

ORIGINALWAR

Polski Spis Funkcji SAIL'a

opracowanie, tłumaczenie oraz rozwinięcie autorstwa: zoNE



Info:

W przypadku, gdy przykłady nie są pełne (z powodów objętościowych poradnika) odsyłam do przeczytania ich pełnej wersji w plikach misji o podanym numerze w przykładzie.

W poradniku znajdują się przykłady kodów użytych w misjach zawartych z grą.
Po więcej szczegółów zachęcam do zagłębienia się w całości kodu misji.

LEGENDA:

{...} => [...] w kodzie

TYPY DANYCH:

String - tekst ~2³¹ znaków (2 GB), np. nazwa jednostki lub zmienna/stała

Integer - liczby z przedziału -2147483648.. 2147483648 lub zmienne je zawierające (można stosować działania)

Boolean - wartość logiczna TRUE lub FALSE

List - lista np. jednostek określana w [], np. [lisc_1,lisc_3,lisc_2] (przy 1 obiekcie nie stosujemy nawiasów)

Real - liczba zmiennoprzecinkowa

Time - format czasu minut\$sekund liczby rzeczywiste np. 3\$0

Przedziały liczbowe oddzielamy "." np. 91..100

Podziękowania:

Serdeczne podziękowania dla: Gothuk - za przejrzanie całego dokumentu w celu wyłapania problematycznych niedociągnięć myślowych i językowych prowadzących do błędnej interpretacji.



Spis Treści:

Nawigacja list.....	004-004
Włącz/Wyłącz.....	004-006
Losowanie	006-007
Inicjalizowanie	008-008
Zapisy.....	009-012
Jednostki/Stwórz/Usuń.....	012-017
Oczekiwanie	018-018
Filtry/Zwracanie	018-021
Ekran	021-022
Dialogi.....	023-025
Medale/Wygrana.....	025-027
Obsługa zbiorów (Com).....	027-038
Obsługa zbiorów - Wstaw kolejne (AddCom)	038-050
Ustalenia/Zwracanie (Get, Set, Is, Find etc.)	051-077
Misje	078-078
Widzenie/Mgła.....	079-080
Środowisko	080-082
Rakieta Syberytowa.....	082-083
Grafika	083-083
Dźwięk.....	083-084
Przygotowanie (Prepare).....	084-086
Makra	087-095
Bazy/Flagi	096-097
Strefy	097-098
Zadania.....	098-101
Koszty/Ilości	101-102
Miny.....	102-105
Właściwości.....	105-106
Artefakty	106-107
Ataki.....	107-108
Dostępność - Fabryka	108-110
Technologie	110-111
Multiplayer.....	111-113
Inne	113-117
Prawa Autorskie	118-118



Nawigacja list:

Replace(list:plist, index:integer, item:tvalue)

zastępuje pozycję na liście o określonym indeksie na inną, zwraca wynik
jeżeli $index > n$ (n =długość listy), to pozycje $n+1$ do $index-1$ zmieniana jest na wartości niezdefiniowane

//Przykład - Misja RU 12

```
export function get_groups;
var i;
begin
{...}
for i:=1 to 5 do begin
  humans:=replace(humans,i,FilterAllUnits([[f_side,ai_s[i]],f_type,unit_human]]) diff hlavouni);
  vehicles:=replace(vehicles,i,FilterAllUnits([[f_side,ai_s[i]],f_type,unit_vehicle]]) diff art_cargo);
  buildings:=replace(buildings,i,FilterAllUnits([[f_side,ai_s[i]],f_type,unit_building]]));
end;
{...}
end;
```

Insert(list:plist, index:integer, item:tvalue)

wstawia element (item) do listy na określonej pozycję (1- przed pierwszym, n - przed n -tym), zwraca wynik
jeżeli $index > n+1$ (n =długość listy), to pozycja $n+2$ do $index-1$ zmieniana jest na wartości niezdefiniowane

//Przykład - Misja AM 15a

```
every 0$1 Marked 24 do // recharge of Show Me Map
begin
  wait(recharge_view);
  can_use_artefact:=Delete(can_use_artefact,1);
  can_use_artefact:=Insert(can_use_artefact,1,true);
  Highlight_Artefact(1);
end;
```

Delete(list:plist, index:integer)

usuwanie elementu z listy, zwraca wynik
jeżeli $index$ jest poza listą, to nie rób nic

//Przykład - Misja AM 15a

```
every 0$1 Marked 24 do // recharge of Show Me Map
begin
  wait(recharge_view);
  can_use_artefact:=Delete(can_use_artefact,1);
  can_use_artefact:=Insert(can_use_artefact,1,true);
  Highlight_Artefact(1);
end;
```

Włącz/Wyłącz:

ExclusiveOn

zatrzymuje wszystkie inne wątki SAIL'a do czasu, gdy ExclusiveOff jest spełniony lub wątek się wykona

//Przykład - Misja RU 01 cont

```
if un=Burlak then
begin
  ExclusiveOn;
  Wait(0$5);
  YouLost('Burlak');
  ExclusiveOff;
  exit;
end {...}
```

ExclusiveOff

anuluje efekt ExclusiveOn

//Przykład - Misja RU 01 cont

```
if un=Burlak then
begin
    ExclusiveOn;
    Wait(0$5);
    YouLost('Burlak');
    ExclusiveOff;
    exit;
end {...}
```

DialogueOn

funkcja, która: przechowuje upływ czasu, przełącza stop_action na true, ustawia ExclusiveOn

//Przykład - Misja AM 03

```
DialogueOn;
// interface_hidden:=true;
if IsOK(patrol1) then
    Say(patrol1,'D8-Rus1-1')
else
    Say(patrol2,'D8-Rus1-1');

Say(MacMillan,'D8-JMM-1');

// interface_hidden:=false;
DialogueOff;
```

DialogueOff

funkcja, która: przywraca upływ czasu, przełącza stop_action na false, ustawia ExclusiveOff

//Przykład - Misja AM 03

```
DialogueOn;
// interface_hidden:=true;
if IsOK(patrol1) then
    Say(patrol1,'D8-Rus1-1')
else
    Say(patrol2,'D8-Rus1-1');

Say(MacMillan,'D8-JMM-1');

// interface_hidden:=false;
DialogueOff;
```

InGameOn

funkcja, która: przechowuje upływ czasu, przełącza ingame_video na true, ustawia ExclusiveOn

//Przykład - Misja AM 07

```
EXPORT Function Action;
begin
    IngameON;
    CenterNowOnXY(48, 22);
    Prepare_base;
    SelectAmericans;
    IngameOFF;
end;
```

InGameOff

funkcja, która: przywraca upływ czasu, przełącza ingame_video na false, ustawia ExclusiveOff

//Przykład - Misja AM 07

```
EXPORT Function Action;  
begin  
    IngameON;  
    CenterNowOnXY(48, 22);  
    Prepare_base;  
    SelectAmericans;  
    IngameOFF;  
end;
```

Losowanie:

Randomize

randomizuje losowe numery SAIL'a

//Przykład - Misja RU 05 cont

```
every 0$10 do  
begin  
    Randomize;  
    enable;  
end;
```

RandomizeAll

randomizuje wszystkie losowe numery w grze

//Przykład - Misja RU 07

```
Every 1$59 do  
Begin  
    RandomizeAll;  
End;
```

Rand(min:integer, max:integer)

zwraca losowe liczby w wybranym zbiorze

//Przykład - Misja RU 06

```
case Rand(1,4) of  
    1: RaiseSailEvent(normal1_utok);  
    2: RaiseSailEvent(normal2_utok);  
    3: RaiseSailEvent(normal3_utok);  
    4: RaiseSailEvent(normal4_utok);  
end;
```

Prob(per:integer)

zwraca true, podaje per w procentach (Rand(1,100)<=per)

//Przykład - Misja AM 10

```
function get_sex;  
begin  
    if Prob(20) then result=sex_female  
        else result=sex_male;  
end;
```

MultiRand(min:integer, max:integer, n:integer)

losowa liczba w wybranym zbiorze wykonana przez n losowań Rand

//Przykład - Misja RU 07

```
Every 2$55 Marked 51 do
Begin
  Wait (MultiRand (0$1, 2$0, 2));
  yashin_start;
End;
```

RandHex(border:boolean)

zwraca koordynaty losowego hex'a w dowolnym miejscu na mapie w formacie [x,y]
jeżeli obramowanie jest true, to sąsiednie hex'y mogą zostać uwzględnione

//Przykład - Misja AM 15a

```
every 0$5.1 trigger TICK>load_global do
var c,sur,i;
begin
  c:=RandHex(false);
  sur:=[];
  for i=1 to 4 do
    sur:=sur ^ [ RandHexXYR(c[1],c[2],1,false)];
  for i in sur do
    CreateCratesXY(5,i[1],i[2],true);
  load_global:=TICK+Rand(global_delay - 0$40,global_delay + 0$60);

  enable;
end;
```

RandHexArea(area:integer, border:boolean)

zwraca koordynaty losowego hex'a w wybranym obszarze w formacie [x,y] (lub 0 jeżeli false)
jeżeli obramowanie jest true, to sąsiednie hex'y mogą zostać uwzględnione

//Przykład - Misja RU 10 cont

```
function create_minefield;
var i,xx;
begin
  mines_list:=[];
  for i=1 to (difficulty*2) do
    begin
      xx:=RandHexArea(minefield_north,false);
      PlaceMine(xx[1],xx[2],enemy,0);
      mines_list:=mines_list ^ xx;
    wait(1);
  end;
```

RandHexXYR(x:integer, y:integer r:integer, border:boolean)

zwraca koordynaty losowego nie nieprawidłowego hex'a w okolicy xy (max. macierzysta jest R) w formacie [x,y] (lub 0 jeżeli false)
jeżeli border jest true, to hex'y obramowania mogą zostać uwzględnione

//Przykład - Misja AM 15

```
For Un in Units_Outside do
begin
  Hex= RandHexXYR(56, 69, 2, false);
  ComMoveXY(Un, Hex[1], Hex[2]);
  AddComHold(Un);
end;
```

Inicjalizowanie:

InitUc

Inicjalizuje jednostkę przywracając domyślne wartości

//Przykład - Misja AM 06 cont

```
if CheckCharacterSet('Denis') then begin
  InitUc;
  InitHc;
  uc_side:=you2;
  uc_nation:=nation_american;
  Denis:=CreateCharacterWithClass('Denis', class_scientistic);
  DialogDenis;
end;
```

InitHc

Inicjalizuje człowieka przywracając domyślne wartości

//Przykład - Misja AM 06 cont

```
if CheckCharacterSet('Denis') then begin
  InitUc;
  InitHc;
  uc_side:=you2;
  uc_nation:=nation_american;
  Denis:=CreateCharacterWithClass('Denis', class_scientistic);
  DialogDenis;
end;
```

InitVc

Inicjalizuje pojazd przywracając domyślne wartości

//Przykład - Misja RU 12

```
masha_def:=LoadVariable('Masha', []);
if masha_def then begin
  InitVc;
  vc_chassis:=masha_def[1];
  vc_engine:=masha_def[2];
  vc_control:=masha_def[3];
  vc_weapon:=masha_def[4];
  masha:=CreateVehicle;
end;
```

InitBc

Inicjalizuje budynek przywracając domyślne wartości

//Przykład - Misja RU 05 cont

```
function place_base(builds);
var i, base;
begin
  InitUc;
  InitBc;
  {...}
  for i in builds do begin
    bc_type:=i[4];
    if bc_type in [b_lab,b_lab_half,b_lab_full] then begin
      bc_kind1:=i[5];
      bc_kind2:=i[6];
    end;
  end;
  {...}
end;
```


Zapisy:

SaveForQuickRestart

używa się go po filmach ingame i wyborze postaci - szybki restart rozpocznie się od tego punktu

```
//Przykład - Misja AM 07
```

```
EVERY 0$1 DO
begin
wait(0$3);
SaveForQuickRestart;
ChangeMissionObjectives('M1');
wait(0$5);
Say(Harrison, 'D2-Har-1');
end;
```

NewCharacter(ident:string)

Ładuje i stworzy nową globalną postać (która nie była nigdy wcześniej zapisana), zwraca jej uchwyt

```
//Przykład - Misja AM 13a
```

```
if debug then begin
uc_side=oth;
changesidefog(oth,al);
jahoda=newcharacter('Belkov');
end;
```

PrepareNewCharacter(ident:string)

Przygotowuje struktury hc i uc dla nowej globalnej postaci (która nie była nigdy wcześniej zapisana)

```
//Przykład - Misja RU 07
```

```
Function zkopiruj_xaviera_jako_rus;
Var Xav, X, Y, Dir, InUnit;
Begin
{...}
PrepareNewCharacter('Xavier');
UC_Side = side_Ru; UC_Nation = NATION_RUSSIAN;
UC_Placed = false;
{...}
End;
```

TestVariable(ident:string)

Zwraca true, jeżeli tu jest zmienna z określonym ident'em z zapisu

```
//Przykład - Misja AM 13a
```

```
joan=0;
if checkcharacter('Joan') then begin
if testvariable('joan_side') then tmp=loadvariable('joan_side',-1)
else buildquery('neny joan_side',['OK']);
if tmp=old_al then joan=createcharacter('Joan');
placeunitxjr(joan,62,25,3,false);
end;
```

TestCharacters(ident:string)

Zwraca true, jeżeli tu jest postać lub grupa postaci z określonym ident'em z zapisu

```
//Przykład - Misja AM 15a
```

```
if TestCharacters('Petrosyan') and CheckCharacterSet('Petrosyan') then
begin
Petrosyan:=CreateCharacter('Petrosyan');
aliance_forces:=aliance_forces ^ Petrosyan;
end;
```

CheckCharacterSet(ident:string)

Zwraca numer postaci z zasobu postaci z zapisu
(jeżeli zbior jest pusty w zapisie, CheckCharacterSet zwraca 0, ale TestCharacter zwraca true)

//Przykład - Misja AM 13a

```
joan=0;
if checkcharacterSet('Joan') then begin
  if testvariable('joan_side') then tmp=loadvariable('joan_side',-1)
  else buildquery('neny joan_side',['OK']);
  if tmp=old_al then joan=createcharacter('Joan');
  placeunitxyr(joan,62,25,3,false);
end;
```

LoadVariable(ident:string, default:tvalue)

Ładuje zmienną z określonym ident'em z zapisu (lub domyślnym, jeżeli ident nie został znaleziony)

//Przykład - Misja AM 15a

```
function prepare_ar_units;
begin
  OmarKilled:=LoadVariable('OmarKilled',false);
  NegotiationOffered:=LoadVariable('NegotiationOffered',false);
  if TestCharacters('Omar') and CheckCharacterSet('Omar') then
    Omar:=CreateCharacter('Omar');

  if Omar and NegotiationOffered then
    Enable(2)
  else
    Enable(3);

  SetSide(Omar,ar_al);
end;
```

CreateCharacterSet(ident:string)

Ładuje zasób postaci z określonym ident'em z zapisu
Tworzy ludzi z niego (oprócz martwych postaci) używając tej samej strony uc_side i zwraca zbiór ich uchwytów

//Przykład - Misja RU 04

```
function nacti_osobu(ident);
begin
  if TestCharacters(ident) then
    if CheckCharacterSet(ident)=1 then
      result:=CreateCharacter(ident)
    else
      result:=CreateCharacterSet(ident)
  else
    result:=0;
end;
```

CreateOneCharacterFromSet(ident:string, index:integer)

Tak samo jak CreateCharacterSet, ale tylko jedna jednostka jest tworzona (jeden z określonego indexu w zasobie)

{patrz przykład: **CreateCharacterSet**}

CreateOneCharacterFromSetWithClass(ident:string, index:integer, cl:integer)

Tak samo jak CreatePeopleFromCharacterSet, ale postać dostaje określona klasę

{brak przykładu}

CreateCharacter(ident:string)

Skrót dla CreateOneCharacterFromSet(ident,1)

//Przykład - Misja AM 15a

```
if TestCharacters('Kozlov') and CheckCharacterSet('Kozlov') then
begin
    Kozlov:=CreateCharacter('Kozlov');
    alianca_forces:=alianca_forces ^ Kozlov;
end;
```

CreateCharacterWithClass(ident:string, cl:integer)

Skrót dla CreateOneCharacterFromSetWithClass(ident,1,cl)

//Przykład - Misja AM 12a cont

```
uc_side=al;
if checkcharacterset('JMM') then john=createcharacterwithclass('JMM',class_soldier)
else buildquery('neny JMM,['OK']);
```

PrepareOneCharacterFromSet(ident:string, index:integer)

Tak samo jak CreateOneCharacter, ale postać jest tylko przygotowywana (nie tworzona jeszcze), zwraca true jeżeli się uda

{brak przykładu}

PrepareCharacter(ident:string)

Skrót do PrepareOneCharacterFromSet(ident,1)

{brak przykładu}

SaveCharacters(humlist:plist, ident:string)

Zapisuje zbiór ludzkich postaci do wyjściowego zapisu i nadaje mu określony ident

W przypadku stosowania postaci specjalnych:

SaveCharacters(handle,name) w jednej misji oraz handle:=CreateCharacter(name) w następnej misji

Dla ludzi, którzy przeżyli z poprzednich wykorzystanych misji

SaveCharacters(set,name) w jednej misji oraz set:=CreateCharacterSet(name) w następnej misji

//Przykład - Misja AM 14a

```
every 0$1 trigger not (IsOK(MacMillan) or IsOK(Burlak)) do
begin
    SaveCharacters(MacMillan,'JMM');
    SaveCharacters(Burlak,'Burlak');

    SaveVariable(JMM_car,'JMM_car');
    SaveVariable(Bur_car,'Bur_car');
    SaveVariable(JMM_load,'JMM_car_load');
    SaveVariable(Bur_load,'Bur_car_load');

    ChangeMap('%_cont','_%cont');
end;
```

SaveVariable(v:tvalue, ident:string)

Zapisuje zmienną do zapisu z określonym identem

//Przykład - Misja RU 02

```
case stav_utoku of
0:SaveVariable(3,'utok_stav');
1:SaveVariable(1,'utok_stav');           // zautocil spravne
2:SaveVariable(2,'utok_stav');           // zautocil spravne, ale s posilama
3:SaveVariable(3,'utok_stav');           // zautocil bez rozkazu
end;
```

DeleteCharacters(ident:string)

Usuwa zasób postaci z zapisu

//Przykład - Misja AM 06 cont

```
DeleteCharacters('opicaci');
DeleteCharacters('other_from5to6');
```

DeleteVariable(ident:string)

Usuwa zasób postaci z zapisu

//Przykład - Misja RU 10 cont

```
if lost_masha then
DeleteVariable('Masha');
```

Jednostki/Stwórz/Usuń:

CharacterSelection(ident:string, min:integer, max:integer, units:plist classes:plist)

Wyświetla menu wyboru postaci i zwraca listę postaci zaznaczonych przez gracza

- ident jest identyfikatorem napisu z Texts.txt

- min i max to minimalna i maksymalna liczba wybranych postaci do zamknięcia dialogu

- units jest listą jednostek, które mogą posiadać specjalne polecenia, które mają wpływ na wszystkie następujące jednostki:

sel_hired - dane postacie są wybrane na początku (domyślnie)

sel_not_hired - dane postacie nie są wybrane na początku

sel_changeable - gracz może ostrzelać/wybrać daną postać

sel_not_changeable - gracz nie może ostrzelać/wybrać daną postać (domyślnie)

sel_change_class - gracz może zmienić klasę danej postaci

sel_dont_change_class - gracz nie może zmienić klasę danej postaci (domyślnie)

sel_ignore_class_nation - dana postać może mieć zmienioną klasę zależnie od narodu

sel_dont_ignore_class_nation - dana postać może być zmieniona tylko na klasę z dango narodu (domyślnie)

- classes jest listą możliwych klas

is list of possibly classes, w które mogą być zmienione jednostki. Wszystkie elementy mogą być w dwóch formatach:

class_number - ta klasa może być używana bez ograniczeń

[class_number,min] - ty musisz zaznaczyć przynajmniej min postaci w tej

[class_number,min,max] - ty musisz zaznaczyć min do max postaci w tej klasie

Example1 (Wybierz maksymalnie pięciu mechaników i żołnierzy z gamma2 and gamma3):

```
CharacterSelection('S1',1,6,
[JMM, sel_not_hired, sel_changeable] ^ gamma2 ^ gamma3,
[class_soldier, class_mechanic])
```

Example2 (Wybierz maksymalnie pięciu mężczyzn z gamma2 do pomocy John i Tim w budowie bazy)

```
CharacterSelection('S2',3,7,
[sel_changeable, JMM, sel_not_changeable, Tim, sel_not_hired, sel_changeable] ^ gamma2,
[class_soldier, class_mechanic, [class_engineer,1], class_scientistic])
```

//Przykład - Misja AM 07

Function SelectAmericans;

begin

// Select your attack group of 10 persons

```
Selected_americans= CharacterSelection(", 9, 9,
[sel_hired, sel_not_changeable, sel_change_class, JMM,
sel_not_hired, sel_changeable, sel_dont_ignore_class_nation,
sel_change_class] ^ All_americans ^ New_americans,
[[class_mechanic, 3, 5], class_soldier, class_scientistic]);
```

```
Selected_americans= Selected_americans ^ [JMM];
```

```
all_americans= all_americans ^ [JMM] ^ new_americans;
```

end;

RewardPeople(humans:plist)

nadaje nagrodę po misji dla listy ludzkich jednostek (używane przed zapisem)

//Przykład - Misja AM 13a

```
every 0$1 trigger koncime or base_out or artef_out do
begin
//if islive(abdul) then wait(0$30); //time to kill abdul
if (getside(masha)=al) or (getside(masha)=oth) then addmedal('Masha',1)
else addmedal('Masha',-1);
if isdead(abdul) then addmedal('Abdul',1)
else addmedal('Abdul',-1);
if isdead(abduluv) then addmedal('Mast',1)
else addmedal('Mast',-1);
givemedals('Main');
rewardpeople(filterallunits([[f_alive],[f_side,al],[f_type,unit_human],f_not,[f_nation,nation_nature]]));
saveitall;
youwin;
end;
```

CreateHuman

Tworzy człowieka używając parametrów uc_ i hc_

//Przykład - Misja RU 07

```
Function vytvor_yelenu;
Begin
UC_Nation = NATION_RUSSIAN; UC_Side = side_Ru2;
HC_name = 'Yelena Trajkov'; HC_sex = SEX_FEMALE;
HC_class = CLASS_SOLDIER; PrepareSoldierSkills (5);
Result = CreateHuman;
End;
```

CreateVehicle

Tworzy pojazd używając parametrów uc_ i vc_

//Przykład - Misja RU 07

```
Function posila_tank;
Begin
UC_Side = side_Ar; UC_Nation = NATION_ARABIAN;
VC_Chassis = AR_MEDIUM_TRIKE; VC_Control = CONTROL_MANUAL;
VC_Engine = ENGINE_COMBUSTION; VC_Weapon = nahoda_seznam ([AR_FLAME_THROWER, AR_ROCKET_LAUNCHER,
AR_MULTIMISSION_BALLISTA]);
VC_Fuel_Battery = Rand (85, 100);
Result = CreateVehicle;
PlaceHumanInUnit (vytvor_cloveka (CLASS_MECHANIC), Result);
End;
```

CreateBuilding

Tworzy budynek używając parametrów uc_ i bc_

//Przykład - Misja RU 05 cont

```
function place_base(builds);
var i, base;
begin {...}
for i in builds do begin
bc_type:=i[4];
if bc_type=b_lab or bc_type=b_lab_half or bc_type=b_lab_full then begin
bc_kind1:=i[5];
bc_kind2:=i[6];
end;
uc_direction:=i[3];
PlaceUnitXY(CreateBuilding, i[1],i[2], false);
end;
{...} end;
```

CreateAndPlaceBuildingXYD(x:integer, y:integer, direction:integer)

Tworzy budynek używając parametrów uc i bc

//Przykład - Misja RU 05 cont

```
function place_base(builds);
var i, base;
begin
  InitUc;
  InitBc;
  uc_side:=arabians;
  uc_nation:=nation_arabian;

  for i in builds do begin
    bc_type:=i[4];
    if bc_type in [b_lab,b_lab_half,b_lab_full] then begin
      bc_kind1:=i[5];
      bc_kind2:=i[6];
    end;
    CreateAndPlaceBuildingXYD(i[1],i[2],i[3]);
  end;
  base:=GetBase(FilterAllUnits([f_btype,b_depot])[1]);
  SetResourceType(base, mat_cans, 450);
  SetResourceType(base, mat_oil, 200);
end;
```

PlaceUnitXY(un:integer, x:integer, y:integer, materialisation:boolean)

Umieszcza jednostkę na określonych koordynatach, zwraca true jeżeli udane
Jeżeli jest materializowanie jest true, postać jest zmateriałizowana

//Przykład - Misja RU 04

```
every 0$1 marked 15 do // Killing Gaydar
begin
  Wait(Gaydar_wait);
  if NOT IsDead(Gaydar) AND sm_pocet3<attack_count then
  begin
    sm_pocet3:=sm_pocet3+1;
    PlaceUnitXY(sm_attack3[sm_pocet3],19,2,false);
    ComMoveToArea(sm_attack3[sm_pocet3],PaloukArea);
    Wait(10);
    AddComAttackUnit(sm_attack3[sm_pocet3],Gaydar);
    enable;
  end
  else
  disable;
end;
```

PlaceUnitArea(un:integer, area:integer, materialisation:boolean)

Umieszcza jednostkę w określonym obszarze, zwraca true jeżeli udane
Jeżeli jest materializowanie jest true, postać jest zmateriałizowana

//Przykład - Misja AM 13 cont

```
every 9$0 do
begin
  setdir(2,burlakv);
  placeunitarea(burlakv,cestasever,false);
  placehumaninunit(burlak,burlakv);
  // commovexy(burlakv,106,118);
  comaggressivemove(burlakv,106,118);
  enable(10);
end;
```

PlaceUnitXYR(un:integer, x:integer, y:integer, r:integer, materialisation:boolean)

Umieszcza jednostkę blisko określonych współrzędnych (max. macierzysta jest r), zwraca true jeżeli udane
Jeżeli jest materializowanie jest true, postać jest zmaterializowana

//Przykład - Misja RU 06

```
every 20$0 do
var a, i;
begin
  InitUc;
  InitHc;
  uc_nation:=nation_american;
  uc_side:=americans;
  for i:=1 to 5 do begin
    PrepareMechanic(0, 5);
    hc_class:=class_soldier;
    a:=CreateHuman;
    PlaceUnitXYR(a, 21,2,5, false);
    ComMoveXY(a, 25,32);
    sol:=sol ^ a;
  end;
end;
```

PlaceUnitAnywhere(un:integer, materialisation:boolean)

Umieszcza jednostkę w dowolnym miejscu na mapie, zwraca true jeżeli udane
Jeżeli jest materializowanie jest true, postać jest zmaterializowana

{brak przykładu}

PlaceHumanInUnit(hum:integer, un:integer)

Umieszcza nieumieszczoną osobę na miejsce umieszczonej lub nieumieszczonej jednostki typu budynek/pojazd

//Przykład - Misja RU 12a cont

```
for un in scientists do
  PlaceHumanInUnit(un,2); // 2 => number of placed lab in UnitList
```

LinkVehicleToHuman(veh:integer, hum:integer)

Łączy pojazd z człowiekiem w Wieży kontrolnej

//Przykład - Misja AM 15

```
EXPORT Function LinkBombs;
begin
  ComEnterUnit(Control_Driver, ArControl);
  Wait(1);
  WaitTask(Control_Driver, 1);
  LinkVehicleToHuman(ArBomb1, Control_Driver);
  LinkVehicleToHuman(ArBomb2, Control_Driver);
end;
```

CreateCratesXY(amount:integer, x:integer, y:integer, materialisation:boolean)

Tworzy skrzynki na danych współrzędnych x,y. Jeżeli materialisation = true to materializuje.

//Przykład - Misja RU 02

```
EVERY 0$10 TRIGGER stav=3 DO
VAR hex;
begin
  wait(Rand(0$30,1$0));
  hex=RandHexArea(bedny_ar,true);
  while HexInfo(hex[1],hex[2])<>0 do hex=RandHexArea(bedny_ar,true);
  CreateCratesXY(4,hex[1],hex[2],true);
  bedny=bedny^[hex];
end;
```

CreateCratesArea(amount:integer, area:integer, materialisation:boolean)

Tworzy skrzynki na danej arei. Jeżeli materialisation = true to materializuje.

//Przykład - Misja AM 10

```
EVERY 1$0 TRIGGER Dial>0 DO
begin
  wait(Rand(1,7200));
  CreateCratesArea(Rand(3,5),Base2,true);
  enable;
end;
```

CreateCratesXYR(amount:integer, x:integer, y:integer, r:integer, materialisation:boolean)

Tworzy skrzynki na danych koordynatach x,y (max. macierzysta jest R). Jeżeli materialisation = true to materializuje.

//Przykład - Misja AM 10

```
function bedny1;
begin
  CreateCratesXYR(Rand(3,5),47,48,3,true);
end;
```

CreateCratesAnywhere(amount:integer, materialisation:boolean)

Tworzy skrzynki w losowym miejscu.

//Przykład - Misja AM 10

```
EVERY 2$0 DO
begin
  wait(Rand(1,7200));
  CreateCratesAnywhere(Rand(3,5),true);
  enable;
end;
```

CreateResourcesXY(typ:integer amount:integer, x:integer, y:integer, materialisation:boolean)

Tworzy zasoby określonego typu na danych koordynatach x,y. Jeżeli materialisation = true to materializuje.

