# Pět hodin poprvé

Někdy týden po zadání jsem se rozhodl začít dělat semestrálku z UPG. Začal jsem s načítáním dat ze souborů do pole. Pak jsem se bohužel na 5 týdnů zasekl, protože jsem nevěděl, jak zobrazit bitmapový obrázek a jak s ním pracovat. Až někdy 27. března den po 6. cvičení jsem se zase pustil do práce a trošku se semestrálkou hnul.

## Co jsem udělal?

* Naučení se Gitem – Úspěšné naklonování souborů do mého počítače.
* Načtení dat – pomocí metody nacitaniDat(), metoda načte hodnoty ze souboru do pole.
* Vytvoření obrázku – metoda createPicture() nejdříve ověří že data v poli se dají převést na barvu. Jestliže se hodnoty nedají převést na barvu (jsou větší než 255) provede se metoda upraveniHodnot(), která upraví data, tak aby největší hodnota nebyla větší jak 255 a zároveň se nezměnil poměr mezi barvami. Následně se data pomocí cyklu vkládají do obrázku.
* Vykreslení obrázku – metoda drawPicture() do které se jako parametr vloží velikost okna, zajistí že se pomocí scalingu vykreslí obrázek. Pod obrázkem je černý čtverec, pro případ, že má okno jiný poměr stran než obrázek.
* Zajištění minimální velikosti obrázku – nastaveno při počátečním nastavení velikosti drawingPanelu. Pomocí poměru stran se zjistí, která strana se musí nastavit na minimální velikost, aby měla i druhá strana minimální velikost a zároveň byl zachován poměr stran.

# Pět hodin podruhé

V dalších pěti hodinách jsem implementoval bilineární interpolaci, dosud byla interpolace pouze pomocí nejbližšího souseda. Chtěl jsem si ušetřit práci, a tak jsem projížděl internet, jestli neexistuje nějaká metoda přímo v Javě, kterou bych mohl použít. Naštěstí jsem takovou metodu našel, takže bilineární interpolace nabyla velkým problémem. Dále jsem se pustil do šipek, které už byli mnohem těžší. Šipky nesmějí za žádných okolností v mapy ven a musejí mít stejnou velikost. K vyřešení tohoto problému jsem modifikoval kód ze třetího cvičení. Jako poslední jsem přidělal metody na vypočítávání nevyššího a nejnižšího bodu a maximálního stoupání.

## Co jsem udělal?

* Bilineární interpolace – implementována do metody drawPicture(), kód a porovnání změny níže.

g2.setRenderingHint(RenderingHints.KEY\_INTERPOLATION,RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

Obsah obrázku text

Popis byl vytvořen automaticky 

**Bilineární interpolace**

**Interpolace pomocí nejbližšího souseda**

* Kreslení šipek–vytvoření metody drawArrow()
  + Nevykreslování mimo mapu – když konec šipky směřuje k polovině okna -> šipka se nevykreslí mimo mapu. Začal jsem tedy počítat směrový vektor šipky, který vede z „počátečního“ bodu (kam šipka ukazuje) do středu mapy.
  + Stejná velikost všech šipek – ze směrového vektoru jsem udělal jednotkový směrový vektor a ten jsem vynásobil délkou šipky (v mém případě 50px)
  + Scaling šipky – mapa může být trochu zvětšená, nebo zmenšená. Musím tedy zařídit, aby šipka vždy ukazovala na stejné místo na mapě. Hodnoty bodu, na který šipka ukazuje tedy musím upravit stejně jako obrázek: x1 = (x1\*scale) + startX;

y1 = (y1\*scale) + startY;

* Nalezeni lokálních extrém
  + Nejvyšší bod – metoda getMax(), prochází se celé pole a porovnává jestli dosud největší zjištěná hodnota je větší než aktuální vstupní.
  + Nejnižší bod – metoda getMin(), stejná implementace jako při hledání maxima
  + Maximální stoupání – metoda getMaxStoupani(), nejdříve se porovnává vertikální absolutní rozdíl mezi dvěmi sousedními indexy. Byl zde problém s tím, že se porovnávali i indexy přes „hranice“ obrázku, tj. první a poslední řádku. Toto jsem vyřešil podmínkou, že se tyto dvě hodnoty nesmějí porovnávat. Horizontální je provedeno obdobným způsobem je se porovnávají dvě hodnoty nad sebou.

# Pět hodin potřetí

Práce je skoro hotova, stačí doladit pár detailů. Šipky se vykreslují blízko sebe. Rozhodnout se jakou barvu budou mít šipky. Doplnit do složky bin složku data, která se z nějakého důvodu smazala. Otestovat, jestli aplikace běží, jak má (spuštění před Run.cmd s argumenty). Nakonec jsem našel ještě chyby při zobrazování oken, které jsem následně opravil.

