

Liste der Scriptfunktionen im TES III: Construction Set

Killfetzner



15. März 2011

Inhaltsverzeichnis

1	Einleitung	7
2	Scriptbefehle	8
3	Scriptfunktionen	10
3.1	Activate	10
3.2	AddItem	10
3.3	AddSoulGem	10
3.4	AddSpell	11
3.5	AddToLevCreature	11
3.6	AddToLevItem	11
3.7	AddTopic	12
3.8	AIActivate	12
3.9	AIEscort	13
3.10	AIEscortCell	13
3.11	AIFollow	14
3.12	AIFollowCell	14
3.13	AITravel	14
3.14	AIWander	15
3.15	BecomeWerewolf	15
3.16	BetaComment	16
3.17	Cast	16
3.18	CellChanged	16
3.19	CellUpdate	17
3.20	CenterOnCell	17
3.21	CenterOnExterior	17
3.22	ChangeWeather	17
3.23	Choice	18
3.24	ClearForceJump	19
3.25	ClearForceMoveJump	19
3.26	ClearForceRun	19
3.27	ClearForceSneak	19
3.28	ClearInfoActor	19
3.29	CreateMaps	19
3.30	Disable	20
3.31	DisableLevitation	20
3.32	DisablePlayerControls	20
3.33	DisablePlayerFighting	20
3.34	DisablePlayerJumping	21
3.35	DisablePlayerLooking	21
3.36	DisablePlayerMagic	21
3.37	DisablePlayerViewSwitch	21
3.38	DisableTeleporting	21
3.39	DisableVanityMode	22
3.40	DontSaveObject	22
3.41	Drop	22
3.42	DropSoulGem	22
3.43	Enable	23
3.44	EnableBirthMenu	23
3.45	EnableClassMenu	23
3.46	EnableInventoryMenu	23
3.47	EnableLevelUpMenuMenu	23
3.48	EnableLevitation	23
3.49	EnableMagicMenu	24
3.50	EnableMapMenu	24

3.51	EnableNameMenu	24
3.52	EnablePlayerControls	24
3.53	EnablePlayerFighting	24
3.54	EnablePlayerJumping	24
3.55	EnablePlayerLooking	24
3.56	EnablePlayerMagic	24
3.57	EnablePlayerViewSwitch	24
3.58	EnableRaceMenu	24
3.59	EnableStatsReviewMenu	24
3.60	EnableStatsMenu	25
3.61	EnableTeleporting	25
3.62	Equip	25
3.63	ExplodeSpell	25
3.64	Face	26
3.65	FadeIn	26
3.66	FadeOut	26
3.67	FadeTo	26
3.68	Fall	27
3.69	FillJournal	27
3.70	FillMap	27
3.71	FixMe	27
3.72	ForceGreeting	27
3.73	ForceJump	28
3.74	ForceMoveJump	28
3.75	ForceRun	28
3.76	ForceSneak	28
3.77	Get	28
3.78	GetAngle	32
3.79	GetArmorType	32
3.80	GetButtonPressed	32
3.81	GetCurrentAIPackage	33
3.82	GetCurrentWeather	33
3.83	GetDeadCount	34
3.84	GetDistance	34
3.85	GetDetected	34
3.86	GetEffect	34
3.87	GetFactionReaction	35
3.88	GetItemCount	35
3.89	GetJournalIndex	36
3.90	GetLineOfSight	36
3.91	GetMasserPhase	36
3.92	GetPCCell	37
3.93	GetPCFacRep	37
3.94	GetPCRank	37
3.95	GetPos	37
3.96	GetRace	38
3.97	GetSecondsPassed	38
3.98	GetSecundaPhase	38
3.99	GetSoundPlaying	39
3.100	GetSpell	39
3.101	GetSpellEffects	39
3.102	GetStartingAngle	40
3.103	GetStartingPos	40
3.104	GetTarget	40
3.105	GetWeaponType	40
3.106	GoToJail	41
3.107	GoodBye	41

3.108	HasItemEquipped	41
3.109	HasSoulGem	42
3.110	Help	42
3.111	HitAttemptOnMe	42
3.112	HitOnMe	42
3.113	HurtCollidingActor	43
3.114	HurtStandingActor	43
3.115	IsWerewolf	43
3.116	Journal	43
3.117	Lock	44
3.118	LoopGroup	44
3.119	LowerRank	44
3.120	MenuMode	44
3.121	MenuTest	45
3.122	MessageBox	45
3.123	Mod	46
3.124	ModFactionReaction	48
3.125	ModPCFacRep	48
3.126	ModRegion	49
3.127	Move	50
3.128	MoveOneToOne	50
3.129	MoveWorld	50
3.130	ObjectReferenceInfo	51
3.131	OnActivate	51
3.132	OnDeath	51
3.133	OnKnockout	51
3.134	OnMurder	51
3.135	OutputObjCounts	52
3.136	OutputRefCounts	52
3.137	PayFine	52
3.138	PayFineThief	52
3.139	PCClearExpell	52
3.140	PCExpell	53
3.141	PCExpelled	53
3.142	PCForce1stPerson	53
3.143	PCForce3rdPerson	53
3.144	PCGet3rdPerson	53
3.145	PCJoinFaction	53
3.146	PCLowerRank	54
3.147	PCRaiseRank	54
3.148	PlaceAtMe	54
3.149	PlaceAtPC	54
3.150	PlaceItem	54
3.151	PlaceItemCell	55
3.152	PlayBink	55
3.153	PlayGroup	55
3.154	PlayLoopSound3D	56
3.155	PlayLoopSound3DVP	56
3.156	PlaySound	56
3.157	PlaySound3D	56
3.158	PlaySoundVP	56
3.159	PlaySound3DVP	56
3.160	Position	56
3.161	PositionCell	57
3.162	PurgeTextures	57
3.163	RaiseRank	57
3.164	Random	57

3.165	RemoveEffects	58
3.166	RemoveFromLevCreature	58
3.167	RemoveFromLevItem	58
3.168	RemoveItem	58
3.169	RemoveSoulGem	59
3.170	RemoveSpell	59
3.171	RemoveSpellEffects	59
3.172	RepairedOnMe	59
3.173	ResetActors	60
3.174	Resurrect	60
3.175	Rotate	60
3.176	RotateWorld	60
3.177	SameFaction	61
3.178	Say	61
3.179	SayDone	61
3.180	ScriptRunning	62
3.181	Set	62
3.182	SetAtStart	64
3.183	SetDelete	64
3.184	SetAngle	65
3.185	SetFactionReaction	65
3.186	SetLevel	65
3.187	SetPCFacRep	66
3.188	SetJournalIndex	66
3.189	SetPos	66
3.190	SetWereWolfAcrobatics	66
3.191	Show	67
3.192	ShowAnim	67
3.193	ShowGroup	67
3.194	ShowMap	67
3.195	ShowRestMenu	67
3.196	ShowSceneGraph	68
3.197	ShowTargets	68
3.198	ShowVars	68
3.199	SkipAnim	68
3.200	StartCombat	68
3.201	StartScript	69
3.202	StopCellTest	69
3.203	StopCombat	69
3.204	StopScript	69
3.205	StopSound	69
3.206	StreamMusic	70
3.207	TestCells	70
3.208	TestInteriorCells	70
3.209	TestModels	70
3.210	TestThreadCells	70
3.211	ToggleAI	71
3.212	ToggleBorders	71
3.213	ToggleClipping	71
3.214	ToggleCollisionBoxes	71
3.215	ToggleCollisionGrid	71
3.216	ToggleCombatStats	72
3.217	ToggleDebugText	72
3.218	ToggleDialogueStats	72
3.219	ToggleFogOfWar	72
3.220	ToggleFullHelp	72
3.221	ToggleGodMode	73

3.222	ToggleGrid	73
3.223	ToggleKillStats	73
3.224	ToggleLights	73
3.225	ToggleLoadFade	73
3.226	ToggleMagicStats	74
3.227	ToggleMenus	74
3.228	TogglePathGrid	74
3.229	ToggleScripts	74
3.230	ToggleScriptOutput	74
3.231	ToggleSky	75
3.232	ToggleStats	75
3.233	ToggleTextureString	75
3.234	ToggleVanityMode	75
3.235	ToggleWater	75
3.236	ToggleWireFrame	76
3.237	ToggleWorld	76
3.238	TurnMoonRed	76
3.239	TurnMoonWhite	76
3.240	UndoWerewolf	76
3.241	Unlock	76
3.242	UsedOnMe	77
3.243	WakeUpPC	77
3.244	XBox	77

Sachverzeichnis 78

4	Anhang	82
4.1	Konstanten	82
4.2	Liste der magischen Effekte	83
4.3	Inventarscripte	86
4.4	Nützliche globale Variablen	88
4.5	Beispielscripte	90
4.5.1	Teleporteramulett	90
4.5.2	Fallgitter	90
4.5.3	Rüstungspuppe	91
4.5.4	Versiegeltes Buch	91

1 Einleitung

Diese Liste ist als Komplettüberarbeitung der Vorgänger-Version aus 2007 entstanden. Es wurden viele neue Informationen hinzugefügt, Fehler und Ungenauigkeiten ausgebessert und die Navigation dank Links und einem Sachverzeichnis entscheidend verbessert. Außerdem verfügen viele Funktionen nun über ein kleines Scriptbeispiel. Vor allem deshalb ist diese Liste mehr als doppelt so lang wie die Vorgängerversion.

Nach wie vor gilt, dass diese Liste keine Anleitung zum Scripten darstellt. Mit dieser Liste kann man sich Informationen über eine bestimmte Funktion suchen oder nach Möglichkeiten suchen, bestimmte Dinge zu erreichen, aber nicht Scripten lernen, wenn man es nicht schon halbwegs beherrscht. Für das Scriptenlernen empfehle ich das Standardwerk *Morrowind Scripting for Dummies 9* von GhanBuriGhan.

Meine Informationen stammen aus der im Construction Set eingebauten Hilfe, Scripting for Dummies, verschiedenen Foren und meiner eigenen Erfahrung. Ich bedanke mich bei meinen Korrekturlesern Crashtestgoblin, Teridan und Tommy.

Aber warum sollte ich überhaupt Scripten können?

Das Spiel The Elder Scrolls III: Morrowind benutzt eine eigene Scriptsprache um interaktive Elemente des Spiels zu steuern. Diese ist unter dem Namen **Morrowind Script** oder **TES Script** bekannt. Da Oblivion ebenfalls eine komplett andere Scriptsprache verwendet, halte ich den Begriff Morrowind Script für weniger missverständlich.

Für Modder sind Scripte ein sehr mächtiges Werkzeug. Mit ihnen lassen sich viele sonst unmögliche Dinge in Plugins realisieren. Zum Beispiel lassen sich Gegenstände bewegen, Aktionen des Spielers detektieren und andere Objekte dynamisch auf diese Aktion reagieren. Allgemein kann man mit Scripten die Interaktivität des Spiel stark erhöhen.

Morrowind Script hat den großen Vorteil, dass es nur über sehr wenige grundlegende Befehle verfügt und somit leicht erlernbar ist. Der Nachteil liegt in der teilweise sehr beschränkten Funktionalität der Scriptsprache. Es gibt viele Bereiche in der Spielmechanik, die von Scripten nicht oder nur unzureichend beeinflusst werden können. Sobald man also über die Standardfunktionen hinausgehen will, ist zwangsläufig Kreativität gefragt.

Wo treffen wir beim Modden eigentlich auf Scripte oder Scriptfunktionen? Da gibt es mehrere Möglichkeiten:

- **Scripte:** Natürlich trifft man Scriptfunktionen in Scripten selbst an. Scripte können mit dem im Construction Set eingebauten Editor erstellt und bearbeitet werden. Scripte gibt es sowohl als lokale, wie auch als globale Variante.
- **Resultboxen von Dialogen:** Auch in den Resultboxen von Dialogen lassen sich Scriptfunktionen ausführen. Es gibt sogar gewisse Funktionen, die nur hier funktionieren. Allgemein liegt die Funktionalität aber weit unter der eines Scripts, da der Code nur ein einziges Mal ausgeführt wird.
- **Konsole:** Die letzte Möglichkeit Scriptfunktionen zu verwenden, stellt die Konsole im Spiel selbst dar. Hier lassen sich vor allem Debuggingfunktionen aufrufen. Allerdings kann sie auch für weitreichende Spielmanipulationen verwendet werden, was den Konsolenfunktionen ungerechter Weise den Ruf von Cheats eingebracht hat. Die Funktionalität der Konsole liegt noch einmal unter der der Resultboxen. Hier lässt sich immer nur ein einziger Befehl gleichzeitig ausführen.

Für diese Liste wurde die deutsche Version von Morrowind zusammen mit den deutschen Versionen von Tribunal und Bloodmoon verwendet, also die aktuellste, offizielle Version. In anderen Versionen können sich einige der Funktionen unterscheiden.

Ich wünsche allen Lesern viel Spaß und Erfolg bei ihren Scriptprojekten.

Killfetzner

2 Scriptbefehle

Jede Programmiersprache verfügt über einen Satz an festen Befehlen. Diese Befehle werden in Morrowind Scriptbefehle (Script Commands) genannt und sind im Vergleich mit höheren Programmiersprachen oder selbst anderen Scriptsprachen nur sehr wenige. Mit diesen Befehlen lassen sich die grundlegenden Funktionen der Scriptsprache, wie deklarieren von Variablen oder prüfen von Bedingungen benutzen.

Hier eine Liste aller Befehle in Morrowind Script:

Name	Funktion
begin	Startet ein Script
end	Beendet ein Script
short	Deklariert eine Short-Variable
long	Deklariert eine Long-Variable
float	Deklariert eine Float-Variable
if	Prüft eine Bedingung und führt im Wahrheitsfall den folgenden Code aus
elseif	Prüft eine Bedingung, wenn die vorhergehende Bedingung falsch war, und führt im Wahrheitsfall den folgenden Code aus
else	Führt den folgenden Code aus, wenn die vorhergehende Bedingung falsch war
endif	Beendet einen if-Block
while	Prüft eine Bedingung und führt den folgenden Code so lange aus, wie die Bedingung wahr ist
endwhile	Beendet einen while-Block
set ... to	Weist einer Variablen einen Wert zu
return	Beendet die Ausführung des Scripts für diesen Frame und springt auf die erste Zeile zurück
;	Kommentar
ID->Funktion	Funktionsaufruf
ID.Variable	Variablenaufruf

Jedes Script in Morrowind ist nach dem selben Schema aufgebaut:

```
begin Scriptname

short var1           ;Deklaration der Variablen
long var2
float var3

set var3 to GameHour ;eigentliches Script

if ( var3 > 7 )
    ;mache irgendwas
elseif ( var3 >= 5 )
    ;mache etwas anderes
else
    ;mache noch etwas anderes
endif

set var1 to 10

while ( var1 > 0 )
    ;mache irgendwas
    set var1 to var1 - 1
endwhile

end
```

Zunächst wird das Script mit *begin* und einem frei wählbaren Scriptnamen gestartet. Danach werden die lokalen Variablen deklariert (alle deklarierten Variablen werden automatisch mit 0 initialisiert). Danach kommen die eigentlichen Scriptbefehle und -funktionen. Zum Schluss wird das Script mit *end* beendet.

Die Variablen können jeweils verschiedene Zahlen verarbeiten. *short* und *long* verarbeiten jeweils ganze Zahlen und *float* eine Gleitkommazahl. Die Grenzen sind:

- **short:** -32,768 bis 32767 (bzw. 16 bit)
- **long:** -2,147,483,648 bis 2,147,483,647 (bzw. 32 bit)
- **float:** Speichert eine reelle Zahl in der Exponentialschreibweise. Da dies nur eine näherungsweise Darstellung ist, kann nicht vorausgesagt werden, wie viele der Nachkommastellen stimmen bis Rechenfehler auftreten. Morrowind-Script unterstützt maximal 6 Nachkommastellen, allerdings sind mit bei Rechenoperationen schon Rechenfehler in der dritten Nachkommastelle aufgefallen. Die größte speicherbare Zahl einer float-Variable liegt bei $\pm 3.4 \cdot e^{\pm 38}$.

Variablennamen müssen mit einem Buchstaben beginnen. Morrowind-Script ist dabei nicht case sensitive, das heißt, es achtet nicht auf Groß- und Kleinschreibung. Variablennamen müssen eindeutig sein, es kann also keinen Namen geben, der sowohl als *short* und als *float* definiert ist. Weiterhin dürfen Variablennamen nicht gleich zu Funktions- oder Befehlsnamen sein. Zusätzlich gibt es noch eine Reihe von reservierten Variablennamen, die nicht für andere Zwecke verwendet werden sollten, eine Liste dieser findet sich im Kapitel Inventarscripte.

Es ist zu beachten, dass Morrowind-Script eine etwas eigenwillige Art der Aktivierung hat. Jedes lokale Script, also jedes Script, das auf einem Objekt liegt, wird einmal in jedem Frame ausgeführt, in dem das Objekt geladen ist (was normalerweise bedeutet, dass sich der Spieler in der gleichen Zelle befindet). Globale Scripte, das sind alle restlichen Scripte, laufen jeden Frame, nachdem sie einmal gestartet wurden, bis sie manuell beendet werden. Entsprechend vorsichtig müssen Scripte geschrieben werden, damit man keine Endlosschleife erzeugt.

3 Scriptfunktionen

In diesem Teil werden die einzelnen Scriptfunktionen in alphabetischer Liste aufgeführt. Zu jeder Funktion werden alle möglichen Parameter, Grenzen oder bekannte Fehler angegeben. Optionale Parameter sind in eckigen Klammern [...] angegeben. Weiterhin gibt es zu vielen Funktionen ein Codebeispiel. Ein thematisch sortiertes Sachverzeichnis ist zwischen dieser Liste und dem Anhang zu finden.

3.1 Activate

"Object_ID"->Activate

Aktiviert das aufrufende Objekt. Das Objekt führt die gleiche Aktion aus, dass vom Spieler in der Standardeinstellung mit der Leertaste ausgelöst wird. Zum Beispiel werden Türen geöffnet oder geschlossen (letzteres funktioniert nur beim Spieler) oder das Inventar von Containern aktiviert.

```
if ( OnActivate == 1 )
    MessageBox "Ihr habt das Siegel aufgebrochen."
    Activate
endif
```

Verwandte Funktionen: *OnActivate*

3.2 AddItem

"Object_ID"->AddItem, "Item_ID", count

Fügt dem aufrufenden Objekt die in *count* angegebene Anzahl des angegebenen Gegenstands ins Inventar ein. Das aufrufende Objekt muss ein Inventar besitzen, also der Spieler, ein NPC, eine Kreatur oder ein Container sein. Wird bei *count* ein negativer Wert angegeben, werden $32768 + count$ Einheiten hinzugefügt, also zum Beispiel werden, wenn *count* gleich -1 ist, 32767 Gegenstände hinzugefügt. *count* akzeptiert keine Variablen!

"Player"->AddItem, "Gold_001", 1000

Verwandte Funktionen: *Drop*, *GetItemCount*, *RemoveItem*

3.3 AddSoulGem

"Object_ID"->AddSoulGem, "Creature_ID", "SoulGem_ID"

Fügt dem aufrufenden Objekt einen Seelenstein des angegebenen Typs mit einer Seele der angegebenen Kreatur ins Inventar ein. Das aufrufende Objekt muss ein Inventar besitzen, also der Spieler, ein NPC, eine Kreatur oder ein Container sein. Die angegebene Kreatur muss keine Seele besitzen (sprich der Seelenwert kann 0 sein). Der entstehende Seelenstein hat in diesem Fall einen Wert von 0. Die Größe des Seelenwertes und die maximale Größe des Seelensteins spielen keine Rolle. Mit dieser Funktion lässt sich jede Seele in jedem Seelenstein unterbringen.

Als Seelenstein können nur die IDs der sechs fest vorgegebenen Seelensteine verwendet werden:

Misc_SoulGem_Azura, *Misc_SoulGem_Common*, *Misc_SoulGem_Grand*, *Misc_SoulGem_Greater*, *Misc_SoulGem_Lesser*, *Misc_SoulGem_Petty*. Wird ein anderer Miscellaneous-Gegenstand als Seelenstein angegeben, wird dieser einfach dem Inventar des aufrufenden Objekts hinzugefügt. Die Scriptfunktion wirkt in diesem Fall wie ein *AddItem*, "*MiscItem_ID*", 1.

```
"Player"->AddSoulGem, "cliff racer", "Misc_SoulGem_Greater"
```

Verwandte Funktionen: *DropSoulGem*, *HasSoulGem*, *RemoveSoulGem*

3.4 AddSpell

```
"Actor_ID"->AddSpell, "Spell_ID"
```

Fügt dem Magiemenu des aufrufenden Actors, also dem Spieler, einem NPC oder einer Kreatur, den angegebenen Zauber hinzu. Mit Zauber ist in diesem Fall jedes Objekt aus dem Spellmaking-Reiter gemeint.

```
"Player"->AddSpell, "ataxia"
```

Verwandte Funktionen: *GetSpell*, *RemoveSpell*

3.5 AddToLevCreature

```
AddToLevCreature, "LeveledCreature_ID", "Creatue_ID", pc_level
```

(Tribunal) Fügt der angegebenen Kreaturenliste die angegebene Kreatur hinzu. Diese kann ab dem angegebenen Level des Spielers generiert werden. Mit einer zusätzlichen Abhängigkeit von Bloodmoon kann statt einer bestimmten Kreatur auch eine weitere Kreaturenliste angegeben werden.

