

# DARK OBERON

**Užívateľská dokumentácia**





# Obsah

<b>1</b>	<b>MINIMÁLNE POŽIADAVKY A INŠTALÁCIA .....</b>	<b>1</b>
1.1	MINIMÁLNE POŽIADAVKY .....	2
1.2	INŠTALÁCIA .....	2
1.2.1	<i>Inštalácia na systémoch Windows</i> .....	2
1.2.2	<i>Inštalácia na systémoch Unix</i> .....	2
<b>2</b>	<b>MANUÁL HRY .....</b>	<b>5</b>
2.1	VŠEOBECNÝ POPIS HRY .....	7
2.2	SPUSTENIE HRY .....	7
2.3	OVLÁDANIE HRY .....	7
2.3.1	<i>Úvodné menu</i> .....	7
2.3.2	<i>Hracia obrazovka</i> .....	10
2.3.3	<i>Akcie s jednotkami</i> .....	12
2.3.4	<i>Významy klávesových skratiek</i> .....	15
2.3.4.1	Klávesové skratky .....	15
2.3.4.2	Pohyby myši .....	16
2.4	KONFIGURAČNÝ SÚBOR CONFIG.CFG .....	16
2.4.1	<i>Nastavenia okna aplikácie</i> .....	17
2.4.2	<i>Video nastavenia</i> .....	17
2.4.3	<i>Audio nastavenia</i> .....	17
2.4.4	<i>Užívateľské nastavenia</i> .....	18
2.4.5	<i>Sieťové nastavenia</i> .....	18
2.5	DÁTOVÉ SÚBORY .....	18
2.5.1	<i>Súbor fonts.dat</i> .....	18
2.5.2	<i>Súbor cursors.dat</i> .....	18
2.5.3	<i>Súbor gui.dat</i> .....	19
2.6	SLOVNÍK POUŽÍVANÝCH TERMÍNOV .....	20
<b>3</b>	<b>VYTVORENIE HRY .....</b>	<b>21</b>
3.1	KONFIGURAČNÉ SÚBORY .....	23
3.1.1	<i>Spoločné črty načítaných položiek a použité skratky</i> .....	23
3.1.2	<i>Popis súborov schém</i> .....	24
3.1.2.1	Hlavička .....	24
3.1.2.2	Sekcia <Materials> .....	25
3.1.2.3	Sekcie <Segments> a <Segment> .....	25
3.1.2.4	Sekcia <Terrain type> .....	26
3.1.2.5	Sekcia <Fragments> .....	26
3.1.2.6	Sekcia <Fragment> .....	27
3.1.2.7	Sekcie <Layers> a <Objects> .....	27
3.1.2.8	Sekcie <Layer> a <Object> .....	28
3.1.3	<i>Popis súborov rás</i> .....	28
3.1.3.1	Hlavička .....	28
3.1.3.2	Sekcia <Units> .....	30
3.1.3.3	Sekcia <Unit> .....	30
3.1.3.4	Sekcia <Buildings> .....	36
3.1.3.5	Sekcia <Building> .....	37

3.1.3.6	Sekcia <Products> .....	39
3.1.3.7	Sekcia <Sources> .....	40
3.1.3.8	Sekcia <Source> .....	40
3.1.4	<i>Popis súborov máp</i> .....	41
3.1.4.1	Hlavička .....	41
3.1.4.2	Sekcia <Players> .....	42
3.1.4.3	Sekcia <StartPoints> .....	42
3.1.4.4	Sekcia <Races> .....	43
3.1.4.5	Sekcia <Race> a <SchemeRace> .....	43
3.1.4.6	Sekcie <Sets> a <Set> .....	44
3.1.4.7	Sekcie <Units>, <Buildings> a <Sources> .....	45
3.1.4.8	Sekcia <Segment> .....	46
3.1.4.9	Sekcia <Fragments> .....	46
3.1.4.10	Sekcia <Layers> .....	47
3.1.4.11	Sekcia <Objects> .....	47
3.2	DÁTOVÉ SÚBORY .....	48
3.2.1	<i>Rasy</i> .....	48
3.2.2	<i>Schémy</i> .....	49
<b>4</b>	<b>MAP EDITOR</b> .....	<b>51</b>
4.1	ÚVOD .....	53
4.2	HLAVNÉ OKNO .....	53
4.3	MAPA .....	54
4.3.1	<i>Práca so súborom</i> .....	54
4.3.2	<i>Editovanie</i> .....	54
4.4	FAREBNÁ SCHÉMA .....	54
4.4.1	<i>Práca so súborom</i> .....	54
4.4.2	<i>Editovanie</i> .....	54
4.5	KLÁVESOVÉ SKRATKY .....	55
<b>5</b>	<b>DATA EDITOR</b> .....	<b>57</b>
5.1	ÚVOD .....	59
5.2	HLAVNÉ OKNO .....	59
5.3	PRÁCA SO SÚBOROM .....	59
5.4	EDITOVANIE DÁT .....	60
5.4.1	<i>Skupina textúr</i> .....	60
5.4.2	<i>Textúra</i> .....	60
5.4.3	<i>Zvukový záznam</i> .....	60
5.5	EXPORT A IMPORT TEXTÚR .....	61
5.6	KLÁVESOVÉ SKRATKY .....	61







# **1 Minimálne požiadavky a inštalácia**

Marián Černý, Matrin Košalko, Peter Knut

## 1.1 Minimálne požiadavky

Pre úspešné spustenie hry a jej plynulý chod je potrebné nasledujúca minimálna konfigurácia počítača:

- **procesor** – AMD Athlon 1GHz alebo odpovedajúci procesor Intel
- **grafická karta** – s podporou OpenGL 1.1, aspoň na úrovni GeForce 2 (64 MB)
- **operačná pamäť** – 512 MB
- **CD ROM** – kvôli inštalácii
- **sieťová karta** – v prípade multiplayer módu
- **zvuková karta** - voliteľne

## 1.2 Inštalácia

### 1.2.1 Inštalácia na systémoch Windows

Na inštalačnom CD disku je dodávaný inštalátor programu Dark Oberon pre systémy Windows. Inštalačný súbor **doberon.msi** stačí spustiť a v priebehu inštalácie zvoliť adresár, do ktorého bude program nainštalovaný (predvolený adresár je **C:\Program Files\dark-oberon\**). Zvyšné prebehne automaticky. V ponuke „Štart“ vytvorí inštalátor odkaz na všetky potrebné súčasti. Program sa spúšťa odkazom z ponuky „Štart“, prípadne spustením súboru **doberon.exe** v adresári, kde bol program nainštalovaný.

### 1.2.2 Inštalácia na systémoch Unix

Projekt potrebuje pre úspešné preloženie knižnicu GLFW vo verzii 2.5., niekedy označovanú ako 2.5.0. Očakáva sa jej štandardné umiestnenie v súbore **/usr/X11R6/lib/libglfw.a**. V prípade, že sa táto knižnica v systéme nenachádza, je potrebné ju nainštalovať. Ak užívateľ nemá právo inštalovať do systémových adresárov, je možné umiestniť súbor **libglfw.a** aj do adresára **dark-oberon/libs** (po kroku 1 nižšie).

Prehľad príkazov potrebných pre inštaláciu na systémoch Unix do aktuálneho adresára:

1. **tar -xvzf /mnt/cdrom/dark-oberon.tgz**
2. **cd dark-oberon**
3. **make**

Tento postup vytvorí spustiteľný súbor hry **<code>doberon</code>**, ktorý sa spustí príkazom **./doberon**.

Popis jednotlivých príkazov:

1. príkaz rozbalí archív **dark-oberon.tgz** z inštalačného CD do adresára **dark-oberon** v aktuálnom adresári. Je potrebné zadať platnú cestu k archívu **dark-oberon.tgz**. Na niektorých systémoch je CD-ROM mechanika prístupná v inom adresári ako **/mnt/cdrom**. Často to býva adresár **/cdrom** a niekedy **/media/cdrom**. CD-ROM mechaniku bude zrejme potrebné sprístupniť príkazom **mount** – napr. **mount /mnt/cdrom** (užívateľ musí mať potrebné práva, vid' **mount(8)** ),
2. zmení aktuálny adresár na adresár **dark-oberon**,



3. Spustí preklad. Štandardne je projekt preložený bez podpory zvuku, pretože knižnica fmod, ktorú používame v hre, nie je dostupná pre všetky platformy Unix. Ak je knižnica v systéme dostupná, je možné projekt skompilovať s podporou zvuku. Je potrebné zadať príkaz: **mount -DSOUND** namiesto jednoduchého príkazu **make**.





## 2 Manuál hry

Valéria Šventová



## 2.1 Všeobecný popis hry

Hra Dark Oberon je strategickou hrou typu Warcraft 2, prípadne C&C – Red Alert. Každý hráč podľa svojich schopností a možností vyrába počas hry jednotky s rôznymi vlastnosťami (tieto vlastnosti je možné nadefinovať v konfiguračnom súbore), stavia budovy, ktoré dokážu prijímať materiál či vyrábať rôzne typy jednotiek, útočí na nepriateľa, ťaží suroviny, atď. Cieľom hry je získať prevahu nad protihráčom a to v absolútnom zmysle, čo znamená zničiť protihráčove jednotky a zabrániť mu vo výrobe ďalších.

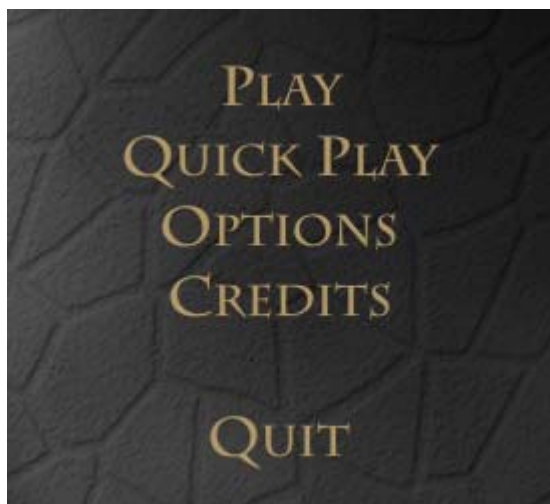
## 2.2 Spustenie hry

Hra sa spúšťa klepnutím na súbor doberon.exe.

## 2.3 Ovládanie hry

### 2.3.1 Úvodné menu

Po spustení hry sa užívateľovi ukáže úvodné menu, situáciu zobrazuje obrázok č.1.



Obrázok 1: Úvodné menu

Jednotlivé položky tohto menu sú popísané v nasledujúcom odstavci.

- **Play** – smeruje k vytvoreniu novej hry, presmeruje hráča na ďalšie menu, v ktorom si môže zvoliť možnosti:
  - **Resume** – umožňuje návrat do už spustenej hry z menu,
  - **Create** – zakladá novú hru. Meno hráča je možné zmeniť v menu, ktoré sa objaví bezprostredne po výbere možnosti CREATE z hlavného menu (obrázok č.2) a klepnutím na Create (výber možnosti Back vráti hráča do predchádzajúceho menu),



Obrázok 2: Create menu

H

Hráč má ďalej možnosť zvoliť si typ mapy (obrázok č.3) klepnutím na danú mapu v sekcii MAP a počítačových hráčov, klepnutím na tlačidlo AddComputer. V poli hráčov sa objaví všetci pripojení hráči vrátane počítačových. Každý z hráčov musí zastupovať iný tábor jednotiek (napríklad Plastic Humans, Plastic Humans – Yellow a pod), ktorý si môže vybrať v rozbaľovacom menu (obrázok č.3).



Obrázok 3: Players, maps, info

- **Disconnect** – ukončí spustenie hry,
- **Connect** – umožní hráčovi zadať server podľa IP adresy alebo sieťového mena. Ak pri pripájaní nastane chyba, dôjde k ukázaní varovanej správy,
- **Back** – výber tejto možnosti vráti hráča do vstupného menu (obrázok č.1),

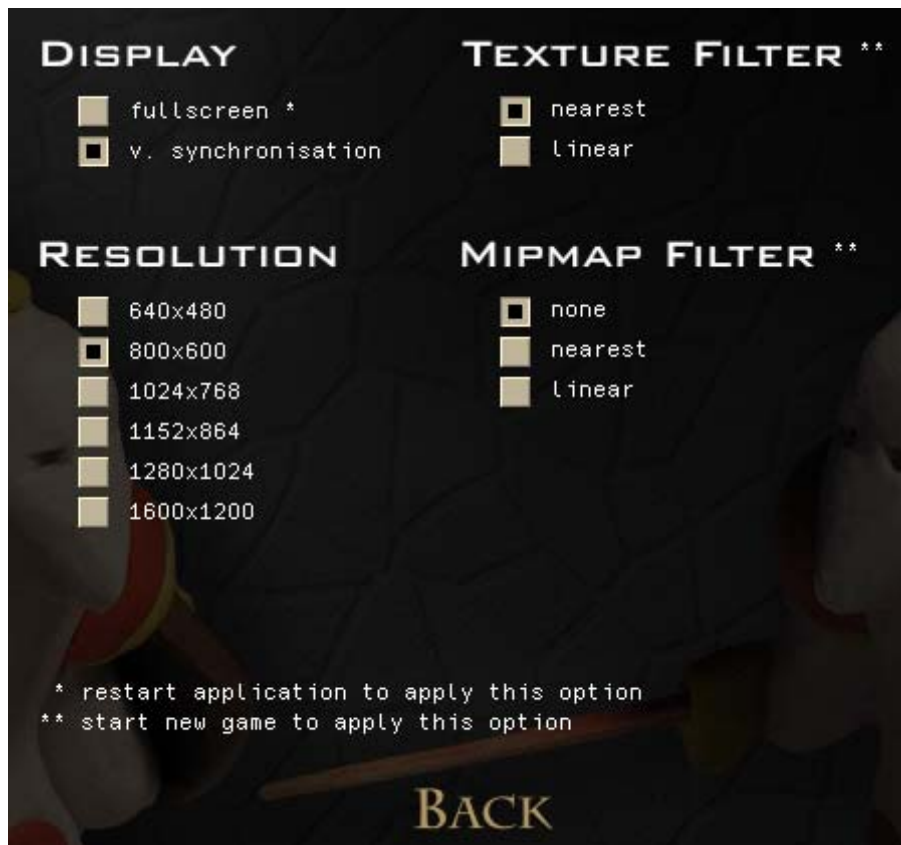
- **Quick Play** – okamžite hráča preniesie priamo do hry, pričom sa vyberie náhodná mapa, náhodná rasa, aspoň jeden počítačový hráč a dôjde k začatiu hry,
- **Options** – táto sekcia menu slúži na nastavenie zvukových (Audio) a obrazových (Video) efektov. Audio efekty znázorňuje obrázok č.4 – je v ňom možné navoliť hlasitosť (Master volume), hlasitosť zvuku a hudby pri práci v menu a hlasitosť zvukov jednotiek a hudby hranej na pozadí počas samotnej hry.



Obrázok 4:Audio nastavenia

Sekcia Video (obrázok č.5) poskytuje nastavenia, z ktorých niektoré sa prejavia až po znuvuspustení aplikácie (sú označené \*), iné aby sa prejavili, je nutné začať novú hru (označené \*\*). Hviezdičkou neoznačené nastavenia majú okamžitý efekt. Ponúkajú sa nastavenia fullscreen, prípadne vertikálnej synchronizácie (synchronizácia vykresľovania s frekvenciou monitoru). Texture filter predstavuje kvalitu zobrazovania textúr pri ich zväčšení a mipmap filter je kvalita pri zmenšení,

- **Credits** – poskytuje informácie o tvorcovi hry, použitej hudbe a technológiách,
- **Quit** – Ukončuje hru, rovnaký efekt má stlačenie klávesy Escape.



Obrázok 5: Video nastavenia

### 2.3.2 Hracia obrazovka

Po spustení hry sa užívateľovi naskytne pohľad na hraciu plochu. Situácia je znázornená na obrázku č.6. V pravej časti je viditeľné menu, ktoré sa mení vzhľadom na označenú jednotku, prípadne vzhľadom na akciu, ktorá je s nejakou jednotkou prevádzaná (bude ukázané neskôr). Menu je možné skryť klepnutím na žltý štvorček v pravom hornom rohu obrazovky. V pravom dolnom rohu menu sa nachádza ikona predstavujúca aktívne segmenty, t.j. tie, ktoré sú viditeľné na mape. Segmenty sú v hre tri a jednotkám je možné v konfiguračných súboroch definovať, v ktorom segmente sa môžu pohybovať, prípadne pri budovách, v ktorom segmente budú stáť. Prepínanie medzi segmentmi umožňuje klepnutie na už spomínanú ikonku v dolnom pravom rohu. Horný pravý roh je vyhradený pre miniatúrnu mapku (radar mapku), ktorá zobrazuje aktuálny segment, resp. segmenty a tiež jednotky, ktoré sa v danom segmente nachádzajú. Táto mapka je kópiou veľkej hracej mapy. Sivý rámik na radar mapke vymedzuje plochu práve viditeľnú na veľkej hracej mape. V strednej časti menu sú viditeľné ikonky s akciami, ktoré môžu jednotky prevádzať. Pokiaľ žiadna jednotka nie je vyznačená, tieto ikony sú neprístupné (je tomu tak aj na obrázku č.6). Pred tým než si popíšeme tieto akcie, za povšimnutie stojí kurzor myši v tvare šípky sivej farby v dolnej časti obrázku č.6. Tvar kurzoru sa počas hry mení v závislosti na prevádzanej akcii.





