UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KATEDRA INFORMATIKY

Animácia fraktálov pri spojitej zmene exponentu

(BAKALÁRSKA PRÁCA)

Ján Kosa

Študijný odbor 9.2.1 Informatika

Školiteľ: RNDr. Martin Samuelčík

BRATISLAVA 2008

Čestne prehlasujem, že som túto prácu robil
samostatne a uviedol som všetku použitú literatúru.

Chcel by som poďakovať vedúcemu bakalárskej práce,
Martinovi Samuelčíkovi, za venovaný čas a cenné rady,
odborné vedenie a pripomienky pri tvorbe tejto práce.
Tiež by som rád poďakoval Jurajovi Cangárovi za ochotu
a ústretovosť pri návrhu a testovaní aplikácie.
Moja vďaka patrí aj rodine, priateľom a všetkým, ktorí ma
podporovali pri tvorbe bakalárskej práce a počas celého štúdia.

Abstrakt

Práca popisuje vývoj a funkcionalitu programu na vytváranie animácií Mandelbrotovej množiny pri spojitej zmene exponentu. Funkcionalita sa neobmedzuje len na uvedenú animáciu, ale obsahuje aj animáciu pohybu po fraktále a animáciu Júliovej množiny. Všetky vytvorené animácie je možné uložiť do avi súboru. Taktiež je možné ukladať vykreslené fraktály do bmp súboru. Program je určený pre širokú verejnosť a pri jeho vytváraní bol kladený dôraz na jednoduchosť ovládania a prívetivosť k užívateľovi. Pri tvorbe programu som sa snažil implementovať vlastné riešenia vzniknutých problémov.

Kľúčové slová: fraktál, exponent, animácia, Mandelbrotova množina, Júliove množiny

The present thesis describes the development and the functionality of an application for creation of Mandelbrot set animations with continuously changing exponent.

Application functionality is not limited to the Mandelbrot set animation - the application is also capable of animating movement along a fractal and a Julia set. Each animation can be saved as avi file. Rendered fractals can be saved into bmp files. The application was designed for use by the broader public and developed with simplicity of use and user friendliness in mind. The present writer also attempted to implement his own solutions to the problems involved.

Keywords: fractal, exponent, animation, Mandelbrot set, Julia set

Obsah:

1. Úvod:

Už od svojho objavenia vzbudzovali fraktály úžas a pozornosť laickej ako aj odbornej verejnosti. Veľmi rýchlo prestali byť len záležitosťou matematikov a fyzikov a dostali sa aj do umenia a iných oblastí života

Hlavnou náplňou tejto bakalárskej práce sú fraktály a najmä ich animácia. Na jej realizáciu slúži program z názvom AFractal (z anglického slovného spojenia Animated Fractal). Pomocou neho sa dá zobraziť Mandelbrotova a Júliova množina, je možné priblížiť si ich ľubovoľnú časť a vytvoriť animáciu zmeny tvaru fraktálu pri zmene exponentu v zadanom rozsahu, animáciu pohybu po fraktále a animáciu zmeny Júliovej množiny.

Tvar Mandelbrotovej množiny závisí od veľkosti exponentu použitého pri jej výpočte a výsledný obraz sa zaujímavým spôsobom mení v závislosti od jeho hodnoty. Vyššie celočíselne exponenty v Mandelbrotovej množine zvyšujú počet osí symetrie. Je zaujímavé sledovať ako presne prebieha zmena od jedného celočíselného exponentu k druhému. Hlavnou úlohou mojej bakalárskej práce bolo vizualizovať práve túto zmenu čo najpresnejšie pomocou animácie, v ktorej sa v každom snímku mení exponent o malú hodnotu tak, aby celá animácia budila dojem plynulosti.

2. Fraktály

V bežnej reči sa za fraktál považuje členitý geometrický objekt, ktorého menšie časti vykazujú známky sebepodobnosti [1][2]. Medzi hlavné vlastnosti fraktálu ako geometrického objektu patrí najmä sebepodobnosť, jednoduchá a rekurzívna definícia, jemnosť aj pri veľkých priblíženiach a nemožnosť ho opísať v tradičnom euklidovskom geometrickom jazyku.

2.1 História

Fraktály sa v prírode vyskytujú od nepamäti. Pohoria, mraky, pobrežia, blesky ale aj snehové vločky sú všetko fraktálne útvary. Taktiež útvary vzniknuté kryštalizáciou vykazujú podobné vlastnosti. Svoje zastúpenie nemajú len v neživej prírode, ale hojné sú aj medzi rastlinami. Asi najlepším príkladom je istý druh karfiolu **[Obr. 1]**, no patria sem aj ulity, papradie alebo zhluky baktérií [3].

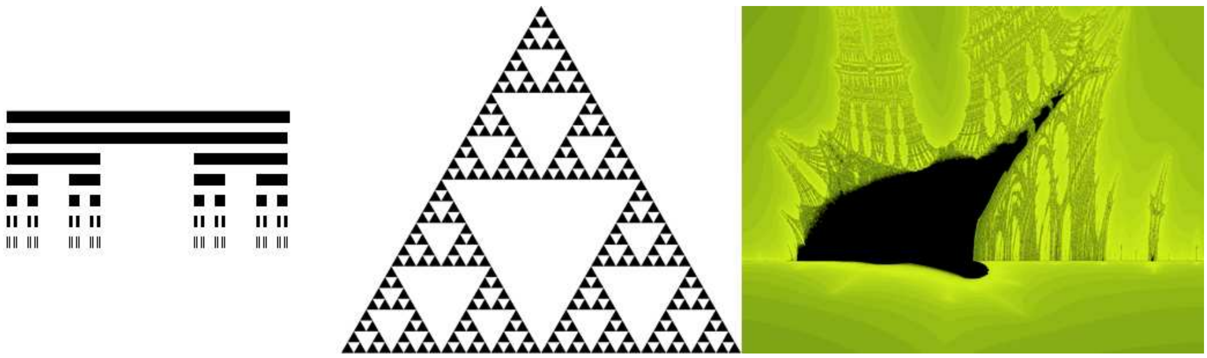
[Obr. 1] Príklady fraktálov v prírode: (zľava) karfiol, snehová vločka, ulita

Výskyt fraktálnych útvarov v prírode bol známy už dlhšiu dobu, chýbal však ich popis a zadefinovanie. To sa do istej miery ako prvému podarilo francúzskemu matematikovi s poľským pôvodom B. Mandelbrotovi pri skúmaní Júliových množín. Zaviedol do matematiky pojem fraktál, jednu z jeho viacerých definícií. Prvé Mandelbrotove stretnutie s fraktálmi bolo počas jeho práce v IBM, kde skúmal problémy pri prenose binárnej informácie medzi dvomi počítačmi. Všimol si že výskyt chýb vykazuje znaky pravidelnosti a nápadne sa podobali na Cantorovu množinu **[Obr 2.]**. Stačilo preposlať informáciu v oblastiach kde bol predpokladaný výskyt chyby a problém bol vyriešený. Okrem toho Mandelbrot skúmal takmer zabudnuté práce francúzskeho matematika Gastona Júliu, ktorý matematicky popisal fraktály, ktoré dnes nesú jeho meno. Mandelbrot sa snažil nájsť spôsob akým Júliove množiny zovšeobecniť a zostrojil akýsi katalóg Júliových množín, Mandelbrotovu množinu, ktorej každému jej bodu zodpovedá iná Júliova množina.

2.1 Generovanie fraktálov

Podľa spôsobu generovania fraktálov ich rozdeľujeme do 3 hlavných skupín:

- **Escape-time fraktály** sú definované rekurentne pre každý bod komplexnej roviny podľa toho, či sa bod pri iterovaní blíži k nekonečnu alebo nie. Patria sem napríklad Mandelbrotova množina, Júliova množina, „burning ship“ fraktál **[Obr 2.]** atď. Samotný fraktál je potom vytvorený tak, že danému bodu sa priradí farba podľa toho, v ktorej iterácii prestal spĺňať podmienku vo výpočte.
- **IFS** (Iterated Function System) sú generované pomocou pravidiel, patrí sem Kochova vločka, Peanova krivka, Sierpinskeho trojuholník, Cantorova množina **[Obr 2.]**
- **Náhodné fraktály** sú generované stochasticky, nie deterministicky, napr. Brownov pohyb, fraktálne pohoria



[Obr 2.] Vľavo: Sierpinskeho trojuholník, vpravo: výrez z fraktálu „Burning ship“

2.2 Podobnosť fraktálov

Fraktály rozdeľujeme podľa podobnosti na:

- presná sebepodobnosť, najsilnejšia podobnosť, vyznačujú sa ňou fraktály vytvorené IFS
- čiastočná sebepodobnosť, jednotlivé časti fraktálu vykazujú znaky podobnosti s menšími odchýlkami, patria sem najmä fraktály zadané rekurentnou funkciou
- štatistická sebepodobnosť, najslabší typ podobnosti. Fraktál má isté štatistické vlastnosti ktoré sa zachovávajú vo všetkých jeho častiach

2.3 Júliove množiny

Júliove množiny skúmali dvaja matematici, Gaston Julia a Pierre Fatou, počas študovania iteratívnych funkcií. Ich práca však nevzbudila širší záujem medzi matematikmi a upadla do zabudnutia na dlhé roky.

