

Vizualizační nástroj Asymptote

T. Brzák¹, P. Koblre², M. M. Machová³, A. Svoboda⁴, J. Tomsa⁵

^{1,2}Gymnázium Děčín, ³Gymnázium Svitavy, ⁴Mendelovo Gymnázium Opava, ⁵Gymnázium Turnov

brzak.t@seznam.cz, peta.koblre@gmail.com, MachovaMichaela@gmail.com, janik.tomsa@gmail.com,
svoboda.adams@gmail.com

Abstrakt

Programovací jazyk Asymptote umožňuje vytvářet vektorovou grafiku. Nabízí široké spektrum možností využití, od vytváření grafů po 3D grafiku. Mnohé z jeho schopností jsme si v rámci miniprojektu mohli vyzkoušet.

1 Úvod

Softwarový nástroj Asymptote generuje kvalitní vektorovou 2D či 3D grafiku s použitím převážně ve vědeckých publikacích či prezentacích. Jeho hlavní výhodou je, že je také programovacím jazykem a ne pouhým grafickým editorem, čemuž vděčí za svou mimořádnou flexibilitu a rozsah uplatnění. Je vhodný pro tvorbu diagramů, grafů, vizualizaci dat, atd. Další výhodou je, že je podporován všemi hlavními operačními systémy (UNIX, MacOS X, Microsoft Windows). Důležité při jeho používání je dobře vyvinuté logické myšlení a důvtip pro práci s příkazy.

2 Přehled možností Asymptote

Programovací jazyk Asymptote vychází svojí syntaxí z programovacího jazyka C, avšak obsahuje mnohá rozšíření navržená s ohledem na jeho zaměření. Příkladem jsou operátory `--`, `^^`, `..` pro spojování geometrických objektů nebo speciální mechanismus předávání parametrů při volání funkcí, který usnadňuje strukturování zdrojového kódu. Obsahuje datové typy pro geometrické objekty, jako je například bod, křivka, atd.

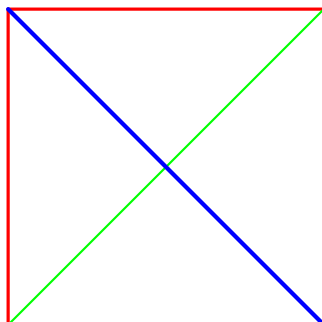
Mezi jeho podstatné vlastnosti patří

- podpora 2D a 3D zobrazování,
- spolupráce s typografickým systémem Latex při vykreslování textu,
- intuitivní koncept kreslení perem (kreslící nástroj s danými vlastnostmi se pohybuje po dané trajektorii),
- snadné geometrické transformace libovolných objektů (rotace, škálování, posunutí),
- řada dalších specifických funkcí v aktivovatelných modulech.

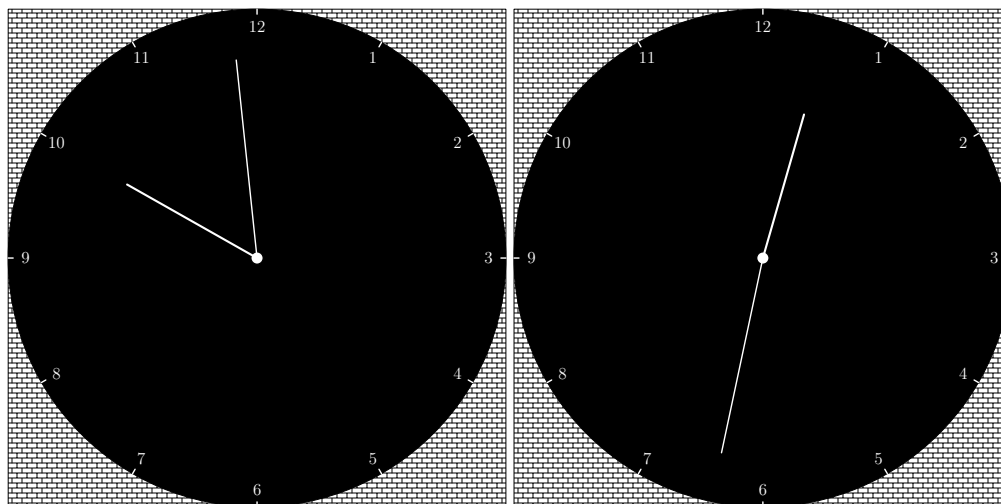
Výstupními formáty jsou PDF, PostScript a SVG, ale pomocí nástroje *convert* je možné získat výstup v celé řadě dalších formátů. Pro zobrazení 3D grafiky obsahuje vlastní prohlížeč s podporou OpenGL.

