

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PARAMETRICKÝ GEOMETRICKÝ NÁČRTNÍK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ROMAN ČIŽMARIK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PARAMETRICKÝ GEOMETRICKÝ NÁČRTNÍK

PARAMETRIC GEOMETRY SKETCH TOOL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ROMAN ČIŽMARIK

VEDOUcí PRÁCE

SUPERVISOR

Prof. Dr. Ing. ZEMČÍK PAVEL

BRNO 2015

Abstrakt

Cílem této práce je navrhnout a implementovat parametrický geometrický náčrtník. Řešení tohoto problému se skládá ze dvou částí, vytvoření knihovny základních funkcí dynamické geometrie a grafického uživatelského rozhraní. Funkcionalitu vytvořené knihovny demonstrují dvě uživatelská rozhraní, každé jiným způsobem. Pomocí výsledné knihovny a toolkitu Qt je možné rychle a jednoduše vytvářet nové geometrické náčrtníky.

Abstract

The aim of this work is to design and implement parametric geometry sketchtool. The solution to this problem consists of two parts, to create a library of base dynamic geometry functions and GUI. Functionality of the created library will be demonstrated by two user interfaces, each using the library in different way. Using the resulting library and toolkit Qt allows to quickly and easily create new geometric sketchtools.

Klíčová slova

Parametrický geometrický náčrtník, dynamická geometrie, 2D zobrazení, Qt.

Keywords

Parametric geometry sketch tool, Dynamic geometry, 2D view, Qt.

Citace

Roman Čížmarik: Parametrický geometrický náčrtník, bakalářská práce, Brno, FIT VUT v Brně, 2015

Parametrický geometrický náčrtník

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Prof. Dr. Ing. Pavla Zemčíka a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Roman Čížmarik
19. května 2015

Poděkování

Chtěl bych poděkovat panu Prof. Dr. Ing. Pavlu Zemčíkovi za odborné vedení práce a cenné rady, návrhy a připomínky, kterými mi pomohl při vypracování této práce.

© Roman Čížmarik, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Súčasný stav dynamickej geometrie	5
2.1	História	5
2.2	Euklidovská a neeuklidovská geometria	6
2.3	Dynamická geometria a DGE	7
2.4	Použitie	8
3	Prehľad súčasného stavu geometrických náčrtníkov	9
3.1	Obecný prehľad	9
3.2	Cabri II PLUS	10
3.3	GeoGebra	11
3.4	Cinderella	12
3.5	TI-Nspire CAS Student Software	14
4	Zhodnotenie súčasného stavu a analýza zadania	15
4.1	Zhrnutie súčasného stavu	15
4.2	Motivácia	16
4.3	Vypracovanie zadania	16
5	Návrh	17
5.1	Klasický prístup k návrhu dátových štruktúr	17
5.2	Geometrické objekty	18
5.2.1	Bod	19
5.2.2	Úsečka a priamka	19
5.2.3	Polygón	20
5.2.4	Kružnica	20
5.3	Zobrazenie geometrických objektov	21
5.4	Používateľské rozhranie geometrického náčrtníku	22
5.5	Používateľské rozhranie parametrického náčrtníku	23
5.6	Zhrnutie funkcionality	24
6	Implementácia	25
6.1	Signály a sloty	26
6.2	Vytváranie geometrických objektov	27
6.3	Vytváranie väzieb medzi objektmi a ich pohyb	27
6.4	Ukladanie a načítanie dát	28

7 Testovanie	29
8 Záver	31
A Obsah CD	34

Kapitola 1

Úvod

V 80. rokoch minulého storočia viedol výrazný pokrok v oblasti informatiky k vzniku nového druhu software, parametrických kresliacich nástrojov. Tento druh software ponúka možnosť vytvoriť základné geometrické primitíva ako sú úsečka, priamka alebo kružnica, a nahradil tak klasické nástroje ako pravítko a kružidlo pri vytváraní geometrických konštrukcií. Zjavnou výhodou použitia počítaču v tejto oblasti je presnosť a rýchlosť. V niekoľko málo krokoch je napríklad možné zostrojiť zložitejšie geometrické konštrukcie, nájsť priesečník dvoch priamok alebo presne zistiť polomer kružnice. Hlavnou výhodou parametrických geometrických náčrtníkov však je, že vytvorený náčrtok nie je iba statický obrázok. Náčrtok sa skladá z grafických objektov medzi ktorými existujú logické väzby, ktoré odpovedajú klasickej geometrii, napríklad priamka je určená dvoma bodmi, trojuholník zasa úsečkami medzi tromi bodmi. S jednotlivými objektmi je možné manipulovať a meniť ich, napríklad uchopením a posunutím bodu, alebo nastavením polomeru kružnice. Používateľ tak môže okamžite pozorovať transformáciu konštrukcie, môže vidieť, ktoré časti sa dynamicky menia a ktoré naopak ostávajú statické. Tieto aplikácie využívajú nový druh geometrie, tzv. dynamickú geometriu, ktorá sa odlišuje od tej klasickej. Jej základnou myšlienkou je, že jednotlivé objekty môžu dynamicky meniť svoje parametre. Tato vlastnosť však so sebou prináša aj nové problémy a otázky, s ktorými sa klasická geometria nepotýka.

Tieto aplikácie našli uplatnenie predovšetkým v školstve. Pri výučbe matematiky a fyziky na základných alebo stredných školách môžu predstavovať neoceniteľnú pomoc. Žiakom pomáhajú pochopiť základné princípy geometrie, vizualizuje geometrické transformácie a ponúkajú zaujímavé interaktívne prostredie, ktoré môže spestriť inak nezaujímavé učivo. Pre učiteľov zasa môžu byť prostriedkom ako ľahko a záživne vysvetliť preberanú látku, prípadne môžu žiakom zadávať nové typy úloh spojené s využitím takýchto aplikácií.

Táto téma ma zaujala už na prvý pohľad. Vybral som si ju najmä preto, lebo v sebe spája dve oblasti, ktoré ma pri štúdiu informatiky zaujali najviac, a síce matematiku a tvorbu používateľských rozhraní. Ďalším dôvodom bol aj fakt, že som už mal s pracou s geometrickými náčrtníkmi skúsenosť a chcel som sa o ich fungovaní dozvedieť viac.

Cieľom tejto práce je vytvoriť knižnicu základných funkcií dynamickej geometrie. Mala by umožňovať najmä vytvorenie a manipuláciu s geometrickými objektmi. Funkčnosť knižnice bude demonštrovaná vytvorením grafického používateľského rozhrania, ktoré bude predstavovať jednoduchý geometrický náčrtník. Tento náčrtník bude schopný vytvárať a upravovať geometrické objekty, spôsobom typickým pre tento typ aplikácií. K nemu bude existovať druhé používateľské rozhranie, ktoré sa zameria na maximálne využitie možnosti parametrickej úpravy konštrukcie. Aplikácie budú kompatibilné. To znamená, že prvá aplikácia bude schopná načítať náčrtok vytvorený v tej druhej a naopak.

Nasledujúca kapitola objasňuje najdôležitejšie pojmy a termíny spojené s touto prácou. Tretia kapitola obsahuje prehľad niektorých geometrických náčrtníkov, ktoré sú v súčasnosti na trhu dostupné. Kapitola 4 zhŕňa poznatky získané v predchádzajúcej kapitole a s ohľadom na ne analyzuje zadanie práce. Ďalej nasleduje popis návrhu a implementácie vlastnej práce. V kapitole Testovanie sú uvedené postupy využité pri overovaní funkčnosti výslednej knižnice. Zaver sumarizuje dosiahnuté výsledky a načrtáva možnosti ďalšieho vývoja.

Kapitola 2

Súčasný stav dynamickej geometrie

Táto kapitola je úvodom do problematiky dynamickej geometrie. Prvá časť sa venuje histórii dynamickej geometrie a vzniku parametrických kresliacich nástrojov. Ďalej sa v kapitole nachádza vymedzenie najdôležitejších pojmov súvisiacich s touto pracou, ako napríklad euklidovská a neeuklidovská geometria, dynamická geometria a geometrický náčrtník. Cieľom tejto kapitoly ani práce však nie je podať encyklopedický výklad o týchto termýnoch. Kapitola je ukončená popisom využitia geometrických konštrukčných nástrojov v praxi.

2.1 História

Súčasný stav geometrických konštrukčných nástrojov, ktorými sa budeme zaoberať v nasledujúcej kapitole, je veľmi úzko spätý s historickým vývinom dynamickej geometrie. Preto považujem za dôležité uviesť krátky náhľad do jej histórie, objasniť príčiny a okolnosti jej vzniku.

Na začiatku 80. rokov minulého storočia zaznamenala oblasť výpočtovej techniky významný pokrok príchodom mikroprocesoru. Mikroprocesory umožnili znížiť počet obvodov na základnej doske počítača, zvýšila sa spoľahlivosť, zmenšili rozmery, zvýšila rýchlosť a kapacita pamäti. Nastáva ústup strediskových staníc v prospech osobných počítačov [16].

S príchodom osobných počítačov vznikla aj nová disciplína, dynamická geometria, ktorá ako prvá začala využívať počítače na zostrojenie základných geometrických objektov. Výhodou bola nie len zvýšená presnosť, ale hlavne dynamickosť. Software je schopný rýchlo reagovať na zmeny akými sú uchopenie a premiestnenie objektu alebo zmena niektorých jeho parametrov. Táto vlastnosť umožňuje v reálnom čase pozorovať zmeny geometrických objektov, môžeme vidieť, ktoré časti objektu sa menia a ktoré ostávajú statické. Priekopníkom v tejto oblasti je software *Cabri Geometry*.

V roku 1981 sa skupina francúzskych vedcov z oblasti kombinatoriky a teórie grafov rozhodla vytvoriť počítačový program, ktorý by pomohol s výskumom v oblasti teórie grafov. Vznikol tak projekt, ktorého cieľom bolo sprístupniť a rozšíriť nové technológie do oblasti vedy a výskumu. Tento projekt bol nazvaný „*Cabier de Brouillon Informatique for Graph Theory, Cabri-Graph*“. Spočiatku sa Cabri stretával so skepticizmom a nedôverou hlavne zo strany matematickej komunity. Tá zastávala názor, že dynamické geometrické prostredie (*DGE - dynamical geometrical environment*) eliminuje kreatívne myslenie v matematike, pretože okamžite zobrazovalo vlastnosti, ktoré by inak museli byť pracne dopočítané.

Dynamická geometria sa potýkala s rôznymi problémami. Jedným z nich bola nejednoznačnosť : ako určiť priesečník priamky a kružnice? Ďalším bolo správanie sa objektov

pri pohybe: ako sa ma správať bod P , patriaci priamke AB , pri pohybe bodu A alebo B ? Vývojársky tím projektu Cabri definoval tieto problémy ako teoretické problémy nového druhu geometrie, dynamickej geometrie, ktorá sa odlišovala od klasickej euklidovskej geometrie. Existencia dynamickej geometrie bola odhalená medzinárodnému publiku v roku 1989 na konferencii v Grenoble, ktorá bola organizovaná vývojárskym tímom Cabri. V tom istom roku bol Cabri Geometry publikovaný vo Francúzsku a získal ocenenie *Apple Award* [9]. Cabri a iné DGE sa čoskoro stali bežnou súčasťou vyučovania matematiky a geometrie v mnohých krajinách. DGE pomáhajú študentom pochopiť geometrické vlastnosti objektov, učiteľom zasa ponúkajú možnosť vyučovanie spestriť.

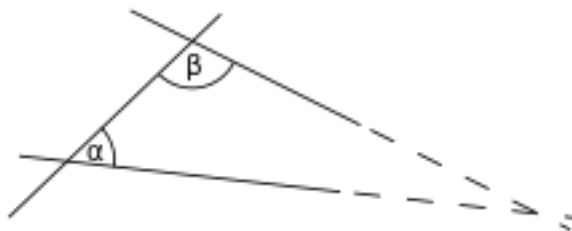
Dnes existuje široká škála geometrických náčrtníkov pracujúcich v 2D ale aj 3D priestore, od jednoduchých aplikácií so základnou funkcionalitou, cez on-line náčrtníky až po profesionálne systémy typu CAD. Základy dynamickej geometrie, ktoré tieto aplikácie implementujú ostali od ich definovanie nezmenené [9]. Dôkazom môže byť aj samotný software Cabri, ktorý je na trhu a stále vo vývoji do dnešného dňa, hoci sa jeho jadro nezmenilo [4]. V súčasnosti je možné pozorovať trend rozšírenia tohto druhu aplikácií na mobilné telefóny, tablety ale aj kalkulačky.

