

Podrobná specifikace ročníkového projektu - Origamist

Martin Pecka

2010

Program Origamist si klade za cíl zjednodušit práci při vytváření návodů na skládání origami modelů. Hlavní práci by měl odvést při vytváření diagramů. Na diagramu je zobrazen skládaný list papíru tak, jak vypadá po určitých krocích. Origamist bude za tvůrce návodu hlídat geometrii papíru, takže tvůrce jen bude zadávat příkazy, kde a jak papír ohnout, a program tyto ohyby znázorní a vytvoří z nich další diagram. Vytvářené diagramy budou obsahovat pomocné prvky pro lepší orientaci (různé stínování různých vrstev papíru, zvýraznění oblasti či zvětšený výřez).

Aplikace bude mít dvě části - tvůrčí a prohlížečí. Prohlížečí část bude fungovat jako applet a bude sloužit především ke snadnému prohlížení vytvořených návodů.

Tvůrčí část bude samostatná JNLP aplikace, která nabídne potřebné ovládací prvky pro tvorbu návodů, jejich načítání, ukládání a export. Návodů by měly jít exportovat primárně do SVG, další možnosti budou např. PDF nebo PNG. Při tvorbě diagramu se bude program snažit pomocí různých heuristik ulehčit práci (např. autodetekce symetrie modelu a automatické nabízení provedení operace na odpovídajících místech).

Společné rysy obou částí

- program bude plně lokalizovatelný, já vytvořím Čj a Aj locales
- obě části budou sdílet uživatelská nastavení, instalované locales atd.
- společná nastavení
 - jazyk programu
 - jazyk pro návody

Prohlížeč

- Java applet
- zpracovávané parametry předané při spuštění:
 - **file** URL - prohlížeč zobrazí jediný návod specifikovaný v předaném URL a nenabídne žádnou možnost zobrazení jiných návodů
 - **files** URL1 URL2 URL3 ... - prohlížeč zobrazí URL1 a nabídne možnost zobrazení ostatních modelů specifikovaných v URL
 - **files** DIR_URL - pokud je obsah adresáře zobrazitelný přes HTTP, zobrazí první model z daného adresáře a nabídne možnost zobrazení ostatních modelů z daného adresáře
 - **files** listing.xml - prohlížeč zobrazí soubory definované v listingu (viz Formát souborů) včetně metainformací
 - **recursive** - pokud je specifikován a **files** ukazuje na adresář, nabídne modely i z podadresářů
 - **startupMode** mode - mode = (page|diagram) - mód, ve kterém je prohlížeč spuštěn
- ještě nejsem rozhodnut, zda by měl prohlížeč při zobrazování více souborů (vyjma formy s listing.xml) všechny modely stáhnout a vytáhnout z nich metainformace, které potom zobrazí, nebo jestli jen zobrazit názvy souborů a jednotlivé modely stahovat až na vyžádání
 - kdyby se všechny modely stahovaly hned na začátku, šlo by třeba i zobrazit jejich náhledy a metainformace apod., ale zase by toto řešení bylo neefektivní, kdyby bylo specifikováno opravdu hodně souborů
 - naopak, kdyby se modely stahovaly až na vyžádání, bylo by to efektivní řešení, ale zase by uživatel viděl jen jména souborů
 - ideálním řešením by bylo používat vždy listing.xml, ale je jasné, že pro jednodušší nasazení je to zbytečný overhead
 - možná zkusit použít HTTP/1.1 Content-Range a natáhnout od každého modelu jen metadata? Tohle bych raději řešil až uvidím, jak velké budou výsledné soubory a zda by to vůbec mělo smysl
- módy prohlížení
 - „papír“ - zobrazí návod tak, jak by se exportoval - zachová počet sloupců a řádků na stránku
 - „jeden diagram - jedna stránka“ - zobrazí jeden krok návodu přes celou zobrazovací oblast
- možnost zoomu
- pokud je definováno více souborů k zobrazení, rozhraní nabídne možnost pro výběr jiného souboru
- možnost zobrazení detailů zobrazeného modelu (autor, název, popis...)
- možnost stažení / exportu modelu, upravení uživatelských nastavení programu

