

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Program pro podporu výuky geometrie



2010

Vojtěch Zlámal

Anotace

Tato práce se zabývá vytvořením podpůrného programu Geomi pro výuku řezů krychle a jehlanu na gymnáziích. Čtenář je uveden do problematiky výuky řezů těchto těles na gymnáziích a pomůcek, které lze k tomuto účelu použít, např. počítačových programů. U nich je rozebrána potřebná funkcionalita a jsou diskutována již existující programová řešení. Druhá polovina práce je věnována popisu programu Geomi z uživatelského i programátorského hlediska jakožto alternativě k uvedeným řešením.

Zde bych rád poděkoval Mgr. Jiřímu Zaccalovi, Ph.D., vedoucímu mé práce, za podněty, cenné rady a čas, který mi věnoval. Dále bych poděkoval Mgr. Miroslavu Hnilicovi a Mgr. Davidu Martinákovi, jejichž poznatky z praxe mi pomohly ke kvalitnějšímu zpracování obsažené látky.

Obsah

1. Úvod	7
2. Problematika výuky řezů na gymnáziích	8
2.1. Výuka	8
2.2. Pomůcky	8
2.3. Pochopení učiva	9
3. Existující řešení	10
3.1. Webová aplikace pro výuku stereometrie	10
3.2. Výuka řezů těles Stereo	11
3.3. Cabri 3D	11
3.4. Dynamická geometrie v prostoru	12
3.5. Říše čísel – Matematika 2	13
3.6. Matlab	13
3.7. Stereometrie	14
3.8. Shrnutí	15
4. Funkcionalita programu Geomi	16
4.1. Teoretická část	16
4.2. Procvičovací část	16
4.3. Testovací část	16
4.4. Bezpečnost	16
5. Použité prostředky	17
5.1. Microsoft Windows XP	17
5.2. Microsoft Visual Studio 2008	17
5.3. Tao Framework	17
6. Uživatelská příručka	18
6.1. Instalace	18
6.2. Základní práce v programu	18
6.2.1. Základní prostředí	18
6.2.2. Okno s textem	19
6.2.3. Univerzální okno řezu	19
6.2.4. Hlavní menu	23
6.3. Rozšířená práce v programu	25
6.3.1. Přihlášení	25
6.3.2. Tvorba cvičení	25
6.3.3. Editace testů	26
6.3.4. Prohlížení výsledků testů	27
6.3.5. Tvorba textů	28

7. Programátorská dokumentace	29
7.1. Třídy	29
7.1.1. Třída About	29
7.1.2. Třída AdminLogin	29
7.1.3. Třída DeleteChapter	29
7.1.4. Třída ExcerciseFile	29
7.1.5. Třída ExcsCreator	32
7.1.6. Třída FindForm	33
7.1.7. Třída ImportDialog	33
7.1.8. Třída LabelDialog	33
7.1.9. Třída MainWindow	33
7.1.10. Třída MyView	36
7.1.11. Třída NewPointPosition	41
7.1.12. Třída PasswdChanger	41
7.1.13. Třída SolutionViewer	41
7.1.14. Třída TestCreator	41
7.1.15. Třída TextFile	43
7.1.16. Třída User	43
7.2. Moduly	43
7.2.1. Modul CubeModel	43
7.2.2. Modul Geometry	44
7.2.3. Modul PointDrawer	45
7.2.4. Modul PyramidModel	45
7.2.5. Modul Security	46
7.2.6. Modul TextDrawer	46
7.3. Používané soubory	46
7.3.1. Soubor Chapters.xml	46
7.3.2. Soubor Index.xml	47
7.3.3. Soubory typu excs	47
7.3.4. Soubory typu test	48
7.3.5. Soubory typu csum	48
7.4. Instalátor	48
Závěr	49
Conclusions	50
Reference	51
A. Obsah příloženého CD	53

Seznam obrázků

1.	Webová aplikace pro výuku stereometrie	10
2.	Výuka řezů těles Stereo	11
3.	Cabri 3D	12
4.	Dynamická geometrie v prostoru	12
5.	Říše čísel – Matematika 2	13
6.	Matlab	14
7.	Webová aplikace Stereometrie	14
8.	Instalace programu Geomi – výběr cílové složky	18
9.	Základní prostředí s otevřeným textovým oknem	20
10.	Jednotlivé nástroje	20
11.	Univerzální okno řezu v režimu procvičování	22
12.	Doplnění informací nutných k importu kapitoly	23
13.	Výběr kapitol k odstranění	24
14.	Dialogové okno určené ke hledání	24
15.	Přihlášení a změna hesla učitele	25
16.	Definice vlastností tvořeného příkladu	26
17.	Editor testů s otevřeným cvičením	27
18.	Prohlížení výsledků testu	28

1. Úvod

Hlavním cílem této práce je vytvoření podpůrného počítačového programu Geomi, který by žákům na gymnáziích usnadnil zvládnutí problematiky řezů krychle a jehlanu a který by zároveň učitelé matematiky dodal další možnosti procvičování a testování. Impulzem k tomu byla vlastní zkušenost z gymnázia, kde pro některé žáky bylo nesmírně obtížné pochopit tuto látku. Na program, který bude využit k tomuto účelu, jsou proto formulovány následující požadavky:

- Aplikace by měla obsahovat výukovou část, ve které by byla popsána teorie. Někdo by mohl namítnout, že tato část je zbytečná, protože přeci existují učebnice stereometrie. K tomu je však nutno podotknout, že využívání programu není vázáno využitím všech jeho částí, učitel tedy nemusí výukovou část používat. Dále lze říci, že pro učitele, kteří pro svou výuku současné učebnice nepoužívají, by neexistence této součásti mohla být velkou nepříjemností.
- Další část by měla tvořit část pro procvičování teorie. Tento požadavek tvoří jádro funkcionality programu, protože možnost neustálého procvičování je základem pro pochopení celé látky.
- Poslední část tvoří testovací část. Měla by existovat možnost testy ukládat a hodnotit. Tato část rozšiřuje možnosti programu a nabízí nový způsob kontroly vědomostí žáků. Je opět možné, že nebude využita, ale zde znovu záleží na rozhodnutí učitele, a proto by zde tato část být měla.
- Aplikace by měla obsahovat část pro učitele, ve které by mohl editovat testy a prohlížet si výsledky testů. Je bezpodmínečně nutné, aby žák neměl přístup ke stejným funkcím programu jako učitel. A to nejen kvůli zabezpečení výsledků testů. Proto by v programu měla existovat sekce určená jen pro učitele.

Tyto požadavky jsou tedy základem pro vytvoření nového programu. Pro vlastní tvorbu však bylo také důležité nejen pečlivě prostudovat způsoby výuky této problematiky na gymnáziích, ale také možnosti využití programů a projektů, které jsou v tuto chvíli dostupné.

2. Problematika výuky řezů na gymnáziích

Výuku řezů na gymnáziích lze považovat za důležitou pro její funkci při rozvoji prostorové představivosti žáků. Tato schopnost hraje pak významnou roli v jejich dalším životě.

2.1. Výuka

Stereometrie, která obsahuje i látku řezů těles, je povinnou složkou výuky matematiky na gymnáziích (viz [12]). Obvykle je tato kapitola zařazena ve druhém nebo třetím ročníku na čtyřletém gymnáziu a v odpovídajících ročnících osmiletých studijních programů. V těchto ročnících bývá týdenní hodinová 3 až 4 hodiny, škola však může v rámci svého ŠVP s touto hodnotou manipulovat. Dále lze studium řezů těles nalézt v předmětu Deskriptivní geometrie a v nepovinných seminářích, pokud je škola nabízí.

Oblíbenost této problematiky u učitelů ani u žáků nelze jednoznačně určit. U učitelů má na tento fakt vliv mnoho činitelů, od profesní specializace až ke klimatu třídy. U žáků oblíbenost ovlivňuje úroveň jejich schopností, pochopení problematiky, osoba učitele a jeho podání látky, ale také mnoho jiných faktorů.

2.2. Pomůcky

Při výuce řezů se nejčastěji používá drátěný model krychle a jehlanu. Jsou to pomůcky názorné, ale lze na nich řez pouze naznačit, ne prakticky provést. Další nevýhodou je fakt, že obvykle tento model má k dispozici pouze učitel. Řešením by bylo mít model pro každého žáka, ale to by bylo pravděpodobně finančně náročné a dosažený efekt by nebyl patrně dostatečný. Další možností by byla tvorba papírových modelů samotnými žáky, což by splňovalo jak finanční nenáročnost, tak dostupnost pro všechny žáky. Nevýhodou tohoto řešení je jednoúlohovost každého modelu, tedy možnost použití jednoho modelu pro jedinou situaci, což by v praxi znamenalo nutnost vytvořit jich velké množství.

Klasickým prostředkem výuky je tabule v učebně a žákův sešit. Tyto pomůcky jsou dostupné a běžně používané, ale jejich nevýhodou je možnost pouze 2D zobrazení. Práce s těmito prostředky pak vyžaduje vyšší úroveň představivosti právě kvůli nutnosti zpětného myšlenkového převodu 2D nákresu do 3D situace. Pokud se ji však žáci naučí správně používat, pak rýsování do sešitu a na tabuli je jednou z nejefektivnějších metod výuky.

Pomůckou kombinující výhody obou předchozích je interaktivní počítačový program. Ten umožňuje vizualizaci modelu, ale také rýsování, což do jisté míry tvoří ekvivalent žákova sešitu. Program sice nikdy plně nenahradí fyzický model tělesa, ale dovoluje simulovat více situací s možností rýsování přímo do nich. Tento způsob práce pak lze ještě rozšířit použitím interaktivní tabule v učebně. Toto pak nahrazuje rýsování na klasické tabuli a přidává výhody modelu tělesa.

2.3. Pochopení učiva

Na pochopení tohoto učiva žáky má vliv mnoho faktorů. K nejdůležitějším patří schopnost prostorové představivosti žáků, která bývá na různých úrovních a také různě rozvinuta. Tato schopnost je tedy vrozenou vlastností a nelze ji měnit, ale rozvíjet její potenciál ano. Z původní domněnky a její konzultace s učiteli s dlouholetou praxí (jmenovitě s Mgr. Miroslavem Hnilicou z Klvaňova gymnázia v Kyjově a Mgr. Davidem Martinákem z Maticního gymnázia v Ostravě) lze usoudit, že přibližně $\frac{1}{3}$ žáků je schopna tuto oblast učiva bez problémů pojmout, $\frac{1}{3}$ s mírnými problémy a poslední část má s tímto tématem velké potíže.

Při klasickém rýsování se také nabízí otázka vhodného způsobu promítání. V zásadě máme dvě možnosti – rovnoběžné promítání a perspektivní promítání. První z nich se využívá ve formě volného rovnoběžného promítání, což je snazší k provedení, ale následný zpětný myšlenkový převod je složitější. Díky snadnosti narýsování se tento způsob používá právě pro rysy na tabuli a v sešitech. Perspektivní promítání je narozdíl od rovnoběžného složitější k narýsování, ale poskytuje reálnější pohled na danou situaci. Právě proto se hojně využívá v počítačové grafice, kde se o vlastní provedení stará algoritmus vykreslování.

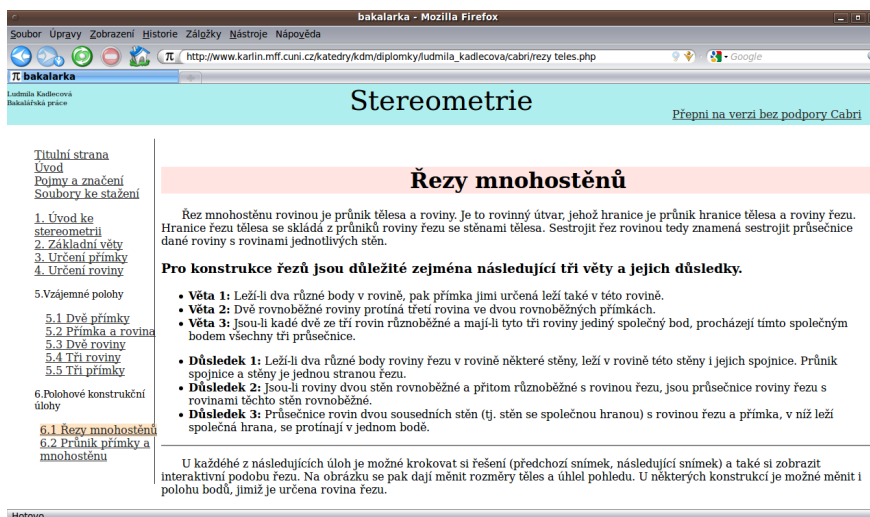
V neposlední řadě je třeba zmínit vliv prostředí. Protože se jedná o látku náročnou, je pro některé žáky obtížné si situaci ve vyučovací hodině správně představit z důvodu časového omezení. Pro tyto žáky je pak důležité si látku znovu projít a nyní již v klidu pochopit. S touto situací může pomoci vhodný počítačový program, který si žák nainstaluje a který mu pomůže celou problematiku pochopit i bez drátěného modelu.

3. Existující řešení

Pokud se podíváme na dostupné programy, které by bylo možno použít jako doplněk výuky řezů krychle a jehlanů, zjistíme, že se jich v současné době nevyskytuje mnoho a ne všechny lze získat zdarma. Mezi následujícími programy lze najít jak moderní software, tak aplikace, které dnes již současnému uživateli nevyhovují.