2.2 Euklidovská a neeuklidovská geometria

Prevažná väčšina súčasných náčrtníkov podporujúcich dynamickú geometriu pracuje v euklidovskej rovine, čiže v rovine spĺňajúcej Euklidove axiómy. Euklides bol grécky matematik, ktorý žil v rokoch 325 a 260 pr.n.l. Vo svojom diele „Základy“ (starogrécky *Stoicheia*) definoval päť axiómou, ktoré musí geometria v rovine spĺňať. Dnes existujú rôzne formy a preklady euklidovských axiómou. Je preto nutné uviesť, že nasledujúca formulácia bola prezvaná z článku Mgr. Lukáša Krumpa, Ph.D [12]:

1. Každými dvoma body lze vest úsečku. (Dnes bychom řekli, že dva různé body jsou koncovými body právě jedné úsečky.)
2. Tuto úsečku lze libovolně prodloužit oběma směry. (Tedy úsečka určuje přímku, která je „na obě strany nekonečna“.)
3. Každou úsečkou lze opsat kružnici. (Neboli je-li dan bod a kladný poloměr, je tím jednoznačně určena kružnice.)
4. Všechny pravé uhly jsou si rovny. (Pravý uhel je podle definice polovina přímého uhlu, axiom tvrdí, že jeho velikost nezávisí na poloze (a pro další účely ji budeme značit $\pi/2$).)
5. Jsou-li dany tři přímky p, q, r tak, že r protíná přímky p, q pod uhly α, β na stejné straně přímky r , a platí-li $\alpha + \beta < \pi$, pak p a q se protínají, a sice na té straně přímky r , kde jsou uhly α, β .

Dôvodom vzniku neeuklidovskej geometrie je práve piaty axióm. Od doby jeho vzniku bolo vytvorených mnoho pokusov tento axióm dokázať, ale výsledkom bolo vždy len prevedenie na iné, ekvivalentné tvrdenie. Až v 19. storočí ruský matematik Nikolaj Ivanovič Lobačevskij predstavil geometriu v rovine, ktorá spĺňala prvé štyri axiómy a opak piateho. Lobačevskú rovinu označujeme ako rovinu hyperbolickú [12]. Ďalším zástupcom neeuklidovskej geometrie je napríklad Riemannová, tiež označovaná ako eliptická, geometria [10]. Neeuklidovskou geometriou sa obecné rozumie taká geometria, ktorá nespĺňa niektorí z vyššie uvedených axiómou.



Obrázek 2.1: Euklidov piaty axióm

2.3 Dynamická geometria a DGE

Ako už bolo spomenuté vyššie dynamická geometria je vedná disciplína, ktorá vznikla v 80. rokoch minulého storočia. Veľmi zjednodušené by sa dalo povedať, že je to „geometria na počítači.“ Čím sa teda odlišuje od klasickej euklidovskej geometrie? Platia v nej rovnaké zákonitosti ako v euklidovskej geometrii popísané vyššie. Najväčším rozdielom je dynamickosť. Termín „dynamický“ sa v matematike vzťahuje k myšlienke pohybu a zmien [7]. Dynamická geometria pracuje s myšlienkou, že jednotlivé objekty konštrukcie môžu meniť svoje parametre, ako napríklad polohu alebo dĺžku, pričom musia zostať zachované vzťahy definované medzi týmito objektmi. To znamená, že s jednotlivými objektmi je možné ľubovoľné manipulovať a konštrukcia sa nesmie dostať do nekonzistentného stavu. Tu ale vzniká problém, s ktorým sa klasická geometria nepotýka a síce nejednoznačnosť. Príkladom môže byť priesečník priamky a kružnice. Tieto objekty nemusia mať žiadny, jeden alebo dva priesečníky a tento stav sa môže dynamicky meniť počas pohybu jedného z objektov. Ulrich Kortenkamp označuje tento problém ako fundamentálny problém dynamickej geometrie [11]. Rovnaký problém môže vzniknúť aj v inom kontexte, napríklad pri analýze funkcií v Riemannovej rovine [11]. Ten istý autor definuje dynamickú geometriu nasledovne [11]:

„Dynamická geometria je teória opisov konštrukčného charakteru, pojednávajúci o funkčných objektoch, pod zmenou parametrov“ (Kortenkamp)

Z tejto definície je zrejmé, že ťažiskom dynamickej geometrie je adekvátna reakcia objektov na akúkoľvek zmenu parametrov.

Geometrické náčrtníky (DGE), sú programy, ktoré implementujú princípy dynamickej geometrie. Medzi základné vlastnosti väčšiny geometrických náčrtníkov patrí možnosť sledovať jednotlivé kroky konštrukcie, prípadne sa vrátiť späť, alebo už vytvorený objekt zmazať. Ďalej je to samozrejme možnosť modifikácie geometrických objektov, či už parametricky alebo v tzv. móde posunutia (*drag-mode*). Príkladom parametrickej zmeny môže byť zadanie presných koordinátou bodu, dĺžky úsečky, veľkosti uhlu alebo polomeru kružnice. Módom posunutia sa obecné rozumie možnosť uchopiť objekt pomocou myši a presunúť ho v reálnom čase [14]. V tejto funkcii sa od seba súčasné geometrické náčrtníky odlišujú, najmä čo sa týka presunu zložitejších objektov, ako napríklad obecný polygón, ktoré sa skladajú z viacerých elementárnych geometrických objektov.

2.4 Použitie

Geometrické konštrukčné nástroje si nemuseli svoje uplatnenie dlho hľadať. Už krátko po vzniku prvých aplikácií tohto typu, na konci 80. rokov minulého storočia, boli nasadené ako pomôcky pri vyučovaní matematiky a geometrie na základných a stredných školách. Počiatok používania software tohto typu v školách bol podporený aj dostupnosťou handheld kalkulačiek od firmy Texas Instruments. Prvý krát bol geometrický náčrtník (Cabri Geometry) použitý pri výučbe v roku 1988 vo Švajčiarskom kantóne Vaud [9].

Geometrické náčrtníky poskytujú pre študentov určitú formu abstrakcie pri geometrických transformáciách. Obzvlášť nápomocne môžu byť pri štúdiu chovania trojrozmerných objektov, ktoré nemusí byť jednoduché si predstaviť [13]. Za zmienku určite stojí aj využitie doplnku fyzikálnych simulácií (napr. Cinderella) na ilustráciu základných fyzikálnych javov [5].

Ďalším možným využitím sú CAD systémy [11]. Tieto systémy sú však omnoho sofistikovanejšie a zvyčajne pracujú v 3D. Využívajú sa najmä v oblasti architektúry a strojného inžinierstva.

Kapitola 3

Prehľad súčasného stavu geometrických náčrtníkov

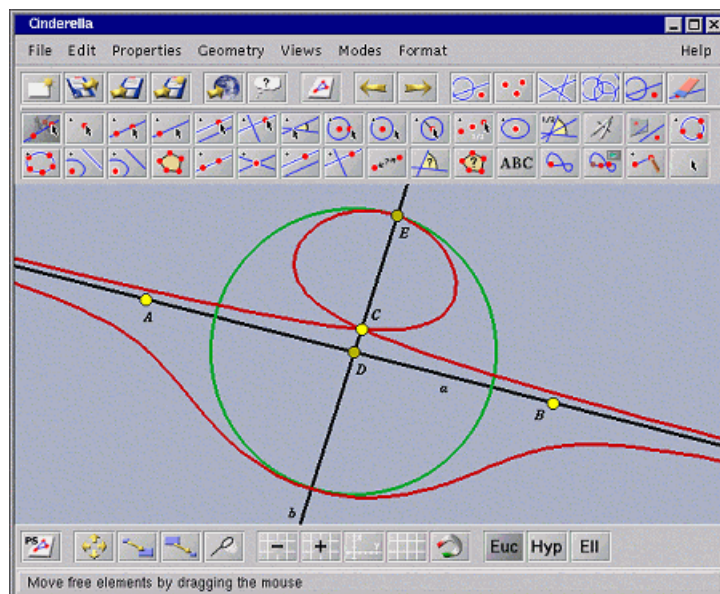
V tejto kapitole je predstavených niekoľko zástupcov súčasných parametrických konštrukčných nástrojov. Vzhľadom na zameranie tejto práce boli vybrane nástroje, ktoré primárne pracujú s euklidovskou geometriou v 2D rovine. Bližšie sú popísané ich používateľské rozhrania, signifikantné vlastnosti a možnosti.

Obsah tejto kapitoly nie je encyklopedický a nedáva si za cieľ popísať všetky dostupné nástroje. Nástroje popísané v tomto prehľade sú azda najznámejšími na trhu a ponúkajú oproti konkurencii zaujímavú funkcionálnosť. Nejedná sa však o popis kompletnej funkcionality ani vymenovanie všetkých vlastností.

3.1 Obecný prehľad

V dnešnej dobe existuje mnoho rôznych nástrojov podporujúcich dynamickú geometriu. Môže sa jednať o klasické desktopové aplikácie pracujúce s 2D alebo 3D geometriou, webové applety alebo špecializovaný software pre handeld kalkulačky. Prevažná väčšina z nich pracuje s euklidovskou geometriou, ale existujú samozrejme aj náčrtníky podporujúce hyperbolicú či eliptickú geometriu. V tomto prehľade budú uvedené aplikácie, ktoré sa zameriavajú na zobrazenie 2D roviny. Pri skúmaní vybraných aplikácií sa zameriame na používateľské rozhranie, funkcionálnosť, export dát, dostupnosť, prípadne budú vyzdvihnuté niektoré špecifické vlastnosti.

Typické používateľské rozhranie geometrického konštrukčného nástroja môžeme vidieť na obrázku 3.1. Používateľské rozhrania tohto typu aplikácie sú v zásade veľmi jednoduché a prehľadné. Obecne je možné povedať, že sa skladajú z troch základných častí a to: klasický menu bar, nástrojovú lištu so základnými funkciami a náčrtnia [15]. Menu bar ponúka prístup k funkciám ako načítanie a uloženie, export, alebo nastavenia používateľského rozhrania. Nástrojová lišta umožňuje rýchly prístup k základným funkciám programu. Funkcie sú logicky zotriedené do skupín a každá skupina je reprezentovaná jednou ikonou na lište. Po kliknutí na ikonu sa zvyčajne zobrazia ostatné funkcie patriace do danej skupiny. Dominantou rozhrania je náčrtnia. Je to plocha, do ktorej používateľ vkladá objekty a kde s nimi môže ďalej pracovať. Typickými vlastnosťami náčrtnia býva priblíženie, oddialenie, zobrazenie osy alebo kockovej mriežky. Menej typickou súčasťou používateľského rozhrania je algebrické okno (napr. GeoGebra). To prehľadne zobrazuje jednotlivé geometrické objekty, ktoré používateľ vytvoril [6].



Obrázek 3.1: Ukážka používateľského rozhrania. Jedná sa o aplikáciu Cinderella

K základnej funkcionalite takmer všetkých geometrických náčrtníkov patrí možnosť zostrojenia alebo zmazanie geometrických objektov, interakcia s nimi, meranie dĺžok, obvodov, objemov či uhlov. Pokročilou funkcionalitou môžeme označiť možnosť vytvárania používateľských makier alebo skriptovanie. Pomocou makier je možné rýchlo znova aplikovať preddefinované predpisy. To umožňuje napríklad vytvoriť makra na konštrukciu jednoduchých objektov a následne z nich zostrojiť zložitejšie konštrukcie. Skriptovaním sa rozumie možnosť vytvárať geometrické konštrukcie pomocou textových inštrukcií. Za týmto účelom je zvyčajné vytvorený vlastný skriptovací jazyk. Nevšednými sú taktiež funkcie ako animácie alebo matematické dôkazy.

Existuje niekoľko spôsobov ako uložiť vytvorenú konštrukciu. Najbežnejším je export scény do statického obrázku rôznych formátov, napríklad PNG, BMP, Scalable Vector Graphics (SVG), alebo PDF. Ďalšou možnosťou je export do interaktívnej webovej stránky v jazyku HTML. Túto možnosť ponúkajú najmä aplikácie napísané v jazyku Java.

