

Fakulta aplikovaných věd – Západočeská univerzita v Plzni

# Dokumentace

K samostatné práci na předmět KIV/UPG

Jan Hinterholzinger

5.5.2020

LS 2019/2020

# Část 1: Pasivní vizualizace toku vody v krajině

---

Jan Hinterholzinger

## Základní informace

Pro vytvoření vizualizace byl použit jazyk Java a postupy předvedené na přednáškách a cvičení předmětu KIV/UPG. Vizualizace je tedy vytvořena za pomoci knihoven Swing a AWT. Vstupní bod programu se nachází ve třídě `me.hintik.simulator.Main` (metodě `main (String[])`). Ve vizualizaci je zapnutá funkce `Antialiasing`, která by měla zajistit zjemnění hran rozmazáním.

## Volba scénáře

Volba scénáře je úvodní uživatelský vstup parametrem, který určuje spuštěný scénář. Při spuštění s žádným parametrem bude automaticky spuštěn výchozí scénář (`=0`). Při zadání špatného parametru (číslo ve špatném rozsahu nebo nečíselnou hodnotu) se program ukončí. Po spuštění programu je spuštěno okno vizualizace, simulace a časovač. Časovač se snaží pravidelně krokovat simulaci po hodnotách 0.02 a překreslit plátno každých 100 ms.

## Vykreslení vodní plochy

Vykreslení vodní plochy bylo vyřešeno bitmapovým obrázkem o velikosti dimenzí scénáře. Hodnoty jednotlivých bodů obrázku jsou nastavovány jako celočíselné RGBA hodnoty podle výsledků metody `buňka.isDry()`. Výsledný obrázek je před vykreslením roztážen v poměru velikosti buněk (`Simulator.getDelta()`) a poté rovnoměrně roztážen do velikosti okna.

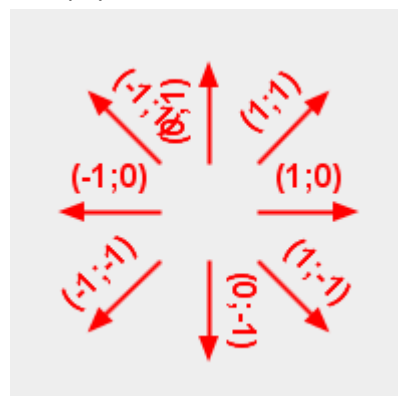
K tomuto postupu jsem dospěl z důvodu snadných a spolehlivých transformací obrázku. Další výhodou použití bitmapy je při použití transparentnosti, vrstvení. Počítám tedy do příštího odevzdání s vykreslením terénu, který bude pravděpodobně vykreslen pod vodní vrstvu.

## Vykreslení vodních zdrojů

### Vykreslení šipky

Je vytvořena pomocná metoda `drawArray(Graphics2D g, double size)`, která vykreslí z počátku souřadného systému v horizontálním směru (směr `(1;0)`) šipku o délce `size`. Metoda je volána s již správným pozicováním a natočením grafického kontextu v metodě `drawWaterFlowLabel(...)`. Rotace je provedena za pomoci metody `Math.atan(double, double)`. Šipka a přilehlý textový popisek nepodléhá změně velikosti při změně okna nebo velikosti buněk. Je nastavená statická velikost šipky na 50 px (není ideální, rozumnější by bylo zvětšovat konstantu velikosti šipky v závislosti na scale obrázku, ale bez závislosti na velikosti buňky – od této konstanty jsou odvíjeny další rozměry šipky a popisku).

Bod počátku šipek v blízkosti okrajů se posouvá dovnitř mapy (pokud se do okraje nevejdou), aby byly šipky i s popisky viditelně vykresleny. To v případě velkého zmenšení okna vede k možnému velkému posunu šipky na mapě. Pravděpodobně nejraději bych situaci řešil uzamknutí nejmenší velikosti okna, aby se tato situace uživateli nestala. Druhý způsob řešení, který jsem mohl zvolit je zvětšit okraje o velikost, o kterou šipky přesahují. Tento postup jsem nakonec nezvolil, protože směr šipky se stále mění a vede to k „třesoucí“ mapě. V takovém případě by byla složité přesně kliknout na určený bod nebo správně označit oblast bez pozastavení simulace.



Obrázek 1 Poloha popisku u různě natočených šipek

### Vykreslení textového popisku

Textový popisek vodního toku má stejné naklonění jako šipka, má tedy stejný směr jako vodní tok. Text se zobrazuje vždy zobrazuje nepřevrácený. To bylo vyřešeno různým natočením a posunutím textu ve směru s X-ovou složkou zápornou a kladnou. Šířka textu je modifikována, aby byla maximálně v šířce šipky. Text popisku je vykreslen na bílý obdélník se zaoblenými rohy, aby byla zajištěna čitelnost textu i na různě barevných podkladech.