Výpočet Júliovej množiny je prekvapivo jednoduchý. Zoberieme si ľubovoľný komplexné číslo **c**. Teraz každý bod komplexnej roviny **z** opakovane umocníme a pripočítame **c** ($z_{n+1} = z_n^2 + c$). Takto vzniknutý číselný orbit môže mať jednu z troch nasledujúcich vlastností:

1. konverguje k nejakému komplexnému číslu
2. vyskytne sa v ňom cyklus, teda $z_n = z_m$ pre $m \neq n$
3. čísla budú chaoticky divergovať

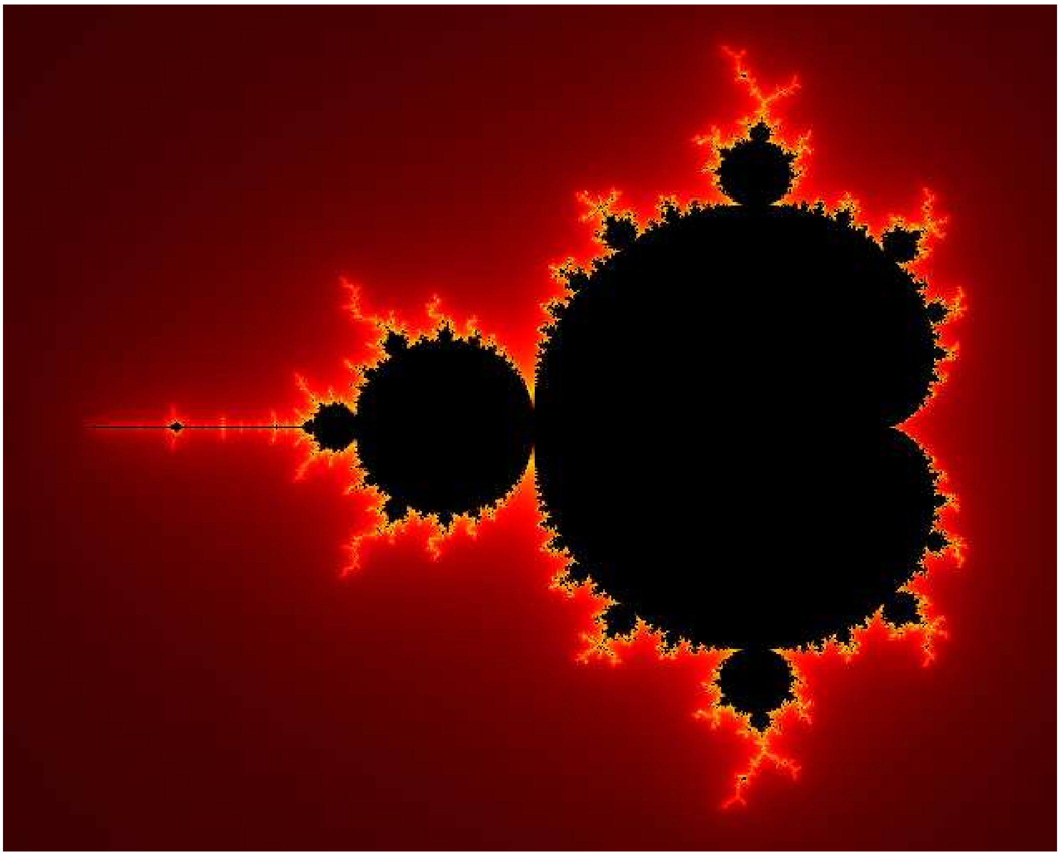
Aby bol výpočet realizovateľný, počítame len niekoľko prvých iterácií. Ak po poslednej iterácii je vzdialenosť bodu od stredu súradnicovej sústavy menšia ako 2, bod patrí do množiny. Keď všetky takto získané body vykreslíme podľa toho či patria alebo nepatria do množiny získame obrázky, vďaka ktorým sa Júliove množiny preslávili **[Obr 3.]**.

[Obr 3.] Rôzne Júliove množiny

2.4 Mandelbrotova množina

Júliove množiny po smrti Gastona Juliu upadli do zapadnutia, až do ich znovuobjavenia Benoitom Mandelbrotom. Juliova práca ho zaujala a pokúsil sa zovšeobecniť Júliove množiny. To sa mu napokon podarilo pomocou inej množiny v komplexnej rovine, ktorá bola po ňom pomenovaná **[Obr 4.]**. Každý bod komplexnej roviny popisuje práve jednu Júliovu množinu. V prípade, že bod v komplexnej rovine patri do Mandelbrotovej množiny, Júliova množina, ktorú určuje, bude spojitá. Mandelbrotova množina je teda akýsi katalóg Júliových množín

Výpočet Mandelbrotovej množiny je veľmi podobný výpočtu Júliovej množiny. Každý bod komplexnej množiny **c** opakovane umocníme podľa rovnakého vzorca ako v prípade Júliovej množiny ($z_{n+1} = z_n^2 + c$), avšak s tým rozdielom $z_0 = 0$. Opakovaním uvedeného postupu získame číselný rad, ktorý buď diverguje, alebo všetky jeho členy ostávajú vo vzdialenosti 2 od stredu súradnicovej sústavy. V prípade, že číselný rad diverguje, bod **c** nepatri do Mandelbrotovej množiny, v opačnom prípade patrí.



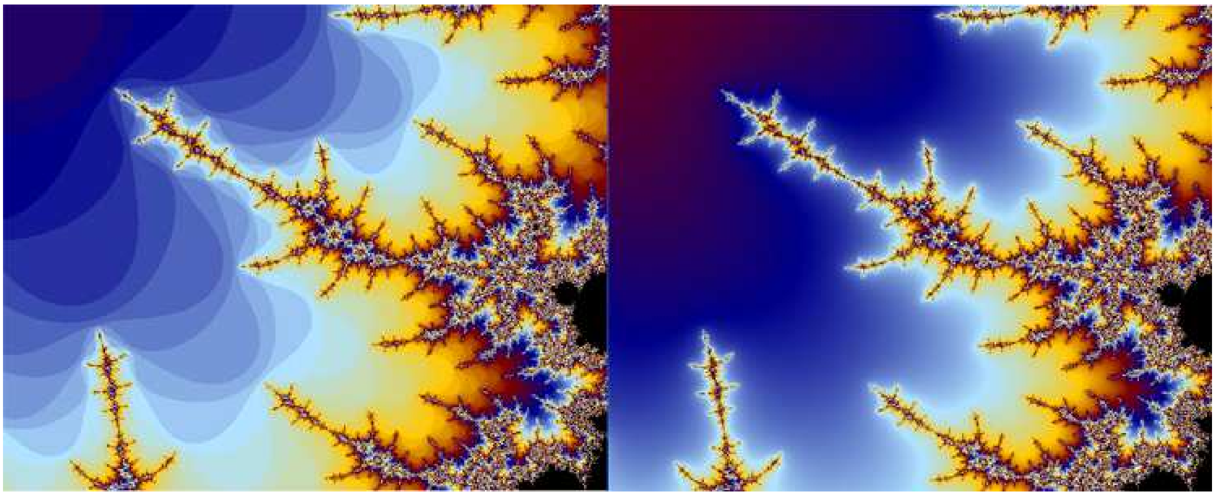
[Obr 4.] Mandelbrotova množina

2.5 Farbenie

Algoritmy, ktoré určia, či bod v komplexnej rovine patri do Mandelbrotovej alebo Júliovej množiny, nám umožnia vytvoriť čiernobiele obrázky. Takéto obrázky sú dostatočné na zobrazenie tvaru množiny, neponúkajú však taký estetický zážitok ako viacfarebné reprezentácie. Existuje viacero spôsobov ako priradiť jednotlivým bodom farbu, bližšie si ukážeme dva, ktoré sú použité v programe AFractal **[Obr 5.]**.

Prvý spôsob vyprodukuje diskrétne zafarbenie. Dodatočnú informáciu pre určenie farby získame sledovaním po koľkej iterácii číslo **z** presiahne vzdialenosť 2 od stredu súradnicovej sústavy. Podľa takto získaného čísla vyberieme farbu z palety a priradíme ju skúmanému bodu. Zvolením vhodnej palety a správnym spôsobom priradenia čísla iterácie ku farbe sa dajú docieliť veľmi zaujímavé zafarbenia Mandelbrotovej a Júliových množín.

Druhy spôsob zafarbenia použitý v programe AFractal je spojité farbenie. Podobne ako v predchádzajúcom prípade sledujeme kedy číslo **z** presiahne vzdialenosť 2 od stredu súradnicovej sústavy. Na rozdiel od diskrétneho zafarbenia necháme algoritmus ešte niekoľkokrát iterovať (konkrétne 5-krát). Následne vypočítame vzdialenosť bodu **z** od stredu súradnicovej sústavy a označíme ho **d**. Index do farebnej palety vypočítame pomocou vzorca `index = i + 1 - log(log(d))/log(m_exp)`.



[Obr 5.] Porovnanie farbiacich metód. Vľavo diskrétne farbenie, vpravo spojité

2.5 Animácie

Animácia je tvorená určitým počtom snímkov. Každý snímok pozostáva z jedného vypočítaného fraktálu. Niektoré nastavenia fraktálu sa v priebehu animácie menia a tým sa mení aj jeho tvar alebo pozícia. Spoločné vlastnosti všetkých animácii sú celkový počet snímkov a počet snímkov za sekundu. Prednastavenou hodnotou počtu snímkov za sekundu je 24. Výslednú dĺžku animácie v sekundách získame vydelením celkového počtu snímkov počtom snímkov za sekundu.

3.2.1 Výpočet neceločíselného exponentu

Pri výpočte animácie spojitej zmeny exponentu je často potrebné umocňovať komplexné číslo na neceločíselný exponent. Takýto výpočet je časovo oveľa náročnejší a výpočet fraktálu s takýmto exponentom trvá výrazne dlhšie ako výpočet fraktálu s celočíselným exponentom.

Pri umocňovaní na neceločíselný exponent musíme najprv komplexné číslo previesť do polárnej formy z = r.e$^{i\varphi}$. r je vzdialenosť bodu od stredu komplexnej roviny, φ je uhol medzi spojnicou komplexného bodu so stredom roviny a reálnou osou. Vzdialenosť r vypočítame ako absolútnu hodnotu komplexného čísla z, teda $r = \sqrt{x.x + y.y}$ kde z = x + y.i. Uhol φ vypočítame ako $\varphi = tg^{-1}\left(\frac{y}{x}\right)$. Komplexné číslo v tomto formáte sa umocňuje veľmi jednoducho. Číslo r umocníme na daný exponent a uhol φ týmto číslom vynásobíme. Nové komplexné číslo je teraz už len potrebné previezť naspäť do karteziánskej formy pomocou vzorca $z = r.\cos(\varphi) + r.\sin(\varphi).i$.