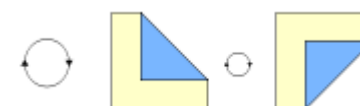
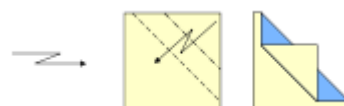
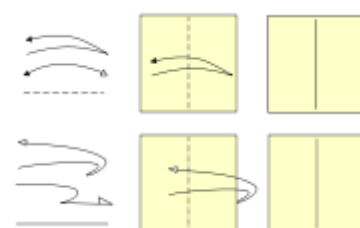
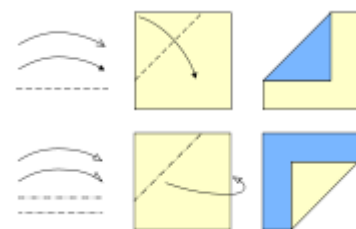
- tisk modelu bych raději řešil upozorněním, že model bude exportován do PDF, odkud se dá snadno vytisknout
- možnost nastavit jazyk návodu (obsahuje-li více jazyků) - defaultně jazyk používaný v rozhraní
- návrh GUI

Ikonové menu - export, download, typ zobrazení, zoom, další / předchozí diagram, nastavení programu, jazyk návodu	
File browser, pokud je k zobrazení více souborů	
Informace o zobrazeném modelu / popis aktuálního kroku při pohledu stránka=diagram	

Editor

- JNLP aplikace
- práce s modely
 - vytvořit nový
 - otevřít stávající (z disku, z URL)
 - export do SVG, PNG, PDF
- model půjde editovat pouze pokud k tomu autor vydá povolení (viz. formát souboru)
 - viz diskuze u formátu souboru na téma zašifrování/přidání hashe - bez nich by po manuálním zásahu do souboru editace šla povolit
- při vytváření nového modelu se program zeptá na následující údaje
 - informace o modelu
 - jméno a homepage autora (uloží se do nastavení, popř. by se mohlo načíst z Environmentu)
 - rok vydání modelu (předvyplněn na aktuální rok)
 - Název modelu (více jazyků)
 - URL originálu (nepovinné)
 - licence - název + buď URL nebo text licence
 - povolení, co lze s modelem dělat - editace, export
 - náhled - buď PNG nebo možnost automatického vygenerování (tj. vytvoří se náhled s názvem modelu a vyobrazením posledního kroku návodu) - program náhled dle potřeb zmenší na nějakou max. velikost (100px x 100px)
 - krátký popis modelu (více jazyků)
 - dlouhý popis modelu (více jazyků)
 - jaký papír použít pro model
 - půjdou zvolit jednotky: cm, inches (další?)
 - rozměry papíru - vlastní, nebo předdefinované (A4, Letter, A4/čtverec, Letter/čtverec...)
 - gramáž papíru
 - poznámka k druhu papíru (více jazyků)
 - běžné základy - existuje několik standardně a hojně využívaných základů, ty by také bylo vhodné nabízet
 - každý základ může vyžadovat přesné rozměry papíru / přesný poměr stran
 - možné rozšíření - manažer základů - u každého modelu by byla možnost „toto je základ“, pak by program tyto základy nabízel také

