

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Filip Pekárek

### Návrh nábytku

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: Mgr. Jiří Kocanda  
Studijní program: informatika, programování

2006

Na tomto místě bych chtěl poděkovat svému vedoucímu bakalářské práce panu Mgr. Jiřímu Kocandovi za podnětné připomínky a vedení.

Dále bych chtěl poděkovat panu Mgr. Martinu Marešovi za připomínky k algoritmu výpočtu nářezových plánů.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 1.8.2006

Filip Pekárek

# Obsah

<b>1</b>	<b>Úvod</b>	<b>6</b>
1.1	Cíl . . . . .	6
1.2	Motivace . . . . .	7
1.3	Obsah dalších kapitol . . . . .	7
<b>2</b>	<b>Návrh</b>	<b>9</b>
2.1	Název . . . . .	9
2.2	Vzhled aplikace . . . . .	9
2.3	Požadavky . . . . .	10
2.4	Části programu . . . . .	11
<b>3</b>	<b>Algoritmus výpočtu nářezových plánů</b>	<b>12</b>
3.1	Analýza problému . . . . .	12
3.2	Návrhy . . . . .	13
3.3	Řešení . . . . .	16
<b>4</b>	<b>Implementace</b>	<b>18</b>
4.1	Výběr vývojových nástrojů . . . . .	18
4.2	Struktura programu . . . . .	18
4.3	Hlavní menu . . . . .	19
4.4	Návrh skříně . . . . .	22
4.5	Prostorový model . . . . .	23
4.6	Nářezové plány . . . . .	24
4.7	Sklad materiálu . . . . .	25
4.8	Seznam součástí . . . . .	26
4.9	Vnitřní datové struktury . . . . .	26
4.10	Odchylky od funkční specifikace . . . . .	27
<b>5</b>	<b>Dokumentace</b>	<b>29</b>
5.1	Funkční specifikace . . . . .	29
5.2	Uživatelská příručka . . . . .	29

5.3	Instalační příručka . . . . .	30
5.4	Programátorská příručka . . . . .	30
<b>6</b>	<b>Závěr</b>	<b>31</b>
6.1	Praktické zkušenosti . . . . .	31
6.2	Další vývoj . . . . .	32
6.3	Splnění cíle . . . . .	33
	<b>Literatura</b>	<b>34</b>

Název práce: Návrh nábytku  
Autor: Filip Pekárek  
Katedra (ústav): Ústav formální a aplikované lingvistiky  
Vedoucí bakalářské práce: Mgr. Jiří Kocanda  
e-mail vedoucího: Jiri.Kocanda@mff.cuni.cz

Abstrakt: Program pro návrh nábytku je aplikace, která umožňuje v uživatelsky přítulném grafickém prostředí vytvářet návrh skříní. Navrhovaná skříň odpovídá požadavkům zadávaných uživateli. Na základě těchto požadavků a seznamu dostupného materiálů program vypočítá optimální nářezový plán. Aplikace také umožňuje uživatelům spravovat vlastní sklad materiálu. Programem vygenerované nářezové plány je možné ručně editovat a tisknout. Seznamy součástí potřebných pro sestavení navrhované skříně jsou také generovány a je možné je vytisknout.

Klíčová slova: návrh, nábytek, skříň, nářezový plán

Title: Furniture design  
Author: Filip Pekárek  
Department: Institute of Formal and Applied Linguistics  
Supervisor: Mgr. Jiří Kocanda  
Supervisor's e-mail address: Jiri.Kocanda@mff.cuni.cz

Abstract: The furniture design program is an application, which allows users to design cases in user friendly graphical interface. Designed cases conform to user's requirements. On the basis of these requirements and the list of available materials, program computes the optimal cutting plan. The application also allows to control the user's material store. Generated cutting plans can be edited and also printed. The list of component parts, necessary for designed case mounting, can be generated. This list can be printed.

Keywords: design, furniture, case, cutting plan

# Kapitola 1

## Úvod

### 1.1 Cíl

Cílem této bakalářské práce je vytvoření programu pro návrh skříní. Program by měl umožňovat navrhovat skříně a vypočítávat optimální nářezové plány z materiálu, které má uživatel na skladě.

Při návrhu skříní je kladen důraz na uživatelsky příjemné grafické prostředí. Skříní by měla být vytvářena postupným zadáváním jednotlivých parametrů uživatelem. Měla by být možná následná změna jednotlivých již zadaných parametrů skříně. Během návrhu by uživatel měl mít k dispozici prostorový model odpovídající navrhované skříní. Program by měl být schopen ukládat a následně nahrávat navrhované skříně do/z vlastního formátu.

Program má uživateli nabídnout nástroj pro správu vlastního skladu materiálu. Položky tohoto skladu mají být použity při výpočtu nářezových plánů.

Program má umět na základě vlastností generované skříně a skladu uživatele materiálu vypočítat optimální nářezové plány vhodných materiálů. Program by měl uživateli umožnit vygenerované nářezové plány upravovat a tisknout.

Uživatel by měl mít k dispozici seznam potřebných částí navrhované skříně. Program by měl být schopen tento seznam vytisknout.

## 1.2 Motivace

Cíl bakalářské práce vychází z nápadu spojení dvou doposud oddělených částí.

Jednou je navrhování prostorového modelu skříně podle požadavků uživatele. V současné době existují programy poskytující prostorové modelování různých tvarů. Generování prostorového modelu skříně v těchto programech není náročné. Příkladem takového programu je „3D Studio Max“. Tento program je špičkový nástroj na poli 3D modelování, animací, vytváření vizuálních efektů a všeobecně zastřešující designovou problematiku vizualizace počítačové grafiky. Je to program pro profesionální vytváření grafických aplikací.

Druhou částí je výpočet nářezových plánů. Existující komerční programy uživateli nabízejí zadat sestavu částí a desky materiálu, z kterých mají být tyto části nařezány. Po zadání všech potřebných nastavení jsou generovány nářezové plány. Příkladem takovýchto programů jsou „Optimik“ nebo „Merick Calc“.

Snahou programu „Furniture“ je spojit tyto dvě myšlenky dohromady a poskytnout uživateli nástroj pro generování optimálních nářezových plánů, jehož součástí je prostorový model navrhované skříně.

## 1.3 Obsah dalších kapitol

Druhá kapitola se zabývá návrhem a analýzou navrhované aplikace. Kapitola popisuje tvorbu funkční specifikace nutné pro správný postup při vývoji nové aplikace. Jsou zde popsány veškeré kroky a rozhodnutí provedené před zahájením implementace.

Třetí kapitola popisuje problémy vzniklé při generování nářezových plánů. Jsou zde popsány autorovy nápady řešení a nakonec je zde popsán implementovaný algoritmus výpočtu nářezových plánů.

