

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

PREZENTAČNÍ SOFTWARE PRO ZNÁZORNĚNÍ ZÁKLADNÍCH JEVŮ V TROJÚHELNÍKU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ŠÁRKA KAŠPAROVÁ

BRNO 2009



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

PREZENTAČNÍ SOFTWARE PRO ZNÁZORNĚNÍ ZÁKLADNÍCH JEVŮ V TROJÚHELNÍKU

PRESENTATION SOFTWARE - TRIANGLE BASIC FACTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ŠÁRKA KAŠPAROVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍT ŠTANCL

BRNO 2009

Abstrakt

Tato bakalářská práce se věnuje vývoji prezentačního softwaru pro podporu výuky matematiky – základní geometrické jevy v trojúhelníku. Připomíná základní poznatky o výuce a trojúhelnících. Ve druhé části se potom zabývá návrhem a implementací programu a nástroji použitými při jeho vývoji.

Abstract

This bachelor's thesis is about development of presentation software supporting teaching of mathematics – triangle basic geometrical facts. It suggests basic knowledge about education and triangles. The second part is about analysis and implementation of the program and tools that were applied in its development.

Klíčová slova

prezentační software, výuka, matematika, geometrie, trojúhelník

Keywords

presentation software, education, mathematics, geometry, triangle

Citace

Šárka Kašparová: Prezentační software pro znázornění základních jevů v trojúhelníku, bakalářská práce, Brno, FIT VUT v Brně, 2009

Prezentační software pro znázornění základních jevů v trojúhelníku

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Víta Štancla a s použitím odborné literatury, kterou jsem všechnu uvedla na konci práce.

.....
Šárka Kašparová
12. května 2009

Poděkování

Na tomto místě bych ráda poděkovala Ing. Vítu Štanclovi za odborné vedení a podnětné konstruktivní připomínky k práci. Dále bych ráda poděkovala Mattu Bínů, studentu Pedagogické fakulty JU za odbornou konzultaci a testování aplikace.

© Šárka Kašparová, 2009.

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 Výuka matematiky	4
2.1 Vhodnost prezentačního softwaru pro výuku	4
2.2 Geometrické jevy v trojúhelníku	4
2.2.1 Trojúhelník	5
2.2.2 Vnitřní úhly	5
2.2.3 Vnější úhly	5
2.2.4 Výšky	5
2.2.5 Těžnice	6
2.2.6 Střední příčky	6
2.2.7 Kružnice opsaná	7
2.2.8 Kružnice vepsaná	7
2.3 Typy trojúhelníků	7
2.3.1 Trojúhelník obecný	7
2.3.2 Trojúhelník ostroúhlý	8
2.3.3 Trojúhelník tupoúhlý	8
2.3.4 Trojúhelník pravoúhlý	8
2.3.5 Trojúhelník rovnoramenný	10
2.3.6 Trojúhelník rovnostranný	10
3 Návrh programu	11
3.1 Grafické uživatelské rozhraní	11
3.1.1 Barevné schéma	11
3.1.2 Uživatelská interakce	11
3.1.3 Menu	13
3.2 Změny velikosti okna	14
4 Vývojové nástroje	16
4.1 wxWidgets	16
4.1.1 Teoretický základ wxWidgets	16
4.1.2 Použití wxWidgets v aplikaci	16
4.2 Code::Blocks	18
5 Implementace programu	20
5.1 Menu	20
5.2 Vykreslování	20
5.2.1 Vrcholy trojúhelníku	21

5.2.2	Názvy geometrických jevů	21
5.3	Ovládání myši	22
5.3.1	Stisknutí levého tlačítka	22
5.3.2	Držení levého tlačítka	23
5.4	Změny velikosti kreslicího plátna	23
5.5	Geometrie	24
5.5.1	Kolmice	24
5.5.2	Průsečík přímek	24
5.5.3	Vzdálenost dvou bodů	25
5.5.4	Střed úsečky	25
5.6	Geometrické jevy	25
5.6.1	Vnitřní úhly	25
5.6.2	Vnější úhly	26
5.6.3	Výšky	26
5.6.4	Těžnice	26
5.6.5	Střední příčky	26
5.6.6	Kružnice opsaná	26
5.6.7	Kružnice vepsaná	27
5.7	Uložení obrázku	27
6	Spouštění a ovládání programu	29
7	Závěr	30
A	Obsah CD	33

Kapitola 1

Úvod

Tato bakalářská práce se zabývá tvorbou prezentačního softwaru pro znázornění základních geometrických jevů v trojúhelníku. Jedná se o software podporující výuku matematiky, a to právě látku zabývající se trojúhelníky, která je dle standardních osnov probírána na druhém stupni základní školy a na nižším stupni osmiletých gymnázií.

V kapitole *Výuka matematiky* jsou popsány obecné vlastnosti trojúhelníků, typy trojúhelníků – ostroúhlý, tupoúhlý, pravoúhlý, rovnoramenný, rovnostranný – a jejich charakteristiky a probírané geometrické jevy spojené s trojúhelníky – vnitřní úhly, vnější úhly, výšky, těžnice, střední příčky, kružnice opsaná, kružnice vepsaná. Toto je názorně ukázáno na obrázcích. Také je zde zohledněna vhodnost prezentačního softwaru pro výuku a ukázány jeho výhody oproti tradičním formám výuky a srovnání s již existujícími matematickými výukovými programy.

Kapitola *Návrh programu* se zabývá postupy a principy při vytváření konceptu a funkčnosti vyvíjeného programu. Je zde popsáno grafické uživatelské rozhraní, jak po stránce vizuální, tedy grafické, což je doloženo obrázky, tak z hlediska ovládání uživatelem.

V kapitole *Vývojové nástroje* jsou popsány teoretické znalosti o vývojových nástrojích, které byly použity při tvorbě aplikace. Jedná se o framework wxWidgets, který velice usnadňuje vývoj aplikací s grafickým uživatelským rozhraním v programovacích jazycích C a C++, a vývojové prostředí Code::Blocks, jež framework wxWidgets podporuje a je možné jej používat v operačních systémech MS Windows, Linux i MAC OS X.

Kapitola *Implementace programu* se věnuje postupu implementace a jejímu konečnému výsledku. Jsou zde popsány jednotlivé metody a způsoby jejich implementace, ovládání menu, vykreslování grafických útvarů a výpočty jednotlivých geometrických jevů a podpůrných geometrických prvků, jako jsou kolmice, průsečíky přímek, vzdálenost dvou bodů a střed úsečky. Též se zde objevují některá chybná řešení, která se zpočátku jevila jako správná, ale nakonec od nich bylo upuštěno.

V kapitole *Spouštění a ovládání programu* jsou popsány způsoby, jak program spustit pod kterými operačními systémy a zjednodušený návod, jak již spuštěný program ovládat a jak s ním pracovat.

Kapitola *Závěr* uvádí zhodnocení celkového výsledku vývoje a implementace aplikace a jsou zde navržena jeho možná budoucí vylepšení.

Kapitola 2

Výuka matematiky

2.1 Vhodnost prezentačního softwaru pro výuku

Matematika, obzvláště geometrie, patří mezi předměty, u nichž nestačí se probíranou látku naučit nazpaměť, nýbrž je třeba jí rozumět, chápat ji a hlavně si dokázat vytvořit vizuální představu toho, co učitel požaduje. Proto je důležité, aby žáci pochopili co nejvíce již při výkladu učitele. Z tohoto důvodu je nutné verbální výklad podpořit názornými materiály, které umožňují lepší vnímání a formování představ, dávají podklady pro zobecňování a abstrakci a pomáhají k lepšímu pochopení podstatných znaků probírané látky. Poznání je díky nim potom trvalejší, protože je aktivizován celý mozek, a nejen jeho levá polovina, která je zodpovědná za řeč. Převzato z [2]

Program, který by umožňoval interaktivní názorné zobrazení, by byl přínosem nejen pro žáky, jimž by osvětloval probíranou látku, ale i pro učitele, kterým by usnadňoval výuku právě díky své interaktivnosti a jednoduchosti ovládání, což je jeho hlavní výhoda oproti klasické tabuli nebo elektronické prezentaci v podobě jednotlivých snímků.

Klasická tabule sice jistou míru interaktivnosti nabízí, je možné jednotlivé jevy přikreslovat či mazat, ale tento způsob výkladu klade jisté nároky na pedagogovu manuální zručnost, pro kterou nemusí mít dispozice. A mazání a přepisování celých již vytvořených příkladů, například kvůli mírnému posunutí jednoho z vrcholů, není zrovna optimální.

Co se týče snímkové prezentace, tak i když je možné v ní vytvořit několik různých příkladů a plynule mezi nimi přecházet, vždy, za současných programů pro vytváření prezentací, budou tyto příklady pouze statické a nebude možné bez obtíží měnit již vytvořené příklady během výkladu.

Geometrické jevy je dále možné zobrazovat v již existujících matematických či grafických programech, například Maple, Matlab, Cabri Geometry nebo GeoGebra, ale ty obvykle nejsou pro tuto funkci optimalizovány, neboť jsou zaměřeny spíše na přímou práci s nimi. Tedy jsou vhodné, pokud má každý žák při výuce k dispozici počítač, ale už ne pro pedagoga, který by chtěl látku promítat přes projektor.

2.2 Geometrické jevy v trojúhelníku

Jevy popsané v této kapitole patří do učiva druhého stupně základní školy, zde jsou uvedeny pro oživení a upřesnění pojmů, kterým se práce bude věnovat v dalších kapitolách. Části této kapitoly jsem převzala z [1].

2.2.1 Trojúhelník

Trojúhelník je definován jako průnik tří polorovin z nichž žádné dvě nejsou rovnoběžné. Body, v nichž se hraniční přímky těchto polorovin protínají, se nazývají vrcholy trojúhelníku a úsečky ohraničené těmito body jsou strany trojúhelníku. Součet délek dvou libovolných stran je vždy větší, než délka strany třetí.

2.2.2 Vnitřní úhly

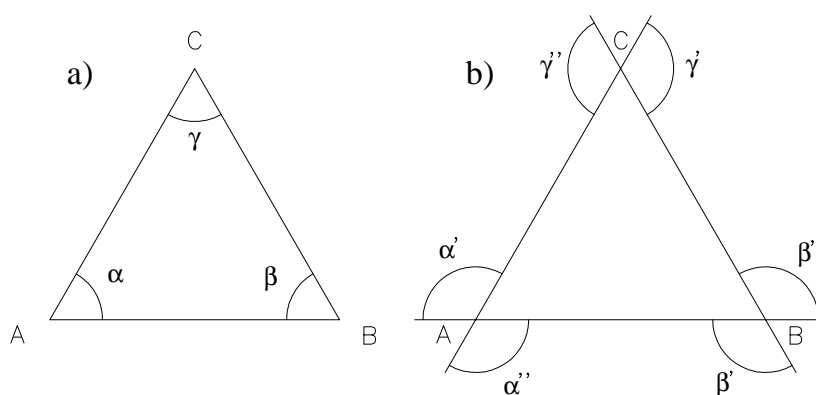
Vnitřní úhly trojúhelníku jsou ty úhly, které svírají strany daného trojúhelníku. Součet vnitřních úhlů v každém trojúhelníku je vždy 180° , tudíž všechny vnitřní úhly jsou menší než tato hodnota. Vnitřní úhly bývají důležité při postupu konstrukce trojúhelníku.

