizuálně spojené jsou vnímány jako více související, než objekty, které vizuálně spojené nejsou. Funkce podobného zaměření by tedy měly být vizuálně sjednoceny.

Původ (Gestaltův zákon seskupení, Zákon blízkosti, Zákon podobnosti, Zákon o vizuální spojitosti) Gestalt přišel první s myšlenkou, že lidé přirozeně vidí objekty organizované do skupin či vzorů. Dále argumentuje, že tento princip existuje protože mysl má dispozici k rozpoznávání skupin a vzorů na základě určitých pravidel.



Obrázek 45: Zákon podobnosti.



Obrázek 46: Zákon o jednotné spojitosti.

Grafické uživatelské rozhraní

Na rozdíl od textového rozhraní umožňuje uživatelskou interakci formou grafických elementů jako jsou tlačítka, formuláře, ikony a jiné. Uživatel pro dosažení cíle přímo manipuluje s grafickými prvky uživatelského rozhraní.

V následující kapitole se budeme věnovat obecným principům grafického rozhraní a poté představíme základní elementy, které toto rozhraní tvoří. Zdroje a zároveň doporučená literatura pro následující kapitolu jsou tyto knihy:

- Designing Interfaces[29]
- Don't Make Me Think, Revisited[30]
- The Essential Guide to User Interface Design[31]
- The Non-Designer's Design Book[32]
- UI is Communication[33]

[29] Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media; Second edition, 2011

[30] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders; 3 edition, 2014

[31] Wilbert O. Galitz. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley; 3 edition, 2007

[32] Robin Williams. *The Non-Designer's Design Book*. Peachpit Press; 4 edition, 2014

[33] Everett N McKay. *UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication*. Morgan Kaufmann; 1 edition, 2013

Grafický layout

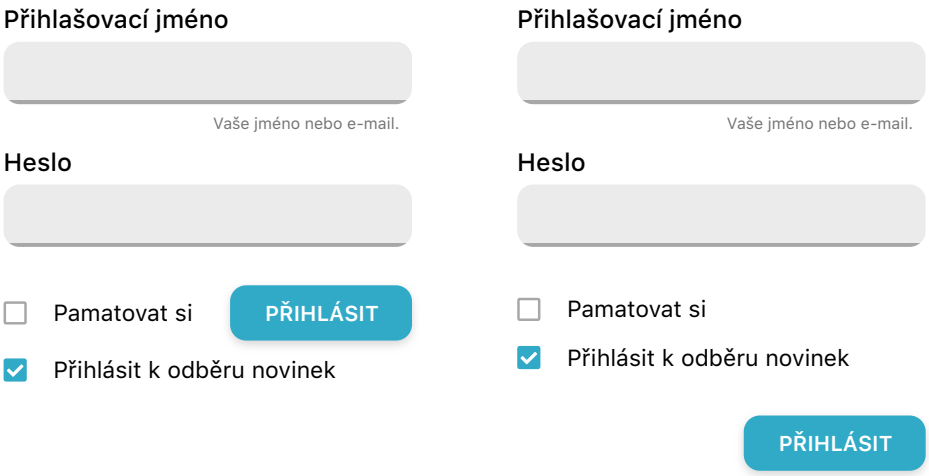
Layout neboli rozvržení je umístění, velikost a vzdálenost UI elementů v rámci obrazovky. Správný layout je kritický pro srozumitelnou a přesnou komunikaci mezi aplikací a uživatelem. Komunikuje účel obrazovky, jakým způsobem ji procházet a číst a v jakém pořadí interagovat s jejími elementy. Kvalita layoutu rovněž ovlivňuje jakou rychlostí jsou schopni uživatelé najít požadovanou informaci/akci v rámci obrazovky.

Pochopení layoutu bývá pro ne-designéry nejtěžší částí návrhu uživatelského rozhraní. Často se zaměřují na technický layout a snaží se elementy umisťovat takovým způsobem aby vyplnili volný prostor. Layout není Tetris a umístění každého elementu musí mít svůj důvod (viz. obr. 47).

Dva způsoby prohlížení: čtení a skenování

Steve Krug ve své knížce „Don't make me think"[34] napsal, že uživatelé

[34] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders; 3 edition, 2014



Obrázek 47: Layout není Tetris, vlevo špatně, vpravo dobře.

nečtou, uživatelé skenují.

Čtení Je velice podobné běžnému čtení textu, cílem je získat podrobné informace. V západní kultuře jsme zvyklí číst zleva doprava. Během čtení přečteme většinu viditelného obsahu/textu, je však možné, že přeskočíme méně zajímavý/čitelný obsah.

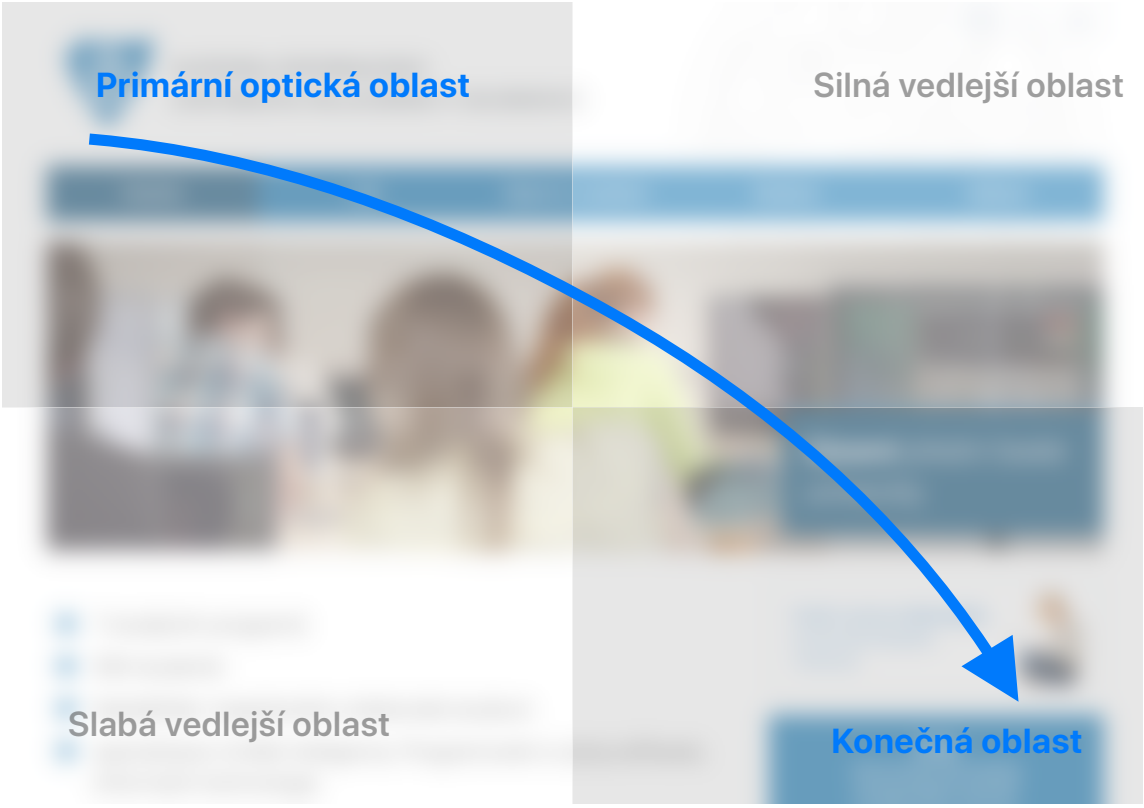
Skenování V případě, že je cílem najít potřebnou informaci. Při této akci uživatelé obvykle detailně nepročítají celou obrazovku, pouze ji skenují. Skenování probíhá z levého horního rohu do pravého dolního, přitom kopíruje pomyslnou křivku. V případě menších zařízení (např. mobilní telefon) je skenování jednoduší – uživatelé sledují obrazovku od horního okraje po spodní. Je tomu tak kvůli tomu, že jsme schopni skenovat celou šířku obrazovky bez nutnosti čtení zleva doprava.

Tyto dva vzory jsou prokládány (viz. obr. 52). Uživatelé skenují dokud je některý z elementů nezaujme, poté přepnou na detailní čtení a čtou do té doby dokud jim daný element poskytuje informace, poté opět začnou skenovat. Tento proces se pak opakuje do té doby, než uživatel nenarazí na spodní okraj obrazovky. Poté se musí rozhodnout zda se posunout níže, nebo přejít na jinou obrazovku.

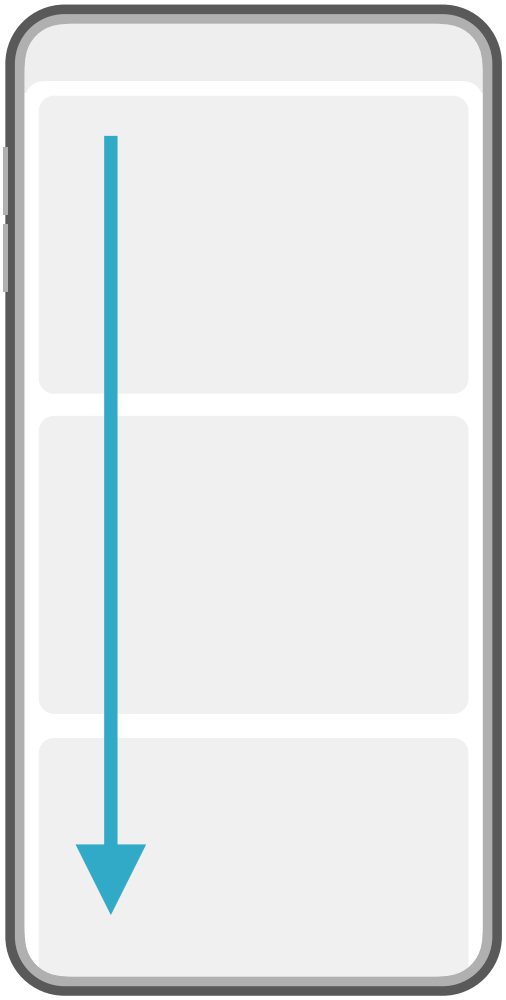
Curabitur sagittis hendrerit ante. Integer rutrum, orci vestibulum ullamcorper ultricies, lacus quam ultricies odio, vitae placerat pede sem sit amet enim. Etiam sapien elit, consequat eget, tristique non, venenatis quis, ante. Fusce consectetuer risus a nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Aenean placerat. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Fusce dui leo, imperdiet in, aliquam sit amet, feugiat eu, orci. Curabitur bibendum justo non orci. Phasellus enim erat, vestibulum vel, aliquam a, posuere eu, velit. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Aenean vel massa quis mauris vehicula lacinia. Vivamus ac leo pretium faucibus. Curabitur bibendum justo non orci. Nulla non arcu lacinia neque faucibus fringilla. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Vestibulum erat nulla, ullamcorper nec, rutrum non, nonummy ac, erat. Nullam faucibus mi quis velit. Fusce consectetuer risus a nunc.

Obrázek 48: Typický způsob čtení pro západní kultury.

Obrázek 49: Styl skenování na větších obrazovkách.



Obrázek 50: Schéma skenování překrývající webovou stránku.



Obrázek 51: Styl skenování na menších obrazovkách.

Obrázek 52: Čtení a skenování uživatel při práci s aplikací či webem kombinuje. Modrá barva značí skenování a červená čtení.

Proč tomu tak je?

Je běžné že uživatelé spěchají. Mnoho lidí potřebuje co v nejkratším čase docílit toho, že najdou hledanou informaci nebo provedou určitou akci. Steve používá připodobnění ke žralokům, uživatelé jsou neustále v pohybu, nikdy se nezastaví.

Dále víme, že většinou není potřeba zdaleka číst veškerý obsah. Skenování nám umožní lokalizovat tu část/funkcionalitu, která nás zajímá.

V neposlední řadě jsme zjistili, že lidé jsou ve skenování dobří. Tuto dovednou používáme nejenom při interakci v počítačem. Jedná se o obecný přístup hledání, který aplikujeme například při čtení novin a časopisů.

Mezi další zajímavé poznatky patří fakt, že uživatelé nehledají optimální řešení. Intuice by nám napovídala, že uživatelé provedou skenování celé aplikace a až poté se rozhodnout jakou akci dále podstoupí. V realitě však uživatelé akce podstoupí mnohem dříve, skenují a jakmile mají *pocit*, že vědí co dál, tak akci provedou. Toto chování pramení částečně také z toho, že pokud uživatel udělá chybu, není za ní dostatečně penalizován[35].

[35] Je zajímavé, že při zasílání peněz v elektronickém bankovnictví jsou lidé obezřetnější než při používání grafického editoru. Tady lze pozorovat nastavení mysli uživatele.