Vzhľadom na to, že pri prevádzaní medzi jednotlivými reprezentáciami sa používajú trigonometrické funkcie a taktiež samotné umocňovanie reálneho čísla na reálny exponent je časovo náročnejšie ako normálne násobenie použité pri umocňovaní komplexného čísla na celočíselný exponent sa tento spôsob používa len pri neceločíselných exponentoch.

3.2.2 NURBS krivka

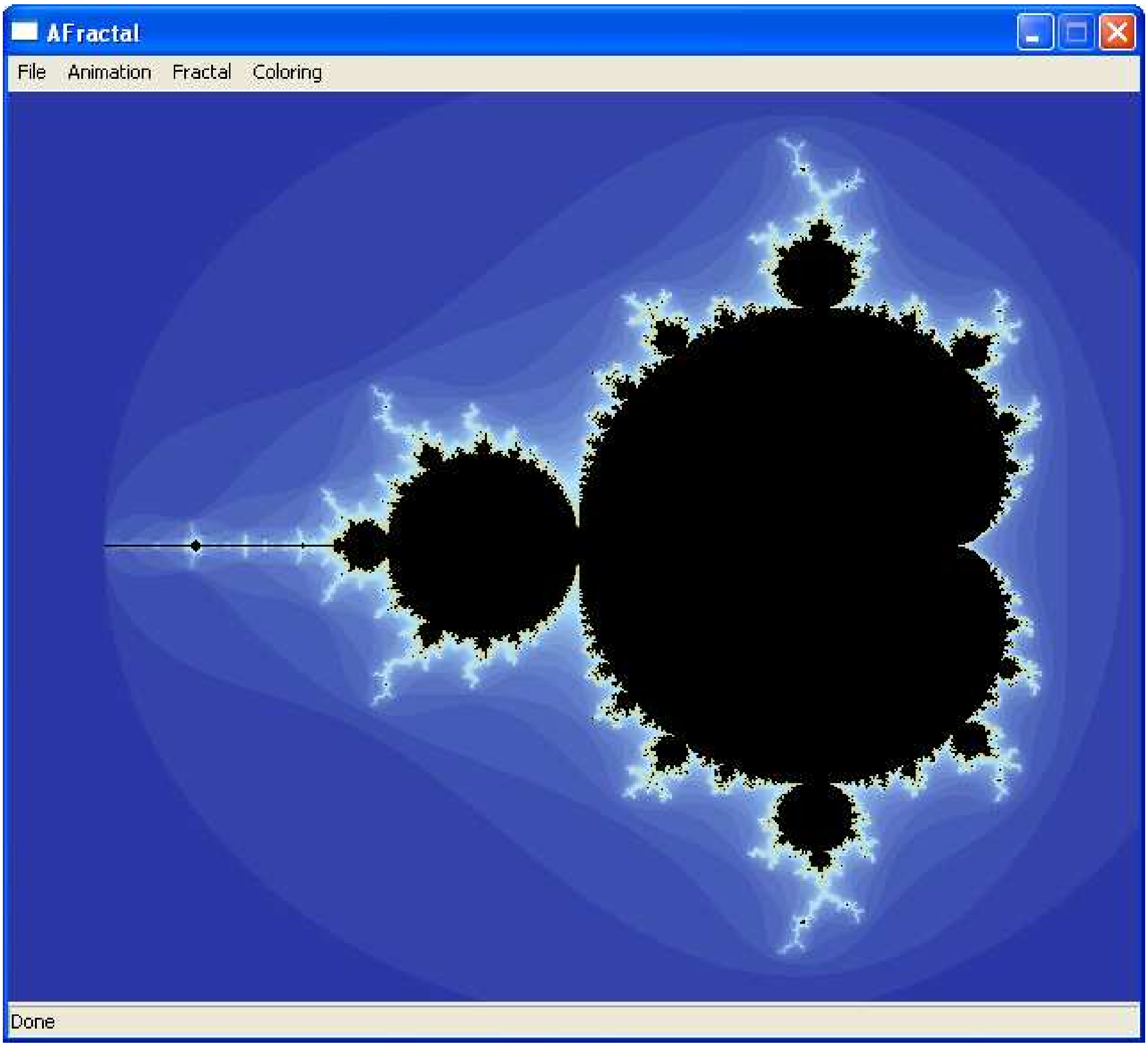
Program AFractal používa NURBS krivky pri animácii Júliovej množiny na reprezentáciu krivky, po ktorej sa pohybuje bod určujúci Júliovu množinu. Kvôli jednoduchosti ovládania nie je možné meniť uzlový vektor a ani váhy jednotlivých vrcholov. Ukázalo sa však, že pre potreby aplikácie je poskytnuté riešenie dostatočné. Hodnoty uzlového vektora sú 0, 1, 2, ... Váhy všetkých vrcholov sú 1.

Krivka si pamätá koľko snímkov ma animácia obsahovať, koľko riadiacich vrcholov krivka obsahuje a taktiež koľko snímkov má byť medzi jednotlivými riadiacimi bodmi. Pomocou týchto údajov je možné jednoducho získať hodnotu požadovaného komplexného čísla len podľa poradového čísla snímku v animácii. Všetky potrebné prepočty vykoná samotná implementácia krivky.

3. AFractal – opis aplikácie

3.1 Užívateľské rozhranie

Aplikácia bola navrhnutá a naprogramovaná s ohľadom na jednoduchosť a prívetivosť ovládania k užívateľovi. Hlavné okno aplikácie pozostáva z menu, grafickej plochy a stavového riadku. Menu položky sú rozdelené do logických kategórií: práca s animáciou fraktálov, nastavenie fraktálu a farbenie.



[Obr 6.] Hlavne okno aplikácie AFractal

Grafická plocha umožňuje skúmanie samotného fraktálu. Ovládanie je jednoduché. Klávesami + a – sa zvyšuje a znižuje počet iterácií, ktoré sa vykonajú pred tým ako sa prehlási, že bod patri do množiny. Zoomovanie je ovládané pomocou myši. Ľavým tlačidlom vyberieme systémom drag&drop oblasť, ktorú chceme priblížiť. Pravým tlačidlom myši sa dostaneme na predchádzajúci stupeň priblíženia.

V spodnej časti aplikačného okna sa nachádza stavový riadok, v ktorom je uvedený stav výpočtu aktuálneho fraktálu. V prípade, že sa počíta novy stav fraktálu (pri priblížení, prípadne zmene poctu iterácií), v stavovom riadku je uvedené koľko percent je vypočítaných. Stav sa obnovuje každých 10%.

3.2 Farbenie

Jednou z najdôležitejších funkcií programu AFractal je farbenie fraktálu. Užívateľovi je k dispozícii viacero farebných paliet, z ktorých si môže ľubovoľne vyberať. Taktiež má možnosť vytvoriť si vlastnú paletu pre oba typy fraktálu.

3.2.1 Farebná paleta

Farebná paleta pozostáva z jednotlivých farieb, ktoré sú priraďované jednotlivým pixlom. Všetky farby sú uložené v spoločnom poli po zložkách (červená, zelená, modrá). Farby palety sú usporiadane podľa počtu iterácií ku ktorým budú priradené. Prvá farba bude priradená k pixlom s najmenším počtom iterácií. Posledne 3 prvky poľa určujú farbu, ktorá bude priradená pixlom patriacim množine. Pre dosiahnutie čo najlepšieho výsledného efektu by prvá a predposledná farba mali byť rovnaké.

Existujú dva spôsoby ako aplikovať paletu na fraktál. Prvý spôsob roztiahne farebnú paletu na celý rozsah iterácií. Prvá farba je namapovaná na body s najmenším počtom iterácií, posledná na body tesne pri množine. V prípade, že posledná farba je zhodná s farbou množiny dosiahneme plynulý prechod medzi množinou a jej okolím. Týmto sa dá dosiahnuť veľmi zaujímavých výsledkov najmä pri použití na Júliovej množine.

Druhy spôsob vytvorí opakujúce sa farebné pásy. Každej farbe palety je priradený celočíselný interval. V prípade, že počet iterácií **i** patri do intervalu danej farby, farba priradená pixlu je výsledkom lineárnej interpolácie medzi farbou patriacou intervalu a nasledujúcou farbou. V prípade, že **i** je väčšie ako horná hranica intervalu predposlednej farby, od **i** odpočítame hodnotu hornej hranice intervalu predposlednej farby. Docielime tým cyklické opakovanie farieb. Aby farbenie bolo rovnako kvalitné aj pri väčšom počte iterácií, je potrebné postupne zväčšovať rozsahy intervalov farieb.

