

## Tvorba misí pro Emergency 3

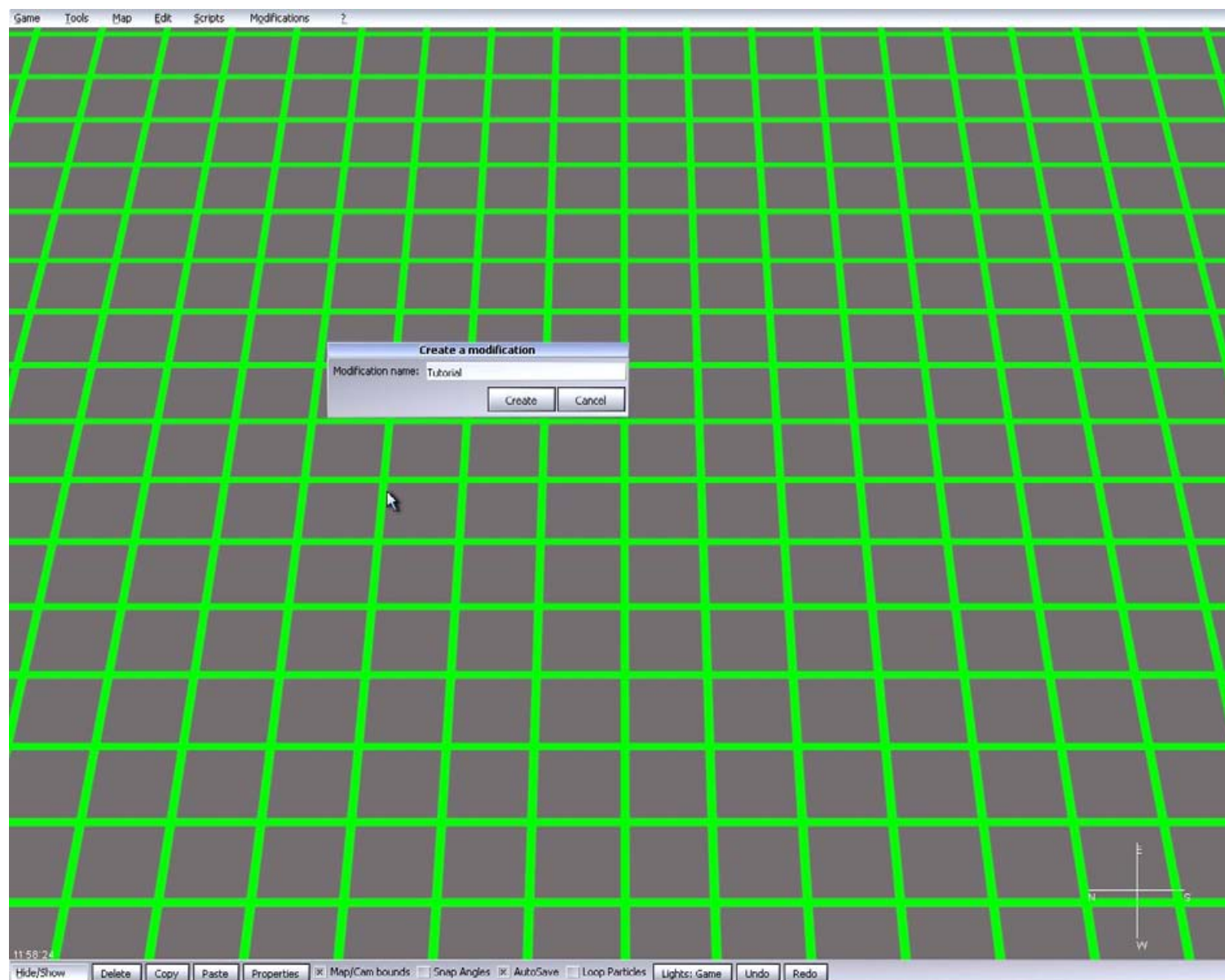
### Předmluva:

Víte, vytváření mise, nebo jakéhokoliv doplňku pro nějakou hru je velice náročný a vysoce vědecký proces a jeho úspěch závisí na mnoha okolnostech, jako třeba, jak se herní editor nahrál do paměti (dobře nebo špatně? Jako když vstanete levou nohou z postele), na počasí a vzájemné poloze planet sluneční soustavy :) No dobře, teď vážně, fakt to není žádná sranda a já říkám předem, že pokud chcete dělat mise do Emergency 3 musíte znát angličtinu, aspoň trochu a mít nějaké zkušenosti s programováním, aspoň nějaké :) Nebo třeba s psaním skriptu pro nějakou jinou hru, protože já vážně nemám čas vysvětlovat programování a překládat celej editor. Rada do začátku, přečtěte si anglický "návod" k editoru, který je v hlavním adresáři hry jako pdf dokument. Pak je taky dobré prolézt soubory hry a podívat se co kde je. Emergency 3 má snad všechna nastavení uložena v xml dokumentech, které se dají dobře pochopit už z významu použitých značek (pokud umíte anglicky :)). Smiřte se s tím, že žádná hra není inteligentní, jen tak vypadá podle toho, jak ji naprogramovali. Takže hledejte, kde je třeba uložen ten text který se mi zobrazuje při načítání mise nebo v průběhu mise. Nebo kde je napsáno jaké jednotky mají být v které misi dostupné. Hra to prostě musí mít někde nadefinované, nenabídne vám v první misi hasičský tank jen proto že ho potřebujete k hašení.

Fajn, tak začneme :)

### Nová modifikace:

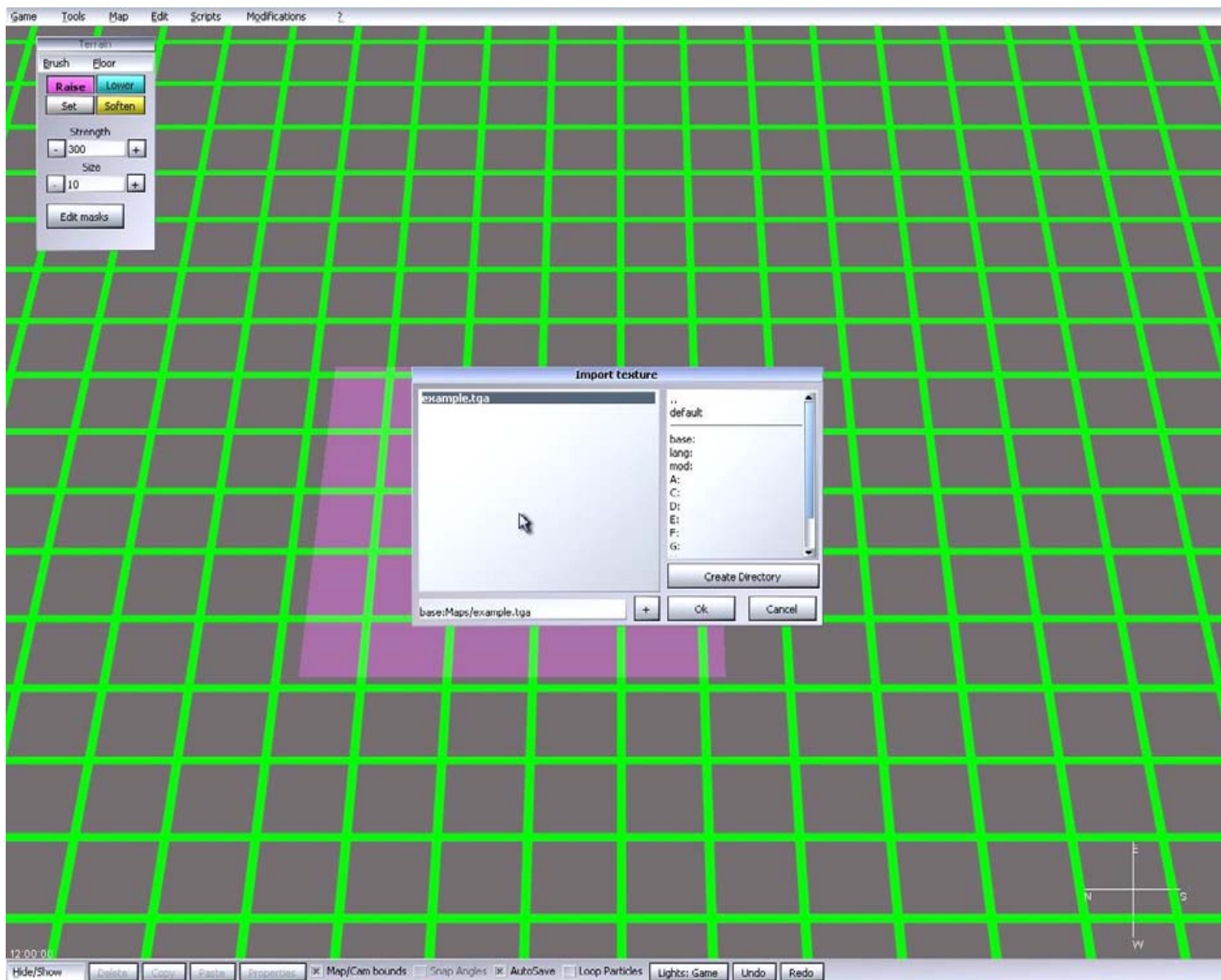
Až spustíte editor, vytvořte si hned novou modifikaci (**Modifications->Create a new mod...**). To je důležité pro to, aby se všechny soubory které vytvoříte, ukládaly do adresáře s modem a vy jste si tak nepo... (počet teček neodpovídá chybějícím písmenům!) původní hru.



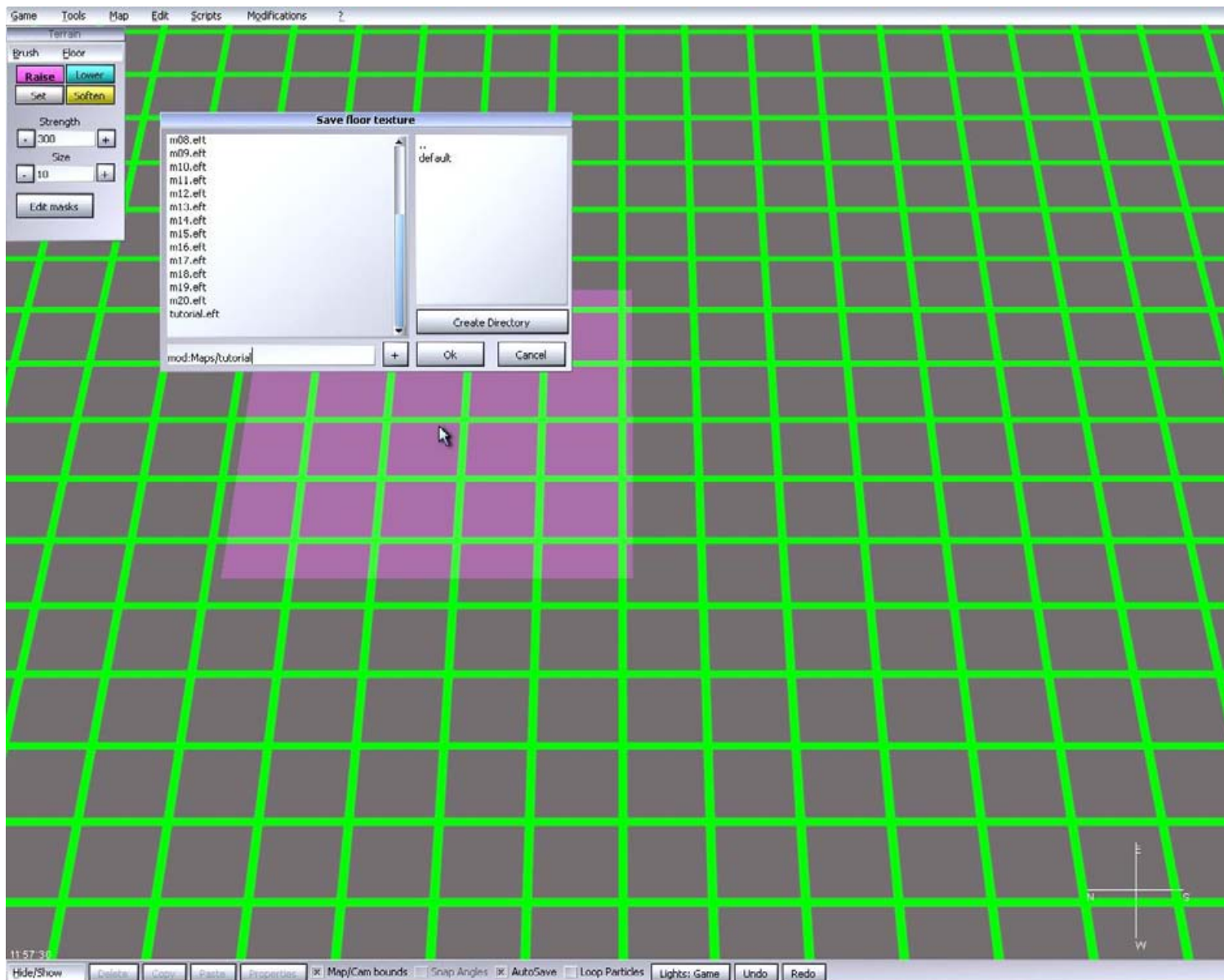
### Něco na podlahu:

To bude asi nejproblematičtější část mise. Teda aspoň pro mě byla. Pro někoho jiného to budou možná spíš skripty. Nicméně, tohle je věc kterou musíte udělat sami. Texturu na "podlahu" musíte vytvořit v nějakém externím programu. Třeba ve Phostoshopu. Tak jsem dělal texturu (která není moc detailní, šlo jenom o ty silnice) pro svoji mapu. Použil jsem na to trávu, kterou jsem získal z ukázkového obrázku example.tga, který je

v originálním adresáři s misemi (Data/Maps). Neporadím vám ani moc s rozměry. Moje původní textura byla velká 4096x4096 pixelů, ale když jsem ji načel do editoru, silnice byly moc široké a vypadalo to hrozně :) Když jsem ji zmenšil na polovinu, bylo to už fajn. Jak je napsáno v anglickém návodu k editoru, 12.5 pixelu odpovídá jednomu metru, jestli se s tím chcete rýsovat :) Až vytvoříte svoji texture, uložte ji ve formátu tga, 24 bitů RGB. Přesně je to napsané taky v tom pdfku. Když to uložíte jinak než zadávají pravidla, editor texture prostě nenačte :) Texture do editoru dostaneme příkazem Import. Nejdřív ale zmáčkněte F2, čímž vyvoláte nástroj pro úpravu terénu. Na něm potom zvolte **Floor->Import Texture** a načtete texture example.tga z adresáře Maps. Pokud tvoříte mapu podle tutoriálu, pokud tvoříte svoji mapu, načtete svoji texture.



Editor se vás následně zeptá, kam chcete uložit tuto texture pro podlahu. Tga texture bude ještě překonvertována na formát eft, což je nějaký vlastní formát Emergency 3 na podlahovou krytinu :) Nechte cestu co tam je a za ní napište jen název. **mod:** v názvu cesty značí, že se to uloží do příslušné cesty v adresáři vašeho modu. V případě našeho tutoriálu pak texture podlahy bude uložena v adresáři Mods/Tutorial/Maps/tutorial.eft.



## Modifikace terénu:

Stejný panel nástrojů přes který jsme načtli texturu na podlahu slouží i k úpravě terénu. K vytváření kopečků či kaňonů a jiných děr :) Máme na výběr ze čtyř nástrojů **Raise**, který vyvýší terén, **Lower**, snižuje terén, **Set**, nevím přesně co dělá, ani jsem ho nepoužil a **Soften**, který, myslím že tomu můžeme říkat že jakýmsi způsobem vyhlazuje terén. Každý nástroj se zapne klikem na příslušné tlačítko. Každý má jinou barvu, takže spolehlivě poznáte, který je zrovna aktivní. Můžete si vybrat též štětec v menu **Brush**. Ne na všechno se hodí čtvercový štětec. Nastavte si také sílu **Strength** štětce, základní hodnota 300 bývá moc silná a taky velikost **Size**, podle toho jak potřebujete. Tyto nástroje jako většina ostatních patří do skupiny, která se špatně popisuje, takže není lepší cesty jak se s nimi naučit, než si to vyzkoušet. Udějte na mapě nějaké kopeček a vůbec co chcete, třeba takhle:

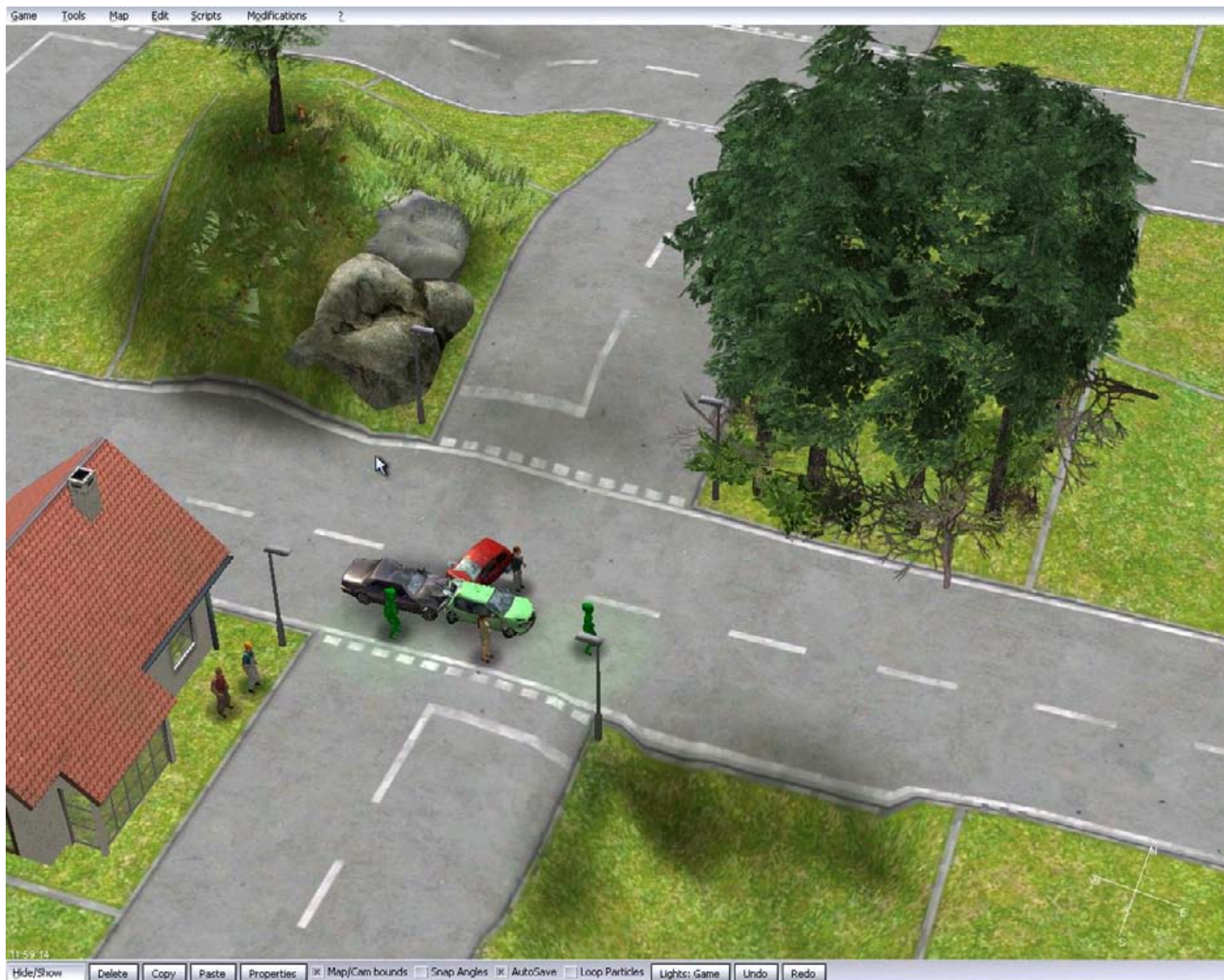




### Editace scény:

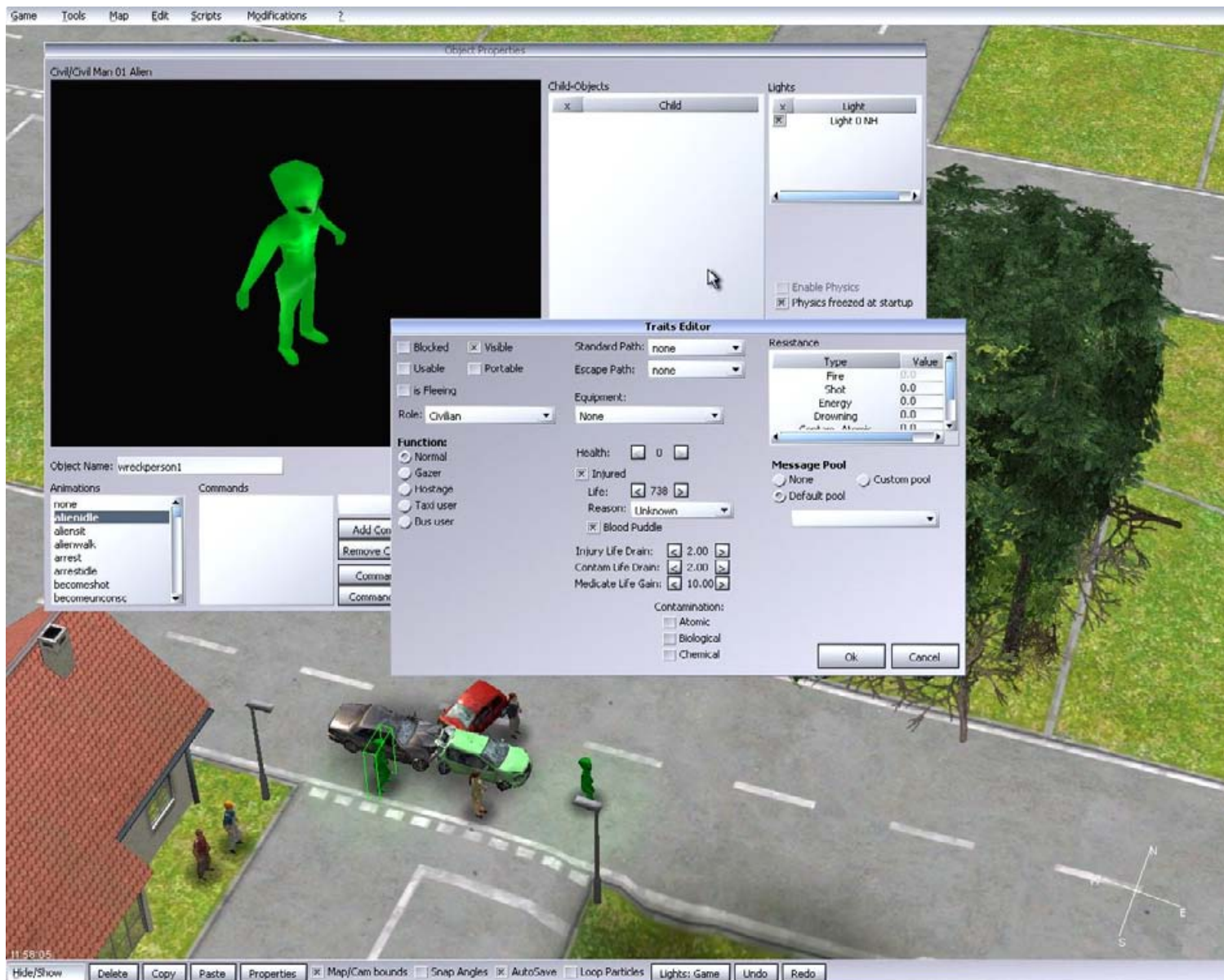
Aneb vložíme si na mapu nějaké baráčky, stromečky atd. Stiskněte F1 nebo klikněte na příkaz **Edit->Scene**. Zobrazí se okénko pro editaci scény. Můžete na něm vidět že můžete na mapu vložit čtyři typy prototypů - objektů, a to **Objekty**, kde jsou stromy, světla, hydranty a další všemožné věci, **Auta**, buď nerozbitá nebo rozbitá, která v sobě mohou schovávat oběť, **Lidičky**, civilisti, ale i záchranné jednotky a **Domy**, statické nebo takové do kterých se dá vejít. Takže si něco vyberte a něco si na mapu poskládejte. Objekty můžete otáčet kolem všech tří os. Slouží k tomu klávesy Shift, Ctrl a Alt vždy za držení pravého tlačítka myšičky a pohybování myši ve směru otáčení. Takže například Alt + pravé myšičko a tažení myši.