Obrázek 2.1 a) názorně ukazuje vnitřní úhly trojúhelníku.

2.2.3 Vnější úhly

Vnější úhly jsou vedlejší k úhlům vnitřním, jejich velikost je tedy 180° minus příslušný vnitřní úhel. Hodnota vnějšího úhlu při jednom vrcholu je tedy shodná se součtem vnitřních úhlů při ostatních vrcholech. Stranami těchto úhlů jsou jedna strana trojúhelníku a přímka, na níž leží strana vedlejší.

Obrázek 2.1 b) názorně ukazuje vnější úhly trojúhelníku.



Obrázek 2.1: a) Vnitřní úhly trojúhelníku, b) Vnější úhly trojúhelníku

2.2.4 Výšky

Výška trojúhelníku je důležitá pro výpočet jeho obsahu. Je znázorněna úsečkou kolmou na stranu trojúhelníku, s počátečním bodem na přímce, na níž tato strana leží, a koncovým

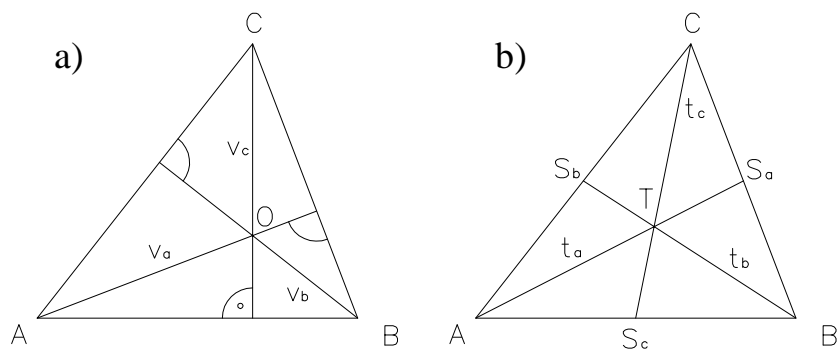
bodem ve vrcholu protilehlému k příslušné straně. Každá strana trojúhelníku má svoji výšku, čím menší strana, tím větší je výška. Průsečík přímek, na nichž leží výšky, se nazývá ortocentrum.

Obrázek 2.2 a) názorně ukazuje výšky v trojúhelníku a jeho ortocentrum.

2.2.5 Těžnice

Těžnice jsou úsečky spojující střed strany trojúhelníku s protilehlým vrcholem a rozdělují trojúhelník na dva menší trojúhelníky se stejným obsahem. Všechny tři těžnice se protínají v jednom bodě, který je nazýván těžiště a bývá označován písmenem T. Těžiště rozděluje každou těžnici v poměru 2:1, přičemž vzdálenost těžiště od vrcholu trojúhelníku je dvojnásobná oproti jeho vzdálenosti od strany trojúhelníku. Těžiště spolu s každými dvěma vrcholy vytváří trojúhelník, obsahy těchto tří nových trojúhelníků jsou shodné. Důležitost těžnic a těžiště bývá uplatněna spíše ve fyzice než v matematice.

Obrázek 2.2 b) názorně ukazuje těžnice v trojúhelníku a jeho těžiště. Trojúhelníky na obrázcích 2.2 a) i 2.2 b) jsou shodné, tudíž je zde vidět rozdíl mezi výškami a těžnicemi.

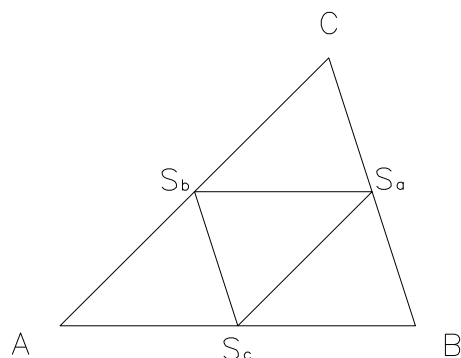


Obrázek 2.2: a) Výšky v trojúhelníku, b) Těžnice v trojúhelníku

2.2.6 Střední příčky

Střední příčky jsou spojnice středů stran trojúhelníku a rozdělují jej na čtyři shodné trojúhelníky - příčkový trojúhelník a tři trojúhelníky při jednotlivých vrcholech. Každá střední příčka je rovnoběžná s jednou stranou trojúhelníku a má velikost poloviny této strany.

Na obrázku 2.3 je možno vidět střední příčky trojúhelníku, je zde i názorně ukázáno, jak střední příčky rozdělují trojúhelník na čtyři menší, které jsou navzájem shodné.



Obrázek 2.3: Střední příčky v trojúhelníku

2.2.7 Kružnice opsaná

Kružnice opsaná prochází všemi třemi vrcholy trojúhelníku. Její střed se nachází v průsečíku os stran trojúhelníku a její poloměr se rovná vzdálenosti středu od libovolného vrcholu trojúhelníku.

Obrázek 2.4 a) znázorňuje kružnici trojúhelníku opsanou.

2.2.8 Kružnice vepsaná

Kružnice vepsaná se dotýká všech stran trojúhelníku, což znamená, že s každou stranou má společný právě jeden bod. Její střed se nachází v průsečíku os úhlů trojúhelníku a její poloměr se rovná vzdálenosti středu od libovolné strany, tedy bodu ležícím na straně trojúhelníku, jímž prochází přímka kolmá na tuto stranu, která zároveň prochází středem kružnice vepsané.

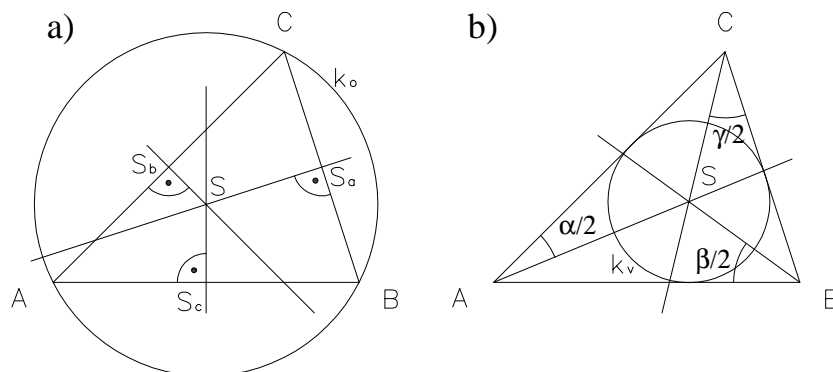
Obrázek 2.4 b) znázorňuje kružnici trojúhelníku vepsanou.

2.3 Typy trojúhelníků

Trojúhelníky jsou různé a dělí se do několika typů, které mají každý své specifické vlastnosti, z nichž některé jsou pevně dané a jiné jsou obecnějšího charakteru.

2.3.1 Trojúhelník obecný

Trojúhelník obecný lze nejlépe charakterizovat jako trojúhelník, jehož žádné dvě strany nejdou shodné a nemá žádné speciální vlastnosti. Obecný trojúhelník může být ostroúhlý i tupoúhlý.



Obrázek 2.4: a) Kružnice trojúhelníku opsaná, b) Kružnice trojúhelníku vepsaná

2.3.2 Trojúhelník ostroúhlý

Všechny úhly v tomto trojúhelníku jsou ostré, tj. každý úhel je menší než 90° . V ostroúhlém trojúhelníku leží ortocentrum a střed kružnice opsané uvnitř trojúhelníku.

Obrázek 2.5 názorně ukazuje rozdíl mezi ostroúhlým a tupoúhlým trojúhelníkem.

2.3.3 Trojúhelník tupoúhlý

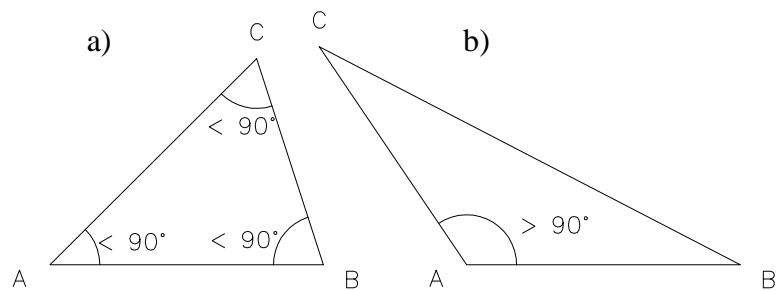
V tomto trojúhelníku je velikost jednoho z úhlů větší než 90° , dva zbývající jsou menší než 90° . Tento úhel leží proti nejdelší straně trojúhelníku. Ortocentrum a střed kružnice opsané leží mimo trojúhelník.

Obrázek 2.5 názorně ukazuje rozdíl mezi ostroúhlým a tupoúhlým trojúhelníkem.

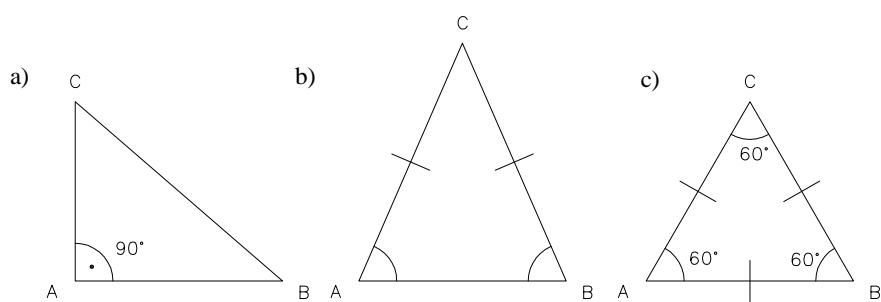
2.3.4 Trojúhelník pravoúhlý

Jeden úhel trojúhelníku má velikost přesně 90° , dva zbývající jsou menší než 90° . Strany, které jej svírají se nazývají odvěsny, strana třetí přepona. Trojúhelník nemůže mít více než jeden pravý úhel, neboť by součet jeho vnitřních úhlů přesahoval 180° , což je nemožné. V pravoúhlém trojúhelníku jsou dvě jeho výšky shodné s odvěsnami, ortocentrum leží ve vrcholu, při němž je pravý úhel. Střed kružnice opsané je shodný se středem přepony.