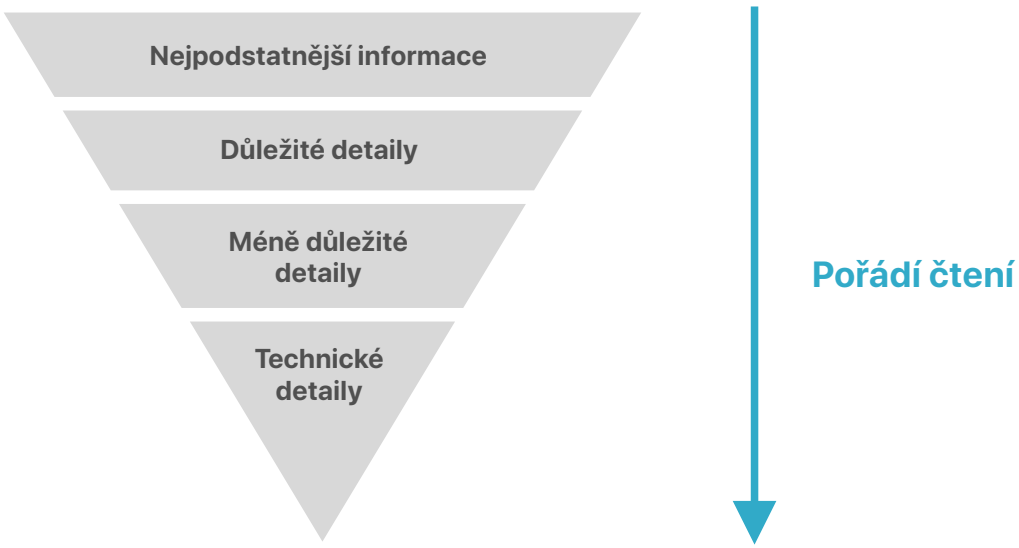
Nejenom, že uživatelé nehledají optimální řešení, často se ani nesnaží pochopit jak aplikace funguje, prostě jen „zkoušejí“. Tento vzorec chování se neobjevuje pouze u začátečníků, ale i u pokročilých uživatelů. Prostě je často dostatečné, že věc funguje, není důležité abychom chápali proč tomu tak je.

S ohledem na dostupné informace pojednávající o chování uživatelů můžeme přizpůsobit přístup k tomu jakým způsobem uživatelské rozhraní navrhujeme. Jedná se především o tyto body:

- Využití známých konvencí.
- Vytváření efektivních hierarchii.
- Rozdělení aplikace do jednoznačně definovaných oblastí.
- Elementy které slouží k interakci by měli být očividně rozeznatelné.
- Eliminovat rušivé elementy.
- Formátovat text aby umožňoval skenování.

Využití známých konvencí

Jeden z nejlepších způsobů jak učinit aplikaci pochopitelnou i ve spěchu je použití již dobře známých konvencí. Konvencí myslíme to jak elementy vypadají, jak se používají a k čemu se používají. Ve většině případů není nutné znovu objevovat kolo.



Obrázek 53: Invertovaná pyramida

Vytváření efektivních hierarchii

V rámci obrazovky by mělo být jasné jaké elementy si jsou podobné, jaké elementy jsou důležité, jaké jsou součástí jiných elementů atd. Jinak řečeno se jedná o definování jasné hierarchie.

Nejdůležitější elementy by měly být největší, nejtlustší, odsazené většími okraji umístěny někde v horní části obrazovky – případně různé kombinace.

Elementy, které spolu logicky souvisí by spolu měly souviset vizuálně. Vizuální zanoření elementů jasně určuje jejich vzájemný vztah.

Správná vizuální hierarchie vytváří jakési předzpracování obsahu, které pak již uživatel nemusí dělat. *Invertovaná pyramida* symbolizuje správnou organizaci obsahu (obr. 53). Uživatel má možnost kdekoli čtení přerušit a ví, že veškerý další obsah bude pouze konkrétnější než on potřebuje.

V neposlední řadě je důležité říct, že tyto pravidla nejsou specifická pouze pro design uživatelského rozhraní, ale jsou hojně využívaná v tiskovinách.

Rozdělení aplikace do jednoznačně definovaných oblastí

Rozdělení obrazovky na oblasti umožní uživatelovi učinit rozhodnutí které oblasti se bude věnovat a jaké oblasti bude ignorovat. Jakmile uživatelé naleznou užitečnou informaci v určité oblasti přestávají prozkoumávat ty ostatní. V extrémním případě se jedná o jev, kdy uživatelé naprosto vytěsňují reklamy na webových stránkách.

Jednoznačně rozeznatelné „klikatelné“ elementy

Elementy zprostředkovávající akci nebo navigaci musí být viditelné. Uživatelé během skenování pozorují a hledají elementy u kterých si myslí, že jsou „klikatelné“[36]. Viditelnost těchto elementů by měla být realizována konzistentním stylem, neboli například všechny odkazy by měly vypadat stejně.

Toto chování je pro lidi rovněž typické i v běžném životě (např. hledání klik dveří).

[36] Lze na ně kliknout myší, nebo „tapnout“ prstem

Eliminovat rušivé elementy

Vizuální šum je velký nepřítel především webu. Uživatelé mají rozdílný práh tolerance pro vizuálně komplexní weby. V kontextu urbanizmu je tento jev nazýván vizuální smog – přehlcenost měst vizuálně nevhodnými billboardy/plakáty. Je dokonce známo, že si uživatelé lepili lístečky na obrazovku jen kvůli tomu aby mohli nerušeně číst text na webu[37].

Je tedy důležité to s viditelností elementů nepřehánět.

[37] Zde vidíme, že různé „readning mode“ funkce mají své opodstatnění

Formátovat text aby umožňoval skenování

V rámci webu uživatelé většinu času tráví prohledáváním textu. Je tedy vhodné nejenom tam text formátovat takovým způsobem aby umožňoval jednoduché skenování.

Toho docílíme například častým používáním nadpisů. U nadpisů je rovněž důležité zajistit dostatečnou vizuální rozdílnost v případě využití více úrovní nadpisů. K více úrovním nadpisů se rovněž váže sémantika nepřeskakování jednotlivých úrovní směrem dolu. Tedy například nadpis úrovně 1 musí být následován nadpisem úrovně 2, nikoli nadpisem úrovně 3.

Dalším krokem je dělit text do menších odstavců. Lidé spoléhají, že odstavec symbolizuje nějakou informaci u které když zjistí, že je nepotřebná, mohou odstavec přeskočit. Dlouhý odstavec podobně jako dlouhou funkci[38] je vhodné rozdělit.

[38] V kontextu programování

Odrážkové seznamy rovněž podporují jednoduché skenování a jsou doporučeným prvkem v rámci textového obsahu.

Obrázek 54: Správné pořadí elementů respektující přirozenou komunikaci.

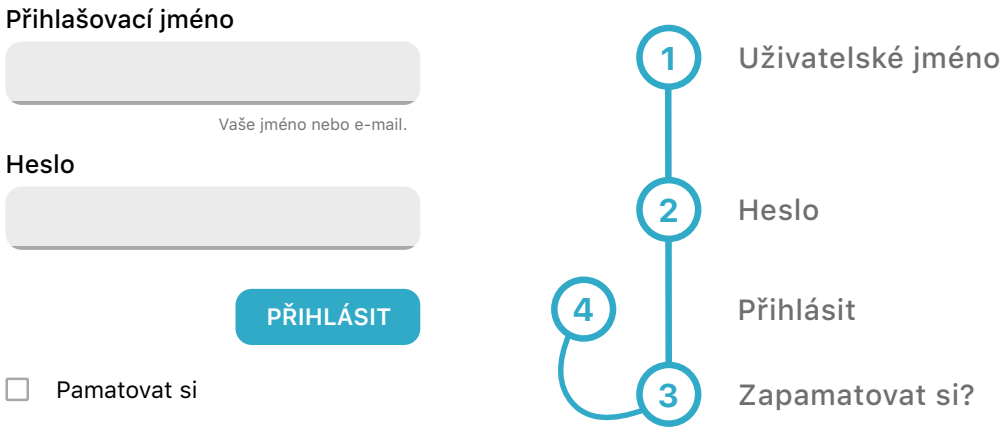
Pořadí elementů na stránce

Pořadí elementu na obrazovce by mělo následovat pořadí běžné konverzace mezi lidmi. Nejvíce je to viditelné například u formulářů.

- **Aplikace:** Jaké je Vaše uživatelské jméno?
- **Uživatel:** František Palacký
- **Aplikace:** Jaké je Vaše heslo?
- **Uživatel:** Bezpecne123
- **Aplikace:** Přejete si zůstat přihlášen?
- **Uživatel:** Ano

Vyváženost

Pokud bychom pomyslně umístili obrazovku na klín (jak je zobrazeno na Obrázku 56), chceme docílit horizontální rovnováhy obsahu. Prakticky to znamená, že elementy je vhodné umisťovat rovnoměrně na levou i pravou stranu obrazovky. Vše však musí zároveň splňovat i ostatní pravidla.



Obrázek 55: Špatné pořadí elementů, nerespektuje přirozenou komunikaci.

Obrázek 56: Vyváženost layoutu.

Čtyři obecné principy

K dosažení výše uvedených cílů nám mohou pomoci následující čtyři principy vycházející z tvarové psychologie (uvedené již dříve na straně 50). Nyní se na ně podíváme z praktického hlediska. Principy budou ilustrovány na obrazovce rozvrhu z mobilní aplikaci UPlikace.

Zákon seskupení Elementy umístěné v ohraničené oblasti působí jako souvislá skupina. Tento jev je viditelný na obrázku 57. Uživatel dostává jasné vodítko, že se jedná o skupinu souvisejícího obsahu.

Zákon blízkosti Elementy umístěné blízko sebe (vhodně oddělené mezerou od okolního obsahu) tvoří skupinu. Příklad si ukážeme v rámci stejné obrazovky na obrázku 58.

Zákon podobnosti Elementy vizuálně podobné tvoří skupinu bez ohledu na jejich umístění. To je důležité aby uživatel viděl souvislosti napříč obrazovkou. Naopak vizuální rozdílnost může uživatele upozornit na odlišnost elementu v kontextu okolních elementů. Tento princip je tedy klíčový pro elementy, které nejsou spojeny do skupiny pomoci vzdálenosti, je nutné vytvořit dostatečný rozdíl oproti okolním elementům (například tedy navigační prvky, tlačítka a další elementy). Demonstrace viditelná na obrázku 59.

Zákon vizuální spojitosti Elementy, které jsou vizuálně podobnější jsou vnímány jakou související. Barva, tvar, rámeček a mnoho dalšího tedy pomáhá seskupovat elementy do stejných významových skupin. Tento princip je podobný principu předchozímu, jedná se zde však o důraz na docílení vizuální podobnosti, nikoli rozdílnosti jak tomu bylo v předchozím principu. Demonstrace rovněž viditelná na obrázku 60.

Mobile-first design

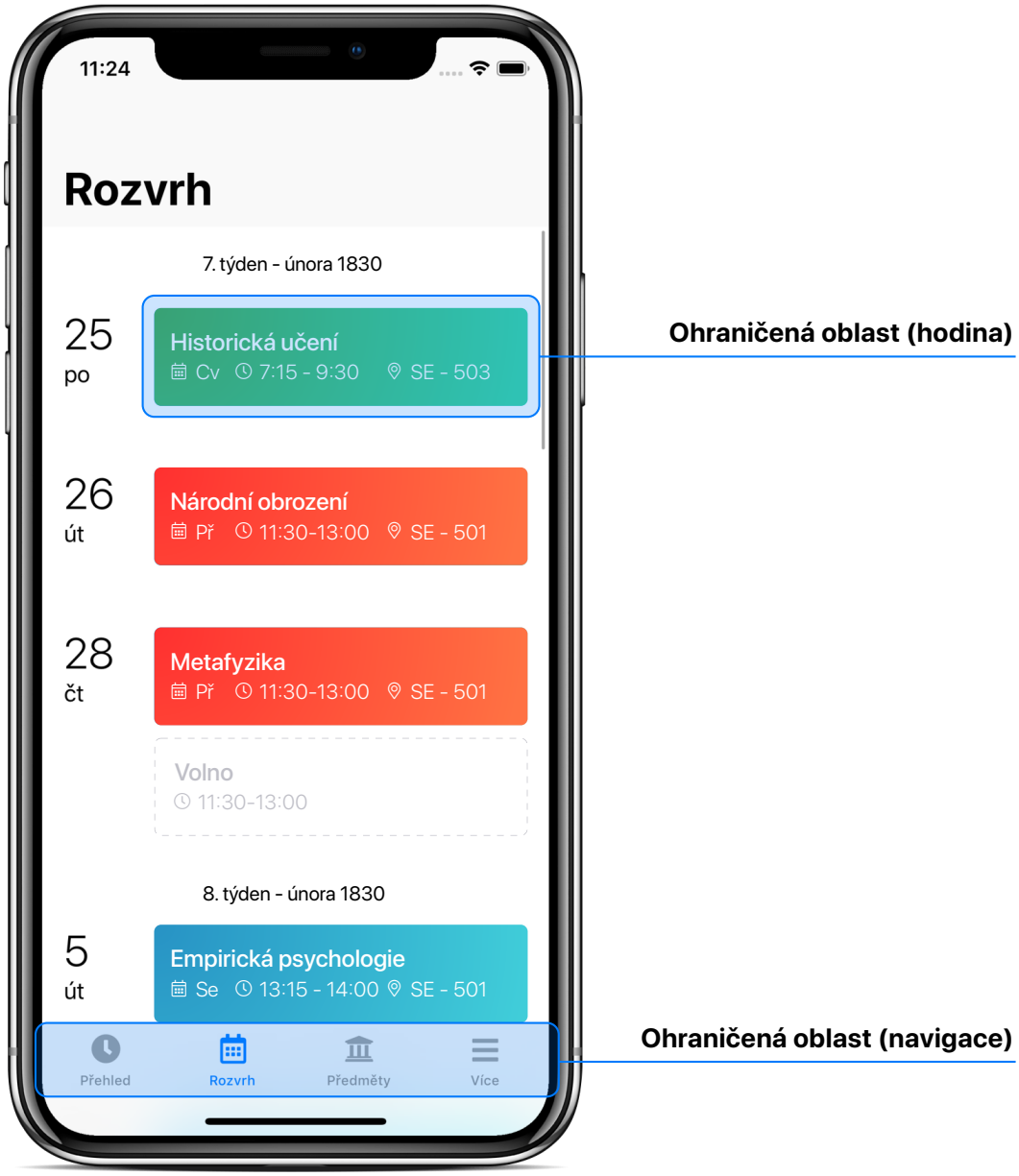
Většina aplikací je již používána na mobilních zařízeních, je tomu tak převážně u webu a webových aplikací. Je tedy vhodné navrhovat layout s ohledem na menší obrazovky.