//Przykład - Babylon

```
EffectTeleport(hex[1],hex[2]);
wait(30);
CreateResourcesXY(mat_artefact,artefact_size[1],hex[1],hex[2],false);
```

CreateResourcesArea(typ:integer, amount:integer, area:integer, materialisation:boolean)

Tworzy zasoby określonego typu na danej arei. Jeżeli materialisation = true to materializuje.

//Przykład - Misja AM 11 cont

```
EVERY 0$3 TRIGGER GetSide(A2_dep)=You or IsDead(A2_dep) DO
VAR sci1;
begin
  wait(0$1);
  CreateResourcesArea(mat_artefact,5,Art_zone,false);
  {...}
  A2_dobyta=true;
end;
```

CreateResourcesXYR(typ:integer, amount:integer, x:integer, y:integer, r:integer, materialisation:boolean)

Tworzy zasoby określonego typu na danych koordynatach x,y (max. macierzysta jest R). Jeżeli materialisation = true to materializuje.

//Przykład - Misja RU12

```
if not h then
  CreateResourcesXYR(mat_artifact, 4, GetX(un), GetY(un),5, false)
else {...}
```


CreateResourcesAnywhere(typ:integer, amount:integer, materialisation:boolean)

Tworzy zasoby określonego typu w określonym miejscu. Jeżeli materialisation = true to materializuje.

{brak przykładu}

CreateDepositXY(x:integer, y:integer, typ:integer)

tworzy dostawę typu mat siberite oraz mat oil na określonych koordynatach

//Przykład - Misja AM 10

```
case nah of
  1,2:CreateDepositXY(150,80,mat_oil);
  3:CreateDepositXY(91,109,mat_oil);
  4:CreateDepositXY(43,71,mat_oil);
end;
```

RemoveDepositXY(x:integer, y:integer)

usuwa dostawę wszelkiego typu na wybranych koordynatach

//Przykład - Babylon

```
function init_map;
var i;
begin // if not required extra oil deposits, then destroy them
  if not def_extra_oil_deposits then
    for i in extra_oil_deposits_locations do
      RemoveDepositXY(i[1],i[2]);
    end;
end;
```

RemoveUnit(un:integer)

Usuwa określoną jednostkę z mapy

//Przykład - Misja AM 15

```
begin
  Say(Roth, 'DA-Roth-3a');
  RemoveUnit(Captured_american);
  True_americans= True_americans ^ Captured_american;
end;
```

DestroyUnit(un:integer)

Niszczy określoną jednostkę (obejmuje usuwanie, jeżeli umieszczone)

//Przykład - Misja AM 12

```
every 0$1 trigger isinarea(kurt_veh,odjezd) marked 10 do
begin
  disable(99);
  destroyunit(kurt_veh);
  if legion=-1 then setattitude(am,ar,att_enemy,true);
end;
```

KillUnit(un:integer)

Zabij określoną jednostkę przez ustawienie jej życia na zero

//Przykład - Misja AM 10 cont

```
if IsLive(Cornel) then KillUnit(Cornel);
```

Oczekiwanie:

Wait(ticks:integer)

czekaj na określoną liczbę chwil, ignoruje tryb asynchroniczny

//Przykład - Misja RU 12

```
if not under_attack[i] then begin
  HledejBudovyNaOprawy(i,buildings[i]);
  wait(0$0.6);
end;
```

DWait(ticks:integer)

czekaj na określoną liczbę chwil, do użycia gdy DialogOn

//Przykład - Misja RU 05 cont

```
if vespod and qr=2 then begin
  DialogueOn;
  DWait(0$4.5);
  DialogueOff;
  vespod2:=true;
end;
```

Filtry/Zwracanie:

FilterAllUnits(filter:plist)

Filtruje listę wszystkich jednostek przy użyciu filtru

//Przykład - Misja RU 06

```
every 3 do
begin
  sol:=FilterAllUnits([[f_side,ai],[f_class,class_soldier]]) ;

  dep:=FilterAllUnits([[f_side,ai],[f_or,[f_btype,b_depot],[f_btype,b_warehouse]]]) ;
  eng:=FilterAllUnits([[f_side,ai],[f_class,class_engineer]]) ;

  fact:=FilterAllUnits([[f_side,ai],[f_or,[f_btype,b_factory],[f_btype,b_workshop]]]) ;
  ctrl:=FilterAllUnits([[f_side,ai],[f_btype,b_control_tower]]) ;
  mec:=FilterAllUnits([[f_side,ai],[f_class,class_mechanic]]) ;

  lab:=FilterAllUnits([[f_side,ai],[f_or,[f_btype,b_lab],[f_btype,b_lab_half],[f_btype,b_lab_full]]]) ;
  sci:=FilterAllUnits([[f_side,ai],[f_class,class_scientistic]]) ;
  sib_dul:=FilterAllUnits([[f_side,ai],[f_btype,b_siberite_mine],[f_distxy,101,93,5]]);
  if sib_dul then
    sib_dul:=sib_dul[1];
end;
```

FilterUnitsInArea(area:integer, filter:plist)

Filtruje listę jednostek w określonym obszarze przy użyciu filtru

//Przykład - Misja AM 11 cont

```
EVERY 0$1 TRIGGER Odebirani DO
begin
  pom=FilterUnitsInArea(ExitAR,[[f_side,Arabi]]);
  for i in pom do RemoveUnit(i);
  if UnitFilter(zradci,[[f_ok]])>0 then enable
    else Odebirani=false;
end;
```

FilterUnitsExceptArea(area:integer, filter:plist)

Filtruje przy użyciu filtru listę jednostek nie będących w określonym obszarze

//Przykład - Misja RU 03

```
while pom>0 do
begin
pom1=FilterUnitsExceptArea(ODBER_AM2,[[f_side,Amici]]);
for i in pom1 do
begin
RemoveUnit(i);
pom=pom diff i;
end;
wait(0$1);
end;
```

UnitFilter(units:plist, filter:plist)

Filtruje listę jednostek (units) przy użyciu filtru

//Przykład - Misja AM 12

```
every 0$1.13 do
var tmp,tmq;
begin
tmp=unitfilter([ruso0,ruso1,ruso2],[[f_ok]]) diff filterunitsinarea(southdef,[[f_side,rus],[f_ok]]);
if tmp then comagressivemove(tmp,95,42);
enable;
end;
```

NearestUnitToXY(units:plist, x:integer, y:integer)

Zwraca najbliższą jednostkę z listy jednostek (units) do pozycji (x,y) lub zero jeżeli units jest puste

//Przykład - Misja RU 11 cont

```
every 0$1+0$0.2 marked 09 do
begin
if devet then enable;
centeronunits(nearestunittoxy(vehiklys,74,54));
end;
```

NearestUnitToUnit(units:plist, un:integer)

Zwraca najbliższą jednostkę z listy jednostek (units) do jednostki un (ignoruje własne un) lub zero

//Przykład - Misja RU 03

```
if vozKirovOut>0 then
begin
veh:=NearestUnitToUnit(vozKirovOut,mech);
ComEnterUnit([mech],veh);
end;
```

AllNearestUnitToUnit(units:plist, un:integer)

Zwraca wszystkie najbliższe jednostki z listy jednostek (units) do jednostki un (ignoruje własne un) lub zero

//Przykład - Misja RU 10 cont

```
every 0$1 trigger See(you,artefact) and not gossudarov_freed do
var un,
list, // seznam zijicich (neumirajicich) ruskych vojaku
nearest; // vojak nejblyze k artefaktu
begin {...}
list:=FilterAllUnits([ [f_side,you], [f_sex,sex_male] ]) diff Burlak;

nearest:=AllNearestUnitToUnit(list,artefact);
{...} end;
```

SortListByListAsc(items:tvalue, values:tvalue)

Zwraca listę elementów (items) posortowaną rosnąco wg towarzyszących liczb całkowitych listy wartości (values)
Jeżeli listy nie są tej samej wielkości, tylko mniejszej długości lista jest używana

//Przykład - Misja RU 05 cont

```
every 0$10 + 0$8 do
var zivoty,i;
begin
  opravit:=[];
  for i in FilterAllUnits([[f_side,arabians],[f_type,unit_building]]) do begin
    if GetLives(i)<950 then begin
      opravit:=opravit ^ i;
      zivoty:=zivoty ^ GetLives(i);
    end;
  end;
  opravit:=SortListByListAsc(opravit, zivoty);
  enable;
end;
```

SortListByListDesc(items:tvalue, values:tvalue)

Zwraca listę elementów (items) posortowanych malejąco wg towarzyszących liczb całkowitych listy wartości (values).
Jeżeli listy nie są tej samej wielkości, tylko mniejszej długości lista jest używana

//Przykład - Misja RU 05 cont

```
function get_x_best(num, skill);
var i, lidi, skills;
begin
  lidi:=FilterAllUnits([[f_side,arabians],[f_type,unit_human]]);
  skills:=[];
  result:=[];
  for i in lidi do
    skills:=skills ^ GetSkill(i, skill);
  SortListByListDesc(lidi, skills);
  for i:=1 to num do
    result:=result ^ lidi[i];
  end;
```

WorstFromListByList(items:tvalue, values:tvalue)

Zwraca SortListByListAsc(items,values)[1] lub zero jeżeli values jest puste. Szybsze od SortListByListDesc

//Przykład - Misja RU 15

```
every 0$7 trigger lesni_opoce marked 09 do
var tmp,tmp1;
begin
  tmp1=[];
  for tmp in lesni_opoce_ do tmp1=tmp1^[getdistunitxy(tmp,134,133)];
  tmp=worstfromlistbylist(lesni_opoce_,tmp1);
  comtamexy([lesc_1,lesc_3,lesc_2],getx(tmp),gety(tmp));
  addcommoveunit([lesc_1,lesc_3,lesc_2],lelabf01);
  wait(rand(0$30,5$0));
  enable;
end;
```

BestFromListByList(items:tvalue, values:tvalue)

Zwraca SortListByListDesc(items,values)[1] lub zero jeżeli values jest puste. Szybsze od SortListByListAsc

//Przykład - Misja AM 13 cont

```
if not tmp then begin
  tmp=filterallunits([[f_side,am],[f_ok],[f_not],[f_nation,nation_nature]]) diff john;
  tmp1=[];
  for tmp2 in tmp do tmp1=tmp1^[getskill(tmp2,skill_scientistic)];
  tmp=bestfromlistbylist(tmp,tmp1);
end {...}
```

SetAttitude(fromside:integer, toside:integer, att:integer, sym:boolean)

określa stosunek att między dwiema stronami (jeżeli sym, to stosunki są też symetryczne)

//Przykład - Misja RU 08

```
begin
  D_HeikeKilled;
  SetLives(Heike,0);
  HeikeKilled:=true;
  SetAttitude(you,merc,att_enemy,true);
end;
```

GetAttitude(fromside:integer, toside:integer)

zwraca stosunek pomiędzy stronami

//Przykład - Misja AM 15

```
Function SetExplorerWayNumber;
var WayNumber;
begin
  if GetAttitude(russians, Explorers_Side) = att_enemy then
    begin
      Explorer_Way_Number= Rand(1,4);
      if Explorer_Way_Number > 1 then Explorer_Way_Number= 2;
    end
  else
    Explorer_Way_Number= Rand(1,2);
  end;
```

SetAlliedVictory(side:integer, state:boolean)

Określa stan zwycięstwa sojusznika dla strony

{brak przykładu}

GetAlliedVictory(side:integer)

zwraca stan zwycięstwa sojusznika po stronie

//Przykład - Alien base

```
if (GetAttitude(i,j)=att_friend) and (GetAttitude(j,i)=att_friend) and
  GetAlliedVictory(i) and GetAlliedVictory(j) then
  vysledek:=true;
```

Ekran:

CenterOnXY(x:integer, y:integer)

Ekran skupia się na określonych koordynatach z przewijaniem

//Przykład - Misja AM 11

```
// capture the base
EXPORT function chmo1;
begin
  CenterOnXY(86,145);
  SetAttitude(You,Arabi,att_enemy,true);
  SetAttitude(Arabi,You,att_enemy,true);
  SaveForQuickRestart;
  wait(0$1);
  ChangeMissionObjectives('M1');
  InGameOff;
end;
```

CenterOnUnits(units:plist)

Ekran skupia się na określonej grupie jednostek (units) z przewijaniem

//Przykład - Misja AM 06

```
function Dialog1;
var i;
begin
{...}
  InGameOn;
{...}
  used:=[JMM] union CharacterSelection('1', 4, 4,
    [sel_change_class,JMM,sel_not_hired]^[Sci,Denis]^[sel_changeable]^(all diff [Sci,JMM,Denis]),
    [class_soldier, class_mechanic, [class_engineer,1], [class_scientistic,1,2]]);

  ComMoveXY(used, 66,75);
  ComMoveXY(UnitFilter(UnitsInside(d), [f_nation,nation_nature]), 66,75);
  CenterOnUnits(used);
  wait(0$4);
  CenterOnUnits(used);
  wait(0$4);
  InGameOff;
end;
```

CenterNowOnXY(x:integer, y:integer)

Ekran skupia się bezpośrednio na określonych koordynatach

//Przykład - Misja AM 07

```
EXPORT Function Action;
begin
  IngameON;
  CenterNowOnXY(48, 22);
  Prepare_base;
  SelectAmericans;
  IngameOFF;
end;
```

CenterNowOnUnits(units:plist)

Ekran skupia się bezpośrednio na określonej grupie jednostek (units)

//Przykład - Misja RU 09 cont

```
on UnitGoesToRed(un) do
begin
  if un=Joan then
  begin
    DialogueOn;
    CenterNowOnUnits(Joan);
    ForceSay(Joan,'DJoanDies-Joan-1');
    SayRadio(JMM,'DJoanDies-JMM-1');
    wait(0$1);
    SetLives(Joan,0);
    KillUnit(Joan);
    DialogueOff;
    wait(wait_mcmillanova_pomsta[difficulty]);
    mcmillanova_pomsta=true;
  end;
  enable;
end;
```

Dialogi:

Say(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident

//Przykład - Misja AM 01 cont

```
every 0$2 trigger IsInArea(Bobby,TrapEnd)or IsInArea(Cyrus,TrapEnd)or IsInArea(JMM,TrapEnd) marked 33 do
begin
    Say(JMM,'D4b-JMM-1');
    SetAreaMapShow(Shw2,0);
    ChangeMissionObjectives('M3c');
end;
```

SayNoFace(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident, jej twarz nie pojawia się

//Przykład - Misja AM 01 cont

```
if cyr and bob then
begin
    async;
    Say(Cyrus,'D6-Cyrus-1');
    sync;
    wait(0$0.3);
    SayNoFace(Bobby,'D6-Bobby-1');
end {...}
```

SayNoName(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident, jej nazwa nie pojawia się - używane dla wrogów

//Przykład - Misja RU 03

```
EXPORT function dialog_scholtze_umrel;
VAR i,pom,pom1;
begin
    pom=0;
    DialogueOn;
    interface_hidden:=true;
    {...}
    pom1=NearestUnitToUnit(pom,schol2);
    if pom1 <> 0 then SayNoName(pom1,'D4b-Sol1-1');
    pom1=NearestUnitToUnit(pom diff pom1,pom1);
    SayNoName(pom1,'D4b-Sol2-1');
    {...}
    interface_hidden:=false;
    DialogueOff;
end;
```

ForceSay(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident nawet jeśli jest umierająca (ale nie martwa)

//Przykład - Misja AM 15

```
begin
    if GetSex(Captured_american) = sex_male then
        ForceSay(Captured_american, 'DA1-Sol1-1')
    else ForceSay(Captured_american, 'DA1-FSol1-1');
end;
```

ForceSayNoName(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident nawet jeśli jest umierająca, jej nazwa nie pojawia się

{brak przykładu}

ForceSayNoFace(un:integer, ident:string)

Określona jednostka mówi zdanie wskazane przez ident nawet jeśli jest umierająca, jej twarz nie pojawia się

//Przykład - Misja AM 11

```
wait(0$2);
SetLives(mrtvola,40);
wait(0$0.5);
ForceSayNoFace(mrtvola,'D1a-Merc1-2');
KillUnit(mrtvola);
```

SayRadio(un:integer, ident:string)

Określona postać mówi przez radio zdanie wskazane przez ident, używane dla jednostek poza mapą

//Przykład - Misja AM 13a

```
case query("QSaveBelkov") of
1:begin
  dialogueon;
  say(john,'D5a-JMM-1');
  sayradio(jahoda,'D5a-Bel-1');
  dialogueoff;
end; {...}
```

SayRadioNoName(un:integer, ident:string)

Określona postać mówi przez radio zdanie wskazane przez ident, jej nazwa nie pojawia się

//Przykład - Misja AM 01 cont

```
SayRadioNoName(Powell,'D1-Pow-1');
Say(JMM,'D1-JMM-2');
SayRadioNoName(Powell,'D1-Pow-2');
ComMoveXY(JMM,80,77);
```

SayEffect(ident:string)

Efekt mówienia wskazany przez ident

//Przykład - Misja AM 06 cont

```
2: begin
  Say(kdo, 'D2b-JMM-1');
  Say(Kurt, 'D2b-Kurt-1');
  Say(kdo, 'D2b-JMM-2');
  Say(Kurt, 'D2b-Kurt-2');
  SayEffect('Kurtfire.wav');
end;
```

Query(ident:string)

Wyświetla zapytanie dialogu określonego w tekście, zwraca numer odpowiedzi (1..n)

//Przykład - Misja AM 14

```
if Query('Q1')=2 then
begin
  DialogueOff;
  interface_hidden:=false;
  exit;
end;
```


SelectiveQuery(ident:string, list:plist)

Wyświetla zapytanie dialogu określonego w tekście, zwraca numer odpowiedzi (1..n)

Lista zawiera numery elementów, które zostaną pokazane

//Przykład - Misja AM 15

```
if not(selection in [5, 6]) then
begin
selection= SelectiveQuery('Q1a', [1, 2]);
ResolveQuery(selection+4, list_of_q);
end;
```

Medale/Wygrana:

AddMedal(ident:string, number:integer)

jeżeli number>0 to dodaj [ident,number] element do gained_medals

jeżeli number<0 to dodaj [ident,-number] element do missing_medals

//Przykład - Misja AM 12

```
case telepmed of
0:addmedal('Teleport',-1);
1:addmedal('Teleport',-2);
2:addmedal('Teleport',1);
end;
```

GiveMedals(ident:string)

Nadanie medalu

//Przykład - Misja RU 05 cont

```
export function Vyhra;
begin
set_medals;
GiveMedals('Main');
RewardPeople(FilterAllUnits([[f_side,you],[f_type,unit_human],[f_nation,nation_russian]]));
Save;
YouWin;
end;
```

YouWin

Wygrałeś

//Przykład - Misja RU 05 cont

```
export function Vyhra;
begin
set_medals;
GiveMedals('Main');
RewardPeople(FilterAllUnits([[f_side,you],[f_type,unit_human],[f_nation,nation_russian]]));
Save;
YouWin;
end;
```

YouLost(ident:string)

Przegrałeś dla określonego ident'a

//Przykład - Misja AM 07

```
// EOM - JMM is death
EVERY 0$5 TRIGGER IsDead(JMM) do
begin
YouLost('JMM');
ExitGame;
end;
```

YouDidSomethingExtraordinary

Podobnie jak YouWin, lecz nie nadaje wygranej. Można użyć do zrobienia innej alternatywnej prawidłowej drogi, która jednak nie prowadzi do wygranej.

//Przykład - Misja AM 01

```
1:begin
{...}
  ingameoff;
  YouDidSomethingExtraordinary;
//  youwin('Saved');
end;
```

YouWinInMultiplayer

Wygrałeś dla trybu multiplayer

//Przykład - Alien base

```
if win then
begin
  wait(0$3);
  YouWinInMultiplayer;
  exit;
end;
```

YouLostInMultiplayer

Przegrałeś dla trybu multiplayer

//Przykład - Alien base

```
if CanWinLoose then
if not FilterAllUnits([[f_side,your_side],[f_or,[f_type,unit_human],[f_btype,b_depot],[f_btype,b_warehouse]]]) then
begin
  wait(0$3);
  YouLostInMultiplayer;
  exit;
end {...}
```

MultiplayerSideAlive(side:integer)

zwraca true, jeżeli istnieje co najmniej jeden grający gracz na tej stronie

//Przykład - Alien base

```
for i=1 to Side_positions do
if Side_positions[i] and (i<>your_side) and MultiplayerSideAlive(i) then
begin
  if not Side_Teams[i] or (Side_Teams[i]<>Side_Teams[your_side]) then
  win:=false;
end;
```

SetTag(index:integer, value:integer)

umieszcza tag[index] dla wartości (value)

//Przykład - Misja RU 06

```
a1:=place_amici(aici1,abase1);
for i in a1 do
  SetTag(i, 1);
a2:=place_amici(aici2,abase2);
for i in a2 do
  SetTag(i, 2);
a3:=place_amici(aici3,abase3);
for i in a3 do
  SetTag(i, 3); {...}
```

GetTag(index:integer)

otrzymuje wartość z tag[index]

//Przykład - Babylon

```
every 0$1 Marked 3 do
var side;
begin
win:=[];
for side=1 to 8 do
if GetTag(side) >= final_count then
win:=win ^ side;
{...}
end;
```

Obsługa zbiorów (Com):

ComMoveXY(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie przejścia do określonych koordynatów

//Przykład - Misja AM 07

```
WaitJob(har_veh, 0.5);
ComExitVehicle(Harrison);
Wait(1);
ComMoveXY(Harrison, 49, 15);
AddComTurnUnit(Harrison, Guard1);
```

ComMoveUnit(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie przejścia do określonej jednostki

//Przykład - Misja AM 14

```
if IsLive(ru_collector) then
begin
ComMoveUnit(ru_collector, ru_depot);
AddComExitVehicle(ru_collector);
ComReturn(free_unit);
end {...}
```

ComMoveToArea(units:plist, area:integer)

zbiór jednostek otrzymuje polecenie przejścia do określonej jednostki strefy

//Przykład - Misja AM 09 cont

```
case ara1_faze of
0: ComMoveXY(Ara1, 58,76);
1: ComPlaceRemoteCharge(Ara1, 0,0, ruDepot1);
2: ComMoveXY(Ara1, 57,76);
3: begin ComMoveToArea(Ara1, arab_exit);
AddComMoveXY(Ara1,37,72);
end;
end;
```

ComAgressiveMove(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie agresywnego przejścia do określonych koordynatów

//Przykład - Misja AM 01 cont

```
if see(russians,JMM) then
begin
enable;
ComAgressiveMove(Bobby,GetX(JMM),GetY(JMM));
exit;
end;
```

ComAttackUnit(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie ataku na określoną jednostkę

//Przykład - Misja RU 12a cont

```
while sc_utok AND IsOK(Rus_Cargo) do
begin
  ComAttackUnit(sc_utok,Artifact_cargo);
  Wait(0$2);
  for un in sc_utok do
    if NOT IsOK(un) then
      sc_utok:=sc_utok diff un;
  end;
```

ComAttackPlace(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie ataku na określone koordynaty

//Przykład - Misja AM 14

```
function DestroyTarget(target,x,y);
var un;
begin
  for un in attack_forces do
    if GetWeapon(un) in [ru_rocket,ru_rocket_launcher] then
      ComAttackPlace(un,x,y)
    else
      ComAttackUnit(un,target);
end;
```

ComCollect(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie zbierania na określonych koordynatach

//Przykład - Misja AM 14

```
if IsLive(ru_collector) then
  ComCollect(un,crate_list[1],crate_list[2])
else
  ComGet(un,crate_list[1],crate_list[2]);
```

ComTurnXY(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie odwrócenia się do określonych koordynatach

//Przykład - Misja AM 07

```
if Guard1 then
begin
  PlaceUnitXY(Guard1, 50, 17, false);
  ComTurnXY(Guard1, 53, 24);
  Guards= Guards ^ Guard1;
end;
```

ComTurnUnit(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie odwrócenia się do określonej jednostki

//Przykład - Misja RU 10 cont

```
InGameOn;
wait(0$2);
ComStop(Gossudarov);
ComTurnUnit(Gossudarov,Burlak);
ComTurnUnit(Burlak,Gossudarov);
CenterOnUnits(Gossudarov);
Say(Gossudarov,'DArtefact-Gos-1');
InGameOff;
```

ComEnterUnit(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie wejścia do określonego pojazdu/budynku

//Przykład - Misja RU 06

```
for i in blds do
  ComEnterUnit(UnitFilter(hums, [[f_side,Getside(i)],[f_dist, i, 30]]), i);
```

ComExitVehicle(units:plist)

zbiór jednostek otrzymuje polecenie wyjścia z pojazdu

//Przykład - Misja AM 05

```
while UnitsInside(car1)^UnitsInside(car2) do begin
  ComExitVehicle([car1, car2]);
  wait(0$1);
end;
```

ComExitBuilding(units:plist)

zbiór jednostek otrzymuje polecenie wyjścia z budynku

//Przykład - Misja AM 15

```
AddMcUnits(Arabian_build_id, Arabian_Engineers);
ComExitBuilding(Arabian_Engineers);
```

ComChangeProfession(units:plist, new_prof:integer)

zbiór jednostek otrzymuje polecenie zmiany profesji na new_prof

//Przykład - Misja AM 10

```
EVERY 0$10 marked 5 do
begin
  ComChangeProfession(ar_eng,class_scientistic);
  AddComChangeProfession(Afi[1],class_engineer);
  AddComExitBuilding(Afi[1]);
  AddComBuild(Afi[1],b_armoury,80,28,0);
  enable(14);
end;
```

ComResearch(units:plist, techn:integer)

zbiór laboratoriów otrzymuje polecenie opracowania technologii techn

//Przykład - Misja AM 10 cont

```
EVERY 0$45 TRIGGER GetTech(tech_Sib1,Arabi)=state_researched DO
begin
  wait(1$0);
  ComResearch(Ar_lab,tech_Sib2);
end;
```

ComConstruct(units:plist, chassis:integer, engine:integer, control:integer, weapon:integer)

zbiór fabryk otrzymuje polecenie stworzenia pojazdu z określonymi komponentami

//Przykład - Misja AM 15

```
if Powell_Attack_Counter = 2 then
begin
  Powell_Attack_Counter= 3;
  // one vehicle more (for Powell)
  ComConstruct(AmFact, us_medium_tracked, engine_solar, control_manual, us_light_gun);
  hex= RandHexArea(Am1_parking, true);
  ComMoveXY(driver, hex[1], hex[2]);
  AddComExitVehicle(driver);
  AddComEnterUnit(driver, AmFact);
end
```

ComPause(units:plist)

zbiór budynków otrzymuje polecenie zatrzymania badań lub tworzenia

{brak przykładu}

ComCancel(units:plist)

zbiór budynków otrzymuje polecenie anulowania badań lub tworzenia

//Przykład - Misja AM 15a

```
if GetTag(base+12) then
begin
  ComCancel(GetTag(base+11));
  wait(5);
  labs_in_use:=labs_in_use diff GetTag(base+11);
  SetTag(base+12,false);
end;
```

ComHeal(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie uleczenia określonej jednostki

//Przykład - Misja AM 06

```
export function zachranka(un);
begin
  if not FilterAllUnits([[f_not,[f_side,ai]], [f_dist,un, 24]]) then begin
    ComHeal(UnitFilter(sci, [[f_dist,un,30]]), un);
    AddComReturn(UnitFilter(sci, [[f_dist,un,30]]));
  end;
end;
```

ComRepairVehicle(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie naprawy określonego pojazdu

//Przykład - Misja AM 01 cont

```
if GetLives(rveh1)<900 then
begin
  ComMoveXY(rveh1,136,88);
  AddComExitVehicle(rveh1);
  repeat
    wait(0$0.5);
  until not IsDrivenBy(rveh1);
  setFuel(rveh1,100);
  ComRepairVehicle(rmech1,rveh1);
end;
```

ComRepairBuilding(units:plist, un:integer)

zbiór jednostek otrzymuje polecenie naprawy określonego budynku

//Przykład - Misja RU 06

```
every 0$3 trigger not under_attack do
begin
  if opravit then begin
    if opravit[1] in can_repair then
      ComRepairBuilding(UnitFilter(eng, [f_dist, opravit[1], (5-GetLives(opravit[1])div 200)* 7]), opravit[1]);
  end;
enable;
end;
```

ComTameXY(units:plist, x:integer, y:integer)

zbiór jednostek otrzymuje polecenie osławiania

//Przykład - Misja RU 15

```
every 0$7 trigger lesni_opoce marked 09 do
var tmp,tmp1;
begin
tmp1=[];
for tmp in lesni_opoce_ do tmp1=tmp1^[getdistunitxy(tmp,134,133)];
tmp=worstfromlistbylist(lesni_opoce_,tmp1);
comtamexy([lesc_1,lesc_3,lesc_2],getx(tmp),gety(tmp));
addcommoveunit([lesc_1,lesc_3,lesc_2],lelabf01);
wait(rand(0$30,5$0));
enable;
end;
```

ComPlaceDelayedCharge(units:plist, x:integer, y:integer, un:integer)

zbiór jednostek otrzymuje polecenie podłożenia materiałów wybuchowych na [x,y] lub w pobliżu jednostki un

{brak przykładu}

ComPlaceRemoteCharge(units:plist, x:integer, y:integer, un:integer)

zbiór jednostek otrzymuje polecenie podłożenia zdalnych ładunków wybuchowych na [x,y] lub w pobliżu jednostki un