Obrázok 6: Hracia plocha

Teraz už k samotným akciám jednotiek. Obrázok č.7 ukazuje ešte raz túto časť menu. Po poradí ikonky predstavujú akcie:

- **Stay** – ak je jednotka v pohybe alebo strieľa, po klepnutí na túto ikonu sa zastaví,
- **Move** – jednotka sa pohne na miesto, na ktoré užívateľ klepol pravým tlačidlom myši po označení jednotky,
- **Attack** – označená jednotka začne útočiť na inú jednotku, ktorá bola označená pravým tlačidlom myši,
- **Mine** – označená jednotka dôjde k zdroju, na ktorý užívateľ klepol pravým tlačidlom myši a začne ťažiť,
- **Repair** – označená jednotka dôjde k budove, ktorú má opravovať a následne ju začne opravovať,
- **Build** – jednotka začne stavať vybranú budovu na vybranom mieste (pokiaľ je to možné).



Obrázok 7: Akcie jednotiek

Na úzkom pruhu v dolnej časti obrazovky sú informácie o materiáloch, ktoré hráčove jednotky získali (množstvo zlata, dreva a uhlia), ďalej informácie o množstve jedla a energie. Veľká hracia mapa ukazuje jednotlivé hráčove aj nepriateľské jednotky, čierna oblasť mapy doposiaľ nebola preskúmaná a je zahalená do vojnovnej hmly, tzv. warfogu.

Po klepnutí ľavým tlačidlom myši na nejakú jednotku sa situácia na hracej ploche aj v menu zmení (popisuje ju obrázok č.7). Dôjde k označeniu jednotky, nad ktorou (v prípade budovy vedľa nej) sa objaví farebný pruh udávajúci, koľko života jednotke zostáva. Tento pruh môže mať tri farby:

- **Zelenú** – udáva rozmedzie života od 100% - 70% života,

- **Žltú** – udáva rozmedzie života od 69% - 30%,
- **Červenú** – menej ako 30%.

V pravej časti obrazovky, v strede menu, je ukázaný obrázok jednotky, ktorá bola označená, s jej pomenovaním a s informáciami o aktuálnom živote a maximálnom živote. Rovnako sú zobrazené ďalšie podrobné informácie (obrázok č.8):

- **Armor** – sila štítu a schopnosť vyhýbať sa,
- **Damage** – minimálny a maximálny rozsah sily zbrane, za znakom '/' je vypísaný polomer ničivého okolia dopadu náboja,
- **Range** – minimálny a maximálny dosah zbrane + presnosť zásahu zbrane,
- **View** – polomer okolia, ktorý jednotka okolo seba odkryje,
- **Max speeds** – maximálna rýchlosť, ktorú jednotka dokáže vyvinúť v jednotlivých segmentoch,
- **Food/Energy** – koľko jedla, resp. energie jednotka odoberie pri svojom vzniku zo zásob daného hráča,
- **Materials** – symboly materiálov, ktoré jednotka dokáže ťažiť, prípadne u zdrojov je to materiál, ktorý zdroj poskytuje,
- **Capacity** – táto informácia sa vyskytuje len pri zdrojoch a udáva aktuálny stav daného materiálu v zdroji.



Obrázok 8: Informácie o jednotke

Nie všetkým jednotkám sa zobrazujú úplne všetky informácie, napr. jednotkám, ktoré nemajú povolené ťažiť žiaden materiál, chýba informácia **Materials** a pod.

Pod týmito popismi v menu sú povolené akcie, ktoré môže jednotka prevádzať. Všetky akcie boli už popísané (obrázok č.8). V tmavšej časti menu sú viditeľné typy obrany jednotiek (obrázok č.9) a to po poradí:

- **Stay** – jednotka zotrúva bez pohybu na svojej pozícii aj v prípade ohrozenia cudzou jednotkou,
- **Guard** – jednotka začne útok v prípade, že v jej dostrele je nepriateľská jednotka,
- **Offensive guard** – podobne ako Guard, naviac, ak sa nepriateľská jednotka vzdiali, obranná jednotka ju začne prenasledovať s cieľom pokračovať v útoku,
- **Agressive guard** – podobné ako Offensive guard, naviac obranná jednotka sa rozbehne za nepriateľskou jednotkou aj v prípade, že na ňu nedostrelí, ale ju vidí.



Obrázok 9: Obranné menu

**POZNÁMKA:** V prípade, že je užívateľom označených viacero jednotiek naraz, v menu je zobrazený obrázok jedného typu jednotiek, výpis „Multi selection“ a ako dostupné sú ukázané len tie akcie, ktoré môžu prevádzať všetky jednotky selekcie.

### 2.3.3 Akcie s jednotkami

Nasledujúci odstavec popisuje, akým spôsobom je možné prevádzať akcie s označenými jednotkami. Pri týchto akciách sa mení kurzor myši podľa toho, či je akcia povolená alebo

zakázaná. Pri popise sa často budeme odvolávať na konfiguračné súbory, ktoré boli popísané v samostatnej časti dokumentácie. Akciami rozumieme:

- **Chodenie** – jednotku označíme ľavým tlačidlom a klepnutím na hraciu plochu pravým tlačidlom určíme cieľ cesty. Jednotka sa okamžite pokúsi do cieľa dostať. Pokiaľ cieľ nie je dostupný, jednotka sa snaží dostať k nemu aspoň čo najbližšie. Samozrejme, jednotky sa môžu pohybovať len v rámci mapy, pokus o pohyb mimo mapy bude zamietnutý. Rovnako je možné označiť viacero jednotiek, ktoré vytvoria formáciu, a poslať ich na nejaké miesto mapy,
- **Ťaženie** – niektoré jednotky majú schopnosť ťažiť istý druh materiálu. Túto vlastnosť je možné nadefinovať v konfiguračnom súbore a pri hre je viditeľná v menu na pravej strane, ako bolo popísané v sekcii o menu. Jednotku označíme a klepneme pravým tlačidlom na zdroj, v ktorom jednotka môže ťažiť. Jednotka do zdroja vojde a po vyťažení určitého množstva materiálu sa zo zdroja vydá na cestu do najbližšej budovy, ktorá prijíma ňou ťažený materiál. Opäť, pokiaľ takáto budova neexistuje, jednotka si udrží vyťaženy materiál, ale zostane stáť vedľa zdroja a ohlásí absenciu prijímajúcej budovy. Jedna jednotka môže ťažiť aj viaceré druhy materiálu,

- **Stavba budov** – schopnosť stavať budovy má jednotka opäť definovanú v konfiguračnom súbore. Klepneme na jednotku, potom v menu na pravej strane vyberieme akciu „Build“ a v dolnej časti menu vyberieme typ budovy, ktorý chceme postaviť. Keďže na postavenie budovy je potrebný materiál, dôjde pri postavení k zníženiu rezerv hráča, ktorý bude budovu vlastniť. Informácie o tom, koľko a akého materiálu je potrebné k postaveniu daného typu budovy môže hráč vidieť, keď myšou postojí na obrázku popisujúcom danú budovu v dolnej časti menu pri stavaní (obrázok č.10 popisuje začiatok stavby - s označenou jednotkou a informáciami o množstve potrebného materiálu).

Nedostatok materiálu spôsobí, že budova postavená nebude. Pri následnom pohybe kurzoru po mapke ikonka kurzoru predstavuje stavanú budovu. Podklad kurzoru sa mení na zeleno a červeno v závislosti na tom, či na miesto na mape,



Obrázok 11: Stavba začiatok



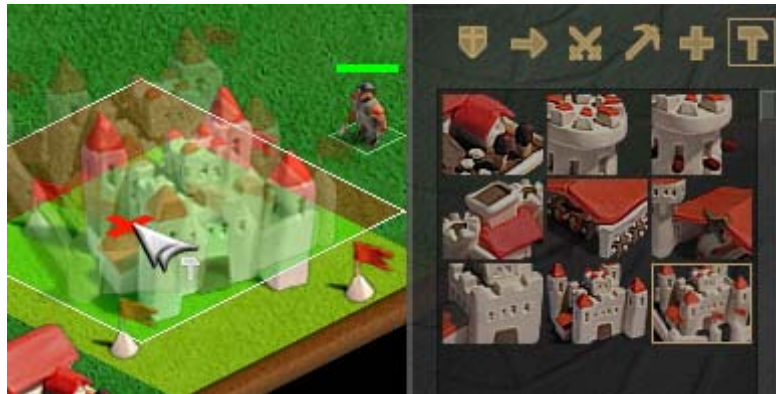
Obrázok 10: Stavba budovy



nad ktorým sa nachádza, je možné budovu postaviť alebo nie (obrázok č 11). Zelená značí pozitívny výsledok. Po klepnutí pravým tlačidlom na mapu označíme miesto, kde bude v budúcnosti budova stáť. Stavajúca jednotka na toto miesto príbehne a začne budovu stavať. Stavanie nie je hromadná akcia, to znamená, že začať stavbu môže len jedna jednotka, ostatné jej ale môžu prísť na pomoc jednoduchým označením a klepnutím pravým tlačidlom na stavanú budovu. Po dostavaní je situácia oznámená hráčovi. Pri stavaní budov, ktoré prijímajú nejaký vyťažovaný materiál sa táto ich schopnosť prejaví až po ich úplnom dostavaní,

- **Upgrade budov** – je veľmi podobný stavaniu, prevádza sa až na určenie miesta stavby budovy rovnako.

Upgradovacie budovy sa stavajú na budovy, ktoré upgradujú, nie na mapu samotnú a pri začatí stavby preberajú život od jednotky, ktorú upgradujú (obrázok č.12 popisuje upgrade budovy, v menu je žltým rámčekom orámovaná budova, ktorá sa na upgrade



Obrázok 12: Upgrade

použije). “Cena“ stavanej jednotky, ktorú užívateľ môže vidieť v menu, keď na chvíľu zastaví myš na obrázku stavanej jednotky, je len orientačná a platí len v prípade, ak upgradovaná budova má plný život. Pokiaľ nie, cena sa zvyšuje v závislosti od aktuálneho života upgradovanej jednotky,

- **Oprava budov** – opravu je možné začať jednoduchým označením jednotky, ktorá môže opravovať daný typ budovy a následným klepnutím pravým tlačidlom myši na danú budovu. Pokiaľ jednotka vybraný typ budovy nedokáže opravovať, chápe to ako výzvu, aby k budove prišla. Podobne ako stavanie, aj oprava budovy stojí hráča čas a materiál. Suma potrebná na opravu je závislá od poškodenia jednotky a od jej typu,
- **Bojovanie** – jednotky schopné boja môžu začať útočiť na vlastné aj cudzie jednotky tak, že ich označíme, v menu vyberieme akciu bojovania a následne pravým tlačidlom ukážeme na jednotku, ktorú chceme zraniť. Pokiaľ je cieľová jednotka mimo dosah - bojujúca jednotka sa k nej snaží dostať na vzdialenosť, z ktorej ju už môže zasiahnuť. Rovnako ako pri ostatných akciách, tak aj pri tejto, bude počas výberu cieľovej jednotky zmenený kurzor myši. Všetky vlastnosti ohľadne zbraní, ich rozptylu, či účinnosti je opäť možné definovať v konfiguračnom súbore,
- **Skrývanie jednotiek** – druh aj počet jednotiek, ktoré môže daná jednotka skryť je opäť definovaný v konfiguračnom súbore. Jednotku je možné do inej jednotky vnoriť jednoduchým označením jednotky a klepnutím pomocou pravého tlačidla na jednotku úkrytu (obrázok č.13, žltá šípka označuje cieľovú jednotku). Podľa rozmerov daná jednotka zaberie istý počet miest v ukývajúcej jednotke. V súvislosti



Obrázok 13: Skrývanie

s budovami táto funkcia môže slúžiť na ukrytie jednotiek pred nepriateľom po dobu boja,

- **Výroba jednotiek** – nie je žiadnym prekvapením, že tak, ako predchádzajúce vlastnosti, aj táto je definovaná v konfiguračnom súbore. Jednotka teda obecné môže produkovať iné jednotky.

Označíme jednotku, potom v menu vyberieme akciu „Build“ a v menu sa nám ponúkne, ktoré jednotky môže produkujúca jednotka vyrábať.

Klepaním na obrázok jednotky,



Obrázok 14: Výroba jednotky

ktorú si prajeme vyrábať, sa budú tieto jednotky hromadiť v dolnej časti menu, spolu s poradovým číslom (obrázok č.14). V tejto časti bude tiež ukázaná informácia o tom, do akej časti je už vyrobená prvá z tejto fronty jednotiek (Progress). Každá vyrábaná jednotka opäť hráča stojí nejaký materiál, tento sa hráčovi odoberá postupne a pokiaľ dôjde k vyčerpaniu materiálu, výroba jednotky je pozastavená až do doby, než bude dostatok materiálu na pokračovanie výroby. Nad vyrábajúcou budovou sa v tomto prípade zobrazí výstražný trojuholníček so symbolom chýbajúceho materiálu.

## 2.3.4 Významy klávesových skratiek

### 2.3.4.1 Klávesové skratky

**POZNÁMKA:** Pri experimentovaní s nižšie popísanými klávesovými skratkami je nutné mať nastavenú anglickú klávesnicu!

Niektorým klávesom bol v hre Dark Oberon priradený špeciálny význam. Nasleduje ich výpis:

- **F5** – zobrazí spodný segment,
- **F6** – zobrazí stredný segment,
- **F7** – zobrazí vrchný segment,
- **F8** – zobrazí všetky segmenty,
- **F9** – zmení rozlíšenie na 640x480,
- **F10** – zmení rozlíšenie na 800x600,
- **F11** – zmení rozlíšenie na 1024x768,
- **Ctrl** – kláves pomáha pri postupnom označovaní jednotiek. Označíme skupinu jednotiek, stlačíme Ctrl a môžeme k už označenej skupine pridať ďalšie jednotky,

- **Ctrl + číslo** – označenú skupinu jednotiek združí do skupiny s daným číslom. Toto číslo sa následne objavuje u každej z jednotiek po jej označení (obrázok č.15). Očíslovanie skupín sa dá predefinovať tak, že sa na jednotku opäť klepne a stlačí Ctrl + číslo. Pokiaľ skupina jednotiek s daným číslom už existuje, bude rozpustená a bude vytvorená nová skupina s daným číslom a to z označených jednotiek. Medzi takto označenými skupinami je možné prepínať sa pomocou stlačenia príslušného čísla skupiny na klávesnici. Takýto spôsob označovania jednotiek môže dopomôcť k jednoduchšej manipulácii s celými skupinami,



Obrázok 15: Efekt skratky Ctrl+číslo

- **Alt + číslo** – nastaví skupinu jednotiek s daným číslom ako aktívnu a vycentruje hraciu plochu tak, aby táto skupina bola v jej strede,
- **Tab** – skrýva a odkrýva panely. Ak bolo v ľavom hornom rohu radaru stlačené tmavé tlačidlo, radar zostane vždy zobrazený,
- **U, u** – umožňuje dopisovanie hráča s ostatnými hráčmi v dolnej časti hracej plochy. Napísanú správu je možné odoslať stlačením klávesy Enter. Písať správy je možné i pri skrytých paneloch.
- **Q** – quit, ukončenie hry,
- **G** – low CPU móde, neprevádza sa vykresľovanie, čo spôsobí nízku spotrebu CPU,
- **S, s** – stay, pokiaľ jednotka napríklad kráča po mape, jej označením a následným stlačením tejto klávesy je možné ju zastaviť. Ostatné skratky už vypíšeme len pre informáciu, ich podrobný popis bol uvedený v odseku 3.2:
- **M, m** – move, **A, a** – attack, **I, i** – mine, **R, r** – repair, **B, b** – build.

### 2.3.4.2 Pohyby myši

- **Pohyb** – pohyb myškou do strán hracej obrazovky spôsobí pohyb po mape,
- **Šípky** – rovnako spôsobia pohyb po mape,
- **Alt + pohyb myšou** – pozícia myši zostáva zachovaná, mení sa len hracia plocha pod ňou,
- **Prostredné tlačidlo myši** – so stlačeným stredným tlačidlom myši je možné mapu uchopiť a následným pohybom myši hraciu plochu potiahnuť v smere pohybu,
- **Pohyb myši v radar. okienku** – pomocou stlačenia ľavého tlačidla myši na sivom obdĺžniku v radarovom okienku a následným pohybom so zachovaním stlačeného tlačidla je možné pohybovať sa po mape v hracom okne,
- **Zoom** – pootočením kolieska myši sa dá meniť priblíženie pohľadu na mapu. Rovnakého efektu je možné docieľiť pomocou kláves +, -.

## 2.4 Konfiguračný súbor config.cfg

Všetky nastavenia prístupné z hlavného menu hry sú uložené v špeciálnom konfiguračnom súbore config.cfg. Tento súbor navyše obsahuje i nastavenia, ktoré v menu nie sú. Ak súbor

neexistuje, tak sa pri spustení hry automaticky vyrobí nový s prednastavenými hodnotami. Prednastavené hodnoty sa použijú i v prípade, že v súbore niektoré položky chýbajú.

Nasleduje kompletný výpis položiek s ich typmi a povolenými hodnotami. T znamená typ parametru položky. Informácie o formáte konfiguračných súborov sa nachádzajú v kapitole o konfiguračných súboroch.