Na obrázku 2.6 a) je znázorněn pravoúhlý trojúhelník.



Obrázek 2.5: a) Trojúhelník ostroúhlý, b) Trojúhelník tupoúhlý



Obrázek 2.6: a) Trojúhelník pravoúhlý, b) Trojúhelník rovnoramenný, c) Trojúhelník rovnostranný

2.3.5 Trojúhelník rovnoramenný

Dvě strany tohoto trojúhelníku mají shodnou délku a nazývají se ramena. Rovnoramenný trojúhelník může být ostroúhlý, tupoúhlý i pravoúhlý. V tomto trojúhelníku je jedna výška shodná s jednou těžnicí. Na této přímce leží těžiště, ortocentrum a středy kružnice opsané i vepsané.

Na obrázku 2.6 b) je znázorněn rovnoramenný trojúhelník.

2.3.6 Trojúhelník rovnostranný

Všechny strany tohoto trojúhelníku mají shodnou délku, všechny vnitřní úhly mají velikost 60° . Úsečky těžnic a výšek jsou shodné, těžiště je shodné s ortocentrem a se středy kružnic vepsané a opsané.

Na obrázku 2.6 c) je znázorněn rovnostranný trojúhelník.

Kapitola 3

Návrh programu

Protože je vytvářený program určen primárně pro podporu výuky matematiky, bylo nutné jej navrhnout tak, aby výuku opravdu podporoval a usnadňoval, neboť špatně navržený by ji mohl zkomplikovat. Proto jsem jej tedy po vytvoření první, testovací verze zkonzultovala se studentem Pedagogické fakulty Jihočeské univerzity, obor matematika a fyzika, který mi dal několik doporučení, z nichž jsem část použila ve vývoji programu.

3.1 Grafické uživatelské rozhraní

Jelikož je vyvíjený program určen pro usnadňování výuky, je nutné, aby jeho grafické uživatelské rozhraní bylo přehledné, srozumitelné, intuitivní a jinak uživatelsky přívětivé. Zvolila jsem tedy klasický vzhled, kde v horní části okna je lišta s jednoduchým menu a většinu prostoru vyplňuje kreslicí plocha, kde bude znázorňován trojúhelník a jeho základní geometrické jevy.

3.1.1 Barevné schéma

Pozadí tohoto programu má světle šedou barvu, která „nebije do očí“ tolik jako bílá a ani není nijak výrazná, takže ostatní barvy na ní mohou vyniknout a „neztrácejí se“.

Pro lepší přehlednost jsou jednotlivé geometrické jevy barevně odlišené a to jak v odstínech, tak v jasů. Jednotlivé barvy jsem volila s ohledem na různorodost, aby nemohlo dojít k záměně¹ jevů. Barevné nastavení je stálé, uživatelská volitelnost by mohla být návrhem na vylepšení programu. Barevně odlišené jsou rovněž názvy zobrazovaných jevů, které mají stejnou barvu jako příslušný jev.

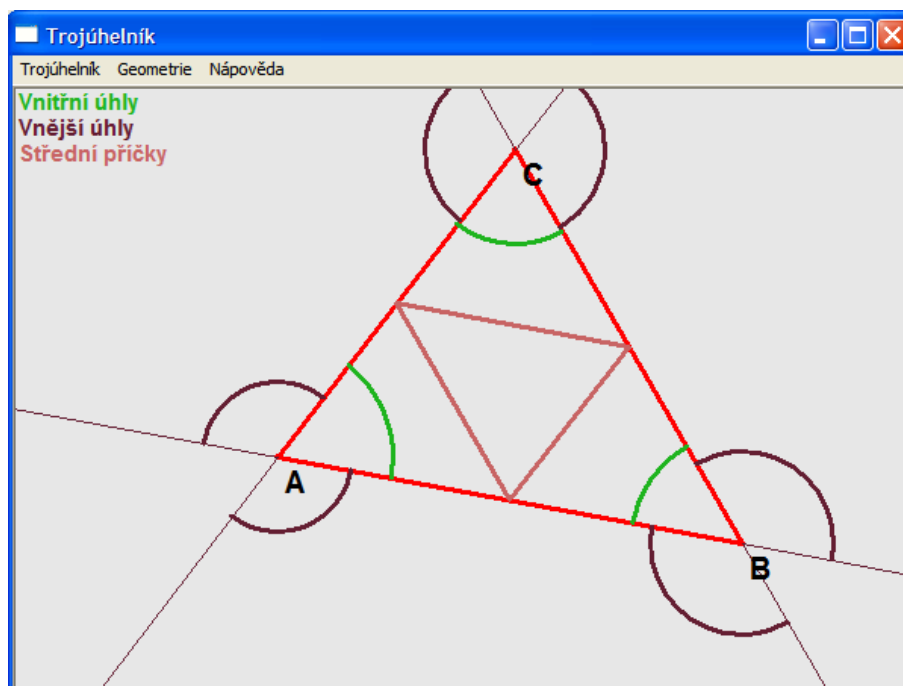
Na obrázcích 3.1, 3.2 a 3.3, si můžete prohlédnout konečné barevné schéma aplikace.

Je ovšem nutné podotknout, že barvy nejsou optimalizované pro prezentování v odstínech šedi.

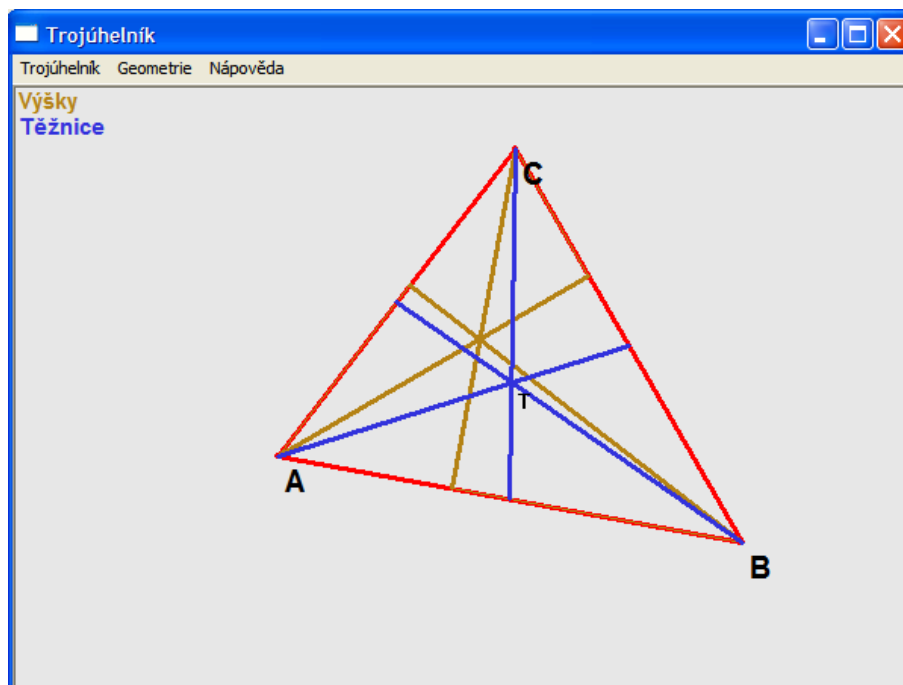
3.1.2 Uživatelská interakce

Jak jsem uvedla výše, program by měl mít výhodu oproti nákresům či prezentaci právě v interaktivnosti a v rámci výuky by měl být co nejintuitivnější, tudíž bylo žádoucí zajistit, aby si uživatel sám zadal body představující vrcholy trojúhelníku. Zadávat by je měl

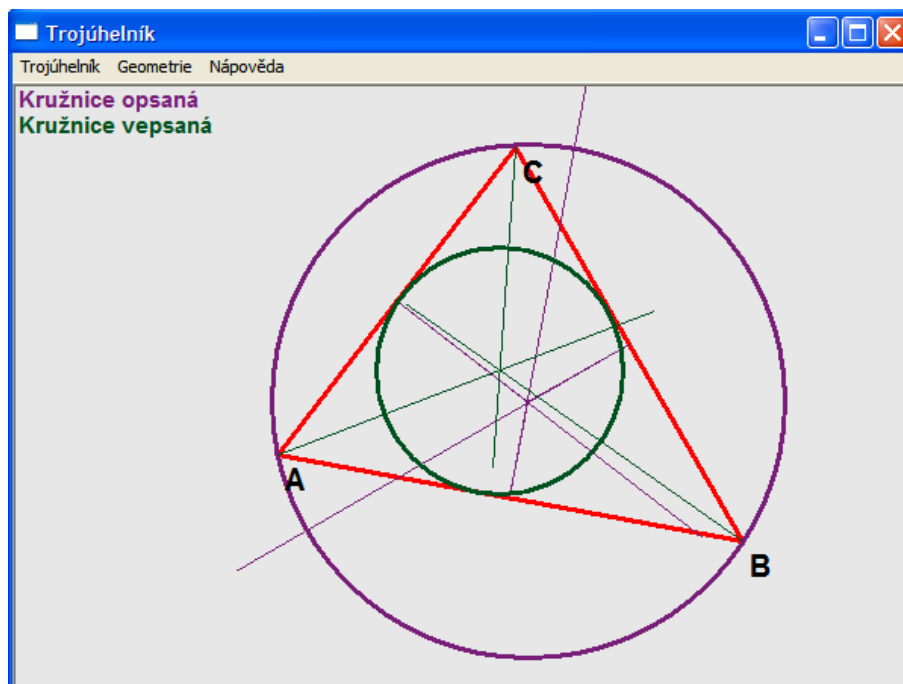
¹Je zřejmé, že například těžnice a kružnici opsanou rozliší téměř každý, nicméně v případě těžnic a výšek tomu už být nemusí.



Obrázek 3.1: Barevné schéma 1: Vnitřní a vnější úhly, střední příčky



Obrázek 3.2: Barevné schéma 2: Výšky a těžnice



Obrázek 3.3: Barevné schéma 3: Kružnice opsaná a vepsaná

kliknutím myši, což je běžným uživatelem počítače považováno za standard, a program by měl přijmout jen tři zadané body, tedy po třech kliknutích již žádné další body nepřidává.