## Co jsem udělal?

* Šipky – šipky nevypadají tak, jak bych si představoval.
  + Šipky jsou blízko sebe - Šipky se na některých mapách vykreslovali blízko sebe (např. na mapě random), vyřešil jsem to tak, že jsem do jedné podmínky dal > a do druhé >=, takže se jedna šipka vykreslí na začátku okna a druhá až na konci.
  + Obsah obrázku měřítko, řetěz, zařízení, příslušenství

    Popis byl vytvořen automatickyObsah obrázku text

    Popis byl vytvořen automatickyObsah obrázku text

    Popis byl vytvořen automatickyBarva šipek – zatím byli šipky jen černé což znamenalo, že na některých mapách nebyly vidět. Rozhodl jsem se tedy udělat šipky dvoubarevné, ve spod černé s použitým Gaussovským filtrem a přes to bílé. Myslel jsem si, že to bude vypadat hezky. Bohužel byl opak pravdou a po přibližně dvou a půl hodinách programování jsem se dopracoval k výsledku, který se mi vůbec nelíbil a vzhledem k tomu, že kód se stal celkem nepřehledný, a navíc by se nafoukl skoro na dvojnásobek, a tak jsem se na to vykašlal a změnil barvu šipky na červenou.

**Nepěkné výsledky mého konvolučního filtrování**

* Zobrazení okna – nějak mi nedošlo, že uživatel může zadat nějaký soubor, který bude mít mnohem větší šířku než výšku a obráceně. Opravil jsem to, tak že jsem přidal mnoho podmínek. Teď by snad nemělo dojít k zobrazení okna o šířce 25000 a výšce 600.
* Ošetření vstupu bez parametru – Původně když se nezadal žádný argument, vyskočil do konzole velký výpis chyb. Snažil jsem se toto ošetřit neboli přidal jsem hlášku a defaultní obrázek který se zobrazí, když není zadán žádný argument při spuštění aplikace.

# Deset hodin poprvé

Po ohodnocení pasivní vizualizace, jsem zjistil že mám špatně popisky, protože se vykreslují mimo obrázek. V následujících 3 hodinách trápení se mi to podařilo opravit. Dále jsem udělal barevnou škálu, tak aby vypadalo jako v mapě. Nakonec jsem začal dělat vrstevnice.

## Co jsem udělal?

* Popisky – opravení popisků, tak aby se nevykreslovali mimo obrázek. Zajistil jsem, aby se popisky nevykreslovali mimo mapu, dokud má písmo menší šířku/výšku než obrázek. Implementace je pomocí metody textVObraze(), který vrací boolean hodnotu jestli se má souřadnice textu upravovat nebo ne, a podle toho se pak určí místo kde se popisek vykreslí.
* Barevná škála – Barevnou škála se tvoří v metodě makePalete(), cílem bylo udělat z černobílého obrázku barevný, a jelikož děláme mapu, řekl jsem si že udělám barevnou škálu odpovídající mapě. První plán byl takový že nejnižší hodnota bude modrá, která bude přecházet do zelené a dále do hnědé.

Obsah obrázku text, barevné, oranžová

Popis byl vytvořen automaticky

**První výsledek přidání barevné škály**

Výsledek sice nebyl úplně špatný, ale mě osobně tam něco scházelo, a tak jsem šel na internet stáhl jsem si mapu barevnou mapu České Republiky a pokusil se o napodobení barevné škály následovně:

* + Obsah obrázku čtverec

    Popis byl vytvořen automaticky0% Modrá
  + 20% Tmavě zelená
  + 40% Světle zelená
  + 60% Žlutá
  + 80% Oranžová
  + 99% Hnědá

**Výsledná barevná škála**

* + 100% Černá

Vytváření barevné škály tedy vypadalo tak, že jsem si otevřel program s nastavováním barvy RGB a zjistil jsem hodnoty r, g a b. Poté jsem si vypočítal, o jaká je tato změna mezi těmito hodnotami mezi jednotlivými barvami, aby nedocházelo ke skokům mezi barvami.

Př.: Modrá má hodnotu R = 0, G = 0, B = 250; Tmavě zelená má hodnotu R = 50, G =150, B = 50.

To znamená, že R se zvětší o 50, G o 150 a B se zmenší o 200. Dále tedy musím implementovat funkci postupného převodu. Pro nás to znamená, že když se nadmořská výška zvětší o 1 m (v případě, že maximální nadmořská výška je maximálně 255) R se zdvíhne o 1, G o 3 a B se zmenší o 4.

V případě, že je nadmořská výška větší jak 255, nejdříve se data upraví metodou, tak aby se poměr mezi hodnotami nezměnil. Ještě to bude chtít vylepši v hlavě se mi rýsuje tak, uvidím, jak se mi podaří implementovat.

Osobně si myslím, že se mi paleta barev povedla, barvy vypadají podobně jako v mapách, což byl také můj cíl. V mapě „hory“ jsem objevil menší easter egg ve formě textu „Toto je správně“.