- základ bude obyčejný model (s trochu pozměněnou strukturou souboru, aby šly zachytit požadavky na rozměry papíru), takže bude možné po vytvoření modelu jednotlivé kroky základu „rozložit“ a např. posledních několik odmazat
 - barva přední / zadní strany papíru
 - možné rozšíření - libovolná textura
- jaký papír použít pro návod (tj. na jak velký papír se bude exportovat návod)
 - rozměry a barva pozadí jako u papíru pro model
 - počet sloupců / řádků, do kterých se budou kreslit diagramy
 - možné rozšíření - defaultně se budou „políčka tabulky“ zaplňovat zleva doprava a shora dolů, mohlo by jít změnit (ale to asi má pramalé praktické využití)
- možnosti editovaného modelu
 - upravit metainformace o modelu (viz. vytváření nového modelu)
 - bude třeba být opatrný vzhledem ke změně velikosti papíru, když už je model rozpracovaný
 - pokud to nebude složité, pak bych ji povolil, ale možnost zakázat změnu velikosti (resp. spíš poměru stran) mi přijde celkem málo omezující
- operace na modelu (obrázky z http://en.wikipedia.org/wiki/Origami_techniques)
 - *mountain/valley fold* - přehnutí papíru spodní strana ke spodní / vrchní k vrchní
 - půjde nastavit úhel - defaultně 180°
 - půjde nastavit počet ovlivněných vrstev
 - *mountain/valley fold and unfold* - vytvoření přehybu
 - půjde nastavit počet ovlivněných vrstev
 - *thunderbolt fold* - cik-cak sklad
 - vznikne spojením *mountain* a *valley*
 - převrácení modelu (přední strana dolů, spodní strana nahoru)
 - otočení modelu podle osy uživatel-monitor
 - několik přednastavených úhlů (90°, 180°), ale i uživatelský úhel



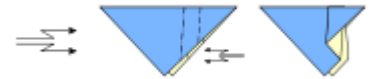
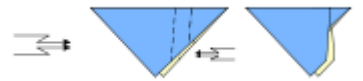
- *Pull* - vytažení papíru, rozložení skladů

- rozloží vybrané sklady



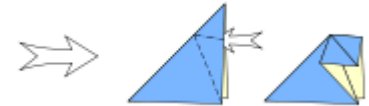
- *Inside/outside crimp fold* - cik-cak dovnitř/ven

- nejdříve se vytvoří obyčejné přehyby a pak se k nim přiřadí tento sklad



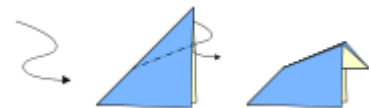
- *Open* - otevřít

- nejdříve se vytvoří obyčejné přehyby a pak se k nim přiřadí otevření



- *Inside/outside reverse fold* - vnitřní/vnější sklad

- půjde nastavit počet ovlivněných vrstev



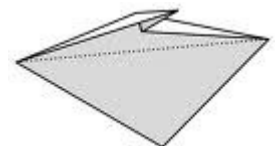
- *Repeat action* - opakovat akci

- uživatel vybere, které sklady se mají opakovat a v jakých oblastech modelu (buď program heuristikou správně pozná, které části kde zopakovat, nebo je uživatel tupě zopakuje, ale pak programu řekne, aby ukázal jen jednu z několika akcí a ostatní nahradil opakovátkem)



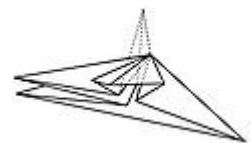
- *rabbit fold* - zaječí sklad

- jde složit na cípu papíru



- *squash fold* - rozplácnutí

- jde složit na cípu s alespoň 2 vrstvami



- možné rozšíření - rotace modelu o libovolný 3D úhel

- sklady na papíru

- tj. čáry, které zůstanou na papíře vidět po složení a rozložení přehybu

- sklady budou blednout s přibývajícím vzdáleností od kroku, kdy vznikly, ale vždy budou aktivní a bude možné kterýkoli sklad kdykoli znovu zvýraznit (popř. mu nastavit stejnou viditelnost „až do odvolání“)

- určování, odkud kam povede přehyb

- jde o to, aby program jednoznačně určil startovní a koncovou pozici přehybu a také vrstvy, které přehyb ovlivní