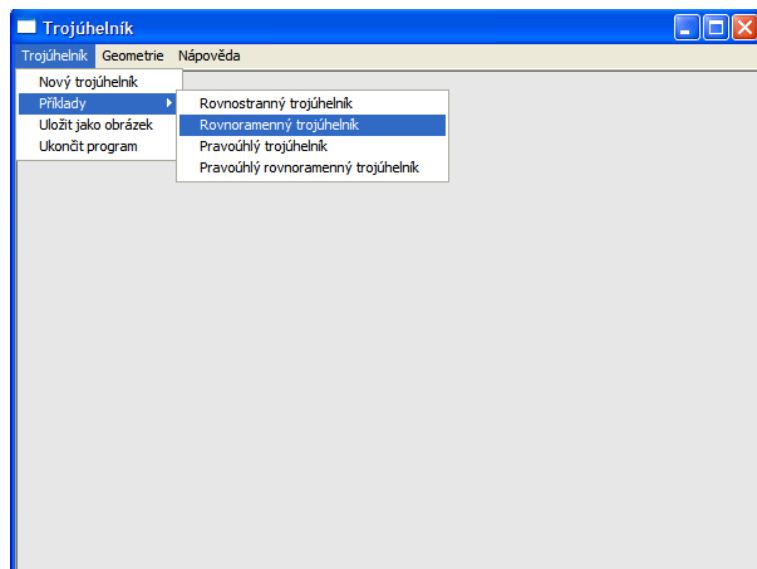
Další funkce vhodná pro posílení interaktivnosti, je možnost pohybování s již vytvořenými vrcholy trojúhelníku, aby učitel mohl lépe znázornit, jak vypadají jednotlivé geometrické jevy a jak se mění při změnách stran a úhlů trojúhelníku.

3.1.3 Menu

Položky v liště menu jsou rozděleny podle svých funkcí do tří skupin, a to *Trojúhelník*, *Geometrie* a *Nápověda*. Názvy skupin a jednotlivých položek jsem volila tak, aby byly výstižné, tedy musí být na první pohled poznat, k čemu slouží. Také by neměly obsahovat mnoho textu, neboť příliš dlouhé názvy by mohly být uživateli nepříjemné.

Ve skupině *Trojúhelník* jsou implementovány položky *Nový trojúhelník*, *Příklady*, *Uložit jako obrázek* a *Ukončit program*.

- *Nový trojúhelník* – po kliknutí na tuto položku má uživatel možnost zadat další trojúhelník, ten původní je smazán. Současně je provedeno nastavení položek v menu *Geometrie* na neaktivní.
- *Příklady* – zde je implementováno automatické vykreslení nového trojúhelníku konkrétního typu, rovnostranný, rovnoramenný, pravoúhlý a rovnostranný pravoúhlý trojúhelník. Jednotlivé metody nastaví souřadnice vrcholů požadovaného trojúhelníku, pokud již nějaký trojúhelník zobrazen je, souřadnice jeho vrcholů jsou změněny.
- *Uložit jako obrázek* – umožňuje uložení momentálně promítaného jevu do podoby bitmapového obrázku o rozměrech současného kreslicího plátna.



Obrázek 3.4: Menu *Trojúhelník* v aplikaci

- *Ukončit program* – bezpečně ukončí program.

Ve skupině *Geometrie* může uživatel nalézt probírané geometrické jevy, jednotlivé položky označují jevy podle svého názvu. Aktivní geometrické jevy jsou označeny zaškrtnutím. Dále se jako vhodné ukázaly položky *Zobrazit vše* a *Skrýt vše*, které v momentálně prezentovaném trojúhelníku zobrazí, nebo skryjí všechny geometrické jevy. Uživatel tedy, pokud chce zobrazit nebo skrýt větší množství jevů najednou, tak nemusí činit pro každý jev zvlášť.

Nápověda obsahuje pouze jedinou položku, a to *O programu*, která ve výchozím webovém prohlížeči zobrazí návod, jak aplikaci používat.

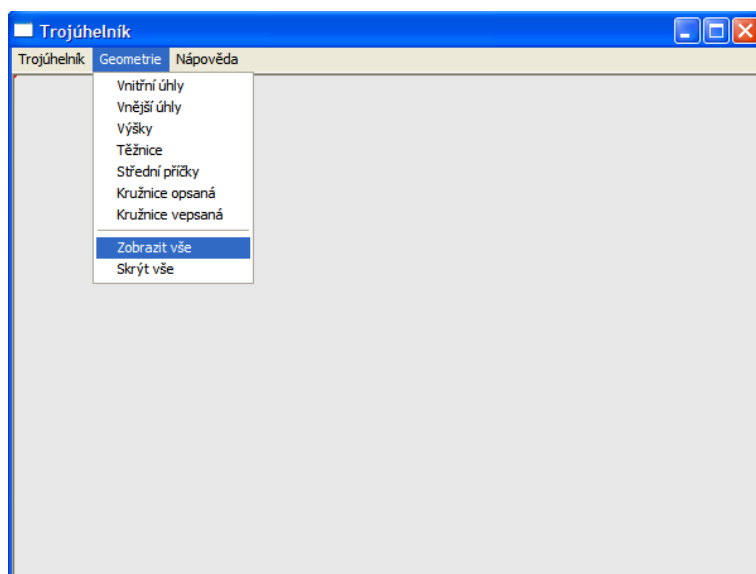
Na obrázcích 3.4 a 3.5 si můžete prohlédnout uspořádání menu v aplikaci.

3.2 Změny velikosti okna

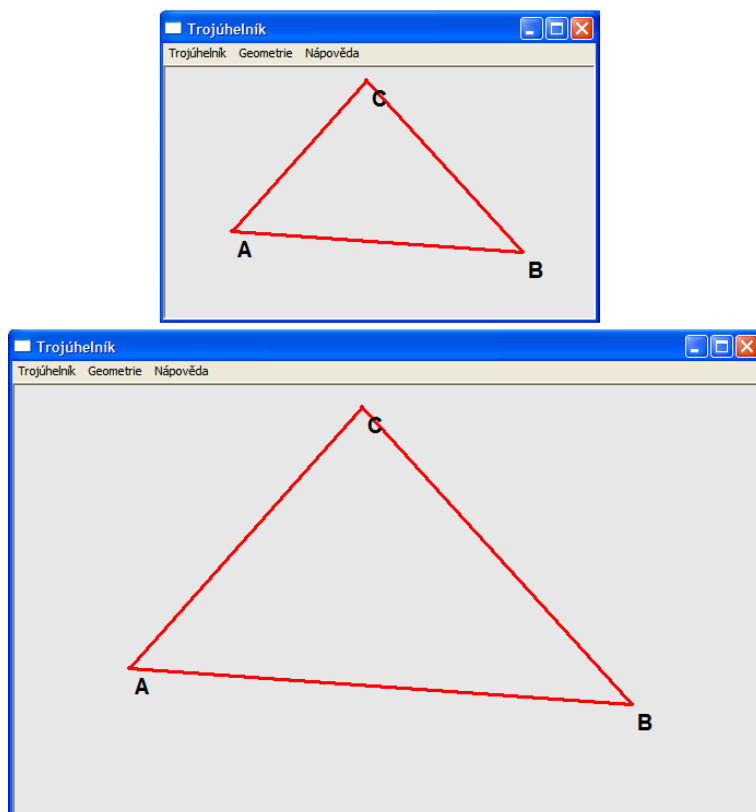
Drobným problémem při vývoji programu se stalo přizpůsobení souřadnic vrcholů trojúhelníku při změně velikosti kreslicího plátna. Při nastavení určeném na začátku implementace se velikost kreslicího plátna mění automaticky, když se změní velikost okna, ale souřadnice vrcholů trojúhelníku zůstávají stejné, tedy nedochází ke zmenšování a zvětšování vykreslených prvků.

Tento efekt je ovšem nežádoucí, neboť pedagog používající tento program by nemohl libovolně měnit velikost okna, například při vícenásobném spuštění aplikace pro porovnání stejných geometrických jevů na různých trojúhelnících, aniž by ohrozil výsledné zobrazení, například při zmenšení by se některé vrcholy trojúhelníku mohly dostat mimo zobrazovanou plochu a při zvětšení by se na zobrazované ploše objevilo příliš mnoho „prázdného“ místa.

Řešením tohoto problému je přepočítání souřadnic vrcholů trojúhelníku podle poměru původní a nové velikosti okna, ukázkou programu s tímto řešením můžete vidět na obrázku 3.6.



Obrázek 3.5: Menu *Geometrie* v aplikaci



Obrázek 3.6: Při změně velikosti okna se změni i souřadnice bodů

Kapitola 4

Vývojové nástroje

4.1 wxWidgets

4.1.1 Teoretický základ wxWidgets

Protože je vyvíjený program zaměřený hlavně na svou grafickou stránku, bylo vhodné najít nástroj, který by tvorbu tohoto programu usnadňoval. Zvolila jsem framework wxWidgets, což je velmi užitečný nástroj pro tvorbu aplikací s grafickým uživatelským rozhraním. Je to framework, který programátorům relativně usnadňuje tvorbu jejich aplikací, například vykreslováním speciálních grafických prvků či obrázků, reagováním na vstup z klávesnice a myši a postaráním se o standardní chování vyvíjených aplikací. Knihovna wxWidgets obsahuje velký počet tříd a metod, které mohou být použity jak přímo, tak pro další úpravy.

wxWidgets je narozdíl od mnoha jiných frameworků primárně multiplatformní. Ačkoli bývá často nazýván vývojovým nástrojem pro tvorbu aplikací s grafickým uživatelským rozhraním, má mnoho užitečných vlastností pro různá stadia vývoje aplikací. Aplikační programové rozhraní je velice podobné na všech podporovaných platformách, díky čemuž je možné aplikaci vyvinout například pod operačním systémem MS Windows a s minimálními, pokud vůbec nějakými, úpravami ji pak překompilovat pod Linuxem či MAC OS X.

Kdekoliv je to možné, wxWidgets používá nativních ovládacích prvků daného systému, a až když to možné není, použije vlastní. Díky tomu nejen poskytuje aplikacím nativní vzhled a chování, ale opravdu nativní je. Toto bývá nesmírně důležité z hlediska přijetí uživatelem, neboť ne všichni uživatelé jsou ochotni rychle se učít novým návykům.