Obsah obrázku mapa

Popis byl vytvořen automatickyObsah obrázku text

Popis byl vytvořen automatickyVrstevnice – Přidělal jsem metodu vetsiNez(), která zjišťuje jestli je nadmořská výška pod danou výškou, vrací pole booleanů, takže teď vím, teď můžu zjistit, v jakých indexech se nadmořská výška přehoupne před nějakou hodnotu.

**Mapa hory**

**Druhý výsledek barevné škály**

# Deset hodin podruhé

Minulý večer jsem začal dělat vrstevnice, celou noc jsem přemýšlel, jak to vykreslit čáry. Nedalo mi to spát, a tak jsem vstával už v půl sedmé ráno (spát jsem šel ve dvě :-D) a začal zkoušet různé metody vykreslení. Nakonec se mi povedlo vykreslit alespoň hranaté vrstevnice a já se mohl další ráno v klidu dospat. Nakonec jsem zvolil něco odpočinkovějšího, první rozšíření – Tisk, který jsme brali na posledním cvičení.

## Co jsem udělal?

* Vykreslení vrstevnic – Přidání metody drawVrstevnice(), věděl jsem že chci využít vytvořeného pravdivostního pole. Metoda tedy funguje tak, že se porovnávají hodnoty v řádku a když se změní pravdivostní hodnota, tak se nakreslí svislá bílá čára. To samé jenom v bledě modrém platí pro svislou osu, kde se nakreslí pro změnu čára vodorovná.

Obsah obrázku mapa

Popis byl vytvořen automaticky

**První verze vrstevnic**

Jak je vidět vrstevnice se opravdu vykreslili, ale nevypadají moc dobře, pokusím se je nejspíš ještě vylepšit, uvidím, jak se mi to povede.

* Přidání timeru – s přidáním vrstevnic se objevil problém, že se vrstevnice překreslují jen když se hýbe s oknem, takže když se pohnulo s oknem rychle, tak se vrstevnice vykreslili špatně, což zrovna nefungovalo u velikých souborů jako např. hory. Problém jsem vyřešil tak, že jsem přidal timer.
* Tisk – Přidání menubaru do okna, funkce tisk je v menu export -> tisk.

# Deset hodin potřetí

V dalších deseti hodinovce jsem se zaměřil na dodělání základních a dalších požadavků interaktivní vizualizace. Nejdříve jsem vytvořil reakci na kliknutí do mapy, kde se po kliknutí zobrazí tečka a nadmořská výška v bodě. Dále jsem pokračoval (i přesto že nemám vyladěné vrstevnice) doplněním legendy a grafů. Po doplnění legendy mi přišlo, že je okno celkem přeplácané, a tak jsem usoudil, že grafy budu vykreslovat v novém okně. Legendu jsem dal do spodní částí okna v novém JPanelu. V tomto bloku jsem vzdal snahu o neopakování se kódu, takže v panelu, která vykresluje legendu je pár úseků skoro stejných jako v panelu, co kreslí mapu.

## Co jsem udělal?

* Kliknutí do mapy
  + Vykreslení bodu s nadmořskou výškou – Pomocí Mouse Listeneru se nejdříve zjišťuje, pomocí dvou podmínek, zda bylo kliknuto na mapu (v případě, že se klikne na okraj, nebo legendu nic se neděje). Dáje se muselo ošetřit, aby bod s číslem neměnil polohu, když se bude okno zmenšovat nebo zvětšovat. Souřadnice X a Y jsou tedy upraveny scalem a zmenšeny o velikost Okraje.
  + Změna barvy nejbližší vrstevnice – Abych změnil barvu správné vrstevnice, musím nejdříve zjistit, která vrstevnice má změnit barvu. O toto se stará metoda najdiNejblizsi(). Metoda prochází seřazené pole, ve kterém jsou hodnoty vrstevnic, a když je nějaká hodnota blíže nebo přesně 25 výškových metrů od nadmořské výšky bodu, bude tato vrstevnice nejblíže. Výjimky jsou krajní body, tedy bod může být vzdálen více jak 25 výškových metrů. Je tedy zde speciální podmínka pro první prvek, tedy jestliže je menší, než první prvek bude nejblíže právě k prvnímu prvku. A zároveň když se projde celé pole a podmínka není splněna a bod je výše než nejvýše vykreslená vrstevnice, takže je nejblíže právě k ní. Nadmořská výška této vrstevnice se ukládá do proměnné zvýrazněné vrstevnice, která se pak vykresluje růžově.

V této funkcionalitě se mi nějak nedařilo získávat správné hodnoty nadmořské výšky v bodě kliknutí při používání jednorozměrného pole, implementoval jsem tedy metodu, kde nahrávám všechna data do dvourozměrného pole, pomocí kterého spouštím metodu najdiNejblizsi().

* Legenda