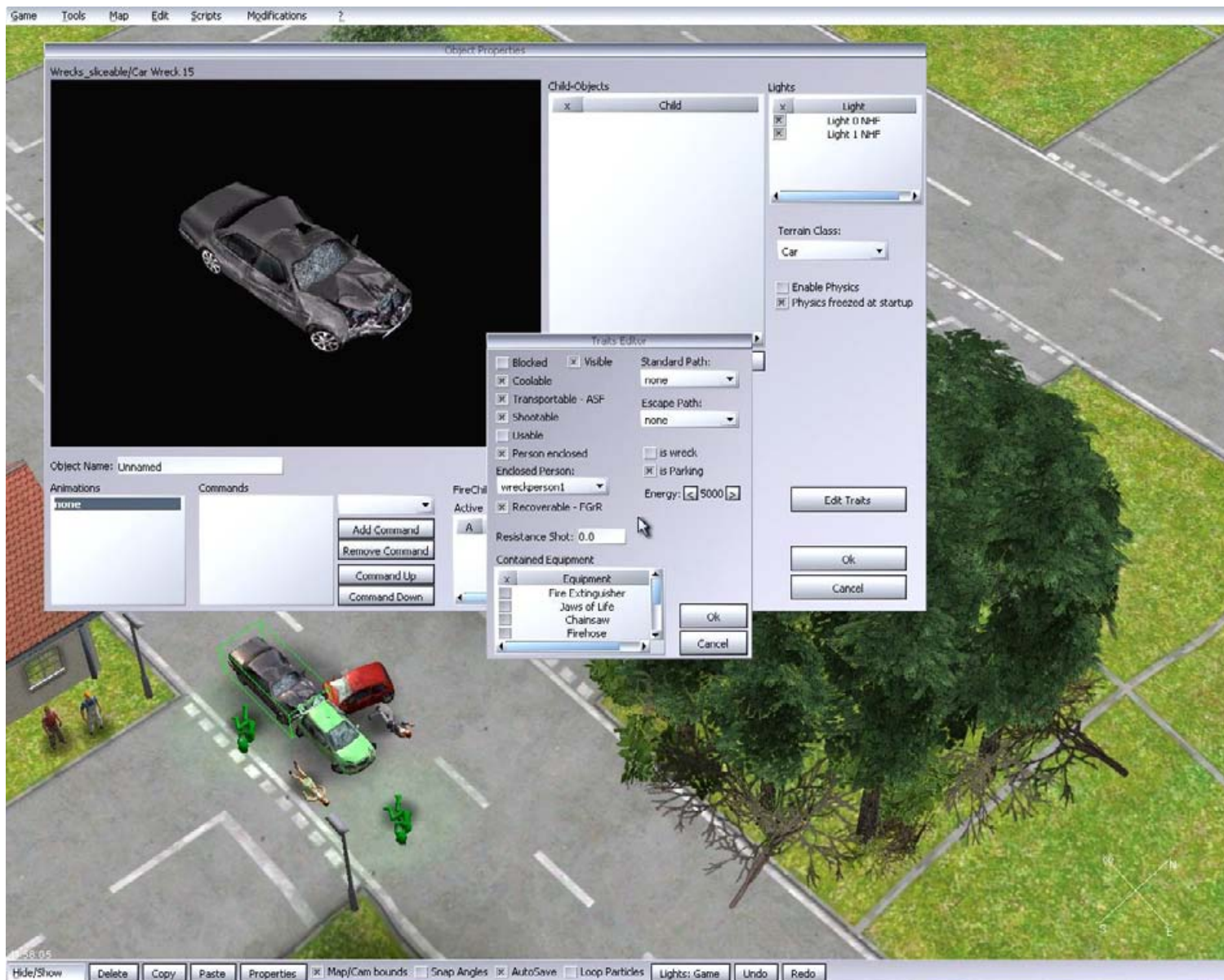
### Nastavení vlastností objektům:

Občas je problém s umístěním objektů. Někdy je chceme prostě jinak než to umístí hra. Objekty bývají nesprávně umístěné, nebo nepěkně umístěné zvláště na kopcích. Abychom mohli objekt umístit tak jak potřebujeme, musíme ho označit kliknutím myši a vyvolat jeho vlastnosti ( v dolní liště klikněte na tlačítko **Properties**. Tam pak v sekci **Placement** zvolte **Custom** a klikněte na **Ok**. To většinou objekt přesune nad terén, že není zarovnan s podlahou. Pomocí klávesy Alt a levého myšítka objekt přesuňte zpět na zem a umístěte ho tak jak potřebujete. V ukázkové mapě jsem takto musel upravit polohu kamenů. Okno s vlastnostmi objektu ale slouží k více věcem. Můžete tam nastavit animaci objektu, to hlavně u lidí. Příkazy, Commands jsem ani nepoužil. Důležité je jméno objektu. U tří postav které jsou u nabouraných aut zapište nějaké jméno do políčka **ObjectName**. Třeba wreckperson1 až wreckperson3. Potom klikněte na tlačítko **Edit Traits**. Zaškrtněte políčko **Injured** a nastavte **Life** na nějakou nižší hodnotu. Pokud chcete aby pod postavou byl krvavý flek, slouží k tomu políčko **Blood Puddle**. Tak bude hra vědět že tyto osoby jsou zraněné a my to pak budeme moci využít abychom zjistili, jestli byly všechny zraněné nebo mrtvé osoby odvezeny do nemocnice. Všechna nastavení pro aliena jsou vidět na následujícím obrázku:



Já mám na mapě celkem čtyři zraněné. U třech jsem nastavil jméno, protože to budou lidé uvnitř nabouraných aut. A jak je dostaneme do aut, to si ukážeme teď. Postupně u všech tří aut nastavíme, který človíček má být uvnitř uvězněn. Otevřete pro každé auta dialog s nastavením a opět klikněte na tlačítko **Edit Traits**. Zaškrtněte políčko **Person Enclosed** a následně se vám pod ním zpřístupní seznam človíčků (**Enclosed Person**), které jsme pojmenovali. Vyberte pro každé auto nějakého. V editoru sice žádnou změnu nevidíte, ale ve hře budou lidičky v autech :) Opět obrázek:





## Ulice:

Hra také potřebuje vědět, kudy mohou jezdit záchranné vozy. To se dozví pomocí ulic. Spusťte **Edit->Streets** nebo Shift+F7. Zobrazí se okénko pro editaci ulic. Klikněte na tlačítko **New**, čímž vytvoříte novou ulici v seznamu. Následně se vám zobrazí i dialog Polyline, pomocí něhož nadefinujeme ulici. Z něj si vystačíme s tlačítky **Add**, **Insert**, **Move** a **Delete**. Klikněte na tlačítko **Add** a pak na mapu. Na mapě se zobrazí bod. Přidejte další o kus dál. Pokud bude ulice širší nebo užší než cesta na podlaze, nebo prostě než jak chcete aby byla cesta široká nebo úzká, chyťte modrý čtverec na okraji cesty pomocí myši a cestu zúžete nebo roztáhněte. Pokud budete chtít přidat bod na existující ulici, použijte tlačítko **Insert**, pro odebrání **Delete** a pro přesun **Move**. Dialog Polyline používá ještě mnoho dalších nástrojů, takže je důležité umět pracovat aspoň s jmenovanými základními čtyřmi tlačítky.auta jezdí vždy po krajních modrých čarách v příslušném směru.



## Cesty:

Cesty mají stejný účel jako ulice. Jenom neslouží záchranářským vozům, ale civilním vozům a pand'ulákům. Pomocí cest tedy nadefinujeme kde mohou jezdit civilní auta a chodit pand'uláci. Editor ulic se spustí příkazem **Edit->Paths**, nebo F7 a používá se stejně jako editor ulic. Jediný rozdíl je v tom, že pokud budete chtít cestu zúžit nebo roztáhnout, musíte držet ještě klávesu Ctrl a pak teprve můžete pohybovat s modrým obdélníkem. Auta i lidé chodí v daném směru vždy po krajních modrých čarách. U cest ještě můžete nastavit jejich typ: **Forward** - chodí a jezdí se od začátku cesty do konce, od prvního bodu k poslednímu, **Backward** - chodí a jezdí se od posledního bodu cesty k prvnímu, **Ping Pong** - auta nebo lidé se na konci cesty otočí a jedou, jdou zpátky, **Loop** - chodí a jezdí pořád dokola. Poslední věc co se tu dá nastavit je rychlost. Pro lidi by rychlost měla být myslím mezi 1.5 a 2.0 a pro auta mezi 5.0 a 7.0 ale přesně je to také v anglickém "manuálu".





## Virtuální objekty:

To jsou objekty speciálního významu, které ve hře nejsou vidět. Například se pomocí nich definuje oblast vody, oblast "země", místa kam se nedá jít, místa kde může hledat záchranářský pes a další. Všechny volby "mluví" do určité míry, samy za sebe. Chce to i trochu představivosti a víry v logiku editoru :) My použijeme virtuální objekt k definování jezera. Editor virtuálních objektů spustíte **Edit->Virtual Objects** nebo klávesou F5. Zobrazí se dialog podobný předchozím. Klikněte na tlačítko **New** pro vytvoření nového virtuálního objektu. Následně se zobrazí dialog Polygon (stejný jako Polyline), který už známe. U vlastnosti **Terrain Type** nastavte pro jezírko *Shallow Water*. Celý nakreslený polygon označte myší a pomocí klávesy Alt a levého myšítka ho přesuňte o něco dolů, jako je to na obrázku. Předtím ale klikněte na tlačítko **Move**, jinak si bude editor myslet že chcete přidávat body do polygonu.