//Przykład - Misja AM 11 cont

```
ComPlaceRemoteCharge(A3_zas1,103,19,0);
AddComMoveXY(A3_zas1,104,3);
AddComCrawl(A3_zas1);
AddComTurnXY(A3_zas1,102,8);
```

ComFireExplosives(units:plist)

zbiór jednostek otrzymuje polecenie odpalenia zdalnie sterowanych ładunków wybuchowych

//Przykład - Misja AM 12

```
every 0$0.5 trigger filterunitsinarea(minefield,[[f_side,am],[f_type,unit_human]]) marked 16 do
begin
comfireexplosives(filterallunits([[f_side,ar],[f_type,unit_human]]));
end;
```

ComLinkTo(units:plist, hum:integer)

zbiór pojazdów otrzymuje polecenie połączenia z określonymi osobami

//Przykład - Misja AM 11

```
ComEnterUnit(A3_rme1,A3_rv1);
ComEnterUnit(A3_rme2,A3_rv2);
ComEnterUnit(A3_rme3,A3_rv3);
wait(0$5);
ComLinkTo([A3_are1,A3_are2,A3_are3],A3_rme1);
ComLinkTo([A3_are4,A3_are5,A3_are6],A3_rme2);
ComLinkTo([A3_are7,A3_are8,A3_are9],A3_rme3);
```

ComUnlink(units:plist)

zbiór pojazdów otrzymuje polecenie odłączenia od mechaników

//Przykład - Misja AM 15a

```
if (GetSide(un)=us) and (GetControl(un)=control_remote) then
begin
ComUnlink(un);
check_remote(0);
end;
```

ComCrawl(units:plist)

zbiór jednostek otrzymuje polecenie czołgania

//Przykład - Misja RU 08

```
every 0$1+0$0.3 marked 102 do // Hunter crawl and walk + Hunting done
begin
  if GetDistUnits(hunter,WantsToAttack(hunter)) < 13 then
    ComCrawl(hunter)
  else
    ComWalk(hunter);
  if NOT HasTask(hunter) then
    disable
  else
    enable;
end;
```

ComWalk(units:plist)

zbiór jednostek otrzymuje polecenie chodzenia

//Przykład - Misja AM 02

```
if IsOk(Cyrus) then
begin
  ComWalk(Cyrus);
  ComMoveXY(Cyrus, 27, 16);
  AddComTurnXY(Cyrus, 27, 15);
end;
```

ComFree(units:plist)

zbiór jednostek otrzymuje polecenie zwolnienia pozycji

//Przykład - Misja AM 02

```
EVERY 0$1 TRIGGER GetLives(RusEM1) < 1000 DO
begin
  if not(IsBusy(RusEM1)) then ComFree(RusEM1);
end;
```

ComHold(units:plist)

zbiór jednostek otrzymuje polecenie utrzymywania pozycji

//Przykład - Misja AM 12a

```
placeunitxy(john,115,14,false);
placeunitxy(roth,118,14,false);
comhold([john,roth]);
```

ComStop(units:plist)

zbiór jednostek otrzymuje polecenie zatrzymania się

//Przykład - Misja AM 03

```
ComStop([patrol1,patrol2]);
AddComMoveXY([patrol1,patrol2],84,71);
AddComWait([patrol1,patrol2],[0$6,0$4,0$2][difficulty]); // Maly oddech prchajaci ruske hlidky
AddComMoveXY([patrol1,patrol2],80,67);
AddComMoveXY([patrol1,patrol2],45,22);
```


ComWait(units:plist, time:integer)

zbiór jednostek otrzymuje polecenie oczekiwania

//Przykład - Misja AM 05 cont

```
arabs_cars:=FilterAllUnits([[f_type, unit_vehicle],[f_side, arabians]]);
if arabs_cars then begin
  if GetWeapon(arabs_cars[1])<>us_cargo_bay then begin
    ComWait(arabs_cars, 0$12);
    AddComMoveToArea(arabs_cars, border);
  end
else
  ComMoveToArea(arabs_cars, border);
end;
```

ComRemember(units:plist)

zbiór jednostek otrzymuje polecenie zapamiętania pozycji

//Przykład - Misja RU 12

```
if nearest then begin
  units:=attack_units isect vehicles[index_strany];
  ComRemember(units);
  AddComAgressiveMove(units, GetX(nearest), GetY(nearest));
  AddComReturn(units);
end;
```

ComReturn(units:plist)

zbiór jednostek otrzymuje polecenie odwrotu

//Przykład - Misja AM 14

```
if IsLive(ru_collector) then
  begin
    ComMoveUnit(ru_collector,ru_depot);
    AddComExitVehicle(ru_collector);
    ComReturn(free_unit);
  end {...}
```

ComBuild(units:plist, bud:integer, x:integer, y:integer, dir:integer)

zbiór jednostek otrzymuje polecenie budowania

//Przykład - Misja AM 04

```
every 0$5+2$0 trigger bunker_done do
  begin
    comfree(ruhengi0);
    combuild(ruhengi1,b_oil_mine,42,54,0);
  end;
```

ComUpgrade(units:plist)

lista jednostek otrzymuje polecenie rozbudowy

//Przykład - Misja AM 15

```
// Upgarade of the new factory
ComUpgrade(GetUnitNumber(78, 12));
repeat
  Wait(1);
until GetBType(GetUnitNumber(78, 12))=b_factory;
ComRepairBuilding(Kozlov ^ Arabian_Engineers, GetUnitNumber(78, 12));
```

```
WaitTask(Kozlov, 1);
```

ComUpgradeLab(units:plist, labtype:integer)

lista jednostek otrzymuje polecenie ulepszenia laboratorium na specjalistyczne

//Przykład - Misja AM 10

```
// upgrade sib. laboratore
EVERY 0$10 marked 3 DO
begin
  e1:=FilterUnitsInArea(Abase,[[f_type,unit_building],[f_and],[f_btype,b_solar_power]]);
  e1:=e1[1];
  AddComMoveXY(ar_eng[2],75,20);
  ComUpgradeLab(lab,b_lab_siberium);
  enable(4);
end;
```

ComPlaceWeapon(units:plist, weap:integer)

lista wieżyczek i bunkrów otrzymuje polecenie zamontowania określonego rodzaju broni

//Przykład - Misja AM 04

```
every 0$6+4$20 trigger build_wrks and Isinunit(ruhmech1)=rubwrks0 do
begin
  commoveunit(ruhengi0,rubware0);
  complaceweapon(rubbunk1,ru_heavy_machine_gun);
  {...}
end;
```

ComAnim(units:plist, atype:integer)

lista jednostek otrzymuje polecenie określonej animacji

{brak przykładu}

ComRefuel(units:plist, un:integer)

lista jednostek otrzymuje polecenie tankowania

//Przykład - Misja AM 09 cont

```
export ArmStarted;
export BarStarted;
function odnes_jeden_oil;
var a;
begin
  {...}
  ComStop(Ara2);
  ComHold(Ara2);
  SetCargo(Ara2, mat_oil, 10);
  a:=GetFuel(AraCar1);
  while GetFuel(AraCar1)<=a do begin
    ComRefuel(Ara2, AraCar1);
    wait(0$1);
  end;
```

ComTransport(units:plist, un:integer, mattype:integer)

lista jednostek otrzymuje polecenie transportowania

//Przykład - Misja AM 15

```
for i= 1 to Number_of_crates do
begin
  if i=1 then ComTransport(Carriers, AmDep, mat_cans)
  else AddComTransport(Carriers, AmDep, mat_cans);
  AddComMoveXY(Carriers, Cans_Positions[i][1], Cans_Positions[i][2]);
  AddComUnload(Carriers);
end;
```

ComInvisible(units:plist)

lista jednostek otrzymuje polecenie niewidzialności

//Przykład - Misja AM 12a cont

```
cominvisible(roth);  
// addcomhold(roth);
```

ComSpaceShift(units:plist, x:integer, y:integer)

lista jednostek otrzymuje polecenie użycia rakiet czaso-przestrzennych

//Przykład - Misja AM 12

```
function registruj_dvojicku(baz,en);  
begin  
  dvojicky:=dvojicky^[[baz,en]];  
  ComSpaceShift(baz,GetX(en)+1,GetY(en)+1);  
  registrovani:=registrovani union en;  
  free_baz:=free_baz diff baz;  
end;
```

ComTimeShift(units:plist, x:integer, y:integer)

lista jednostek otrzymuje polecenie użycia rakiet spowalniających

//Przykład - Misja AM 12

```
if att_began and attackers then comtimeshift([baz0,baz1,baz2,baz3,baz4,baz5],87,67);
```

ComHack(units:plist, un:integer)

lista jednostek otrzymuje polecenie hack

{brak przykładu}

ComTeleportExit(units:plist, x:integer, y:integer, un:integer)

lista jednostek otrzymuje polecenie umieszczenia wyjścia z teleportu un na [x,y]

{brak przykładu}

ComHiddenCamera(units:plist, x:integer, y:integer)

lista jednostek otrzymuje polecenie umieszczenia ukrytej kamery

//Przykład - Misja RU 09

```
ComEnterUnit(Arabi_enter[1],Ar_lab);  
ComPlaceRemoteCharge(diet,78,47,0);  
AddComEnterUnit(diet,Ar_bar);  
ComPlaceRemoteCharge(Arabi_enter[2],78,43,0);  
AddComEnterUnit(Arabi_enter[2],Ar_bu1);  
ComHiddenCamera(Arabi_enter[3],64,38);  
AddComHiddenCamera(Arabi_enter[3],85,52);  
AddComHiddenCamera(Arabi_enter[3],101,51);  
AddComEnterUnit(Arabi_enter[3],Ar_dep);  
ComPlaceRemoteCharge(Arabi_enter[4],82,46,0);  
AddComEnterUnit(Arabi_enter[4],Ar_bu2);
```

ComContaminate(units:plist, x:integer, y:integer)

lista jednostek otrzymuje polecenie zanieczyszczenia syberyty

//Przykład - Misja AM 15

```
Wait(0$5);  
ComContaminate(Omar_Scientists, 177, 69);
```

ComUnload(units:plist)

lista jednostek otrzymuje polecenie wyładowania ładunku

{brak przykładu}

ComGet(units:plist, x:integer, y:integer)

lista jednostek otrzymuje polecenie podniesienia ładunku

//Przykład - Misja RU 12a cont

```
every 0$1 marked 7 do // Loss Artefact checking
begin
  if FindArtifact(4) then
  begin
    ComGet(Rus_Cargo,FindArtifact(4)[1],FindArtifact(4)[2]);
    Wait(3);
    AddComMoveToArea(Rus_Cargo,EnterArea);
  end;
  enable;
end;
```

ComGive(units:plist, un:integer)

lista jednostek otrzymuje polecenie przekazania ładunku

{brak przykładu}

ComCarabine(units:plist)

lista szeków otrzymuje polecenie CarabineOn (zmień na karabin)

//Przykład - Misja RU 05 cont

```
every 0$2 trigger IsInUnit(Omar)=mastodon do
begin
  ComCarabine(Omar);
  enable(99);
end;
```

ComSabre(units:plist)

lista szeków otrzymuje polecenie SabreOn (zmień na szablę)

//Przykład - Misja RU 05 cont

```
every 0$2 trigger not IsInUnit(Omar)=mastodon marked 99 do
var mytick;
begin
  ComSabre(Omar);
  wait(0$1);
  mytick:=tick+0$5;
  {...}
end;
```

ComSailEvent(units:plist, num:integer)

jednostka wywołuje losowe zdarzenie SailEvent(num)

{brak przykładu}

ComStand(units:plist)

zbiór jednostek otrzymuje polecenie stój

//Przykład - Misja AM 07 cont

```
if Frank then ComStand(opofrank);
```

ComAttackSoporific(units:plist, un:integer)

atakuj jednostki un nasenną amunicją

//Przykład - Misja RU 09

```
EVERY 0$5 TRIGGER IsOK(Am_snipe) DO
VAR u,sl;
begin
sl=false;
if Attacks(Am_snipe)<0 then
begin
u=Attacks(Am_snipe);
if GetType(u)<>unit_vehicle
then sl=true
else if GetControl(u)=control_manual then sl=true;
if sl then if Prob(45) then ComAttackSoporific(Am_snipe,u);
end;
enable;
end;
```

ComDismantle(units:plist, un:integer)

lista jednostek otrzymuje polecenie demontażu budynku un

//Przykład - Misja RU 06

```
on Contact(side1, side2) do
var a;
begin
{...}
while IsLive(sib_dul) and IsLive(aIeng) do begin
a:=GetLives(sib_dul)-25*difficulty;
if a>1 then
SetLives(sib_dul, a);
ComDismantle(aIeng, sib_dul);
wait(0$1);
end;
{...}
end;
```

ComRecycle(units:plist, un:integer)

lista pojazdów otrzymuje polecenie do ponownego wykorzystania w budynku un

{brak przykładu}

ComLinkToBase(units:plist, un:integer)

magazyn lub skład otrzymuje polecenie do połączenia budynku un do bazy

{brak przykładu}

ComBuildBehemoth(units:plist, bud:integer, x:integer, y:integer, dir:integer)

zbiór jednostek otrzymuje polecenie budowania behemota

//Przykład - Misja AM 15

```
for Un in Behemot_Engineers do
begin
ComBuildBehemoth(Un, b_behemoth, Behemot_Places[i][1], Behemot_Places[i][2], 5);
i= i+1;
end;
```

ComForceInvisible(units:plist)

lista jednostek otrzymuje komendę niewidzialności

{brak przykładu}

Obsługa zbiorów - Wstaw kolejne (AddCom):

AddComMoveXY(units:plist, x:integer, y:integer)

zbiór jednostek otrzyma polecenie ruszenia do określonych koordynatów

//Przykład - Misja AM 03

```
CenterOnUnits([patrol1,patrol2]);

wait(200-(difficulty*45));

ComStop([patrol1,patrol2]);
AddComMoveXY([patrol1,patrol2],84,71);
AddComWait([patrol1,patrol2],[0$6,0$4,0$2][difficulty]); // Maly oddych prchajjici ruske hlidky
AddComMoveXY([patrol1,patrol2],80,67);
AddComMoveXY([patrol1,patrol2],45,22);
```

AddComMoveUnit(units:plist, un:integer)

zbiór jednostek otrzyma polecenie do ruszenia do określonej jednostki

//Przykład - Misja AM 05

```
function dialogue;
var qr, sol1, i, mytick, mechs, sols;
begin
{...}
  CenterOnXY(48,14);
  sol1:=WhoSayAny(nic_nerikaji, you, sex_male, 0, 0);
  AddComMoveUnit([Lisa, Frank, sol1], JMM);
  AddComMoveXY(Gladstone,46,15);
{...}
end;
```

AddComMoveToArea(units:plist, area:integer)

zbiór jednostek otrzyma polecenie ruszenia do określonej strefy

//Przykład - Misja RU 05

```
on DestinationUnreachable(un) do
begin
  ComStop(un);
  wait(0$0.5);
  AddComMoveToArea(un, ru_exit);
end;
```

AddComAgressiveMove(units:plist, x:integer, y:integer)

zbiór jednostek otrzyma polecenie agresywnego ruszenia na określone koordynaty

//Przykład - Misja AM 03

```
ComAgressiveMove([patrol1,patrol2],130,68);
AddComAgressiveMove([patrol1,patrol2],124,102);
AddComAgressiveMove([patrol1,patrol2],66,100);
wait(0$15);
```

AddComAttackUnit(units:plist, un:integer)

zbiór jednostek otrzyma polecenie atakowania określonej jednostki

//Przykład - Misja RU 08

```
every 0$15 marked 23 do // Heike reaction on MercHandedOver (kill'em all)
var filter,un;
begin
  ComAttackUnit(Heike,Burlak);
  filter:=FilterAllUnits([[f_side,you],[f_type,unit_human]]);
  for un in filter do
  begin
    AddComAttackUnit(Heike,un);
    Wait(10);
  end;
end;
enable;
end;
```

AddComAttackPlace(units:plist, x:integer, y:integer)

zbiór jednostek otrzyma polecenie atakowania na określonych koordynatach

//Przykład - Misja AM 15

```
EXPORT Function RSiberiteAttack(Rocket);
var American_units, Selected_unit, Factories, Teleport1, Teleport2, Un;
begin
{...}

  AddComAttackPlace(Rocket, GetX(Selected_unit), GetY(Selected_unit));
end;
```

AddComCollect(units:plist, x:integer, y:integer)

zbiór jednostek otrzyma polecenie zbierania na określonych koordynatach

//Przykład - Misja RU 15a

```
export function remove_bad_crates(side);
var list,un,x,y;
begin
{...}

  list:=FilterAllUnits([ [f_side,side], [f_type,unit_human], [f_class,class_engineer], [f_ok] ]);

  for un in list do
  if not GetUnitMc(un) then
  begin
    ComRemember(un);
    AddComCollect(un,x,y);
    AddComReturn(un);
  end;
end;
```

AddComTurnXY(units:plist, x:integer, y:integer)

zbiór jednostek otrzyma polecenie odwrócenia się do określonych koordynatach

//Przykład - Misja AM 02

```
CenterOnXY(27, 15);
if IsOk(John) then
begin
  ComWalk(John);
  ComMoveXY(John, 24, 14);
  AddComTurnXY(John, 27, 15);
end;
```

AddComTurnUnit(units:plist, un:integer)

zbiór jednostek otrzyma polecenie odwrócenia się do określonej jednostki

//Przykład - Misja RU 12b

```
addcomturnunit([platonov,gleb],burlak);
addcomturnunit([burlak],gleb);
```

AddComEnterUnit(units:plist, un:integer)

zbiór jednostek otrzyma polecenie wejścia do określonego pojazdu/budynku

//Przykład - Misja RU 05 cont

```
PlaceUnitXY(Mastodon, 130,67, false);
PlaceUnitXYR(Omar, GetX(Mastodon),GetY(Mastodon), 5, false);
ComCarabine(Omar);
AddComEnterUnit(Omar, Mastodon);
```

AddComExitVehicle(units:plist)

zbiór jednostek otrzyma polecenie wyjścia z pojazdu

//Przykład - Misja AM 14a cont

```
ComMoveXY(MacMillan,192,82);
ComMoveXY(Burlak,193,86);

AddComExitVehicle([MacMillan,Burlak]);
```

AddComExitBuilding(units:plist)

zbiór jednostek otrzyma polecenie wyjścia z budynku

//Przykład - Misja AM 14a

```
function MacMillan_To_Engineer;
var depot;
begin
depot:=FilterAllUnits([ f_or, [f_btype,b_depot], [f_btype,b_warehouse]]);
if depot then
begin
depot:=depot[1];
InGameOn;
if IsInUnit(MacMillan) then
ComExitBuilding(MacMillan);
while GetClass(MacMillan)<>class_engineer do
begin
CenterOnUnits(MacMillan);
ComEnterUnit(MacMillan,depot);
if IsInUnit(MacMillan)=depot then
begin
ComChangeProfession(MacMillan,class_engineer);
AddComExitBuilding(MacMillan);
end;
wait(0$1);
end;
InGameOff;
end;
enable;
end;
```


AddComChangeProfession(units:plist, new_prof:integer)

zbiór jednostek otrzyma polecenie zmiany profesji na new_prof

//Przykład - Misja RU 09 cont

```
if IsInUnit(A1)<>Ar_lab then
begin
  ComExitBuilding(A1);
  AddComEnterUnit(A1,Ar_lab);
  AddComChangeProfession(A1,class_scientistic);
end;
```

AddComResearch(units:plist, techn:integer)

zbiór laboratoriów otrzyma polecenie opracowania technologii techn

//Przykład - Misja AM 15

```
// Run the develop and construction
ComResearch(AmLab, tech_Tech1);
AddComResearch(AmLab, tech_Tech2);
AddComResearch(AmLab, tech_Tech3);
```

AddComConstruct(units:plist, chassis:integer, engine:integer, control:integer, weapon:integer)

zbiór fabryk otrzyma polecenie stworzenia pojazdu z określonymi komponentami

//Przykład - Misja AM 15

```
// Vehicles for the first attack
AddComConstruct(AmFact, us_medium_tracked, engine_solar, control_manual, us_light_gun);
AddComConstruct(AmFact, us_medium_tracked, engine_solar, control_manual, us_light_gun);
AddComConstruct(AmFact, us_medium_tracked, engine_solar, control_manual, us_double_gun);
AddComConstruct(AmFact, us_medium_tracked, engine_solar, control_manual, us_double_gun);
```

AddComPause(units:plist)

zbiór budynków otrzyma polecenie zatrzymania badań lub tworzenia

{brak przykładu}

AddComCancel(units:plist)

zbiór budynków otrzyma polecenie anulowania badań lub tworzenia

{brak przykładu}

AddComHeal(units:plist, un:integer)

zbiór jednostek otrzyma polecenie uleczenia określonej jednostki

//Przykład - Misja RU 08

```
if Attacks(ar_human[1])=Kurt then
begin
  ComStop([Kurt,Mercenary1]);
  SetSide([Kurt,Mercenary1],arab);
  ComStop(ar_human[1]);
  Wait(2);
  AddComMoveUnit(ar_human[1],Kurt);
  if GetLives(Kurt)<300 then AddComHeal(ar_human[3],Kurt);
end;
```

AddComRepairVehicle(units:plist, un:integer)

zbiór jednostek otrzyma polecenie naprawy określonego pojazdu

//Przykład - Misja RU 03

```
every 0$5 marked 201 do
var
    veh, mech;
begin
    enable;
    for veh:=1 to vozKirov do
    begin
        {...}
        if (vozKirovState[veh] = stNotReady) and IsInArea(vozKirov[veh],CAuta) then
        begin
            vozKirovState:=replace(vozKirovState,veh,stRepair);
            ComExitVehicle(mechKirov[veh]);
            wait(5);
            AddComRepairVehicle([mechKirov[veh]],vozKirov[veh]);
        end;
    end;
end;
```

AddComRepairBuilding(units:plist, un:integer)

zbiór jednostek otrzyma polecenie naprawy określonego budynku

//Przykład - Misja AM 09 cont

```
ComMoveXY([Eng1, Eng2, Eng3],34,6);
AddComRepairBuilding([Eng1, Eng2, Eng3], Depot1);
```

AddComTameXY(units:plist, x:integer y:integer)

zbiór jednostek otrzyma polecenie oswojania

{brak przykładu}

AddComPlaceDelayedCharge(units:plist, x:integer, y:integer, un:integer)

zbiór jednostek otrzyma polecenie podłożenia materiałów wybuchowych na [x,y] lub w pobliżu jednostki un

//Przykład - Misja AM 10 cont

```
EVERY 0$1 TRIGGER Arabi_utoci=true DO
VAR arabi_v_zakladne;
VAR pp;
VAR un;
begin
    {...}
    // tapkal
    pp=FilterAllUnits([[f_side,You],[f_type,unit_building]]);
    if pp>0 then
    begin
        un=NearestUnitToUnit(tapkal,pp);
        ComExitBuilding(tapkal);
        AddComPlaceRemoteCharge(tapkal,0,0,un);
        AddComPlaceDelayedCharge(tapkal,GetX(un)+1,GetY(un)+1,0);
        AddComMoveXY(tapkal,78,10);
        AddComFireExplosives(tapkal);
        AddComAgressiveMove(tapkal,128,19);
    end;
end;
```

AddComPlaceRemoteCharge(units:plist, x:integer, y:integer, un:integer)

zbiór jednostek otrzyma polecenie podłożenia zdalnych ładunków wybuchowych na [x,y] lub w pobliżu jednostki un

//Przykład - Misja AM 10 cont

```
EVERY 0$1 TRIGGER Arabi_utoci=true DO
VAR arabi_v_zakladne;
VAR pp;
VAR un;
begin
{...}
pp=FilterAllUnits([[f_side,You],[f_type,unit_building]]);
if pp>0 then
begin
un=NearestUnitToUnit(tapkal,pp);
ComExitBuilding(tapkal);
AddComPlaceRemoteCharge(tapkal,0,0,un);
AddComPlaceDelayedCharge(tapkal,GetX(un)+1,GetY(un)+1,0);
AddComMoveXY(tapkal,78,10);
AddComFireExplosives(tapkal);
AddComAgressiveMove(tapkal,128,19);
end;
end;
```

AddComFireExplosives(units:plist)

zbiór jednostek otrzyma polecenie odpalenia zdalnie sterowanych ładunków wybuchowych

//Przykład - Misja AM 10 cont

```
EVERY 0$1 TRIGGER Arabi_utoci=true DO
VAR arabi_v_zakladne;
VAR pp;
VAR un;
begin
{...}
pp=FilterAllUnits([[f_side,You],[f_type,unit_building]]);
if pp>0 then
begin
un=NearestUnitToUnit(tapkal,pp);
ComExitBuilding(tapkal);
AddComPlaceRemoteCharge(tapkal,0,0,un);
AddComPlaceDelayedCharge(tapkal,GetX(un)+1,GetY(un)+1,0);
AddComMoveXY(tapkal,78,10);
AddComFireExplosives(tapkal);
AddComAgressiveMove(tapkal,128,19);
end;
end;
```

AddComLinkTo(units:plist, hum:integer)

zbiór pojazdów otrzyma polecenie połączenia z określonymi osobami

{brak przykładu}

AddComUnlink(units:plist)

zbiór pojazdów otrzyma polecenie odłączenia od mechaników

{brak przykładu}

AddComCrawl(units:plist)

zbiór jednostek otrzyma polecenie czołgania

//Przykład - Misja AM 02

```
if not(Skip) then AddComCrawl(John);
```

AddComWalk(units:plist)

zbiór jednostek otrzyma polecenie chodzenia

//Przykład - Misja AM 02

```
if not(Skip) then ComWait(John, 0$3);
AddComWalk(John);
AddComTurnUnit(John, Lisa);
```

AddComFree(units:plist)

zbiór jednostek otrzyma polecenie zwolnienia pozycji

//Przykład - Misja AM 02

```
EVERY 0$0.1 MARKED 10 do
begin
AddComEnterUnit(RusEM1, vehicle1);
AddComFree(RusEM1);
end;
```

AddComHold(units:plist)

zbiór jednostek otrzyma polecenie utrzymywania pozycji

//Przykład - Misja AM 02

```
EVERY 0$0.5 Marked 3 DO
var answer, all, i;
begin
if FilterAllUnits([f_side, americans]) > 1 then
begin
if FilterAllUnits([f_side, americans], [f_dist, Mikhail, 15]) > 0 then
begin
all=FilterAllUnits([f_side, americans], [f_dist, Mikhail, 15]); // all units near then 10
AddComHold(all); // stop the work
AddComHold(Followed);
AddComTurnUnit(all, Mikhail);
AddComTurnUnit(Followed, Mikhail);
{...}
end
else enable;
end;
end;
```

AddComStop(units:plist)

zbiór jednostek otrzyma polecenie zatrzymania się

//Przykład - Misja AM 11

```
function jdi_na_przukum(pru);
begin
ComStop(pru);
AddComExitBuilding(pru);
AddComExitVehicle(pru);
AddComStop(pru);
{...}
end;
```

AddComWait(units:plist, time:integer)

zbiór jednostek otrzyma polecenie oczekiwania

//Przykład - Misja AM 01 cont

```
AddComMoveXY(JMM,80,78);
AddComWait(JMM,0$3);
AddComMoveXY(JMM,84,80);
```

AddComRemember(units:plist)

zbiór jednostek otrzyma polecenie zapamiętania pozycji

//Przykład - Misja AM 05

```
wait(0$2);
AddComRemember(all_units);
AddComMoveXY(JMM,46,15);
```

AddComReturn(units:plist)

zbiór jednostek otrzyma polecenie odwrotu

//Przykład - Misja AM 14

```
function ukradni_macmillanovi_auto;
var un;
begin
  for un in ru_sol do
    if IsOK(un) then
      ComEnterUnit(un,truck1);
      AddComAgressiveMove(ru_sol,69,35);
      AddComReturn(ru_sol);
      wait(0$20);
    end;
  end;
```

AddComBuild(units:plist, bud:integer, x:integer, y:integer, dir:integer)

zbiór jednostek otrzyma polecenie budowania

//Przykład - Misja RU 12a

```
every 0$1+0$0.2 do // Alliance build base
var filter,un,skill,inzenyr;
begin
  if IsInArea(Burlak,StartBuildArea) then
    begin
      {...}
      Wait(10);
      AddComBuild(inzenyr,b_oil_power,126,17,5);
      EnableExclamations;
      disable;
    end
  else
    enable;
  end;
end;
```

AddComUpgrade(units:plist)

lista jednostek otrzyma polecenie rozbudowy

//Przykład - Misja RU 05 cont

```
function build_base(builds);
var stavari, i, a;
begin
  for i in builds do begin
    stavari:=get_stavari;
    {...}
    while not BuildingsInProgress(arabians) do begin
      if i[3]=upgrade then
        AddComUpgrade(get_build(i[1],i[2],i[4]))
      else
        AddComBuild(stavari, i[4], i[1],i[2],i[3]);
        wait(0$1);
      end;
    end;
  end;
end;
```

AddComUpgradeLab(units:plist, labtype:integer)

lista jednostek otrzyma polecenie ulepszenia laboratorium na specjalistyczne

//Przykład - Misja AM 10

```
EVERY 0$3 marked 220 do
begin
  AddComUpgradeLab(R_lab,b_lab_spacetime);
  AddComUpgradeLab(R_lab,b_lab_weapon);
{...}
end;
```

AddComPlaceWeapon(units:plist, weap:integer)

lista wieżyczek i bunkrów otrzyma polecenie zamontowania określonego rodzaju broni

//Przykład - Misja AM 10

```
EVERY 0$3 marked 219 DO
begin
  cheat_source(R_depot,20,0,10);
  AddComPlaceWeapon(R_bunkr,ru_heavy_machine_gun);
end;
```