V následujícím příkladu je uveden kód, který generuje kresbu na obrázku 1.

```
size(300,300);
path ctverec=(0,0)--(1,0)--(1,1)--(0,1)--cycle;
pen p=red+3;
draw(ctverec,p);
draw((0,0)--(1,1),green+2);
draw((0,1)--(1,0),blue+4);
```



Obrázek 1: Příklad jednoduché kresby



Obrázek 2: Obrázek hodin pro čas 21:59 (vlevo) a pro čas 12:32 (vpravo)

3 Výsledky

Následující sekce prezentují výsledky jednotlivých členů týmu.

3.1 Zobrazení ciferníku hodin (Tomáš Brzák)

Dostal jsem za úkol udělat funkční analogové hodiny, které by ukazovaly čas podle zadaných parametrů (hodiny, minuty). Pokud je tedy čas ve formátu $h:m$, pak úhel α , který svírá minutová ručička se svislicí vypočítáme podle

$$\alpha = m * \pi / 30$$

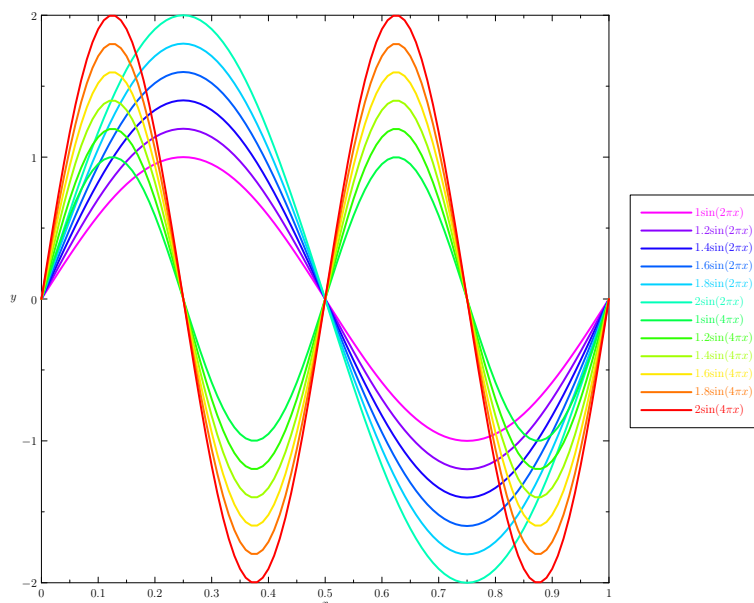
Úhel β , který svírá hodinová ručička se svislicí, potom

$$\beta = h * \pi / 6 + \pi / 360 * m$$

Na konec jsem přidal pozadí připomínající zeď a "pohrál" si s barvami a velikostmi ručiček a čísel. Výsledek je vidět na obrázku 2

3.2 Automaticky generované grafy (Michaela Mia Machová)

Během své práce v Asymptote, jsem se seznámila s modulem graph, který slouží k vykreslování grafů matematických funkcí nebo dat. Ve výsledku se mi podařilo naprogramovat graf pro funkci $y = A \sin(2k\pi x)$ pro různé hodnoty parametru A a k , které se mění v cyklu, jak si můžeme všimnout na obrázku 3. Pro větší přehlednost jsem ještě v závěru využila barevného odlišení jednotlivých křivek a přidala automaticky se měnící legendu grafu.



