

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

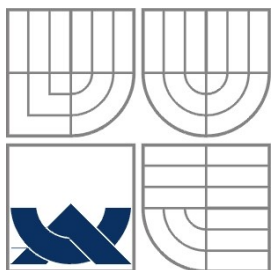
GENERÁTOR MAP PRO HRY
(RPG, POČÍTAČOVÉ, STOLNÍ)

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

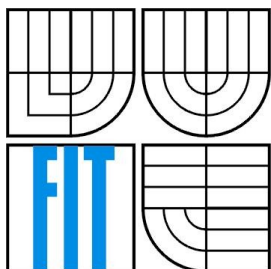
AUTOR PRÁCE
AUTHOR

JIŘÍ HAŠA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

GENERÁTOR MAP PRO HRY (RPG, POČÍTAČOVÉ, STOLNÍ)

MAP GENERATOR USEFUL FOR GAMES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ HAŠA

VEDOUcí PRÁCE
SUPERVISOR

ING. ALEŠ SMRČKA

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Haša Jiří**

Obor: Informační technologie

Téma: **Generátor map pro hry (RPG, počítačové, stolní)**

Kategorie: Modelování a simulace

Pokyny:

1. Seznamte se s nároky a požadavky pro náhodnou tvorbu 2D map užívaných v počítačových nebo stolních hrách. Nastudujte algoritmy pro generování stavového prostoru.
2. Navrhněte lehce konfigurovatelný systém, který umožní vytvořit náhodnou mapu podle požadavků uživatele/programu.
3. Implementujte takový systém modulárním způsobem a vytvořte modul pro export map do grafického formátu.
4. Zhodnoťte dosažené výsledky a navrhněte možné pokračování projektu.

Literatura:

- dle pokynů vedoucího projektu

Při obhajobě semestrální části projektu je požadováno:

- první dva body

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrčka Aleš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jiří Haša**
Id studenta: 84382
Bytem: Dvořákova 27/18, 682 01 Vyškov
Narozen: 06. 10. 1984, Vyškov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Generátor map pro hry (RPG, počítačové, stolní)
Vedoucí/školitel VŠKP: Smrčka Aleš, Ing.
Ústav: Ústav inteligentních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

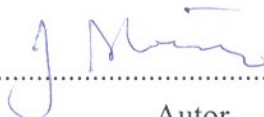
Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel


.....

Autor

Abstrakt

Tato práce se zabývá způsoby vytváření map pro počítačové nebo stolní hry. Je v ní uveden přehled několika editorů a generátorů map pro různé typy her. K práci je přiložen program, který funguje jako obecný modulární generátor map.

Klíčová slova

Mapa, generátor, náhoda, vrstva, hra

Abstract

The subject for this thesis is map generation for computer or desk games. Examples of map editors and generators of several game genres are included. Program attached to this document works as general modular map generator.

Keywords

Map, generator, random, layer, game

Citace

Haša Jiří: Generátor map pro hry (RPG, počítačové, stolní). Brno, 2007, Bakalářská práce, FIT VUT v Brně.

GENERÁTOR MAP PRO HRY (RPG, POČÍTAČOVÉ, STOLNÍ)

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Aleše Smrčky.

Další informace mi poskytl Ing. Aleš Smrčka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Aleši Smrčkovi za cenné rady a připomínky, které mi značně pomohly s řešením této práce

Dále bych rád poděkoval ing. Janu Hašovi za jeho vydatnou pomoc při opravě chyb v textové části práce.

© Jiří Haša, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	8
Seznam ilustrací.....	9
1 Úvod.....	11
2 Mapy – význam a využití.....	12
2.1 Vrstvy map.....	12
2.2 Význam map pro hry.....	14
2.2.1 Herní pole, hrací kameny.....	14
2.2.2 Typy map ve hrách – RPG.....	15
2.2.3 Postup kreslení základní herní mapy.....	17
2.3 Editory herních map.....	18
2.3.1 3D Akce (UnrealEd).....	18
2.3.2 RPG (Neverwinter Nights).....	18
2.3.3 Strategie (C&C Generals).....	19
2.3.4 Shrnutí.....	19
2.4 Generátory ve hrách.....	20
2.4.1 Diablo II.....	20
2.4.2 Civilisation IV.....	21
2.4.3 Open Transport Tycoon Deluxe.....	21
2.4.4 Shrnutí.....	22
3 Systém pro generování map.....	23
3.1 Prostředí pro běh aplikace.....	23
3.1.1 Python.....	23
3.1.2 wxPython.....	23
3.2 Hlavní aplikace.....	24
3.2.1 Systém pro správu modulů.....	24
3.2.2 Zobrazení a export výsledků.....	25
3.3 Moduly.....	25
3.3.1 Nadmořská výška (heightmap).....	26
3.3.2 Moře (oceanmap).....	28
3.3.3 Řeky (rivermap).....	28
3.3.4 Zavlažované oblasti (watermap).....	29
3.3.5 Teplotní pásma (temperaturemap).....	29
3.3.6 Les (forestmap).....	30
4 Závěr.....	31

Literatura a zdroje na internetu.....	32
Seznam příloh.....	33
Seznam ilustrací.....	33
Příloha 1: Manuál k programu.....	34
Spuštění a systémové požadavky.....	34
Popis dialogů programu.....	34
Popis dialogů modulů:.....	36

Seznam ilustrací

mapa povrchu Evropy.....	12
nadmořská výška.....	13
řeky a jezera.....	13
moře.....	13
dostupnost vody.....	13
lesy.....	13
výsledek kombinace jednotlivých složek.....	13
Interpretovaná výšková mapa se znázorněním dostupnosti vody.....	14
čtvercová políčka.....	15
šestiúhelníková políčka.....	16
Unreal editor	18
Neverwinter nights editor.....	18
C&C Generals editor.....	19
Diablo II, mapa katakomb.....	20
herní mapa Civilisation IV.....	21
herní mapa Open TTD.....	21
2 iterace.....	26
3 iterace.....	26
1 iterace.....	26
4 iterace.....	27
5 iterací.....	27
6 iterací.....	27
lineární interpolace.....	27
cosinovská interpolace.....	27
řeky na výškové mapě.....	28
vrstva zavlažovaných oblastí.....	29

teplotní mapa.....	29
mapa lesů.....	30
hlavní okno.....	34
základní konfigurace.....	35
dialóg správce exportu.....	35
okno zobrazující aktuální stav pro export.....	35
konfigurace heightmap.....	36
konfigurace rivermap.....	36
konfigurace temperaturemap.....	37

1 Úvod

V oblasti počítačových a deskových her se stále častěji setkáváme s možností vytvářet nebo importovat vlastní mapy, nebo je aspoň vytváří automaticky vestavěný generátor. Touto cestou je herním fanouškům nabízena cesta k přetváření hry k obrazu svému.

U deskových her je přímo nutné, aby samotní hráči, nebo přinejmenším pán hry ručně vytvářel mapy všech prostředí, do kterých se děj hry může přesunout. Tato činnost je časově velice náročná a vyžaduje velikou dávku fantazie. Ne všichni lidé mají potřebné zkušenosti, znalosti a schopnosti k tomu, aby byli schopni sami si vytvořit kvalitní herní mapy.

Účelem této práce je zjistit, jaký způsob kreslení mapy je nejvhodnější, zhodnotit jednotlivé způsoby vytváření map pro deskové i počítačové hry a z těchto získaných znalostí vybrat nejvhodnější způsob generování vhodný pro praktické řešení generátoru map a ten implementovat.

Tato práce je rozdělena na 2 části. V části **Mapy – význam a využití** se zaměříme na obecné vlastnosti map, metody kreslení map pro deskové hry a stavbu herních prostředí pro hry počítačové. Pokusíme se v této části textu také nalézt obecné řešení pro postup, jaký by měl být dodržován při generování mapy. V druhé části **Systém pro generování map** se seznámíme s algoritmickým řešením vlastního generátoru map.

2 Mapy – význam a využití

Po celou historii si člověk vyznačuje důležitá místa. Od počátečních kreseb na stěnách jeskyní se tyto obrázky vyvinuly v sofistikované kresby zobrazující do detailu téměř každý kout Země. Právě díky mapám a jejich přesnému zpracování lze uskutečnit takové věci jako je navigace lodí kolem útesu nebo noční lety mezi městy.