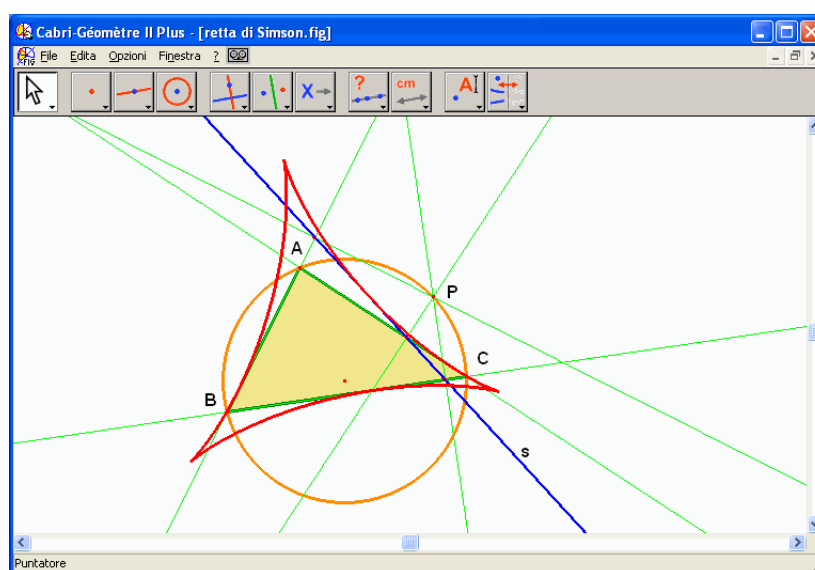
Väčšina súčasných geometrických náčrtníkov je voľne dostupná pod proprietárnou licenciou. Niektoré však majú limitovanú skúšobnú dobu (typický 30 dní). Po jej vypršaní je funkčnosť aplikácie značne obmedzená (Cabri II Plus) alebo je nutné program zakúpiť (TI-Nspire). Vo zvyšku kapitoly budú popísané štyri vybrané aplikácie. Zvolené boli aplikácie, ktoré poskytujú neobvyklú funkčnosť, prípadne zaujímavé rozšírenia.

3.2 Cabri II PLUS

Produkty firmy Cabrilog patria medzi najznámejšie programy zaoberajúce sa dynamicou geometriou. Táto firma pôsobí na trhu už vyše 20 rokov a jej produkty sa používajú najmä ako výukové programy. Azda najvýznamnejším je Cabri II Plus. Tento 2D geometrický náčrtník má slúžiť hlavne ako pomôcka pri vyučovaní geometrie, algebry alebo fyziky na stredných a základných školách. Existuje aj jeho varianty pracujúce s 3D geometriou (Cabri3D) a verzia pre kalkulačky spoločnosti Texas Instruments (Cabri Jr.).

Cabri II Plus je charakteristický prehľadným užívateľským rozhraním, kde sú všetky

funkcie programu zahrnuté v niekoľkých ikonách s výstižnými obrázkami. Centrálnu časť rozhrania tvorí samozrejme nákresňa, ako je u programov tohto typu zvykom. Pod ňou sa nachádza stavový riadok, ktorý používateľovi oznamuje aká funkcia programu je momentálne zvolená. Práca s programom je intuitívna a nápadne sa podoba rysovaníu pomocou pravítka a kružidla. Medzi základne objekty, ktoré je možné skonštruovať patria: bod, priamka, úsečka, vektor, polygón a kružnica. Program ponúka širokú škálu funkcií nad týmito objektmi, napríklad: zostrojenie rovnobežky, kolmice, stredového bodu, rotácia objektu okolo osy alebo bodu, vyčíslenie vzdialenosti, dĺžky, objemu alebo uhlu. Samozrejmosťou je modifikovanie objektov ťahaním myšou. Pri každej akcii sa užívateľovi pri kurzore myši zobrazí krátka napovedá, ktorá mu oznamuje aká funkcia, prípadne s ktorým objektom bude vykonaná. Cabri II Plus ponúka aj možnosť vytvorenia animácie, pomocou ktorých sa dajú vytvoriť zaujímavé fyzikálne simulácie. Ďalšími zaujímavými funkciami sú modifikácia objektu pomocou rovníc, skriptovanie či možnosť vytvorenia používateľských makier. Program podporuje exportovanie do HTML, PNG, JPG a BMP.



Obrázek 3.2: Cabri II Plus

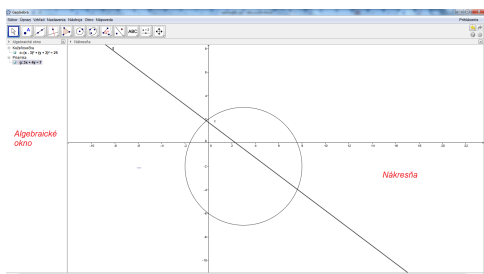
Nevýhodou programu je, že nie je voľne dostupná plná verzia a licencia je pomerne nákladná. Po vypršaní skúšobnej doby, sa program spustí v demo verzii, s ktorou je možné pracovať maximálne 15 minút a dáta nie je možné uložiť ani exportovať [4].

3.3 GeoGebra

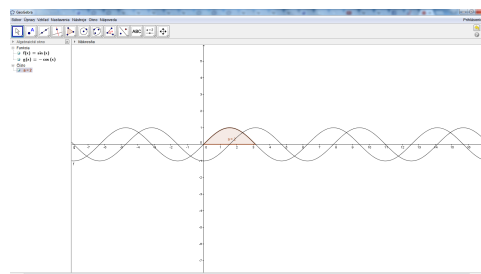
Tvorcom GeoGebry je Markus Hohenwarter, ktorý ju začal vyvíjať v roku 2001 na Univerzite Salzburg ako súčasť svojej diplomovej práce. GeoGebra je multiplatformný software, ktorý spojuje dynamickú geometriu, algebru a matematickú analýzu. Nie je to iba geometrický náčrtník, ponúka radu ďalších funkcií ako napríklad vstavaný CAS alebo tabuľkový procesor.

Užívateľské rozhranie je podobne ako u Cabri II Plus. Funkcie programu sú roztriedené do skupín, ktoré sú reprezentované ikonami s obrázkami. Základnými geometrickými objektmi GeoGebry sú bod, vektor, priamka, polygón, kruhové výseky a matematické funkcie. Objekty je možné vytvárať a manipulovať s nimi rovnako ako v ostatných geometrických

náčrtníkoch. GeoGebra obsahuje algebrické okno, v ktorom sú zobrazené objekty popísané matematickými rovnicami. Pri manipulácii s objektmi je možné sledovať ako sa ich algebrický popis dynamicky mení, respektíve je možné s objektom manipulovať upravením príslušnej rovnice. Objekty je taktiež možné vytvárať zadáním rovnice do vstupného poľa, napríklad: priamka g môže byť zadaná ako: $g = 3x + 4y = 7$ alebo kruh c ako $c = (x - 3)^2 + (y + 2)^2 = 25$.



Obrázek 3.3: GeoGebra – základné geometrické objekty



Obrázek 3.4: GeoGebra – základné matematické funkcie a integrály

Zadaná rovnica nemusí popisovať iba geometrické objekty, môže to byť ľubovoľná matematická funkcia, napríklad funkciu sínus je možné zadať ako $f(x) = \sin(x)$. GeoGebra ponúka aj funkcie akými sú určitý a neurčitý integrál, alebo derivácia funkcie. Výsledkom neurčitého integrálu a derivácie je nová funkcia s odpovedajúcou rovnicou. Výsledkom určitého integrálu je obsah plochy, ktorá je graficky znázornená na nákrese.

Vytvorené konštrukcie môžu byť exportované do rôznych formátov statických obrázkov (PNG, SVG, PDF, EMF) alebo animovaných obrázkov GIF. GeoGebra ponúka aj možnosť vytvorenia dynamickej stránky vo forme HTML kódu alebo vygenerovanie kódu pre LaTeX . Od roku 2011 je možné konštrukcie ukladať priamo na GeoGebra Tube, oficiálny GeoGebra repozitár obsahujúci voľne dostupné interaktívne výučné materiály. Program je napísaný v jazyku Java a okrem desktopovej aplikácie je na webe dostupný spustiteľný applet.

GeoGebra je voľne dostupný open-source software pod licenciou GPL pre nekomerčné účely. Pre komerčné účely je potrebné zakúpiť špeciálnu licenciu. Zaujímavosťou je, že aplikácia je dostupná až v 55 jazykových lokalizáciách, vrátane češtiny a slovenčiny [6].

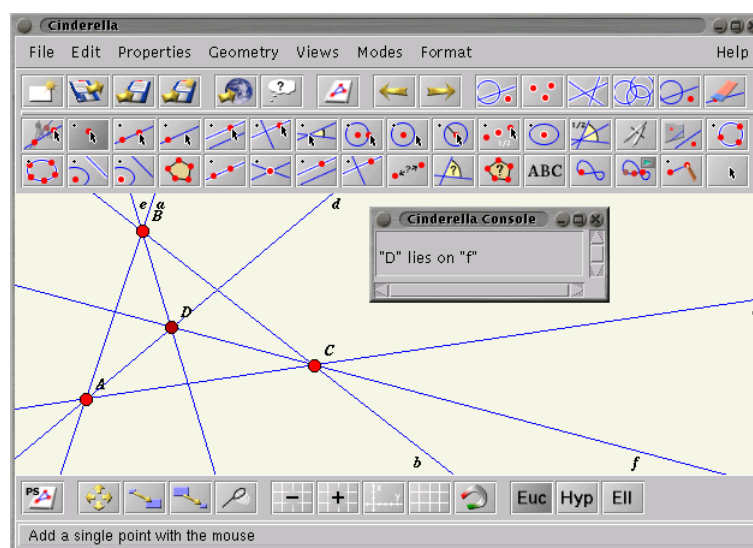
3.4 Cinderella

Cinderella je ďalším typickým zástupcom DGE. Tento projekt vznikol v roku 1992 počas konferencie kombinatoriky na *Mittag-Leffler Institute* vo Švédsku. Jeho tvorcami sú Henry Crapo a Jürgen Richter-Gebert. Projekt bol spočiatku vyvíjaný na NeXT platforme, ktorá bola v tom čase známa vďaka svojej softwareovej architektúre. Už po niekoľkých týždňoch bol vytvorený funkčný prototyp. V auguste roku 1996 sa jeden zo zakladajúcich členov projektu, Jürgen Richter-Gebert, a jeho kolega Ulrich H. Kortenkamp rozhodli začať celý projekt úplne od začiatku a prepísať ho do jazyku Java, pretože platformy NeXT sa už skoro vôbec nepoužívali. Cieľom druhého projektu bolo zachovať funkcionality prvého a rozšíriť ho o rysy neeuklidovskej geometrie. Druhé vydanie programu bolo zverejnené v septembri 1996. Táto verzia vyhrala ocenenie „*Multimedia Innovation Award*“, ale autori s ňou aj napriek tomu neboli spokojní. Program trpel nestálosťami spôsobenými matematickými nejednoznačnosťami akou je napríklad priesečník priamky a kružnice. Tieto nejednoznačnosti

mohli v programe vyústiť k nekonzistenciám konštrukcie. Preto došlo v roku 1998 k tretiemu, finálnemu, prepísaniu programu [15].

Používateľské rozhranie je veľmi podobne ako u predchádzajúcich zástupcov. Cinderella umožňuje zostrojenie bodov, priamok, kružníc, kužeľosečiek, polygónov, meranie vzdialenosti a uhlov. K základnej funkcionalite ďalej patrí zostrojenie priesečníkov priamok a kužeľosečiek, rovnobežiek, kolmíc a uhlových symetrií.

Cinderella navyše ponúka niekoľko zaujímavých funkcií, ktoré žiadna z vyššie spomenutých aplikácií nemá. Provou je tzv. „*theorem checker*“. Je to funkcia, ktorá skúma všetky kroky, ktoré používateľ pri konštrukcii urobil a hlási všetky netriviálne teoremy, ktoré nájde (obr. 3.5).



Obrázek 3.5: Cinderella – „theorem checker“

Ďalej je to možnosť zobraziť jednu konštrukciu vo viacerých pohľadoch (oknách). Tie môžu byť rôzne priblížené, respektíve oddialené, alebo môžu zobrazovať rozdielne časti konštrukcie. Zmeny prevedené v hociktorom z pohľadov sa prejavujú vo všetkých ostatných [11].

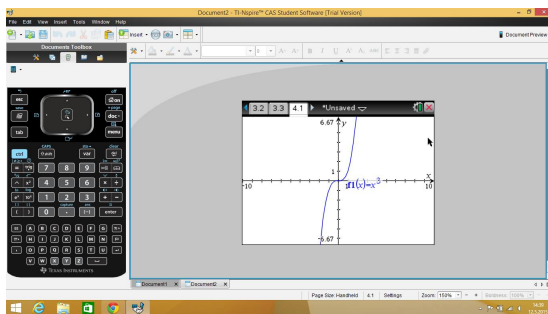
Ďalším významným doplnkom je prostredie pre vytváranie fyzikálnych simulácií Cyn-diLab [5]. Vytváranie modelu a navigácia v tomto prostredí sú podobne geometrickej časti. Používateľ si vyberie z ponuky elementov, ako napríklad voľná hmota, gravitácia alebo pružina, vytvorí svoj model a spustí simuláciu. Počas behu simulácie je možné kedykoľvek do modelu zasiahnuť. Pomocou skriptovacieho jazyka CindyScript je možné k vytvoreným konštrukciám a simuláciám pridať špecifické chovanie.

