



Tworzenie własnych poziomów w Colobot v.1.5.1

1. Instalacja własnych poziomów

Instalacja i /lub kreacja własnych poziomów wymaga posiadania COLOBOTA w wersji 1.7 (lub wyższej). W przypadku konieczności uaktualnienia posiadanej wersji COLOBOTA, należy ściągnąć patch ze strony: www.colobot.com.

Poziomy stworzone przez użytkowników dostępne są po kliknięciu na przycisku: **"Poziomy"**, znajdującym się w Menu Głównym:



Wszystkie poziomy stworzone przez użytkowników znajdują się w katalogu: user\, znajdującym się w katalogu instalacyjnym Colobota. Domyślna lokalizacja to :

- C:\Program Files\Colobot\user\

Jeśli folder: user\ nie istnieje, stwórz go.

Poziomy stworzone przez użytkowników mogą zawierać jedną lub wiele misji. Każdy poziom stworzony przez użytkownika odpowiada jednemu wersowi, w menu po lewej stronie:



Poziom stworzony przez użytkownika, fizycznie jest folderem zawierającym szereg plików. W powyższym przykładzie, poziom: „Sample #1” znajduje się w folderze:

- C:\Program Files\Colobot\user\sample01\

Zwróć uwagę na to, iż nazwa poziomu wyświetlająca się w menu po lewej stronie jest inna niż nazwa folderu. Dzieje się tak, ponieważ nazwa ćwiczenia znajduje się w pliku: sample01\scene00.txt. Definiuje ją komenda:

- Title.E text="Sample #1" resume="Sample"

Nazwy misji widoczne po prawej stronie, odpowiadają kolejnym plikom: sample01\sceneNN.txt :

Plik	Tytuł
sample01\scene01.txt	Alert
sample01\scene02.txt	Mc Gywer

Nazwy plików z misjami muszą być oznaczone kolejnymi numerami. Załóżmy, iż istnieje następująca struktura plików:

- scene00.txt
- scene01.txt
- scene02.txt
- scene05.txt

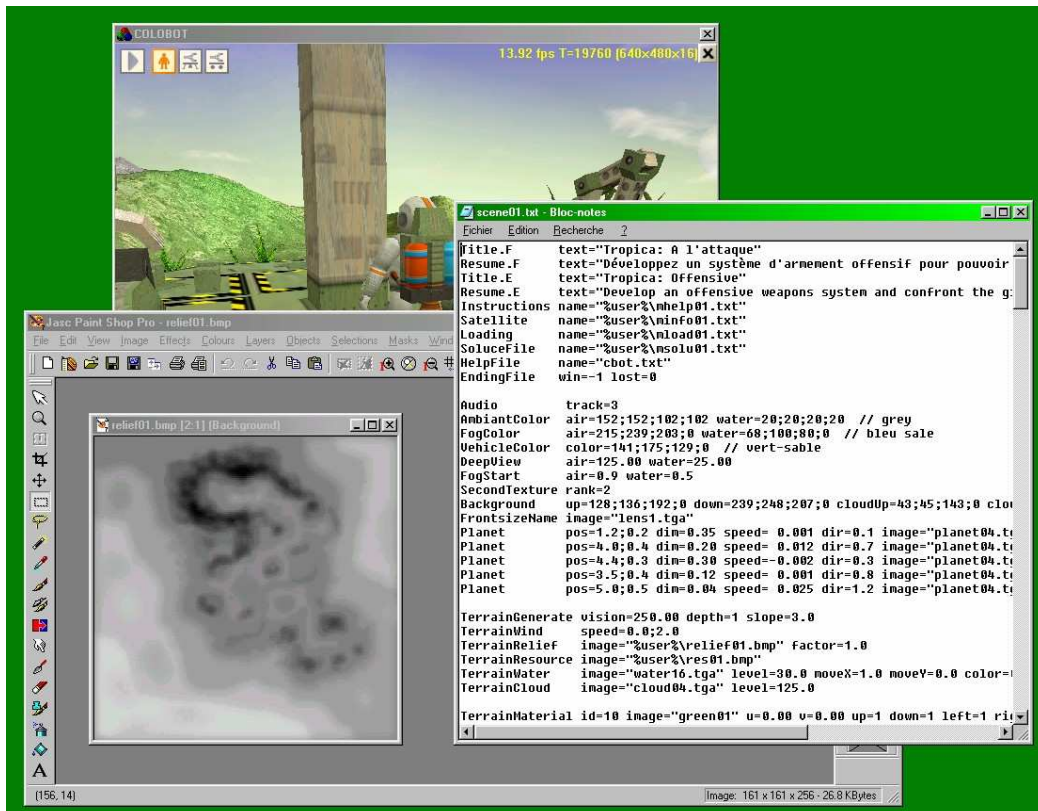
W przypadku takim dostępne będą tylko dwie misje, o numerach 01 i 02. Misja numer 05 nie pojawi się na liście, gdyż brakuje misji 03 i 04.

2. Tworzenie poziomów a właściwy warsztat pracy

Podczas tworzenia własnych poziomów często niezbędnym jest korzystanie z programów innych niż COLOBOT. Generalnie program można zwinąć do paska zadań, za pomocą kombinacji ALT+TAB. Niemniej bardziej wygodnym rozwiązaniem będzie uruchomienie COLOBOTA w oknie. Aby uruchomić COLOBOTA w oknie, kliknij na „Opcje”, a następnie na zakładkę : „Urządzenie”.



Po wykonaniu tych czynności odznacz opcję: "Pełny ekran" i kliknij na przycisku: "Zastosuj zmiany". Pamiętaj o tym, iż COLOBOT uruchomiony w oknie pracuje zawsze w rozdzielczości 640x480 pikseli.



3. Tworzenie własnych poziomów

Plik: `sample01.zip` zawiera dwie przykładowe misje. Rozpakuj zawartość tego pliku do katalogu: `user\`, który znajduje się w katalogu instalacyjnym COLOBOTA. Domyślna lokalizacja to:

- `C:\Program Files\Colobot\user\sample01\`

Jeśli folder: `user\` nie istnieje, stwórz go.



Powyższe obrazki pochodzą z przykładowych misji znajdujących się w pliku: `sample01.zip`.

Misje te będą przewijać się przez całą tą instrukcję. Folder: user\sample01\ zawiera następujące, 24 pliki :

Plik	Zawartość	
	mhhelp01.txt	
	Instrukcje dla misji #1	
mhhelp02.txt	Instrukcje dla misji #2	
minfo01.txt	Informacja satelitarna dla misji #1	Teksty widoczne w SatCom
minfo02.txt	Informacja satelitarna dla misji #2	
mload01.txt	Wyjaśnienia programów wysyłanych przez Houston dla misji #1	
mload02.txt	Wyjaśnienia programów wysyłanych przez Houston dla misji #2	
msolu01.txt	Solucja (Rozwiązanie dla misji #1	
msolu02.txt	Solucja (Rozwiązanie dla misji #2	
sant01.txt	Skrypt startowy ładowany bezpośrednio do konkretnych mrówek	Skrypty
skill.txt	Skrypt startowy ładowany do konkretnych Dział lub Dział Organicznych.	
scene00.txt	Nazwa poziomu	Poziomy
scene01.txt	Opis misji #1	
scene02.txt	Opis misji #2	
back01.bmp	Gwiazdy dla misji #1	Obrazy
cloud02.bmp	Chmury dla misji #2	
lens02.bmp	Refleksy słoneczne dla misji #2	
nevada1.bmp	Podłoże dla misji #2 (piasek-skały1)	
nevada2.bmp	Podłoże dla misji #2 (skały1-skały2)	
relief01.bmp	Rzeźba terenu dla misji #1	
relief02.bmp	Rzeźba terenu dla misji #2	
res01.bmp	Surowce dla misji #1	
res02.bmp	Surowce dla misji #2	
terra01.bmp	Tekstura podłoża dla misji #1	
water01.bmp	Tekstura wody dla misji #1	

Wszystkie pliki tekstowe (.txt) mogą być kreowane i modyfikowane za pomocą edytorów tekstowych (np. Notatnika). Pliki obrazów (.bmp) mogą być modyfikowane za pomocą edytorów grafiki (np. Paint, GIMP, PaintShop, PhotoShop, czy PhotoPaint).

Plik: user\sample01\scene01.txt zawiera opis pierwszej misji :

Komenda	Użyte pliki
Title.E text="Alert"	
Resume.E text="Your base is under alert ..."	
Instructions name="%user%\mhhelp01.txt"	Instrukcje dla SatCom
Satellite name="%user%\minfo01.txt"	Raport satelitarny dla SatCom
Loading name="%user%\mload01.txt"	Programy wysyłane przez Houston dla SatCom
SoluceFile name="%user%\msolu01.txt"	Solucja dla SatCom
HelpFile name="cbot.txt"	Oficjalna pomoc programistyczna dla SatCom
...	
Background image="%user%\back01.bmp"	Gwiazdy widoczne na niebie
FrontsizeName image="lens1.tga"	Promienie słoneczne
...	
TerrainRelief image="%user%\relief01.bmp"	Rzeźba terenu
TerrainResource image="%user%\res01.bmp"	Surowce
TerrainWater image="%user%\water01.bmp"	Tekstura wody
...	

TerrainInitTextures image="%user%\terra.bmp" dx=1 dy=1 table=1	Tekstura podłoża
...	
CreateObject pos=12.50;-112.50 dir=1.0 type=WingedOrgaShooter power=0.2 script1="%user%\skill.txt"	Skrypt startowy ładowany do danego robota
...	


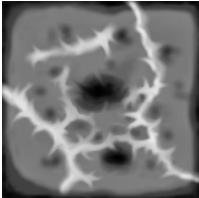
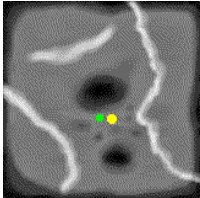

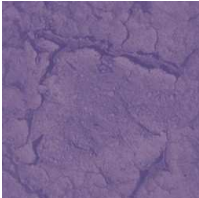
Niektóre z komend znajdujących się w pliku: user\sample01\scene01.txt wiążą go z innymi plikami. Przykład:

- `Instructions name="%user%\mhelp01.txt"`

Powyższa komenda wskazuje komendy, które mają być wyświetlone w SatCom, by wyjaśnić co musi być zrobione w danej misji.

Jeśli nazwa pliku poprzedzona jest frazą: %user%, to zostanie on zapożyczony z folderu misji (w tym przykładzie będzie to lokalacja: user\sample01\).

Oto obrazki znajdujące się w katalogu: user\sample01\, wchodzące w skład pierwszej misji :

				
back01.bmp	relief01.bmp	res01.bmp	water01.bmp	terra001.bmp
512x347	161x161	161x161	256x256	256x256
16 milionów kolorów	256 poziomów szarości	256 kolorów	16 milionów kolorów	16 milionów kolorów



3.1. Pliki standardowe

Wszystkie pliki wchodzące w skład danego ćwiczenia muszą znajdować się w tym samym katalogu. W przypadku misji przykładowej będzie to katalog: `user\sample01\`. Możliwe jest też tworzenie odwołań do plików standardowych COLOBOTA, znajdujących się w folderach standardowych COLOBOTA. Dzięki temu nie trzeba powtarzać dziesiątki razy tych samych komend. Przykład::

- `HelpFile name="cbot.txt"`

Powyższa fraza oznacza, iż plik: `cbot.txt` znajduje się w standardowym katalogu pomocy Colobota:

- `C:\Program Files\Colobot\help\cbot.txt`

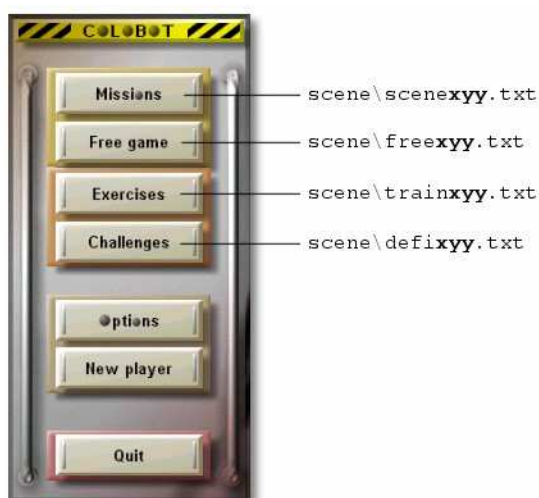
W normalnym wypadku plik musi być poprzedzony frazą: `%user%\`, by wskazać na konkretny folder misji. Plik pozbawiony tego prefiksu, otwierany jest z jednego ze standardowych folderów Colobota:

Folder standardowy	Zawartość
<code>scene\</code>	Opisy ćwiczeń i misji
<code>help\</code>	Instrukcje wyświetlane w SatCom .
<code>diagram\</code>	Obrazy wyświetlane w SatCom .
<code>textures\</code>	Obrazy rzeźby terenu i surowce.
<code>script\</code>	Programy startowe CBOT ładowane do robotów.