V dnešní době člověk zakresluje do map nejen okolí, které lze vnímat lidskými smysly, ale také to, co je člověku umožněno vnímat za použití speciálních přístrojů. Setkáváme se tedy s mapami či plány, které zobrazují například města, pobřeží, kontinenty, mořské proudy nebo třeba intenzitu geomagnetického pole. Zkrátka, mapovat se dá téměř vše co je rozloženo v prostoru a vytváří tak jisté mapovatelné oblasti.



ilustrace 1: mapa povrchu Evropy

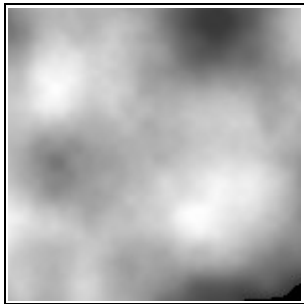
Způsoby vytváření map se za dobu jejich existence změnily stejně jako mapy samy. V minulosti bylo nutné, aby kartograf osobně navštívil všechna místa, která chtěl zakreslit (Pomineme-li kreslení map podle jiných map). S postupem času se objevovaly nové metody, jako je třeba letecké a v současnosti i satelitní snímání. Je nutno podotknout, že u metod měření, kde výsledky nejsou kontrolovatelné pohledem, jsou získané snímky nejprve převedeny do číselné podoby a až takto vzniklé pole dat je možné převést na obrázek.

2.1 Vrstvy map

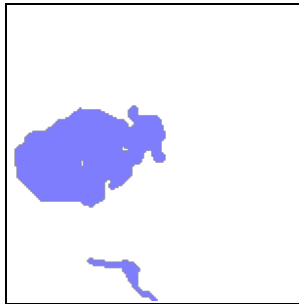
Mapovatelná oblast, například krajina, se skládá z různých prvků, které jsou mezi sebou vzájemně provázány, přestože jde o objekty jiného druhu. Podívejme se na obrázek klasické české vesnice: Několik domů u cesty, náves, kostelík na kopci, strouha napájející místní rybníček, přilehlé sady nebo políčka a okolní lesy a kopce. Pokud bychom chtěli zmapovat oblast popsanou těmito slovy, možná by nám stačilo pouze překreslit fotografii. Ale co když ona vesnice obsahuje i prvky, které nejsou vidět? Může jít třeba o vodovodní potrubí či kanalizace nebo třeba hlučnost prostředí.

Při zakreslování map je vždy důležité vědět, co má výsledná mapa obsahovat za informace a do jaké míry detailu je bude zachycovat. Dále je vhodné na jedné mapě zachycovat vzájemně související data. Pokud se máme držet těchto zásad, nejsme schopni zobrazit na jedné mapě veškeré dostupné informace. Abychom popsalí zadanou oblast, v našem případě onu vesnici, je nutné si získaná data

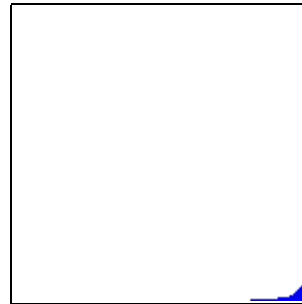
rozdělit do skupin podle typu. Odlišíme tedy data týkající se různých prvků skutečnosti do několika vzájemně souvisejících **vrstev**. Díky tomu získáme sadu **vrstev** nesoucích informace o mapovaném celku odděleně. Tento způsob zpracování otevírá široké možnosti týkající se dalšího zobrazování nebo slebo sledování souvislostí mezi zaznamenanými daty. Na tomto principu jsou založeny Geografické informační systémy.



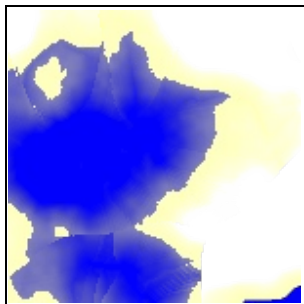
ilustrace 2: nadmořská výška



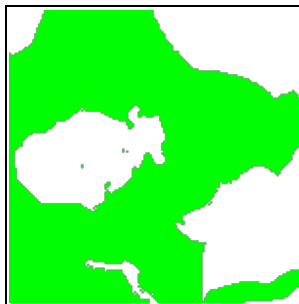
ilustrace 3: řeky a jezera



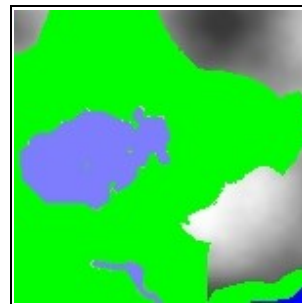
ilustrace 4: moře



ilustrace 5: dostupnost vody

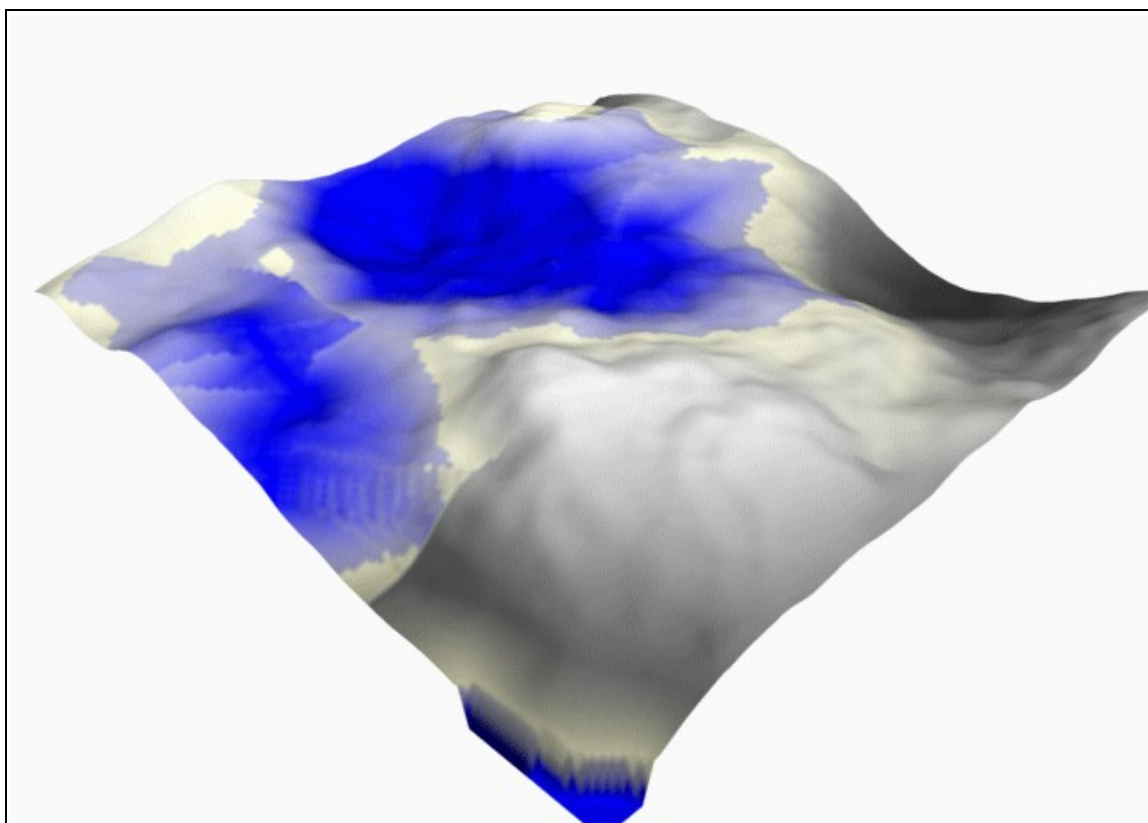


ilustrace 6: lesy



ilustrace 7: výsledek kombinace jednotlivých složek

Tyto obrázky ukazují příklad rozdělení mapy na jednotlivé **vrstvy** a jsou také vidět souvislosti mezi jednotlivými prvky v této oblasti. Lesy například se vyskytují jenom v oblasti, kde je dostupná voda, ale nevyskytují se přímo na vodní hladině. Je také vidět, že čím větší je rozdíl nadmořské výšky hladiny vody a místa, které je měřeno, tím méně je tam voda dostupná. Je zřejmé, že tento způsob zaznamenávání map je výhodný zejména proto, že jednotlivá data je možné interpretovat samostatně a s pomocí této metody vytvořit model mapované oblasti se znázorněním vybraných dat, jak je vidět na následující ilustraci:



Ilustrace 8: Interpretovaná výšková mapa se znázorněním dostupnosti vody

Zde je vidět interpretace vrstvy nadmořské výšky jako souřadnice polygonů 3D objektu, jako textura byla použita vrstva zobrazující dostupnost vody. Data zobrazovaná touto vrstvou byla dále použita k výpočtu několika dalších vrstev.

Důležitými prvky všech map jsou objekty znázorňující části skutečnosti nacházející se v dané oblasti. Takovými objekty mohou být jednotlivé domy, značky u silnice nebo jakékoli významné prvky skutečnosti, které je nutné navzájem odlišit od jiných objektů vlastního typu nějakou unikátní vlastností, třeba adresou. Objekty se rovněž umísťují do vrstev, ovšem tam lze znázornit pouze prostorovou informaci, kterou obsahují (půdorys, tvar) ostatní informace o takových objektech už musí být uloženy v databázi, která spolupracuje se systémem spravujícím vlastní mapu.