Za zmienku určite stojí aj fakt, že Cinderella podporuje zobrazovanie nie len v euklidovskej rovine, ale tiež hyperbolickej a eliptickej. Medzi jednotlivými zobrazeniami je možné jednoducho prepínať a prípadné akcie v hociktorej z rovín budú korektne interpretované [11].

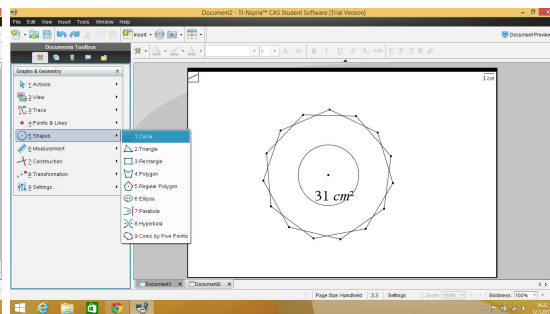
Cinderella podporuje exportovanie do formátov PFD, PNG, JPG. Keďže je aplikácia napísaná v jazyku Java je možné celú konštrukciu exportovať do interaktívnej webovej stránky vo forme HTML kódu. Aplikácia je voľne dostupná pod proprietárnou licenciou [5].

3.5 TI-Nspire CAS Student Software

Poslednou zmienou aplikáciou je TI-Nspire CAS Student Software od firmy Texas Instruments. Tento software sa odlišuje od predchádzajúcich, pretože to nie je iba klasický geometrický náčrtník. Svojou funkcionalitou kopíruje možnosti grafických kalkulačiek TI-Nspire CAS. Funkcionalita programu je rozdelená do nasledujúcich módov : kalkulačka, graf, dáta a štatistiky, geometria a tabuľkový procesor. Natívny vzhľad programu (obr. 3.6) pripomína prostredie grafickej kalkulačky, avšak používateľ má možnosť zmeniť zobrazenie na prostredie počítačovej aplikácie (obr. 3.7).



Obrázek 3.6: TI-Nspire CAS Student Software – Natívny vzhľad aplikácie



Obrázek 3.7: TI-Nspire CAS Student Software – mód geometrie

Pre nás bude najzaujímavejší samozrejme mód geometrie. V tomto móde je možné vytvárať geometrické objekty a pracovať s nimi rovnako ako vo vyššie popísaných aplikáciách. Množina objektov, ktoré je možné zostrojiť je pomerne široká. Okrem základných objektov je to napríklad obdĺžnikový mnohoúhelník (obr. 3.7), hyperbola alebo parabola. Ďalej tu nájdeme typické funkcie ako meranie dĺžky, objemu, uhlu, zostrojenie rovnobežky, priamky, stredového bodu, alebo nájdenie priesečníku. Veľmi zaujímavými sú funkcie transformácie, medzi ktoré patri symetria, reflexia, presunutie, rotácia a dilatácia. Po zvolení niektorej z funkcií sa v ľavom hornom rohu scény zobrazí ikona reprezentujúca vybranú akciu. Pri prechode kurzorom cez túto ikonou sa zobrazí krátka výstižná napovedá, oznamujúca používateľovi ako sa daná funkcia používa.

Mód grafov slúži na vytváranie a analýzu funkcií. Používateľ môže zadaním rovnice vytvoriť jej graf a následne ho analyzovať. Program ponúka množinu funkcií na analýzu grafov, ako napríklad minimum, maximum, derivácia alebo integrál. Geometrické objekty je taktiež možné vytvárať pomocou rovníc. Nie je však potrebné poznať matematické rovnice geometrických objektov. Program poskytuje praktické rozhranie s preddefinovanými rovnicami ilustrované obrázkami. Používateľ si tak môže vybrať z ponuky rovníc a doplniť len niektoré parametre. V tomto móde je dostupné aj zobrazenie v 3D.

Program disponuje značným množstvom rozličných funkcií, čoho dôkazom je aj takmer 600 stranový používateľský manuál [1]. Ukladanie je možné do špeciálneho formátu TI-Nspire Documents (TNS), ktorý je kompatibilný s kalkulačkami firmy Texas Instruments.

Kapitola 4

Zhodnotenie súčasného stavu a analýza zadania

Táto kapitola zhrňuje poznatky o vybraných geometrických náčrtníkoch popísaných v predchádzajúcej kapitole. Ďalej v tejto kapitole čitateľ nájde odôvodnenie prečo a za akým účelom táto práca vznikla. Kapitolu uzatvára stanovenie cieľov práce na základe zadania.

4.1 Zhrnutie súčasného stavu

Aplikácie predstavené v predchádzajúcej kapitole predstavujú pomerne široké spektrum geometrických náčrtníkov pracujúcich v 2D rovine. Samozrejme existuje celá rada ďalších náčrtníkov. Rozsiahlejší zoznam súčasných geometrických konštrukčných nástrojov je dostupný na webe [2]. Nasleduje tabuľka zhrňujúca vlastnosti študovaných náčrtníkov. Tabuľka skúma podporu nevšednej funkcionality, možnosti exportu a licenciu, pod ktorou je daný software dostupný. V poslednom stĺpci sú uvedené špecifické funkcie, ktoré daná aplikácia oproti konkurencii ponúka.

	Skriptovanie	Makrá	Animácie	Export	Licencia	Extra
Cabri II Plus	Áno	Áno	Áno	HTML,PNG, JPG,BMP	GPL	Hyperbolická geometria, podpora TI kalkulačiek
GeoGebra	Áno	Áno	Áno	HTML,PNG,SVG, EMF,GIF,PDF	Proprietárna	CAD, 3D zobrazenie, jazykové lokalizácie
Cinderella	Áno	Áno	Áno	HTML,PNG, JPG,PDF	Proprietárna	Hyperbolická a eliptická geometria, CyndyLab
TI-Nspire Student Software	Áno	Áno	Áno	TNS	GPL	Plná funkcionality kalkulačiek TI-Nspire CAS

Tabuľka 4.1: Zhrnutie vlastností skúmaných aplikácií

Z tabuľky je vidieť, že všetky vybrané náčrtníky ponúkajú podporu skriptovania, tvorbu makri a animácií. Ďalším spoločným rysom je široká ponuka exportu.

Základná funkcionality aplikácií predstavených v predchádzajúcej kapitole je v podstate veľmi podobná. Taktiež spôsob vytvárania geometrických konštrukcií sa nápadne podobná. Výnimkou je aplikácia Cinderella, kde sa objekty ako priamka, úsečka alebo polygón vytvárajú inak, než by mohol používateľ očakávať. Napríklad ak chceme vytvoriť polygón je najprv nutné definovať body, následne zvoliť funkciu na vytvorenie polygónu a dané body prepojiť. Navyše takto vytvorený polygón potom nie je možné presúvať ako jeden objekt. Ďalším z menších nedostatkov je nevýrazné zobrazenie bodov v aplikácii Cabri II Plus, čo však vynahrádza promptná napovedá zobrazovaná pri kurzore.

4.2 Motivácia

Ako bolo spomenuté už vyššie skúmané aplikácie sú, aj napriek svojej širokej škále funkcií, veľmi podobné v spôsobe konštrukcie. Dôraz je kladený na vytváranie a manipulovanie s objektmi pomocou myši. Parametrická úprava konštrukcie pritom zostáva v úzadí. Zaujímavý by mohol byť parametrický náčrtník, ktorý by bol schopný načítať uložený prototyp konštrukcie, vytvorený pomocou klasického geometrického konštrukčného nástroju. Takýto parametrický náčrtník by potom ponúkal možnosť celú konštrukciu rýchlo a presne upraviť pomocou zmeny niektorých parametrov. Predstavme si, že by sme mali k dispozícii jednoduchý nákres čerpadla. Pomocou takejto aplikácie by bolo možné rýchlo upraviť parametre prototypu a vytvoriť tak čerpadlo s požadovanými vlastnosťami.

Ďalším z dôvodov prečo som sa rozhodol pre túto prácu je aj fakt, že ma geometrické náčrtníky zaujali už na strednej škole. Pri výbere témy bakalárskej práce som sa zameral predovšetkým na tie, ktoré vyžadovali tvorbu grafického používateľského rozhrania.

4.3 Vypracovanie zadania

Cieľom mojej práce je navrhnuť a implementovať knižnicu funkcií typických pre geometrické konštrukčné nástroje (ďalej len „knižnica“). Funkčnosť tejto knižnice bude demonštrovať jednoduchý geometrický náčrtník, v ktorom bude možné vytvárať a modifikovať geometrické objekty pomocou myši, tak ako je u aplikácii tohto typu zvykom. Tento náčrtník by mal využívať takmer kompletnú funkčnosť knižnice. Ďalšia aplikácia bude zameraná na využitie možnosti parametrickej úpravy objektov. Tento parametrický náčrtník bude schopný načítať konštrukciu vytvorenú v prvej aplikácii a parametricky meniť jej atribúty. Pri zmene niektorého z parametrov musia samozrejme ostať zachované všetky väzby, definované v pôvodnej konštrukcii. Parametrický náčrtník pravdaže nebude ochudobnený o možnosť vytváranie objektov. Tie sa však do scény nebudú vkladať klasickým rysovaním pomocou pohybu myši, ale definovaním ich parametrov. Výsledná knižnica by mala spĺňať nasledujúce kritériá:

- Možnosť vytvoriť tieto geometrické objekty: bod, úsečka, priamka, rovnobežka, kolmica, polygón, kružnica
- Konštrukcia objektov rysovaním pomocou myši alebo definovaním parametrov
- Modifikácia objektov presunutím (*drag-mode*) a parametricky
- Korektná reakcia pri modifikácii závislých objektov
- Možnosť označenia parametrov konštrukcie
- Vytvorenie animácie
- Uloženie a načítanie nákresu

Pri tvorbe tejto knižnice bude použitý toolkit Qt 5.3.2. Qt je multiplatformný aplikačný toolkit, využívajúci jazyk C++. Používateľské rozhrania vytvorené pomocou tohto toolkitu používajú natívny vzhľad operačného systému, takže aplikácia sa vždy prispôbi používanému prostrediu. Navyše je pri tvorbe grafických používateľských rozhraní možné využiť vstavaný *designer*, ktorý ich tvorbu značne urýchľuje. Qt ponúka celú radu ďalších zaujímavých funkcií, ktoré budú bližšie popísané v kapitolách 5 a 6.

Kapitola 5

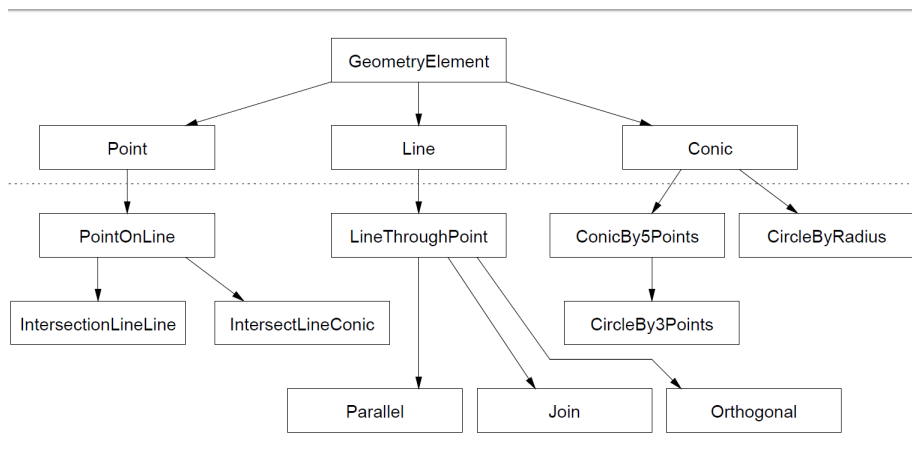
Návrh

Vhodný návrh dátových štruktúr je kritický pre správne fungovanie a rýchlosť aplikácie. Takisto má veľký vplyv na možnosť prípadnej rozšíriteľnosti programu. Prvá časť tejto kapitoly popisuje typický návrh dátových štruktúr používaný vo väčšine aplikácií dynamickej geometrie. Nasledujúca časť sa venuje návrhu dátových štruktúr geometrických objektov použitých v tejto práci. V podkapitole 5.3 sa nachádza popis zobrazenia týchto objektov.