AddComAnim(units:plist, atype:integer)

lista jednostek otrzyma polecenie określonej animacji

{brak przykładu}

AddComRefuel(units:plist, un:integer)

lista jednostek otrzyma polecenie tankowania

{brak przykładu}

AddComTransport(units:plist, un:integer, mattype:integer)

lista jednostek otrzyma polecenie transportowania

//Przykład - Misja AM 15

```
EXPORT Function SendMaterial;
var Un, i, Cans_Positions, Free_Place;
begin
{...}
  Cans_Positions= [[63, 103], [68, 101], [66, 96], [64, 99], [59, 99], [67,104], [69,95]];
  for i= 1 to Number_of_crates do
    begin
      if i=1 then ComTransport(Carriers, AmDep, mat_cans)
      else AddComTransport(Carriers, AmDep, mat_cans);
      AddComMoveXY(Carriers, Cans_Positions[i][1], Cans_Positions[i][2]);
      AddComUnload(Carriers);
    end;
  {...}
end;
```

AddComInvisible(units:plist)

lista jednostek otrzyma polecenie niewidzialności

{brak przykładu}

AddComSpaceShift(units:plist, x:integer, y:integer)

lista jednostek otrzyma polecenie użycia rakiet czaso-przestrzennych

//Przykład - Misja AM 10 cont

```
EVERY 0$3 marked 247 DO
VAR i;
begin
  if vojaci then
  begin
    for i in CREATE_VOJACI_LIST do PlaceUnitArea(i,RUEV,false);
    VOJUTOK=FilterUnitsInArea(RUEV,[[f_side,Rusi]]);
    AddComAgressiveMove(VOJUTOK,122,40);
    for i in VOJUTOK DO
    begin
      AddComSpaceShift(i,109,40);
      AddComSailEvent(i,666);
      AddComTimeShift(i,107,41);
      AddComSailEvent(i,666);
      AddComSpaceShift(i,110,30);
      AddComSailEvent(i,666);
    end;
  }
end;
```

AddComTimeShift(units:plist, x:integer, y:integer)

lista jednostek otrzyma polecenie użycia rakiet spowalniających

//Przykład - Misja AM 10 cont

```
EVERY 0$3 marked 247 DO
VAR i;
begin
  if vojaci then
  begin
    for i in CREATE_VOJACI_LIST do PlaceUnitArea(i,RUEV,false);
    VOJUTOK=FilterUnitsInArea(RUEV,[[f_side,Rusi]]);
    AddComAgressiveMove(VOJUTOK,122,40);
    for i in VOJUTOK DO
    begin
      AddComSpaceShift(i,109,40);
      AddComSailEvent(i,666);
      AddComTimeShift(i,107,41);
      AddComSailEvent(i,666);
      AddComSpaceShift(i,110,30);
      AddComSailEvent(i,666);
    end;
  }
end;
```

AddComHack(units:plist, un:integer)

lista jednostek otrzyma polecenie hack

{brak przykładu}

AddComTeleportExit(units:plist, x:integer, y:integer, un:integer)

lista jednostek otrzyma polecenie umieszczenia wyjścia z teleportu un na [x,y]

{brak przykładu}

AddComHiddenCamera(units:plist, x:integer, y:integer)

lista jednostek otrzyma polecenie umieszczenia ukrytej kamery

//Przykład - Misja RU 09

```
EVERY 0$1 DO
begin
  ComEnterUnit(Arabi_enter[1],Ar_lab);
  ComPlaceRemoteCharge(diet,78,47,0);
  AddComEnterUnit(diet,Ar_bar);
  ComPlaceRemoteCharge(Arabi_enter[2],78,43,0);
  AddComEnterUnit(Arabi_enter[2],Ar_bu1);
  ComHiddenCamera(Arabi_enter[3],64,38);
  AddComHiddenCamera(Arabi_enter[3],85,52);
  AddComHiddenCamera(Arabi_enter[3],101,51);
  AddComEnterUnit(Arabi_enter[3],Ar_dep);
  ComPlaceRemoteCharge(Arabi_enter[4],82,46,0);
  AddComEnterUnit(Arabi_enter[4],Ar_bu2);
  {...}
end;
```

AddComContaminate(units:plist, x:integer, y:integer)

lista jednostek otrzyma polecenie zanieczyszczenia syberytu

{brak przykładu}

AddComUnload(units:plist)

lista jednostek otrzyma polecenie wyładowania ładunku

//Przykład - Misja RU 06

```
every 0$5 trigger kecali and FilterUnitsInArea(north_base, [[f_side,you]]) do
begin
  wait(0$10);
  nakladaky;
  ComMoveXY([accar1,accar2],53,8);
  DTruck;
  AddComMoveXY([accar1,accar2],45,14);
  AddComMoveXY([accar1,accar2],33,14);
  AddComMoveXY([accar1,accar2],27,32);
  AddComUnload([accar1,accar2]);
  AddComMoveXY([accar1,accar2],42,79);
  AddComExitVehicle([acman1, acman2]);
  sol:=sol union [acman1, acman2];
  a3:=a3 union [acman1, acman2];
end;
```

AddComGet(units:plist, x:integer, y:integer)

lista jednostek otrzyma polecenie podniesienia ładunku

//Przykład - Misja RU 12a cont

```
AddComGet(Gossudarov,FindArtifact(4)[1],FindArtifact(4)[2]);
Wait(0$5);
ComExitBuilding(scientists);
AddComMoveXY(scientists^Gossudarov,50,58);
```

AddComGive(units:plist, un:integer)

lista jednostek otrzyma polecenie przekazania ładunku

{brak przykładu}

AddComCarabine(units:plist)

lista szejków otrzyma polecenie CarabineOn

{brak przykładu}

AddComSabre(units:plist)

lista szejków otrzyma polecenie SabreOn

{brak przykładu}

AddComSailEvent(units:plist, num:integer)

jednostka wywołuje losowe zdarzenie SailEvent(num)

//Przykład - Misja AM 10 cont

```
EVERY 0$3 marked 247 DO
VAR i;
begin
if vojaci then
begin
for i in CREATE_VOJACI_LIST do PlaceUnitArea(i,RUEV,false);
VOJUTOK=FilterUnitsInArea(RUEV,[[f_side,Rusi]]);
AddComAgressiveMove(VOJUTOK,122,40);
for i in VOJUTOK DO
begin
AddComSpaceShift(i,109,40);
AddComSailEvent(i,666);
AddComTimeShift(i,107,41);
AddComSailEvent(i,666);
AddComSpaceShift(i,110,30);
AddComSailEvent(i,666);
end;
end;
{...}
end;
disable_me(247);
enable;
end;
```

AddComStand(units:plist)

zbiór jednostek otrzyma polecenie stój

//Przykład - Misja RU 05 cont

```
every 0$10 trigger defence do
begin
ComMoveXY(defence[1], 121,69);
AddComStand(defence[1]);
enable;//-----
end;
```

AddComAttackSoporific(units:plist, un:integer)

atakuj jednostki un nasenną amunicją

{brak przykładu}

AddComDismantle(units:plist, un:integer)

lista jednostek otrzyma polecenie demontażu budynku un

{brak przykładu}

AddComRecycle(units:plist, un:integer)

lista pojazdów otrzyma polecenie do ponownego wykorzystania w budynku un

```
every 0$2 trigger see(ar,masha) marked 14 do
var tmp;
begin
{ comtransport(inzynyrka,ar_depot,mat_oil);
wait(0$0.5);
comrefuel(inzynyrka,masha);
// while not getfuel(masha) do wait(0$1);
// while not isidle(inzynyrka) do wait(0$1);
addcomenterunit(inzynyrka,masha);
comreturn(abduluv);
while not isinunit(inzynyrka) do wait(0$1);
commovexy(masha,140,65);
addcomrecycle(masha,arfact00);
while not isdead(masha) do wait(0$1);
changemissionobjectives('M4b');}
{...}
end;
```

AddComLinkToBase(units:plist, un:integer)

magazyn lub skład otrzyma polecenie do połączenia budynku un do bazy

{brak przykładu}

AddComBuildBehemoth(units:plist, bud:integer, x:integer, y:integer, dir:integer)

zbiór jednostek otrzyma polecenie budowania behemotha

{brak przykładu}

AddComForceInvisible(units:plist)

lista jednostek otrzymuje komendę niewidzialności

//Przykład - Misja AM 15

```
// or continue the way
else if not(HasTask(Explorer1)) and not(HasTask(Explorer2)) and
(Step_On_The_Way < Way_of_Explorers[Explorer_Way_Number]) then
begin
Wait(0$2);
if not(SeeXY(americans, Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][1],
Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][2]))
or (Rand(1,20) = 1) then
begin
ComMoveXY(Explorer1, Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][1],
Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][2]);
AddComHold(Explorer1);
AddComForceInvisible(Explorer1);
ComMoveXY(Explorer2, Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][1],
Way_of_Explorers[Explorer_Way_Number][Step_On_The_Way][2]);
AddComHold(Explorer2);
AddComForceInvisible(Explorer2);

Step_On_The_Way= Step_On_The_Way+1;
end;
end;
```

Ustalenia/Zwracanie (Get, Set, Is, Find etc.):

SetRememberedX(un:integer, value:integer)

ustala ComReturn koordynaty x jednostki

//Przykład - Misja AM 15

```
// Send Rus_Doctors to Heal area
for Un in Rus_Doctors do
begin
hex= RandHexArea(Rus_Heal, true);
SetRememberedX(Un, hex[1]);
SetRememberedY(Un, hex[2]);
end;
ComReturn(Rus_Doctors);
```

SetRememberedY(un:integer, value:integer)

ustala ComReturn koordynaty y jednostki

//Przykład - Misja AM 15

```
// Send Rus_Doctors to Heal area
for Un in Rus_Doctors do
begin
hex= RandHexArea(Rus_Heal, true);
SetRememberedX(Un, hex[1]);
SetRememberedY(Un, hex[2]);
end;
ComReturn(Rus_Doctors);
```

SetDir(un:integer, dir:integer)

ustala kierunek dla nieumieszczonej jednostki

//Przykład - Misja AM 13a

```
every 10$05 do
begin
uc_side=oth;
uc_nation=nation_russian;
vc_control=control_manual;
vc_engine=engine_combustion;
vc_chassis=ru_medium_wheeled;
vc_weapon=ru_gatling_gun;
masha=createvehicle;
setmark(masha,1);
setfuel(masha,0);
setdir(masha,3);
setlives([masha,burlak],499);
placeunitxy(masha,116,4,false);
placeunitxy(burlak,116,2,false);
commoveunit(burlak,masha);
wait(0$0.3);
commovexy(masha,109,9);
end;
```

SetLives(units:plist, value:integer)

ustal życie określonej jednostki przez wartość (value)

//Przykład - Misja RU 03

```
EVERY 0$4 TRIGGER arabsky_utok_zacal DO
begin
if GetLives(RU1_el1)<1000 and GetLives(RU1_el1)<150 then SetLives(RU1_el1,150);
if GetLives(RU1_el2)<1000 and GetLives(RU1_el2)<150 then SetLives(RU1_el2,150);
if IsOK(RU1_el1) or IsOK(RU1_el2) then enable;
end;
```

SetSide(units:plist, value:integer)

ustal stronę wybranej jednostki przez wartość (value)

//Przykład - Misja RU 08

```
if filter>0 then
begin
repeat
Wait(0$1);
until NOT HasTask(Heike);
RemoveUnit(Heike);
PlaceHumanInUnit(Heike,filter[1]);
SetSide(Heike,you);
end;
```

SetSideBase(base:integer, value:integer)

ustal stronę skończonej bazy

//Przykład - Misja RU 12

```
case side of
ai_s[1]: begin
SetSideBase(GetBase(depots[1]), you);
SetSide(humans[1], you);
SetSide(vehicles[1], you);
end;
{...}
end;
```

SetSkill(units:plist, sk:integer, value:integer)

ustal umiejętność sk dla określonej osoby przez value

//Przykład - Misja AM 15

```
uc_nation= nation_russian;
Kozlov= NewCharacter("Kozlov");
if Kozlov then
begin
SetSkill(Kozlov, skill_combat, 2);
SetSkill(Kozlov, skill_engineering, 8);
SetSkill(Kozlov, skill_mechanical, 3);
SetSkill(Kozlov, skill_scientistic, 10);
End {...}
```

GiveSkillBonus(units:plist, sk:integer, value:integer)

zwiększ podstawowe umiejętności sk dla określonej postaci przez wartość (value)

{brak przykładu}

SetAttr(units:plist, attr:integer, value:integer)

ustal atrybut attr określonej jednostki (units) osoby przez wartość (value)

//Przykład - Misja RU 01 cont

```
function prepare_pokryshkin;
begin
uc_side:=russians;
Pokryshkin:=NewCharacter("Pokryshkin");
SetAttr(Pokryshkin,attr_speed,12);
SetAttr(Pokryshkin,attr_stamina,8);
exit;
{...}
end;
```

SetFuel(units:plist, percent:integer)

ustal paliwo dla określonych pojazdów spalinowych lub na baterie słoneczne z listy

//Przykład - Misja AM 13a

```
every 10$05 do
begin
{...}
masha=createvehicle;
setmark(masha,1);
setfuel(masha,0);
setdir(masha,3);
setlives([masha,burlak],499);
placeunitxy(masha,116,4,false);
placeunitxy(burlak,116,2,false);
commoveunit(burlak,masha);
wait(0$0.3);
commovexy(masha,109,9);
end;
```

SetBLevel(units:plist, blevel:integer)

ustal podstawowy poziom budynku(ów)

//Przykład - Misja AM 04

```
case difficulty of
2:begin
setskill([ruhmech0,ruhmech2],skill_mechanical,2);
complaceweapon(rubbunk0,ru_gatling_gun);
setskill([ruhsold2,ruhsold6,ruhsold7],skill_combat,3);
setskill(ruhsold1,skill_combat,2);
setblevel([rubbrwr0,rubbrwr1],2);
setblevel([rubbunk0],2);
end;
{...}
end; //case
```

SetMark(units:plist, mark:integer)

ustal oznaczenie jednostce (flaga masha)

//Przykład - Misja AM 13a

```
every 10$05 do
begin
uc_side=oth;
uc_nation=nation_russian;
vc_control=control_manual;
vc_engine=engine_combustion;
vc_chassis=ru_medium_wheeled;
vc_weapon=ru_gatling_gun;
masha=createvehicle;
setmark(masha,1);
setfuel(masha,0);
setdir(masha,3);
setlives([masha,burlak],499);
{...}
end;
```

TeleportExit(un:integer, x:integer, y:integer)

tworzy wyjście na [x,y] dla teleportu un

//Przykład - Misja RU 15

```
teleportexit(altelp00,7,5);
```

HiddenCamera(x:integer, y:integer, side:integer)

tworzy ukrytą kamerę

//Przykład - Misja AM 11 cont

```
EXPORT function dialog_kamery;
begin
  if not IsOK(A3_solop) then exit;
  Say(A3_solop,'D7a-Ar1-1');
  HiddenCamera(87,3,You);
  HiddenCamera(105,39,You);
  HiddenCamera(134,56,You);
  HiddenCamera(122,51,You);
  HiddenCamera(91,8,You);
  ChangeMissionObjectives('MCarOut');
  SetAreaMapShow(baziny,0);
end;
```

SetLastMission(humans:plist, number:integer)

zmienia ostatnią misję, w której postaci ludzkie są obecne

{brak przykładu}

GiveMissionExperience(humans:plist, number:integer)

daje domyślne doświadczenie dla misji number dla postaci humans

{brak przykładu}

GetType(un:integer)

zwraca typ jednostki (unit_human, unit_vehicle, unit_building)

//Przykład - Misja RU 10 cont

```
// cekani na Burlakovu smrt
on UnitDestroyed(un) do
begin
  if un=Burlak then
    burlaks_end;
  if GetSide(un)=you and GetType(un)=unit_human then
    lost_russians:=lost_russians+1;
  if un=Masha then
    lost_masha:=true;
end;
```

GetNation(un:integer)

zwraca narodowość jednostki

//Przykład - Misja AM 15a

```
function sort_aliance_by_the_nation;
var un,us_list,ru_list;
begin
  us_list:=[];
  ru_list:=[];

  for un in aliance_forces do
    if GetNation(un)=us then
      us_list:=us_list ^ un
    else
      ru_list:=ru_list ^ un;

  aliance_forces:=us_list ^ ru_list;
end;
```

GetIdent(un:integer)

zwraca ident określonej jednostki (dla debugowania list etc.)

{brak przykładu}

GetX(un:integer)

zwraca koordynat x jednostki

//Przykład - Misja AM 15

```
EVERY 0$2 TRIGGER Center_To_Powell DO
var X, Y, NewX, NewY;
begin
  if IsInUnit(Powell) = Powell_vehicle then
    begin
      X= GetX(Powell_vehicle);
      Y= GetY(Powell_vehicle);
      NewX= ShiftX(X, GetDir(Powell_vehicle), 5);
      NewY= ShiftY(Y, GetDir(Powell_vehicle), 5);
      CenterOnXY(NewX, NewY);
    end
  else
    begin
      X= GetX(Powell);
      Y= GetY(Powell);
      NewX= ShiftX(X, GetDir(Powell), 2);
      NewY= ShiftY(Y, GetDir(Powell), 2);
      CenterOnXY(NewX, NewY);
    end;
  enable;
end;
```

GetY(un:integer)

zwraca koordynat y jednostki

//Przykład - Misja AM 15

```
EVERY 0$2 TRIGGER Center_To_Powell DO
var X, Y, NewX, NewY;
begin
  if IsInUnit(Powell) = Powell_vehicle then
    begin
      X= GetX(Powell_vehicle);
      Y= GetY(Powell_vehicle);
      NewX= ShiftX(X, GetDir(Powell_vehicle), 5);
      NewY= ShiftY(Y, GetDir(Powell_vehicle), 5);
      CenterOnXY(NewX, NewY);
    end
  else
    begin
      X= GetX(Powell);
      Y= GetY(Powell);
      NewX= ShiftX(X, GetDir(Powell), 2);
      NewY= ShiftY(Y, GetDir(Powell), 2);
      CenterOnXY(NewX, NewY);
    end;
  enable;
end;
```

GetRememberedX(un:integer)

zwraca ComReturn koordynat x jednostki

// Przykład - Misja AM 07

```
if (GetLives(West_mech) < 600) and (IsDrivenBy(West_veh) = West_mech) then
  begin
    if IsOk(Center_mech) and IsOk(Center_veh) and (State_MechC = 0) then // goes to help
      begin
        State_MechC= 4;
        ComMoveXY(Center_veh, GetRememberedX(West_veh), GetRememberedY(West_veh));
        AddComHold(Center_veh);
      end;
  end;
```

GetRememberedY(un:integer)

zwraca ComReturn koordynat y jednostki

// Przykład - Misja AM 07

```
if (GetLives(West_mech) < 600) and (IsDrivenBy(West_veh) = West_mech) then
  begin
    if IsOk(Center_mech) and IsOk(Center_veh) and (State_MechC = 0) then // goes to help
      begin
        State_MechC= 4;
        ComMoveXY(Center_veh, GetRememberedX(West_veh), GetRememberedY(West_veh));
        AddComHold(Center_veh);
      end;
  end;
```

GetDir(un:integer)

zwraca kierunek jednostki

//Przykład - Misja AM 15

```
EVERY 0$2 TRIGGER Center_To_Powell DO
var X, Y, NewX, NewY;
begin
  if IsInUnit(Powell) = Powell_vehicle then
    begin
      X= GetX(Powell_vehicle);
      Y= GetY(Powell_vehicle);
      NewX= ShiftX(X, GetDir(Powell_vehicle), 5);
      NewY= ShiftY(Y, GetDir(Powell_vehicle), 5);
      CenterOnXY(NewX, NewY);
    end
  else
    begin
      X= GetX(Powell);
      Y= GetY(Powell);
      NewX= ShiftX(X, GetDir(Powell), 2);
      NewY= ShiftY(Y, GetDir(Powell), 2);
      CenterOnXY(NewX, NewY);
    end;
  enable;
end;
```

GetSide(un:integer)

zwraca stronę jednostki

//Przykład - Misja AM 13a

```
every 0$1 trigger koncime or base_out or artef_out do
begin
  //if islive(abdul) then wait(0$30); //time to kill abdul
  if (getside(masha)=al) or (getside(masha)=oth) then addmedal('Masha',1)
  else addmedal('Masha',-1);
  {...}
end;
```


GetLives(units:plist)

zwraca przeciętny poziom życia określonej jednostki (units)

//Przykład - Misja AM 03

```
for un in repair do
  if (GetLives(un)<1000) and IsDrivenBy(un) then
    ComExitVehicle(un);
```

GetClass(un:integer)

zwraca klasę określonej jednostki (units) ludzkiej

//Przykład - Misja AM 07

```
// No of people will be engineer
for Un in All_americans ^ JMM do
  if (GetClass(Un) = class_engineer) then SetClass(Un, class_soldier);
```

GetSex(un:integer)

zwraca płeć określonej jednostki (units) ludzkiej

//Przykład - Misja AM 15

```
Function CaptureDialogue(Captured_american);
var Join;
begin
  Join= 0;
  DialogueON;
  case Captured_american of
    JMM: ForceSay(JMM, 'DA1-JMM-1');
    {...}
  else
    begin
      if GetSex(Captured_american) = sex_male then
        ForceSay(Captured_american, 'DA1-Sol1-1')
      else ForceSay(Captured_american, 'DA1-FSol1-1');
    end;
  end;
```

GetSkill(units:plist, sk:integer)

zwraca przeciętny poziom umiejętności sk określonej jednostce (units) ludzkiej

//Przykład - Misja AM 11 cont

```
function get_skills(un);
VAR pom1,pom2,pom3,pom4;
begin
  pom1:=GetSkill(un,skill_combat);
  pom2:=GetSkill(un,skill_engineering);
  pom3:=GetSkill(un,skill_mechanical);
  pom4:=GetSkill(un,skill_scientistic);
  result:=[pom1,pom2,pom3,pom4]
end;
```

GetAttr(units:plist, attr:integer)

zwraca przeciętny atrybut attr określonej postaci (units) ludzkiej

//Przykład - Misja AM 06 cont

```
for t1 in FilterAllUnits([[f_side,russians],[f_type,unit_human]]) do begin
  SetAttr(t1, attr_stamina, GetAttr(t1,attr_stamina)+1);
  SetAttr(t1, attr_speed, GetAttr(t1,attr_speed)+1);
  SetSkill(t1, skill_combat, GetSkill(t1,skill_combat)+1);
end;
```

GetFuel(units:plist)

określa przeciętny poziom paliwa określonemu pojazdowi spalinowemu lub na baterie słoneczne z listy

//Przykład - Misja AM 06 cont

```
every 0$1.0 do
var set_fuel, fuel, depot;
begin
  for depot in FilterAllUnits([[f_btype, b_depot],[f_side,you]]) do
    for set_fuel in FilterAllUnits([[f_type, unit_vehicle], [f_side, you], [f_dist, depot, 8]]) do begin
      fuel:=GetFuel(set_fuel);
      if fuel<=80 then
        SetFuel(set_fuel, fuel+20);
      end;
    end;
  enable;//-----
end;
```

GetEngine(un:integer)

zwraca typ silnika pojazdu

//Przykład - Misja AM 03

```
every 0$2 trigger GetEngine(IsInUnit(Lisa))=engine_solar Marked 7 do
begin
  Wait(0$2);
  EnableExclamations;
  async;
  Say(Lisa,'D2c-Lisa-1');
  sync;

  AddComMoveXY(Lisa,GetX(Lisa),GetY(Lisa)-2);
  {...}
  DisableExclamations;
end;
```

GetControl(un:integer)

zwraca typ sterowania pojazdu

//Przykład - Misja RU 11 cont

```
setside([buldoz1,buldoz2],ru2);
if getcontrol(buldoz1)=control_manual then begin
  comexitvehicle(buldoz1);
  wait(0$1);
  comenterunit(popov,buldoz1);
  popin=true;
end;
```

GetWeapon(un:integer)

zwraca typ broni pojazdu

//Przykład - Misja AM 05 cont

```
every 0$1.0 do
begin
  enter_near_vehicle(make_mess, 4, FilterAllUnits([[f_side, you],[f_type, unit_vehicle]]));
  arabs_cars:=FilterAllUnits([[f_type, unit_vehicle],[f_side, arabsians]]);
  if arabs_cars then begin
    if GetWeapon(arabs_cars[1])<>us_cargo_bay then begin
      ComWait(arabs_cars, 0$12);
      AddComMoveToArea(arabs_cars, border);
    end
    else
      ComMoveToArea(arabs_cars, border);
    end;
  end;
  enable;
end;
```

GetChassis(un:integer)

zwraca typ podwozia pojazdu

//Przykład - Misja RU 15a

//----- Checking if all vehicles has tracked chassis

```
function All_Tracked(list);
var un,vysledek;
begin
  vysledek:=true;
  for un in list do
    if GetChassis(un) in [us_light_wheeled, us_medium_wheeled, ru_medium_wheeled] then
      begin
        vysledek:=false;
        break;
      end;
    end;
  result:=vysledek;
end;
```

GetBType(un:integer)

zwraca typ budynku, jeżeli un nie jest budynkiem, zwraca -1

//Przykład - Misja RU 05 cont

```
on BuildingComplete(build) do
begin
  if GetSide(build)=you then begin
    if not depot and GetBType(build)=b_depot then
      depot:=build;
    end
  else begin
    if GetBType(build) in [b_depot,b_warehouse] then
      dep1:=build+0;
    if GetBType(build) in [b_workshop,b_factory] then
      fact1:=build+0;
    if GetBType(build) in [b_lab,b_lab_half,b_lab_full] then
      lab1:=build+0;
    if GetBType(build) in [b_breastwork,b_bunker,b_armoury,b_barracks] then
      fort:=fort ^ [build];
    end;
  end;
end;
```

GetBLevel(un:integer)

zwraca poziom budynku, jeżeli un nie jest budynkiem, zwraca -1

//Przykład - Misja AM 12a cont

```
function set_vars;
var tmp,tmp1;
begin
  {...}

  if difficulty=3 then begin
    tmp=filterallunits([[f_side,am],[f_or,[f_btype,b_bunker],[f_btype,b_turret]]]);
    for tmp1 in tmp do setblevel([tmp1],getblevel(tmp1)+2);
  end;
end;
```

GetLabKind(un:integer, which:integer)

zwraca rodzaj laboratorium, which (która) jest dla 1 lub 2 części

//Przykład - Misja AM 15a

```
function get_possible_research(side);           // zjuztuje upgrady laboratorii (weapon, siberit, ...)
var un,vysledek,
    lab_list,           // seznam laboratorii
    upgrade;           // seznam upgradu jednotlivych laboratorii
begin
    vysledek:=[];
    upgrade:=[];
    lab_list:=get_labs(side);
    for un in lab_list do
        upgrade:=upgrade ^ GetLabKind(un,1) ^ GetLabKind(un,2);
    for un in upgrade do
        vysledek:=vysledek ^ (un-8);

    result:=vysledek;
end;
```

GetBWeapon(un:integer)

zwraca typ broni (w przypadku wiezyczki lub bunkra, inaczej zero)

//Przykład - Misja RU 12

```
//rebuild base
every 0$10 + 0$0.0 do
var i, j, weap, k;
begin
    for i in UnitFilter(fort, [mf_bunk]) do
        if not GetBWeapon(i) then begin
            weap:=0;
            for k:=1 to 5 do
                for j in builds[k] do
                    if GetX(i)=j[2] and GetY(i)=j[3] and GetBType(i)=j[1] then begin
                        weap:=j[5][1];
                        break;
                    end;
                if weap then
                    ComPlaceWeapon(j, weap);
            end;
        enable;
    end;
```

GetExtPositions(un:integer)

zwraca liste pozycji rozbudowan [[x,y],[x,y],[x,y],[x,y],[x,y]]

//Przykład - Misja AM 15a

```
function ExtensionBuild(factory, extension);
var i, vysledek, list;
begin
    vysledek:=false;
    if extension>0 then           // technologie potrzebuje pristavek
        begin
            list:=GetExtPositions(factory);
            for i in list do
                if GetBType(HexInfo(i[1],i[2]))=extension then
                    begin
                        vysledek:=true;
                        break;
                    end;
            end;
        end
    else           // technologie nepotrebuje pristavek
        vysledek:=true;
    result:=vysledek;
end;
```

GetMark(un:integer)

zwraca wartość oznaczenia jednostki un (flaga masha)

{brak przykładu}

ShiftX(x:integer, dir:integer, dist:integer)

zwraca koordynat x przesunięty przez macierzysty hex dist w kierunku dir

//Przykład - Misja AM 15

```
EVERY 0$2 TRIGGER Center_To_Powell DO
var X, Y, NewX, NewY;
begin
  if IsInUnit(Powell) = Powell_vehicle then
    begin
      X= GetX(Powell_vehicle);
      Y= GetY(Powell_vehicle);
      NewX= ShiftX(X, GetDir(Powell_vehicle), 5);
      NewY= ShiftY(Y, GetDir(Powell_vehicle), 5);
      CenterOnXY(NewX, NewY);
    end
  else
    begin
      X= GetX(Powell);
      Y= GetY(Powell);
      NewX= ShiftX(X, GetDir(Powell), 2);
      NewY= ShiftY(Y, GetDir(Powell), 2);
      CenterOnXY(NewX, NewY);
    end;
  enable;
end;
```

