

Masarykova univerzita, Fakulta informatiky



Diplomová práce

Editor hierarchických mozaík

Brno 2003

Bc. Ján Bokor

Prehlásenie:

Prehlasujem, že táto práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

V Brně 14. 4. 2003

.....

Ján Bokor

Pod'akovanie:

Na tomto mieste by som sa rád poďakoval vedúcemu diplomovej práce prof. Ivovi Serbovi za jeho inšpiráciu, plnú podporu a celkovú pomoc s prácou ako i prof. J. Pelikánovi, ktorý mi pomohol pri riešení niektorých problémov, počas programovania v prostredí Delphi.

Zhrnutie

Táto práca pojednáva o hierarchických mozaikách, zaraďuje ich do ostatnej množiny mozaik, popisuje ich tvorbu, interpretáciu vhodnou dátovou štruktúrou. Analyzuje ich vlastnosti a možnosti. Skúma princípy jej postupného vytvárania ako aj dôsledky rôznych postupov. Snaží sa, ak nie rozšíriť obzor poznania tohto typu mozaik, tak aspoň zahľadiť sa na túto problematiku iným uhlom pohľadu. Ďalej popisuje prostredie aplikácie na tvorbu týchto mozaik vrátane ovládania tejto aplikácie.

Kľúčové slová

Mozaika, hierarchická mozaika, dlaždica, množina základných dlaždíc, množina prípustných transformácií, rozkladové pravidlo.

OBSAH

Úvod	4
1. Pojem mozaiky a základné rozdelenie	5
1. 1. Periodické mozaiky	7
1. 2. Neperiodické a aperiodické mozaiky	8
2. Hierarchické mozaiky	10
2. 1. Definícia	10
2. 2. Rozkladové pravidlá a hierarchia	11
3. Návrh a realizácia programového diela	15
3. 1. Verzia 1	15
3. 2. Finálna verzia	16
3. 2. 1. Celková stavba projektu	17
3. 2. 2. Základné princípy	21
3. 2. 3. Realizácia transformácií	27
3. 3. Problémy a riešenia	30
4. Popis a ovládanie Editoru hierarchických mozaik	30
4. 1. Popis prostredia	30
4. 2. Stručný popis funkcií horného menu	31
4. 3. Načítanie a uloženie bitmapy	32
4. 4. Tvorba a práca s rozkladovými pravidlami	33
4. 5. Škálovanie	36
4. 6. Práca s príkazovým zásobníkom	37
4. 7. Zoznam klávesových skratiek	39
Záver	41
Literatúra	42

"Svet sa skladá z nedeliteľných častíc, atómov." - Demokritos (asi 460 - 370 pr. Kr.)

Úvod

Čo sú to mozaiky? Ako ich presne definovať? Čo s nimi možno robiť? Zdá sa vám, že mozaiky sú len obrazy poskladané z nejakých častí? V podstate je to pravda. Ale nie je dôvod ich len preto zavrhnúť a nezaoberať sa nimi. Mozaiky nám môžu poskytnúť iný návod na definovanie obrazov, iný uhol pohľadu na rôzne predmety, ktoré nimi možno definovať. Vieme, že mnohé reálne objekty je možné modelovať prostredníctvom fraktálov. A ak zvolíme ten správny uhol pohľadu, nedávajú nám mozaiky aspoň podobné možnosti? Už Demokritos povedal, že svet sa skladá z atómov. To oni sú tie časti obrovskej mozaiky, ktorej sme všetci súčasťou.

Táto práca si kladie za úlohu oboznámiť vás s problematikou mozaík. Ukázať, že mozaiky môžu byť mocným nástrojom či už k dekorácii interiérov, tvorbe zaujímavých textúr, alebo dokonca modelovaniu určitého prostredia.

Takže v prvej časti sa pokúsime vytvoriť akúsi formálnu definíciu mozaiky. Popíšeme si jej vlastnosti, ukážeme, ako sa vo všeobecnosti mozaika tvorí, aké je treba dodržiavať pravidlá pri jej tvorbe a napokon si popíšeme dva druhy asi najznámejších druhov mozaik. Periodickú mozaiku a špirálnu mozaiku.

Druhá časť pojednáva výlučne o hierarchických mozaikách. Tu si ich definujeme popíšeme ich vlastnosti a hlavne tvorbu, kvôli ktorej vďačia za svoj názov. Budeme sa snažiť zobecniť pravidlá jej tvorby, dať im nejaký konkrétny tvar. Popíšeme si ako ich navrhovať a aplikovať.

Tretia časť, tou je realizácia programového diela Editora hierarchických mozaik. Ako algoritmicke realizovať tvorbu týchto mozaik. Akú dátovú štruktúru som použil v tomto Editore. Na akom princípe fungujú jednotlivé operácie. S akými problémami som sa potýkal. Ako som niektoré riešil.

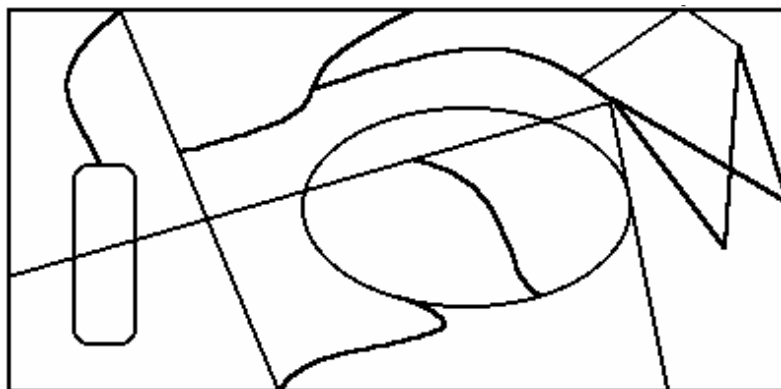
Posledná štvrtá časť už len popisuje hotové programové dielo. Jeho vzhľad, ovládanie, tvorbu hierarchických mozaík. Nájdu sa aj nejaké dobré rady, napríklad čoho sa treba vyvarovať, ako najlepšie postupovať a podobne.

Celá práca je obohatená obrázkami, ktoré majú poskytnúť či už nejaký príklad niečoho, alebo schematicky znázorniť niektoré postupy alebo vnútornú štruktúru. Dúfam, že je písaná zrozumiteľne a mnohých ďalej inšpiruje.

1. Pojem mozaiky a základné rozdelenie

Väčšina ľudí si pod pojmom mozaika predstaví buď niečo s opakujúcim sa vzorom, alebo niečo poskladané z plôšok rôznych tvarov a farieb. V každom prípade vždy pomyslia na nejaký ľudský výtvor, na niečo, čo vytvoril človek, aby vyriešil nejaký estetický problém. Avšak nie je náhodou príroda tým najväčším tvorcom mozaik? Stačí pozrieť na listy na stromoch, na ich žilkovanie, ktoré vytvára akúsi symetrickú štruktúru listu, na zvlnenú vodnú hladinu, líku posiatu kvetmi, dalo by sa vymenúvať donekonečna. Vo vzduchu ostáva visieť nevyzpytovaná otázka. Je všetko to, čo tu bolo vymenované naozaj mozaika? Ako ju presne definovať?

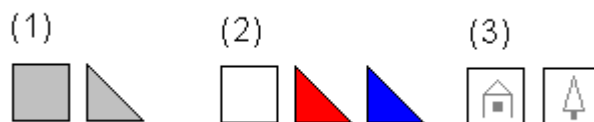
Možno práve touto cestou: Mozaika je matematická štruktúra pozostávajúca z priestoru, ktorý ju tvorí. Na tomto priestore existuje disjunktný rozklad na podpriestory, ktoré nazveme dlaždicami. Môžeme tak povedať, že mozaika sa skladá z dlaždíc. (Strom teda nemôže byť mozaikou, ktorej dlaždice sú len listy.) Ďalej je nutné predpokladať, že dlaždice musia mať takú istú dimenziu, ako daný priestor. Čiže v 2D rovine môžu byť dlaždicami trojuholník, štvorec, kruh a podobne. V 3D priestore zasa kváder, ihlan, guľa, alebo iný 3D objekt (neobsahujúci menejrozmerné časti napr. vyčnievajúce čiary a pod). A tieto dlaždice, musia daný priestor vyplniť bezo zvyšku.



Obr. 1 Príklad mozaiky, rozdelenie priestoru na dlaždice

Na obrázku môžeme vidieť jednu mozaiku, ktorá spĺňa všetky podmienky našej definície. Nosným priestorom je časť roviny, obdĺžnik, rozdelený na jednotlivé dlaždice, ktoré sa neprekrývajú a tento priestor vyplňajú celý. Je to naozaj mozaika. Pre nás však istým spôsobom nevyhovuje. Pozostáva totiž z dlaždíc, z ktorých je každá jedinečná a teda by bolo pre nás v tomto prípade jednoduchšie popísať celý "obraz" ako taký a nie ako nejakú mozaiku. Takže ako prvé by sme mali obmedziť tzv. množinu základných dlaždíc a prípustné transformácie dlaždíc.

Pod množinou základných dlaždíc rozumieme množinu takých rôznych tvarov a obsahov, ktoré jedine môžu byť použité pri stavbe mozaiky. Táto množina musí byť minimálne jednoprvková.



Obr. 2 Príklady množín základných dlaždíc v rovine

V prípade (1) vidíme, že mozaika bude pozostávať len zo štvorcov a trojuholníkov. Pomocou množiny základných dlaždíc (2), môžeme použiť biely štvorec, a dva rôzne svetlé trojuholníky. Príklad (3) tiež ukazuje, že nielen tvarom sa musia základné dlaždice líšiť. Mali by sme taktiež uvažovať aj množinu prípustných transformácií, ktoré môžeme na dlaždice aplikovať. Malo by tu byť predovšetkým posunutie, ktoré využívame na ukladanie dlaždíc do nosného priestoru. Ďalej sa veľmi často vyskytuje rotácia dlaždíc, čo znamená, že dlaždice môžu byť v mozaike rôzne natočené. Môžeme mať však ešte veľa iných transformácií ovplyvňujúcich ako tvar (škálovanie, ohýbanie, zošikmovanie atď.), tak aj obsah (zosvetlenie, zpriehľadnenie, rôzny morfung atď.).

Ako prípustné transformácie sa často využívajú diskkrétne transformácie. Napríklad kroková rotácia (otáčanie o pevne daný uhol), diskkrétne posunutie (posunutie po pevne zadaných osách o pevne danú dĺžku) a podobne.

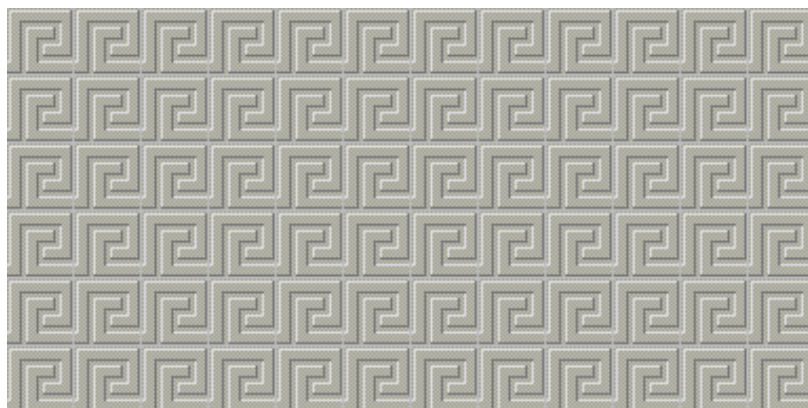
Ďalej by bolo vhodné zadať si pojem obrazca. Pod obrazcom budeme rozumieť výslednú štruktúru zostavenú z dlaždíc. Na prvý pohľad mozaika a obrazec reprezentujú to isté. Avšak ak povieme, že obrazec je popísaný nejakou mozaikou, je na mieste otázka, či ten istý obrazec nepopisuje viacero mozaik. To nás privádza k definícii ekvivalencie mozaik. Vravíme, že dve mozaiky sú navzájom ekvivalentné, ak popisujú ten istý obrazec.

Množina základných dlaždíc a množina prípustných transformácií nemusia byť až také nezávislé, ako by sa na prvý pohľad mohlo zdať. Transformácia je vo všeobecnosti nejaká operácia, ktorá istým vlastnostiam nejakého objektu, v našom prípade dlaždici alebo mozaiky, priradí vlastnosti iné. A teda k mozaike s viacprvkovou množinou základných dlaždíc určite existuje ekvivalentná mozaika s jednoprvkovou množinou základných dlaždíc, kde množina prípustných transformácií je rozšírená o transformácie premeny základnej dlaždice na ostatné základné dlaždice mozaiky prvej.

1. 1. Periodické mozaiky

Periodické mozaiky môžeme definovať, cez obrazce, ktoré popisujú. Ku každej periodickej mozaike musí existovať k nej ekvivalentná mozaika s jednoprvkovou množinou základných dlaždíc a s jednoprvkovou množinou prípustných transformácií, totiž posunutím. V každej periodickej mozaike sa preto musí nachádzať tzv. fundamentálna oblasť, ktorú keď budeme rekurzívne obklopovať jej kópiami, dostaneme ten istý obrazec. Je zrejmé, že fundamentálna oblasť je dlaždicou vyššie spomínanej ekvivalentnej mozaiky.

V každej periodickej mozaike sa dá nájsť nekonečne veľa rôznych fundamentálnych oblastí. V 2D priestore to môžu byť obdĺžniky, kosodĺžniky, šesťuholníky atď. Napríklad pokiaľ máme periodickú mozaiku s jednou základnou dlaždicou, tak fundamentálnu oblasť tvorí práve táto dlaždica, jej posunutie o ľubovoľný smer a ľubovoľnú vzdialenosť, zjednotenie dvoch susedných dlaždíc a podobne.