### 2.4.1 Nastavenia okna aplikácie

- **fullscreen** – T: **bool**. Zadáva, či sa má aplikácia spustiť v režime na celú obrazovku alebo v okne. **true** – celá obrazovka, **false** – okno,
- **resolution** – T: **string**. Rozlíšenie (rozmary) okna aplikácie. Povolené hodnoty sú: “640x480”, “800x600”, “1024x768”, “1152x864”, “1280x1024”, “1600x1200”,
- **vert\_sync** – T: **bool**. Použitie synchronizácie vykresľovania s frekvenciou monitora. **true** – zapnutá, **false** – vypnutá synchronizácia.

### 2.4.2 Video nastavenia

- **texture\_filter** – T: **string**. Kvalita zobrazenia textúr pri ich zväčšovaní. Povolené hodnoty: “nearest”, “linear”,
- **mipmap\_filter** – T: **string**. Kvalita zobrazenia textúr pri ich zmenšovaní. Povolené hodnoty: “none”, “nearest”, “linear”,
- **warfog\_color** – T: 3 x **byte**. Farba warfogu po zložkách (červená, zelená, modrá zložka). Každá zložka sa nachádza v intervale <0, 255>,
- **warfog\_intensity** – T: **byte**. Intenzita warfogu v intervale <0, 100>. 0 – úplne priehľadný warfog, 100 – úplne nepriehľadný,
- **show\_fps** – T: **bool**. Zadáva, či sa má na obrazovke vypisovať informácia o počte vykreslených snímkov za sekundu,
- **show\_disconnect\_warning** – T: **bool**. Zadáva, či sa má po odhlásení zo servera prípadne ukončení hry zobrazovať dialóg s varovaním,
- **max\_frame\_rate** – T: **int**. Maximálny povolený počet snímkov za sekundu. Celočíselná hodnota v intervale <0, 1000>,
- **map\_move\_speed** – T: **byte**. Rýchlosť pohybu mapy. V intervale <1, 100>,
- **map\_zoom\_speed** – T: **byte**. Rýchlosť približovania / oddiaľovania mapy. V intervale <1, 100>.

### 2.4.3 Audio nastavenia

- **snd\_master\_volume** – T: **byte**. Globálne nastavenie hlasitosti v intervale <0, 100>,
- **snd\_menu\_music\_volume** – T: **byte**. Nastavenie hlasitosti hudby v menu. Hodnota v intervale <0, 100>,
- **snd\_menu\_sound\_volume** – T: **byte**. Nastavenie hlasitosti zvukov menu. Hodnota v intervale <0, 100>,

- **snd\_game\_music\_volume** – T: **byte**. Nastavenie hlasitosti hudby v hre. Hodnota v intervale  $\langle 0, 100 \rangle$ ,
- **snd\_game\_sound\_volume** – T: **byte**. Nastavenie hlasitosti zvukov v hre. Hodnota v intervale  $\langle 0, 100 \rangle$ ,
- **snd\_menu\_music** – T: **bool**. Zadáva, či má v menu hrať hudba,
- **snd\_game\_music** – T: **bool**. Zadáva, či má v hre hrať hudba,
- **snd\_unit\_speech** – T: **bool**. Zadáva, či má byť zapnuté ozvučenie jednotiek pri ich označovaní a iných akciách.

#### 2.4.4 Užívateľské nastavenia

- **player\_name** – T: **string**. Meno hráča,
- **address** – T: **string**. Posledne použitá adresa pre pripojenie k inému hráčovi,
- **sensitivity** – T: **byte**. Citlivosť myši v intervale  $\langle 0, 100 \rangle$ .

#### 2.4.5 Sieťové nastavenia

- **net\_server\_port** – T: **int**. Port serveru v intervale  $\langle 1024, 65535 \rangle$  použitý pri vytváraní novej hry.

### 2.5 Dátové súbory

Všetky textúry pre grafické rozhranie menu a rôznych panelov v hre ako aj interné zvuky a hudba sú uložené v dátových súboroch nachádzajúcich sa v adresári /dat. Pre tieto súbory platí, že ich vnútorná štruktúra je pevná a nesmie sa meniť. V prípade, že by užívateľ mal záujem zmeniť napr. pozadie menu alebo jednotlivé tlačidlá, musí dodržať presné pravidlá.

#### 2.5.1 Súbor *fonts.dat*

Súbor *fonts.dat* obsahuje jednu skupinu textúr s jednou textúrou zobrazujúcou vedľa seba jednotlivé písmená abecedy a znaky používané pri zobrazovaní textu. Nedoporučuje sa túto textúru akokoľvek meniť. Jej forma je príliš zviazaná s aplikáciou.

#### 2.5.2 Súbor *cursors.dat*

V súbore *cursors.dat* sa nachádzajú textúry pre kurzor myši. Je tu uložená jedna skupina textúr obsahujúca 19 textúr v nasledujúcom poradí:

1. **default** – základný kurzor pre grafické rozhranie,
2. **select** – kurzor v prípade, že sa nachádza nad jednotkou,
3. **select\_plus** – kurzor v prípade, že sa nachádza nad jednotkou pri stlačení klávese Ctrl,
4. **can\_move** – kurzor pre akciu chodenia ak je povolené,
5. **cant\_move** – kurzor pre akciu chodenia ak je zakázané,
6. **can\_attack** – kurzor pre akciu útočenia ak je povolené,



7. **cant\_attack** – kurzor pre akciu útočenia ak je zakázané,
8. **can\_mine** – kurzor pre akciu ťaženia ak je povolené,
9. **cant\_mine** – kurzor pre akciu ťaženia ak je zakázané,
10. **can\_repair** – kurzor pre akciu opravovania ak je povolené,
11. **cant\_repair** – kurzor pre akciu opravovania je zakázané,
12. **can\_build** – kurzor pre akciu stavania ak je povolené,
13. **cant\_build** – kurzor pre akciu stavania ak je zakázané,
14. **can\_hide** – kurzor pre akciu skrývania ak je povolené,
15. **cant\_hide** – kurzor pre akciu skrývania ak je zakázané,
16. **eject** – kurzor pre akciu vyloženia skrytých jednotiek,
17. **circle\_in** – sekundárna textúra zobrazujúca šípky smerujúce dovnútra z kruhu. Používa sa v prípade ak je akcia jednotky povolená a má súvislosť so zmenou pozície jednotky,
18. **circle\_out** – sekundárna textúra zobrazujúca šípky smerujúce von z kruhu. Používa sa v prípade vykladania skrytých jednotiek,
19. **cross** – sekundárna textúra zobrazujúca kríž zákazu pre prípad nepovolennej akcie.

### 2.5.3 Súbor *gui.dat*

Textúry a zvuky grafického rozhrania sa nachádzajú v súbore *gui.dat*. Je tu šesť skupín textúr:

1. **panels** – obsahuje pozadie menu (*bg\_menu*), pozadie pre hlavný panel hry (*bg\_panel*) a textúru pre dialóg Credits (*bg\_credits*),
2. **action\_buttons** – textúry tlačidiel pre jednotlivé akcie. Ich význam je zrejmý z identifikátoru. Sú to v poradí: *bt\_stay*, *bt\_move*, *bt\_attack*, *bt\_mine*, *bt\_repair*, *bt\_build*. Za nimi nasledujú textúry pre tieto tlačidlá v stlačenom stave,
3. **menu\_buttons** – tu sa nachádzajú textúry pre tlačidlá jednotlivých menu a dialógov: *bt\_play*, *bt\_quic\_play*, *bt\_options*, *bt\_credits*, *bt\_quit*, *bt\_resume*, *bt\_create*, *bt\_connect*, *bt\_disconnect*, *bt\_back*, *bt\_video*, *bt\_audio*, *bt\_ok*, *bt\_yes*, *bt\_no*, *bt\_cancel*,
4. **segment\_buttons** – tlačidlá pre zobrazenie rôznych segmentov: *bt\_segment\_0*, *bt\_segment\_1*, *bt\_segment\_2*, *bt\_segment\_all*,
5. **labels** – nadpisy používané v dialógoch: *lbl\_display*, *lbl\_resolution*, *lbl\_texture\_filter*, *lbl\_mipmap\_filter*, *lbl\_master*, *lbl\_menu*, *lbl\_game*, *lbl\_map*, *lbl\_info*, *lbl\_name*, *lbl\_ip*, *lbl\_players*,
6. **guard\_buttons** – tlačidlá pre zmenu typu bránenia: *bt\_ignore*, *bt\_guard*, *bt\_offensive*, *bt\_aggressive*.

Zvuky sú nasledujúce:

1. **menu\_music** – hudba pre menu,
2. **button\_click** – zvuk pre klepnutie na tlačidlo menu,
3. **game\_music** – hudba pre hru,

4. **component\_click** – zvuk pre klepnutie na aktívny prvok menu ako je zaškrtávacie políčko a pod.

## 2.6 Slovník používaných termínov

- **rasa** – druh jednotiek hráča (napríklad ľudia, obry, mimozemšťania...). Každý hráč na mape musí mať jedinečnú rasu,
- **schéma** – prostredie, v ktorom sa hra odohráva (napríklad mesiac). V schéme sa definujú všetky elementy prostredia, z ktorých bude poskladaná mapa (jeden mesačný kráter),
- **mapel** – najmenšia adresovateľná jednotka mapy. Pojem vznikol podobným spôsobom ako slovo „pixel“ = **picture element**, teda „mapel“ = **map element**. V mapeloch sú uvádzané mnohé vlastnosti jednotiek v konfiguračných súboroch,
- **fragment** – schémový element, z ktorého sa tvorí povrch mapy. Fragmenty sú mapované na celý povrch jednej vrstvy mapy a určujú výšku a obtiažnosť každého mapelu,
- **segment** – jedna vrstva mapy. Mapa má 3 vrstvy: „podzemie“, „povrch“ a „vzduch“. Jednotky sa môžu pohybovať vo všetkých troch vrstvách. Každá z vrstiev má svoje textúry a svoje fragmenty,
- **warfog** – oblasť mapy, ktorú už hráč niekedy objavil, no momentálne na ňu nevidí žiadna jeho jednotka,
- **textúra** – dvojrozmerný obrázok používaný grafickou kartou,
- **hyperhráč** – fiktívny hráč, ktorý „vlastní“ schémové jednotky, budovy a hlavne zdroje. Jeho jednotky a budovy nemajú žiadnu (okrem „okrasnej“) funkciu. Zo zdrojov však ostatní hráči ťažia materiály potrebné pre rozvoj,
- **leader (siete)** – počítač, ktorý vytvára hru. Ostatní hráči sa na neho pripájajú. Leader vlastní všetkých počítačových hráčov – teda aj hyperhráča,
- **follower (siete)** – počítač, ktorý sa napája na už vytvorenú hru,
- **PathFinder** – funkcia používaná na hľadanie cesty v trojrozmernej mape,
- **leader (hľadanie cesty)** – jednotka, ktorá bola vybraná za „vodcu“ skupiny jednotiek pri skupinovom chodení. Kvôli optimalizácii a „držaniu formácie skupiny jednotiek“ sa cesta v mape hľadá len raz – práve pre vodcu (leadera) – ostatné jednotky ju len prevezmú,
- **event (fronta správ)** – typ správy používaný frontou správ. Event mení stav jednotky, ktorej je adresovaný,
- **request (fronta správ)** – typ správy používaný frontou správ. Request je žiadosť o vykonanie nejakej akcie a nemení stav jednotky, ktorej bol adresovaný.



# 3 Vytvorenie hry

Martin Košalko



### 3.1 Konfiguračné súbory

Projekt Dark Oberon nie je jediná konkrétna strategická hra s pevne definovanými jednotkami, prostredím a mapami. V svojej podstate je projekt výpočtovým strojom (šablónou) pre rôzne inštancie strategických hier typu Warcraft II (C&C), ktoré je možné definovať práve pomocou konfiguračných súborov. Pre úspešné vytvorenie konkrétnej hry je potrebné nadefinovať typy jednotiek spolu s ich vlastnosťami (rasu), prostredie, v ktorom sa budú dané typy pohybovať (schému) a nakoniec z elementov prostredia poskladať konkrétnu mapu. Význam a vzájomné vzťahy jednotlivých typov súborov je vidieť najlepšie na príklade:

- **Schéma** – mesačná krajina
- **Rasy** – astronauti, mimozemšťania
- **Mapa** – jeden konkrétny mesačný kráter

Nasleduje presnejší popis:

- **Schéma** - Už na prvý pohľad je jasné, že základom každej hry je schéma, teda prostredie, v ktorom sa bude hra odohrávať. Schéma nie je závislá na ostatných súboroch, no musí k nej existovať takzvaná schémová rasa. Je to rasa špecifická pre dané prostredie – v našom prípade by to mohli byť mesačný „domorodci“ spolu s ich prírodnými zdrojmi. Pre samotnú hru nie sú dôležité jednotky a budovy schémovej rasy, ktoré plnia viac-menej okrasnú funkciu. Dôležité sú práve prírodné zdroje, z ktorých budú ostatné rasy ťažiť materiály potrebné pre svoj rozvoj.
- **Rasy** - Rasy sú závislé na schéme. V hlavičke definície rasy musia byť uvedené schémy (minimálne jedna platná), v ktorých môže byť táto rasa použitá. Je totiž ťažko predstaviteľné, aby na mesačnom povrchu bojovali napríklad potápači. Závislosť rasy na schéme spočíva napríklad v definícii typov povrchov, kde sa môže jednotka pohybovať, alebo v definícii typov materiálov, ktoré môže jednotka ťažiť.
- **Mapy** - Mapa je vytvorená pomocou elementov definovaných v schéme (a teda je na schéme závislá). Podobne ako v rasách je v hlavičke mapy uvedená schéma, z ktorej sa majú čerpať stavebné elementy. Na rozdiel od rás však musí byť uvedená práve jedna schéma. Mapa tvorí herný plán, v ktorom vystupujú hráči (či už počítačový alebo skutoční). Keďže program nedokáže nijako graficky rozlíšiť jednotky dvoch rôznych hráčov s tou istou rasou, je potrebné, aby mal každý hráč na mape inú rasu. Preto sa v mape okrem maximálneho počtu hráčov definujú aj rasy, z ktorých sa môže vyberať. Ak bude v mape uvedených menej rás ako maximálny počet hráčov, nebude možné vybrať do hry maximálny počet hráčov, ale iba počet rovný počtu uvedených rás. Každá rasa uvedená v mape musí byť samozrejme použiteľná v schéme mapy.

#### 3.1.1 Spoločné črty načítaných položiek a použité skratky

- **textúry** – Ak bude zadaná hodnota (textový identifikátor skupiny textúr), ktorá neexistuje v dátovom súbore príslušnej rasy, program ohlásí chybu. Zadáva sa práve jeden identifikátor skupiny textúr.
- **zvuky** - Je možné zadať hodnotu „none“, čo znamená, že sa nepoužije žiadny zvuk. Ak však bude zadaná hodnota, ktorá neexistuje v dátovom súbore príslušnej rasy, program ohlásí chybu. Je možné zadať niekoľko identifikátorov, pričom v hre sa vždy náhodne vyberie jeden z nich.

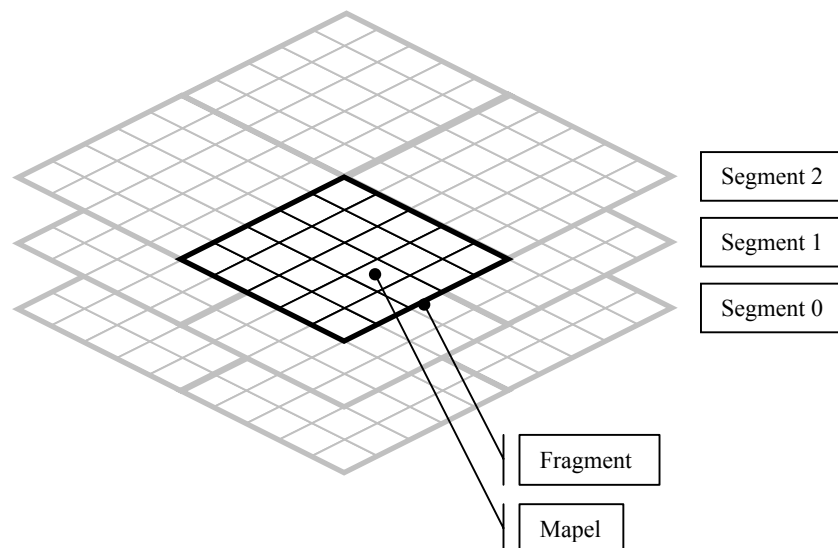
Pri popise jednotlivých položiek konfiguračných súborov sú v zátvorkách použité skratky s nasledovným významom:

- **T (type)** - očakávaný typ premennej (string, integer, byte, bool, float),
- **R (range)** – očakávaný rozsah hodnôt,
- **D (default value)** – v prípade, že bude zadáný zlý typ, rozsah, alebo neplatná hodnota, bude použitá preddefinovaná hodnota,
- **N (necessary item)** – určuje, či je položka povinná.

### 3.1.2 Popis súborov schém

Súbory schém majú príponu „sch“ a ich jednoznačný identifikátor, používaný ako odkaz v ostatných konfiguračných súboroch, je meno súboru bez prípony. Ako už bolo spomenuté, ku každej schéme musí existovať tzv. schémová rasa. Všetky dostupné súbory schém (spolu so súborami schémových rás) sú lokalizované v podadresári „schemes“ koreňového adresára hry. Schéma definuje prostredie, v ktorom sa hra bude odohrávať. Prostredie hry je postupne možné zložiť z niekoľkých typov elementov. Najmenším terénnym elementom (zložkou prostredia) je terénny typ - „**Terrain type**“. Terénne typy sa pomocou štvorcových zoskupení terénnych typov – „**Fragmentov**“ mapujú na najmenšie adresovateľné jednotky mapy – „**Mapely**“. Z mapelov sú poskladané tri vrstvy prostredia – „**Segmenty**“. Segmenty sú vždy 3 a predstavujú podpovrchovú vrstvu (podzemie), povrchovú vrstvu (zem) a nadpovrchovú vrstvu (vzduch).