3.1. Webová aplikace pro výuku stereometrie

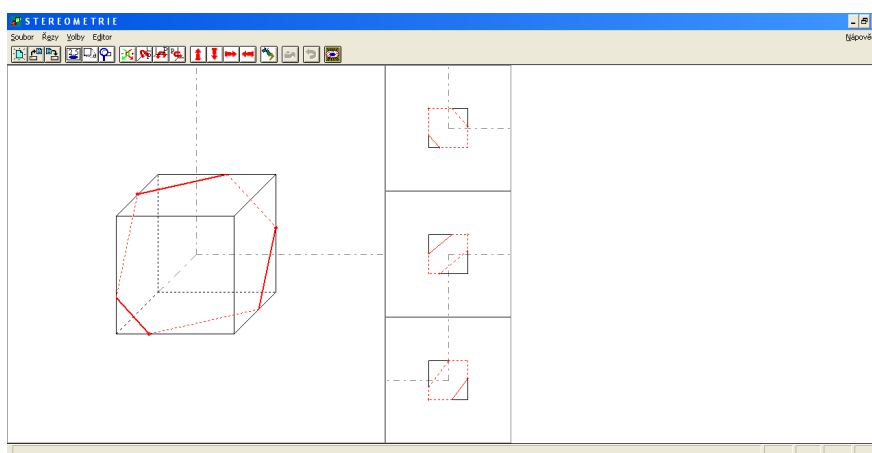
Tato bakalářská práce Ludmily Kadlecové z MFF UK v Praze [5] je zpracována jako interaktivní internetová stránka obsahující výukové texty, ukázková řešení i cvičení. Teoretická část i některá cvičení jsou ilustrovány obrázky nebo interaktivním modelem, který však potřebuje k práci přítomnost doplňku **Cabri 3D plugin**. Tento doplněk ale není povinný, aplikace nabízí možnost procházení i bez něj. Pokud má uživatel dotyčný modul instalován, pak se funkcionality aplikace rozrůstá také na možnost změny polohy daných útvarů i jednotlivých prvků zadání. Dále pak v případě řezů tělesa tento doplněk umožňuje krokování postupu nebo animaci namísto série obrázků, které se zobrazí, pokud není tento modul v počítači přítomen. Celá práce zabírá látku jak obecných poloh útvarů v prostoru, tak řezy mnohostěnů a průnik přímky s mnohostěnem. Pokud zaměříme pozornost na práci žáka v této aplikaci, pak nelze opominout, že přítomná cvičení nejsou interaktivní (je zde pouze možnost zobrazit řešení) a že práce neobsahuje testovací část.



Obrázek 1. Webová aplikace pro výuku stereometrie

3.2. Výuka řezů těles Stereo

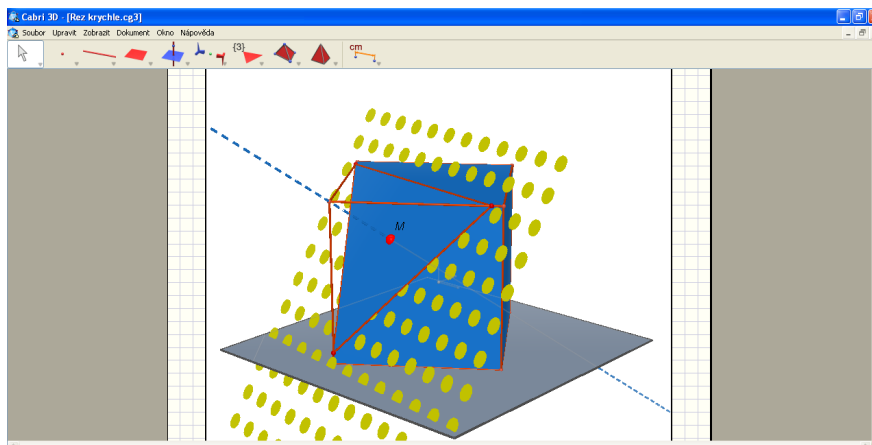
Program vytvořený Radovanem Janečkem [4] jako diplomová práce na PřF UP v Olomouci v roce 1996 bude současnému uživateli pravděpodobně připadat zastaralý, avšak některými svými vlastnostmi předčí i moderní software zde popsaný. Touto vlastností je možnost spojování různých těles do celků a jejich následné řezání. Uživatel může také měnit polohu jednotlivých těles a prvků řezu, což mu umožňuje sledovat změny řezu tělesa. U vlastního řezu zde není zobrazen postup, ale pouze výsledek. Dále pak je nutné uvést, že v programu není zpracována ani teoretická část, ani žákovy procvičování nebo testování, což dohromady snižuje didaktickou hodnotu tohoto softwaru.



Obrázek 2. Výuka řezů těles Stereo

3.3. Cabri 3D

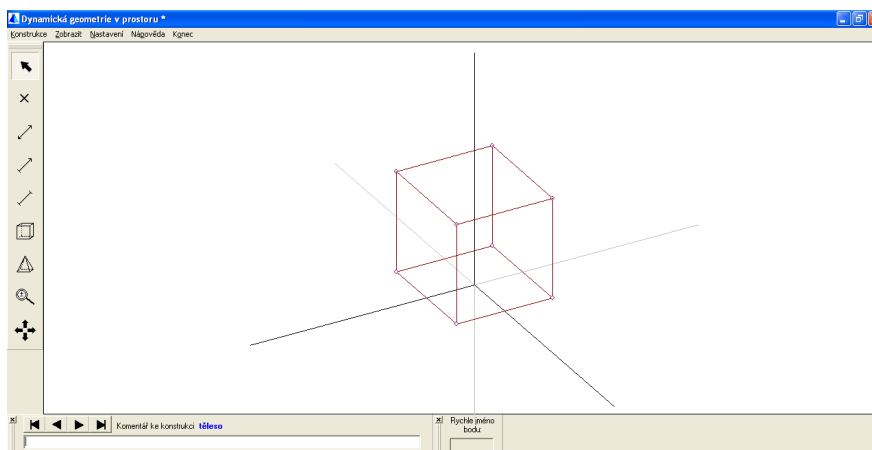
Tento komerční software [9] je vyvíjen firmou Cabrilog SAS stejně jako program **Cabri** pro planimetrické konstrukce, volně dostupná je pouze zkušební verze, která neumožňuje ukládání nebo tisk a jejíž funkčnost je omezena na 30 dní. Dlouhým vývojem tohoto softwaru bylo dosaženo velmi propracovaného systému práce s tělesy, stejně tak ovládání jednotlivých funkcí programu. Celý program není specializován na nějakou specifickou oblast geometrie, ale lze v něm provádět všechny stereometrické konstrukce. Není zde zabudován systém zobrazování teorie, což je ale u čistě konstrukčního programu pochopitelné. S tím souvisí také nepřítomnost testovací části. Možností, jak využít toto prostředí ve výuce, by byla sada demonstrativních příkladů, kterou bohužel výrobce nenabízí a učitel by si ji musel připravit sám bez existence šablony.



Obrázek 3. Cabri 3D

3.4. Dynamická geometrie v prostoru

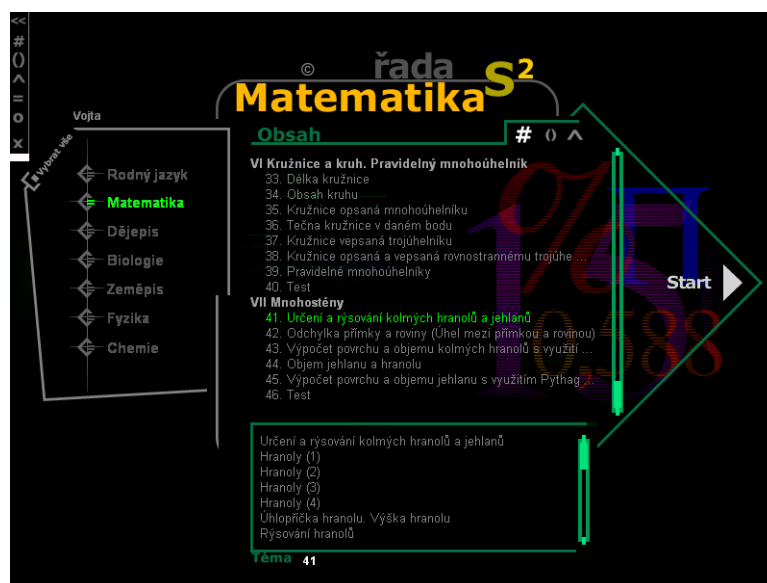
Výrobce tohoto programu je již zavedená firma Pachner, vzdělávací software, s. r. o., která jej distribuuje zdarma ve formě volně stažitelného ISO obrazu instalačního disku. Také v tomto případě se jedná o konstrukční systém, ve kterém není implementováno zobrazení teorie řezů ani kontrolované procvičování nebo testování. Program, obdobně jako **Cabri 3D**, umožňuje velmi vyspělou manipulaci s jednotlivými tělesy, narozdíl od něj však obsahuje zabudovanou funkci, která provede řez daného tělesa. Bohužel tato funkce opět zobrazí pouze výsledek bez možnosti náhledu postupu konstrukce řezu. Takto provedený řez se chová dynamicky, což v praxi znamená, že úpravou jednotlivých prvků zadání se mění aktuální podoba řezu.



Obrázek 4. Dynamická geometrie v prostoru

3.5. Říše čísel – Matematika 2

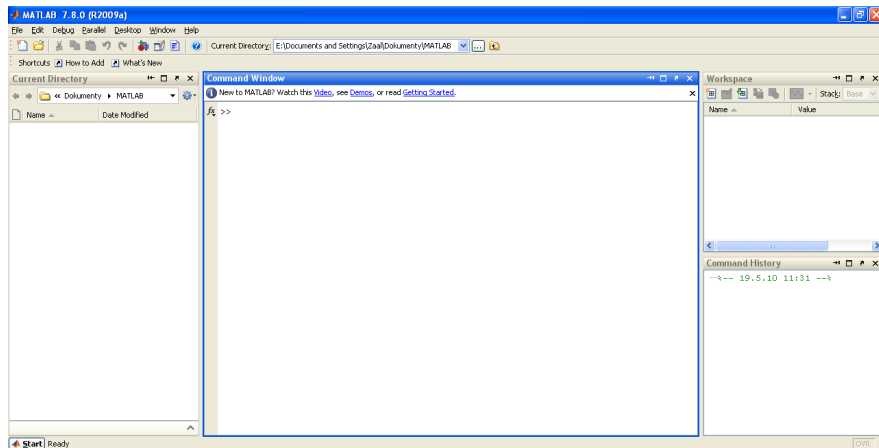
Firma LANGMaster International, s. r. o., je známým výrobcem mnoha výukových programů pro většinu středoškolských předmětů. Ve svém projektu **Říše čísel** [14] se však tělesy zabývá jen povrchně, ve výukové části jsou zde uvedeny pouze základní informace o mnohostěnech bez možnosti další úpravy učitelem. Naopak obecně celý systém zpracování látky je na velmi vysoké úrovni, zahrnuje moduly pro teorii, ukázky, procvičování a testování.



Obrázek 5. Říše čísel – Matematika 2

3.6. Matlab

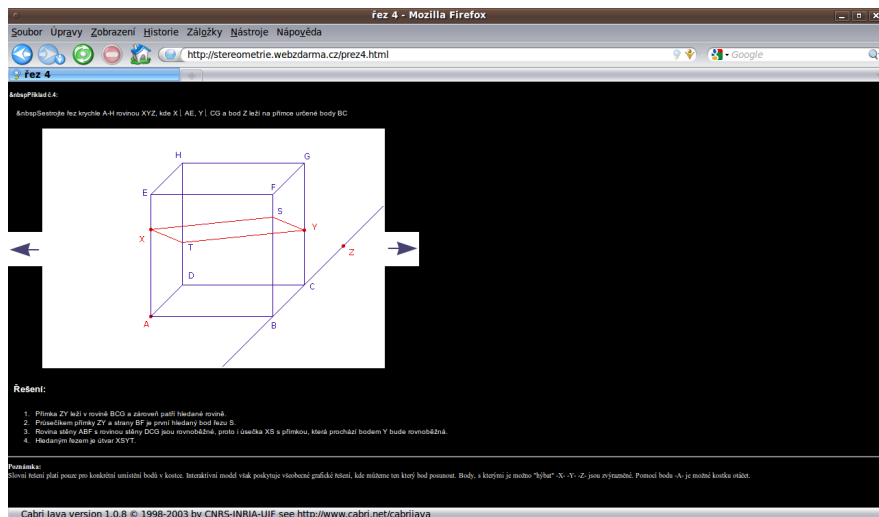
Velmi silným nástrojem na poli matematiky obecně je právě tento program společnosti Mathworks, který lze pomocí Virtual Reality Toolboxu rozšířit tak, aby v něm bylo možno pracovat s tělesy a jejich řezy. Bohužel jsem neměl možnost tento doplněk vyzkoušet, jelikož se jedná o placený software a na Katedře informatiky není zakoupen. Tato skutečnost také omezuje jeho případné využití na školách pro samostatnou přípravu žáků.