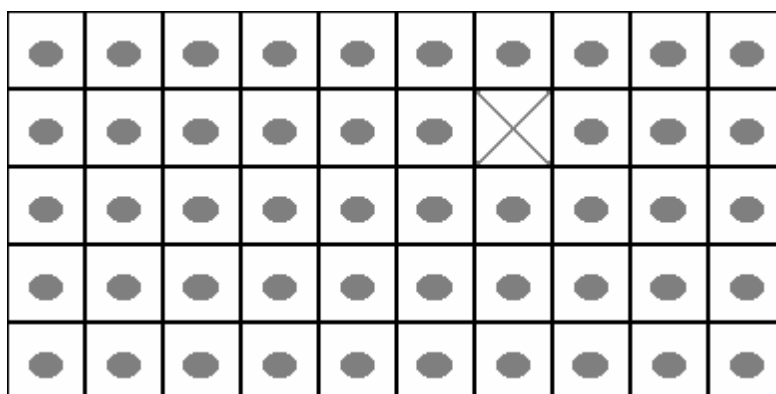


Obr. 3 Periodická mozaika

Periodické mozaiky sa pre svoju jednoduchosť používajú najčastejšie. Taktiež ľudia si pod pojmom mozaika predstavia najskôr tento typ mozaiky, niečo, čo sa opakuje, čo sa dá vidieť napríklad v architektúre, na odevoch a pod. V počítačovej grafike sa vo veľkom používajú ako textúry popisujúce rôzne povrchy.

1. 2. Neperiodické a aperiodické mozaiky

Neperiodickou mozaikou je každá mozaika, ktorá nie je periodická. A teda nedá sa v nej nájsť fundamentálna oblasť. Periodicitu stačí narušiť vytvorením kazu, v podobe novej dlaždice rovnakého tvaru, ktorú vymeníme za jednu dlaždicu periodickej mozaiky. To nám znemožní vytvoriť akúkoľvek fundamentálnu oblasť.



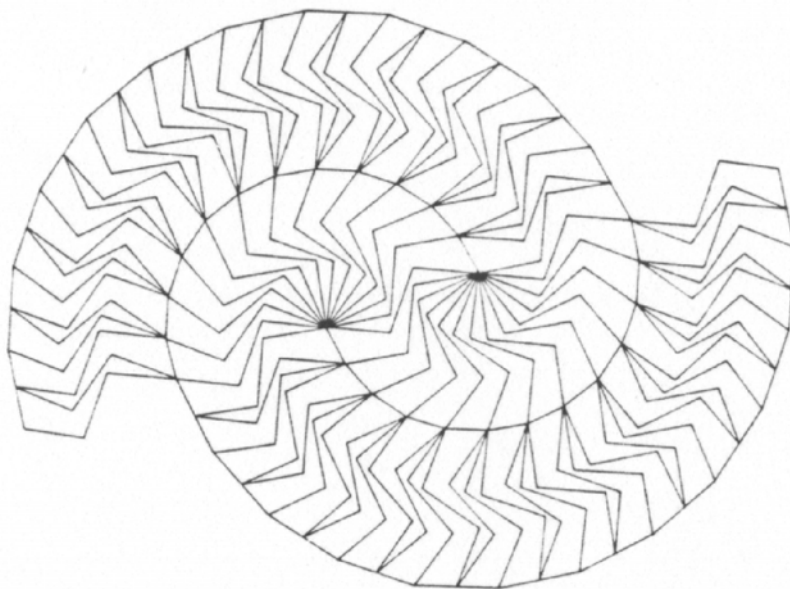
Obr. 4 Príklad neperiodickej mozaiky

Avšak takto vytvorené mozaiky často nemajú takmer žiadnu estetickú hodnotu a sú pre nás nezaujímavé. Pokiaľ potrebujeme vytvoriť na periodickej mozaike nejaký kaz (napríklad prasklinu v múre), je vhodnejšie na dané miesto v periodickej mozaike nalepiť obrázok s definovanou priehľadnosťou (tzv. masku). Túto masku je potom možné aplikovať na viacero textúr reprezentujúcich rôzne povrchy, okrem toho táto maska môže byť škálovaná nezávisle na dlaždicach mozaiky a teda použitie tohoto riešenia dáva ďaleko väčšie možnosti.

Aperiodickou mozaikou nazveme mozaiku, ktorej základná množina dlaždíc ani nedovoľuje vytvorenie periodickej mozaiky. Uvedený obrázok tak zobrazuje síce neperiodickú, avšak nie aperiodickú mozaiku. Lebo pozostáva z dvoch typov dlaždíc, ktoré, ak by sa striedali, alebo ak by sa jedna z nich nepoužila, vytvorili by periodickú mozaiku. Teda z množiny základných dlaždíc neperiodickej mozaiky môžeme vytvoriť periodickú mozaiku, kdežto z množiny základných dlaždíc aperiodickej mozaiky už toto nie je možné.

Neperiodické mozaiky tvoria veľkú skupinu mozaik. Ich neperiodicita ich robí prirodzenejšími a zaujímavejšími. Dôležitými skupinami týchto mozaik sú špirálne mozaiky a hierarchické mozaiky. Definícia špirálnej mozaiky je podobná periodickej. Podstatný rozdiel je v

množine, prípustných transformácií, ktorá je rozšírená o rotáciu. Klasickými dlaždicami špirálnych mozaik bývajú rovnoramenné trojuholníky, presnejšie trojuholníky, ktoré vzniknú rozkladom pravidelného n -uholníka na trojuholníky, alebo tvary, ktoré vzniknú určitými deformáciami ramien takýchto trojuholníkov.



Obr. 5 Príklad špirálnej mozaiky

Okolo špirálnych mozaik, tvare ich dlaždíc a rozmiestnenia v priestore by sa taktiež dalo popísať mnoho riadkov, ale keďže tieto mozaiky nie sú predmetom tejto práce, nebudem ich tu ďalej rozoberať.

2. Hierarchické mozaiky

2. 1. Definícia

Hierarchické mozaiky sú založené na princípe opakovaného rozkladu nosného priestoru hierarchickej mozaiky na podpriestory istého tvaru. Tieto podpriestory sa vyplňajú modifikovanými kópiami pôvodnej mozaiky, čím sa v mozaike znásobuje počet dlaždíc. V každej iterácii je pritom možné použitie iného rozkladu. Na začiatku je mozaika zložená z jedinej dlaždice, ktorá kompletne vyplní nosný priestor.

Formálne je možné hierarchickú mozaiku definovať indukciou. Mozaika pozostávajúca z jedinej dlaždice je hierarchická. Mozaiku, ktorá vznikne rozkladom nosného priestoru hierarchickej mozaiky na podpriestory, ktoré sú bezozvyšku vyplnené prípustnými transformáciami modifikovanými kópiami tejto mozaiky, nazveme taktiež hierarchickou.

Množina základných dlaždíc v hierarchickej mozaike je teda jednoprvková a jej tvar určuje nosný priestor mozaiky. Množina prípustných transformácií okrem posunutia však musí nutne obsahovať aj škálovanie, alebo inú, obsah znižujúcu transformáciu dlaždice. Totiž rozkladom nosného priestoru na podpriestory, musí každý tento podpriestor vyplniť bezozvyšku práve jedna kópia pôvodnej mozaiky (v súlade s prípustnými transformáciami). To taktiež obmedzuje použitie ľubovoľného rozkladu. Pokiaľ by sa v množine prípustných transformácií nenachádzala takáto transformácia, tak by to vyhovovalo jedinému rozkladu, a to jednoprvkovému, v ktorom by sa daný priestor rozložil sám na seba. Ide o triviálny rozklad, ktorý nám pôvodnú mozaiku ďalej nerozvinie, čo pre nás nie je zaujímavé.



Obr. 6 Príklad mozaiky vzniknutej rozkladom

Ako ukazuje obrázok, transformácie posunutia a škálovania postačujú na vytvorenie neperiodickej hierarchickej mozaiky. Základná dlaždica bola len rôzne škálovaná v osách x a y . Rozklad nosného priestoru reprezentuje pravidlo rozloženia dlaždíc. Ale my by sme mohli pokračovať ďalej. Nosným priestorom bol tentokrát obdĺžnik. Daný rozklad ho rozdelil len na ďalšie obdĺžniky. Na takto vzniknutý obrazec sa opäť dá aplikovať nejaký rozklad (rozkladové pravidlo), takže na celú mozaiku sa môžeme pozerat' ako na jedinú dlaždicu, ktorej tvar je rovnaký ako tvar pôvodnej dlaždice.

Hierarchické mozaiky tvoríme v jednotlivých iteráciách, ako napríklad fraktály. Začíname jedinou základnou dlaždicou. Môžeme povedať, že to je hierarchická mozaika nultej iterácie. Po aplikovaní prvého rozkladového pravidla dostávame mozaiku prvej iterácie, po druhom rozklade druhej iterácie atď.

2. 2. Rozkladové pravidlá a hierarchia

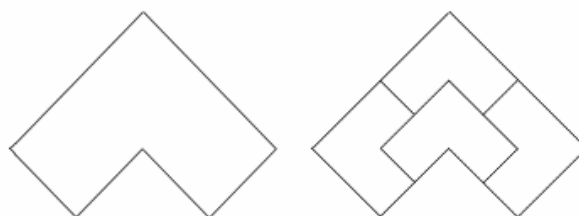
Definícia rozkladového pravidla. Majme množinu prípustných transformácií a tvar základnej dlaždice. Úplný disjunktný rozklad priestoru daného tvarom základnej dlaždice nazveme rozkladovým pravidlom, ak všetky jeho prvky (tj. podpriestory) budú kompletne vyplnené práve jednou kópiou nejakej hierarchickej mozaiky, modifikovanej prípustnými transformáciami, ktorej nosný priestor popisuje tvar základnej dlaždice. Ak navyše každý prvok rozkladu nesie so sebou informáciu o parametroch týchto transformácií, hovoríme, že tento rozklad tvorí rozkladové pravidlo pre hierarchickú mozaiku.

Definícia hierarchie. Hierarchiou je každá taká ľubovoľná množina rozkladových pravidiel, ktorej prvky sa vzťahujú k určitému typu nosného priestoru a určitým daným transformáciám. Čiže napríklad rozkladové pravidlo pre nejaký 2D útvar nemôže byť v jednej hierarchii s rozkladovým pravidlom pre nejaký 3D objekt a pod.

Bližšie sa budeme prirodzene zaujímať o 2D mozaiky, tj. mozaiky, ktorých nosným priestorom bude nejaký 2D uzatvorený útvar. Väčšinou pôjde o obdĺžniky alebo trojuholníky. Vymyslieť hierarchiu pre kruh môže byť obtiažne. Keďže nás zaujímajú výlučne konečné rozklady, so samotnou transformáciou škálovania by sme neboli schopný nájsť nejaký rozklad, ktorý by rozložil kruhový priestor na konečný počet kruhových podpriestorov. Množina prípustných transformácií by teda musela obsahovať transformáciu prevádzajúcu kruh na nejaký vhodnejší tvar, ktorý by spolu s ostatnými transformáciami mohol vyplniť kruhový obsah (napr. transformácia

prevádzajúca kruh na štvrtkruh a rotácia po 90-tich stupňoch). Všeobecne ide o rôzne morfoviacie transformácie, ktoré ovplyvňujú ako tvar, tak aj obsah dlaždice. A keďže hierarchické mozaiky často tvoríme v niekoľkých iteráciách, sú z praktického hľadiska kruhové hierarchické mozaiky výpočtovo ďaleko náročnejšie. Preto sa obmedzíme na vhodnejšie útvary a jednoduchšie transformácie. Bude to hlavne kroková rotácia po 90-tich stupňoch, preklápanie okolo osi x alebo y a diskrétno škálovanie.

Definícia diskrétno škálovania. Pod diskrétnym škálovaním rozumieme transformáciu, ktorá znásobuje dĺžku objektu v smeroch základných ôs súradnicovej sústavy, pričom koeficienty zväčšenia resp. zmenšenia sú určené racionálnym číslom. V prípade hierarchických mozaik dochádza iba k zmenšovaniu dlaždíc, takže uvedené koeficienty musia byť menšie, nanajvýš rovné jednej.



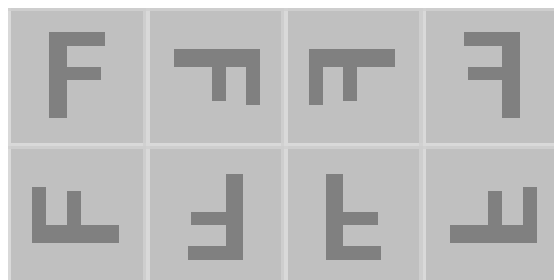
Obr. 7 Základná dlaždica a rozkladové pravidlo

Rozkladové pravidlo na obrázku 7 využíva spomínanú krokovú rotáciu a diskrétno škálovanie. Veľkosť dlaždíc tu v každej osi skraccujeme na polovicu a potom ich vhodným natočením zložíme do pôvodného tvaru.

Kroková rotácia a diskrétno škálovanie však nie sú jedinými jednoduchými transformáciami, ktorými sa budeme ďalej zaoberať. Ešte by som spomenul preklápanie. Keď hovoríme o dlaždici, musíme rozpoznať ako jej tvar tak aj obsah. Totiž tvar mnohých dlaždíc býva často súmerný minimálne podľa jednej osi (viď dlaždicu na obr. 7). Potom by sa mohlo zdať, že preklápanie podľa tejto osi je zbytočné. Ak však uvažujeme aj obsah dlaždice, ktorý reprezentuje istý vzor mozaiky, ten súmerný byť nemusí a aplikácie preklápania nám môžu priniesť veľmi pôsobivé výsledky.

V ďalšom texte sa obmedzíme na základné dlaždice tvaru štvorca a na už spomínané jednoduché transformácie. Pre preklápanie bude stačiť, ak budeme uvažovať iba preklápanie okolo osi y . Preklopenie obrazu okolo osi x dostaneme ako preklopenie okolo osi y a následnou rotáciou

o 180 stupňov. Ohliadnuc od škálovania, môžeme teda jedinú základnú dlaždicu použiť len ôsmymi rôznymi spôsobmi. Čo nám umožňuje vytvoriť množinu ôsmich transformácií bez parametrov, ktoré sú ekvivalentné s množinou obsahujúcou rotáciu po 90-tich stupňoch a preklápanie.