Obrázek 3: Automaticky generované grafy funkcí

3.3 3D model (Adam Svoboda)

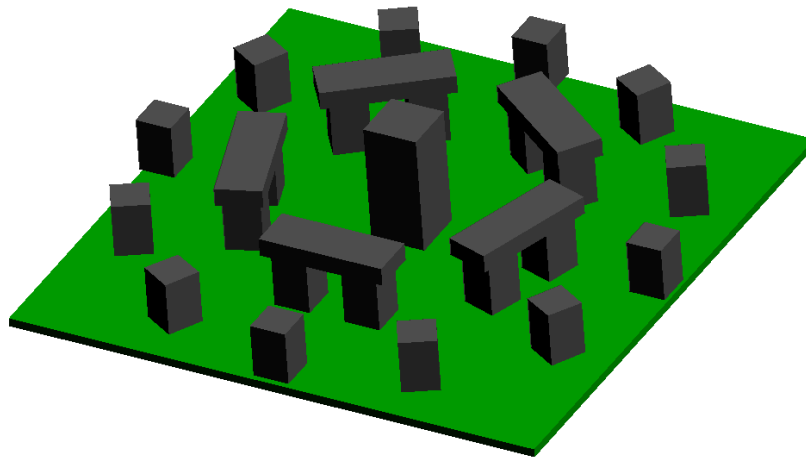
Při vytvoření 3D scény jsme zvolili téma Stonehenge, na kterém se dá ukázat základní schopnosti Asymptote ohledně posunování, nastavování velikostí a rotace celých objektů. Prvotní problém byl jak se dá snadno vytvořit krychle. Rozhodli jsme se ji sestavit ze 6 povrchů, avšak nastal problém s vykreslováním (zadní strana překryla strany vpředu). Tento problém vyřešil program Adobe reader, který toto dokázal hravě překonat. Po vytvoření krychle už zbyla jen snadná práce. Nastavením velikostí (scale) jsme udělali oblouk a udělali z něj proceduru. Tento oblouk jsme zrotovali kolem osy 5 krát. Podobným způsobem jsme zhotovili jednoduchý menhir, také z něj udělali funkci a zrotovali ve větší vzdálenosti 12 krát. Na závěr jsme dali největší menhir do středu osy a vytvořili zem (úprava krychle pomocí scale). Tímto způsobem jsme vytvořili velice jednoduchý Stonehenge (viz obrázek 4).

3.4 Hvězda (Petr Koblre)

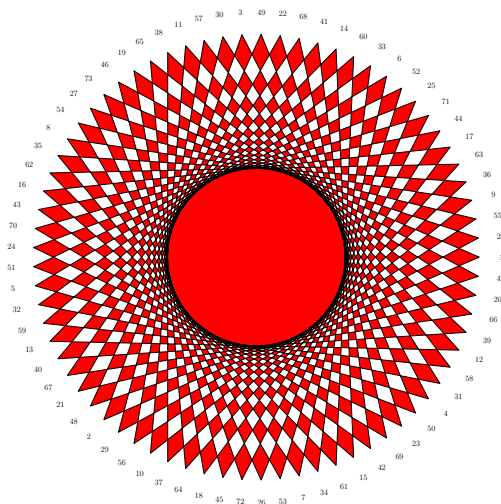
Pro vykreslení 73($n=73$) cípé hvězdy jsem použil proměnnou i která se v každém cyklu zvětšovala o 46 a dělila výsledným počtem cípů hvězdy a zbytek bylo nové výchozí i pro další cyklus: $i = (i + 46) \bmod n$. Výsledná hodnota i reprezentovala pozice vrcholů rovnoměrně rozdělených na jednotkové kružnici. Hvězda vznikala postupným spojováním pomocí rovných čar a ke každému vrcholu jsem přidal číslo(1-73) označující pořadí v jakém byly vrcholy spojovány, jak je vidět na obrázku 5. Příklad pro hvězdu s 47 cípů, i zvětšující se o 16 je na obrázku 6.