Obrázek 6. Matlab

3.7. Stereometrie

Posledním zaznamenaným projektem je webová aplikace [3] vytvořená Pavlem Holmanem, v současnosti však již není pravděpodobně udržovaná. Velkou výhodou je možno zde spatřit v jednoduchém způsobu práce s tělesy a dynamičnosti celého systému. Teoretická stránka problematiky je zde obsažena i s ilustracemi jednotlivých situací. Pro jednotlivé situace je nastíněn postup konstrukce, není však zobrazen ani jako animace, ani jako vizuální sled kroků. Dále pak tento projekt neobsahuje procvičovací nebo testovací část.



Obrázek 7. Webová aplikace Stereometrie

3.8. Shrnutí

Všechny uvedené projekty mají tedy své výhody a nevýhody, propracované části a nevhodná řešení. Za největší nedostatek je možno považovat absenci procvičování, tedy vlastní konstrukce řezu uživatelem, která se průběžně vyhodnocována a tím uživatele směřuje ke správnému řešení. Nejbližší k odstranění tohoto nedostatku je program Cabri 3D, případně program Dynamická geometrie v prostoru, do kterého by bylo možno vytvořit sadu příkladů zabývajících se řezy těles.

Dalším velkým nedostatkem je vynechání teoretické části se zobrazením postupu konstrukce řezu, což neumožňuje uživateli připomenutí již získaných znalostí. Tento problém řeší jediný projekt – Webová aplikace pro výuku stereometrie. Bohužel však neumožňuje uživateli tyto konstrukce vytvářet a uživatel je tedy odsunut do pozice pouhého pozorovatele.

V souhrnu lze říci, že se na současném trhu nevyskytuje program, který by plně vyhovoval jako vhodný program pro podporu gymnaziální geometrie se zaměřením na řezy těles.

4. Funkcionalita programu Geomi

Funkcionalitu programu Geomi lze rozdělit do tří hlavních částí – teoretickou, procvičovací a testovací. Každá z nich je specifická nejen činností žáka, ale i učitele.

4.1. Teoretická část

Tato část umožňuje zobrazování výukových textů uchovávaných v souborech typu RTF, což dovoluje text formátovat a vkládat do něj obrázky. Výukové texty učitel připravuje v externím textovém editoru, na který je zvyklý. Důležitou funkcí, kterou lze označit za přechodovou mezi teoretickou částí a procvičovací částí, je prohlížení dané situace na situačním modelu a krokování postupu řešení, které učitel připravil.

4.2. Procvičovací část

V této části žák řeší pomocí vizuálních nástrojů úlohu vytvořenou učitelem. Během toho má stále k dispozici všechny výukové texty a ilustrční příklady jako nápovědu, pokud by si nebyl jistý. Nedílnou součástí je možnost zkontrolovat si správnost provedení úkolu podle vzorového řešení vytvořeného učitelem v návrhovém prostředí.

4.3. Testovací část

Testováním ověřuje učitel žákovy dovednosti a vědomosti, které získal při studiu. Program Geomi umožňuje testování ve formě sady příkladů, které žák postupně řeší v určitém časovém limitu. Žákovo řešení je následně uloženo do předem určené složky tak, aby si jej mohl učitel následně prohlédnout. V této části nabízí program Geomi pro učitele editor testů a nástroj k hromadnému prohlížení řešení jednotlivých žáků.

4.4. Bezpečnost

Program Geomi poskytuje zabezpečení jednotlivých výukových textů, cvičení a testů pomocí kontrolního součtu umístěného ve zvláštním souboru přiloženém k původnímu souboru. Dále pak žákovi není umožněno využívat nástrojů určených pro učitele, protože k tomu je třeba se v programu přihlásit jako učitel, což je chráněno heslem.

5. Použité prostředky

Program Geomi byl vyvíjen za pomoci následujících prostředků.

5.1. Microsoft Windows XP

Tento operační systém byl zvolen proto, že je v současné době stále nejrozšířenějším operačním systémem na trhu a ve školách obzvláště. Není totiž možné očekávat, že jen kvůli tomuto programu by žáci spouštěli jiný operační systém než ten, ve kterém pracují většinu času.

5.2. Microsoft Visual Studio 2008

Vývojové prostředí MS Visual Studio nabízí kompletní soubor funkcí důležitých pro vývojáře aplikací, z nichž jmenujme například vizuální návrh jednotlivých formulářů, integrovanou nápovědu k danému programovacímu jazyku, automatické formátování zdrojového kódu z důvodu přehlednosti a v neposlední řadě také nástroje k ladění aplikace. Spolu s Microsoft .NET Frameworkem umožňují relativně snadný návrh programů pro operační systém Windows.

5.3. Tao Framework

Důležitou součástí programu Geomi je grafické zobrazení jednotlivých 3D modelů, k čemuž lze v zásadě použít dva systémy knihoven - OpenGL a DirectX. Druhá jmenovaná je vyvíjena společností Microsoft a tedy by měla lépe spolupracovat s operačním systémem Windows. Jako výhodu lze označit fakt, že uživatelé ji mají již většinou nainstalovánu, protože ji využívá velké množství aplikací. Její velkou nevýhodou je však její složitost a velikost příslušného SDK (Software Development Kit – balík nástrojů určených pro vývoj software).

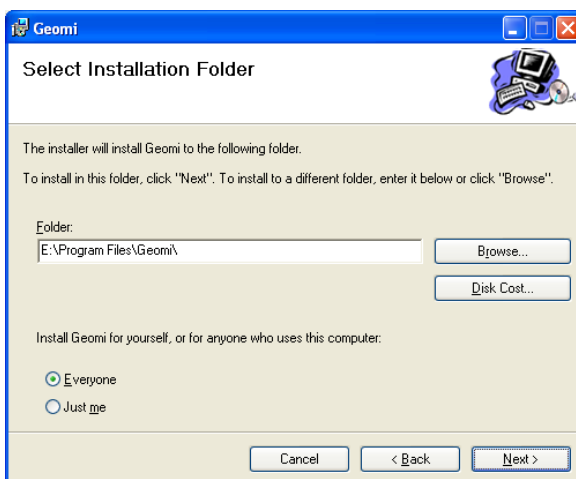
Naopak knihovny OpenGL poskytují jednoduché programátorské rozhraní, což spolu s faktem, že grafické rozhraní použité v programu Geomi pro zobrazování modelů nevyžaduje složité operace, je možno označit za rozhodující vlastnost. Protože však .NET Framework nepodporuje přímo spolupráci s OpenGL, bylo nutno použít systém Tao Framework, který zavádí možnost práce s OpenGL přímo do Visual Studia. Tento systém je volně dostupný ve formě instalačního balíku, ale ke spuštění výsledné aplikace jej celý instalovat nemusíme, programu stačí jen několik málo nevelkých knihoven, které si obvykle ve svém instalačním balíku sám dodá.

6. Uživatelská příručka

6.1. Instalace

Program Geomi je určen pro operační systém Microsoft Windows XP SP2 a ke svému běhu vyžaduje knihovny Microsoft .NET Framework minimální verze 3.5 a knihovny Tao Framework minimální verze 2.1.0. Kvůli tomu byl vytvořen instalátor, který tyto závislosti řeší. Nedoporučuje se spouštět program Geomi bez instalace, protože pak není možné zaručit správné spuštění programu.

Vlastní instalace se zahajuje spuštěním instalátoru `Geomi-Setup.msi`. Během instalace je uživatel požádán o zadání složky, kam se má program nainstalovat. Pokud v počítači nejsou přítomny knihovny .NET Framework, uživatel bude mít možnost je doinstalovat. Instalátor také vytvoří zástupce v programové nabídce *Start* v nové složce *Geomi*. Tím je celá instalace dokončena. Program nenabízí samostatný odkaz k odinstalaci, ta se provádí standardními nástroji operačního systému.



Obrázek 8. Instalace programu Geomi – výběr cílové složky

6.2. Základní práce v programu

Spuštěním programu se do tohoto módu přechází automaticky, není tedy třeba se přihlašovat.

6.2.1. Základní prostředí

Celá aplikace je tvořena hlavním oknem, které je rozděleno na dvě části – průzkumník kapitol vlevo a plochu pro jejich zobrazení vpravo. Uspořádání je

částečně variabilní v tom smyslu, že lze měnit plochu zabranou průzkumníkem na úkor zobrazovací plochy, případně lze průzkumník nezobrazit (viz nabídka [Zobrazení](#)).

Průzkumník kapitol Místem k zobrazení jednotlivých odkazů na kapitoly je průzkumník. Zobrazuje je v hierarchickém uspořádání, které může být buď definované uživatelem pomocí čísel kapitol nebo rejstříkové, tedy takové uspořádání, že k důležitým pojmům jsou přiřazeny jednotlivé kapitoly. Poklepáním se vybraná kapitola otevře na zobrazovací ploše. Existují čtyři druhy zobrazovaných kapitol:

- **Text** (zobrazen černým písmem) – zobrazuje text.
- **Ukázka** (zobrazeny zeleným písmem) – zobrazuje vzorové řešení dané úlohy. Pokud se jedná o zadaný řez třemi body, umožňuje také volbu Náhled situace, která zjednodušeně zobrazuje danou situaci pro snadnější pochopení.
- **Procvičování** (zobrazeno oranžovým písmem) – nechává žáka sestrojít řez s možnostmi kontroly správnosti a s možnostmi nahlédnout do textů a ukázek.
- **Test** (zobrazen červeným písmem) – žák zde řeší zadané úlohy v daném časovém limitu bez možnosti nahlédnout do textů nebo ukázek. Řešení těchto úloh se ukládá na pozici zvolenou učitelem. Po spuštění testu je žák požádán, aby vložil své jméno, což je určeno k identifikaci jeho řešení. Čas na vyřešení úloh je omezen a je zobrazován v titulku okna testu.

Zobrazovací plocha Tato část hlavního okna slouží k zobrazení jednotlivých textů, ukázek, cvičení a testů. Tato se zobrazují jako samostatná okna vázaná na zobrazovací plochu, lze s nimi tedy standardně manipulovat, ale pouze uvnitř této plochy.

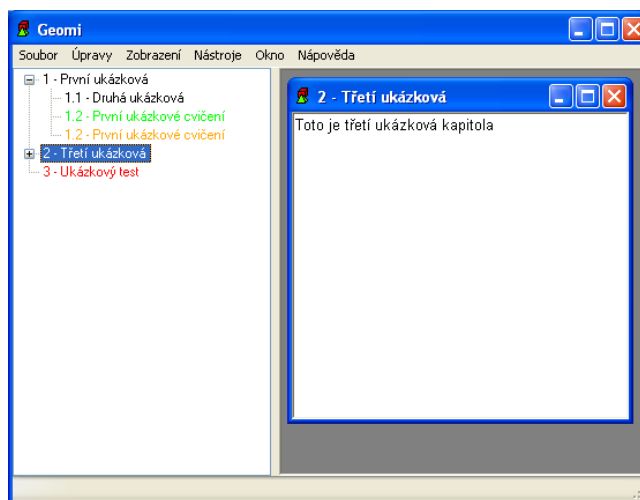
6.2.2. Okno s textem

Toto velmi jednoduché okno pouze zobrazuje text, který v něm však nelze měnit.

6.2.3. Univerzální okno řezu

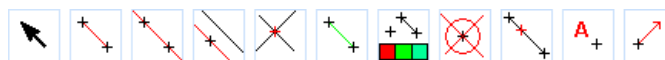
Pro ukázky, procvičování i testy je toto okno s mírnými odchylkami společné. Obsahuje plochu pro vlastní práci s řezem, nástroje a také tlačítka funkcí určených pro danou situaci.

Zadání V horní části okna je prostor pro zadání zobrazované úlohy. Tento prostor je editovatelný pouze při návrhu cvičení (viz [Tvorba cvičení](#)).



Obrázek 9. Základní prostředí s otevřeným textovým oknem

Panel nástrojů Pro vlastní práci s řezem se využívají nástroje seskupené do panelu nástrojů. Při zobrazení ukázky však není dostupný, protože zde není potřeba.



Obrázek 10. Jednotlivé nástroje. Zleva Žádná akce, Úsečka, Přímka, Rovnoběžka, Průsečík, Obarvená úsečka, Obarvit objekt, Smazat objekt, Umístit bod, Přidat popisek, Přidat vektor

- **Žádná akce** – pomocí tohoto nástroje se s vlastním řezem neprovádí žádná operace, slouží pouze pro otáčení modelu.
- **Úsečka** – tímto nástrojem se vytváří postupným označením dvou bodů úsečka.
- **Přímka** – jedná se o nástroj podobný předchozímu s tím rozdílem, že místo úsečky se vytvoří přímka procházející určenými dvěma body.
- **Rovnoběžka** – pro vytvoření přímky procházející daným bodem a rovnoběžné s určenou úsečkou či přímkou se použije tento nástroj. Jako první se vybere požadovaná přímka / úsečka a následně bod, kterým má nová přímka procházet.
- **Průsečík** – použitím tohoto nástroje a postupným označením určených přímk / úseček dojde k vytvoření nového bodu právě v místě jejich průsečíku.