Obr. 8 Osem spôsobov uloženia štvorcovej dlaždice

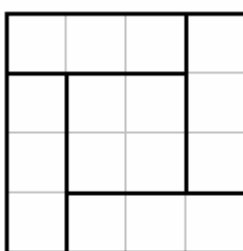
Aké môžeme použiť rozkladové pravidlá? Sme viazaný na mozaiky tvaru štvorca a vyššie spomínané prípustné transformácie. Tieto transformácie tentoraz používame na celú mozaiku, tj. aplikujeme ich na každú dlaždicu mozaiky. Transformácie rotácie po 90-tich stupňoch a preklápanie nemenia tvar mozaiky. Jediná transformácia, ktorá ho môže zmeniť je diskkrétne škálovanie. V hierarchických mozaikách, ako už bolo spomínané, dlaždice iba zmenšujeme. Pričom zmenšenie v osi jednej nemusí byť rovnaké, ako v osi druhej. A teda po použití tejto transformácie môžeme dostať tvar obdĺžnika. Nič iné sa pomocou týchto transformácií nedá vytvoriť.

Rozkladové pravidlo preto musí pozostávať z rozkladu, ktorý rozloží štvorec na množinu menších štvorcov a obdĺžnikov. Každý prvok tohto rozkladu bude niesť informáciu o použití rotácie alebo preklopenia dlaždice. Ako už bolo spomenuté, akékoľvek kombinácie týchto dvoch transformácií sa dajú realizovať jediným použitím jednej z vyššie uvedených ôsmich transformácií bez ďalších parametrov.

Na rozkladové pravidlo sa teda dá pozeráť ako na mozaiku. Je to množina nejakých dlaždíc rozmiestnených do akéhosi základného tvaru. V našom prípade je množina základných dlaždíc osemprvková a množina prípustných transformácií obsahuje okrem posunutia len diskkrétne škálovanie. Tým zabezpečíme, že každá dlaždica rozkladového pravidla nám dá informáciu o použitých transformáciách rotácie a preklápania. Použitie transformácií posunutia a diskkrétneho škálovania vyčítame so samotného rozkladu.

Celá tvorba hierarchických mozaik sa tak dá previesť cez jedinú operáciu, a to aplikácie rozkladového pravidla na mozaiku. A keďže rozkladové pravidlo je taktiež mozaikou, hovoríme tu o aplikovaní jednej mozaiky na druhú.

Tvorba rozkladových pravidiel. Keďže rozkladové pravidlo je mozaikou, ktorej nosným priestorom je štvorec, tak týmto pravidlom podľa definície mozaiky bude nejaký rozklad štvorca. Ten navrhne nasledovne. Štvorec rozdelíme na $m \times n$ rovnakých obdĺžnikov. Dostávame akúsi mriežku, ktorú využijeme na definovanie hraníc dlaždíc rozkladového pravidla, vid' obr. 9.



Obr. 9 Tvorba rozkladového pravidla

Definícia rozmeru rozkladového pravidla. Nech rozkladové pravidlo má tvar štvorca o strane dĺžky 1. Potom rozmerom rozkladového pravidla budeme rozumieť také minimálne prirodzené číslo m , pre ktoré je dĺžka každej dlaždice rozkladového pravidla v osi x presne i/m a dĺžka každej dlaždice v osi y presne j/m , kde i a j sú nejaké prirodzené čísla rôzne od nuly. Tieto čísla i a j určujú taktiež relatívnu veľkosť dlaždice.

Poznámka: Keďže rozkladové pravidlo môžeme interpretovať ako mozaiku, podobne môžeme vysloviť i definíciu rozmeru mozaiky.

Ako budeme dlaždice, resp. mozaiky škálovať? Do každého prvku rozkladu v rozkladovom pravidle musíme vložiť práve jednu kópiu mozaiky predchádzajúcej iterácie. Ak hovoríme o použití nejakej transformácie na mozaiku, budeme tým myslieť aplikovanie tejto transformácie na každú jej dlaždicu. Škálovanie budeme vykonávať až po aplikovaní rotácie prípadne preklápania mozaiky. Koeficienty škálovania pre os x budú i/m a pre os y j/m . Kde m je rozmer rozkladového pravidla.

Rozkladové pravidlá netreba voliť zbytočne zložitú. Pokiaľ sa dá nejaké rozkladové pravidlo vytvoriť pomocou iných, jednoduchších pravidiel, tak je vhodnejšie do hierarchie zaradiť skôr tieto pravidlá, ako to zložitejšie. Pravidlá by mali v sebe obsahovať určitú symetriu. Treba si uvedomiť, že práve tieto pravidlá určujú spájanie obrazcov do výslednej mozaiky.

3. Návrh a realizácia programového diela

Editor hierarchických mozaik bol kompletne vyvinutý v prostredí Borland Delphi 4. Vybral som si ho preto, lebo mám s ním asi najviac skúseností a je to vývojové prostredie pod operačným systémom platformy MS Windows. Navyiac vyššiu verziu Delphi (v čase dokončenia projektu verzia 7) sa mi nepodarilo nainštalovať na mojom počítači, na jedinom, na ktorom som mal možnosť pracovať. Tj. pentium 120 MHz, 16 MB RAM s operačným systémom MS Windows 95.

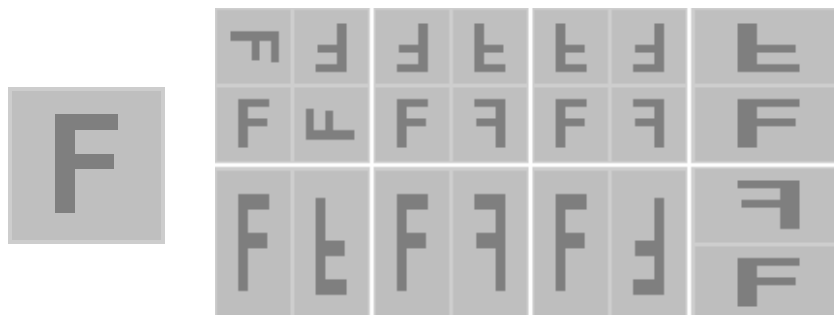
3. 1. Verzia 1

Prvá verzia editora hierarchických mozaik obsahovala len napevno dané rozkladové pravidlá, ktoré boli prebraté z článku pána Glassnera [1]. Tj. verzia 1 umožňovala vytvárať hierarchické mozaiky len pod jedinou hierarchiou. Taktiež tu nebola rozvinutá nejaká dátová štruktúra mozaiky, žiadna pamäť postupu jej tvorby a už vôbec nie jej editácie. Celá mozaika bola uložená v jedinej bitmapovej dátovej štruktúre. Program umožňoval načítanie bitmapy vo formáte bmp a to považoval za hierarchickú mozaiku nulte iterácie. Bitmapa vo všeobecnosti nemusí byť tvaru štvorca. Editor ju však bez akéhokoľvek varovania ureže do požadovaného tvaru.

Tvorba hierarchickej mozaiky. Hierarchická mozaika sa tvorí aplikáciou rozkladových pravidiel. Tieto pravidlá využívajú nejaké transformácie, v našom prípade sú to rotácia o 90° , rotácia o 180° , rotácia o 270° a preklopenie okolo osi y . Diskrétné škálovanie je závislé na tvare rozkladového pravidla, preto bude aplikované ako posledné.

Aplikácia rozkladového pravidla. Editor si aktuálnu iteráciu mozaiky ukladá vo forme dvoch bitmáp. Na začiatku je v oboch bitmapách základná dlaždica. Prvá bitmapa je v prvom kroku okopírovaná toľko krát, z koľkých prvkov pozostáva rozkladové pravidlo. Na každú kópiu sú aplikované príslušné transformácie rotácie a preklápania. Až potom dochádza k škálovaniu. V predchádzajúcej kapitole bolo spomenuté, že každú kópiu mozaiky aktuálnej iterácie musíme zoškálovať i/m –krát v osi x a j/m –krát v osi y . Čísla i a j závisia od rozmerov jednotlivých dlaždíc v rozkladovom pravidle, tj. pre každú dlaždicu môžu byť iné. A každá kópia je zviazaná s určitou dlaždicou rozkladového pravidla. Algoritmus teda každú kópiu zväčší i -násobne v osi x a j -násobne v osi y , v závislosti od veľkosti súvisiacej dlaždice v rozkladovom pravidle. Potom sa už modifikované kópie pôvodnej mozaiky poskladajú podľa rozkladového pravidla. Vzniknutá

bitmapa je teda m násobne väčšia ako bitmapa predchádzajúcej iterácie. Do druhej bitmapy sa potom uloží zmenšená kópia, ktorá sa zobrazí a ktorú je potom možné ďalej škálovať.



Obr. 10 Základná dlaždica a pevne dané rozkladové pravidlá verzie 1.

Tieto rozkladové pravidlá boli prevzaté z článku pána Glasnera, ako už bolo spomínané, preto by som zachoval aj ich značenie. Prvý riadok, zľava sú to $p4$, $p2mg$, $p2mm$ a $reflectY$. Druhý riadok potom $halfslideX$, pm , $r2$ a pg .

Neefektivita tohoto riešenia spočíva vo veľkom množstve operácií nad bitmapami. A to nad bitmapami, ktoré sú každou iteráciou štvornásobne väčšie (všetky pravidlá v použitej hierarchii majú rozmer 2). Totiž pri aplikovaní nasledujúceho pravidla sa toto aplikuje na prvú (nezmenšenú) bitmapu mozaiky. Preto bolo nevyhnutné prísť s iným, možno efektívnejším riešením, ktoré okrem iného poskytuje väčšie možnosti.

3. 2. Finálna verzia

3. 2. 1. Celková stavba projektu

Ako bolo spomínané, Editor hierarchických mozaik vznikol ako samostatný projekt vo vývojovom prostredí Borland Delphi 4. Pozostáva zo šiestich jednotiek, päť z nich je zviazaných s hlavným formulárom aplikácie, dialógovými a informačnými formulárami. Posledná jednotka *htiles*, pozostáva z definície samotnej dátovej štruktúry mozaiky a z procedúr realizujúcich určité transformácie s bitmapami a touto štruktúrou. Jednotka *htiles* obsahuje aj významnú procedúru *ApplyTile*, ktorá aplikuje na dátovú štruktúru mozaiky nejaké rozkladové pravidlo.

Jednotka definujúca hlavné okno aplikácie sa volá *main*. Pozostáva hlavne z objektov knižnice VCL (Visual Component Library), ako sú tlačítka, hlavné menu, skrolovacie lišty a podobne.

Odtiaľto sú volané takmer všetky procedúry z jednotky *htiles*. Jednotka *main* taktiež obsahuje procedúry vykreslenia mozaiky a neskôr som k nej premiestnil z *htiles* aj procedúry škálovania.

Aplikácia obsahuje dialógové okná načítania a uloženia súboru, tie sú však súčasťou VCL. Dialógové okná definované v ďalších dvoch jednotkách slúžia na orezávanie bitmapy, pokiaľ ako základná dlaždica bola vybraná bitmapa s nezhodnou šírkou a výškou a výber rozkladového pravidla pri nejakých operáciách nad príkazovým zásobníkom, čo bude spomenuté ďalej.

Posledné dve jednotky definujú informačné okná *Ovládanie* a *O programe*.

3. 2. 2. Základné princípy

Editor hierarchickým mozaikám by však nemal byť obmedzený na pár rozkladových pravidiel. Sú tu predsa obrovské možnosti aj napriek istým obmedzeniam, ktoré sme spomínali v časti 2. Už pri programovaní prvej verzie som si uvedomoval, že mozaiku treba interpretovať nejakou vhodnou dátovou štruktúrou. Ďalej hierarchická mozaika, to je aj postup, ktorým vznikala. Postupnosť rozkladových pravidiel z množiny nejakej hierarchie. Možnosť vrátiť sa v tomto postupe späť, čo vo verzii 1 nebolo možné. Vytvoriť si vlastnú hierarchiu s vlastnými rozkladovými pravidlami. To by malo byť minimum, čo má editor hierarchických mozaik poskytovať.

Pri vývoji verzie 1 som ešte netušil, že rozkladové pravidlá môžu mať štruktúru mozaiky, dokonca štruktúru hierarchickej mozaiky prvej iterácie. Toto zistenie malo za následok, že rovnaká dátová štruktúra pre mozaiku môže byť použitá i na rozkladové pravidlo. Tu je výpis definície dátovej štruktúry hierarchickej mozaiky:

```
Segment = record  
X, Y, ExpX, ExpY: word;  
Typ: Stav;  
end;
```

```
HTile = record  
bgrnd: TColor;  
ResOfTile, ResOfBmp: word;  
bmp: array [0..7] of TBitmap;  
Item: array of Segment;  
end;
```

Jednotlivé položky štruktúry *HTile*:

- bgrnd* - Farba pozadia mozaiky. Podrobnejšie to bude vysvetlené ďalej.
- ResOfTile* - Aktuálny rozmer mozaiky.
- ResOfBmp* - Strana štvorca bitmapy základnej dlaždice v pixeloch.
- bmp* - pole ôsmich bitmáp, ktoré zodpovedajú ôsmim rôznym kombináciám transformácií rotácie a preklápania.
- Item* - dynamické pole reprezentujúce všetky dlaždice v mozaike.

Štruktúra *Segment* popisuje konkrétnu dlaždicu:

- X, Y* - Súradnice ľavého horného rohu dlaždice v mriežke určenej rozmerom mozaiky (*ResOfTile*).
- ExpX, ExpY* - Relatívna veľkosť dlaždice vzhľadom k rozmeru mozaiky. ($1 \sim 1/m$)
- Typ* - Jedna z hodnôt: R0, R90, R180, R270, MR0, MR90, MR180, MR270. Vysvetlené ďalej.

Uvedená štruktúra dokonca nepožaduje, aby jednotlivé dlaždice museli vyplňať celý priestor štvorcového obsahu. Môžu vzniknúť medzery, čo je v rozpore s uvedenou definíciou mozaiky. Preto túto medzeru môžeme chápať ako novú dlaždicu s jednofarebným obsahom. Práve farba tejto dlaždice je daná atribútom *bgrnd*. Pre úplnosť treba dodať, že tieto medzery, je možné vyplniť dlaždicami štvorcového alebo obdĺžnikového tvaru.

V danej dátovej štruktúre je ďalej možné prekryvanie dlaždíc. To je opäť v rozpore s definíciou mozaiky. Korigovať to môžeme tak, že do množiny prípustných transformácií zavedieme aj orezávanie dlaždíc. Ak máme dve dlaždice, ktoré sa prekývajú, tak vždy je orezávaná dlaždica tá, ktorá má v poli *Item* menší index. Je to dané tým, že pri vykresľovaní mozaiky vykresľujeme dlaždice v poradí, v akom sú uložené v poli *Item* a teda dlaždica s vyšším indexom prekresľuje dlaždicu s nižším, čím ju orezáva.

