

Řezání mapy

1. Všeobecné

Obdélníková síť

- bude potřeba nějak zjistit, jaký obdélník souřadnic mapa zabírá (zatím to vidím na načtení celého souboru :()), nebo číslovat čtverečky podle globálních souřadnic (a některé se prostě nenaplní)
- velikost čtverečků teda zatím napevno, pokud budou moc velké, nic se s tím nedá dělat
- názvy souborů: „A-B-C.m“, kde:
 - A je číslo vrstvy
 - B je horizontální číslo čtverečku (může být zem. délka)
 - C je vertikální číslo čtverečku (může být zem. šířka)
- hranice obdélníků budou silné právě „1“ (tj. 0,00001 stupně), jejich body budou patřit do obou obdélníků, které jimi hraničí, „rohý“ do všech čtyř (kvůli dělení polygonů, nebudu vytvářet 2 nebo 4 sousedící body, budu duplikovat jeden)

Paměť

- chtěl bych načíst celou mapu do paměti najednou, je někdo proti?
- případně můžu po uběhnutí určité doby (10MB např.) zapsat vše do souboru a pokračovat, zbytek naappendovat

Hlavička

- informace o obdélníku, který mapa zabírá
- počet vrstev
- název
- ... ?

2. Postup

- Na začátku se všechno načte do paměti, přičemž se budou data z textového převádět na interní formát (POI, POLYLINE a POLYGON budou třídy, se kterými se bude dát lépe pracovat)
- obdélník ve obdélníkové síti budou také třídy, do kterých bude možné třídy POI, POLYLINE a POLYGON ukládat do vektorů
- Každý POI, POLYLINE a POLYGON se rozdělí a výsledky se budou ukládat do jednotlivých obdélníků, kam patří
- obdélník bude vytvořen až tehdy, bude-li ho potřeba
- Nakonec se obsah všech obdélníků v Daweově binárním formátu vyblíže do souborů

3.Řezání POI

- Triviální, zjistí se, do kterého čtverečku se vlezou

4.Řezání POLYLINE

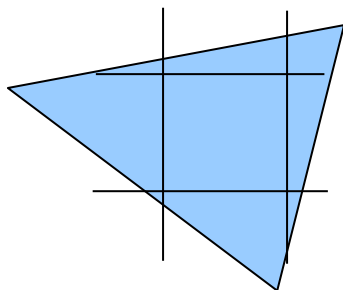
Algoritmus řezání:

- po řadě zkoušet, do kterých obdélníků padnou sousední body:
 - když budou stejné, nic se neděje, pokračuji v té samé čáře
 - když budou různé, rozdělím čáru (a krajní úsečku dám do obou dílů)
 - přesahuje-li úsečka více než jeden obdélník, bude nutné přidat ji do všech, přes které prochází (spočítáním, které hranice obdélníků protíná)obdélník
- prostřednímu ze všech dílů přidám LABEL, ostatní nebudou mít žádný
 - toto trochu nechápu – LABEL by měla mít každá POLYLINE, nepochopil jsem, jestli chceš přiřazovat LABEL přímo nějaké úsečce, nebo nějakému prostřednímu obdélníku ze všech obdélníků, kterými POLYLINE prochází
 - druhá z možností by nebyla příliš dobrý, třeba Labe by na podrobné mapě nemělo moc popisků...
- díly vrátím a rozházím do obdélníků, kam patří (a může se stát, že víc dílů přijde do jednoho obdélníka, když nějaká čára obdélník opustí a pak se do něj vrátí – budou z ní dvě)

5.Řezání POLYGON

Algoritmus řezání – pro jeden POLYGON:

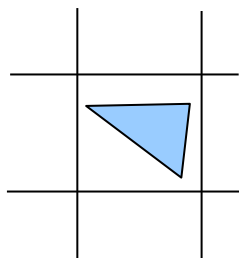
1. Provede se triangulace (tj. vyrobí se trojúhelníky, se kterými se bude dál pracovat).
2. Pro každý trojúhelník se zjistí, kde se protínají jeho úsečky s hranicemi obdélníků.
 - Průsečíky se rozdělí podle toho, na které úsečce leží.
 - Pro obdélníky, které celé leží „uvnitř“ oblasti vytyčené průsečíky (tj. padnou celou plochou do trojúhelníka, aniž by se někde protínaly s jeho hranami)¹, se vytvoří POLYGON odpovídající celé jejich ploše (a můžu je v následujícím bodě vynechat).