2.2 Význam map pro hry

2.2.1 Herní pole, hrací kameny

Všechny hry jsou založeny na manipulaci objekty vyskytujícími se v určitém stavovém poli podle zadaných pravidel. Může jít třeba o slovní fotbal, kde se řadí slova vybraná z množiny slov jazyka za sebe podle shody prvních a posledních písmen jednotlivých slov, nebo o šachy, kde se hrací figury přemísťují po herním poli podle pravidel pro jednotlivé tahy.

Mapy nabízejí skoro neomezené možnosti pro hry, protože čím je složitější herní prostředí, tím více může být hra zajímavá. Na druhou stranu mapy obsahují spoustu informací, které nemusí být pro hry důležité, jde zejména o přílišné množství informací o povrchu herní plochy a nespočet možností, na jaké souřadnice je možné umístit hrací kameny. Proto se ve hrách setkáváme s rozdělením herní plochy na omezený počet políček, která svojí velikostí vytváří kompromis mezi podrobností herního plánu a potřebou jednoduchosti manipulace a získávání informací o kamenech jako jsou absolutní poloha v herním poli, vzájemná poloha či vzdálenost.

Protože jedním ze základních předpokladů zábavnosti hry patří její spád a dynamika, je nutné brát v potaz, jaký systém bude zpracovávat tato data a tomu přizpůsobit herní systém a hlavně herní pole. Pokud má hru hrát člověk, a zpracovávat data má také člověk, hra má na rozdíl od počítačových her relativně jednoduchý systém, malý počet herních kamenů a snadno pochopitelná pravidla jejich vzájemné interakce.

2.2.2 Typy map ve hrách – RPG

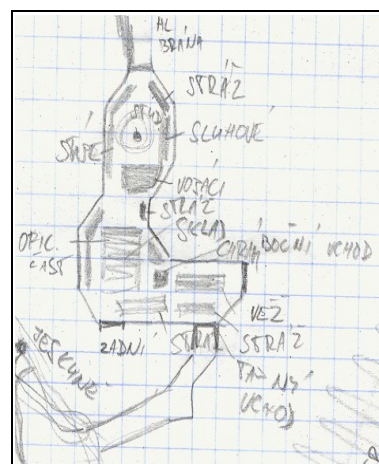
Mezi hry, které nejvíce využívají map patří RPG (Role Playing Game) hry, neboli hry na hrdiny. Jejich obsahem je, že skupina dobrodruhů se musí dostat z místa A do místa B a tam něco udělat, ovšem cesta z A do B je plná nástrah a dobrodruzi musí po cestě navštívit různé množství lokalit, které musí být dopředu zakresleny v mapách pána hry. Čím je taková mapa propracovanější, tím je hra zábavnější a hlavně méně jednotvárná. Když se podíváme na podrobněji na způsob kreslení map v takových hrách, zjistíme, že tyto mapy se dělí podle tvaru políček na 3 základní druhy:

- čtvercová políčka
- šestiúhelníková políčka
- žádná políčka (Volná mapa)

Podívejme se postupně na klady a zápory těchto způsobů:

2.2.2.1 Čtvercová políčka

Jednoznačně nejrozšířenějším a také nejjednodušším systémem políček jsou malé čtverečky. Každý je na ně zvyklý ze školních let a kromě toho se do nich jednoduše zakreslují veškeré pravoúhlé tvary, jaké se vyskytují v obytných budovách (chodby, místnosti apod.), městech (ulice, domy), nebo libovolných umělých prostředích. Co se týče měření vzdálenosti mezi políčky, většinou se řeší prostým odpočtem políček mezi políčky A a B. Málokdy dochází k situacím, že je odchylka směru od jedné z os taková, že by se rozdíl vzdálenosti mezi naměřenou a skutečnou vzdáleností nedal zanedbat.

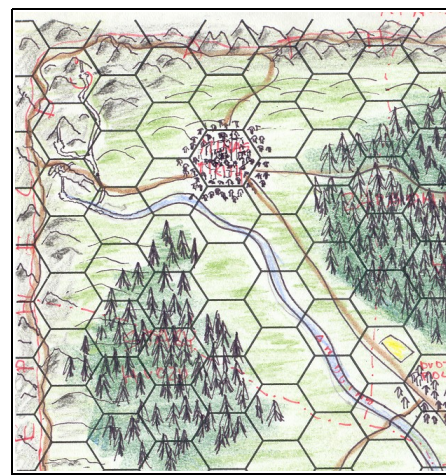


ilustrace 9: čtvercová políčka

Mezi jednoznačné nevýhody patří nevhodnost pro znázornění kulatých předmětů a největší problémy nastávají při tom, že políčka mají jenom 4 sousední pole, takže když je kolem dobrodruhů několik nepřátel v těsné vzdálenosti, někteří se jeví dále, než ve skutečnosti jsou. Dalším problémem je pohyb po diagonále, který bývá raději úplně vypouštěn, aby se předešlo potížím s tím, že situace ve hře naprosto odporuje zdravému rozumu, například když dobrodruh projde skrze mezeru mezi dvěma dotýkajícími se zdmi. Kvůli těmto nesrovnalostem a problémům je v pokročilejších hrách tento systém nahrazen šestiúhelníkovými políčky.

2.2.2.2 Šestiúhelníková políčka

V tomto systému je herní pole rozděleno na malé šestiúhelníky. Ty už dostatečně dobře umožňují zakreslování krajiny ve volné přírodě, protože se s nimi i při značné velikosti dají zakreslit poměrně zdařilé tvary, navíc hrany objektů se různě klikatí a to vytváří dojem přirozenosti. Další výhodou je, že pohyb po diagonále už není tak špatně proveditelný a vzdálenost předmětů se dobře zobrazuje ve více směrech. Měření vzdálenosti se provádí většinou za pomoci provázku nebo jiné pomůcky a odpočítají se hexy (tak se políčka nazývají), na kterých provázek leží.



ilustrace 10: šestiúhelníková políčka

Mezi jeden z menších problémů patří obtíže se zakreslováním rovných hran, ale ten je odstraněn tím, že každý hex je v případě potřeby možno rozdělit na menší pole a to spojením dvou vrcholů šestiúhelníku, které hráč uzná za vhodné. V praxi se s tímto systémem můžeme setkat nejen na poli deskových her, ale i u počítačových tahových RPG a strategií.

2.2.2.3 Žádná políčka (Volná mapa)

Tento způsob vytváření mapy je ze všech herních způsobů kreslení map tím, který se nevíce blíží ke skutečným mapám. Není rozdělen na žádná políčka a tím pádem se vyznačuje nekonečným množstvím směrů pohybu a zakreslitelných tvarů. S tím jsou spojeny velké nároky na představivost těch, kdo mapy vytváří, protože žádný z tvarů není dopředu definován a tak je třeba vše vymyslet. Zatímco u „políčkových“ způsobů se můžeme chytit předlohy v podobě předkreslených buněk a tak vést například cesty a řeky pouze po hranách, v tomto případě si musíme umět dokonale představit souvislosti mezi jednotlivými prvky mapy, aby nedocházelo k nesmyslným situacím ve kterých třeba voda teče do kopce.

Další problém, který vzniká při hraní s takovým typem map je v tom, že spoluhráči si musí mapu obkreslovat, což způsobuje větší či menší podobnost map, proto se taková situace řeší tím způsobem, že místo kreslení se vezme kopie mapy a zakryje se nějakým práškem, třeba moukou a ta

se z již prozkoumaných míst odmetá štětečkem. Protože tento systém má oproti ostatním nejvíce možností a mapa zakreslená bez políček se dá vždy na políčkovou mapu převést, budeme se v tomto textu dále zabývat pouze tímto způsobem kreslení map.

2.2.3 Postup kreslení základní herní mapy

Proto, aby při kreslení mapy nevznikaly problémy s logickou návazností prvků krajiny, je třeba při vytváření mapy postupovat přísně logicky a zakreslovat jednotlivé *vrstvy* tak, abychom při kreslení jistého prvku krajiny znali všechny faktory, které mohou ovlivňovat rozmístění, tvar a druh prvku.

Obecně se začíná kreslení rozvržením celkového rázu krajiny, to znamená jinými slovy nakreslit výškovou mapu. Ta může vypadat rozdílně podle toho, jestli ji kreslíme ručně (pak půjde většinou o vrstevnicovou mapu) nebo ji kreslíme na počítači (s největší pravděpodobností pole odstínů šedé).

Když máme nakreslenou výškovou mapu, je vhodné rozmístit další nejvýznamější prvky krajiny jako je vodstvo. Při kreslení pobřeží nebo ostrovní oblasti si určíme v jaké nadmořské výšce se nachází hladina (při kreslení oblasti jezer se spojitou hladinou se postupuje stejně) a všechna místa s nižší nadmořskou výškou „zaplavíme“. Řeky zakreslíme způsobem, že podle svého uvážení vybereme místa pramenů a z těch vedeme řeku cestou největšího spádu až do moře nebo k okraji mapy, přičemž dbáme na to, že pokud narazíme na nějaké lokální minimum, musíme zakreslit jezero.