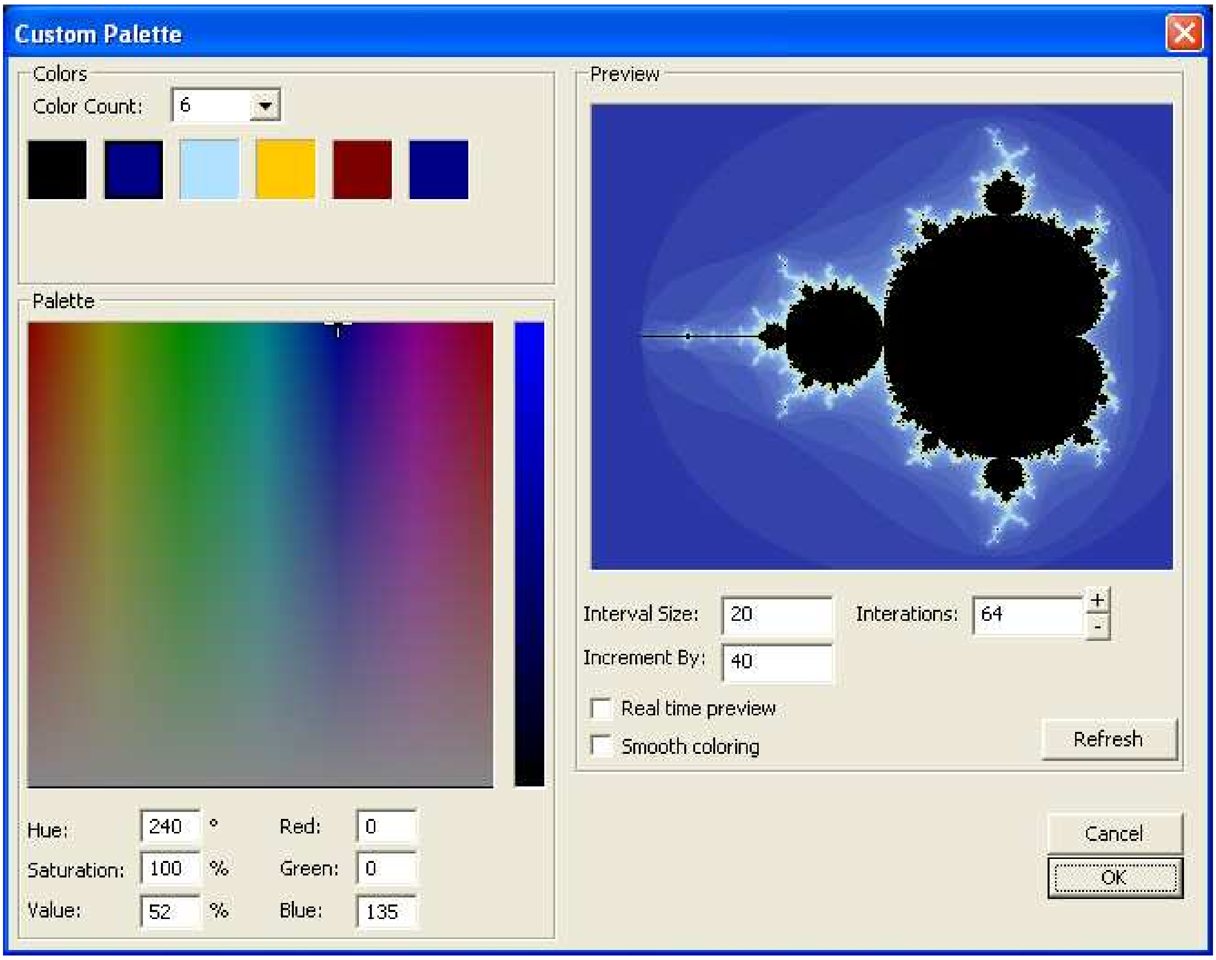
3.2.2 Vytvorenie vlastnej palety

Program AFractal umožňuje vytvoriť si jednu vlastnú paletu pre každý typ fraktálu. Po vytvorení novej palety sa predchádzajúca prepíše. Paleta sa vytvára pomocou dialógu ktorý otvoríme cez položku „Custom Palette“ v menu „Coloring“.

V ľavej časti dialógu sa nachádzajú ovládacie prvky na nastavenie jednotlivých farieb palety. Pomocou poľa so zoznamom v hornej časti vyberieme počet farieb. Minimálny počet je 3, prvá farba znázorňuje farbu bodov patriacich do množiny, nasledujú farby priradené jednotlivým intervalom iterácií postupne od najmenšieho počtu po najväčší. Maximálny počet farieb v palete je 14. Konkrétne farby sa nastavujú jedným z 3 spôsobov.

1. zadávaním hodnôt vo formáte HSV
2. zadávaním hodnôt vo formáte RGB
3. výberom konkrétnej farby z farebnej palety

Pravá časť dialógu obsahuje náhľad aktuálnej palety spolu s nastavením rozsahov intervalov farieb. Položka „Interval Size“ určuje veľkosť prvého farebného intervalu, „Increment By“ určuje o koľko bude každý ďalší interval väčší. V náhľade môžeme vidieť diskrétne aj spojité farbenie. Ktoré z nich je použité závisí od hodnoty „Smooth coloring“ v začiarkávacom políčku. Paleta sa dá uložiť pomocou tlačidla OK, po jeho stlačení sa dialóg zavrie a nová paleta sa aplikuje na aktuálny fraktál.



[Obr 7.] Dialóg na vytvorenie vlastnej palety

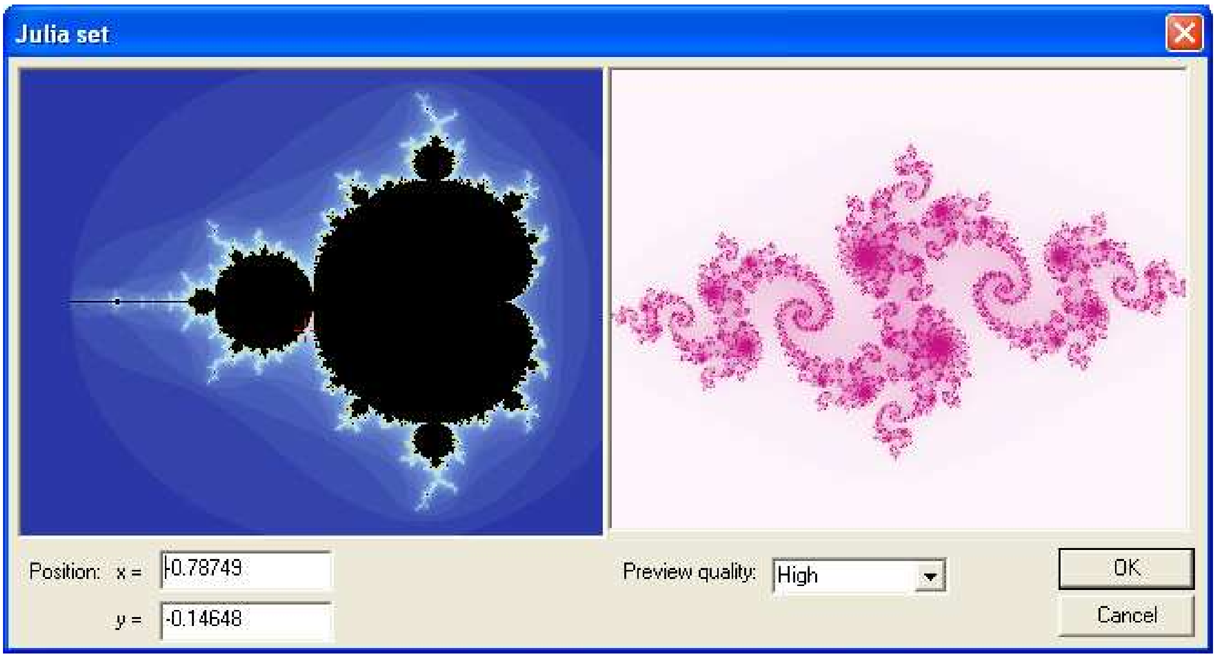
3.3 Menu „Fractal“

3.3.1 Zmena typu fraktálu

V súčasnosti podporuje program AFractal len dva typy fraktálov, Júliovu a Mandelbrotovu množinu. Typ fraktálu sa vyberá z menu položky „Fractal type“ umiestnenej v menu „Fractal“. Po kliknutí na položku „Mandelbrot“ sa nastaví ako aktuálny fraktál Mandelbrotova množina so základnými parametrami.

Po zvolení položky „Julia“ sa otvorí dialógové okno na vyber Júliovej množiny. Vzhľadom na úzku spojitosť medzi množinami, dialóg na výber Júliovej množiny obsahuje obraz Mandelbrotovej množiny, pomocou ktorého si môžeme zvoliť parameter určujúci tvar Júliovej množiny. Posúvaním bodu po Mandelbrotovej množine v ľavej časti dialógu sa v reálnom čase mení náhľad Júliovej množiny v pravej časti. Po potvrdení výberu tlačidlom „Ok“ sa v hlavnom okne aplikácie objaví Júliova množina, ktorú si užívateľ vybral. Dialóg ponuka tri kvality náhľadu:

1. Low – nízke rozlíšenie
2. Medium – vysoké rozlíšenie, menší počet iterácii
3. High – vysoké rozlíšenie, väčší počet iterácií



[Obr 8.] Dialóg na vyber Júliovej množiny

3.3.2 Nastavenia fraktálu

Pomocou dialógu „Fractal settings“ môžeme nastaviť všetky potrebné vlastnosti fraktálu. Dá sa nastaviť stred, hĺbka priblíženia, počet iterácií a hodnota exponentu použitého vo vzorci na výpočet fraktálu.

Medzi ďalšie možnosti na ovládanie vlastnosti patrí zvýšenie a zníženie počtu iterácií. Tieto zmeny nie sú lineárne. Pri zvýšení sa aktuálny počet vynásobí konštantou 1.5. Pri znížení sa počet vydelí rovnakou konštantou.

3.4 Animácia fraktálu

Program AFractal ponuka viacero možnosti animácie. Každú animáciu môžeme vytvoriť v jednom z troch rozlíšení: 320x256, 640x512, 800x640. Po vypočítaní animácie ju môžeme uložiť do súboru ako video. Rozlíšenie videa bude rovnaké ako to, v ktorom sme animáciu vypočítali.

Po nastavení želaných parametrov je potrebné vypočítať jednotlivé snímky animácie. Výpočet sa spustí po stlačení tlačidla „Compute“. Po jeho stlačení sa otvorí nové dialógové okno s ukazovateľom priebehu, ukazujúcim stav výpočtu. Výpočet je možné kedykoľvek ukončiť tlačidlom „Cancel“. Po dopočítaní animácie sa dialógové okno ukazujúce stav výpočtu automaticky zavrie, v dialógu s nastaveniami animácie sa tlačidlo „Compute“ zmení na „Run“ a povolí sa tlačidlo „Save to File“.