Ďalej sa táto kapitola venuje návrhu grafických používateľských rozhraní geometrického a parametrického náčrtníku. Tieto aplikácie slúžia ako demonštrácia funkcionality knižnice. Cieľom bolo vytvoriť jednoduché a prehľadné používateľské rozhrania. Práca s aplikáciami by mala byť intuitívna a nenáročná, tak pre skúsenejšieho používateľa, ako aj pre používateľa, ktorý pracuje s aplikáciami tohto druhu po prvý krát. Kapitulu uzatvára zhrnutie funkcionality knižnice.

5.1 Klasický prístup k návrhu dátových štruktúr

Štúdium klasického návrhu dátových štruktúr objektovo orientovaného geometrického softwaru malo významný vplyv na návrh použitý pri tvorbe tejto práce. Návrh, ktorý bude popísaný v tejto podkapitole sa využíva vo väčšine aplikácií tohto typu [11]. Hierarchia tried je znázornená na obrázku 5.1.



Obrázek 5.1: Hierarchia tried klasického návrhu [11]

Základnou triedou, od ktorej dedia všetky ostatné, je **GeometryElement**. Táto trieda

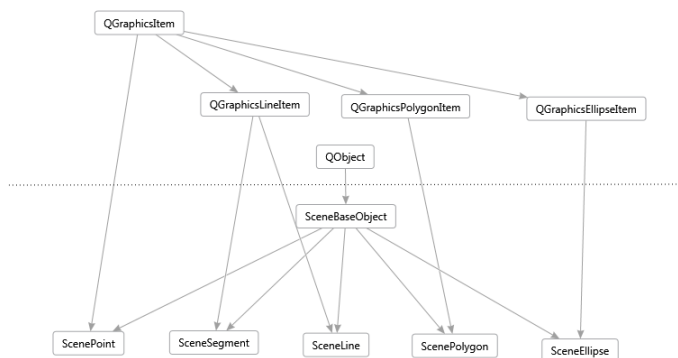
obsahuje metódy a atribúty potrebné pre všetky geometrické objekty, ako napríklad informácia o farbe alebo názve a metódy na zobrazenie alebo aktualizáciu. Dôležitým je taktiež zoznam závislých objektov.

Od nej sú odvodené triedy reprezentujúce základné geometrické objekty, ktoré sú v danej aplikácii dostupné. Na obrázku 5.1 to sú `Point`, `Line` a `Conic`. Tie obsahujú navyše dodatočné dáta, ako napríklad koordináty bodu, dĺžku a smernicu priamky alebo polomer kružnice.

Deriváciou od týchto tried vznikajú špeciálne typy daných prvkov. Tieto triedy implementujú určitý druh závislosti na jednom alebo viacerých elementoch. Napríklad bod, ktorý je priesečníkom dvoch priamok reprezentuje trieda `IntersectionLineLine`, kolmicu zase trieda `Parallel`. Pri zmene pozície niektorého elementu sú najprv zmenené koordináty daného prvku. Následne sa prechádza zoznam priamo závislých objektov a vypočíta sa ich nová pozícia na základe typu závislosti. Na tomto mieste by mohla rekurgia viesť k nekonečnému zanoreniu, preto je nutné kontrolovať, či už daný objekt bol aktualizovaný alebo nie [11].

5.2 Geometrické objekty

V tejto podkapitole bude popísaný návrh dátových štruktúr pre geometrické objekty dostupné v knižnici. Princíp návrhu vychádza z klasického návrhu dátových štruktúr popísaného vyššie. Hlavným rozdielom je vytváranie závislosti medzi objektmi a ich revalidácia pri modifikácii scény (viď 6). Hierarchia tried je zrejmá z obrázku 5.2. Prerušovanou čiarou sú oddelené triedy implementované toolkitom Qt od vlastnej implementácie. Tieto triedy sú v diagrame uvedené preto, lebo majú nezanedbateľný vplyv na vlastnosti a funkcionality implementovaných tried. Pri návrhu bol kladený dôraz na jednoduchosť a možnosť rozširiteľnosti jednotlivých tried.



Obrázek 5.2: Hierarchia tried použitá v knižnici

Už na prvý pohľad je viditeľná paralela medzi týmto a klasickým návrhom dátových štruktúr. Všetky dostupné geometrické objekty sú potomkami triedy `SceneBaseObject`. Táto trieda implementuje spoločné vlastnosti geometrických objektov. Konkrétne sú to informácie o názve a farbe objektu, a signály oznamujúce zmenu pozície (viď 6.3).

5.2.1 Bod

Základom každej geometrickej konštrukcie je bod. V knižnici je reprezentovaný triedou **ScenePoint**. Bod sa môže v scéne vyskytovať ako samostatný (voľný) objekt, ako stred úsečky, môže patriť jednému alebo viacerým objektom súčasne. Fundamentálnou vlastnosťou všetkých geometrických objektov je samozrejme ich pohyblivosť. Polohu bodu je možné meniť viacerými spôsobmi. Samozrejmosťou je uchopenia a presunutie bodu pomocou myši. Druhou možnosťou je zadanie presných koordinátou. V prípade, že je bod súčasťou iného objektu, je viazaný, tak pri zmene polohy rodičovského objektu pochopiteľne tiež zmení svoju pozíciu. Výnimku predstavuje stred úsečky. Tento bod nie je samostatne pohyblivý, nie je ho možné uchopiť a premiestniť ani mu priradiť koordináty. Pohybuje iba pri presune úsečky, ktorej náleží.

Bod nemusí náležať iba jednému objektu. Počet objektov, ktoré je možné naviazať na jeden bod nie je v rámci návrhu nijak obmedzený. Podmienkou, ktorú je nutné splniť je, aby na pohyb takéhoto bodu korektne reagovali všetky naviazané objekty. Vďaka tejto vlastnosti je možné vytvoriť zaujímavé, komplexnejšie konštrukcie.

Informácie, ktoré si bod uchováva sú jeho súradnice v scéne, parameter udávajúci pozíciu v rámci úsečky a údaj o viazanosti. Tento údaj si uchováva len z dôvodu obmedzenia redundandnosti pri ukladaní scény. V prípade, že bod náleží úsečke, uchováva si navyše parameter, udávajúci jeho pozíciu v rámci danej úsečky. Pri zmene dĺžky úsečky sa na základe tohto parametru vypočíta nová pozícia bodu. Tento parameter je konštantný a môže nadobúdať hodnoty od 0 do 1. Napríklad pre stredový bod úsečky bude mať hodnotu 0.5.

5.2.2 Úsečka a priamka

Úsečku reprezentuje trieda **SceneSegment**. Jediné informácie, ktoré si táto trieda uchováva je zoznam bodov, ktoré jej náležia. Tento zoznam je však potrebný len pri ukladaní scény. Pre samotnú úsečku sú dôležité len jej začiatočný a koncový bod. Modifikovať ju je možné tromi spôsobmi: presunom krajných bodov, uchopením a premiestnením alebo nastavením dĺžky. Pri zmene dĺžky sa posunie koncový bod tak, aby mala výsledná úsečka požadovanú dĺžku a zároveň ostal zachovaný jej smerový vektor. Hoci je táto trieda vo svojej podstate jednoduchá, zohráva dôležitú úlohu ako stavebný prvok polygónu.

K realizácii priamky slúži trieda **SceneLine**. Trieda obsahuje tri druhy chovania. Prvým je priamka definovaná dvoma bodmi a smerovým vektorom medzi nimi. Ostatnými dvoma sú rovnobežka, respektíve kolmica, k úsečke alebo priamke prechádzajúca bodom. Na rozdiel od prvého typu priamky sú tieto dve závislé na objekte s ktorým sú rovnobežné, respektíve kolmé. Pri zmene smerového vektoru závislého objektu sa odpovedajúcim spôsobom zmení aj smerový vektor priamky. U tohto typu priamky nie je možné priamo nastaviť smerový vektor. Povolenie takejto modifikácie by viedlo k porušeniu kolmosti alebo rovnobežnosti. Tento problém by sa dal odstrániť, ak by sa spoločne so zmenou smerového vektoru priamky menil aj objekt na, ktorom je závislá. Otázkou je, či by bolo takéto chovanie pre užívateľa očakávané.

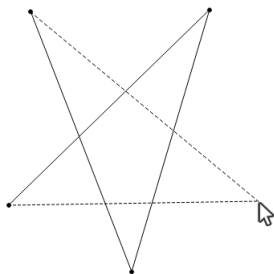
U priamky prechádzajúcej dvoma bodmi je možné smerový vektor ľubovoľne meniť presunom jedného z bodov, ktorými prechádza. Všetky priamky je taktiež možné presúvať uchopením pomocou myši. V tomto prípade ostane samozrejme smerový vektor nezmenený. Zaujímavou vlastnosťou priamky je jej predĺženie pri zmene viditeľného výrezu scény, čím je imitovaná „nekonečnosť“ priamky.

5.2.3 Polygón

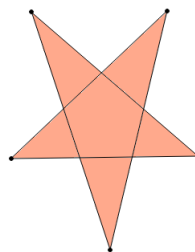
Ďalším dostupným geometrickým objektom knižnice je polygón, reprezentovaný triedou **ScenePolygon**. Polygón môže byť definovaný ľubovoľným počtom bodov, najmenej však tromi, ktoré sú prepojené úsečkami. Body a úsečky, z ktorých sa polygón skladá, sú inštaniami tried popísaných vyššie. S týmito objektmi je možné v rámci polygónu manipulovať rovnakým spôsobom, ako keby boli neviazané. Vďaka tomu je možné polygón modifikovať presunom jednotlivých bodov, strán alebo celého objektu. Táto vlastnosť umožňuje veľmi flexibilnú a intuitívnu manipuláciu s objektom. Užívateľ tak môže veľmi rýchlo dosiahnuť požadovaný tvar polygónu. Túto možnosť neponúkajú niektoré súčasné geometrické nástroje. Vo svojej podstate však ostáva táto trieda jednoduchá a obsahuje iba informácie o bodoch a úsečkách, z ktorých sa skladá.

Zaujímavou funkciou knižnice, ktorá sa týka polygónov je parametrické vkladanie trojuholníkov. Tá umožňuje vytvoriť trojuholník až tromi spôsobmi: definovaním koordinátou vrcholov, zadáním dĺžok strán alebo zadáním dĺžky základne a príslúchajúcej výšky. V poslednom prípade bude vytvorený trojuholník vždy rovnostranný. Veľmi užitočné a nápomocné však môže byť vytvorenie trojuholníku definovaním dĺžok jeho strán. Vo väčšine prípadom kedy chce užívateľ vytvoriť nejaký konkrétny trojuholník, má k dispozícii práve tieto údaje. Vďaka tejto funkcii nie je nutné dopočítavať polohy vrcholov, alebo hľadať daný trojuholník presúvaním vrcholov.

Samozrejmosťou je vytváranie polygónu spôsobom bežným v geometrických nástrojoch, čiže definovaním vrcholov kliknutím na náčrtu. Počas konštrukcie sa zobrazujú prerušované pomocné čiary, ktoré definujú tvar polygónu, ktorý by vznikol ak by bol ďalší vrchol umiestnený na aktuálnu pozíciu kurzoru v scéne (obr. 5.3). Vďaka tomu má užívateľ vizuálnu spätnú väzbu na pohyb kurzoru a nemusí odhadovať alebo si predstavovať tvar výsledného polygónu. Konštrukciu je navyše možné pohodlne ukončiť dvojklikom a polygón sa automaticky dokreslí na naznačený tvar (obr. 5.4).



Obrázek 5.3: Polygón pred dokončením



Obrázek 5.4: Polygón po dokončení

5.2.4 Kružnica

Posledným geometrickým objektom dostupným v súčasnej verzii knižnice je kružnica. Tá je zastúpená triedou **SceneEllipse**. Už samotný názov napovedá, že táto trieda by mala byť schopná realizovať nie len kružnicu ale hocijakú kužeľosečku. To bolo zámerom pri definovaní tejto triedy, ku implementácii ostatných kužeľosečiek však bohužiaľ nedošlo. V súčasnosti je podporovaná konštrukcia kružnice, definovanej stredom a bodom ležiacim na kružnici. Kružnicu je možné modifikovať viacerými spôsobmi. Očakávanými sú uchopenie a presun celej kružnice a zadanie polomeru. Zaujímavé je pozorovať správanie kružnice pri

presune jednotlivých bodov, ktorými je definovaná. Napríklad pri pohybe stredového bodu, ostáva bod ležiaci na jej obode statický. Kružnica tak musí meniť svoj polomer ale zároveň musí prechádzať daným bodom.