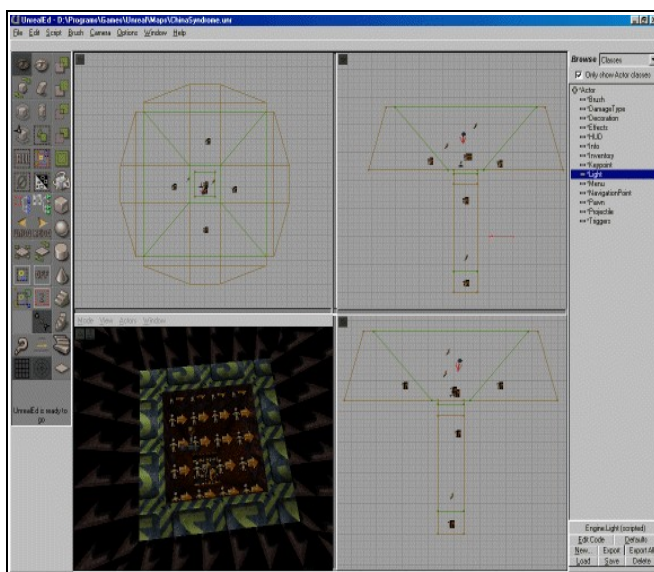
Znázornění ostatních prvků v krajině více záleží na uvážení toho, kdo mapu kreslí, obecně lze říct, že se postupuje od větších a významějších prvků k menším a závislejším detailům. Kreslení a vymýšlení každé mapy je velice namáhavá činnost, která může zabrat klidně i několik hodin na jednu malou lokalitu.

2.3 Editory herních map

Editory jsou programové nástroje, které v počítačových hrách umožňují uživateli vytvářet vlastní mapy z předem připravených prvků, které jsou obsaženy v herním engine (systému pro zobrazování stavu hry). Jednotlivé editory jsou většinou použitelné jen pro jeden, nebo po jistých úpravách pro sadu herních titulů. V této kapitole se podíváme na zástupce editorů pro různé žánry her.

2.3.1 3D Akce (UnrealEd)

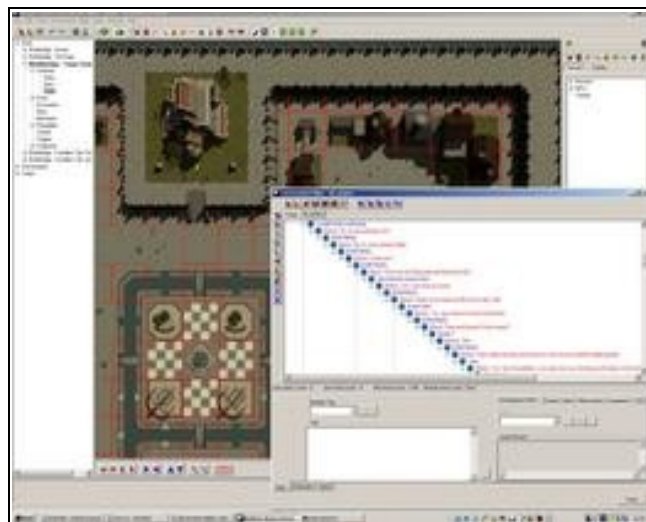
Klasickým zástupcem editorů pro 3D akční hry je program **UnrealEd**, který je dodáván s hrou. Na první pohled je vidět, že jde o program, který funguje jako běžný program pro editaci trojrozměrných objektů s tím, že umožňuje skriptování a programování funkcí jednotlivých objektů. Výstupem programu jsou nepříliš rozsáhlé mapy, které jsou svojí velikostí a mírou detailu vhodné pro zběsilé akce, které se v nich pak odehrávají. Herní mapa není rozdělena na žádná políčka a člověk má plnou svobodu s vytvářením libovolných tvarů, jaké si dokáže vymyslet.



ilustrace 11: Unreal editor

2.3.2 RPG (Neverwinter Nights)

Jako zástupce editorů pro RPG hry jsem vybral editor pro známou hru **Neverwinter Nights** z dílny **Bioware**. Hra se odehrává v trojrozměrném prostředí, které je složeno ze čtvercových, na sebe navazujících políček. Při vyvážení mapy je nejprve nutné zvolit typ prostředí a podle toho program vybere nejvhodnější sadu polí, ze kterých se bude skládat základní ráz krajiny. Po jejím vytvoření je možné dále umísťovat

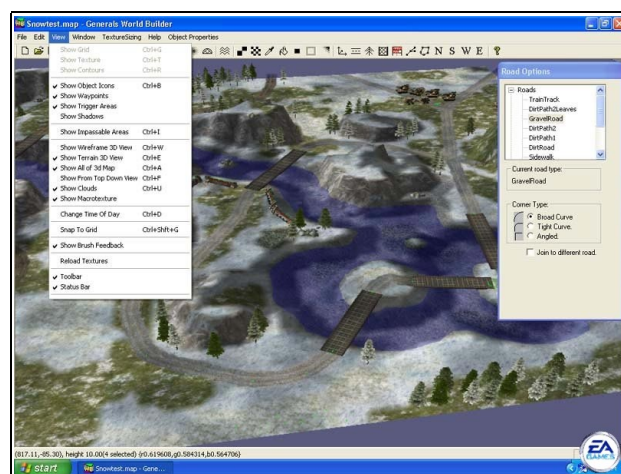


ilustrace 12: Neverwinter nights editor

jednotlivé objekty jako jsou truhlice, lampy a podobně. Přestože tento systém tvorby map se může zdát jako dostatečný, člověk po chvíli používání programu narazí na opakující se místa. Tento neduh je sice možné odstranit zvětšením množství políček, ale i tak jde o značně svazující systém. Na druhou stranu je třeba poznamenat, že tento způsob je uživatelsky velice příjemný a umožňuje rychlou tvorbu herních map.

2.3.3 Strategie (C&C Generals)

Jako zástupce editorů pro strategické hry poslouží editor pro tvorbu map do hry C&C Generals. Herní mapa je jednodílná plocha s měnitelnými souřadnicemi jednotlivých vrcholů polygonů, což umožňuje uživateli dostatečnou svobodu při vytváření krajiny. Zajímavým způsobem je zde řešeno vytváření řek a jezer. Obojí jsou definovány jako polygonové objekty, s plnou možností editace, takže lze vytvářet řeky s velkým spádem. Celá mapa je rozdělena do dvou základních vrstev. V první je povrchová plocha krajiny a v druhé ostatní objekty.



ilustrace 13: C&C Generals editor

2.3.4 Shrnutí

Obecně lze říci, že editory herních map se začaly hojně používat až s příchodem 3D herních prostředí, které umožňuje snadné matematické vyjádření povrchu herní mapy. U starších titulů některých her se editor sice dodával, ale jeho možnosti byly značně omezené. Bylo tomu hlavně proto, že šlo v podstatě o skládání jednotlivých složek mapy interpretovaných jako obrázky vedle sebe. Mapy vytvářené v editorech jsou vždy vázané na konkrétní engine, něco jako obecné řešení se neobjevuje. Narazil jsem sice na hru **Blade of Darkness**, která používala jako mapy soubory vytvořené v programu 3D MAX, ale to tato metoda se dnes moc nepoužívá, protože každá firma používá vlastní engine a editor si radši vytvoří na míru sama.

Co se týče způsobu vytváření map v editorech, je u každého editoru mírně odlišný, v rozsahu od práce v kompletním 3D editoru až po vybírání předdefinovaných políček.

2.4 Generátory ve hrách

Oproti editorům, které vyžadují, aby byly veškeré objekty rozmístovány ručně uživatelem, jsou generátory jistým druhem editoru, kde uživatele nahrazuje počítač. Program rozmísťuje jednotlivé prvky mapy podle předem definovaných závislostí. Ke své činnosti využívá algoritmy jejichž vstupem jsou výsledky jiných algoritmů nebo parametry zadané uživatelem. V této kapitole se nejprve podíváme v jakých hrách se generují jaké typy map a poté si představíme některé základní algoritmy, které se používají.

Generátory zaručují hráč velkou míru znovuhratelnosti bez toho, aby se člověk musel nějak zatěžovat vytvářením nových map.

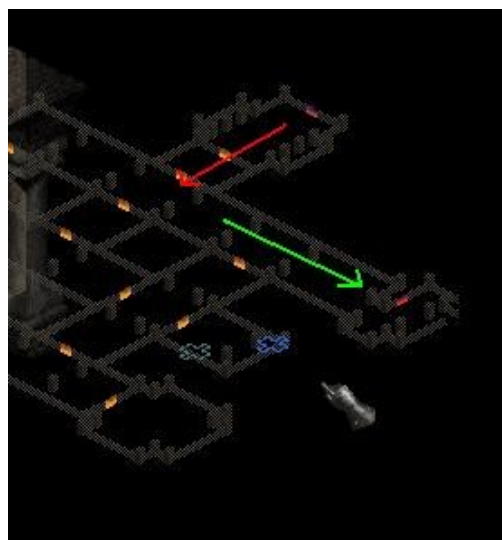
2.4.1 Diablo II

Jednou z věcí které proslavily tuto hru byl právě generátor map, který se spouští automaticky při každém načtení hry. To znamená, že herní mapa je pokaždé jiná. Generátor šikovným způsobem kombinuje prvky vytvářené automaticky s místy, která jsou dána příběhem a jako taková jsou předdefinována. Generátor vytváří dva základní druhy map:

- interiéry (katakomy, vězení)
- exteriéry (pastviny, poušť...)