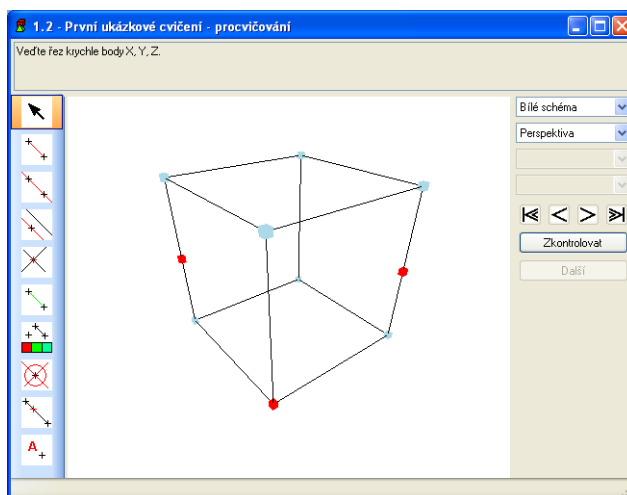
- **Obarvená úsečka** – tento nástroj slouží k vytvoření úsečky stejně jako nástroj Úsečka, ale v tomto případě je nová úsečka obarvena barvou příslušnou řezu a čára není čárkovaná. Obarvená úsečka slouží speciálně ke zvýraznění obrysu výsledného řezu.
- **Obarvit objekt** – nástroj Obarvit objekt slouží ke změně barvy jakéhokoli objektu řezu, což je určeno především pro lepší orientaci v řezu a také pro zvýraznění výsledného řezu.
- **Smazat objekt** – tímto nástrojem lze smazat jakýkoliv objekt řezu kromě prvků zadání.
- **Umístit bod** – k umístění bodu na úsečku tak, aby jeho vzdálenost od krajních bodů byla v určitém poměru, lze použít tento nástroj. Uživatel je vyzván, aby zadal požadovaný poměr a následně označil krajní body určené úsečky. Zadaná poměrná vzdálenost se vztahuje k prvnímu bodu výběru.
- **Přidat popisek** – tento nástroj slouží k označení jednotlivých bodů, přímkem nebo úsečkem. V označení lze použít pouze písmena anglické abecedy a číslice. Po výběru objektu určeného k označení je uživatel vyzván, aby zadal nové pojmenování v dialogovém okně. Toto okno se také automaticky zobrazuje po přidání libovolného bodu do řezu.
- **Přidat vektor** – viz [Rozšířená práce – Tvorba cvičení](#).

Plocha pro práci s řezem Toto prostředí slouží k vlastní práci s řezem, tedy především použitím nástrojů popsaných výše. Jsou však zde další možnosti, které nabízí. První z nich je rotace modelu za pomoci levého tlačítka myši (stisknutím a tažením), pokud je zvolena Žádná akce. Otáčení kolečkem myši nebo klávesy Ctrl a levého tlačítka myši (stisknutí a tažení) lze objekt přibližovat nebo oddalovat. Posun modelu je prováděn klávesami Home (doleva), End (doprava), PageUp (nahoru) a PageDown (dolů). Při zobrazení náhledu situace (viz [Průzkumník kapitol](#)) lze pravým tlačítkem myši přesouvat bod zadání řezu (stisknutím a tažením určeného bodu) ve směru určeném učitelem (případně směry), pokud je zadán. Takto lze bod přesouvat ve dvou směrech, pokud jsou však zadány tři, pak přesun ve třetím směru je umožněn pomocí kombinace klávesy Ctrl a pravého tlačítka myši stejným způsobem jako pro první dva směry.

Další funkce Na pravé straně okna jsou položky pro další funkce dostupné v dané situaci (vzorové řešení, procvičování, test).

- **Rozbalovací nabídka pro změnu promítání** – zde lze nastavit použité promítání (rovnoběžné, perspektivní).

- **Rozbalovací nabídka pro změnu barevného schématu** – touto nabídkou lze vybírat mezi třemi barevnými schématy – bílým, černým a vytvořeným učitelem.
- **Rozbalovací nabídka pro způsob zobrazení ukázky** – mezi zobrazením náhledu situace a zobrazením vzorového řešení je možno přepínat touto nabídkou, která je však dostupná pouze v zobrazení ukázky (viz [Průzkumník kapitola](#)) a pouze v případě, že se jedná o typ úlohy s předem zadanými třemi body řezu.
- **Tlačítka pohybu v konstrukci** – pro krokování postupu konstrukce lze využít tato čtyři tlačítka – Na začátek konstrukce, Krok zpět, Krok vpřed, Na konec konstrukce. Jsou dostupná ve všech uvedených situacích.
- **Tlačítko Zkontrolovat** – pouze pokud je spuštěno procvičování, lze použít toto tlačítko na kontrolu správnosti řešení. To je prováděno porovnáním zvýrazněného řezu vzorového řešení se zvýrazněnými objekty v aktuální konstrukci.
- **Tlačítko Další** – tímto tlačítkem přechází žák v režimu testu od jedné úlohy k další, nelze se však již vrátit zpět. V jiných režimech není toto tlačítko aktivní.
- **Tlačítko Uložit jako zadání, Uložit jako řešení** – viz [Rozšířená práce – Tvorba cvičení](#).



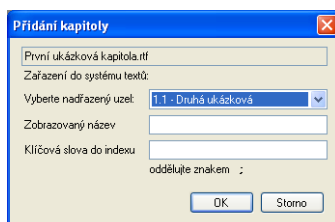
Obrázek 11. Univerzální okno řezu v režimu procvičování

6.2.4. Hlavní menu

Zde jsou uvedeny položky hlavního menu určené pro základní práci s programem.

Soubor Tato nabídka slouží k vlastní práci se soubory. Program využívá tři typy souborů odlišených různými příponami. Soubor s příponou **rtf** je souborem textu, přípona **excs** označuje cvičení a přípona **test** právě test. Je nutné zdůraznit, že aby program mohl daný soubor přečíst, musí být v daném umístění přítomen také soubor, jehož název je rozšířen o příponu **csum**, což je soubor s kontrolním součtem.

- **Otevřít samostatné cvičení** – otevře soubor cvičení v samostatných dvou oknech (Náhled situace / Vzorové řešení a Procvičování) mimo plochu v hlavním okně pro to určenou.
- **Otevřít samostatný test** – otevře soubor testu v samostatném okně obdobně jako předchozí položka.
- **Import textu, cvičení, testu** – umožňuje uživateli přidat kapitoly do průzkumníka. Po zvolení vlastních souborů určených k importu, je uživatel vyzván, aby doplnil informace o názvu, zařazení kapitoly a pojmech restříku, ke kterým se vztahuje.

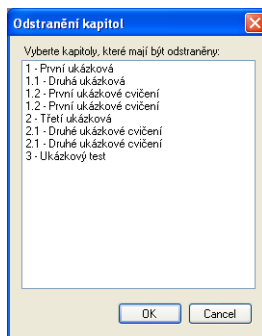


Obrázek 12. Doplnění informací nutných k importu kapitoly

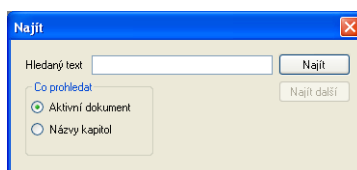
- **Odstranit kapitolu** – pomocí dialogového okna umožní uživateli odstranit kapitoly z průzkumníka. Odstraněním kapitoly dojde k automatickému odstranění všech kapitol, které jsou řazeny jako její podkapitoly.
- **Konec** – ukončí práci s programem.

Úpravy

- **Najít** – tato volba zobrazí dialogové okno vyhledávání, což umožňuje vyhledávat v názvech kapitol nebo v obsahu právě aktuálního textového okna.



Obrázek 13. Výběr kapitol k odstranění



Obrázek 14. Dialogové okno určené ke hledání

Zobrazení Touto nabídkou lze ovládat zobrazení Průzkumníka kapitol a jeho uspořádání.

- **Průzkumník kapitol** – odznačením této položky dojde k nezobrazení průzkumníka kapitol a k zneprůstupnění následujících dvou voleb.
- **Rejstřík** – výběrem této položky dojde k přeuspořádání průzkumníka kapitol tak, že kapitoly budou řazeny pod jednotlivé pojmy v restříku.
- **Strom kapitol** – výběrem této položky dojde k přeuspořádání průzkumníka kapitol tak, že kapitoly budou řazeny tak, jak bylo zvoleno při importu.

Nástroje Viz [Rozšířená práce](#).

Okno Tato nabídka obsahuje nástroje pro základní práci se zobrazovanými okny na ploše k tomu určené, kromě toho jsou zde také odkazy na právě zobrazená okna.

- **Kaskáda** – rozmístí otevřená okna kaskádovitě
- **Uspořádat horizontálně** – rozmístí okna tak, že všechna budou vyplňovat plnou šířku plochy a budou naskládána pod sebou.
- **Uspořádat vertikálně** – rozmístí okna tak, že všechna budou vyplňovat plnou výšku plochy a budou naskládána vedle sebe.

Nápověda

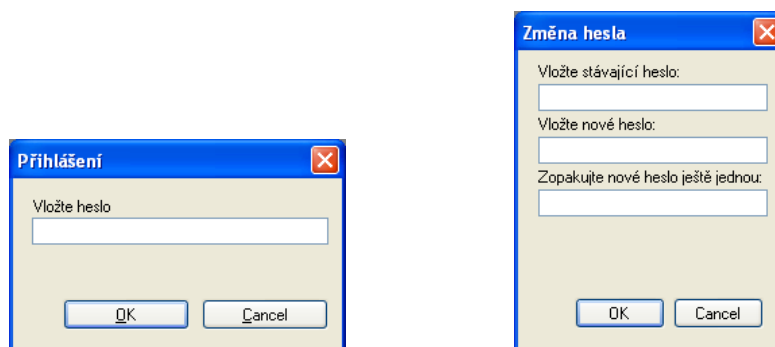
- **Dokumentace** – zobrazí externí soubor s popisem použití.
- **O programu** – zobrazí informace o programu.

6.3. Rozšířená práce v programu

Tento režim je určen především pro učitele, nijak však neomezuje základní práci v programu. Protože jsou zde nástroje, které nejsou určeny pro žáka, je vstup do tohoto režimu podmíněn přihlášením. Učíteli se následně zobrazí nové možnosti v nabídce Nástroje, kde se v základním režimu nachází pouze nabídka Přihlásit se jako učitel.

6.3.1. Přihlášení

Vlastní přihlášení probíhá po vybrání nabídky Přihlásit se jako učitel přes dialogové okno, kde je učitel vyzván k vložení hesla. Toto heslo lze v programu změnit přes nabídku Změnit heslo.



Obrázek 15. Přihlášení a změna hesla učitele

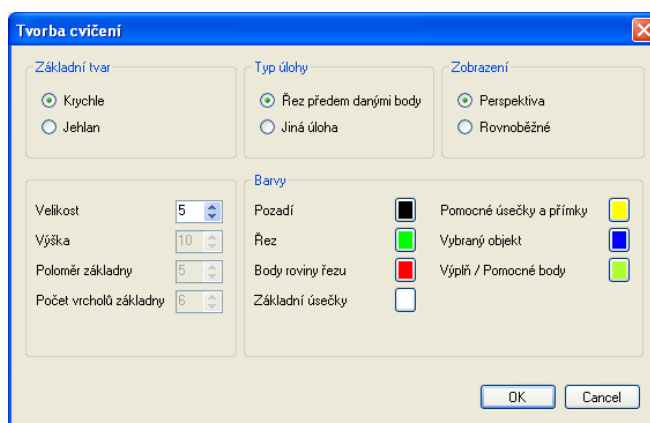
6.3.2. Tvorba cvičení

K vytvoření nového cvičení je třeba, aby učitel definoval, bude-li základním tělesem krychle nebo jehlan, vlastnosti tohoto těles, typ úlohy a preferovaný způsob promítání. Dále může učitel změnit barevné schéma použité při zobrazování řezu. Typem úlohy se myslí případ, kdy bude úloha řezu dána předem známými třemi body, nebo případ, kdy bude úloha zadána jinak. V prvním případě pak v návrhu cvičení musí učitel vyznačit příslušnou barvou (barva bodů roviny řezu) tři body zadání řezu.

Po potvrzení daných voleb se zobrazí univerzální okno řezu se zvoleným tělesem. Učitel v něm do textového pole vyplní zadání úlohy, na tělese vyznačí prvky zadání a potvrdí to tlačítkem Uložit jako zadání. Následně učitel na tomto

tělese vytvoří vzorové řešení pro žáky, kde bude obrys řezu zvýrazněn určenou barvou (barva řezu) pomocí obarvených úseček a bodů. Cvičení pak učitel uloží tlačítkem Uložit jako řešení. Tím je cvičení hotovo.

V situaci návrhu cvičení je dostupná nabídka panelu nástrojů Přidat vektor. Přidáním vektoru je žákovi umožněno, aby v zobrazení náhledu situace (viz [Průzkumník kapitol](#)) mohl daným bodem pohybovat a tím se podívat, jak se daná situace může měnit. Vlastní přidání vektoru se provede tak, že se nejprve označí bod, kterému tato funkčnost bude přidána, a následně koncový bod tohoto pomyslného vektoru.