Upřednostnění mobilních zařízení při procesu návrhu se nazývá *Mobile-first design*[39]. Navrhováním layoutu pro menší obrazovky jsme nuceni upřednostnit jednoduchost před komplexností, tento přístup má tedy pozitiva i pro aplikace, které nebudou v drtivé většině používány na mobilních zařízeních[40].

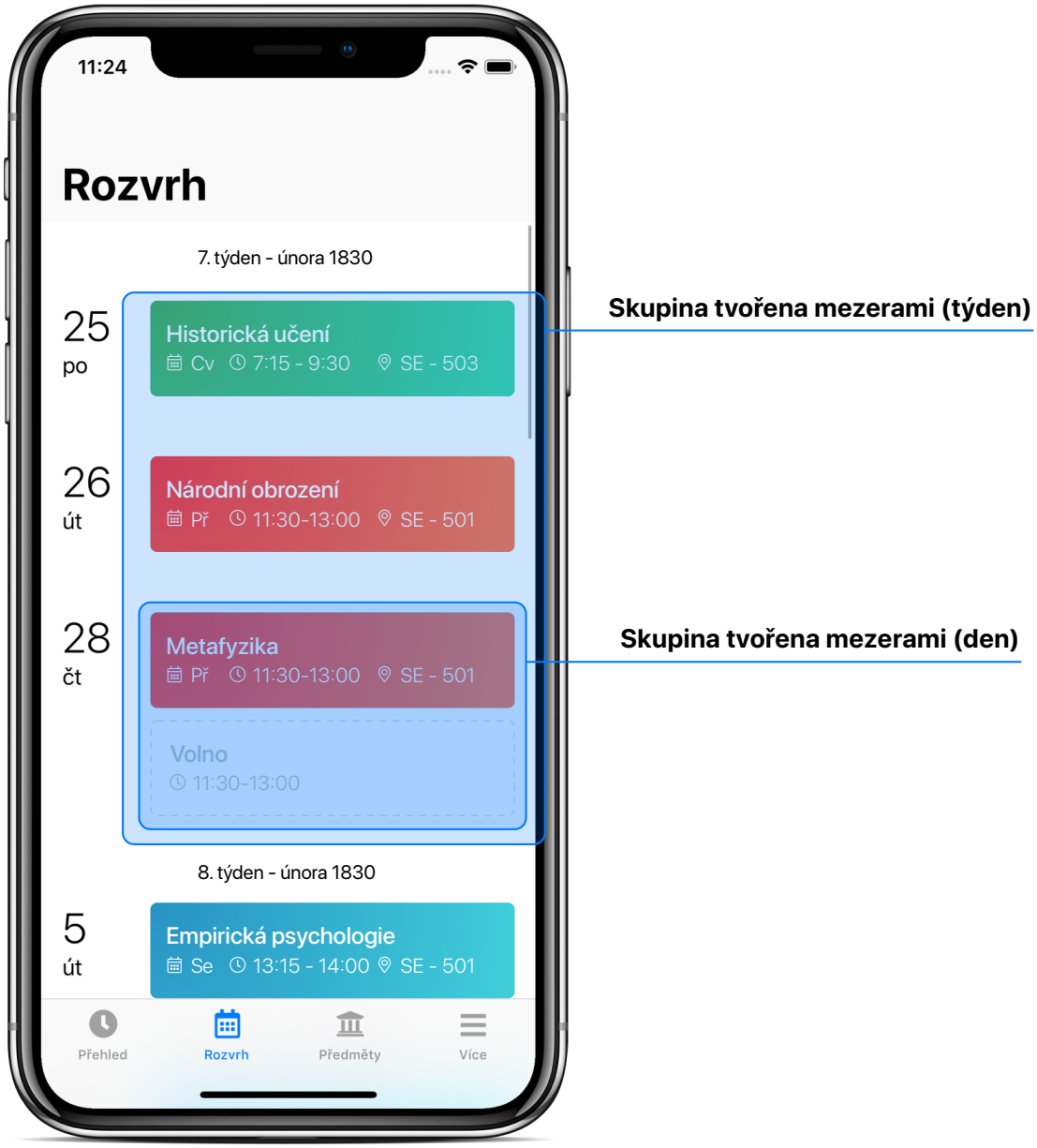
Mobile-first design tedy zlepšuje:

[39] Martin Michálek. *Vzhůru do (responzivního) webdesignu*. Martin Michálek - Vzhůru dolů, 2018

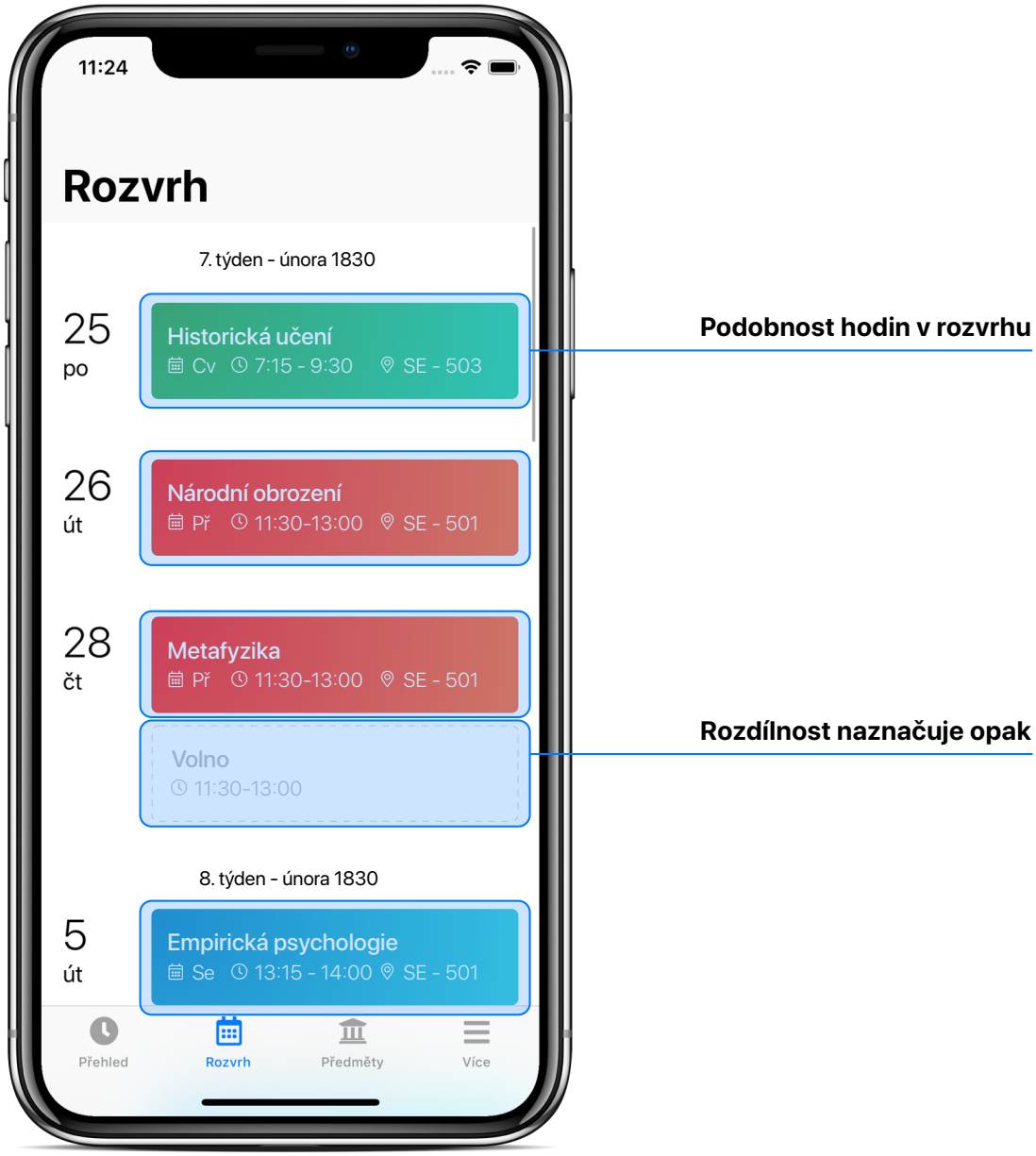
[40] Nehledě na to, že v dnešní době by každá webová stránka/aplikace, měla nějakým způsobem rovněž fungovat na mobilních zařízeních



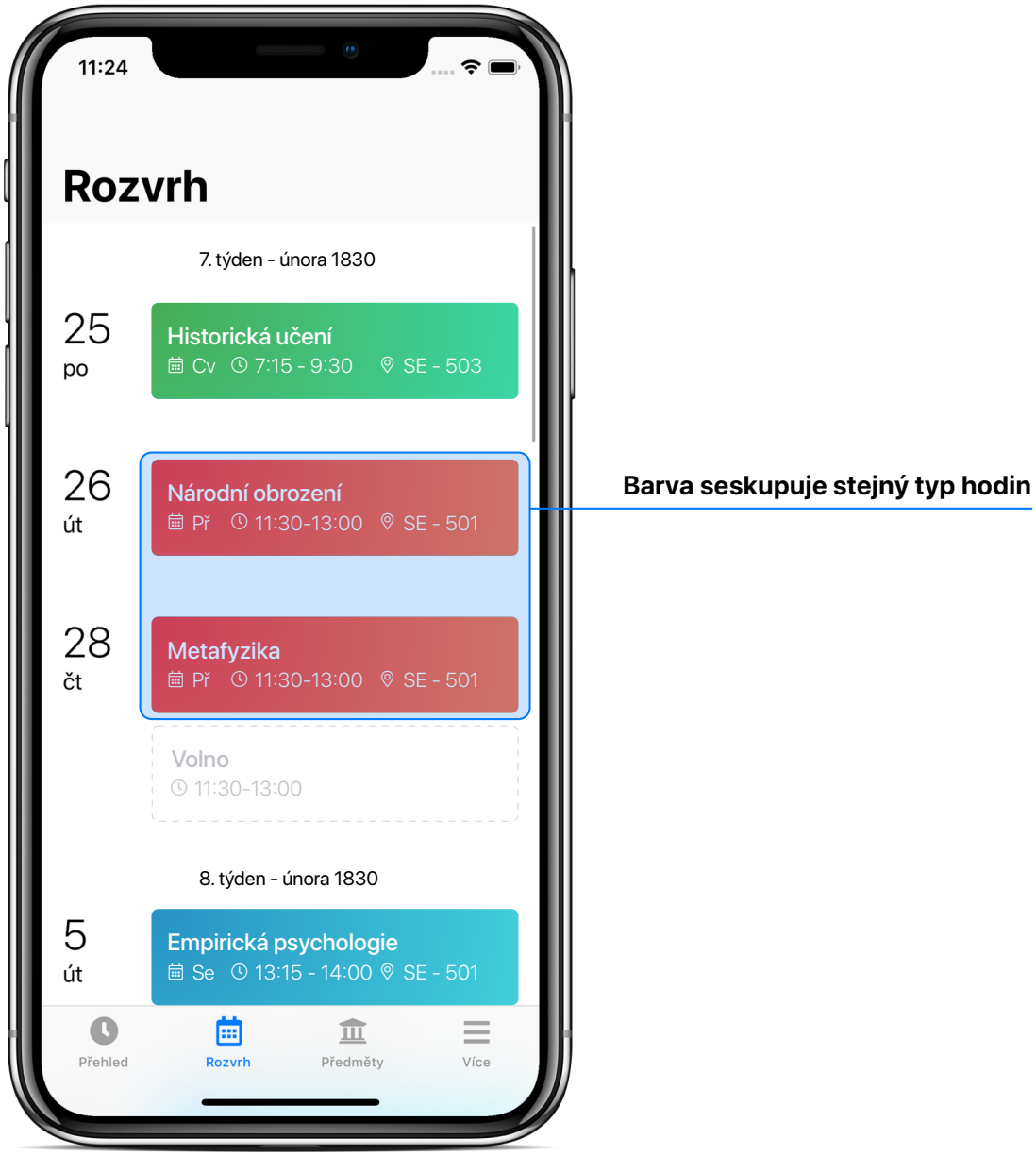
Obrázek 57: Demonstrace zákonu seskupení.