ShiftY(y:integer, dir:integer, dist:integer)

zwraca koordynat y przesunięty przez macierzysty hex dist w kierunku dir

//Przykład - Misja AM 15

```
EVERY 0$2 TRIGGER Center_To_Powell DO
var X, Y, NewX, NewY;
begin
  if IsInUnit(Powell) = Powell_vehicle then
    begin
      X= GetX(Powell_vehicle);
      Y= GetY(Powell_vehicle);
      NewX= ShiftX(X, GetDir(Powell_vehicle), 5);
      NewY= ShiftY(Y, GetDir(Powell_vehicle), 5);
      CenterOnXY(NewX, NewY);
    end
  else
    begin
      X= GetX(Powell);
      Y= GetY(Powell);
      NewX= ShiftX(X, GetDir(Powell), 2);
      NewY= ShiftY(Y, GetDir(Powell), 2);
      CenterOnXY(NewX, NewY);
    end;
  enable;
end;
```

GetBase(un:integer)

zwraca bazę dla określonych budynków (lub zero, jeżeli budynek nie istnieje)

//Przykład - Misja RU 05 cont

```
function place_base(builds);
var i, base;
begin
  InitUc;
  InitBc;
  uc_side:=arabians;
  uc_nation:=nation_arabian;

  for i in builds do begin
    bc_type:=i[4];
    if bc_type in [b_lab,b_lab_half,b_lab_full] then begin
      bc_kind1:=i[5];
      bc_kind2:=i[6];
    end;
    CreateAndPlaceBuildingXYD(i[1],i[2],i[3]);
  end;
  base:=GetBase(FilterAllUnits([f_btype,b_depot])[1]);
  SetResourceType(base, mat_cans, 450);
  SetResourceType(base, mat_oil, 200);
end;
```

GetResourceType(base:integer, mattype:integer)

zwraca ilość materiału określonego typu w określonej bazie

//Przykład - Misja RU 03

```
// doplnovani ropy bazi RU1
EVERY 1$0 DO
begin
  if GetResourceType(GetBase(RU1_dep),mat_oil)<50
  then AddResourceType(GetBase(RU1_dep),mat_oil,100);
  enable;
end;
```

AddResourceType(base:integer, mattype:integer, amount:integer)

dodaje zasoby określonego typu do określonej bazy

//Przykład - Misja AM 10

```
// pridani konzerv,ropy,sioberitu do base
EXPORT function cheat_source(dep,cans,oil,siberite);
begin
  Dbase=GetBase(dep);
  AddResourceType(Dbase,mat_cans,cans);
  AddResourceType(Dbase,mat_oil,oil);
  AddResourceType(Dbase,mat_siberit,siberite);
end;
```

SetResourceType(base:integer, mattype:integer, amount:integer)

ustala zasoby określonego typu do określonej bazy

//Przykład - Misja RU 05 cont

```
function place_base(builds);
var i, base;
begin
  {...}
  base:=GetBase(FilterAllUnits([f_btype,b_depot])[1]);
  SetResourceType(base, mat_cans, 450);
  SetResourceType(base, mat_oil, 200);
end;
```

GetEnergy(base:integer)

zwraca listę wartości mocy: [need,needmax,prod,prodmax]

{brak przykładu}

GetResources(base:integer)

zwraca listę zasobów bazy: [cans,oil,sib]

//Przykład - Misja AM 15a

```
function Show_me_map(side,x,y);
var vysledek,base;
begin
  vysledek:=true;
  base:=GetBase(nearest_labs[1]);
  if GetResources(base)[3] >= sib_to_view then
    begin
      PlaceSeeing(x,y,side,-15);
      SetResourceType(base,mat_siberit,GetResources(base)[3]-sib_to_view);
      view_data:=[x,y,side];
    end
  else
    vysledek:=false;
    result:=vysledek;
  end;
end;
```

CanCarry(un:integer)

zwraca true jeżeli jednostką jest inżynier mała-inżynier lub komora

{brak przykładu}

Carry(un:integer)

zwraca true jeżeli jednostka transportuje coś

//Przykład - Misja AM 10

```
EVERY 0$4 marked 16 DO
begin
  if not Carry(auta[1]) then begin
    ComMoveXY(auta[1],64,23);
    enable(17);
  end
  else enable;
end;
```

CanCarryNext(un:integer)

zwraca true jeżeli jednostka jest pełna

{brak przykładu}

GetResourceTypeXY(x:integer, y:integer)

zwraca typ zasobu w określonym hex'ie lub zero

{brak przykładu}

GetResourceAmountXY(x:integer, y:integer)

zwraca ilość zasobów na określonym hex'ie

//Przykład - Misja RU 02

```
// odpis beden
EVERY 0$13 TRIGGER stav>2 and stav<7 Do
VAR i;
VAR a;
begin
  a=[];
  for i in bedny do if GetResourceAmountXY(i[1],i[2])<>0 then a=a^[i];
  bedny=a;
  enable;
end;
```

ChangeResourceAmountXY(x:integer, y:integer, n:integer)

zmienia ilości istniejących zasobów na określonym hex'ie

{brak przykładu}

EraseResourceArea(ar:integer, typ:integer)

kasuje wszystkie zasoby określonego typu w określonej strefie

//Przykład - Babylon

```
function DestroyArtefact(hex,art);
var base,side;
begin
  MoveAreaXY(artefact_end,hex[1],hex[2]);
  EffectTeleport(hex[1],hex[2]);
  Wait(30);
  EraseResourceArea(artefact_end,mat_artefact);

  base:=(art-1)*100;
  If GetTag(base+11) then
  begin
    side:=GetSide(GetTag(base+11));
    if GetTag(base+12) then
    begin
      ComCancel(GetTag(base+11));
      Wait(10);
      SetTag(base+12,false);
    end;
    if not GetTag(base+side*10+11) then
      DeInit_Lab(side,art); // deleting icon from lab
    SetTag(base+11,0);
  end;
end;
```

GetResourceArea(ar:integer, typ:integer)

zwraca wszystkie zasoby określonego typu w określonej strefie

//Przykład - Misja AM 15a

```
every 0$5.4 trigger TICK>load_local_cpu do
var un;
begin
  if GetResourceArea(us_crates,mat_cans)<250 then
    CreateCratesArea(Rand(3,5),us_crates,true);
  if GetResourceArea(ru_crates,mat_cans)<250 then
    CreateCratesArea(Rand(3,5),ru_crates,true);
  load_local_cpu:=TICK+Rand(cpu_delay - 0$20,cpu_delay + 0$20);
  enable;
end;
```


GetCargoType(units:plist)

zwraca typ ładunku (mat_none, mat_cans, mat_oil, mat_siberite, mat_multi)

//Przykład - Misja RU 05 cont

```
export function Save;
var cars, i, cargotype;
begin
    SaveCharacters(Omar, 'Omar');
// nas hrdina
    SaveCharacters(Burlak, 'Burlak');
// novi v misi
    if not IsInUnit(Titov) then SetLives(Titov, 0);
    if not IsInUnit(Dolgov) then SetLives(Dolgov, 0);
    if not IsInUnit(Lipshchin) then SetLives(Lipshchin, 0);
    if not IsInUnit(Karamazov) then SetLives(Karamazov, 0);
    if not IsInUnit(Petrovova) then SetLives(Petrovova, 0);
    if not IsInUnit(Gleb) then SetLives(Gleb, 0);
    if not IsInUnit(Petrosyan) then SetLives(Petrosyan, 0);
    {...}

    SaveVariable(Stevens_killed, 'StevensKilled');
    SaveVariable(action_was_fast, 'OvesyenkoTime');

    cars:=[];
    for i in FilterUnitsInArea(exit_rus, [[f_side,you],[f_type,unit_vehicle]]) do begin
        cargotype:=GetCargoType(i);
        cars:=cars ^ [[GetChassis(i),GetWeapon(i),GetNation(i),cargotype,GetCargo(i,cargotype)]];
    end;
    SaveVariable(cars, 'cars_ru_6');
end;
```

GetCargo(units:plist, mattype:integer)

zwraca (łącznie) ilość mattype

//Przykład - Misja AM 12

```
//odjezd prazdnych nakladaku
var odjet,odjet2,odjet3;
every 0$1+0$0.1 trigger (isplaced(c11) and (not getcargoc(c11,mat_cans)) and (not getcargoc(c11,mat_siberit))) or (isplaced(c12) and not
getcargoc(c12,mat_siberit)) marked 5 do
var tmp;
begin
    if isplaced(c12) and (not getcargoc(c12,mat_siberit)) then begin tmp=c12;c12=0;end
    else begin tmp=c11;c11=0;end;
    setside(tmp,am_odj);
    commovexy(tmp,150,101);
    odjet=odjet-1;
    if odjet>0 then enable;
end;
```

SetCargo(units:plist, mattype:integer, amount:integer)

ustala ilość mattype (każdej jednostki na liście) zwraca true jeżeli (wszystkie) udane

//Przykład - Misja RU 08

```
case i of
1:
begin
    SetCargo(auto[1],mat_cans,100);
    ComMoveXY(auto[1],ShiftX(x,d,5),ShiftY(y,d,5));
    Wait(1);
    AddComExitVehicle(auto[1]);
end;
{...}
end;
```

AddCargo(units:plist, mattype:integer, amount:integer)

dodaje ilość mattype (każdej jednostki na liście) zwraca true jeżeli (wszystkie) udane

//Przykład - Misja AM 10 cont

```
EVERY 0$1 DO
```

```
begin
  ComEnterUnit(LS,Ar_lab);
  AddComChangeProfession(LS,class_scientistic);
  ComEnterUnit(tapkal,Ar_bar);
  ComEnterUnit(Dietrich,Ar_bar);
  AddComChangeProfession(Dietrich,class_soldier);
  {...}
```

```
// nejakie surowiny do aut
AddCargo(auto1,mat_siberit,20);
AddCargo(auto1,mat_oil,30);
AddCargo(auto2,mat_cans,40);
AddCargo(auto2,mat_oil,40);
end;
```

See(side:integer, un:integer)

zwraca true jeżeli strona widzi określoną jednostkę

//Przykład - Misja AM 04

```
every 0$1 trigger see(amer,rubdepo0) or see(amer,rubwrks0) do
begin
  hint('Capturing2');
end;
```

SeeXY(side:integer, x:integer, y:integer)

zwraca true gdy strona dobrze widzi określony hex

//Przykład - Misja Am 01 cont

```
// russians guards, burlak cannot pass
every 0$1 trigger not SeeXY(russians,86,53) marked 1 do
  var un;
  begin
    if pokryshkin_met then
      exit;
    un:=team[Rand(1,team)];
    ComMoveXY(un,86,53);
    repeat
      Wait(0$1);
    until SeeXY(russians,86,53) or pokryshkin_met;
    enable;
  end;
```

SeeArea(side:integer, area:integer)

zwraca true jeżeli określona strefa jest widoczna dla strony

//Przykład - Misja RU 05 cont

```
// Dialogue Arabs (when Arabians are first spotted)
every 0$5 trigger dialogy and (SeeArea(you, vidi_je) or See_any(you, fort)) do
begin
  DArabs;
  DSurrender;
end;
```

GetFreeMode(un:integer)

zwraca true jeżeli jednostka jest w trybie wolnym

{brak przykładu}

GetDistUnits(un1:integer, un2:integer)

zwraca odległość dwóch jednostek, 99999 jeżeli jedna lub obie nie są na mapie

//Przykład - Misja RU 01 cont

```
if GetDistUnitXY(Burlak,76,53)>15 then
begin
Say(Belkov,'D7b-Bel-3');
repeat
wait(0$2);
until GetDistUnits(Burlak,Belkov)<10;
Say(Belkov,'D7b-Bel-3a');
end;
```

GetDistUnitXY(un1:integer, x:integer, y:integer)

zwraca odległość między jednostką a hex'em, 99999 jeżeli jednostka nie jest na mapie

//Przykład - Misja AM 15

```
// Alliance units go out of the map
EVERY 0$1 MARKED 16 DO
var Un;
begin
for Un in Captured_Units_of_Alliance do
if GetDistUnitXY(Un, 226, 197) < 2 then
begin
RemoveUnit(Un);
Captured_Units_of_Alliance= Captured_Units_of_Alliance diff Un;
end;
if Captured_Units_of_Alliance > 0 then
begin
enable;
for Un in Captured_Units_of_Alliance do
ComMoveXY(Un, 226, 197);
end;
end;
```

GetDistXY(x1:integer, y1:integer, x2:integer, y2:integer)

zwraca odległość dwóch hex'ów

//Przykład - Misja RU 10 cont

```
if GetDistXY(mines_list[(mine*2)-1],mines_list[(mine*2)],GetX(un),GetY(un))<dist then
begin
LaunchMineAtPos(mines_list[(mine*2)-1],mines_list[(mine*2)],enemy);
if not first_mine_exploded then
begin
first_mine_exploded:=true;
mine_xy:=[mines_list[(mine*2)-1],mines_list[(mine*2)]];
end;
end;
```

GetDistUnitArea(un:integer, area:integer)

zwraca odległość między jednostką i najbliższym hex'em w strefie (0 jeżeli to jest w strefie)

//Przykład - Misja RU 06

```
function buch_crate; //bedny padaji do areii nejlepe blizko Burlaka
var a, h, c;
begin
  a:=IsInUnit(Burlak);
  if not a then
    a:=Burlak;

  repeat
    c:=c+1;
    h:=RandHexArea(bedny, false);
  until c>100 or GetDistUnitXY(Burlak, h[1], h[2])<25 or GetDistUnitArea(Burlak, bedny)>15;
  CreateCratesXY(5, h[1],h[2],true);
  padlo:=padlo+1;
end;
```

IsLive(un:integer)

zwraca true jeżeli jednostka jest na mapie i nie zginęła

//Przykład - Misja AM 06 cont

```
//Dialog9
every 0$1.0 trigger Joan_P and not IsLive(Joan) do
begin
  ChangeMissionObjectives('MJa');
end;
```

IsDead(un:integer)

zwraca true jeżeli jednostka jest martwa (zero życia lub doszczętnie zniszczona)

//Przykład - Misja AM 11

```
// zmena technologii v arabske pri zisku arabske zakladny
EVERY 0$10 TRIGGER GetSide(A1_tov) or IsDead(A1_tov) DO
begin
  SetTech(tech_Flame,You,state_researched);
  SetTech(tech_SibPow,You,state_researched);
  SetTech(tech_SibEng,You,state_researched);
  SetTech(tech_Mortar,You,state_researched);
end;
```

IsOK(un:integer)

zwraca true jeżeli jednostka jest na mapie i nie umiera

//Przykład - Misja AM 14

```
// Ruska hlidka na obhlidce silnice - moznost nalezeni MacMillanova auta
function ukradni_macmillanovi_auto;
var un;
begin
  for un in ru_sol do
    if IsOK(un) then
      ComEnterUnit(un,truck1);
  AddComAgressiveMove(ru_sol,69,35);
  AddComReturn(ru_sol);
  wait(0$20);
end;
```

IsDying(un:integer)

zwraca true jeżeli jednostka jest na mapie i umiera

//Przykład - Misja AM 03

```
if IsDead(MacMillan) or IsDying(MacMillan) then
begin
  YouLost('0');
  exit;
end;
```

IsConstructed(build:integer)

zwraca niezerową liczbę, jeżeli budowany budynek jest w budowie lub jeżeli jest ulepszany

//Przykład - Misja AM 10

```
// uklad jednostek po stavbe armoury
EVERY 0$10 marked 6 do
begin
  if IsConstructed(arm)=0 then
  begin
    ComEnterUnit(Afi[1],dep);
    enable(8);
  end
  else enable;
end;
```

IsPlaced(un:integer)

zwraca true jeżeli jednostka un jest umieszczona

//Przykład - Misja AM 12

```
every 0$1 trigger isplaced(kurt) and (not hastask(kurt_veh)) marked 99 do
begin
  commovexy(kurt_veh,150,101);
  enable;
end;
```

IsSelected(units:plist)

zwraca true jeżeli zbiór jednostek jest zaznaczony

//Przykład - Misja RU 10 cont

```
repeat
  wait(0$1);
until IsSelected(artefact);
```

IsAt(un:integer, x:integer, y:integer)

zwraca true jeżeli jednostka jest w określonych koordynatach

//Przykład - Misja AM 02

```
// John
if IsOk(John) And not(Skip) then ComWait(John, 0$0.5);
if IsOk(John) And not(Skip) then AddComTurnUnit(John, Lisa);
if not(IsAt(John, 36, 63)) then
begin
  ComMoveXY(John, 36, 63);           // At any time
  AddComTurnUnit(John, Lisa);
end;
```

IsInArea(un:integer, area:integer)

zwraca true jeżeli jednostka jest w określonej strefie (może prowadzić lub być wewnątrz)

//Przykład - Misja Am 14

```
function at_home(list);
var un,vysledek;
begin
  vysledek:=true;
  for un in list do
    if IsOK(un) and not IsInArea(un,RU_Base) then
      begin
        vysledek:=false;
        break;
      end;
  result:=vysledek;
end;
```

InArea(x:integer, y:integer, area:integer)

zwraca true jeżeli [X,Y] jest w określonej strefie

//Przykład - Misja RU 10 cont

```
every 0$2 trigger FilterUnitsInArea(under_attack, [ [f_side,you] ]) do
begin
  enable(7);
end;
```

IsInUnit(hum:integer)

zwraca ident jednostki, w której jednostka przebywa (np. człowiek w budynku) lub zero

//Przykład - Misja AM 03

```
if IsInArea(MacMillan,Am_Base) and not (GetType(IsInUnit(MacMillan))=unit_building) then
begin
  ComMoveXY(MacMillan,62,91);
  hold_list:=hold_list ^ MacMillan;
end;
```

IsDrivenBy(veh:integer)

zwraca ident kierowcy od człowieka prowadzącego pojazd, inaczej zero

//Przykład - Misja RU 10 cont

```
every 0$1 Marked 7 do
var un,list;
begin
  list:=FilterUnitsInArea(base_flee, [ [f_side,enemy] ]);

  for un in list do
    if UnitsInside(un) then
      begin
        RemoveUnit(IsDrivenBy(un));
        RemoveUnit(un);
      end
    else
      RemoveUnit(un);

  enable;
end;
```

IsControlledBy(veh:integer)

zwraca kierowcy kierowcy zdalnie sterującego dany pojazd, inaczej zero

//Przykład - Misja AM 06 cont

```
every 0$3 do
var ok, sez, i;
begin
  sez=[];
  ok:=false;
  for i in FilterAllUnits([f_control,control_remote]) do begin
    if IsControlledBy(i) in sez then begin
      ok:=true;
      break;
    end
  else
    sez:=sez ^ i;
  end;
  if ok then begin
    ChangeMissionObjectives('MRa');
  end
  else
    enable;
end;
```

UnitsInside(un:integer)

zwraca listę jednostek wewnątrz budynku (lub pojazdu) un

//Przykład - Misja RU 03

```
// wyroba v americke tovarne (pokud jsou v ni vic nez 3 mechanici)
EVERY 0$17 DO
VAR pom;
begin
  pom=UnitsInside(Am_fac);
  if pom>3 then ComConstruct(Am_fac,us_light_wheeled,engine_solar,control_manual,us_machine_gun);
  while UnitsInside(Am_fac)>3 do wait(0$10);
  enable;
end;
```

HasTask(un:integer)

zwraca true jeżeli jednostka ma zadanie od gracza lub SAIL'a

//Przykład - Misja AM 03

```
ComReturn(Frank);
repeat
  wait(10);
  if GetLives(Frank)<900 then
    SetLives(Frank,GetLives(Frank)+50);
until not HasTask(Frank);
RemoveUnit(Frank);
```

IsBusy(un:integer)

zwraca true jeżeli jednostka ma zadania (nawet własne)

//Przykład - Misja AM 02

```
// MARKED 13 - mikhails_escape
EVERY 0$1 TRIGGER IsOk(Mikhail) and not(IsBusy(Mikhail)) MARKED 13 do
begin
  ComMoveXY(Mikhail, target_of_escape[1], target_of_escape[2]);
end;
```

IsIdle(un:integer)

zwraca true jeżeli jednostka nic nie robi

//Przykład - Misja RU 05 cont

```
export function f_arcar1_exit;
begin
  while not IsInArea(ArScout1, ar_in_area) do begin
    if IsIdle(arscout1) then begin
      ComAgressiveMove(ArScout1,127,135);
      AddComAgressiveMove(ArScout1,185,134);
      // AddComMoveToArea(ArScout1, ar_in_area);
    end;
    DU_wait(ArScout1);
  end;
  RemoveUnit(ArScout1);
end;
```

IsTamedBy(un:integer)

zwraca numer tresera danego zwierzęcia, inaczej zero

{brak przykładu}

Crawls(units:plist)

zwraca ident czołgających się jednostek na liście (0=false)

//Przykład - Misja AM 01

```
every 0$0.4 trigger crawls(john) marked 04 do
begin
  // dialogueon;
  if not killhim then say(joan,'DXComments-Joan-1');
  // dialogueoff;
end;
```

WantsToAttack(un:integer)

zwraca ident jednostki której celem jest komenda ataku na jednostkę un lub zero

//Przykład - Misja RU 08

```
every 0$1+0$0.3 marked 102 do // Hunter crawl and walk + Hunting done
begin
  if GetDistUnits(hunter,WantsToAttack(hunter)) < 13 then
    ComCrawl(hunter)
  else
    ComWalk(hunter);
  if NOT HasTask(hunter) then
    disable
  else
    enable;
end;
```

Attacks(un:integer)

zwraca ident jednostki która obecnie jest atakowana przez jednostkę un (cel liczy się jako atak) lub zero

//Przykład - Misja AM 01 cont

```
function someone_attacks_breastwork;
var un;
begin
  result:=false;
  for un in your_units do
    if Attacks(un)=am_bunker then
      result:=true;
  end;
```


GetTech(tech:integer, side:integer)

zwraca stan technologii dla strony

//Przykład - Misja AM 15a

```
DeInit_Lab(size);
if GetTech(tech_Artifact,you)=state_enabled then
state:=art_gray
```

SetTech(tech:integer, side:integer, state:integer)

ustala stan technologii dla strony

//Przykład - Misja AM 15

```
// Set technologies to researched
SetTech(tech_Gun, americans, state_researched);
SetTech(tech_SolPow, americans, state_researched);
SetTech(tech_SolEng, americans, state_researched);
```

GetRestrict(btype:integer, side:integer)

zwraca ograniczenia btype dla strony

//Przykład - Misja RU 08

```
function read_parameters;
var i;
begin
you:=3;
{...}
AvailableBuildings=[];
for i:=0 to 50 do
if GetRestrict(i,you)=1 then
AvailableBuildings:=AvailableBuildings^i;
AvailableBuildings:=AvailableBuildings diff [0,2,4,6,7,8,9,10,11,12,14,22,31,32,33,39,40];
{...}
end;
```

SetRestrict(btype:integer, side:integer, state:boolean)

ustala ograniczenia btype dla strony

//Przykład - Misja AM 05 cont

```
if tech=tech_Radar then begin
ChangeMissionObjectives('M2a');
SetRestrict(b_ext_radar, you, state_enabled);
MyHint('Extensions');
end;
```

Researched(side:integer, tech:integer)

zwraca true jeżeli tech jest przebadana (obsolete)

//Przykład - Misja AM 10

```
// vyzkum 3
EVERY 0$43 TRIGGER Researched(Arabi,tech_Sib2) DO
begin
wait(1$0);
SetTech(tech_Sib3,Arabi,state_enabled);
ComResearch(lab,tech_Sib3);
end;
// utok arabu pokud je to moc rychle !!!
EVERY 0$13 TRIGGER Researched(You,tech_Sib1) DO
begin
arabi_utoci=true;
end;
```

GetTechProgress(tech:integer, side:integer)

zwraca postęp przebadania technologii dla strony

//Przykład - Misja RU 06

```
every 0$5 do
begin
// mezeitim mohla spadnout bedna proto je tu test
if [(padlo<4),(padlo<4),(padlo<3)][difficulty] then
case true of
//nutno zachovat opacne poradí
GetTechProgress(tech_MatPred, you)>95 : buch_crate;
GetTechProgress(tech_MatPred, you)>70 : buch_crate;
GetTechProgress(tech_MatPred, you)>40 : buch_crate;
GetTechProgress(tech_MatPred, you)>1 : buch_crate;
end;
// mezeitim muze spadnout bedna proto je tu test znovu
if [(padlo<4),(padlo<4),(padlo<3)][difficulty] then
enable;
end;
```

SetTechProgress(tech:integer, side:integer, percentage:integer)

ustala postęp przebadania technologii dla strony

{brak przykładu}

GetWorkingProgress(units:plist)

zwraca (przeciętny) postęp działań jednostki(ek) - budowa etc.

{brak przykładu}

SetWorkingProgress(units:plist, percentage:integer)

ustala postęp działań jednostki(ek) - budowa etc. (nie badań!)

//Przykład - Misja AM 15

```
// Control of siberite rocket construction
Every 0$1 TRIGGER Construction_of_Arabian_rocket and IsOk(GetUnitNumber(78, 12)) MARKED 10 DO
var progress;
begin
if (tick < Arabian_siberite_attack_time) and (Number_of_Arabian_rocket = 1) and
IsOk(GetUnitNumber(78, 12)) and GetSide(GetUnitNumber(78, 12)) = arabs then
begin
progress= 100*(tick-Starting_Construction_Time)/
(Arabian_siberite_attack_time-Starting_Construction_Time) div 1;
if progress >0 then
progress= progress-1;
SetWorkingProgress(GetUnitNumber(78, 12), progress);
enable;
end
else if (tick >= Arabian_siberite_attack_time) and (Number_of_Arabian_rocket = 1) and
IsOk(GetUnitNumber(78, 12)) then
begin
SetWorkingProgress(GetUnitNumber(78, 12), 99);
end;
end;
```

GetListOfCratesInArea(area:integer)

zwraca listę pozycji skrzynek w strefie

//Przykład - Misja RU 05 cont

```
every 0$5 do
var a, i;
begin
  crates:=[];
  a:=GetListOfCratesInArea(ar_crates_site);
  for i:=1 to (a div 2) do
    crates:=crates ^ [[a[i*2-1],a[i*2]]];
  enable;
end;
```

GetUnitActivity(un:integer)

zwraca jednostki aktywne (stałych w UDatStr.pas)

{brak przykładu}

GetTaskList(un:integer)

zwraca listę list [task,x,y,obj,par1,par2,patrol] siiiib (UDatStr - zadanie...)

//Przykład - Misja RU 08

```
on Command(comm) marked 601 do // trigger s Kurtem a Mercenary1, aby nedelali nic, co nemaji
var i;
begin
  {...}
  if IsOK(Mercenary1) then
    if GetTaskList(Mercenary1)<>[] then
      begin
        for i:=1 to (0+GetTaskList(Mercenary1)) do
          begin
            if (GetTaskList(Mercenary1)[i][1] in ['F','H','M','U','V','a','h','u','v','~','^','>','+',';','4','{']) OR (GetTaskList(Mercenary1)[i][1]='B' AND
GetTaskList(Mercenary1)[i][5]=36) then
              else
                begin
                  RemoveTasks(Mercenary1);
                  D_MercRefuseBuild;
                end;
              end;
            end;
          end;
        end;
      end;
end;
```

GetUnitNumber(x:integer, y:integer)

zwraca ident jednostki na pozycji x, y

//Przykład - Misja AM 02

```
EVERY 0$5 TRIGGER GetResourceType(GetBase(GetUnitNumber(52, 10)), mat_cans) > 20 DO
begin
  Hint('Building');
end;
```

SetClass(units:plist, value:integer)

ustala od razu klasę jednostek do wartości (value)

//Przykład - Misja AM 12a cont

```
every 0$0.47 trigger isinarea(john,weaplab) do
begin
  setclass(john,class_bazooker);
  enable(02);
end;
```

UnitsLinked(un:integer)

zwraca numer podpiętych pojazdów do jednostki

//Przykład - Misja RU 06

```
every 0$3 do
var i, cm;
begin
  if ctrl then begin
    cm:=UnitFilter(UnitsInside(ctrl[1]), [[f_class, class_mechanic]]);
    for i in cm do
      if UnitsLinked(i)>= GetSkill(i, skill_mechanical) then
        cm:=cm diff i;
    for i in attack do begin
      if not IsControlledBy(i) then
        case cm of
          1: ComLinkTo(i, cm[1]);
          2: if UnitsLinked(cm[1]) <= UnitsLinked(cm[2]) then
              ComLinkTo(i, cm[1])
            else
              ComLinkTo(i, cm[2]);
          3: if UnitsLinked(cm[1]) <= UnitsLinked(cm[2]) then
              if UnitsLinked(cm[1]) <= UnitsLinked(cm[3]) then
                ComLinkTo(i, cm[1])
              else
                ComLinkTo(i, cm[3])
            else
              if UnitsLinked(cm[2]) <= UnitsLinked(cm[3]) then
                ComLinkTo(i, cm[2])
              else
                ComLinkTo(i, cm[3]);
        end;
    end;
  end;
end;
enable;
end;
```

FindMaxSkill(units:plist, skill:integer)

zwraca jednostkę z max umiejętnością z units

//Przykład - Misja AM 15a

```
every 0$3.9 trigger GetTag(281) do
var un,sci;
begin
  un:=FilterAllUnits([ [f_side,you], [f_nation,nation_russian], [f_sex,sex_male] ]);
  un:=un diff [MacMillan, Burlak, Roth, Gossudarov, 0];
  if un then
    sci:=FindMaxSkill(un,skill_scientistic);

  { ... }
end;
```

FindMaxSkill2(units:plist, skill:integer)

zwraca jednostkę z max umiejętnością z units (w tym nieumieszczone)

//Przykład - Misja RU 10

```
function prepare_units;
var un,x,y,placed,num,veh,important,depot_list,depot,factory_list,mechanics,lab_list,scientists;
begin
  { ... } for un=1 to num do
    mechanics:=mechanics ^ FindMaxSkill2(people diff mechanics,skill_mechanical);

  for un=1 to num do
    scientists:=scientists ^ FindMaxSkill2(people diff (mechanics ^ scientists),skill_scientistic); { ... }
end;
```

FindNearestPoint(list_of_points:plist, x:integer, y:integer)

zwraca najbliższy punkt x y z list_of_points

{brak przykładu}

GetResourceVisibility(x:integer, y:integer, side:integer)

zwraca true, jeżeli dostawa na pozycji x, y jest widoczna dla strony

//Przykład - Misja RU 06

```
every 0$2 trigger GetResourceVisibility(101,93,you) marked 66 do
begin
    DSibResearch;
end;
```

SetResourceVisibility(x:integer, y:integer, side:integer)

ustala dostawę na pozycji x, y widoczną dla strony

//Przykład - Misja RU 05 cont

```
every 0$0.1 do
begin
    SetResourceVisibility(131, 51, arábians);
    case difficulty of
        1: objeovani_zasilek:=[3,6,4,4,1,4,2,2,3,5,0,2,3,5,1,2,1,2,1,2,1,2];
        2: objeovani_zasilek:=[2,5,3,2,1,3,2,2,3,4,0,2,3,4,0,2,0,2,0,2,0,2];
        3: objeovani_zasilek:=[2,4,2,2,0,3,1,2,3,3,0,2,3,2,0,2,0,2,0,2,0,2];
    end;
    ar_objeovani_zasilek:=[5,6,4,4,2,5,2,4,6,5,2,4,6,5,5,4,5,2,9,8,6,5];
    Randomize;
end;
```

GetListOfResourcesInArea(area:integer)

zwraca listę pozycji dostaw zgodnie z typem, [x,y,typ,...]