Możesz tworzyć referencje (powiązania) do plików znajdujących się w tych folderach lub kopiować je do katalogu ćwiczeniowego i tam modyfikować je pod kątem danego ćwiczenia. **Nie wolno ci jednak modyfikować plików znajdujących się w folderach standardowych Colobota!** Mogłoby to doprowadzić do uszkodzenia całej gry oraz do sytuacji, w której Twoje poziomy nie działałyby na innych komputerach.

Wszystkie standardowe pliki dla ćwiczeń i zadań, znajdują się w folderze: `scene\` :

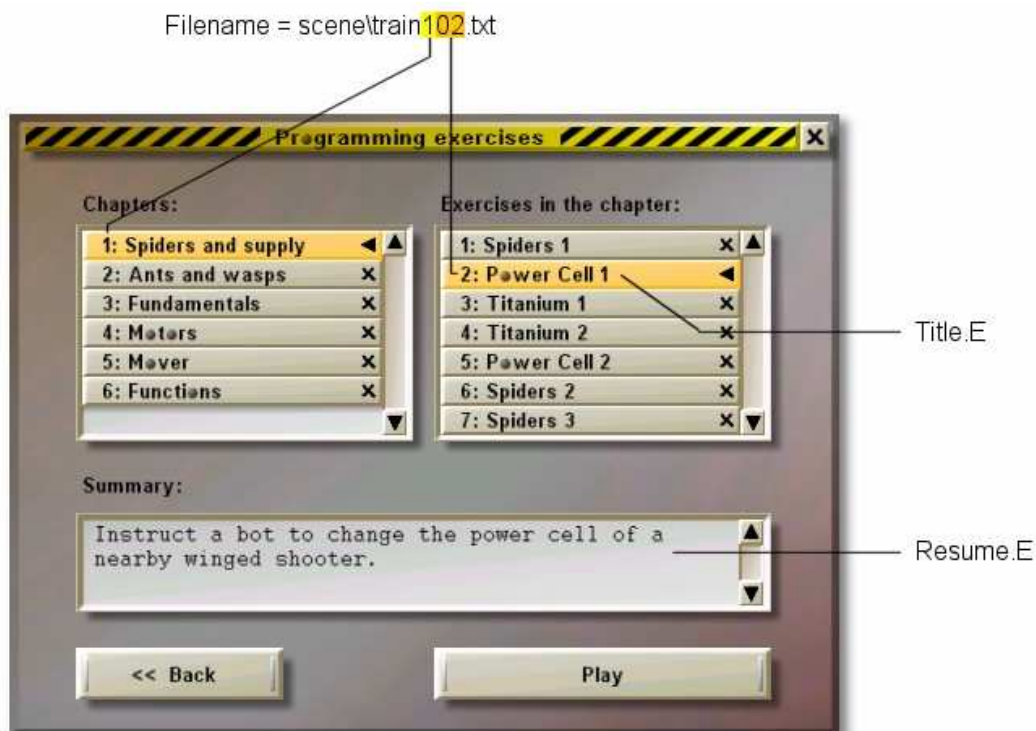
<code>scene\scenexyy.txt</code>	Misje
<code>scene\freexyy.txt</code>	Swobodne gry
<code>scene\trainxyy.txt</code>	Ćwiczenia
<code>scene\defixyy.txt</code>	Wyzwania



Trzycyfrowa liczba składa się z:

x	Numeru rozdziału	Od 1 do 9
yy	Numeru ćwiczenia	Od 01 do 99

Na przykład: train102.txt oznacza plik przynależny do drugiego ćwiczenia, pierwszego rozdziału.



3.2. Opis misji i/lub ćwiczenia

Opis misji determinuje rzeźbę terenu, tekstury, pozycję startową różnych robotów, surowce, roślinność itd. Znajduje się on w pliku:

user\sample01\sceneNN.txt. Każdy wers pliku opisującego misję zaczyna się od komendy, która składa się z hasła i parametrów. Puste wersy są ignorowane. Znak podwójnego slash: // oznacza początek komentarza, który kończy się z końcem wersu. Colobot jest bardzo wrażliwy na poprawność wprowadzanych komend – nawet wielkość liter jest istotna. Należy zwracać szczególną uwagę na poprawność ich treści i na to czy np. pomiędzy treścią, a znakiem „=” nie znajdują się spacje.

Poprawna formuła :

- Title.E text="Alert"

Niepoprawna formuła :

- title.E text = "Alert" // Nie zachowana właściwa wielkość znaków, spacje przed i po znaku: "=".

3.2.1. Nagłówek

Nagłówek determinuje nazwę ćwiczenia i/lub misji i nazwy plików z wyjaśnieniami.

```
Title.E text="Alert"
```

Jest to krótka nazwa misji. Pojawia się ona w menu, na liście po prawej stronie.

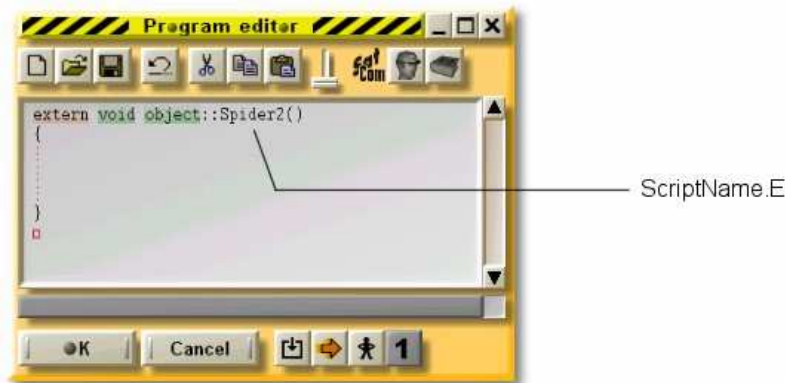
```
Resume.E text="Your base is under alert ..."
```

Krótki opis misji. Pojawia się on w oknie pomiędzy dwoma listami.

Uwaga : Komendy Title.P i Resume.P umożliwiają wprowadzenie tekstów, które będą widoczne w polskiej wersji COLOBOTA.

```
ScriptName.E text="Spider2"
```

Jest to domyślna nazwa, nadawana nowym programom.



```
Instructions name="%user%\mhelp01.txt "
```

Jest to nazwa pliku, który zawiera instrukcje dla danej misji. Będą one widoczne w **SatCom**.

Wszystkie pliki instrukcji powinny rozpoczynać się od frazy: "mhelp".

```
Satellite name="%user%\minfo01.txt "
```

Jest to nazwa pliku zawierających raport satelitarny. Będzie on widoczny w **SatCom**. Wszystkie pliki raportów satelitarnych powinny rozpoczynać się od frazy: "minfo".

```
Loading name="%user%\mload01.txt "
```

Jest to nazwa pliku zawierającego programy wysyłane przez Houston. Będą one widoczne w **SatCom**.

Wszystkie programy wysyłane przez Houston powinny rozpoczynać się od frazy: "mload".

```
SoluFile name="%user%\msolu01.txt "
```

Jest to nazwa pliku zawierającego rozwiązanie dla danej misji. Będzie ona widoczna w **SatCom**.

Wszystkie pliki rozwiązań powinny rozpoczynać się od frazy: "msolu".

```
HelpFile name="cbot.txt "
```

Jest to nazwa pliku zawierającego instrukcje programistyczne. Wyświetlają się one po naciśnięciu klawisza F2. Plik standardowy: cbot.txt znajduje się w katalogu help\. Zwykle wszystkie misje mają odwołania do tych samych instrukcji, zawartych w pliku: cbot.txt.



```
EndingFile win=-1 lost=0
```

Jest to animacja, która ma być użyta po zakończeniu misji (pomyślnym lub też i niepomyślnym):

- win=-1: .brak animacji po zakończeniu ćwiczenia. Misja po prostu kończy się po starcie statku.
- win=2: scene\win002.txt będzie użyta.
- win=123: scene\win123.txt będzie użyta.
- lost=0: scene\lost000.txt będzie użyta.

Nie możesz umieścić opisu tych animacji w folderze własnego poziomu. Musisz skorzystać ze standardowych plików: winxxx.txt, zlokalizowanych w katalogu: scene\. Pliki te nie mogą być modyfikowane.

```
MessageDelay factor=5
```

Fraza ta umożliwia zwiększenie czasu, przez jaki będą widoczne wiadomości. Wiadomości wyświetlają się w górnej części ekranu.

3.2.2. Atmosfera

Część ta opisuje kolory, mgłę, niebo itp., a także muzykę odtwarzaną w tle.

```
Audio track=0
```

Określisz tu numer ścieżki audio, która ma być odtwarzana podczas danej misji. Oto lista ścieżek:

Numer ścieżki	Planeta
0	Księżyc (cisza)
2	Ziemia
3	Tropika
4	Krystalia
5	Saari
6	Wulkania
7	Centuria
8	Orfenia
9	Nowa Ziemia

```
AmbiantColor air=102;102;102;102 water=20;20;20;20
```

Jest to kolor biernych źródeł światła dla nieba i obszarów podwodnych.

Na kolor składają się 4 komponenty: red (czerwień)/green (zieleń)/blue (niebieski)/alpha. Wartości tych barw oscylują pomiędzy 0 (brak), a 255 (maksymalne natężenie koloru). Komponent alpha jest ignorowany. Przykład :

- color=175;209;215;0 // piaskowo-niebieski

Planeta	Powietrze	Woda
Ziemia	120;90;0;0	20;20;20;20
Księżyc	102;102;102;102	20;20;20;20
Tropika	152;152;102;102	20;20;20;20
Krystalia	102;102;102;102	20;20;20;20
Saari	136;136;102;102	20;20;20;20
Wulkania	102;102;102;102	20;20;20;20
Centuria	102;102;102;102	20;20;20;20
Orfenia	102;68;68;102	20;20;20;20
Nowa Ziemia	102;102;102;102	20;20;20;20





FogColor air=180;222;255;0 water=10;20;100;0

Jest to kolor obiektów znajdujących się w oddali. Zobacz też DeepView i FogStart.

Planeta	Powietrze	Woda
Ziemia	100;100;90;0	67;80;100;0
Księżyc	0;0;0;0	67;80;100;0
Tropika	215;239;203;0	68;100;80;0
Krystalia	180;222;255;0	10;20;100;0
Saari	254;245;146;0	67;80;100;0
Wulkania	205;86;21;0	200;100;0;0
Centuria	176;176;181;0	10;100;20;0
Orfenia	50;30;10;0	67;80;100;0
Nowa Ziemia	208;200;223;0	94;153;180;0

VehicleColor color=175;209;215;0

Jest to kolor robotów i budynków.

			
208;206;196;0	141;175;129;0	234;129;26;0	158;120;82;0
Ziemia	Tropika	Wulkania	Orfenia

GreeneryColor color=250;187;69;0

Jest to kolor roślinności.

InsectColor color=189;171;51;0

Jest to kolor insektów.

Planeta	Kolor pojazdów	Kolor roślinności	Kolor insektów
Ziemia	168;158;118;0	161;151;41;0	
Księżyc	208;206;196;0		
Tropika	141;175;129;0		
Krystalia	175;209;215;0		
Saari	158;143;68;0		
Wulkania	234;129;26;0	250;187;69;0	
Centuria	117;158;64;0		
Orfenia	158;120;82;0		189;171;51;0
Nowa Ziemia	200;196;174;0		

```
DeepView air=100.00 water=25.00
```

Jest to maksymalne pole widzenia. Wartość ta wyrażona jest w metrach. Żaden z obiektów znajdujących się poza „polem widzenia” nie będzie widoczny.

```
FogStart air=0.1 water=0.1
```

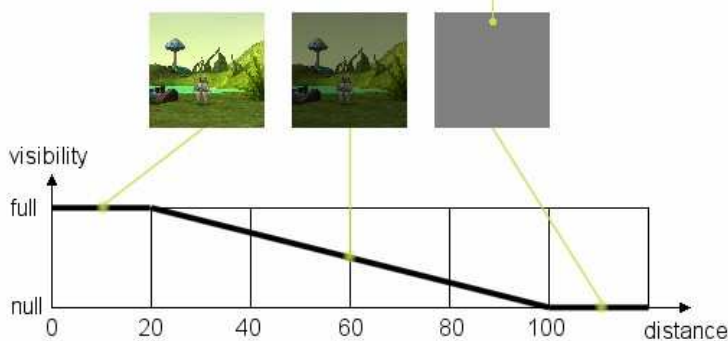
Parametr ten wpływa na to jak gęsta będzie mgła w swym początkowym punkcie. Im większą wartość będzie miał parametr „pole widzenia” (DeepView) tym więcej obiektów przyjmie kolor mgły (FogColor). Parametr ten ma za zadanie symulację mgły. Wartość 0.1 to gęsta mgła. Wartość 0.9 to lekka mgła.