Čtvrtá kapitola popisuje vlastní implementaci programu. Popisuje výběr nástroje, ve kterém byla aplikace vyvíjena. Tato kapitola zachycuje vnitřní strukturu programu. Také popisuje jednotlivé položky hlavního menu, návrh hlavních vnitřních struktur použitých v programu a vzniklé odchylky od původní funkční specifikace.

V páté kapitole jsou stručně popsány dokumenty, které jsou nutnou součástí softwarového díla. Důležitá je formální úprava dokumentů a jejich funkčnost.

Závěrečná kapitola shrnuje nové praktické zkušenosti, možnosti dalšího vývoje a vylepšení vytvořené aplikace. Na závěr hodnotí splnění cíle bakalářské práce.



# Kapitola 2

## Návrh

### 2.1 Název

Vytvořená aplikace slouží k navrhování skříní, a proto se nabízí název „Návrh nábytku“, nebo přesněji „Návrh skříní“. Pro zjednodušení se nabízel kratší název „Nábytek“. Nakonec byl název změněn na anglické univerzální slovo „Furniture“, který bude dále používán.

### 2.2 Vzhled aplikace

V dnešní době již nejsou programy vytvářeny jen za účelem vysoké funkčnosti, jak tomu bývalo dříve, ale také jejich vzhled a jednoduchost ovládání je velmi důležitá. Většina moderních programů poskytuje velice dobré uživatelské rozhraní a snaží se uživatelům nabídnout co největší volnost v jeho nastavení. Právě vzhled aplikace totiž zanechá v uživateli první dojmy o programu a pokud jsou tyto dojmy pozitivní, vede to uživatele k dalšímu zkoumání, co vlastně ta dobře vypadající aplikace umí. Naopak pokud je vzhled nevydařený, může to uživatele odradit a dát mu impuls poohlédnout se po něčem jiném. Proto jsem považoval vzhled aplikace za velice podstatnou část programu a snažil jsem se jí věnovat velkou pozornost. Veškerá menu aplikace by měla být dostupná přes klávesové zkratky.

## 2.3 Požadavky

### Vytvoření skříně

Uživatel si bude moci navrhnout libovolnou pravoúhlou skříň. Uživatel bude mít možnost zadávat jednotlivé parametry skříně a také je v případě potřeby měnit a upravovat podle svých představ. Navrhované modely skříní bude možné ukládat a nahrávat do/z souboru.

### Model skříně

Program na základě zadaných požadavků uživatelem bude zobrazovat odpovídající prostorový model skříně. Tento model skříně bude reagovat na veškeré změny provedené uživatelem do stávající struktury skříně. Modelem bude možné otáčet kolem svislé a vodorovné osy a bude ho možné zvětšovat a zmenšovat.

### Sklad s materiálem

Program uživateli poskytne nástroj pro evidenci materiálu. Ve skladě bude mít uživatel svůj materiál. Při výpočtu nářezových plánů bude použit materiál z tohoto skladu.

### Nářezové plány

Aplikace bude schopna vypočítat optimální nářezové plány vhodných materiálů z uživatelského skladu. Generované nářezové plány bude možné ručně upravovat, ukládat a také zpětně nahrávat do/z souboru.

### Seznam součástí

Uživatel bude mít k dispozici seznam jednotlivých částí navrhované skříně. Seznam bude také volitelně obsahovat rozměry lišt potřebných pro polepení všech hran skříně. Generovaný seznam bude možné tisknout.

### Licence

Vytvořený program bude volně šiřitelný a zdrojové soubory budou volně k dispozici.

### Platforma

Program nemusí být přenositelný. Program bude vyvíjen pouze pro platformu Windows.

## 2.4 Části programu

Aplikace „Furniture“ se skládá z několika hlavních částí, které jsou na sobě závislé. Jednotlivé části a jejich úkol je popsán níže.

### Zadávání parametrů

Tato část programu má za úkol poskytnout uživateli prostředek pro zadávání vlastností navrhované skříně. Musí být možné editovat a upravovat každou část skříně samostatně.

### Prostorový model

Úkolem této části programu je co nejpříjemněji zobrazovat aktuální strukturu navrhované části. Zobrazovaný model by měl být schopen reagovat na požadavky uživatele. Například přibližování a otáčení.

### Sklad materiálu

Dalším požadavkem na program „Furniture“ je schopnost spravovat uživatelův sklad materiálu. Tento sklad má být následně použit při generování nářezových plánů.

### Nářezové plány

Optimální výpočet nářezových plánů navrhované skříně z uživatelova materiálu ve skladě má za úkol čtvrtá část aplikace. Program má také umožňovat vygenerované nářezové plány upravovat a tisknout.

### Seznam součástí

Poslední část aplikace má za úkol generovat seznam částí, z kterých se navrhovaná skříň skládá. Generované seznamy je možné prohlížet a tisknout.

# Kapitola 3

## Algoritmus výpočtu nářezových plánů

### 3.1 Analýza problému

Program Furniture má poskytovat nástroj pro výpočet optimálních nářezových plánů uživatelova materiálu. Jedná se o nářezání co nejmenšího počtu desek materiálu tak, aby po provedení řezů vznikly všechny části navrhované skříně. Na žádost uživatele budou odřezky vzniklé po nářezání všech částí přidány zpět do uživatelova skladu a v budoucnu s nimi bude program počítat. Je důležité aby vzniklé odřezky byly co nejvíc využitelné. To znamená, aby byly málo členité a měly velkou plochu.

Uživatel má na skladě desky různých rozměrů a materiálů. Každá skříň navrhovaná uživatelem se skládá z několika částí. Na začátku návrhu je jich pět. Levá, pravá, horní, spodní a zadní stěna. Každá s těchto vyjmenovaných stěn tvoří samostatnou část a má vlastní délku  $d_i$ , šířku  $s_i$ , tloušťku  $t_i$  a svůj materiál  $m_i$ . V průběhu práce s programem, kdy uživatel přidá řekněme patro, přibude do seznamu částí další položka, která má rozměry  $d_j, s_j, t_j, m_j$ .

Hledaný algoritmus tedy dostane na vstup množinu jednotlivých částí nebo spíše desek, ze kterých se skříň skládá. Poté je na vstupu seznam všech desek materiálu, který má uživatel k dispozici.

Nejdůležitějšími parametry jsou tloušťka a materiál. Proto vstupní data nejprve trochu upravíme. Jelikož část z materiálu  $m_i$  a o tloušťce  $t_i$  se dá nařezat pouze z desky  $(m_j, t_j)$ , která je ze stejného materiálu a má stejnou tloušťku ( $m_i = m_j, t_i = t_j$ ). To znamená, že seznam všech částí skříně bude rozdělen do menších seznamů, kde v jednom seznamu budou

pouze díly stejné tloušťky a stejného materiálu. Nyní můžeme předpokládat, že na vstupu budeme mít části skříně a díly materiálu stejné tloušťky a materiálu.