//Przykład - Misja AM 15a

```
// nalezeni vseh siberitovych lozisek v ruske zakladne
function get_sib_list;
var list,un,
    vysledek;
begin
    vysledek:=[];
    list:=GetListOfResourcesInArea(ru_base);
    for un=1 to (list div 3) do
        if list [(un*3)]=source_siberium then
            vysledek:=vysledek ^ [ list[(un*3)-2], list[(un*3)-1] ];
    end;
    result:=vysledek;
end;
```

FindMinSkill(units:plist skill:integer)

zwraca jednostkę z min umiejętnością z units

//Przykład - Misja AM 15

```
3: begin
    Un= FindMinSkill(Uns, skill_mechanical);
    The_original_mechanical_Skill= GetSkill(Un, skill_mechanical);
    SetSkill(Un, skill_mechanical, 5); // 10
    Uns= Uns diff Un;
    ComEnterUnit(Un, AmFact);
    AddComChangeProfession(Un, class_mechanic);
end;
```

Misje:

ChangeMissionObjectives(ident:string)

zmienia cele misji, ident jest identyfikatorem napisu z Texts.txt

//Przykład - Misja AM 12a cont

```
on enterbuilding(buil,un) do
  if (un=john) and (buil=amlab) then begin
    setclass(john,class_bazooker);
    settech(tech_soporific,al,state_researched);
    changemissionobjectives('M1a');
    enable(02);
  end;
```

HintSpec(ident:string, kind:integer)

dodaje podpowiedź (rodzaj: 0-hint, 1-info, 2-error, 3-first)

//Przykład - Misja AM 01

```
// saveforquickrestart;
wait(0$0.3);
changemissionobjectives('M0');
hintSpec('Hints',3);
wait(0$1);
hint('Character');
wait(0$1);
```

Hint(ident:string)

dodaje podpowiedź

//Przykład - Misja AM 02

```
EVERY 0$2 TRIGGER See(americans, dep1) or See(americans, dep2) do
  begin
    Hint('CapturingBuilding');
  end;
```

ChangeMap(ident:string, fname:string)

zmienia mapę, ident identyfikuje tekst podczas ładowania, fname jest nazwą mapy w tym samym katalogu (bez ścieżki)

//Przykład - Misja AM 14a

```
// Save dulezitych promennych a reload mapy po odjezdu obou hrdinu
every 0$1 trigger not (IsOK(MacMillan) or IsOK(Burlak)) do
  begin
    SaveCharacters(MacMillan,'JMM');
    SaveCharacters(Burlak,'Burlak');

    SaveVariable(JMM_car,'JMM_car');
    SaveVariable(Bur_car,'Bur_car');
    SaveVariable(JMM_load,'JMM_car_load');
    SaveVariable(Bur_load,'Bur_car_load');

    ChangeMap('%_cont','_%cont');
  end;
```

ExitGame(var res:tvalue)

kończy grę (debug)

{brak przykładu}

Widzenie/Mgła:

PlaceSeeing(x:integer, y:integer, side:integer, range:integer)

umieszcza strefę widzenia (odrzuca pole przy przeglądaniu radarem)

```
//Przykład - Misja AM 15
```

```
Function JMM_warns;  
var Remaining_americans, Remaining_unnamed_males, sold;  
begin  
  placeseeing(157, 7, americans, -15);  
  placeseeing(165, 21, americans, -15);  
  CenterNowOnUnits([Omar, Heike]);  
  Wait(0$0.5);  
  {...}  
end;
```

RemoveSeeing(x:integer, y:integer, side:integer)

usuwa strefę z widzenia (wszystkie mieszczące się strefy)

```
//Przykład - Misja 15a
```

```
every 0$1 trigger view_data do  
begin  
  wait(0$10);  
  RemoveSeeing(view_data[1],view_data[2],view_data[3]);  
  view_data:=0;  
  enable;  
end;
```

RevealFogArea(side:integer, area:integer)

ujawnia mgłę dla określonej strony w określonej strefie

```
//Przykład - Misja AM 03
```

```
every 0$5 Marked 5 do  
begin  
  wait(0$10);  
  RevealFogArea(you,odkryj_mapu2);  
  SayRadio(Frank,'D3b-Frank-1');  
end;
```

HideArea(side:integer, area:integer)

ustala określonej strefie niewidoczność dla określonej strony

```
{brak przykładu}
```

ShowArea(side:integer, area:integer)

ujawnia na strefie to, co ukryte zostało przez HideArea

```
{brak przykładu}
```

ResetFog(var res:tvalue)

resetuje mgłę wojny dla wszystkich stron na czarno

```
//Przykład - Alien base
```

```
your_side:=Player_Side;  
ResetFog;  
prepare_specfog;
```

GetSideFog(side:integer)

zwraca numer mgły określonej strony

//Przykład - Misja RU 07

```
Function set_attitudes;
Begin
  // Burlak a Xavier sdílejí mlhu a jsou kámoši.
  ChangeSideFog (side_Xavier, GetSideFog (side_Ru));
  ChangeSideFog (side_Ru2, GetSideFog (side_Ru));
  {...}
End;
```

ChangeSideFog(side:integer, fognumber:integer)

ustal mgłę określonej strony do określonego numeru mgły

//Przykład - Misja AM 04

```
// retreat of remaining russian units
every 0$1 trigger filterallunits([[f_side,rus],[f_ok],[f_type,unit_human]]<=5 do
begin
  zbytek=filterallunits([[f_side,rus],[f_ok],[f_type,unit_human]]);
  commovetoarea(zbytek,retr_ar); //124,95);
  changesidefog(rus,amer);
  setattitude(rus,amer,att_friend,false);
  commovetoarea(filterunitsexceptarea(targetarea,[[f_side,amer],[f_ok],[f_type,unit_human]]),retr_ar);
  ingame_video=true;
  interface_hidden=true;
  centeronunits(zbytek);
  retreat=true;
end;
```

FogOff(sw:integer)

jeżeli sw jest true, mgła jest wyłączona

```
starting
begin
  FogOff(false);

  Randomize;
  RandomizeAll;

  {...}
end;
```

BuildingsInProgress(side:integer)

zwraca true jeżeli budowa wszelkich budynków określonej strony nie jest skończona

//Przykład - Misja RU 05 cont

```
while BuildingsInProgress(arabians) do
  wait(0$1);
```

Środowisko:

ReplaceEnvironment(x:integer, y:integer, envset:integer, num:integer)

zastępuje środowisko na x, y przez elementy num z zasobu

{brak przykładu}

RemoveEnvironment(x:integer, y:integer)

usuwa środowisko na x, y

//Przykład - Misja RU 01 cont

```
if difficulty <> 1 then
  RemoveEnvironment(48,26);
if difficulty <> 2 then
  RemoveEnvironment(48,27);
bunker_see = [ See1, See2, See3 ] [difficulty];
```

RemoveEnvironmentWithoutRebuild(x:integer, y:integer)

usuwa środowisko na x, y, nie liczą się parametry, używane tylko wtedy gdy normalne RemoveEnvironment następuje

//Przykład - Misja AM 15

```
Function LaunchMines;
var mines_pos, mines_pos1, i, j;
begin
  // Mines positions on the stones
  mines_pos= [[168,18], [169,20], [167,20], [164,18], [164,19], [163,19],
             [164,21], [165,21], [163,20], [164,22], [162,20]];
  // Mines positions on the factory
  mines_pos1= [[169,34], [169,35], [167,32], [170,31], [173,35], [174,39]];

  // Launch the mines
  for i= 1 to mines_pos do
    begin
      // Launch the environment
      MineExplosion(mines_pos[i][1], mines_pos[i][2], 1);
      Wait(Rand(1, 3));
      if i < mines_pos then
        RemoveEnvironmentWithoutRebuild(mines_pos[i][1], mines_pos[i][2])
      else
        begin
          // Launch the factory
          for j= 1 to mines_pos1 do
            begin
              MineExplosion(mines_pos1[j][1], mines_pos1[j][2], 1);
              Wait(Rand(1,3));
            end;
          RemoveEnvironment(mines_pos[i][1], mines_pos[i][2]);
        end;
    end;
  end;
end;
```

PlaceEnvironment(x:integer, y:integer, envset:integer, num:integer)

tworzy środowisko num z zasobu envset na x, y

//Przykład - Alien base

```
// eventually place flag
if def_morale_flags then
begin
  PlaceEnvironment(ShiftX(x,dir,10),ShiftY(y,dir,10),12,2);
  battle_flags_id:=battle_flags_id ^ SetBattleFlag(side,ShiftX(x,dir,10),ShiftY(y,dir,10),
            [ [bfo_defend_bonus_human,100],
              [bfo_defend_bonus_vehicle,50],
              [bfo_defend_bonus_building,20],
              [bfo_height,100] ]);
end;
```

PlaceRandEnvironment(x:integer, y:integer, envset:integer)

tworzy losowe środowisko z zasobu envset na x, y

{brak przykładu}

IsEnvironment(x:integer, y:integer)

zwraca true jeżeli jest środowisko na x, y

{brak przykładu}

PlaceTreesToArea(area:integer, sets:plist, count:integer, trials:integer, dist:integer)

tworzy count drzew w strefie, ustawia wybrane losowo z listy zasobów dla każdego drzewa tworzy taką samą ilość prób (trials) i wybiera z jedno mniejszego zakresu macierzystego dist uwaga drzewo w odległości 1 jest dwa razy słabsze, niż drzewo w odległości 2 etc.

//Przykład - Misja RU 08

```
PlaceTreesToArea(TreeArea1,[2,10,11,27,35],40,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
PlaceTreesToArea(TreeArea2,[9,10,27,35,48],30,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
PlaceTreesToArea(TreeArea3,[2,9,10,11,27],60,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
PlaceTreesToArea(TreeArea4,[9,10,27,35,48],90,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
PlaceTreesToArea(TreeArea5,[2,9,10,11,48],120,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
PlaceTreesToArea(TreeArea6,[2,11,27,35,48],45,10,4); //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
```

ListEnvironmentArea(area:integer)

zwraca listę środowisk w strefie [[x,y,envset,num],...]

//Przykład - Misja RU 08

```
SaveVariable(trotskyj,'Trotskyj');
SaveVariable(ListEnvironmentArea(WholeMapArea),'Environment');
```

PlaceEnvironmentList(list:plist)

tworzy środowiska [[x,y,envset,num],[x,y,envset]...] num może nie być określone

//Przykład - Misja RU 09

```
// sazeni stromecku do mapy
function prepare_map;
var map_env;
begin
  map_env:=LoadVariable('Environment',[]);
  if map_env then
    PlaceEnvironmentList(map_env);
end;
```

RemoveEnvironmentArea(area:integer)

usuwa wszystkie środowiska w strefie

{brak przykładu}

Rakieta Syberytowa:

SiberiteRocket(x:integer y:integer, n:integer, list:tvalue)

odzworowanie n-tą fazę wybuchu rakiety syberytowej na liście jednostek

//Przykład - Misja AM 15

```
case Number_of_Arabian_rocket of
  1: begin
    Construction_of_Arabian_rocket= false; // We will not control it in the next
    Number_of_Arabian_rocket= Number_of_Arabian_rocket+1;
    SiberiteRocketAttack(veh, 46, 64, 1);
  end;
  2: SiberiteRocketAttack(veh, 155, 60, 2);
end;
```

PrepareSiberiteRocket(var res:tvalue)

przygotowuje wybuch rakiety sybertyowej

//Przykład - Misja AM 14a cont

```
DialogueOn;
interface_hidden:=true;
CenterOnXY(103,52);
wait(0$3);
PrepareSiberiteRocket;
SendSiberiteRocket(103,52);           //!!!!!!  VYBUCH  !!!!!
interface_hidden:=false;
DialogueOff;
```

SiberiteExplosion(x:integer, y:integer)

tworzy wybuch sybertyowy

//Przykład - Misja RU 15

```
placeseeing(63,77,ru,11);
centernowonxy(63,77);
wait(0$0.3);
siberiteexplosion(63,77);
```

Grafika:

GetGraphicsGamma(var res:tvalue)

zwraca korekcję gamma z ustawień

{brak przykładu}

SetGraphicsGamma(gamma:integer)

ustala korekcję gamma wyświetlania

{brak przykładu}

GetGraphicsBrightness(var res:tvalue)

zwraca jasność z ustawień

{brak przykładu}

SetGraphicsBrightness(brightness:integer)

ustala jasność wyświetlania

{brak przykładu}

Dźwięk:

PlaySoundXY(x:integer, y:integer, ident:string)

odtwarza dźwięk o idencie na koordynatach x,y

{brak przykładu}

MusicIsPlaying(var res:tvalue)

zwraca true jeżeli muzyka jest odtwarzana

//Przykład - Misja AM 01

```
every 0$1 do
begin
  if not MusicIsPlaying then
  if killhim then
  begin
    InsertMusic('musicdemo2.wav');
    AddMusic("");
  end
  else
    PlayMusic('musicdemo4.wav');
enable;
end;
```

Przygotowanie (Prepare):

PrepareAttr(sex:integer)

przygotowuje losowe atrybuty dla określonej płci (lub losową, jeżeli zero)

{brak przykładu}

PrepareClassSkills(cl:integer, level:integer)

przygotowuje umiejętności człowieka z klasą cl (ta umiejętność jest przybliżeniem określonego level'u)
ujemny level oznacza, że ten człowiek jest specjalistą

//Przykład - Misja RU 14 cont

```
Function vytvor_postavu (Cls);
Var X, Y;
Begin
  // Vytvořit postavu.
  UC_Side = side_Le; UC_Nation = NATION_ARABIAN;
  HC_Sex = Rand (SEX_MALE, SEX_FEMALE);
  HC_Class = Cls;
  PrepareClassSkills (Cls, 7);
  Result = CreateHuman;
End;
```

PrepareSoldierSkills(level:integer)

przygotowuje umiejętności dla żołnierza, główna umiejętność jest w okolicy określonego level'u (używa PrepareClass)

//Przykład - Misja RU 07

```
Function vytvor_yelenu;
Begin
  UC_Nation = NATION_RUSSIAN; UC_Side = side_Ru2;
  HC_name = 'Yelena Trajkov'; HC_sex = SEX_FEMALE;
  HC_class = CLASS_SOLDIER; PrepareSoldierSkills (5);
  Result = CreateHuman;
End;
```

PrepareEngineerSkills(level:integer)

przygotowuje umiejętności dla inżyniera, główna umiejętność jest w okolicy określonego level'u (używa PrepareClass)

//Przykład - Misja RU 13

```
Function vytvor_inzenyra;
Begin
  UC_Nation = NATION_RUSSIAN; UC_side = 3;
  HC_name = "";
  HC_sex = Rand (SEX_MALE, SEX_FEMALE);
  HC_class = CLASS_ENGINEER;
  PrepareEngineerSkills (4);
  Result = CreateHuman;
End;
```

PrepareMechanicSkills(level:integer)

przygotowuje umiejętności dla mechanika, główna umiejętność jest w okolicy określonego level'u (używa PrepareClass)

//Przykład - Misja RU 14 cont

```
Function vytvor_nakladak;
Var Ridic;
Begin
  // Vytvořit auťák.
  UC_Side = side_Upr; UC_Nation = NATION_ARABIAN;
  VC_Chassis = AR_HALF_TRACKED; VC_Control = CONTROL_MANUAL;
  VC_Weapon = AR_CARGO_BAY; VC_Engine = ENGINE_SIBERITE;
  Result = CreateVehicle;
  // Vytvořit řidiče.
  HC_Sex = Rand (SEX_MALE, SEX_FEMALE);
  HC_Class = CLASS_MECHANIC; PrepareMechanicSkills (4);
  Ridic = CreateHuman;
  PlaceHumanInUnit (Ridic, Result);
End;
```

PrepareScientistSkills(level:integer)

przygotowuje umiejętności dla naukowca, główna umiejętność jest w okolicy określonego level'u (używa PrepareClass)

{brak przykładu}

PrepareCustomSkills(list:plist)

przygotowuje losowe umiejętności dla człowieka tworzonoego, podaje listę czterech poziomów jako parametr: jeżeli jeden z parametrów jest ujemny, wtedy ta umiejętność jest jako podstawowa

{brak przykładu}

PrepareHuman(sex:integer, cl:integer, lev:integer)

przygotowuje parametry dla człowieka z określonej płci, klasy i poziomu (0=losowy dla płci lub klasy) ujemny poziom oznacza, że ten człowiek jest specjalistą

//Przykład - Misja RU 04

```
case Rand(1,100) of
  1..10:PrepareHuman(0,class_soldier,Rand(3,6));
  11..60:PrepareHuman(0,class_mechanic,Rand(3,6));
  61..90:PrepareHuman(0,class_engineer,Rand(3,6));
  91..100:PrepareHuman(0,class_scientistic,Rand(3,6));
end;
```

PrepareSoldier(sex:integer, lev:integer)

przygotowuje parametry dla żołnierza z określonej płci i poziomu (0=losowy dla płci)

//Przykład - Misja AM 15

```
case (i mod 4) of
0: begin
  PrepareSoldier(sex_male, 7);
end;
{...}
end;
```

PrepareEngineer(sex:integer, lev:integer)

przygotowuje parametry dla inżyniera z określonej płci i poziomu (0=losowy dla płci)

//Przykład - Misja AM 15

```
case (i mod 4) of
0: begin
  PrepareSoldier(sex_male, 7);
end;
1: begin
  PrepareScientist(sex_male, 7);
end;
2: begin
  PrepareMechanic(sex_male, 7);
end;
3: begin
  PrepareEngineer(sex_male, 7);
end;
end;
```

PrepareMechanic(sex:integer, lev:integer)

przygotowuje parametry dla mechanika z określonej płci i poziomu (0=losowy dla płci)

//Przykład - Misja AM 15

```
case (i mod 4) of
0: begin
  PrepareSoldier(sex_male, 7);
end;
1: begin
  PrepareScientist(sex_male, 7);
end;
2: begin
  PrepareMechanic(sex_male, 7);
end;
{...}
end;
```

PrepareScientist(sex:integer, lev:integer)

przygotowuje parametry dla naukowca z określonej płci i poziomu (0=losowy dla płci)

//Przykład - Misja AM 15

```
case (i mod 4) of
0: begin
  PrepareSoldier(sex_male, 7);
end;
1: begin
  PrepareScientist(sex_male, 7);
end;
{...}
end;
```

Makra:

WaitForMc(mcid:integer)

czeka na zakończenie makro mcid

```
//Przykład - Misja RU 13 cont
```

```
// Aż skończy makro, spustit trigger, który bude přesunovat jednotky do únikové zóny.
```

```
Every 0$1 Marked 12 do
```

```
Begin
```

```
    WaitForMc (MakroMove);
```

```
    Enable (13);
```

```
End;
```

ExistMc(mcid:integer)

jeżeli makro mcid istnieje, zwraca true, inaczej false

```
//Przykład - Misja AM 15
```

```
if not(ExistMc(Russian_reg_id)) then
```

```
    begin
```

```
        Russian_reg_id= McRegistry(russians, [[mc_reg_refresh_time, 0$5],
```

```
            [mc_reg_area_to_guard, attack_area],
```

```
            -mc_reg_only_important,
```

```
            mc_reg_ignore_fog,
```

```
            mc_reg_buildings]);
```

```
    end {...}
```

KillMc(mcid:integer)

jeżeli makro mcid istnieje, zamyka makro i zwraca true inaczej zwraca false

```
//Przykład - Misja AM 15
```

```
EXPORT Function AlliancePersonIsDead;
```

```
begin
```

```
    if FilterAllUnits([[f_side, alliance], [f_alive], [f_type, unit_human]]) = 0 then
```

```
        begin
```

```
            KillMc(Alliance_reg_id);
```

```
            KillMc(Alliance_def_id);
```

```
            KillMc(Alliance_wood_reg_id);
```

```
            Disable(Not_end_of_alliance);
```

```
        end;
```

```
end;
```

GetUnitMc(un:integer)

zwraca mcid makra z użyciem jednostki un zwraca zero jeżeli nie użyta

```
//Przykład - Misja AM 14
```

```
every 0$5 trigger GetListOfCratesInArea(ru_crates) do
```

```
var un,free_unit,x,y,max,i,clear_crates,
```

```
    old_class,attempt;
```

```
begin
```

```
    {...}
```

```
    i:=russians_forces diff reserved_people;
```

```
    for un in i do
```

```
        begin
```

```
            if not (HasTask(un) or GetUnitMc(un)) then // pouzít HasTask nebo radsí IsBusy ???????????
```

```
                begin
```

```
                    free_unit:=un;
```

```
                    break;
```

```
                end;
```

```
            end;
```

```
        {...}
```

```
        Enable;
```

```
end;
```

GetMcUnits(mcid:integer)

zwraca listę jednostek z makrem mcid

//Przykład - Misja RU 05 cont

```
every 0$2 do
begin
  todef1:=fort ^ attack ^ defence ^ mor ^ sol;

  AddMcUnits(idDef1, todef1 diff GetMcUnits(idDef1));
  RemoveMcUnits(idDef1, GetMcUnits(idDef1) diff todef1);

  enable;
end;
```

AddMcUnits(mcid:integer, units:plist)

zmienia listę jednostek z makrem mcid zwraca true jeżeli udane

//Przykład - Misja RU 15a

```
// Zasobovani obrany a utoku pocitacu novymi daty
on McRegistryResultChanges(mcid,n,s) do
var side;
begin
  for side=1 to id_reg do
  if id_reg[side]=mcid then
  begin
    AddMcUnits(mcid,prepare_defend(side));
    SendForces(side);
    break;
  end;
end;
```

RemoveMcUnits(mcid:integer, units:plist)

zmienia listę jednostek z makrem mcid zwraca true jeżeli udane

//Przykład - Misja RU 05 cont

```
every 0$2 do
begin
  todef1:=fort ^ attack ^ defence ^ mor ^ sol;

  AddMcUnits(idDef1, todef1 diff GetMcUnits(idDef1));
  RemoveMcUnits(idDef1, GetMcUnits(idDef1) diff todef1);

  enable;
end;
```

ClearMcUnits(mcid:integer)

czyści listę jednostek z makrem mcid zwraca true jeżeli udane

//Przykład - Misja RU 12

```
every 0$10 do
begin
  {...}
  ClearMcUnits(idDef1);
  AddMcUnits(idDef1, todef1);
  ClearMcUnits(idDef2);
  AddMcUnits(idDef2, todef2);
  ClearMcUnits(idDef3);
  AddMcUnits(idDef3, todef3);
  {...}
  enable;
end;
```


GetMcUnitsSpec(mcid:integer, spec:integer)

zwraca listę jednostek spec z makro mcid

{brak przykładów}

AddMcUnitsSpec(mcid:integer units:plist spec:integer)

zmienia listę jednostek spec z makro mcid zwraca true jeżeli udane

//Przykład - Misja RU 12a cont

```
every 0$10+0$0.3 marked 1 do // Creating Ru_Straz
var i,bazukr;
begin
  if straz<bazooker then
  begin
    bazooker:=bazooker+1;
    while straz<bazooker do
    begin
      uc_side:=rus;
      uc_nation:=nation_russian;
      PrepareHuman(sex_male,class_bazooker,7+difficulty);
      hc_name="";
      bazukr:=CreateHuman;
      AddMcUnitsSpec(rus_mcrep_id,bazukr,1);
      straz:=straz^bazukr;
      PlaceUnitArea(bazukr,EnterArea,false);
      ComAgressiveMove(bazukr,66,48);
    end;
  end;
enable;
end;
```

RemoveMcUnitsSpec(mcid:integer, units:plist, spec:integer)

zmienia listę jednostek spec z makro mcid zwraca true jeżeli udane

//Przykład - Misja RU 12a cont

```
on UnitDestroyed(human) do
var un;
begin
  if human=Burlak then YouLost('Burlak');
  if human=Gossudarov then YouLost('Gossudarov');
  if un=Masha then lost_masha:=true;
  if human in straz then
  begin
    straz:=straz diff human;
    RemoveMcUnitsSpec(rus_mcrep_id,human,1);
    killed_russians:=killed_russians+1;
  end;
end;
```

ClearMcUnitsSpec(mcid:integer, spec:integer)

czyści listę spec jednostek z makro mcid zwraca true jeżeli udane

{brak przykładu}

McBase(prior:integer, units:plist, build:plist, research:plist, construct:plist, options:plist)

uruchamia makro i zwraca mcid

build musi być BuildList, researchs - lista tech, construct - niezaimplementowane

//Przykład - Misja AM 15a

```
every 0$5 trigger us in cpu_list do
var side,list,
    engineers;
begin
list=[];
for side=1 to 8 do
if (side in cpu_list) and not ExistMc(mc_list[side]) then
begin
engineers:=request_people(side,class_engineer,5);
list:=list ^ McBase(3,engineers,b_list[side],0,0,0);
end
else
list:=list ^ 0;
mc_list:=list;
enable;
end;
```

McMove(prior:integer, units:plist, waypoints:plist, options:plist)

uruchamia makro i zwraca mcid

units - lista jednostek, co jeszcze...

waypoints - jak sama nazwa mówi...

następujące opcje:

mc_move_wait - czekać wzajemnie na innych poza 4 hex'ami odległości, można podać dystans w hex'ach jako parametr

mc_move_through - naprawdę przechodzić PRZEZ waypoints, domyślnie 6 hex'ów najbliższych nich

mc_move_waitwp - czeka na waypoints dla pozostałych jednostek

mc_move_dontshoot - nie strzelając do wrogów

mc_move_dontcapture - nie przechwytuj pojazdów

mc_move_agressive - bądź !!##\$\$% AGRESYWNY !!!

mc_move_ownstuck - bez timeout przy przeszkodach strony przy nerwowości 100, po prostu rób to ZAWSZE...

//Przykład - Misja RU 09

```
// utok cislo 3 na rusaky
EVERY 0$19 TRIGGER Am_utoky=3 DO
VAR i;
VAR cesta;
begin
wait(cas_utoku_amiku[difficulty]);
for i in AM_UTOK_3_LIST do PlaceUnitArea(i,Am_enter,false);
case cislo_base of
1:cesta=[[40,46],[70,62],[96,104],[70,100],[49,78]];
2:cesta=[[40,47],[61,52],[59,76],[59,85],[71,99],[97,103]];
3:cesta=[[40,47],[61,52],[87,42],[84,20],[100,30],[128,41],[139,45],[148,65]];
end;
McMove(1,AM_UTOK_3_LIST,cesta,[mc_move_dontcapture,mc_move_agressive,mc_move_waitwp]);
Am_utoky=4;
end;
```

//Przykład - Misja RU 02

```
// amici zautoci na belkova a sejmou ho
EXPORT function amici_attack_belkov;
VAR i;
begin
if GetSide(bel)<>You then
begin
for i in AMICI_UTOK_BELKOV_LIST do PlaceUnitArea(i,Zpet_bel,false);
McMove(1,AMICI_UTOK_BELKOV_LIST,[[44,6],[36,5],[36,10],[GetX(bel),GetY(bel)],[52,2]],[mc_move_agressive]);
end;
end;
```

McRegistry(side:integer, options:plist)

Uruchamia makro do automatycznej rejestracji wrogów strony w określonych odstępach czasu i zwraca mcid. Każda rejestracja daje ilość i siłę wszystkich zarejestrowanych jednostek w wyniku. Jeżeli rejestracja zwraca inny wynik, niż poprzednio (lub niezerowy w pierwszej rejestracji), wywołuje zdarzenie:

McRegistrationResultChanges(mcid,count,strength).

Wartość siły nie ma realnego znaczenia, używa go tylko dla komparacji z innym wynikiem z makro lub do komparacji z zerem.