Warnung: Diese Funktion sollte nur auf selbst erstellte Listen angewendet werden, da die Änderungen im Spielstand gespeichert werden!

```
AddToLevCreature, "bm_ex_hirforest", "cliff racer", 1
```

Verwandte Funktionen: *RemoveFromLevCreature*

3.6 AddToLevItem

```
AddToLevItem, "LeveledItem_ID", "Item_ID", pc_level
```

(Tribunal) Fügt der angegebenen Gegenstandsliste den angegebenen Gegenstand hinzu. Dieser kann ab dem angegebenen Level des Spielers generiert werden. Statt eines Gegenstandes kann auch eine weitere Gegenstandsliste angegeben werden. **Warnung:** Diese Funktion sollte nur auf selbst erstellte Listen angewendet werden, da die Änderungen im Spielstand gespeichert werden!

```
AddToLevItem, "random_drinks_01", "potion_nord_mead", 5
```

Verwandte Funktionen: *RemoveFromLevItem*

3.7 AddTopic

AddTopic, "Themenname"

Fügt das angegebene Thema der Themenliste des Spielers hinzu, das heißt der Spieler kann dieses Thema nun in Dialogen auswählen.

AddTopic, "Die neusten Gerüchte"

Verwandte Funktionen: -

3.8 AIActivate

"Actor_ID"->AIActivate, "Object_ID"[, reset]

(**Tribunal**) Die Funktion ist zwar auch im normalen Morrowind enthalten, dort aber nicht funktionsfähig.

Der aufrufende Actor, also ein NPC oder einer Kreatur, wird versuchen den angegebenen Gegenstand zu aktivieren. Sollte dies gelingen, führt er je nach Gegenstandstyp eine bestimmte Aktion aus:

Objekttyp	Aktion
Actor/Container	Zum Ziel laufen
Activator	Löst OnActivate aus
(normale) Tür	Öffnen (aber nicht Schließen, wenn Tür geöffnet)
(teleportierende) Tür	Betreten, wenn Ziel sich in der gleichen Zelle befindet, sonst Absturz
Buch/Schriftrolle	Öffnet das Buch zum Lesen (für den Spieler!)
sonstiger Gegenstand	Aufnehmen

Der angegebene Gegenstand muss *reference persistent* sein. Befindet sich der Gegenstand außerhalb der Reichweite des aufrufenden Actors, wird er versuchen zu ihm zu gehen. In diesem Fall muss der Ort des Gegenstands über Pathgrids mit der aktuellen Position des Actors verbunden sein und die Entfernung sollte nicht zu groß gewählt werden. Wenn der Gegenstand seine Position nach Aufrufen der Funktion verändert (zum Beispiel ein NPC der weiterläuft), läuft der aufrufende Actor nur zur ehemaligen Position und dreht sich an dieser wild im Kreis. Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

Wichtig: Diese Funktion sendet kein *AIPackageDone*! Somit muss die Ausführung auf anderem Weg detektiert werden oder der Actor läuft ohne Ende auf der Stelle, friert ein oder führt die selbe Aktion immer wieder aus (bei Büchern sehr nervig, da sie den Spieler jedes Mal ins Lesemenü zwingen).

```
short doonce
if ( doonce == 0 )
    if ( GetDistance, "Player" < 512 )
        set doonce to 1
        AIActivate, "act_dwe_lever_a"
    endif
endif
```

Verwandte Funktionen: *AIEscort*, *AIEscortCell*, *AIFollow*, *AIFollowCell*, *AITravel*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.9 AIEscort

"Actor_ID"->AIEscort, "Actor2_ID", duration, xpos, ypos, zpos[, reset]

Der aufrufende Actor eskortiert den angegebenen Actor, zum Beispiel den Spieler, zur angegebenen Position im Exterior. Das heißt er läuft zu dem angegebenen Punkt, wartet aber, wenn der zu eskortierende Actor sich zu weit von ihm entfernt (mehr als 512 Editoreinheiten). Nähert sich dieser wieder (auf weniger als 256 Editoreinheiten), läuft der erste Actor weiter. Die angegebene Position muss über Pathgrids mit seiner aktuellen Position verbunden sein und die Entfernung sollte nicht zu groß gewählt werden. *Duration* gibt die maximale Zeit in Stunden an, die der aufrufende Actor den zweiten Actor eskortiert, bevor er den Versuch abbricht. Wird als Dauer 0 angegeben, wird keine Zeitbeschränkung ausgewertet. Die Funktion sendet ein *AIPackageDone* wenn der aufrufende Actor den in *xpos*, *ypos*, und *zpos* angegebenen Ort erreicht oder die maximale Dauer erreicht ist. Verlässt der eskortierte Spieler das Exterior durch eine Teleporttür, während er nahe genug am eskortierenden Actor ist, wird dieser mitteleportiert. In der Interiorzelle bewegt sich dieser aber nicht weiter.

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```
short doonce
if ( doonce == 0 )
    set doonce to 1
    AIEscort, "Player", 0, -25000, -12340, 1018 ;Balmora, vor dem Hlaalu-Ratsgebäude
endif
```

Verwandte Funktionen: *AIActivate*, *AIEscortCell*, *AIFollow*, *AIFollowCell*, *AITravel*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.10 AIEscortCell

"Actor_ID"->AIEscortCell, "Actor2_ID", "Zellenname", duration, xpos, ypos, zpos[, reset]

Der aufrufende Actor eskortiert den angegebenen Actor, zum Beispiel den Spieler, zur angegebenen Position in der angegebenen Innenzelle. Das heißt er läuft zu dem angegebenen Punkt, wartet aber, wenn der zu eskortierende Actor sich zu weit von ihm entfernt (mehr als 512 Editoreinheiten). Nähert sich dieser wieder (auf weniger als 256 Editoreinheiten), läuft der erste Actor weiter. Die angegebene Position muss über Pathgrids mit seiner aktuellen Position verbunden sein und die Entfernung sollte nicht zu groß gewählt werden. *Duration* gibt die maximale Zeit in Stunden an, die der aufrufende Actor den zweiten Actor eskortiert, bevor er den Versuch abbricht. Wird als Dauer 0 angegeben, wird keine Zeitbeschränkung ausgewertet. Die Funktion sendet ein *AIPackageDone* wenn der aufrufende Actor den in *xpos*, *ypos*, und *zpos* angegebenen Ort erreicht oder die maximale Dauer erreicht ist. Wechselt der eskortierte Spieler die Zelle durch eine Teleporttür während der nahe genug am eskortierenden Actor ist, wird dieser mitteleportiert. In der neuen Zelle bewegt sich dieser aber nicht weiter.

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```
short doonce
if ( doonce == 0 )
    set doonce to 1
    AIEscortCell, "Player", "Balmora, MagiERGilde", 0, -628, -1080, -754
endif
```

Verwandte Funktionen: *AIActivate*, *AIEscort*, *AIFollow*, *AIFollowCell*, *AITravel*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.11 AIFollow

"Actor_ID"->AIFollow, "Actor2_ID", duration, xpos, ypos, zpos[, reset]

Die genaue Umkehrfunktion zu *AIEscort*. Der aufrufende Actor folgt dem angegebenen Actor bis zur angegebenen Exteriorposition. *Duration* gibt die maximale Zeit in Stunden an, die der aufrufende Actor dem zweiten Actor folgt, bevor er den Versuch abbricht. Wird als Dauer 0 angegeben, wird keine Zeitbeschränkung ausgewertet. Die Funktion sendet ein *AIPackageDone* wenn der aufrufende Actor den in *xpos*, *ypos*, und *zpos* angegebenen Ort erreicht oder die maximale Dauer erreicht ist. Wechselt der eskortierende Spieler die Zelle durch eine Teleporttür oder durch einen Reiseservice, wird der folgende Actor mitteleportiert.

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```
short doonce
if ( doonce == 0 )
    set doonce to 1
    AIFollow, "Player", 0, 0, 0, 0 ;folgt dem Spieler für immer
endif
```

Verwandte Funktionen: *AIActivate*, *AIEscort*, *AIEscortCell*, *AIFollowCell*, *AITravel*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.12 AIFollowCell

"Actor_ID"->AIFollowCell, "Actor2_ID", "Zellenname", duration, xpos, ypos, zpos[, reset]

Die genaue Umkehrfunktion zu *AIEscortCell*. Der aufrufende Actor folgt dem angegebenen Actor bis zur angegebenen Position in der angegebenen Zelle. *Duration* gibt die maximale Zeit in Stunden an, die der aufrufende Actor dem zweiten Actor folgt, bevor er den Versuch abbricht. Wird als Dauer 0 angegeben, wird keine Zeitbeschränkung ausgewertet. Die Funktion sendet ein *AIPackageDone* wenn der aufrufende Actor den in *xpos*, *ypos*, und *zpos* angegebenen Ort erreicht oder die maximale Dauer erreicht ist. Wechselt der eskortierende Spieler die Zelle durch eine Teleporttür oder durch einen Reiseservice, wird der folgende Actor mitteleportiert.

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```
short doonce
if ( doonce == 0 )
    set doonce to 1
    AIFollowCell, "Player", "Balmora, Magierrgilde", 0, -628, -1080, -754
endif
```

Verwandte Funktionen: *AIActivate*, *AIEscort*, *AIEscortCell*, *AIFollow*, *AITravel*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.13 AITravel

"Actor_ID"->AITravel, xpos, ypos, zpos[, reset]

Der aufrufende Actor geht zur angegebenen Position in der aktuellen Zelle. Die angegebene Position muss über Pathgrids mit seiner aktuellen Position verbunden sein und die Entfernung sollte nicht zu groß gewählt werden.

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```

short doonce
if ( doonce == 0 )
    set doonce to 1
    AITravel, -628, -1080, -754
endif

```

Verwandte Funktionen: *AIActivate*, *AIEscort*, *AIEscortCell*, *AIFollow*, *AIFollowCell*, *AIWander*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.14 AIWander

"Actor_ID"->AIWander, distance, duration, hour[, idle1, ..., idle9][, reset]

Bringt den aufrufenden Actor dazu sich zufällig um seine derzeitige Position zu bewegen. *distance* definiert, wie weit er sich maximal von seinem Startpunkt entfernt, *duration*, wie lange er maximal braucht um zu seinem Ausgangspunkt zurückzukehren und *hour* um welche Tageszeit er mit seiner Wanderung beginnt. Die neun idle-Werte bezeichnen die Wahrscheinlichkeiten, dass er während er stillsteht bestimmte Animationen ausführt (zum Beispiel entspricht idle4 der Wahrscheinlichkeit den Kopf zu schütteln). Diese müssen nicht im Funktionsaufruf angegeben sein. Die idle-Werte müssen Werte zwischen 0 und 100 annehmen. *duration* und *hour* akzeptieren nur Werte zwischen 0 und 23.

Die einzelnen idle-Parameter stehen für folgende Animationen:

Nummer	Animation
idel1	Stillstehen
idel2	Gewicht von einem Bein auf das andere verlagern
idel3	Umsehen
idel4	Kopf kratzen und schütteln
idel5	Menschenfrau: nichts
	Kleidung an der Schulter ordnen
idel6	Menschenfrau: Hand an die Hüfte
	Menschen: Gähnen und Strecken
idel7	Tierrassen: Auf Hände schauen
	Auf rechte Hand schauen
idel8	Hand ans Kinn legen
idel9	Menschen: nach Waffe greifen
	Tierrassen: Kopf kratzen und Füße scharren

Die Funktion des optionalen Parameters *reset* ist unbekannt, allerdings scheint der Actor den Befehl nicht auszuführen, wenn irgendein Wert für *reset* angegeben wird.

```

short doonce
if ( doonce == 0 )
    set doonce to 1
    AIWander, 512, 5, 0, 0, 100 ;führt immer idle2 aus
endif

```

Verwandte Funktionen: *AIActivate*, *AIEscort*, *AIEscortCell*, *AIFollow*, *AIFollowCell*, *AITravel*, *GetAIPackageDone*, *GetCurrentAIPackage*

3.15 BecomeWerewolf

"NPC_ID"->BecomeWerewolf

(Bloodmoon) Verwandelt den aufrufenden NPC in seine Werwolfform.

Verwandte Funktionen: *IsWerewolf*, *SetWerewolfAcrobatics*, *UndoWerewolf*

3.16 BetaComment

BetaComment, "Text"
BC, "Text"

(Konsole) Um diese (zum Debugging sehr praktische) Funktion zu benutzen, muss zunächst in der *Morrowind.ini* eine Änderung vorgenommen werden. In der Sektion *[General]* muss in der Zeile *Beta Comment File* eine Textdatei eingetragen werden, zum Beispiel *Beta Comment File=betacomment.txt*. Danach muss eine leere Textdatei dieses Namens im Morrowind-Hauptverzeichnis erstellt werden.

Im Spiel kann diese Funktion nur benutzt werden, wenn die Konsole geöffnet ist und mit der Konsole ein beliebiges Objekt ausgewählt ist (ID des Objekts steht in der Titelleiste der Konsole in Klammern hinter dem Wort Konsole). Durch diese Funktion wird der in Anführungszeichen stehende Text zusammen mit einigen anderen Daten als Zeile in der vorher angegebenen Textdatei gespeichert. Dies sieht dann zum Beispiel so aus:

9/7/2010 (14:08) Morrowind.esm 8/12/2002 (16:55) Killfetzter active_de_p_bed_09 Balmora, Kriegergilde 444 -489 -314 "Beispieltext"

Diese Daten geben der Reihe nach Systemdatum, Systemzeit, Name der zuerst geladenen ESM-Datei, Datum und Uhrzeit der letzten Änderung der ESM-Datei, Windows-Benutzername, ID des ausgewählten Objekts, Zelle des ausgewählten Objekts, X-, Y- und Z-Koordinate des ausgewählten Objekts und den eingegebenen Text wieder. Somit ist es möglich während eines Test viele Kommentare aufzuschreiben und diese mit allen nötigen Informationen in geordneter Form zu besitzen. Die so angelegte Datei kann auch als tabgetrennte Tabelle in eine Tabellenverarbeitung geladen werden.

Verwandte Funktionen: -

3.17 Cast

"Object_ID"->Cast, "Spell_ID", "Actor_ID"

Das aufrufende Objekt zaubert den angegebenen Zauber auf den angegebenen Actor. Zauber und Ziel müssen zueinander passen (also ein Zauber auf sich selbst muss auch auf den aufrufenden Actor zielen)! Wichtig ist hierbei, dass nur ein Actor auf sich selbst gerichtete Zauber sprechen kann. Damit Activator diese Funktion auch benutzen können, muss *ExplodeSpell* benutzt werden. Ist das aufrufende Objekt kein Actor, darf es sich nicht in einem Inventar befinden und muss eine Textur besitzen (das Mesh EditorMarker.nif funktioniert zumindest nicht). Auf diese Art ausgelöste Zauber verbrauchen keine Magie und gelingen immer. Zauber auf Ziel werden ausgeführt, unabhängig davon, ob sie ihr Ziel überhaupt erreichen können.

Cast, "poison bloom", "Player"

Verwandte Funktionen: *ExplodeSpell*

3.18 CellChanged

CellChanged

Die Funktion liefert für einen Frame eine 1 wenn die Zelle gewechselt wurde, sonst eine 0.

```
if ( CellChanged == 1 )
    MessageBox "Ihr habt eine neue Zelle betreten."
endif
```

Verwandte Funktionen: -

3.19 CellUpdate

Funktion defekt! Die Funktion sollte die aktuelle Zelle neu laden.

Verwandte Funktionen: -

3.20 CenterOnCell

CenterOnCell, "Zellenname"
COC, "Zellenname"

(Konsole) Teleportiert den Spieler in der angegebenen Zelle. Der Spieler erscheint (meistens) an der Position des zuerst gesetzten DoorMakers in der Zelle oder wenn keiner existiert an der Position des zuletzt hinzugefügten Objekts (was ihn meistens bewegungsunfähig macht und ein kurzzeitiges Abschalten der Kollision mit *ToggleCollision* erforderlich macht).

Die Funktion kann zwar auch in Scripts verwendet werden, da die genaue Zielposition aber nicht bekannt ist, ist die Funktion *PositonCell* vorzuziehen.

Verwandte Funktionen: *CenterOnExterior*, *Position*, *PositionCell*

3.21 CenterOnExterior

CenterOnExterior, xpos, ypos
COE, xpos, ypos

(Konsole) Teleportiert den Spieler in Exteriorzelle mit den angegebenen X- und Y-Koordinaten. Der Spieler erscheint in der Mitte der Exteriorzelle. Da er in anderen Objekten erscheinen kann, ist ein kurzzeitiges Abschalten der Kollision mit *ToggleCollision* manchmal erforderlich.

Die Funktion kann zwar auch in Scripts verwendet werden, bietet aber nur eine ungenauere Zielbestimmung als die Funktion *Position*, weswegen diese vorzuziehen ist. Das Beispiel schickt den Spieler zum Assarnibibiberg:

```
COE, 14, -4
```

Verwandte Funktionen: *CenterOnCell*, *Position*, *PositionCell*

3.22 ChangeWeather

ChangeWeather, "Region_ID", type

Ändert das aktuelle Wetter in der angegebenen Region auf das angegebene Wetter. Die Zuordnung von Typen zu Wetter ist wie folgt:

Typ	Wetter
0	Sonnenschein
1	Leicht bewölkt
2	Nebbig
3	Stark bewölkt
4	Regen
5	Gewitter
6	Aschesturm
7	Peststurm
(Typen nur mit Bloodmoon verfügbar)	
8	Schnee
9	Blizzard

Die Regions-IDs können im Construction Set unter World→Regions gefunden werden. Die Standardregionen sind (da teilweise nicht konsistent durchbenannt):

Region	Region-ID
Ascadia-Inseln	Ascadian Isles Region
Aschländer-Region	Ashlands Region
Azuraküste	Azura's Coast Region
Bitterküste	Bitter Coast Region
Molag Amur	Molag Mar Region
Roter Berg	Red Mountain Region
Sheogorad	Sheogorad
Weidenländer	Grazelands Region
Westspalte	West Gash Region
Gramfeste	Mournhold Region
Brodir Hain	Brodir Grove Region
Felsaad-Küste	Felsaad Coast Region
Hirstaang-Wald	Hirstaang Forest Region
Isinfier-Ebenen	Isinfier Plains Region
Moesring-Berge	Moesring Mountains Region
Thirsk	Thirsk Region

Das Beispiel ändert das Wetter an der Bitterküste zu einem Gewitter:

```
ChangeWeather, "Bitter Coast Region", 5
```

Verwandte Funktionen: *GetCurrentWeather*, *GetWindSpeed*, *ModRegion*

3.23 Choice

```
Choice, "Antwort 1", return1[, "Antwort 2", return2, ..., "Antwort 20", return20]
```

(Dialog) Diese Funktion kann nur während Dialogen ausgegeben werden. Es ist egal, ob sie aus einer Resultbox oder einem Script aufgerufen wird, allerdings sollte die Resultbox die weitaus häufigere Anwendung sein. Die Funktion ist ähnlich der MessageBox. Beide geben dem Spieler eine Möglichkeit Entscheidungen durch verschiedene Antworten zu geben. Bei der *Choice*-Funktion gibt es keine direkte Möglichkeit eine Frage zu stellen, diese muss entweder im zuvor angezeigten Dialog enthalten sein oder durch einen *MessageBox*-Aufruf in den Dialog geschrieben werden. Zusätzlich muss bei der *Choice*-Funktion der Rückgabewert jeder Antwort nach dieser angegeben werden. Dies ermöglicht es zwar in einer verzweigten Dialogstruktur viele Antwortkombinationen auf den gleichen Dialog zu führen (da mehrere Antworten den gleichen Rückgabewert haben), allerdings kann es auch zu unvorhergesehenen Problemen führen, wenn die Rückgabewerte nicht eindeutig sind. So können vor allem in den großen Themen wie

"Befehle" oder "Greeting" lüngenau konstruierte *Choice*-Funktionen, oder Antworten darauf, dazu führen, dass man die Antwort eines komplett anderen *Choice*-Aufrufs erhält. Die vom Spieler getroffene Antwort einer *Choice* kann über eine Dialogantwort gefunden werden. Dazu muss diese Antwort **über** der Antwort stehen, die die *Choice* ausgelöst hat und in den Feldern *Function/Variable* eine Zeile mit *Function*→*Choice* gewählt haben.

Choice, "Fahrt fort.", 1, "Kommt auf den Punkt.", 2, "Ihr langweilt mich zu Tode. Sterbt!", 5

Verwandte Funktionen: *MessageBox*

3.24 ClearForceJump

3.25 ClearForceMoveJump

3.26 ClearForceRun