V schéme sa samozrejme mapely nedefinujú, no sú tu uvedené pre lepšie pochopenie vzťahov medzi elementmi prostredia. Podobne sa v schéme nedefinuje ani konečná podoba segmentu (tá sa definuje samozrejme v konfiguračnom súbore mapy), no v schéme sú pre každý segment definované typy fragmentov a terénnych typov.



Okrem elementov prostredia obsahuje definíciu typov materiálov a hlavičku schémy so základnými informáciami o schéme.

#### 3.1.2.1 Hlavička

Hlavička obsahuje základné informácie o schéme.

- **name** – meno schémy. Samotný program ho nijako nepoužíva, je to iba informácia. (*T: string, R: 1024 znakov, D: „“, N: nie*),
- **author** – meno autora súboru. V programe sa nevyužíva, je to len informácia navyše. (*T: string, R: 1024 znakov, D: „“, N: nie*).

```
name "Plastic World"
author "PP team"
```

### 3.1.2.2 Sekcia <Materials>

Sekcia obsahuje definície typov materiálov. Definované môžu byť maximálne 4 typy materiálov, no musí sa podariť načítať aspoň jeden. Sekcia obsahuje nasledujúce položky a podsekcie:

- **count** – počet typov materiálov. Program sa pokúsi prečítať zadaný počet sekcií <Material>. Ak sa to nepodarí, načítavanie schémy skončí chybou. (*T*: **integer**, *R*: maximálny počet typov materiálov = 4 >= *r* >= 1, *D*: 1, *N*: áno),
- <Material> - podsekcia obsahujúca definíciu jedného typu materiálu.
  - **id** – jednoznačný textový identifikátor typu materiálu. (*T*: **string**, *R*: 1024 znakov, *D*: „Material X“ – názov sekcie, *N*: nie),
  - **name** – meno materiálu zobrazujúce sa v hre ako popis. (*T*: **string**, *R*: 1024 znakov, *D*: **id** – identifikátor materiálu, *N*: nie),
  - **tg\_id** – položka pre materiál určuje textový identifikátor skupiny textúr (definovaný v dátovom súbore príslušnej schémy) obrázka materiálu. Využíva sa ako ikona pri zobrazení aktuálneho stavu materiálu hráča a ako identifikátor nedostatku materiálu. (*T*: **string**, *R*: 1024 znakov, *D*: -, *N*: áno).

```
<Materials>
  count 1
  <Material 0>
    id "gold"
    name "Gold"
  </Material 0>
</Materials>
```

### 3.1.2.3 Sekcie <Segments> a <Segment>

Jedinou úlohou sekcie <Segments> je obaliť 3 podsekcie <Segment 0>, <Segment 1> a <Segment 2>. Všetky dôležité informácie sú až v uvedených podsekciami, kde je možné nájsť definície prostredia.

- **name** – meno segmentu. (*T*: **string**, *R*: 1024 znakov, *D*: „Segment X“ – názov sekcie, *N*: nie),
- **count** – počet terénnych typov daného segmentu. Program sa pokúsi načítať zadaný počet sekcií <Terrain type>. Ak sa to nepodarí, načítavanie schémy skončí s chybou. (*T*: **integer**, *R*: >= 0, *D*: 0, *N*: áno),
- <Terrain type> - podsekcia obsahujúca podrobné informácie o type terénu v danom segmente,
- <Fragments> - podsekcia obsahujúca definície fragmentov daného segmentu,
- <Layers> - podsekcia definujúca vrstvy – špeciálne terénne elementy, ktoré sa dajú umiestniť kdekoľvek v mape a lokálne menia vlastnosti povrchu, na ktorom sú umiestnené. Vrstvy sa môžu navzájom aj prekrývať a nie sú vykreslené vždy pod jednotkami v mape. Dobrý príklad je bažina, ktorá je vždy pod jednotkami a mení obtiažnosť terénu,

- **<Objects>** - podsekcia definujúca terénne elementy podobné vrstvám, ale objekty sa nemôžu vzájomne prekrývať a sú zahrnuté do triediacich algoritmov textúr. Ako príklad možno uviesť strom.

```
<Segment 1>
  count 9
  name "Earth"

  <Terrain type 0>
  ...
</Terrain type 0>
<Fragments>
  ...
</Fragments>
<Layers>
  ...
</Layers>
<Objects>
  ...
</Objects>
</Segment 1>
```

### 3.1.2.4 Sekcia <Terrain type>

Sekcia obsahuje presnú definíciu terénneho typu.

- **name** – meno terénneho typu. (*T*: **string**, *R*: 1024 znakov, *D*: „Terrain type X“ – názov sekcie, *N*: nie),
- **layer** – vrstva terénneho typu. Vrstvu možno chápať ako výšku terénu. Práve pomocou intervalu dvoch vrstiev sa definuje, kam môže jednotka chodiť - je teda možné definovať, že jednotka sa môže pohybovať po všetkých vrstvách (pre lepšiu predstavu sa hodí slovo výškach) zo zadaného intervalu. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: nie),
- **difficulty** – položka definuje, ako obtiažne sa po danom povrchu pohybuje. Očakávané je číslo z intervalu  $<0, 1>$ , kde 0 znamená, že rýchlosť pohybu jednotky je rovná maximálnej rýchlosti jednotky v danom segmente a zo zvyšujúcou hodnotou sa rýchlosť jednotky priamo úmerne znižuje, až 1 znamená, že jednotka sa na danom povrchu nedokáže pohybovať. (*T*: **float**, *R*:  $1 \geq r \geq 0$ , *D*: 0, *N*: nie).

```
<Terrain type 6>
  name "Rocks"
  layer 30
  difficulty 0
</Terrain type 6>
```

### 3.1.2.5 Sekcia <Fragments>

Sekcia obsahuje definície pomenovaných štvorcových zoskupení terénnych typov.

- **tg\_default\_id** - textový identifikátor skupiny textúr (definovaného v dátovom súbore príslušnej schémy) obrázka prednastaveného terénneho typu o rozmere 1x1 mapel. Tento sa použije v prípade, že celá mapa nie je pokrytá fragmentmi – vyplnia sa ním prázdne miesta. (*T*: **string**, *R*: 1024 znakov, *D*: -, *N*: nie),



- **default\_terrain\_id** – vrstva prednastaveného terénneho typu o rozmere 1x1 mapel, ktorým sa vyplnia miesta nepokryté fragmentmi. Je možné zadať hodnotu „none“, čo znamená, že terénny typ bude priehľadný. (*T: integer, R: >= 0, D: 0, N: áno*),
- **count** – počet typov fragmentov daného segmentu. Program sa pokúsi načítať zadaný počet sekcií <. Ak sa to nepodarí, načítavanie schémy skončí chybou. (*T: integer, R: >= 0, D: 0, N: áno*),
- **<Fragment>** - podsekcia definujúca vlastnosti jedného fragmentu.

```
<Fragments>
  tg_default_id "default"
  default_terrain_id 10
  count 80
  <Fragment 4>
  ...
</Fragment 4>
</Fragments>
```

### 3.1.2.6 Sekcia <Fragment>

Sekcia obsahuje definíciu jedného konkrétneho fragmentu.

- **size** – veľkosť fragmentu v mapeloch. Fragment je vždy štvorcový. (*T: byte, R: >= 1, D: 1, N: nie*),
- **tg\_id** – textový identifikátor skupiny textúr (definovaného v dátovom súbore príslušnej schémy) povrchu fragmentu. Je možné zadať hodnotu „none“, čo znamená, že sa nepoužije žiadna textúra a fragment bude plne priehľadný. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **terrain\_id** – Očakávaných je **size\*size** výšok (vrstiev) terénnych typov, ktoré určujú výšku danej časti fragmentu. Ak je zadaných málo hodnôt, načítavanie schémy skončí s chybou. Ak je zadaná hodnota neexistujúcej vrstvy, do úvahy sa vezme najbližšia vyššia existujúca vrstva. (*T: integers, R: >= 0, D: 0, N: áno*).

```
<Fragment 4>
  size 3
  tg_id "rocks_se"
  terrain_id 30 30 30 30 30 10 30 30 30
</Fragment 4>
```

### 3.1.2.7 Sekcie <Layers> a <Objects>

- **count** – počet objektov (vrstiev), ktoré sa budú načítavať. Ak ich nebude dosť, načítavanie schémy skončí s chybou. (*T: integer, R: >= 0, D: 0, N: áno*),
- **<Object>**, **<Layer>** - definície jednotlivých objektov a vrstiev.

```
<Layers>
  count 1
  <Layer 1>
  ...
</Layer 1>
</Layers>
```

### 3.1.2.8 Sekcie <Layer> a <Object>

Elementy prostredia <Layer> a <Object> majú úplne totožné vlastnosti, no sú chápané trochu inak. Zatiaľčo vrstva je chápaná ako „dvojmerná“, objekt je „trojmerný“. Dve vrstvy umiestnené na mape sa môžu prekryvať (terén modifikuje tá, čo bola pridaná ako posledná) a vykresľujú sa vždy pod jednotkami toho istého segmentu. Dva objekty sa prekryvať nesmú a ich textúry sú ako všetky ostatné jednotky triedené. Aj objekty a aj vrstvy slúžia na lokálne zmeny terénu.

- **id** – textový identifikátor objektu (vrstvy), ktorý sa používa ako odkaz na typ v ostatných konfiguračných súboroch.. (*T: string*, *R: 1024 znakov*, *D: „Object X“*, *N: nie*),
- **name** – názov objektu. (*T: string*, *R: 1024 znakov*, *D: id* – identifikátor objektu, *N: nie*),
- **width** – šírka podstavy objektu (vrstvy) v mapeloch. (*T: byte*, *R: >= 1*, *D: 1*, *N: nie*),
- **height** – dĺžka podstavy objektu (vrstvy) v mapeloch. (*T: byte*, *R: >= 1*, *D: 1*, *N: nie*),
- **tg\_id** – textový identifikátor skupiny textúr (definovaný v dátovom súbore príslušnej schémy) daného objektu (vrstvy). (*T: string*, *R: 1024 znakov*, *D: -, N: áno*),
- **terrain\_id** – položka má totožný význam ako rovnomenná položka sekcie <Fragment>.

```
<Object 0>
  id "tree"
  name "Tree"
  width 2
  height 2
  tg_id "o_tree"
  terrain_id 10 10 10 10
</Object 0>
```

### 3.1.3 Popis súborov rás

Súbory rás majú príponu „rac“ a ich jednoznačný identifikátor používaný ako odkaz v ostatných konfiguračných súboroch je meno súboru bez prípony. Obsah súboru a jeho umiestnenie v adresárovej štruktúre závisí od toho, či súbor definuje schémovú, alebo hráčsku rasu. Schémová rasa je umiestnená v podadresári „schemes“ koreňového adresára hry a meno súboru (až na príponu) musí byť totožné s menom súboru schémy. Ostatné (hráčske) súbory rás sú umiestnené v podadresári „races“ koreňového adresára hry, kde je pre každú rasu očakávaný ešte podadresár s rovnakým názvom ako rasa. Hlavným dôvodom existencie konfiguračných súborov rás je možnosť definovať druhy jednotiek, budov a zdrojov a ich podrobných vlastností. Okrem toho súbor obsahuje hlavičku s informáciami o rase ako takej. Obsahový rozdiel hráčskych a schémovej rasy spočíva v definícii zdrojov v schémovej rase.

#### 3.1.3.1 Hlavička

Sekcia obsahuje údaje platné pre celú rasu a formálne informácie o súbore.

- **name** – meno rasy. Meno zadané v tejto položke sa zobrazí v menu pri vyberaní rasy. (*T: string*, *R: 1024 znakov*, *D: meno súboru rasy*, *N: nie*),

- **author** – meno autora súboru. V programe sa nevyužíva, je to len informácia navyše. (T: **string**, R: 1024 znakov, D: „“, N: nie),
- **schemes** – v položke sú očakávané textové identifikátory schém, pod ktorými môže byť rasa použitá. Ak nebude zadaná žiadna schéma, rasu nebude možné vybrať v menu. (T: **string**, R: 1024 znakov, D: „“, N: nie),
- **tg\_food\_id**, **tg\_energy\_id** – textový identifikátor skupiny textúr (definovaný v dátovom súbore príslušnej rasy) obrázka jedla (energie) – používa sa ako ikona pri zobrazení aktuálneho stavu jedla (energie) hráča a ako identifikátor nedostatku jedla (energie). (T: **string**, R: 1024 znakov, D: -, N: áno),
- **tg\_burning\_id** – textový identifikátor skupiny textúr plameňa definovaný v dátovom súbore príslušnej rasy. Textúra je využitá pri budovách a zdrojoch, ktoré nemajú definovanú vlastnú položku **tg\_burning\_id**. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **snd\_burning** – textový identifikátor zvuku horiacej jednotky (definovaný v dátovom súbore príslušnej rasy). Zvuk sa použije pri budovách a zdrojoch, ktoré nemajú definovaný vlastný zvuk. Prehráva sa v okamihu označenia horiacej jednotky. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **snd\_dead** – textový identifikátor zvuku umierajúcej pohyblivej jednotky. Zvuk sa použije v prípade, že jednotka nemá definovanú položku **snd\_dead**. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **snd\_explosion** – textový identifikátor zvuku explodujúcej (padajúcej) budovy, prípadne zdroja. Zvuk sa použije len v prípade, že budova (zdroj) nemá definovanú položku **snd\_explosion**. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **snd\_error** – textový identifikátor zvuku (definovaného v dátovom súbore), ktorý sa použije pri neúspešnom pokuse o vykonanie nejakej akcie kurzorom. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **snd\_placement** – textový identifikátor zvuku, ktorý sa použije pri umiestnení budovy do mapy kurzorom myši. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **snd\_construction** – textový identifikátor zvuku, ktorý sa použije, keď stavaná budova zmení textúru, prípadne pri označení stavanej budovy. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **snd\_building\_selected** – textový identifikátor zvuku použitý pri označení budovy alebo zdroja a to len v prípade, že budova (zdroj) nemá definovaný vlastnú položku **snd\_selected**. (T: **string**, R: 1024 znakov, D: -, N: áno),

```
name "Plastic Humans"
author "[:-)] PP team"
schemes "plastic"

tg_food_id "food"
tg_energy_id "energy"
tg_burning_id "building_burning"
snd_error error
snd_placement placement
snd_construction construction
snd_burning burning
```

```
snd_dead dead
snd_explosion explosion1 explosion2 explosion3
snd_building_selected building_selected
```

### 3.1.3.2 Sekcia <Units>

Sekcia obsahuje definície typov a vlastností pohyblivých jednotiek. Podľa akcií, ktoré môžu pohyblivé jednotky vykonávať, ich rozdeľujeme na dve skupiny: bojovníci (FORCE UNITS) a pracanti (WORKER UNITS), ktorí môžu navyše ťažiť materiály zo zdrojov, stavať budovy a opravovať ich. Sekcia obsahuje nasledujúce položky a sekcie:

- **count** - počet typov pohyblivých jednotiek rasy. Program sa pokúsi prečítať zadaný počet sekcií <Unit>. Ak sa to nepodarí, načítavanie skončí chybou. (*T: integer*, *R: >=0*, *D: 0*, *N: áno*),
- **<Unit X>** - definícia jedného konkrétneho typu pohyblivej jednotky.

```
<Units>
  count 10
  <Unit 0>
  ...
  </Unit 0>
</Units>
```

### 3.1.3.3 Sekcia <Unit>

Sekcia obsahuje definíciu typu jednotky.