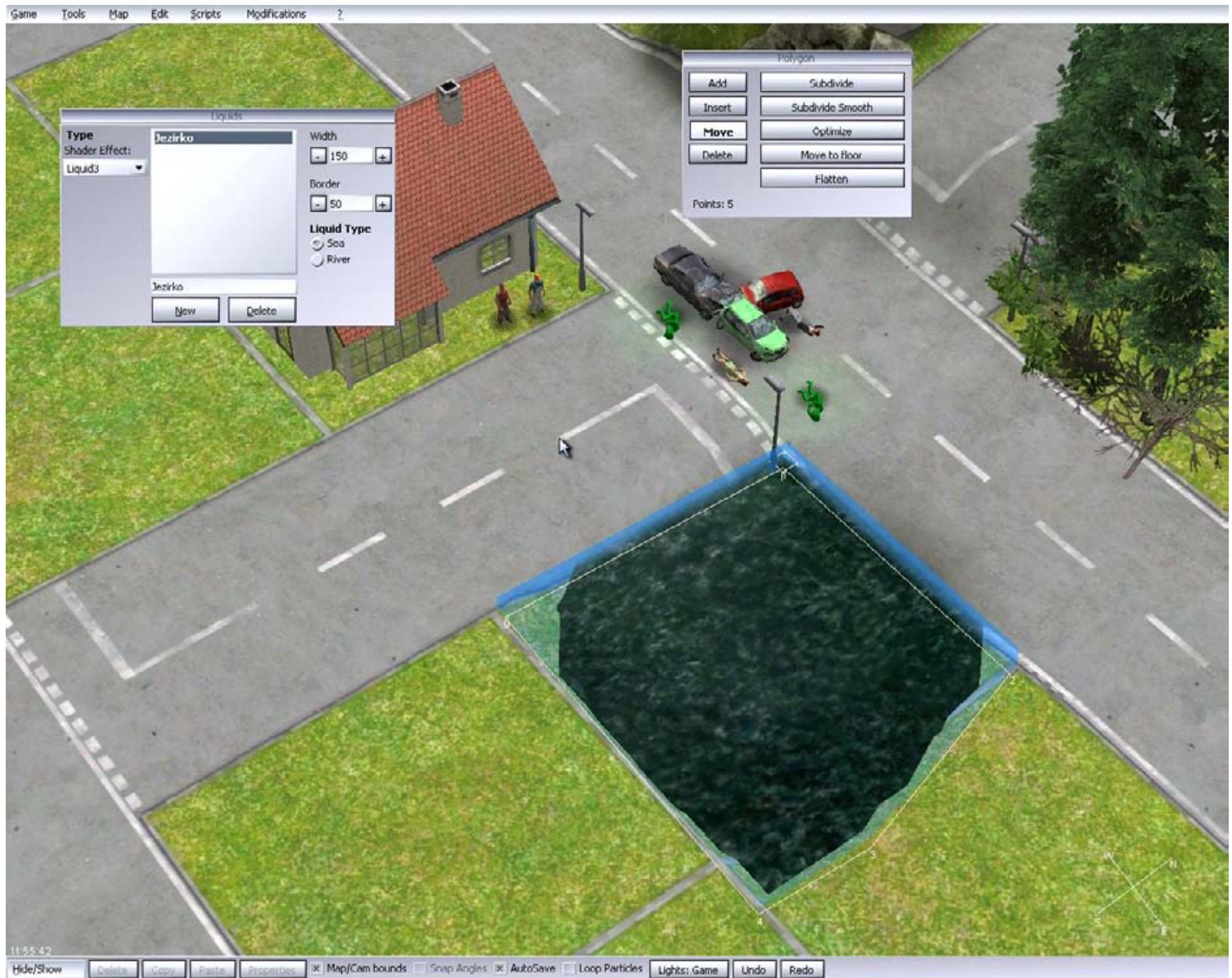




## Voda:

Z mé zkušenosti vím, že je většinou lepší vytvořit nejdřív virtuální objekt pro jezírko nebo řeku a až potom jezírko nebo řeku. Prošel jsem si mnoho mazání a úprav než řeka v mé první mapě začala nějak fungovat. Tak doufám, že když jsem zařadil Virtuální objekty v tutoriálu jako první, že vám ušetřím nějaké to mazání a nové zkoušení :) Řeky, jezera a ostatní jiné vodní louže se přidávají přes editor **Edit->Liquids** nebo klávesa F4. Je to to samé jako dřív, tlačítkem **New** přidáme nový objekt a pomocí dialogu Polygon ho nakreslíme. Pokud se jedná o řeku, zaškrtněte políčko **River**, pak se hodí i parametr **Width**, kterým definujete šířku řeky. V případě vodních louží různých rozměrů použijte volbu **Sea**. V rozevíracím seznamu **Shader Effect** si pak ještě vyberte jak má řeka nebo louže vypadat :). Po nakreslení polygonu louže, nebo řeky je pak občas vhodné celý nový objekt vybrat pomocí myši a pomocí známe kombinace Alt + levé myšítko ho přeunout do vhodné výškové pozice :) No a jestli máte šťastný den a budete chtít vložit paňáčka do vody, jakmile se s ním přemístíte nad vodu, nastaví se u něj animace tonutí a také bude na úrovni vodní hladiny. Pokud máte smůlu, nebude se topit nebo nebude na úrovni vodní hladiny a pak vám nezbude než hledat v čem je chyba. Většinou pomůže snížení terénu, poupravení virtuálního objektu nebo předělání vody nebo všechno najednou :(

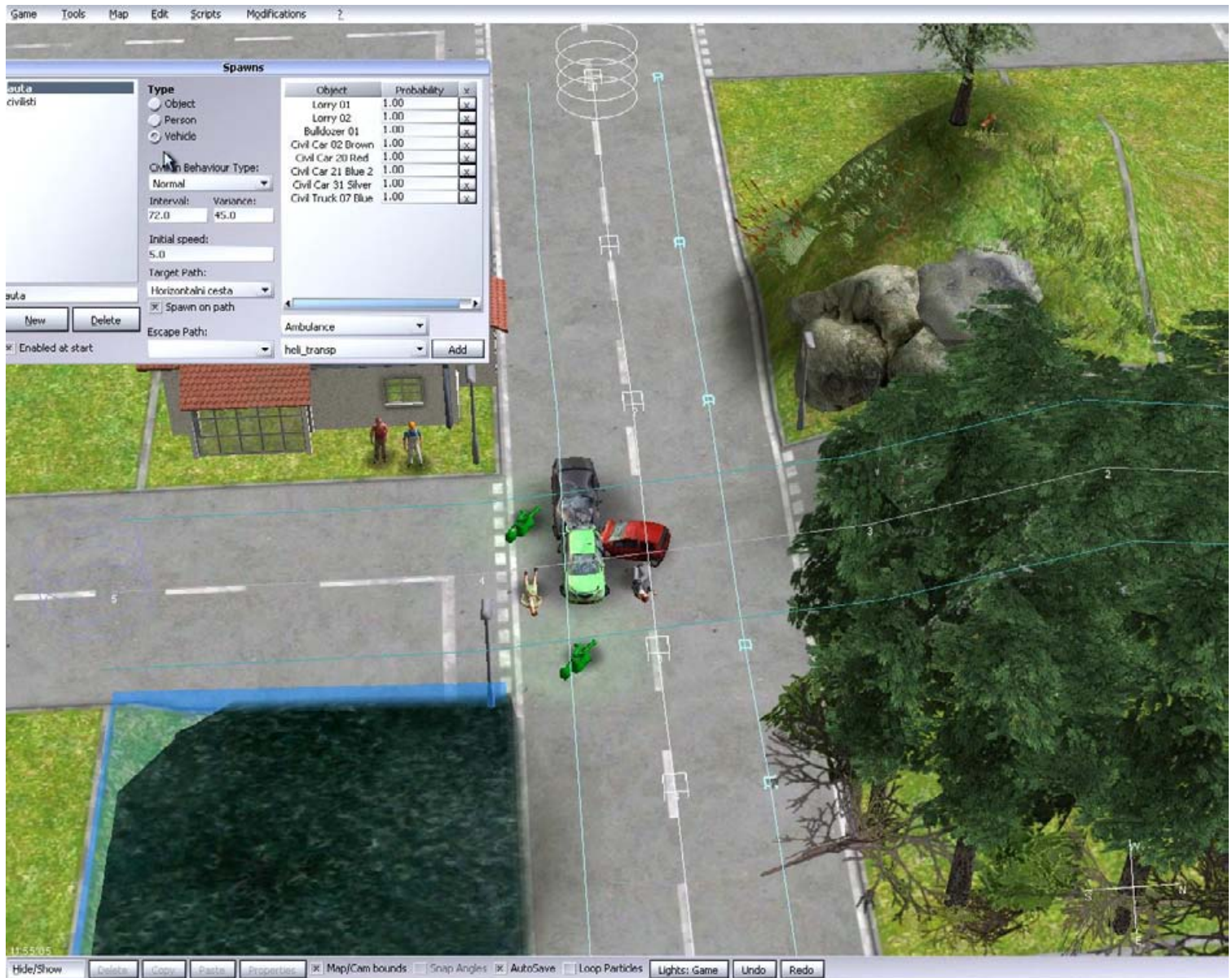




## Body "zrození"

Další z věcí co se špatně překládá :) Spawn Points - F9 nebo **Edit->Spawn Points** slouží k definování míst, bodů kde se budou generovat auta jezdící po cestách a chodci. Nový bod vytvoříme jako vždy tlačítkem **New**. První ze všeho si zvolíte typ **Type**. Nevím k čemu se přesně používá typ objekt, ale já jsem zatím použil jen Person nebo Vehicle pro lidi nebo auta. Pak můžete nastavit další vlastnosti, jako chování lidí, interval v kterém se budou generovat **Interval**, různorodost (myslím) **Variance** (ohledně těchto hodnot, nevím jaké by měly být, házím je tam od oka :)), počáteční rychlost **Initial Speed**, pro lidi nastavte maximálně 2 a pro auta aspoň 5. **Target Path** je cesta po které se budou vygenerovaní lidé nebo auta pohybovat, takže nějakou zvolte. Poslední věcí je přidání lidí nebo aut do seznamu. Aby hra věděla z jakých lidí nebo aut má vybírat. K tomu slouží seznam úplně vpravo, dva rozevírací seznamy pod ním a tlačítko **Add**. Určitě přijdete na to jak se to používá :)