```
"NPC_ID"->ClearForceJump
"NPC_ID"->ClearForceMoveJump
"NPC_ID"->ClearForceRun
```

(**Tribunal**) Hebt ein *ForceJump*, *ForceMoveJump*, *ForceRun* auf dem aufrufenden NPC auf.

Verwandte Funktionen: *ClearForceSneak*, *GetForce...*, *Force...*

3.27 ClearForceSneak

```
"NPC_ID"->ClearForceSneak
```

Hebt ein *ForceSneak* auf dem aufrufenden NPC auf.

Verwandte Funktionen: *ClearForce...*, *GetForce...*, *Force...*

3.28 ClearInfoActor

ClearInfoActor

(**Dialog**) Diese Funktion kann nur in der Resultbox von Dialogantworten aufgerufen werden. Sie verhindert, dass die entsprechende Antwort in der Themensektion des Spielertagebuchs auftaucht.

Verwandte Funktionen: -

3.29 CreateMaps

CreateMaps, "Plugin.esp"

(**Konsole**) Um diese sehr nützliche, aber auch sehr spezielle Funktion nutzen zu können, muss zunächst im Morrowind-Hauptverzeichnis ein neuer, leerer Ordner mit Namen *Maps* angelegt werden. Danach muss in der Morrowind.ini in der Sektion *[General]* die Zeile *Create Maps Enable=0* auf *Create Maps Enable=2* geändert werden.

Diese Funktion erlaubt es Screenshots jeder einzelnen Exteriorzelle des Spiels zu machen. Diese Screenshots sind identisch zu den lokalen Karten im Spiel und je Zelle 256x256 Pixel groß. Sie werden im Bitmap-Format im zuvor angelegten Maps-Ordner gespeichert. Die Wahl der esp-Datei scheint den Ausgang des Aufrufs nicht zu beeinflussen.

Warnung: Die Ausführung dieser Funktion dauert sehr lange (> 1 Stunde), da jede Zelle des Spiels geladen wird.

CreateMaps, "Morrowind.esm"

Verwandte Funktionen: -

3.30 Disable

"Object_ID"->Disable

Macht das aufrufende Objekt unsichtbar und deaktiviert seine Kollisionsabfrage. Scripte auf diesem Objekt werden weiterhin ausgeführt. Der Effekt wird mit *Enable* aufgehoben. **Achtung:** Das Deaktivieren der Kollisionsabfrage funktioniert nur in Innenzellen direkt. In Außenzellen muss zusätzlich die Zelle neu geladen werden, damit auch die Kollisionspositionen aktualisiert werden.

Verwandte Funktionen: *Enable*, *GetDisabled*, *SetDelete*

3.31 DisableLevitation

DisableLevitation

Macht Levitation unmöglich, das heißt jeder Levitationszauber auf dem Spieler wird sofort aufgehoben und der Wert des GameSettings *sLevitateDisabled* (Sektion Menus, Standardinhalt: "Levitation ist hier nicht möglich") eingeblendet.

Anmerkung Da das TribunalMain-Script jeden Frame ein *EnableLevitation* aufruft, sollte eine zuverlässige Deaktivierung auch jeden Frame aufgerufen werden. Dies befreit aber nicht von der Pflicht selbst dafür zu sorgen, dass die Levitation wieder erlaubt wird.

Verwandte Funktionen: *EnableLevitation*

3.32 DisablePlayerControls

DisablePlayerControls

Alle Menus verschwinden und der Spieler kann sich nicht mehr bewegen oder kürzer der Spieler kann sich nur noch umsehen. Die Funktion wird mit *EnablePlayerControls* wieder aufgehoben.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.33 DisablePlayerFighting

DisablePlayerFighting

Der Spieler kann keine Waffe mehr ziehen oder wegstecken. Wenn er jedoch bereits eine Waffe gezogen hat oder er eine über die Kurzwahltasten erreichen kann, kann er trotzdem noch kämpfen.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.34 DisablePlayerJumping

DisablePlayerJumping

Der Spieler kann nicht mehr springen.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.35 DisablePlayerLooking

DisablePlayerLooking

Der Blick des Spieler wird automatisch in die im Editor nördliche Richtung ausgerichtet (Achtung: In Interiorzellen zeigt Norden im Editor oft in eine andere Richtung als im Spiel.) und er kann sich nicht mehr umsehen.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.36 DisablePlayerMagic

DisablePlayerMagic

Der Spieler kann sich nicht mehr auf einen Zauber vorbereiten. Wenn er jedoch die Hände bereits zum Zaubern gehoben hat, kann er weiterhin zaubern. Außerdem kann er, wenn er einen Zauber auf den Kurzwahltasten gespeichert hat, diesen Zauber nach wie vor ausführen.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.37 DisablePlayerViewSwitch

DisablePlayerViewSwitch

Der Spieler kann nicht mehr zwischen den einzelnen Ansichten (First Person, Third Person, Vanity-Mode, rotierend) wechseln. Er bleibt in der Ansicht, die während des Funktionsaufrufs aktiv war.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*

3.38 DisableTeleporting

DisableTeleporting

Macht Teleportation mit Zaubereffekten (Almsivi Intervention, Göttlicher Eingriff, Rückkehr) unmöglich. **Achtung:** Gesciptete Teleporte wie Teleporteramulette sind hiervon nicht betroffen! **Anmerkung** Da das TribunalMain-Script jeden Frame ein *EnableTeleporting* aufruft, sollte eine zuverlässige Deaktivierung auch jeden Frame aufgerufen werden. Dies befreit aber nicht von der Pflicht selbst dafür zu sorgen, dass die Teleportation wieder erlaubt wird.

Verwandte Funktionen: *EnableTeleporting*

3.39 DisableVanityMode

DisableVanityMode

Die Kamera rotiert nicht mehr selbstständig um den Spieler, wenn sich dieser zu lange nicht bewegt hat.

Verwandte Funktionen: *DisablePlayer...*, *EnablePlayer...*, *GetPlayer...Disabled*, *EnableVanityMode*, *GetVanityModeDisabled*

3.40 DontSaveObject

"Object_ID"->DontSaveObject

Veränderungen des aufrufenden Objekts werden nicht im Spielstand gespeichert. Dies ist zum Beispiel eine Möglichkeit Spielstände klein zu halten, wenn ein Plugin viele periodisch oder zufällig bewegte Objekte hat, zum Beispiel Fahnen im Wind oder Ähnliches.

Achtung: Die Funktion muss jeden Frame aufgerufen werden!

Verwandte Funktionen: -

3.41 Drop

"Actor_ID"->Drop, "Item_ID", count

Der aufrufende Actor lässt die in *count* angegebene Anzahl des angegebenen Items zu Boden fallen. Es sollte sicher gegangen werden, dass er aufrufende Actor die zu droppenden Items auch besitzt, da sonst schwere Inventarfehler auftreten können. *count* akzeptiert keine Variablen! Weiterhin ist die Funktion in dem Sinne verbuggt, dass jeder fallengelassene Gegenstand zu Füßen des Spielers erscheint, egal wer ihn fallengelassen hat.

"Player"->Drop, "Gold_001", 1000

Verwandte Funktionen: *AddItem*, *GetItemCount*, *RemoveItem*

3.42 DropSoulGem

"Actor_ID"->DropSoulGem, "Creature_ID"

Der aufrufende Actor lässt einen Seelenstein mit der angegebenen Seele fallen. Der Actor muss einen solchen Seelenstein im Inventar haben, sonst kann das Spiel abstürzen. Hat der Actor mehrere Seelensteine mit dieser Art von Seele im Inventar, wird der Seelenstein mit der ältesten Seele (also der zuerst gefangenen) zuerst fallen gelassen. Weiterhin ist die Funktion in dem Sinne verbuggt, dass jeder fallengelassene Seelenstein zu Füßen des Spielers erscheint, egal wer ihn fallengelassen hat.

"Player"->DropSoulGem, "cliff racer"

Verwandte Funktionen: *AddSoulGem*, *HasSoulGem*, *RemoveSoulGem*

3.43 Enable

"Object_ID"->Enable

Macht das aufrufende Objekt wieder sichtbar und aktiviert seine Kollisionsabfrage, wenn es vorher disabled war. **Achtung:** Das Aktivieren der Kollisionsabfrage funktioniert nur in Innenzellen direkt. In Außenzellen muss zusätzlich die Zelle neu geladen werden, damit auch die Kollisionspositionen aktualisiert werden.

Verwandte Funktionen: *Disable*, *GetDisabled*

3.44 EnableBirthMenu

3.45 EnableClassMenu

EnableBirthMenu

EnableClassMenu

Öffnet das Sternzeichen- beziehungsweise Klassenauswahlmenu. Wird zur Charaktererstellung benötigt. **Achtung:** Änderungen am Spieler nach der Charaktererstellung können zu schweren Fehlern führen!

Verwandte Funktionen: *Enable...Menu*

3.46 EnableInventoryMenu

EnableInventoryMenu

Erzeugt das Inventar für den Spieler. Wird zur Charaktererstellung benötigt.

Verwandte Funktionen: *Enable...Menu*

3.47 EnableLevelUpMenuMenu

EnableLevelUpMenu

Öffnet das Levelaufstiegsfenster.

Verwandte Funktionen: *Enable...Menu*

3.48 EnableLevitation

EnableLevitation

(**Tribunal**) Erlaubt das Levitieren wieder, wenn es vorher disabled war.

Verwandte Funktionen: *DisableLevitation*

3.49 EnableMagicMenu

3.50 EnableMapMenu

EnableMagicMenu

EnableMapMenu

Erzeugt das Zaubermenu beziehungsweise die Karte für den Spieler. Wird zur Charaktererstellung benötigt.

Verwandte Funktionen: *Enable...Menu*

3.51 EnableNameMenu

EnableNameMenu

Öffnet das Namensauswahlfenster des Spielers. Wird zur Charaktererstellung benötigt.

Verwandte Funktionen: *Enable...Menu*

3.52 EnablePlayerControls

3.53 EnablePlayerFighting

3.54 EnablePlayerJumping

3.55 EnablePlayerLooking

3.56 EnablePlayerMagic

3.57 EnablePlayerViewSwitch

EnablePlayer...

Hebt die entsprechenden Disable-Funktionen auf.

Verwandte Funktionen: *DisablePlayer...*, *GetPlayer...Disabled*

3.58 EnableRaceMenu

EnableRaceMenu

Öffnet das Rassenauswahlfenster des Spielers. Wird zur Charaktererstellung benötigt.

Achtung: Änderungen am Spieler nach der Charaktererstellung können zu schweren Fehlern führen!

Verwandte Funktionen: *Enable...Menu*

3.59 EnableStatsReviewMenu

EnableStatsReviewMenu

Öffnet das Kontrollfenster vom Ende der Charaktererstellung. Wird zur Charaktererstellung benötigt.

Achtung: Änderungen am Spieler nach der Charaktererstellung können zu schweren Fehlern führen!

Verwandte Funktionen: *Enable...Menu*

3.60 EnableStatsMenu

EnableStatsMenu

Erzeugt das Statusfenster des Spielers. Wird zur Charaktererstellung benötigt.

Verwandte Funktionen: *Enable...Menu*

3.61 EnableTeleporting

EnableTeleporting

Erlaubt das Teleportieren wieder, wenn es zuvor disabled war.

Verwandte Funktionen: *DisableTeleporting*

3.62 Equip

"NPC_ID"->Equip, "Item_ID"[, count]

(Tribunal) Die Funktion war auch in Original-Morrowind vorhanden, allerdings verbuggt. Erst mit Tribunal ist sie korrekt zu verwenden.

Die Funktion zwingt den aufrufenden NPC die (optionale) Anzahl des angegebenen Gegenstands auszurüsten. Für Gegenstände, die keine Rüstung, Kleidung oder Waffe sind, löst dies die Funktion aus, wie wenn man sie im Inventarmenu auf das Charakterbild zieht, also zum Beispiel Trinken für Tränke. Für diesen zweiten Fall ist auch der *count*-Parameter gedacht. Der aufrufende NPC kann somit gezwungen werden mehrere gleiche Tränke oder Zutaten auf einmal zu nehmen.

Achtung: Es sollte vor diesem Aufruf sicher gestellt werden, dass der aufrufende NPC die auszurüstenden Gegenstände auch in seinem Inventar hat, da sonst Fehler auftreten können!

```
if ( "Player"->GetItemCount, "daedric club" >= 1 )
  if ( "Player"->HasItemEquipped, "daedric club" == 0 )
    "Player"->Equip, "daedric club"
  endif
endif
```

Verwandte Funktionen: *HasItemEquipped*

3.63 ExplodeSpell

"Activator_ID"->ExplodeSpell, "Spell_ID"

(Tribunal) Diese Funktion lässt den aufrufenden Activator einen Zauberspruch auf Berührung auf sich selbst sprechen. Damit der Zauber einen spürbaren Effekt bekommt, muss er eine Area besitzen.

```
"Act_BM_axe_hrothmund"->ExplodeSpell, "fire bite"
```

Verwandte Funktionen: *Cast*

3.64 Face

"Actor_ID"->Face, xpos, ypos

Diese Funktion lässt den aufrufenden Actor seine derzeitige Tätigkeit unterbrechen und ihn zu den angegebenen Koordinaten blicken. Nach einem kurzen Moment fährt er mit seiner vorherigen Tätigkeit fort. Als Koordinaten können auch Variablen übergeben werden.

"aengoth"->Face, 1234, -432

Verwandte Funktionen: -

3.65 FadeIn

FadeIn, time

Lässt einen (teilweise) undurchsichtigen Bildschirm in der angegebenen Zeit in Sekunden wieder zur normalen Durchsichtigkeit zurückkehren. Die Zeit muss kleiner als 10 Sekunden sein.

Verwandte Funktionen: *FadeOut*, *FadeTo*

3.66 FadeOut

FadeOut, time

Setzt die Durchsichtigkeit des Bildschirms auf 0, sprich der Spieler sieht außer dem HUD nur noch schwarz. *time* gibt die Zeit in Sekunden an, bis der Bildschirm undurchsichtig ist. Die Zeit muss kleiner als 10 Sekunden sein.

Verwandte Funktionen: *FadeIn*, *FadeTo*

3.67 FadeTo

FadeTo, alpha, time

Verändert die Durchsichtigkeit des Bildschirms zum angegebenen *alpha*-Wert. *alpha* ist in Prozent angegeben, 0 entspricht dem normalen, durchsichtigen Bildschirm und 100 einem schwarzen, undurchsichtigen. *time* gibt die Zeit in Sekunden an, bis der Bildschirm die gewünschte Durchsichtigkeit erreicht hat. Die Zeit muss kleiner als 10 Sekunden sein.

FadeTo, 50, 3

Verwandte Funktionen: *FadeIn*, *FadeOut*

3.68 Fall

"Actor_ID"->Fall

Der aufrufende Actor fällt auf die Bodenhöhe zurück. Der Actor nimmt dabei keinen Fallschaden. Dies funktioniert nicht, wenn der aufrufende Actor den Effekt eines Levitationszaubers spürt. Allerdings funktioniert es gegen fliegende Kreaturen, diese verlieren durch diese Funktion ihre Flugfähigkeit.

Verwandte Funktionen: -

3.69 FillJournal

FillJournal

(**Konsole**) Fügt jeden möglichen Eintrag in das Tagebuch des Spielers ein.

Achtung: Die Ausführung dauert sehr lange!

Verwandte Funktionen: *FillMap*

3.70 FillMap

FillMap

(**Konsole**) Markiert jede besondere Exteriorzelle auf der Karte des Spielers, also alle Exteriorzellen, die keinen Regionsnamen besitzen.

Verwandte Funktionen: *FillJournal*

3.71 FixMe

"Object_ID"->FixMe

Teleportiert das aufrufende Objekt 128 Editoreinheiten zur Seite. Bei der ersten Eingabe wird der Spieler nach vorne, bei der zweiten nach rechts, bei der dritten nach hinten und der vierten nach links teleportiert. Dies ist zum Beispiel praktisch, wenn ein NPC eine Tür versperrt oder man selbst in einer Wand hängen geblieben ist.

Verwandte Funktionen: -

3.72 ForceGreeting

"Actor_ID"->ForceGreeting

Zwingt den Spieler in einen Dialog mit dem aufrufenden Actor. Dabei ist es egal, wo sich der Actor befindet oder ob der Actor überhaupt Dialogoptionen besitzt.

Achtung: Es sollte sichergestellt sein, dass nur ein Dialog gleichzeitig stattfinden kann. Mit *MenuMode* lässt sich dies leicht überprüfen.

Verwandte Funktionen: -

3.73 ForceJump

3.74 ForceMoveJump

3.75 ForceRun

```
"NPC_ID"->ForceJump  
"NPC_ID"->ForceMoveJump  
"NPC_ID"->ForceRun
```

(**Tribunal**) Zwingt den aufrufenden NPC auf der Stelle zu hüpfen beziehungsweise sich nur noch hüpfend oder rennend fortzubewegen. Die Funktionen werden bei jedem Laden von einem Spielstand oder durch die entsprechenden Clear-Funktionen wieder aufgehoben.

Verwandte Funktionen: *ClearForce...*, *ForceSneak*, *GetForce...*

3.76 ForceSneak

```
"NPC_ID"->ForceSneak
```

Zwingt den aufrufenden NPC zu schleichen. Die Funktion wird bei jedem Laden von einem Spielstand oder durch die entsprechende Clear-Funktion wieder aufgehoben.

Verwandte Funktionen: *ClearForce...*, *Force...*, *GetForce...*

3.77 Get