*Spoločné vlastnosti bojovníkov (FORCE UNIT) a pracantov (WORKER UNIT):*

- **id** – textový identifikátor typu jednotky využívaný na jeho jednoznačnú identifikáciu. (*T: string*, *R: 1024 znakov*, *D: „Unit X“ = názov aktuálnej sekcie*, *N: nie*),
- **name** – meno druhu jednotky používané ako popis v hre. (*T: string*, *R: 1024 znakov*, *D: id = identifikátor typu jednotky*, *N: nie*),
- **size** – šírka a výška druhu jednotky v mapeloch, t.j. najmenších adresovateľných jednotiek plochy (*T: byte*, *R: 15 >= r > 0*, *D: 1*, *N: nie*),
- **materials** – množstvo materiálov potrebných na vyrobenie (vycvičenie) typu jednotky. Musí byť uvedených aspoň toľko čísel, koľko bolo definovaných materiálov v príslušnej schéme. (*T: floats*, *R: >= 0*, *D: 0*, *N: nie*),
- **max\_life** – maximálna hodnota života typu jednotky (*T: integer*, *R: >= 1*, *D: 1*, *N: nie*),
- **max\_speed** – maximálna hodnota rýchlosti typu jednotky pre každý segment v mapeloch za sekundu. Očakávané sú tri čísla (počet segmentov) (*T: floats*, *R: >= 0.01*, *D: 0.01*, *N: nie*),
- **max\_rotation\_speed** – maximálna hodnota rýchlosti otáčania typu jednotke pre každý segment v stupňoch za sekundu. Očakávané sú tri čísla (počet segmentov) (*T: integers*, *R: >= 1*, *D: 1*, *N: nie*),
- **selection\_height** – výška rámčeka označujúceho jednotku v pixeloch (*T: byte*, *R: >= 0*, *D: 40*, *N: nie*),

- **burning\_position** – X-ová a Y-ová relatívna súradnica plameňa voči jednotke v pixeloch. Očakávané sú 2 hodnoty. (T: **float, float**, R: -, D: 0,20, N: nie),
- **item\_type** – položka označuje typ jednotky. Očakáva sa jedna z hodnôt: „f“- bojovník (FORCE UNIT), „w“- pracant (WORKER UNIT). V prípade, že hodnota je „w“ sa navyše budú načítavať položky špecializované pre pracantov. (T: **string**, R: 1 znak, D: „f“, N: áno),
- **view** – dohľad jednotky od svojho okraja v mapeloch. (T: **byte**, R: MAX\_MAP\_SIZE = 240 >= r >= 1, D: 1, N: nie),
- **energy** – údaj hovoriaci o tom, koľko energie pridáva (kladná hodnota), alebo potrebuje (záporná hodnota) každá jednotka daného typu. (T: **integer**, R: -, D: 0, N: nie),
- **food** – údaj hovoriaci o tom, koľko jedla pridáva (kladná hodnota), alebo potrebuje (záporná hodnota) každá jednotka daného typu. (T: **integer**, R: >= 0, D: 0, N: áno),
- **move\_terrain\_id** - minimálna a maximálna výška terénu, kde sa môže jednotka pohybovať v danom segmente. Očakávané sú tri (počet segmentov) dvojice čísel. Program kontroluje, či načítavané minimum je menšie ako maximum (ak nie je, za maximum sa vezme minimum). (T: **integers**, R: >= 0, D: 0, N: nie),
- **land\_terrain\_id** - minimálna a maximálna výška terénu, kde môže jednotka pristáť v danom segmente (ale nie pohybovať sa). Očakávané sú tri (počet segmentov) dvojice čísel. Program kontroluje, či načítavané minimum je menšie ako maximum (ak nie je, za maximum sa vezme minimum). (T: **integers**, R: >=0, D: 0, N: nie),
- **min\_exists\_segment** – index najnižšieho segmentu, kde môže jednotka existovať. (T: **byte**, R: počet segmentov = 3 > r >= 0, D: 0, N: nie),
- **max\_exists\_segment** - index najvyššieho segmentu, kde môže jednotka existovať. (T: **byte**, R: počet segmentov > r >= **min\_exists\_segment**, D: 0, N: nie),
- **min\_max\_visible\_segment\_id** – pre každý segment definovaná dvojica minimálneho a maximálneho indexu segmentu, kde jednotka vidí. Očakávané sú tri (počet segmentov) dvojice indexov, pričom sa kontroluje, aby načítavané minimum bolo menšie ako maximum. (T: **bytes**, R: počet segmentov = 3 > r >= 0, D: 0, N: nie),
- **land\_segment\_id** – index segmentu, kam sa jednotka dostane, keď zastane. Do tohto segmentu je prvorado pridávaná po vyrobení. (T: **byte**, R: počet segmentov = 3 > r >= 0, D: 0, N: nie),
- **max\_hided\_units** – maximálny počet jednotiek miesta, ktoré je schopná v sebe jednotka prenášať. Jedna jednotka miesta odpovedá bojovej jednotke s veľkosťou 1 mapel štvorcový. (T: **byte**, R: >=0, D: 0, N: nie),
- **can\_hide** – zoznam textových identifikátorov typov jednotiek, ktoré môže daný typ v sebe prenášať. (T: **strings**, R: 1024 znakov, D: „“, N: nie),
- **tg\_picture\_id** – textový identifikátor skupiny textúr obrázka jednotky v menu definovaný v dátovom súbore príslušnej rasy. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **tg\_stay\_id** – textový identifikátor skupiny textúr stojacej jednotky definovaný v dátovom súbore príslušnej rasy. (T: **string**, R: 1024 znakov, D: -, N: áno),

- **tg\_anchor\_id** – textový identifikátor skupiny textúr kotviacej (pristátej) jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa použije textúra **tg\_stay\_id**. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_move\_id** – textový identifikátor skupiny textúr pohybujúcej sa jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa použije textúra **tg\_stay\_id**. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_land\_id** – textový identifikátor skupiny textúr pristávajúcej a odpristávajúcej jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa použije textúra **tg\_stay\_id**. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_rotate\_id** – textový identifikátor skupiny textúr točiacej sa jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa použije textúra **tg\_stay\_id**. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_attack\_id** – textový identifikátor skupiny textúr útočiacej jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa použije textúra **tg\_stay\_id**. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tq\_dying\_id** – textový identifikátor skupiny textúr umierajúcej jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa stav jednotky preskočí. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_zombie\_id** – textový identifikátor skupiny textúr mŕtvej (rozkladajúcej sa) jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa stav jednotky preskočí. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_projectile\_id** – textový identifikátor skupiny textúr náboja definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa nepoužije žiadna textúra a náboj bude neviditeľný. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **tg\_burning\_id** – textový identifikátor skupiny textúr plameňa definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu “none”, čo znamená, že sa nepoužije žiadna textúra. (*T: string, R: 1024 znakov, D: -, N: áno*),
- **snd\_ready** – textový identifikátor zvuku (z dátového súboru) jednotky, ktorá skončila akciu a je pripravená na ďalšiu. (*T: strings, R: 1024 znakov, D: -, N: áno*),
- **snd\_selected** - textový identifikátor zvuku (z dátového súboru) jednotky, ktorá bola práve označená. (*T: strings, R: 1024 znakov, D: -, N: áno*),
- **snd\_command** - textový identifikátor zvuku (z dátového súboru) jednotky, ktorá bola práve poslaná na nejakú akciu. (*T: strings, R: 1024 znakov, D: -, N: áno*),
- **snd\_fireon** - textový identifikátor zvuku (z dátového súboru) jednotky, ktorá nabíja. (*T: strings, R: 1024 znakov, D: -, N: áno*),
- **snd\_fireoff** - textový identifikátor zvuku (z dátového súboru) jednotky, ktorá práve vystrelila. (*T: strings, R: 1024 znakov, D: -, N: áno*),
- **snd\_hit** - textový identifikátor zvuku (z dátového súboru) dopadajúceho náboja. (*T: strings, R: 1024 znakov, D: -, N: áno*),

- **snd\_burning** - textový identifikátor zvuku (z dátového súboru) horiacej jednotky. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **snd\_dead** - textový identifikátor zvuku (z dátového súboru) umierajúcej jednotky. Je možné zadať hodnotu „none“, čo spôsobí, že sa použije zvuk definovaný položkou **snd\_dead** definovanou v hlavičke rasy. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **is\_offensive** – ak má táto položka hodnotu true, budú sa načítavať aj útočné vlastnosti jednotky, inak nie. (T: **bool**, R: -, D: false, N: nie),
- **offensive\_aggressivity** – položka určuje prednastavený stupeň agresivity jednotiek daného typu. Položka môže nadobúdať nasledujúce hodnoty: „aggressive”- jednotka zaútočí na nepriateľskú jednotku v okamihu, keď ju má v doohľade a bude ju prenasledovať, „offensive”- jednotka zaútočí na nepriateľskú jednotku v okamihu, keď na ňu dostrelí a bude ju prenasledovať, „guarded”- jednotka zaútočí na nepriateľskú jednotku v okamihu, keď na ňu dostrelí, no nebude ju prenasledovať (ostane stáť na mieste), „ignore”- jednotka úplne ignoruje cudzie jednotky. (T: **string**, R: „aggressive“, „offensive“, „guarded“, „ignore“, D: „ignore“, N: nie),
- **offensive\_accuracy** – presnosť zbrane. Udáva sa ako číslo z intervalu <0,1>, pričom, 0 znamená nepresná (na vzdialenosť 10 mapelov je maximálna odchýlka 3 mapely), 1 znamená presná (v absolútnom význame). (T: **float**, R: 1>= r >=0, D: 0, N: nie),
- **offensive\_range** – minimálna a maximálna vzdialenosť v mapeloch počítaná od okraja jednotky, na ktorú je jednotka schopná strieľať. Program kontroluje, či načítavané minimum je menšie ako maximum (ak nie je, modifikuje sa maximum). Očakávajú sa dve čísla. (T: **bytes**, R: >=1, D: 0, N: nie),
- **offensive\_scope** – polomer výbuchy strely pri dopade v mapeloch. (0 znamená 1x1 mapel, 1 znamená 3x3 mapely ...) (T: **byte**, R: >= 0, D: 0, N: nie),
- **offensive\_shotable\_seg\_min\_max** – index minimálneho a maximálneho segmentu, kam jednotka dostrelí. Kontroluje sa, či index minimálneho segmentu je menší ako index maximálneho segmentu (ak nie, index maximálneho je modifikovaný). Očakávaná je dvojica čísel. (T: **bytes**, R: počet segmentov = 3 > r >= 0, D: 0, N: nie),
- **gun\_power\_min\_max** – minimálna a maximálna sila útoku. Vždy je garantovaná minimálna sila útoku, ku ktorej sa náhodne pridá sila tak, aby nebola prekročená maximálna sila útoku. Očakávaná je dvojica čísel, pričom sa kontroluje, aby načítavané minimálne číslo bolo menšie ako maximálne. (T: **integers**, R: >= 0, D: 0, N: nie),
- **gun\_shot\_speed** – rýchlosť náboja v mapeloch za sekundu. (T: **float**, R: >= 0.01, D: 0.01, N: nie),
- **offensive\_shot\_time** – čas, ktorý jednotka potrebuje na jeden výstrel (od stisnutia spúšte po okamih, kedy náboj vyletí z hlavne). (T: **float**, R: >= 0, D: 0, N: nie),
- **offensive\_wait\_time** – čas, ktorý je jednotka „nepoužiteľná“ po vystrelení (čas potrebný na vychladnutie hlavne po výstrele, kým nie je možné znova nabíjať). (T: **float**, R: >= 0, D: 0, N: nie),
- **offensive\_feed\_time** – čas potrebný na nabitie zbrane. (T: **float**, R: >= 0, D: 0, N: nie),

- **offensive\_flags** – položka určuje špeciálne vlastnosti zbrane. Očakávaných je niekoľko príznakov, prípadne príznak “none”, ktorý hovorí, že zbraň nemá žiadne špeciálne vlastnosti. Očakávané hodnoty: “gun\_same\_segment”- jednotka môže atakovať len jednotky v segmenke, kde sa práve nachádza, “gun\_damage\_buildings”- útok jednotky ohrozuje aj budovy, “gun\_damage\_sources”- útok jednotky ohrozuje aj zdroje, “gun\_notlanding”- jednotka nedokáže stierať, keď kotví (po pristátí). (T: **strings**, R: 1024 znakov, D: „none“, N: nie),
- **defence\_armour** – položka vyjadruje silu obrany a je protikladom k sile útoku. (T: **integer**, R:  $\geq 1$ , D: 1, N: nie),
- **defence\_protection** – položka vyjadruje schopnosť jednotky uhýbať útoku. Očakávané je číslo medzi 0 (jednotka neuhýba) a 1 (jednotka uhýba). Ak má jednotka schopnosť uhýbania, znižuje sa útok na ňu o malé náhodné číslo. (T: **float**, R:  $1 \geq r \geq 0$ , D: 0, N: nie),
- **features** – položka určuje špeciálne vlastnosti typu jednotky. Očakávaných je niekoľko príznakov, prípadne príznak “none”, ktorý hovorí, že jednotka nemá žiadne špeciálne vlastnosti. Očakávané hodnoty: “have\_to\_land”- jednotka nemôže ostať stáť, ale musí pristáť (ak nemôže, tak sa jej v pravidelných intervaloch znižuje život), “heal\_when\_stay”- jednotka sa dokáže sama ozdravovať, keď stojí (čas ozdravenia nastavuje položka **heal\_time**), “heal\_when\_anchor”- jednotka sa sama ozdravuje, keď kotví (podobne ako v predchádzajúcom prípade je čas obnovenia nastavený položkou **heal\_time**). (T: **strings**, R: 1024 znakov, D: „none“, N: nie),
- **heal\_time** – čas potrebný na ozdravenie o jednu jednotku života v sekundách. Berie sa do úvahy len v prípade, že jednotka má niektorú z nasledovných špeciálnych vlastností: “heal\_when\_stay”, “heal\_when\_anchor”. (T: **float**, R:  $\geq 0$ , D: 0, N: nie),

**Špeciálne vlastnosti pracanta (WORKER UNIT):** načítajú sa len v prípade, že položka **item\_type** má hodnotu „w“.

- **max\_amount** – maximálne množstvo materiálov, ktoré je schopný pracant odnieť. Očakávaných toľko čísel, koľko materiálov bolo definovaných v príslušnej schéme. (T: **integers**, R:  $\geq 0$ , D: 0, N: nie),
- **allowed\_materials** – očakávaný je zoznam textových identifikátorov, ktorý určuje, ktoré materiály môže pracant ťažiť. V prípade zlého textového identifikátoru, program ohlásí varovanie, no načítanie to nepreruší. (T: **strings**, R: 1024 znakov, D: „“, N: nie),
- **mining\_time** – čas potrebný na vyťaženie jednej jednotky materiálu zo zdroja pre každý materiál. Očakávaných je toľko čísel, koľko bolo v príslušnej schéme nadefinovaných materiálov. Vykladanie daného materiálu je vždy 10 krát pomalšie. (T: **floats**, R:  $\geq 0$ , D: 0, N: nie),
- **mining\_sound\_shift** – položka pre každý typ materiálu definuje časový posun pustení prvého zvuku od začiatku ťaženia v sekundách. Očakávaných je toľko čísel, koľko bolo definovaných materiálov v príslušnej schéme. (T: **floats**, R:  $\geq 0$ , D: 0, N: nie),
- **mining\_sound\_time** – položka pre každý typ materiálu definuje čas medzi dvoma zvukmi toho istého pracanta v sekundách. Očakávaných je toľko čísel, koľko bolo definovaných materiálov v príslušnej schéme. (T: **floats**, R:  $\geq 0$ , D: 1, N: nie),



- **snd\_mine\_materialX** – textový identifikátor zvuku (z dátového súboru) ťažiackej jednotky pre materiál X. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **snd\_workcomplete** – textový identifikátor zvuku (z dátového súboru) pracanta, ktorý ukončil akciu. (T: **strings**, R: 1024 znakov, D: -, N: áno),
- **tg\_mine\_id** – textový identifikátor skupiny textúr ťažiackej jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu "none", čo znamená, že sa použije textúra stojacej jednotky. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **tg\_repair\_id** – textový identifikátor skupiny textúr opravujúcej a stavajúcej jednotky definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu "none", čo znamená, že sa použije textúra stojacej jednotky. (T: **string**, R: 1024 znakov, D: -, N: áno),
- **repairing\_time** – položka definuje čas potrebný na opravenie (postavenie) jednej jednotky života budovy, ktorú pracant opravuje (stavia) v sekundách. (T: **float**, R: >= 0, D: 0, N: nie),
- **can\_build** – očakávaný je zoznam textových identifikátorov budov, ktoré môže pracant stavať. Všetky zadané budovy môže automaticky aj opravovať. V prípade, že bude zadaný nesprávny identifikátor budovy, program zobrazí upozornenie. (T: **strings**, R: 1024 znakov, D: „“, N: nie),
- **can\_repair** - očakávaný je zoznam textových identifikátorov budov a jednotiek, ktoré môže pracant opravovať. Nie je nutné uvádzať do zoznamu budovy, ktoré môže pracan stavať, pretože ti môže automaticky aj opravovať. V prípade, že bude zadaný nesprávny identifikátor budovy (jednotky), program zobrazí upozornenie. (T: **strings**, R: 1024 znakov, D: „“, N: nie).

```

<Unit 1>
  id "peasant"
  name "Peasant"
  size 1
  view 6
  energy 0
  food -10
  materials 400 5 0
  max_life 70
  max_speed 10 10 10
  max_rotation_speed 1440 1440 1440
  selection_height 35
  burning_position 0 0
  item_type w
  max_amount 100 20 0
  allowed_materials "gold" "wood"
  can_build "farm" "tower" "cannontower" "barracks" "shed"
  can_repair "catapult" "airship"
  mining_time 0.1 0.33 0
  repairing_time 0.2
  mining_sound_shift 0 0 0
  mining_sound_time 0 1.0 0
  move_terrain_id 0 0 8 25 0 0
  land_terrain_id 0 0 0 0 0 0

```

```

min_exist_segment_id 1
max_exist_segment_id 1
min_max_visible_segment_id 1 2 1 2 1 2
land_segment_id 1
max_hided_units 0
can_hide
tg_picture_id peasant_picture
tg_stay_id peasant_stay
tg_anchor_id none
tg_move_id peasant_stay
tg_land_id none
tg_rotate_id none
tg_attack_id peasant_stay
tg_mine_id peasant_stay
tg_repair_id peasant_stay
tg_dying_id none
tg_zombie_id none
tg_projectile_id none
tg_burning_id none
snd_ready peasant_ready
snd_selected peasant_selected1 peasant_selected2
snd_command peasant_command1 peasant_command2
snd_workcomplete peasant_workcomplete
snd_mine_material0 none
snd_mine_material1 peasant_chop1 peasant_chop2 peasant_chop3
snd_mine_material2 peasant_chop1 peasant_chop2
snd_mine_material3 none
snd_fireon none
snd_fireoff peasant_hit
snd_hit none
snd_burning none
snd_dead none
is_offensive true
offensive_aggressivity guarded
offensive_accuracy 0.8
offensive_flags none
offensive_range 1 1
offensive_shot_time 0
offensive_wait_time 0
offensive_feed_time 0.8
offensive_scope 0
offensive_shotable_seg_min_max 1 1
gun_power_min_max 5 20
gun_shot_speed 100
defence_armour 2
defence_protection 0.3
features heal_when_stay
heal_time 5.0
</Unit 1>