- pozice by neměla záviset na měřítku (tj. zvětšení papíru by ji nemělo ovlivnit)
- při výběru bodu by měly být existující sklady „magnetické“ („síla magnetu“ odpovídá tomu, jak moc již sklad zeslábl; průsečíky budou magnetické ještě trochu víc)
- jediná reprezentace, která mě napadla, je přiřadit každé straně papíru float/double 0-1 a používat jako souřadnice tyhle relativní
 - určitě se bude hodit uložit ještě směrový vektor skladu
 - zatím jsem si nedokázal, že takováto reprezentace opravdu postačí na všechny operace, které chci s modelem provádět
- možné vylepšení - drag&drop ohýbání papíru, stačilo by vzít myši cíp papíru a automaticky by se nabízely možné formy přehybů
- reprezentace skladů
 - teoreticky není nutné sklady nijak reprezentovat, protože vyplynou z operací, které s papírem provedeme
 - nicméně si teď myslím, že by bylo vhodné sklady nějak explicitně reprezentovat (kvůli snížení výpočetní náročnosti)
 - zde mě napadá reprezentace taková, že by každý sklad byl definován množinou úseček (tu stačí spočítat jednou, když sklad vznikne, pak se již nikdy nemění), identifikátorem kroku, kdy byl vytvořen, a „směrem“ (*valley/mountain*)
- reprezentace kroků
 - krok bude mít své pořadové číslo
 - ale raději bych mi přiřadil i jednoznačné ID, kdybych v budoucnu zavedl podporu pro mazání jiného než posledního kroku (což ne vždy musí jít, skoro bych řekl, že to půjde málokdy) popř. vložení kroku mezi dva další (to by bylo možné, například vložit krok s obrázkem, kde se programově nic neděje)
 - krok bude mít svůj popis (více jazyků)
 - krok bude sestávat z uspořádaného seznamu operací, každá z nich může mít nastaveno, zda se pro ni má zobrazit značka (např. právě skrývání kroků, které zaštití opakovátko)
 - kroky půjde slučovat a rozdělovat (např. pro příjemnou práci se základy) - při sloučení by program mohl zvýraznit operace ve slučovaných krocích a dát na výběr, které z nich skrýt/zobrazit
 - nabízí se otázka, co s popisky - dal bych možnosti: ponechat první, ponechat poslední, sloučit popisky do jednoho
- vykreslovací algoritmus
 - tady bude ještě třeba odvést kus práce
 - přemýšlím, zda oddělit vykreslování od kontroly, zda je požadovaný krok validní (křížení papíru apod.), nebo zda to nechat v jedné třídě (tj. pokusit se daný krok vykreslit a když to nepůjde nedovolit ho)

- vychytávky
 - heuristika na odhalení symetrických částí a automatické nabízení opakování kroku na všech částech
- ovládání
 - základní ovládání bude myší (vybírání bodů, volba nástrojů) a klávesnicí (volba nástrojů)
 - klávesové zkratky by bylo vhodné soustředit do jedné části klávesnice, spíš bych preferoval levou
 - zoom - Ctrl+kolečko myši, klávesové zkratky
 - bude potřeba skrývat vrchní vrstvy (aby šel vyznačit sklad, který tyto vrstvy zakrývají) - kolečko myši, klávesové zkratky
 - přeskočit mezi editací diagramu a popisku - F2
 - následující diagram (vytvořit další krok) - Tab (Shift+Tab - předchozí diagram)
 - smazat krok - Del, Esc (je pěkně dosažitelný v levé části, ale mohl by uživatele mást - stejně ale před smazáním kroku vyskočí potvrzovací dialog)

Formát souborů

- Soubor s modelem
 - pokud se v některé verzi změní interpretace některého příkazu, bude obsahovat proceduru (jednosměrnou) na konverzi ze starší verze na novější
 - ukládání vždy jen v nejnovější verzi