Tieto vlastnosti len vychádzajú z možností navrhutej dátovej štruktúry. Prvá vlastnosť dokonca umožňuje vybudovať mozaiku s dvojprvkovou množinou základných dlaždíc. Mohlo by sa zdať, že toto je zasa v rozpore s definíciou hierarchickej mozaiky. Ale stačí, len zmeniť uhol pohľadu. Na novú dlaždicu sa môžeme pozeráť ako na dlaždicu, ktorá vznikla z jedinej základnej dlaždice, na ktorú aplikujeme novú transformáciu. Transformáciu, ktorá zafarbí obsah celej dlaždice danou farbou.

Aby sme to zhrnuli, uvedené možnosti editora hierarchických mozaik naozaj umožňujú tvorbu hierarchických mozaik s jednoprvkovou množinou základných dlaždíc a transformáciami posunutia, rotácie po 90°, preklápania, diskrétného škálovania, ale aj orezávania a zafarbovania dlaždice. Použitie jednotlivých transformácií závisí výlučne od rozkladových pravidiel, takže posledné dve transformácie vôbec nemusíme použiť.

Čo presne znamenajú atribúty X , Y , $ExpX$ a $ExpY$. X a Y sú súradnice ľavého horného rohu dlaždice. Môžu nadobúdať len hodnoty 0 až $m-1$, kde m je rozmer mozaiky, alebo rozkladového pravidla. Tieto súradnice sa teda viažu k pomyselnéj mriežke veľkosti $m \times m$. K tej istej mriežke sa viažu aj hodnoty $ExpX$ a $ExpY$. Obe musia byť minimálne rovné jednej a vyjadrujú veľkosť dlaždice v tejto mriežke.

Popis rozkladového pravidla v uvedenej dátovej štruktúre. Oproti samotnej hierarchickej mozaike tu sú nejaké drobné rozdiely. Prvým je atribút *bgrnd*, ktorý tu nemá žiaden význam. Bolo by síce zaujímavé, vytvoriť pravidlá s rôzne farebnými medzerami, ale myslel som, že by bolo vhodnejšie tento atribút skôr zviazať so samotnou mozaikou. Mozaikou s istou základnou dlaždicou, ku ktorej by bolo možné túto farbu vhodne zladiť.

Druhým nepodstatným a tiež nedefinovaným atribútom v rozkladovom pravidle je pole ôsmich bitmáp *bmp*. Tieto bitmapy sa generujú len pre mozaiku súčasne s definíciou základnej dlaždice, tj. s načítaním bitmapy, ktorá má reprezentovať túto základnú dlaždicu, čím zefektívňujú hlavnú aplikáciu rozkladového pravidla na mozaiku. Nemusí sa tu počítať rotácia a preklápanie bitmapy, stačí len okopírovať jednu z už transformovaných bitmáp.

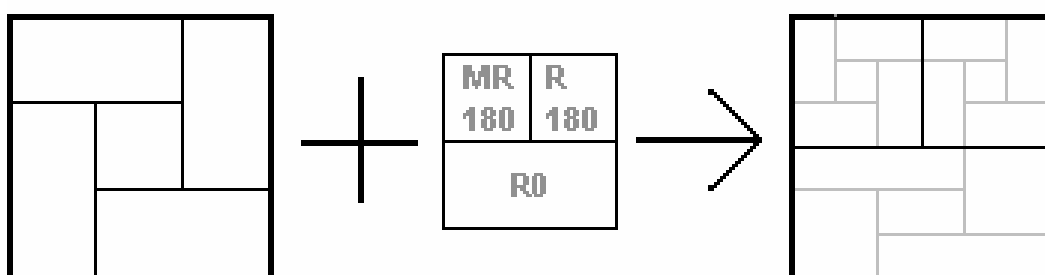
Každá dlaždica, či už rozkladového pravidla, alebo aktuálnej mozaiky obsahuje atribút *Typ*. Tento atribút, ako už bolo spomenuté obsahuje jednu z ôsmich hodnôt: R0, R90, R180, R270, MR0, MR90, MR180 alebo MR270. Ide o kódy transformácií rotácie a preklápania. Tu je výpis ich významu:

- R0** - Žiadna rotácia ani preklopenie.
- R90** - Rotácia o 90° .
- R180** - Rotácia o 180° .
- R270** - Rotácia o 270° .
- MR0** - Preklopenie okolo osi x .
- MR90** - Preklopenie okolo osi x a následná rotácia o 90° .
- MR180** - Preklopenie okolo osi x a následná rotácia o 180° .
- MR270** - Preklopenie okolo osi x a následná rotácia o 270° .

Samotná aplikácia rozkladového pravidla sa teraz neprevádza na bitmape, ale na samotnej štruktúre mozaiky. Táto štruktúra pozostáva hlavne z dynamického poľa reprezentujúceho počet dlaždíc v mozaike. Je zrejmé, že aplikáciou rozkladového pravidla o n dlaždiciach sa počet dlaždíc v mozaike z n -násobí.

Presný postup: Máme aktuálnu hierarchickú mozaiku, na ktorú chceme aplikovať určité rozkladové pravidlo. Ako prvé vytvoríme prázdnu výslednú mozaiku. Potom prechádzame poľom *Item* v rozkladovom pravidle. Prechádzame teda všetkými jeho dlaždicami. V každom prechode

vytvoríme kópiu pôvodnej mozaiky. Na základe atribútu *Typ* dlaždice rozkladového pravidla otrotujeme príp. preklopíme kópiu mozaiky. Tj. každej jej dlaždici upravíme atribúty *Typ*, *ExpX* a *ExpY*. Na základe atribútu *ExpX* a *ExpY* dlaždice rozkladového pravidla zoškálujeme transformovanú kópiu mozaiky. Tj. znásobíme atribúty *ExpX* a *ExpY* a upravíme súradnice dlaždíc *X* a *Y* jej dlaždiciam. Nakoniec presunieme všetky dlaždice kópie na výsledné miesto podľa polohy dlaždice v rozkladovom pravidle. Všetky operácie sa prevádzali v podstate len v poli *Item* kópie mozaiky. Takže už stačí iba pridávať toto pole *Item* na koniec poľa *Item* výslednej mozaiky. Ešte sa upraví atribút *ResOfTile*, ktorý sa znásobí s rozmerom rozkladového pravidla a výsledná mozaika je kompletná.



Obr. 11 aplikácia rozkladového pravidla. Zľava doprava: pôvodná mozaika, rozkladové pravidlo a výsledná mozaika.

Vykreslenie samotnej mozaiky. Obraz mozaiky sa jednoducho poskladá z jednotlivých dlaždíc. Transformácie rotácie a preklápania sa už neprevádzajú. Podľa atribútu *Typ* konkrétnej dlaždice, sa len vyberie jedna z ôsmich bitmáp v poli *bmp*, podľa *ExpX* a *ExpY* sa patrične zväčší a vykreslí do dátovej štruktúry bitmapy. Zmenšuje sa až nakoniec, a to podľa premennej *scale*. V tejto premennej je uložené koľkonásobne sa výsledný obraz zmenší. Užívateľ si má totiž možnosť zväčšiť mozaiku až *m* násobne, kde *m* je rozmer tejto mozaiky. Jediné, čo sa počas vykresľovania mozaiky teda vykonáva, je škálovanie. Zväčšujú sa iba jednotlivé bitmapy dlaždíc (pokiaľ je to potrebné) a zmenšuje sa až výsledný obraz.

Premenná *scale* je na začiatku rovná jednej. Aplikáciou rozkladového pravidla rozmeru *m* sa z *m*-násobí. Premenná *scale* určuje koľko násobne sa má výsledný obraz zmenšiť. Pokiaľ si užívateľ zmenší mozaiku, premenná *scale* sa zdvojnásobí a obraz sa vykreslí dvojnásobne kratší v oboch osiach. Pokiaľ si užívateľ obraz zväčší, premenná *scale* sa zmenší celočíselne na polovicu. Minimálna hodnota premennej *scale* je 1, čiže mozaika sa nedá zväčšiť viac ako *m*-násobne.

Naopak obmedzenie pre zmenšenie vychádza zo samotného algoritmu zmenšovanie bitmapy, ktorú môže zmenšiť menej ako 256 krát z každej strany.

Ďalšou možnosťou je pamäť postupu tvorby mozaiky. Jednu mozaiku je možné vytvoriť nad istou hierarchiou. Hierarchiu tvorí množina rozkladových pravidiel. Editor umožňuje použitie hierarchií s maximálne 12-tímy pravidlami. Túto hierarchiu interpretuje ako 12-prvkové pole záznamov, v ktorých jedným atribútom je meno rozkladového pravidla a druhým dátová štruktúra hierarchickej mozaiky, v ktorej je uložené.

Použitím určitého rozkladového pravidla sa ukladá jeho identifikačné číslo (index, pod ktorým je v hierarchii vedené) do tzv. príkazového zásobníka, ktorý je interpretovaný dynamickým poľom. Editor umožňuje prevádzať niektoré operácie nad týmto zásobníkom, čím je možné meniť postup tvorby samotnej mozaiky.

Najdôležitejšou operáciou nad príkazovým zásobníkom je asi možnosť vracat' sa v postupe späť. Odobrať posledné použité pravidlo. Vďaka zásobníku vieme, ktoré pravidlo sme použili posledné. Teda poznáme počet jeho dlaždíc, označme ho m . Ak má aktuálna mozaika n dlaždíc, tak pred použitím tohto pravidla ich mala práve n/m . Stačí v mozaike, v poli *Item* nechať len prvých n/m dlaždíc. Podľa prvej dlaždice a posledne použitého pravidla tieto dlaždice upraviť. Tj. prípadne ich zoškálovať späť, presunúť späť a orotovať späť. Nesmieme zabudnúť na odstránenie indexu posledne použitého pravidla zo zásobníka.

Viacnásobným opakovaním tohto postupu sa môžeme vrátiť k akejkoľvek iterácii, prípadne až k základnej dlaždici. Ak si navyše zapamätáme postupnosť rušených príkazov, môžeme zrušiť použitie ľubovoľného pravidla a aplikovať automaticky všetky, čo boli po ňom, zameniť použitie ľubovoľného pravidla za iné alebo vložiť kdekoľvek do postupu nové pravidlo.

3. 2. 3. Realizácia transformácií

V tejto kapitole sa zameriam na realizáciu jednotlivých transformácií realizovaných v jednotke *htile* a výpočtových algoritmov. Budem ich uvádzať v zjednodušenom zápise a následným popisom. Jednotlivé procedúry tejto programovej jednotky by som rozdelil na dve časti. Jedna sa zaoberá samotnými bitmapami, druhá pracuje s dátovou štruktúrou mozaiky.

Algoritmy na prácu s bitmapami:

Bitmapa je určená farebnými hodnotami jednotlivých pixelov. Tieto hodnoty sú uložené v dvojrozmernom poli dĺžky $n \times n$ (tj. 0 až $n-1$ v každej osi), keďže pracujeme výlučne so štvorcovými bitmapami. Farebné hodnoty chápeme ako trojicu (R, G, B) , kde R, G, B sú prirodzené

čísla od 0 po 255. Tieto čísla reprezentujú jednotlivé intenzity troch základných farieb: červenej, zelenej a modrej. Ich kombináciou je možné vytvoriť ľubovoľnú ľudským okom vnímateľnú farbu alebo odtieň (vlastne konkrétne len čosi nad 16.7 milióna). Farebnú hodnotu obrazu na súradniciach x, y budeme značiť $p(x, y)$.

Preklopenie okolo osi x :

Algoritmus:

Pre každé $j = 0$ až $(n-1)/2$ rob: pre každé $i = 0$ až $n-1$ vymeň $p(i, j)$ s $p(i, n-j)$.

Popis:

V tejto transformácii sa vymieňajú riadky. Preklápajú sa okolo stredu, čiže prvý riadok sa preklopí na posledný, druhý na predposledný atď. Operácia je symetrická, čiže keď prvý riadok kopírujeme na posledný, súčasne môžeme posledný vložiť na prvý.

Preklopenie okolo osi y :

Algoritmus:

Pre každé $i = 0$ až $(n-1)/2$ rob: pre každé $j = 0$ až n vymeň $p(i, j)$ s $p(n-i-1, j)$.

Popis:

To isté ako v minulej transformácii, ibaže tu sa pracuje so stĺpcami.

Rotácia o 90° :

Algoritmus:

Pre každé $j = 0$ až $(n-1)/2$ rob: pre každé $i = j$ až $n-j-2$ rob:

do $p(i, j)$ vlož $p(n-j-1, i)$,

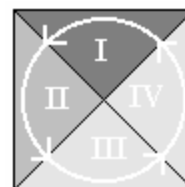
do $p(n-j, i)$ vlož $p(n-i-1, n-j-1)$,

do $p(n-i-1, n-j-1)$ vlož $p(j, n-i-1)$,

do $p(j, n-i-1)$ vlož pôvodné $p(i, j)$.

Popis:

Ide tu vlastne o akúsi cyklickú výmenu. Celý obraz je rozdelený na 4 trojuholníkové časti. Každý pixel z časti prvej sa presunie na svoje miesto do časti druhej, odtiaľ prehodí jeden pixel do časti tretej, až napokon jeden pixel z časti IV príde na uvoľnené miesto v prvej časti.



Obr. 12

Rotácia o 180° :

Algoritmus:

Pre každé $j = 0$ až $(n-1)/2$ rob: pre každé $i = 0$ až $n-1$ vymeň $p(i, j)$ s $p(n-i-1, n-j-1)$.

Popis:

Rotácia o 180° je to isté ako preklopenie obrazu okolo jeho stredy. Tj. dá sa to realizovať ako preklopenie okolo osi x a následne okolo osi y .

Rotácia o 270° :

Algoritmus:

Pre každé $j = 0$ až $(n-1)/2$ rob: pre každé $i = j$ až $n-j-2$ rob:

do $p(i, j)$ vlož $p(j, n-i-1)$,

do $p(j, n-i)$ vlož $p(n-i-1, n-j-1)$,

do $p(n-i-1, n-j-1)$ vlož $p(n-j-1, i)$,

do $p(n-j-1, i)$ vlož pôvodné $p(i, j)$.

Popis:

To isté ako pri rotácii o 90° , len v obrátenom poradí.