Výstupem algoritmu má být opět seznam, ale tentokrát už pouze použitých desek z uživatelova skladu. Na jednotlivých deskách budou znázorněny nářezové plány. Použitých desek má být co nejméně a desky mají být co nejvíc využity. To znamená, že vzniklý odpad má být co nejmenší. Každá deska má být optimálně nářezána, aby provedením všech řezů materiálů vznikly všechny části navrhované skříně.

V praxi se používají dva typy řezů:

Gilotinový.

Každý provedený řez musí být proveden po celé délce materiálu v jednom směru. Během řezání nelze změnit směr. Každým řezem vzniknou dvě nové části.

Negilotinový.

Během řezání je možné zastavit a změnit směr řezu o devadesát stupňů vlevo nebo vpravo.

Program „Furniture“ je určen pro návrh skříní z dřevěných materiálů. V praxi se při práci s dřevěnými materiály používají pouze gilotinové řezy, protože negilotinové jsou technicky náročné. Proto je v aplikaci implementován výpočet optimálních nářezových plánů pomocí gilotinových řezů.

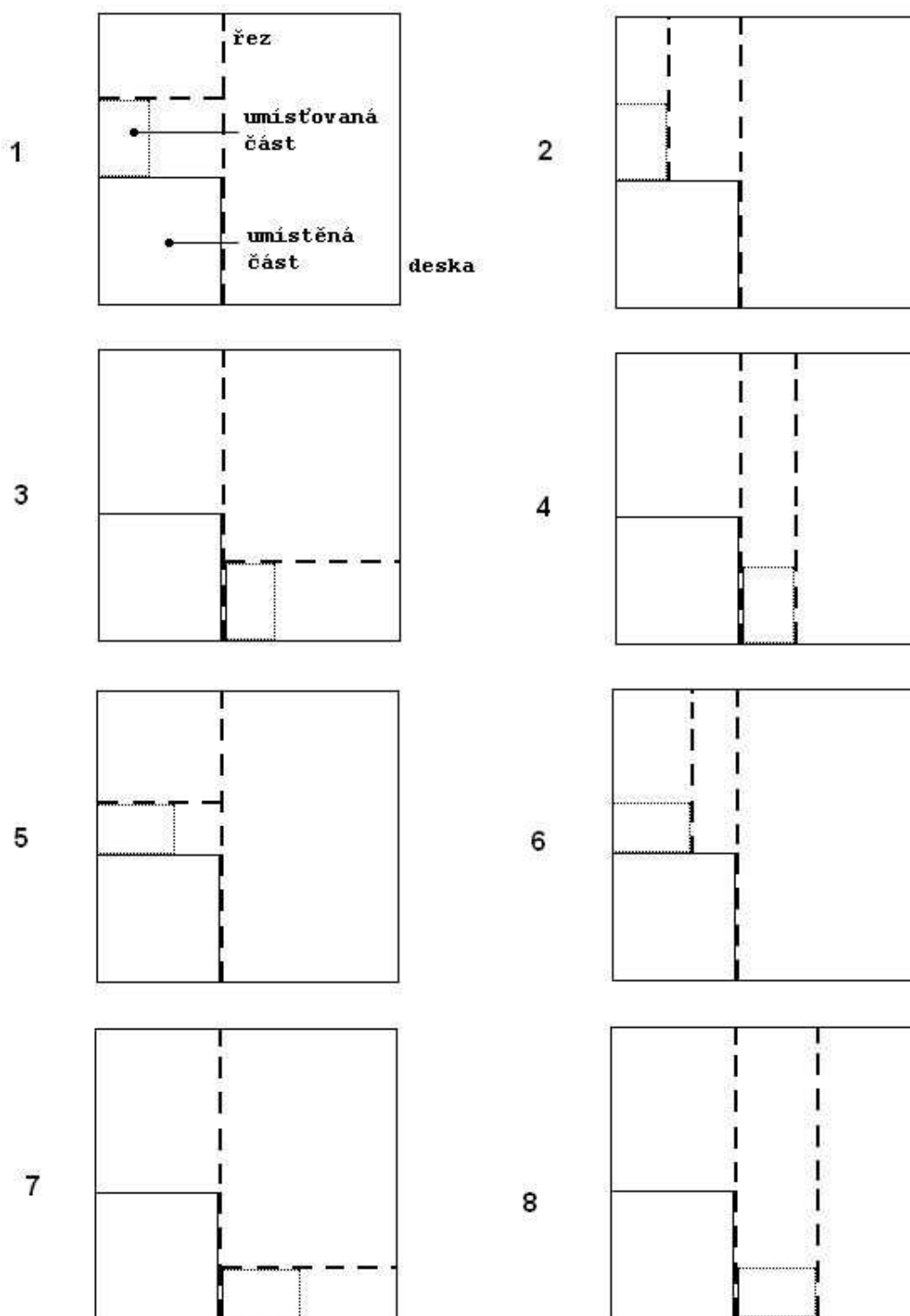
## 3.2 Návrhy

Vzniklý problém patří do kategorie NP-úplných problémů, což jsou problémy, jejichž výpočtová složitost je větší než polynomiální.

První a velmi intuitivní řešení je vyzkoušet všechny možné kombinace umístění jednotlivých částí. Vzhledem ke kategorii problému, do kterého zapadá je toto řešení velmi neefektivní a časově i paměťově příliš náročné. Při malé velikosti vstupního seznamu částí by doba výpočtu byla přijatelná, ale už při počtu dílů kolem sedmi by výpočet trval příliš dlouhou dobu. Navíc by zde také musela být funkce, která by ohodnotila aktuální výpočet a následně vybírala nejlepší řešení. Tento algoritmus nebyl

z důvodu výpočetní složitosti implementován.

Dalším nápadem bylo nějakým způsobem optimalizovat postup popsaný výše. Použití heuristiky mělo přinést vylepšení. Tím vylepšením bylo seřadit si vstupní seznam dílů skříně podle velikosti. V našem případě podle obsahu, konkrétně od největšího k nejmenšímu. V takovém seznamu samozřejmě mohlo být několik dílů stejného obsahu, ale to nevadí. Algoritmus by poté pracoval obdobně jako v předchozím případě. Rozdíl by byl pouze v tom, že by se díly ze seznamu braly postupně ze předu (tedy podle velikosti) a zkoušely by se všechny možnosti jak je uspořádat. Také díky gilotinovým řezům se v jednom kroku algoritmus rozvětví na mnoho nových větví. Viz Obrázek 3.1, který znázorňuje jeden krok výpočtu navrhovaného algoritmu. Je zde znázorněna deska materiálu, do které je již jedna část skříně umístěna a další část v pořadí je k ní přidávána. Opět zde musí být pomocná funkce, která by všechny výsledky výpočtu ohodnocovala a nakonec by vybrala to nejlepší řešení. Tento algoritmus není optimální, a proto také nebyl implementován.