Obrázek 16. Definice vlastností tvořeného příkladu

6.3.3. Editace testů

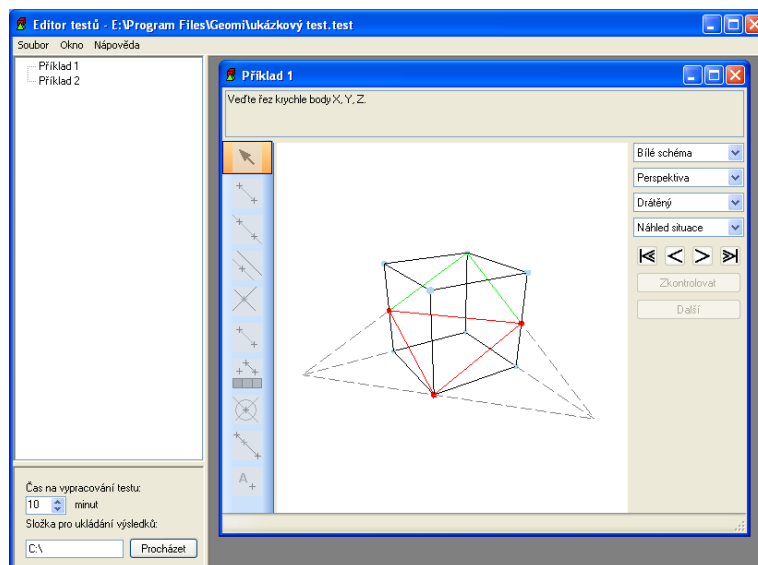
K tvorbě a editaci testů slouží nástroj Editor testů. Ten je otevřen v samostatném okně a nabízí vlastní menu. Nabídky Okno a Nápověda jsou obdobné jako hlavním okně (viz [Základní práce v programu](#)), nabídka Soubor je popsána níže.

Průzkumník testových úloh Tato oblast v levé části okna slouží ke zobrazení odkazů na jednotlivá cvičení testu, poklepáním na odkaz se cvičení zobrazí jako okno na ploše k tomu určené. Ve spodní části průzkumníka se nachází položky pro určení časového limitu testu a místa, kam se budou ukládat řešení žáků.

Plocha pro zobrazení úloh Tato plocha má obdobné určení jako plocha pro zobrazení hlavního okna programu Geomi (viz [Základí prostředí](#)), je tedy místem, kde se zobrazují jednotlivá cvičení.

Nabídka Soubor hlavního menu

- **Nový** – vytvoří prázdný test.
- **Otevřít** – otevře pro úpravy uložený test.
- **Uložit** – uloží rozpracovaný test.
- **Uložit jako** – uloží test do souboru včetně cvičení.
- **Přidat nové cvičení** – otevře nástroj ke tvorbě nových cvičení s tím, že nové cvičení je pak automaticky přidáno do testu.
- **Přidat existující cvičení** – přidá do testu cvičení ze souboru na disku.
- **Odebrat cvičení** – zobrazí dialogové okno, ve kterém může učitel odebrat cvičení z testu.
- **Konec** – ukončí práci s editorem testů.

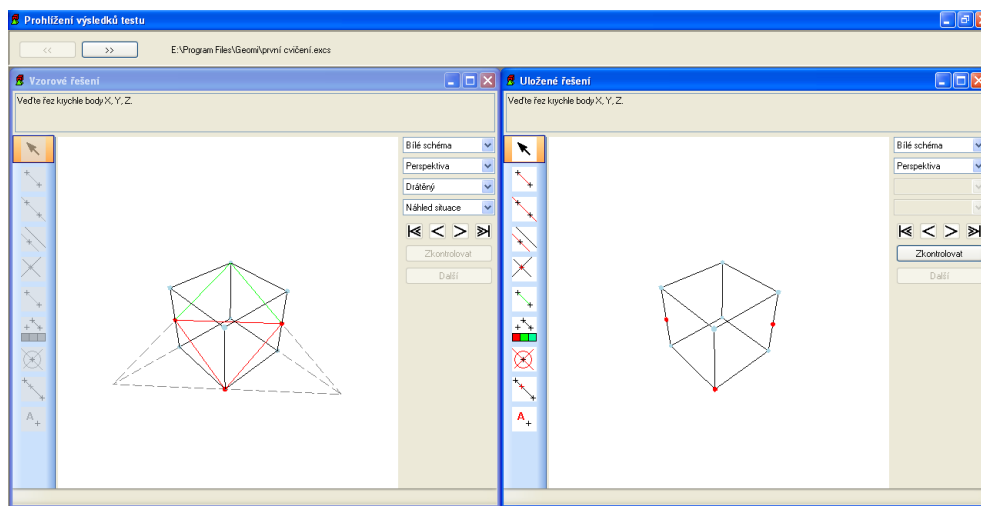


Obrázek 17. Editor testů s otevřeným cvičením

6.3.4. Prohlížení výsledků testů

Jednotlivá řešení testových úloh jsou ukládána jako cvičení, učitel je tedy může jednotlivě prohlížet nástrojem Otevřít samostatné cvičení. Pro tuto situaci je však vhodnější nástroj Prohlížení výsledků testů, který umožňuje otevřít celou sadu souborů a postupně je prohlížet pomocí k tomu určených tlačítek.

Prohlížení výsledků testu je otevřeno jako samostatné okno, ve kterém jsou vždy zobrazeny dvě podokna – vzorové řešení daného příkladu a řešení uskutečněné žákem. Tato podokna jsou opět univerzální okna řezu s přístupnými příslušnými funkcemi.



Obrázek 18. Prohlížení výsledků testu

6.3.5. Tvorba textů

V programu není obsažen speciální nástroj na tvorbu výukových textů, učitel si může zvolit libovolný textový editor, který podporuje ukládání ve formátu `rtf`. K takto vytvořenému textu je však nutné dopočítat kontrolní součet, aby bylo možné tento text zobrazit. K tomu slouží nabídka Přepočítat kontrolní součet, kde učitel vybere patřičný soubor a program k němu součet dopočítá.

7. Programátorská dokumentace

Tato část obsahuje programátorskou dokumentaci programu Geomi. Protože některé postupy jsou velmi specifické, bylo nutné je řešit za použití uvedených zdrojů.

7.1. Třídy

Zde je uveden popis jednotlivých tříd a jejich funkcionality.

7.1.1. Třída About

Jedná se o standardní formulář, který nabízí Visual Studio.

7.1.2. Třída AdminLogin

Tento formulář slouží k zadání hesla učitelem, žádnou funkčnost nenabízí.

7.1.3. Třída DeleteChapter

Formulář *DeleteChapter* obsahuje *ListBox* pro výběr kapitol, které mají být smazány.

- **Function GetResult() As String()** – tato funkce vrací pole řetězců – vybrané položky ke smazání.

7.1.4. Třída ExcerciseFile

Formulář *ExcerciseFile* je univerzálním oknem řezu použitým pro ukázkou, procvičování, návrh cvičení i test. Obsahuje komponenty *MyView* a *ToolStrip*, pomocí kterých je prováděna práce s tělesem.

- **Enum Mode** – tento výčtový typ obsahuje čtyři režimy funkcionality formuláře: Overview, Excercise, Creation, Test.
- **Event SuccessfullySaved(...)** – tato událost je použita k signalizaci úspěšného uložení při tvorbě nového cvičení do testu.
- **Sub New(...)** – vytvoření nové instance lze udělat pomocí čtyř procedur *New*: pro zobrazení cvičení (náhled, procvičování), test, návrh cvičení s tělesem krychle a návrh cvičení s tělesem jehlanu.
- **Sub ExcerciseFile_Load(...)** – tato základní procedura je volána při zobrazení formuláře a zajišťuje nastavení výchozích hodnot a zobrazení tělesa v patřičném režimu.

- **Sub LoadOverview()** – zajišťuje zobrazení náhledu situace.
- **Sub LoadExcercise()** – zajišťuje zobrazení procvičování.
- **Sub LoadCreator()** – zajistí správná nastavení při návrhu cvičení.
- **Sub LoadTest()** – nastaví okno pro zobrazení testu, zjistí identifikaci žáka dialogem *User* a nahraje první cvičení pomocí *LoadTestExcercise*.
- **Sub SetOutputFile(...)** – touto procedurou je z vnějšku nastavován výstupní soubor, do kterého má být nově tvořené cvičení uloženo.
- **Sub ComboBox1_SelectedIndexChanged(...)** – zde je prováděna změna režimu náhled situace na zobrazení vzorového řešení a zpět.
- **Sub ComboBox2_SelectedIndexChanged(...)** – zde je prováděna volba barevného schématu pomocí *DoChangeColors*.
- **Sub DoChangeColors()** – provede vlastní změnu barev.
- **Sub ComboBox3_SelectedIndexChanged(...)** – změna typu promítání je zajištěna zde.
- **Sub ComboBox4_SelectedIndexChanged(...)** – v režimu náhledu situace je zde zajištěna změna drátěného modelu na vyplněný a zpět.
- **Sub ComboBox4_VisibleChanged(...)** – navrácí výchozí nastavení při přechodu zpět do náhledu situace.
- **Sub MyView1_AskForNewLabel(...)** – tato procedura je navázána na událost *AskForNewLabel* komponenty *MyView* a zobrazí okno pro zadání nového popisku a tento popisek pak předá zpět.
- **Sub MyView1_AskIfDelete(...)** – navázána na událost *AskIfDelete* zobrazí dialog, ve kterém je uživatel dotázán, zda chce vybraný objekt smazat.
- **Sub MyView1_CheckSteps()** – tato procedura je navázána na událost změny pozice v postupu konstrukce, zavolá proceduru *DoCheckSteps*.
- **Sub DoCheckSteps(...)** – zde je provedena kontrola možnosti pohybu v postupu konstrukce a podle toho je umožněn přístup k příslušným tlačítkům.
- **Sub MyView1_IWantToTellYouSomething(...)** – pokud komponenta *MyView* potřebuje něco sdělit uživateli, provede to přes událost, na kterou ja navázána tato procedura. Ta zobrazí zaslou zprávu ve stavovém řádku.

- **Sub LoadFile(...)** – tato procedura slouží k nahrání cvičení ze souboru pomocí *DoLoadFile*.
- **Sub DoLoadFile(...)** – zde je provedeno vlastní načtení obsahu XML souboru do objektů *mainObj*, *colors*, *points*, *problempoints*, *lines*, *problemlines*.
- **Sub CheckButton_Click(...)** – obsluhou tlačítka pro kontrolu správnosti řešení je pověřena tato procedura, která využívá funkce *IsComplete* komponenty *MyView*.
- **Sub NextButton_Click(...)** – pro posun na další cvičení v testu je použita tato procedura. Zavoláním *SaveUserSolution* uloží žákovo řešení a pomocí *LoadTestExcercise* přejde na další cvičení testu.
- **Sub SaveUserSolution()** – tato procedura provede uložení žákova řešení.
- **Sub SetExtension(...)** – zde je nastavena správná přípona souborů cvičení pro ukládání návrhu.
- **Sub SaveButton_Click(...)** – tato procedura je volána při ukládání zadání v návrhu cvičení. Vyvolá dialog pro výběr cílového souboru.
- **Sub SaveButton2_Click(...)** – provede uložení vzorového řešení do souboru se zadáním cvičení.
- **Sub WriteBegining(...)** – tato procedura slouží k zapsání první části (obecných informací o těles a barev) do souboru cvičení.
- **Sub WriteDictionary(...)** – pomocí této procedury jsou přepsány vlastnosti uložené v kolekci do jednotlivých XML elementů a jejich atributů.
- **Sub LoadTestExcercise(...)** – zde je nahráváno testové cvičení pomocí *LoadExcercise* a provedeno počáteční nastavení testu.
- **Sub Timer1_Tick(...)** – test je omezen zadaným časem a tato procedura jej sleduje.
- **Sub ExcerciseFile_FormClosing(...)** – v případě návrhu cvičení, pokud uživatel zapomene uložit řešení a ukončuje práci, je dotázán, zda tak nechce učinit.
- **Sub Button1_Click(...)** a **Sub Button1_Click(...)** – provádí posun vpřed a vzad v postupu konstrukce.
- **Sub Button3_Click(...)** a **Sub Button4_Click(...)** – provádí posun na začátek a na konec konstrukce.

- **Sub UnCheckAll()** – tato procedura provede odznačení všech nástrojů v komponentě *ToolStrip*.
- **Sub Pointer()** – tato procedura provede signalizaci žádné akce komponentě *MyView*.
- **Sub ToolStripButton1_Click(...)** až **Sub ToolStripButton11_Click** – signalizují vybraný nástroj komponentě *MyView*.
- **Function IsEverythingOK() As Boolean** – slouží ke zjištění, zda během načítání nedošlo k nestandardní situaci.

7.1.5. Třída *ExcscCreator*

Formulář *ExcscCreator* slouží k zadání obecných informací o navrhovaném cvičení. Není zde volána žádná externí funkcionalita, vlastní tvorba cvičení je pak zajištěna na jiném místě.

- **Event ExcscCreatorIsClosing(...)** – tato událost slouží k signalizaci ukončení práce formulářem s tím, že jsou její pomocí předány výsledné hodnoty.
- **Sub ExcscCreator_Disposed(...)** – na ukončení práce s formulářem je navázána tato procedura, která pouze zajistí signalizaci události *ExcscCreatorIsClosing*.
- **Sub OK_Button_Click(...)** – jedná se o proceduru, která je navázána na stisk potvrzovacího tlačítka. Tato procedura pouze uloží zvolené hodnoty a uzavře okno formuláře.
- **Sub GetResult(...)** – tato procedura umožňuje získat výsledné hodnoty zvolené uživatelem, je však určena pro pevně kontrolované zobrazení tohoto formuláře mateřskou procedurou nebo funkcí.
- **Sub Cancel_Button_Click(...)** – standardní procedura pro ukončení práce bez potvrzení hodnot.
- **Sub RadioButton1_CheckedChanged(...)** – zvolením krychle jako základního tělesa dojde pomocí této procedury k aktivaci vstupního pole pro zadání velikosti hrany krychle.
- **Sub RadioButton2_CheckedChanged(...)** – zvolením jehlanu jako základního tělesa dojde pomocí této procedury k aktivaci příslušných vstupních polí pro zadání charakteristik tohoto tělesa.