Ve velké míře využívá wxWidgets principů objektově orientovaného programování, každé okno aplikace je reprezentováno objektem v jazyce C++. Díky tomu, že dobře definované základní chování těchto objektů je již součástí komponent wxWidgets, má programátor značně zjednodušenou roli při sestavování těchto komponent tak, aby tvořily harmonický celek v podobě funkční a uživatelsky přívětivé aplikace. Převzato z [3]

4.1.2 Použití wxWidgets v aplikaci

wxWidgets jsem zvolila z důvodů uvedených již v popisu tohoto frameworku, hlavní roli zde hrálo velké usnadnění práce s grafickými prvky v jazyce C++. Jako zdroj informací jsem využívala oficiální internetové stránky [4]. Hlavičkové sobory tříd wxWidgets, které jsem použila ve vyvíjeném programu jsou:

- wx/wx.h – pro práci s okny
- wx/event.h – pro obsluhování událostí, například kliknutí myši, nebo příkaz z menu
- wx/menu.h – pro vytvoření lišty menu, položek menu a práci s nimi
- wx/filefn.h – pro práci s adresáři, nalezení cesty k nápovědě. Využívá třídu `wxPathList`
- wx/dc.h – pro vykreslování jednotlivých jevů
- wx/image.h – pro práci s obrázky, konkrétně ukládání v bitmapovém formátu

wxDC

Třída `wxDC` je device context, objekt, v němž jsou definovány grafické prvky a text, které mají být zobrazovány na obrazovce počítače nebo tisknuty tiskárnou. Je to abstraktní základní třída a objekt této třídy nemůže být vytvořen přímo, nýbrž musí být použita některá ze tříd z této odvozených, jako jsou například `wxPaintDC`, `wxClientDC`, `wxWindowDC`, `wxScreenDC`, `wxMemoryDC` nebo `wxPrinterDC`. Device contexty, které jsou spojeny s okny, což jsou `wxClientDC`, `wxWindowDC` a `wxPaintDC`, používají písma a barvy dle oken, zatímco ostatní třídy používají systémové hodnoty, proto je důležité příslušná písma a barvy před jejich používáním správně nastavit.

V aplikaci byly použity device contexty tříd `wxPaintDC` a `wxMemoryDC`.

wxMenu

Menu je vyskakovací nebo rolovací seznam položek, z nichž jedna může být vybrána před tím, než menu zmizí. Jednotlivé položky menu mají celočíselný identifikátor (ID), který je důležitý pro výběr akce nebo změně samotného menu. Položka se speciálním identifikátorem o hodnotě -1 je oddělovač, této položce není přiřazena žádná akce, ale bývá zobrazována jako čára pro oddělení částí menu.

Položky v menu mohou být normální, zaškrťovací¹ nebo přepínací². Normální položky nemají žádné specifické znaky. Položky zaškrťovací mají boolovský příznak, který je zobrazován vedle názvu položky. Hodnota tohoto příznaku se změní obvykle při kliknutí na položku, ale může být změněna přímo metodou `wxMenu::Check()`, podle parametru buď na „zaškrtnuto“, nebo „nezaškrtnuto“. Stav této hodnoty lze zjistit zavoláním metody `wxMenu::IsChecked()`.

Přepínací položky jsou podobné těm zaškrťovacím, ovšem pokud je jedna aktivní, ostatní ve skupině musí být neaktivní. Přepínací skupina musí být tvořena souvislou řadou několika položek. Přepínací položky se ve vyvíjeném programu nevyskytují, zde jsou uvedeny pouze pro úplnost.

wxEvent

Událost je struktura uchovávající informaci o tom, jaká metoda se má zavolat, nebo jaký příkaz provést, nastane-li v programu situace, která toto vyžaduje. `wxEvent` je abstraktní základní třída, z níž vycházejí další třídy událostí.

¹přeloženo z anglického „check item“

²přeloženo z anglického „radio item“

Řízení událostí ve wxWidgets může probíhat dvěma způsoby. Buď přes tabulku událostí³, která jednotlivé události řídí staticky, tedy k jednotlivým událostem jsou přiřazeny příslušné metody během překladu programu, nebo přes volání `wxEvtHandler::Bind<>()`, jež může být použito dynamicky, tedy události přiřadit řídicí metodu v závislosti na běhu programu. Tabulky událostí mohou řídit pouze události předem definované, takže jejich použití je méně přizpůsobivé, než použití `Bind<>()`. Na druhou stranu jsou tabulky událostí stručnější, centralizují události na jednom místě a je s ní snadnější předcházet chybám, jež mohou na základě nepředvídatelného chování uživatele.

Vyvíjený program, který je součástí této bakalářské práce, využívá právě tabulku událostí. V programu jsou použity čtyři třídy událostí, a to `wxMouseEvent`, `wxPaintEvent`, `wxSizeEvent` a `wxCommandEvent`.

- `wxMouseEvent` – Tato třída uchovává informace o událostech generovaných myši, což jsou stisknutí a uvolnění tlačítka myši a pohyby myši.
- `wxPaintEvent` – Událost této třídy je volána pokaždé, když okno aplikace vyžaduje překreslení svého obsahu. Pro řízení událostí třídy `wxPaintEvent` je vždy nutné vytvořit objekt třídy `wxPaintDC`, i když nemusí být aplikací používán. Jinak je možné, že pod operačním systémem MS Windows bude docházet k chybnému překreslování.
- `wxSizeEvent` – Události velikosti uchovávají informace o změnách velikostí `wxWindow`. Pokud je změněna velikost okna, zavolá se makro pro řízení událostí `EVT_SIZE`. Při změně velikosti okna je obvykle „poškozena“ jen část okna a je potřeba překreslit pouze danou část. Ovšem pokud vykreslování závisí přímo na velikosti okna, což je případ i aplikace vyvíjené v rámci této bakalářské práce, je potřeba explicitně vyčistit celý device context. Z tohoto důvodu se používá volání metody `wxWindow::Refresh()` pro zneplatnění obsahu celého okna.
- `wxCommandEvent` – Tato třída uchovává informace o příkazových událostech, které vycházejí z množství jednoduchých řídicích prvků. V aplikaci je použito makro `EVT_MENU`, které je generováno při výběru položky v menu.

wxImage

Tato třída zapouzdřuje platformově nezávislé obrázky. Tyto obrázky mohou být vytvořeny z dat nebo pomocí `wxBitmap::ConvertToImage()`. Hotový obrázek lze pro používání nahrát ze souboru v různých formátech a je rozšiřitelný do jiných formátů prostřednictvím služby formátu obrázku⁴. Dostupné funkce mohou být použity na základní úpravy obrázku, neboť pracují s nastavením a zjištěním stavu jednotlivých bitů obrázku.

Objekt třídy `wxImage` nemůže být vytvořen přímo z objektu třídy `wxDC` a obráceně. Místo toho musí být vytvořen platformově specifický objekt třídy `wxBitmap`, který zde hraje roli prostředníka.

4.2 Code::Blocks

Code::Blocks je volně šiřitelné multiplatformní vývojové prostředí zaměřené na programovací jazyky C a C++. Je vyvíjeno v jazyce C++ s využitím vývojového nástroje wxWidgets. Poslední stabilní verze vyšla v únoru roku 2008. Oficiálně vydávané verze jsou pro

³přeloženo z anglického „event table“

⁴přeloženo z anglického „image format handler“

MS Windows, Linux a MAC OS X. Code::Blocks podporuje použití více překladačů, například GCC/MinGW, Microsoft Visual C++, Watcom, Borland C++ a další, a je možné do něj importovat projekty z jiných vývojových prostředí, konkrétně Dev-C++ a Microsoft Visual C++. Dále lze v tomto prostředí tvořit projekty s DirectX, OpenGL, GLUT, Irrlich, Ogre, Qt, SDL, wxWidgets a jinými. Převzato z webových stránek [5], kde je prostředí Code::Blocks také volně ke stažení a jsou zde k dispozici podrobné návody.

Textový editor prostředí Code::Blocks je založen na editační komponentě Scintilla, což je volně šiřitelná editovací komponenta pro Win32 a GTK+. Pro zpřehlednění zdrojového kódu umožňuje číslování řádků, zvýrazňování syntaxe, skrývání částí kódu, nebo dokončování kódu.

Zvýrazňování syntaxe je velice užitečné, protože nejen opticky oddělí klíčová slova programovacího jazyka od číselných hodnot, názvů proměnných a komentářů, ale i dočasně zvýrazní závorku párovou k aktuálně označené, což je v drtivé většině případů velmi přínosné. Každý programátor jistě alespoň jednou zažil situaci, kdy marně hledal chybu v kódu, a přitom měl jen špatně umístěné závorky.

Skrývání částí kódu je výhodné zejména pro delší kódy, kdy je možné celou funkci, metodu, cyklus či podmíněný výraz skrýt a zobrazit až ve chvíli, kdy je potřeba upravit nebo zjistit údaj uvnitř části tohoto skrytého celku.

Dokončování kódu znamená, že editor rozpozná, který název proměnné nebo klíčového slova chce programátor napsat, a rozbalí nabídku s možnostmi. Je to užitečné u delších názvů proměnných a metod, jejichž název si programátor nemusí pamatovat přesně do písmene, ale stačí začátek a přibližný zbytek.

Kapitola 5

Implementace programu

Program byl dle zadání vytvářen v jazyce C++ za použití objektově orientovaných principů. Využíváno bylo též výše zmíněného vývojového nástroje wxWidgets a vývojového prostředí Code::Blocks. Informace o potřebných třídách a metodách wxWidgets jsem převzala z oficiální dokumentace na webových stránkách [4].

Zdrojové soubory vytvářeného programu jsou `main.cpp`, `myframe.cpp` a `mypanel.cpp` a k nim odpovídající hlavičkové soubory `main.h`, `myframe.h` a `mypanel.h`.

V souborech `myframe.cpp` a `myframe.h` jsou implementovány vlastnosti okna aplikace a způsob řízení položek v menu.

Soubory `mypanel.cpp` a `mypanel.h` implementují vykreslování a výpočetní logiku aplikace.

`main.cpp` a `main.h` zajišťují zobrazení okna aplikace.

5.1 Menu

Lišta menu byla vytvořena pomocí třídy wxWidgets `wxMenuBar`, samotné položky potom pomocí třídy `wxMenu`.

Jednotlivé položky menu používají pro volání příslušných metod dříve zmíněnou tabulku událostí, kde je pro každou položku deklarována jedna událost a s ní spojená metoda.

Položka v menu *Nový trojúhelník* je implementována metodami `MyFrame::OnMyReset()` a `MyPanel::ResetTriangle()`, které zajistí smazání trojúhelníku, tedy nastavení všech souřadnic bodů na `-1`, všech konstant pro vykreslování jednotlivých geometrických jevů na hodnotu `false`, vynulování počítadla vrcholů a uložení současné velikosti kreslicí plochy.

Kliknutí na *Ukončit program* zavolá metodu `MyFrame::OnQuit()`, která využitím příkazu `Close()` bezpečně ukončí program.

Položky ve skupině *Geometrie* volají přes tabulku událostí metody pro vykreslování jednotlivých geometrických jevů, tyto jevy budou popsány dále.

5.2 Vykreslování

Základem pro vykreslování jednotlivých jevů je device context třídy `wxPaintDC`, jehož přítomnost je vyžadována v metodě `MyPanel::OnPaint()`, která je zavolána pokaždé, když dojde k události vyvolávající překreslení okna. Pro tuto událost je ve wxWidgets makro `EVT_PAINT`, jež je generováno po zavolání metody `wxWindow::Refresh()`. Tato metoda je volána pokaždé, když uživatel provede změnu vyžadující překreslení okna aplikace.