### Převedení modelu do okna vizualizace

Velikost samotného modelu je vypočítán jako  $\text{dimenze} * \text{velikost buňky}$ . Do téhle velikosti je také roztažen model a ten je poté roztažen na velikost okna. Vzniknou mi tedy 2 měřítka. Po jejich vynásobení dostanu měřítko pro přímé roztažení do okna. Nutno podotknout, že se z horizontálního a vertikálního měřítka pro roztažení do okna vybírá pouze to menší, aby nebyla celá vizualizace nesprávně roztažena do některé strany. Krom měřítka se také podle velikosti vizualizace posouvá bod umístění, aby vizualizace byla ve středu obrazovky.

## Část 2: Interaktivní vizualizace

---

Jan Hinterholzinger

## Vykreslení mapy terénu

### Vykreslení do obrázku

Terén vykresluji obdobně jako u vodního toku do bitmapového obrázku, který vykresluji pod obrázek s vodní plochou. Do obrázku se vloží barevné informace vypočítané z výšky terénu v daném bodě. Obrázek je dále rovnoměrně roztažen do panelu a posunut na střed.

### Barevná škála

Barevná škála je de-facto přechod od zelené barvy přes žlutou k červené („vzorec“ v metodách `Color getColorByValue(int)` a `Color getColorByHeight(double)` ve třídě `DrawingPanel`). Barevná škála se vždy zasadí mezi minimální a maximální výškou ve scénáři. Tím je zaručeno, že scénáře s nízkými hodnotami výšky (včetně nejvyššího bodu) nebudou jednobarevné. Barevná škála se pro přehlednost a čitelnost rozdělí na 8 barev a výše zmíněné metody vypočítají, která barva se na daný bod použije.

## Legenda

Pro úplnost mapy je požadována legenda, která obsahuje popisky barev a prvků obsažené v mapě. Legenda se nachází v levé části okna. Nachází se ve stejném `JPanelu` jako mapa samotná, aby bylo možné vytisknout mapu společně s legendou.

## Ovládací panel a kontrola simulace

Ovládací panel se nachází v pravé části okna a obsahuje ovládací prvky k simulaci. Umožňuje posun mapy do stran, přiblížení a oddálení mapy, změnu rychlost, pozastavit simulaci a změnu nástroje.

### Navigace

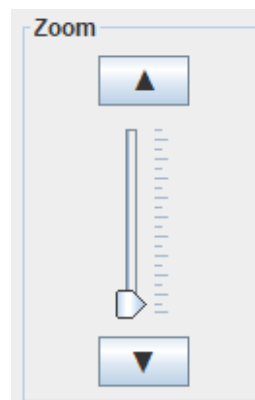
Ovládací panel obsahuje standardní navigační růžice pro posun do stran. A tlačítkem ve středu pro obnovení původní pozici mapy. Posun mapy lze i pomocí myši se stisknutým pravým tlačítkem myši. Posun nemá stanovené limity a je možno s mapou posouvat takřka neomezeně a uživatel si tak může mapu polohovat dle svých preferencí. Příklad „ztraceného okna“ řeší tlačítko pro reset polohy.



Obrázek 2 Navigační růžice

### Přiblížení/oddálení

Mapu lze přiblížovat a oddalovat sliderem a tlačítky v ovládacím panelu. Zoom je reálné číslo, kterým roznásobím hodnotu scale, kterým vyplňuji mapu na velikost panelu. Toto číslo měním ovládáním skokově o 0.1. Z toho důvodu jsem musel zvolit spodní hranici, aby se tato hodnota nedostala pod 0. Hranice jsem tedy zvolil na desetinásobek pro zvětšení a půlnásobek zvětšení pro zmenšení (dvojnásobné zmenšení). Přijdou mi to rozumné hodnoty, které mají reálné využití v zadaných scénářích.



Obrázek 3 Ovládání zoomu

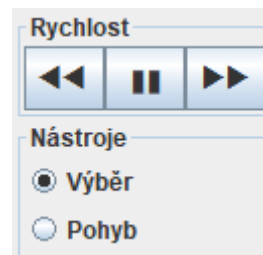
### Rychlost Simulace

Rychlost simulace řeším reálným číslem, které roznásobí původní atribut předávaný metodě `nextStep(double)` (hodnota 0,02). Rychlost simulace opět zvětšuji skokově o hodnotu 0,2. Hranice rychlostí jsem nastavil na 0,2násobek a 3násobek. Tyto hodnoty lze samozřejmě v kódu pozměnit, ale je potřeba, aby výsledný posun simulace byl v intervalu (0,1).

Simulaci lze příslušným také pozastavit a znovu spustit. To je zajištěno jednoduchou pravdivostní proměnnou obsahující informaci, zda je simulace pozastavena či nikoli. Podle této informace se provede nebo neprovede posun simulace a nepřipočítává se také čas simulace.