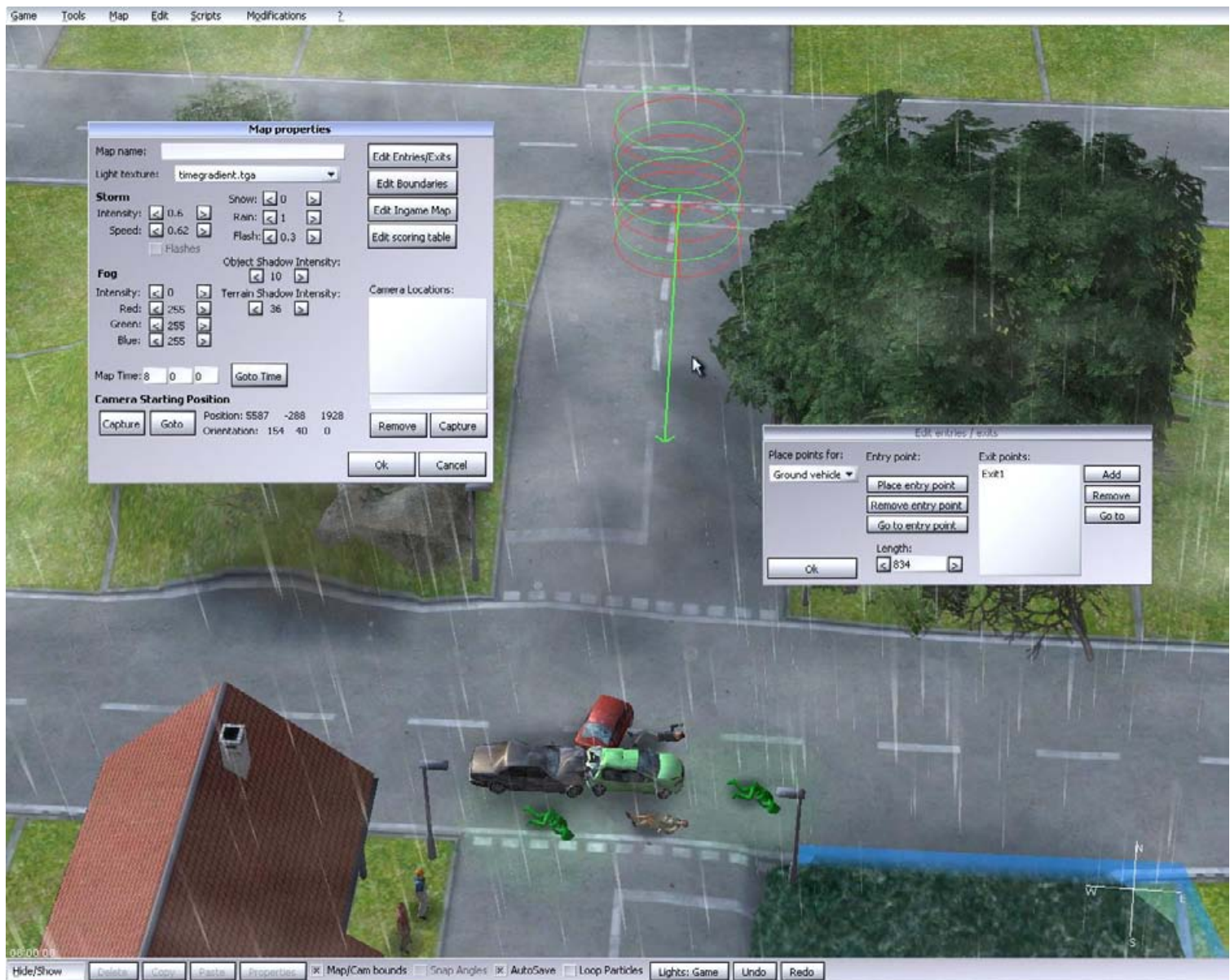




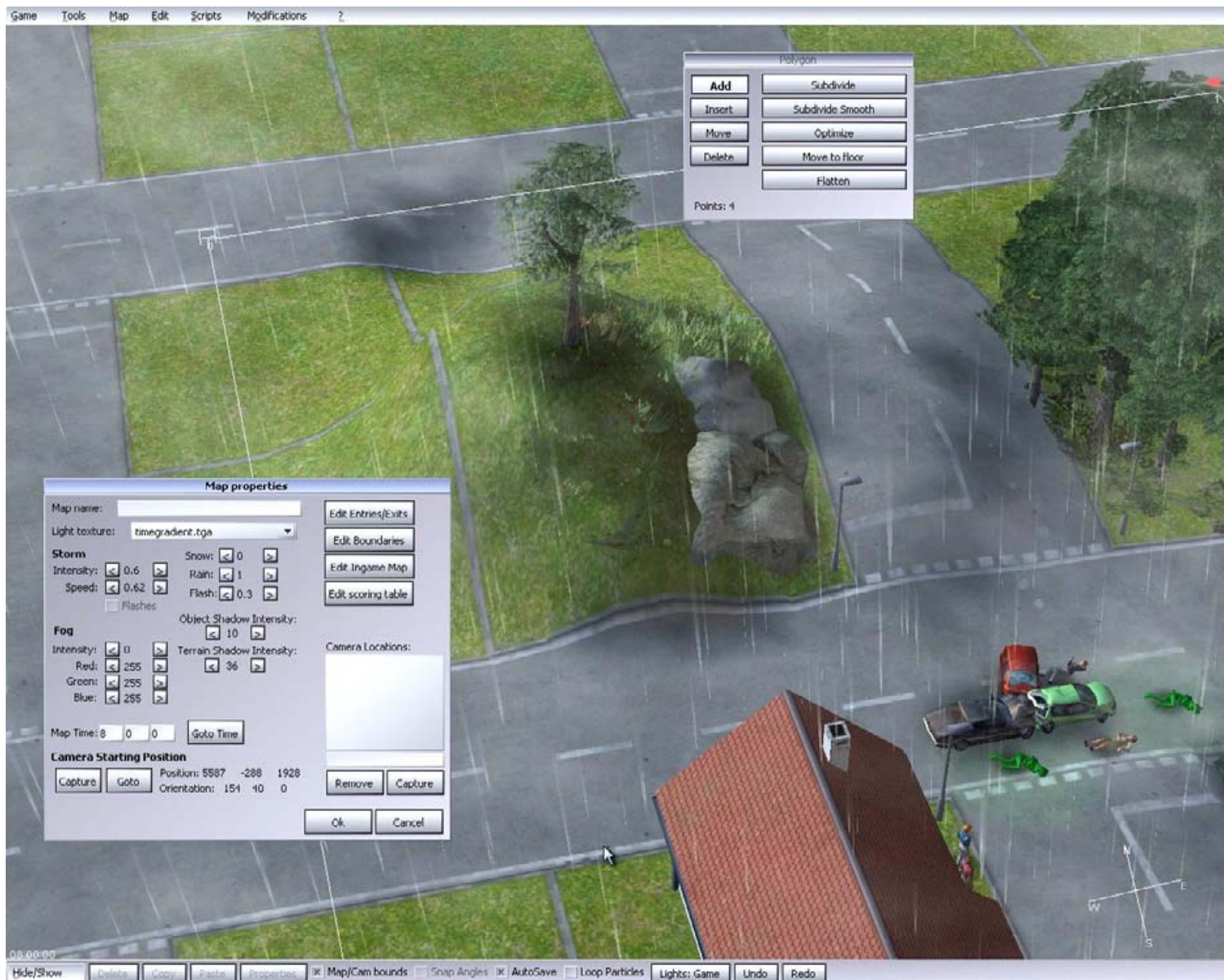
## Nastavení mise:

Jedna z posledních položek kterou se budeme v editoru zabývat. Otevřete panel pro nastavení mise **Edit->Map properties** nebo **Shift+F9**. Otvře se vám okénko, na kterém si můžete nastavit mnoho věcí :) Například Bouřku, její intenzitu a rychlost. Můžete si nastavit zda má pršet, nebo sněžit, a samozřejmě také jak hodně. Nechybí ani blesky, u kterých můžeme volit jak často mají blyškat :) Další možností je nastavení mlhy, můžeme zvolit její intenzitu a barvu. Nezapomeňte nastavit čas, kdy se mise odehrává. Jedním z důležitých nastavení, je také nastavení startovní pozice kamery, tj. první místo které uvidíte po startu mise. Většinou místo katastrofy :) Takže si nějaké místo najděte, natočte si kameru jak potřebujete a stiskněte tlačítko **Capture** pod textem **Camera Starting Position**. Chtělo by to také, aby hra věděla odkud mohou přijet záchranné jednotky. K tomu slouží tlačítko **Edit Entries/Exits**, po jehož stisku se jako vždy objeví další okno. Ale nebojte, je jednoduché. Jeho levá část se věnuje umístění bodů, kudy na mapu vjíždějí záchranáři. Můžete nastavit bod pro suchozemské jednotky, vodní i létající :) Vyberte si typ ze seznamu **Place points for:** a klikněte na tlačítko **Place Entry Point**. Pro každý typ je možné mít jen jeden přístupový bod! Na mapě se objeví zelený válec (vlastně jenom 4 kružnice nad sebou :) se zelenou šipkou. Pomocí myši přemístíte tento válec kam potřebujete a pomocí pravého tlačítka myši natočte šipku do správného směru. Ukazuje v jakém směru jednotky přijedou, takže většinou to chce nastavit podle nějaké příjezdové cesty :) Pravá část okna se zase věnuje místům, v kterých budou záchranáři mapu opouštět. Těch můžete mít kolik chcete. Přidáte ho pomocí tlačítka **Add** a na mapě se objevuje jako červený válec (opět čtyři kružnice nad sebou :)). Umístěte ho stejným způsobem jako vstupní bod, pomocí myšičky. Šipku nastavovat nemusíte, pač ji tyto body nemají :)



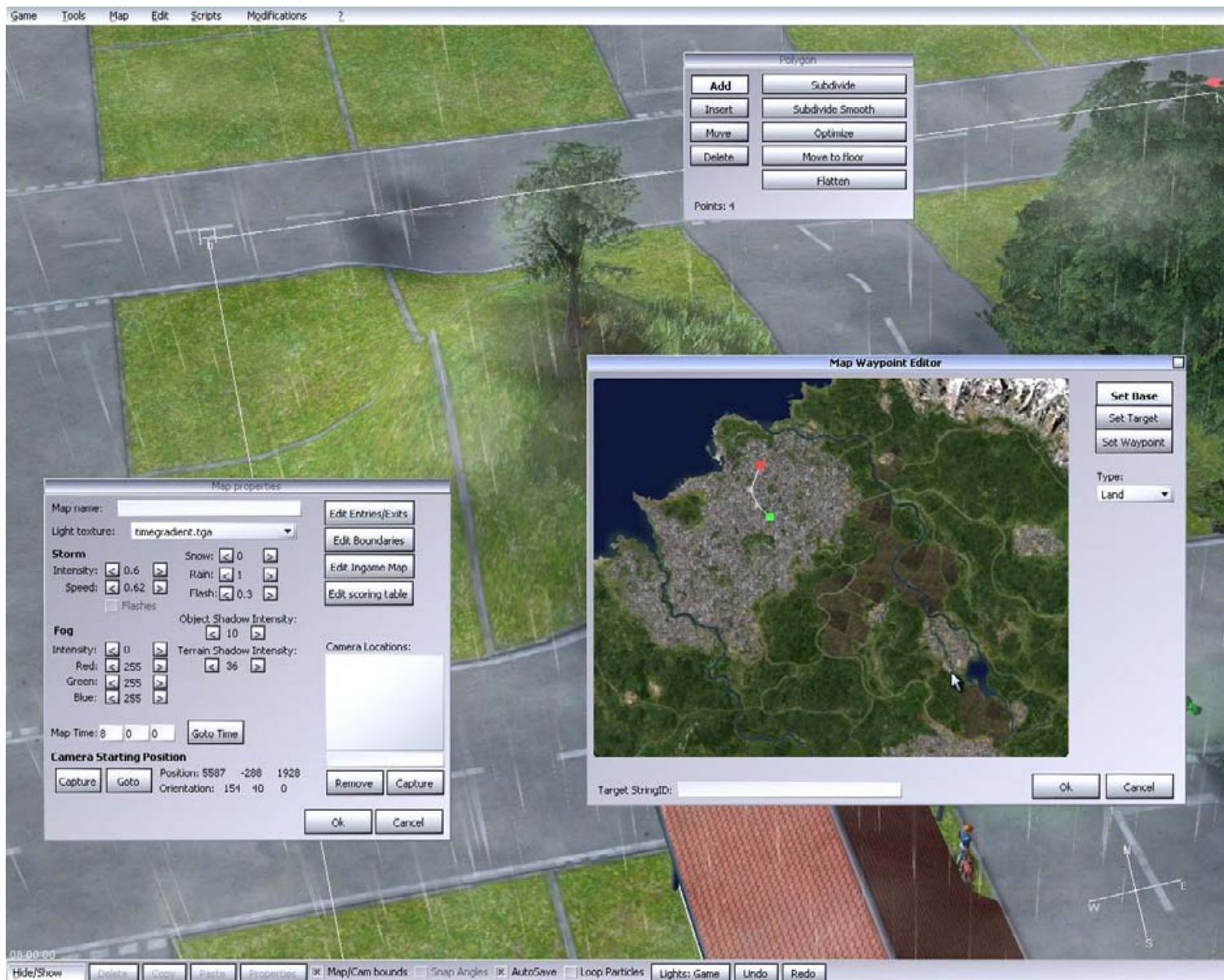


Další důležitou věcí je nastavení okrajů mapy. Je to velmi jednoduché. Klikněte na tlačítko **Edit Boundaries**. Objeví se známý dialog Polygon. Stejným způsobem jako jsme definovali, kreslili polygony dříve, nakreslete polygon kolem celé mapy. Tento polygon, v mém případě má jen čtyři vrcholy, je v mém případě čtverec :) Nyní se s kamerou nemohu pohybovat za tuto definovanou hranici. To je hlavně kvůli tomu, aby nebylo vidět, že tam dál už nic není! :) Až zavřete okno pro nastavení mapy, budou tyto hranice fungovat i v editoru. Pokud by vás omezovaly v další tvorbě, můžete je v editoru vypnout zrušením zaškrtnutí políčka **Map/Cam Bounds** na spodní liště editoru. Ale nedjřív musíme probrat zbylá dvě tlačítka :)