Jediné, co tyto dva druhy map odlišuje je celkové uspořádání koridorů, které jsou v interiéru mnohem užší a hranice jsou reprezentovány jako zdi. V exteriéru jsou koridory širší a hranice jsou reprezentovány ploty,

skalami, dunami apod. Algoritmus pracuje na principu náhodného umístění políček, která jsou poté propojena. Generátor také musí zajišťovat, aby byla herní mapa průchodná, to znamená, že musí existovat cesta mezi vstupním a výstupním bodem mapy. Poté, co je vygenerována základní vrstva, umísťují se na herní plán různé další objekty, jejich rozmístění je víceméně náhodné v rámci políčka, v rámci mapy se vyskytují na předem definovaných typech polí.



ilustrace 14: Diablo II, mapa katakomb

2.4.2 Civilisation IV

Generátor map pro tuto hru je založen na principu náhodného výběru políček s různými vlastnostmi povrchu (kopec, pobřeží, řeka), poté se na takto vzniklou mapu aplikuje vrstva kvality povrchu, která určuje, zda dané políčko bude reprezentovat poušť, tundru, step, či jiný typ krajiny. Nakonec se přidají objekty jako jsou stromy, lesy, a obchodovatelné zboží. Generátor ke spuštění potřebuje vložit od uživatele (hráče) několik parametrů. Jde zejména o následující prvky zadání:

- velikost mapy
- podíl souše a vody
- průměrná teplota světa
- průměrná vlhkost světa
- množství ostrovů

Tato kombinace políčkového systému v kombinaci s generováním textury podle parametrů zadaných uživatelem prokazuje velice dobré výsledky při krátké době vytváření mapy a je možné tak vygenerovat skutečně věrohodně vypadající mapy (ovšem v rámci limitů daných velikostí políčka).

2.4.3 Open Transport Tycoon Deluxe

Open verze (verze s veřejnými zdrojovými kódy) jedné z nejznámějších budovatelských strategií obsahuje integrovaný generátor map Terra Genesis Perlin, který využívá k vytvoření výškové mapy prostředí **Perlinovu šumovou funkci** (princip je vysvětlen později v textu). Poté co se vygeneruje povrchová mapa, je rozdělena na políčka ve všech osách. Podle parametrů se přidávají další objekty jako jsou stromy, městečka, továrny apod. Jejich umístění je víceméně náhodné. Z



ilustrace 15: herní mapa Civilisation IV



ilustrace 16: herní mapa Open TTD

hlediska generování výškové mapy, která je ve hře reprezentována pseudo 3D mřížkou, je tento

způsob vhodný zejména pro snadnou konfigurovatelnost algoritmu parametry a krátkou dobu výpočtu.

2.4.4 Shrnutí

Generátory používané ve hrách jsou většinou vzájemně odlišné stejnou měrou jako editory. Většina her, na které jsem narazil používá systém generování map výběrem předdefinovaných políček. Tím je zaručeno, že výsledná mapa bude použitelná a zároveň se tímto způsobem lze vyhnout striktně vrstevnatému systému vytváření map, který je výpočetně a hlavně časově náročnější.

Jednou z mála her, která využívá obecného postupu při vytváření herní mapy je **Open TTD**, který využívá algoritmu Perlinovy šumové funkce.

3 Systém pro generování map

V této kapitole se zaměříme na popis algoritmičtých řešení využitých při řešení problému spojených s funkcí obecného modulárního systému pro generování map. Poté se zaměříme na algoritmy, které používají jednotlivé přiložené moduly.

3.1 Prostředí pro běh aplikace

Jedním z hlavních požadavků na tuto aplikaci byla její **přenositelnost**, a aby nevyžadovala ke svému běhu **žádný komerční software**. Dále je nutné, aby aplikace umožňovala **přidávání dalších součástí**, což mě vedlo k tomu, že jsem zvolil k implementaci interpretovaný jazyk Python, který svými možnostmi umožňuje všechny požadované vlastnosti aplikace.

3.1.1 Python

Zde bych si dovil citovat autora Geon z článku Python - popis jazyka (<http://programujte.com/view.php?cislocclanku=1970010106-Python---popis-jazyka>):

„Python je interpretovaný, interaktivní a objektově orientovaný programovací jazyk. Jednoduše se učí a je to mocný programovací jazyk. Má výkonné vysokoúrovňové datové struktury a jednoduchý, přesto mocný přístup k objektovému programování.

Pythonovská elegantní syntaxe a dynamické typování, společně s jeho interpretovanou povahou, ho činí ideálním jazykem pro skriptování a rychlý vývoj aplikací v mnoha oblastech na většině platform.

Tyto vlastnosti činí Python ideální volbou a to hlavně z pohledu rychlosti a jednoduchosti implementace modulárního systému. Python je jazyk, který po importu pythonovského souboru ke zdrojovým datům dokáže pracovat s importovanými daty jako s objektem a tak přistupovat k jednotlivým prvkům jako jsou proměnné, třídy a funkce s pomocí tečkové notace. Tato vlastnost v kombinaci s možností importu souborů, které nejsou známy při psaní zdrojového kódu, znamená, že není třeba vymýšlet vlastní skriptovací jazyk a překladač.

3.1.2 wxPython

WxPython je toolkit pro vytváření grafických uživatelských rozhraní, který je součástí open source projektu wxWidgets. Umožňuje rychlé a efektivní psaní *GUI* pro přenositelné programy, protože je implementován pro platformy Windows, většinu Unix nebo Unixu podobných systémů (Linux) a Mac OS.

3.2 Hlavní aplikace

Hlavní aplikace se skládá ze dvou základních částí, které jsou spuštěny postupně. Po startu aplikace se nejprve vyhledají dostupné moduly, a jejich názvy se poté předají objektu, který má na starost konfigurování jednotlivých modulů, jejich puštění a předávání dat, která moduly potřebují. Tuto část programu budeme nazývat systém pro správu modulů.

Druhá část slouží k zobrazování a exportu výsledků, které jsou poskytnuty systémem pro správu modulů.

3.2.1 Systém pro správu modulů

Systém pro správu modulů je systém, který obstarává veškerou práci s moduly, to znamená především:

1. vyhledání a import dostupných modulů
2. umožnění konfigurace jednotlivých modulů
3. postupné spuštění
4. předávání výsledků a vstupních dat

Jakmile jsou jednotlivé moduly vyhledány a jejich názvy předány třídě pro správu modulů, je pro každý importovaný modul vytvořen a zobrazen formulář umožňující konfiguraci a aktivaci modulu. Deaktivované moduly nebudou zařazeny do seznamu spouštěných modulů.

Před samotným spouštěním je třeba od každého modulu zjistit, jaká data jsou pro jeho běh potřeba a jaká data modul vyvábí. Toho je dosaženo přečtením hlavičky modulů, která tyto informace obsahuje. Následně dojde k vytvoření prázdného pole názvů dat, které vytváří moduly, které jsou přidávány do spouštěcího pořadí. Poté se v cyklu prochází seznam aktivních modulů a vybírají se moduly, jejichž seznam požadavků na data je podmnožinou seznamu již vrácených dat. Program tuto činnost dělá tak dlouho, dokud se po dva kroky za sebou startovní pole nezmění. Následně je zjištěno, jestli má spouštěcí pole stejný počet záznamů jako pole aktivních modulů. Pokud tomu tak není, je zjištěn seznam modulů, jejichž požadavky není možné splnit a uživatel je dotázán, zda si přeje tyto moduly vynechat a pokračovat ve spouštění.

Při spuštění modulu je volána funkce `execute` (t.j. funkce zajišťující provedení vlastního algoritmu) s parametry nastavení hlavního programu a spouštěcí strukturou. Každý záznam této struktury je složen z jména modulu, pořadí spuštění a odkazem na výsledná data modulu. Modul po dokončení své činnosti vrací data ve formátu pole osmibitových hodnot. Tento způsob vytváření výsledků později umožňuje data snadno interpretovat jako obrázek.

Poté, co jsou spuštěny všechny moduly a jejich výsledky jsou uloženy do struktury pro předávání hodnot se spouštěcí část ukončí a data se předají objektu pro zobrazení a export výsledků.