Zväčšenie r -krát v osi x a s -krát v osi y :

Algoritmus:

Zväčši bitmapu na $r.n \times s.n$ pixelov.

Pre každé $j = n-1$ až 0 rob: pre každé $i = n-1$ až 0 rob:

do $p(r.i+p, s.j+q)$ vlož $p(i, j)$ pre každé $p: 0 \leq p < r$ a pre každé $q: 0 \leq q < s$.

Popis:

V prvom rade musíme zväčšiť samotnú bitmapu. Každý pixel sa po škálovaní zväčší na $r \times s$ pixelov. Postupujeme zprava doľava a zdola nahor, aby sme si neprepisovali hodnoty, s ktorými sa ešte bude počítať.

Zmenšenie r -krát v osi x a s -krát v osi y :

Algoritmus:

Pre každé $j = 0$ až $(n-1)/s$ rob: pre každé $i = 0$ až $(n-1)/r$ rob:

do $p(i, j)$ vlož aritmetický priemer hodnôt $p(r.i+p, s.j+q)$, pre všetky kombinácie $p: 0 \leq p < r$ a $q: 0 \leq q < s$.

Zmenši bitmapu na $n/r \times n/s$ pixelov.

Popis:

Opačný postup. Do jedného pixelu sa vkladá $r \times s$ pixelov. Aritmetický priemer farebných hodnôt sa robí nasledovne. Každá farebná hodnota sa skladá z troch hodnôt. Pre každú hodnotu sa vypočíta aritmetický priemer a tieto tri aritmetické priemery tvoria novú farebnú hodnotu. Tento priemer sa však môže počítať iba z toľkých pixelov, koľko zvládne dátový typ *word*, čo je obmedzenie pre zmenšovanie obrazu.

Použitie bitmapových operácií v editori. Transformácie rotácie a preklápania okolo osi y sú použité len pri načítaní novej bitmapy ako novej dlaždice. Preklápanie okolo osi x nie je použité vôbec. Zmenšovanie bitmapy sa používa na zmenšenie výsledného obrazu pred samotným vykreslením, ktorí si zachováva rozlíšenie jednotlivých dlaždíc a teda jeho celkové rozlíšenie je $m.ResOfBmp$, kde m je rozmerom mozaiky. Zväčšovanie bitmapy. Nikdy sa nezväčšuje výsledný obraz. K zväčšovaniu dochádza len pri jeho tvorbe. Ten sa postupne skladá z dlaždíc uložených v poli *Item*. Pokiaľ nebolo použité pravidlo s dlaždicami iných rozmerov ako 1×1 , toto škálovanie je zbytočné, lebo celý obraz sa skladá iba z bitmáp z poľa *bmp*. Pokiaľ sa však v mozaike nachádza jediná dlaždica s *ExpX* alebo *ExpY* rôznym od jedničky, toto škálovanie sa prevádza. Hlavne preto je efektívnejšie používanie pravidiel s jednoduchými dlaždicami. Operácia zväčšenia bitmapy je vždy realizovaná na kópiu, ktorú si vytvorí z jednej z bitmáp v poli *bmp*.

Algoritmy na prácu s mozaikou:

Uvedené algoritmy pracujú s vyššie popísanou dátovou štruktúrou. Označme preto m , ako rozmer mozaiky a p ako pomocnú premennú. Ostatné značenie bude podľa atribútov dátovej štruktúry. Hlavný rozdiel medzi mozaikou a bitmapou je v tom, že kým bitmapa je vlastne dvojrozmerným poľom, mozaika je zoznamom dlaždíc. Preto i algoritmy, budú vyzerat' úplne inak. Napríklad tu nepôjde o výmenu dlaždíc, ako to bolo predtým s pixelami. Tu sa budú len meniť hodnoty jednotlivých atribútov každej dlaždice.

Preklopenie okolo osi x :

Algoritmus:

Pre každú dlaždicu rob: do Y vlož $ResOfTile-Y-ExpY$ a uprav Typ (podľa tabuľky 1).

Popis:

Čiže každú dlaždicu preklápame okolo stredu mozaiky. Okrem zmeny atribútu *Typ*, ktorým prevedieme akoby preklopenie obsahu dlaždice, musíme zmeniť i jej súradnice. Treba si

prítom uvedomiť, že každá dlaždica má istú veľkosť a obyčajným preklopením nejakého bodu dostaneme jeho presný zrkadlový obraz. Tj. z ľavého horného rohu dostaneme ľavý spodný. Ale my potrebujeme súradnicu toho horného. Preto vo výpočte musíme odčítať i veľkosť dlaždice v osi y , teda $ExpY$. X -ové súradnice, ako aj veľkosti dlaždíc v oboch osiach sa touto operáciou nemenia.

Pred operáciou	R0	R90	R180	R270	MR0	MR90	MR180	MR270
Po operácii	MR180	MR90	MR0	MR270	R180	R90	R0	R270

Tabuľka 1.

Preklopenie okolo osi y :

Algoritmus:

Pre každú dlaždicu rob: do X vlož $ResOfTile-X-ExpX$ a uprav Typ (podľa tabuľky 2).

Popis:

Analogicky ako minule, len namiesto y -ových súradníc sa menia súradnice x -ové.

Pred operáciou	R0	R90	R180	R270	MR0	MR90	MR180	MR270
Po operácii	MR0	MR270	MR180	MR90	R0	R270	R180	R90

Tabuľka 2.

Rotácia o 90° :

Algoritmus:

Pre každú dlaždicu rob:

vymeň $ExpX$ s $ExpY$,

do p vlož X , do X vlož Y a do Y vlož $m-p-ExpY$,

uprav Typ (podľa tabuľky 3).

Popis:

Táto transformácia mení proporcie dlaždice. Preto je potrebné vykonať výmenu hodnôt $ExpX$ a $ExpY$. Pri rotácii dlaždíc podobne, ako pri preklápaní nesmieme zabúdať na to, že potrebujeme súradnice ľavého horného rohu. Jeho orotovaním totiž dostávame ľavý dolný roh, preto je nutné vo výpočte pre Y ešte odčítať veľkosť dlaždice v osi y .

Pred operáciou	R0	R90	R180	R270	MR0	MR90	MR180	MR270
Po operácii	R90	R180	R270	R0	MR90	MR180	MR270	MR0

Tabuľka 3.

Rotácia o 180°:

Algoritmus:

Pre každú dlaždicu rob:

do X vlož $m-X-ExpX$ a do Y vlož $m-Y-ExpY$,

uprav Typ (podľa tabuľky 4).

Popis:

Rotácia o 180° sa dá chápať i ako preklopenie okolo stredu mozaiky. Proporcie dlaždíc sa nemenia, mení sa len Typ . Preklopením ľavého horného rohu tentoraz dostávame pravý spodný preto v oboch súradniciach po preklopení ešte odčítať samotné veľkosti dlaždice v daných osiach.

Pred operáciou	R0	R90	R180	R270	MR0	MR90	MR180	MR270
Po operácii	R180	R270	R0	R90	MR180	MR270	MR0	MR90

Tabuľka 4.

Rotácia o 270°:

Algoritmus:

Pre každú dlaždicu rob:

vymeň $ExpX$ s $ExpY$,

do p vlož Y , do Y vlož X a do X vlož $m-p-ExpX$,

uprav Typ (podľa tabuľky 5)

Popis:

Situácia je podobná ako pri Rotácii o 90°. Tentoraz orotovaním súradníc dostávame pravý horný roh dlaždice, preto od hodnoty X musíme ešte odčítať $ExpX$.

Pred operáciou	R0	R90	R180	R270	MR0	MR90	MR180	MR270
Po operácii	R270	R0	R90	R180	MR270	MR0	MR90	MR180

Tabuľka 5.

Zväčšenie r -krát v osi x a s -krát v osi y :

Algoritmus:

Pre každú dlaždicu rob:

do $ExpX$ vlož $r.ExpX$, do $ExpY$ vlož $s.ExpY$,

do X vlož $r.X$, do Y vlož $s.Y$.

Popis:

V prvom rade každá dlaždica zväčší svoju veľkosť, r -násobne v osi x a s -násobne v osi y .

Treba si však uvedomiť, že zväčšením sa menia aj súradnice každej dlaždice, pokiaľ sú nenulové. Atribút *Typ* ostáva tentoraz nemenný.

Presunutie mozaiky o r/m v osi x a s/m v osi y :

Algoritmus:

Pre každú dlaždicu rob: do X vlož $X+r$, do Y vlož $Y+s$.

Popis:

Triviálna operácia posunutia.

Využitie jednotlivých operácií. Prakticky takmer všetky sa využívajú počas aplikácie rozkladového pravidla a vracania sa v zásobníku. Teda s výnimkou preklápania okolo osi x , ktorú editor nevyužíva vôbec. Podrobný postup pri aplikácii rozkladového pravidla a odstránenia posledného použitého pravidla je popísaný v predchádzajúcej kapitole.

3.3. Problémy a riešenia

Najväčším problémom je pomalosť grafických výpočtov. Pred realizáciou finálnej verzie editora som sa snažil nájsť riešenie, ako obmedziť viacnásobné prechody bitových máp. Totiž v prvej verzii každé pravidlo pozostávalo z maximálne štyroch dlaždíc. Jeho aplikácia na mozaiku mala za následok, že pre každú dlaždicu rozkladového pravidla (dlaždice samotnej mozaiky neboli

nikde zaznamenané) sa počítalo prípadné preklopenie mozaiky, orotovanie mozaiky a tiež prípadné zväčšenie mozaiky. A napokon sa celá bitmapa (štvornásobne väčšia) následne zmenšovala. Pre úplnosť treba dodať, že v prípade rozkladového pravidla so štyrmi dlaždicami k žiadnemu zväčšovaniu pôvodnej mozaiky nedochádza. Iba pri rozkladových pravidlách s dvoma dlaždicami sa prevedie dvojnásobné predlžovanie jednej strany. Celkovo sa teda prechádzal pixelový priestor bitmapy pôvodnej mozaiky maximálne 12 krát. Pritom neškálovaná bitmapa mozaiky je každou iteráciou štvornásobne väčšia.

Aj preto tu bola nutnosť vytvoriť dátovú štruktúru mozaiky. Hierarchická mozaika sa predsa skladá len z jednej základnej dlaždice. Pri jej vykresľovaní sa mnohokrát používajú tie isté transformácie preklápania a rotácie. Preto vzniklo pole ôsmich bitmáp *bmp*, do ktorého sa hneď po načítaní bitmapy základnej dlaždice uloží osem kombinácií aplikácie preklápania a rotácie. Pri samotnom vykreslení mozaiky sa už nerotuje a nepreklápa žiadna bitmapa. Toto riešenie má však za následok dlhšie načítanie bitmapy, ktoré je cítiť pri bitmapách s väčším rozlíšením.

Pri vykresľovaní mozaiky oproti verzii 1 pribudlo skladanie jej výslednej bitmapy z bitmáp jednotlivých dlaždíc. Bitmapy týchto dlaždíc sa už len prípadne zväčšujú. Následné zmenšenie celého obrazu ostalo, rovnako ako aj ponechanie si nezmenšeného obrazu. To kladie so stúpajúcou iteráciou hierarchickej mozaiky aj väčšie nároky na pamäť.

So stúpajúcou iteráciou je aj dlhší výpočet mozaiky. Ani nie tak transformácia samotnej dátovej štruktúry, ako samotné vykreslenie je veľmi časovo náročné. Začal som sa síce zaoberať podporou DirectX v Delphi, ale nestihol som získať dostatočné množstvo skúseností a napokon by to znamenalo prerobiť celý projekt, čo by som už možno nestihol. A tak som skôr pridal na formulár komponentu sledujúcu priebeh výpočtu, aby užívateľ vedel, že sa niečo deje.

Ďalšie urýchlenie vykreslenia spočíva v možnosti vypnúť výsledné zmenšovanie obrazu. Pre tvorbu mozaik je vhodné nepoužívať príliš zložité a rozsiahle bitmapy, skôr sa zamerať na jednoduché vzory a používať bitmapy s rozlíšením niekoľko desiatok pixelov. Toto rozlíšenie sa behom napr. štyroch iterácií aj tak z 16-násobí. Vypnutím výsledného škálovania užívateľ nemusí výsledný obraz získaný z malej dlaždice naspäť zväčšovať.

Vykresľovanie teda prebieha v dvoch fázach. Skladanie obrazu z jednotlivých dlaždíc a škálovania výsledného obrazu. Priebeh výpočtu každej fázy má užívateľ možnosť vidieť na komponente, ktorá sa na okne aplikácie v tomto čase objaví.

Tiež vykreslenie mozaiky prešlo istými zmenami. Procedúra, ktorá toto vykonáva je volaná jedným parametrom, ktorý určuje, čo je potrebné vykonať. Pokiaľ aplikujeme rozkladové pravidlo, načítame novú bitmapu dlaždice, alebo upravíme príkazový zásobník, je potrebné vykonať obe

fázy, ak teda nie je zapnutá voľba neškálovať výslednú bitmapu. V prípade manuálneho škálovania, ktoré vykonáva užívateľ, nie je potrebné znovu bitmapu skladať, iba ju zmenšiť. A v prípade zmeny veľkosti okna aplikácie, či posunutia skrolovacích líšt, stačí prekresliť len už zoškálovaný obraz.

S výslednou uloženou bitmapou môžu byť problémy pri niektorých aplikáciách. Napríklad sa mi nepodarilo nejakú mozaiku vytvorenú editorom otvoriť vo Photoshope 4.0 ani Photoshope 6.0. Preto som to skúšal s windowsovským Paintbrushom, prehliadačom ACD See a Internet Explorerom. Všetky načítali bitmapu bez problémov. Preto by nemal byť problém následne uložiť ju v nejakom vhodnejšom formáte.