Możliwe opcje (należy zauważyć, że defarea może być podawana w dwóch formatach, jako 'areanumber' lub jako '[x,y,r]'):

[mc_reg_refresh_time, ticks] ... odstęp czasu między dwoma rejestracjami (domyślnie 0\$1)
[mc_reg_area_to_guard, defarea] ... rejestry wrogów w określonej strefie (domyślnie none)
[mc_reg_area_to_protect, defarea] ... rejestry wrogów atakujących jednostki w określonej strefie (domyślnie none)
[mc_reg_units_to_protect, listofunits] ... rejestry wrogów atakujących określone jednostki (domyślnie none)
[mc_reg_units_to_guard, listofunits] ... rejestry określonych wrogów (domyślnie none)
[mc_reg_expire_stops_to_attack, ticks] ... jak długo po zaprzestaniu ataku przez wroga rejestracja wygasa (domyślnie 0\$3)
[mc_reg_expire_leaves_area, ticks] ... jak długo po opuszczeniu przez wroga strefy rejestracja wygasa (default 0\$2)
(-)mc_reg_ignore_fog ... ignoruje mgłę więc widać wszystkich wrogów w strefie (domyślnie false)
(-)mc_reg_only_important ... nie ustawia zdarzenia jeżeli wynik zmieni się z niezerowej wartości na inną niezerową wartość (domyślnie false)

//Przykład - Misja AM 15

// McRegistry

```
Alliance_reg_id= McRegistry(alliance, [[mc_reg_area_to_guard, Al_Base],  
                                     [mc_reg_refresh_time, 0$2],  
                                     [mc_reg_area_to_protect, Al_Base],  
                                     [mc_reg_units_to_protect, units_to_protect],  
                                     -mc_reg_only_important,  
                                     -mc_reg_ignore_fog]);  
  
Alliance_wood_reg_id= McRegistry(alliance, [[mc_reg_area_to_guard, Ali_wood],  
                                           [mc_reg_refresh_time, 0$5],  
                                           -mc_reg_only_important,  
                                           -mc_reg_ignore_fog]);
```

{...}

//Przykład - Misja RU 12

```
var rDef1, rDef2, rDef3, rDef4, rDef5; {...}  
every 0$1 do  
begin {...}  
  rDef1:=McRegistry(ai_s[1], [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, nearA],[mc_reg_area_to_protect,nearA]]);  
  rDef2:=McRegistry(ai_s[2], [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, nearB],[mc_reg_area_to_protect,nearB]]);  
  rDef3:=McRegistry(ai_s[3], [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, nearC],[mc_reg_area_to_protect,nearC]]);  
  rDef4:=McRegistry(ai_s[4], [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, nearD],[mc_reg_area_to_protect,nearD]]);  
  rDef5:=McRegistry(ai_s[5], [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, nearE],[mc_reg_area_to_protect,nearE]]);  
{...} end;
```

McRegistryChangeOptions(mcid:integer, options:plist)

zmienia opcje McRegistry z ident mcid

//Przykład - Misja AM 15

```
{...}  
else if Area_changed then  
  begin  
    McRegistryChangeOptions(Russian1_reg_id, [[mc_reg_area_to_guard, attack_area]]);  
  end;  
{...}
```

McDefend(prior:integer, regid:integer, units:plist, options:plist)

uruchamia McDefend makro i zwraca mcid

Tutaj musi być uruchomione McRegistry makro do startu McDefend - przeciwnicy są rejestrowani przez to McRegistry. McDefend rozpoczyna się po każdej rejestracji tego McRegistry. Parametry:

[mc_def_advantage, percent] ... jak wielka przewaga nad wrogiem jest wystarczająca (domyślnie 200, podwójna)
[mc_area_dont_leave, defarea] ... strefa, w której wszystko musi być obronione (domyślnie none)
(-)mc_change_to_soldiers ... ustawia tutaj, jeżeli McDefend może przemienić ludzi w żołnierzy
(-)mc_change_to_mechanics ... ustawia tutaj, jeżeli McDefend może przemienić ludzi w mechaników

//Przykład - Misja RU 06

```
var rDef1, rDef2;
export idDef1, idDef2;
export toDef1, toDef2;
every 0$1 do
begin
  rDef1:=McRegistry(ai, [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, def_base],[mc_reg_area_to_protect,def_base]]);
  rDef2:=McRegistry(ai, [[mc_reg_refresh_time, 0$2],[mc_reg_area_to_guard, def_base3],[mc_reg_area_to_protect,def_base3]]);
  {...}
  idDef1:=McDefend(10, rDef1, toDef1,[mc_def_advantage, mcdef_difficulty]);//-mc_change_to_soldiers, -mc_change_to_mechanics);
  idDef2:=McDefend(10, rDef2, toDef2,[mc_def_advantage, na_kopci_mcdef_difficulty]);
  {...}
end;
```

McAttack(prior:integer, regid:integer, units:plist, options:plist)

uruchamia McAttack makro i zwraca mcid

Nie musi być uruchomione McRegistry makro do startu McAttack - przeciwnicy są rejestrowani przez to McRegistry.

McAttack startuje po każdej rejestracji tego McRegistry.

regid - id rejestru - pilnuje jednostek area_to_guard z mc_registry

units - lista jednostek posłusznych twoim rozkazom

opcje:

[-]mc_no_stop ... zawsze czołgaj przez strefę rejestrowaną
[-]mc_murder ... zabij te PRZEKŁĘTE świnie leżące na ziemi!
[mc_area_dont_leave, defarea] ... nie zostawaj w tej strefie (domyślnie none)
[mc_retreat_lives_people,lives] ... wycofanie dla retreat_area_people gdy kobiecie/mężczyźni punkty życia (hitpoints) spadają poniżej lives
[mc_retreat_lives_vehicles,lives] ... wycofanie dla retreat_area_vehicle gdy pojazdowi hitpoints zejdzie poniżej lives
[mc_retreat_area_people, defarea]
[mc_retreat_area_vehicle, defarea]
[mc_out_of_fuel, level] ... porzuć zdarzenie, jeżeli paliwo zejdzie poniżej poziomu (level)

Zachowanie McAttacku:

Skanuje określoną strefę do czasu, gdy zobaczy nieprzyjaciela.

Następnie przypisuje niezdolność celu Nevhodnost_cilu dla strony McAttacku, najmniejszy dla najsłabszej jednostki.

Nieodpowiedni jest znormalizowany przez wartość 100.

Kiedy jednostki zabiją wroga makro zakonczy się, chyba że ustawisz flagę no_stop.

W makro można ustawić strefy (mc_area_dont_leave), których nie można opuścić oraz w których jednostki muszą zawrócić.

Makro z ustawioną opcją mc_out_of_fuel wyłącza się u jednostek, którym poziom paliwa zmniejszył poniżej ustalonej granicy.

Tworzy listę jednostek (celi), na której umieszczeni są aktualnie atakowani, a następnie ustala cel na -1.

Zapewni to nieatakowanie umierających jednostek (którzy dostaną bardzo korzystny Nevhodnost_cile).

Kiedy McAttack zakończy się, ustawi na wszystkie jednostki które były atakowane niezdolność -1.

Domyślnie McAttack nie uśmierca umierających. Da się to zmienić używając opcji mc_murde, wtedy gdy postać jest umierająca, makro zmienia jej niezdolność na -1 i będzie otrzymywała niskie uszkodzenia.

Po zakończeniu McAttacku te jednostki pozostają przy określonej niezdolności, dzięki temu będą mogły zostać zabite przez każdą przechodnią jednostkę naszej z strony.

Nie wiemy, jak rozwiązać ten problem (tj. aby i umierające jednostki w tym trybie miały po zakończeniu McAttack niezdolność -1).

McAttack skończy, gdy nie ma już żadnej jednostki, które by mogły atakować lub jeżeli nie widzi nieumierających jednostek przeciwnika.

Po zakończeniu McAttacku jednostki nie mają zadań, więc wykańczają wszystkich, których widzą.

```
//Przykład - Misja RU 12
```

```
export BehMcRegId,BehMcAttId;
export function BehemothUtok;
begin
  BehMcRegId:=McRegistry(ai_s[1], [[mc_reg_refresh_time, 0$4],
    [mc_reg_units_to_protect, [Platonov]],
    [mc_reg_units_to_guard, [Burlak]],
    [mc_reg_area_to_protect,near_base_area[plat_i]],
    mc_reg_ignore_fog,
    -mc_reg_only_important]);

  BehMcAttId:=McAttack(12, BehMcRegId, [behemoth],
    [[mc_retreat_area_vehicles, nearA],
    [mc_retreat_area_people, nearA],
    [mc_retreat_lives_vehicles, 600],
    [mc_retreat_lives_people, 800],
    mc_no_stop]);
end;
```

```
-----
//Przykład - Misja RU 15
```

```
//legion krade siberitofku hm na nejaký kradeni kaslu, bude utok :)
var attid; //DODELAT mohl bych udelal ze kdyz vyhrozoval tak zautoci hned, jinak az po chvíli
every 0$2 trigger rusacimajisib and not legdestr do
var tmp,tmp1,tmp2,regid_le;
begin //|lethiefsar|==322
{...}

tmp=filterunitsinarea(lethiefsar,[[f_or,[f_weapon,ar_rocket_launcher],[f_weapon,ar_gun],[f_weapon,ar_flame_thrower]]]);
tmp1=filterunitsinarea(lethiefsar,[[f_weapon,ar_selfpropelled_bomb]]);
tmp2=filterunitsinarea(lethiefsar,[[f_weapon,ar_radar]]);

regid_le=mcregistry(le,[[mc_reg_area_to_guard,target],mc_reg_ignore_fog]);
attid=mcattack(100,regid_le,tmp^tmp1^tmp2,[mc_no_stop{,[mc_area_dont_leave,target]}]);

end;
```

McPatrol(prior:integer, regid:integer, units:plist, options:plist)

uruchamia McPatrol makro i zwraca mcid

Tu nie musi być uruchomione McRegistry makro do startu McPatrol - przeciwnicy są rejestrowani przez to McRegistry.

McPatrol startuje po każdej rejestracji tego McRegistry.

jednostki przechodzą przez strefę i próbują pozostać niezauważeni przez dłuższy czas, więc robią to ZAWSZE

(np. McPatrol kończy się jeżeli wszystkie jednostki w niej są marter

porzuć zdarzenie gdy tylko jednostki zobaczą wroga

prawie wszystkie opcje tak jak w McAttack, z wyjątkiem mc_murder i mc_no_stop, których nie ma. Ponadto jest:

[-]mc_pat_aggressive ... bądź agresywny podczas patrolowania

```
//Przykład - Misja AM 15
```

```
if not(ExistMc(Russian1_patr_id)) and (Hlp_New_Patrol_Group <> 0) then
begin
  R1Patrol_group= Hlp_New_Patrol_Group;
  Russian1_patr_id= McPatrol(100, Russian1_reg1_id, Hlp_New_Patrol_Group, [mc_no_stop]);
end
```

McSkirmish(prior:integer, units:plist, options:plist)

uruchamia McSkirmish makro i zwraca mcid

opcje - [[mcs_PosList, list], [mcs_Strategy, list], [mcs_Parking, parkingarea]]

{brak przykładu}

McSSetPosList(mcid:integer, list:plist)

ustala listę pozycji budynku dla McSkirmish mcid

{brak przykładu}

McSAddVehicles(mcid:integer, weapons:plist, chassis:plist, engines:plist, controls:plist, parkingarea:integer)

tworzy nową definicję pojazdu dla McSkirmish mcid, zwraca numer grupy

{brak przykładu}

McSChangeVehicles(mcid:integer, number:integer, weapons:plist, chassis:plist, engines:plist, controls:plist, parkingarea:integer)

uaktualnia definicję pojazdu dla McSkirmish mcid, numer grupy

{brak przykładu}

McSAddFortification(mcid:integer, kinds:plist, weapons:plist)

tworzy nową definicję fortyfikacji dla McSkirmish mcid, zwraca numer definicji

{brak przykładu}

McSChangeFortification(mcid:integer, number:integer, kinds:plist, weapons:plist)

uaktualnia nową definicję fortyfikacji dla McSkirmish mcid, numer definicji

{brak przykładu}

McSSetStrategy(mcid:integer, list:plist)

ustala główną listę strategii dla McSkirmish mcid

{brak przykładu}

McSGetVehList(mcid:integer, number:integer)

zwraca listę pojazdów w grupie number

{brak przykładu}

McSParking(mcid:integer, veh:plist, park:boolean)

zmienia listę zaparkowanych pojazdów veh zostaną dodane (park=true) or usunięte (park=false)

{brak przykładu}

McSConditionalStrategyFilter(str:plist)

filtry warunkowe strategii do strategii bez warunku (główna lista strategii)

{brak przykładu}

McRepair(prior:integer, units:plist, others:plist, options:plist)

uruchamia McRepair makro i zwraca mcid

jednostki (units) będą się trzymać innych (others) (i dodatkowo z McWithdraw)

opcje - [[mcr_hum_area, area], [mcr_veh_area, area], (-)mcr_area_only, [mcr_safety, 7],

[mcr_hum_limit, 600], [mcr_veh_limit, 1000], [mcr_bui_limit, 1000],

[mcr_repv_area, area], [mcr_capv_area, area], [mcr_repb_area, area], [mcr_capb_area, area]]

//Przykład - Misja RU 12a cont

```
every 0$3+0$0.5 marked 401 do // Russian McRepair
```

```
var un,CanReturnToDepot,filter;
```

```
begin
```

```
{...}
```

```
if NOT ExistMc(rus_mcrep_id) then
```

```
    rus_mcrep_id:=McRepair (100,rus_mcrep_units,
```

```
        rus_mcrep_bui^rus_mcrep_units,
```

```
        [[mcr_hum_area, BluekherArea],[mcr_bui_limit, 1000],
```

```
        mcr_area_only,[mcr_safety,7],[mcr_repb_area, BluekherArea]]);
```

```
{...}
```

```
end;
```

//Przykład - Misja RU 12a cont

```
every 0$10+0$0.5 marked 401 do // Russian McRepair
```

```
var un,CanReturnToDepot,filter;
```

```
begin
```

```
{...}
```

```
if NOT ExistMc(rus_mcrep_id) then
```

```
    rus_mcrep_id:=McRepair (100,rus_mcrep_units,
```

```
        rus_mcrep_bui^rus_mcrep_units,
```

```
        [[mcr_hum_area, BluekherRepairArea],
```

```
        [mcr_repb_area, BluekherArea]]);
```

```
{...}
```

```
end;
```

McWithdraw(prior:integer, repid:integer, units:plist, options:plist)

uruchamia McWithdraw makro i zwraca mcid najpierw powinien być uruchomiony McRepair

opcje - [[mcw_hum_lo, 400], [mcw_hum_hi, 800], [mcw_veh_lo, 400], [mcw_veh_hi, 1000],

[mcw_hum_area, defarea], [mcw_veh_area, defarea], [mcw_corpse_area, defarea]]

strefy zostaną skopiowane z McRepair z repid przed sprawdzeniem tej opcji

{brak przykładu}

McCustom(prior:integer, units:plist)

uruchamia puste makro i zwraca mcid

//Przykład - Misja AM 15a

```
every 0$2 do
```

```
var list,un;
```

```
begin
```

```
for un in wounded_un do
```

```
if not (un in healing_un) then
```

```
begin
```

```
    healing_un:=healing_un ^ un;
```

```
    McCustom(100,un);
```

```
    ComMoveToArea(un,Heal[GetSide(un)]);
```

```
    wounded_un:=wounded_un diff un;
```

```
end;
```

```
enable;
```

```
end;
```

Bazy/Flagi:

GetBuildList(base_or_list:tvalue)

tworzy BuildList z całej bazy lub listy jednostek (budynków)

//Przykład - Misja AM 14

```
export function init_keep_base;
var un,list;
begin
  b_list:=[];

  un:=FilterAllUnits([ [f_side,russians], [f_type,unit_building] ]);

  if un then
    b_list:=GetBuildList(GetBase(un[1]));

  Enable(2);
end;
```

SetBattleFlag(side:integer, x:integer, y:integer, options:plist)

ustala flagę wojny dla określonej strony na określone koordynaty i zwraca flagid

efekt flagi jest zależny od opcji, opcje domyślne:

[[bfo_defend_bonus_human,80],[bfo_defend_bonus_vehicle,40],[bfo_defend_bonus_building,0],[bfo_range,17],[bfo_height,100]]
distance: 0..range: full effect range+1..range*2: decreasing effect

//Przykład - Alien base

```
// prepares units of specified side
function prepare_side(side);
var pos,x,y,dir,i,r,sol,br,dep,neng,nsci,nmech,nsol;
begin
  Side_IsSpec := Replace(Side_IsSpec,Side,Multiplayer_GetPlayerIsSpec(Side));
  {...}
  x:=depot_def[pos][1];
  y:=depot_def[pos][2];
  dir:=depot_def[pos][3];
  {...}

  // eventually place flag
  if def_morale_flags then
    begin
      PlaceEnvironment(ShiftX(x,dir,10),ShiftY(y,dir,10),12,2);
      battle_flags_id:=battle_flags_id ^ SetBattleFlag(side,ShiftX(x,dir,10),ShiftY(y,dir,10),
        [ [bfo_defend_bonus_human,100],
          [bfo_defend_bonus_vehicle,50],
          [bfo_defend_bonus_building,20],
          [bfo_height,100] ]);
    end;
  {...}
end;
```

KillBattleFlag(flagid:integer)

usuwa flagę wojny flagid (zwraca true jeżeli udane)

//Przykład - Babylon

```
on UnitDestroyed(j) do
begin
  if not (getBType(j) in [b_depot,b_warehouse]) then exit;
  KillBattleFlag(FindBattleFlag(GetX(j),GetY(j)));
end;
```


FindBattleFlag(x:integer, y:integer)

zwraca flagid flagi wojny na [x,y] lub -1

//Przykład - Babylon

```
on UnitDestroyed(j) do
begin
if not (getBType(j) in [b_depot,b_warehouse]) then exit;
KillBattleFlag(FindBattleFlag(GetX(j),GetY(j)));
end;
```

GetBattleFlagSide(flagid:integer)

zwraca stronę istniejącej flagi wojny

{brak przykładu}

ChangeBattleFlagSide(flagid:integer, side:integer)

zmienia stronę istniejącej flagi wojny

{brak przykładu}

ChangeBattleFlagOptions(flagid:integer, options:plist)

zmienia opcje istniejącej flagi wojny

//Przykład - Alien base

```
function Update_Flags;
var un;
begin
for un in battle_flags_id do
ChangeBattleFlagOptions(un,[ [bfo_defend_bonus_human,strength],
[bfo_defend_bonus_vehicle,strength div 2],
[bfo_defend_bonus_building,strength div 5],
[bfo_height,strength] ]);
end;
```

Strefy:

SetAreaMapShow(area:integer, id:integer)

pokazuje strefę na mapie i/lub minimapie

//Przykład - Misja AM 13

//dialog 6b

```
on vehicleconstructed(veh,fact) marked 13 do
if getweapon(veh)=us_siberium_rocket then begin
dialogueon;
say(john,'D6b-JMM-1');
sayradio(powell,'D6b-Pow-1');
dialogueoff;
changemissionobjectives('M3');
setareamapshow(target,1);
disable(13);
enable(04);
end;
```

MoveAreaXY(area:integer, x:integer, y:integer)

przenosi strefę, lewy górny róg jest na określonych koordynatach

//Przykład - Misja RU 15a

```
every 0$7.7 trigger ar_bad do
var un,list;
begin
list=[];
for un in ar_bad do
begin
MoveAreaXY(crates_check,un[1],un[2]);
if GetListOfCratesInArea(crates_check) then
list:=list ^ [un];
end;
ar_bad:=list;
enable;
end;
```

Zadania:

SendSiberiteRocket(x:integer, y:integer)

rakieta syberytyowa eksploduje na hex'ie x,y

//Przykład - Misja AM 14a cont

```
DialogueOn;
interface_hidden:=true;
CenterOnXY(103,52);
wait(0$3);
PrepareSiberiteRocket;
SendSiberiteRocket(103,52);           //!!!!!! VYBUCH !!!!!!
interface_hidden:=false;
DialogueOff;
```

CanBeResearched(labor:integer, tech:integer)

zwraca jeżeli tech jest badana w laboratorium (jeżeli tech jest już zbadana zwraca false)

//Przykład - Misja AM 15a

```
function research_possible(side,tech);           // zjistovani seznamu laboratorii, ve kterych se da tech vynalezat
var un,vysledek,
lab_list;           // seznam laboratorii
begin
lab_list:=get_labs(side);

if GetTech(tech,side) in [state_disabled, state_researched] then
vysledek:=[]
else
for un in lab_list do
if CanBeResearched(un,tech) then
vysledek:=vysledek ^ un;

result:=vysledek;
end;
```

PlaceWeaponTurret(un:integer, bc_weapon:integer)

umieszcza bc_weapon na wieżyczce lub bunkrze un

//Przykład - Misja AM 06

```
function base;
var i, tmp, btype, Delta;
begin
{...}
tmp:=CreateAndPlaceBuildingXYD(i[1], i[2], i[3]);
if bc_type in [b_bunker,b_turret] then
  if i=6 then
    PlaceWeaponTurret(tmp, i[6])
  else
    PlaceWeaponTurret(tmp, us_machine_gun);
end;
{...}
end;
```

Przykład - Misja RU 10

```
if bc_type in [b_bunker, b_turret] then
begin
  weap:=[ru_gun,ru_rocket_launcher,ru_heavy_gun];
  if (building>6) then
    begin
      if (building[7] in weap) then
        PlaceWeaponTurret(un,building[7])
      else
        PlaceWeaponTurret(un,weap[Rand(1,weap)]);
    end
  else
    PlaceWeaponTurret(un,weap[Rand(1,weap)]);
end;
```

CopyTasks(un_dest:integer, un_source:integer)

kopiuje zadania z jednostki un_source do unit_dest (zastępując, niepatroluj)

{brak przykładu}

AddCopyTasks(un_dest:integer, un_source:integer)

kopiuje zadania z jednostki un_source do unit_dest (dodaje, niepatroluj)

{brak przykładu}

SetTaskList(un:integer, list:plist)

ustala zadania z tasklist (GetTaskList format) - używaj ostrożnie!

{brak przykładu}

AddTaskList(un:integer, list:plist)

dodaje zadania z tasklist (GetTaskList format) - używaj ostrożnie!

{brak przykładu}

CanBeConstructed(factory:integer, chassis:integer, engine:integer, control:integer, weapon:integer)

zwraca jeżeli pojazd może być wyprodukowany w fabryce

//Przykład - Misja RU 05 cont

```
every 0$15 trigger IsOk(fact1) and IsIdle(fact1) and ((defence+attack)<(5+difficulty*2)) do
var a, ec, cm, ac, podvozky,ovladani, def_pod, att_pod, tmec;
begin
  ec:=engine_combustion;
  cm:=control_manual;
  podvozky:=AvailableChassisList(fact1);
  ovladani:=AvailableControlList(fact1);
  if control_remote in ovladani then
    ac:=control_remote
  else
    ac:=cm;
  att_pod:=ar_half_tracked;

  case true of
    ar_half_tracked in podvozky : def_pod:=ar_half_tracked;
    ar_medium_trike in podvozky : def_pod:=ar_medium_trike;
  end;

  tmec:=UnitFilter(mec, [[f_outside]]);
  while tmec do begin
    ComExitVehicle(tmec);
    AddComEnterUnit(tmec, fact1);
    DU_wait(tmec);
    tmec:=UnitFilter(mec, [[f_outside]]);
  end;

  case true of
//cargo
    cargo<1 and CanBeConstructed(fact1, ar_medium_trike, ec, cm, ar_cargo_bay): begin
      ComConstruct(fact1, def_pod, ec, cm, ar_cargo_bay);
      to_cargo:=true;
    end;
//defence
    defence>=1 and defence<3 and
    CanBeConstructed(fact1, def_pod, ec, cm, ar_gun): begin
      ComConstruct(fact1, def_pod, ec, cm, ar_gun);
      to_defence:=true;
    end;
  {...}
//remote
    remote<1 and
    CanBeConstructed(fact1, ar_medium_trike, ec, cm, ar_control_tower): begin
      ComConstruct(fact1, ar_medium_trike, ec, cm, ar_control_tower);
      to_remote:=true;
    end;
//attack
    tick<20$0 and attack<3 and
    CanBeConstructed(fact1, ar_hovercraft, ec, ac, ar_light_gun): begin
      ComConstruct(fact1, ar_hovercraft, ec, ac, ar_light_gun);
      to_attack:=true;
    end;{...}
  end;
enable://-----
end;
```

RemoveTasks(units:plist)

usuwa wszystkie zadania jednostek

//Przykład - Misja RU 12

```
on Command(comm) do
begin
if not Getside(Eisenstein)=you and IsInUnit(Eisenstein) then
    RemoveTasks(Eisenstein);
if not Getside(Davidov)=you and IsInUnit(Davidov) then
    RemoveTasks(Davidov);
if not Getside(Tsaritsyn)=you and IsInUnit(Tsaritsyn) then
    RemoveTasks(Tsaritsyn);
if not Getside(Borodin)=you and IsInUnit(Borodin) then
    RemoveTasks(Borodin);
end;
```

Koszty/Ilości:

CostOfVehicle(chassis:integer, engine:integer, control:integer, weapon:integer)

zwraca koszt pojazdu na liście [cans,oil,sib]

//Przykład - Misja AM 12

```
var fourthplus,chs,wps;
every 1$0+15$0 do
var ch,wp,tmp,tmp1,tmp2;
begin
fourthplus=fourthplus+1;
ch=0;
case fourthplus of //4 diff=3 is rta=[10,18,5,4,7]
1..rta[1]:
if not (fourthplus mod rta[3]) then begin
telep=[];
telepattno=telepattno+1;
ch=chs[1];
wp=wps[1];
end;
(rta[1]+1)..rta[2]:
if not ((fourthplus-rta[1]) mod rta[4]) then begin
telep=[];
telepattno=telepattno+1;
ch=chs[2];
wp=wps[2];
end;
else
if not ((fourthplus-rta[2]) mod rta[5]) then begin
telep=[];
telepattno=telepattno+1;
ch=chs[3];
wp=wps[3];
end;
end;
enable;
if not ch then exit;
tmp=costofvehicle(ch[1],engine_siberite,control_computer,wp[1]);//[cans,oil,sib]
tmp1=costofvehicle(ch[2],engine_siberite,control_computer,wp[2]);
tmp2=costofvehicle(ch[3],engine_siberite,control_computer,wp[3]);
addresourcetype(getbase(ru_fact),mat_cans,tmp[1]+tmp1[1]+tmp2[1]);
addresourcetype(getbase(ru_fact),mat_oil,tmp[2]+tmp1[2]+tmp2[2]);
addresourcetype(getbase(ru_fact),mat_siberit,tmp[3]+tmp1[3]+tmp2[3]);

comconstruct(ru_fact,ch[1],engine_siberite,control_computer,wp[1]);
addcomconstruct(ru_fact,ch[2],engine_siberite,control_computer,wp[2]);
addcomconstruct(ru_fact,ch[3],engine_siberite,control_computer,wp[3]);

end;
```

CostOfBuilding(kind:integer, nation:integer)

zwraca koszt budynku na liście [cans,oil,sib]

{brak przykładu}

CostOfWeapon(weapon:integer)

zwraca koszt broni na liście [cans,oil,sib]

{brak przykładu}

SeeGreyArea(side:integer, area:integer)

zwraca ile hex'ów z określonej strefy widzi strona + szare hex'y

//Przykład - Misja RU 02

```
EVERY 0$1 DO
VAR pom;
begin
pom=SeeGreyArea(You,Narea);
if pom>=procento_odkrytych_pro_vyhru[difficulty]
then begin
    ChangeMissionObjectives('M10');
    ChangeMissionObjectives('M10a');
end
else enable;
end;
```

Miny:

MineExplosion(x:integer, y:integer, wound:integer)

wybuch miny na x, y

//Przykład - Misja AM 15

```
Function LaunchMines;
var mines_pos, mines_pos1, i, j;
begin
// Mines positions on the stones
mines_pos= [[168,18], [169,20], [167,20], [164,18], [164,19], [163,19],
[164,21], [165,21], [163,20], [164,22], [162,20]];
// Mines positions on the factory
mines_pos1= [[169,34], [169,35], [167,32], [170,31], [173,35], [174,39]];
// Launch the mines
for i= 1 to mines_pos do
begin
// Launch the environment
MineExplosion(mines_pos[i][1], mines_pos[i][2], 1);
Wait(Rand(1, 3));
if i < mines_pos then
RemoveEnvironmentWithoutRebuild(mines_pos[i][1], mines_pos[i][2])
else
begin
// Launch the factory
for j= 1 to mines_pos1 do
begin
MineExplosion(mines_pos1[j][1], mines_pos1[j][2], 1);
Wait(Rand(1,3));
end;
RemoveEnvironment(mines_pos[i][1], mines_pos[i][2]);
end;
end;
end;
```

PlaceMine(x:integer, y:integer, side:integer, typ:integer)

postaw minę na pozycji x, y. typ=0 - zdalna mina (wynik to numer miny), typ>0 (mina czasowa - czas w chwilach (1..100))

//Przykład - Misja RU 10 cont

```
// vytvoreni arabskeho minoveho pole
function create_minefield;
var i,xx;
begin
  mines_list:=[];
  for i=1 to (difficulty*2) do
    begin
      xx:=RandHexArea(minefield_north,false);
      PlaceMine(xx[1],xx[2],enemy,0);
      mines_list:=mines_list ^ xx;
      wait(1);
    end;
  for i=1 to difficulty do
    begin
      xx:=RandHexArea(minefield_east,false);
      PlaceMine(xx[1],xx[2],enemy,0);
      mines_list:=mines_list ^ xx;
      wait(1);
    end;
  {...}end;
```