```
"Object_ID"->Get...
```

Get ist die wahrscheinlich vielseitigste Funktion in Morrowind-Script. Genau genommen sind es einige hundert, einzelne Funktionen, der Übersichtlichkeit halber fasse ich sie aber in diesem Abschnitt zusammen. Jede *Get*-Funktion liefert eine Zahl zurück. Welche Zahlen zurückgeliefert werden ist abhängig davon, welche *Get*-Funktion benutzt wird. Es gibt grundsätzlich drei verschiedene Arten von Rückgabewerten: ganze Zahlen (in short- oder long-Variablen zu speichern), Gleitkommazahlen (in float-Variablen zu speichern) und bool'sche Werte. Letztere bezeichnen eine Wahrheitsentscheidung. Diese Funktionen liefern entweder eine 0 (für falsch) oder eine 1 (für wahr) zurück. Am besten werden diese Werte in einer short-Variablen gespeichert.

Weiterhin lassen sich die *Get*-Funktionen danach unterscheiden, von welchen Objekt sie aufgerufen werden können. Im Folgenden wird eine Tabelle aller *Get*-Funktionen gegeben, zusammen mit ihrem Rückgabewert, von wem sie aufgerufen werden können und gegebenenfalls einer Erklärung. *Get*-Funktionen, die zusätzliche Parameter oder Bedingungen benötigen oder besondere Rückgabewerte haben, werden später separat behandelt.

Name	Rückgabe	aufrufbar von	Erklärung
GetAIPackageDone	bool	Actor (außer PC)	1 in dem Frame wenn ein AIPackage abgearbeitet ist
GetArmorType	short	NPC + PC	siehe unten
GetAttacked	bool	Actor	1 wenn Actor jemals angegriffen wurde
GetBlightDisease	bool	Actor	1 wenn Actor mit Pest infiziert ist
GetButtonPressed	short	-	siehe unten
GetCollidingActor	bool	Object	1 wenn irgendein Actor die Kollisionsabfrage des Objekts auslöst
GetCollidingPC	bool	Object	1 wenn der Spieler die Kollisionsabfrage des Objekts auslöst
GetCommonDisease	bool	Actor	1 wenn Actor mit gewöhnlicher Krankheit infiziert ist
GetCurrentAIPackage	short	Actor (außer PC)	siehe unten
GetCurrentTime	float	-	aktuelle Spielzeit auf zwei Nachkommastellen gerundet
GetCurrentWeather	short	-	siehe unten
GetDeadCount	short	-	siehe unten
GetDisabled	bool	Object	1 wenn Objekt disabled
GetDistance	float	Object	siehe unten
GetDetected	bool	Actor	siehe unten
GetEffect	bool	Actor	siehe unten
GetFactionReaction	short	-	siehe unten
GetForceJump	bool	NPC + PC	(Tribunal)
GetForceMoveJump	bool	NPC + PC	(Tribunal)
GetForceRun	bool	NPC + PC	(Tribunal)
GetForceSneak	bool	NPC + PC	(Tribunal)
GetHealthGetRatio	float	Actor	Bruchteil der maximalen Lebensenergie
GetHealthRatio	-	-	Funktion defekt! <i>GetHealthGetRatio</i> verwenden
GetInterior	bool	-	1 wenn sich der Spieler in Innenzelle befindet
GetItemCount	long	Object	siehe unten
GetJournalIndex	short	-	siehe unten
GetLineOfSight	bool	Object	siehe unten
GetLocked	bool	Door + Cont.	1 wenn verschlossen
GetMasserPhase	short	-	siehe unten
GetPCCell	bool	-	siehe unten
GetPCCrimeLevel	long	-	Kopfgeld des Spielers
GetPCFacRep	short	-	siehe unten
GetPCInJail	bool	-	(Bloodmoon) 1 wenn Spieler im Gefängnis
GetPCJumping	bool	-	(Tribunal) 1 wenn Spieler springt
GetPCRank	short	-	siehe unten
GetPCRunning	bool	-	(Tribunal) 1 wenn Spieler rennt
GetPCSleep	bool	-	1 wenn Spieler rastet oder wartet
GetPCSneaking	bool	-	(Tribunal) 1 wenn Spieler schleicht
GetPCTraveling	bool	-	(Bloodmoon) 1 wenn Spieler Schnellreise benutzt
GetPCVisionBonus	float	-	Summierte Stärke aller Infravisionseffekte auf dem Spieler / 100
GetPlayerControlsDisabled	bool	-	
GetPlayerDistance	float	Object	Funktion im Compiler unbekannt!
GetPlayerFightingDisabled	bool	-	
GetPlayerJumpingDisabled	bool	-	

GetPlayerLookingDisabled	bool	-	
GetPlayerMagicDisabled	bool	-	
GetPlayerViewSwitchDisabled	bool	-	
GetRace	bool	NPC + PC	siehe unten
GetSecondsPassed	float	-	siehe unten
GetSecundaPhase	short	-	siehe unten
GetSecundusPhase	-	-	Funktion defekt! <i>GetSecundaPhase</i> verwenden
GetSoundPlaying	bool	Object	siehe unten
GetSpell	bool	Actor	siehe unten
GetSpellEffects	bool	Actor	siehe unten
GetSpellReadied	bool	NPC + PC	(Tribunal) 1 wenn Actor die Hände zum Zaubern gehoben hat
GetSquareRoot	float	-	(Tribunal) siehe unten
GetStandingActor	bool	Object	1 wenn irgendein Actor auf dem Objekt steht
GetStandingPC	bool	Object	1 wenn der Spieler auf dem Objekt steht
GetStartingAngle	float	Object	siehe unten
GetStartingPos	float	Object	siehe unten
GetTarget	bool	Actor	siehe unten
GetVanityModeDisabled	bool	-	
GetWaterLevel	float	-	(Tribunal) Höhe des Wasserstandes, Absturz wenn Interior kein Wasser besitzt
GetWeaponDrawn	bool	NPC + PC	(Tribunal) 1 wenn Actor die Waffe gezogen hat
GetWeaponType	short	NPC + PC	siehe unten
GetWerewolfKills	short	-	(Bloodmoon) Anzahl der NPC-Kills seit der letzten Verwandlung des Spielers in einen Werwolf, 0 wenn nicht in Werwolf-form
GetWindSpeed	float	-	aktuelle Windgeschwindigkeit auf zwei Nachkommastellen genau
GetStrength	short	Actor	aktuelle Attributwerte (einschließlich aller Modifikationen durch Zaubereffekte)
GetIntelligence	short	Actor	
GetWillpower	short	Actor	
GetAgility	short	Actor	
GetSpeed	short	Actor	
GetEndurance	short	Actor	
GetPersonality	short	Actor	
GetLuck	short	Actor	
GetAcrobatics	short	NPC + PC	aktuelle Fertigkeitswerte (einschließlich aller Modifikationen durch Zaubereffekte)
GetAlchemy	short	NPC + PC	
GetAlteration	short	NPC + PC	
GetArmorer	short	NPC + PC	
GetAthletics	short	NPC + PC	
GetAxe	short	NPC + PC	
GetBlock	short	NPC + PC	
GetBluntWeapon	short	NPC + PC	
GetConjuration	short	NPC + PC	
GetDestruction	short	NPC + PC	
GetEnchant	short	NPC + PC	
GetHandToHand	short	NPC + PC	

GetHeavyArmor	short	NPC + PC	
GetIllusion	short	NPC + PC	
GetLightArmor	short	NPC + PC	
GetLongBlade	short	NPC + PC	
GetMarksman	short	NPC + PC	
GetMediumArmor	short	NPC + PC	
GetMercantile	short	NPC + PC	
GetMysticism	short	NPC + PC	
GetRestoration	short	NPC + PC	
GetSecurity	short	NPC + PC	
GetShortBlade	short	NPC + PC	
GetSneak	short	NPC + PC	
GetSpear	short	NPC + PC	
GetSpeechcraft	short	NPC + PC	
GetUnarmored	short	NPC + PC	
GetHealth	short	Actor	aktueller Wert, nicht Maximalwert
GetMagicka	short	Actor	aktueller Wert, nicht Maximalwert
GetFatigue	short	Actor	aktueller Wert, nicht Maximalwert
GetLevel	short	Actor	(Tribunal)
GetDisposition	short	NPC	Stimmung des NPCs gegenüber dem Spieler
GetFight	short	Actor (außer PC)	AI-Wert
GetFlee	short	Actor (außer PC)	AI-Wert
GetAlarm	short	Actor (außer PC)	AI-Wert
GetHello	short	Actor (außer PC)	AI-Wert
GetReputation	short	NPC + PC	Reputation des Actors
GetArmorBonus	short	Actor	Summierte Stärke aller Zaubereffekte des angegebenen Typs auf dem Actor
GetAttackBonus	short	Actor	Schild
GetBlindness	short	Actor	Angriff festigen
GetCastPenalty	short	Actor	Lärm
GetChameleon	short	Actor	
GetDefendBonus	short	Actor	Zuflucht
GetFlying	short	Actor	Levitation
GetInvisible	bool	Actor	1 wenn Unsichtbarkeit aktiv
GetParalysis	bool	Actor	1 wenn Lähmung aktiv
GetResistBlight	short	Actor	
GetResistCorprus	short	Actor	
GetResistDisease	short	Actor	
GetResistFire	short	Actor	
GetResistFrost	short	Actor	
GetResistMagicka	short	Actor	
GetResistNormalWeapons	short	Actor	
GetResistParalysis	short	Actor	
GetResistPoison	short	Actor	
GetResistShock	short	Actor	
GetSilence	bool	Actor	1 wenn Stille aktiv
GetSuperJump	short	Actor	Sprung
GetSwimSpeed	short	Actor	Flossen
GetWaterBreathing	bool	Actor	1 wenn Festes Wasser aktiv
GetWaterWalking	bool	Actor	1 wenn Wasseratmung aktiv
GetAngle	float	Object	siehe unten
GetPos	float	Object	siehe unten
GetScale	float	Object	(Tribunal)

3.78 GetAngle

"Object_ID"->GetAngle, axis

Liefert den aktuellen Winkel des aufrufenden Objekts bezüglich der angegebenen Achse in Grad. Für *axis* werden die Buchstaben X, Y und Z akzeptiert.

"de_p_desk_01_galbedir"->GetAngle, X

Verwandte Funktionen: *GetPos*, *SetAngle*, *SetPos*

3.79 GetArmorType

"NPC_ID"->GetArmorType, position

Die Funktion liefert einen Wert für die Rüstungsart, die das Rüstungsteil an der angegebenen Position des aufrufenden Actors, besitzt. Eine Liste der möglichen Positionen und Rückgabewerte ist hier zu finden:

Position	Rüstungsteil	Rückgabe	Rüstungsart
0	Helm		
1	Kürass		
2	Linke Schulterplatte		
3	Rechte Schulterplatte		
4	Beinschienen	-1	Keine Rüstung
5	Stiefel	0	Leichte Rüstung
6	Linker Handschuh	1	Mittlere Rüstung
7	Rechter Handschuh	2	Schwere Rüstung
8	Schild		
9	Linke Armschiene		
10	Rechte Armschiene		

```
if ( "Player"->GetArmorType, 5 == 2 )
    MessageBox, "Ihr tragt schwere Stiefel."
endif
```

Verwandte Funktionen: *GetWeaponType*

3.80 GetButtonPressed

GetButtonPressed

Die Funktion liefert für einen Frame die Nummer des MessageBox-Buttons zurück, den der Spieler angeklickt hat. Der erste Button liefert eine 0, der zweite eine 1 und so weiter. Ansonsten liefert die Funktion -1 zurück.

```

short doonce
short button
if ( doonce == 0 )
    MessageBox, "Klickt OK.", "OK"
    set doonce to 1
else
    set button to GetButtonPressed
    if ( button == 0 )
        MessageBox, "Na endlich."
        set doonce to 2
    endif
endif
endif

```

Verwandte Funktionen: *MessageBox*

3.81 GetCurrentAIPackage

"Actor_ID"->GetCurrentAIPackage

Gibt den Wert des vom aufrufenden Actors derzeit ausgeführten AI-Packages zurück. Wenn sich der Actor im Kampf befindet, wird das AIPackage von vor Kampfbeginn zurückgegeben. Eine Liste der Rückgabewerte steht hier:

Rückgabe	AI-Package
-1	Keine Aufgabe (Idle)
0	Wander
1	Travel
2	Escort
3	Follow
4	Activate

Verwandte Funktionen: *AI...*, *GetAIPackageDone*

3.82 GetCurrentWeather

GetCurrentWeather

Gibt den Wert des aktuellen Wetters in der Zelle des Spielers zurück. In einer Innenzelle ohne Wetter wird der Wert des zuletzt erlebten Wetters zurückgegeben. Eine Liste der Rückgabewerte steht hier:

Rückgabe	Wetter
0	Sonnenschein
1	Leicht bewölkt
2	Neblich
3	Stark bewölkt
4	Regen
5	Gewitter
6	Aschesturm
7	Peststurm
(Typen nur mit Bloodmoon verfügbar)	
8	Schnee
9	Blizzard

Verwandte Funktionen: *ChangeWeather*, *GetWindSpeed*, *ModRegion*

3.83 GetDeadCount

GetDeadCount, "Actor_ID"

Liefert die Anzahl der NPCs oder Kreaturen des entsprechenden Typs, die seit Beginn des Spiels gestorben sind. Damit kann man zum Beispiel abfragen, ob ein bestimmter (einmaliger) NPC schon getötet wurde.

```
if ( GetDeadCount, "caius cosades" >= 1 )
    MessageBox, "Ihr seid verdammt."
endif
```

Verwandte Funktionen: -

3.84 GetDistance

"Object_ID"->GetDistance, "Object2_ID"

Liefert den Abstand zwischen den beiden Objekten in Editoreinheiten als float zurück.

```
if ( GetDistance, "Player" < 512 )
    MessageBox, "BUMM!"
endif
```

Verwandte Funktionen: -

3.85 GetDetected

"Actor_ID"->GetDetected, "Actor2_ID"

Liefert eine 1, wenn der zweite Actor vom aufrufenden Actor bemerkt wurde.

Warnung: Die Funktion ist sehr langsam. Sie sollte nur in Ausnahmefällen und sehr sparsam benutzt werden.

```
if ( GetDetected, "Player" < 512 )
    MessageBox, "Habe ich Euch, Ihr Dieb!"
endif
```

Verwandte Funktionen: -

3.86 GetEffect

"Actor_ID"->GetEffect Effect_ID

Liefert wahr, wenn der aufrufende Actor in irgendeiner Form (ob als Zauber, Fähigkeit, Trank oder sonstiges) von dem angegebenen Effekt betroffen ist. Eine Liste der Effekt-IDs befindet sich im Anhang.

Achtung: Diese Funktion ist stark verbuggt. Zum Einen führt jeder Funktionsaufruf außer dem unten gezeigten zum Absturz des Spiels und zum Anderen gibt es eine Reihe von Effekten, die selbst in dieser Form zum Absturz des Spiels führen können. Die angezeigten Fehler sind nicht konsistent (sie unterscheiden sich also von Auftreten zu Auftreten) und können somit nur mit extremer Mühe diesen Funktionen zugeordnet werden. Mindestens folgende Effekte sind verbuggt:

- sEffectRestoreFatigue

- sEffectParalyze (kann mit *GetParalysis* umgangen werden)
- sEffectSilence (kann mit *GetSilence* umgangen werden)
- sEffectSummonFrostAtronach
- sEffectSummonLesserBonewalker
- sEffectSummonScamp
- sEffectWaterBreathing (kann mit *GetWaterBreathing* umgangen werden)

Da jede Wertzuweisung für eine Variable oder jede Vergleichsoperation mit dieser Funktion ebenfalls zum Absturz führen kann, muss folgende Konstruktion verwendet werden (das Setzen des Kommas und der Anführungszeichen kann ebenfalls zum Absturz führen):

```
short effdamhealth
if ( "Player"->GetEffect sEffectDamageHealth )
set effdamhealth to 1
endif
```

Verwandte Funktionen: *GetSpellEffects*, *RemoveEffects*

3.87 GetFactionReaction

GetFactionReaction, "Faction1_ID", "Faction2_ID"

Liefert einen Wert wie sehr die erste Fraktion die zweite Fraktion mag. Dies lässt sich im Fractions-Menü im Construction Set einstellen.

```
if ( GetFactionReaction, "Telvanni", "Mages Guild" < 0 )
  MessageBox, "Die Telvanni mögen die Magiergilde nicht."
endif
```

Verwandte Funktionen: *ModFactionReaction*, *SetFactionReaction*

3.88 GetItemCount

"Object_ID"->GetItemCount, "Item_ID"

Liefert die Anzahl der angegebenen Gegenstände im Inventar des aufrufenden Objekts. Das Objekt muss dafür ein Inventar besitzen.

Achtung: Wenn die Geldmenge des Spielers gespeichert werden soll, sollte auf jeden Fall eine long-Variable benutzt werden.

```
long goldamount
set glodamout to "Player"->GetItemCount, "Gold_001"
```

Verwandte Funktionen: *AddItem*, *Drop*, *RemoveItem*

3.89 GetJournalIndex

GetJournalIndex, "JournalTopic_ID"

Liefert den höchsten Journalindex, den der Spieler zu dem benannten Thema bereits in seinem Tagebuch hat.

```
if ( GetJournalIndex, "C3_DestroyDagoth" == 20 )
    Journal, "C3_DestroyDagoth", 50
endif
```

Verwandte Funktionen: *Journal*, *SetJournalIndex*

3.90 GetLineOfSight

"Object_ID"->GetLineOfSight, "Object2_ID"
"Object_ID"->GetLOS, "Object2_ID"

Liefert eine 1, wenn zwischen den beiden Objekten eine Sichtlinie besteht.

Warnung: Die Funktion ist sehr langsam. Sie sollte nur in Ausnahmefällen und sehr sparsam benutzt werden.

```
if ( GetLOS, "Player" == 1 )
    MessageBox, "Ich sehe Euch."
endif
```

Verwandte Funktionen: -

3.91 GetMasserPhase

GetMasserPhase

Liefert einen Wert für die Mondphase von Masser (dem großen der beiden Monde). Jede Mondphase dauert drei Tage. Folgende Werte werden zurückgegeben:

Rückgabe	Mondphase
0	Neumond oder Interior
1	zwischen Neumond und Halbmond
2	Halbmond
3	zwischen Halbmond und Vollmond
4	Vollmond

Verwandte Funktionen: *GetSecundaPhase*

3.92 GetPCCell

GetPCCell, "Zellenname"

Liefert eine 1, wenn sich der Spieler in der angegebenen Zelle befindet. Der Zellenname muss existieren, allerdings liefert die Funktion auch eine 1, wenn man sich in einer Zelle befindet, deren Name mit dem angegebenen Zellennamen beginnt. So liefert zum Beispiel jede Interiorzelle in Balmora eine 1, wenn als Zellenname Balmora angegeben wird.

```
if ( GetPCCell, "Balmora" == 1 )
    if ( GetPCCell, "Balmora, Magieryilde" == 1 )
        MessageBox, "Ihr befindet Euch in der Magieryilde von Balmora."
    else
        MessageBox, "Ihr befindet Euch irgendwo sonst in Balmora."
    endif
endif
```

Verwandte Funktionen: *GetInterior*

3.93 GetPCFacRep

GetPCFacRep, "Faction_ID"

Liefert den Ruf des Spielers bei der angegebenen Fraktion.

```
if ( GetPCFacRep, "Redoran" >= 100 )
    MessageBox, "Ihr genießt einen guten Ruf bei den Redoran."
endif
```

Verwandte Funktionen: *ModPCFacRep*, *SetPCFacRep*

3.94 GetPCRank

GetPCRank, "Faction_ID"

Liefert den Rang des Spielers in der angegebenen Fraktion. Ist der Spieler kein Mitglied, liefert die Funktion -1 zurück.

```
if ( GetPCRank, "Redoran" == 8 )
    MessageBox, "Ihr seid ein Ratsherr im Hause Redoran"
endif
```

Verwandte Funktionen: *PCJoinFaction*, *PCLowerRank*, *PCRaiseRank*

3.95 GetPos

"Object_ID"->GetPos, axis

Liefert die aktuelle Position des aufrufenden Objekts bezüglich der angegebenen Achse in Editoreinheiten. Für *axis* werden die Buchstaben X, Y und Z akzeptiert.

```
"de_p_desk_01_galbedir"->GetPos, X
```

Verwandte Funktionen: *GetAngle*, *SetAngle*, *SetPos*

3.96 GetRace

"NPC_ID"->GetRace, "Race_ID"

Liefert eine 1, wenn der aufrufende NPC der angegebenen Rasse angehört.

```
if ( "aengoth"->GetRace, "Wood Elf" == 1 )
    MessageBox, "Aengoth ist ein Bosmer."
endif
```

Verwandte Funktionen: -

3.97 GetSecondsPassed

GetSecondsPassed

Liefert die vergangenen Sekunden seit dem letzten Frame als float-Wert. Dies stellt die einzige Scriptmöglichkeit dar, kurze Zeitspannen zu messen. Da die gemessenen Zeitspannen von der Bildwiederholrate und somit (unter anderem) vom verwendeten PC und dem Ort im Spiel abhängig sind, kann man keine glatten Werte erwarten. Bedingungen sollten also immer mit größer oder kleiner gestellt werden.

```
float timer
set timer to timer + GetSecondsPassed
if ( timer > 10 )
    set timer to 0
    MessageBox, "Schon wieder 10 Sekunden rum."
endif
```

Verwandte Funktionen: -

3.98 GetSecundaPhase

GetSecundaPhase

Liefert einen Wert für die Mondphase von Secunda (dem kleinen der beiden Monde). Jede Mondphase dauert drei Tage. In der Hilfe des Construction Sets ist die Funktion fälschlicherweise als GetSecundus-Phase aufgeführt. Folgende Werte werden zurückgegeben:

Rückgabe	Mondphase
0	Neumond oder Interior
1	zwischen Neumond und Halbmond
2	Halbmond
3	zwischen Halbmond und Vollmond
4	Vollmond

Verwandte Funktionen: *GetMasserPhase*

3.99 GetSoundPlaying

"Object_ID"->GetSoundPlaying, "Sound_ID"

Liefert eine 1, wenn der angegebene Sound vom aufrufenden Objekt gespielt wird. Dies stellt eine der wenigen Möglichkeiten dar, bestimmte Aktionen des Spielers zu detektieren, wie zum Beispiel zaubern.

```
short doonce
if ( "Player"->GetSoundPlaying, "mysticism cast" == 1 )
  if ( doonce == 0 )
    MessageBox, "Ihr habt einen Mystikzauber gewirkt."
    set doonce to 1
  endif
endif
```

Verwandte Funktionen: *PlayLoopSound3D*, *PlaySound*, *StopSound*

3.100 GetSpell

"Actor_ID"->GetSpell, "Spell_ID"

Liefert eine 1, wenn sich der angegebene Zauber im Inventar des aufrufenden Actors befindet.

```
"Player"->GetSpell, "poisonbloom"
```

Verwandte Funktionen: *AddSpell*, *RemoveSpell*

3.101 GetSpellEffects

"Actor_ID"->GetSpellEffects, "Spell_ID"

Liefert eine 1, wenn sich der angegebene Zauber auf den aufrufenden Actor auswirkt. Dabei ist es egal, wer den Zauber auf den Actor gesprochen hat.