Przykład:

- DeepView air=100.00 // pole widzenia na 100 metrów
- FogStart air=0.2
- FogColor air=128;128;128;128 // szary

Odległość od obserwatora	
0 do 20 metrów	Normalna widoczność
20 do 100 metrów	Widoczność ograniczona stopniowo narastającą mgłą
100 metrów i więcej	Zerowa widoczność

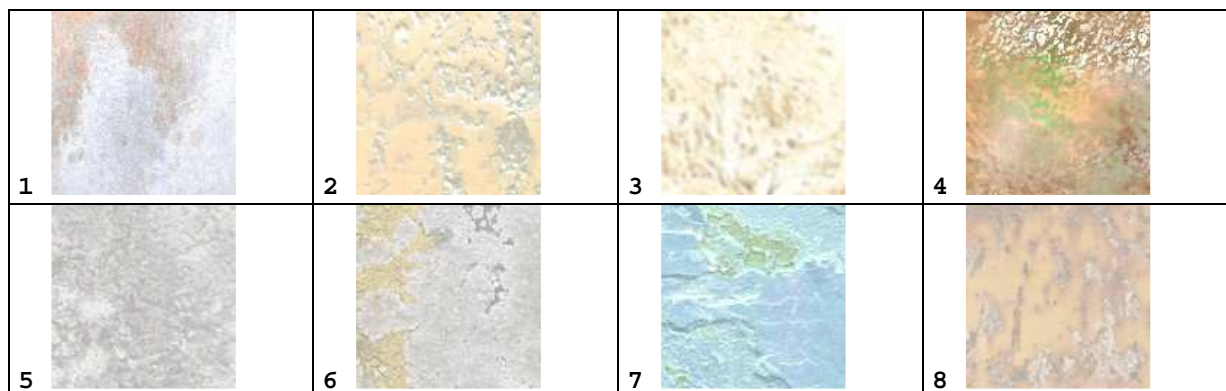
```
DeepView air=100.00
FogStart air=0.2
FogColor air=128;128;128;128
```



Oto wykres zmniejszania się widoczności dla w/w parametrów. Oś „visibility” jest osią widoczności. Oś „distance” jest osią odległości. Wartość „full” oznacza pełną widoczność. Wartość „null” oznacza widoczność zerową.

```
SecondTexture rank=3
```

Są to tekstury służące do symulacji brudu na robotach i budynkach. Możesz korzystać z wartości od 1 do 8:



```
Background image="back01.tga" up=76;105;226;0 down=192;250;255;0
```

Jest to tekstura, stanowiąca o wyglądzie nieba. Możesz wybierać spośród następujących tekstur:



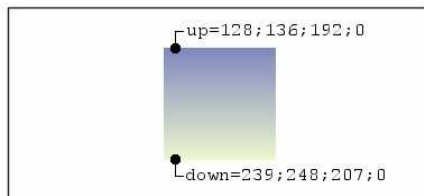
Jeśli nazwa obrazka poprzedzona jest frazą: %user%, to zostanie on zapożyczony z foldera własnego poziomu. Możesz korzystać z plików .bmp lub .tga. Tekstura nieba nie może przekroczyć wymiaru 1280x480 pikseli.

Wartości up i down są używane, jeśli obrazek nie zostanie zadeklarowany lub jeśli opcja „niebo” zostanie wyłączona w opcjach Colobota.

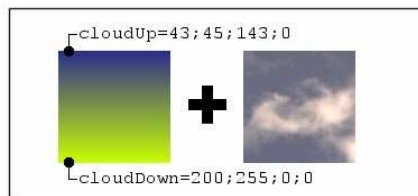
Oba te kolory determinowane są przez swoje komponenty RGBA (red (czerwień), green (zieleń), blue (niebieski), alpha). Kolor nieba to kolor zaczynający się od barwy koloru up, stopniowo schodzący do barwy koloru down. Jeśli nie będziesz chciał skorzystać z tekstury nieba, po prostu pomiń część image. Parametry cloudUp i cloudDown umożliwiają określenie koloru nieba za chmurami (komenda TerrainCloud). Jeśli chmury nie występują (z powodu niskiej wydajności komputera), to zostaną użyte kolory up i down. Te dwa kolory powinny jak najbardziej imitować barwę „prawdziwych” chmur.

```
Background up=128;136;192;0 down=239;248;207;0 cloudUp=43;45;143;0
cloudDown=200;255;0;0
```

Sky

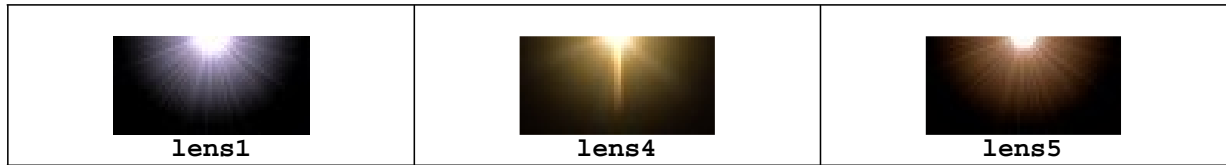


Sky



```
FrontsizeName image="lens5.tga"
```

Jest to nazwa pliku z teksturą efektów świetlnych.

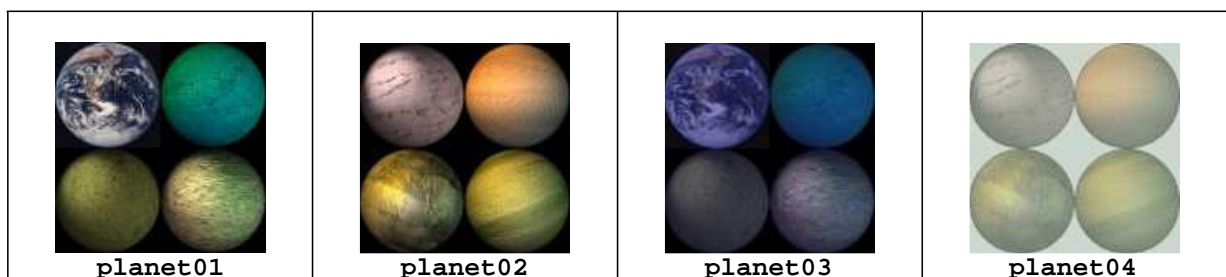
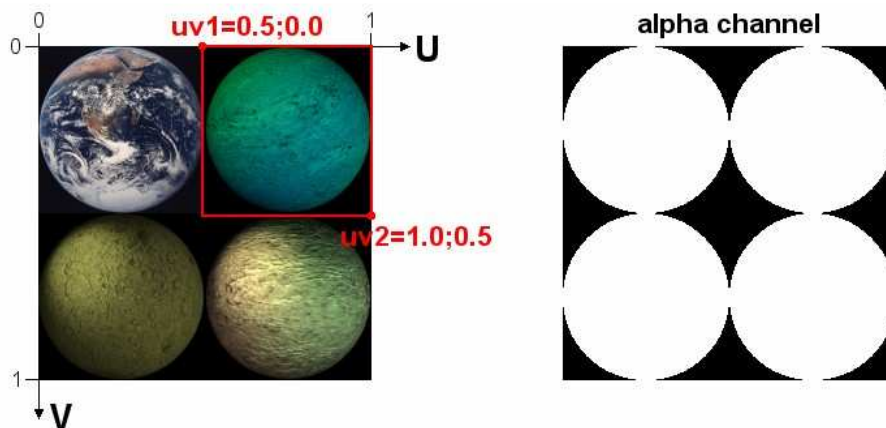


Jeśli nazwa obrazka poprzedzona jest frazą: %user%, to jest on zlokalizowany w folderze poziomu użytkownika (w przypadku ćwiczenia przykładowego będzie to: user\sample01\). Możesz korzystać z plików graficznych typu .bmp lub .tga. Plik nie może przekroczyć rozmiaru 512x256 pikseli. Domyślnym rozmiarem jest tu 128x64 pikseli.

```
Planet mode=0 pos=1.2;0.2 dim=0.35 speed=0.001 dir=0.1 image="planet04.tga"
uv1=0.0;0.5 uv2=0.5;1.0
```

Za pomocą tej frazy możesz zdeterminować czy planeta widoczna na niebie będzie nieruchoma, czy też mobilna.

- mode=0 : jest to planeta widoczna podczas normalnej fazy gry.
- mode=1 : jest to planeta widoczna podczas filmu wyświetlanego podczas podróży, o ile statek kosmiczny został stworzony za pomocą komendy: CreateObject run=11. (zobacz rozdział 3.2.5.2).
- pos to początkowa pozycja planety na niebie. Pierwsza wartość to kierunek. Druga wartość to wysokość.
- dim to rozmiar planety. Możesz korzystać z wartości od 0.02 dla bardzo małych gwiazd (takich jak **Krystalia**) do 0.5 dla prawdziwie wielkich planet (np. widok Ziemi z Księżycyca).
- Speed to prędkość planety. Generalnie jest ona bardzo niska (0 dla planet nieruchomych). Typowe wartości oscylują wokół 0.001
- dir to pionowy wektor kierunku planety. Jeśli dir ma wartość 0 to planeta stoi w tym samym punkcie na niebie. Typowe wartości oscylują wokół 0.1
- image to nazwa obrazka, który ma zostać użyty (zobacz niżej).
- Pliki graficzne planet zawierają zawsze 4 różne planety. uv1 to współrzędne w górnym-lewym rogu. uv2 to współrzędne w dolnym-prawym rogu obrazka planety. Współrzędne w górnym-lewym rogu wynoszą 0;0. Współrzędne w dolnym-prawym rogu wynoszą 1;1.



Jeśli nazwa obrazka poprzedzona jest frazą: %user%, to jest on zlokalizowany w folderze poziomu użytkownika (w przypadku ćwiczenia przykładowego będzie to: user\sample01\). Musisz używać plików typu: .tga. Kanał alpha determinuje rysy planety. Plik graficzny planety nie powinien przekraczać rozmiaru: 256x256 pikseli.

3.2.3. Podłoże

Część ta determinuje rzeźbę terenu i jego teksturę. Możesz tu także ustawić ilość wody znajdującej się na planszy i jej poziom, a także ilość i rodzaj surowców znajdujących się pod ziemią.

```
TerrainGenerate vision=250 depth=1 hard=0.6
```

- `vision` determinuje do jakiej odległości ma być generowany teren. Dystans ten musi być co najmniej dwa razy większy od większej z dwóch wartości `DeepView` (`air` lub `water`).
- `depth` musi być zawsze ustawione na 1.
- `hard` determinuje twardość podłoża, który wpływa na głośność kroków astronauty.

```
TerrainWind speed=0;-5
```

Jest to wektor wskazujący kierunek i siłę wiatru. Na księżycu nie ma wiatru. Odzwierciedla to fraza: `speed=0;0`. Fraza: `speed=10;0` odpowiada natomiast silnemu wiatrowi wiejącemu z zachodu na wschód. Wiatr wpływa na ruch chmur (`TerrainCloud`), dymu i flag. Wiatr nie ma jednak żadnego wpływu na poruszanie się robotów.

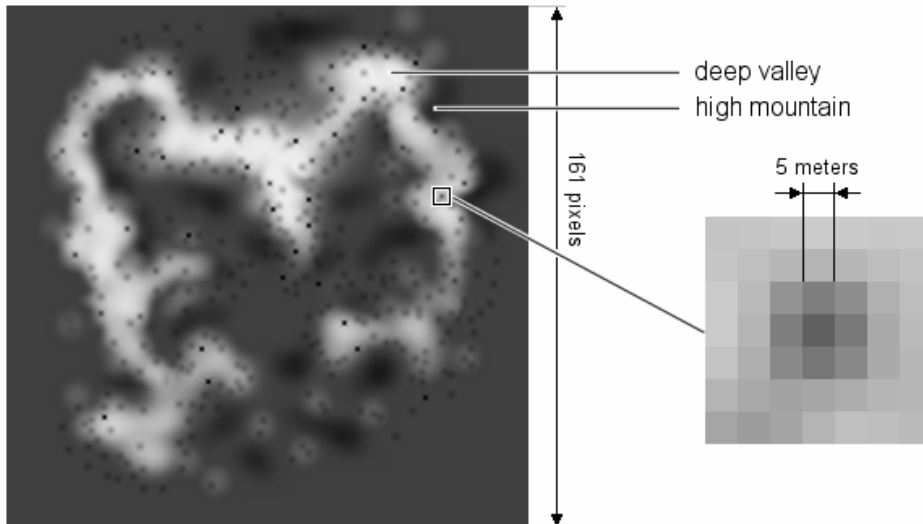
```
TerrainBlitz sleep=60 delay=5 magnetic=100
```

Możesz tu aktywować przypadkowe wyładowania elektryczne (burze). W standardowych misjach Colobota, komenda ta używana jest w misjach na **Orfenii**.

- `sleep` to czas do wystąpienia pierwszych błyskawic. Zwłoka ta umożliwi Ci wybudowanie stacji odgromnikowej (`PowerCaptor`).
- `delay` to średnia zwłoka (w sekundach) pomiędzy dwoma błyskawicami.
- Każdy metalowy obiekt otoczony jest przez wirtualną sferę, przyciągającą pioruny. `magnetic` określa promień (w metrach) tej sfery. Jeśli piorun uderzy w obręb strefy `magnetic`, obiekt zostanie zniszczony.