5.3 Zobrazenie geometrických objektov

Na dynamické vytváranie, zobrazovanie a manipuláciu s geometrickými objektmi je v knižnici využitý tzv. „*Graphics View Framework*“ toolkitu Qt. Jeho architektúra sa skladá zo scény (*scene*), pohľadu (*view*) a grafických objektov (*item*).

Scénu predstavuje trieda `QGraphicsScene` a zabezpečuje nasledovné:

- Rýchle rozhranie pre prácu s veľkým množstvom objektov
- Šírenie udalosti myši a klávesnice každému objektu
- Riadenie stavu objektu

Scéna funguje ako kontajner 2D objektov, reprezentovaných triedou `QGraphicsItem`. Objekty môžu byť do scény vkladané a následné je možné s nimi pracovať prostredníctvom vyhľadávacích funkcií. Mechanizmus šírenia udalosti v scéne plánuje propagáciu udalosti objektom a taktiež medzi objektmi samotnými. Napríklad ak scéna prijme udalosť kliknutia myši, propaguje túto udalosť objektom, ktoré sa nachádzajú na danej pozícii v scéne. Objekty sa môžu nachádzať v rôznych stavoch, napríklad objekt môže byť označený. V tomto stave je možné s objektom manipulovať.

Trieda `QGraphicsView` predstavuje pohľad, ktorý vizualizuje obsah scény. Na jednu scénu je možné naviazať viacero pohľadov, čím je možné docieľiť rôznych zobrazení rovnakých dát. Pohľad je schopný transformovať súradný systém scény pomocou transformačných matic. To umožňuje pokročilé navigačné funkcie, ako je zväčšovanie a rotácie.

Bázovou triedou pre grafické objekty v scéne je `QGraphicsItem`. Táto trieda okrem iného poskytuje nasledujúcu funkcionálnosť:

- Spracovanie udalosti myši (klik, dvojklik, pohyb, stlačenie a uvoľnenie tlačidiel)
- Spracovanie vstupu z klávesnice
- Detekcia kolízií
- Zoskupovanie grafických objektov

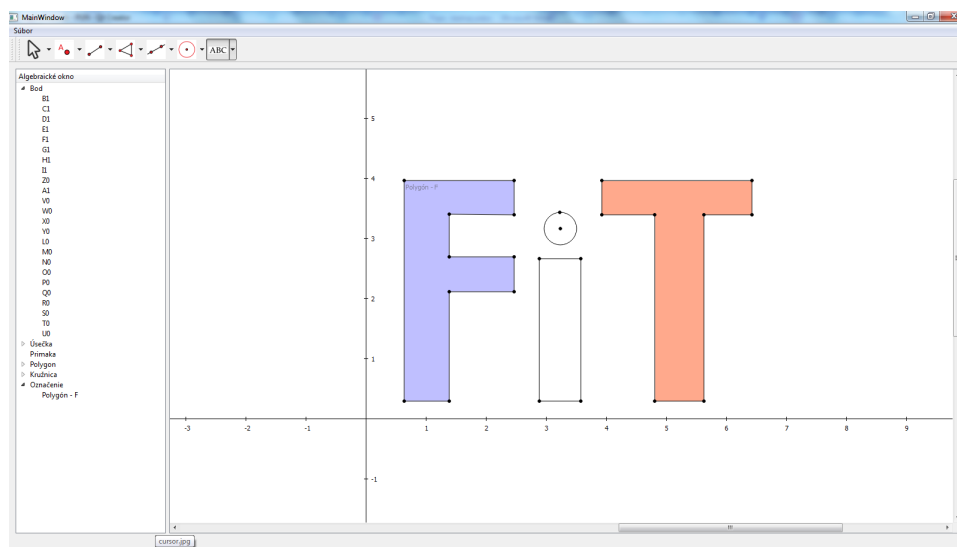
Grafické objekty existujú v lokálnom súradnom systéme a podobne ako pohľad, sú schopné ho transformovať pomocou transformačných matic. To znamená, že je možné prevádzať rotácie a zväčšovania individuálnych objektov. Medzi objektmi je možné vytvárať vzťah rodič - dieťa. Jeden objekt (rodič) môže obsahovať ďalšie objekty (deti), na ktoré budú aplikované všetky transformácie rodičovského objektu [3].

Scéna, pohľad a všetky geometrické objekty v knižnici sú odvodené práve od týchto tried. Ich funkcionálnosť je rozšírená a prispôbena účelom tejto práce. Takmer kompletná funkcionálnosť ponúkaná knižnicou je navyše dostupná prostredníctvom triedy `GraphicsScene`, ktorá predstavuje scénu. Existujú dva zásadné dôvody prečo je tomu tak. Prvý dôvod vychádza zo samotnej podstaty tejto triedy. Ako bolo spomenuté vyššie úlohou scény je vytváranie a uchovávanie geometrických objektov. Navyše sú dostupné práve funkcie na vkladanie objektov do scény a manipuláciu s nimi. Logika však ostáva pre programátora,

ktorý by sa rozhodol knižnicu použiť, skrytá. Ďalším dôvodom je jednoduchosť použitia. Stačí vytvoriť inštanciu tejto triedy a kompletná funkcionálnosť scény je dostupná. Korektné zobrazenie scény zabezpečuje pohľad `SceneView` dostupný z knižnice.

5.4 Používateľské rozhranie geometrického náčrtníku

Pri návrhu používateľského rozhrania geometrického náčrtníku boli rešpektované zvyklosti zaužívané u aplikácií tohto druhu. Celé používateľské rozhranie je rozdelené na štyri časti: klasické menu, panel nástrojov s funkciami programu, zoznam vytvorených objektov a scénu (obr. 5.5).



Obrázek 5.5: Používateľské rozhranie geometrického náčrtníku

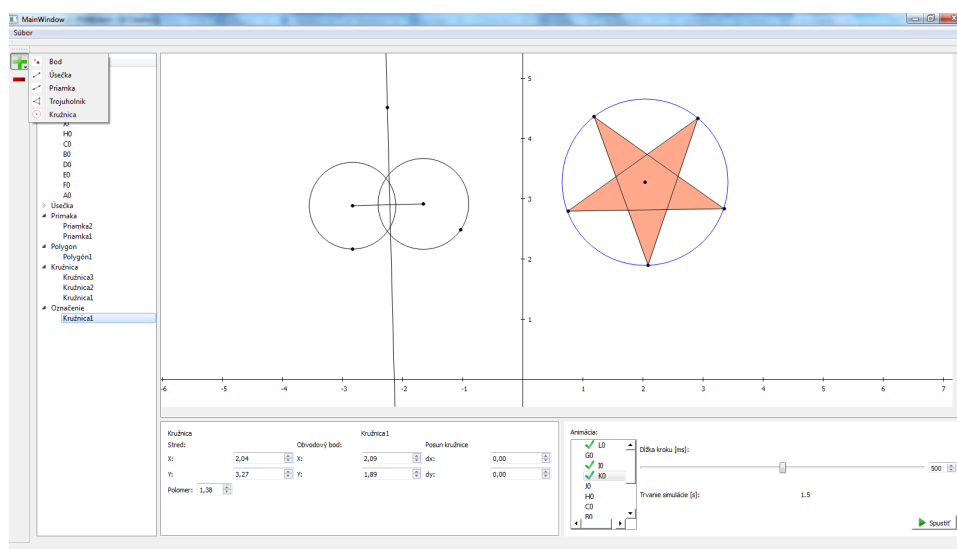
V najvrchnejšej časti používateľského rozhrania sa nachádza menu, ktoré umožňuje uloženie a načítanie konštrukcie a ukončenie programu. Pod menu sa nachádza panel nástrojov. Ako je u programov tohto typu zvykom, sú funkcie programu rozdelené do skupín. Ikony panelu nástrojov predstavujú jednotlivé skupiny. Obrázok ikony by mal byť dostatočne výstižný na to, aby sa aj používateľ, ktorý s programom pracuje po prvýkrát, v ňom dokázal bez problémov zorientovať. Po kliknutí na ikonu sa zobrazí zoznam funkcií prislúchajúcich do danej skupiny. Jednotlivé funkcie sú reprezentované obrázkom a slovným popisom. Po zvolení niektorej z funkcií sa obrázok ikony zmení podľa vybranej funkcie a daná ikona sa zvýrazní.

Najväčšiu časť používateľského rozhrania tvorí scéna. Je to interaktívna plocha, do ktorej je možné pomocou myši vkladať zvolené geometrické objekty, upravovať ich a manipulovať s nimi. Scéna obsahuje už preddefinované osy. Naľavo od náčrtníka sa nachádza okno, ktoré obsahuje zotriedený zoznam všetkých vytvorených geometrických objektov podľa ich typu a poskytuje k nim rýchly prístup. Súčasťou rozhrania je taktiež dialógové okno, ktoré sa zobrazí po dvojkliku na bod. Prostredníctvom tohto okna je možné nastaviť presné koordináty bodu.

5.5 Používateľské rozhranie parametrického náčrtníku

Rozhranie je už na prvý pohľad trochu komplexnejšie ako to predchádzajúce (obr. 5.5). Skladá sa z nasledujúcich častí: menu, bočný nástrojový panel, zoznam objektov, scéna, formulár s informáciami o vybranom objekte a formulár animácie.

Menu je totožné s tým, ktoré bolo popísané v predchádzajúcej podkapitole. Scéna je inštanciou rovnakou triedy ako v predchádzajúcej aplikácii, avšak je tu malá zmena v jej nastavení. Táto scéna nemá povolené vkladanie ani manipuláciu objektov pomocou myši. Namiesto toho sú tieto operácie realizované parametricky. Na vkladanie objektov do scény slúži tlačidlo s ikonou zeleného plus na bočnom paneli nástrojov. Po kliknutí na ikonu sa zobrazí zoznam objektov, ktoré je možné vytvoriť. Jednotlivé objekty sú ilustrované obrázkom a slovným popisom. Po zvolení niektorého z objektov sa zobrazí dialógové okno, v ktorom používateľ definuje parametre daného objektu.



Obrázek 5.6: Používateľské rozhranie parametrického náčrtníku

V tejto aplikácii je vo väčšej miere využitá funkcionálna zoznamu objektov. Pri prechode kurzorom nad menom objektu v zozname sa daný objekt zvýrazní v scéne. Ak užívateľ klikne na meno objektu v zozname, tak sa v ľavej dolnej časti obrazovky zobrazia detailné informácie o danom objekte. Rozloženie formulára s informáciami sa dynamicky mení podľa typu objektu. Zobrazené parametre objektu je možné modifikovať a ihneď pozorovať zmeny v konštrukcii. U kružnice je možné meniť polohu jej stredu, bodu na jej obvodě, polomer a posunutie v smere x-ovej a y-ovej osy. Úsečke je možné nastaviť pozícia jej koncových bodov a dĺžku. U priamky zase koordináty bodov, ktorými prechádza. Polygón je možné modifikovať pomocou jeho vrcholov. V zozname objektov je však možné nájsť jednotlivé elementy, z ktorých sa daný objekt skladá.

Vďaka možnosti parametrickej zmeny vlastností objektov je možné vytvoriť jednoduché animácie. V pravej dolnej časti obrazovky sa nachádza formulár animácie. V súčasnosti je možné vytvárať iba animácie bodov, avšak väzby medzi objektmi ostávajú zachované aj počas behu animácie. Používateľ má k dispozícii vyfiltrovaný zoznam všetkých bodov v scéne, ktoré môže použiť pri tvorbe svojej animácie. Na pravo od zoznamu bodov sa nachádza jazdec (*slider*), ktorým je možné definovať časový rozdiel medzi jednotlivými krokmi. Tým je možné urýchliť alebo spomaliť priebeh animácie. Vpravo od jazdca sa

nachádza číselník zobrazujúci aktuálnu hodnotu časového rozdielu. Číselník je samozrejme tiež editovateľný, takže používateľ môže zadať priamo číselnú hodnotu.

Na vytvorenie animácie je potrebné aby používateľ pridelil zoznam x-ových a y-ových súradníc každému bodu, ktorý má byť animovaný. Tie sa načítajú z excelovského súboru. Po dvojkliku na bod v zozname môže užívateľ vybrať excelovský súbor uložený na disku. Ak sú dáta úspešne načítané, zobrazí sa pri danom bode zelená ikona, signalizujúca, že tento bod bude súčasťou animácie. Počet bodov v animácii nie je nijak obmedzený. Počet koordinátou u vybraných bodov sa nemusí zhodovať, dĺžka animácie je determinovaná maximálnym počtom súradníc.