5.2.1 Vrcholy trojúhelníku

Program používá pro uchování hodnot vrcholů trojúhelníku pole celých čísel, `bod_x[]` a `bod_y[]`, což jsou souřadnice jednotlivých bodů v x-ové a y-ové ose, přičemž bod `[0,0]` se nachází v levém horním rohu kreslicího plátna. Program dále uchovává hodnoty v proměnných `fixed_bod_x[]` a `fixed_bod_y[]`, které mají zásadní význam při změnách velikosti kreslicího plátna a přepočítávání souřadnic bodů, což bude zmíněno později.

Během implementace programu došlo k nežádoucím efektům u některých jevů, neboť pořadí vrcholů trojúhelníku mohlo být zadáno jak po, tak proti směru hodinových ručiček a u některých metod `wxWidgets`, například `wxDC::DrawArc()` pro vykreslení oblouku, záleží na pořadí počátečního a koncového bodu. Bylo tedy nutné, aby v každém případě bylo pořadí vrcholů trojúhelníku zadáno jedním směrem. Proto při každé změně souřadnic některého z vrcholů počítá metoda `MyPanel::SwapCoord()` z-ovou souřadnici vektorového součinu dvou stran trojúhelníku. Oba tyto vektory mají počáteční bod ve druhém zadaném vrcholu, tedy vrcholu B. Pokud je vypočtená souřadnice kladná, metoda zamění souřadnice vrcholů B a C.

Popisy vrcholů program zobrazí až po vykreslení všech požadovaných jevů. Při rychlosti současných počítačů je časový rozdíl mezi vykreslením na počátku a vykreslením na konci metody okem nepostřehnutelný, tudíž jediný důvod tohoto uspořádání je ten, aby bylo zajištěno, že popisy vrcholů nebudou znázorněnými jevy překryty.

5.2.2 Názvy geometrických jevů

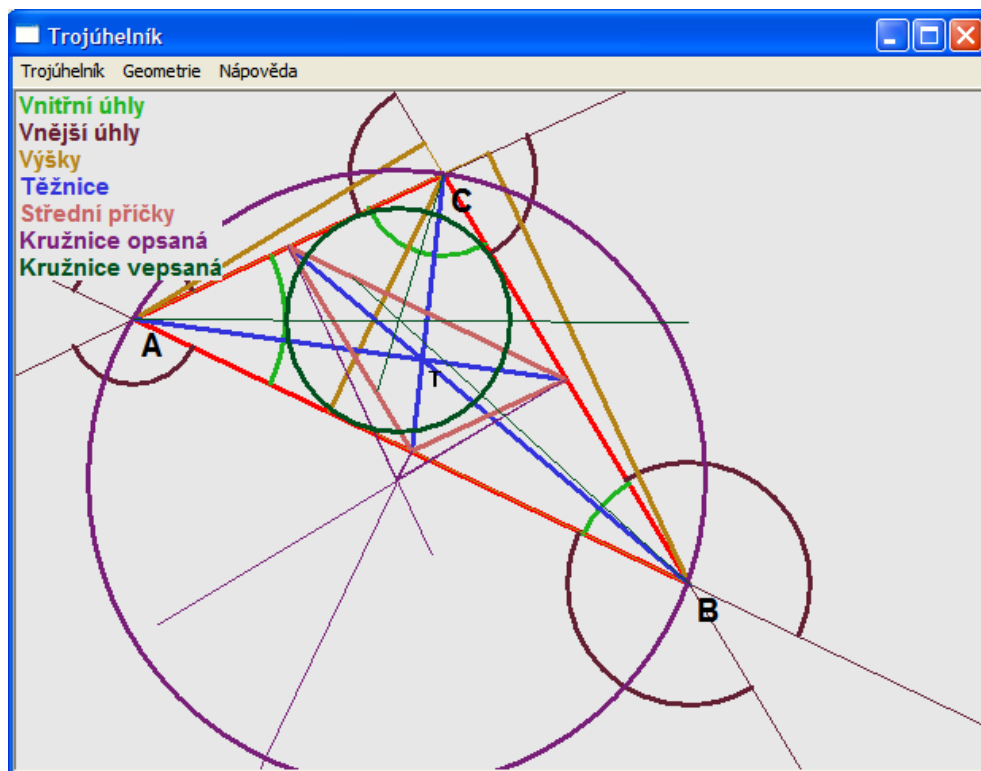
Vykreslování názvů jednotlivých geometrických jevů je z technického hlediska při použití `wxWidgets` věc téměř triviální, ovšem z celkového pohledu, kdy je nutno vzít v úvahu i estetický vzhled aplikace, je potřeba nad tím popřemýšlet. Samotné vykreslování je prováděno metodou `wxWidgets wxDC::DrawText()`, která jako parametry přijímá textový řetězec, jenž se má vykreslit a souřadnice levého horního rohu tohoto textu. Umístění textu jsem zvolila v levém horním rohu kreslicího plátna, kde je menší pravděpodobnost, že bude při výkladu „překážet“.

Způsobů, jak byly texty vykreslovány, se během vývoje programu vystřídalo několik. Zpočátku měl každý název daný svou pevnou polohu a zobrazil se až s příslušným jevem. To se ukázalo jako nepříliš vhodné, neboť při určitých kombinacích jevů vznikala mezi jednotlivými nápisy „prázdná“ místa, která působila rušivě. Navíc při určitých polohách trojúhelníku některé jevy zasahovaly do nápisů a činily je tak špatně čitelnými.

Dalším způsobem bylo permanentní zobrazování jevů, kde před jejich vykreslením byl vykreslen obdélník o stejné barvě jako pozadí programu, a tím se zabránilo zasahování jevů do textu. Nicméně názvy působily rušivým dojmem, pokud zrovna nebyly zobrazeny všechny jevy současně.

Konečný způsob implementace vykreslování názvů pracuje s polem textových řetězců `factNames[]`, do něhož se ukládají názvy právě vykreslovaných jevů. Druhé pole, `colour2[]`, ukládá jejich barvy. S každým vykresleným jevem je současně inkrementována celočíselná proměnná `position`, která udává pozici dané barvy a názvu v polích. Po vykreslení všech požadovaných geometrických jevů poté udává jejich počet a na její hodnotě je závislá výška obdélníku vykreslovaného bezprostředně pod názvy jevů. Názvy jsou zobrazeny až na konci metody `MyPanel::OnPaint()`, je tedy zaručeno, že nebudou ničím překryty.

Na obrázku 5.1 je možno prohlédnout si konečnou implementaci vykreslování názvů. Zobrazení všech jevů sice vypadá chaoticky, ale při výuce se málokdy stane, aby je pedagog



Obrázek 5.1: Náhled na vykreslování názvů

potřeboval zobrazit všechny, tudíž i prostor, který zabírají názvy jevů, bude menší a je méně pravděpodobné, že bude zasahovat do trojúhelníku.

5.3 Ovládání myši

Program je z převážné části ovládán myší, což je pro běžného uživatele nejpřirozenější způsob ovládání jakéhokoli programu s grafickým uživatelským rozhraním. Každá změna způsobená prostřednictvím myši generuje makro `EVT_MOUSE_EVENTS`, které je deklarované v tabulce událostí. Pokud k takovéto události dojde, je zavolána metoda `MyPanel::OnMyMouse()`, která tuto událost dokáže buď zpracovat, nebo ignorovat.

Zpracovávané události jsou stisknutí levého tlačítka myši (kliknutí) a držení levého tlačítka myši (tažení).

5.3.1 Stisknutí levého tlačítka

Metoda `MyPanel::OnMyMouse()` nejprve zjistí, zda došlo ke stisku levého tlačítka. Pokud došlo, metoda `wxMouseEvent::LeftDown()` vrací hodnotu `true`. Dále je zjištěno, zda je počítadlo vrcholů trojúhelníku menší než hodnota proměnné `quantity`, jež označuje počet vrcholů trojúhelníku, tedy ještě nebyly zadány všechny. Jestliže je i tato podmínka splněna, pomocí `wxMouseEvent::GetX()` a `wxMouseEvent::GetY()` jsou zjištěny souřadnice polohy myši, které jsou uloženy do proměnných `bod_x[]`, `bod_y[]`, `fixed_bod_x[]` a `fixed_bod_y[]`. Poté metoda zvýší počítadlo vrcholů trojúhelníku.

Ve chvíli, kdy jsou známé souřadnice všech bodů trojúhelníku, je provedena kontrola pořadí vrcholů metodou `MyPanel::SwapCoord()` a do proměnných `pansize_x` a `pansize_y` jsou uloženy šířka a výška kreslicího plátna.

5.3.2 Držení levého tlačítka

Tažení myši za účelem pohybování libovolným vrcholem trojúhelníku je zjištěno pomocí metody `wxMouseEvent::LeftIsDown()`, která vrací hodnotu `true` v případě, že levé tlačítko myši je stisknuté, tedy dole. Pokud toto nastane, program následně zkontroluje, zda všechny vrcholy trojúhelníku mají již souřadnice a zda se myš nachází v blízkosti nějakého z nich. Blízkostí je zde myšlen nejen daný vrchol trojúhelníku, ale všechny body od něj ve vzdálenosti určené konstantou `vicinity`, jež je nastavena na hodnotu tři, což se jeví jako optimální vzhledem k tloušťce čáry. Testování na blízkost bodu je prováděno v příkazu `if-else-if`, kdy je zkontrolován první zadaný bod, když nevyhovuje, tak druhý a když ani ten nevyhovuje, tak třetí. Kód to sice není optimální, ale je takto zaručeno, že se pohne pouze jeden bod, i když budou oba velmi blízko sebe. V počáteční verzi programu byl v tomto místě cyklus `for` a podmínka testování uvnitř těla cyklu, ale objevovaly se zde chyby, které bránily hladkému běhu programu po spuštění.

Pokud se tedy myš v této blízkosti nachází a levé tlačítko je stisknuté, změní se souřadnice příslušného bodu na souřadnice polohy myši. Tím tedy při tažení myši dochází k plynulým změnám polohy bodu. Ovšem při příliš rychlých pohybech myši aplikace „nestíhá“ a myš se dostane mimo rozsah, kde je možné s měnit polohu vrcholu trojúhelníku. Proto je nutné pohybovat myši pomalu. Současně s proměnnými `bod_x[]` a `bod_y[]` pro konkrétní vrchol trojúhelníku se mění i proměnné `fixed_bod_x[]` a `fixed_bod_y[]` pro všechny vrcholy trojúhelníku. Toto má zásadní význam při změnách velikosti okna a kreslicího plátna, jak bude uvedeno dále.