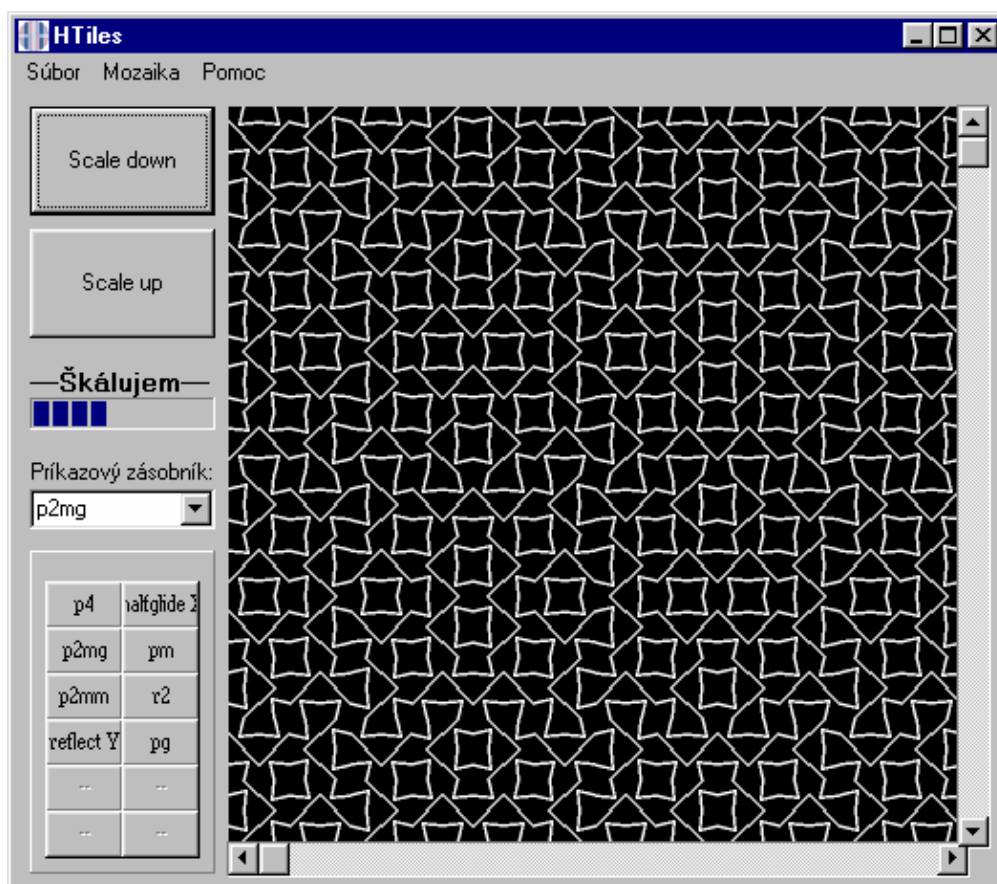
Ku koncu som sa rozhodol obohatiť program o niektoré klávesové skratky. Hlavne pri práci so zásobníkom, lebo sa mi zdalo trochu máťúce najskôr myšou označiť pravidlo v zásobníku a potom cez horné menu vybrať nejaký príkaz (zruš toto pravidlo, zameň za iné pravidlo a pod.). Preto hneď po označení príslušného pravidla je možné stlačiť napr. klávesu *Delete* a dané pravidlo sa zruší. A keď už som s tými skratkami začal, obohatil som nimi i niektoré ďalšie funkcie škálovania a otvorenia či uloženia bitmapy. Kompletný zoznam klávesových skratiek je vypísaný v kapitole 4. 7.

Ešte taký detail. Ďalším problémom, ktorý sa môže vyskytnúť je diakritika. Program komunikuje s užívateľom v slovenčine, takže v prípade nevhodného nastavenia systému sa nesprávne zobrazujú písmenká s dĺžňami a mäkčeňmi. Všimol som si to na počítačoch v jednej učebni.

4. Popis a ovládanie Editora hierarchických mozaik

4. 1. Popis prostredia

Editor hierarchických mozaik *HTiles* pozostáva zo spustiteľného súboru *htile.exe*, *default.hrh* a adresára *dlazdice* v ktorom musí byť bitmapa *default.bmp*. Súšťa sa prirodzene súborom *htile.exe*. Súbor *default.hrh* obsahuje základnú hierarchiu, ktorá sa automaticky načíta. V prípade, že tento súbor chýba, editor vyhlási chybu: „Zlyhalo otvorenie súboru default.hrh !!!“ a spustí editor bez načítania akejkoľvek hierarchie. V prípade, že chýba súbor *default.bmp* v adresári *dlazdice*, alebo tento súbor síce existuje, ale nie je to formát bmp, editor vypíše správu: „Nemôžem nahráť súbor default.bmp z adresára dlazdice !!!“ a vôbec sa nespustí. V tom prípade je nutné do tohto adresára nakopírovať nejakú bitmapu vo formáte bmp a premenovať ju na *default.bmp*. Malo by ísť o štvorcovú bitmapu, s malým rozlíšením (dôvod vid' kapitola 3. 3.).



Obr. 13 Vzhľad Editora hierarchických mozaik

Program by mal byť spustiteľný pod ľubovoľným operačným systémom MS Windows (všetky verzie som neskúšal). Hlavné okno aplikácie pozostáva z horného menu, vľavo sú zhora nadol tlačítka na zväčšenie a zmenšenie obrazu, v prípade vykresľovania grafické znázornenie postupu, príkazový zásobník a panel s tlačítkami jednotlivých rozkladových pravidiel, ktoré sú tu označované ako príkazy. Vpravo je zobrazovací priestor mozaiky.

Hlavné okno aplikácie je škálovateľné, tj. je možné do určitej miery meniť jeho veľkosť a maximalizovať ho. V prípade, že rozmer bitmapy je väčší ako rozmer priestoru na zobrazenie mozaiky, k tomuto priestoru pribudnú skrolovacie lišty, ktoré umožňujú bitmapu v tomto priestore posúvať.

4. 2. Stručný popis funkcií horného menu

Položka *Súbor*:

Využíva sa na prácu so súbormi a na ukončenie programu. Jednotlivé podpoložky:

- | | |
|---------------------------|---|
| <i>Otvor bitmapu</i> | - Načítanie bitmapy ako základnej dlaždice. |
| <i>Ulož bitmapu</i> | - Uloženie zobrazeného obrazu. |
| <i>Načítaj hierarchiu</i> | - Načítanie rozkladových pravidiel zo súboru. |
| <i>Koniec</i> | - Ukončenie programu. |

Položka *Mozaika*:

Ide najmä o prácu s príkazovým zásobníkom. Väčšina príkazov predpokladá výber nejakej položky z tohto zásobníka. Jednotlivé podpoložky menu *Mozaika*:

- | | |
|---------------------------------|--|
| <i>Reset</i> | - Návrat k základnej dlaždici. |
| <i>Zruš posledný príkaz</i> | - Zrušenie poslednej aplikácie rozkladového pravidla. |
| <i>Zruš vybraný príkaz</i> | - Zrušenie vybranej aplikácie rozkladového pravidla. |
| <i>Zruš po vybraný príkaz</i> | - Zrušenie všetkých aplikácií rozkladového pravidla až po určitú iteráciu. |
| <i>Zameň vybraný príkaz</i> | - Výmena rozkladového pravidla v postupe za iné. |
| <i>Vlož pred vybraný príkaz</i> | - Vloženie rozkladového pravidla do postupu. |
| <i>Farba medzery</i> | - Nastavenie farby prípadnej medzery v mozaike. |
| <i>Neškáluj</i> | - Zapnutie/vypnutie nezmenšovania výslednej mozaiky. |

Položka *Pomoc*:

- | | |
|-------------------|--|
| <i>Ovládanie</i> | - Výpis jednoduchého návodu k ovládaniu editora. |
| <i>O programe</i> | - Názov a verzia programu. |

4. 3. Načítanie a uloženie bitmapy

Pri načítaní si treba uvedomiť, že s touto bitmapou pracujeme ako so základnou dlaždicou. Pri jej načítaní sa automaticky prevádzajú operácie rotácie i preklápania, aby boli predpripravené pre vykresľovanie mozaiky z tejto dlaždice vznikutej. Preto sa nesnažte načítať bitmapu reprezentujúcu povedzme nejakú fotografiu definovanú stotisícami pixelov. Kým by editor tieto operácie s touto bitmapou previedol trvalo by to dlhšiu dobu a neskôr by sa s tým aj ťažšie pracovalo. Ako dlaždice sú skôr použiteľné menšie bitmapy s jednoduchým vzorom.

To, že sa načítava základná dlaždica neznamená zrušenie mozaiky. Pokiaľ máte nejakú vytvorenú a načítate novú bitmapu, tak dôjde k nahradeniu pôvodnej dlaždice novou a vykreslí sa celá mozaika s touto novou dlaždicou. Opäť ak načítate nejakú veľkú bitmapu a máte vytvorenú mozaiku vysokej iterácie, z množstvom dlaždíc, môžu nastať veľké problémy s pamäťou. Povedzme máte vytvorenú mozaiku piatej iterácie rozmeru 32. V tomto štádiu načítate bitmapu s rozlíšením 1024 x 1024. Editor vyrobí 8 bitmáp tohto rozlíšenia, a bude sa snažiť vyprodukovať výslednú bitmapu mozaiky. Len táto bitmapa bude mať rozlíšenie 32 768 x 32 768, čo by zabralo 3 GB pamäte. Preto sa neoplatí načítavať veľké bitmapy a keď chcete predísť tomu, aby sa nová bitmapa ihneď nahradila do vytvorenej mozaiky, treba túto mozaiku zrušiť príkazom *Reset* pod položkou *Mozaika* z horného menu.

Samotné načítanie vykonáme voľbou *Otvor bitmapu* pod položkou *Súbor* horného menu. Otvorí sa dialógové okno pre načítanie súborov zobrazujúce len súbory s príponou bmp. Otvoriť sa dá najviac jeden súbor.

Ako základná dlaždica musí mať táto bitmapa tvar štvorca. Pokiaľ toto načítaná bitmapa nespĺňa, otvorí sa dialógové okno, v ktorom si máte možnosť vybrať ako danú bitmapu orezať. Obrázok 14 ukazuje dialógové okno pre bitmapu, ktorej šírka je väčšia ako výška. Dialógové okno pre bitmapu, ktorá má naopak väčšiu výšku ako šírku má voľbu namiesto orezať zľava orezať zhora a namiesto orezať sprava orezať zdola.

Voľba orezať sprava, znamená, že výsledná bitmapa bude mať tvar štvorca, ostane len pravá časť bitmapy, ľavá sa ureže. Podobne pre orezávanie zľava, prípadne zhora a zdola. Nechať stred znamená orezať bitmapu rovnako z oboch strán, tj. orezanie jej pravého aj ľavého, resp. horného a dolného okraja.



Obr 14.

Uloženie bitmapy. Ukladá sa práve zobrazený obraz. Tj. pokiaľ je nejaký naškálovaný, uloží sa v takej veľkosti, v akej je práve zobrazený. Prirodzene, pokiaľ je vyobrazený obraz väčší ako zobrazovací priestor, uloží sa tento obraz celý.

Bitmapa sa ukladá voľbou *Ulož bitmapu* pod položkou *Súbor* v hornom menu. Otvorí sa dialógové okno pre uloženie súboru. Bitmapa sa dá uložiť iba do súboru formátu bmp. Príponu .bmp pri zadávaní názvu súboru písať nemusíte. Pokiaľ sa z nejakého dôvodu nedá uložiť táto bitmapa editor vypíše správu s chybou.

4. 4. Tvorba a práca s rozkladovými pravidlami

Hierarchická mozaika vzniká postupným aplikovaním rozkladových pravidiel z istej množiny, ktorú voláme hierarchia. Hneď po spustení aplikácie, sa načíta hierarchia zo súboru *default.hrh*. Pôvodne ide o hierarchiu s rozkladovými pravidlami vyobrazenými na obrázku 10.

Samotná aplikácia pravidiel, tu označovaných ako príkazov, sa prevádza jednoduchým kliknutím na jednotlivé tlačítka s ich názvami, ktoré sa nachádzajú na panely vpravo dole. Hierarchia v tomto editore je obmedzená dvanástimi rozkladovými pravidlami. Pokiaľ je týchto pravidiel menej, tak zvyšné tlačítka sú označené dvojicou pomlčiek „--“ a sú neaktívne.

Vytváranie mozaiek čoraz väčších iterácií si vyžaduje čoraz dlhší čas výpočtu, lebo sa tu pracuje z čoraz väčšími bitmapami. Veľkosť výslednej bitmapy sa pritom zvyšuje exponenciálne. Prakticky a esteticky využiteľné mozaiky sa však dajú získať po tretej, štvrtej alebo maximálne piatej iterácii. Pokiaľ chcete v aplikácii rozkladových pravidiel pokračovať ďalej, je vhodný aj takýto postup: Vytvorte mozaiku maximálne piatej iterácie. Naškáľujte ju na požadovanú veľkosť, pokiaľ možno na tú najmenšiu, kde je možné rozpoznať vzor. Výsledok uložte a následne načítajte

ako novú základnú dlaždicu. Predtým by bolo vhodné zresetovať mozaiku (viď kapitola 4. 3.) voľbou *Reset* pod položkou *Mozaika* v hornom menu, aby ste sa vyhli jej tvorbe s novou dlaždicou. A môžete pokračovať s tvorbou mozaiky vyšších iterácií.

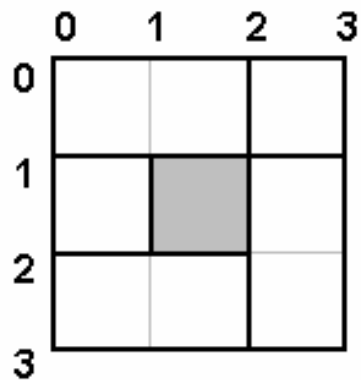
Tvorba vlastnej hierarchie. Predpokladá sa, že hierarchia je uložená v súbore s príponou *.hrh*. Tento súbor je obyčajným textovým súborom obsahujúcim záznamy jednotlivých rozkladových pravidiel, ktorý je editovateľný ľubovoľným textovým editorom, ktorý je schopný uložiť čistý text, bez formátovania a akýchkoľvek značiek. Tu je príklad jedného takého záznamu:

```
p4
scale: 2
items: 4
0 0 1 1 R270
1 0 1 1 R180
0 1 1 1 R0
1 1 1 1 R90
```

V prvom riadku je uvedené meno pravidla, pod akým bude v editore hierarchických mozaiek vystupovať. Druhý riadok uchováva rozmer rozkladového pravidla. V tomto prípade je to 2. Tretí riadok udáva počet dlaždíc v rozkladovom pravidle a ostatné riadky sa vzťahujú každý k jednej dlaždici. Tie sa skladajú zo štyroch číselných hodnôt a kódu reprezentujúceho preklopenie a natočenie danej dlaždice.