```
short doonce
if ( "Player"->GetSpellEffects, "poisonbloom" == 1 )
  if ( doonce == 0 )
    MessageBox, "Ihr wurdet von einem Giftblüte-Zauber getroffen."
    set doonce to 1
  endif
endif
```

Verwandte Funktionen: *GetEffect*, *RemoveSpellEffects*, *RemoveEffects*

3.102 GetStartingAngle

"Object_ID"->GetStartingAngle, axis

Liefert den Winkel bezüglich der angegebenen Achse, die das aufrufende Objekt im Construction Set einnimmt, in Grad. Für *axis* werden die Buchstaben X, Y und Z akzeptiert.

"de_p_desk_01_galbedir"->GetStartingAngle, Z

Verwandte Funktionen: *GetStartingPos*, *SetAtStart*

3.103 GetStartingPos

"Object_ID"->GetStartingPos, axis

Liefert die Position bezüglich der angegebenen Achse, die das aufrufende Objekt im Construction Set einnimmt, in Editoreinheiten. Für *axis* werden die Buchstaben X, Y und Z akzeptiert.

"de_p_desk_01_galbedir"->GetStartingPos, X

Verwandte Funktionen: *GetStartingAngle*, *SetAtStart*

3.104 GetTarget

"Actor_ID"->GetTarget, "Actor2_ID"

Liefert eine 1, wenn der aufrufende Actor den angegebenen Actor als Angriffsziel hat. Der aufrufende Actor kann nicht der Spieler sein.

"caius cosades"->GetTarget, "Player"

Verwandte Funktionen: *StartCombat*, *StopCombat*

3.105 GetWeaponType

"NPC_ID"->GetWeaponType

Liefert eine Zahl zurück, die für die Waffenart steht, die der aufrufende Actor gerade ausgerüstet hat. Die Werte für Pfeil und Bolzen scheinen nicht erreichbar zu sein, es wird immer die Waffe (oder -1) zurückgegeben.

Rückgabe	Waffenart
-1	keine Waffe (Hände)
0	Kurzwaffe
1	Langwaffe, einhändig
2	Langwaffe, zweihändig
3	Stumpfe Waffe, einhändig
4	Stumpe Waffe, zweihändig, kurze Reichweite
5	Stumpe Waffe, zweihändig, große Reichweite
6	Speer
7	Axt, einhändig
8	Axt, zweihändig
9	Bogen
10	Armbrust
11	Wurfwaffe
12	Pfeil
13	Bolzen

Verwandte Funktionen: *GetArmorType*

3.106 GoToJail

GoToJail

Schickt den Spieler ins Gefängnis. Er wird zum nächsten Gefängnismarker teleportiert, Zeit verstreicht, in denen er sein Kopfgeld abarbeitet, seine Fertigkeiten verschlechtern sich und seine gestohlenen Gegenstände werden konfisziert. Selbst wenn der Spieler kein Kopfgeld hat, verbringt er einen Tag im Kerker.

Verwandte Funktionen: *GetPCCrimeLevel*, *ModPCCrimeLevel*, *PayFine*, *PayFineThief*, *SetPCCrimeLevel*

3.107 GoodBye

GoodBye

(Dialog) Diese Funktion kann nur während Dialogen ausgegeben werden. Es ist egal, ob sie aus einer Resultbox oder einem Script aufgerufen wird, allerdings sollte die Resultbox die weitaus häufigere Anwendung sein. Die Funktion blendet eine Antwortmöglichkeit mit dem Wert des GameSettings *sGoodbye* (unter Menus, Standard: "Lebt wohl") ein, die den Spieler zwingt den Dialog zu beenden.

Verwandte Funktionen: -

3.108 HasItemEquipped

"NPC_ID"->HasItemEquipped, "Item_ID"

Liefert eine 1, wenn der aufrufende Actor das angegebene Item ausgerüstet hat. Dies funktioniert nur für Waffen, Rüstungen und Kleidungen. Verbrauchsgegenstände können so nicht abgefragt werden.

"Player"->HasItemEquipped, "daedric culb"

Verwandte Funktionen: *Equip*, *GetArmorType*, *GetWeaponType*

3.109 HasSoulGem

```
"Actor_ID"->HasSoulGem, "Creature_ID"
```

Die Funktion liefert die Zahl der Seelensteine mit der angegebenen Seele im Inventar des aufrufenden Actors zurück.

```
short soulcount  
set soulcount to "Player"->HasSoulGem, "cliff racer"
```

Verwandte Funktionen: *AddSoulGem*, *DropSoulGem*, *RemoveSoulGem*

3.110 Help

Help

(**Konsole**) Zeigt eine Liste der reinen Konsolenfunktionen und ihrer Abkürzungen in der Konsole an.

Verwandte Funktionen: -

3.111 HitAttemptOnMe

```
"Actor_ID"->HitAttemptOnMe, "Weapon_ID"
```

Liefert für einen Frame eine 1, wenn der aufrufende Actor von irgendjemand anderem versucht wird mit der angegebenen Waffe getroffen zu werden. Es ist egal, ob der Angriff trifft, geblockt wird oder ausgewichen wird (aber nur durch Geschicklichkeit, nicht durch tatsächliche Bewegung), der Versuch zählt.

```
"heart_akulakhan"->HitAttemptOnMe, "Keening"
```

Verwandte Funktionen: *HitOnMe*

3.112 HitOnMe

```
"Actor_ID"->HitOnMe, "Weapon_ID"
```

Liefert für einen Frame eine 1, wenn der aufrufende Actor von der angegebenen Waffe getroffen wird.

```
"heart_akulakhan"->HitOnMe, "Keening"
```

Verwandte Funktionen: *HitAttemptOnMe*

3.113 HurtCollidingActor

"Object_ID"->HurtCollidingActor, damage

Jeder Actor, der das aufrufende Objekt berührt, nimmt Schaden. Der Schaden pro Sekunde wird in *damage* angegeben.

"act_sotha_tenderizer"->HurtCollidingActor, 9999999

Verwandte Funktionen: *GetCollidingActor*, *GetCollidingPC*, *HurtStandingActor*

3.114 HurtStandingActor

"Object_ID"->HurtStandingActor, damage

Jeder Actor, der auf dem aufrufenden Objekt steht, nimmt Schaden. Der Schaden pro Sekunde wird in *damage* angegeben.

"in_lava_1024"->HurtStandingActor, 20

Verwandte Funktionen: *GetStandingActor*, *GetStandingPC*, *HurtCollidingActor*

3.115 IsWerewolf

"NPC_ID"->IsWerewolf

Liefert eine 1, wenn sich der aufrufende Actor im Moment in der Werwolfgestalt befindet.

Verwandte Funktionen: *BecomeWerewolf*, *SetWerewolfAcrobatics*, *UndoWerewolf*

3.116 Journal

Journal, "JournalTopic_ID", index

Fügt dem Tagebuch des Spielers den Tagebucheintrag mit dem entsprechenden Index aus dem entsprechenden Journalthema hinzu.

Journal, "C3_DestroyDagoth", 50

Verwandte Funktionen: *GetJournalIndex*, *SetJournalIndex*

3.117 Lock

"Object_ID"->Lock, level

Verschließt das aufrufende Objekt mit einem Schloss den angegebenen Levels. Als Objekt können nur Türen und Container angegeben werden. Als Level kann jede ganze Zahl von 0 bis 100 angegeben werden. Wird ein Level von 0 angegeben, ist die Tür im Spiel verschlossen und reagiert entsprechend. Sie wird aber nicht als mit einem Schloss belegt angezeigt und kann somit nicht mit einem Dietrich geöffnet werden. Dies ist also perfekt für Türen, die nur mit einem Schlüssel zu öffnen sein sollen.

Nicht teleportierende Türen, die in ihrer geöffneten Stellung stehen, wenn die Funktion aufgerufen wird, schließen sich automatisch. Dies wird aber nicht korrekt verarbeitet. Beim erneuten Öffnen der Tür, dreht sich diese in die andere Richtung.

"CharGen_cabindoor"->Lock, 0

Verwandte Funktionen: *GetLocked*, *Unlock*

3.118 LoopGroup

"Object_ID"->LoopGroup, Groupname, count[, flag]

Spielt die angegebene Animationsgruppe auf dem aufrufenden Objekt in einer Schleife so oft, wie in *count* angegeben. Das aufrufende Objekt muss die angegebene Animationsgruppe besitzen. Der optionale Parameter *flag* gibt an, wie die Animation gestartet wird.

1 bedeutet, dass die aktuell laufende Animation zuerst beendet wird, bevor die Schleife gestartet wird. Dies ist die Standardeinstellung.

2 bedeutet, dass die aktuell laufende Animation abgebrochen wird und die neue Animation sofort startet.

3 bedeutet, dass die aktuell laufende Animation abgebrochen wird und die neue Animation sofort am Anfang der Schleife gestartet wird. Der Unterschied zu 2 ist nicht bekannt.

"furn_banner_councilclub"->LoopGroup, Idle3, 5

Verwandte Funktionen: *PlayGroup*, *SkipAnim*

3.119 LowerRank

"NPC_ID"->LowerRank

Der aufrufende NPC (nicht der Spieler) wird in seiner Fraktion um einen Rang degradiert.

Verwandte Funktionen: *RaiseRank*

3.120 MenuMode

MenuMode

Liefert eine 1, wenn der Spieler irgendein Menu geöffnet hat, das schließt unter anderem Dialoge, Inventar (Rechtsklick oder Container), Messageboxen mit Buttons, das Spielmenu (ESC) und die Konsole ein. Dies ist eine der wichtigsten Funktionen von Morrowind Script, da fast jedes Script entweder nur laufen soll, wenn kein Menu offen ist (Bewegungsscripts) oder nur, wenn ein Menu offen ist (Inventar-scripts). Deswegen werden viele Scripte eine solche Konstruktion direkt nach der Variablendeklaration besitzen:

```

if ( MenuMode == 1 )
    return
endif

```

Verwandte Funktionen: *MenuTest*, *ToggleMenus*

3.121 MenuTest

MenuTest[, flag]

Diese Funktion kann dazu verwendet werden, bestimmte Menus zu öffnen und zu schließen. Es lassen sich nur Effekte feststellen, wenn *MenuMode* 1 ist. Wird kein Flag angegeben schließt die Funktion das geöffnete Inventar. Die betroffenen Menus werden bei Ausführung der Funktion auch aktualisiert. Als optionale Flags können angegeben werden:

Parameter	Wirkung
0	Gleiche Wirkung wie ohne Parameter
1	Öffnet eine Dialogbox, die nur einen OK-Button und die Nachricht <i>PUT IN SOME TEXT</i> enthält
3	Bei geschlossenem Inventar (sonstiges Menu offen): Öffnet und maximiert Statusmenu in der Bildmitte (Wirkung wie beim Klicken des Buttons in der rechten oberen Ecke des Menus) Bei geöffnetem Inventar: Holt das Statusmenu in den Vordergrund (nur bemerkbar, wenn sich die Fenster überlappen)
4	Wie bei 3, nur mit Inventarmenu
5	Wie bei 3, nur mit Magiemenu
6	Wie bei 3, nur mit Kartenmenu

Verwandte Funktionen: -

3.122 MessageBox

MessageBox, "Anzuzeigender Text"[, var1, var2, ...][, "Antwort 1", ..., "Antwort 9"]

Blendet eine Nachricht ein, die zusätzlich bis zu 9 Buttons erzeugen kann (Antwort 1 – Antwort 9). Wenn Antwortmöglichkeiten gegeben sind, schließt sich das Fenster erst nach klicken auf eine der Antworten. Welche Antwort geklickt wurde lässt sich mit *GetButtonPressed* ausgeben. Ist keine Antwortmöglichkeit gegeben, bleibt das Fenster abhängig von der Zeichenanzahl der Nachricht eingeblendet. Die Dauer pro Zeichen wird im GameSetting *fMessageTimePerChar* festgelegt und ist standardmäßig auf 0,1 Sekunden gestellt.

Es ist auch möglich Werte von Variablen im Text anzuzeigen. Im Text muss dafür ein Platzhalter verwendet werden: *%f* oder *%.yf*. In der ersten Version wird jede Zahl mit 6 Nachkommastellen angegeben. In der zweiten Version wird die Zahl mit der in *y* festgelegten Anzahl von Nachkommastellen ausgegeben. Somit sollte für Variablen vom Typ *short* oder *long* der Platzhalter *%.0f* verwendet werden. In Messageboxen lassen sich auch einige Stringkonstanten ausgeben. Diese müssen mit einem ^ (Zirkumflex) angeführt werden. Eine Liste aller Stringkonstanten kann im Anhang gefunden werden.

Wenn eine MessageBox während eines Dialoges aufgerufen wird, wird der Inhalt der MessageBox als weiße Schrift in einen neuen Absatz geschrieben. Dies funktioniert allerdings nur mit Messageboxen ohne Antworten.

```

MessageBox, "Ganz einfache Messagebox."
---
MessageBox, "Der Name des Spielers ist ^PCName."
---
long gold
set gold to "Player"->GetItemCount, "Gold_001"
MessageBox, "Ihr besitzt %.0f Draken.", gold
---
MessageBox, "Es ist %.2f Uhr am %.0f. %.0f. im Jahr 3Ä %.0f.", GameHour, Day, Month, Year
---
MessageBox, "Diese Nachricht verschwindet erst, wenn Ihr OK anklickt.", "OK"
---
MessageBox, "Es ist schon nach %.0f Uhr. Ihr solltet schlafen gehen.", GameHour, "OK"

```

Verwandte Funktionen: *Choice*, *GetButtonPressed*

3.123 Mod

"Object_ID"->Mod..., value

Mod ist eine der vielseitigsten Funktion in Morrowind Script. Genau genommen sind es sehr viele einzelne Funktionen, der Übersichtlichkeit halber fasse ich sie aber in diesem Abschnitt zusammen. Diese Funktionen haben immer gemeinsam, dass sie den aktuellen Wert einer bestimmten Eigenschaft um den in *value* angegebenen Wert ändern. Diese Funktionen akzeptieren bedauerlicherweise keine Variablen. Weiterhin lassen sich die *Mod*-Funktionen danach unterscheiden, von welchen Objekt sie aufgerufen werden können. Im Folgenden wird eine Tabelle aller *Mod*-Funktionen mit gültigen aufrufenden Objekten und gegebenenfalls einer Erklärung gegeben. *Mod*-Funktionen, die zusätzliche Parameter oder Bedingungen benötigen, werden später separat behandelt.

Mod-Funktionen akzeptieren quasi alle übergebenen Werte. Viele Eigenschaften haben aber bestimmte Ober- und Untergrenzen, die mit einer *Mod*-Funktion nicht überschritten werden können. So kann zum Beispiel mit dem *ModCurrentHealth* nicht mehr als das maximal verfügbare Leben erreicht werden.

Änderungen an Zaubereffekten werden nicht unter den aktiven Zaubern angezeigt, sind aber trotzdem wirksam. Alle Effekte, die entweder an oder aus sind (zum Beispiel Festes Wasser) können auf beliebige ganzzahlige Werte geändert werden. Sie sind immer aktiv, wenn der Wert nicht 0 ist. Werden sie durch einen Zaubereffekt aktiviert, wird der Wert auf 1 gesetzt.

Name	aufrufbar von	Erklärung
ModFactionReaction	-	siehe unten
ModPCCrimeLevel	-	Kopfgeld des Spielers
ModPCFacRep	-	siehe unten
ModPCVisionBonus	-	Summierte Stärke aller Infravisionseffekte auf dem Spieler / 100
ModRegion	-	siehe unten
ModWaterLevel	-	(Tribunal) Höhe des Wasserstandes, Absturz wenn Interior kein Wasser besitzt, keine Wirkung im Exterior

ModStrength	Actor	aktuelle Attributwerte (einschließlich aller Modifikationen durch Zaubereffekte)
ModIntelligence	Actor	
ModWillpower	Actor	
ModAgility	Actor	
ModSpeed	Actor	
ModEndurance	Actor	
ModPersonality	Actor	
ModLuck	Actor	
ModAcrobatics	NPC + PC	aktuelle Fertigkeitswerte (einschließlich aller Modifikationen durch Zaubereffekte)
ModAlchemy	NPC + PC	
ModAlteration	NPC + PC	
ModArmorer	NPC + PC	
ModAthletics	NPC + PC	
ModAxe	NPC + PC	
ModBlock	NPC + PC	
ModBluntWeapon	NPC + PC	
ModConjuration	NPC + PC	
ModDestruction	NPC + PC	
ModEnchant	NPC + PC	
ModHandToHand	NPC + PC	
ModHeavyArmor	NPC + PC	
ModIllusion	NPC + PC	
ModLightArmor	NPC + PC	
ModLongBlade	NPC + PC	
ModMarksman	NPC + PC	
ModMediumArmor	NPC + PC	
ModMercantile	NPC + PC	
ModMysticism	NPC + PC	
ModRestoration	NPC + PC	
ModSecurity	NPC + PC	
ModShortBlade	NPC + PC	
ModSneak	NPC + PC	
ModSpear	NPC + PC	
ModSpeechcraft	NPC + PC	
ModUnarmored	NPC + PC	
ModHealth	Actor	Ändert den Maximalwert
ModMagicka	Actor	Ändert den Maximalwert
ModFatigue	Actor	Ändert den Maximalwert
ModCurrentHealth	Actor	Ändert den aktuellen Wert
ModCurrentMagicka	Actor	Ändert den aktuellen Wert
ModCurrentFatigue	Actor	Ändert den aktuellen Wert
ModDisposition	NPC	Stimmung des NPCs gegenüber dem Spieler
ModFight	Actor (außer PC)	AI-Wert
ModFlee	Actor (außer PC)	AI-Wert
ModAlarm	Actor (außer PC)	AI-Wert
ModHello	Actor (außer PC)	AI-Wert
ModReputation	NPC + PC	Reputation des Actors

		Summierte Stärke aller Zaubereffekte des angegebenen Typs auf dem Actor
ModArmorBonus	Actor	Schild
ModAttackBonus	Actor	Angriff festigen
ModBlindness	Actor	
ModCastPenalty	Actor	Lärm
ModChameleon	Actor	
ModDefendBonus	Actor	Zuflucht
ModFlying	Actor	Levitation
ModInvisible	Actor	
ModParalysis	Actor	
ModResistBlight	Actor	
ModResistCorprus	Actor	
ModResistDisease	Actor	
ModResistFire	Actor	
ModResistFrost	Actor	
ModResistMagicka	Actor	
ModResistNormalWeapons	Actor	
ModResistParalysis	Actor	
ModResistPoison	Actor	
ModResistShock	Actor	
ModSilence	Actor	
ModSuperJump	Actor	Sprung
ModSwimSpeed	Actor	Flossen
ModWaterBreathing	Actor	
ModWaterWalking	Actor	
ModScale	Object	(Tribunal)

3.124 ModFactionReaction

ModFactionReaction, "Faction1_ID", "Faction2_ID", value

Ändert die Einstellung der ersten Fraktion gegenüber der zweiten um den angegebenen Wert.

```
ModFactionReaction, "Telvanni", "Mages Guild", 2
```

Verwandte Funktionen: *GetFactionReaction*, *SetFactionReaction*

3.125 ModPCFacRep

ModPCFacRep, "Faction_ID", value

Ändert den Ruf des Spielers bei der angegebenen Fraktion um den angegebenen Wert.

```
ModPCFacRep, "Redoran", 10
```

Verwandte Funktionen: *GetPCFacRep*, *SetPCFacRep*

3.126 ModRegion

`ModRegion, "Region_ID", clear_var, cloudy_var, foggy_var, overcast_var, rain_var, thunder_var, ash_var, blight_var[, snow_var, blizzard_var]`

Achtung: Diese Funktion unterscheidet sich abhängig davon, ob Bloodmoon installiert ist, oder nicht. Die beiden optionalen Parameter sind nur mit Bloodmoon verfügbar. Werden sie nicht angegeben werden sie als 0 angenommen.

Diese Funktion ändert die Wetterwahrscheinlichkeiten in der angegebenen Region. Die 7 (oder 9) Variablen geben jeweils die Wahrscheinlichkeit das entsprechende Wetter zu erhalten in Prozent an. **Wichtig:** Alle Variablen müssen sich zu 100 Prozent addieren!

Im Gegensatz zu *ChangeWeather* sind die Änderungen dieser Funktion dauerhaft. Alle 20 Stunden Spielzeit wird ein neues Wetter nach der mit diesem Aufruf vorgegebenen Wahrscheinlichkeitsverteilung ausgewürfelt.

Die Variablen beeinflussen folgende Wettertypen:

Variable	Wetter
clear_var	Sonnenschein
cloudy_var	Leicht bewölkt
foggy_var	Nebelig
overcast_var	Stark bewölkt
rain_var	Regen
thunder_var	Gewitter
ash_var	Aschesturm
blight_var	Peststurm
	(Typen nur mit Bloodmoon verfügbar)
snow_var	Schnee
blizzard_var	Blizzard

Die Regions-IDs können im Construction Set unter World→Regions gefunden werden. Die Standardregionen sind (da teilweise nicht konsistent durchbenannt):

Region	Region-ID
Ascadia-Inseln	Ascadian Isles Region
Aschländer-Region	Ashlands Region
Azuraküste	Azura's Coast Region
Bitterküste	Bitter Coast Region
Molag Amur	Molag Mar Region
Roter Berg	Red Mountain Region
Sheogorad	Sheogorad
Weidenländer	Grazelands Region
Westspalte	West Gash Region
Gramfeste	Mournhold Region
Brodir Hain	Brodir Grove Region
Felsaad-Küste	Felsaad Coast Region
Hirstaang-Wald	Hirstaang Forest Region
Isinfier-Ebenen	Isinfier Plains Region
Moesring-Berge	Moesring Mountains Region
Thirsk	Thirsk Region

Das Beispiel ändert die Wetterwahrscheinlichkeiten an der Bitterküste zu 50% Nebel und je 25% Sonne und Regen:

```
ModRegion, "Bitter Coast Region", 25, 0, 50, 0, 25, 0, 0, 0
```

Verwandte Funktionen: *ChangeWeather*, *GetCurrentWeather*, *GetWindSpeed*

3.127 Move

"Object_ID"->Move, axis, value

Bewegt das aufrufende Objekt mit der in *value* angegebenen Geschwindigkeit bezüglich der **relativen** Achse. Die relative Achse ist eine andere Achse als die feste Achse der Welt (zum Beispiel Z für Oben). Je nachdem, ob ein Objekt gedreht ist, unterscheiden sich die relative und die absolute Achse. Diese Funktion kann zum Beispiel dafür benutzt werden, dass sich ein Pfeil immer in die Richtung seiner Spitze bewegt, egal in welche Richtung er gedreht ist.

Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden. Die Werte werden als Editoreinheiten pro Sekunde interpretiert. Tatsächlich wird bei jedem Aufruf nur der entsprechende Bruchteil der Bewegung ausgeführt, die der Zeit seit dem letzten Frame entspricht. Um eine kontinuierliche Bewegung zu erreichen, muss die Funktion also jeden Frame aufgerufen werden. *value* akzeptiert (leider) keine Variablen!

Die Funktion hat keine Auswirkungen auf NPCs, Kreaturen oder den Spieler, für diese ist *SetPos* zu verwenden.