Obrázek 58: Demonstrace zákonu blízkosti.



Obrázek 59: Demonstrace zákonu podobnosti.



Obrázek 60: Demonstrace zákonu vizuální spojitosti.

Obrázek 61: Demonstrace ovladatelnosti mobilního telefonu v orientaci na výšku.

- Srozumitelnost a přehlednost.
- Zavádí snadnější lineární způsob konzumace obsahu.
- Zvětšuje velikost ovládacích prvků.

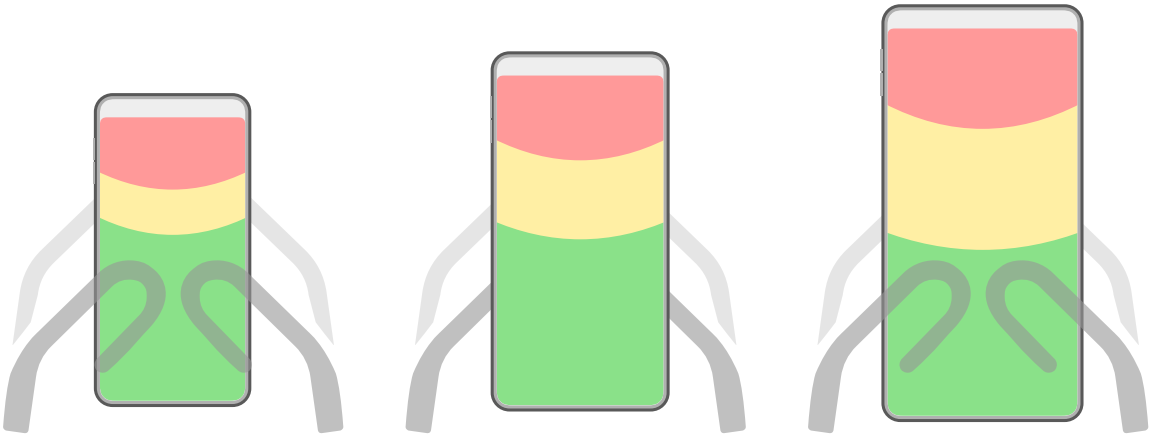
To jakým způsobem uživatelé drží mobilní zařízení má velký vliv na styl a možnost jeho ovládání. Běžný uživatel střídá různé styly držení a orientace mobilního zařízení.

Většina uživatelů drží telefon v jedné ruce a dotykovou obrazovku ovládá palcem, pouze 28% uživatelů drží telefon v jedné ruce a dotykovou obrazovku ovládá ukazováčkem druhé ruky. Jakým způsobem je ovladatelnost zařízení ovlivněna je možné vidět na obrázku 61. Zelená oblast je dosažitelná nejlépe, žlutá hůře a červená nejhůře. Je nutné podotknout, že s zvětšováním zařízení se rovněž zhoršuje jeho ovládání.

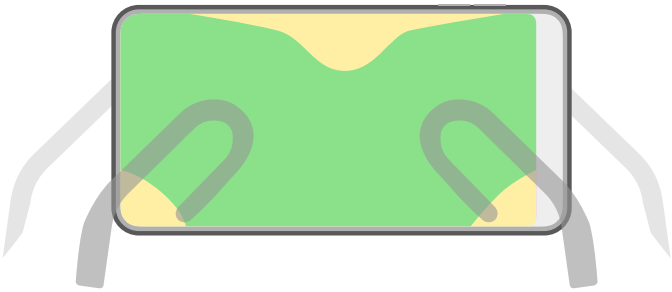
Při držení telefonu oběma rukama, je situace o něco lepší. Toto držení uživatelé používají nejčastěji při psaní. Demonstrace je opět viditelná na obrázku 62.

Orientace na šířku je nejčastěji používaná při hraní her nebo sledování multimediálního obsahu. V této orientaci je dosažitelná skoro celá obrazovka (viz. obr. 63).

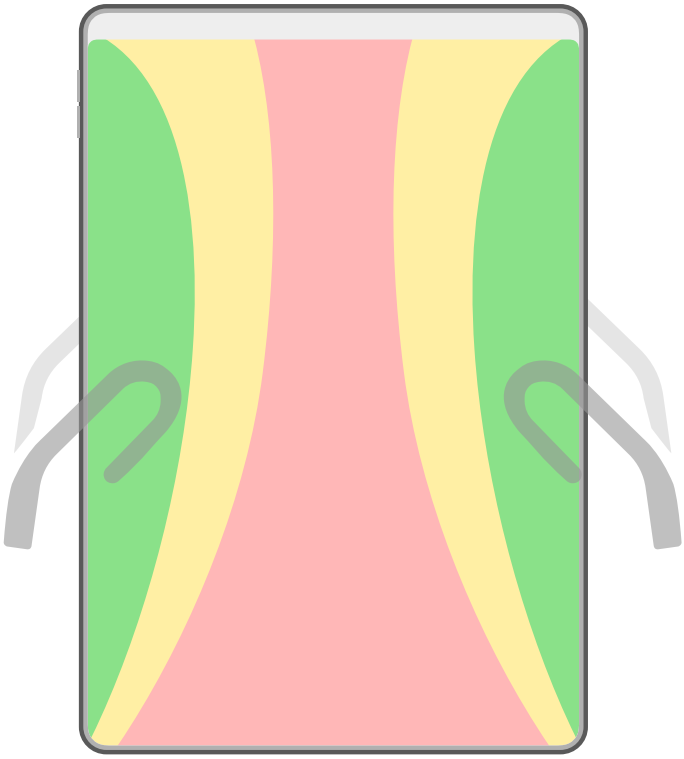
U větších zařízení jako je tablet je nejčastější držení oběma rukama, dosažitelnost obrazovky je ilustrována na obrázku 64.



Obrázek 62: Demonstrace ovladatelnosti mobilního telefonu drženého oběma rukama v orientaci na výšku.

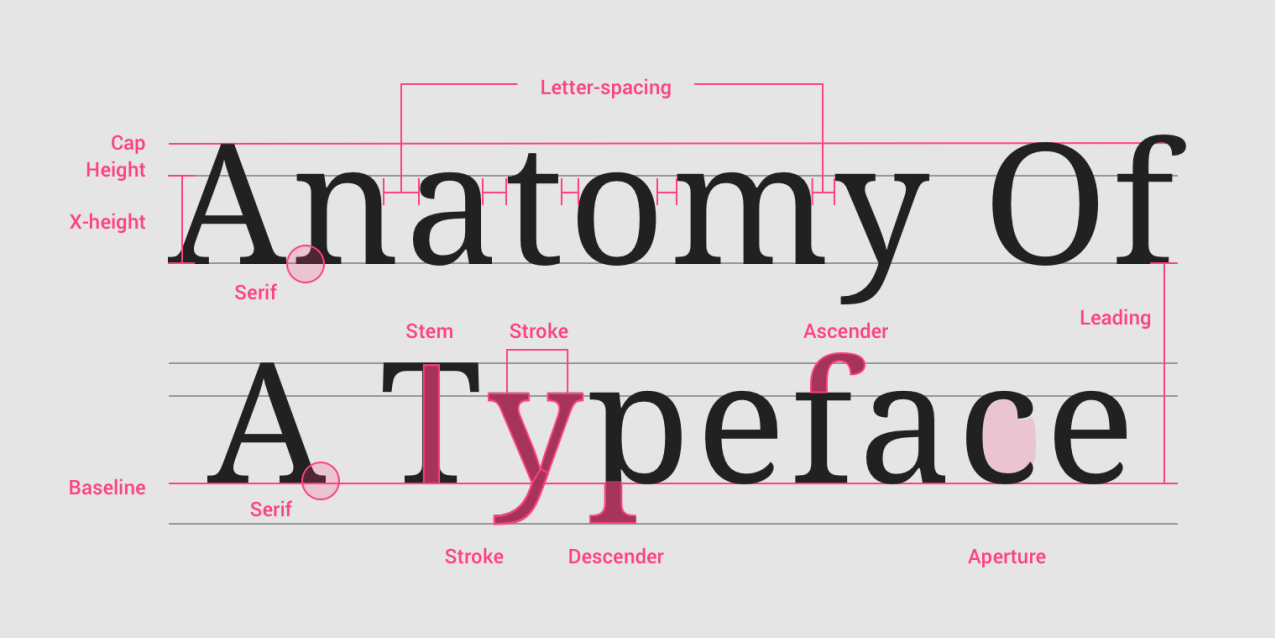


Obrázek 63: Demonstrace ovladatelnosti mobilního telefonu drženého oběma rukama v orientaci na šířku.



Obrázek 64: Demonstrace ovladatelnosti tabletu drženého oběma rukama v orientaci na výšku.

Typografie



Obrázek 65: material.io – Understanding typography

„I decided to take a calligraphy class to learn how to "learn calligraphy". I learned about serif and sans-serif typefaces, about varying the space between different letter combinations, about what makes great typography great. It was beautiful. Historical. Artistically subtle in a way that science can't capture. And I found it fascinating. None of this had any hope of any practical application in my life. But 10 years later, when we were designing the first Macintosh computer, it all came back to me. And we designed it all into the Mac. It was the first computer with beautiful typography. If I had never dropped in on that single course in college, the Mac would never have multiple typefaces or proportionally spaced fonts. And since Windows just copied the Mac, it's likely that no personal computer would have them."

- Steve Jobs,

V následující podkapitole bych se rád velice okrajově věnoval typografii[41]. Typografie je umění a pravidla stylu, vzhledu a struktury písma se záměrem poskytnout čtenáři estetický a dobře čitelný text. Rovněž můžeme říct, že se jedná o transformaci jazyka do dekorativního vizuálního elementu. Typografie je poměrně bohatá na svoji historii, která začíná někde okolo 11-12. století.

Proč je typografie důležitá? Většina vizuální krásy designu pramení právě z písma. Špatně provedená typografie má znatelný dopad na kvalitu designu uživatelského rozhraní[42].

[41] Google. Material design guidelines. `https://material.io/design`

[42] Jak esteticky tak funkčně – například v čitelnosti textu

Klíčové pojmy z typografie uvádím v anglickém jazyce, je to nejčastější forma jejich výskytu v různých příručkách případně softwaru.

Font vs. Typeface

Font a *typeface* jsou často zaměňované pojmy. Typeface si můžeme představit jako „rodinu“ a jednotlivé fonty jsou její členové. Typeface je tedy styl designu písma, který obsahuje kompletní sadu znaků ve všech velikostech a tloušťkách. Font je potom grafická reprezentace textového znaku konkrétní typeface, velikosti a tloušťky.

Ve většině případu[43] jsou tyto zájmy již volně zaměňovány.

[43] Převážně s nástupem digitální typografie

Mean vs. Base line

Znaky textu jsou nejčastěji umístěny na pomyslné rovné lince – řádku. Tuto linku nazýváme *base line* – značí spodní okraj těla znaku. V kontrastu *mean line* značí horní okraj těla znaku. Celá situace je ilustrována na obrázku 66.

Obrázek 66: Mean line vs. Baseline

Size, weight, height

Vlastnosti umožňující designérům rozlišit důležitost různých informací.

Size *Size* neboli velikost písma, rovněž různé jednotky „px“, „pt“, „em“, atd.

Weight *Weight* neboli tloušťka písma, určena číselnou hodnotou 100, 200, 300, atd. případně hodnotami „light“, „medium“, „bold“ atd.

Height *Height* nebo také *x-height*, česky pak výška znaku. Odvíjí se od znaku písmena „x“. Situace je ilustrována na obrázku 67.

Obrázek 67: X-height

Ascender vs. Descender

Ascender je část znaku sahající nad mean line (například u znaku „b“ nebo „h“). V kontrastu, *Descender* je část znaku sahající pod baseline (například u znaku „g“ nebo „j“). Situace je viditelná na obrázku 68.

Whitespace

Whitespace nebo často také *negative space* je volná plocha mezi grafickými entitami. Whitespace ovlivňuje čitelnost textu. Ačkoli je whitespace neviditelný, jedná se o jeden z nejdůležitějších prvků designu.