- **Sub Button1_Click(...)** až **Sub Button6_Click(...)** – tyto procedury slouží k zobrazení dialogu pro výběr barvy.
- **Sub SetDefaults(...)** – pro nastavení základních hodnot mateřskou procedurou nebo funkcí slouží tato procedura.

7.1.6. Třída FindForm

Tento formulář slouží k vyhledávání.

- **Sub Button1_Click(...)** – pokud je zaškrtnuta možnost *Aktivní dokument*, projde tato procedura po stisknutí tlačítka *Najít* právě zobrazený dokument a vyznačí v něm hledaný výraz, pokud je však označena možnost *Názvy kapitol*, bude procedura hledat výraz v *Průzkumníku kapitol*.
- **Sub Button2_Click(...)** – tato procedura se pokusí po stisknutí tlačítka *Najít další* vyhledat další výskyt daného výrazu podle zadaných podmínek.
- **Function FindInNodes(...) As Integer** – vlastní prohledání stromu v *Průzkumníku kapitol* je realizováno právě touto funkcí, která vrací index daného uzlu.

7.1.7. Třída ImportDialog

Tento dialog slouží k zadání informací nutných ke správnému zařazení importovaného dokumentu do stromu kapitol a rejstříku, vlastní funkčnost nenabízí.

7.1.8. Třída LabelDialog

K zadání popisku pro vykreslovaný objekt slouží tento dialog, vlastní funkčnost však nenabízí.

7.1.9. Třída MainWindow

Tento formulář je hlavním oknem aplikace, proto tedy obsahuje mnoho funkcí, které slouží k celkové koordinaci.

- **Enum TreeStateValues** – tento výčet je použit pro určení režimu, který se vztahuje ke komponentě *TreeView*. Může jím být rejstřík nebo strom kapitol.
- **Event TreeChanged()** – tato událost slouží k vnitřní signalizaci změny stromu kapitol.
- **Sub MainWindow_FormClosing(...)** – pro kontrolu, zda uživatel při ukončování práce s programem nezapomněl na otevřený editor testů, slouží tato procedura.

- **Sub MainWindow_Load(...)** – základní procedura, která nastavuje defaultní hodnoty po spuštění programu.
- **Sub TextImport_Click(...)** – procedura vázaná na položku menu pro import textů. Výběr souborů je proveden standardním dialogem a vybrané soubory jsou pak zpracovány pomocí *DoImport*.
- **Sub ExcerImport_Click(...)** – obdoba *TextImportClick* pro import cvičení.
- **Sub TestImport_Click(...)** – obdoba *TextImportClick* pro import testů.
- **Sub ChangeTree()** – tato procedura provede načtení kapitol z pomocných struktur do *TreeView* v závislosti na aktuálním režimu.
- **Sub TreeViewLoad()** – do pomocných struktur v programu je zde načten obsah souborů *Chapters.xml* a *Index.xml*.
- **Function AddNode(...) As TreeNode()** – tato funkce akceptuje jako argument XML element, který převede na uzel zařaditelný do *TreeView*. Využívá se při načítání struktury kapitol.
- **Function ReadableName(...) As String** – z názvu kapitoly a z jejího čísla zařazení tak, jak je to v XML elementu souboru *Chapters.xml* a *Index.xml*, vytvoří zobrazovaný popis.
- **Sub DoImport(...)** – tato procedura provede vlastní import kapitol. Soubory zkopíruje do pracovní složky programu a pak je zařadí rejstříku a stromu kapitol pomocí *TreeImport* a *IndexImport*.
- **Sub TreeSerialize(...)** – všechny názvy uzlů v daném stromu seřadí do jednorozměrného pole.
- **Function GetNewChapterNumber(...) As String** – slouží k výpočtu čísla kapitoly pro uložení do souborů rejstříku a stromu kapitol.
- **Function TreeImport (...) As String** – na základě výběru kapitoly, pod kterou má být nová kapitola zařazena, ji vloží do souboru struktury. Návrátovou hodnotou je číslo zařazení v podobě řetězce.
- **Function XmlChildNodeLocate(...) As XmlNode** – slouží k nalezení XML elementu podle zobrazovaného popisu kapitoly.
- **Sub IndexImport(...)** – na základě uvedených slov, zařadí kapitolu do struktury rejstříku.
- **Sub WriteNewNode(...)** – tato procedura obstarává zápis XML elementu kapitoly do souboru.

- **Sub MainWindow_TreeChanged()** – na signalizaci události *TreeChanged* je navázána tato procedura. Vyvolá opětovné načtení obsahu komponenty *TreeView*.
- **Sub TreeView1_NodeMouseDoubleClick(...)** – na tuto proceduru je nasměrována obsluha poklepání na uzel *TreeView*. To vyvolá otevření příslušného souboru. K tomu slouží procedury *OpenTextFile*, *OpenExcerciseFile* a *OpenTestFile*.
- **Sub OpenTextFile(...)** – otevře *TextFile* jako MDI potomka a zobrazí v něm obsah textového souboru.
- **Sub OpenExcerciseFile(...)** – zobrazí zvolený soubor cvičení buď jako MDI potomka nebo jako samostatné okno. Obě varianty však použijí okno *ExcerciseFile*.
- **Sub OpenTestFile(...)** – pro zobrazení testu použije tato procedura také *ExcerciseFile*, ale v jiném režimu. Současně zakáže zobrazování jakýchkoliv jiných kapitol.
- **Sub MenuItem5_Click(...)** a **Sub MenuItem6_Click(...)** – jsou navázány na položky menu určené ke změně rozbrazení *TreeView* (rejstřík, strom kapitol).
- **Sub MenuItem7_Click(...)** – tato procedura je navázána na položku menu určenou pro tvorbu nového cvičení. Vyvolá dialog *ExcscCreator*.
- **Sub MenuItem8_Click(...)** – položka menu určená pro tvorbu a editaci testů je navázána právě na tuto proceduru, která vyvolá otevření okna *TestCreator*.
- **Sub ECreator_ExcscCreatorIsClosing(...)** – událost uzavření okna *ExcscCreator* je navázána na tuto proceduru, která zjistí nastavené hodnoty a s nimi volá otevření okna *ExcerciseFile* v režimu návrhu cvičení.
- **Sub MenuItem10_Click(...)** – ukončuje práci s programem.
- **Sub MenuItem4_Click(...)** – slouží k otevření samostatného cvičení do dvou oken.
- **Sub MenuItem25_Click(...)** – slouží k otevření samostatného testu.
- **Sub MenuItem14_Click(...)** – slouží k zobrazení okna *About*.
- **Sub MenuItem16_Click(...)** – slouží k otevření nástroje na prohlížení výsledků testu *SolutionViewer*

- **Sub MenuItem18_Click(...)** až **Sub MenuItem20_Click(...)** – tyto procedury slouží k uspořádání zobrazených oken – MDI potomků.
- **Sub MenuItem22_Click(...)** – slouží k zobrazení dialogu *FindForm* určeného pro hledání.
- **Sub MenuItem23_Click(...)** – tato procedura slouží k zobrazení nebo skrytí panelu s *TreeView*.
- **Sub tWindow_Disposed(...)** – ukončení práce s testem opět umožní otevírání kapitol.
- **Sub MenuItem26_Click(...)** – tato procedura je navázána na položku menu pro odstranění kapitol. Spustí se dialog *DeleteChapter* a vybrané kapitoly se odstraní ze souborů struktury. Následně je volána procedura *Reorganize*, která zajišťuje napravení integrity dat.
- **Sub Reorganize(...)** – provede reorganizaci v souborech struktury tak, aby si záznamy o kapitolách odpovídaly.
- **Sub ReNewChapterNumbers(...)** – provede přepočtení čísel zařazení jednotlivých kapitol.
- **Sub MenuItem27_Click(...)** – vyvolá dialog pro přihlášení učitele *AdminLogin*, případně provede odhlášení.
- **Sub MenuItem29_Click(...)** – obsluhuje nabídku menu pro přepočtení kontrolního součtu. Ten je proveden funkcí *CreateNewHashFile* modulu *Security*.
- **Sub MenuItem30_Click(...)** – tato procedura slouží ke změně hesla zavoláním *ChangePassword* modulu *Security*.
- **Sub TCreator_FormClosed(...)** – po uzavření editoru testů odstraní odkaz na jeho instanci.
- **Sub MenuItem13_Click(...)** – zobrazuje uživatelskou příručku.

7.1.10. Třída MyView

Tato třída tvoří vykreslovací plochu pro kreslení pomocí OpenGL a také práci se zobrazenou grafikou. Koncepce zavedení zobrazovací plochy jako samostatné třídy je převzata ze stránek Tao Framework [15] a způsob výběru objektů na této ploše je proveden podle OpenGL Programming Guide [11].

Samotná třída je odvozena od třídy *SimpleOpenGLControl* z balíku *TaoFramework*. Postupy vykreslování jsou odvozeny od postupů uvedených v sérii článků publikovaných na serveru Root.cz [13].

- **Enum Mode** – je použit pro určení režimu zobrazení, obsahuje dvě hodnoty: *presentation* a *drawing*.
- **Enum Figure** – tento výčet je určen pro typ zobrazovaného tělesa. Může nabýt hodnot *cube* nebo *pyramid*.
- **Enum ObjToDraw** – tímto výčtem je obsažena množina všech akcí, které lze s modelem provést.
- **Structure Line** – struktura pro popis přímek a úseček.
- **Structure Point** – struktura pro popis bodů.
- **Sub New()** – vytvoření nové instance.
- **Sub SetPropriets(...)** – tyto dvě procedury umožňují nastavit obecné vlastnosti tělesa.
- **Sub GenerateCubeOnly(...)** – voláním této procedury je zobrazen základní model krychle.
- **Sub GeneratePyramidOnly(...)** – voláním této procedury je zobrazen základní model jehlanu.
- **Sub SetPoints(...)** – touto procedurou se nastavují body tělesa a zadání pomocí funkce *LoadPoints*
- **Function LoadPoints(...) As Point()** – zde jsou převedeny informace o bodech z kolekce do struktur typu *Point*.
- **Sub SetLines(...)** – v tomto případě jsou zde nastaveny zobrazované přímky a také přímky obsažené v zadání.
- **Function LoadLines(...) As Line()** – zde jsou převedeny informace o přímkách z kolekce do struktur typu *Line*.
- **Sub SetSolution(...)** – tato procedura je užita pro nastavení bodů a přímek řešení.
- **Sub SetSolid(...)** – pomocí této procedury je nastaven paramater *_isSolid*, což je využito v módu *presentation*.
- **Sub SetColor(...)** – zde probíhá nastavení barvného schématu.

- **Sub ChangePointsColor(...)** a **Sub ChangeLinesColor(...)** – při změně barevného schématu je nutno projít všechny body a přímký a jejich barvu změnit na novou.
- **Sub SetMode(...)** – slouží k nastavení režimu práce.
- **Sub SetObjectToDraw(...)** – pomocí této funkce nastavuje nástroj v objektu *ToolStrip* v okně *ExcerciseFile* prováděnou akci.
- **Sub SetProjection(...)** – nastavení typu promítání.
- **Sub SetStartPosition(...)** – nastavení pozice modelu.
- **Sub SetNewPointPosition(...)** – nastavení pozice nově vytvářeného bodu na úsečce.
- **Function GetMainObject() As Dictionary(Of String, String)** – tato funkce slouží k vytvoření kolekce obsahující informace o tělese.
- **Function GetColors() As Dictionary(Of String, String)** – vrací kolekci barev použitých ve schématu.
- **Function GetPoints() As Dictionary(Of String, Object)** – pomocí funkce *SavePoints* vrací kolekci vykreslených bodů.
- **Function GetSortedPoints() As Dictionary(Of String, Object)** – vrací uspořádanou kolekci vykreslených bodů.
- **Function GetSolutionPoints() As Dictionary(Of String, Object)** – vrací kolekci bodů obsažených v *SolutionPoints*.
- **Function SavePoints(...)** **As Dictionary(Of String, Object)** – provede převod zvolených bodů do kolekce.
- **Function SortPoints(...)** **As Point()** – provede seřazení bodů tak, aby body zadání byly na začátku.
- **Function GetLines() As Dictionary(Of String, Object), Function GetSolutionLines() As Dictionary(Of String, Object), Function SaveLines(...)** **As Dictionary(Of String, Object)** – obdoba výše popsaných funkcí zde určených pro přímký.
- **Sub Init()** – inicializace kreslicího plátna.
- **Sub SetPerspectiveProjection()** – nastavení typu promítání pomocí *SetProjectionMatrix*.
- **Sub SetProjectionMatrix()** – nastavení vlastní matice promítání.