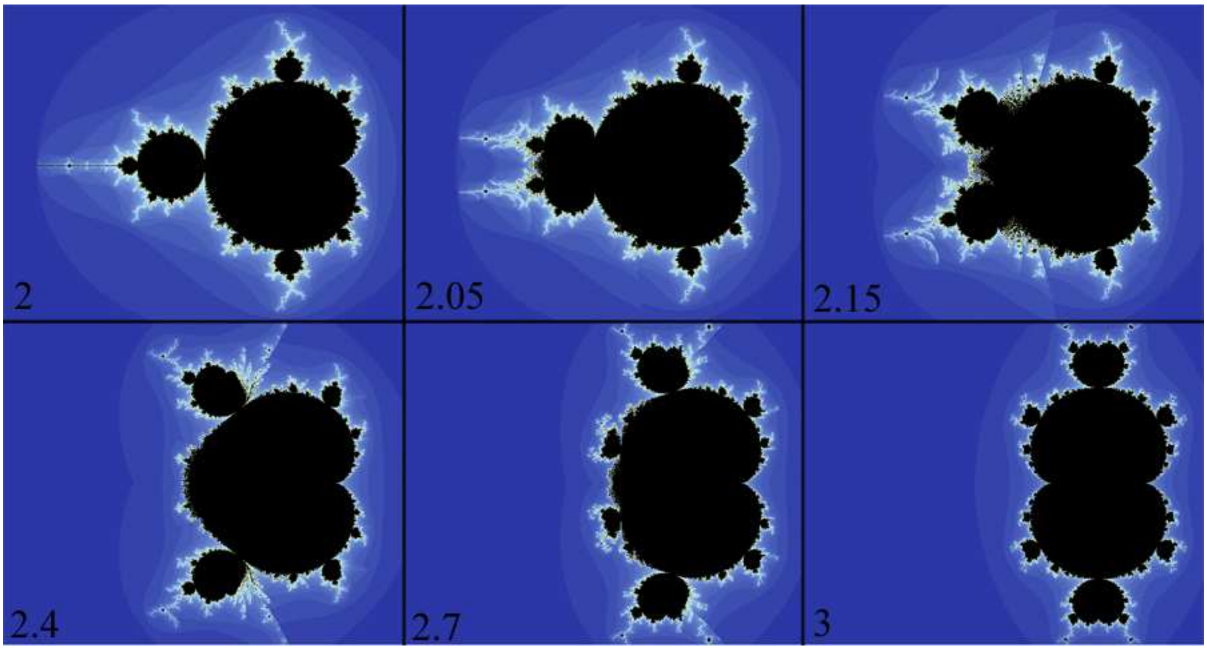
Po stlačení tlačidla „Run“ sa otvorí prehrávač animácie. Ide o veľmi jednoduchý video prehrávač schopný prezerať animáciu ako celok, alebo po jednotlivých snímkoch.

Po stlačení tlačidla „Save to File“ sa otvorí dialóg na uloženie súboru. Po vybratí cesty a mena súboru sa animácia uloží do video súboru.

Aplikácia si môže pamätať len jednu vypočítanú animáciu. V prípade, že sa v pamäti už nachádza vypočítaná animácia, pri pokuse o vytvorenie novej bude užívateľ na túto skutočnosť upozornený varovnou správou. Animácia uložená v pamäti sa prepíše v okamihu kedy sa začne počítať nová animácia.

3.4.1 Animácia zmeny exponentu

Najdôležitejším typom animácie, ktorú program AFractal ponúka, je animácia Mandelbrotovej množiny pri spojitej zmene exponentu použitého pri jej výpočte. Je to zároveň aj animácia, ktorá sa najjednoduchšie ovláda. Na jej vytvorenie je potrebné len nastaviť začiatočnú a koncovú hodnotu exponentu v dialógu „Animation of exponent change“. Obe hodnoty musia byť väčšie ako 0. Hodnota exponentu sa pri výpočte animácie lineárne interpoluje medzi zadanými hraničnými hodnotami.



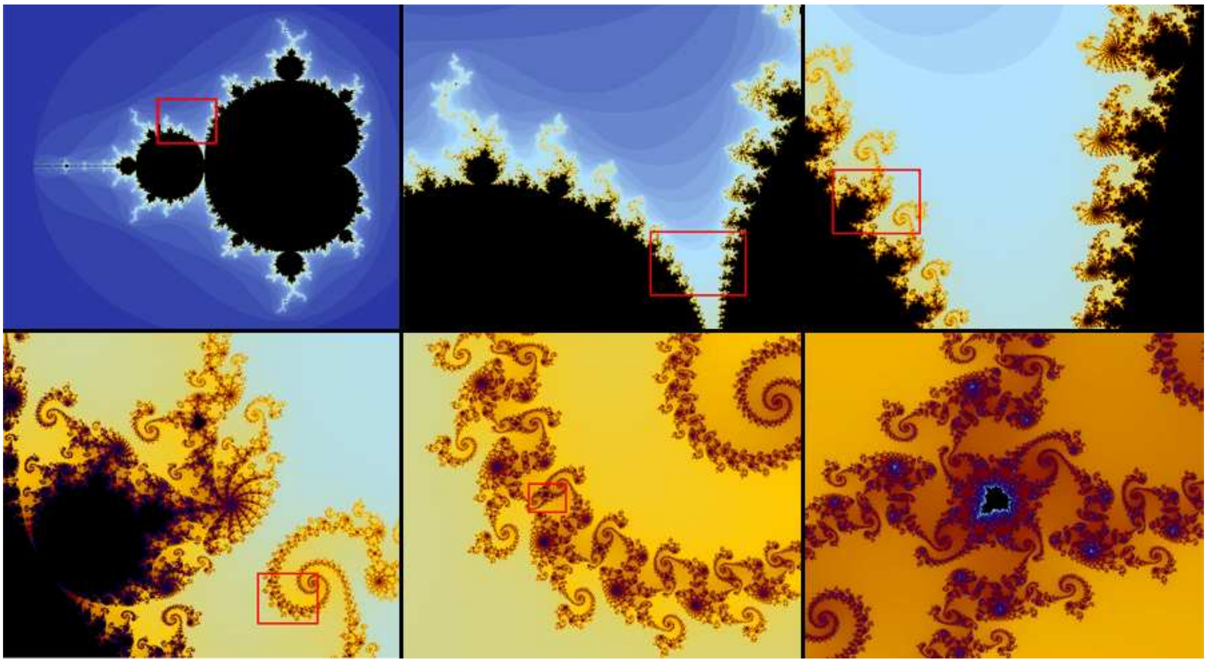
[Obr 9.] Ukážka zmeny exponentu pri výpočte Mandelbrotovej množiny

3.4.2 Animácia priblíženia

Ďalším typom animácie je animácia priblíženia. Z dôvodu jednoznačnosti je potrebné programu presne povedať kedy chceme začať nahrávať pohyb po fraktále. To je možné vykonať cez položku „Start recording“, ktorá sa nachádza pod „Animation of zoom“ pod menu „Animation“. Rovnakým spôsobom je nutné oznámiť aj ukončenie nahrávania položkou „Stop recording“. Po ukončení nahrávania je možné animáciu vypočítať. Dialóg s nastaveniami pri tomto type animácie najjednoduchší. Dá sa nastaviť iba celkový počet snímkov, počet snímkov za sekundu a rozlíšenie animácie.

Pri nahrávaní pohybu po fraktále sa v každom kroku uloží hodnota priblíženia, súradnice stredu a počet iterácii. Pri výpočte sa následne pre každý snímok použijú hodnoty získané prirodzenou kubickou interpoláciou týchto údajov. Výsledkom je plynulý pohyb po spojitej krivke.

Vzhľadom na spôsob pohybu po fraktále existujú dva spôsoby ako sa dá interpretovať krok spať. Je možné, že išlo o napravenie nesprávneho priblíženia, no mohlo to byť aj úmyselné preskúmanie „slepej uličky“. Nie je možné s istotou určiť, ktorý z prípadov nastal a taktiež by nebolo veľmi dobré obmedzovať užívateľov len na jeden prípad. Z tohto dôvodu je možné kedykoľvek toto správanie zmeniť nastavením hodnoty položky „Record full path“ nachádzajúcej sa pod „Animation of zoom“



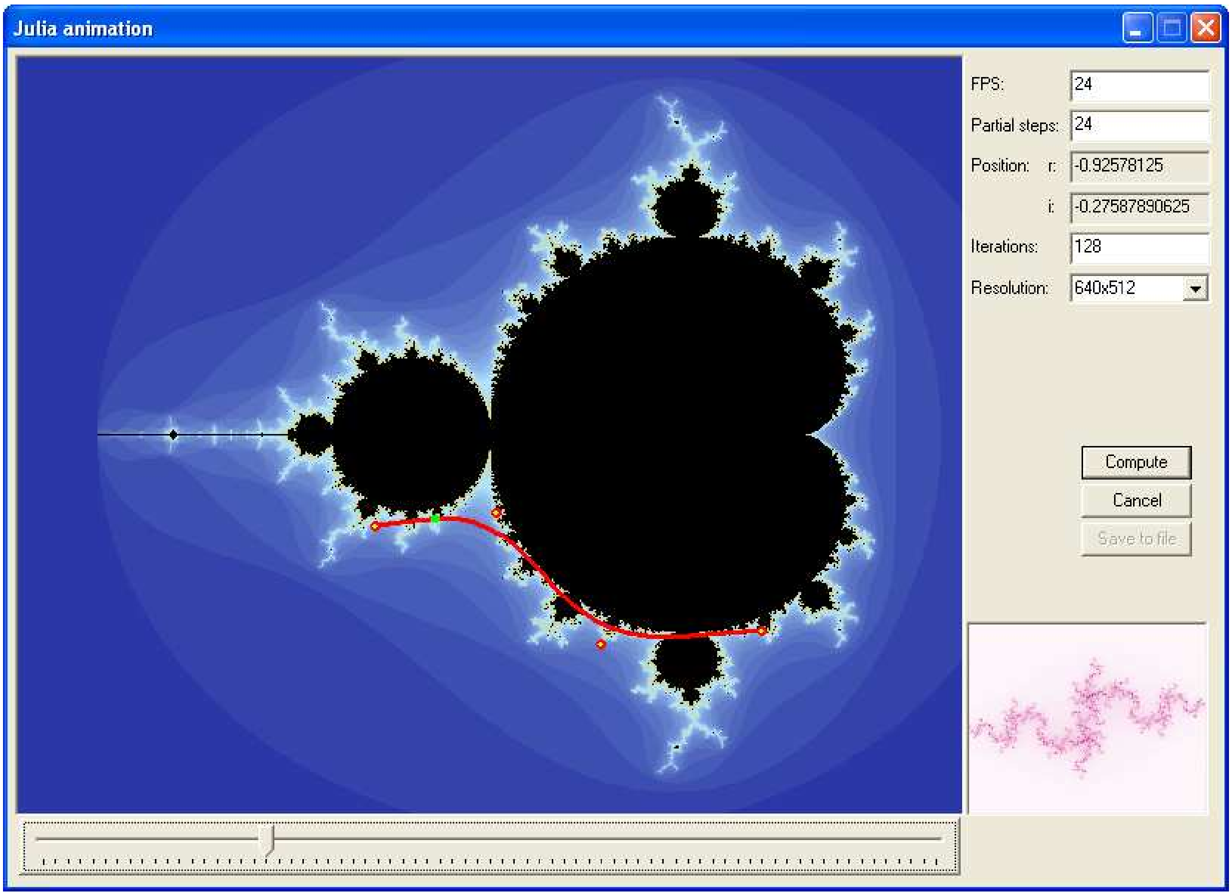
[Obr 10.] Ukážka animácie priblíženia. Červený obdĺžnik znázorňuje oblasť na nasledujúcom obrázku