Prvá dvojica čísel, to sú súradnice ľavého horného rohu dlaždice. Môžu nadobúdať rozsah 0 až $m-1$, kde m je rozmer rozkladového pravidla. Editor to však nekontroluje. Pokiaľ je dlaždica mimo rozkladové pravidlo, tak sa proste nezobrazí. Druhá dvojica čísel určuje veľkosť dlaždice. Minimálne však musia byť rovné jednej. Editor to opäť nekontroluje, ale tentoraz použitie pravidla s nulovou veľkosťou dlaždice vedie k zmrznutiu aplikácie.

Príklad tvorby rozkladového pravidla. Na obrázku 15 máme pravidlo, ktoré chceme zapísať do súboru s hierarchiou. Možno toto pravidlo nedá práve estetické výsledky, ale pre vysvetlenie bude dobré. V prvom rade by malo mať nejaké meno. Nazvime ho teda *moje*. Vidíme, že sme ho nakreslili do mriežky 3×3 . Rozmer tohto pravidla je naozaj 3, do menšej mriežky ho už nedostaneme. Takže druhý zápis záznamu bude *scale: 3*. A keďže sa skladá z piatich častí (to v strede bude medzera), ďalším riadkom editoru oznámime aj túto skutočnosť, *items: 5*. Píšeme malými písmenami. a nasledujú už len riadky jednotlivých dlaždíc. Na poradí nezáleží. Napríklad dlaždica vpravo dole by mala súradnice $2\ 1$ a veľkosť $1\ 2$, lebo v osi x zaberá jedno políčko mriežky a v osi y políčka dve.



Obr. 15 Definícia rozkladového pravidla

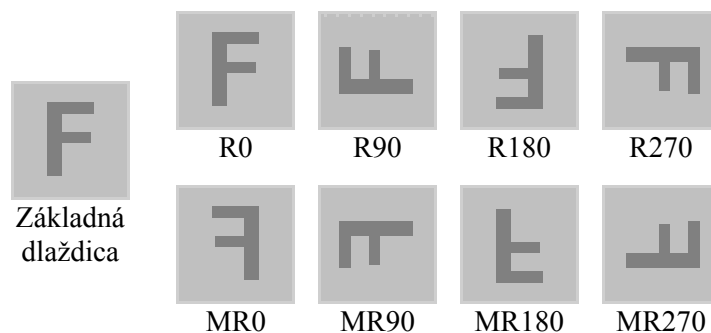
Tu je celý zápis tohto rozkladového pravidla:

```

moje
scale: 3
items: 5
0 0 2 1 R0
2 0 1 1 MR0
0 1 1 1 MR180
2 1 1 2 R0
0 2 2 1 R270

```

Čo znamenajú posledné údaje v riadkoch jednotlivých dlaždíc už bolo spomenuté v kapitole 3. 2. 2. Ide o kódy operácií rotácie a preklápania okolo osi y . Celkovo existuje osem kombinácií použitia týchto transformácií. Obrázok 16 zobrazuje ich aplikáciu na základnú dlaždicu.



Obr. 16

Čo sa ešte tvorby pravidiel týka. V jednoduchosti je krása. V praxi sú osvedčené pravidlá rozmeru 2. Niečo by sa dalo dosiahnuť i pravidlami rozmeru 3 alebo 4. Ale pravidlá vyšších rozmerov sú buď dosiahnuteľné kombináciou jednoduchších pravidiel, alebo často neprinášajú žiaden estetický úžitok.

Keď už máme viac takýchto pravidiel definovaných uložíme ich do súboru s príponou .hrh. V editore hierarchických mozaik potom je možné potom túto hierarchiu načítať voľbou *Načítaj hierarchiu* pod položkou *Súbor* horného menu. Ak máte v tomto momente vytvorenú nejakú mozaiku, editor vás upozorní na jej zrušenie. Inak príkazy uložené v príkazovom zásobníku by nezodpovedali novým príkazom. V prípade nesprávneho zápisu súboru s hierarchiou, editor vyhlási chybu „Chybný formát súboru!“ a ukončí nahrávanie hierarchie. Tj. pokiaľ editor už nahral nejaké pravidlá bez chyby, tak tieto teraz považuje za hierarchiu.

Ešte pár poznámok k pravidlám s medzerou. Aplikovanie takýchto pravidiel na mozaiku má taktiež za následok vytvorenie medzery. Farbu tejto chýbajúcej dlaždice možno zvoliť kliknutím na voľbu *Farba medzery* pod položkou *Mozaika* horného menu.

4. 5. Škálovanie

Tvorba hierarchických mozaik pozostáva z rozkladu priestoru základnej dlaždice. Avšak my tu pracujeme s bitmapovými dlaždicami, ktoré počas zmenšovania môžu strácať svoje vzory. Aby sa tak nestalo, editor si ukladá výslednú mozaiku m -násobne zväčšenú, kde m je rozmerom aktuálnej iterácie mozaiky. Akým spôsobom si má užívateľ aspoň sčasti upraviť veľkosť výsledného obrazu?

Na hlavnom okne aplikácie sú dve veľké tlačítka *Zväčši* a *Zmeňši*. Ich použitie je zrejmé. Zmeňšovať sa dá mozaika takmer neobmedzene. Každým stlačením tohto tlačítka sa každá strana obrázku zmenší dvojnásobne. So zväčšovaním to môže byť trochu inak. Jednak zväčšovať sa dá maximálne m -násobne oproti pôvodnému obrázku dlaždice, kde m je rozmerom mozaiky v aktuálnej iterácii. Taktiež zväčšenie v každej strane nemusí byť dvojnásobné. Napr. trojnásobné zväčšenie môže nastať ak bolo pri tvorbe mozaiky použité pravidlo rozmeru 3. Pokiaľ sa zväčšuje na maximálnu možnú veľkosť obrazu, tak sa v podstate nič neprepočítava, lebo sa jednoducho vykreslí výsledná bitmapa v celej svojej veľkosti.

Editor tiež implicitne zmeňšuje zobrazovanú mozaiku. Po použití rozkladového pravidla ju zobrazí v tej istej veľkosti ako pred ním. Tomu sa dá zabrániť zapnutím voľby *Neškáľuj* pod

položkou *Mozaika* v hornom menu. Keď je táto voľba zapnutá, nedochádza k spätnému škálovaniu a po aplikovaní nejakého pravidla je výsledný obraz m -krát väčší ako bitmapa základnej dlaždice, kde m je rozmer mozaiky po použití tohto pravidla. Tlačítka *Zmenši* a *Zväčši* fungujú aj v tomto prípade (hlavne *Zmenši*, zväčšiť sa bitmapa po aplikácii nejakého pravidla už v tomto prípade nedá).

4. 6. Práca s príkazovým zásobníkom

Príkazový zásobník je štruktúra, v ktorej je uložený postup tvorby hierarchickej mozaiky. Aplikáciou nejakého pravidla sa na vrchol zásobníka pridá poradové číslo tohto pravidla. Inak prácu s týmto zásobníkom vykonáva väčšina príkazov pod položkou *Mozaika* v hornom menu. Preto ich tu popíšem podrobnejšie.

Ale ešte predtým, ako tak učiním, by som mal poznamenať, že niektoré príkazy sú späté s tzv. vybraným príkazom. Myslí sa tým užívateľom vybraným príkazom z príkazového zásobníku. Tieto voľby predpokladajú predošlý výber nejakého použitia pravidla. Na hlavnom okne aplikácie je komponenta, pomocou ktorej sa k tomuto zásobníku užívateľ dostane. Nachádza sa medzi tlačítkami škálovania a panelom tlačítok jednotlivých rozkladových pravidiel, je taktiež označená ako príkazový zásobník. Text v políčku reprezentuje vybraný príkaz. Tento text sa nedá prepísať. Pokiaľ je aktuálnou mozaikou zatiaľ len základná dlaždica je táto komponenta neaktívna a text v políčku „-- prázdny --“, čo značí, že príkazový zásobník je prázdny, preto sa s ním okrem vkladania prvkov (teda aplikovania nejakého pravidla) nedá nič robiť.

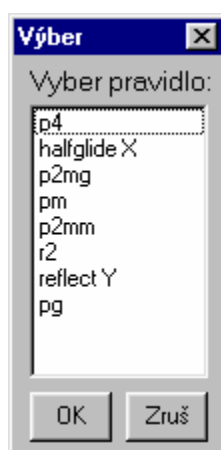
Ak však zásobník prázdny nie je, kliknutím na tlačítko šípky nadol hneď pri tomto texte sa rozvinie zoznam použitých pravidiel usporiadaných od posledne použitého po prvé. Jednotlivé pravidlá sa v tomto zozname môžu opakovať, lebo akékoľvek pravidlo môže byť použité viacnásobne. Po otvorení tohto zoznamu si užívateľ môže vybrať jedno pravidlo a to bude to vybrané pravidlo, na ktoré sa vzťahuje tých niekoľko podpoložiek položky *Mozaika* horného menu. Po akejkoľvek operácii nad týmto zásobníkom, sa ako vybraný príkaz vždy nastaví posledne aplikované pravidlo.

Voľba *Reset*. Nesúvisí až tak s týmto zásobníkom. Ide o návrat k pôvodnej dlaždici a tým o zrušenie všetkých rozkladových pravidiel, ktoré boli pri tvorbe mozaiky vykonané. Zásobník sa úplne vyprázdni.

Voľba *Zruš posledný príkaz*. Vyberie sa posledné pravidlo zo zásobníka bez ohľadu na to, či je práve vybrané alebo nie. Je to krok späť v postupe tvorby mozaiky. Viacnásobným opakovaním tejto voľby je možné dostať sa až k pôvodnej dlaždici. Pokiaľ túto voľbu vyberieme, keď je zásobník prázdny, editor nám túto skutočnosť oznámi.

Voľba *Zruš vybraný príkaz*. Vymaže zo zásobníka vybraný príkaz a upraví mozaiku. Celkovo to prebieha tak, že sa postupne rušia posledné príkazy až kým sa nezruší aj ten vybraný. Postup rušenia jednotlivých príkazov si však editor zapamätá a následne ich bez toho posledne zrušeného vykoná. Pokiaľ je vybraný príkaz ten posledný, operácia je totožná s voľbou *Zruš posledný príkaz*.

Voľba *Zruš po vybraný príkaz*. Postupne sa rušia posledné príkazy, až kým sa vybraný nestáva posledným. Je to viacnásobné použitie operácie *Zruš posledný príkaz*. Touto operáciou sa nedá dostať k základnej dlaždici, lebo ak je vybraný prvý použitý príkaz, tak sa zmaže všetko, okrem neho. Čiže ten tam ostáva. Návrat k základnej dlaždici sa realizuje operáciou *Reset*. Ak je vybraný posledný príkaz, nestane sa nič. Ak je vybraný predposledný príkaz, operácia je totožná s operáciou *Zruš posledný príkaz*.



Obr. 17 Voľba zámenny alebo vloženia pravidla

Voľba *Zameň vybraný príkaz*. Pokiaľ nie je príkazový zásobník prázdny, objaví sa dialógové okno ako na obrázku 17. Na ňom si máte možnosť vybrať pravidlo, ktoré chcete za vybrané pravidlo zameniť. Svoj výber potvrdíte stlačením tlačítka *OK*. Inak sa nič nevykoná. Postup editora je podobný, ako v prípade *Zruš vybraný príkaz*. Rušia sa postupne posledne aplikované pravidlá vrátane vybraného. Editor si tieto pravidlá pamätá (okrem toho, ktoré sa má meniť), aby ich po aplikácii pravidla, ktoré má zameniť za to pôvodné, opäť vykonal.

Voľba *Vlož pred vybraný príkaz*. Postup podobný ako pri voľbe *Zameň vybraný príkaz*. Len s tým rozdielom, že keď sa spätne aplikujú zrušené pravidlá, vykonáva sa i to vybrané. Iterácia mozaiky sa týmto pravidlom zvyšuje.

Práca s príkazovým zásobníkom sa dá výrazne urýchliť použitím klávesových skratiek, ktoré sú uvedené v nasledujúcej kapitole.

4. 7. Zoznam klávesových skratiek

Klávesové skratky sú niečo, bez čoho by práca s mnohými programami bola oveľa zdĺhavejšia. Začiatočníci ich často síce nepoužívajú, ale pokiaľ sa s programom pracuje často, tak namiesto zdĺhavého behania po menu, prípadne po tlačítkach ušetríme veľa času stlačením nejakej dvoj- maximálne trojkombinácie kláves.

Editor hierarchických mozaik, možno takým programom nebude, ale i tak je obohatený vyššie spomenutými možnosťami. Tieto sú využiteľné hlavne pri práci s príkazovým zásobníkom. Mnoho volieb v položke *Mozaika* horného menu totiž súvisí s predtým vybraným príkazom v zásobníku. Tj. treba vybrať príkaz, ísť na horné menu otvoriť ponuku *Mozaika* a tam vybrať nejakú operáciu. Preto je jednoduchšie toto menu z toho vynechať a po vybraní nejakého príkazu, stačí už len stlačiť príslušné klávesy.

A tu sú tie skratky:

- | | |
|-------------------------|--|
| <i>Ctrl – O</i> | - Otvorenie dialógového okna pre načítanie bitmapy do editora. |
| <i>Ctrl – S</i> | - Otvorenie dialógového okna pre uloženie obrázku do súboru. |
| <i>Backspace</i> | - Zrušenie posledne aplikovaného pravidla. |
| <i>Ctrl – Backspace</i> | - Zresetovanie mozaiky. Návrat k základnej dlaždici. |
| <i>Delete</i> | - Zrušenie vybraného príkazu v príkazovom zásobníku. |
| <i>Shift – Delete</i> | - Zrušenie všetkých príkazov z vrchola príkazového zásobníka až po vybraný príkaz v tomto zásobníku. |
| <i>Ctrl – U</i> | - Zamenenie pravidla vybraného príkazu z príkazového zásobníka za pravidlo iné. |
| <i>Ctrl – Insert</i> | - Vloženie nového príkazu do príkazového zásobníka pred vybraný príkaz tohoto zásobníka. |
| + (na num. kláv.) | - Zväčšenie obrazu pokiaľ je to možné (viď kapitola 4. 5.) |

- (na num. kláv.) - Zmenšenie obrazu pokiaľ je to možné.
- * (na num. kláv.) - Maximálne možné zväčšenie obrazu.