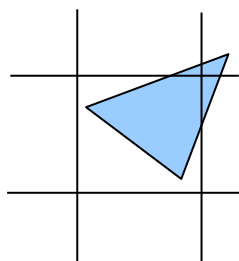
¹ Takové obdélníky lze najít projitím průsečíků trojúhelníka s vertikálními hranicemi obdélníků podle rostoucí X-souřadnice. Pro každou hranici mám dva průsečíky, omezující trojúhelník „shora“ a „zdola“. Jsou-li mezi nimi víc než 2 horizontální hranice obdélníků pro víc než 2 vertikální hranice obdélníků za sebou, potom obdélníky vytyčené těmito hranicemi do trojúhelníku spadají celé.

3. Trojúhelník se rozdělí podle obdélníkové sítě – vytvoří se POLYGONy z bodů trojúhelníka, rohů obdélníků a průsečíků, které by dohromady daly původní trojúhelník. Přitom ukládám vytvořené POLYGONy k obdélníkům, do kterých padnou.

- Nemám-li žádné průsečíky s obdélníky pro nějaký trojúhelník (tedy pokud trojúhelník celý padne do jednoho obdélníka), je zároveň výsledným POLYGONem a můžu tento bod dále přeskočit.

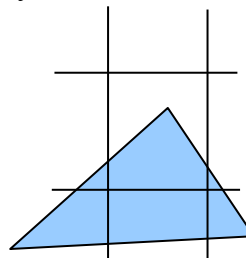
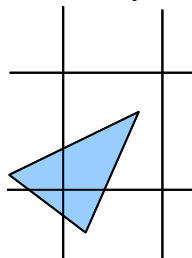


- Uvnitř obdélníka pak leží max. 2 body trojúhelníka, nebo jeden nebo žádný.
- Mám-li dva body uvnitř obdélníka, do výsledného POLYGONU zařadím nejdřív je. Mám určitě dva průsečíky čar na hranicích obdélníka, ty vložím ve správném pořadí (vedle druhého z bodů



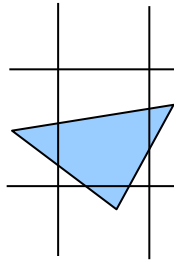
dám průsečík ležící na úsečce, která vede z něj) do výsledku a skončím. Neleží-li oba průsečíky na stejné hranici obdélníka, vložím mezi ně roh obdélníka.

- Pozn.: je potřeba dodržovat správný směr ukládání bodů, jinak by mohl být význam chápán dvojmo (Polish formát to tak má a tedy se z něj dá zjistit, který směr je používán)
- Mám-li jeden bod uvnitř obdélníka, budu mít dva průsečíky úseček, které z něj vedou, plus možná dva průsečíky třetí úsečky. Nenajdu-li průsečíky s třetí úsečkou, postupuji stejně jako v



předchozím případě. (může se stát, že úsečky neprocházejí sousedícími hranami obdélníka, pak je třeba dát dva vrcholy obdélníka a ne jen jeden jak je v předchozím viz druhý obrázek) Jinak spojím vždy dva průsečíky, které leží na stejné hranici (budou určitě po dvou na dvou hranicích) tak, aby s bodem trojúhelníka sousedily body úseček, které z něj vycházejí.

- Nemám-li žádný bod trojúhelníka uvnitř obdélníka, spojím vždy průsečíky, patřící stejným úsečkám a ty pak pospojím mezi sebou nejkratší cestou po hranici (přidám chybějící „roh“).



- Pokud mám jen jednu úsečku (dva průsečíky), přidám ty rohy obdélníka, které leží na stejné straně jako třetí bod trojúhelníka².
4. V rámci každého obdélníka se přímo sousedící POLYGONy, pocházející z toho samého původního, slepí.
- To jde tak, že začnu u prvního, všechny ostatní testuji, mají-li s ním společné 2 body a postupně je přilepuji (po každém přilepení musím začít znova).
 - Nejde-li už nic přilepit, je jeden z POLYGONŮ vyřízený a pokračuji dalším, na který nabalují zbylé.
 - Má-li POLYGON LABEL, dostane ho jedna z částí, která se připleče „prostředního“ ze všech obdélníka, kam padne (mohl by být problém stejný jako POLYLINE, i když nejspíš ne tak velký jako u POLYLINE)

6. Podpůrné geometrické algoritmy

1. Nálezení bodu do obdélníka
2. Průsečík přímek: vrátit bod
3. Průsečík přímek: vrátit násobek směrového vektoru první z nich (možná ?)
4. Zjistit, zda bod leží severně / západně od přímky a vrátit -1 / 0 (leží-li na přímce) / 1
5. Triangulace

² Podpůrný algoritmus 4.