Obrázek 68: Ascender vs. Descender

Alignment

Alignment neboli zarovnání určuje k jakému okraji (případně středu) je text zarovnaný. Zarovnání spojuje jinak nespojitý text do bloků. Zarovnání existují tři druhy: „left“, „center“, „right“, „justified“. Jednotlivá zarovnání jsou demonstrována na obrázku 70.

Nejvíce problematické z nich je justified. V případě použití justified zarovnání mohou vznikat takzvané *řeky* (nejlépe pochopitelné z obrázku 71. Justified zarovnání vynucuje zarovnaní bloku textu z obou stran, z toho důvodu nejsou mezery mezi slovy konstantní. Tento problém můžeme jednoduše vyřešit zalamováním slov, poté je možné zarovnání justified používat bez obav[44].

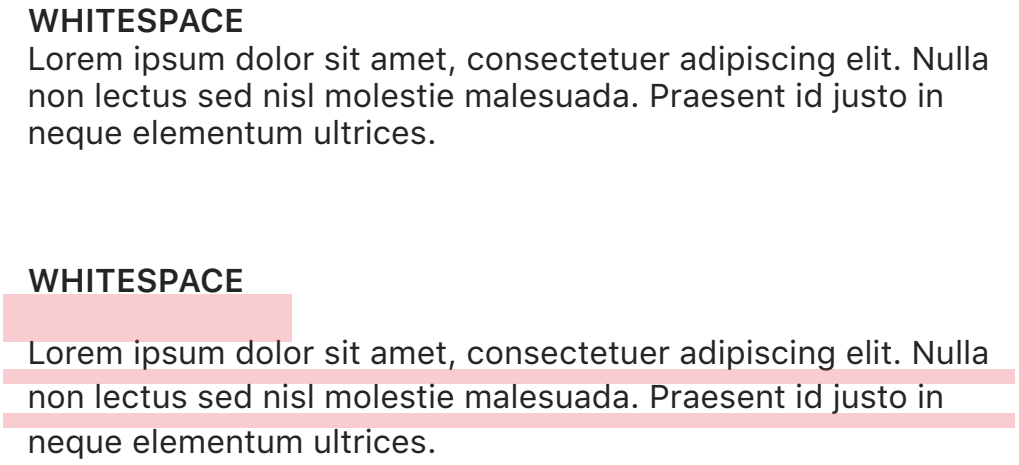
Obecně ale platí, že ve většina případů bychom si měli vystačit se zarovnáním doleva.

[44] Například text, který právě čtete využívá zarovnání justified se zalamováním slov.

Tracking vs. Kerning

Tracking je proces upravování globální velikosti mezery mezi znaky. Správná volba velikosti mezery učiní text vzdušným a dobře čitelným.

Kerning je proces velice podobný trackingu, jedná se však o upravování mezery mezi dvěma konkrétními znaky. Jedná se tedy o individuální případy kdy tato změna vede k čitelnějšímu nebo esteticky



Obrázek 69: Whitespace (vyznačený červeně).

působivějšímu textu.

Leading

Leading určuje velikost mezery mezi dvěma baseline. Správně nastavený leading pomáhá čtenáři jednoduše přejít z řádku na řádek. Standardní velikost leadingu je 120% velikosti písma.

Serif vs. Sans-serif

Písma dělíme na dvě hlavní kategorie, *Serif* neboli *patková* písma a *Sans-serif* neboli *bezpatková*. Rozdíl je patrný na první pohled. Patková písma obsahují tzv. *patky* (viditelné na obrázku 75.

Dříve platilo, že patkové fonty nacházely místo především v tiskovinách. Důvod by však prostý, obrazovky dřívější doby nedisponovaly vysokým rozlišením, což znemožňovalo detailně vykreslit často malé patky písma. V dnešní době obrazovek s vysokým rozlišením se jedná o volbu založenou na čisté preferenci[45]. Rovněž se neprokázalo, že by patkové písma byly lépe čitelná než písma bezpatková.

[45] Nielsen Norman Group. Serif vs. sans-serif fonts for hd screens. https://www.nngroup.com/articles/serif-vs-sans-serif-fonts-hd-screens

LEFT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla non lectus sed nisl molestie malesuada. Praesent id justo in neque elementum ultrices.

RIGHT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla non lectus sed nisl molestie malesuada. Praesent id justo in neque elementum ultrices.

JUSTIFIED

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla non lectus sed nisl molestie malesuada. Praesent id justo in neque elementum ultrices.

CENTER

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla non lectus sed nisl molestie malesuada. Praesent id justo in neque elementum ultrices.

Obrázek 70: Demonstrace typů zarovnání.

JUSTIFIED

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla non lectus sed nisl molestie malesuada. Praesent id justo in neque elementum ultrices.

Obrázek 71: Demonstrace „řeky“ v případě zarovnání justified.

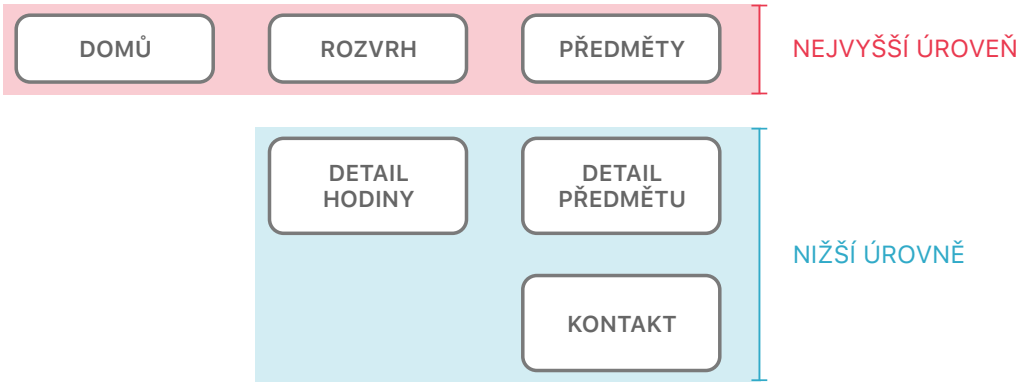
Obrázek 72: Demonstrace trackingu.

Obrázek 73: Demonstrace kerningu.

Obrázek 74: Demonstrace leadingu.

Ahg

Obrázek 75: Serif písmo se zvýrazněnými patkami.



Obrázek 76: Jednotlivé vrstvy aplikace.

Navigace

V následující sekci bych rád nastínil základní princip navigace, který je popsán v příručce Material Design[46]. Navigace je akt pohybu mezi jednotlivými obrazovkami aplikace/webu za účelem dokončit určitou akci.

[46] Google. Understanding navigation. `https://material.io/design/navigation/understanding-navigation.html`

Vrstvy aplikace

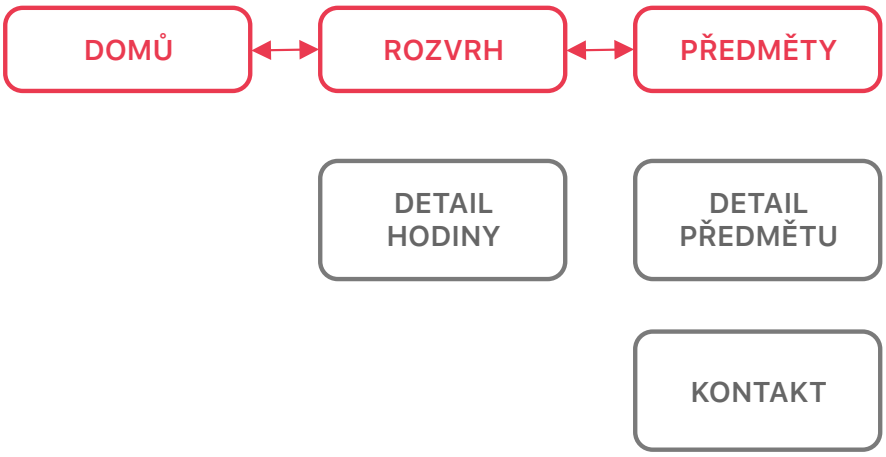
Každá aplikace obsahuje několik vrstev navigace. Každá vrstva je tvořena jednotlivými obrazovkami. Navigace poté probíhá mezi obrazovkami[47]. Příklad vrstev aplikace je viditelný na obrázku 76.

[47] Ať už v jedné vrstvě nebo napříč vrstvami

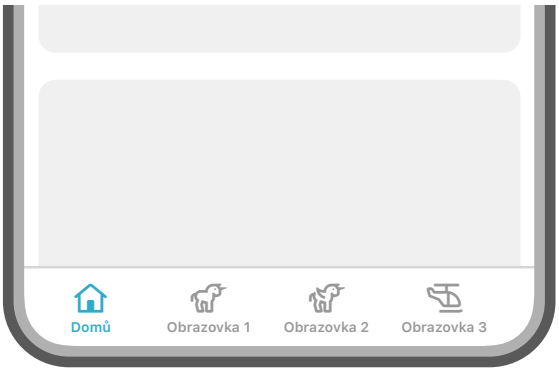
Typy navigace

Typy navigace jsou určeny tím, jakým způsobem uživatel přechází mezi obrazovkami, zda například sestupuje z jedné vrstvy na druhou, nebo zda se pouze pohybuje v rámci jedné vrstvy.

Boční navigace Jedné se o navigaci mezi hlavními obrazovkami aplikace. Je to tedy pohyb po nejvyšší úrovni aplikace a z této vrstvy se nikam nepřechází (obrázek 77). Tento typ navigace můžeme také nazvat *hlavní navigace*. Hlavní navigace by měla umožňovat přecházet mezi *všemi* hlavními obrazovkami.



Obrázek 77: Boční navigace v rámci hlavních obrazovek.



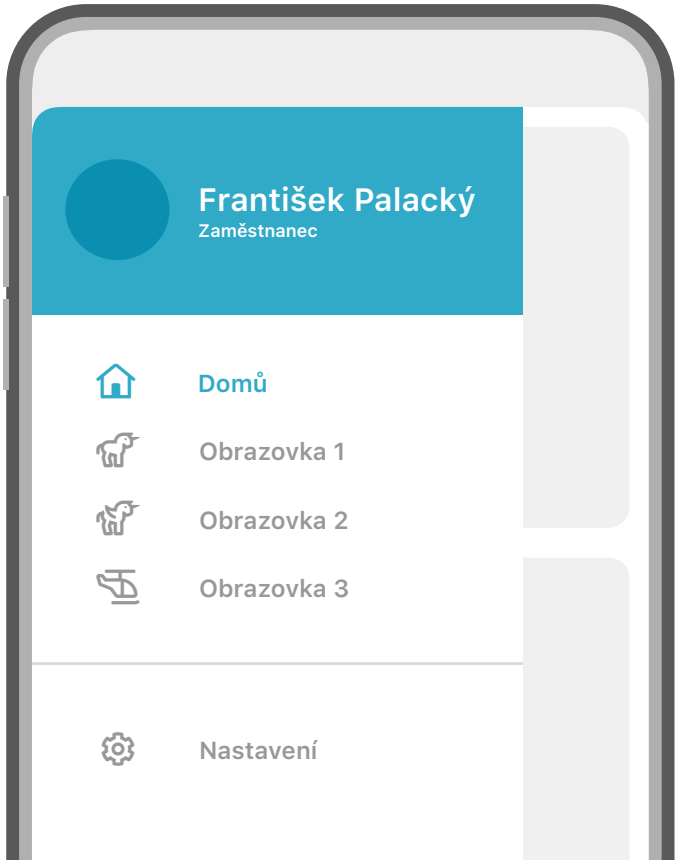
Obrázek 78: Hlavní navigace v dolní části obrazovky.

Hlavní navigace je realizovaná převážně prostřednictvím navigačních menu. Příklady hlavní navigace jsou viditelné na obrázku 78 a obrázku 79. Věřím, že použité elementy poznáváte, větší pozornost jim budeme věnovat později.

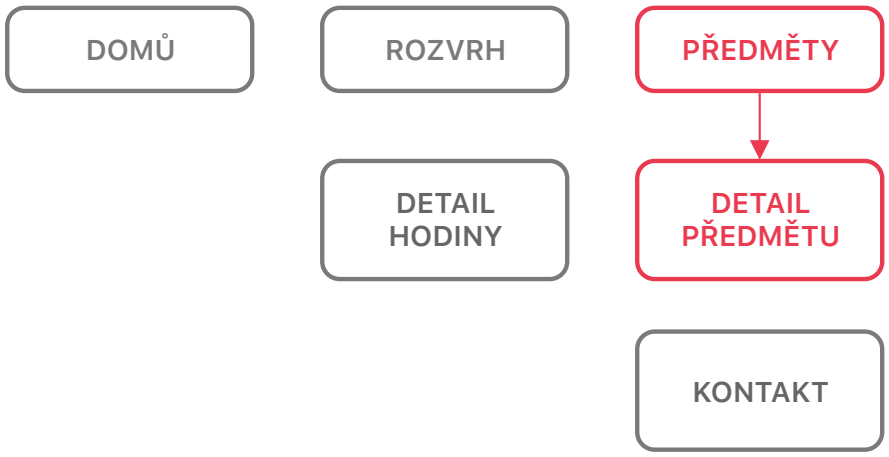
Dopředná navigace Pohyb mezi obrazovkami po sobě jdoucích vrstvách (obrázek 80). Při dopředné navigaci je možné vrstvy rovněž přeskakovat. Dopředná navigace je realizovaná skrze tlačítka, obrázky a další elementy (příklad na obrázku 81).