Klávesové skratky *Backspace* a *Delete* sú v podstate takmer rovnaké, keďže okrem manuálneho výberu nejakého príkazu z príkazového zásobníka, je vždy označený ako vybraný príkaz ten posledný.

Záver

Hlavnou úlohou tejto práce bolo vypracovanie programového diela, ktorým je možné tvoriť hierarchické mozaiky princípom im vlastným. I keď trieda hierarchických mozaik je ďaleko obsiahlejšia myslím, že Editor hierarchických mozaik spĺňa tie požiadavky, ktoré naň boli kladené. Napriek tomu tu bolo ešte množstvo nápadov, ktoré som chcel uplatniť a množstvo iných riešení, ktoré by som použil teraz. Človek sa učí po celý život a s programovaním je to vždy tak. Vytvoríte si nejaký model a až keď ho využívate a ďalej na ňom pracujete zisťujete, ako ho vylepšiť. Veľmi často prichádzate k tomu, že treba celý projekt prerobiť.

Ale aby som sa vrátil k tým nápadom a postrehom, ktoré ma napadali behom programovania a písania tejto práce a ktoré neboli realizované buď preto, lebo som s tým nemal dostatok skúseností, alebo jednoducho som sa obával, že to časovo nezvládnem. Veľmi mi v editore chýba grafické rozhranie na tvorbu rozkladových pravidiel, ďalej škálovanie by sa dalo efektnejšie zrealizovať prostredníctvom tzv. MIP máp a mohlo by byť trochu interaktívnejšie. Teraz taktiež nechápem, ako ma mohla nenapadnúť možnosť použiť základnú dlaždicu rôzne natočenú a preklopenú. Použitím pravidiel by sme totiž nedostali ten istý obrazec, len inak natočený alebo preklopený. Sú tu opäť nové možnosti.

Čo sa písomného diela týka, bola tu hlavne snaha nazrieť iným uhlom pohľadu na problematiku mozaik, neobmedziť ich priestorovo, popísať ich možno trochu inak, položiť si otázku, či by sa nedalo prostredníctvom mozaik niečo modelovať a vytvoriť k tomu vhodný nástroj. Mozaiky totiž mnohí ľudia chápu len ako obrazce, v skutočnosti môžu predstavovať oveľa viac. Stačí ich vložiť do ďalšieho priestoru.

Pri hierarchických mozaikách, ktoré boli hlavným predmetom tohto diela som sa snažil postupovať od všeobecnosti po konkrétny typ hierarchických mozaik, snažiac sa zachytiť čo najviac možností, ktoré sa s takouto štruktúrou dajú robiť. Zameral som sa hlavne na opis ich vlastností daných ich definíciou. Jednou z ciest, ako ďalej pokračovať, je podrobnejšie analyzovať tieto možnosti, nájsť ich praktické využitia ako pri tvorbe obrazcov, či rôznych textúr tak i dokonca pri modelovaní zaujímavých trojrozmerných štruktúr.

Dúfam, že som dal prácou podnet k ďalšiemu skúmaniu možností mozaik, ktorých nie je zďaleka tak málo, ako by sa na prvý pohľad mohlo zdať. Moja práca ich určite aspoň zopár uvádza. Využitie programového diela môže pomôcť k tvorbe textúr, ktoré slúžia ako pozadia resp. textúr ktoré sú nanášané na 3D objekty. Ale taktiež využitie tohto programu v predmete *Výtvarnej Informatiky*, kde jeho užívatelia môžu získať aspoň zjednodušenú predstavu o tomto type mozaik.

Literatúra

- [1] GLASSNER, Andrew. Andrew Glassner's Notebook. IEEE Computer Graphic. 1998, no.7/8, s. 104-109.
- [2] SLAVOJ, Písek. Delphi (začínáme programovat), podrobný průvodce začínajícího uživatele. Grada Publishing a.s. 2002, ISBN 80-247-0547-8
- [3] SOBOTA Branislav, Počítačová grafika a jazyk C. Kopp 1995, ISBN 80-85828-52-9
- [4] GRÜNBAUM, Branko; SHEPHARD, G. C. Tilings and Patterns. New York: W. H. Freeman, 1987. ix, Chapter 10: Aperiodic tilings. ISBN 0716711931
- [5] GOODMAN-STRAUSS, C. Aperiodic Hierarchical Tilings [online]. University of Arkansas, Fayetteville, Arkansas, 1999. Dostupně z WWW: <<http://comp.uark.edu/~cgstrauss/papers/index.html>>
- [6] GRIMM, U. Aperiodic Tilings on the Computer [online]. Chemitz: Technical University Chemitz, 1999. Dostupně z WWW: <<http://www.arxiv.org/abs/cond-mat/0009277>>
- [7] SENECHAL, Marjorie. Quasicrystals and geometry. Cambridge: Cambridge university press, 1995. ISBN 0521372593.
- [8] ŽIDEK, M. Diplomová práce. Penrose – mozaiky. Vysoké učení technické v Brně 2001.

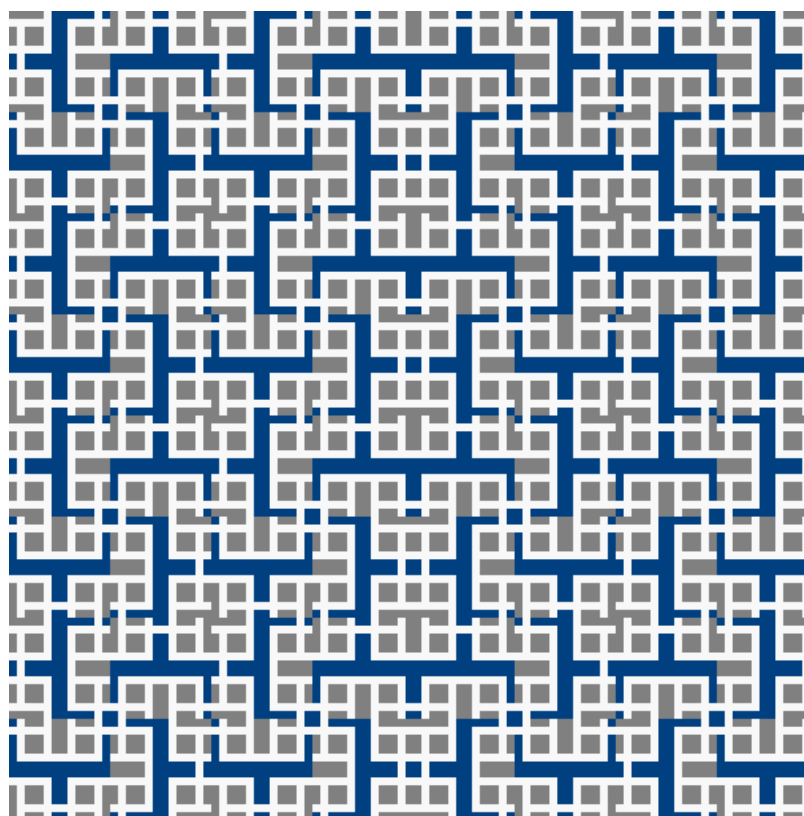
PRÍLOHA – Ukážky mozaík vytvorených Editorom hierarchických mozaík

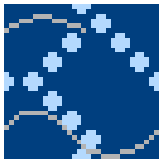
Všetky tu vyobrazené mozaiky vznikali nad hierarchiou uvedenou v kapitole 3. 1. a boli vytvorené aplikáciou popísanou v štvrtej časti.



Základná dlaždica

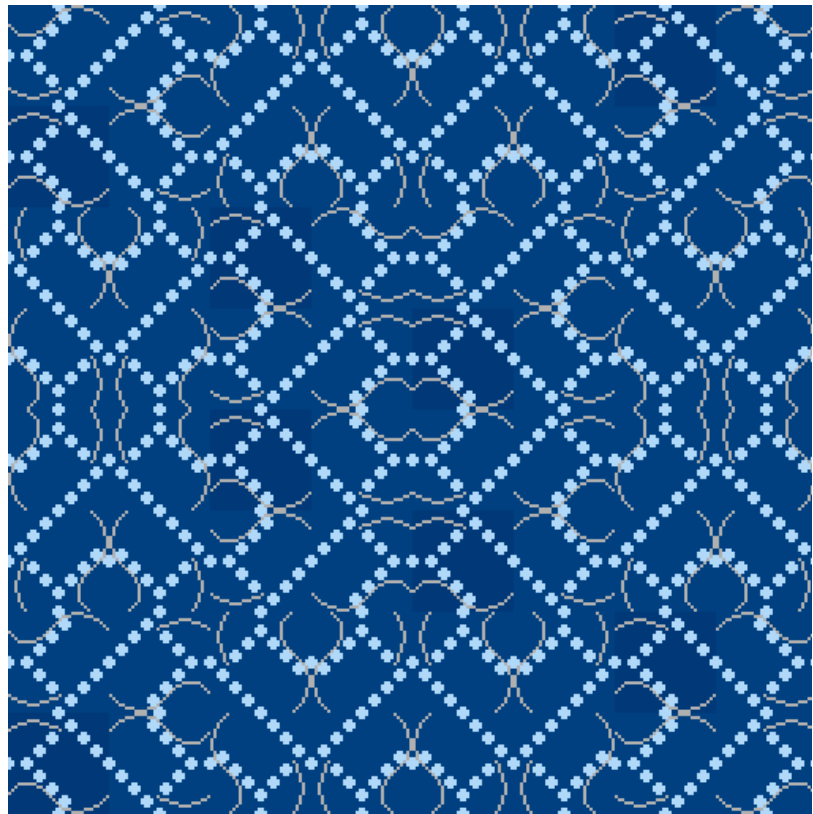
Postup: p2mg, p4, p4, p2mg





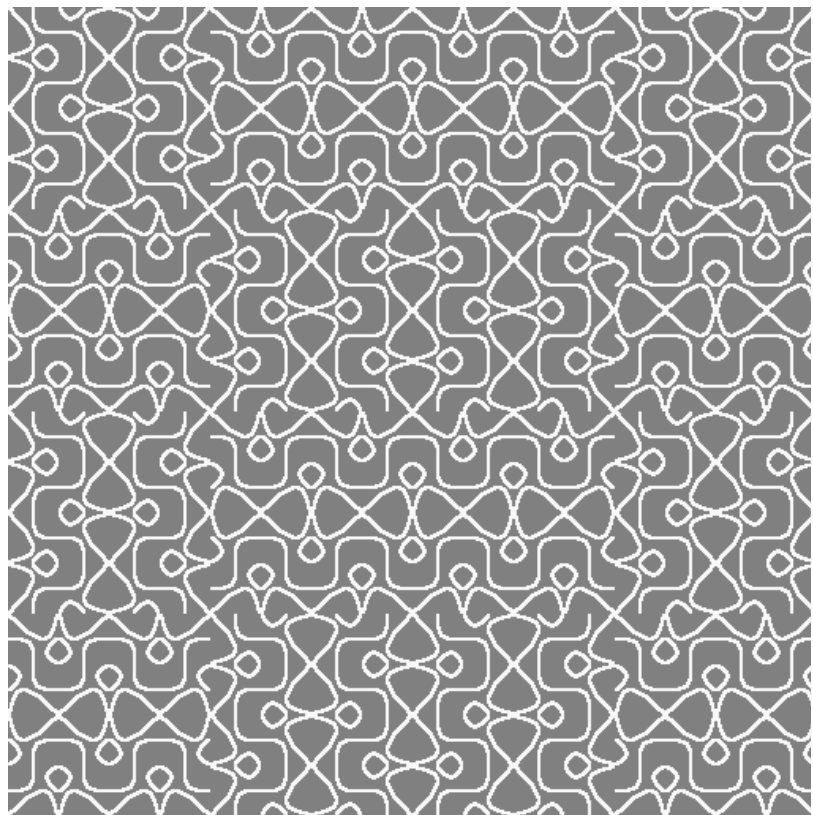
Základná dlaždica

*Postup: r2, reflect Y, p4,
p2mm*



Základná dlaždica

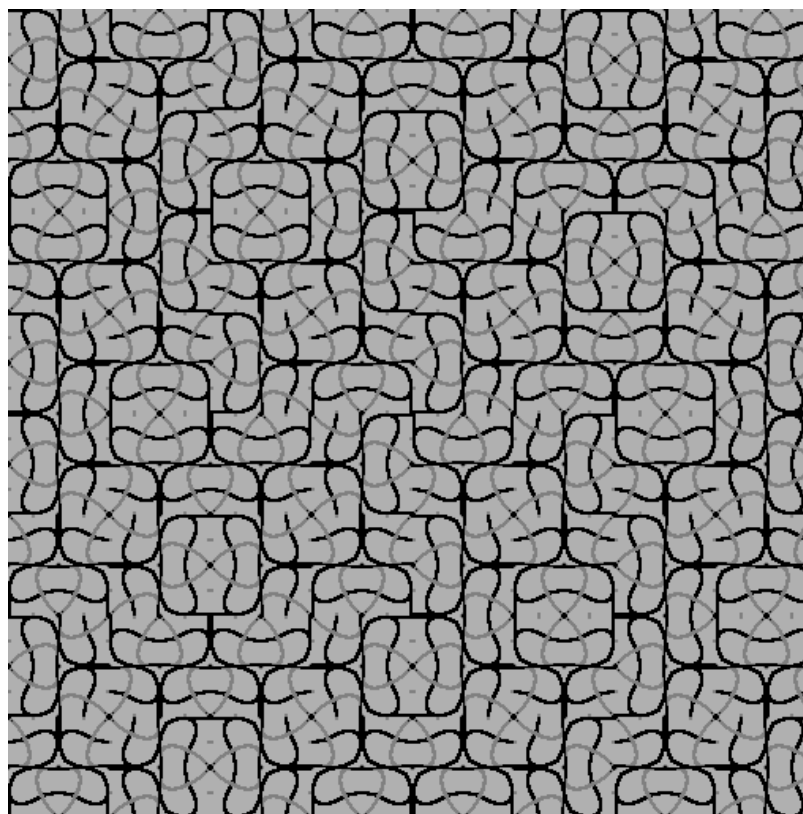
*Postup: p2mg, p2mm, p4,
p2mg*





Základná dlaždica

Postup: p2mg, p4, p2mg, p4



Základná dlaždica

Postup: p2mg, p4, p2mm, p4