3.2.2 Zobrazení a export výsledků

Integrovaný modul pro zobrazování a část výsledků umožňuje výsledná data jednotlivých modulů interpretovat jako vrstvy mapy a ty zobrazovat. Uživatel má v této fázi k dispozici formulář, ve kterém jsou uvedeny názvy všech výstupních dat, pořadí ve kterém jsou promítány a s jakou maximální viditelností bodu. (Viditelnost má rozsah 0-100 % a určuje, do jaké míry je schopna vrstva přebarvit již položený bod jiných vrstev)

Program zajišťuje, že při změně pořadí nemůže dojít k souběžnému vykreslení dvou vrstev. Toho je dosaženo tak, že při změně hodnoty pořadí je stará hodnocena přesunuta na místo, kde se původně nacházela hodnota nová, takže dojde k přehození hodnot.

Poté co uživatel provede nastavení a stiskne tlačítko „Make“, jsou do výsledného obrázku přidány postupně jednotlivé vrstvy a to tímto způsobem:

- pro 8 bitovou vrstvu se data ztrojí a jako alfa kanál (hodnota určující průhlednost bodu) se vezme hodnota z konfigurace
- pro 16 bitovou hodnotu se třetí bajt dosadí průměr prvních dvou, alfa kanál se vezme hodnota z konfigurace
- pro 24 bitovou vrstvu se přidá alfa kanál, do kterého se nastaví hodnota z konfigurace
- pro 32 bitovou vrstvu se porovná hodnota viditelnosti s hodnotou z konfigurace, pokud je větší, použije se hodnota z konfigurace, jinak se nechá původní

3.3 Moduly

Jako moduly jsou importovány veškeré *.py soubory, které jsou nalezeny v adresáři modules. Každý modul obsahuje hlavičku, funkci pro konfiguraci a funkci pro spuštění modulu. Jakákoli další datová struktura je nepovinná.

Hlavička se skládá z těchto záznamů:

- type: identifikuje dále soubor ke generátoru
- version: číslo verze modulu
- name: jméno modulu, zobrazuje se na kontrolních prvcích
- input: seznam názvů dat, která jsou třeba pro spuštění
- output: jméno výstupních dat

Nyní se podívejme na algoritmická řešení jednotlivých modulů.

3.3.1 Nadmořská výška (heightmap)

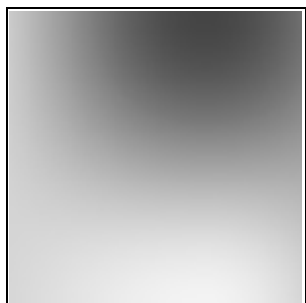
3.3.1.1 Implementovaná metoda

Modul generování mapy nadmořské výšky nevyžaduje žádná vstupní data, ke generování mapy využívá **Perlinovy šumové funkce**. Mezi její hlavní vlastnosti patří:

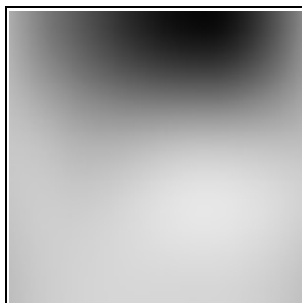
- neměnnost při otáčení a posunutí
- spojitost
- omezené frekvenční spektrum
- opakovatelnost výsledku

První dva znaky nám zaručují to, že výstup funkce se nebude pro daný bod měnit při jakémkoli posunutí, nebo otočení. Reprodukovatelnost dat nám zaručuje, že pro stejné vstupní parametry bude výsledek vždy shodný.

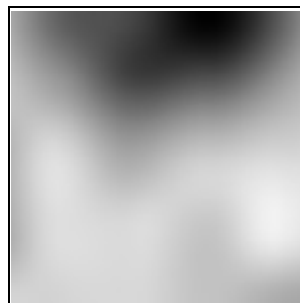
Metoda potřebuje ke své funkci dvě další funkce, kterými jsou generátor náhodných čísel a interpolační funkce, která dopočítává hodnoty bodů, jejichž hodnota není přímo určena náhodným generátorem. Tento výpočet se pro každý bod provede několikrát a výsledná hodnota bodu je součtem výsledků těchto iterací vynásobených hodnotou persistence (schopnost udržovat vliv iterace na hodnotu výsledku), která se v každém kroku zmenší na polovinu své hodnoty. Tím je dosaženo toho, že s každou další iterací se výsledná hodnota bodu mění čím dál méně. Kromě toho se s každou iterací násobí obě vstupní souřadnice bodu, takže se neustále zvětšuje frekvence této funkce. Další věc, která se mění s každou iterací je vzdálenost bodů, které jsou určeny přímo generátorem náhodných čísel, tím je zajištěna ještě větší konfigurovatelnost generátoru, i když výsledky vypadají věrohodně i bez této volby. Následující obrázky byly generovány s hodnotou persistence 50% a bez přídavného zmenšování vzdálenosti přímo generovaných bodů.



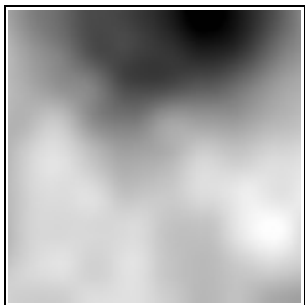
ilustrace 19: 1 iterace



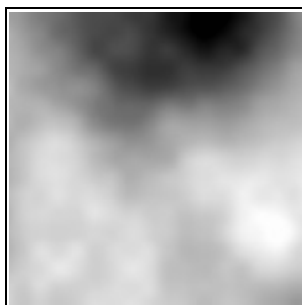
ilustrace 17: 2 iterace



ilustrace 18: 3 iterace



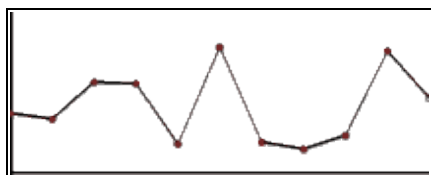
ilustrace 20: 4 iterace



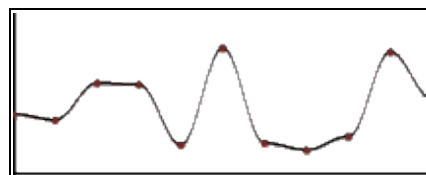
ilustrace 21: 5 iterací



ilustrace 22: 6 iterací



ilustrace 23: lineární interpolace



ilustrace 24: cosinová interpolace

Pro dopočítávání bodů, které nejsou určeny přímo generátorem náhodných čísel, se používá kosinová interpolace. Ta je sice výpočetně mírně náročnější než interpolace lineární, ale zato poskytuje nesrovnatelně lepší výsledky, protože nevytváří žádné ostré přechody. Cosinová interpolace vytváří pěkně zaoblené tvary. Principem této metody je nastavení funkce cosinus takovým způsobem, že interval funkce $\langle 0, \pi/4 \rangle$ promítne do intervalu $\langle 0, \text{velikost interpolační mřížky} \rangle$, podobným způsobem se posunou funkční hodnoty, aby odpovídaly hodnotám krajních bodů interpolační mřížky. Pokud je první bod mřížky menší než druhý, funkce \cos se násobí -1 . Poté už se ostatní body dopočítávají jako funkční hodnoty takto upravené funkce cosinus.

3.3.1.2 Ostatní metody

Přestože metoda Perlinovy šumové funkce podává v rámci vytváření čistě náhodných povrchových map věrohodné a rychle spočítané výsledky, nemělo by se zapomínat i na ostatní metody:

Náhodné přesouvání středního bodu

tato metoda vychází z předdefinovaného tvaru, jehož vrcholy mají náhodné hodnoty. Potom se rekurzivně opakuje postup:

- najde se střed čtverce a lineární interpolací se spočítá jeho hodnota
- k hodnotě středu čtverce se přičte náhodná hodnota
- vypočítají se hodnoty uprostřed hran čtverce
- pro nově vzniklé čtverce se aplikuje stejný postup, zmenší se rozsah náhodného čísla, které se přičítá k hodnotě bodu uprostřed

Metoda náhodných poruch

Při metodě náhodných poruch se generovaná oblast rozděluje náhodnou přímkou na dvě části. Ke všem bodům z první části se přičte náhodné číslo, které reprezentuje odchylku výšky bodu od průměrné výšky bodů v rozích čtverce. Od všech bodů druhé oblasti se toto číslo odečte. Tento postup se iterativně opakuje.

Metoda fyzikální simulace

Tato metoda využívá ke generování výškové mapy fyzikálních postupů, které simulují tektonické děje pro vytváření horských masivů a poté aplikují algoritmy simulující erozi. Tato metoda poskytuje nejvěrohodněji vypadající tvar terénu, ale je výpočetně zdaleka nejnáročnější

3.3.2 Moře (oceanmap)

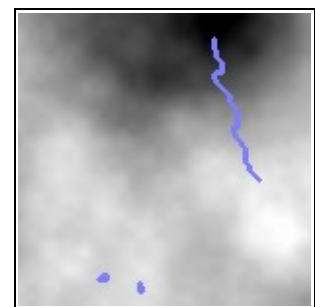
Modul generování hladiny mořské hladiny pracuje na jednoduchém principu zaplavení všech oblastí jejichž nadmořská výška je menší, než hodnota zvolená uživatelem. Toho je dosaženo tím, že pro každý bod na mapě se ověří, jestli je jeho úroveň nad hladinou moře. Pokud je stejná nebo menší, je odpovídající bod na mapě vodní hladiny moře označen jako hladina. Algoritmus vyžaduje hotovou výškovou mapu jako vstupní data.