- XML formát

```
<origami version="integer:verze_souboru">
  <author>
    <name>Jméno autora</name>
    <homepage>volitelně URL adresa autorovy homepage</homepage>
  </author>
  <name>Název modelu</name>
  <name lang="jazyk">
    Název modelu v jiném jazyce (nepovinný)
  </name>
  <year>Rok vydání modelu</year>
  <shortdesc>Krátký popis modelu</shortdesc>
  <shortdesc lang="jazyk">
    Krátký popis modelu v jiném jazyce (nepovinný)
  </shortdesc>
  <description>Popis modelu</description>
  <description lang="jazyk">
    Popis modelu v jiném jazyce (nepovinný)
  </description>
  <license>
    <name>Název licence, pod kterou je model vydán</name>
    <homepage>URL, kde je text licence k nalezení</homepage>
    <content>
      volitelně text licence, pokud není vystavena na webu
    </content>
  </license>
  <original>
    volitelně URL vedoucí na originální návod, ze kterého byl
    tento vytvořen
  </original>
  <permissions>
    <!-- Povolení, co je možné s modelem dělat -->
    <permission>export</permission>
    <permission>edit</permission>
  </permissions>
  <thumbnail>
    <image type="image/png">Base64 obrázek</image>
  </thumbnail>
  <paper>
    <size>
      <width>20</width>
```

```

        <height>30</height>
        <unit>cm</unit>
    </size>
    <color>
        <background>#ffffff</background>
    </color>
    <!-- Definice matice, do které se budou kroky zakreslovat -->
    <cols>3</cols>
    <rows>3</rows>
</paper>
<model>
    <paper>
        <size>
            <width>20</width>
            <height>20</height>
            <unit>cm</unit>
        </size>
        <color>
            <foreground>#ffffff</foreground>
            <background>#aaaaaa</background>
        </color>
        <weight>Gramáž papíru v g/m²</weight>
        <note>Poznámka k druhu papíru</note>
        <note lang="jazyk">
            Poznámka k druhu papíru v jiném jazyce (nepovinná)
        </note>
    </paper>
    <steps>
        <!-- Pro každý krok jeden záznam <step> -->
        <step id="1">
            <description>Popisek kroku</description>
            <description lang="jazyk">
                Popisek kroku v jiném jazyce (nepovinný)
            </description>

            <!-- Definují, kolik řádků a sloupců má diagram
                zabírat -->
            <colspan>1</colspan>
            <rowspan>1</rowspan>

            <zoom>
                Pokud bude zoom jiný než v předchozím kroku,
                uvede se (defaultně 100)
            </zoom>

            Zde bude seznam operací provedených v tomto kroku
            nebo
            <image type="image/png">Base64 obrázek</image>

```

```

        </step>
        <step id="2">
        ...
        </step>
        ...
    </steps>
</model>
</origami>

```

- zvažuji možnost přidání nějakého „hashe“, který by zaručil fungující možnost zákazu editace (editace by byla chráněná heslem) - jen vůbec nevím, jak do toho
 - napadlo mě přidat k souboru, kde autor zamýšlí ochranu, nějaký digitální podpis, ale pořád nevidím, jak bych zabránil tomu, aby „útočník“ soubor ručně upravil (např. smazal podpis) a přemluvil program, aby editaci povolil
- Soubor se seznamem souborů, které má zobrazit prohlížeč
 - XML formát

```

<listing>
  <files>
    <file src="URL souboru">
      <author>
        <name>Jméno autora</name>
        <homepage>
          volitelně URL adresa autorovy homepage
        </homepage>
      </author>
      <name>Název modelu</name>
      <name lang="jazyk">
        Název modelu v jiném jazyce (nepovinný)
      </name>
      <year>Rok vydání modelu</year>
      <shortdesc>Krátký popis modelu</shortdesc>
      <shortdesc lang="jazyk">
        Krátký popis modelu v jiném jazyce (nepovinný)
      </shortdesc>
      <license>
        <name>
          Název licence, pod kterou je model vydán
        </name>
        <homepage>
          URL, kde je text licence k nalezení
        </homepage>
      </license>
      <original>
        volitelně URL vedoucí na originální návod,
        ze kterého byl tento vytvořen
      </original>
    </file>
  </files>
</listing>

```

```

        <thumbnail>
            <image type="image/png">Base64 obrázek</image>
        </thumbnail>
    </file>
    <file src="URL2">
        ...
    </file>
    ...
</files>
<categories>
    <category name="Název kategorie" id="ID">
        <name lang="jazyk">Název v jiném jazyce</name>
        <files>...</files>
        <categories>...</categories>
    </category>
    ...
</categories>
</listing>

```

- všechny vnitřní položky <file> jsou nepovinné, parametr src povinný je
- prohlížeč zobrazí ve výpisu souborů ke každému modelu všechny informace, které o něm listing poskytuje
- pokud je definována sekce <categories>, prohlížeč zobrazí modely v každé kategorii nějakým vhodným způsobem