3.4.3 Animácia Júliovej množiny

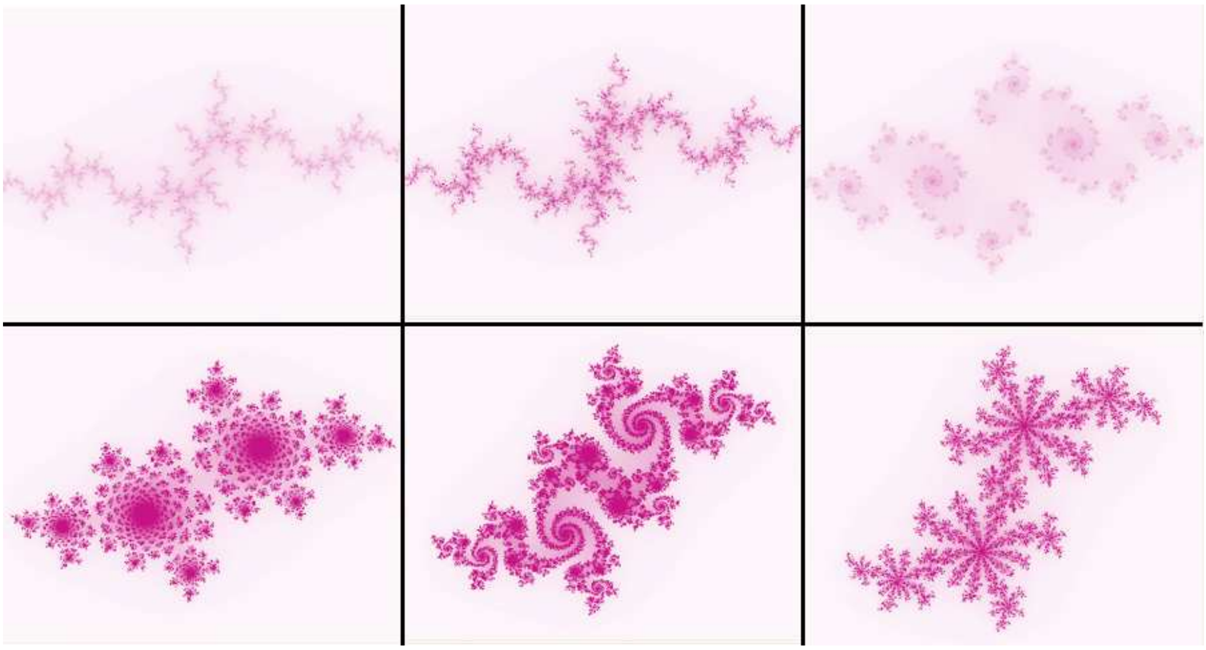
Posledný typ animácie, ktorú program AFractal ponúka je animácia Júliovej množiny. Postupnou zmenou bodu **z** určujúceho fraktál môžeme dosiahnuť plynulý prechod medzi jednotlivými množinami. Pre dosiahnutie najlepších výsledkov by mala byť zmena čísla **z** čo najmenšia. To dosiahneme pohybovaním bodu po užívateľom vytvorenej krivke. Pri dostatočnej jemnosti krivky sa dajú dosiahnuť veľmi zaujímavé animácie, ktoré pôsobia veľmi plynulým dojmom.

Dialóg na vytvorenie tejto animácie je najkomplikovanejší zo všetkých troch. Jeho najväčšiu časť tvorí obraz Mandelbrotovej množiny na ktorom sa vytvára krivka po ktorej sa bude pohybovať bod určujúci Júliovu množinu. Ide o NURBS krivku. Jej uzlový vektor a váhy sú pevne dané, jediné čo sa dá meniť, sú jej riadiace body. Pri priblížení myšou k niektorému z riadiacich bodov sa tento zvýrazní a je možné s ním manipulovať systémom drag&drop. Kliknutím pravým tlačidlom myši na zvýraznený bod sa tento vymaže. Nové body sa vytvárajú kliknutím ľavým tlačidlom myši do prázdneho priestoru. Každý nový bod sa pripojí na koniec krivky, nie je možné pridávať body inde ako na koniec. Pri akejkoľvek zmene sa krivka okamžite automaticky prepočíta.

Pod obrazom Mandelbrotovej množiny sa nachádza posuvník, ktorý určuje aktuálnu pozíciu bodu na krivke. Aktuálna pozícia bodu sa vypisuje v editovacích poliach v pravej časti dialógu. Medzi nastaviteľné parametre animácie patri počet iterácií, rozlíšenie, počet snímkov za sekundu a taktiež jemnosť krivky. Jemnosť krivky sa zadáva ako počet krokov medzi dvomi susediacimi riadiacimi bodmi. Celkový počet krokov teda získame zo vzorca (počet riadiacich bodov – 1)*(počet krokov medzi riadiacimi bodmi). V pravej dolnej časti dialógu sa nachádza náhľad aktuálnej Júliovej množiny, ktorý sa aktualizuje pri akejkoľvek zmene krivky, pripadne zmene pozície posuvníka.



[Obr 11.] Dialóg na nastavenie animácie Júliovej množiny



[Obr 12.] Priebeh animácie z Obr 11.

4. AFractal – realizácia

Program AFractal je naprogramovaný v jazyku C++ s použitím knižnice volaní Win32. Všetka grafika sa vykresľuje pomocou rozhrania GDI [4]. V počiatočných štádiách vývoja som zväzoval použiť na vykresľovanie rozhranie OpenGL kvôli animáciám, no ukázalo sa, že GDI uspokojuje všetky nároky aplikácie a aj pri najvyššom zaťažení je vykresľovanie plynulé.

Pri návrhu aplikácie som sa snažil viac zamerať na efektivitu ako na modulárnosť a možnosti rozšírenia. Vďaka princípom OOP, ktoré sú v aplikácii využité však nie je pridávanie nových typov fraktálov a animácií veľmi zložité, je ale potrebne urobiť niekoľko zmien aj v jadre aplikácie. Všetky fraktály musia dediť od základnej triedy `CFractal`, každý typ animácie je reprezentovaný animačným objektom, ktorý musí dediť od triedy `CAnimationObject`.

4.1 `CFractal` a odvodené triedy

Trieda `CFractal` je základnou triedou pre oba typy fraktálov použité v programe, takisto aj pre všetky prípadné ďalšie typy, ktoré by sa do programu doplnili. Ide o abstraktnú triedu s niekoľkými virtuálnymi metódami. Bližšie si popíšeme niektoré členské metódy a premenné ako aj spôsob výpočtu jednotlivých typov fraktálov.

4.1.1 Opis triedy `CFractal`

Premenná `m_fractal` je smerník na pole bajtov v ktorom je uložený obraz fraktálu. Pole ma veľkosť 640*512*3 = 984960 bajtov. Ako môžeme vidieť, každý pixel je charakterizovaný trojicou bajtov, každý pre jednu farbu (modrá, zelená, červená). Rozsah farieb je od 0 po 255.

Štyri reálne premenné (`m_xMin, m_yMin, m_xMax, m_yMax`) určujú hranice aktuálne zobrazovanej časti fraktálu. Používajú sa pri jeho výpočte a sú menené pri každom priblížení a vzdialení. Na dosiahnutie čo najväčšieho možného priblíženia je pri týchto hodnotách použitá najpresnejšia dostupná reprezentácia reálneho čísla.

Premenná `m_zoomDepth` určuje hĺbku priblíženia aktuálneho výrezu fraktálu. Čím viac je fraktál zväčšený, tým väčšia je hodnota tejto premennej. Keby bol fraktál štvornásobne zväčšený, hodnota premennej by bola 4.

Premenná `m_exp` určuje hodnotu exponentu použitého pri výpočte fraktálu, kvôli optimalizácii rozlišujeme prípady kedy je exponent celočíselný a kedy nie a každý prípad riešime inak. `m_iterationCount` určuje koľkokrát sa výpočet iteruje pokiaľ sa prehlási, že skúmaný bod patrí do množiny.

Premenná `m_fractalBuffer` je smerník na pole, kde každý prvok tohto pola je obraz fraktálu. Ide o zásobník predchádzajúcich výrezov fraktálu, ktorý sa využíva pre rýchle odďaľovanie.

Metóda `SetPixel` priradí pixlu určenému parametrami `x` a `y` farbu z farebnej palety v závislosti od parametru `dist`. Tato funkcia je použitá pri spojitom aj diskrétnom farbení, kvôli tomu je potrebné aby parameter určujúci farbu bol reálne číslo. Funkcia zisti do ktorého intervalu farieb parameter patri a následne podľa jeho hodnoty interpoluje medzi dvoma farbami určujúcimi tento interval.

Metóda `ZoomIn` najprv uloží aktuálny obraz fraktálu do zásobníka `m_fractalBuffer` a potom priblíži časť fraktálu určenú parametrami `x1, y1, x2, y2`. Tieto parametre určujú súradnice obdĺžnika vybraného na obrazovke určujúceho, ktorá časť fraktálu sa má priblížiť. Tieto je najprv potrebné prekonvertovať na reálne súradnice vo fraktále pomocou hodnoty `m_zoomDepth` a súradníc hraníc aktuálneho fraktálu. Následne sa upraví pomer strán na potrebnú veľkosť 4/5 a prepočítajú sa nové súradnice hraníc a hĺbka priblíženia. Funkcia vracia `false` pokiaľ sú súradnice `x1, y1, x2, y2` príliž blízko seba, teda vyznačený obdĺžnik bol príliž malý. V tomto prípade sa výrez nepriblíži. V opačnom prípade sa vráti `true` a fraktál je potrebne prepočítať s novými hodnotami.