3.3.3 Řeky (rivermap)

3.3.3.1 Implementovaná metoda

Modul generování řek potřebuje ke svému spuštění jako vstup výškovou mapu. Podle parametrů, které zadá uživatel se vybere množina souřadnic bodů, ve kterých může být pramen. Následně program vyberirá z této množiny náhodný bod, ze kterého „teče“ řeka

Pro samotné generování řeky je použit upravený algoritmus hledání cesty implementovaný metodou **Depth first search** (Metoda založená na prohledávání stavového prostoru „do hloubky“). Určí se limitující hodnota nadmořské výšky, která se u prvního bodu nastaví na 255, což je maximální hodnota v poli nadmořské výšky. Tato hodnota zajistí, že pokud se pramen objeví v lokálním minimu, pokusí se hranice tohoto minima překonat. Poté co jsou určeny souřadnice pramene, algoritmus zjistí souřadnice nejnižšího okolního bodu, který ještě není součástí řeky. Pokud je jeho výška menší, než limitní výška, přidá se výběr okolních bodů na zásobník a zjistí se nová limitní hodnota, ta je vždy druhou nejmenší dostupnou výškou celého okolí řeky.



ilustrace 25: řeky na výškové mapě

Pokud se ve vybraném okolí nenachází žádný bod nižší, než je výška aktuálního bodu, je nastavena výška aktuálního bodu na nejnižší hodnotu okolí a algoritmus vybere znovu nejnižší bod z přechozího výběru. Souřadnice všech bodů, jejichž okolí se zjišťovalo jsou zároveň přidány do pole souřadnic řeky, tím je umožněno, aby algoritmus pro hledání nejkratší cesty vytvářel po cestě jezera. Algoritmus končí, pokud bylo dosaženo bodu s výškou 0 a nebo pokud je pole otevřených bodů prázdné.

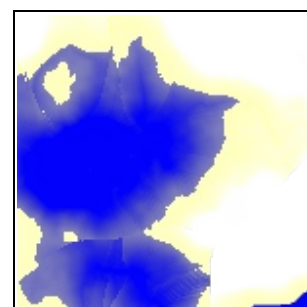
3.3.3.2 Ostatní metody

Metoda sledování průtoku

Tato metoda se řadí k fyzikálně věrným metodám a její výhodou je, že vytváří velice věrohodné rozložení řek a jezer. Principem této metody je, že se na každý bod mapy umístí jistá hladina vody a poté se pro každý bod simuluje stékání vody do níže položených sousedních bodů. Pro každý bod se měří průtok a v bodech s největším průtokem se vytvoří řeky.

3.3.4 Zavlažované oblasti (watermap)

Vstupními daty tohoto modulu jsou výstupy modulů **oceanmap** a **rivermap**, tedy veškeré body, na kterých leží vodní hladina. Uživatel zadá, jaká je maximální vzdálenost bodu od vodní hladiny, aby se dal považovat za zavlažovaný. Algoritmus podle tohoto parametru vybere množinu souřadnic všech bodů, které mají souřadnice x,y maximálně o hodnotu zadanou uživatelem rozdílnou od libovolného bodu z pole vodní hladiny. Poté se pro každý takový bod postupně spočítá přesná vzdálenost ke všem bodům vodní hladiny, přitom se ukládají souřadnice bodu s nejmenší hodnotou vzdálenosti, ta se po spočítání všech vzdáleností zaznamená do mapy.

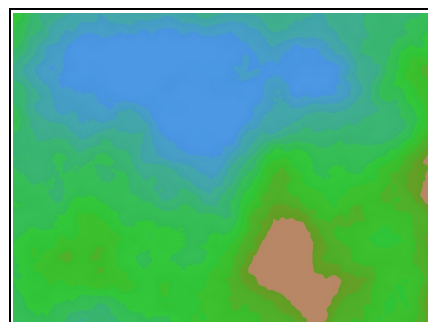


ilustrace 26: vrstva zavlažovaných oblastí

3.3.5 Teplotní pásma (temperaturemap)

Modul vytváření teplotních pásem vyváží mapovou vrstvu znázorňující teplotu v každém bodě mapy. K tomu potřebuje tato funkce několik parametrů. Uživatel zadá, jaká teplota je v každém rohu mapy a určí, jak moc klesá teplota s narůstající nadmořskou výškou.

Algoritmus vytvoří teplotní mapu, která je výsledkem lineární aproximace stavů v rozích mapy, přitom v každém bodě ještě zjišťuje, jaká je aktuální hodnota bodu na výškové

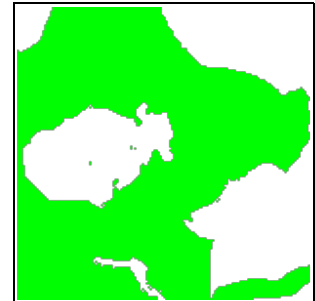


ilustrace 27: teplotní mapa

mapě a podle toho odečte od místní teploty hodnotu (nadmořská výška) * (teplotní úbytek na metr). Výsledná hodnota je poté pro lepší čitelnost znázorněna barevně.

3.3.6 Les (forestmap)

Modul porostu pracuje podobně jako modul teplotních pásem, zjišťuje v každém bodě mapy, jestli je daný bod zavlažovaný, pokud ano vypočítá ze zadaných parametrů, jestli je množství dostupné vody větší, než je nutné k tomu aby byla splněna podmínka pro zalesnění oblasti. Pokud tomu tak je, je daný bod označen jako les.



ilustrace 28: mapa lesů

4 Závěr

Závěrem lze říci, že po prostudování způsobů vytváření map v různých herních editorech nebo generátorech se podařilo nalézt obecný a univerzální způsob pro vytváření herních map založený na vzájemně závislých postupně spouštěných modulech, které mají přístup k datovým strukturám vytvořeným ostatními moduly. Tento systém vytváření map lze použít až na vytváření plánek různých interiérů nebo map exteriérů. Jediné, co je k tomu třeba je dostatečný počet modulů, které budou vytvářet tyto datové struktury. Není tak problém nakonec dosáhnout toho, že se podaří vygenerovat poměrně věrohodný plánec města zasahující mírou detailu až na úroveň předmětů položených na stole.

Vzhledem k tomu, že jednotlivé moduly mají možnost upravovat i pořadí spoštění dalších modulů, je možné v tomto programu vytvořit simulační systém s využitím zpětné vazby různých modulů. Pokud by v modulech byly implementovány algoritmy pro fyzikální simulaci, dalo by se tohoto programu využít třeba při simulaci proudící kapaliny vytvářející si v terénu koryto. Další využití se nabízí v exportu do souborů specifických pro nějaké jiné programy, třeba hry. Toho lze snadno dosáhnout prostým rozšířením systému o další modul.

Největším přínos této práce je pro komunitu hráčů nejen RPG her. Konečně mají k dispozici obecný systém pro generování map. Pokud budou chtít své mapy tvořit na počítači, není pro ně lepší volba než využít tohoto snadno rozšiřitelného programu.

Literatura a zdroje na internetu

- [1] Novák, P. Procedurální model krajiny. Praha, 2005, Diplomová práce, FEL ČVUT v Praze
- [2] Stránky o jazyku Python: <http://www.python.org/>
- [3] Stránky o toolkitu wxPython: <http://www.wxpython.org/>
- [4] Stránky profesora Kena Perlina: <http://mrl.nyu.edu/~perlin/>
- [5] Stránky věnované Perlinově šumové funkci:
http://freespace.virgin.net/hugo.elias/models/m_perlin.htm
- [6] Služba vyhledávání obrázků <http://www.google.com>

Seznam příloh

Příloha 1. Manuál k programu

Příloha 2. Zdrojové texty

Příloha 3. Instalační soubory Python a wxPython pro Win32 a Linux

Příloha 4. CD

Příloha 1: Manuál k programu

Spuštění a systémové požadavky

Program ke svému spuštění vyžaduje, aby bylo na počítači nainstalovaný interpret Python s toolkitem wxPython pro zobrazování uživatelského prostředí. Veškeré instalační soubory pro systémy Linux a Win32 jsou součástí příloh.