5.6 Zhrnutie funkcionality

V súčasnosti poskytuje knižnica možnosť vytvorenia a modifikácia nasledujúcich objektov: bod, úsečka, priamka, polygón a kružnica. K základným funkciám patria zostrojenie rovnobežky, priamky, vyčíslenie dĺžky, uloženie a načítanie nákresu, ukladanie geometrických objektov do stromovej štruktúry a označenie objektu.

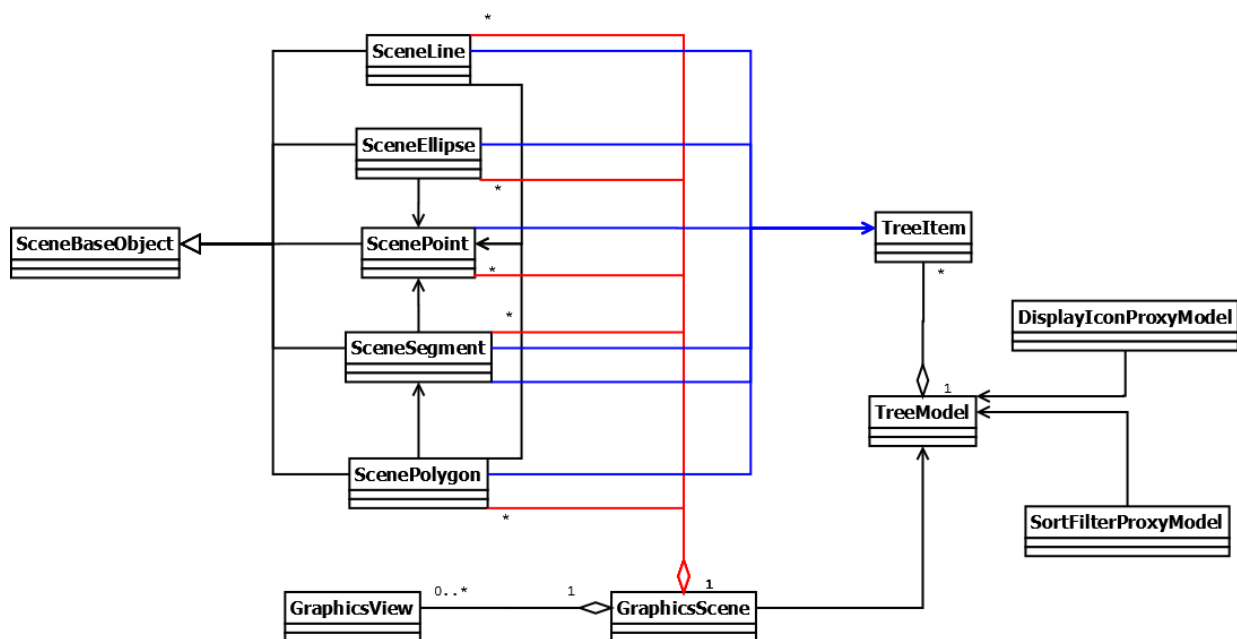
Vytvorenú konštrukciu je možné uložiť a znova načítať, pričom je garantované zachovanie vytvorených väzieb medzi objektmi. Geometrické objekty sa navyše ukladajú aj do stromovej štruktúry implementovanej triedou `TreeModel`, ktorá je odvodená od triedy tool-kitu Qt `QAbstractItemModel`. V tejto triede sú uložené referencie na objekty vytvorené užívateľom a sú prehľadné zotriedené podľa typu objektu. V grafickom používateľskom rozhraní je potom možné obsah tejto triedy vizualizovať (obr. 5.5). Takáto zoznam významne uľahčuje vyhľadávanie a orientáciu v nákrese.

Užitočnou môže byť aj funkcia označenia objektu. Používateľ môže pomocou tejto funkcie zvoliť ľubovoľný objekt scény a priradiť mu alternatívne označenie. Primárnym využitím tejto funkcie je sprehľadnenie a urýchlenie vyhľadávania v konštrukcii. Predstavme si napríklad, že by sme v geometrickom náčrtníku vytvorili nákres čerpadla. V potenciálne zložitom nákrese by sme si mohli vhodným pomenovaním označiť napríklad kružnicu predstavujúcu prietok. Ak by sme potom takto uložený nákres otvorili v parametrickom náčrtníku, veľmi rýchlo a jednoducho by sme, pod záložkou „Označenie“ v zozname objektov, označenú kružnicu našli. Na obrázku je možné 5.5 vidieť použitie tejto funkcie. Označený je polygón v tvare písmena „F“. Objekt, ktorý ma pridelené označenie zmení svoju implicitnú farbu na modrú.

Kapitola 6

Implementácia

Táto kapitola popisuje spôsob implementácie fundamentálnych problémov dynamickej geometrie, a síce vytváranie a pohyb objektov. Základnou vlastnosťou každého geometrického objektu je možnosť ho presúvať. Pri pohybe však musia ostať zachované väzby definované medzi objektmi. Prvá časť tejto kapitoly sa venuje mechanizmu signálov a slotov toolkitu Qt, ktorý je pri riešení týchto problémov kritický. Nasledujúca časť sa venuje problematike vytvárania geometrických objektov. Spôsob implementácie väzieb a pohybu objektov je popísaný v podkapitole 6.3. Kapitole je ukončená popisom funkcií načítania a uloženia dát.



Obrázek 6.1: UML diagram – model tried

Na obrázku 6.1 je znázornený zjednodušený UML diagram celej knižnice. Kvôli prehľadnosti diagram neobsahuje metódy ani atribúty tried. V centrálnej časti diagramu sú znázornené jednotlivé geometrické objekty. Z diagramu môžeme vyčítať odvodenie týchto tried od triedy `SceneBaseObject`, odôvodnené v kapitole 5. Medzi jednotlivými objektmi si môžeme všimnúť vzťahy asociácie. Tento vzťah naznačuje, že zložitejšie geometrické objekty sa skladajú z tých jednoduchších a uchovávajú si na ne referencie. Napríklad úsečka je tvorená

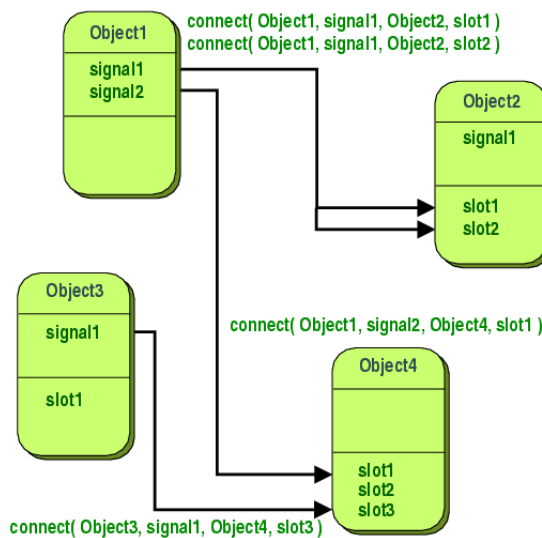
bodmi, polygón zasa bodmi a úsečkami.

Červenou farbou je znázornený vzťah agregácie medzi triedami geometrických objektov a triedou `GraphicsScene`. Tento vzťah je očakávaný a znamená, že scéna obsahuje kolekciu geometrických objektov. Vzťah medzi triedami `GraphicsScene` a `GraphicsView` naznačuje, že na jednu scénu nemusí existovať žiadny alebo až n pohľadov, ktoré vizualizujú jej obsah. Scéna ďalej obsahuje odkaz na inštanciu triedy `TreeModel`, ktorá vytvára stromovú štruktúru z inštancii triedy `TreeItem`. Tá si musí uchovávať odkaz na geometrický objekt, ktorý zastupuje, aby mohla ponúknuť prístup k jeho metódam. V diagrame znázornené modrou farbou.

Triedy `DisplayIconProxyModel` a `SortFilterProxyModel`, predstavujú iba zástupcov pôvodnej stromovej štruktúry geometrických objektov. Tieto triedy umožňujú v pôvodnej štruktúre buď filtrovať podľa rôznych parametrov, alebo jednotlivým položkám pridať špeciálne vlastnosti, ako napríklad ikonu.

6.1 Signály a sloty

Jednou z kľúčových vlastností toolkitu Qt využitých pri tvorbe knižnice je mechanizmus tzv. „*signals & slots*“. Tento mechanizmus je používaný na komunikáciu medzi objektmi a je alternatívou *callback* funkcií. Nevýhodou *callback* funkcií je, že sa pri ich volaní neprevádza typová kontrola a volaná metóda musí poznať ukazovateľ na metódu, ktorá ju zavolať. Typickým prípadom použitia signálov a slotov je reakcia objektu na vstup od používateľa. Napríklad ak používateľ stlačí tlačidlo okna Zavrieť, pravdepodobne chceme zavolať metódu okna `close()`. Pre ich použitie sa definuje spojenie signálu so slotom pomocou metódy `connect()`, pričom musí odpovedať počet a typ argumentov. Tento mechanizmus potom zaisťuje, že po emitovaní signálu bude zavolaný odpovedajúci slot s parametrami signálu. Signály a sloty sú voľne spojené, to znamená, že trieda, ktorá signál emitovala nevie ktorý slot jej signál príjme a slot nevie, či sú k nemu pripojené nejaké signály. Sloty nie sú nijako limitované počtom pripojených signálov a signály môžu byť pripojené k ľubovoľnému počtu slotov. Tento mechanizmus dovoľuje vytváranie nezávislých komponentov a navyše zaisťuje typovú kontrolu [3].



Obrázek 6.2: Qt – Signály a sloty

6.2 Vytváranie geometrických objektov

Ako už bolo zmienené vyššie, knižnica podporuje dva druhy vytvárania geometrických objektov. Prvým spôsobom je vkladanie objektov do scény interaktívnym rysovaním pomocou myši. Je to primárny spôsob vytvárania objektov v aplikáciách venujúcich sa dynamickej geometrii. Základným predpokladom interaktívneho rysovania je pochopiteľne schopnosť spracovať udalosti pohybu a kliknutia myši. Na to sú využité metódy `mouseMoveEvent` a `mouseReleaseEvent` toolkitu Qt. Metóda `mouseMoveEvent` zachytáva udalosti pohybu myši nad scénou, `mouseReleaseEvent` zase umožňuje zistiť koordináty kurzoru v scéne v momente vzniku udalosti uvoľnenia tlačidla myši. Spôsob akým scéna s touto informáciou naloží, záleží na tom v akom stave sa momentálne nachádza. Stav scény je determinovaný aktuálne vybranou funkciou a jej štádiom vykonávania. Funkciu, ktorú má scéna pri spracovaní udalosti vykonať je možné zadať pomocou metódy `setSceneFunction`. Zoznam funkcií, ktoré scéna podporuje je určený výpočtom `sceneFunctions`. Napríklad ak je zvolená funkcia vytvorenia kružnice, reakciou na prvé kliknutie bude vytvorenie kružnice a stredového bodu. V tomto momente začne scéna reagovať aj na udalosť pohybu myši a dynamicky meniť polomer vzhľadom na stred kružnice a kurzor myši. Po ďalšom kliknutí sa zakreslí bod ležiaci na kružnici, konštrukcia toho objektu sa ukončí a scéna prejde do počiatočného stavu vytvárania kružnice.

Parametrické vytváranie objektov, je vo svojej podstate veľmi podobné, líši sa však v spôsobe získavania parametrov objektu. Zatiaľ čo pri prvom spôsobe boli tieto informácie získané dynamicky ako koordináty kurzoru v scéne počas konštrukcie objektu, v tomto prípade sú očakávaným vstupom odpovedajúcich metód. Knižnica podporuje dynamické vytvorenie bodu, úsečky, priamky, trojuholníku a kružnice. Rozdielom je tiež, že pri takomto type konštrukcie musia byť koordináty niektorých bodov dopočítané. Príkladom môže byť vytvorenie trojuholníku pomocou dĺžok strán. V tomto prípade bola na určenie pozície bodov využitá kosínusová veta [8].

6.3 Vytváranie väzieb medzi objektmi a ich pohyb

Väzba medzi objektmi predstavuje určitý druh závislosti. Znamená to, že objekty spolu komunikujú a reagujú na zmenu pozície objektov, na ktorých sú závislé. Táto vlastnosť je dosiahnutá pomocou systému signálov a slotov.