Zpětná navigace Jak už název napovídá jedná se o zpětný pohyb v rámci obrazovek. Zpětná navigace může být *chronologická* nebo *hierarchická*. Chronologická zpětná navigace respektuje přes jaké obrazovky uživatel na aktuální obrazovku přišel. Hierarchická zpětná navigace postupuje zpět po vrstvách, které nepřeskakuje.

Implementace zpětné navigace záleží na platformě, v drtivé většině je realizována některou formou „šipky zpět“. Příklad chronologické navigace je možné vidět na obrázku 83. Příklad hierarchické navigace je možné vidět na obrázku 84.

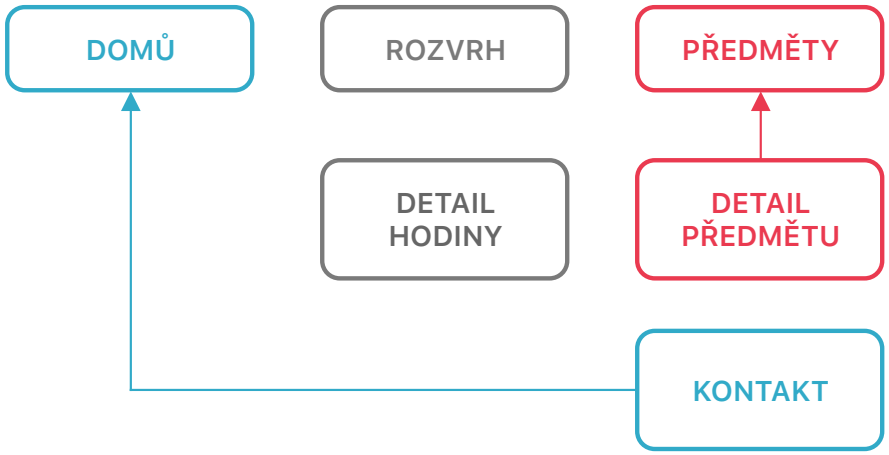


Obrázek 79: Hlavní navigace vysouvající se z boku obrazovky.



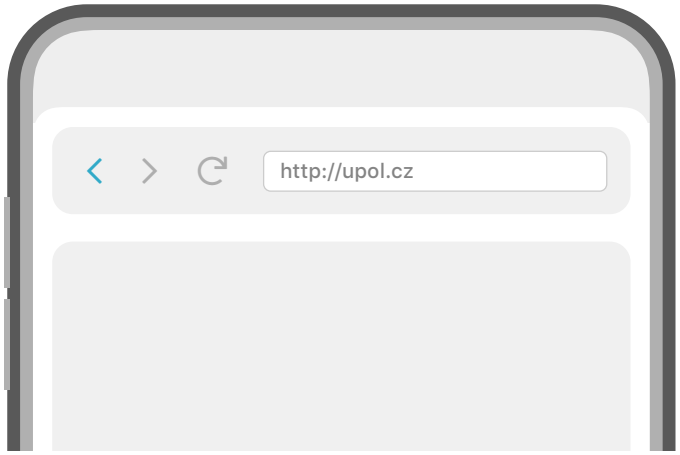
Obrázek 80: Dopředná navigace mezi obrazovkami.

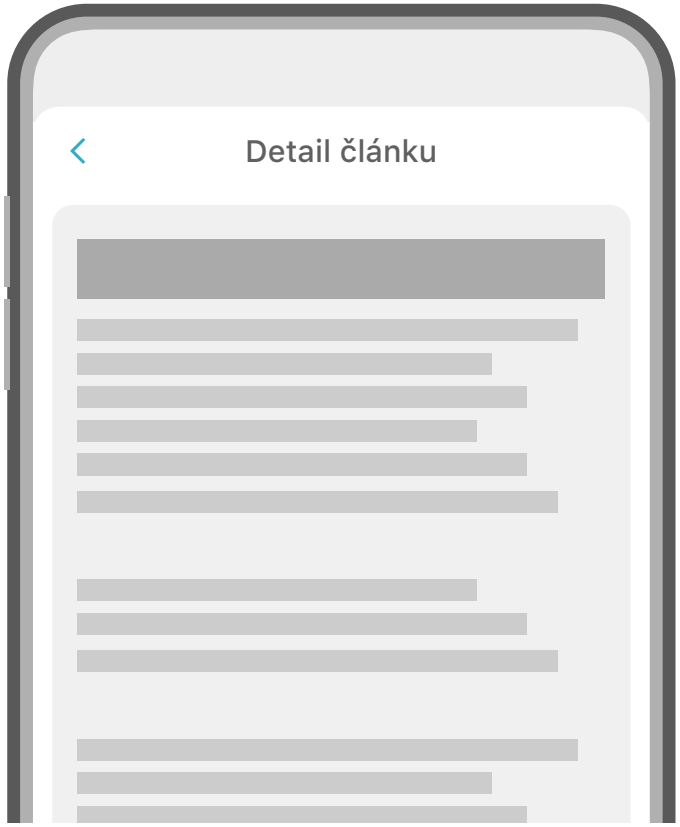
Obrázek 81: Dopředná navigace při zobrazování detailu článku.



Obrázek 82: Zpětná navigace mezi obrazovkami.

Obrázek 83: Chronologická zpětná navigace, příklad z web. prohlížeče.





Obrázek 84: Hierarchická zpětná navigace, příklad zobrazení detailu článku a následné vrácení zpět.

Elementy grafického uživatelského rozhraní

V předchozí kapitole byly nastíněny základní principy grafického uživatelského rozhraní, v této kapitole podrobně popisuji jednotlivé elementy z kterých je tvořena většina GUI. Každý element je zde popsán co nejobecněji, u některých elementů tato snaha vedla k určitým kompromisům. Je tedy možné, že v praxi budou u některých platforem některé elementy použity lehce odlišným způsobem.

Příkazy a akce

V následující sekci se budeme věnovat elementům reprezentující příkazy a akce. Některé elementy mohou obecně sloužit k více účelům, proto toto rozdělení berte pouze jako orientační.

Button

Button neboli tlačítko je jeden z nejpoužívanějších elementů. Jedná se o jasný element interakce, většina uživatelů mu rozumí. Je tomu tak především proto, že tlačítka se vyskytují všude kolem nás.

Tvarově je nejlepší následovat příklady z reálného světa. Nejklasičtější tvar je tlačítko hranaté nebo kulaté. Zaoblené rohy přitahují uživatelovu pozornost [48], obecně však platí, že své místo mají tlačítka se zaoblenými i ostrými rohy.

[48] Shan Shen. Make sense of rounded corners on buttons. https://uxdesign.cc/make-sense-of-rounded-corners-on-buttons-dfc8e13ea7f7/

Největším problémem u tlačítek je jejich viditelnost (obrázek 88). Jak již bylo zmíněno, tento problém začal být aktuálnější hlavně s příchodem *Flat designu*. Tlačítka bez výplně, případně s pouhým rámečkem (takzvané *ghost button*) mohou značně ovlivňovat viditelnost (obrázek 86). Textová tlačítka pak zastávají funkci méně výrazných (sekundárních) tlačítek.

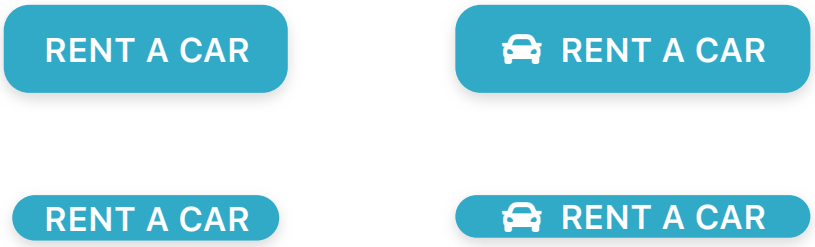
Tlačítko kromě popisu může obsahovat ostatní prvky – například ikonu. V jakékoli variantě tlačítka je důležité dodržovat rozumné vnitřní okraje (takzvaný *padding*). Tato situace je viditelná na obrázku 87.

Tlačítka se mohou vyskytovat v několika stavech. Běžně se uvádí těchto pět stavů: default, hover, focus, pressed, disabled (obrázek 89).

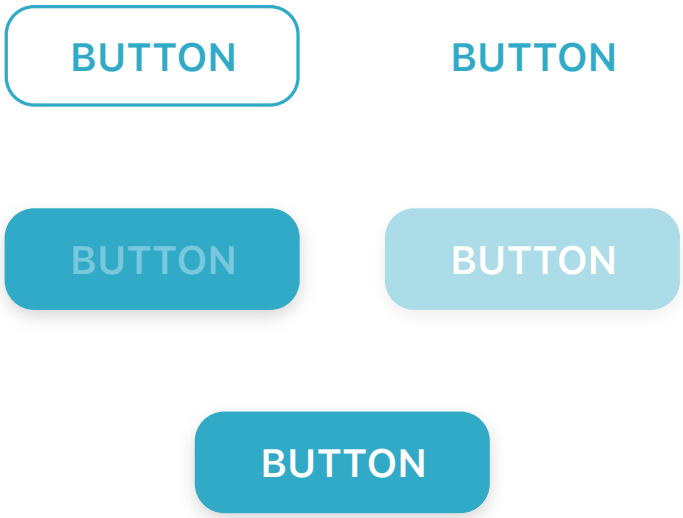
- **Default** – Výchozí stav tlačítka.

Obrázek 85: Tlačítko jako jasný element interakce.

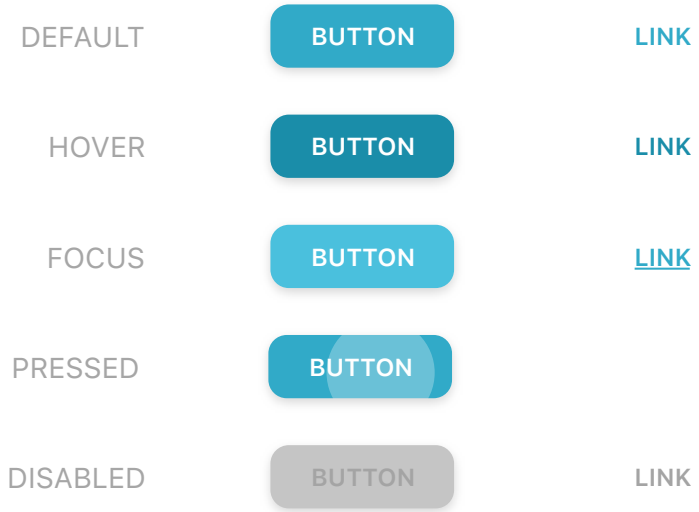
Obrázek 86: Ghost button vs. klasické tlačítko z Flat designu.



Obrázek 87: Příklad tlačítek bez ikony/s ikonou a špatným/správným paddingem.

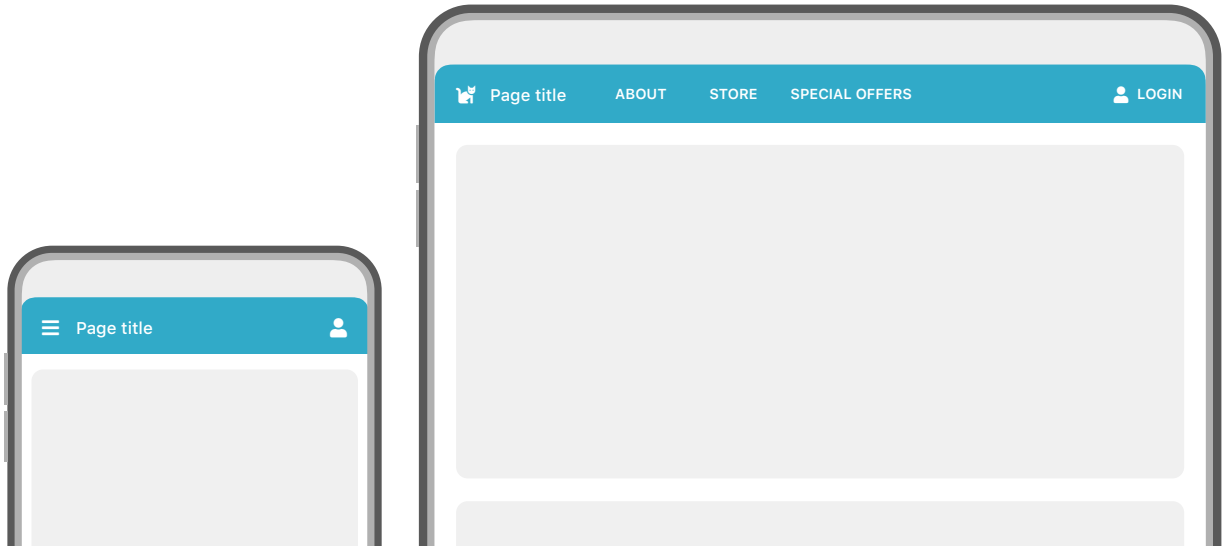


Obrázek 88: Tlačítka s různou viditelností, které jsou podle Vás nejlépe a nejhůře viditelná/rozeznatelná?