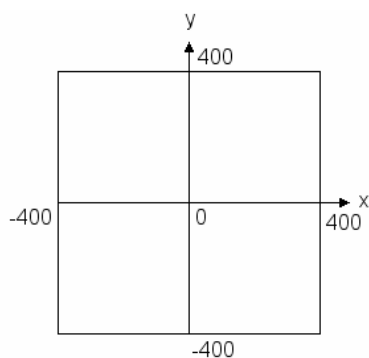
```
TerrainRelief image="%user%\relief01.bmp" factor=1.0
```

Obraz rzeźby terenu przedstawiony jest w 256 poziomach szarości. Odpowiadający mu plik graficzny: .bmp jest obrazem o wymiarach dokładnie 161 x 161 pikseli. Kolor biały odpowiada najniższej możliwej wysokości. Kolor czarny odpowiada najwyższej możliwej wysokości. `factor` jest to współczynnik rozciągnięcia, którego wartość oscyluje przeważnie w okolicach wartości: 1.0. W takim wypadku obraz w 256 poziomach szarości odpowiada wysokości 64 metrów. Jeden poziom odpowiada zatem różnicy 0,25 metra.



Oto przykładowy wykres rzeźby terenu. Lokacja opisana jako „high mountain” to wysoka góra. Lokacja opisana jako „deep valley” to głęboka dolina.

Współrzędne środkowego piksela obrazka: 80;80 odpowiadają współrzędnym 0;0 metrów w COLOBOCIE. Współrzędne -400;400, w COLOBOCIE odpowiadają punktowi ekstremalnie wysuniętemu na północny-zachód. Jeden piksel wykresu topograficznego to obszar o wielkości 5x5 metrów w COLOBOCIE.



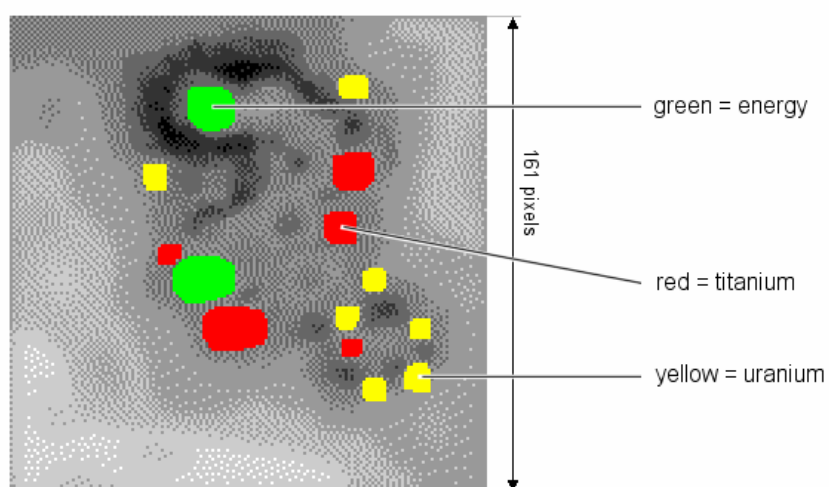
Świat COLOBOTA ma rozmiar 800x800 metrów. Rzeźba terenu może być modyfikowana za pomocą edytorów graficznych, takich jak np.: PaintShop lub poprzez dowolne podmiany plików: `reliefxx.bmp` znajdujących się w folderze: `textures\`.

```
TerrainResource image="%user%\res01.bmp"
```

Obrazek ten determinuje rozmieszczenie surowców w podłożu. Musi być on plikiem graficznym: .bmp o 256 kolorach, rozmiarze 161 x 161 pikseli i w standardowej palecie Windows.

Kolor	RGB	Numer w palecie	Zawartość podłoża
Czerwony	255;0;0	5	Tytan
Zielony	0;255;0	30	Energia
Żółty	255;255;0	35	Uran
Zielony	0;104;0	24	Klucz A
Zielony	51;104;0	25	Klucz B
Zielony	102;104;0	26	Klucz C
Zielony	153;104;0	27	Klucz D

Wszystkie inne kolory, wraz z odcieniami szarości będą ignorowane.



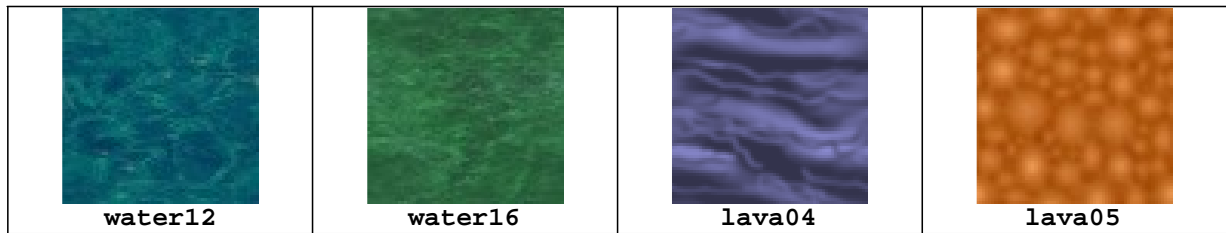
Oto przykładowy wykres zawartości podłoża. Obszar opisany jako „green=energy” to oznaczony na zielono obszar złóż energetycznych. Obszar opisany jako „red=titanium” to oznaczony na czerwono obszar złóż tytanu. Obszar opisany jako „yellow=uranium” to oznaczony na żółto obszar złóż uranu.

Najprostszą metodą sporządzenia tego wykresu jest skorzystanie z szarego wykresu używanego w TerrainRelief i przekonwertowanie go do 256 kolorów.

```
TerrainWater image="water16.tga" level=30.0 moveX=1.0 moveY=0.0  
color=0;240;100;0 brightness=0.2
```

Komenda ta determinuje wygląd wody i lawy. Niemożliwe jest występowanie jeziora wodnego i jeziora lawy w tej samej misji.

- `image` określa to czy w danej misji ma występować lawa, czy też woda.
- `level` to poziom wody lub lawy w odniesieniu do absolutnej wartości zero, która równoznaczna jest białemu kolorowi na mapie topograficznej (TerrainRelief). Wysokość (współrzędna z) robota jest w takiej sytuacji zawsze zależna od poziomu wody (lub lawy). Wartość negatywna oznacza, iż robot znajduje się pod wodą. Poziom wszystkich jezior (wody lub lawy) w danej misji jest taki sam. Nie jest możliwe, by w tej samej misji jedno jezioro było w dolinie, a drugie na szczycie wzniesienia.
- `moveX` to amplituda horyzontalnego ruchu na powierzchni.
- `moveY` to amplituda ruchu wertykalnego.
- `color` i `brightness` umożliwiają modyfikację koloru tekstur.



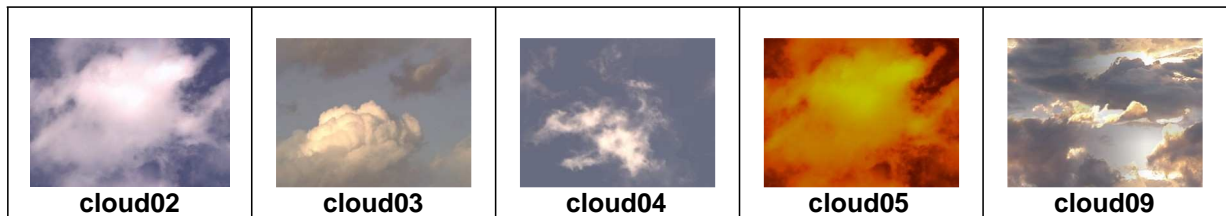
Jeśli nazwa obrazka poprzedzona jest frazą: %user%, to musi się on znajdować w folderze ćwiczeniowym (w przypadku ćwiczenia przykładowego będzie to: user\sample01\). Możesz korzystać z plików graficznych: .bmp lub .tga. Ich rozmiar nie może jednak przekroczyć rozmiaru 256x256 pikseli.

```
TerrainLava mode=1
```

- Jeśli zadeklarowana jest fraza: mode=1, astronauta zginie natychmiast po kontakcie z lawą. W trybie tym kamera nigdy nie zejdzie poniżej poziomu lawy.

```
TerrainCloud image="cloud05.tga" level=125.0
```

Obraz ten jest wykorzystywany przy modelowaniu ruchomych chmur. Ruch chmur zależy od kierunku wiatru (zobacz: TerrainWind). Możesz skorzystać z następujących obrazów:



Jeśli nazwa obrazka jest poprzedzona frazą: %user%, to znajduje się on w folderze ćwiczeniowym (w przypadku ćwiczenia przykładowego będzie to: user\sample01\). Możesz korzystać z plików graficznych: .bmp lub .tga. Ich rozmiar nie powinien przekroczyć rozmiaru 640x480 pikseli.

- level to maksymalna wysokość chmur. Pułap chmur musi być wyższy co najmniej o kilkadziesiąt metrów od wierzchołka najwyższej góry.

Za pomocą komendy: Background i komend: cloudUp oraz cloudDown możesz określić kolor nieba za chmurami.

3.2.4. Tekstury podłoża

Istnieją dwa systemy pokrywania podłoża teksturami:

	Komendy	Znaczenie
1	TerrainInitTextures	Tekstury kładzione są "na ślepo". Innymi słowy wybrana tekstura pokrywa cały teren na mapie, niezależnie od jego rzeźby i tym podobnych uwarunkowań.
2	TerrainMaterial TerrainInit TerrainLevel	Na teksturę składa się szereg elementów zależnych od rzeźby i wysokości terenu. I tak np. pod wodą znajdzie się inny piasek niż na powierzchni. Na średnich wysokościach pojawiają się rośliny, a na wysokościach większych śnieg itd.

Misje rozgrywane na **Księżycu** korzystają z pierwszego systemu. Misje rozgrywane na **Nowej Ziemi** korzystają natomiast z systemu drugiego.

Jako, że komendy mapujące są bardzo skomplikowane, powinieneś korzystać z komend istniejących w innych ćwiczeniach. Oto kilka powierzchniowych detali na ich temat:

```
TerrainInitTextures image="mars" dx=4 dy=2 table=1;2;3;4;5;6;7;8
```

Komenda ta umożliwi przypisanie konkretnej tekstury do podłoża. W powyższym przykładzie tekstura podłoża składa się z 8, następujących obrazków:

mars001.tga	mars002.tga	mars003.tga	mars004.tga
mars005.tga	mars006.tga	mars007.tga	mars008.tga

W wyniku powyższej komendy, kompozycja tych 8 tekstur zostaje powtórzona we wszystkich kierunkach.

Z tego też powodu prawa krawędź: mars002.tga będzie stykać się z lewą krawędzią: mars003.tga.

Prawa krawędź: mars004.tga będzie też stykać się z lewą krawędzią: mars001.tga.

Górna krawędź: mars002.tga będzie stykać się z dolną krawędzią: mars006.tga. itd.

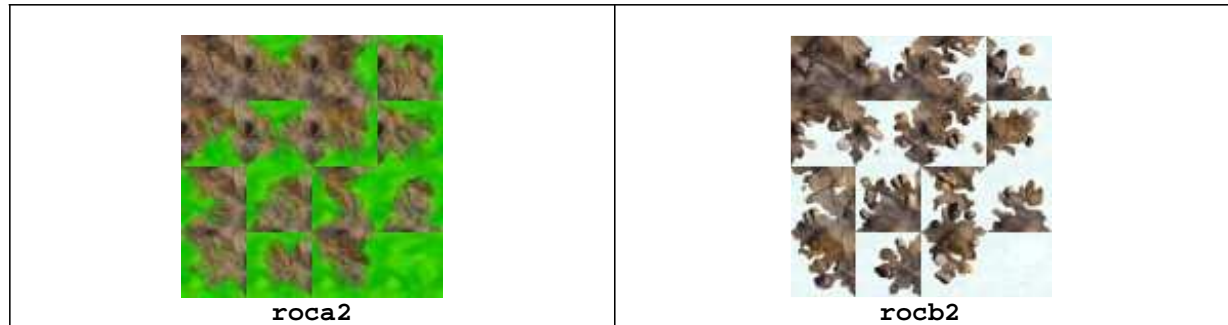
Nazwa pliku składa się z nazwy obrazka i 3 cyfr. Jeśli nazwa obrazka poprzedzona jest frazą: %user%, to znajduje się on w katalogu własnego poziomu użytkownika (w przypadku ćwiczenia przykładowego będzie to: user\sample01\). Możesz korzystać z plików graficznych: .bmp lub .tga. Przykład :

- TerrainInitTextures image="%user%\moon.bmp" dx=1 dy=1 table=15

Powyższa formuła oznacza użycie pliku: user\sample01\moon015.bmp.

```
TerrainMaterial id=1 image="roca2" u=0.00 v=0.00 up=1 down=1 left=1 right=1
hard=0.8
```

Definicje podłoża korzystają z numerów id. Np. **Nowa Ziemia** korzysta z następujących tekstur:



Każda tekstura składa się z 16 kawałków (4x4), tak jak to widać na powyższych obrazkach. Do poszczególnych id przypisane są główne charakterystyki terenu. Wartości id oscylują pomiędzy 1 i 4 :