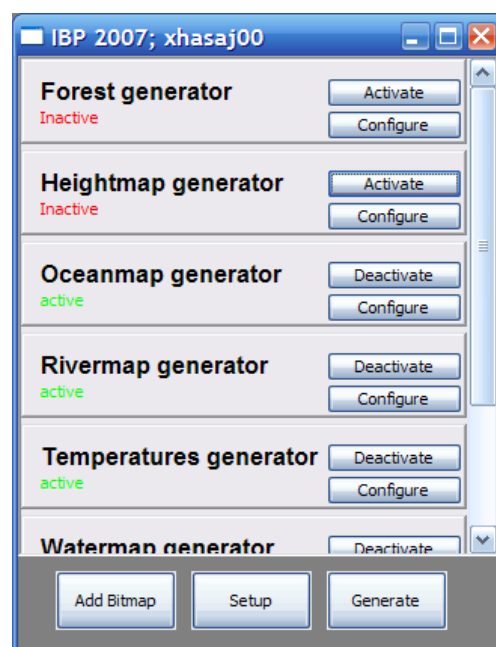
Popis dialogů programu

Popis hlavního okna

Po spuštění se zobrazí hlavní okno aplikace, které je vidět na ilustraci 29. Hlavní plochu okna zabírá seznam všech načtených modulů. U každého modulu se zobrazuje, zde je nebo není aktivní. Tento stav se dá přepnout stisknutím tlačítka **Activate** nebo **Deactivate**. Tlačítko **Configure** zobrazí okno konfigurace příslušící k danému modulu. Pokud žádná konfigurace není možná, nebo v modulu není napsána metoda pro konfiguraci, je tlačítko znepřístupněné.

V ovládací oblasti programu jsou 3 tlačítka:

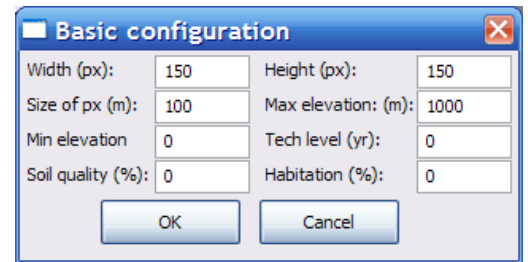
- **Add Bitmap**: Přidá do seznamu modulů modul bitmap, který slouží k nahrazení libovolných výstupních dat daty importovanými z obrázku
- **Setup**: Zobrazí dialog základních parametrů programu.
- **Generate**: Spustí vlastní generování mapy aktivními moduly. Po dokončení generování se zobrazí dialog Canvas controls.



ilustrace 29: hlavní okno

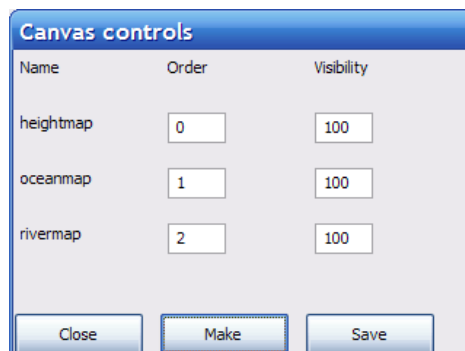
Dialog základní konfigurace umožňuje nastavit tyto parametry:

- **Width:** šířka mapy v pixelech
- **Height:** výška mapy v pixelech
- **Size of px:** horizontální rozměr pixelu v metrech
- **Max elevation:** Určuje jakou hodnotu má maximální dosažitelná hodnota výškové mapy v metrech
- **Min elevation:** Určuje jakou hodnotu má minimální dosažitelná hodnota výškové mapy v metrech
- **Tech level:** Určuje letopočet na mapě, toto je vhodné zejména pro moduly, které generují různé objekty pro odlišná časová období
- **Soil quality:** Obecná kvalita půdy na mapě
- **Habitation:** Určuje míru zalidnění na mapě.

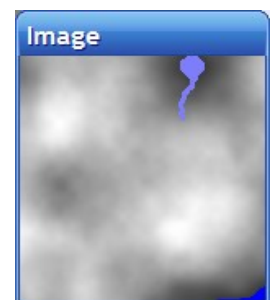


ilustrace 30: základní konfigurace

Dialog Canvas controls umožňuje nastavení dat pro export. Kolonka **Order** umožňuje editovat pořadí v jakém je vrstva přidána do obrázku, kolonka **visibility** zobrazuje maximální viditelnost vrstvy. Tlačítko **Make** provede sjednocení všech vrstev mapy do obrázku a výsledek zobrazí v okně Image. Tlačítkem **Save** se vyvolá dialog pro uložení souboru ve formátu png.



ilustrace 31: dialog správce exportu

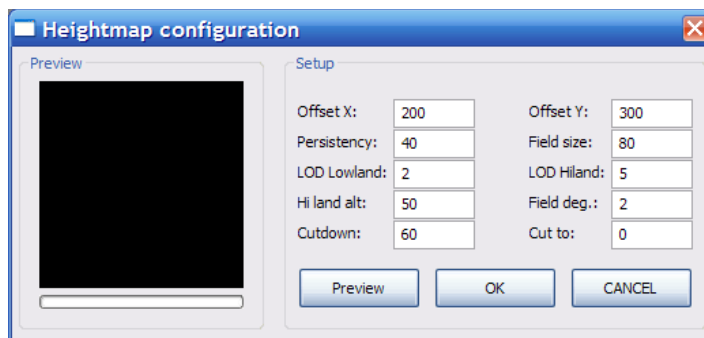


ilustrace 32: okno zobrazující aktuální stav pro export

Popis dialogů modulů:

Heightmap:

- **OffsetX:** Odsazení souřadnicového systému na ose x
- **OffsetY:** Odsazení souřadnicového systému na ose y

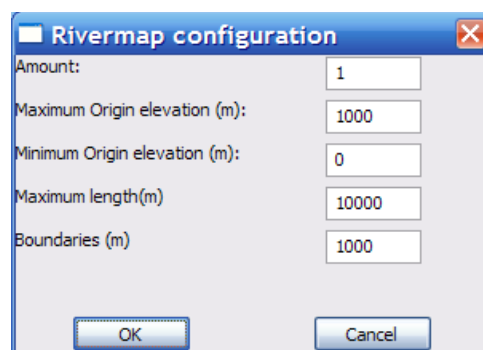


Ilustrace 33: konfigurace heightmap

- **Persistency:** Míra zachovávání vlivu výsledku jednotlivých iterací. Hodnoty v rozmezí 0-100. Čím větší je, tím členitější je výsledná mapa. Pro hladkou mapu doporučuji hodnotu 30 - 40, pro hrubé mapy 50 – 70.
- **Field size:** Počáteční velikost hrany interpolovaného čtverce. Pro větší členitost je vhodné volit menší hodnoty. Doporučená hodnota je zhruba polovina velikosti větší strany mapy
- **LOD hiland a lowland:** Určují počet iterací generátoru pro body v nížinné a horské oblasti. Tím je možné dosáhnout členitějších horských oblasti, zatímco nížiny jsou tvořeny zaoblenými kopečky. Doporučuji hodnoty nedávat moc rozdílné, nebo je vidět hranice mezi oblastmi
- **Hi land alt.:** Určuje nadmořskou výšku, od které se oblast považuje za horskou
- **Field deg:** Určuje přídavné zmenšování interpolační mřížky
- **Cutdown:** Určuje minimální hranici bodu pro generovaný terén. Pokud je nadmořská výška bodu menší, je bod nastaven na hodnotu v poli **Cut to**

Rivermap:

- **Amount:** určuje počet pramenů na mapě
- **Maximum origin elevation:** Určuje maximální nadmořskou výšku, ve které může být umístěn pramen
- **Minimum origin elevation:** Určuje minimální nadmořskou výšku, ve které může být umístěn pramen

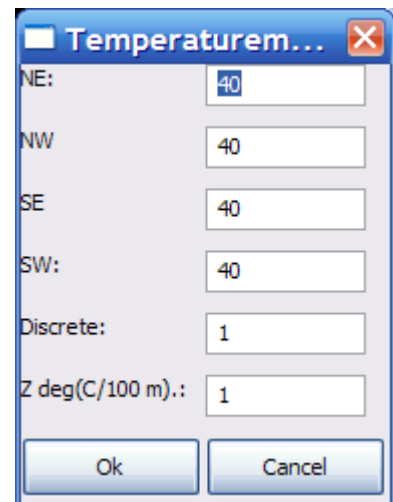


ilustrace 34: konfigurace rivermap

- **Maximum length:** Určuje maximální délku řeky. Toto omezení se v drtivé většině případů neprojeví, slouží pouze k určení maximálního počtu otevřených uzlů v algoritmu hledání cesty.
- **Boundaries:** Minimální vzdálenost pramene od kraje mapy

Temperaturemap:

- **NE, NW, SE, SW:** Nastavené teploty v jednotlivých rozích mapy
- **Discrete:** Určuje míru diskretizace hodnot. Výsledný počet diskrétních hladin je určen $255/\text{discrete}$
- **Z deg (C/100 m):** Určuje míru poklesu teploty ve stupních na 100 metrů nadmořské výšky



NE:	40
NW:	40
SE:	40
SW:	40
Discrete:	1
Z deg(C/100 m):	1

*ilustrace 35: konfigurace
temperaturemap*