LaunchMine(number:integer)

eksplozna zdalnej miny, podłożonej przez PlaceMine. NIE UŻYWAJ numeru, to się zmienia

{brak przykładu}

LaunchMineAtPos(x:integer, y:integer, side:integer)

aktywuj minę na pozycji x, y, położonej przez PlaceMine

// Przykład - Misja RU 10 cont

```
// AI minoveho pole ;-))
export mine_xy;
every 0$1 trigger FilterUnitsInArea(minefield, [ [f_side,you] ]) and mines_list Marked 1 do
var un,list,mine,
  dist, // vzdalenost na kterou se ma reagovat
  pom_list; // nahradni seznam min, ktore zatim nevybuchly
begin
  list:=FilterUnitsInArea(minefield, [ [f_side,you] ]);
  pom_list:=[];
  for un in list do
    begin
      if GetType(un)=unit_human then
        dist:=3
      else
        dist:=4;
      for mine=1 to (mines_list div 2) do
        if GetDistXY(mines_list[(mine*2)-1],mines_list[(mine*2)],GetX(un),GetY(un))<dist then
          begin
            LaunchMineAtPos(mines_list[(mine*2)-1],mines_list[(mine*2)],enemy);
            if not first_mine_exploded then
              begin
                first_mine_exploded:=true;
                mine_xy:=[mines_list[(mine*2)-1],mines_list[(mine*2)]];
              end;
            end
          else
            pom_list:=pom_list ^ mines_list[(mine*2)-1] ^ mines_list[(mine*2)]; // razeni nevybuchlych bomb
          mines_list:=pom_list;
          pom_list:=[];
        end;
      enable;
    end;
```

ViewMineAtPos(x:integer, y:integer, side:integer)

pokazuje minę na pozycji x, y dla side_fog[side]

//Przykład - Misja RU 10 cont

```
function show_mines;
var i;
begin
  for i=1 to (mines_list div 2) do
    ViewMineAtPos(mines_list[(i*2)-1],mines_list[(i*2)],you);

  Heike_group:=[] ^ Heike;
  PlaceUnitArea(Heike,heike_placed,false);
  uc_side:=neutral;
  uc_nation:=nation_arabian;
  uc_direction:=Rand(0,5);
  for i=1 to (4-difficulty) do
    begin
      PrepareSoldier(0,5);
      Heike_group:=Heike_group ^ CreateHuman;
      PlaceUnitArea(Heike_group[i+1],heike_placed,false);
    end;
end;
```

MineAtPos(x:integer, y:integer)

zwraca true, jeżeli mina jest na pozycji x, y inaczej false

//Przykład - Misja RU 09

```
// odpaleni min
EVERY 0$1 TRIGGER Arabi_vztah=0 DO
VAR pom;
begin
  pom=FilterUnitsInArea(Armina1,[[f_side,You]]);
  if pom>0 and MineAtPos(78,47) then ComFireExplosives(diet);
  pom=FilterUnitsInArea(Armina2,[[f_side,You]]);
  if pom>0 and MineAtPos(78,43) then ComFireExplosives(A2);
  pom=FilterUnitsInArea(Armina3,[[f_side,You]]);
  if pom>0 and MineAtPos(82,46) then ComFireExplosives(A4);
  enable;
end;
```

MineOfUnit(un:integer)

zwraca [x,y], jeżeli jednostka posiada zdalny ładunek inaczej zwraca false

//Przykład - Misja RU 09 cont

```
// vojaci
ComExitBuilding(Arabi_enter);
AddComEnterUnit(Arabi_enter,Ar_bar);
AddComChangeProfession(Arabi_enter,class_mortar);
RemoveMineOfUnit(diet);
RemoveMineOfUnit(A2);
RemoveMineOfUnit(A4);
AddComPlaceRemoteCharge(diet,0,0,Am_at6);
AddComPlaceRemoteCharge(A2,0,0,Am_at7);
AddComPlaceRemoteCharge(A3,0,0,Am_bar);
AddComPlaceRemoteCharge(A4,0,0,Am_at4);
AddComMoveXY(Arabi_enter,88,9);
```


RemoveMineOfUnit(un:integer)

usuwa umieszczony zdalny ładunek jednostki, jeżeli istnieje

//Przykład - Misja RU 09 cont

```
// wojaci
ComExitBuilding(Arabi_enter);
AddComEnterUnit(Arabi_enter,Ar_bar);
AddComChangeProfession(Arabi_enter,class_mortar);
RemoveMineOfUnit(diet);
RemoveMineOfUnit(A2);
RemoveMineOfUnit(A4);
AddComPlaceRemoteCharge(diet,0,0,Am_at6);
AddComPlaceRemoteCharge(A2,0,0,Am_at7);
AddComPlaceRemoteCharge(A3,0,0,Am_bar);
AddComPlaceRemoteCharge(A4,0,0,Am_at4);
AddComMoveXY(Arabi_enter,88,9);
```

Właściwości:

BuildingStatus(un:integer)

zwraca stan budynku (bs_???)

//Przykład - Misja AM 15

```
// The first factory
if (R_Attack_Group > 0) then
  if not(BuildingStatus(GetUnitNumber(195, 84)) = bs_working) and
    CanBeConstructed(GetUnitNumber(195, 84),
      R_Attack_Group[1][1], R_Attack_Group[1][2],
      R_Attack_Group[1][3], R_Attack_Group[1][4]) then
    begin
      ComConstruct(GetUnitNumber(195, 84),
        R_Attack_Group[1][1], R_Attack_Group[1][2],
        R_Attack_Group[1][3], R_Attack_Group[1][4]);
      R_Attack_Group= Delete(R_Attack_Group, 1);
      Wait(1);
    end;
```

ContactTime(sides:plist)

czas od ostatniego kontaktu (parametr: side lub [side1,side2])

//Przykład - Misja AM 09 cont

```
on BuildingCaptured(build, origside, eng) do
begin
  if not byd2a and origside=you2 then
    dialog2a;
  if EndEnable and (build=rudepot1) and origside=you then begin
    TimerEnable:=true;
    TimerCas:=ContactTime(russians);
    SetSideBase(GetBase(rudepot1),russians);
  end;
end;
```

InBattle(sides:plist)

wykrywanie walki (parametr: side lub [side1,side2])

//Przykład - Misja AM 10

```
// jestli muze kecat, nebo se prave bojuje
function testfight;
begin
  while InBattle([You,Rusi]) do wait(0$0.5);
end;
```

GetProperties(un:integer)

zwraca listę właściwości jednostek [speed, defend, sight, range, eff1, eff2, eff3]

{brak przykładu}

ImagineProperties(vehb:integer, human:integer, cl:integer)

zwraca listę właściwości jednostek dla vehb z ludzi wewnątrz (cl 0 oznacza bieżącą klasę)

{brak przykładu}

CanCarryHowMuch(un:integer)

zwraca ile materiałów może posiadać jeszcze jednostka

//Przykład - Misja RU 14 cont

```
Function naloz (Veh, Bud, Mat, Amount);
Var Base, Res;
Begin
  Base = GetBase (Bud);
  Result = CanCarryHowMuch (Veh);
  If Result > Amount then Result = Amount;
  Res = GetResourceType (Base, Mat);
  If Result > Res then Result = Res;
  SetResourceType (Base, Mat, Res - Amount);
  AddCargo (Veh, Mat, Result);
End;
```

Artefakty:

SetArtifactRes(side:integer, state:boolean)

pozwalia na badanie technologii artefaktu

//Przykład - Misja AM 15a

```
// Initialize artefact from defined values
function Get_Init_Artefacts(art_icons, art_events, art_difficulty, art_cost, art_recharge);
var un,i,side,base;
begin
  {...}
  // allow research alien technology
  for side=1 to 8 do
    SetArtifactRes(side,state_enabled);
  {...}
end;
```

SetArtifactUse(side:integer, number:integer, state:integer, lab:integer)

umożliwia korzystanie z ikon artefakt w laboratoriach dla strony...

number - art_use_? state - art_? lab - unit number, wartość ujemna lub zero dla wszystkich laboratoriów

//Przykład - Misja RU 15a

```
// Highlightovani a shadowovani ikon pri pouziti artefaktu
function Shadow_Artefact(size);
begin
  SetArtifactUse(you,art_status[size][1],art_gray,nearest_labs[size]);
end;

function Highlight_Artefact(size);
var state;
begin
  {...}
  SetArtifactUse(you,art_status[size][1],state,nearest_labs[size]);
end;
```

FindArtifact(size:integer)

znajduje artefakt o rozmiarze size i zwraca jego pozycję [x,y] lub false

//Przykład - Misja RU 15

```
every 0$1 trigger rusacimajisib do
begin
  if not findartifact(5) then
    artef_gone=true
  else
    enable;
end;
```

Ataki:

PriorityAttack(side:integer, un:integer)

prioritytet ataku strony side na jednostkę un

//Przykład - Misja AM 03

```
function lets_destroy_americans;
var i,attackers1,attackers2;
begin
  PriorityAttack(russians,sklad);
  attackers1:=[];
  attackers2:=[];
  for i in ru_trucks do
    if (0+attackers1)>(0+attackers2) then
      attackers2:=attackers2 ^ i
    else
      attackers1:=attackers1 ^ i;

  ComRemember(attackers1);
  ComRemember(attackers2);
  Attack_US_Base(attackers1,attackers2);

  wait(0$4);
  ComReturn(attackers1);
  ComReturn(attackers2);
end;
```

DoNotAttack(side:integer, un:integer)

nieagresja dla strony side na jednostkę un

//Przykład - Misja 01 cont

```
every 0$0.3+0$0.15 trigger attacking_bunker do
var un;
begin
  for un in trappers diff [Burlak,pokryshkin] do
    begin
      if GetLives(un)<500 then
        DoNotAttack(americans,un);
      end;
    enable;
  end;
```

NormalAttack(side:integer, un:integer)

zniesienie specjalnego ataku strony side na jednostkę un

//Przykład - Misja AM 15

```
Function CaptureWithoutShow(Captured_american, Explorer1, Explorer2);
begin
  while IsInUnit(Captured_american) <> 0 do
    begin
      ComExitVehicle(IsInUnit(Captured_american));
      Wait(1);
    end;
    SetSide(Captured_american, americans1);
    ComStop(Captured_american);
    DoNotAttack(alliance, Captured_american);
    DoNotAttack(Explorers_side, Captured_american);
    if Explorer1 <> 0 then TeleportUnit(Explorer1, 245, 186, 3, true);
    if Explorer2 <> 0 then TeleportUnit(Explorer2, 245, 186, 3, true);
    TeleportUnit(Captured_american, 245, 186, 3, true);
    Wait(0$3);
    CaptureDialogue(Captured_american);
    NormalAttack(alliance, Captured_american);
    NormalAttack(Explorers_side, Captured_american);
    HealExplorers;
  end;
```

EnableExclamations(var res:tvalue)

włącz okrzyki

//Przykład - Misja AM 03

```
// Dialogue 2c (if a car is driven by some character)
every 0$2 trigger GetEngine(IsInUnit(MacMillan))=engine_solar Marked 7 do
begin
  DisableExclamations;
  Say(MacMillan,'D2c-JMM-1');
  if JMMinvehicle then
    Say(MacMillan,'D2c-JMM-1a');
  EnableExclamations;
end;
```

DisableExclamations(var res:tvalue)

wyłącz okrzyki

//Przykład - Misja AM 03

```
// Dialogue 2c (if a car is driven by some character)
every 0$2 trigger GetEngine(IsInUnit(MacMillan))=engine_solar Marked 7 do
begin
  DisableExclamations;
  Say(MacMillan,'D2c-JMM-1');
  if JMMinvehicle then
    Say(MacMillan,'D2c-JMM-1a');
  EnableExclamations;
end;
```

Dostępność - Fabryka:

AvailableChassisList(un:integer)

zwraca listę dostępnych komponentów (podwozia) - fabryka, warsztat

//Przykład - Misja AM 15a

```
if un[1] in AvailableChassisList(factory) then
  components:=components ^ un[1]
else
```

AvailableEngineList(un:integer)

zwraca listę dostępnych komponentów (silniki) - fabryka, warsztat

//Przykład - Misja AM 15a

```
// define engine
if un[2] in AvailableEngineList(factory) then
  components:=components ^ un[2]
else
  if (not components[1] in [us_light_wheeled, ar_hovercraft, ar_light_trike]) and (Get_Resources(side)[3]>9) then
    components:=components ^ engine_siberite
  else
    components:=components ^ engine_combustion;
```

AvailableControlList(un:integer)

zwraca listę dostępnych komponentów (sterowanie) - fabryka, warsztat

//Przykład - Misja 15a

```
// define control // <---- upravit podle poctu lidi a tak podobne
if un[3] in AvailableControlList(factory) then
  components:=components ^ un[3]
else
  begin
    if control_remote in controls then
      components:=components ^ control_remote
    else
      if control_computer in controls then
        components:=components ^ control_computer
      else
        components:=components ^ control_manual;
  end;
```

AvailableWeaponList(un:integer)

zwraca listę dostępnych komponentów (bronie) - fabryka, warsztat

//Przykład - Misja AM 15a

```
// checkovani Amiku zda muzou vyrobit siberitovku
every 0$30.5 trigger TICK>time_to_end Marked 4 do
var factory,zdroje;
begin
  factory:=FilterAllUnits([ [f_side,us], [f_btype,b_factory] ]);
  if factory then
    begin
      factory:=factory[1];
      if us_siberium_rocket in AvailableWeaponList(factory) then
        begin
          zdroje:=GetResources(GetBase(factory));
          if (zdroje[1]>70) and
            ((zdroje[2]>30) or (zdroje[3]>70)) and
            (zdroje[3]>60) then
            begin
              if zdroje[3]>70 then
                ComConstruct(factory,us_morphling,engine_combustion,control_remote,us_siberium_rocket)
              else
                ComConstruct(factory,us_morphling,engine_siberite,control_remote,us_siberium_rocket);
              us_sib_wait:=false;
            end
          else
            us_sib_wait:=true;
        end;
      end;
    end;
  enable;
end;
```

Constructable(chas:integer, eng:integer, con:integer, weap:integer)

zwraca true, jeżeli pojazd może być wyprodukowany

{brak przykładu}

Technologie:

GetBuildingTechReq(kind:integer, nation:integer)

zwraca technologie wymaganych dla rodzaju budynku, narodu (0 - wolne -1 - niedostępna dla narodu)

{brak przykładu}

GetTechTechsReq(tech:integer)

zwraca listę technologii wymaganych do opracowania tech

{brak przykładu}

GetTechLab(tech:integer)

zwraca rodzaj laboratorium wymaganego do opracowania tech

{brak przykładu}

TechNationAvailable(tech:integer, nation:integer)

zwraca true jeżeli tech jest dostępna dla narodu

{brak przykładu}

GetActResearch(un:integer)

zwraca technologie aktualnie opracowywana w laboratorium un, 0 jeżeli nie lub spec

//Przykład - Misja RU 13 cont

```
// Nastavit použití artefaktu.  
If LabySArtefaktem then  
Begin  
  For L in LabySArtefaktem do  
    If GetActResearch (L) then nastav_pouziti (ART_GRAY, L)  
    else nastav_pouziti (ART_INSTANT, L);  
  End else zrus_pouziti_artefaktu;
```

SetSpecResearch(un:integer, time:integer, auto:boolean)

ustala (laboratorium) un dla SpecResearch automatyczny tryb czasu - dłuższe używaj ostrożnie!

//Przykład - Misja AM 15a

```
// klikani na ikony v laboratorich
on ArtifactUsed(side,number,p1,p2) do
var un,succes;
begin
if number<=3 then
begin
SetSpecResearch(nearest_labs[number],art_difficulty[number],true);
SetWorkingProgress(nearest_labs[number],art_progres[number]);
art_in_research:=art_in_research ^ number;
for un=1 to art_num do
if nearest_labs[un]=nearest_labs[number] then
DeInit_Lab(un);
end
else
begin
succes:=false;
case number of
// volani samotne artefaktovni funkce
4: succes:=Show_me_map(side,p1,p2);
5: succes:=Destroy_siberite(p1);
6: succes:=Mass_Teleport(p1,p2);
end;

if succes then
begin
can_use_artefact:=Delete(can_use_artefact,number-3);
can_use_artefact:=Insert(can_use_artefact,number-3,false); // oznaceni ikony do seda

Shadow_Artefact(number-3); // zasednuti ikony
Enable(20+number); // zpusleni counteru pro dalsi pouziti artefaktu
end;
end;
end;
```

SetNoActivity(un:integer)

resetuje aktywność (laboratorium) un używaj BARDZO ostrożnie!

//Przykład - Misja RU 13 cont

```
// Zruší probíhající použití artefaktu.
Function zrus_pouziti_artefaktu;
Begin
// Zrušit výzkum.
SetNoActivity (MericiLaborator);
MericiLaborator = 0;
End;
```

Multiplayer:

GetMultiplayerSetting(index:integer)

zwraca wartość ustawień trybu multiplayer z określonego index'u

```
// Przykład - Alien base

// game settings for real Multiplayer
if MultiPlayer then
begin
  def_base_level:=      GetMultiplayerSetting(0);
  def_amount_of_people:= GetMultiplayerSetting(1);
  def_skill_level:=     GetMultiplayerSetting(2);
  def_starting_resources:= GetMultiplayerSetting(3);
  def_shipments_density:= GetMultiplayerSetting(4);
  def_extra_oil_deposits:= GetMultiplayerSetting(5);
  def_extra_sib_deposits:= GetMultiplayerSetting(6);

  if Game_Type=1 then // Kill'em All
    def_shared_vision:=  False
  else
    def_shared_vision:=  GetMultiplayerSetting(10);

  def_morale_flags:=    GetMultiplayerSetting(11);
  def_siberite_detection:= GetMultiplayerSetting(12);
  def_people_respawning:= GetMultiplayerSetting(13);
  def_amount_of_apemen:= GetMultiplayerSetting(14);
  def_amount_of_siberit:= GetMultiplayerSetting(15);
  def_public_score:=    GetMultiplayerSetting(16);
  def_siberite_bomb:=    GetMultiplayerSetting(17);
  def_builduptime:=     GetMultiplayerSetting(20);
end
```

SetMultiScore(side:integer, score:integer)

ustala wynik multiplayer dla wybranej strony

```
//Przykład - Babylon

// Checking for people in the island
function count_king;
var side,list,un;
begin
  side:=[];
  list:=FilterUnitsInArea( king_territory, [ [f_type,unit_human], f_not, [f_inside] ] );
  for un in list do
    side:=side union GetSide(un);

  if side<>1 then
    exit;
  side:=side[1];

  SetTag(side, GetTag(side)+1);
  SetMultiScore(side, GetTag(side) div 10);
  if GetTag(side)>=final_count then
    Enable(3);
end;
```

GetMultiScore(side:integer)

otrzymuje wynik multiplayer dla wybranej strony

{brak przykładu}

ResetMultiScore(var res:tvalue)

resetuje wynik multiplayer dla wybranej strony

//Przykład - Babylon

```
export function init_kings_counting;
var i;
begin
  mp_with_score:=true;
  ResetMultiScore;

  case def_kings_age of      // setting time of the Age
  0: final_count:= 500 ;
  1: final_count:= 1000 ;
  2: final_count:= 1500 ;
  3: final_count:= 2000 ;
  end;

  Enable(2);
end;
```

Inne:

RaiseSailEvent(num:integer)

bepośrednio wywołuje losowe zdarzenie SailEvent(num)

//Przykład - Misja RU 05 cont

```
every 3$12 + 1$0 trigger vsichni_gone do
begin
  SetFuel([ArScout1,ArScout2,ArScout3], 100);
  PlaceUnitArea(ArScout1, exit_area, false);
  if difficulty>=2 then
    PlaceUnitArea(ArScout2, exit_area, false);
  if difficulty>=3 then
    PlaceUnitArea(ArScout3, exit_area, false);
  RaiseSailEvent(5051);
  if difficulty>=2 then
    RaiseSailEvent(5052);
  if difficulty>=3 then
    RaiseSailEvent(5053);
  enable;
end;
```

HexInfo(x:integer, y:integer)

zwraca numer jednostki na hex'ie, 0 dla wolnego i -1 dla środowiska

//Przykład - Misja RU 06

```
function get_build(x,y,btype);
begin
  result:=HexInfo(x,y);
  if result<1 then result:=0;
  if btype then
    if btype<>GetBType(result) then result:=0
  else
    if unit_building<>GetType(result) then result:=0;
end;
```

TeleportUnit(un:integer, x:integer, y:integer, range:integer, sound:boolean)

spróbuj teleportować jednostkę un do [x,y]. Używa efektu warp. Sound opcjonalne. zwraca true, jeżeli udane.
Tylko dla jednostek zewnętrznych. Używa zakresu do auto-wyszukiwania innych celi jeżeli [x,y] jest niezdatny do użytku.

//Przykład - Misja RU 15a

```
function Mass_Teleport(x,y);
var vysledek,base,
    list,un;
begin
    vysledek:=true;
    base:=GetBase(nearest_labs[3]);
    if (GetResources(base)[3] >= sib_to_teleport) then
        begin
            SetAreaMapShow(teleport_area,1);
            for un=1 to 11 do
                begin
                    display_strings:=[(11-un)];
                    wait(0$1);
                end;
            display_strings:="";
            SetAreaMapShow(teleport_area,0);

            list:=FilterAllUnits([ f_not, [f_type,unit_building] ]);
            for un in list do
                TeleportUnit(un,x,y,12,true);          // 12 je vzdalenost, do ktore se to bude pokouset umistnit
            end
        end
    else
        vysledek:=false;
        result:=vysledek;
    end;
end;
```

GetCustomRate(basicrate:integer, tab:integer, skill:integer)

zwraca premię basicrate (lub karę) dla umiejętności (-10..50) używa skilltable tab (0..6)
Zwykle używane umiejętności są 0..10. Tabela 0 nie daje premii, table 6 daje dużą premię.
np. tempo badań naukowca to GetCustomRate(zaklad,4,GetSkill(human,skill_scientistic))

//Przykład - Misja RU 12 comment

```
GetCustomRate(zaklad,4,GetSkill(human,skill_scientistic))
```

ValidHex(x:integer, y:integer)

true jeżeli hex x,y jest valid (0 - invalid, 1 - bordering, 2 - normal).

//Przykład - Misja AM 01

```
every 0$0.5 trigger (not killhim) and follow do //joan followuje JMM
var tmp,tmp2,tmp3,x,y,px,py,dir;
begin
    tmp=getdistunits(john,joan);
    x=getx(john); y=gety(john);
    tmp2=[]; tmp3=[];
    if tmp>=7 then begin
        for dir=0 to 5 do begin
            px=shiftx(x,dir,tmp-2);
            py=shifty(y,dir,tmp-2);
            if ValidHex(px,py) then
                begin
                    tmp2=tmp2^[px,py];
                    tmp3=tmp3^[getdistunitxy(joan,px,py)];
                end
            end;
        tmp=bestfromlistbylist(tmp2,tmp3);
        commovexy(joan,tmp[1],tmp[2]);
    end;
    enable;
end;
```

Contaminate(side:integer, x:integer, y:integer)

bezwzględne zanieczyszczenie dostawy syberytu jeżeli udane, (wywołuje warunek oraz) zwraca true

//Przykład - Misja AM 15

```
Function StartTheAttack;
begin
  ComMoveXY(Omar_vehicles ^ Masto, 184, 82);
  Wait(0$3);
  McMove(100, Heike ^ Omar_Soldiers, [[184, 82]], mc_no_stop);
  Wait(0$5);
  ComContaminate(Omar_Scientists, 177, 69);
end;
```

SelectUnits(units:plist)

dodaje zaznaczenie do jednostki

{brak przykładu}

DeselectUnits(units:plist)

usuwa zaznaczenie z jednostki

//Przykład - Misja AM 15

```
{...}
  if IsSelected(Friend) then
    DeselectUnits(Friend);
{...}
```

AddExperience(hum:integer, sk:integer, points:integer)

dodaje punkty w umiejętności sk dla człowieka hum

//Przykład - Misja RU 08

```
every 0$0.1+0$0.05 marked 203 do // Adding experiency for shooters
var un,shoot_filter;
begin
  shoot_filter:=FilterUnitsInArea(ShootArea,[[f_type,unit_human],[f_class,class_engineer],[f_hastask],[f_nation,nation_russian]]);
  // display_strings:=[shoot_filter]; //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  if shoot_filter<>[] then
    for un in shoot_filter do
      if GetTaskList(un)<>[] then
        if (GetTaskList(un)[1][1]='A') AND (InArea(GetTaskList(un)[1][2],GetTaskList(un)[1][3],ShootArea)) then
          AddExperience(un,skill_combat,1);
          enable;
        end;
```

EffectTeleport(x:integer, y:integer)

tworzy efekt na hex'ie [x,y]

//Przykład - Misja AM 13a

```
on destinationunreachable(un) do
if (un=ruv1) or (un=ruv2) or (un=ruv3) then begin
  effectteleport(getx(un),gety(un));
  comhold(un);
  wait(50);
  destroyunit(un);
end;
```

ArtContamination(x:integer, y:integer, percentage:integer)

tworzy skażoną strefę (percentage procent zanieczyszczenia rakietą syberytową)

//Przykład - Babylon

```
//----- Artefact num. 3 -----//
export function Contamine(side,p1,p2);
var vysledek,old_side,
    station;
begin
    vysledek:=true;
    station:=GetBase(GetTag(200+side*10+15));

    if station and (GetResources(station)[3] >= GetTag(206)) then
        begin
            ArtContamination(p1,p2,15);
            SetResourceType(station,mat_siberit,GetResources(station)[3]-GetTag(206));
        end
    else
        vysledek:=false;

    result:=vysledek;
end;
```

EnableVideoExclamations(var res:tvalue)

okrzyki video włączone

//Przykład - Misja RU 03

```
CenterOnXY(142,80);
DisableVideoExclamations;
InGameOff;
interface_hidden:=true;
kecy_behem_utoku_arabu;
InGameOn;
EnableVideoExclamations;
```

DisableVideoExclamations(var res:tvalue)

okrzyki video wyłączone

//Przykład - Misja RU 03

```
CenterOnXY(142,80);
DisableVideoExclamations;
InGameOff;
interface_hidden:=true;
kecy_behem_utoku_arabu;
InGameOn;
EnableVideoExclamations;
```

SetBName(units:plist, bname:string)

ustala budynkowi wyświetlanie ciągu znaków określony przez użytkownika

//Przykład - Misja AM 05 cont

```
export depot, laboratore, budovy;
on BuildingComplete(build) do
begin
    budovy:=budovy union build;
    {...}
    if GetBType(build)=b_depot then begin
        depot:=build;
        SetBName(build, 'delta');
    end;
    {...}
end;
```

AreaToMatrix(area:integer)

tworzy macierz ze strefy

{brak przykładu}

WaypointsFromMatrix(matrix:plist, areas:plist)

tworzy waypoints bezpośrednio używając macierzu stref. Strefy są na liście: [[x,y,r],...] i tu musi być co najmniej jeden element na liście. Droga jest tworzona losowo: losowe punkty w strefach są wybierane, jedna z najlepszych dróg poprzez te punkty jest wybierana losowo.

{brak przykładu}

SideShoot(un:integer)

zwraca stronę, która strzelała do jednostki un

{brak przykładu}

UnitShoot(un:integer)

zwraca jednostkę, która strzelała do jednostki un

{brak przykładu}

SetUnitDisplayNumber(un:integer, num:integer)

wyświetla numer num nad jednostką un (nawet w finale Original War)... zero oznacza nie wyświetlać nic

{brak przykładu}

CopySkills(un1:integer, un2:integer)

kopiuje umiejętności między jednostkami (un1 ->un2)

//Przykład - Alien base

```
// returns list of created units
function prepare_people(side,count,cl,x,y,r, var commander);
var j,i,un;
begin
  result=[];
  for i=1 to count do
    begin
      {...}
      un:=CreateHuman;

      if global_defined then
        begin
          j:=i;
          if cl=class_soldier then
            j:=i+soldiers_created;
          CopySkills(global_units[cl][j],un);
          SetAttr(un, attr_speed, GetAttr( global_units[cl][j], attr_speed) );
          SetAttr(un, attr_stamina, GetAttr( global_units[cl][j], attr_stamina) );

        end;
      {...}
    end;
  end;
```

Bibliografia

Opisy większości funkcji zostały przetłumaczone z dokumentu powstałego w oparciu o zapiski programistów	http://www.owsupport.com/files/SAILFunctions.rtf
Przykłady kodów zostały przytoczone z plików umieszczonych w grze	Original War Game



PRAWA AUTORSKIE:

Niniejszy kurs, w tym jego struktura, a w szczególności teksty, tłumaczenia i zdjęcia stanowią wyłączną własność Daniela Gabryśa oraz podlegają ochronie zgodnie z ustawą z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (tj. Dz. U. z 2000r., Nr 80, poz. 904, z późn. zm.).

Wszelkie teksty autorstwa innych osób, dodane do kursu jako rozwinięcie kursu lub treści, które zostały przetłumaczone, są własnością ich autorów i umieszczone za ich zgodą.

Żadna część tego kursu, łącznie z tekstami, zdjęciami, grafiką oraz znaki towarowe, nie może być kopiowana ani rozpowszechniana, w tym również w celu wykorzystania w całości lub w części w innych kursach, w publikacjach elektronicznych jak również w wersji materialnej, bez pisemnej zgody autorów konkretnych części kursu.