```
while ( "de_p_desk_01_galbedir"->GetPos, X < 1000 )
    "de_p_desk_01_galbedir"->Move, X, 100
endwhile
```

Verwandte Funktionen: *GetPos*, *MoveWorld*, *SetPos*

3.128 MoveOneToOne

MoveOneToOne

MOTO

Nach Aufruf dieser Funktion wird die Rennanimation mit der gleichen Geschwindigkeit abgespielt, wie die Gehgeschwindigkeit. Dies hat zur Folge, dass man sich in der Third Person-Ansicht nur noch mit Gehgeschwindigkeit bewegen kann. Diese Funktion beeinflusst die First Person-Ansicht nicht. Zweite Eingabe der Funktion hebt die Wirkung auf.

Verwandte Funktionen: -

3.129 MoveWorld

"Object_ID"->MoveWorld, axis, value

Bewegt das aufrufende Objekt mit der in *value* angegebenen Geschwindigkeit bezüglich der **absoluten** Achse. Die absolute Achse ist die feste Achse der Welt (zum Beispiel Z für Oben). Je nachdem, ob ein Objekt gedreht ist, unterscheiden sich die relative und die absolute Achse. Diese Funktion kann zum Beispiel dafür benutzt werden, dass sich ein Objekt immer nach oben bewegt, egal in welche Richtung es gedreht ist.

Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden. Die Werte werden als Editoreinheiten pro Sekunde interpretiert. Tatsächlich wird bei jedem Aufruf nur der entsprechende Bruchteil der Bewegung ausgeführt, die der Zeit seit dem letzten Frame entspricht. Um eine kontinuierliche Bewegung zu erreichen, muss die Funktion also jeden Frame aufgerufen werden. *value* akzeptiert (leider) keine Variablen!

Die Funktion hat keine Auswirkungen auf NPCs, Kreaturen oder den Spieler, für diese ist *SetPos* zu verwenden.

```
while ( "de_p_desk_01_galbedir"->GetPos, X < 1000 )
    "de_p_desk_01_galbedir"->MoveWorld, X, 100
endwhile
```

Verwandte Funktionen: *GetPos*, *Move*, *SetPos*

3.130 ObjectReferenceInfo

ObjectReferenceInfo
ORI

(Konsole) Zeigt Informationen des ausgewählten Objekts, wie seine ID, seine Zelle und die esm/esp-Datei, die es ins Spiel eingefügt hat, in der Konsole an.

Verwandte Funktionen: -

3.131 OnActivate

"Object_ID"->OnActivate

Liefert für einen Frame eine 1, wenn versucht wird das aufrufende Objekt zu aktivieren. Die Aktivierung eines Objekts wird abgebrochen, wenn ein *OnActivate* aufgerufen wird! Deswegen ist es nötig, eine Aktivierung des Objekts, wenn gewünscht, im folgenden Scriptblock manuell mit der *Activate*-Funktion auszulösen. Ein Activator kann nur dann ein *OnActivate* abfangen, wenn er einen Namen hat.

```
if ( OnActivate == 1 )
    PlaySound, "LockedDoor"
endif

---

if ( OnActivate == 1 )
    Journal, "C3_DestroyDagoth", 50
    Activate
endif
```

Verwandte Funktionen: *Activate*

3.132 OnDeath

3.133 OnKnockout

3.134 OnMurder

"Actor_ID"->OnDeath
"Actor_ID"->OnKnockout
"Actor_ID"->OnMurder

Liefert für einen Frame eine 1, wenn der aufrufende Actor stirbt, zu Boden geschlagen (Treffer, wenn Ziel keine Ausdauer mehr hat) oder ermordet (Angreifer gewinnt) wird.

Verwandte Funktionen: *GetDeadCount*

3.135 OutputObjCounts

OutputObjCounts

(**Konsole**) Zeigt eine Liste mit der Anzahl der Objekte in den einzelnen Objektklassen in Morrowind in der Konsole.

Verwandte Funktionen: *OutputRefCounts*

3.136 OutputRefCounts

OutputRefCounts

(**Konsole**) Zeigt eine Liste mit den verschiedenen Referenzen in den einzelnen Objektkategorien in Morrowind, wobei auch solche Informationen, wie zum Beispiel die Anzahl der verschlossenen Türen angezeigt wird.

Verwandte Funktionen: *OutputObjCounts*

3.137 PayFine

PayFine

Veranlasst die KI die Verfolgung des Spielers wegen Verbrechen einzustellen. Gleichzeitig werden alle gestohlenen Gegenstände aus dem Inventar entfernt und in die nächste Truhe mit Beweisstücken transportiert. Das Kopfgeld muss per Hand auf 0 gesetzt werden (mit der Funktion *SetPCCrimeLevel, 0*).

Verwandte Funktionen: *GetPCCrimeLevel, GoToJail, ModPCCrimeLevel, PayFineThief, SetPCCrimeLevel*

3.138 PayFineThief

PayFineThief

Veranlasst die KI die Verfolgung des Spielers wegen Verbrechen einzustellen, allerdings behält der Spieler seine gestohlenen Gegenstände. Das Kopfgeld muss per Hand auf 0 gesetzt werden (mit der Funktion *SetPCCrimeLevel, 0*).

Verwandte Funktionen: *GetPCCrimeLevel, GoToJail, ModPCCrimeLevel, PayFine, SetPCCrimeLevel*

3.139 PCClearExpell

PCClearExpell, "Fraction_ID"

Hebt den Ausschluss des Spielers aus der angegebenen Fraktion wieder auf.

Verwandte Funktionen: *PCExpell, PCExpelled*

3.140 PCExpell

PCExpell, "Faction_ID"

Schließt den Spieler aus der angegebenen Fraktion aus.

Verwandte Funktionen: *PCClearExpell*, *PCExpelled*

3.141 PCExpelled

PCClearExpelled, "Faction_ID"

Liefert eine 1, wenn der Spieler aus der angegebenen Funktion ausgeschlossen ist.

```
if ( PCExpelled, "Telvanni" == 1 )
    PCClearExpell, "Telvanni"
endif
```

Verwandte Funktionen: *PCClearExpell*, *PCExpell*

3.142 PCForce1stPerson

3.143 PCForce3rdPerson

PCForce1stPerson
PCForce3rdPerson

Die Ansicht des Spielers wechselt automatisch in die First beziehungsweise Third Person-Ansicht. Der Spieler kann aber jederzeit die Ansicht wieder wechseln, wenn dies nicht mit *DisablePlayerViewSwitch* verhindert wird.

Verwandte Funktionen: *DisablePlayerViewSwitch*, *EnablePlayerViewSwitch*, *GetPlayerViewSwitchDisabled*, *PCGet3rdPerson*

3.144 PCGet3rdPerson

PCGet3rdPerson

Liefert eine 0, wenn sich der Spieler in der First Person-Ansicht befindet. Sonst, also in der Third Person-Ansicht, im Vanity Mode und im Umschauen-Modus, liefert die Funktion eine 1.

Verwandte Funktionen: *DisablePlayerViewSwitch*, *EnablePlayerViewSwitch*, *GetPlayerViewSwitchDisabled*, *PCForce1stPerson*, *PCForce3rdPerson*

3.145 PCJoinFaction

PCJoinFaction, "Faction_ID"

Nimmt den Spieler in die angegebene Fraktion auf. Er nimmt automatisch den untersten Rang ein.

Verwandte Funktionen: *GetPCRank*, *PCLowerRank*, *PCRaiseRank*

3.146 PCLowerRank

3.147 PCRaiseRank

```
PCLowerRank, "Faction_ID"
PCRaiseRank, "Faction_ID"
```

Degradiert beziehungsweise befördert den Spieler in der angegebenen Fraktion um einen Rang.

Achtung: Ist der Spieler kein Mitglied der Fraktion, wird er durch diesen Funktionsaufruf zum Mitglied im untersten Rang gemacht.

Verwandte Funktionen: *GetPCRank*, *PCJoinFaction*

3.148 PlaceAtMe

```
"Object_ID"->PlaceAtMe, "Object2_ID", count, distance, position
```

(**Bloodmoon**) Platziert die angegebene Anzahl des angegebenen Objekts in der angegebenen Entfernung zum aufrufenden Objekt. Der Parameter *position* gibt an, wo das Objekt in Bezug zum aufrufenden Objekt erscheinen soll. 0 bedeutet vor dem aufrufenden Objekt, 1 hinter diesem und 2 und 3 stehen für die linke und rechte Seite. Das Beispiel erschafft einen Hunger 128 Editoreinheiten hinter Caius Cosades:

```
"caius cosades"->PlaceAtMe, "hunger", 1, 128, 1
```

Verwandte Funktionen: *PlaceAtPC*, *PlaceItem*, *PlaceItemCell*

3.149 PlaceAtPC

```
PlaceAtPC, "Object2_ID", count, distance, position
```

Platziert die angegebene Anzahl des angegebenen Objekts in der angegebenen Entfernung zum Spieler. Der Parameter *position* gibt an, wo das Objekt in Bezug zum Spieler erscheinen soll. 0 bedeutet vor dem Spieler, 1 hinter diesem und 2 und 3 stehen für die linke und rechte Seite. Das Beispiel erschafft einen Hunger 128 Editoreinheiten hinter dem Spieler:

```
PlaceAtPC, "hunger", 1, 128, 1
```

Verwandte Funktionen: *PlaceAtMe*, *PlaceItem*, *PlaceItemCell*

3.150 PlaceItem

```
PlaceItem, "Object_ID", xpos, ypos, zpos, zrot
```

Erschafft eine neue Referenz des angegebenen Objekts an den angegebenen Koordinaten und mit der angegebenen Rotation um die Z-Achse im aktuellen Bezugsrahmen. In einer Interiorzelle also in dieser Zelle beziehungsweise irgendwo im Exterior.

Hinweis: Diese Funktion akzeptiert lokale Variablen!

```
PlaceItem, "hunger", 123, 456, 789, 180
```

Verwandte Funktionen: *PlaceAtMe*, *PlaceAtPC*, *PlaceItemCell*

3.151 PlaceItemCell

PlaceItemCell, "Object_ID", "Zellenname", xpos, ypos, zpos, zrot

Erschafft eine neue Referenz des angegebenen Objekts an den angegebenen Koordinaten und mit der angegebenen Rotation um die Z-Achse in der angegebenen Zelle. Um ein Objekt in einer nicht benannten Exteriorzelle zu platzieren, kann der Name einer beliebigen benannten Exteriorzelle angegeben werden. In diesem Fall reichen der Funktion die Koordinaten den Unterschied zwischen der angegebenen Zelle und den Koordinaten zu finden und setzt den Gegenstand automatisch in der den Koordinaten entsprechenden Zelle ab. Das Beispiel erschafft einen Hunger vor dem Eingang von Dagoth Ur.

Hinweis: Diese Funktion akzeptiert lokale Variablen!

```
PlaceItemCell, "hunger", "Balmora", 19525, 70400, 12000, 180
```

Verwandte Funktionen: *PlaceAtMe*, *PlaceAtPC*, *PlaceItemCell*

3.152 PlayBink

PlayBink, "Videodatei.bik", flag

Pausiert das Spiel und startet die Wiedergabe des angegebenen Videos. Das Video muss im Bink-Format vorliegen und im Unterordner des Morrowind-Hauptverzeichnisses *Data Files/Video* gespeichert sein. Der Parameter *flag* gibt an, ob das Video mit ESC abgebrochen werden kann. Ist der Wert 1, ist dies möglich, bei 0 nicht.

Verwandte Funktionen: -

3.153 PlayGroup

"Object_ID"->PlayGroup, Groupname[, flag]

Spielt die angegebene Animationsgruppe auf dem aufrufenden Objekt einmal ab. Das aufrufende Objekt muss die angegebene Animationsgruppe besitzen. Der optionale Parameter *flag* gibt an, wie die Animation gestartet wird.

1 bedeutet, dass die aktuell laufende Animation zuerst beendet wird, bevor die Schleife gestartet wird. Dies ist die Standardeinstellung.

2 bedeutet, dass die aktuell laufende Animation abgebrochen wird und die neue Animation sofort startet.

3 bedeutet, dass die aktuell laufende Animation abgebrochen wird und die neue Animation sofort am Anfang der Schleife gestartet wird. Der Unterschied zu 2 ist nicht bekannt.

```
"furn_banner_councilclub"->PlayGroup, Idle3
```

Verwandte Funktionen: *LoopGroup*, *SkipAnim*

3.154 PlayLoopSound3D**3.155 PlayLoopSound3DVP**

```
"Object_ID"->PlayLoopSound3D, "Sound_ID"
"Object_ID"->PlayLoopSound3DVP, "Sound_ID", volume, pitch
```

Das aufrufende Objekt spielt den angegebenen Sound in einer Endlosschleife ab. Der Sound ertönt dabei von der Position des aufrufenden Objekts aus. Die Variante mit Lautstärke *volume* und Tonhöhe *pitch* erfordert eine zusätzliche Angabe dieser Werte. Beide Parameter akzeptieren float-Werte. Die Lautstärke kann zwischen 0.0 und 1.0 angegeben werden, scheint aber keine Auswirkung auf den Sound zu haben. Die Tonhöhe kann zwischen 0 und 2 angegeben werden, wobei 0.0 volle Bässe, 1.0 normalen Ton und 2.0 volle Höhen bedeutet. Um einen LoopSound zu beenden wird die Funktion *StopSound* benötigt.

```
if ( GetSoundPlaying, "lava layer" == 0 )
    PlayLoopSound3DVP, "lava layer", 1.0, 1.0
endif
```

Verwandte Funktionen: *GetSoundPlaying*, *PlaySound*, *StopSound*

3.156 PlaySound**3.157 PlaySound3D****3.158 PlaySoundVP****3.159 PlaySound3DVP**

```
"Object_ID"->PlaySound, "Sound_ID"
"Object_ID"->PlaySound3D, "Sound_ID"
"Object_ID"->PlaySoundVP, "Sound_ID", volume, pitch
"Object_ID"->PlaySound3DVP, "Sound_ID", volume, pitch
```

Das aufrufende Objekt spielt den angegebenen Sound einmal ab. Bei den Varianten ohne 3D wird der Sound am Ort des Spielers abgespielt, sonst ertönt der Sound von der Position des aufrufenden Objekts aus. Die Varianten mit Lautstärke *volume* und Tonhöhe *pitch* erfordern eine zusätzliche Angabe dieser Werte. Beide Parameter akzeptieren float-Werte. Die Lautstärke kann zwischen 0.0 und 1.0 angegeben werden, scheint aber keine Auswirkung auf den Sound zu haben. Die Tonhöhe kann zwischen 0 und 2 angegeben werden, wobei 0.0 volle Bässe, 1.0 normalen Ton und 2.0 volle Höhen bedeutet.

Verwandte Funktionen: *GetSoundPlaying*, *PlayLoopSound3D*, *StopSound*

3.160 Position

```
"Object_ID"->Position, xpos, ypos, zpos, zrot
```

Platziert das aufrufende Objekt mit der angegebenen Rotation um die Z-Achse an der angegebenen Position im Exterior. Diese Funktion akzeptiert lokale float-Variablen. Allerdings nur float-Variablen. Andere Variablentypen werden immer als 0 interpretiert.

Achtung: Der Aufruf dieser Funktion im Dialog sollte vermieden werden, da dies zum Spielabsturz führen kann. Der Aufruf in der Konsole ist verbüggt.

```
Player->Position, 123, 456, 789, 90
```

Verwandte Funktionen: *PositionCell*

3.161 PositionCell

"Object_ID"->PositionCell, xpos, ypos, zpos, zrot, "Zellenname"

Platziert das aufrufende Objekt mit der angegebenen Rotation um die Z-Achse an der angegebenen Position in der angegebenen Zelle. Diese Funktion akzeptiert lokale float-Variablen. Allerdings nur float-Variablen. Andere Variablentypen werden immer als 0 interpretiert. Ist der angegebene Zellenname ungültig, erfolgt der Teleport immer an die (0,0,0)-Position im Exterior. **Achtung:** Der Aufruf dieser Funktion im Dialog sollte vermieden werden, da dies zum Spielabsturz führen kann. Der Aufruf in der Konsole ist verbuggt.

Player->PositionCell, 2.730, 959.274, -105.817, 180, "Gramfeste, Königspalast: Empfangsbereich"

Verwandte Funktionen: *Position*

3.162 PurgeTextures

PurgeTextures

PT

(**Konsole**) Die Funktion lädt alle derzeit geladenen Texturen neu.

Verwandte Funktionen: -

3.163 RaiseRank

"NPC_ID"->RaiseRank

Befördert den aufrufenden NPC in seiner Fraktion um einen Rang.

Verwandte Funktionen: *LowerRank*

3.164 Random

Random, maximum

Liefert eine ganzzahlige Zufallszahl, für die gilt $0 \leq x < \text{maximum}$. Wird also für *maximum* zum Beispiel 100 angegeben liefert die Funktion eine Zahl zwischen (einschließlich) 0 und 99.

Die Zufallszahl wird nur jeden Frame aktualisiert. Es kann mit dieser Funktion also nur eine Zufallszahl pro Frame erzeugt werden.

```
short zufall
set zufall to Random, 2
if ( zufall == 0 )
    MessageBox, "Der Münzwurf ergibt Kopf."
else
    MessageBox, "Der Münzwurf ergibt Zahl."
endif
```

Verwandte Funktionen: -

3.165 RemoveEffects

"Actor_ID"->RemoveEffects, effectnumber

Entfernt alle auf den aufrufenden Actor wirkenden Effekte mit der angegebenen Nummer. Eine Zuordnung der Nummern zu den Effekt-IDs kann man im Anhang finden.

```
if ( "Player"->GetEffect sEffectDamageHealth )
"Player"->RemoveEffects, 23
endif
```

Verwandte Funktionen: *GetEffect*, *GetSpellEffects*, *RemoveSpellEffects*

3.166 RemoveFromLevCreature

RemoveFromLevCreature, "LeveledCreature_ID", "Creatue_ID", pc_level

(Tribunal) Entfernt die angegebene Kreatur mit dem angegebenen Level aus der angegebenen Kreaturenliste. Mit einer zusätzlichen Abhängigkeit von Bloodmoon kann statt einer bestimmten Kreatur auch eine weitere Kreaturenliste angegeben werden.

Warnung: Diese Funktion sollte nur auf selbst erstellte Listen angewendet werden, da die Änderungen im Spielstand gespeichert werden!

```
RemoveFromLevCreature, "bm_ex_hirforest", "cliff racer", 1
```

Verwandte Funktionen: *AddToLevCreature*

3.167 RemoveFromLevItem

RemoveFromLevItem, "LeveledItem_ID", "Item_ID", pc_level

(Tribunal) Entfernt den angegebenen Gegenstand mit dem angegebenen Level aus der angegebenen Gegenstandsliste. Statt eines Gegenstandes kann auch eine weitere Gegenstandsliste angegeben werden.

Warnung: Diese Funktion sollte nur auf selbst erstellte Listen angewendet werden, da die Änderungen im Spielstand gespeichert werden!

```
RemoveFromLevItem, "random_drinks_01", "potion_nord_mead", 5
```

Verwandte Funktionen: *AddToLevItem*

3.168 RemoveItem

"Object_ID"->RemoveItem, "Item_ID", count

Dem aufrufenden Objekt wird die angegebene Anzahl des angegebenen Gegenstands aus dem Inventar entfernt. Der aufrufende Gegenstand muss ein Inventar besitzen und es sollte sicher gestellt sein, dass die nötige Anzahl von Gegenständen sich auch in diesem Inventar befindet, da sonst schwere Fehler auftreten können.

```
if ( "Player"->GetItemCount, "Gold_001" >= 1000 )
"Player"->RemoveItem, "Gold_001", 1000
endif
```

Verwandte Funktionen: *AddItem*, *Drop*, *GetItemCount*

3.169 RemoveSoulGem

"Object_ID"->RemoveSoulGem, "Creature_ID"

Dem aufrufenden Objekt wird ein Seelenstein mit der angegebenen Seele aus dem Inventar entfernt. Das Objekt muss ein Inventar besitzen und einen solchen Seelenstein im Inventar haben, sonst kann das Spiel abstürzen. Hat das Objekt mehrere Seelensteine mit dieser Art von Seele im Inventar, wird der Seelenstein mit der ältesten Seele (also der zuerst gefangenen) zuerst entfernt.