Obrázek 3.1: Jeden krok algoritmu

### 3.3 Řešení

Jako optimální řešení se ukázal algoritmus založený na myšlence dynamického programování. Dynamické programování je odvětví optimalizace. Dynamickým programováním rozumíme takový postup, kdy vyřešíme zadanou úlohu nejprve pro zadání menší velikosti a pak nalezená řešení zkombinujeme dohromady, abychom získali řešení původní úlohy.

Základní myšlenka je následující. Budeme mít funkci, která dostane jako vstupní parametry množinu částí skříně, které chceme rozřezat z desek materiálu. Výstupem této funkce bude opět množina, ale tentokrát minimálních obsahů, do kterých se vejdou všechny vstupní části. Výstupní množina obsahů je optimální, a proto odpovídá myšlence dynamického programování. Během výpočtu funkce jsou generovány obsahy, které jsou do výstupní množiny přidávány podle relace  $\ll$ . Tato relace má následující předpis.  $A = \ll (A, B)$  když deska  $A$  má rozměry  $x_a, y_a$  a deska  $B$   $x_b, y_b$  a platí  $x_a \leq x_b$  &  $y_a \leq y_b$ , nebo  $x_a \leq y_b$  &  $y_a \leq x_b$ . Pokud neplatí ani jedna z těchto podmínek, jsou dané obsahy neporovnatelné a do výstupní množiny budou přidány oba. Obě podmínky relace  $\ll$  jsou testovány pouze pokud materiál, s kterým je právě počítáno lze libovolně otáčet. Pokud s používaným materiálem nelze otáčet, je testována pouze první podmínka. Nastavení zda se s materiálem smí nebo nesmí otáčet odpovídá nastavení orientované kresby materiálu. Pokud je v programu nastaveno, že daný materiál má orientovanou kresbu, nelze s ním libovolně otáčet a je s ním vždy počítáno pouze ve své původní poloze.

Implementovaný algoritmus vypadá následovně. Uvedu příklad, kdy na vstupu máme čtyři díly, které si očísľujeme 1 až 4. Na začátku zavoláme funkci  $\ll (1, 2, 3, 4)$ , která nám vrátí množinu minimálních obsahů, které obsahují všechny čtyři části. Funkce  $\ll$  s parametry (1,2,3,4) bude počítat následovně. Nejprve rozdělí svůj výpočet na menší a jednodušší výpočty a na každou podúlohu zavolá rekurzivně sama sebe. Po každém zavolání funkce  $\ll$  bude vrácen minimální obsah. Funkce všechny vypočítané obsahy porovná podle relace popsané výše a z těchto obsahů vygeneruje výslednou minimální optimalizovanou výstupní množinu obsahů. Pro názornost uvedu přesně postup pro náš vzorový vstup (1,2,3,4). Funkce spočítá všechny možné kombinace jak rozložit vstupní množinu.



1 | 2 3 4  
 2 | 1 3 4  
 3 | 1 2 4  
 4 | 1 2 3

1 2 | 3 4  
 1 3 | 2 4  
 1 4 | 2 3

Toto jsou všechny možné kombinace. Nám jde pouze o výpočet minimálních nutných obsahů, proto například rozložení  $1\ 2\ | \ 3\ 4$  je stejné s rozložením  $2\ 1\ | \ 3\ 4$  a  $2\ | \ 1\ 3\ 4$  je stejné s  $3\ 1\ 4\ | \ 2$ . Funkce  $\ll$  tedy vždy rozdělí výpočet na menší části a poté skládá vrácené výsledky dohromady. Funkce volá rekurzivně sama sebe až do té doby, kdy jako vstupní parametr dostane pouze jednu část. V tento okamžik vrací dvouprvkovou množinu. Prvky této množiny mají stejné obsahy a rozdíly jsou pouze v rozměrech. První je část předaná parametrem a druhá je stejná část otočená o devadesát stupňů. Pokud je vstupní část čtverec, je výstupní množina pouze jednoprvková.

Během výpočtu výše popsaného algoritmu je několik výpočtů počítáno vícekrát, a proto je potřeba jednotlivé dílčí výsledky ukládat a před každým výpočtem je potřeba nejprve zkontrolovat zda vstupní kombinace již nebyla jednou počítána. Pokud ano, vrátí se již jednou spočítaný optimální výsledek.

Jsme ve fázi, kdy nám první zavolání funkce  $\ll (1, 2, 3, 4)$  vrátilo množinu minimálních obsahů. Pokud množina obsahuje více prvků, jsou všechny minimální a vzájemně neporovnatelné. Musíme si tedy jedno z řešení vybrat sami. Jako kritérium výběru byl zvolen maximální rozdíl vzniklý při odečtení rozměrů řezané desky a desky představující nařezané části skříně vypočítaného obsahu. Tím pádem nám po nařezání všech částí skříně vznikne co největší možný souvislý kus odpadu, který bude lépe využitelný při případném dalším řezání. Algoritmus během výpočtu ukládá jednotlivé dílčí výsledky a pamatuje si jakému obsahu odpovídá jaké rozložení částí. V okamžiku, kdy máme vybraný nejvhodnější obsah, stačí pouze projít uložené výpočty a dospějeme k výslednému optimálnímu nářezovému plánu materiálu.

# Kapitola 4

## Implementace

### 4.1 Výběr vývojových nástrojů

Protože při vývoji aplikace byl kladen velký důraz na uživatelsky přívětivé grafické prostředí, vhodné vývojové prostředí by mělo poskytovat rychlý grafický nástroj pro návrh oken, různých dialogů pro komunikaci s uživatelem a podporu grafické knihovny OpenGL. Na základě předchozích zkušeností vývojáře byl pro implementaci vybrán jazyk C++.