```

### 3.1.3.4 Sekcia <Buildings>

Sekcia obsahuje definície vlastností statických jednotiek rasy – budov (BUILDING UNITS) a tovární (FACTORY UNITS). Statické jednotky sa delia na spomínané dva typy podľa

činností, ktoré môžu vykonávať (továrne môžu navyše vyrábať (cvičiť) pohyblivé jednotky rasy). Obsahuje nasledujúce položky a podsekcie:

- **count** - počet typov statických jednotiek rasy. Program sa pokúsi prečítať zadaný počet sekcií <Building>. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- <Building X> - definícia jedného konkrétneho typu statickej jednotky.

```
<Buildings>
  count 5
  <Building 0>
  ...
  </Building 0>
</Buildings>
```

### 3.1.3.5 Sekcia <Building>

V sekcii je možné nájsť podrobné definície vlastností budov. Mnohé položky budú totožné s položkami sekcie <Unit> (aj textovo a aj významovo) a preto budú len vymenované. Ostatné položky budú podrobne popísané v nasledujúcom zozname:

- **id, name, materials, max\_life, view, selection\_height, burning\_position, energy, food, max\_hided\_units, can\_hide, tg\_picture\_id, tg\_stay\_id, tg\_dying\_id, tg\_zombie\_id, tg\_projectile\_id, is\_offensive, defence\_armour, defence\_protection, offensive\_aggressivity, offensive\_accuracy, offensive\_flags, offensive\_range, offensive\_shot\_time, offensive\_wait\_time, offensive\_feed\_time, offensive\_scope, offensive\_shootable\_seg\_min\_max, gun\_power\_min\_max, gun\_shot\_speed** – všetky tieto položky majú rovnaký význam ako rovnomené položky sekcie <Unit>,
- **item\_type** – položka označuje typ jednotky. Očakáva sa jedna z hodnôt: „b”- budova (BUILDING UNIT), „a”- továreň (FACTORY UNIT). V prípade, že hodnota je „a” sa navyše budú načítavať položky špecializované pre továrne. (*T*: **string**, *R*: 1 znak, *D*: „b”, *N*: áno),
- **width** – šírka podstavy jednotky v mapeloch. Na rozdiel od pohyblivých jednotiek nemusia mať budovy a továrne štvorcovú podstavu. (*T*: **byte**, *R*:  $15 \geq r > 0$ , *D*: 1, *N*: nie),
- **height** – dĺžka podstavy budovy v mapeloch. (*T*: **byte**, *R*:  $15 \geq r > 0$ , *D*: 1, *N*: nie),
- **min\_energy** – položka vyjadruje, koľko percent z energie uvedenej v položke **energy** musí budova dostávať, aby bez obmedzení plnila svoju funkciu. Aktuálna energia pripadajúca na budovu sa získa pomerným rozpočítaním energie hráča na všetky jednotky. Pri poklese aktuálnej energie pod hodnotu v položke **min\_energy** prestane továreň vyrábať nové jednotky (výroba sa spustí automaticky ak sa dosiahne aspoň minimálna energia). Očakávaná je percentuálna hodnota. (*T*: **integer**, *R*:  $100 \geq r \geq 0$ , *D*: 100, *N*: nie),
- **build\_terrain\_id** - minimálna a maximálna výška terénu, kde je možné budovu postaviť v danom segmente určenom položkou **exists\_segment\_id**. Očakávaná je dvojica čísel. Program kontroluje, či načítavané minimum je menšie ako maximum (ak nie je, za maximum sa vezme minimum). (*T*: **integers**, *R*:  $\geq 0$ , *D*: 0, *N*: nie),
- **exists\_segment\_id** – položka určuje index segmentu, v ktorom je budova postavená. (*T*: **byte**, *R*: počet segmentov =  $3 > r \geq 0$ , *D*: 0, *N*: nie),

- **min\_max\_visible\_segment\_id** – podobný význam rovnomennej položky sekcie **<Unit>**, ale len pre jediný segment určený položkou **exists\_segment\_id**. Očakávaná je teda dvojica indexov určujúcich minimálny a maximálny viditeľný segment, pričom sa kontroluje, aby načítavané minimum bolo menšie ako maximum. (*T: byte*, *R: počet segmentov = 3 > r >= 0*, *D: 0*, *N: nie*),
- **ancestor** – položka určuje, či daný typ jednotky je vylepšením (rozšírením) iného typu jednotky (vtedy je hodnotou položky jednoznačný textový identifikátor typu), alebo úplne nový typ (hodnota položky je „none“). Ak má typ predchodcu, musí mať rovnakú veľkosť a existovať v rovnakom segmente, pretože sa ho inak nepodarí postaviť. Následník sa stavia na rovnaké pozície, ako jeho predchodca. Očakávaný je teda textový identifikátor iného typu jednotky, prípadne text „none“. (*T: string*, *R: 1024*, *D: „none“*, *N: áno*),
- **tg\_build\_id** – textový identifikátor skupiny textúr stavajúcej sa jednotky definovaný v dátovom súbore príslušnej rasy, ktoré sa zobrazujú postupne podľa vyspelosti stavby. Je možné zadať hodnotu „none“, čo znamená, že sa použije textúra stojacej jednotky. (*T: string*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **tg\_burning\_id** – textový identifikátor skupiny textúr plameňa definovaný v dátovom súbore príslušnej rasy. Je možné zadať hodnotu „none“, čo znamená, že sa použije textúra **tg\_burning\_id** definovaná v hlavičke rasy. (*T: string*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **snd\_selected** - textový identifikátor zvuku (z dátového súboru) jednotky, ktorá bola práve označená. Je možné zadať hodnotu „none“, čo znamená, že sa má použiť hodnota položky **snd\_building\_selected** definovanej v hlavičke rasy. (*T: strings*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **snd\_burning** - textový identifikátor zvuku (z dátového súboru) horiacej budovy. Ak bude zadaná hodnota „none“, použije sa hodnota položky **snd\_burning** definovanej v hlavičke rasy. (*T: strings*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **snd\_explosion** - textový identifikátor zvuku (z dátového súboru) explózie (pádu) budovy. V prípade hodnoty „none“ sa použije hodnota položky **snd\_explosion** z hlavičky rasy. (*T: strings*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **allowed\_materials** – očakávaný je zoznam textových identifikátorov materiálov, ktoré je budova schopná prijímať. Ak bude zadaný neexistujúci identifikátor, program vypíše varovanie. (*T: strings*, *R: 1024 znakov*, *D: -*, *N: nie*),
- **<Products>** - špeciálna vlastnosť tovární. Sekcia definuje produkty, ktoré môže továreň vyrábať.

```

<Building 0>
  id "townhall"
  name "Town Hall"
  item_type a
  width 7
  height 7
  materials 1200 800 0
  max_life 1000
  view 7
  selection_height 60
  burning_position 10 35

```

```

energy 0
min_energy 0
food 0
build_terrain_id 10 10
exist_segment_id 1
min_max_visible_segment_id 1 2
ancestor none
max_hided_units 5
can_hide "peasant"
tg_picture_id townhall_picture
tg_stay_id townhall_stay
tg_build_id townhall_build
tg_attack_id none
tg_dying_id none
tg_zombie_id townhall_zombie
tg_projectile_id none
tg_burning_id none
snd_selected townhall_selected
snd_burning none
snd_explosion none
allowed_materials "gold" "wood"
is_offensive false
defence_armour 20
defence_protection 0
<Products>
...
</Products>
</Building 0>

```

### 3.1.3.6 Sekcia <Products>

Sekcia je načítaná len v prípade definície továrne – položka **item\_type** má hodnotu „a“. Definuje pohyblivé typy jednotiek, ktoré je továreň schopná vyrábať (iná predstava: cvičiť vojakov v táboroch).

- **count** - počet produktov továrne. Program sa pokúsi prečítať zadaný počet sekcií <Product>. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- **<Product>** - sekcia obsahujúca informácie o každom produkte zvlášť.
  - **product** – očakávaný je textový identifikátor pohyblivej jednotky danej rasy. (*T*: **string**, *R*: 1024 znakov, *D*: -, *N*: áno),
  - **product\_time** – čas potrebný na vytvorenie jednej jednotky v sekundách. (*T*: **float**, *R*:  $\geq 0$ , *D*: 0, *N*: áno).

```

<Products>
count 1
<Product 0>
product "paesant"
product_time 20
</Product>
</Products>

```

### 3.1.3.7 Sekcia <Sources>

Sekcia je špeciálna pre schémovú rasu, hráčske rasy ju neobsahujú (ak by ju mali, tak bude ignorovaná). Definujú sa v nej podrobné vlastnosti zdrojov materiálov, z ktorých v hre hráči získavajú suroviny na svoj rozvoj. Obsahuje nasledujúce položky a podsekcie:

- **count** - počet typov zdrojov. Program sa pokúsi prečítať zadaný počet sekcií <Source>. Ak sa to nepodari, načítavanie skončí chybou. (*T: integer*, *R: >= 0*, *D: 0*, *N: áno*),
- **<Source X>** - definícia jedného konkrétneho typu zdroja.

```
<Sources>
  count 4
  <Source 0>
  ...
  </Source 0>
</Sources>
```

### 3.1.3.8 Sekcia <Source>

Obsahuje podrobné definície vlastností typu zdroja. Mnohé položky majú rovnaký význam ako rovnomenné položky sekcie <Building>, a preto budú len vymenované.

- **id, name, width, height, max\_life, selection\_height, burning\_position, max\_hided\_units, can\_hide, tg\_picture\_id, tg\_stay\_id, tg\_zombie\_id, tg\_burning\_id, tg\_dying\_id, snd\_selected, snd\_explosion, snd\_burning, build\_terrain\_id, exists\_segment\_id, defence\_armour, defence\_protection** - tieto položky majú rovnaký význam ako rovnomenné položky sekcie <Building>,
- **capacity** – položka určuje maximálnu kapacitu zdroja. (*T: integer*, *R: >= 0*, *D: 0*, *N: nie*),
- **offer\_material** - Očakávaný je textový identifikátor materiálu (definovaný v príslušnej schéme), ktorý je možné v zdroji ťažiť. (*T: string*, *R: 1024 znakov*, *D: -*, *N: áno*),
- **renewable** - položka rozhoduje o tom, či sa daný zdroj bude sám obnovovať – napríklad les sám rastie. (*T: bool*, *R: true/false*, *D: false*, *N: nie*),
- **time\_of\_first\_regeneration** – položka sa berie do úvahy len v prípade, že je zdroj obnoviteľný. Určuje čas v sekundách potrebný na obnovenie prvej jednotky materiálu v zdroji po jeho úplnom vyťažení – na príklade lesa je to čas, kým vyklíči semienko zo zeme. (*T: float*, *R: >= 0*, *D: 1*, *N: nie*),
- **time\_of\_reeneration** - berie sa do úvahy len v prípade obnoviteľných zdrojov. Určuje čas v sekundách potrebný na obnovenie každej ďalšej jednotky materiálu v zdroji. (*T: float*, *R: >= 0*, *D: 1*, *N: nie*),
- **inside\_mining** – položka určuje, či má pracant pri ťažení z tohto zdroja zmiznúť z mapy (materiál ťaží zvnútra zdroja - baňa), alebo má ostať v mape (ťaženie lesa). (*T: bool*, *R: true/false*, *D: false*, *N: nie*),
- **hideable** - pomocou položky sa určuje, čo sa má stať v prípade, že zdroj je vyťažенý. V prípade hodnoty „true“ je zdroj odstránený z mapy (textúra vyťažенého zdroja však ostáva viditeľná) a dá sa po danom mieste chodiť. V prípade hodnoty „false“ je miesto nedostupné. (*T: bool*, *R: true/false*, *D: false*, *N: nie*).

```
<Source 1>
  id "forest"
  name "Forest"
  width 3
  height 3
  max_life 400
  capacity 500
  selection_height 50
  burning_position 0 40
  max_hidden_units 0
  can_hide
  tg_picture_id forest_picture
  tg_stay_id forest_stay
  tg_zombie_id none
  tg_burning_id none
  tg_dying_id none
  snd_selected none
  snd_explosion none
  snd_burning none
  build_terrain_id 10 10 10 10 10 10
  exist_segment_id 1
  defence_armour 1
  defence_protection 20
  offer_material "wood"
  time_of_regeneration 3
  time_of_first_regeneration 600
  renewable true
  inside_mining false
  hideable true
</Source 1>
```

### 3.1.4 Popis súborov máp

Súbory máp sú umiestnené v podadresári „maps“ koreňového adresára hry a majú príponu „.map“. Ako už bolo uvedené v úvode kapitoly o konfiguračných súboroch, mapy sú závislé na schéme (pretože z terénnych elementov schémy je mapa pokladaná) a tak isto sú závislé na rasách, ktorých jednotky sa v mape pohybujú. Na závislosť máp, schém a rás sa však dá pozerat' aj z druhej strany – od používateľa programu, ktorý hru začína tým, že si zvolí mapu (nie schému, ako to robí tvorca hry). V okamihu výberu mapy program skontroluje, či existuje schéma uvedená v mape a či sú rasy uvedené v mape správne definované (či existujú a či sú tvorené na schému uvedenú v mape). Ak niečo nie je v poriadku, hra sa nemôže začať.

Súbory máp sú členené na dva rozsiahlejšie celky – definícia samotnej mapy pomocou elementov schémy a definícia hráčov = rás (hráči na mape musia mať rôzne rasy, pretože by nebolo možné graficky rozlíšiť dvoch hráčov) a ich štartovných pozícií.

#### 3.1.4.1 Hlavička

Hlavička obsahuje základné informácie o mape a pomocné informácie o súbore mapy.

- **name** – názov mapy. Nepoužíva sa ako identifikátor (tým je meno súboru bez prípony), no zobrazuje sa pri výbere mapy v hlavnom menu hry. (*T*: **string**, *R*: 1024 znakov, *D*: „“, *N*: nie),



- **author** – meno autora súboru. V programe sa nevyužíva, je to len informácia navyše. (*T*: **string**, *R*: 1024 znakov, *D*: „“, *N*: nie),
- **width** – šírka mapy v mapeloch – najmenších adresovateľných jednotiek mapy. (*T*: **byte**, *R*: maximálna veľkosť mapy = 240  $\geq r \geq 1$ , *D*: 1, *N*: áno),
- **height** – dĺžka (výška) mapy v mapeloch. (*T*: **byte**, *R*: maximálna veľkosť mapy = 240  $\geq r \geq 1$ , *D*: 1, *N*: áno),
- **scheme** – očakávaný je textový identifikátor schémy, pre ktorú je mapa vytvorená. Pri výbere mapy z menu sa kontroluje, či zadaná schéma existuje a či je správne definovaná. (*T*: **string**, *R*: 1024 znakov, *D*: „“, *N*: áno).

```
name "Trial map"
author "PP team"
width 80
height 70
scheme "plastic"
```

### 3.1.4.2 Sekcia <Players>

Prvá z dvoch veľkých celkov definície mapy – definuje sa tu počet hráčov, ich štartovacie pozície a rasy, ktoré si môžu vybrať. Každá mapa má maximálny počet hráčov, no aby mohli skutočne všetci hrať, musí byť definovaných dostatočný počet štartovacích pozícií a tiež dostatočný počet rás. Počet hráčov, ktorí môžu hrať na mape, je minimum z čísel: počet rás, počet štartovacích pozícií, maximálny počet hráčov. Sekcia má nasledujúce položky a podsekcie:

- **max\_count** – maximálny prípustný počet hráčov na mape. (*T*: **integer**, *R*: maximálny počet hráčov = 8  $\geq r \geq 0$ , *D*: 0, *N*: áno),
- **<Start Points>** - definuje všetky možné štartovacie pozície hráčov v mape,
- **<Races>** - zoznam všetkých rás, ktoré môžu byť na mape použité.

```
<Players>
  max_count 2
  <Start Points>
  ...
</Start Points>
  <Races>
  ...
</Races>
</Players>
```

### 3.1.4.3 Sekcia <StartPoints>

Sekcia definuje všetky možné štartovacie pozície na mape. Program ich po spustení hry náhodne prideli hráčom tak, aby žiadni dvaja nezačínali hru na rovnakej pozícii. Zabezpečuje sa tým istá neopakovateľnosť každej hry.