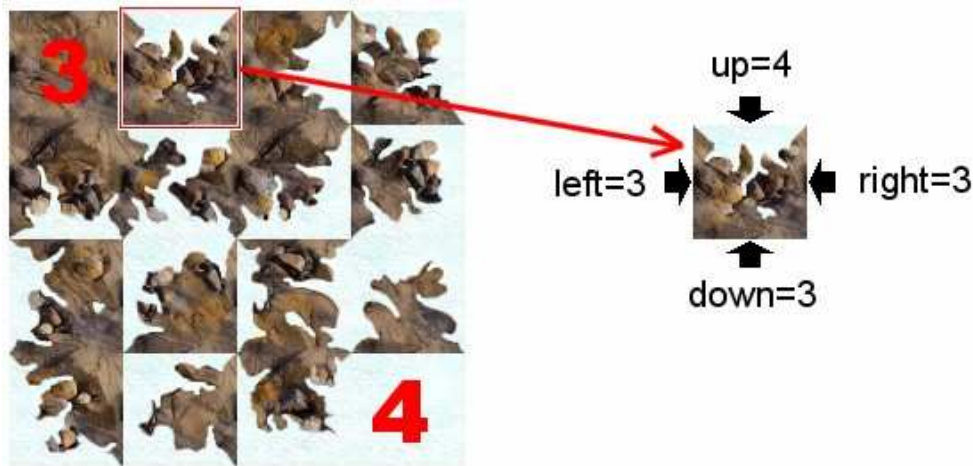
- id=1 image="roca2" u=0.00 v=0.00 up=1 down=1 left=1 right=1 // skały
- image="roca2" u=0.25 v=0.00 up=2 down=1 left=1 right=1
- image="roca2" u=0.50 v=0.00 up=1 down=1 left=1 right=2
- image="roca2" u=0.75 v=0.00 up=2 down=1 left=1 right=2
- image="roca2" u=0.00 v=0.25 up=1 down=2 left=1 right=1
- image="roca2" u=0.25 v=0.25 up=2 down=2 left=1 right=1
- image="roca2" u=0.50 v=0.25 up=1 down=2 left=1 right=2
- image="roca2" u=0.75 v=0.25 up=2 down=2 left=1 right=2
- image="roca2" u=0.00 v=0.50 up=1 down=1 left=2 right=1
- image="roca2" u=0.25 v=0.50 up=2 down=1 left=2 right=1
- image="roca2" u=0.50 v=0.50 up=1 down=1 left=2 right=2
- image="roca2" u=0.75 v=0.50 up=2 down=1 left=2 right=2
- image="roca2" u=0.00 v=0.75 up=1 down=2 left=2 right=1
- image="roca2" u=0.25 v=0.75 up=2 down=2 left=2 right=1
- image="roca2" u=0.50 v=0.75 up=1 down=2 left=2 right=2
- id=2 image="roca2" u=0.75 v=0.75 up=2 down=2 left=2 right=2 // trawa
- id=3 image="rocb2" u=0.00 v=0.00 up=1 down=1 left=1 right=1 // skały
- ...
- id=4 image="rocb2" u=0.75 v=0.75 up=3 down=3 left=3 right=3 // śnieg

Tekstury bez id są zdefiniowane jako umożliwiające „przejście”. Komendy up, down, left i right (górze, dół, lewo, prawo) nadają teksturze id najbliższej tekstury w znajdującej się w określonej lokacji.

Przykład :

- `image="rocb2" u=0.25 v=0.00 up=4 down=3 left=3 right=3`

To drugi kawałek, w pierwszym wersie następującego obrazka (id=3:skały i id=4:śnieg) :



`hard` wpływa na twardość podłoża. Wartość ta oscyluje pomiędzy 0 i 1. Twardość podłoża wpływa na głośność kroków astronauty.

Uwaga : Typy podłoża są używane w misjach na Ziemi do tworzenia dróg (np. `scene\scene101.txt`). Trzy id odpowiadają drogom Wschód/Zachód (id=3), Północ/Południe (id=4) i skrzyżowaniu (id=5).

Jeśli nazwa obrazka poprzedzona jest frazą: `%user%`, to znajduje się on w folderze własnego poziomu użytkownika (w przypadku ćwiczenia przykładowego będzie to: `user\sample01\`). Możesz korzystać z plików graficznych: `.bmp` i `.tga`. Ich rozmiar nie może przekroczyć rozmiaru: 512x512 pikseli.

```
TerrainInit id=3
```

Komenda ta ujednocza całe podłoże danej mapy.

```
TerrainLevel id=10;11;10;11;12;13 min=0 max=99 slope=0.0 freq=100
```

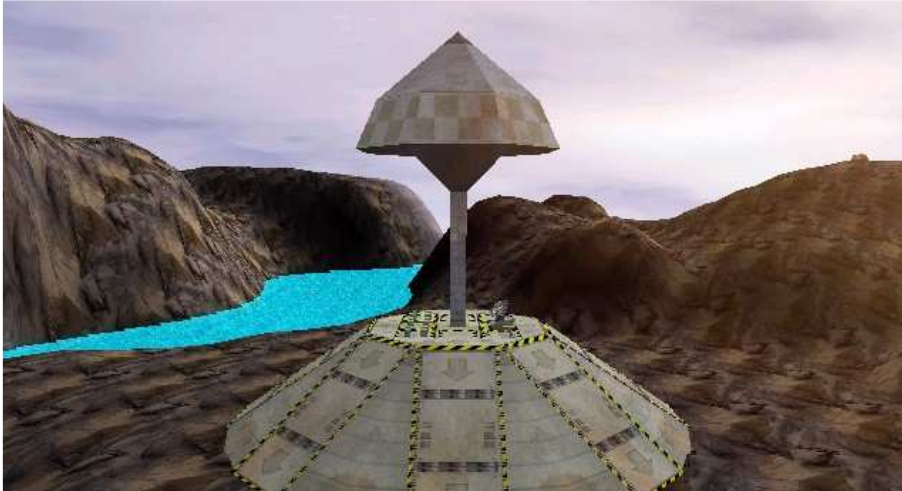
Komenda ta pozwala wybrać podłoże dla danej mapy. Podłoże jest wybierane po uwzględnieniu szeregu parametrów:

- `id` to typ podłoża, który ma zostać użyty. Jeśli zadeklarujesz kilka typów, zostanie on wybrany losowo. Możliwe jest też zastosowanie proporcji. Przykład: deklaracja `id=10;10;10;11` to polecenie użycia 3 razy więcej typu 10 niż 11.
- `min` i `max` determinuje minimalną i maksymalną wysokość terenu, pomiędzy którymi ma zostać użyty typ podłoża. Pamiętaj o posługiwaniu się wysokościami absolutnymi (zamiast wysokościami nad poziomem wody/lawy).
- `slope` determinuje na jakim nachyleniu terenu powinien zostać umieszczony dany typ podłoża. Np. ustawienie: `slope=9` umieści dany typ terenu tylko na bardzo płaskich obszarach. Jest to użyteczne dla np. docelowo śnieżnych typów terenów itd.
- `center` determinuje środek opcjonalnej strefy, w której zostanie zastosowany typ. Jeśli zostaną zdefiniowane parametry `center` i `radius` to typ podłoża zostanie zaaplikowany tylko w obrębie określonego obszaru.
- `radius` determinuje promień opcjonalnej strefy.
- `freq` umożliwi Ci przydzielenie typu podłoża tylko dla określonego procentu podłoża. Np. przy ustawieniu: `freq=25`, tylko 25 % powierzchni planety (w odniesieniu do `min`, `max`, `slope`, etc.) zostanie pokryte danym typem podłoża.

Możesz użyć szeregu komend: `TerrainLevel`, które zostaną wykonane w narzuconej kolejności.

Oto kilka przykładów zastosowania komend, na przykładzie **Nowej Ziemi** :

```
TerrainInit id=1 // skały
```



```
TerrainInit id=1 // skały
```

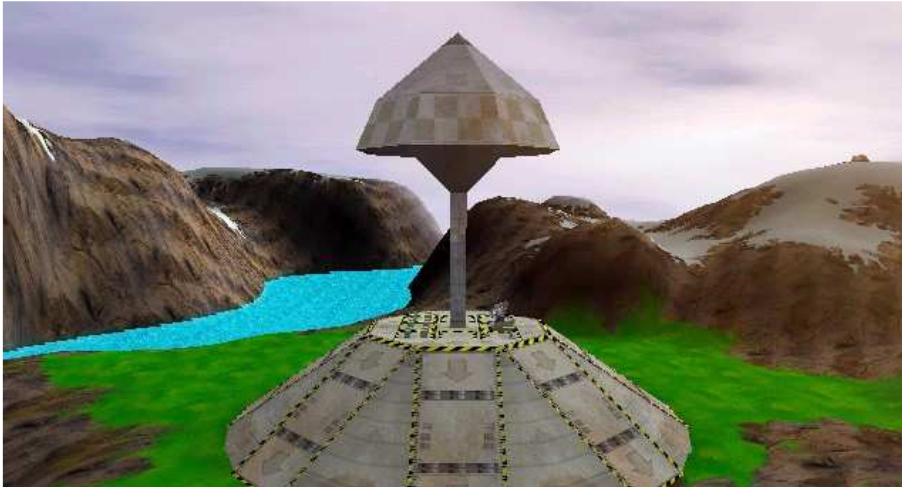
```
TerrainLevel id=2 min=25 max=37 slope=3.0 freq= 70 // trawa
```



```
TerrainInit id=1 // skały
```

```
TerrainLevel id=2 min=25 max=37 slope=3.0 freq= 70 // trawa
```

```
TerrainLevel id=4 min=37 max=99 slope=9.0 freq= 80 // śnieg
```





TerrainCreate

Powyższa komenda musi kończyć wszystkie komendy terenowe. Jej wstawienie powoduje wykonanie komend znajdujących się w bloku.

```
GroundSpot pos=-125;100 radius=40 color=190;220;160
```




Za pomocą tej komendy możesz określić kolor poszczególnych obszarów. Możesz np. przydzielić zielony kolor dla obszaru z bujną roślinnością.. pos i radius wpływają na rozmiar strefy, w obrębie której ma zostać zmieniony kolor.

Przykłady :

	
	radius=30 color=150;255;150
	
radius=50 color=255;0;0	radius=10 color=255;0;0

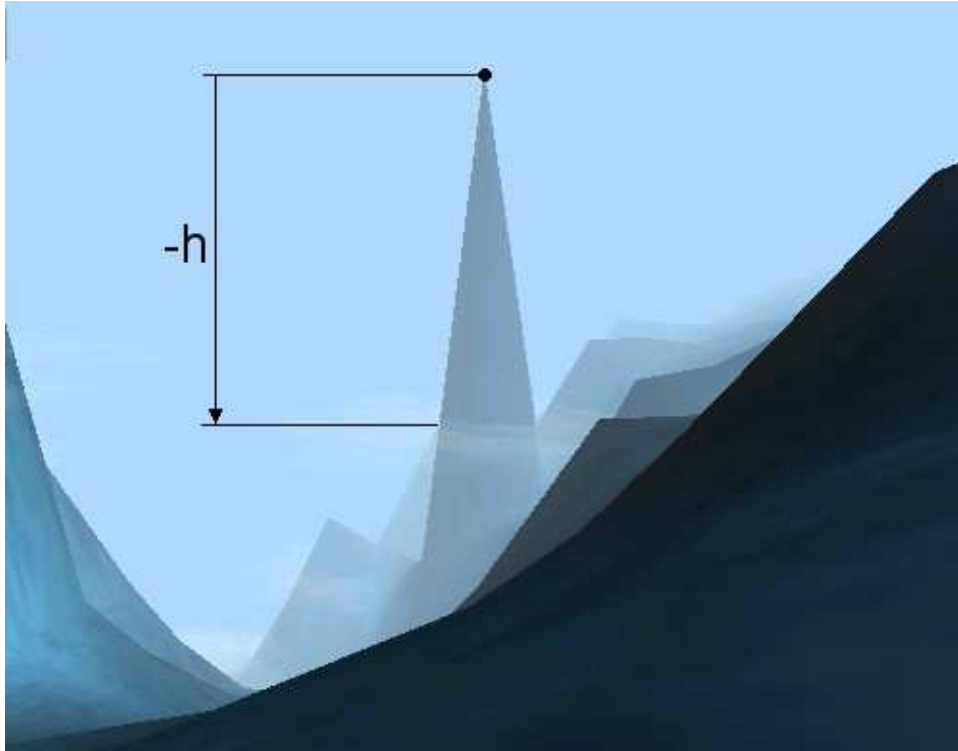
```
GroundSpot color=0;128;255 min=-10 max=24 smooth=10
```

Możesz także wpływać na kolor jezior, poprzez określenie minimalnej i maksymalnej wysokości oddziaływania koloru:

		
	color=0;255;0 min=-10 max=28.75	color=0;0;255 min=-10 max=28.75

```
CreateFog pos=57;69 height=3 dim=20 delay=4.0 type=4
```

Powyższa fraza umożliwia tworzenie horyzontalnych poziomów mgły. Poziomy te widoczne są szczególnie na planszy Krystalia 4 (Zaginiona dolina). `Height` to relatywna odległość od podłoża w pozycji: `pos`. Jeśli pozycja strefy znajduje się na wierzchołku, to `height` może być ujemne. W takim przypadku mgła otoczy dany wierzchołek. `delay` to szybkość rotacji warstwy.