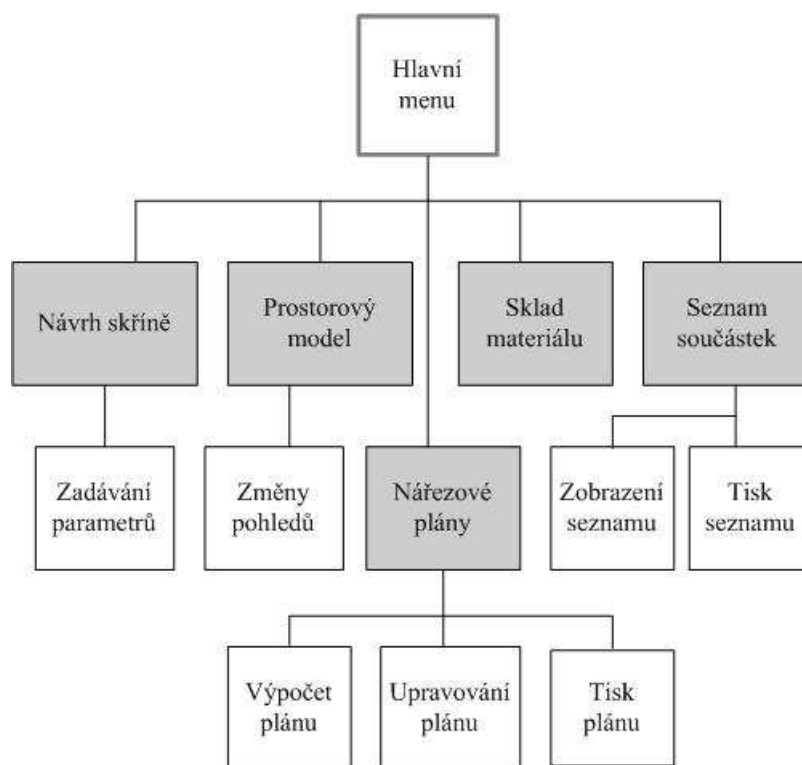
Jako vhodné nástroje splňující výše uvedené podmínky se ukázaly:

1. Borland C++ Builder
2. Microsoft Visual Studio 2003

Obě varianty jsou přibližně ekvivalentní. Z praktických důvodů (dostupnost prostředí a zkušenost vývojáře s prostředím) bylo zvoleno druhé prostředí tj. MS Visual Studio 2003.

### 4.2 Struktura programu

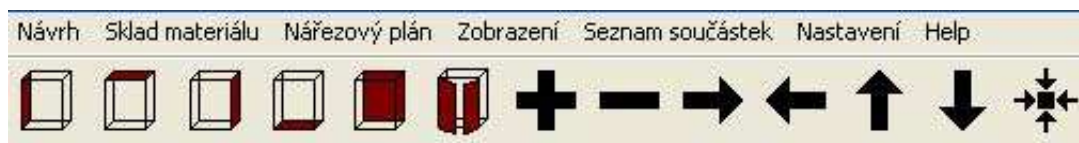
Program je rozdělen do čtyř hlavních modulů znázorněných na obrázku 4.1 šedě. Moduly jsou logicky rozděleny podle funkce, kterou mají v celém systému splňovat. Pro názornost jsou moduly pojmenovány podle svého úkolu v programu. Každý z modulů je implementován zvlášť v samostatném souboru. Rozhraní mezi těmito hlavními čtyřmi moduly poskytuje hlavní menu programu „Furniture“ viz Obrázek 4.2. Úkoly a funkčnost jednotlivých modulů jsou popsány v následujících kapitolách.



Obrázek 4.1: Struktura programu

### 4.3 Hlavní menu

Hlavní menu je součástí hlavního okna aplikace. Poskytuje propojení mezi všemi moduly programu „Furniture“ na obrázku 4.1. Pomocí něho se uživatel může dostat ke všem prostředkům a nástrojům, které aplikace nabízí. Toto menu je implementováno v samostatném a hlavním souboru celého systému a je zobrazeno na obrázku 4.2.



Obrázek 4.2: Hlavní menu

Menu má dvě části. Jednak první řádek, kde jsou položky pro přepínání mezi hlavními nástroji programu. Uživatel tyto položky může vyvolat buď klasickým kliknutím myši nebo také pomocí klávesových zkratk. Všechny klávesové zkratky se používají stejně. Nejprve uživatel stiskne

a drží stisknutou klávesu ALT. Následně se v každé položce nabídky podtrhne jedno písmenko. Po jejím stisku společně se stisknutou klávesou ALT se vyvolá příslušná nabídka. Druhý řádek nabídky obsahuje klasická tlačítka. Při spuštění programu obsahuje nabídka pouze první řádek. V okamžiku kdy uživatel začne nový nebo otevře již stávající návrh skříně, zobrazí se i druhý řádek nabídky.

### **Návrh**

Tato položka menu uživateli umožňuje vlastní návrh skříně. Uživatel si může vybrat, zda chce vytvořit úplně nový návrh skříně, otevřít již uložený návrh, uložit právě probíhající návrh skříně, ukončit stávající návrh a nebo ukončit celý běh programu.

### **Sklad materiálu**

Výběrem této položky se uživateli zobrazí nové okno s materiálem. Obsluhu programu přebírá modul „Sklad materiálu“.

### **Nářezový plán**

Kliknutím na tuto položku menu se uživateli zobrazí nové okno, které má na starosti veškerou práci ohledně nářezových plánů. V tento okamžik veškerou práci přebírá modul „Nářezové plány“

### **Zobrazení**

Tato položka nabídky je pouze pro nastavení zobrazení oken programu při návrhu skříně. Uživatel si může zvolit, zda chce zobrazit okno znázorňující členitost skříně (strom součástí) a okno, kde se vypisují vlastnosti vybraných částí.

### **Seznam součástí**

Výběrem této položky je řízení programu přepnuto do modulu „Seznam součástí“ popsaného v kapitole 4.7.














### **Nastavení**

Pomocí této položky mohou uživatelé nastavit hodnoty tloušťky a odsazení desek používaných při nových návrhů skříní.

## Help

Po stisku tohoto tlačítka se uživateli zobrazí další dvě nabídky, kde jedna položka vyvolá dialog se základními informacemi o programu a druhá spustí kompletní uživatelskou příručku (help) k programu „Furniture“.

V druhém řádku nabídky jsou vytvořena tlačítka, která mají co nejlépe a nejpřesněji znázorňovat svou funkci a jejich význam je následující:

-  V okně vlastnosti zobrazí parametry levé stěny skříně.
-  V okně vlastnosti zobrazí parametry horní stěny skříně.
-  V okně vlastnosti zobrazí parametry pravé stěny skříně.
-  V okně vlastnosti zobrazí parametry spodní stěny skříně.
-  V okně vlastnosti zobrazí parametry zadní stěny skříně.
-  V okně vlastnosti zobrazí informace o zavírání skříně.
-  Přiblíží prostorový model skříně.
-  Oddálí prostorový model skříně.
-  Otočí modelem skříně vpravo podél svislé osy.
-  Otočí modelem skříně vlevo podél svislé osy skříně.
-  Otočí modelem skříně nahoru podél vodorovné osy skříně.
-  Otočí modelem skříně dolů podél vodorovné osy skříně.
-  Vrátí model skříně do původní polohy.