Obrázek 89: Jednotlivé stavy tlačítka.

- **Hover** Stav tlačítka po najetí kurzoru.
- **Focus** Stav tlačítka po označení například klávesnicí (důležité pro přístupnost).
- **Pressed** Stav tlačítka pokud je zmáčknuto (důležité pro dotykové obrazovky).
- **Disabled** Stav tlačítka, které je aktuálně zakázáno a nelze aktivovat



Obrázek 90: Navigation bar na zařízeních různé velikosti.

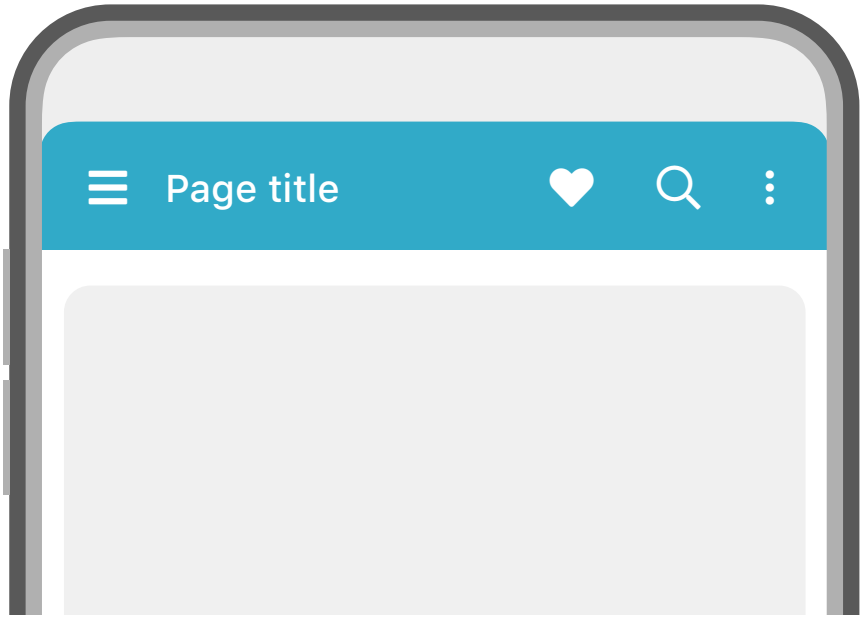
Navigation Bar

Navigation Bar, česky navigační lišta slouží jak už název napovídá k hlavní navigaci v rámci aplikace.

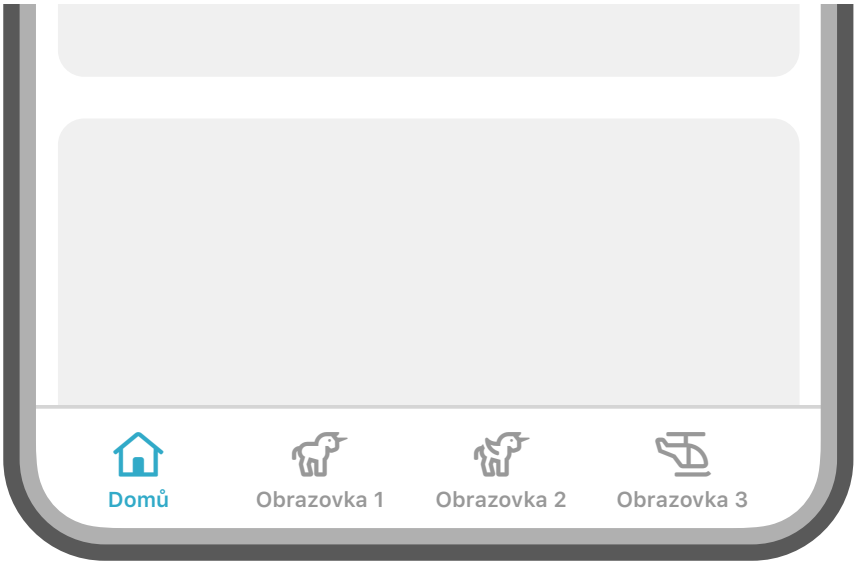
Navigační lišta má na různých platformách mnoho podob a názvů. Lišta bývá zpravidla umístěna na horním okraji a její pozice může být fixovaná – při scrollování zůstává připnuta na horní okraj obrazovky. Navigační lišta může měnit svoji podobu na základě kontextu, obecně by však měla být konzistentní napříč aplikací.

V některých případech může navigační lišta obsahovat akce spojené s aktuální obrazovkou. Jedná se tak o spojení navigační lišty a lišty nástrojů (o ní si povíme později). Toto spojení je časté u mobilních telefonů a zařízení s menší obrazovkou.

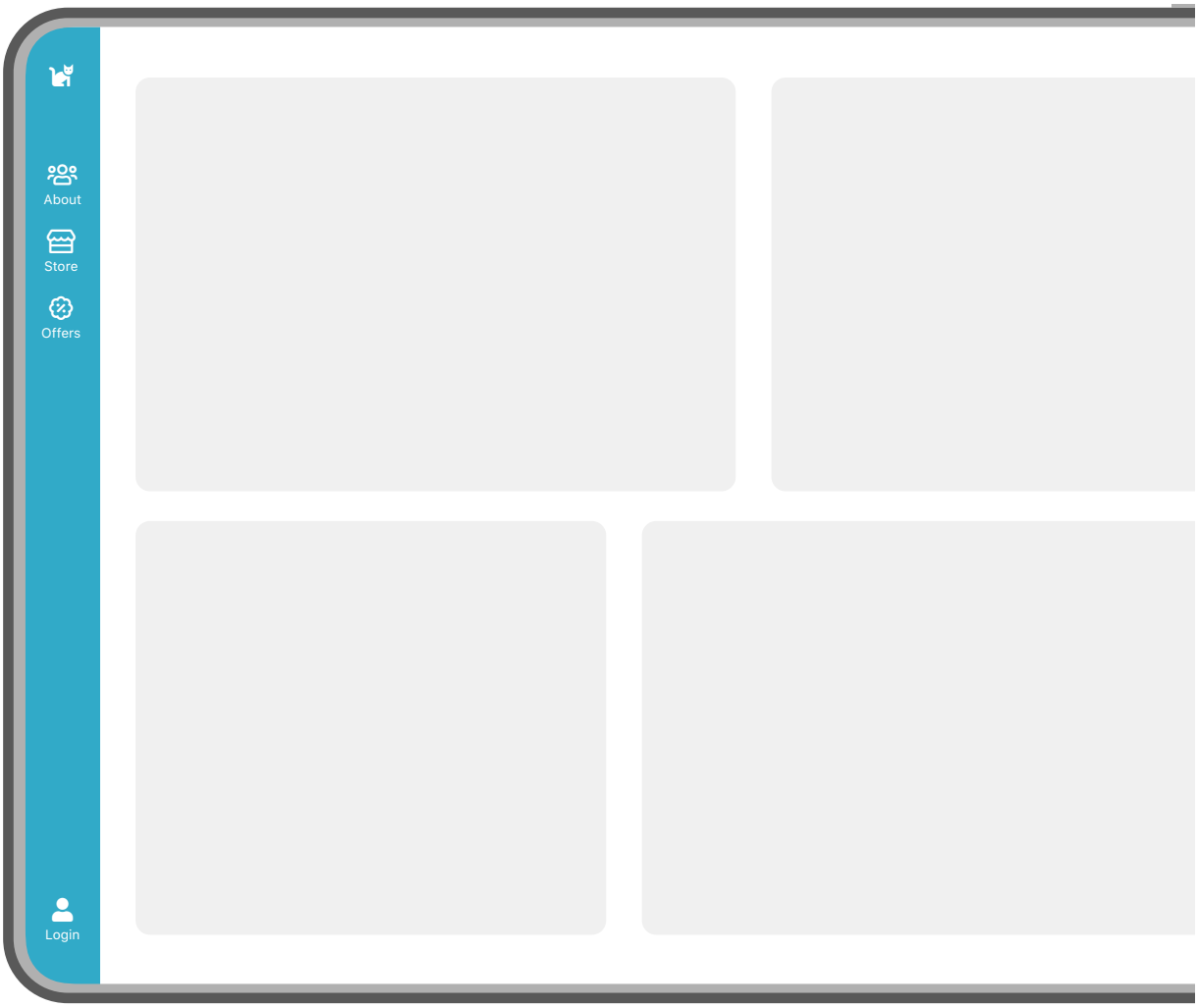
U širokých obrazovek je populární využívat vertikální navigační lišty. V případě vertikální navigační lišty je však často nutné využívat ikon (delší text se na šířku lišty nevleze). Tato varianta je viditelná na obrázku 93.



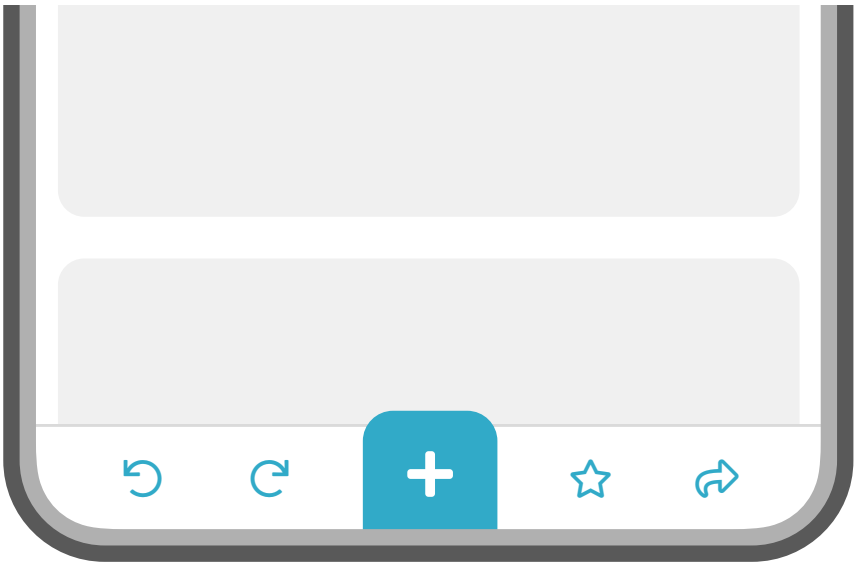
Obrázek 91: Spojení navigační lišty a lišty nástrojů.



Obrázek 92: Navigační lišta umístěná v dolní části obrazovky mobilního telefonu.



Obrázek 93: Vertikální varianta navigační lišty.



Obrázek 94: Lišta nástrojů na mobilním telefonu.

Tool Bar

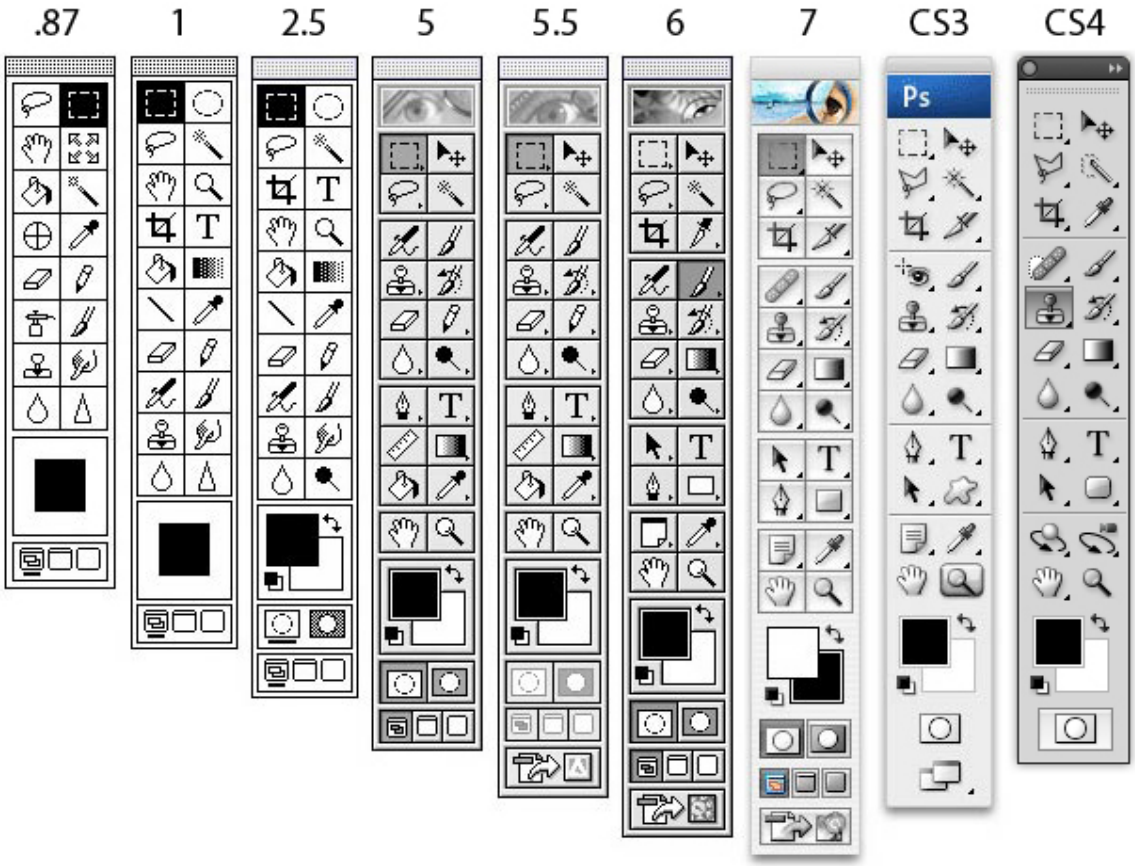
Tool Bar (česky lišta nástrojů) na rozdíl od navigační lišty poskytuje akce dostupné pro aktuální kontext aplikace. Akce jsou symbolizovány za pomoci textu a ikon (případně jakékoli kombinace).