5.4 Změny velikosti kreslicího plátna

Jak jsem uvedla výše, přepočítání souřadnic vrcholů trojúhelníku podle poměru původní a nové velikosti okna je více než vhodné pro vyvíjenou aplikaci. Proto jsou v programu dvě globální proměnné, `pansize_x` a `pansize_y`, které uchovávají relativně stále počáteční hodnoty velikosti okna. Toto je velice důležité při počítání poměru velikosti nového a starého okna, neboť pokud by se po každém zvětšení či zmenšení okna měla nová velikost stávat starou, docházelo by k deformacím zobrazovaných jevů kvůli velmi malým poměrům změn při postupném zvětšení či zmenšení. Nynější nastavení programu proto uchovává hodnoty velikosti okna, které byly určeny při spuštění programu a které se mění pouze při zadání nového trojúhelníku a při změnách polohy jednotlivých vrcholů trojúhelníku.

Samotné přepočítávání souřadnic je implementováno v metodě `MyPanel::MyResize()`, která si nejprve zjistí novou velikost okna pomocí metody `wxDC::GetSize()` a ze získaného výsledku následně pomocí metod `wxSize::GetWidth()` a `wxSize::GetHeight()` převede výšku a šířku do celočíselných hodnot. Poté vypočítá poměry změny ve směru osy `x` i osy `y` a menším z nich vynásobí hodnoty proměnných `fixed_bod_x[]` a `fixed_bod_y[]`, čímž dostane nové souřadnice vrcholů trojúhelníku. Důvodem pro vybírání menšího poměru je zachování tvaru trojúhelníku při zvětšení či zmenšení pouze jednoho rozměru okna. Proměnné označující poměry změny a pomocné proměnné pro nové souřadnice jsou typu `double` a až hotové výsledky jsou převedeny na celočíselné hodnoty z důvodu přesného převodu do nových souřadnic.

5.5 Geometrie

Protože program uchovává pouze souřadnice vrcholů trojúhelníku, je nutné při každé změně vykreslování, tedy při každém volání metody `MyPanel::OnPaint()`, spočítat všechny jevy, které je nutné vykreslit.

Pro vykreslování všech přímků program využívá metodu `wxWidgets wxDC::DrawLine()`, která jako parametry vyžaduje souřadnice počátečního a koncového bodu.

Oblouky jsou vykreslovány pomocí metody `wxWidgets wxDC::DrawArc()`. Její parametry jsou souřadnice počátečního bodu oblouku, přibližné souřadnice koncového bodu oblouku a střed kružnice, jejíž část oblouk tvoří. Přibližnost souřadnic koncového bodu spočívá v tom, že je možné zadat jakýkoli bod ležící na přímce procházející středem kružnice a bodem, kde je opravdu potřeba oblouk ukončit. Tato metoda vykresluje oblouk proti směru hodinových ručiček od počátečního bodu.

Vykreslení kružnic je zajištěno metodou `wxWidgets wxDC::DrawCircle()`, jejímiž parametry jsou souřadnice středu kružnice a její poloměr v pixelech, což je celočíselná hodnota.

Všechny souřadnice jsou zde zadávány jako dvojice celočíselných proměnných, neboť je většinou zapotřebí různých hodnot pro osu x a pro osu y .

5.5.1 Kolmice

U některých geometrických jevů je potřeba pracovat s přímkami kolnými na strany trojúhelníku. Aplikace to řeší tak, že odečtením souřadnic vrcholů u příslušné strany získá vektor přímky. Tento vektor poté převede na vektor jemu kolný a posune jej do bodu, jímž má kolmice procházet. Výsledkem jsou nové souřadnice počátečního a koncového bodu úsečky kolmé na příslušnou stranu trojúhelníku.

Následující rovnice ukazují, jak program při znalosti pouze souřadnic vrcholů trojúhelníku ABC zjišťuje souřadnice bodu D , jenž leží na přímce procházející bodem C , která je kolmá na stranu AB :

$$x_d = y_a - y_b + x_c \quad (5.1)$$

$$y_d = x_b - x_a + y_c \quad (5.2)$$

5.5.2 Průsečík přímků

Velmi často je také nutné zjistit průsečík dvou a více přímků, ať už jde o přímky pomocné, nebo o stranu trojúhelníku a pomocnou přímku. Pro tuto situaci program využívá metodu `MyPanel::MyIntersect()`. Tato metoda nemá žádné vstupní parametry, ale využívá globálních proměnných `st_bod_x[]` a `st_bod_y[]` pro počáteční a `ko_bod_x[]` a `ko_bod_y[]` pro koncové body úseček ležících na požadovaných přímkách. Tyto proměnné jsou pole celých čísel o velikosti tří položek. Metoda využívá parametrického vyjádření přímků podle rovnice:

$$ax + by + c = 0 \quad (5.3)$$

kde a a b jsou x -ová a y -ová souřadnice vektoru kolmého na vektor přímky a c je vypočítáno podle následující rovnice:

$$c = -(a \cdot st_bod_y + b \cdot st_bod_x) \quad (5.4)$$

Parametry a , b , c metoda zjistí pro obě přímky, jejichž průsečík potřebuje vypočítat, a jeho souřadnice poté určí podle rovnic:

$$x = \frac{a_1 \cdot c_2 - a_2 \cdot c_1}{a_2 \cdot b_1 - a_1 \cdot b_2} \quad (5.5)$$

$$y = \frac{b_2 \cdot c_1 - b_1 \cdot c_2}{a_2 \cdot b_1 - a_1 \cdot b_2} \quad (5.6)$$

5.5.3 Vzdálenost dvou bodů

Pro zjištění vzdálenosti dvou bodů je využívána metoda `MyPanel::MyDistance()`, která jako vstupní parametry přijímá souřadnice těchto dvou bodů a její návratová hodnota je získaná vzdálenost v podobě celého čísla. Samotnou vzdálenost metoda počítá z rozdílu x-ových a y-ových souřadnic bodů na principu Pythagorovy věty podle rovnice

$$vzdálenost = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5.7)$$

5.5.4 Střed úsečky

Velice často je pro vykreslení či spočítání některých jevů nutné znát střed nějaké úsečky, u níž známe pouze souřadnice počátečního a koncového bodu. Je to velmi jednoduchá operace, program nejprve použije ternární operátor pro zjištění, která ze souřadnic má menší hodnotu, a k této přičte polovinu absolutní hodnoty rozdílu obou souřadnic. Tento postup je stejný jak pro x-ovou, tak pro y-ovou osu.

5.6 Geometrické jevy

Všechny zobrazované jevy je nutné počítat a vykreslovat pro každou stranu nebo vrchol zvlášť, proto se u každého jevu objevuje cyklus `for` a výpočet je prováděn třikrát, ovšem pokaždé pro jinou stranu nebo vrchol. Kružnice opsaná a vepsaná jsou sice vykreslovány pouze jednou, ale pro jejich pomocné přímky, které je nutné znázornit pro všechny strany trojúhelníku, platí situace zmíněná výše.

Každý geometrický jev trojúhelníku, který je implementován v této aplikaci, má přiřazenou boolovskou proměnnou, jejíž hodnota určuje, zda se příslušný jev vykreslí, či nikoliv. Při spuštění aplikace jsou tyto proměnné nastaveny na hodnotu `false`, tudíž žádný z jevů se nevykreslí. Hodnoty těchto proměnných je možné změnit kliknutím na příslušný jev v menu *Geometrie* nebo na položku *Nový trojúhelník* v menu *Trojúhelník*, která všechny tyto „vykreslovací“ proměnné nastaví na `false`. Kliknutí na kteroukoli položku v menu *Trojúhelník – Příklady* nemá na hodnoty těchto proměnných vliv, tedy současné jevy zůstanou zobrazené, i když se změní typ trojúhelníku.

5.6.1 Vnitřní úhly

Počáteční bod oblouku vnitřního úhlu je vypočítán jako bod ležící na straně trojúhelníku v jedné čtvrtině této strany blíže k vrcholu, jehož vnitřní úhel se právě vykresluje. Následující rovnice znázorňuje výpočet počátečního bodu oblouku a je použit pro osu x i pro osu y :

$$pocatecniBod = soucasnyVrchol - \frac{soucasnyVrchol - vrcholPoSmeruHodin}{4} \quad (5.8)$$

Koncový bod oblouku je vypočítán stejně, ovšem s tím rozdílem, že místo vrcholu po směru hodinových ručiček je vzat vrchol proti směru hodinových ručiček. Středový bod oblouku je současný vrchol trojúhelníku.

5.6.2 Vnější úhly

Vykreslování vnějších úhlů je velmi podobné vnitřním, pouze je nutné dočasně si vytvořit pomocné přímký, které vzniknou protažením stran trojúhelníku. Počáteční a koncový bod pro metodu `wxDC::DrawLine()` jsou vypočítány podle následujících rovnic:

$$pomocnyBod = soucasnyVrchol + (soucasnyVrchol - vrcholPoSmeruHodin) \quad (5.9)$$

$$pomocnyBod = vrcholPoSmeruHodin + (vrcholPoSmeruHodin - soucasnyVrchol) \quad (5.10)$$

Poté program vykreslí pomocné přímký, které mají počáteční bod v současném vrcholu trojúhelníku a koncový bod ve vypočítaném pomocném bodě.

Počáteční a koncový bod oblouku označující vnější úhel jsou vypočítány stejným způsobem jako u vnitřních úhlů, ovšem vrcholy v rovnici jsou zaměněny dle potřeby aplikace.

5.6.3 Výšky

Pro výpočet souřadnic počátečního a koncového bodu výšek bylo nutné nejprve zjistit vektor kolmý na současnou stranu trojúhelníku, tento vektor poté posunout do bodu protilehlého k současné straně. Následně program vypočítá průsečík strany a takto vzniklé kolmice pomocí metody `MyPanel::MyIntersect()`. Poté je pomocí metody `wxDC::DrawLine()` vykreslena pomocná přímká z vrcholu trojúhelníku do vypočteného průsečíku a následně jsou zobrazeny výšky. Pomocné přímký jsou nutné v případě, že by trojúhelník byl tupouhlý, neboť jak jsem již uvedla výše, v tupouhlém trojúhelníku leží ortocentrum, tudíž i dvě z výšek mimo jeho obsah.

5.6.4 Těžnice

Pro vykreslení těžnic musí program spočítat pouze střed strany trojúhelníku a ten poté použít jako počáteční bod těžnice, jejíž konečný bod je shodný s protilehlým vrcholem trojúhelníku.

Po vykreslení všech těžnic program pomocí metody `MyPanel::MyIntersect()` vypočítá polohu těžiště a zobrazí jeho popis.

5.6.5 Střední příčky

Vykreslení středních příček program zajistí vypočítáním souřadnic středů dvou stran trojúhelníku a tyto body použije jako počáteční a koncový bod pro zobrazení požadovaných úseček pomocí metody `wxDC::DrawLine()`.