Bazová trieda `SceneBaseObject`, od ktorej sú odvodené všetky geometrické objekty obsahuje signál `posChanged`, ktorý je pre fungovanie celej knižnice fundamentálny. Je emitovaný vždy keď objekt zmení svoju pozíciu v scéne. Vytvorenie závislosti medzi objektmi spočíva v prepojení tohto signálu s odpovedajúcim slotom, zaistujúcim prekreslenie daného objektu. Výhodou tohto prístupu je, že objekty si nemusia uchovávať žiadny zoznam závislých objektov ani nie je prekresľovaná celá scéna pri pohybe jedného objektu. Objekty si uchovávajú iba zoznam elementov, z ktorých sú tvorené. Tento zoznam sa využíva v momente, keď je objekt vyzvaný aby sa prekreslil. Vtedy si od prvkov zo zoznamu vyžiada ich aktuálnu pozíciu a prekreslí sa. Napríklad úsečka si uchováva odkazy na svoje krajné body, polygón zasa zoznam vrcholov a úsečiek, ktoré ho definujú. Bod neobsahuje žiadny zoznam objektov, keďže je to elementárny objekt. Ako bolo spomenuté vyššie, počet objektov, ktoré je možné na jeden bod napojiť nie je nijak implementačne obmedzený. Stačí pritom prepojiť signál `posChanged` daného bodu s požadovaným objektom. Pri zmene pozície bodu bude upovedomený každý objekt, ktorý je napojený na tento signál. Vďaka tomu je možné vytvárať zaujímavé geometrické konštrukcie a sledovať zmeny rôzne prepojených objektov.

Ďalším typom závislosti sú rovnobežky a kolmice. Tieto objekty sú priamo závislé na smerovom vektore rodičovskej priamky alebo úsečky a na pozícii bodu, ktorým prechádzajú. Tento druh priamky si uchováva referencie na body určujúce smerový vektor rodičovského objektu. Pri požiadaní o prekreslenie sa vypočíta nový smerový vektor z aktuálnych súradníc týchto bodov a to s ohľadom na to či sa jedná o kolmicu alebo rovnobežku. Na jednu priamku alebo úsečku môže byť opäť naviazaných ľubovoľne veľa rovnobežiek alebo kolmíc, bez toho aby si o tom objekt uchovával nejakú informáciu.

Príkladom efektivity tohto prístupu môže byť mnohouholník, ktorý môže byť tvorení akýmkoľvek počtom bodov väčším než tri. Je možné presúvať každý bod, stranu alebo polygón ako celok. Neprevádzajú sa pri tom žiadne komplikované kontroly, ktorý bod alebo strana sa presunula a ako má byť polygón prekreslený, ako tomu bolo v počiatočnej fáze návrhu. Navyše je možné na každý bod polygónu naviazať ľubovoľný počet a typ objektov, ku každej strane je možné zostrojiť ľubovoľne mnoho rovnobežiek alebo kolmíc a je pri tom zaručené korektné chovanie každého objektu.

6.4 Ukladanie a načítanie dát

Bez možnosti uloženia a načítania dát by vytváranie konštrukcií asi nemalo veľký zmysel. Implementácia oboch týchto funkcií je preto neoddeliteľnou súčasťou knižnice. Vytvorenie konštrukciu je možné uložiť pomocou metódy scény `saveDrawing`, načítať zase pomocou `loadDrawing`. Parametrom oboch funkcií je absolútna cesta k súboru vo formáte XML. Na prehľadávanie súborového systému a výber, prípadné vytvorenie, súboru sú použité natívne dialógové okná operačného systému, na ktoré je používateľ zvyknutý z bežných aplikácií. Na vytváranie a spracovanie XML súborov bola použitá knižnica `QtXML`.

Funkcia ukladania prechádza zoznam všetkých objektov, ktoré vytvoril používateľ a volá ich metódu `fillSaveData`. Táto metóda vytvorí uzol XML dokumentu a vyplní ho relevantnými dátami o danom objekte. Tie samozrejme záležia na triede objektu ale vo všeobecnosti sú to typ, meno objektu a zoznam elementov, z ktorých sa skladá.

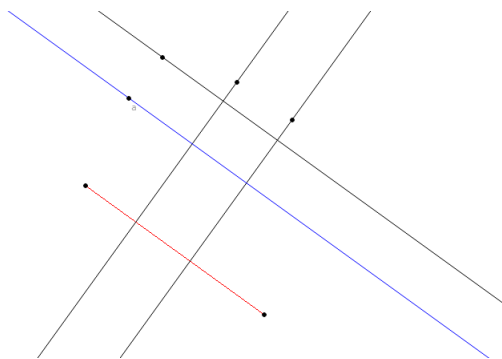
Funkcia načítania dát prechádza uložený XML dokument po uzloch a vkladá do scény príslušné geometrické objekty. Na to sú využité metódy parametrického vytvárania objektov. Aby boli zachované väzby definované v pôvodnom nákrese kontroluje sa existencia každého objektu. Ak objekt neexistuje, tak sa jednoducho vytvorí s danými parametrami, ak sa však už v scéne nachádza, tak sa prepoja potrebné signály a sloty, čím sa vytvoria pôvodné väzby.

Kapitola 7

Testovanie

Funkčnosť výslednej knižnice bola testovaná experimentálne pomocou vytvorených grafických používateľských rozhraní. Sadá testov sa skladala z rôznych geometrických konštrukcií, od najjednoduchších elementárnych objektov až po zložitejšie konštrukcie pokrývajúce kompletnú funkcionálnu knižnicu.

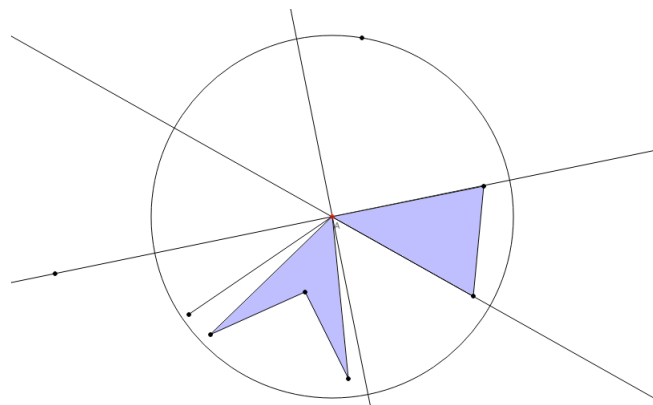
Pri testovaní základných geometrických objektov bolo najviac chýb odhalených u kružnice. Pri presune bodu ležiaceho na kružnici sa polomer zväčšoval dvojnásobne, pri posune stredového bodu zase tento bod nereagoval vôbec. Parametrické nastavovanie polomeru tiež spôsobovalo nekorektné chovanie tohto bodu. Ďalším testom bola odhalená zásadná chyba v implementácii kolmíc. Tento test spočíval v zostrojení kolmice k úsečke a následným zostrožením druhej kolmice k prvej. Očakávaným chovaním je, že druhá kolmica bude s úsečkou rovnobežná. Tento test je ilustrovaný na obrázku 7.1, červenou farbou je vyznačená pôvodná úsečka, modrou zase skúmaná priamka. Po vytvorení konštrukcie tomu tak aj bolo, ale po zmene smerového vektora úsečky sa druhá priamka stále kolmou na úsečku, čo bolo nekorektné chovanie. Vďaka týmto testom boli odhalené chyby v návrhu a implementácii základných geometrických objektov.



Obrázek 7.1: Test kolmosti

Po odstránení týchto chýb nasledovali komplexnejšie testy. Táto sadá sa zameriavala hlavne na väzby vytvorené medzi viacerými objektmi, ale obsahovala aj záťažové testy. Napríklad bolo na jeden bod naviazaný relatívne veľký počet objektov rôzneho druhu a bolo skúmané ich správanie pri pohybe tohto bodu. Tieto testy prebehli bez problémov. Na obrázku 7.2 je možné vidieť test, v ktorom boli na bod v strede naviazané všetky typy objektov, ktoré knižnica v súčasnosti ponúka.

V priebehu testovania bolo odhalená celá rada chýb a nedostatkov, či už v návrhu



Obrázek 7.2: Test väzieb medzi objektmi

alebo implementácii, ktoré boli následné opravené. Bolo vynaložené maximálne úsilie na otestovanie rozličného druhu konštrukcií a všetky funkcie, ktoré knižnica obsahuje. Nedá sa však tvrdiť, že by výsledky testovania bezvýhradné zaručovali bezchybnosť všetkých možných konštrukcií.

Kapitola 8

Záver

Cieľom tejto práce bolo vytvoriť knižnicu funkcií dynamickej geometrie a jej funkcionality demonštrovať v grafickom používateľskom rozhraní. Tento cieľ bol splnený.

Druhá kapitola tejto práce uvádza čitateľa do problematiky dynamickej geometrie a zhrňuje najdôležitejší pojmy spojené s touto prácou. Naštudoval som niektoré dostupné parametrické kresliace nástroje, ich funkcionality a vlastnosti. Ich prehľad je uvedený v tretej kapitole. Získané poznatky sú zhrnuté kapitole 4, čím bol splnený prvý bod zadania. Kapitola 5 obsahujú návrh dátových štruktúr a grafických používateľských rozhraní geometrických náčrtníkov, čo splňuje druhý a tretí bod zadania. Kapitola 6 popisuje spôsob riešenia najdôležitejších problémov, ktoré sa vyskytli pri implementácii. Tým bol splnený štvrtý bod zadania. Posledný bod zadania splňa táto kapitola.

Výsledkom práce je knižnica základných funkcií dynamickej geometrie a dve grafické používateľské rozhrania, využívajúce jej funkčnosť rôznymi spôsobmi. Vďaka tejto práci som sa dozvedel ako fungujú parametrické geometrické náčrtníky a zdokonalil som sa v tvorbe aplikácii pomocou toolkitu Qt.

V práci je ďalej možné pokračovať rozšírením funkcionality knižnice alebo vylepšením používateľských rozhraní. Možnými rozšíreniami knižnice sú napríklad funkcia na nájdenie priesečníku, implementácia kriviek alebo podpora iných druhov geometrie.

Literatura

- [1] Student Software Guidebook [online]. education.ti.com/guides, 2015. [cit. 2015-05-12].
- [2] List of interactive geometry software [online]. http://en.wikipedia.org/wiki/List_of_interactive_geometry_software, 2015. [cit. 2015-05-14].
- [3] Qt Documentation [online]. <http://doc.qt.io/>, 2015 [cit. 2015-05-16].
- [4] Cabri [online]. <http://www.cabri.com/>, [cit. 2015-05-10].
- [5] GeoGebra [online]. <http://www.cinderella.de/>, [cit. 2015-05-10].
- [6] GeoGebra [online]. <https://www.geogebra.org/>, [cit. 2015-05-10].
- [7] Math Teacher Link [online]. <http://mtl.math.uiuc.edu/node/16>, [cit. 2015-05-10].
- [8] BARTSCH, H.-J.: *Matematické vzorce*. Praha: Academia, Čtvrté vydání, 2006, ISBN 80-200-1448-9.
- [9] BLUME, G. W.; HEID, M. K.: *Research on technology and the teaching and learning of mathematics*. Reston, VA: National Council of Teachers of Mathematics, 2008, ISBN 978-193-1576-215.
- [10] COXETER, H.: *Non-Euclidean geometry*. Washington, D.C.: Mathematical Association of America, sixth vydání, 1998, ISBN 08-838-5522-4.
- [11] Kortenkamp, U.: *Foundations of Dynamic Geometry*. Disertační práce, Zurich, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 1999.
- [12] KRUMP, L. v.: Neeuklidovská geometrie [online]. <http://www.talnet.cz/documents/18/f0cd37a3-1d98-4ff8-b186-ab15f51e2ec6>, [cit. 2015-05-10].
- [13] LABORDE, C.: Dynamic Geometry Environments as a Source of Rich Learning Contexts for the Complex Activity of Proving. *Educational Studies in Mathematics*, ročník 44, č. 1-2, 2000: s. 151–161, ISSN 0013-1954, doi:10.1023/A:1012793121648.
- [14] LEUNG, A.: Dragging in a Dynamic Geometry Environment Through the Lens of Variation. *International Journal of Computers for Mathematical Learning*, 2008, ISSN 1382-3892.
- [15] RICHTER-GEBERT, J.; KORTENKAMP, U. H.: *User manual for the interactive geometry software Cinderella*. New York: NSpringer, 2000, ISBN 35-406-7139-0.

- [16] ZELENÝ, J.; MANNOVÁ, B.: *Historie výpočetní techniky*. Praha: Scientia, první vydání, 2006, ISBN 80-869-6004-8.

Dodatek A

Obsah CD

Src	Zložka so zdrojovými súbormi
Bin	Zložka s preloženými aplikáciami
Report	Zložka so zdrojovými kódmi tejto správy
xcizma04_BP.pdf	Technická správa v elektronickej podobe
readme.txt	Nápoveda na spustenie aplikácie