### Nástroje

Přepínání nástrojů obsahuje dvě polohy „Výběr“ a „Posun“ a mění se chování při stisknutí levém tlačítku. Při volbě „Výběr“ se levým tlačítkem vybírají body nebo oblast, pro který chceme zobrazit graf výšky hladiny. Při volbě „Posun“ se chování myši při stisknutí levého tlačítka změní na přesun mapy stejně jako standardně pravým tlačítkem.



Obrázek 4 Ovládání rychlosti a výběr nástrojů

### Možnosti okna

Okno nabízí také nabídku různých možností. Tyto možnosti lze nalézt na vrchu okna (pod Windows lištou). Nabídka je vytvořena jako JMenuBar. Pro mé potřeby jsem vytvořil vlastní třídu, která dědí třídu JMenuBar, a do ní implementuji potřebné možnosti i s posluchači.

### Ukončení

Možnosti obsahují tlačítko (Soubor->Ukončit) pro ukončení aplikace. Aplikaci ukončuji příkazem `System.exit(0)`.

### Zobrazení nebo skrytí různých komponent okna

V aplikaci lze zaškrtnutými tlačítky v možnosti okna (Okno) jednotlivé dodatečné panely okna (legenda a ovládací panel) odebrat. Prováním to odebráním daného panelu z rodičovského panelu a opětovným vrácením. Rodičovský komponent musí být po této proceduře validován. Tato funkcionality může být užitečná v případě případného odstranění rušivých elementů.

### Zobrazení nebo skrytí různých komponent mapy

Jelikož mám vykreslování samotné mapy rozděleno do několika vrstev (vodní plocha, terén a značky), napadlo mě přidat možnost pro skrytí a zobrazení jednotlivých vrstev. Tyto možnosti se nachází v nabídce „Mapa“. Využití může opět být pro odstranění rušivé nebo nepotřebné vrstvy. Uživatel například může chtít vidět pouze členitost terénu bez vodního toku a značek.

### Tisk

V nabídce je možnost také tisku mapy (Soubor->Tisk). Funkce připraví vizualizaci mapy s legendou pro tisk a vyvolá systémovou nabídku pro tisk. Pro tisk bez legendy je možné ji odebrat v nabídce (Okno->Legenda).

### Grafy

Pro zobrazování grafu jsem použil knihovnu JFreeChart, která mimo jiné zajišťuje i vyplnění grafu do okna. Protože je vytváření grafu pro oba druhy grafů podobné a mají společné vlastnosti, má každý graf vlastní třídu dědící od abstraktní třídy, která obsahuje společné vlastnosti. Je také zajištěno, že může být otevřeno více oken s grafy v současnou dobu a grafy se budou aktualizovat.

### Ukládání dat

Pro zobrazení „historických“ dat v grafu je potřeba je uchovávat. Tyto data uchovávám v seznamu, do kterého je ukládám každý desátý (včetně počátku) posun simulace. Protože jsem narazil na problém nedostatku paměti (`OutOfMemoryError`), zvolil jsem tedy možnost redukce dat jednoduchým vzorkováním (každý druhý prvek se odebere), která se volá po přidání 500. prvku. Tímto krokem jsem minimálně oddálil případ ukončení aplikace.

Tento problém jsem takto řešil až poté co jsem se uchýlil k redukci dat. Totiž je naimplementováno převzatou třídou `MemoryWarningSystem`, že poté co je překročena hranice 80% obsazené paměti se ukončí sběr dat a informuje o tom vyskakovacím oknem uživatele. Tato implementace v programu zůstala pro případ „kdyby náhodou něco“.

Při každém ukládání dat se také pošlou nové informace otevřeným grafům, aby se postupně aktualizovali a nemusely načítat všechna data znovu celá.

### Volba bodu

Volbu bodu pro zobrazení grafu hladiny vody v bodě provádím jednoduchým kliknutím v `MouseListeneru`. Před zavoláním vytvoření grafu se samozřejmě ověří, zda se daný bod nachází na modelu. Nakonec se zavolá vytvoření grafu. Graf zobrazuje hladinu vody v daném vodě v závislosti na simulačním čase.

### Volba území

Volba území je také provedena pomocí `MouseListeneru`, kde se uchovává bod počátečního kliku a po uvolnění tlačítka se po ověření, že se nejedná o klik vypočítají souřadnice a velikost označeného území na modelu. V průběhu označování se zobrazuje okraj označovaného obdélníku. Po vypočtení správných souřadnic se vytvoří nová instance grafu. Graf zobrazuje průměrnou výšku hladiny vody bodů, kde se voda nachází v závislosti na simulačním čase.

### Tisk grafu

Knihovna `JFreeChart` umožňuje graf i vytisknout. Lze tak učinit kliknutím pravým tlačítkem na graf a vybrat možnost „Tisk...“. Otevře se standardní windows print dialog.