Typ	Wygląd
0 i 1	Warstwa niebieskawa/sinawa.
2 i 3	Czerwono-pomarańczowy okrąg.
4 i 5	Warstwa szara.
6 i 7	Warstwa żółtawa.

```
MaxFlyingHeight max=40
```

Maksymalna wysokość absolutna (zignorowanie poziomu wody), do której może wnieść się astronauta lub robot.

3.2.5. Obiekty

W sekcji tej możesz stworzyć szereg obiektów 3D.

BeginObject

Komenda ta rozpoczyna sekcję obiektów i musi być ona zadeklarowana przed komendą: CreateObject.

CreateObject pos=7;-10 dir=1.5 type=Me

Powyższa komenda umożliwia stworzenie obiektu. Jeśli wskazaną pozycją (pos) będzie platforma Statku kosmicznego, to w chwili rozpoczęcia misji, obiekt będzie znajdował się na jego pokładzie.

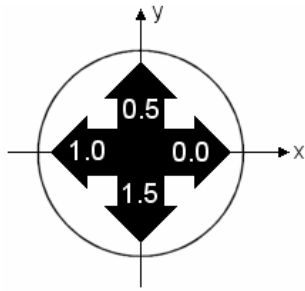
pos determinuje pozycję obiektu. Najprostszym sposobem na wskazanie właściwej pozycji jest przełączenie się w tryb debugowania i skierowanie astronauty do pożądanej lokacji.

Aby aktywować tryb debugowania, należy wcisnąć kombinację klawiszy: Ctrl+Break, wprowadzić komendę: showstat i zatwierdzić ją klawiszem Enter. Następnie należy ponownie wcisnąć klawisze: Ctrl+Break i wprowadzić komendę: showpos. Od teraz pozycja wybranego obiektu będzie stale widoczna w lewym dolnym rogu ekranu :



```
CreateObject pos=-9.96;47.45 dir=1.7 type=TitaniumOre
```

Kierunek jest liczbą oscylującą pomiędzy wartościami 0 i 2 :



Oto lista wszystkich typów obiektów :

Ogólne :

```
type=Me           // astronauta
type=SpaceShip    // statek kosmiczny
```

Roboty :

```
type=PracticeBot // robot treningowy
type=TargetBot   // robot cel
```

Transportery (na kołach, na gąsienicach, latający, na nogach)

```
type=WheeledGrabber
type=TrackedGrabber
type=WingedGrabber
type=LeggedGrabber
```

Działa (na kołach, na gąsienicach, latający, na nogach)

```
type=WheeledShooter
type=TrackedShooter
type=WingedShooter
type=LeggedShooter
```

Działa organiczne (na kołach, na gąsienicach, latający, na nogach)

```
type=WheeledOrgaShooter
type=TrackedOrgaShooter
type=WingedOrgaShooter
type=LeggedOrgaShooter
```

Szperacze (na kołach, na gąsienicach, latający, na nogach)

```
type=WheeledSniffer
type=TrackedSniffer
type=WingedSniffer
type=LeggedSniffer
```

```
type=Thumper (Uderzacz)
type=PhazerShooter (Działo fazowe)
type=Recycler (Recykler)
type=Shielder (Osłaniacz)
type=Subber (Nurek)
```

Budynki :

```
type=Derrick (Kopalnia)
type=BotFactory (Fabryka robotów)
type=PowerStation (Elektrownia)
type=Converter (Huta)
type=RepairCenter (Warsztat)
type=DefenseTower (Wieża obronna)
type=AlienNest (Gniazdo obcych)
type=ResearchCenter (Centrum badawcze)
```

```

type=RadarStation (Radar)
type=ExchangePost (Stacja przekaźnikowa informacji)
type=PowerPlant (Fabryka ogniów elektrycznych)
type=AutoLab (Laboratorium)
type=NuclearPlant (Elektrownia atomowa)
type=PowerCaptor (Stacja odgromnikowa)
type=Vault (Skrytka)
type=StartArea (Podest startowy)
type=GoalArea (Podest końcowy)
type=Target1 // cel dla ćwiczeń lotniczych
type=Target2
type=Houston // centrum kontroli misji

```

Obiekty przenośne :

```

type=TitaniumOre (Ruda tytanu)
type=UraniumOre (Ruda uranu)
type=Titanium (Kostka tytanu)
type=PowerCell (Zwykłe ogniwo elektryczne)
type=NuclearCell (Atomowe ogniwo elektryczne)
type=OrgaMatter (Materia organiczna)
type=BlackBox (Czarna skrzynka)
type=KeyA..D (Klucze A, B, C i D)
type=TNT // materiały wybuchowe

```

Rośliny i obiekty :

```

type=Greenery0..4 // małe, standardowe rośliny
type=Greenery5..7 // mała koniczynka
type=Greenery10..14 // rozkwitająca roślina
type=Greenery15..19 // paprotka
type=Tree0..3 // wysokie drzewo
type=Mushroom1 // nieszkodliwy grzyb
type=Mushroom2 // grzyb korozyjny
type=MegaStalk0..5 // wielka, dziwna roślina

type=Quartz0..3 // od małego do dużego kwarcu
type=Barrier0 // krótka barierka
type=Barrier1 // długa barierka
type=ApolloLEM // tylko na Księżycu :
type=ApolloJeep (Pojazd księżycowy)
type=ApolloFlag (Flaga amerykańska)
type=ApolloModule (Moduł Apollo)
type=ApolloAntenna (Mała antena satelitarna)

```

Wraki:

```

type=WreckBotw1..2 // roboty z kołami
type=WreckBott1..2 // roboty z małymi gąsienicami
type=WreckBotr1..2 // roboty z dużymi gąsienicami

```

Ruiny:

```

type=RuinBotFactory (Ruiny fabryki robotów)
type=RuinDoor // drzwi huty
type=RuinSupport // ruiny radaru
type=RuinRadar // podstawa radaru
type=RuinConvert // ruiny huty
type=RuinBaseCamp // podstawa statku kosmicznego
type=RuinHeadCamp // dach statku kosmicznego

```

Wrogowie:

```

type=AlienQueen // Królowa obcych
type=AlienEgg // Jajo obcych
type=AlienAnt // Mrówka

```

```

type=AlienSpider (Pająk)
type=AlienWasp (Osa)
type=AlienWorm (Robak)

```

Wskaźniki:

```

type=PowerSpot // znacznik źródła energii w podglebiu
type=TitaniumSpot // znacznik złoża tytanu w podglebiu
type=UraniumSpot // znacznik złoża uranu w podglebiu
type=KeyA..DSpot // znacznik klucza w podglebiu
type=WayPoint // krzyż dla ćwiczeń
type=BlueFlag (Niebieska flaga)
type=RedFlag (Czerwona flaga)
type=GreenFlag (Zielona flaga)
type=YellowFlag (Żółta flaga)
type=VioletFlag (Fioletowa flaga)

```

Różne :

```

type=Mine // mina
type=Portico // żuraw przesuwalny
type=Bag // zestaw przetrwania
type=Home // mały ładny domek (na Nowej Ziemi)
type=Tech // inżynier z Houston
type=Firework (fajerwerki)

```

CreateObject może zawierać więcej parametrów :

```
CreateObject ... script1="%user%\scharge2.txt"
```

Jest to nazwa programu CBOT, który ma być załadowany na 1 slot programowy robota lub insekta. Za pomocą komend od: script1 do script10, możesz załadować do 10 programów. W celu dokonania referencji do uniwersalnego programu, znajdującego się w katalogu: script\, usuń frazę: %user%\ znajdującą się przed nazwą skryptu.

```
CreateObject ... run=1
```

Numer programu, który zostanie wykonany z chwilą wystartowania misji. Komenda ta jest przydatna w kontekście wrogów i robotów, mających inne zadania niż stanie w miejscu.

```
CreateObject ... select=1
```

Jeden i tylko jeden obiekt może mieć przypisany parametr: select=1. Obiekt ten po załadowaniu misji zostanie wybrany, a także zostanie na niego skierowane oko kamery.

```
CreateObject ... power=0.5
```

Parametr ten determinuje początkowy stan ogniwa elektrycznego robota. Może on przyjąć następujące wartości::

Wartość	Znaczenie
-1	Robot rozpoczyna ćwiczenie bez ogniwa.
0	Robot rozpoczyna ćwiczenie z pustym ogniwem.
0..1	Robot rozpoczyna ćwiczenie z mniej lub bardziej naładowanym zwykłym ogniwem.
1..100	Robot rozpoczyna ćwiczenie z mniej lub bardziej naładowanym ogniwem atomowym.

```
CreateObject ... range=100
```

Jest to model lotu. Im większa wartość tym wolniej nagrzewa się silnik w trakcie lotu. Domyślną wartością jest tu 30. Roboty budowane w trakcie ćwiczenia zawsze charakteryzują się domyślnym modelem lotu.

```
CreateObject ... shield=1
```

Jest to początkowy stan osłon. 1 to w pełni naładowane osłony. 0 to wyładowane osłony. Jeśli robot z wyładowanymi osłonami zostanie trafiony, będzie to równoznaczne jego zniszczeniu. Możliwe jest też asygnowanie wartości pośrednich.

```
CreateObject ... magnifyDamage=2
```

Zwiększa lub zmniejsza ilość obrażeń, jaką otrzymuje robot w wyniku wrogiego ostrzału lub zderzenia z innym robotem. Roboty budowane w *trakcie* ćwiczenia zawsze charakteryzują się domyślną wytrzymałością.

Wartość	Efekt
0.5	Podwójna wytrzymałość
1	Domyślna wytrzymałość
2	Podwójne obrażenia

```
CreateObject ... proxyActivate=1 proxyDistance=10
```

ProxyDistance oznacza odległość, na jaką należy zbliżyć się do obiektu, by mógł on zostać "znaleziony".

```
CreateObject ... selectable=0
```

Parametr ten uniemożliwia wybranie danego obiektu. Jest on używany na niektórych misjach dla robotów, które po prostu jeżdżą w kółko lub wykonują pewne zadania niezależnie od gracza.

3.2.5.1. Wiersz poleceń

Wiersz poleceń umożliwia wysyłanie argumentów do programu. Stworzymy program, który stworzy mrówkę :

```
CreateObject pos=-55;243 cmdline=-55;243;-62;234 dir=0.0 type=AlienAnt  
script1="ant04.txt" run=1
```

Program: ant04.txt otrzyma 4 argumenty -55, 243, -62 i 234. Argumenty te mogą zostać użyte przez program. Tutaj parametry są współrzędnymi (x;y) wskazującymi na punkty, pomiędzy którymi będzie chodzić mrówka:

- point nav1, nav2;
- nav1.x = cmdline(0);
- nav1.y = cmdline(1);
- nav2.x = cmdline(2);
- nav2.y = cmdline(3);

Zmienna: nav1 będzie wynosić (-55;243), zmienna: nav2 to (-62;234). Za pomocą wiersza poleceń możesz możesz testować różne programy pod kątem innej wartości ich argumentów. Możesz użyć do 10 argumentów w jednym programie.

3.2.5.2. Statek kosmiczny

```
CreateObject pos=0.00;0.00 dir=0.0 type=SpaceShip run=1
```

Statek kosmiczny nie obsługuje żadnego z programów CBOT. Niemniej komenda run może przyjąć następujące wartości :

Komenda	Akcja na początku misji
run=0	Statek rozpocznie misję z otwartym włazem
run=1	Statek wylądowuje, a następnie otworzy się właz
run=2	Statek zostanie dostarczony za pomocą żurawia (Ziemia #3)
run=3	Specjalny tryb dla zwycięstwa i porażki
run=11	Międzygwiazdna podróż, a potem lądowanie

Podczas podróży międzygwiazdnej możesz wybrać widoczne gwiazdy za pomocą komendy: Planet mode=1 (zobacz komendę: Planet w rozdziale 3.2.2).

3.2.5.3. Jajo

```
CreateObject pos=38;298 dir=1.5 type=AlienEgg autoValue1=20
autoType=AlienAnt autoString="ant10.txt" run=1
```

Jaja składane są przez królową obcych. Wykluwają się z nich obcy.

```
CreateObject ... autoValue1=20
```

Jest to czas pomiędzy złożeniem jaja, a wykluciem się z niego insekta (w sekundach).

```
CreateObject ... autoType=AlienAnt
```

Zdefiniujesz tu typ insekta, który wykluje się z jaja.

```
CreateObject ... autoString="ant10.txt"
```

Jest to program CBOT, który zostanie zaimplementowany do insekta. Podobnie jak w powyższych ćwiczeniach można zastosować prefiks: %user%\.

```
CreateObject ... run=1
```

Zdefiniujesz tu czy jajo ma być aktywne, czy też nie. Fraza: run=0 symbolizuje jajo, z którego nigdy nie wykluje się insekt.