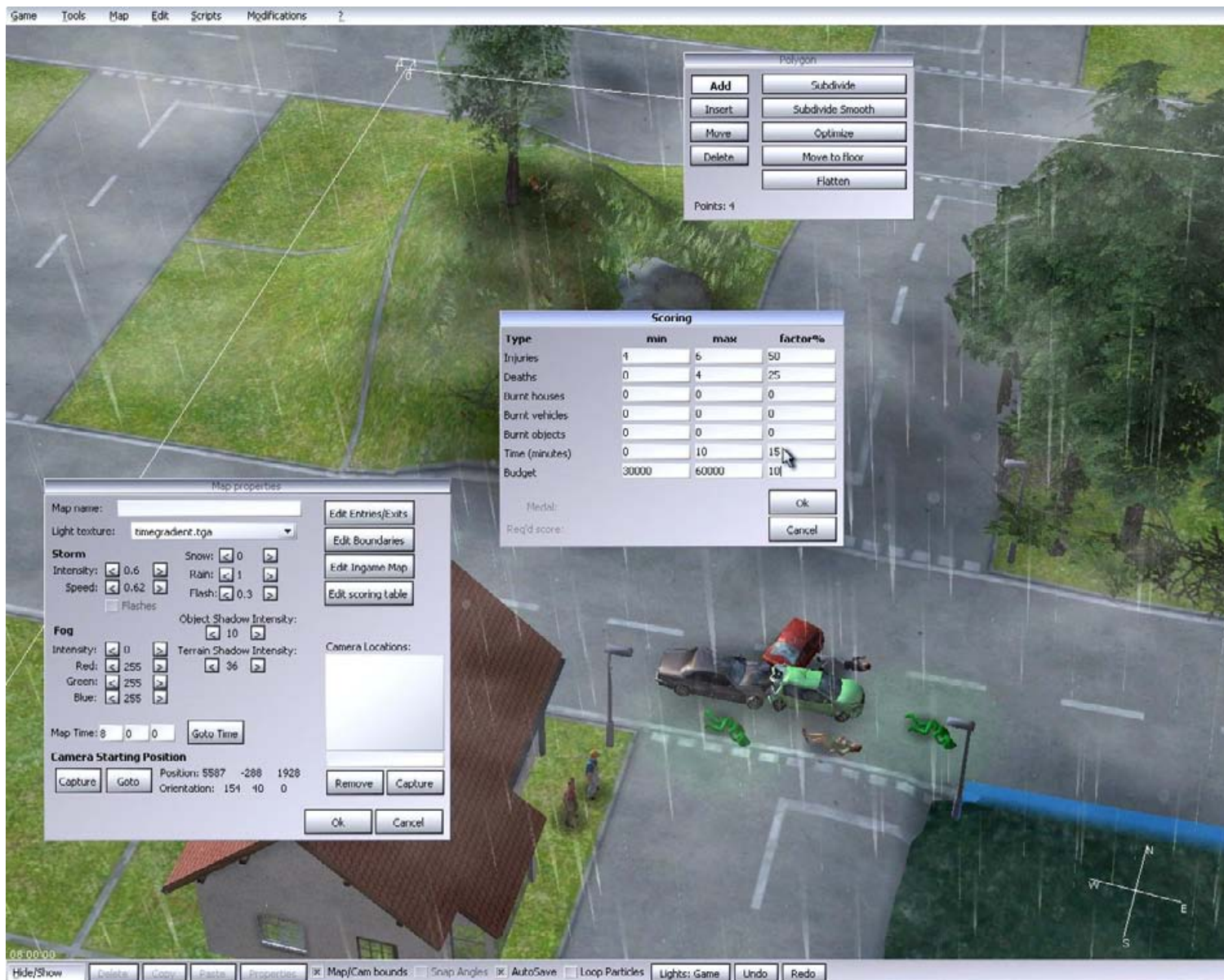


Pomocí tlačítka **Edit Ingame Map** nadefinujete pozici základny a místa katastrofy na té velké mapě, která je i ve hře. No je to vidět na obrázku :) Základnu umístíte pomocí tlačítka **Set Base**, bod cesty ke katastrofě tlačítkem **Set Waypoint** a místo katastrofy tlačítkem **Set Target**. Základna je zelený čtvereček a místo katastrofy červený.





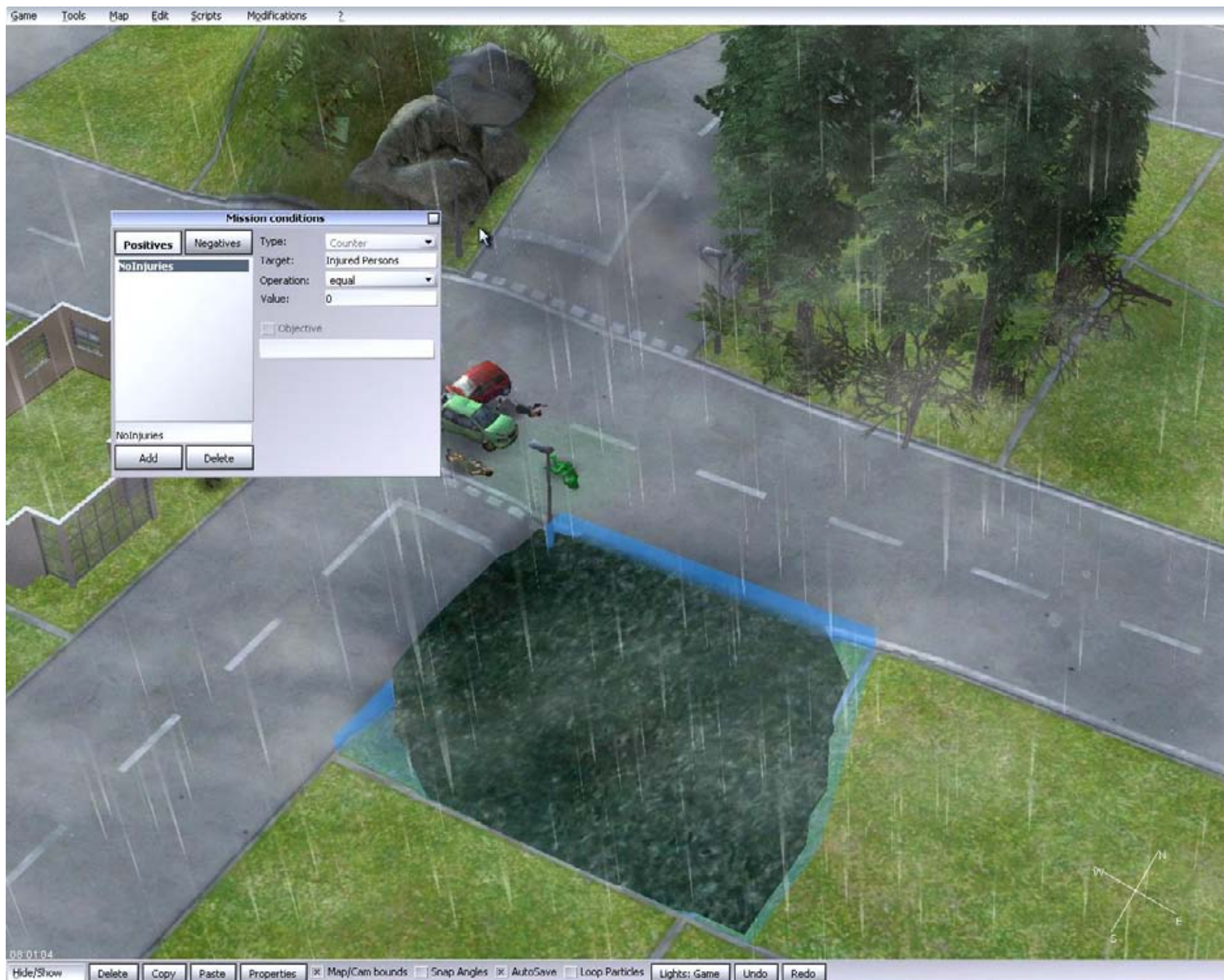
Posledním tlačítkem **Edit scoring table** nadefinujete způsob jakým se bude počítat výsledné skóre. Pro každou věc, za kterou se počítají nebo strhávají body, musíte nastavit minimální a maximální hodnotu a podíl jaký to má na celkovém hodnocení. To je poslední sloupec a v součtu by to mělo dávat 100% :). Tak například, v naší zkušební mapě jsou na začátku zraněné 4 osoby a 2 čumilové. Takže jsem nastavil minimální počet zraněných na 4 a maximální na 6, přičemž to má váhu 50% na celkovém hodnocení. To znamená, že pokud se během zásahu nikdo nezraní a všechny čtyři zraněné zachráníme, dostaneme za to právě 50% v hodnocení. Většinu hodnot házím od oka, ale jestli chcete počítat kolik vás může zásah nejmíň stát, kolik toho může minimálně a maximálně shořet, můžete :)



### Aby nám mise vůbec skončila:

Hra obsahuje několik počítadel (čítačů), které počítají různé věci, jako například počet zraněných lidí, počet mrtvých lidí, počet hořících objektů a mnoho dalšího. K těmto počítadlům se přistupuje pomocí názvu a my je použijeme k tomu, abychom hře řekli, kdy má mise skončit úspěchem, a kdy neúspěchem. To provedeme pomocí tlačítka **Edit->Mission conditions** nebo Ctrl+Shift+C. Okýnko obsahuje dva seznamy, mezi nimiž je možné přepínat tlačítka **Positives** a **Negatives**. Jak už napovídají názvy tlačítek, první seznam obsahuje podmínky pro splnění mise a druhý podmínky pro ukončení mise. Tlačítkem **Add** přidáme do seznamu novou podmínku. V políčku **type** nechte **Counter**. Do políčka **Target** vždy napište jméno počítadla, kterého se podmínka týká. Jejich seznam je v pdf návodu u hry, někde na konci v sekci Mission conditions. Do políčka **Value** pak napište hodnotu, s jakou má být hodnota počítadla srovnávána a v rozevřacím seznamu **Operation** zvolte, zda musí být hodnota čítače menší, rovna, větší než zadaná hodnota. Na ukázkovém obrázku je vidět nastavení podmínky pro nulový počet zraněných, tzn. že všichni zranění byli zachráněni. Dále jsem ještě nastavil negativní podmínku, aby mise skončila pokud zemřou více jak dva lidé. Ale na to už si otevřete editor a podívejte se jak to mám nastavené, pokud na to nepřijdete sami :) Nejsem si jistý zda tohle vůbec musíte nastavovat. Mise bez skriptu stejně neskončí. Některé původní mise nějaké podmínky nastavené mají, jiné vůbec, tak nevím. Asi jak se komu chce :?)





Tím jsme skončili v editoru. Můžete uložit mapu a vypnout editor. Aby jste si mohli misi vyzkoušet ve hře, budeme muset udělat ještě pár opatření.