- **Sub OnDisplay()** – zde je prováděno vlastní vykreslování. Pokud je nastaven režim *Presentation* a je platný příznak nutnosti přepočtu modelu, je pomocí funkcí modulu *Geometry* proveden výpočet bodů řezu, jeho obrysu a případně krajních ploch výplně řezu, nejedná-li o zobrazení drátěného modelu. Tyto objekty jsou vykresleny samostatně od ostatních prvků, které jsou zobrazeny funkcí *DrawObjects*.
- **Sub DeleteObjectAnswer(...)** – tato funkce je volána po zodpovězení otázky ohledně smazání vybraného objektu. Objekt ve skutečnosti není smazán, je pouze odsunut v postupu konstrukce na místo, které již není zobrazováno. V příznaku je mu však nastaveno původní umístění pro případ obnovení.
- **Sub AddLabel(...)** – zde je provedeno vlastní pojmenování objektu po zadání popisku uživatelem.
- **Sub DrawObjects(...)** – zde je provedeno vlastní vykreslování objektů. Vstupní parametr udává, zda se jedná o standardní vykreslování nebo o vykreslování pro potřeby výběru objektu.
- **Sub OnResize(...)** – při změně rozměrů plátna upraví viditelnou oblast.
- **Sub MyView_Invalidated(...)** – po zneplatnění komponenty nastaví příznak nutnosti překreslení.
- **Sub MyView_KeyDown(...)** – obsluhuje tlačítka klávesnice.
- **Sub MyView_KeyUp(...)** – klávesa *Crtl* má to výsadní postavení, že je potřeba na její stisknutí reagovat i během jiné akce, takže po jejím uvolnění jsou zde přepnuty příznaky zpět na výchozí hodnoty.
- **Sub ProcessHits(...)** – tato důležitá procedura provádí zvolené akce s právě vybraným objektem. Číselné označení tohoto objektu je uloženo v zásobníku s tím, že pro úsečky je sudé a pro body liché.
- **Sub MyView_MouseDown(...)** – obsluhu výběru pomocí tlačítka myši provádí tato procedura. Pokud se neprovádí žádná akce, pak pracuje s modelem jako s celkem, jinak volá proceduru *Selection* určenou pro výběr objektu na plátně.
- **Sub Selection(...)** – tato procedura nechá vykreslit všechny objekty ve speciálním režimu určeném pro výběr a následně vybraný objekt předá proceduře *ProcessHits*.

- **Sub MyView_MouseMove(...)** – pokud se pracuje s tělesem jako s celkem, pak zde prováděna změna parametrů ovlivňujících zobrazení tělesa (rotace, posun, přiblížení). Pokud je však vybrán bod s možností posuvu, pak souřadnice bodu se mění právě zde pomocí procedury *PointMovement*.
- **Sub PointMovement(...)** – mění souřadnice bodu v závislosti na vektoru, který je mu přidělen.
- **Sub MyView_MouseUp(...)** – vrací hodnoty příznaků nastavených během manipulace na výchozí hodnoty.
- **Sub MyView_MouseWheel(...)** – zde je prováděna jiná možno přibližování a oddalování pomocí pohybu kolečka myši.
- **Sub MyView_Paint(...)** – při potřebě překreslení komponenty při nastaveném příznaku je zde volána procedura *OnDisplay*.
- **Sub SendMessage(...)** – tato procedura je použita pro signalizaci události, pokud je potřeba sdělit informaci uživateli.
- **Function IsComplete() As Boolean** – pro zjištění, zda provedený řez souhlasí s vzorovým řešením, je použita tato funkce. Porovnání je provedeno na základě shodných objektů zvýrazňujících řešení.
- **Function CountCutPoints() As Integer** – vrátí počet bodů, které jsou označeny jako body zadání řezu.
- **Sub NextStep()** – provede posun vpřed v postupu konstrukce pomocí nové hodnoty indexů pro pořadí vykreslovaných objektů.
- **Sub PreviousStep()** – analogicky provede posun vzad.
- **Sub RemoveFromLists()** – tato procedura odstraní objekty, které jsou mimo interval zobrazovaných indexů pořadí.
- **Function FindUpperOrderBound(...) As Integer** – nalezne nejvyšší hodnotu indexu vykreslovacího pořadí.
- **Function AllowPrevious() As Boolean** – slouží ke zjištění, zda je možné postupovat zpět v postupu konstrukce.
- **Function AllowNext() As Boolean** – obdobná funkce pro postup vpřed.
- **Sub ResetOrderCounter()** – nastaví hodnoty indexů pořadí všech objektů na 0.

7.1.11. Třída `NewPointPosition`

Tento dialog slouží k zadání pozice ve tvaru zlomku nově umisťovaného bodu.

- **Function `GetResult() As Single`** – pro zjištění výsledné hodnoty je určena tato funkce, ze tvaru zlomku je hodnota převedena na destinné číslo.

7.1.12. Třída `PasswdChanger`

Dialog `PasswdChanger` slouží ke změně hesla učitele pomocí funkce `SetNewPassword` modulu `Security`.

7.1.13. Třída `SolutionViewer`

Tento formulář je řešen jako MDI okno se dvěma tlačítky zajišťujícími posun v množině otevřených cvičení.

- **Sub `New(...)`** – vytvoření nové instance tohoto formuláře je možné pouze se zadáním množiny souborů určených k otevření.
- **Sub `SolutionViewer_Load(...)`** – tato procedura, která je volána při zobrazení formuláře, pomocí `LoadFile` otevře první ze souborů.
- **Sub `LoadFile(...)`** – zde je prováděno samotné otevření souboru cvičení. Nejprve procedura uzavře okna předchozího cvičení a poté otevře okna aktuálního cvičení. Tato okna pak uspořádá.
- **Sub `Button1_Click(...)`** – jedná se o obsluhu tlačítka pro zobrazení předchozího souboru, ta se porovede procedurou `LoadFile`.
- **Sub `Button2_Click(...)`** – jedná se o obsluhu tlačítka pro zobrazení následujícího souboru, ta se porovede procedurou `LoadFile`.

7.1.14. Třída `TestCreator`

Funkcionalitu vytváření a editace testů zajišťuje tento formulář. Je řešen jako MDI okno spolu s komponentou `TreeView` v levé části určenou k zobrazení struktury testu. Dále pak tento formulář obsahuje menu a vstupní pole pro zadání časového limitu a složky pro uložení řešení.

- **Sub `New(...)`** – vytvoření nové instance je možné pouze se zadáním přípon používaných souborů.
- **Sub `NovýToolStripMenuItem_Click(...)`** – tato procedura je určena pro vytvoření prázdného testu, je volána z menu.
- **Sub `Clear()`** – návrat stavu jednotlivých komponent do výchozího stavu zajišťuje právě tato procedura.

- **Sub OtevřítToolStripMenuItem_Click(...)** – pro načtení uloženého testu je volána tato procedura. Soubor testu je určen pomocí standardního dialogu.
- **Sub MenuItem18_Click(...)** až **Sub MenuItem20_Click(...)** – tyto procedury slouží k uspořádání zobrazených oken – MDI potomků.
- **Sub MenuItem7_Click(...)** – pro nahrání již existujícího souboru cvičení do testu slouží tato procedura. Výběr souboru je opět proveden standardně.
- **Sub TreeView1_NodeMouseDoubleClick(...)** – obsluhu dvojitého kliknutí na uzel v komponentě *TreeView* obsluhuje tato procedura zavoláním *LoadExcercise*.
- **Sub LoadExcercise(...)** – tato procedura provede vlastní nahrání souboru cvičení do nového okna (MDI potomka tohoto formuláře)
- **Sub MenuItem6_Click(...)** – stejně jako u hlavního okna probíhá odstraňování pomocí dialogu *DeleteChapter*, který je volán právě touto procedurou, která je aktivována z menu. Následně jsou pak zde vybraná cvičení smazána.
- **Sub MenuItem10_Click(...)** – ukončí editor, je volána z menu.
- **Sub Button1_Click(...)** – tato procedura zajišťuje doplnění složky pro uložení výsledků do příslušného vstupního pole pomocí dialogu pro výběr složek.
- **Sub MenuItem8_Click(...)** – vybráním příslušné položky menu je aktivována tato procedura, která slouží k vytvoření nového cvičení a jeho zařazení do testu. Pro tvorbu používá dialog *ExcrcCreator* a okno *ExcerciseFile*.
- **Sub exf_SuccessfullySaved(...)** – tato procedura je navázána na událost úspěšného uložení nově tvořeného cvičení do dočasného souboru. Jejím účelem je přidat tento soubor do seznamu příkladů aktuálního testu.
- **Sub MenuItem4_Click(...)** – položku menu *Uložit* obsluhuje tato procedura standardním způsobem.
- **Sub MenuItem3_Click(...)** – standardním způsobem je také obsloužena položka *Uložit jako* touto procedurou.
- **Sub DoSaveAs()** – tato procedura provádí výběr souboru pro uložení a následně volá *DoSave*.

- **Sub DoSave()** – zde je provedeno vlastní uložení do předem vybraného souboru a vytvoření souboru s kontrolním součtem.
- **Sub TestCreator_FormClosing (...)** – pro případ, že by uživatel zapomněl uložit rozpracovaný test a ukončoval by editor, tato procedura zajistí zobrazení kontrolní otázky a případně následné uložení pomocí *DoSave* nebo *DoSaveAs*
- **Sub MenuItem13_Click(...)** – zobrazuje uživatelskou příručku.

7.1.15. Třída `TextFile`

Tento formulář slouží pouze k zobrazení text v komponentě *RichTextBox*, vlastní funkčnost neobsahuje.

7.1.16. Třída `User`

Dialog *User* slouží k zadání jména a příjmení žáka při otevření testu.

- **Function GetUser() As String** – touto funkcí lze zjistit zde zadanou identifikaci studenta.

7.2. Moduly

Funkce a procedury byly do jednotlivých modulů odděleny kvůli přehlednosti, protože v těchto skupinách zajišťují specifickou funkčnost.

7.2.1. Modul `CubeModel`

Obsahem tohoto modulu jsou nástroje pro geometrickou práci s modelem krychle.

- **Function OnCube(...)** **As Boolean** – tato funkce slouží ke zjištění, zda bod leží na povrchu krychle.
- **Function OnSameSide(...)** **As Boolean** – ke zjištění, zda dva body leží na téže straně, slouží tato funkce. Je využita pro automatické tvorbu řezu.
- **Function CubeCut(...)** **As Single()()** – pomocí průsečíků zadané roviny s hranami krychle a jejich následným uspořádáním počítá tato funkce řez krychle. Návrátovou hodnotou je pole bodů řezu reprezentovaných jako pole souřadnic.
- **Function PolygonContains(...)** **As Boolean** – touto funkcí lze zjistit, zda pole již neobsahuje daný bod. Vlastní porovnání je provedeno funkcí *IsSamePoint* modulu *Geometry*.

- **Function MainPlanes(...)** **As Line()** – tato funkce slouží k výpočtu pomocných čar, které jsou průsečnicemi roviny řezu s rovinami danými souřadnicovými osami.
- **Function Cube(...)** **As Single()()** – základní funkcí pro výpočet vrcholů krychle je funkce *Cube*.
- **Function CubeSides(...)** **As Single()()()** – tato funkce vrací jednotlivé strany krychle jako pole vrcholů dané strany.
- **Function CubeLines(...)** **as Line()** – pro výpočet jednotlivých hran krychle slouží tato funkce.
- **Function CubeSecondaryLines(...)** **as Line()** – návratovou hodnotou této funkce je pole pomocných úseček, na kterých jsou umístěny body řez v případě, že neleží na hranách krychle. Této funkce je využito při automatickém výpočtu řezu.

7.2.2. Modul Geometry

Tento modul obsahuje výpočty základní geometrické konstrukcí.

- **Function Vector(...)** **As Double()** – tato funkce slouží pro výpočet vektoru ze dvou daných bodů. Vektor je navrácen, jak pole složek.
- **Function CrossProduct(...)** **As Double()** – k výpočtu vektorového součinu slouží tato funkce.
- **Function Plane(...)** **As Double()** – pomocí této funkce se počítá obecná rovnice roviny z normálového vektoru a bodu roviny.
- **Function PointFromPlaneDistance(...)** **As Double** – touto funkcí lze zjistit vzdálenost bodu od dané roviny.
- **Function FurthestPointFromPlane(...)** **As Single()** – tato funkce slouží ke zjištění nejvzdálenějšího bodu z dané množiny od určené roviny.
- **Function PlaneLineIntersection(...)** **As Single()** – pro zjištění průsečíků roviny a přímky lze použít tuto funkci, která jako návratovou hodnotu vrací souřadnice nalezeného průsečíku.
- **Function PlaneSegmentIntersection(...)** **As Single()** – obdobně jako předchozí funkce je zde počítán průsečík s tím rozdílem, že zde se jedná o průsečík úsečky a roviny.
- **Function HalfSpaceDirection(...)** **As Boolean** – tato funkce slouží ke zjištění, zda daný bod leží v poloprostoru určeném danou rovinou a jejím normálovým vektorem.

- **Function CreateLine(...)** **As Line** – dvě funkce s totožným názvem jsou určeny k vytvoření přímky ze dvou bodů nebo z bodu a směrového vektoru.
- **Function CreateSegment(...)** **As Line** – zde je vytvořena úsečka ze dvou zadaných bodů.
- **Function SegmentIntersection(...)** **As Single()** – tato funkce počítá průsečík dvou úseček.
- **Function IsSamePoint(...)** **As Boolean** – protože není možno dosáhnout absolutní přesnosti při práci s desetinnými čísly na počítači, porovnává tato funkce souřadnice dvou bodů se danou přesností.
- **Function IsSameLine(...)** **As Boolean** – touto funkcí se porovnávají přímky. Srovnání probíhá na základě krajních bodů.
- **Function SegmentPoint(...)** **As Single()** – ze zadaných krajních bodů a z čísla určujícího polohu vypočte tato funkce souřadnice bodu na úsečce.