3.2.6. Oświetlenie

```
CreateLight dir=0.0;-1.0;0.0 color=0.6;0.6;0.6 type=Terrain
```

Będziesz mógł tutaj kompleksowo zdefiniować oświetlenie występujące na planszy. `dir` to wektor kierunku ($x; z; y$). Amplituda wektora jest ignorowana. Wartość współrzędnej „z”, dla większości światła wynosi: „-1”. Wartość taka oznacza światło padające z góry na dół ($x;-1;y$)-`color` determinuje kolor światła, który jest wypadkową trzech barw: red (czerwieni), green (zieleni) i blue (niebieskiego). Generalnie wartości tych parametrów oscylują pomiędzy -1 i 2 :







Wartość	Efekt
-1.0	Absorbacja światła (zjawisko nie występujące w prawdziwym świecie)
0.0	Brak oświetlenia
1.0	Normalne oświetlenie
2.0	Przejaskrawienie

Type może przyjąć następujące wartości:

Type	Efekt
Terrain	Oświetlenie podłoża.
Object	Oświetlenie obiektów (robotów, budynków, roślin, etc.).
Quartz	Lekki kwarc (np. na planecie Krystalia) z dowolnymi ruchami światła, naśladującymi refleksy.

Poniższa tabela przedstawia różne typy oświetlenia Obiektów w kontekście robotów :



		
<pre>dir=-1;-1; 1 color=0.6;0.6;0.6 dir= 1;-1; 1 color=0.3;0.3;0.3 dir=-1;-1;-1 color=0.3;0.3;0.3 dir= 1;-1;-1 color=0.2;0.2;0.2</pre>	<pre>dir=-1;-1; 1 color=2.0;0.0;0.0 dir= 1;-1; 1 color=0.3;0.3;0.3 dir=-1;-1;-1 color=0.3;0.3;0.3 dir= 1;-1;-1 color=0.2;0.2;0.2</pre>	<pre>dir=-1;-1; 1 color=0.6;0.6;0.6 dir= 1;-1; 1 color=0.3;0.3;0.3 dir=-1;-1;-1 color=2.0;0.0;0.0 dir= 1;-1;-1 color=0.2;0.2;0.2</pre>
		
<pre>dir= 1;-1; 1 color=0.3;0.3;0.3</pre>	<pre>dir=-1;-1; 1 color=0.6;0.6;0.6</pre>	<pre>dir=-1;-1;-1 color=0.3;0.3;0.3</pre>

```
WaterColor color=-0.6;-0.1;-0.1
```

Komenda ta umożliwia edycję koloru obszaru znajdującego się pod wodą. Powyższe wartości to kolor ciemno-niebieski. Te trzy komponenty są dodawane do wszystkich światła stworzonych za pomocą komendy: CreateLight. .

Na przykład `color=-0.6;-0.1;-0.1`, pod wodą będzie wyglądał w następujący sposób :

Powietrze	Woda (-0.6;-0.1;-0.1)
<code>color= 0.63; 0.63; 0.63 type=Terrain</code>	<code>color= 0.03; 0.53; 0.53 type=Terrain</code>
<code>color=-0.70;-0.70;-0.70 type=Terrain</code>	<code>color=-1.50;-0.80;-0.80 type=Terrain</code>
<code>color= 1.40; 1.40; 1.40 type=Terrain</code>	<code>color= 0.80; 1.30; 1.30 type=Terrain</code>
<code>color= 0.56; 0.56; 0.56 type=Object</code>	<code>color=-0.04; 0.46; 0.46 type=Object</code>
<code>color= 0.32; 0.32; 0.32 type=Object</code>	<code>color=-0.28; 0.22; 0.22 type=Object</code>
<code>color= 0.32; 0.32; 0.32 type=Object</code>	<code>color=-0.28; 0.22; 0.22 type=Object</code>
<code>color= 0.16; 0.16; 0.16 type=Object</code>	<code>color=-0.44; 0.06; 0.06 type=Object</code>

```
CreateSpot pos=12;50;-75 color=1.00;-0.20;0.10 type=Terrain
```

Powyższa komenda jest niezwykle rzadko używana. Umożliwia ona oświetlenie wybranego obszaru. Zamiast tej komendy, powinieneś korzystać raczej z komendy: GroundSpot.

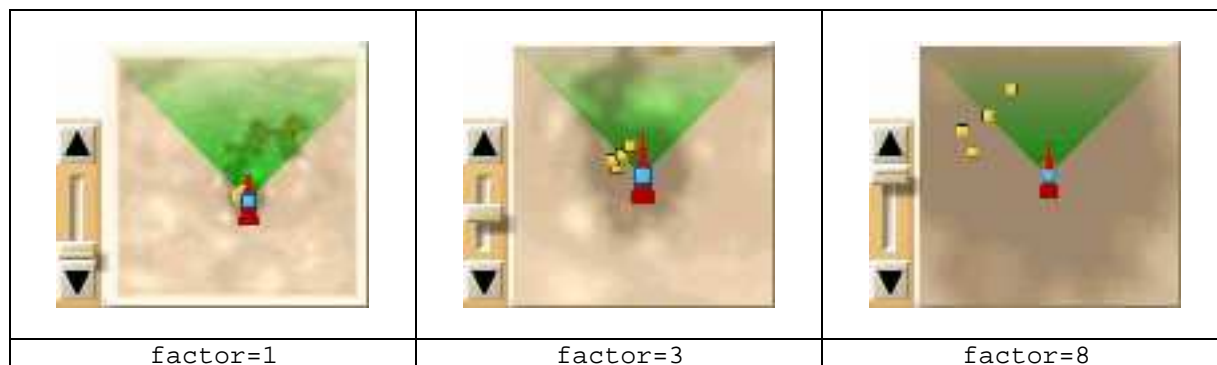
3.2.7. Mini-mapa

```
MapColor floor=104;185;205 water=154;235;255
```

Są to kolory, które zostaną użyte przez mini-mapę jako kolor podłoża i wody.

```
MapZoom factor=4
```

Jest to domyślny współczynnik przybliżenia dla mini-mapy. Wartości możliwe do wstawienia oscylują pomiędzy 1 (widoczna jest cała mapa), a 8 (najsilniejsze przybliżenie).



3.2.8. Opcje

Poniżej znajdziesz opcje determinujące możliwości działania względem danego ćwiczenia lub misji.

```
NewScript name="%user%\tower1.txt" type=WheeledGrabber
```

Dzięki tej komendzie możesz automatycznie wyposażać roboty w program: `CreateObject script1="name" type=WheeledGrabber`. Możesz też automatycznie przypisywać wybrane programy do różnego rodzaju produkowanych typów robotów.

- `type` determinuje typ robota, do którego ma zostać załadowany program. Typ `All` umożliwi Ci załadowanie danego programu do robotów każdego typu.
- `Name` jest to nazwa pliku zawierającego program.


```
EnableBuild type=PowerStation
```

Oto lista budynków, które mogą być wybudowane przez astronautę. `Type` może przyjąć jedną z następujących wartości:

<code>ResearchCenter</code>	Centrum badawcze
<code>BotFactory</code>	Fabryka robotów
<code>Converter</code>	Huta
<code>PowerStation</code>	Elektrownia
<code>RadarStation</code>	Radar
<code>RepairCenter</code>	Warsztat
<code>DefenseTower</code>	Wieża obronna
<code>PowerPlant</code>	Fabryka ogniw elektrycznych
<code>Derrick</code>	Kopalnia
<code>NuclearPlant</code>	Elektrownia atomowa
<code>AutoLab</code>	Laboratorium
<code>PowerCaptor</code>	Stacja odgromnikowa
<code>ExchangePost</code>	Stacja przekaźnikowa informacji

Lista specjalnych akcji, przydatnych dla astronauty :

<code>FlatGround</code>	Weryfikacja tego, czy grunt jest płaski
<code>Flag</code>	Dodanie/usunięcie flag



```
EnableBuild type=ResearchCenter
EnableBuild type=BotFactory
EnableBuild type=Converter
EnableBuild type=PowerStation
EnableBuild type=RadarStation
EnableBuild type=NuclearPlant

EnableBuild type=FlatGround
EnableBuild type=Flag
```

W powyższym przykładzie, przycisk elektrowni atomowej nie jest widoczny, ponieważ nie została ona jeszcze wynaleziona.

```
EnableResearch type=WINGER
```

Oto lista projektów naukowych, które mogą zostać zrealizowane w Centrum badawczym i odpowiadających im wynalazków:

TRACKER	Tracked*
WINGER	Winged*
THUMPER	Thumper
SHOOTER	*Shooter
TOWER	DefenseTower
PHAZER	PhazerShooter
SHIELDER	Shielder
ATOMIC	NuclearPlant

Oto lista projektów naukowych, które mogą zostać zrealizowane w Laboratorium:

iPAW	Legged*
iGUN	*OrgaShooter

```
DoneResearch type=WINGER
```

Oto lista projektów naukowych, które mogą być dostępne od początku ćwiczenia:

TRACKER	Tracked*
WINGER	Winged*
THUMPER	Thumper
SHOOTER	*Shooter
TOWER	DefenseTower
PHAZER	PhazerShooter
SHIELDER	Shielder
ATOMIC	NuclearPlant
iPAW	Legged*
iGUN	*OrgaShooter
RECYCLER	Recycler
SUBBER	Subber
SNIFFER	*Sniffer

Przykład

- `DoneResearch type=SUBBER`

Po zastosowaniu tej komendy, podwodny robot: Subber będzie mógł zostać wyprodukowany już z chwilą rozpoczęcia ćwiczenia.



```
EnableResearch type=TRACKER
EnableResearch type=SHOOTER
EnableResearch type=TOWER
EnableResearch type=ATOMIC
```

```
DoneResearch type=TRACKER
DoneResearch type=SHOOTER
DoneResearch type=TOWER
```

W powyższym przykładzie zrealizowanymi projektami są projekty: TRACKER, SHOOTER oraz TOWER. Do zrealizowania zaś, wciąż pozostaje projekt ATOMIC.

3.3. Warunki zakończenia misji

```
EndMissionTake pos=0;0 dist=1000 type=Me lost=0
EndMissionTake pos=0;0 dist=1000 type=WheeledGrabber lost=0
EndMissionTake pos=0;0 dist=1000 type=Titanium min=4
EndMissionTake pos=0;0 dist=1000 type=AlienAnt min=0 max=0
```

Są to kryteria, determinujące moment zakończenia się ćwiczenia. Musisz użyć osobnego wersu dla każdego typu, który chcesz przetestować.

Przykładowo, 2 wers decyduje o tym, w jakich okolicznościach ćwiczenie zakończy się niepowodzeniem, a wers 3 o tym, w jakich okolicznościach zostanie ono zakończone sukcesem.

Zapoznaj się też z informacjami o komendzie: `EndingFile`

Oto znaczenie powyższego programu:

Misja zakończy się niepowodzeniem, jeśli *jeden* z poniższych warunków będzie prawdziwy (true) :

- Liczba astronautów wyniesie 0
(`EndMissionTake pos=0;0 dist=1000 type=Me lost=0`)
- Liczba zbieraczy na kołach wyniesie 0
(`EndMissionTake pos=0;0 dist=1000 type=WheeledGrabber lost=0`)

Misja zakończy się sukcesem lub statek kosmiczny będzie mógł wystartować, jeśli *wszystkie* poniższe warunki będą prawdziwe (true) :

- Będą co najmniej 4 kawałki rudy tytanu.
(`EndMissionTake pos=0;0 dist=1000 type=Titanium min=4`)
- Będzie dokładnie zero mrówek.
(`EndMissionTake pos=0;0 dist=1000 type=AlienAnt min=0 max=0`)

```
... pos=x;y dist=radius
```

Środek okręgu, w którym znajdować się mają testowane obiekty. Fraza: `Pos=0;0 dist=1000` oznacza, iż obiekty mają być testowane na całej mapie. Możesz także określić precyzyjną pozycję obiektu, który ma zostać przetestowany. Komenda jest użyteczna do sprawdzania, czy pewne obiekty znajdują się na statku kosmicznym.

```
... type=Me
```

Typ obiektu, który ma być przetestowany (zapoznaj się z tekstem nt. komendy: `CreateObject` w celu zapoznania się z typami obiektów).

```
... lost=0
```

Jeśli liczba obiektów osiągnie ten poziom, ćwiczenie kończy się niepowodzeniem.

```
... min=4
```

Jeśli liczba obiektów jest niższa od tego limitu, to start statku jest niewykonalny (a co za tym idzie i zakończenie misji).

```
... max=0
```

Jeśli liczba obiektów jest wyższa od tego limitu, to start statku jest niewykonalny (a co za tym idzie i zakończenie misji).

3.4. Początkowa pozycja kamery

```
Camera eye=0.00;5.00;0.00 lookat=0.00;1.00;0.00 delay=0
```

Na początku misji kamera celuje w wybrany obiekt (`CreateObject select=1`). Za pomocą tej komendy możesz wybrać inną pozycję i kierunek. Kamera uda się wtedy (prędkość można zdefiniować za pomocą: `delay`) do pozycji zdeterminowanej przez wybrany obiekt.