- **count** – počet štartovacích pozícií, ktoré sa program pokúsi prečítať. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- **start\_point\_X** – definuje súradnice štartovacej pozície v mape. Program kontroluje, či bola zadaná pozícia v mape. Parametre štartovacej pozície sú kvôli obmedzeniu veľkosti súboru uvedené v jednom riadku. Očakávané sú 2 čísla:



- **x** – x-ová súradnica pozície. (*T*: **integer**, *R*: **width**  $\geq r \geq 0$ , *D*: 0, *N*: áno),
- **y** – y-ová súradnica pozície. (*T*: **integer**, *R*: **height**  $\geq r \geq 0$ , *D*: 0, *N*: áno).

```
<Start Points>
  count 1
  start_point_0 65 48
</Start Points>
```

#### 3.1.4.4 Sekcia <Races>

V sekcii sú definované všetky prípustné rasy. Z už uvedených dôvodov musia mať hráči na mape rôzne rasy. Okrem „reálnych“ rás je tu definovaná aj takzvaná „schémová“ rasa, pomocou ktorej sa do mapy dostanú hlavne zdroje – sú teda vlastnené „schémovým“ hráčom. Ostatné rasy môžu vlastniť len budovy a pohyblivé jednotky.

- **count** – počet rás (sekcii <Race>), ktoré sa program pokúsi prečítať. Ak sa to nepodarí, načítavanie skončí chybou. Ide iba o rasy, ktoré si vyberú reálni hráči. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- <Race> - sekcia definuje všetky jednotky a budovy, ktoré bude mať hráč danej rasy na začiatku hry,
- <SchemeRace> - sekcia definuje jednotky, budovy a zdroje, ktoré sú vlastnené schémovým hráčom. Táto sekcia sa vyskytuje iba raz.

```
<Races>
  count 2
  <Race 0>
    ...
  </Race 0>
  <SchemeRace>
    ...
  </SchemeRace>
</Races>
```

#### 3.1.4.5 Sekcia <Race> a <SchemeRace>

Aby sa predišlo jednotvárnym začiatkom hry, program umožňuje definovať rôzne počiatkové stavy hráča tým, že sa definuje rôzny počet iných, prípadne inak rozložených jednotiek. Program po spustení hry náhodne vyberie jeden z definovaných počiatkových stavov hráča a jednotky umiestni do mapy relatívne k náhodne vybranej štartovacej pozícii (viď sekcia <Start Points>). Všetky štartovacie pozície jednotiek a budov hráča sú teda zadávané relatívne k „fiktívnej štartovacej pozícii“, ktorá sa namapuje na skutočnú štartovaciu pozíciu mapy. Ďalším rozdielom medzi reálnym a schémovým hráčom je v tom, že schémový hráč má jediný štartovací bod, ktorý je pevne určený v začiatku mapy (pozícia [0,0]). Pozície jednotiek, budov a zdrojov schémového hráča sú teda zadávané relatívne k pozícii [0,0] – teda absolútne.

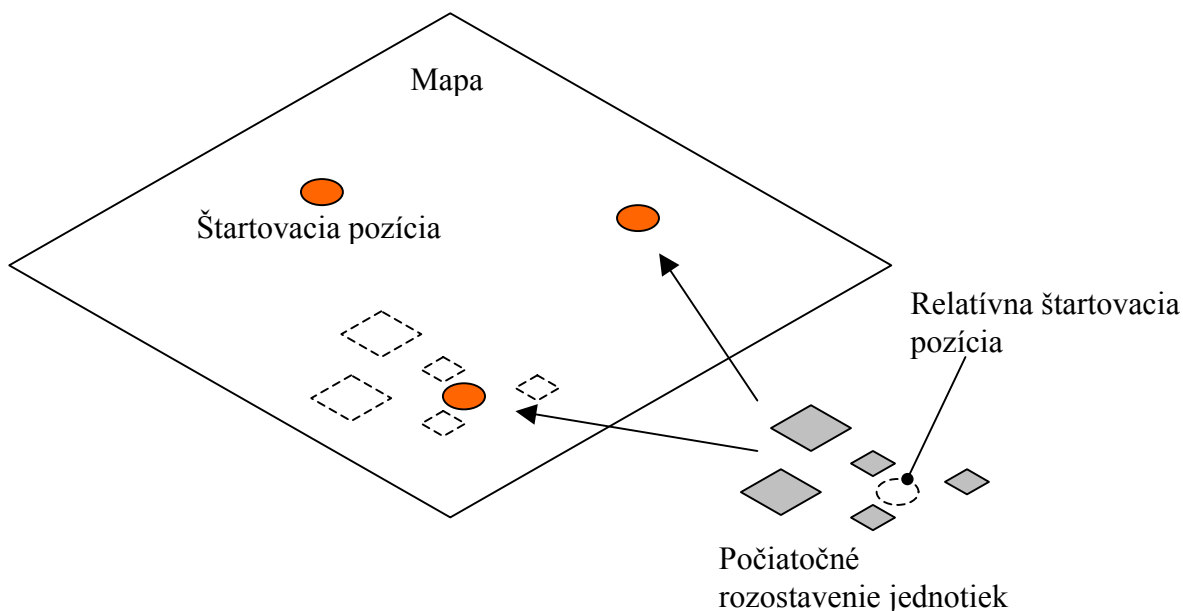
- **name** – textový identifikátor rasy. Súbor tejto rasy musí existovať a musí byť korektný, inak program skončí chybou. (*T*: **string**, *R*: 1024 znakov, *D*: „“, *N*: áno),
- <Sets> - podsekcia definujúca všetky počiatkové stavy hráča, ktorý si zvolil danú rasu. V definícii schémovej rasy sekcia nie je, sú tam priamo sekcie <Units>, <Buildings>, <Sources> (viď nižšie).

```

<Race>
  name "human-yellow"

  <Sets>
  ...
</Sets>
</Race 1>

```



### 3.1.4.6 Sekcie <Sets> a <Set>

Obe sekcie plnia viac-menej obalovú funkciu pre definície počiatočných stavov hráča.

- **count** – položka má štandardný význam – určuje počet sekcií <Set>, ktoré sa program pokúsi prečítať. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- **<Set>** - definícia jedného konkrétneho štartovacieho stavu hráča.
  - **init\_materials\_amount** – počiatočné množstvá materiálov hráča pre daný štartovací stav. Očakávaných je toľko čísel, koľko bolo definovaných materiálov v príslušnej schéme. (*T*: **integes**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
  - **<Units>**, **<Buildings>**, **<Sources>** - definície jednotiek. Sekcia <Sources> je očakávaná iba pri definícii schémovej rasy, inak je ignorovaná.

```

<Sets>
  count 1
  <Set 0>
    init_materials_amount 1500 1000 1000
    <Units>
    ...
  </Units>
  <Buildings>
  ...
</Buildings>

```

```

<Sources>
...
</Sources>
</Set 0>
</Sets>

```

### 3.1.4.7 Sekcie <Units>, <Buildings> a <Sources>

Sekcia obsahuje definície jednotiek, budov a zdrojov jedného štartovacieho stavu hráča.

- **count** – určuje počet položiek príslušnej sekcie, ktoré sa program pokúsi prečítať. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- **unit\_X** – definuje typ jednotky z príslušnej rasy, pozíciu v mape, natočenie a počiatočný život. Očakávaných je 6 čísel:
  - **unit\_id** – identifikátor typu jednotky (číslo *Z* z definície sekcie <Unit *Z*> v schéme),
  - **x, y, z** – pozícia ľavého dolného rohu jednotky v mape. (*T*: **integer**, *R*: maximálna veľkosť mapy  $\geq r \geq 0$ , *D*: 0, *N*: áno),
  - **direction** – natočenie jednotky (očakáva sa číslo z intervalu <0,7> reprezentujúce smer). (*T*: **integer**, *R*:  $7 \geq r \geq 0$ , *D*: 0, *N*: áno),
  - **life** – percentuálne vyjadrenie aktuálneho života z maximálneho života daného typu jednotky. (*T*: **integer**, *R*:  $100 \geq r \geq 0$ , *D*: 0, *N*: áno).
- **building\_X** – definuje typ budovy z príslušnej rasy, pozíciu v mape a počiatočný život. Očakávané sú 4 čísla:
  - **building\_id** – identifikátor typu budovy (číslo *Z* z definície sekcie <Building *Z*> v schéme),
  - **x, y** – pozícia ľavého dolného rohu budovy v mape. (*T*: **integer**, *R*: maximálna veľkosť mapy  $\geq r \geq 0$ , *D*: 0, *N*: áno),
  - **life** – percentuálne vyjadrenie aktuálneho života z maximálneho života daného typu budovy. (*T*: **integer**, *R*:  $100 \geq r \geq 0$ , *D*: 0, *N*: áno).
- **source\_X** - definuje typ zdroja z príslušnej rasy, pozíciu v mape, počiatočný život a počiatočný stav materiálu v zdroji (aktuálnu kapacitu). Očakávaných je 5 čísel:
  - **source\_id** – identifikátor typu zdroja (číslo *Z* z definície sekcie <Source *Z*> v schéme),
  - **x, y** – pozícia ľavého dolného rohu zdroja v mape. (*T*: **integer**, *R*: maximálna veľkosť mapy  $\geq r \geq 0$ , *D*: 0, *N*: áno),
  - **life** – percentuálne vyjadrenie aktuálneho života z maximálneho života daného typu zdroja. (*T*: **integer**, *R*:  $100 \geq r \geq 0$ , *D*: 0, *N*: áno),
  - **material\_balance** – absolútna vyjadrenie aktuálneho stavu materiálu v zdroji. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno).

```

<Units>
count 1

```

```

    unit_0 "footman" 9 0 1 0 70
</Units>
<Buildings>
    count 2
    building_0 "fort" 0 0 80
    building_1 "farm" -3 0 100
</Buildings>
<Sources>
    count 2
    source_0 "goldmine" 33 5 75 50000
    source_1 "goldmine" 70 25 50 50000
</Sources>

```

### 3.1.4.8 Sekcia <Segment>

Segmenty predstavujú druhý veľký celok definície mapy. V súboroch máp sú očakávané tri sekcie: <Segment 0>, <Segment 1> a <Segment 2>, ktoré vyjadrujú jednotlivé vrstvy mapy. Dosahuje sa tým „trojrozmernosť“ priestoru, v ktorom sa môžu pohybovať jednotky hráčov. Jednotlivé segmenty vyjadrujú postupne podpovrchovú vrstvu (podzemie), povrch (zem) a nadpovrchovú vrstvu (vzdušný priestor). Každá sekcia pomocou fragmentov definovaných v schéme popisuje „povrch“ vrstiev a tým dostupnosť políčka – mapelu (najmenšej adresovateľnej jednotky mapy) jednotlivým jednotkám hráča. Navyše je do mapy možné umiestniť objekty a vrstvy definované v schéme.

- <Fragments> - podsekcia obsahujúca zoznam fragmentov mapujúcich sa na povrch mapy,
- <Layers> - podsekcia obsahujúca zoznam vrstiev vložených do mapy,
- <Objects> - podsekcia obsahuje inštancie objektov pridaných do mapy.

```

<Segment 1>
  <Fragments>
    ...
  </Fragments>
  <Layers>
    ...
  </Layers>
  <Objects>
    ...
  </Objects>
</Segment 1>

```

### 3.1.4.9 Sekcia <Fragments>

Sekcia <Fragments> je najdôležitejšou sekciou celého konfiguračného súboru mapy. Obsahuje mapovanie fragmentov definovaných v schéme na povrch mapy a tým určuje, ktoré políčko mapy – mapelu bude pre jednotku dostupné a ako rýchlo sa po ňom bude jednotka pohybovať. Má nasledujúce položky:

- **count** – počet fragmentov. Program sa pokúsi prečítať zadaný počet položiek **fragment\_X**. Ak sa to nepodarí, načítavanie skončí chybou. (*T*: **integer**, *R*:  $\geq 0$ , *D*: 0, *N*: áno),
- **fragment\_X** – definuje typ fragmentu definovaného v schéme a jeho súradnice v mape. Program kontroluje, či je celý fragment umiestnený v mape. Ak nie je, do

mapy sa nepridá a prázdne miesto sa vyplní príslušným počtom prednastavených fragmentov veľkosti 1x1 mapel zo schémy. Dva fragmenty sa nesmú prekryvať. Ak bude zadaný chybný fragment, nebude do mapy pridaný. Parametre fragmentu sú kvôli obmedzeniu veľkosti súboru uvedené v jednom riadku (dobré viditeľné na príklade). Očakávané sú tri čísla:

- **typ** – identifikátor typu fragmentu (číslo X sekcie <Fragment X> zo schémy). (*T: integer, R: >= 0, D: 0, N: áno*),
- **x** – x-ová súradnica ľavého dolného rohu fragmentu v mape. (*T: integer, R: >= 0, D: 0, N: áno*),
- **y** – y-ová súradnica ľavého dolného rohu fragmentu v mape. (*T: integer, R: >= 0, D: 0, N: áno*).

```
<Fragments>
  count 200
  fragment_0 0 70 65
  fragment_1 0 75 65
</Fragments>
```

#### 3.1.4.10 Sekcia <Layers>

Sekcia určuje, ktoré typy vrstiev zo schémy sa použijú v mape a definuje ich umiestnenie. Vrstvy lokálne menia výšku terénu.

- **count** – počet načítavaných vrstiev. Program sa pokúsi prečítať zadaný počet položiek **layer\_X**. Ak sa to nepodarí, načítavanie skončí chybou. (*T: integer, R: >= 0, D: 0, N: áno*),
- **layer\_X** – položka definuje typ vrstvy definovanej v schéme a jej súradnice v mape. Program kontroluje, či celá vrstva leží v mape – ak nie, nebude do mapy pridaná. Vrstvy sa môžu prekryvať a výška terénu daného mapelu je určená naposledy pridanou vrstvou. Podobne ako fragmenty majú vrstvy svoje parametre v riadku. Očakávané sú tri čísla:
  - **typ** – identifikátor typu vrstvy (číslo X sekcie <Layer X> zo schémy). (*T: integer, R: >= 0, D: 0, N: áno*),
  - **x** – x-ová súradnica ľavého dolného rohu vrstvy v mape. (*T: integer, R: >= 0, D: 0, N: áno*),
  - **y** – y-ová súradnica ľavého dolného rohu vrstvy v mape. (*T: integer, R: >= 0, D: 0, N: áno*).

```
<Layers>
  count 1
  layer_0 0 25 0
</Layers>
```

#### 3.1.4.11 Sekcia <Objects>

Sekcia obsahuje zoznam objektov, ktoré majú byť pridané do mapy. Objekt má podobné vlastnosti ako vrstva. Líšia sa spôsobom vykresľovania a tým, že sa nesmú prekryvať (podrobne je to vysvetlené v definícii schémy). Objekty sa z mapy načítavajú rovnako ako vrstvy, len sa kontroluje, aby sa neprekryvali. Očakávané sú tri čísla: **typ, x, y**

```
<Objects>
  count 4
  object_0 0 0 70 45
</Objects>
```

## 3.2 Dátové súbory

Dátové súbory združujú audio a video záznamy používané v hre Dark Oberon. Jednotlivé konfiguračné súbory hry sa môžu na tieto záznamy odkazovať prostredníctvom textových identifikátorov.

Dátový súbor obsahuje dva typy záznamov, na ktoré je možné sa odkazovať: skupiny textúr a zvuky. Na skupiny textúr sú kladené presné pravidlá podľa toho, na aký účel bude skupina použitá. Nasledujúce kapitoly popisujú zoznam týchto pravidiel podľa typu odkazu. Spoločným pravidlom pre všetky skupiny je existencia aspoň jednej textúry v skupine. Textúry môžu byť animované.

Detailný popis dátových súborov sa nachádza v dokumentácii k Data Editoru.

### 3.2.1 Rasy

Dátový súbor pre rasy môže obsahovať nasledujúce typy skupín textúr (delenie podľa názvu položky v konfiguračnom súbore rasy):

- **tg\_food\_id** – skupina obsahuje dve textúry. Prvou je ikona pre jedlo s rozmermi snímku 15x12. V prípade odlišných rozmerov bude textúra deformovaná. Druhá textúra sa používa ako znamenie v prípade nedostatku jedla (napr. pri stavaní jednotiek v továrňach),
- **tg\_energy\_id** – rovnako ako pri tg\_food\_id,
- **tg\_material0\_id, tg\_material1\_id, tg\_material2\_id, tg\_material3\_id** – rovnako ako pri tg\_food\_id,
- **tg\_burning\_id** – jednotlivé textúry predstavujú stupeň požiaru od najmenšieho po najväčší,
- **Jednotky**
  - **tg\_picture\_id** – skupina obsahuje jednu textúru s obrázkom jednotky. Predpokladajú sa rozmery 50x40 bodov. Obrázok s inými rozmermi bude deformovaný,
  - **tg\_stay\_id, tg\_anchor\_id, tg\_move\_id, tg\_rotating\_id, tg\_attack\_id** – obsahujú práve jednu alebo osem textúr pre každý smer jednotky,
  - **tg\_projectile\_id** – obsahuje práve jednu alebo osem textúr pre každý smer náboja,
  - **tg\_dying\_id** – podobne ako tg\_stay\_id. Navyše platí pravidlo, že pre dĺžku stavu v ktorom sa jednotka nachádza je použitá dĺžka animácie,
  - **tg\_zombie\_id** – podobne ako tg\_stay\_id. Navyše platí, že dĺžka animácie textúry je upravená podľa dĺžky stavu,
  - **tg\_burning\_id** – jednotlivé textúry predstavujú stupeň požiaru od najmenšieho po najväčší,

- **Budovy**
  - **tg\_picture\_id, tg\_projectile\_id, tg\_burning** – rovnako ako u jednotiek,
  - **tg\_stay\_id** – jednotlivé textúry predstavujú stupeň zničenia budovy od plného života až po nulový život,
  - **tg\_build\_id** – jednotlivé textúry predstavujú stupeň stavania budovy od základov až po stupeň tesne pred dokončením budovy. Pre posledný stupeň stavania sa využíva prvá textúra z tg\_stay\_id,
  - **tg\_dying\_id, tg\_zombie\_id** – práve jedna textúra. Pre dĺžky animácií platia pravidlá ako u jednotiek.