## 4.4 Návrh skříně

Tato část programu má na starosti zpracování požadavků uživatele týkajících se návrhu skříně. Jedná se o přidávání nových částí do skříně, zadávání nových rozměrů různých částí skříně, zobrazování a úprava již zadaných parametrů. Tento modul se také stará o ukládání a načítání návrhu z/do souboru na pevném disku uživatele. Toto ukládání je implementováno pomocí binárního zápisu do uživatelem zvoleného souboru. Takový soubor má příponu `fur`. Tento název byl zvolen tak, aby byl jedinečný a nemohl se krýt s nějakým již existujícím souborem a také aby připomínal název programu „Furniture“. Při instalaci systému je pomocí zápisu do registrů operačního systému uživatele s touto příponou asociovaná ikona. Tato asociace ikony skříně byla volena pro účel lepšího přehledu uživatele o souborech na svém počítači.

Při spuštění nového návrhu se uživateli zobrazí pouze prázdná skříň bez zavírání. Uživatel může měnit parametry jednotlivých stěn skříně a přidávat nové položky do skříně. Pokud je skříň prázdná a nemá žádné vnitřní členění, musí si uživatel při přidávání první nové části vybrat, zda chce přidat nové patro nebo přepážku. Pokud už je skříň členitá, platí, že do přepážek se přidávají patra a do pater přepážky.

Každou požadovanou změnu na navrhované skříni aplikace vyhodnotí a pokud je možná, provede ji a uživatele o tom informuje. Pokud není změna z nějakého důvodu možná, uživatele o tom také informuje a neprovede ji. V případě, kdy je změna jen z části nepřesná, zeptá se program uživatele, zda může automaticky upravit parametry. Pokud s tím uživatel souhlasí, upraví je, změní parametry a uživatele o tom informuje.

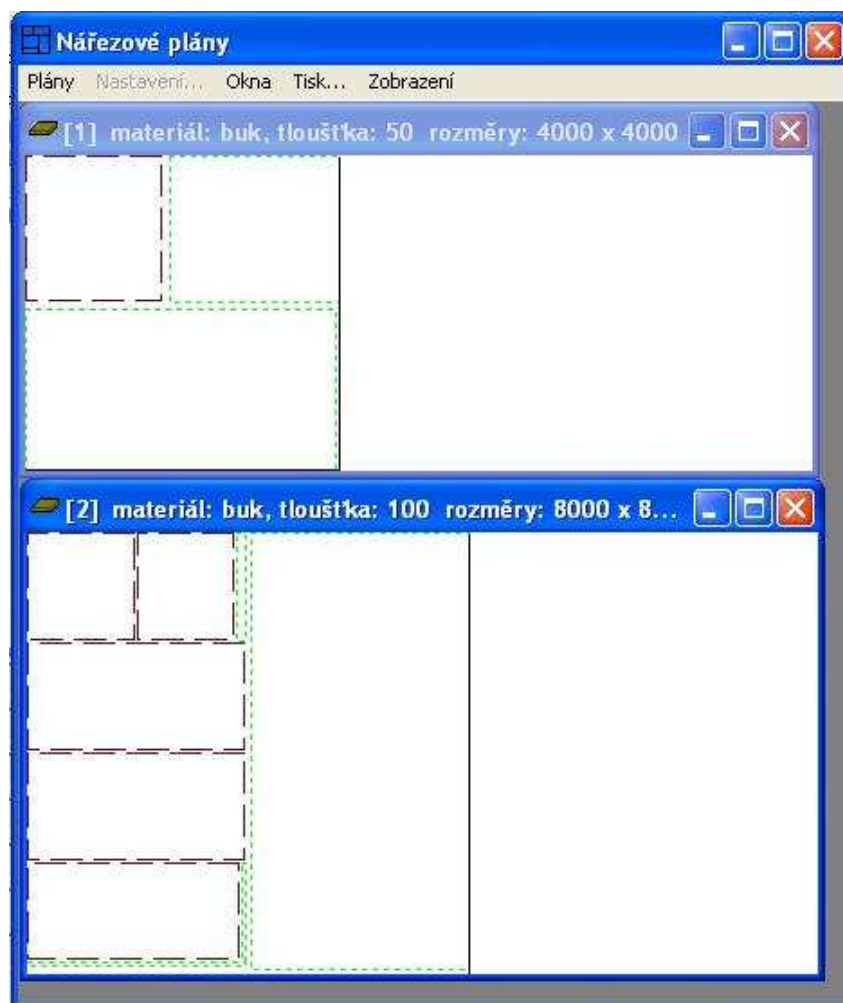
## 4.5 Prostorový model



Obrázek 4.3: Ukázka prostorového modelu

Na obrázku 4.3 je ukázka skříně z programu. Prostorový model odpovídá aktuálnímu nastavení parametrů skříně. Například na obrázku je skříně s jednou přepážkou, kde v druhém oddílu jsou tři patra. Pro lepší představu uživatele program umožňuje otáčení modelem kolem svislé i vodorovné osy. Model je možné přibližovat a oddalovat. Možnost otáčení skříně kolem dvou os se ukázalo postačující vzhledem k potřebám uživatele. Během testování programu se ale může stát, že se model dostane do takové pozice, kdy ani uživatel sám neví, kde se jaká část skříně na obrázku právě nachází. Pro tento případ bylo přidáno tlačítko, které vrátí prostorový model do původní pozice.

## 4.6 Nářezové plány



Obrázek 4.4: Nářezové plány

Tato část programu je algoritmicky nejnáročnější. Výpočet optimálních nářezových plánů byl implementován podle algoritmu popsaného v kapitole 3.3. Veškerá práce s nářezovými plány se odehrává v novém okně aplikace. Při výpočtu plánů se může stát, že bude potřeba použít více desek materiálu. Pro přehlednost je vhodné, aby byla každá použitá deska zobrazena ve vlastním okně. Při programování proto bylo použito MDI (Multiple document interface), což je technika správy oken v programech. MDI je také popisováno jako označení aplikací, které umožňují uvnitř sebe sama otevřít (a spravovat) v jednom okamžiku větší



množství dokumentů či souborů. Typickým příkladem MDI aplikací jsou některé textové editory, které umožňují pracovat s více dokumenty zároveň. Každá používaná deska materiálu je zobrazena v jednom MDI Child okně a vše je dohromady obsaženo uvnitř jednoho okna. Viz Obrázek 4.4.

Pokud je otevřen nový nebo již uložený návrh skříně, může si uživatel nechat vypočítat optimální nářezové plány. V první řadě aplikace vyhledá vhodné desky materiálu, které má uživatel na skladě. Pokud jich nebude dostatek, upozorní o tom aplikace uživatele a výpočet nebude proveden. Vygenerované nářezové plány je možno pomocí kliknutí myši jednoduše upravovat podle potřeb uživatele. Nářezové plány je také možno ukládat a nahrávat. Program nabízí dva typy zobrazení plánů. Jedním je „návrh“ a druhým jsou „kóty“. V prvním případě je plán vykreslen pouze pomocí barevných čar různých typů a uživatel může plán měnit. V druhém případě se dané části už měnit nedají a u každé části jsou pomocí kót zobrazeny jejich rozměry. Tento druhý typ je vhodnější pro tisk plánů, které aplikace také umožňuje. Co se týče nastavení nářezových plánů, může si uživatel zvolit tloušťku řezu, s kterou má algoritmus počítat. Tloušťka se zadává v milimetrech.