## Vlastní kampaň

Bohužel ve hře není nějaký mód jako samostatné mise, takže každou novou misi, nebo mise musíme zařadit jako novou kampaň. Proto budeme muset vytvořit, nadefinovat novou kampaň. Kampaň je definovaná v souboru **campaign.xml** v adresáři **Specs**. Proto si i ve svém modu vytvořte tento adresář a v něm tento soubor. Stačí jen dokument z poznámkového bloku nebo jiného textového editoru uložit jako normální textový soubor a dát mu příponu xml. Pokud jsem to již neřikal dřív, tak hra hledá při načítání modu všechny soubory v adresáři modu a když je tam nenajde, načte je z původního, originálního umístění. Pokud tedy nenajde soubor campaign.xml v adresáři Specs naší modifikace, načte campaign.xml z adresáře Specs, který je ale v adresáři data a obsahuje definici původní kampaně, kde není naše mise :) To platí pro většinu souborů. Tím že vytvoříte stejné soubory v adresáři modu vlastně překryjete ty původní. No, nyní ale k té kampani. V souboru campaign.xml naší kampaně by jste měli mít toto:

```
<?xml version="1.0" encoding="UTF-16" ?>
<campaign>
  <mission name="MISSION01" index="1" order="0" budget="56000" map="mod:Maps/tutorial.e3m"
script="mod:Scripts/Game/Mission/murdock_tutorial.script" />
</campaign>
```

První řádek nás nemusí zajímat. Vlastně nás zajíma jen řádek se značkou *mission*. Parametr **name** určuje jak se bude mise jmenovat. **MISSION01** je název mise, tedy přesně řečeno řetězce v jednom ze souborů v adresáři s aktuálním jazykem. Ten nadefinujem za chvíli :) **Index** je jen identifikační číslo mise, ale nemá vliv na to zda je mise první či poslední v kampani. To určuje až atribut **order**, je roven 0 a to znamená že mise je v kampani první. Čísluje se od nuly, tzn. že druhá mise by měla order=1 :) **Budget** je množství peněz které by jste měli mít v misi k dispozici, ale myslím že hra se tím ani moc neřídí, pač jsem tam měl jinou hodnotu. **Map** určuje cestu k mapě, myslím že k tomu není třeba víc co dodávat. **mod:** v cestě už jsem vysvětloval někde na začátku tutoriálu :) Poslední, **script**, je cesta ke skriptu mise.

## Hlášky :) :

Před chvílí jste do souboru campaign.xml nadefinovali jméno mise MISSION01. Tahle konstanta, a jiné, musí být také někde nadefinovaná :) Pokud jste nezkoumali překlad, tak vám řeknu, že všechny hlášky ve hře jsou uloženy v adresáři **data/lang/cz**. Stejně adresáře si vytvoříme i v adresáři našeho modu. Můžeme je pojmenovat jinak a nemusí jich být ani tolik, ale abychom udrželi nějakou "normalizaci", necháme to tak. Do vytvořeného adresáře uložíme naše soubory s hláškama. Já jsem si vytvořil tři soubory: **murdock\_mission.xml**, **murdock\_objectives.xml** a **murdock\_fail.xml**. První soubor obsahuje název mise a ten úvodní text, co se zobrazí při načítání. Druhý soubor obsahuje úkoly pro misi a třetí hlášky které by se měly zobrazit pokud v misi neuspějete. Vypíšu tady jen první soubor, ostatní vypadají stejně, i když ne úplně. No, podívejte se do nich sami :)

### murdock\_mission.xml:

```
<?xml version="1.0" encoding="UTF-16" ?>
<strings>
  <string name="MISSION01">Ukázková mise</string>
  <string name="ID_01">Ukázková mise. Zachraňte všechny a všechno :).\nVytvořil Šílený Murdock
- silenymurdock@volny.cz</string>
</strings>
```

První řádek nás opět nemusí zajímat, ale v každém souboru musí být :) Značka **strings** značí začátek nějaké kolekce, seznamu řetězců, hlášek. Každá položka, řetězec, hláška se definuje ve značce **string**. Tato značka má jen parametr **name**, což je jméno řetězce, hlášky. Myslím že k tomu není třeba víc co dodávat.

## Máme vlastní soubory s hláškama :) :

Je fajn, že máme vlastní soubory s hláškama, ale dokud to nebude vědět hra, bude pořád používat ty ve svém adresáři *data/lang...a tak dále* :). Vratně se zpět do adresáře *Specs* (v adresáři našeho modu), kam jsem uložili soubor campaign.xml. Zkopírujte si sem z původního adresáře soubor **metastrings.xml** a upravte ho, aby jeho obsah byl následovný:

```
<?xml version="1.0" encoding="UTF-16" ?>
<meta>
  <file name="lang:commands.xml" />
  <file name="mod:Lang\cz\murdock_fail.xml" />
  <file name="lang:hints.xml" />
  <file name="mod:Lang\cz\murdock_missions.xml" />
  <file name="mod:Lang\cz\murdock_objectives.xml" />
  <file name="lang:strings.xml" />
  <file name="lang:talk.xml" />
  <file name="lang:mainmenu.xml" />
  <file name="lang:tutorial.xml" />
  <file name="lang:poi.xml" />
  <file name="lang:infotexts.xml" />
</meta>
```

Změny oproti originálnímu souboru jsou tučně :) Tak, pokud si pamatujete všechno a rozumíte všemu co jsem až doteď napsal a víte i něco navíc, pochopíte co jsme touto změnou udělali i bez vysvětlování :) Původní soubory **fail.xml**, **objectives.xml** a **missions.xml** jsme nahradili těmi našimi a zbytek jsme ponechali. Tím máme zajištěno, že v misi budou použiti naše hlášky, ale zároveň nepřijdeme o ostatní texty, jako informace o jednotkách a tak dále. Pokud by jste v souboru nechali jenom ty tři editované řádky, chyběly by vám ve hře všechny ostatní texty, čili vlastně celý překlad. Tedy chci říct doufám, pač jsem to nezkoušel :) Také by to mělo fungovat, pokud ty tři řádky prostě do souboru přidáte, mezi značky **meta**, samozřejmě a ty původní si tam necháte, čili nepřepíšete je :)

## Jednotky dostupné v misi:

V každé misi máte většinou přístupné jiné jednotky, tedy různé hasičské vozy a tak dále. To je samozřejmě taky někde napsané, co kde má a může být :) Budeme pracovat opět s určitým souborem z adresáře **Specs** a tentokrát jde o soubor **vehicles.xml**. Pro účely naší ukázkové mise vlastně soubor ani nemusíme překopírovávat do našeho adresáře, protože v něm nemusíme nic měnit, pač nám budou stačit jednotky, které jsou dostupné v první misi. Ale vysvětlím vám, jak to změnit pokud by jste chtěli :) Takže si ten soubor přeci jenom zkopírujte. V tomto souboru jsou nadefinovány všechny záchranářské jednotky, jaké mají modely, kdo v nich může jezdit, jaké vybavení převáží a také v které misi je potřeba :) V souboru jsou skupiny pro všechny sbory, pro záchranku, pro hasiče, polici a podpůrnou techniku. Skupina vždy začíná i končí značkou **group**. Jednotlivé jednotky, vozy, jsou definovány značkou **vehicle**. Ukážeme si jednu část souboru, definici pohotovostní ambulance:

```
<vehicle name="ID_NAME_NEF" prototype="mod:Prototypes/Vehicles/Ambulance/nef_body.e3p" space="1"
loadspace="0" requires="1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20" speed="120"
price="150">
  <interface id="ID_PARAMEDICS_NEF" image="base:UI/Game/vehicleimg/win_vehicleinfo_img_nef.dds"
bigimage="base:UI/Game/vehicleimg/win_vehicleinfo_img_nef.dds" trivia="ID_TRIVIA_NEF"
purpose="ID_PURPOSE_NEF" tooltip="ID_TOOLTIP_NEF" />
  <crew name="ID_NAME_DOCTOR" prototype="mod:Prototypes/Persons/Ambulance/doctorm.e3p"
alternative="mod:Prototypes/Persons/Ambulance/doctorf.e3p" space="1" requires="1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16 17 18 20" price="75" freeplayprice="75">
```



```

    <interface id="ID_DEFAULT" image="base:UI/Game/personimg/doctor_small" bigimage="base:UI/
Game/personimg/doctor_big.dds" trivia="ID_TRIVIA_DOCTOR" purpose="ID_PURPOSE_DOCTOR"
tooltip="ID_TOOLTIP_DOCTOR" />
    </crew>
</vehicle>

```

Zajímá nás zatím jen parametr **requires**, který jsem i zvýraznil. Obsahuje čísla misí, v kterých má být jednotka přístupná. V této ukázce je vidět, že jak vozidlo, tak jeho posádka je potřeba, je přístupná ve všech 20 originálních misích :) Ještě k významu čísel, 1 znamená první mise, 2 druhá a tak dále. Bude se to shodovat s parametrem **index** dané mise v souboru **campaign.xml**, který už známe. Tedy chci říct doufám, pokud ne, tak snad s parametrem **order** :)

### Skript:

Konečná fáze. Pak už by vám měla mise zcela fungovat :) Možná vás trochu zklamá, ale vysvětlím vám jen základní věci ohledně skriptu a ukážu jak přidávat úkoly a zjišťovat jestli jsou splněny. V naší ukázkové misi budeme hlídat jen jestli byly odvezeni všichni zranění, nic jiného. Pokud nebudete ve svojí misi chtít nic jiného, můžete skript použít jako šablonu. Ale jak už jsem psal v úvodu, alespoň bez základní znalosti programování se neobejdete a pokud programování aspoň trochu rozumíte, vykoukáte a opíšete spoustu věcí ze skriptů originálních misí. Tady je tedy skript pro naši misi:

```

//konstanta, ktera urcuje maximalni pocet lidi kteri muzou v misi zemrit
const int MAX_DEAD_PERSONS = 3;

/*Objekt TutorialMission rozsiruje objekt skript mise. Proste je to objekt
ktery obsahuje nekolik funkci. Nektere z nich maji specialni vyznam a spusti se v dany cas
nebo pokud nastane dana udalost. Muzete ale take definovat vlastni funkce, ktere budou vykonavat
potrebne doplujici veci
*/
object TutorialMission : MissionScript
{
    /*Konstruktor. Spusti se uplne jako prvni pri vytvareni tohoto objektu, tedy by se snad
dalo rict ze pri vytvoreni skriptu
*/
    TutorialMission()
    {
    }

    /*Tato funkce se spusti pri startu mise. Takze se do ni hodi napsat nejaky
pocatecni kod, ktery vse nastavi tak jak by to melo na zacatku byt
*/
    void Start()
    {
        //spustime soundtrack. Soundtrack je definovan v souboru Audio/Soundtracks/
soundtracks.xml
        //kadzdy soundtrack obsahuje nekolik "skladeb", kazdou pro ruznou uroven akce, napeti
        //to urcuje parametr level
        //Uroven urcuje druhy parametr funkce
        Audio::PlaySoundtrack("1", 0.0f);
        //pridani ukolu ktery ma hrac splnit. Text je definovan v souboru murdock_objectives.xml
        Mission::AddObjective("M01_SAVE_ALL");
    }

    /*Tato funkce se spusti pokud uplyne doba nejakeho casovace, citace.
casovac jeho doba uplynula pozname podle jeho jmena, coz je prvni parametr funkce.
Druhy jsem zatim nevyuzil, ale hadam ze to je aktualni cas v casovaci
*/
    void OnTimer(const char *Timer, float Time)
    {
    }

    /*Tato funkce se hadam spusti pokazde nez zacne nejaka jednotka vykonavat nejaky hracuv
prikaz*/
    ActionCallbackResult OnPreAction(const char *Action, ActionCallback* Data)
    {
    }

    /*Nevim presne k cemuz je tahle funkce. Snad se spusti pokud nejaky objekt opusti misi, auto,
zachranari nebo tak*/
    void OnMissionLeft(GameObject *Obj)
    {
    }
}