### 3.2.2 Schémy

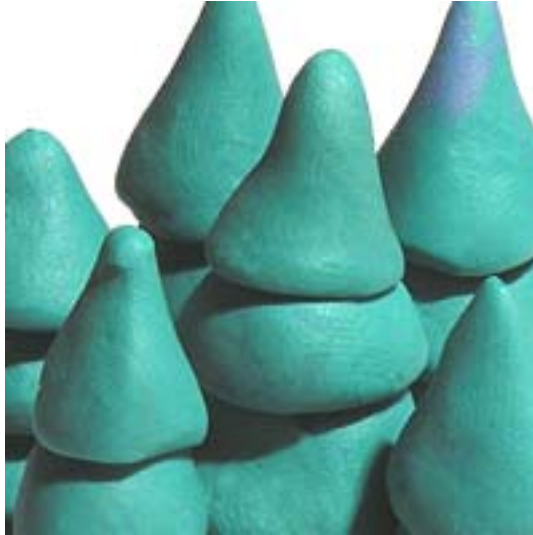
Dátový súbor pre schému obsahuje jednak záznamy spojené so schémou samotnou a potom záznamy schémovej rasy. Schémová rasa používa rovnaké typy záznamov ako bežná rasa, navyše môže obsahovať skupiny textúr pre zdroje materiálov.

Typy skupín textúr podľa odkazu:

- **Fragmenty** – ak obsahuje skupina viac textúr, vyberie sa pre zobrazenie fragmentu náhodná z nich. Všetky textúry musia mať pre správne vykreslenie nastavený typ „Terrain Fragment“,
- **Objekty** – podobne ako u fragmentov sa pre zobrazenie vyberie náhodná textúra zo skupiny,
- **Zdroje:**
  - **tg\_stay\_id** – skupina obsahuje buď jednu textúru alebo páry počet textúr, kde jednotlivé dvojice po sebe nasledujúcich textúr zobrazujú zdroj podľa zostatku materiálu, ktorý sa v ňom nachádza (od plného zdroja po prázdny). Prvá z dvojice textúr reprezentuje zdroj v normálnom stave, druhá v stave, keď zo zdroja ťaží nejaká jednotka,
  - **tg\_picture\_id, tg\_burning, tg\_dying, tg\_zombie** – rovnaké pravidlá ako u budov rasy.







## 4 Map Editor

Peter Knut



## 4.1 Úvod

Program Map Editor umožňuje vytvárať a upravovať konfiguračné súbory máp používaných v hre Dark Oberon. Tento editor nie je úplný, je zameraný iba na editovanie povrchu máp. Ostatné súčasti mapy (ako napr. zdroje alebo štartovné pozície hráčov) je potrebné doplniť ručne. Ak už mapa obsahuje tieto súčasti, po editovaní budú zachované.

Ďalším obmedzením editora je to, že predpokladá rovnakú veľkosť všetkých fragmentov vo všetkých segmentoch mapy. (Vysvetlenie pojmov mapel, fragment a segment mapy je uvedené v dokumentácii k vytvoreniu vlastnej mapy v hre Dark Oberon.)

Pre lepšie odlíšenie zobrazovaných fragmentov a orientáciu v mape je možné definovať farebnú schému, v ktorej je k jednotlivým číslam fragmentov uvedená príslušná farba a názov fragmentu. Táto schéma sa dá uložiť na disk do samostatného súboru a opätovne otvoriť.

Po spustení aplikácie sa automaticky vytvorí nová prázdna farebná schéma a maximálna mapa so štandardnou veľkosťou fragmentu 5 mapelov.

## 4.2 Hlavné okno

Hlavné okno aplikácie je rozdelené na dve základné časti. Na ľavej strane sa nachádza tabuľka fragmentov zobrazujúca tiež farebnú schému, zvyšnú časť okna tvorí panel s vyobrazením mapy.

Zo samotnej mapy sa v editore zobrazuje vždy zvolený segment v tvare pravidelnej mriežky. Každé políčko mriežky reprezentuje jeden fragment. Jednotlivé políčka (fragменты) sú od seba odlíšené číslom a farbou. Medzi segmentmi je možné prepínať v ľavej dolnej časti okna.



Obrázok 16: Hlavné okno aplikácie.

## 4.3 Mapa

### 4.3.1 Práca so súborom

Príkazmi z menu Map je možné vytvoriť novú mapu, otvoriť existujúcu mapu alebo uložiť otvorenú mapu. Pri vytváraní novej mapy sa zobrazí dialóg „New map“ s vlastnosťami mapy. Tu je potrebné zadať požadované rozmery mapy a veľkosť fragmentov. Obidva údaje sú v mapeloch. Veľkosť fragmentov nesmie byť väčšia ako niektorý z rozmerov mapy. V prípade, že rozmery mapy nie sú násobkom veľkosti fragmentov, budú zmenšené na najbližší násobok.

Podobne pri otváraní mapy je potrebné v dialógu „Fragments size“ zadať veľkosť fragmentov, ktorú mapa používa.

Vlastnosti mapy je možné kedykoľvek zmeniť v dialógu „Map properties“ (príkaz menu Map/Properties).

Súčasne môže byť otvorená a editovaná práve jedna mapa.

### 4.3.2 Editovanie

Fragmenty sa do mapy vkladajú klepnutím ľavým tlačidlom myši na príslušné miesto v zobrazenej mriežke. Vkladaný fragment je možné zmeniť v tabuľke fragmentov.

Pravým tlačidlom myši sa vloží fragment s číslom nula.

## 4.4 Farebná schéma

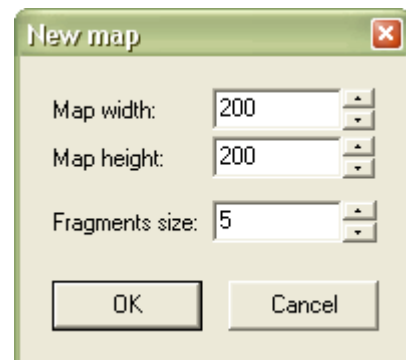
### 4.4.1 Práca so súborom

Príkazy z menu Scheme slúžia na vytvorenie novej farebnej schémy, otvorenie existujúcej schémy alebo uloženie otvorenej schémy. Používanú schému je možné zmeniť kedykoľvek počas editovania mapy.

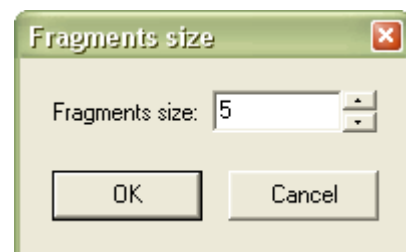
Súčasne môže byť otvorená práve jedna schéma.

### 4.4.2 Editovanie

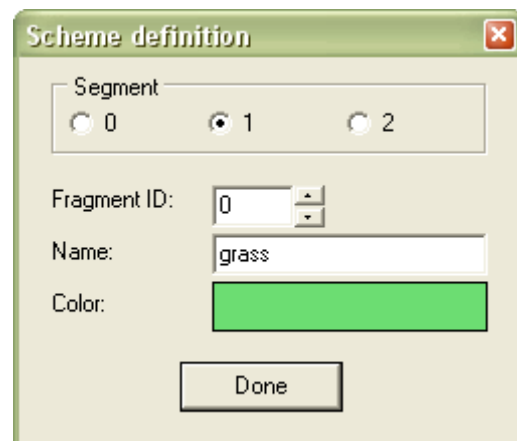
Farebná schéma sa edituje pomocou dialógu „Scheme definition“ (príkaz Scheme/Definition). Pole segment udáva požadovaný segment. Pri zmene čísla fragmentu sa automaticky zobrazí jeho aktuálny názov a farba. Tieto dva údaje je



Obrázok 17: Dialóg "New map"



Obrázok 18: Dialóg "Fragments size"



Obrázok 19: Dialóg "Scheme definition"

možné zmeniť (farba sa mení klepnutím na obdĺžnik zobrazujúci farbu fragmentu). Maximálny počet fragmentov je 256.

Tlačidlo Done ukončí editovanie schémy a zavrie dialóg.

## 4.5 Klávesové skratky

Ctrl+N	Vytvorí novú mapu.
Ctrl+O	Otvorí existujúcu mapu.
Ctrl+S	Uloží otvorenú mapu pod novým názvom.
Shift+Ctrl+N	Vytvorí novú farebnú schému.
Shift+Ctrl+O	Otvorí existujúcu farebnú schému.
Shift+Ctrl+S	Uloží otvorenú farebnú schému pod novým názvom.
F1	Zobrazí okno O programe.





# 5 Data Editor

Peter Knut





## 5.1 Úvod

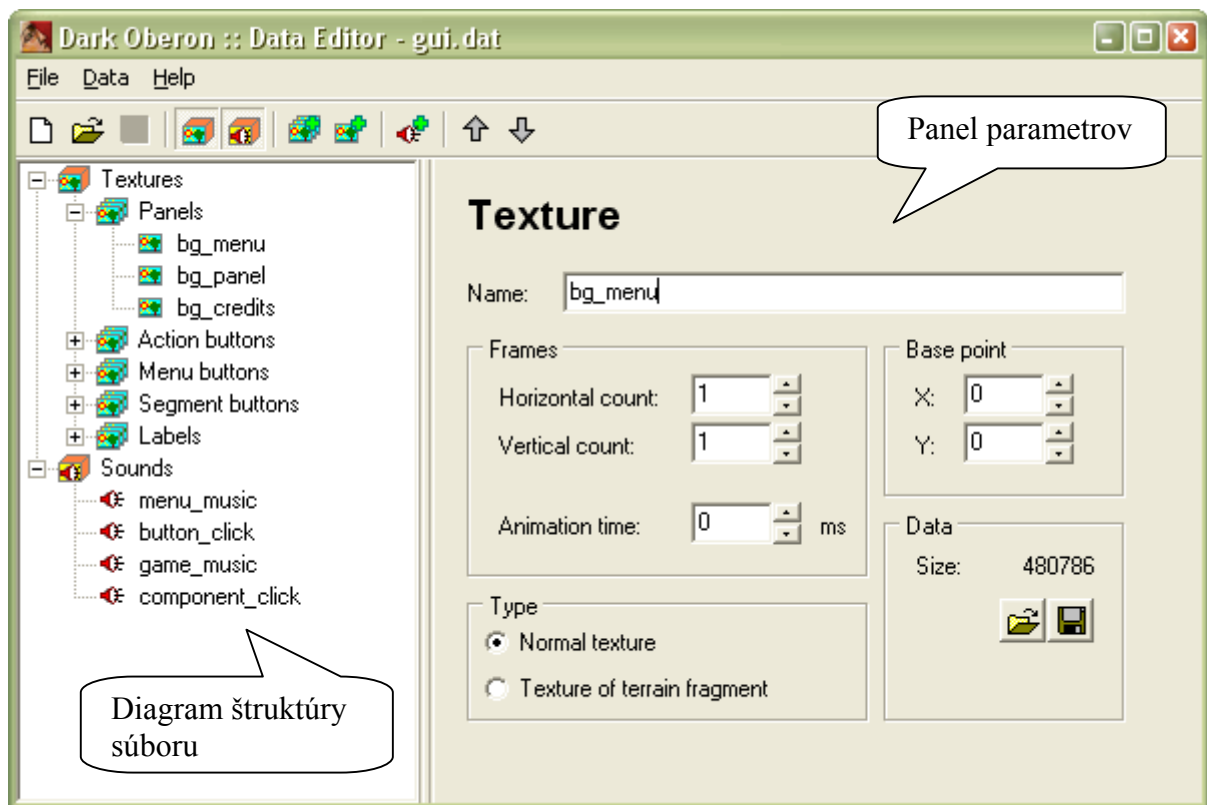
Program Data Editor umožňuje vytvárať a upravovať dátové súbory používané v hre Dark Oberon. Dátový súbor (\*.dat) obsahuje dva základné typy záznamov: textúry (obrazové dáta) a zvuky.

Textúry sú organizované v skupinách, pričom každá textúra patrí do práve jednej skupiny. Skupina nemôže obsahovať ďalšie skupiny. Počet textúr v skupine je neobmedzený.

Zvukové záznamy nie sú organizované, ich počet je rovnako neobmedzený.

## 5.2 Hlavné okno

Hlavné okno aplikácie je rozdelené na dve základné časti. Na ľavej strane sa nachádza stromový diagram zobrazujúci štruktúru dátového súboru, zvyšnú časť okna tvorí panel pre editovanie parametrov jednotlivých záznamov.



Obrázok 20: Hlavné okno aplikácie.

## 5.3 Práca so súborom

Príkazmi z menu File je možné vytvoriť nový prázdny súbor, otvoriť existujúci súbor, uložiť a zavrieť otvorený súbor. Pri zatváraní neuloženého súboru sa zobrazí výzva pre uloženie. Súčasne môže byť otvorený a editovaný iba jeden súbor.

Aktuálna podporovaná verzia dátového súboru je 3. Editor umožňuje otvoriť i súbory v predchádzajúcich verziách, pričom tieto budú automaticky prevedené na aktuálnu verziu.

## 5.4 Editovanie dát

Aby bolo možné pridávať dátové položky, je potrebné najprv aktivovať požadovaný typ dát. Tieto typy sa aktivujú a deaktivujú príkazmi: Data/Textures pre textúry a Data/Sounds pre zvuky. Po aktivovaní je možné príkazmi z menu Data pridávať, mazať a presúvať jednotlivé záznamy. Pri deaktivovaní sa všetky záznamy patriace k príslušnému typu zmažú.

Pri pridávaní záznamu je užívateľ automaticky vyzvaný, aby zvolil zdrojový súbor pre textúru či zvuk. Zdrojový súbor sa po načítaní stáva súčasťou dátového súboru.

Označením položky v diagrame sa na paneli zobrazia parametre, ktoré je možné editovať.

### 5.4.1 Skupina textúr

Tu je možné zmeniť iba názov skupiny. Tento názov môže obsahovať neobmedzený počet znakov.

### 5.4.2 Textúra

Textúry v hre Dark Oberon sú animované. Animácia sa dosiahne tým, že zdrojový obrázok textúry je rozdelený na jednotlivé snímky. Všetky snímky majú rovnakú veľkosť a sú usporiadané do pravidelnej mriežky. Príklad takejto textúry je na obrázku.

V poli Frames dá nastaviť horizontálny a vertikálny počet snímkov v textúre a dĺžka animácie v milisekundách. Počet snímkov je v intervale  $\langle 1, 100 \rangle$ , dĺžka animácie v intervale  $\langle 0, 10000 \rangle$ .

Parametre X, Y v poli Base point definujú bod v snímku, ktorý určuje stred súradnicovej sústavy použitej pri vykresľovaní. Štandardne je tento stred v bode  $[0, 0]$ . Pri nenulových hodnotách dôjde k lokálnemu posunutiu textúry. Napr. pri hodnotách  $[10, 5]$  sa textúra posunie o 10 bodov doľava a 5 bodov dole. Je možné zadať i bod mimo veľkosti snímku. Povolené hodnoty sú v rozsahu  $\langle -1024, 1024 \rangle$ .

Typ textúry určuje spôsob, akým sa textúra vykresľuje. Ak je textúra určená pre fragment terénu, je potrebné nastaviť typ na Texture of terrain fragment. Inak je typ nastavený na Normal texture.

V poli Data sa nachádza údaj o veľkosti zdrojového súboru a tlačidlá pre načítanie nového súboru a uloženie súboru na disk. Zdrojovým súborom pre textúru je obrázok vo formáte TGA. Maximálna veľkosť obrázku je 1024x1024 bodov.

Názov textúry je neobmedzený.



Obrázok 21: Príklad animovanej textúry

### 5.4.3 Zvukový záznam

Ako zdrojový súbor pre zvuk je možné použiť formáty: WAV, MP2, MP3, OGG, RAW, MOD, S3M, XM, IT, MID, RMI, SGT. Použitý formát je vyznačený v poli Format.

Zvukový záznam môže byť dvojakého typu:

- **Sample** – zdrojový súbor je pri načítaní dátového súboru uložený do pamäte a teda je k nemu veľmi rýchly prístup. Tento typ je vhodný pre krátke a časté zvuky,

- **Stream** – zdrojový súbor sa nenačíta dopredu. Pri prehrávaní sa číta priamo z disku. Vhodný pre veľmi dlhé a málo časté zvuky. Napr. pre hudbu v pozadí hry.

Názov zvukového záznamu je neobmedzený.

## 5.5 Export a import textúr

Príkazy File/Export/All textures a File/Import/All textures slúžia pre hromadný export a import zdrojových súborov pre textúry.

Exportované súbory majú tvar <zadané meno><vygenerované číslo>.tga. Číslovanie začína číslom 1000 a pre každý nasledujúci obrázok sa zväčšuje o jedna. Pri importovaní stačí zvoliť jeden zo súborov takéhoto tvaru. Počet vstupných obrázkov sa musí zhodovať s počtom textúr v dátovom súbore.

## 5.6 Klávesové skratky

Ctrl+N	Vytvorí nový dátový súbor.
Ctrl+O	Otvorí existujúci dátový súbor.
Ctrl+S	Uloží otvorený súbor.
F1	Zobrazí okno O programe.
Del	Zmaže položku označenú v diagrame súboru.