## 4.7 Sklad materiálu

Sklad uživateli umožňuje spravovat svůj materiál. Sklad je zobrazován v samostatném okně jako seznam. V jednom řádku je jedna položka skladu. Každý řádek je rozdělen do pěti sloupců, ve kterých jsou uvedeny jednotlivé parametry desek. Jsou to materiál, délka, šířka, tloušťka a počet.

Sklad materiálu je možné editovat. Do skladu je možné přidávat nové záznamy se sedmi přednastavenými dekory. Těmito dekory jsou buk, černý grafit, hrušeň, javor, olše, třešeň a světle šedá. Pro každý tento dekor je použita vlastní odpovídající bitmapa. Uživatel si do tohoto seznamu může přidat novou položku. Každá z těchto přidaných položek bude v programu reprezentována jednou univerzální bitmapou. Celý sklad je uložen v jednom souboru na pevném disku uživatele. Operace se souborem jsou prováděny binárně, takže vnější úprava například pomocí textového editoru není možná.

## 4.8 Seznam součástí

Posledním modulem je modul generující seznam součástí navrhované skříně. Generovaný seznam je vypsán do nového okna. Položky generovaného seznamu jsou jednotlivé části skříně. V každém řádku je jedna položka. Řádky jsou rozděleny do čtyř sloupců, v kterých jsou uvedeny jednotlivé parametry. Jsou to id, materiál, rozměry a počet. Podle uživateli volby jsou do seznamu přidány další záznamy, které znázorňují rozměry hranovacích pásek potřebných pro olepení předních hran skříně. Tyto pásy odpovídají rozměrům horní, spodní, levé, pravé stěny skříně a předním stěnám pater a přepážek skříně. Vygenerovaný seznam je možné vytisknout.

## 4.9 Vnitřní datové struktury

Pro program „Furniture“ bylo potřeba navrhnout dvě hlavní vnitřní struktury, které jsou popsány v této kapitole.

První z nich je datová struktura navrhované skříně. Každá navrhovaná skříň se skládá z hlavních pěti částí, kterými jsou spodní, horní, levá, pravá a zadní stěna. Poté je zde volitelný parametr, kterým je zavírání dveří. Vnitřní datová struktura pojmenovaná *TSkrin* tyto parametry obsahuje. Poté bylo potřeba nějakým vhodným způsobem reprezentovat vnitřní členitost skříně. Skříň může obsahovat pouze patro nebo přepážku.

Původní návrh byl pro každou z těchto dvou možností nadefinovat vlastní strukturu. Pro každou strukturu by bylo potřeba nadefinovat specifické operace pro práci s těmito strukturami. Například struktura pro patro by obsahovala parametry jako jsou poloha, tloušťka, šířka, výška, odsazení a seznam přepážek uvnitř daného patra. Struktura pro přepážku by obsahovala stejné parametry. Polohu, tloušťku, šířku, výšku a odsazení. Lišili by se pouze v seznamu pater obsažených v dané přepážce. Během návrhu těchto vyjmenovaných vlastností a návrhu všech potřebných funkcí se jako vhodnější ukázalo navrhnout pouze jednu univerzální datovou strukturu jak pro patro, tak pro přepážku. Výsledná implementovaná struktura je tedy pouze jedna jediná a obsahuje již zmíněné parametry. Navržená struktura je rekurzivní, protože pokud je daná část skříně dále členitá, obsahuje seznam prvků té samé struktury. Všechny implementované funkce struktury byly tomuto návrhu přizpůsobeny a

jsou univerzální. Detailnější programátorský popis dané struktury je obsažen v programátorské příručce.

Druhou vnitřní strukturou je datová struktura vypočítaných nářezových plánů. Navrhovaná struktura musela být rychlá a schopná reagovat na možné změny vzniklé uživatelským zásahem do vygenerovaných nářezových plánů. Tato struktura byla nakonec implementována jako seznam jednotlivých částí, kde každá část má své specifické vlastnosti jako jsou šířka, délka a absolutní souřadnice. Každá část také obsahuje ukazatele na všechny části ze seznamu, které s ní logicky souvisí.

## 4.10 Odchylky od funkční specifikace

Některé požadavky ve funkční specifikaci se v průběhu vývoje ukázaly jako zbytečné, a proto nebyly implementovány. Některé se ukázaly jako nepraktické a byly upraveny. Jedná se o následující požadavky:

1. Průvodce zadáváním parametrů skříně.
2. Drátěný a reálný model skříně.
3. Standardní hodnoty rozměrů skříně.

Ve funkční specifikaci systému je uvedeno, že parametry skříně budou zadávány pomocí průvodce. Tento průvodce měl mít několik kroků, které by postupně uživateli umožnili nastavit jednotlivé parametry skříně. Zadávání požadavků uživatelem je v aplikaci „Furniture“ implementováno natolik přehledně a intuitivně, že žádný pomocný průvodce nebylo potřeba implementovat. Pokud se i přesto uživatel dostane do nesnáží a neví jak má postupovat, součástí programu je kompletní uživatelská příručka, kde nalezne potřebné informace a postupy.

V programu je implementován pouze reálný prostorový model skříně. Požadavky na otáčení a přibližování byly implementovány. Reálný model umožňuje volit viditelnost jednotlivých částí skříně nezávisle na sobě. Je natolik propracovaný, že pro svou variabilitu a funkčnost nebylo nutné implementovat drátěný model skříně. U každé části navrhované skříně má uživatel nabídku a možnost volby, zda chce tuto část v modelu zobrazovat a nebo nezobrazovat. Je proto například možné skrýt zobrazení pravé stěny skříně a uživatel má tak možnost vidět navrhovanou skříň skrz pravou stěnu.

Ve funkční specifikaci je uvedeno, že rozměry skříně budou zadávány v jednotkách rozměru šanonů, což je 338 milimetrů. Tento nápad vznikl na základě rozměrů strojově vyráběných kancelářských skříní. Tam se právě takovýto rozměr používá. Pro větší komplexnost a variabilitu aplikace „Furniture“ při návrhu skříní byla jednotka rozměrů změněna na jeden milimetr. V celém programu „Furniture“ je základní jednotkou jeden milimetr.

# Kapitola 5

## Dokumentace

### 5.1 Funkční specifikace

Funkční specifikace zachycuje požadavky zadavatele softwarového díla specifikované v průběhu analýzy a návrhu aplikace. Na základě Funkční specifikace by mělo vzniknout samotné softwarové dílo. V závěrečných fázích vývoje aplikace slouží Funkční specifikace k ověření splnění požadavků zadavatele a k tvorbě funkčních testovacích scénářů.