```
if ( "Player"->HasSoulGem, "cliff racer" >= 1 )
    "Player"->RemoveSoulGem, "cliff racer"
endif
```

Verwandte Funktionen: *AddSoulGem*, *DropSoulGem*, *HasSoulGem*

3.170 RemoveSpell

"Actor_ID"->RemoveSpell, "Spell_ID"

Entfernt den angegebenen Zauber aus dem Magiemenu des aufrufenden Actors. Mit Zauber ist in diesem Fall jedes Objekt aus dem Spellmaking-Reiter gemeint.

```
"Player"->RemoveSpell, "ataxia"
```

Verwandte Funktionen: *AddSpell*, *GetSpell*

3.171 RemoveSpellEffects

"Actor_ID"->RemoveSpellEffects, "Spell_ID"

Entfernt alle Effekte des angegebenen Zaubers vom aufrufenden Actor, wenn dieser Zauber auf ihn wirkt. Dabei ist es egal, wer diesen Zauber gewirkt hat.

```
if ( "Player"->GetSpellEffects, "alit bite" == 1 )
    "Player"->RemoveSpellEffects, "alit bite"
endif
```

Verwandte Funktionen: *GetEffect*, *GetSpellEffects*, *RemoveEffects*

3.172 RepairedOnMe

"Item_ID"->RepairedOnMe, "RepairItem_ID"

Liefert eine 1, wenn die aufrufende Waffe oder Rüstung von dem angegebenen Werkzeug repariert wurde.

```
if ( RepairedOnMe, "repair_journeyman_01" == 1 )
    MessageBox, "Ihr habt mich mit einem Schmiedehammer des Gesellen repariert."
endif
```

Verwandte Funktionen: -

3.173 ResetActors

ResetActors
RA

Die Funktion setzt alle NPCs und Kreaturen in der aktuellen Zelle an ihre Startposition zurück. Eigentlich ist die Funktion für die Konsole gedacht, zum Beispiel wenn NPCs durch Texturen gefallen sind oder ein NPC eine Tür versperrt. Die Funktion ist aber auch aus Scripten heraus aufrufbar und könnte durchaus verschiedene Nutzen erfüllen.

Verwandte Funktionen: *SetAtStart*

3.174 Resurrect

"Actor_ID"->Resurrect

Belebt den aufrufenden Actor nach seinem Tod neu. Dabei werden alle Werte auf die ursprünglich vorgegebenen Werte zurückgesetzt.

Verwandte Funktionen: -

3.175 Rotate

"Object_ID"->Rotate, axis, value

Bewegt das aufrufende Objekt mit der in *value* angegebenen Winkelgeschwindigkeit um die **relative** Achse. Die relative Achse ist eine andere Achse als die feste Achse der Welt (zum Beispiel Z für Oben). Je nachdem, ob ein Objekt gedreht ist, unterscheiden sich die relative und die absolute Achse. Diese Funktion kann zum Beispiel dafür benutzt werden, dass sich ein Pfeil immer um seine Längsachse dreht, egal in welche Richtung er gedreht ist.

Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden. Die Werte werden als Grad pro Sekunde interpretiert. Tatsächlich wird bei jedem Aufruf nur der entsprechende Bruchteil der Bewegung ausgeführt, die der Zeit seit dem letzten Frame entspricht. Um eine kontinuierliche Bewegung zu erreichen, muss die Funktion also jeden Frame aufgerufen werden. *value* akzeptiert (leider) keine Variablen!

Die Funktion hat keine Auswirkungen auf NPCs, Kreaturen oder den Spieler, für diese ist *SetAngle* zu verwenden.

```
while ( "de_p_desk_01_galbedir"->GetAngle, X < 180 )  
    "de_p_desk_01_galbedir"->Rotate, X, 100  
endwhile
```

Verwandte Funktionen: *GetAngle*, *RotateWorld*, *SetAngle*

3.176 RotateWorld

"Object_ID"->RotateWorld, axis, value

Bewegt das aufrufende Objekt mit der in *value* angegebenen Winkelgeschwindigkeit um die **absolute** Achse. Die absolute Achse ist die feste Achse der Welt (zum Beispiel Z für Oben). Je nachdem, ob ein Objekt gedreht ist, unterscheiden sich die relative und die absolute Achse.

Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden. Die Werte werden als Grad pro Sekunde interpretiert.

Tatsächlich wird bei jedem Aufruf nur der entsprechende Bruchteil der Bewegung ausgeführt, die der Zeit seit dem letzten Frame entspricht. Um eine kontinuierliche Bewegung zu erreichen, muss die Funktion also jeden Frame aufgerufen werden. *value* akzeptiert (leider) keine Variablen!

Die Funktion hat keine Auswirkungen auf NPCs, Kreaturen oder den Spieler, für diese ist *SetAngle* zu verwenden.

```
while ( "de_p_desk_01_galbedir"->GetAngle, X < 180 )
    "de_p_desk_01_galbedir"->RotateWorld, X, 100
endwhile
```

Verwandte Funktionen: *GetAngle*, *Rotate*, *SetAngle*

3.177 SameFaction

```
"NPC_ID"->SameFaction
```

Liefert eine 1, wenn der Spieler Mitglied in der gleichen Fraktion wie der aufrufende NPC ist.

Verwandte Funktionen: -

3.178 Say

```
"Object_ID"->Say, "Soundfile.mp3", "Untertitel"
```

Das aufrufende Objekt (Kreatur, NPC, Activator oder der Spieler selbst) spricht die angegebene MP3-Datei oder anders die MP3-Datei wird als Sound von der Position des aufrufenden Objekts abgespielt. Die Datei muss sich im *Data Files/Sound*-Ordner befinden und in einem MPEG-1 Layer-III mit Bitrate von 64 kbps, Samplingfrequenz von 44100 kHz und Mono vorliegen. Andere Einstellungen können funktionieren, müssen aber nicht.

Im Parameter für den Untertitel kann der Text angegeben werden, der eingeblendet wird, wenn der Spieler Untertitel aktiviert hat.

```
Say, "vo\Misc\CharGenWoman1.wav", "Je schneller Ihr geht, desto schneller können wir weiter."
```

Verwandte Funktionen: *PlaySound3D*, *SayDone*

3.179 SayDone

```
"Object_ID"->SayDone
```

Liefert eine 1, wenn das aufrufende Objekt gerade nichts sagt, also keine *Say*-Funktion ausgeführt wird.

```
if ( SayDone == 1 )
    Say, "vo\Misc\CharGenWoman2.wav", "Ja? Seid Ihr verwirrt? Wir warten!"
endif
```

Verwandte Funktionen: *Say*

3.180 ScriptRunning

ScriptRunning, "Script_ID"

Liefert eine 1, wenn das angegebene Script ausgeführt wird.

```
if ( ScriptRunning, "CharGen" == 0 )
    StartScript, "CharGen"
endif
```

Verwandte Funktionen: *StartScript*, *StopScript*

3.181 Set

"Object_ID"->Set..., value

Set ist eine der vielseitigsten Funktion in Morrowind Script. Genau genommen sind es sehr viele einzelne Funktionen, der Übersichtlichkeit halber fasse ich sie aber in diesem Abschnitt zusammen. Diese Funktionen haben immer gemeinsam, dass sie den aktuellen Wert einer bestimmten Eigenschaft auf den in *value* angegebenen Wert gesetzt werden. Diese Funktionen akzeptieren lokale Variablen. Der Variablentyp muss dazu immer zu dem Speicherwert passen. In der Regel sind dies short- oder long-Werte. Funktionen, die float-Werte erwarten sind markiert.

Weiterhin lassen sich die *Set*-Funktionen danach unterscheiden, von welchen Objekt sie aufgerufen werden können. Im Folgenden wird eine Tabelle aller *Set*-Funktionen mit gültigen aufrufenden Objekten und gegebenenfalls einer Erklärung gegeben. *Set*-Funktionen, die zusätzliche Parameter oder Bedingungen benötigen, werden später separat behandelt.

Set-Funktionen akzeptieren quasi alle übergebenen Werte. Jede Eigenschaft hat aber bestimmte Ober- und Untergrenzen, die mit einer *Set*-Funktion nicht überschritten werden können.

Änderungen an Zaubereffekten werden nicht unter den aktiven Zaubern angezeigt, sind aber trotzdem wirksam. Alle Effekte, die entweder an oder aus sind (zum Beispiel Festes Wasser) können auf beliebige ganzzahlige Werte gesetzt werden. Sie sind immer aktiv, wenn der Wert nicht 0 ist. Werden sie durch einen Zaubereffekt aktiviert, wird der Wert auf 1 gesetzt.

Name	aufrufbar von	Erklärung
SetAtStart	Objekt	siehe unten
SetDelete	Objekt	(Tribunal) siehe unten
SetFactionReaction	-	siehe unten
SetJournalIndex	-	siehe unten
SetPCCrimeLevel	-	Kopfgeld des Spielers
SetPCFacRep	-	siehe unten
SetPCVisionBonus	-	Summierte Stärke aller Infravisionseffekte auf dem Spieler / 100, erwartet float-Wert
SetRegion	-	siehe unten
SetWaterLevel	-	(Tribunal) Höhe des Wasserstandes, Absturz wenn Interior kein Wasser besitzt, keine Wirkung im Exterior, erwartet float-Wert
SetWerewolfAcrobatics	-	(Bloodmoon) siehe unten

SetStrength	Actor	aktuelle Attributwerte (einschließlich aller Modifikationen durch Zaubereffekte)
SetIntelligence	Actor	
SetWillpower	Actor	
SetAgility	Actor	
SetSpeed	Actor	
SetEndurance	Actor	
SetPersonality	Actor	
SetLuck	Actor	
SetAcrobatics	NPC + PC	aktuelle Fertigkeitswerte (einschließlich aller Setifikationen durch Zaubereffekte)
SetAlchemy	NPC + PC	
SetAlteration	NPC + PC	
SetArmorer	NPC + PC	
SetAthletics	NPC + PC	
SetAxe	NPC + PC	
SetBlock	NPC + PC	
SetBluntWeapon	NPC + PC	
SetConjuration	NPC + PC	
SetDestruction	NPC + PC	
SetEnchant	NPC + PC	
SetHandToHand	NPC + PC	
SetHeavyArmor	NPC + PC	
SetIllusion	NPC + PC	
SetLightArmor	NPC + PC	
SetLongBlade	NPC + PC	
SetMarksman	NPC + PC	
SetMediumArmor	NPC + PC	
SetMercantile	NPC + PC	
SetMysticism	NPC + PC	
SetRestoration	NPC + PC	
SetSecurity	NPC + PC	
SetShortBlade	NPC + PC	
SetSneak	NPC + PC	
SetSpear	NPC + PC	
SetSpeechcraft	NPC + PC	
SetUnarmored	NPC + PC	
SetHealth	Actor	Ändert den Maximalwert
SetMagicka	Actor	Ändert den Maximalwert
SetFatigue	Actor	Ändert den Maximalwert
SetLevel	Actor	(Tribunal) siehe unten
SetDisposition	NPC	Stimmung des NPCs gegenüber dem Spieler
SetFight	Actor (außer PC)	AI-Wert
SetFlee	Actor (außer PC)	AI-Wert
SetAlarm	Actor (außer PC)	AI-Wert
SetHello	Actor (außer PC)	AI-Wert
SetReputation	NPC + PC	Reputation des Actors

SetArmorBonus	Actor	Summierte Stärke aller Zaubereffekte des angegebenen Typs auf dem Actor
SetAttackBonus	Actor	Schild
SetBlindness	Actor	Angriff festigen
SetCastPenalty	Actor	Lärm
SetChameleon	Actor	
SetDefendBonus	Actor	Zuflucht
SetFlying	Actor	Levitation
SetInvisible	Actor	
SetParalysis	Actor	
SetResistBlight	Actor	
SetResistCorprus	Actor	
SetResistDisease	Actor	
SetResistFire	Actor	
SetResistFrost	Actor	
SetResistMagicka	Actor	
SetResistNormalWeapons	Actor	
SetResistParalysis	Actor	
SetResistPoison	Actor	
SetResistShock	Actor	
SetSilence	Actor	
SetSuperJump	Actor	Sprung
SetSwimSpeed	Actor	Flossen
SetWaterBreathing	Actor	
SetWaterWalking	Actor	
SetAngle	Object	siehe unten
SetPos	Object	siehe unten
SetScale	Object	(Tribunal) erwartet float-Wert

3.182 SetAtStart

"Object_ID" -> SetAtStart

Setzt das aufrufende Objekt an seine Startposition zurück. Auch der Startwinkel wieder hergestellt.

Verwandte Funktionen: *GetStartingAngle*, *GetStartingPos*, *ResetActors*

3.183 SetDelete

SetDelete, 1

(**Tribunal**) Diese Funktion löscht die aufrufende Referenz nicht wiederherstellbar. Dies ist nur möglich, wenn das Objekt vorher Disabled wurde. Weiterhin sollte in dem Frame, in dem diese Funktion aufgerufen wird, keine weitere Scriptfunktion aufgerufen werden.

Objekte mit *reference persistent* zu löschen, kann zu Fehlern führen. Deswegen sollte diese Funktion auch nur im lokalen Script des zu löschenden Objekts aufgerufen werden.

Theoretisch kann der Aufruf im gleichen Frame noch mit *SetDelete*, 0 aufgehoben werden.

```

short doonce
if ( GetDisabled == 1 )
    if ( doonce == 0 )
        set doonce to 1
        SetDelete, 1
    endif
endif
endif

```

Verwandte Funktionen: *Disable*, *GetDisabled*

3.184 SetAngle

"Object_ID"->SetAngle, axis, value

Setzt den aktuellen Winkel des aufrufenden Objekts bezüglich der angegebenen Achse in Grad auf den angegebenen Wert. Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden, nur Variablen müssen vom Typ float sein. Alle Zahlen werden modulo 360 genommen und negative Zahlen danach von 360 abgezogen.

```
"de_p_desk_01_galbedir"->SetAngle, X, 90
```

Verwandte Funktionen: *GetAngle*, *GetPos*, *RotateWorld*, *SetPos*

3.185 SetFactionReaction

SetFactionReaction, "Faction1_ID", "Faction2_ID", value

Setzt die Einstellung der ersten Fraktion gegenüber der zweiten auf den angegebenen Wert.

```
SetFactionReaction, "Telvanni", "Mages Guild", 2
```

Verwandte Funktionen: *GetFactionReaction*, *ModFactionReaction*

3.186 SetLevel

"Actor_ID"->SetLevel, value

(Tribunal) Diese Funktion setzt das Level des aufrufenden Actors auf den angegebenen Wert. Diese Funktion akzeptiert keine Variablen!

Wird diese Funktion auf den Spieler angewendet, wird die Anzeige im Statusmenu nicht aktualisiert. Um das zu bereinigen, müssen folgende Funktionen aufgerufen werden:

```

"Player"->SetLevel, 1
MenuTest, 3
ToggleMenus
ToggleMenus

```

Vielen Dank an Tommy für diese Informationen!

Verwandte Funktionen: *GetLevel*

3.187 SetPCFacRep

SetPCFacRep, "Faction_ID", value

Setzt den Ruf des Spielers bei der angegebenen Fraktion auf den angegebenen Wert.

SetPCFacRep, "Redoran", 10

Verwandte Funktionen: *GetPCFacRep*, *ModPCFacRep*

3.188 SetJournalIndex

SetJournalIndex, "JournalTopic_ID", value

Setzt den Journalindex des angegebenen Themas auf den angegebenen Wert. Diese Funktion fügt allerdings nicht den entsprechenden Tagebucheintrag in das Tagebuch des Spielers ein. Der angegebene Journalindex muss nicht existieren, somit kann man dies als eine Art globale Variable benutzen, da man diesen gesetzten Wert trotzdem mit *GetJournalIndex* abfragen kann. Dabei ist aber zu beachten, dass beim Laden der Wert automatisch auf den höchsten tatsächlichen Tagebucheintrag zu diesem Thema gesetzt wird.

Verwandte Funktionen: *GetJournalIndex*, *Journal*

3.189 SetPos

"Object_ID"->SetPos, axis, value

Setzt die aktuelle Position des aufrufenden Objekts bezüglich der angegebenen Achse auf den angegebenen Wert. Für *axis* werden die Buchstaben X, Y und Z akzeptiert. *value* verarbeitet float-Werte, es können aber auch beliebige andere Werte angegeben werden, nur Variablen müssen vom Typ float sein.

"de_p_desk_01_galbedir"->SetPos, X

Verwandte Funktionen: *GetAngle*, *GetPos*, *MoveWorld*, *SetAngle*

3.190 SetWereWolfAcrobatics

"Actor"->SetWereWolfAcrobatics

Setzt den Akrobatikwert des aufrufenden Actors auf den im Gamesetting *fWereWolfAcrobatics* (unter Gameplay *rightarrow* Settings *rightarrow* Gameplay) angegebenen Wert. Dieser Wert wird als verändert markiert. Wird diese Funktion aufgerufen, während der Actor sich in Werwolf-Form befindet, wird der Wert bei der Rückverwandlung automatisch wieder auf den ursprünglichen Wert gesetzt. Wird die Funktion jedoch in der Normalform aufgerufen, ist die Veränderung dauerhaft und irreversibel (außer durch einen manuellen *Mod*- beziehungsweise *Set*-Aufruf).

Warum diese Funktion existiert und der Akrobatikwert nicht wie die anderen 39 Werwolfwerte automatisch gesetzt wird, ist nicht bekannt.

Verwandte Funktionen: *BecomeWerewolf*, *IsWerewolf*, *UndoWerewolf*

3.191 Show

Show, globalvar

(**Konsole**) Zeigt den Wert der angegebenen globalen Variablen in der Konsole an. Wenn in der Konsole ein Objekt ausgewählt ist, können auch lokale Variablen angezeigt werden.

Show, GameHour

Verwandte Funktionen: *ShowVars*

3.192 ShowAnim

ShowAnim
sa

(**Konsole**) Zeigt einige Informationen über den gerade in der Konsole ausgewählten NPC. Darunter die laufenden Animationen, ob er eine Waffe gezogen hat oder einen Zauber in Bereitschaft hat, wie oft er bereits angegriffen wurde, mit welcher Geschwindigkeit er sich bewegt, usw.

Verwandte Funktionen: -

3.193 ShowGroup

ShowGroup
sg

(**Konsole**) Zeigt alle Mitglieder der Gruppe des in der Konsole ausgewählten Actors. Zu seiner Gruppe gehören alle Personen oder Kreaturen, die ihm folgen oder denen er selbst folgt.

Verwandte Funktionen: -

3.194 ShowMap

ShowMap, "Zellenname"

Zeigt alle Exteriorzellen mit dem angegebenen Zellennamen auf der Weltkarte des Spielers als gelbe Quadrate an.

ShowMap, "Balmora"

Verwandte Funktionen: -

3.195 ShowRestMenu

ShowRestMenu

Öffnet das Rasten-Menü.

Verwandte Funktionen: *GetPCSleep*

3.196 ShowSceneGraph

ShowSceneGraph

(**Konsole**) Minimiert Morrowind in die Taskleite und öffnet ein Fenster mit Informationen über das Rendering der aktuellen Ansicht.

Verwandte Funktionen: -

3.197 ShowTargets

ShowTargets
st

(**Konsole**) Zeigt alle Gegner des in der Konsole ausgewählten Actors an. Dies sind normalerweise sein aktuelles Ziel und alle Personen oder Kreaturen in seiner Gruppe.

Verwandte Funktionen: *GetTarget*

3.198 ShowVars

ShowVars
sv

(**Konsole**) Zeigt die Werte der globalen Variablen, der lokalen Variablen in laufenden globalen Scripten und der lokalen Variablen im Script des ausgewählten Objekts.

Verwandte Funktionen: *Show*

3.199 SkipAnim

SkipAnim

Alle Animationen des aufrufenden Objekts werden für diesen Frame pausiert. Um ein Objekt tatsächlich einfrieren zu lassen, muss diese Funktion also jeden Frame aufgerufen werden.

Verwandte Funktionen: *LoopGroup*, *PlayGroup*

3.200 StartCombat

"Actor_ID"->StartCombat, "Actor2_ID"

Der aufrufende Actor beginnt mit einem Angriff auf den zweiten, angegebenen Actor. Hat der aufrufende Actor einen zu geringen Fight-Wert wird er einen Angriff auf den Spieler von sich aus beenden.

Verwandte Funktionen: *GetTarget*, *StopCombat*

3.201 StartScript

```
["Object_ID"->]StartScript, "Script_ID"
```

Diese Funktion startet ein globales Script. Wird ein aufrufendes Objekt angegeben oder die Funktion von einem lokalen Script oder einer Dialogresultbox aufgerufen, läuft dieses Script als so genanntes *targeted Script*. Das sind globale Scripte, die wie ein lokales Script eine Aufruf-ID auf jeden Funktionsaufruf übergeben. Diese ID ist gleich der ID, des Objekts, auf die dieses Script zielt.

Verwandte Funktionen: *ScriptRunning*, *StopScript*

3.202 StopCellTest

```
StopCellTest  
SCT
```

(**Konsole**) Stoppt einen aktiven Zellentest.

Verwandte Funktionen: *TestCells*, *TestInteriorCells*, *TestThreadCells*

3.203 StopCombat

```
"Actor_ID"->StopCombat, "Actor2_ID"
```

Der aufrufende Actor beendet mit einem Kampf dem zweiten, angegebenen Actor. Hat der aufrufende Actor einen zu hohen Fight-Wert wird er einen Angriff auf den Spieler von sich aus fortführen. Gleiches gilt für Angriffe auf Kreaturen durch NPCs mit hohem Alarm-Wert.

Verwandte Funktionen: *GetTarget*, *StartCombat*

3.204 StopScript

```
StopScript, "Script_ID"
```

Beendet das angegebene globale Script am Ende des Frames. Ein Script kann diese Funktion auf sich selbst aufrufen. Im Editor führt dies beim ersten Speichern zu einer (logischen) Fehlermeldung, dass das zu stoppende Script nicht bekannt ist (es wurde ja noch nicht gespeichert). Ein zweites Speichern behebt diesen Fehler. Wird ein bereits gespeichertes Script um diese Funktion erweitert, erscheint keine Fehlermeldung. Auch lokale Scripte können diese Funktion aufrufen, allerdings werden diese im nächsten Frame automatisch wieder gestartet.

Verwandte Funktionen: *ScriptRunning*, *StartScript*

3.205 StopSound

```
"Object_ID"->StopSound, "Sound_ID"
```

Stoppt die Wiedergabe des angegebenen Sounds, wenn dieser vom aufrufenden Objekt wiedergegeben wird. Diese Funktion ist (abgesehen von einem Zellenwechsel) die einzige Möglichkeit einen Loop-Sound abzubrechen.