```

```

}

/*Tahle funkce se spusti pokud se spusti nejaky spinac. Prvni parametr udava jmeno spinace a
druhy kdo ho spustil*/
void OnTrigger(const char *Trigger, Actor *Collider)
{
}

/*Tato funkce se spusti pokud hrac zada nejake jednotce nejaky prikaz*/
ActionCallbackResult OnPostAction(const char *Action, ActionCallback* Data)
{
}

/*Tato funkce se spusti pokud hrac zrusi akci kterou provadi nejaka jednotka*/
ActionCallbackResult OnAbortAction(const char *Action, ActionCallback* Data)
{
}

/*Tato funkce je dulzita. Ta rozhoduje o tom, jestli mise probiha, byla neuspesna nebo je
splnena
Tato mise se vola velmi casto!*/
MissionState GetMissionState()
{
    //pokud pocet mrtvych prekroci limit, tak je mise neuspesna
    //pocet mrtvych pocita citac, který jsme nastavili v editoru
    if (Mission::GetCounter("Person deaths") > MAX_DEAD_PERSONS)
        return MISSION_FAILED;
    //Pokud byly splneny vsechny ukoly, mise skončila uspesne
    if (Mission::AllObjectivesAccomplished())
    {
        return MISSION_SUCCEEDED;
    }
    //pokud je pocet zranenych nula, byli vsichni zraneni nebo mrtvi odvezeni,
    //nastav ze byl splnen ukol, ale to ma cenu jen pokud uz nebyl splnen, coz
    //nejdriv zkontrolujeme
    if (Mission::GetCounter("Injured Persons") == 0)
    {
        //pokud ukol jeste nebyl splnen, tak nastav ze ted splnen je
        if (!Mission::IsObjectiveAccomplished("M01_SAVE_ALL"))
            Mission::SetObjectiveAccomplished("M01_SAVE_ALL", true);
    }
    //jinak, pokud jeste nebyli vsichni zraneni nebo mrtvi odbvezeni z mapy, nastav
    //ze tento ukol jeste nebyl splnen. Coz ma cenu, pokud byl splnen.
    else
    {
        //pokud ukol byl splnen, nastav ze uz to neplati a ze splnen neni
        if (Mission::IsObjectiveAccomplished("M01_SAVE_ALL"))
            Mission::SetObjectiveAccomplished("M01_SAVE_ALL", false);
    }
    //pokud se dostaneme sem, znamena to, ze mise porad pokracuje
    return MISSION_RUNNING;
}

/*Taktez velmi dulezita funkce. Spusti se pri ukladani pozice*/
bool SerializeTo(ScriptSerializer *Serializer)
{
    /*Pokud si ve skriptu definujete nejake promenne, ktere pouzivate k ruznym ucelum
je nutne je pri ukladani pozice take uložit, aby pri nahrani pozice vsechno bylo v tom
stavu
jako pred uložením. Konstanty ukladat nemusíte. Nasledující řádky jsou jen demonstrační,
Slouží
k uložení verze:
const int Version = 0x0100;
Serializer->Write(Version);*/

    return true;
}

/*Tato mise se spusti naopak při načítání uložené pozice*/
bool SerializeFrom(ScriptSerializer *Serializer)

```



```

{
    /*To co jste si pomoci predchozi funkce ulozili si tady musite zase precist, nacist
    Nacitejte ve stejnem poradi jako jste ukladali, jinak to nactete spatne a mise se pak bude
    chovat jinak nez chcete!
    Demonstrace nacteni verze:
    int Version;
    Serializer->Read(Version);*/

    return true;
}

/*Pokud hrac v misi selze, tato funkce vrati duvod, proc se tak stalo*/
const char *GetFailReason()
{
    //Pokud zemrelo vice lidi nez bylo povoleno, vrat retezec kde je to ulozeno.
    //Hlasku TOO_MANY_DEADS jsem definoval v souboru murdock_fail.xml
    if (Mission:GetCounter("Person deaths") >= MAX_DEAD_PERSONS)
        return "TOO_MANY_DEADS";

    //jiny pripad by nemel nastat, ale pro jistotu pokud se stane neco co jsme necekali, tak
    //vrat retezec informujici o tom ze vlastne ani my sami nevime co se stalo, ale ze je
    nutne hrat
    //misi znova :)
    return "UNKNOWN";
}

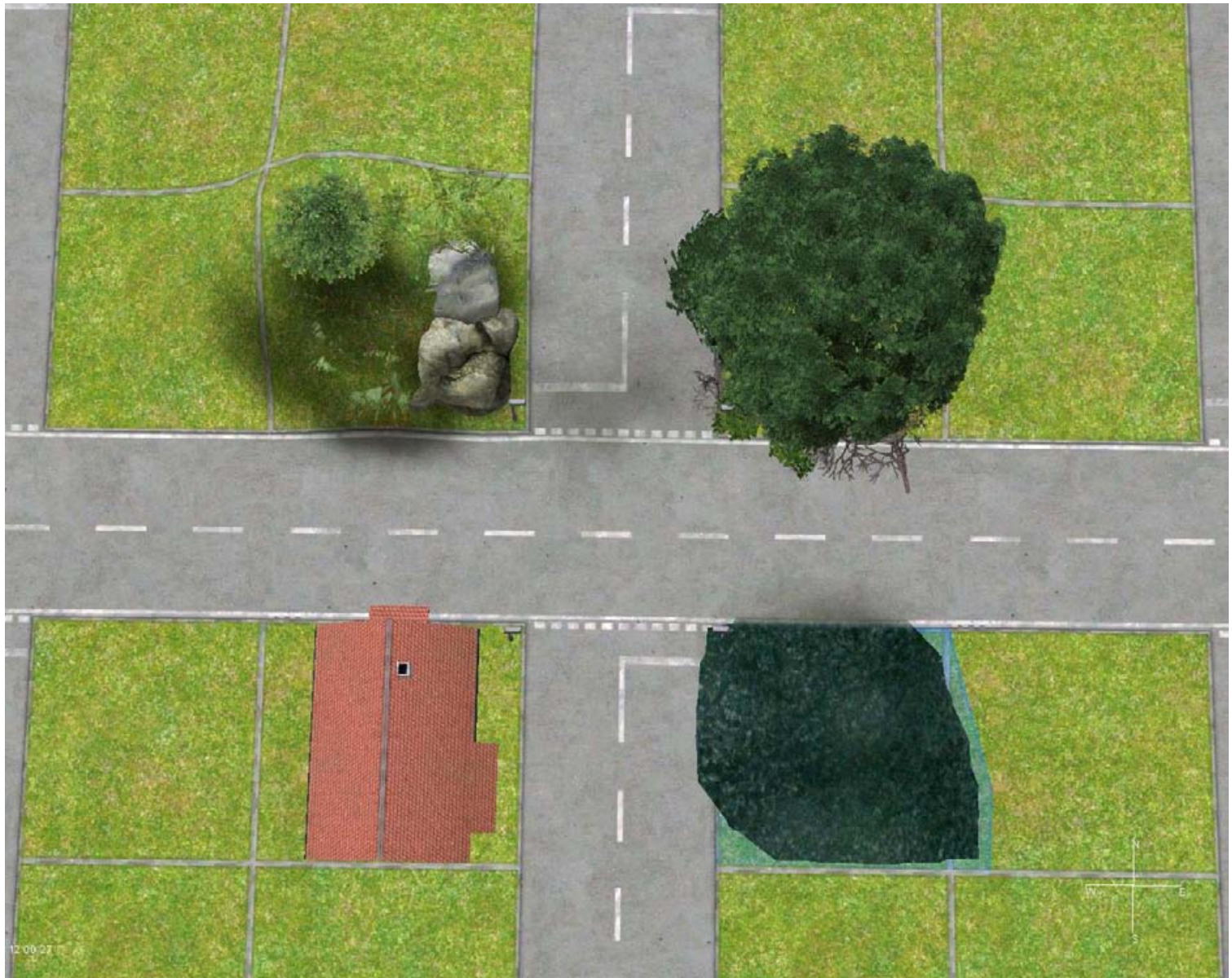
/*Tato funkce se spusti pri prijezdu zachranarske jednotky na misto katastrofy*/
void OnSquadVehicleArrived(Vehicle *Vehicle_)
{
}
};

```

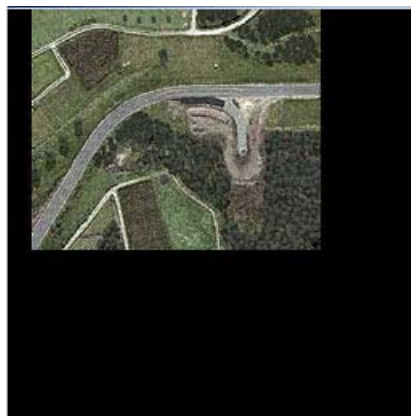
Víc vám toho o skriptování opravdu neřeknu, protože to by bylo na celou knihu a na to nemám čas :( Pokud se o tom chcete dozvědět víc a tvořit více nápadité mise, musíte holt prolézat původní mise a zkoumat jejich skripty a jak je to propojené s misí v editoru. Bohužel, herní vývojáři nebo distributoři, nevím kdo to vždycky propaguje jak bude mít jejich hra editor misí a jak bude perfektně rozšiřitelná a pak ke hře dostaneme žádný nebo neúplný návod. Prostě takový, aby jste nebyli schopni se to naučit a tvořit stejně dobré mise nebo většinou lepší než autoři. To samé je s Emergency 3, nikde jsem nenašel seznam funkcí co bych ve skriptu mohl používat, to si je pak můžu už jen vycucát z prstu a pak teprve psát skripty. A nebo používat jen to co můžu opsát z původních skriptů :(

### Satelitní obraz mapy:

To je obrázek mapy v tom malém okénku ve hře, kde můžete sledovat rozmístění všech vašich jednotek. V pdf návodu je docela složitý postup jak tenhle obrázek vytvořit. Faktem ale je, že k tomu budete potřebovat nejspíš Photoshop a plugin pro zpracování dds souborů. Spustíte editor a načtete svoji mapu. Dole na liště je menu **Hide/Show**. Pomocí něj schovete všechny položky co jsou v menu jako počasí, auta, pand'uláky atd. Nechte jen objekty. Prostě aby to vypadalo neobydleně :) Potom stiskněte klávesu **Shift + Enter**. Tím nastavíte čas na 12:00. Potom stiskněte klávesu **Pause**, čímž se kamera přepne do módu, kdy s ní můžete libovolně pohybovat. Pomocí kurzorových kláves a myši nastavte kameru tak, aby jste viděli mapu celou z ptačí perspektivy. Nějak takhle:

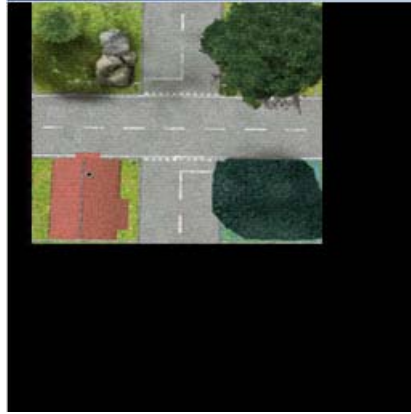


Udějte screenshot. Buď pomocí klávesy **Print Screen (Prt Scrn)** a otevřete v nějakém editovacím programu, nejlépe ve Photoshopu tenhle screenshot. Otevřete ještě nějaký satelitní obrázek mapy z nějaké původní mise. Ten je vždy uložen ve stejném adresáři jako mise a má stejné jméno jako mise, jen má příponu dds. Třeba pro první misi je to soubor **m01.dds**. Ten vypadá takhle:



Z vašeho screenshotu vykopírujte oblast, která zachycuje vaši mapu. Touto vykopírovanou oblastí překryjte tu v otevřeném satelitním obrázku. Nejlépe to určitě pochopíte z dalšího obrázku:





Obrázek ještě upravte pomocí filtrů, jestli chcete :) a uložte ho do adresáře, kde je uložena vaše mise. Tedy soubor .e3m a pojmenujte ho stejně jako se jmenuje mise. V mém případě to tedy bude tutorial.dds.

### **A máme to :) :**

Ted' bychom měli mít celou misi kompletní. Pokud vám to nefunguje, tak jak má, je někde chyba :)

Šílený Murdock ([silenymurdock@volny.cz](mailto:silenymurdock@volny.cz))