Funkční specifikace slouží uživateli, který nemusí být vzdělaný v oblasti informačních technologií. Proto nezachází do technických detailů a pokud možno se vyhýbá používání příliš odborných pojmů.

### 5.2 Uživatelská příručka

Uživatelská příručka popisuje softwarové dílo z uživatelského hlediska. Cílem uživatelské příručky je co možná nejsrozumitelnějším způsobem seznámit uživatele s aplikací. Proto byla napsána formou klasického help souboru pro windows. Nicméně obsahuje popis všech vlastností aplikace a poskytuje uživateli detailní popis všech částí aplikace.

Pro názornost a možnost čtení tohoto dokumentu bez nutnosti spuštění aplikace Furniture je text příručky doplněn o ukázkové obrázky většiny oken aplikace.

Pro úplnost a uživatelovo pohodlí byl vytvořen jednoduchý a krátký návod jak navrhnout jednoduchou skříň, přidat materiál do skladu a vygenerovat nářezový plán.

## 5.3 Instalační příručka

Instalační příručka popisuje postup instalace softwarového díla, doplňujících součástí a popis potřebných nastavení systémových prostředí a dalších aplikací. Instalační příručka dále obsahuje popis kroků, které je nutno provést před prvním spuštěním aplikace.

Instalační příručka aplikace Furniture je velmi jednoduchá a krátká, jelikož celá instalace programu je také jednoduchá a je řízena pomocí průvodce.

## 5.4 Programátorská příručka

Programátorská příručka vysvětluje principy, na jejichž základě aplikace vnitřně funguje. Popisuje použité technologie, datový model, logiku rozdělení zdrojových kódů do modulů, logiku názvosloví identifikátorů apod. Součástí Programátorské příručky je i detailní Referenční příručka ke zdrojovým kódům aplikace.

Programátorská příručka je psaná formálním jazykem a v trpném rodu. Časté je používání anglických výrazů a někdy i infromatického žargonu. Je to způsobeno příliš rychlým vývojem informačních technologií, na který spisovná čeština nedokáže dostatečně rychle reagovat.

# Kapitola 6

## Závěr

### 6.1 Praktické zkušenosti

Během této bakalářské práce jsem se seznámil s metodikou a postupy potřebné pro správný vývoj nového softwarového díla. Naučil jsem se co vše je potřeba pro úplné a správné splnění požadavků na vytvářené dílo. Uvědomil jsem si nutnost a funkčnost veškeré dokumentace potřebné při vytváření programů.

Před programováním aplikace „Furniture“ jsem neměl zkušenosti s programováním grafických prvků jako je prostorový model skříně. Nejprve jsem proto musel hledat a zjišťovat které možné nástroje právě programování grafiky umožňují. Nakonec jsem jako nejvhodnější shledal OpenGL (Open Graphics Library), což je průmyslový standard specifikující multiplatformní rozhraní (API) pro tvorbu aplikací počítačové grafiky. Implementace OpenGL existují pro prakticky všechny počítačové platformy, na kterých je možno vykreslovat grafiku. Základní funkcí OpenGL je vykreslování do obrazového rámce (framebufferu). Umožňuje vykreslování různých základních primitiv (bodů, úseček, mnohoúhelníků a obdélníků pixelů) v několika různých režimech. Veškerá činnost OpenGL se řídí vydáváním příkazů pomocí volání funkcí a procedur (kterých OpenGL definuje cca 250). Dříve tomu tak nebylo, ale dnes už je OpenGL velmi dobře zdokumentováno i v českém jazyce, takže podle mého názoru začít s OpenGL může každý. Existují i spousty tutoriálů o práci v OpenGL. Více o OpenGL na [?].

V závěrečných fázích vývoje aplikace (tj. především ve fázi funkčního testování) byla aplikace testována jak autorem, tak několika dalšími uživateli.

Význam jednotlivých tlačítek a postupy při návrhu skříně se jevily jako přirozené a na první pohled jasné. Prostorový model skříně odpovídající právě navrhované skříně se ukázal jako velmi názorný a pro uživatele příjemný.

O průběhu výpočtu nářezových plánů uživatele informuje progress bar, takže uživatel má stále kontrolu nad programem. Následné upravování vygenerovaných plánů je také intuitivní pouze pomocí myši.

Praktické zkušenosti s programem Furniture ukázaly, že práce s programem je pro uživatele příjemná a intuitivní a tudíž splnila požadavek na přívětivost prostředí.

## 6.2 Další vývoj

V průběhu vývoje a při testování aplikace Furniture se objevilo množství zlepšovacích návrhů, které nebyly součástí funkční specifikace. Některé, pro správné fungování aplikace nevyhnutelné požadavky, byly do výsledné aplikace dodatečně implementovány. Zbývající, zejména zlepšovací návrhy, nebylo již z časových důvodů možné realizovat a jsou uvedeny v následujícím seznamu pro případ dalšího vývoje aplikace.

Možnosti dalšího vývoje:

- jiné než pravoúhlé tvary
- další materiály
- negilotinové řezy
- další typy nábytku



## 6.3 Splnění cíle

V rámci této bakalářské práce byla vytvořena aplikace Furniture sloužící pro pohodlné návrhy skříní a následné generování optimálních nářezových plánů. Důraz byl kladen na uživatelskou přívětivost.

Vývoj aplikace probíhal převážně podle zásad stanovených zadáním. Nejdříve byla vytvořena podrobná Funkční specifikace, na základě které byla naprogramována výsledná aplikace „Furniture“. K softwarovému dílu byla napsaná veškerá požadovaná dokumentace, tj. Uživatelská příručka s jednoduchým návodem, Instalační příručka a Programátorská příručka. Projektovou dokumentaci nahrazuje rozsáhlejší Bakalářská práce.

Vytvořená aplikace téměř přesně odpovídá Funkční specifikaci. Výjimkou je několik málo požadavků vyjmenovaných v 3.6, které se v průběhu vývoje ukázaly jako zbytečné a proto nebyly implementovány.

# Literatura

- [1] Microsoft corp. The MSDN library. Worl Wide Web,  
<http://msdn.microsoft.com>
- [2] Charles Petzold: *Programming Windows by Charles Petzold*,  
Microsoft Press, 1988.
- [3] Builder. World Wide Web, *Učíme se WinApi*,  
<http://www.builder.cz/art/cpp/winapi1.html>
- [4] Wikipedia. World Wide Web, *Dynamic programming*,  
<http://en.wikipedia.org>
- [5] World Wide Web, *OpenGL*,  
<http://www.opengl.org>
- [6] Nutsch W.: *Konstrukce nábytku*, Grada, 2003.