V prípade, že v zásobníku m_fractalBuffer sa nachádzajú uložené predchádzajúce obrazy, metóda ZoomOut prepise aktuálny obraz fraktálu posledným obrazom uloženým v zásobníku, patrične upraví hodnoty hraníc a hĺbku priblíženia a vráti false. V prípade, že v zásobníku neostali žiadne predchádzajúce stavy, fraktál sa oddiali do polovičnej hĺbky priblíženia. Teda v prípade, že hĺbka bola 4, po zavolaní metódy ZoomOut s prázdnym zásobníkom bude hĺbka 2.
Návratová hodnota funkcii ZoomIn a ZoomOut hovorí, či je potrebne fraktál prepočítať, alebo stačí len prekresliť obraz.

Metóda Compute slúži na vypočítanie farebných hodnôt pixlov fraktálu. Metoda je navrhnutá tak aby umožňovala počítanie jedného fraktálu viacerými vláknami. Parameter n určuje koľko vlákien naraz počíta tento fraktál, parameter y určuje o koľké vlákno v poradí ide. Výpočet sa delí na jednotlivé stĺpce. V prípade, že chceme vypočítať celý fraktál jedným vláknom zavoláme funkciu s parametrami y = 0, n = 1.

4.1.2 Opis triedy CMandelbrot

Trieda CMandelbrot dedí od triedy CFractal. Nepridáva žiadne nové metódy ani premenné. Predefinuje však metódu, v ktorej sa fraktál vypočítavá.

4.1.3 Opis triedy CJulia

Trieda CJulia rovnako ako trieda CMandelbrot dedí od triedy CFractal. Okrem zdedených metód a premenných pridáva nasledovné:

Premenná m_z obsahuje komplexné číslo určujúce danú Júliovu množinu. Taktiež boli pridané aj metódy pracujúce s touto premennou SetSeed a GetSeed.

Metóda Compute je veľmi podobná metóde Compute z triedy CMandelbrot. Jediný rozdiel je v tom, že pri výpočte Mandelbrotovej množiny komplexné číslo z nastavujeme pre každý pixel zvlášť, pri výpočte Júliovej množiny používame hodnotu premennej m_z.

4.2 `CAnimObject` a odvodené triedy

Trieda `CAnimObject` je základnou triedou pre všetky typy animácií použité v programe, takisto aj pre všetky prípadné ďalšie typy, ktoré by sa do programu doplnili. Ide o abstraktnú triedu s niekoľkými virtuálnymi metódami. Bližšie si popíšeme niektoré členské metódy a premenné ako aj spôsob výpočtu jednotlivých typov fraktálov.

4.2.1 Opis triedy `CAnimObject`

`m_frames` je pole smerníkov na fraktály. Pri vytvorení inštancie triedy `CAnimObject` sa tieto fraktály nainicializujú a pripravia na následné počítanie.

Premenná `m_numFrames` obsahuje počet snímkov animácie. Určuje počet fraktálov, ktoré sú v animácii a teda aj veľkosť pola `m_frames`.

Premenná `m_quality` obsahuje jednu z troch hodnôt určujúcich rozlíšenie animácie. Podľa tejto premennej sa fraktálom z poľa `m_frames` nastaví správna veľkosť. Možné hodnoty sú:

- 1 – rozlíšenie 320x256
- 2 – rozlíšenie 640x512
- 3 – rozlíšenie 800x600

Metóda `CreateAVI` vytvorí video súbor za použitia snimkov z pola `m_frames` pomocou triedy Avi, ktorej autormi sú Lucian Wischik a René Nyffenegger [6].

4.2.1 Opis triedy `CExpAnimObject`

Trieda `CExpAnimObject` slúži na vytvorenie animácie spojitej zmeny exponentu. Okrem metód a premenných zdedených od rodičovskej triedy `CAnimObject` obsahuje aj dve nové premenné.

Premenné `m_startExp` a `m_endExp` obsahujú počiatočnú a koncovú hodnotu exponentu, ktorá sa pri výpočte animácie lineárne interpoluje pre každý fraktál.

Okrem nových premenných trieda `CExpAnimObject` predefinuváva metódu `Compute.` Tato metóda zistí aktuálnu hodnotu exponentu pre každý fraktál a spustí jeho počítanie.

4.2.1 Opis triedy `CMoveAnimObject`

Trieda `CMoveAnimObject` slúži na vytvorenie animácie priblíženia. Táto trieda pridáva niekoľko pomocných premenných na prácu s krivkou, po ktorej sa pohybuje kamera. Okrem nich pridáva aj samotnú krivku

Trieda `CAnimationPath` zaobaľuje prirodzenú kubickú krivku a ponúka rozhranie na jednoduchú prácu s ňou. Umožňuje vytvoriť ľubovoľne dlhú krivku s ľubovoľnou jemnosťou. Každý riadiaci vrchol krivky obsahuje hodnotu exponentu, počet iterácií, ako aj hodnotu hraníc výrezu fraktálu.

Predefinovaná metóda `Compute` zistí pre každý fraktál animácie aktuálne hodnoty a nastaví fraktál, ktorý sa následne vypočíta.

4.2.1 Opis triedy `CJuliaAnimObject`

Trieda `CJuliaAnimObject` slúži na vytvorenie animácie Júliovej množiny. Táto trieda rovnako ako trieda `CMoveAnimObject` pridáva niekoľko pomocných premenných na prácu s krivkou, po ktorej sa bude pohybovať bod určujúci Júliovu množinu. Okrem nich pridáva aj samotnú krivku. Tentoraz ide o NURBS krivku.

Trieda `CNURBS` zaobaľuje NURBS krivku a ponúka rozhranie na jednoduchú prácu s ňou. Umožňuje vytvoriť ľubovoľne dlhú krivku s ľubovoľnou jemnosťou. Nedá sa však nastaviť váha jednotlivým vrcholom a ani uzlový vektor.

Predefinovaná metóda `Compute` je opäť veľmi jednoduchá. Jej jedinou úlohou je zistenie aktuálnej hodnoty komplexného bodu určujúceho Júliovu množinu pomocou NURBS interpolácie a vypočítanie tejto množiny.

4.3 Multithreading

V súčasnosti je badať výrazný trend v návrhu procesorov tykajúci sa viacjadrových riešení. Ukazuje sa, že po najbližšie roky sa budú výrobcovia zameriavať na pridávanie väčšieho počtu jadier, zatiaľ čo rýchlosť jednotlivých jadier sa bude zvyšovať len pomaly. Rozhodol som sa čo najviac prispôsobiť tomuto trendu a navrhnúť aplikáciu AFractal tak, aby dokázala do čo najväčšej miery využiť ponúkanú možnosť súbežného počítania.

Vzhľadom na charakter výpočtu fraktálu, kedy výpočet jedného pixlu nezávisí od výpočtu žiadneho iného, je veľmi jednoduché zaviesť multithreading. Mnou zvolený prístup je automaticky škálovateľný a veľmi jednoduchý na použitie. Pri spustení aplikácie sa pomocou CPUID inštrukcie zistí počet jadier procesora. Na základe tohto počtu sa vytvorí príslušný počet vlákien, medzi ktoré sa následne delí výpočet. Vzhľadom na to, že všetky vlákna vznikajú pri spustení programu a ostávajú až do jeho ukončenia, je práca s nimi rýchla a vyžaduje len minimálne zvýšenie spotreby prostriedkov na ich spravovanie. Vlákno je aktívne iba počas počítania. Keď dokončí úlohu a už nie je viac čo počítať, uspí sa.

Vlákna sa vytvárajú pomocou volania funkcie `CreateThread` z knižnice volaní Win32 [5]. Identifikátor vlákna sa po jeho vytvorení uloží do poľa aby sa s ním následne mohlo pracovať. Každé vlákno sa vytvorí v suspendovanom režime a nastaví sa mu nízka priorita aby náročný výpočet nebránil bežnej práci s počítačom.

V prípade, že sa počíta jeden fraktál, výpočet sa delí na stĺpce. Ak **n** je počet vlákien, potom každé vlákno počíta každý **n**-tý stĺpec, začínajúc od **i**-teho, kde **i** je rôzne pre každé vlákno a **i** $\in$ `<0,` **n**`)`. Takto docielime to, že sa vypočíta každý stĺpec a práca sa rovnomerne rozdelí medzi všetky vlákna. Výnimku tvorí niekoľko posledných stĺpcov v prípade, že počet stĺpcov nie je deliteľný počtom vlákien. Týmto spôsobom môžeme počítanie rozdeliť až medzi 640 vlákien.

V prípade výpočtu animácie je delenie jednoduchšie. Každé vlákno počíta jeden celý fraktál. V prípade, že zoznam fraktálov čakajúcich na vypočítanie nie je prázdny, po dopočítaní aktuálneho fraktálu dostane vlákno ďalší fraktál zo zoznamu. Je tu však rovnaký problém s poslednými fraktálmi. V prípade, že na dopočítanie ostáva menej fraktálov ako je voľných vlákien, niektoré vlákna môžu ostať nevyužité.

5. Výsledky

Vzhľadom na to, že výpočet fraktálov môže byť veľmi časovo náročný, pravdepodobne najdôležitejšia charakteristika je rýchlosť. V nasledujúcej tabuľke je uvedených niekoľko meraní dĺžky výpočtu rôznych časti Mandelbrotovej množiny s vypnutým aj zapnutým multithreadingom. Všetky merania boli vykonané na procesore Intel® Core™ 2 DUO E6400 s frekvenciou 3 GHz. Každé meranie bolo vykonané 5 krát, uvedená je priemerná hodnota. Rozlíšenie meraného fraktálu bolo vo všetkých prípadoch 1280x1024. V prípade, že niektoré meranie vykazovalo výnimočne veľkú odchýlku, z dôvodu zaťaženia procesoru inými procesmi, bolo opakované.