Podobně jako navigační lišta je umísťován do horní části obrazovky, není to však striktním pravidlem. Především na telefonech je často možné lištu nástrojů vidět na spodním okraji obrazovky (obrázek 94)[49]. Na webu se pak lišta nástrojů může vyskytovat kdekoli na stránce.

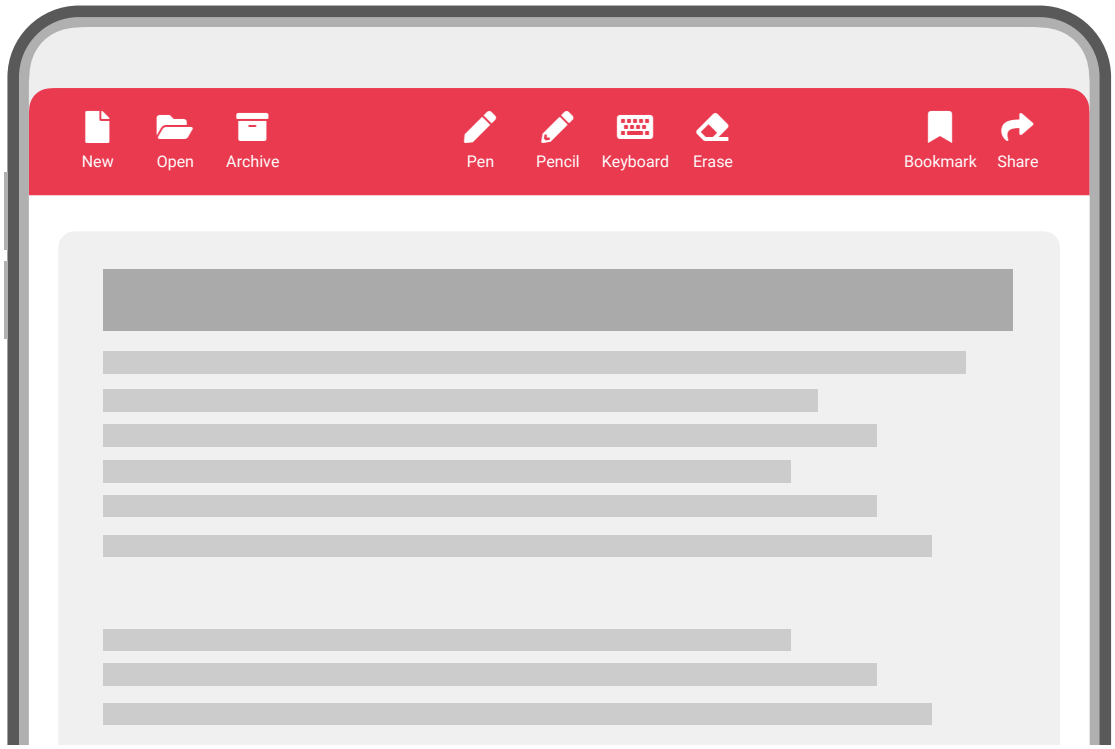
Časté je rovněž použití vertikální lišty nástrojů (příklad z programu Photoshop viditelný na obrázku 95).

V rámci lišty nástrojů je vhodné prvky rozdělovat do skupin po 3 až 7 prvcích dle Millerova pravidla uvedeného dříve (obrázek **??**).

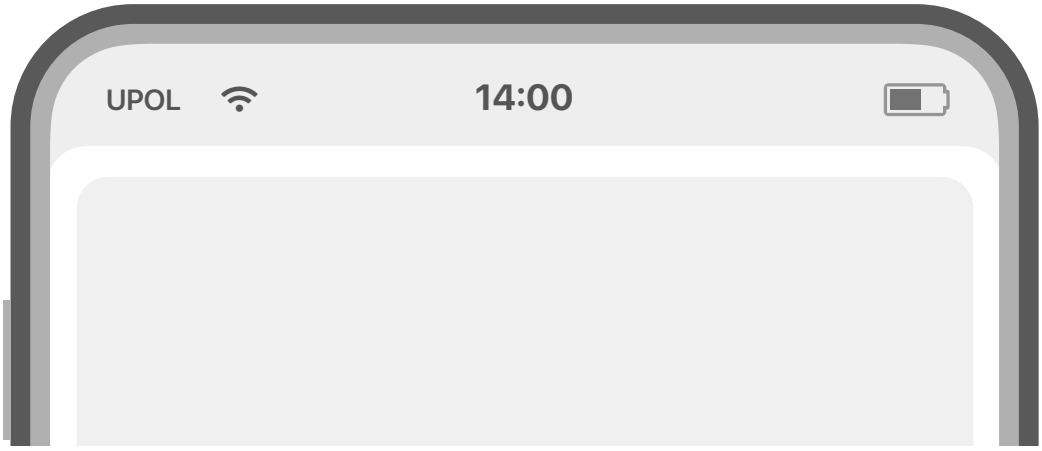
[49] Dolní pozice lišty nástrojů je postupně vytlačena navigační lištou umístěnou ve spodní části obrazovky mobilního telefonu



Obrázek 95: Vertikální lišta nástrojů z programu Photoshop.



Obrázek 96: Horizontální lišta nástrojů respektující Millerovo pravidlo.

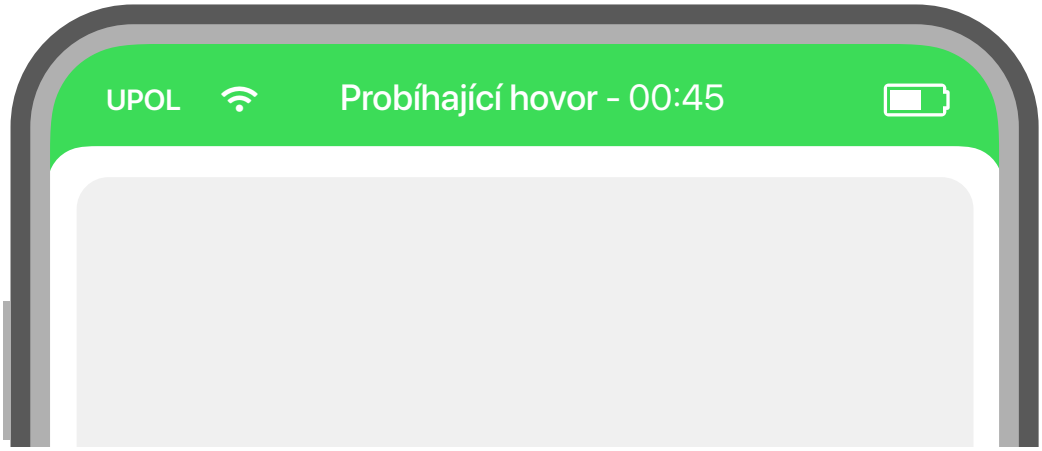


Obrázek 97: Stavová lišta na mobilních zařízeních.

Status Bar

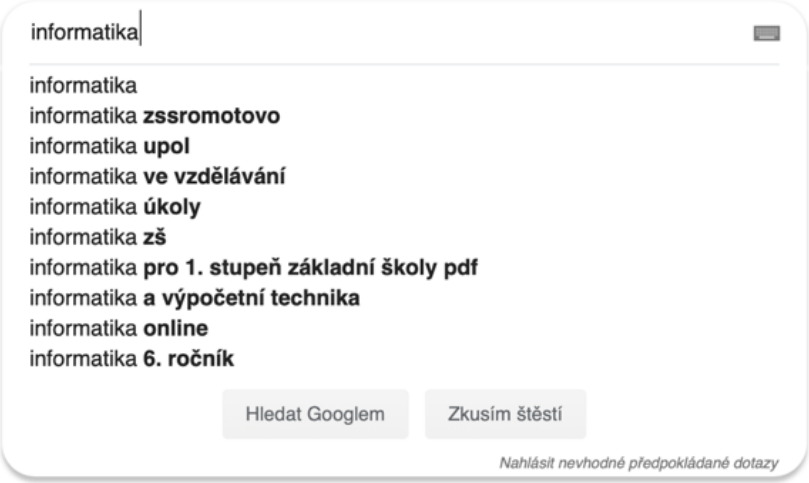
Status bar (česky stavová lišta) obsahuje aktuální informace o aplikaci/zařízení nebo probíhající operaci. Může se nacházet jak v horní (mobilní zařízení) tak dolní (web a desktop aplikace) části obrazovky. Okrajově může obsahovat i prvky sloužící k interakci.

Stavová lišta může měnit barvu na základě probíhající akce (obrázek 98), případně nemusí být zobrazena trvale (obrázek 99).



Obrázek 98: Stavová lišta během probíhajícího hovoru.

Obrázek 99: Stavová lišta, která je zobrazena pouze když je potřeba (náhled cíle odkazu).



Obrázek 100: Vyhledávací lišta s našeptáváním na webu Google.com.

Search Bar

Search Bar (česky vyhledávací lišta) slouží ke zprostředkování vyhledávání v rámci aplikace. V základu se jedná o modifikovaný textový vstup. Vyhledávací lištu však uvádím jako samostatný element z důvodu jeho rozšíření a dodatečné funkcionality.

Jak už bylo řečeno tvarově se jedná o klasický textový vstup, který standardně obsahuje ikonu a text. V některých případech je vyhledávací lišta spojena s potvrzovacím tlačítkem na zahájení vyhledávání, častěji však tuto funkci supluje klávesa Enter.

Důležitou přidanou funkcionalitou je našeptávání možných hledaných frází. Našeptávání výrazně zlepšuje použitelnost tohoto elementu a snižuje zátěž na paměť uživatele.

Lišta vyhledávání se může nacházet kdekoli v aplikaci, nejčastěji však najde své místo v rámci navigační lišty nebo lišty nástrojů.

Literatura

[1] Wilbert O. Galitz. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley; 3 edition, 2007.

[2] Google. Material design guidelines. https://material.io/design.

[3] Google. Understanding navigation. https://material.io/design/navigation/understanding-navigation.html.

[4] Nielsen Norman Group. The aesthetic-usability effect. https://www.nngroup.com/articles/aesthetic-usability-effect/.

[5] Nielsen Norman Group. Flat ui elements attract less attention and cause uncertainty. https://www.nngroup.com/articles/flat-ui-less-attention-cause-uncertainty/.

[6] Nielsen Norman Group. Serif vs. sans-serif fonts for hd screens. https://www.nngroup.com/articles/serif-vs-sans-serif-fonts-hd-screens.

[7] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 1st edition edition, 2013.

[8] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders; 3 edition, 2014.

[9] Everett N McKay. *UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication*. Morgan Kaufmann; 1 edition, 2013.

[10] Martin Michálek. *Vzhůru do (responzivního) webdesignu*. Martin Michálek - Vzhůru dolů, 2018.

[11] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1994.

[12] Donald A. Norman. *The Design of Everyday Things*. Basic Books, revised and expanded edition edition, 2013.

[13] Shan Shen. Make sense of rounded corners on buttons. https://uxdesign.cc/make-sense-of-rounded-corners-on-buttons-dfc8e13ea7f7/.

[14] Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media; Second edition, 2011.

[15] Robin Williams. *The Non-Designer's Design Book*. Peachpit Press; 4 edition, 2014.

[16] Dennis L. Byrnes Wingfield, Arthur. *The psychology of human memory*. New York: Academic Press, 1981.

[17] Jon Yablonski. Laws of ux. https://lawsofux.com.