5.6.6 Kružnice opsaná

Pro základ kružnice opsané je nutné nejprve zjistit a vykreslit osy stran trojúhelníku, tj. kolmice procházející jejich středem, a vypočítat souřadnice jejich průsečíku pomocí metody

`MyPanel::MyIntersect()`. Poté program využitím metody `MyPanel::MyDistance()` vypočítá vzdálenost již zjištěného průsečíku a vrcholu A. Vzhledem k tomu, že vzdálenost od všech vrcholů je stejná, nezáleží na tom, který vrchol vezme v úvahu. Tuto vzdálenost poté použije jako poloměr kružnice opsané, kterou vykreslí pomocí `wxDC::DrawCircle()`.

Ovšem v případě os stran je nutné, aby byly vykresleny správným směrem jak v případě ostroúhlého trojúhelníku, kde střed kružnice opsané leží uvnitř, tak v případě trojúhelníku tupoúhlého, kde střed leží mimo trojúhelník. Proto program před jejich vypočtením za použití Pythagorovy věty zkontroluje, zda je příslušná strana naproti ostrému či tupému úhlu a v případě tupého úhlu změní znaménko pomocné proměnné, aby se kolmice vykreslovala opačným směrem, tedy mimo trojúhelník. Aplikace také po vypočtení středu znovu překreslí pomocné přímky, neboť u trojúhelníků s velkým tupým úhlem se ne vždy povede pomocné čáry vykreslit až ke středu kružnice.

Na pravouhlý trojúhelník, kde střed kružnice opsané leží ve středu přepony, program situaci netestuje.

5.6.7 Kružnice vepsaná

Pro kružnici vepsanou je potřeba zjistit osy úhlů trojúhelníku, což pouze ze souřadnic jeho vrcholů lze jen velmi obtížně. Aplikovala jsem tedy řešení, které osu úhlu zjistí sice nepřímou, ale správně a především jednoduše, tudíž je velmi efektivní.

Jak jsem uvedla v kapitole 2.3.5, v rovnoramenném trojúhelníku leží střed kružnice vepsané na těžnici vycházející z vrcholu, v němž se stýkají jeho ramena, tudíž tato těžnice je zároveň i osou úhlu. Ovšem ne všechny trojúhelníky jsou rovnoramenné, je tedy nutné, aby program vytvořil pomocný bod, který by reprezentoval vrchol trojúhelníku posunutý po přímce jeho strany tak, aby nový, pomocný trojúhelník byl rovnoramenný a aby aktuální vrchol ležel naproti imaginární straně. Toho je dosaženo přičtením k souřadnicím současného vrcholu vektor strany vynásobený délkou strany vedlejší a vydělený svou vlastní délkou. Strana se tedy buď prodlouží, nebo zkrátí tak, aby nový trojúhelník byl rovnoramenný. Poté aplikace vypočte střed nově vytvořené imaginární strany a vykreslí pomocnou přímku. Po vykreslení všech pomocných přímek je pomocí metody `MyPanel::MyIntersect()` vypočítán jejich průsečík, který označuje střed kružnice vepsané.

Kružnice vepsaná se dotýká všech stran trojúhelníku, tudíž s každou z nich má společný právě jeden bod, jehož zjištění je nutné pro následný výpočet poloměru kružnice. Tento bod dotyku leží na průsečíku strany a její kolmice procházející středem kružnice vepsané. Opět jsou použity metody `MyPanel::MyIntersect()` pro zjištění souřadnic bodu dotyku a `MyPanel::MyDistance()` pro výpočet poloměru kružnice. Ten poté, spolu se souřadnicemi středu, aplikace použije jako vstupní parametry pro metodu `wxDC::DrawCircle()`.

5.7 Uložení obrázku

Pro možnost uložení momentálně promítaného jevu do podoby bitmapového obrázku nebylo možné použít již vytvořený device context třídy `wxPaintDC`, který je vyžadován metodou `MyPanel::OnPaint()`. Jeho obsah je ovšem to, co je potřeba uložit, tudíž bylo nutné data zkopírovat do nově vytvořeného device contextu třídy `wxMemoryDC`, pomocí něhož je modifikován objekt třídy `wxBitmap`, který poskytuje základ pro uložení obrázku.

Kopírování mezi device contexty je prováděno metodou `wxWidgets wxDC::Blit()`.

Samotné uložení je implementováno v metodě `MyPanel::SaveImage()`, která využívá třídu `wxFileDialog` z `wxWidgets`, jež pomocí metody `wxBitmap::SaveFile()` uloží obrázek v bitmapovém formátu.

Kapitola 6

Spouštění a ovládání programu

Spustitelný soubor programu je *trojuhelnik.exe* a lze jej spustit pod operačním systémem Microsoft Windows XP a Vista. Spustitelnou verzi pro operační systém Linux se bohužel vytvořit nepodařilo. Pro zobrazení nápovědy využívá program externího souboru *manual.html*, který je nutné mít umístěn ve stejném adresáři jako spustitelný soubor. Pro správnou funkci nápovědy je též vhodné do tohoto adresáře umístit adresář *pictures*, v němž jsou potřebné vysvětlující obrazové soubory.

Ovládání programu je velmi intuitivní a je poměrně snadné se v něm zorientovat i bez pomoci nápovědy, která je ovšem v programu zabudována. Po spuštění se na obrazovce objeví okno s lištou menu. Pro zobrazení nápovědy k programu využijte v menu *Nápověda–O programu*. Ve výchozím internetovém prohlížeči se zobrazí webová stránka s podrobným návodem.

Vrcholy trojúhelníku lze zadat kliknutím myši na příslušná místa v šedé ploše programu nebo položkou v menu *Trojúhelník–Příklady*, kde vyberete konkrétní typ trojúhelníku.

Zobrazení jednotlivých jevů je možné pomocí lišty menu v záložce *Geometrie*, kde kliknutím na příslušný jev jej zobrazíte. Zobrazený jev bude označen zaškrtnutím a opětovným kliknutím je možné jej vypnout. Zde jsou také přítomné položky *Zobrazit vše* a *Skrýt vše*, které buď zobrazí nebo skryjí všechny implementované geometrické jevy.

Smazání starého a možnost zadání nového trojúhelníku provedete opět jednoduše přes menu, a to *Trojúhelník–Nový trojúhelník*.

Uložení obrázku najdete v *Trojúhelník–Uložit jako obrázek*. Objeví se dialogové okno, kde jen zadáte cestu, kam se obrázek uloží a napíšete jeho název. Obrázek bude uložen ve formátu bmp.

Program ukončíte buď standardním zavřením okna nebo přes menu kliknutím na *Trojúhelník–Ukončit program*.

Kapitola 7

Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci pro znázornění základních jevů v trojúhelníku, a tedy pro usnadnění výuky této látky, a tohoto cíle se podařilo dosáhnout.

Práce s frameworkem wxWidgets byla velkým přínosem při vývoji grafického uživatelského rozhraní, které má v takovémto typu aplikace stejnou důležitost, jako její „vnitřní logika“.

Díky správě projektů, kterou poskytovalo vývojové prostředí Code::Blocks, byl vývoj aplikace také velmi usnadněn.

Jako nejobtížnější na této práci se ukázalo nalezení učitele matematiky, který by byl ochotný poskytnout mi k této problematice konzultaci. Tu nakonec poskytl student Pedagogické fakulty Jihočeské univerzity, jehož některé návrhy jsem v aplikaci použila, jiné by mohly být implementovány dodatečně jako možnost rozšíření funkcí aplikace. Hotovou verzi programu též otestoval a vyjádřil se k ní takto:

Program Trojúhelník je vhodnou doplňkovou učební pomůckou pro výuku příslušné oblasti geometrie, především na základěškolské úrovni. Umožňuje zobrazit základní geometrické prvky trojúhelníků, byť zde není vysvětlena či vykreslena jejich konstrukce. Proto, a rovněž díky prvkům dynamické geometrie, se hodí především k deduktivním metodám výuky.

Ovládání programu je snadné, a mohou jej tedy používat učitelé i sami žáci. Jediným vážným nedostatkem je pozbytí charakteristické vlastnosti trojúhelníka, je-li dynamicky manipulováno s body předdefinovaných objektů - lze například změnit pravý úhel pravoúhlého trojúhelníka či poměry délek stran rovnostranného.

Program by neměl sloužit k vysvětlení jevů, ale rozhodně je praktický k jejich upevnění formou časově nenáročných hry motivující zejména obecné důkazové úlohy. V tomto ohledu se Trojúhelník může měřit s profesionálními programy typu Cabri Geometry, byť jeho pole působnosti je zatím velmi omezeno.

- Matt Bínů, 3. rok studia PF JU v Českých Budějovicích, obor M-F/SŠ

Mezi navrhovaná vylepšení patřilo možnost zobrazení pouze některých částí jevů, například pouze jednu těžnici nebo úhel, možnost zapínání a vypínání čtverečkového pozadí, soustavy Oxy a souřadnice bodů v ní, ilustrování základních vět – Pythagorovy, Euklidových, krokování standardní konstrukce trojúhelníku, či sbírka úloh v podobě předkreslených zadání, které by bylo možné tisknout. Nicméně pro implementaci většiny z těchto vylepšení,

snad kromě zobrazování čtverečkovaného pozadí, by bylo nutné začít psát program znovu úplně jinými způsoby a postupy, než jsem při vývoji této aplikace použila.

Komentované zdrojové soubory jsou dostupné na přiloženém disku CD, kde je též uložen spustitelný soubor programu, manuál k obsluze programu ve formátu pdf a tato technická zpráva ve formátu pdf a zdrojových kódech \LaTeX .

Literatura

- [1] Halouzka, A.: *Přehled učiva k maturitní zkoušce z matematiky*. Fortuna, 2002, iISBN 80-7168-808-8.
- [2] Pražáková, V.; Čapková, M.: *Základy Psychologie a biologické základy vývoje [online]*. 2008-10-30 [cit. 2009-04-17].
- [3] Smart, J.; Hock, K.; Csomor, S.: *Cross-Platform GUI Programming with wxWidgets [online]*. Prentice-Hall, 2006, iISBN 0-13-147381-6.
- [4] Smart, J.; Roebling, R.; Zeitlin, V.; aj.: wxWidgets 2.8.10: A portable C++ and Python GUI toolkit [online]. <http://docs.wxwidgets.org/stable/>, 2009-02 [cit. 2009-05-08].
- [5] WWW stránky: Code::Blocks [online]. <http://www.codeblocks.org>, 2009-03-19 [cit. 2009-05-08].

Dodatek A

Obsah CD

Příložené CD obsahuje

- Zdrojové kódy programu
- Soubor programu spustitelný na MS Windows
- Manuál k programu ve formátu .pdf
- Technickou zprávu BP ve formátu .pdf
- Zdrojové soubory technické zprávy BP pro L^AT_EX