5.1 Porovnanie rýchlostí na viacjadrových procesoroch

Počet iterácií	**C**	**Hĺbka vnorenia**	**Počet vlákien**	**Čas výpočtu [s]**
64	-0,75 + 0.i	1	2	0,210
64	-0,75 + 0.i	1	1	0,407
1600	-1,18533 + 0,30532.i	204200	2	3,388
1600	-1,18533 + 0,30532.i	204200	1	6,623

Tab. 1 Porovnanie rýchlostí výpočtu Mandelbrotovej množiny pri jednom a dvoch vláknach

Z uvedených výsledkov je ľahko vidieť, že pri zvyšovaní počtu vlákien úmerne klesá čas potrebný na výpočet. Taktiež môžeme pozorovať veľmi malú réžiu spojenú so správou viacerých vlákien a z toho vyplývajúcu takmer zdvojnásobenú rýchlosť výpočtu pri zdvojnásobení počtu jadier.

5.2 Porovnanie rýchlostí výpočtu neceločíselného exponentu

Nasledujúce merania poukazujú na rýchlostný rozdiel pri počítaní celočíselných a neceločíselných exponentov. Keďže pri zmene exponentu sa mení aj tvar množiny, nemá porovnávanie výpočtu rôznych exponentov veľmi veľkú výpovednú hodnotu. Z tohto dôvodu bol ako posledné meranie prevedený výpočet fraktálu s celočíselným exponentov metódou, ktorá sa využíva pri výpočte neceločíselného. Rozlíšenie fraktálu pri každom meraní bolo 1280x1024, počet iterácií 128, stred C = -0.75 + 0.i, hĺbka vnorenia bola 1.

Exponent	Čas výpočtu [s]
2	1,43
2,1	6,68
2,5	6,73
5,2	7,32
2+	5,75

Tab. 2 Porovnanie rýchlostí výpočtu celočíselného a neceločíselného exponentu

5.3 Porovnanie rýchlostí výpočtu jednotlivých typov animácií

Časy výpočtov jednotlivých typov animácií sa výrazne líšia. Konkrétne hodnoty sú uvedené v tabuľke 3. Všetky animácie mali nastavené 128 iterácii. Animácia priblíženia bola rovnaká ako animácia z obrázku 10. Animácia Júliovej množiny bola rovnaká ako na obrázku 12. Animácia zmeny exponentu sa počítala od exponentu 2 po exponent 3.

Typ animácie	Počet snímkov	Rozlíšenie	Čas výpočtu [s]
Animácia priblíženia	120	320x256	8,9
		640x512	35,2
		800x640	54,8
Animácia Júliovej množiny	72	320x256	1,2
		640x512	3,9
		800x640	6,3
Animácia zmeny exponentu	72	320x256	32,1
		640x512	129,8
		800x640	204,2

Tab. 3 Porovnanie rýchlostí výpočtov jednotlivých typov animácie

5.4 Pamäťová náročnosť

Aplikácia po spustení zaberá približne 35MB operačnej pamäte. Táto spotreba sa zvyšuje pri prezeraní fraktálu. Pri prvých desiatich vnoreniach sa alokuje každým vnorením ďalší 1MB pamäte, kvôli rýchlemu vráteniu do predchádzajúcej pozície. Najväčšie nároky na pamäť si kladú animácie. Pre každý snímok animácie je potrebné alokovat pamäť, jej veľkosť je závislá od veľkosti snímku. Presne hodnoty sú uvedené v tabuľke 4.

Rozlíšenie	Veľkosť jednej snímky
320x256	250 KB
640x512	1 MB
800x640	1,5 MB

Tab. 4 Veľkosti snímkov animácie pri jednotlivých rozlíšeniach

Pre dvojsekundovú animáciu v rozlíšení 640x512 to je približne 50MB. Pri dlhších animáciách tak môžeme dosiahnuť až niekoľko stoviek MB, pripadne niekoľko GB pamäte pri veľmi dlhých animáciách.

5.5 Ukladanie výsledkov

Animácie aj jednotlivé obrázky sa dajú ukladať do súborov. V prípade animácií sú to avi [6] video súbory, v prípade obrázkov ide o obrázky typu bmp [7]. Veľkosti výsledných avi súborov sú uvedené v tabuľke 5. Ide o rovnaké animácie ako v tabuľke 3.

Typ animácie	Počet snímkov	Rozlíšenie	Veľkosť súboru
Animácia priblíženia	120	320x256	1,16 MB
		640x512	6.13 MB
		800x640	8.91 MB
Animácia Júliovej množiny	72	320x256	724 KB
		640x512	2,20 MB
		800x640	3,19 MB
Animácia zmeny exponentu	72	320x256	758 KB
		640x512	2,08 MB
		800x640	2,93 MB

Tab. 5 Porovnanie veľkosti súborov s animáciami

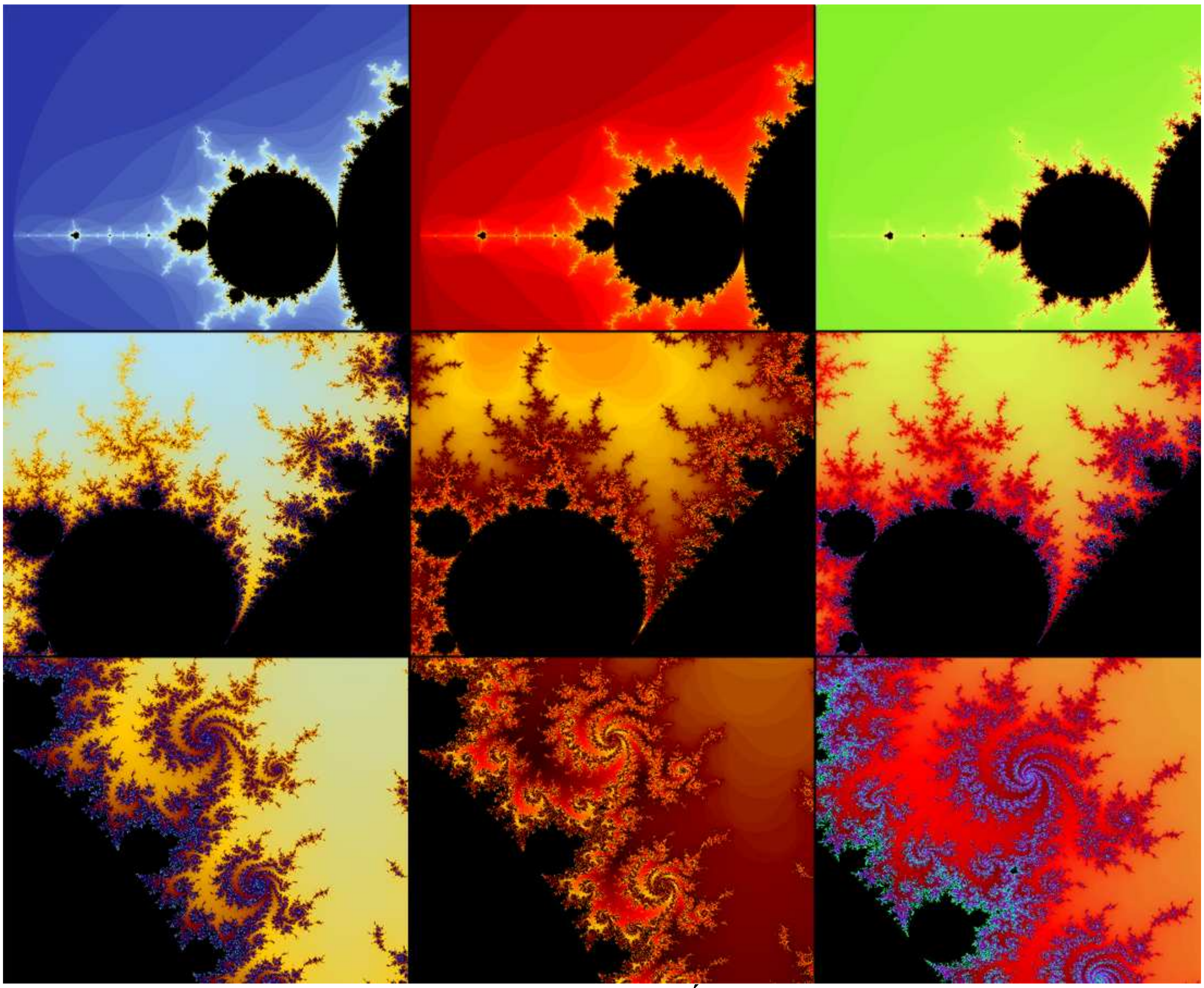
Obrázky vo formáte bmp na rozdiel od videosúborov majú rovnakú veľkosť bez ohľadu na jeho obsah. Jediný faktor ovplyvňujúci výslednú veľkosť súboru sú rozmery obrázku. V tabuľke 6 sú uvedené niektoré vybrané rozmery a im prislúchajúca veľkosť súboru.

Rozmery	Veľkosť súboru
800x600	1,37 MB
640x512	2,25 MB
1280x1024	3,75 MB

Tab. 6 Veľkosti súborov

5.6 Farebné palety

Aplikácia AFractal ponúka na výber 5 rôznych paliet, z toho 3 pre Mandelbrotovu množinu a 2 pre Júliovu množinu. Na **[Obr 13.]** sú uvedené výrezy množín s rôznymi paletami pre porovnanie.



[Obr 13.] Ukážka rôznych paliet. Každý stĺpec znázorňuje inú paletu

Použité zdroje

[1] B. Mandelbrot: The Fractal Geometry of Nature. ISBN 07-167-1186-9

[2] http://en.wikipedia.org/wiki/Fractal

[3] http://www.miqel.com/fractals_math_patterns/visual-math-natural-fractals.html

[4] http://msdn.microsoft.com/sk-sk/library/ms536795%28en-us,VS.85%29.aspx

[5] http://msdn.microsoft.com/en-us/library/ms684841.aspx

[6] http://www.wischik.com/lu/programmer/avi_utils.html

[7] http://msdn.microsoft.com/en-us/library/ms532311(VS.85).aspx