7.2.3. Modul PointDrawer

- **Sub DrawPoint(...)** – tyto procedury slouží k vykreslení bodu jako krychle malých rozměrů. Obsahují možnost vykreslení v režimu *Select*.

7.2.4. Modul PyramidModel

Tento modul obsahuje nástroje pro geometrickou práci s jehlanem obdobně jako modul *CubeModel* pro práci s krychlí.

- **Function Pyramid(...)** **As Single()()** – tato funkce slouží k výpočtu jednotlivých vrcholů jehlanu.
- **Function PyramidSides(...)** **As Single()()()** – k výpočtu jednotlivých stran jehlanu slouží tato funkce.
- **Function PyramidLines(...)** **As Line()** – zde se vytváří jednotlivé hrany jehlanu.
- **Function PyramidCut(...)** **As Single()** – tato funkce slouží pro výpočet jednotlivých vrcholů řezu jehlanu danou rovinou.
- **Function PyrSecondaryLines(...)** **As Line()** – pomocí této funkce se počítají pomocné čáry pro body mimo hrany jehlanu.

7.2.5. Modul Security

Modul *Security* slouží k zabezpečení jednotlivých souborů, se kterými program pracuje. Využívá k tomu hashovací funkci MD5.

- **Function CheckFile(...)** **As Boolean** – touto funkcí lze ověřit, zda je u zpracovávaného souboru přítomen soubor s kontrolním součtem a jestli je tento součet správný.
- **Sub CreateNewHashFile(...)** – tato procedura vytvoří nový soubor s kontrolním součtem příslušný k zadanému souboru.
- **Function LogIn(...)** **As Boolean** – pomocí této funkce lze přihlásit učitele do programu přes dialog *AdminLogin* a kontrolní funkci *IsRightPassword*.
- **Function ComputeHash(...)** **As String** – zde je obsažen vlastní hashovací mechanismus.
- **Sub ChangePassword(...)** – tato jednoduchá procedura slouží pouze k zobrazení dialogu pro změnu hesla učitele.
- **Function IsRightPassword(...)** **As Boolean** – pomocí této funkce lze zkontrolovat, zda zadané heslo odpovídá uloženému heslu.
- **Function SetNewPassword(...)** **As Boolean** – touto funkcí se provádí vlastní změna hesla učitele na základně zadaného starého a nového hesla.

7.2.6. Modul TextDrawer

Tento modul slouží k vykreslování popisků jednotlivých objektů pomocí bitové mapy. V poli jsou zde uloženy bitové mapy znaků anglické abecedy a číslic.

- **Sub DrawString(...)** – tato procedura provede vykreslení znaků řetězce.

7.3. Používané soubory

Program Geomi používá ke svému běhu kromě knihoven .NET Framework a Tao Framework také několik externích souborů.

7.3.1. Soubor Chapters.xml

Tento soubor slouží k uchování struktury kapitol. Kořenovým elementem je *seznam*. Každá kapitola je pak uložena jako jeden XML element *kapitola*. Tato struktura předpokládá stromové uspořádání, což je prakticky provedeno přidáváním kapitol jako potomků nadřazené kapitoly. V každém elementu je uchována informace o názvu jako atribut *name*, cestě k souboru v atributu *path* a čísle zařazení v *no*.

7.3.2. Soubor Index.xml

Zde jsou uchovány informace o rejstříku kapitol. Uložení kapitol je obdobné jako v souboru `Chapters.xml`, s tím rozdílem, že kořenovým elementem je *rejstrik* že jsou ukládána jako potomci speciálního elementu, které uchovává informace o slovu, ke kterému jsou přiřazeny. Tímto elementem je *slovo* s atributem *name*, jehož hodnotou je slovo zařazené do rejstříku.

7.3.3. Soubory typu excs

Tyto soubory slouží k uchování jednotlivých cvičení. Jejich struktura je z části závislá na tělese, ze kterého se vychází. Kořenovým elementem je *excercise*.

Informace o tělese Obecné informace o tělese jsou uloženy v elementu *object* jako jeho potomci. Jednotlivá obecná nastavení pak jsou určena hodnotou atributu *value* příslušného elementu. Pokud je tělesem krychle, pak se objeví element *size*. V případě jehlanu se zde vyskytují elementy *height*, *baseradius* a *characteristic*.

- **figure** – pokud je hodnota atributu 0, pak se jedná o krychli, pokud je 1, pak tělesem je jehlan.
- **size** – tento element uchovává hodnotu velikosti krychle.
- **height** – zde je uložena výška jehlanu.
- **baseradius** – tento element slouží k uchování poloměru kružnice opsané základně jehlanu.
- **characteristic** – zde je uložen počet vrcholů základny jehlanu.
- **xrot, yrot** – uložena je zde velikost úhlu počáteční rotace kolem osy x , y .
- **shiftx, shifty, shiftz** – v těchto elementech jsou uloženy hodnoty posunutí v ose x , y a z .
- **regularcut** – tento element udává, jestli je příklad řezem s zadanými třemi body.

Dalším elementem je *text*, uvnitř kterého je uložen text zadání příkladu. Za ním následuje element *colors*, ve kterém jsou opět jako potomci uvedeny jednotlivé barvy použité ve cvičení.

- **pointcolor** – zde je uložena barva pro body zadání řezu.
- **linecolor** – hodnotou toho elementu je barva úseček tělesa.

- **backgroundcolor** – tento element obsahuje hodnotu barvy pozadí.
- **cutcolor** – barva zde obsažená je použita na zvýraznění řezu.
- **seclinecolor** – tato barva je užita pro pomocné přímky a úsečky.
- **solidcolor** – v tomto elementu je uložena barva výpně tělesa v náhledu situace.
- **selectioncolor** – touto barvou je zvýrazněn objekt, které uživatel vybere.

Body, přímky, úsečky Dále jsou zde uloženy informace o bodech a přímkách zadání, což je uloženo v elementech *problempoints* a *problemlines*, a bodech a přímkách řešení v elementech *points* a *lines*. Nerozlišují se zde přímky a úsečky, protože v počítačové grafice přímky reálně neexistují.

Každý bod je uložen v samostatném elementu *point* s atributy x , y , z , což jsou jeho souřadnice. Dále pak tento element obsahuje atributy *color*, ve kterém je uložena barva bodu, *label* s popisem bodu a *order*, který v sobě ukrývá umístěn v pořadí v postupu konstrukce. Pokud je k bodu přiřazen vektor pohybu, pak jak potomek je přítomen element *vector* s atributy x , y , z , což jsou jednotlivé složky.

Přímky jsou uloženy jako jako elementy *line* s atributy *color*, *label*, *order* stejného významu jako u bodů. Atributem, který se však u bodů nevyskytuje, je *stipple*, terý udává, zda je vykreslená čára čárkovaná. Krajní body přímky jsou uloženy ve zjednodušených elementech *point* (pouze se souřadnicemi) jako potomci elementu *line*.

7.3.4. Soubory typu test

Soubory typu test jsou také XML dokumenty, zde však s kořenovým elementem *test*. Jako potomci tohoto elementu jsou zde elementy *excercise*, které obsahují jednotlivá cvičení. Jejich struktura je shodná se strukturou elementů *excercise* v samostatných souborech cvičení.

7.3.5. Soubory typu csum

Tyto soubory slouží k uchování kontrolních součtů, které jsou vypočteny pomocí funkcí nabízené modulem Security. Zvláštní postavení má soubor *admin.csm*, ve kterém je uchováno heslo učitele.

7.4. Instalátor

Kvůli potřebě nestandardních knihoven byl k programu vytvořen instalátor přímo jako součást projektu. Takto vzniklý instalační balíček Windows Installer Service (MSI) tyto knihovny zajistí a umožní tak běh programu na daném počítači.

Závěr

Aplikace Geomi byla vytvořena tak, aby plně vyhovovala na začátku formulovaným požadavkům a přitom poskytovala jednoduché a přehledné uživatelské rozhraní. Samotný program nalezne využití přímo v hodinách matematiky, ale také v domácí přípravě žáků s tím, že příslušný obsah by měl být zpracován až danou školou podle učebnic, se kterými škola pracuje. Učitel v této aplikaci získává nástroj pro snadnou tvorbu cvičení a testů. Vzhledem k možnostem nástrojů v okně řezu nebude jeho využití omezeno jen touto problematikou, ale bude použitelný také pro jiné prostorové úlohy.

Conclusions

Computer program Geomi was created to satisfy all the previously formulated demands, providing simple and well-arranged user interface at the same time. This application can be used in lessons of mathematics as well as during students' home preparation, with attention to the fact that the content should be transformed according to the books used by the school. Program Geomi is a teachers' powerful tool which helps them to create exercises and tests very easily. Moreover, owing to possibilities of tools placed in the window of construction, it can also be used for other stereometrical tasks.

Reference

- [1] BOŽEK, Miloš. *Matematika pro II. ročník gymnázií : Geometrie v prostoru*. Praha : Státní pedagogické nakladatelství, 1989. 112 s. Učebnice pro střední školy. ISBN 80-04-24339-8.
- [2] BRANT, Petr. *Dynamická geometrie v prostoru* [počítačový program]. Ver. 2.0.0.24. Praha: Pachner, vzdělávací software, s.r.o., 2007. Dostupné z WWW: <<http://www.pachner.cz>>.
- [3] HOLMAN, Pavel. *Stereometrie* [online]. 2005 [cit. 2010-01-20]. Dostupné z WWW: <<http://stereometrie.webzdarma.cz/menu.html>>.
- [4] JANEČEK, Radovan. *Výuka řezů těles Stereo* [online]. Olomouc, 1996. Diplomová práce. Univerzita Palackého, Přírodovědecká fakulta. Dostupné z WWW: <<http://www.deskriptiva.unas.cz/janecekr.zip>>.
- [5] KADLECOVÁ, Ludmila. *Webová aplikace pro výuku stereometrie* [online]. Praha, 2008. Bakalářská práce. Univerzita Karlova v Praze, Matematicko fyzikální fakulta. Dostupné z WWW: <http://www.karlin.mff.cuni.cz/katedry/kdm/diplomky/ludmila_kadlecova>.
- [6] PETROUSTOS, Evangelos. *Myslíme v jazyku Visual Basic.NET : knihovna programátora. 1. díl*. Praha : Grada Publishing, 2003. 675 s. Myslíme v ... ISBN 80-247-0371-8.
- [7] PETROUSTOS, Evangelos. *Myslíme v jazyku Visual Basic.NET : knihovna programátora. 2. díl*. Praha : Grada Publishing, 2003. 538 s. Myslíme v ... ISBN 80-247-0372-6.
- [8] POMYKALOVÁ, Eva. *Matematika pro gymnázia : Stereometrie*. 3. vydání. Praha : Prometheus, 2001. 223 s. Učebnice pro střední školy. ISBN 80-7196-178-7.
- [9] *Cabri 3D* [počítačový program]. Ver. 2.1.2. Grenoble: Cabrilog SAS, 2007. Dostupné z WWW: <<http://www.cabri.com>>. Pouze zkušební verze.
- [10] *Microsoft Developer Network* [online]. 2010. Dostupné z WWW: <<http://www.msdn.com>>.
- [11] *OpenGL Programming Guide* [online]. Ver. 1.1. 1997. Dostupné z WWW: <<http://www.glprogramming.com/red/>>.
- [12] *Rámcový vzdělávací program pro gymnaziální vzdělávání*. Praha : Výzkumný ústav pedagogický v Praze, 2007. 104 s. Dostupné z WWW: <<http://www.vuppraha.cz>>. ISBN 978-80-87000-11-3.

- [13] *Root.cz: Grafická knihovna OpenGL*. Root.cz, 2003– . Dostupné z WWW: <http://www.root.cz/serialy/graficka-knihovna-opengl/>
- [14] *Říše čísel* [počítačový program]. Praha: LANGMaster International, s. r. o., 2006.
- [15] *Tao Framework* [online]. 2010.
Dostupné z WWW: <http://www.taoframework.com>.
- [16] *Visual Basic .NET* [online]. 2010.
Dostupné z WWW: <http://www.vbnet.com>.

A. Obsah příloženého CD

Příložené CD obsahuje zdrojové soubory, sestavenou aplikaci a další data uspořádaná v následujících adresářích:

`bin/`

Tento adresář obsahuje instalátor `GEOMI-SEUP.MSI`, jehož spuštění provede vlastní instalaci. Dále je zde přítomen soubor `SETUP.EXE`, který je automaticky generován při tvorbě instalačního balíčku a slouží k případnému dodání příslušné verze Windows Installer Service.

`doc/`

Zde je obsažena dokumentace k práci ve formátu PDF a dokument, který slouží jako nápověda ve vlastním programu. K nim jsou tu umístěny jejich zdrojové soubory, obrázky v nich obsažené a také použité soubory závazných stylů.

`src/`

V tomto adresáři jsou obsaženy všechny zdrojové soubory programu a také instalátor knihovny Tao Framework `TAOFramework-2.1.0-SETUP.EXE`, která je nutná k sestavení programu.

`readme.txt`

Instrukce k instalaci a ke spuštění jsou uloženy v tomto souboru.

Dále CD obsahuje:

`data/`

Zde jsou uloženy ukázkové soubory, které je možno v programu importovat.