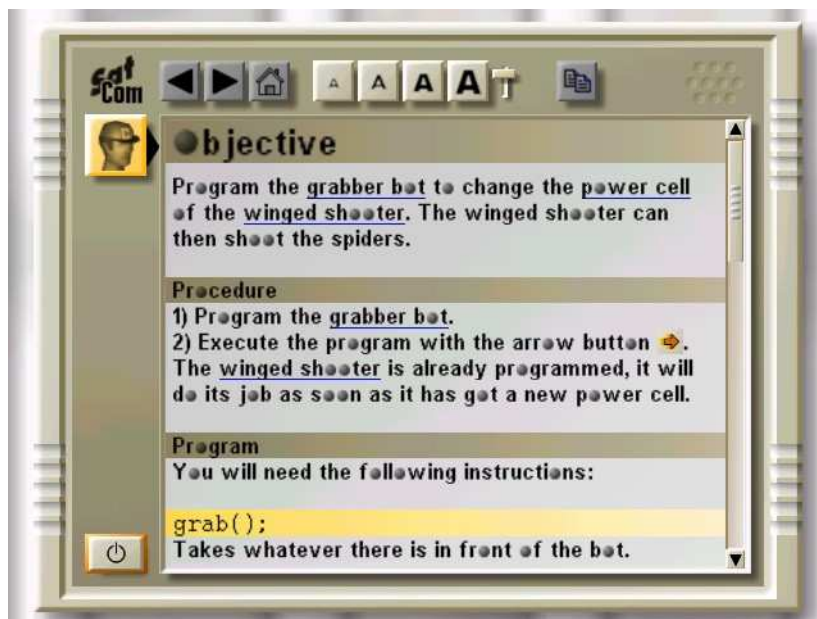
4. Formatowanie tekstu dla wiadomości SatCom

Pliki zawierające oficjalną pomoc i instrukcje Colobota, wyświetlane w **SatCom** zlokalizowane są w folderze help\. Teksty te zawierają specjalne komendy formatujące, rozpoczynające się od znaku: "\" i kończące się znakiem średnika: ";". Akapit jest ograniczony przez znak zakończenia wiersza: "¶".

Oto próbka kodu źródłowego:

```
\b;Objective¶
Program the \l;grabber bot\ u object\botgr; to change the \l;power cell\ u
object\power; of the \l;winged shooter\ u object\botfj;. The winged shooter
can then shoot the spiders.¶
¶
\t;Procedure¶
1) Program the \l;grabber bot\ u object\botgr;. ¶
2) Execute the program with the arrow button \button 21;. ¶
The \l;winged shooter\ u object\botfj; is already programmed, it will do its
job as soon as it has got a new power cell.¶
¶
\t;Program¶
You will need the following instructions:¶
\c;¶
\s;grab();\n;¶
\n;Takes whatever there is in front of the bot.¶
\c;¶
```

A tak będzie to wyglądać w **SatCom** :



\b;

Duży tytuł na brązowym tle. Komenda ta musi znajdować się na początku wiersza i kończy się automatycznie z końcem wiersza.

\t;

Napisy na brązowym tle. Komenda ta musi znajdować się na początku wiersza i kończy się automatycznie z końcem wiersza.

\s;

Napis na żółtym tle. Komenda ta musi znajdować się na początku wiersza i kończy się automatycznie z końcem wiersza. Komenda ta jest często używana do wyświetlania programów CBOT i tabel.

```
\tab;
```

Tabela na pomarańczowym tle. Tabele tego typu są często używane w raportach satelitarnych. Komenda ta musi znajdować się na początku wiersza i kończy się automatycznie z końcem wiersza.

```
\l;hypertext\u file;
```

Hiperłącze (odnośnik). Innymi słowy jest to słowo otoczone przez: `\l;i\u;`, będące nazwą linku. Komenda: `\u;` wskazuje, do którego pliku ma odnosić się link (tekst). Przykładowo, `\u object\botgr;` wskaże na standardowy plik: `help\object\botgr.txt`. Inny przykład: `\u %user%\mlink;` wskaże na konkretny plik w katalogu własnego poziomu: `user\sample01\mlink.txt`.

```
\c;
```

Jest to polecenie użycia czcionki Courier. Komenda ta jest używana w programach CBOT.

```
\n;
```

Jest to polecenie użycia normalnej czcionki.

```
\image %user%\filename dx dy;
```

Komenda ta umożliwia wyświetlenie obrazu znajdującego się w pliku graficznym: `.bmp`. Plik ten znajduje się w konkretnym folderze własnego poziomu. Jeśli prefiks `%user%` zostanie pominięty to plik ten zostanie zapożyczony z katalogu standardowego: `diagram\`.

Przykład: `\image %user%\sample01 8 8;` wyświetli plik `user\sample01\sample01.bmp`.

Przykład: `\image sniff1 12 12;` wyświetli plik `diagram\sniff1.bmp`.

`dx` to szerokość obrazka, a `dy` to wysokość obrazka. Obrazek zostanie rozciągnięty do pożądanego rozmiaru. Wartości `12 12` zaowocują obrazkiem kwadratowym.

```
\token;
```

Jest to polecenie użycia pomarańczowego tła. Komenda: `\norm;` przywraca standardowe tło.

```
\type;
```

Jest to polecenie użycia zielonego tła. Tło to jest najczęściej wykorzystywane przy typach zmiennych CBOT. Komenda: `\norm;` przywraca standardowe tło.

```
\const;
```

Jest to polecenie użycia czerwonego tła. Tło to jest najczęściej wykorzystywane przy stałych CBOT. Komenda: `\norm;` przywraca standardowe tło.

```
\key;
```

Zmiana tła na szare. Komenda: `\norm;` przywraca standardowe tło.

```
\norm;
```

Powrót do standardowego tła po zastosowaniu komend: `\token;`, `\type;`, `\const;` `\key;`.

```
\key name;
```

Komenda ta pokazuje nazwę klawisza. Np. `\key help;` wyświetli nazwę klawisza, który aktywuje **SatCom**. Domyślnie będzie to klawisz F1. Niemniej gracz ma pełną swobodę w konfigurowaniu klawiszy, w takim wypadku wyświetlony będzie symbol skonfigurowanego przez niego klawisza. Nigdy nie pisz bezpośrednio F1, używaj zamiast tego komendy `\key help;`

Oto domyślne ustawienia klawiszy:

Nazwa	Domyślny klawisz	Akcja
Left	Strzałka w lewo	Skręt w lewo
Right	Strzałka w prawo	Skręt w prawo
Up	Strzałka do góry	Ruch do przodu
Down	Strzałka w dół	Ruch do tyłu
Gup	Shift	W górę
Gdown	Control	W dół
Camera	Spacja	Zmiana widoku kamery
Desel	numpad 0	Wybór poprzedniego obiektu
Action	Enter	Domyślna akcja dla wybranego robota
Near	numpad +	Przybliżenie kamery
Away	numpad -	Oddalenie kamery
Next	Tabulator	Wybór następnego robota lub budynku
Human	Home	Wybór astronauty
Quit	Esc	Wyjście
Help	F1	Ogólne instrukcje w SatCom
Prog	F2	Pomoc programistyczna CBOT w SatCom
Cbot	F3	Pomoc odnośnie bieżących instrukcji w SatCom
Visit	numpad . (przecinek)	Wyświetlenie lokacji błędu
speed10	F4	Prędkość gry x1.0
speed15	F5	Prędkość gry x1.5
speed20	F6	Prędkość gry x2.0

Aby nazwy klawiszy wyświetlane były na szarym tle, powinieneś użyć kombinacji poniższych komend:
`\key;``\key help;``\norm;`

5. Problemy

W celu rozwiązania niektórych problemów związanych z Colobot, możliwa jest edycja pliku: `colobot.ini`, znajdującego się domyślnie w lokalizacji:

- `C:\Program Files\Colobot\colobot.ini`

Nie należy dodawać nowych wersów! Należy modyfikować już istniejące wartości. Nie wstawiaj spacji. Zachowaj szczególną uwagę przy edytowaniu zawartości tego pliku.

Jeśli wyświetlanie roślinności jest bardzo słabe lub w ogóle żadne, być może koniecznym będzie ograniczenie wyświetlania obiektów specjalnych :

- `[Setup]`
- `GadgetQuantity=1.00`
-

Jeśli roślinność wciąż nie jest widoczna, spróbuj :

- `[Engine]`
- `AlphaMode=0`

lub

- `[Engine]`
- `AlphaMode=2`

Jeśli wokół cieni pojawia się kwadrat, spróbuj:

- `[Engine]`
- `WhiteSrcBlend=9`
- `WhiteDestBlend=6`

lub

- `[Engine]`
- `WhiteSrcBlend=6`
- `WhiteDestBlend=3`
-

Jeśli powyższe działania nie przyniosą skutków, możesz w ogóle usunąć cienie:

- `[Engine]`
- `WhiteSrcBlend=0`
- `WhiteDestBlend=0`
- `[Setup]`
- `GroundShadow=0`

Obiekt znajdujący się pomiędzy wybranym obiektem, a kamerą powinien stać się przezroczysty. Jeżeli tak się nie stanie, spróbuj:

- `[Engine]`
- `StateColor=0`

lub

- `[Engine]`
- `StateColor=1`

6. Tuning

W celu odnalezienia najlepszych ustawień względem swojego komputera, możesz nakazać programowi stałe wyświetlanie wartości FPS. Aby tego dokonać, wciśnij kombinację klawiszy: `Control+Break`, wpisz komendę: `"showstat"` i zatwierdź ją klawiszem: `Enter` :

- `Ctrl+Break showstat Enter`

Po zastosowaniu powyższej komendy, w górnej części ekranu wyświetli się aktualna wartość FPS i kilka innych parametrów:

- 32.46 fps T=11558 (640x480x16)

Pierwsza liczba to FPS (fps = "klatki na sekundę"), jest to liczba obrazów wyświetlanych w czasie sekundy (im więcej tym bardziej płynne wyświetlanie grafiki). Druga liczba to liczba trójkątów obecnych na planszy. Trzy liczby w nawiasach to kolejno: rozdzielczość ekranu i paleta barw (w bitach).

Wszystkie ustawienia znajdują się w pliku: colobot.ini. Plik ten może edytowany za pomocą edytora tekstowego (np. za pomocą Notatnika).

7. Zrzeczenie się gwarancji

EPSITEC I/LUB JEGO DOSTAWCY NIE ODPOWIADAJĄ ZA POPRAWNOŚĆ DANYCH I GRAFIKI ZAWARTYCH W TYM DOKUMENCIE. INFORMACJE I GRAFIKA W NIM ZAWARTE DOSTARCZANE SĄ "TAKIE JAKIE SĄ" BEZ JAKICHKOLWIEK GWARANCJI. EPSITEC I/LUB JEGO DOSTAWCY ZRZEKAJĄ SIĘ NINIEJSZYM WSZELKIEJ ODPOWIEDZIALNOŚCI I WSZYSTKICH GWARANCJI, NIEZALEŻNIE OD TEGO CZY SĄ ONE WYRAŹNIE OKREŚLONE CZY DOMNIEMANE, W TYM, W SZCZEGÓLNOŚCI, DO: GWARANCJI W ZAKRESIE ZASTOSOWANIA I PRZYDATNOŚCI DO OKREŚLONEGO CELU. PONADTO, EPSITEC NIE STWIERDZA ANI NIE GWARANTUJE, ŻE INFORMACJE DOSTĘPNE W TYM DOKUMENCIE SĄ DOKŁADNE, PEŁNE LUB POZBAWIONE JAKICHKOLWIEK BŁĘDÓW. EPSITEC I/LUB JEGO DOSTAWCY NIE MOGĄ BYĆ ODPOWIEDZIALNI ZA SZKODY POWSTAŁE W WYNIKU UŻYWANIA TEGO DOKUMENTU LUB W POWIĄZANIU Z KORZYSTANIEM Z NIEGO. EPSITEC W ŻADNYM RAZIE NIE BĘDZIE ODPOWIEDZIALNA ZA JAKIEKOLWIEK POWSTAŁE SZKODY I KOSZTY WTÓRNE, UBOCZNE, POŚREDNIE, SZKODY SZCZEGÓLNE, ANI ODSZKODOWANIA ZA STRATY MORALNE I WSZELKIE INNE ROSZCZENIA.

8. Zespół projektowy gry

- Daniel Roux
- Denis Dumoulin
- Otto Kölbl
- Michael Walz
- Didier Gertsch

8.1. Zespół wewnętrznych beta-testerów

- Adrien Roux
- Didier Raboud
- Nicolas Beuchat
- Joël Roux
- Michael Jubin
- Daniel Sauthier
- Nicolas Stubi
- Patrick Thévoz

8.2. Wydawca

EPSITEC SA
Mouette 5
CH-1092 Belmont

colobot@epsitec.ch
www.colobot.com

8.3. Informacje o poradniku

Polska wersja poradnika została poprawiona przez administratora Polskiego Portalu COLOBOTa. Nazewnictwo zostało dostosowane do oryginalnej wersji Colobot 1.9/pl dystrybuowanej przez Mantę Multimedia. Usunięte zostały liczne błędy merytoryczne.

Polski Portal COLOBOTa:

Strona WWW - www.colobot.xt.pl

Forum – www.colobot.xt.pl/forum

e-mail administratora: pl.colobot@gmail.com