3.5 Fraktální množina kruhů (Jan Tomsa)

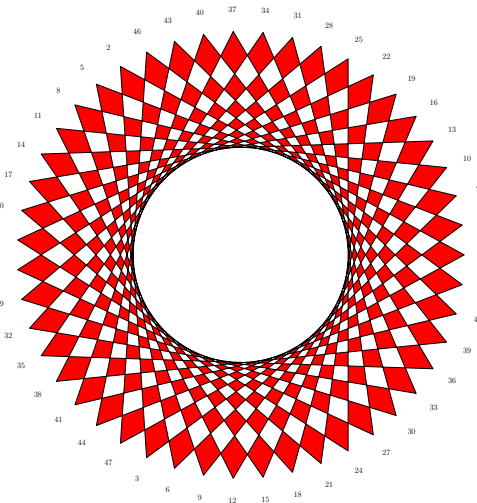
Cílem mého snažení bylo vytvořit nějaký efektní obrazec, který by ihned zaujal diváka. Vždy se mi líbily fraktální množiny. Zkusil jsem tedy něco podobného vytvořit. Jako základ se vzal běžný kruh. Na ten program umístil ze všech stran další kruhy polovičního poloměru (a odlišné barvy). A na každý z těchto nových kruhů se umístí zase další kruhy polovičního poloměru (oproti předchůdci). Celý algoritmus tvoření je tedy založen na rekurzivním volání procedury, která na daný kruh "nalepí" již zmíněné menší kruhy. Rekurze pracuje na tom principu, že funkce/procedura volá sama sebe. Volá se tedy stále stejná procedura, ovšem s pokaždé s jinými parametry závislými na parametrech předchozího kruhu. Program pak systematicky "prochází" celý obrazec až do dané úrovně (hloubky) a postupně vykresluje patřičně velké kruhy na správné místo. Výsledek mého snažení je vidět na obrázku 7.



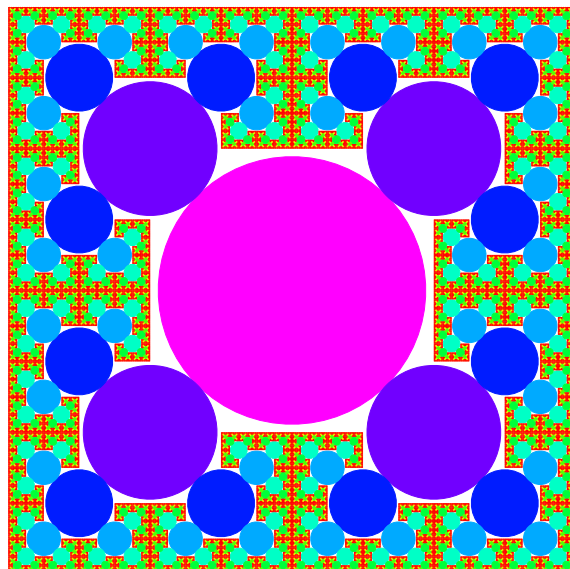
Obrázek 4: Stonehenge



Obrázek 5: 73 cípá hvězda



Obrázek 6: 47 cípá hvězda



Obrázek 7: Fraktální množina kruhů

4 Závěr

V rámci miniprojektu jsme se během dvou dnů seznámili s programovacím jazykem Asymptote. Program je volně stažitelný, jazyk není těžký na naučení a výsledkem je kvalitní grafika.

Poděkování: Rádi bychom poděkovali všem, kteří se podíleli na přípravě Týdne vědy na FJFI ČVUT a rovněž vedoucímu našeho miniprojektu Ing. Pavlu Strachotovi.

Reference

- [1] P. Strachota: *Počítačová grafika* (přednášky). FJFI ČVUT, Praha, 2011.
- [2] Žára, Beneš, Sochor, Felkel: *Moderní počítačová grafika*. Computer Press, 2005. ISBN: 80-251-0454-0
- [3] Asymptote documentation. <http://asymptote.sourceforge.net/>