```
if ( GetSoundPlaing, "lava layer" == 1 )  
    StopSound, "lava layer"  
endif
```

Verwandte Funktionen: *GetSoundPlaying*, *PlayLoopSound3D*, *PlaySound*

3.206 StreamMusic

`StreamMusic, "Dateiname.mp3"`

Spielt die angegebene Musik als Hintergrundmusik. Die Musikdatei muss sich im Morrowind-Unterordner *Data Files/Music/* befinden. Diese Musik wird durch andere eintretende Musikstücke unterbrochen (z.B. Kampfmusik).

```
StreamMusic, "Battle/MW battle1.mp3"
```

Verwandte Funktionen: *PlaySound*

3.207 TestCells

3.208 TestInteriorCells

`TestCells`
`TestInteriorCells`

(**Konsole**) Lädt alle Zellen beziehungsweise alle Innenzellen in alphabetischer Reihenfolge. Dieser Test kann mit *StopCellTest* unterbrochen werden.

Verwandte Funktionen: *StopCellTest*, *TestThreadCells*

3.209 TestModels

`TestModels`
`t3d`

(**Konsole**) Lädt alle Objekte und meldet fehlende Meshes.

Verwandte Funktionen: -

3.210 TestThreadCells

`TestThreadCells`

(**Konsole**) Lädt alle Zellen beziehungsweise alle Innenzellen in alphabetischer Reihenfolge. Dieser Test kann mit *StopCellTest* unterbrochen werden.

Verwandte Funktionen: *StopCellTest*, *TestCells*, *TestInteriorCells*

3.211 ToggleAI

ToggleAI
tai

Die Eingabe dieser Funktion schaltet die künstliche Intelligenz ab. Alle Kreaturen und NPCs bleiben auf der Stelle stehen und spielen noch maximal ihre aktuelle Warteanimation ab. Auch jeglicher Kampf wird sofort abgebrochen. Dialoge sind hiervon nicht betroffen. Eine zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *AI...*, *GetAIPackageDone*, *GetCurrentAIPackage*, *StartCombat*, *StopCombat*

3.212 ToggleBorders

ToggleBorders
tb

(Konsole) Zeigt die Grenzen der Exteriorzellen als farbige Linien auf dem Boden an. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.213 ToggleClipping

ToggleClipping
tcl

Hebt die Kollisionsabfrage und die Schwerkraft für alle Kreaturen und Personen auf. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *GetCollidingActor*, *GetCollidingPC*, *GetStandingActor*, *GetStandingPC*, *HurtCollidingActor*, *HurtStandingPC*

3.214 ToggleCollisionBoxes

ToggleCollisionBoxes
tcb

(Konsole) Zeigt die rechteckigen und normalerweise unsichtbaren Collisionboxes aller Meshes an. Somit kann man zum Beispiel erkennen, an welchem Objekt man festhängt, wenn der Spieler anscheinend gegen eine unsichtbare Wand läuft. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.215 ToggleCollisionGrid

ToggleCollisionGrid
tcg

(Konsole) Zeigt eine Matrix mit sehr vielen Nullen und einigen anderen Zahlen, sobald man mit etwas kollidiert. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.216 ToggleCombatStats

ToggleCombatStats
tcs

(Konsole) Zeigt Informationen über die Kampffaktionen des Spielers in der Konsole an. Darunter wie hoch seine Trefferwahrscheinlichkeit bei Angriffen ist, wie wahrscheinlich er selbst von einem angreifenden Gegner getroffen wird und wie viel Schaden Treffer zugefügt haben. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *ToggleDialogueStats*, *ToggleKillStats*, *ToggleMagicStats*, *ToggleStats*

3.217 ToggleDebugText

ToggleDebugText
tdt

(Konsole) Blendet eine ganze Reihe von für das Debugging sehr nützlichen Informationen ins HUD ein. Drunter die genaue Kollisionsposition des Spielers, seine genaue Geschwindigkeit, die aktuell abgespielte Animation, den Winkel zur Z-Achse und Werte für FPS. ZUSätzlich werden auch die Werte der globalen Variablen GameHour und TimeScale angezeigt. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.218 ToggleDialogueStats

ToggleDialogueStats
tds

(Konsole) Zeigt Informationen über die Gespräche des Spielers in der Konsole an. Darunter zum Beispiel welchen Effekt Bestechungen hatten. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *ToggleCombatStats*, *ToggleKillStats*, *ToggleMagicStats*, *ToggleStats*

3.219 ToggleFogOfWar

ToggleFogOfWar
tfow

Schaltet den schwarzen Nebel auf der lokalen Karte des Spielers aus. Die gesamte Karte wird aufgedeckt. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.220 ToggleFullHelp

ToggleFullHelp
tfh

(Konsole) Zeigt im Informationsfenster jedes Objekts zusätzlich dessen Besitzer und das auf ihm laufende Script an. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.221 ToggleGodMode

ToggleGodMode
tgm

(Konsole) Macht den Spieler (fast) unverwundbar. Der Spieler kann nach wie vor aus Scripten resultierenden Schaden (zum Beispiel *ModCurrentHealth*) und Fallschaden erleiden. Zusätzlich verbraucht der Spieler keine Magie und Ausdauer mehr und Waffen, Rüstungen und Verbrauchsgegenstände nutzen sich nicht mehr ab. Die Ausdauerregeneration wird ausgeschaltet. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.222 ToggleGrid

ToggleGrid
tg

(Konsole) Zeigt ein Gitter der geladenen Zellen. Die Zelle in der Mitte ist die Zelle, in der der Spieler sich aufhält in grüner Farbe. Die grauen Zellen sind bereits vorgeladen, die rote Zelle wird geladen und die gelben müssen noch geladen werden. Diese Funktion ist praktisch, um die Zellenkoordinaten für die COE-Funktion zu erhalten. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.223 ToggleKillStats

ToggleKillStats
tkS

(Konsole) Zeigt Informationen über die Tötungen des Spielers an. Es wird angezeigt, wer getötet wurde, wie viele Tötungen der Spieler insgesamt hatte und wie viele als Werwolf. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *ToggleCombatStats*, *ToggleDialogueStats*, *ToggleMagicStats*, *ToggleStats*

3.224 ToggleLights

ToggleLights

(Konsole) Unbekannter Effekt.

Verwandte Funktionen: -

3.225 ToggleLoadFade

ToggleLoadFade
tlf

Schaltet den Ladebalken beim Betreten einer neuen Zelle ab. Zweite Eingabe schaltet ihn wieder ein.

Verwandte Funktionen: -

3.226 ToggleMagicStats

ToggleMagicStats
tms

(Konsole) Zeigt Informationen über die ablaufenden Zaubersprüche an. Darunter zum Beispiel welcher Teil der Zauberanimation abläuft, wer den Zauber gewirkt hat, wer das Ziel ist und welche Effektnummer der Zauber besitzt. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: *ToggleCombatStats*, *ToggleDialogueStats*, *ToggleKillStats*, *ToggleStats*

3.227 ToggleMenus

ToggleMenus
tm

Macht alle Menus unsichtbar (auch die Konsole). Um nach Konsoleneingabe weiterspielen zu können, muss diese mit ^ geschlossen werden. Um die Menus wieder sichtbar zu machen: Konsole öffnen und zweimal rechtsklicken. In Scripten wird der Effekt mit einer zweiten Eingabe aufgehoben.

Verwandte Funktionen: *MenuTest*

3.228 TogglePathGrid

TogglePathGrid
tpg

(Konsole) Zeigt die im Editor definierten Wege für die NPCs und Kreaturen an. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.229 ToggleScripts

ToggleScripts

(Konsole) Schaltet die Ausführung von Scripts ab. Zweite Eingabe hebt den Effekt auf.

Achtung: Es ist durchaus möglich diese Konsolenfunktion (wie fast alle) in einem Script aufzurufen. Verständlicherweise würde dies aber das Spiel unspielbar machen, da es nicht mehr möglich wäre die Scripts zu aktivieren, da dazu die Scripts ja laufen müssten...

Verwandte Funktionen: -

3.230 ToggleScriptOutput

ToggleScriptOutput
tso

(Konsole) Unbekannter Effekt.

Verwandte Funktionen: -

3.231 ToggleSky

ToggleSky
ts

(**Konsole**) Ersetzt den Himmel durch eine schwarze Fläche und die Sonnenbeleuchtung wird dauerhaft auf Nacht gesetzt. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.232 ToggleStats

ToggleStats
tst

(**Konsole**) Aktiviert/Deaktiviert alle vier Stats-Funktionen auf einmal.

Verwandte Funktionen: *ToggleCombatStats*, *ToggleDialogueStats*, *ToggleKillStats*, *ToggleMagicStats*

3.233 ToggleTextureString

ToggleTextureString
tts

(**Konsole**) Unbekannter Effekt.

Verwandte Funktionen: -

3.234 ToggleVanityMode

ToggleVanityMode
tvm

Zwingt den Spieler in die rotierende Außenansicht, die sich normalerweise aktiviert, wenn er sich zu lange nicht bewegt hat. Er kann diese Einstellung nicht mehr verlassen. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.235 ToggleWater

ToggleWater

(**Konsole**) Unbekannter Effekt.

Verwandte Funktionen: -

3.236 ToggleWireFrame

ToggleWireFrame
twf

(**Konsole**) Stellt die Welt als Gittermodell dar. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.237 ToggleWorld

ToggleWorld
tw

(**Konsole**) Alles außer dem Wasser, dem Himmel und dem Spieler wird nicht mehr dargestellt. Sämtliche Objekte und Actors sind aber trotzdem aktiv. Zweite Eingabe hebt den Effekt auf.

Verwandte Funktionen: -

3.238 TurnMoonRed

3.239 TurnMoonWhite

TurnMoonRed
TurnMoonWhite

(**Bloodmoon**) Verändert die Farbe Secundas, des kleineren Mondes, auf die in der *Morrowind.ini* in der Sektion *[Moons]* angegebene *Script Color* (Standardwert: 255,20,20) beziehungsweise setzt die Farbe wieder auf das normale Aussehen zurück.

Verwandte Funktionen: -

3.240 UndoWerewolf

"NPC_ID"->UndoWerewolf

(**Bloodmoon**) Verwandelt den aufrufenden Actor aus der Werewolfform in die Normalform zurück.

Verwandte Funktionen: *Become Werewolf*, *Is Werewolf*, *SetWerewolfAcrobatics*

3.241 Unlock

"Object_ID"->Unlock

Entfernt jedes Schloss von dem aufrufenden Objekt. Das Objekt muss dazu eine Tür oder ein Container sein. Fallen können nicht entfernt werden.

Verwandte Funktionen: *GetLocked*, *Lock*

3.242 UsedOnMe

"Object_ID"->UsedOnMe, "Object2_ID" (?)

Funktion defekt! Oder korrekte Funktionsweise bisher noch von niemanden entdeckt.

Verwandte Funktionen: *OnActivate*

3.243 WakeUpPC

WakeUpPC

"Weckt"den Spieler, wenn er zum Zeitpunkt des Funktionsaufrufs rastet oder wartet.

```
if ( GameHour > 6 )
    if ( GetPCSleep == 1 )
        WakeUpPC
        MessageBox, "Aufwachen, Schlafmütze!"
    endif
endif
```

Verwandte Funktionen: *GetPCSleep*

3.244 XBox

XBox

Liefert eine 1, wenn das Spiel auf der X-Box läuft. Da keine Modifikationen für die X-Box erstellt werden können, ist diese Funktion für das eigene Scripting nutzlos.

Verwandte Funktionen: -

Sachverzeichnis

Animation

- LoopGroup, 44
- PlayGroup, 55
- SkipAnim, 68

Bewegung

- CenterOnCell, 17
- CenterOnExterior, 17
- Disable, 20
- Fall, 27
- GetAngle, 32
- GetPos, 37
- GetStartingAngle, 40
- GetStartingPos, 40
- Move, 50
- MoveWorld, 50
- Position, 56
- PositionCell, 57
- Rotate, 60
- RotateWorld, 60
- SetAngle, 65
- SetAtStart, 64
- SetPos, 66

Bloodmoon

- BecomeWerewolf, 15
- GetPCInJail, 28
- GetPCTraveling, 28
- GetWerewolfKills, 29
- IsWerewolf, 43
- PlaceAtMe, 54
- SetWereWolfAcrobatics, 66
- TurnMoonRed, 76
- TurnMoonWhite, 76
- UndoWerewolf, 76

Charaktererstellung

- EnableBirthMenu, 23
- EnableClassMenu, 23
- EnableInventoryMenu, 23
- EnableMagicMenu, 24
- EnableMapMenu, 24
- EnableNameMenu, 24
- EnableRaceMenu, 24
- EnableStatsMenu, 25
- EnableStatsReviewMenu, 24

Dialog

- AddTopic, 12
- Choice, 18
- ClearInfoActor, 19
- ForceGreeting, 27
- GetJournalIndex, 36
- GoodBye, 41
- Journal, 43

- SetJournalIndex, 66

Fraktionen

- GetFactionReaction, 35
- GetPCFacRep, 37
- GetPCRank, 37
- LowerRank, 44
- ModFactionReaction, 48
- ModPCFacRep, 48
- PCClearExpell, 52
- PCExpell, 53
- PCExpelled, 53
- PCJoinFaction, 53
- PCLowerRank, 54
- PCRaiseRank, 54
- RaiseRank, 57
- SameFaction, 61
- SetFactionReaction, 65
- SetPCFacRep, 66

Interaktion

- Activate, 10
- CellChanged, 16
- GetButtonPressed, 32
- GetCollidingActor, 28
- GetCollidingPC, 28
- GetDetected, 34
- GetDistance, 34
- GetInterior, 28
- GetLineOfSight, 36
- GetPCCell, 37
- GetSpellReadied, 29
- GetStandingActor, 29
- GetStandingPC, 29
- GetWeaponDrawn, 29
- GetWerewolfKills, 29
- GoToJail, 41
- HitAttemptOnMe, 42
- HitOnMe, 42
- HitStandingActor, 43
- HurtCollidingActor, 43
- MessageBox, 45
- OnActivate, 51
- OnDeath, 51
- OnKnockout, 51
- OnMurder, 51
- PayFine, 52
- PayFineThief, 52
- UsedOnMe, 77

Inventar

- AddItem, 10
- AddSoulGem, 10
- Drop, 22
- DropSoulGem, 22

- Equip, 25
- GetArmorType, 32
- GetItemCount, 35
- GetWeaponType, 40
- HasItemEquipped, 41
- HasSoulGem, 42
- RemoveItem, 58
- RemoveSoulGem, 59
- RepairedOnMe, 59
- Künstliche Intelligenz
 - AIActivate, 12
 - AIEscort, 13
 - AIEscortCell, 13
 - AIFollow, 14
 - AIFollowCell, 14
 - AITravel, 14
 - AIWander, 15
 - Face, 26
 - GetAIPackageDone, 28
 - GetCurrentAIPackage, 33
 - GetTarget, 40
 - StartCombat, 68
 - StopCombat, 69
 - ToggleAI, 71
- Konsole
 - BetaComment, 16
 - CenterOnCell, 17
 - CenterOnExterior, 17
 - CreateMaps, 19
 - FillJournal, 27
 - FillMap, 27
 - FixMe, 27
 - Help, 42
 - ObjectReferenceInfo, 51
 - OutputObjCounts, 52
 - OutputRefCounts, 52
 - PurgeTextures, 57
 - ResetActors, 60
 - Show, 67
 - ShowAnim, 67
 - ShowGroup, 67
 - ShowSceneGraph, 68
 - ShowTargets, 68
 - ShowVars, 68
 - StopCellTest, 69
 - TestCells, 70
 - TestInteriorCells, 70
 - TestModels, 70
 - TestThreadCells, 70
 - ToggleAI, 71
 - ToggleBorders, 71
 - ToggleClipping, 71
 - ToggleCollisionBoxes, 71
 - ToggleCollisionGrid, 71
 - ToggleCombatStats, 72
 - ToggleDebugText, 72
 - ToggleDialogueStats, 72
 - ToggleFogOfWar, 72
 - ToggleFullHelp, 72
 - ToggleGodMode, 73
 - ToggleGrid, 73
 - ToggleKillStats, 73
 - ToggleLights, 73
 - ToggleLoadFade, 73
 - ToggleMagicStats, 74
 - ToggleMenus, 74
 - TogglePathGrid, 74
 - ToggleScriptOutput, 74
 - ToggleScripts, 74
 - ToggleSky, 75
 - ToggleStats, 75
 - ToggleTextureString, 75
 - ToggleVanityMode, 75
 - ToggleWater, 75
 - ToggleWireFrame, 76
 - ToggleWorld, 76
- Magie
 - AddSpell, 11
 - Cast, 16
 - DisableLevitation, 20
 - DisableTeleporting, 21
 - EnableLevitation, 23
 - EnableTeleporting, 25
 - ExplodeSpell, 25
 - GetEffect, 34
 - GetSpell, 39
 - GetSpellEffects, 39
 - RemoveEffects, 58
 - RemoveSpell, 59
 - RemoveSpellEffects, 59
- Sonstiges
 - AddToLevCreature, 11
 - AddToLevItem, 11
 - BecomeWerewolf, 15
 - DontSaveObject, 22
 - Enable, 23
 - GetMasserPhase, 36
 - GetSecundaPhase, 38
 - Lock, 44
 - MenuMode, 44
 - MenuTest, 45
 - ModWaterLevel, 46
 - PlaceAtMe, 54
 - PlaceAtPC, 54
 - PlaceItem, 54
 - PlaceItemCell, 55
 - PlayBink, 55
 - Random, 57
 - RemoveFromLevCreature, 58
 - RemoveFromLevItem, 58
 - ScriptRunning, 62

- SetDelete, 64
- SetWaterLevel, 62
- ShowMap, 67
- StartScript, 69
- StopScript, 69
- TurnMoonRed, 76
- TurnMoonWhite, 76
- UndoWerewolf, 76
- Unlock, 76
- WakeUpPC, 77
- XBox, 77
- Sound
 - GetSoundPlaying, 39
 - PlayLoopSound3D, 56
 - PlayLoopSound3DVP, 56
 - PlaySound, 56
 - PlaySound3D, 56
 - PlaySound3DVP, 56
 - PlaySoundVP, 56
 - Say, 61
 - SayDone, 61
 - StopSound, 69
 - StreamMusic, 70
- Spielerkontrollen
 - ClearForceJump, 19
 - ClearForceMoveJump, 19
 - ClearForceRun, 19
 - ClearForceSneak, 19
 - DisablelayerLooking, 21
 - DisableLevitation, 20
 - DisablePlayerControls, 20
 - DisablePlayerFighting, 20
 - DisablePlayerJumping, 21
 - DisablePlayerMagic, 21
 - DisablePlayerViewSwitch, 21
 - DisableTeleporting, 21
 - DisableVanityMode, 22
 - EnableLevelUpMenu, 23
 - EnableLevitation, 23
 - EnablePlayerControls, 24
 - EnablePlayerFighting, 24
 - EnablePlayerJumping, 24
 - EnablePlayerLooking, 24
 - EnablePlayerMagic, 24
 - EnablePlayerViewSwitch, 24
 - EnableTeleporting, 25
 - FadeIn, 26
 - FadeOut, 26
 - FadeTo, 26
 - ForceJump, 28
 - ForceMoveJump, 28
 - ForceRun, 28
 - ForceSneak, 28
 - GetForceJump, 28
 - GetForceMoveJump, 28
 - GetForceRun, 28
 - GetForceSneak, 28
 - GetPlayerControlsDisabled, 28
 - GetPlayerFightingDisabled, 28
 - GetPlayerJumpingDisabled, 28
 - GetPlayerLookingDisabled, 29
 - GetPlayerMagicDisabled, 29
 - GetPlayerViewSwitchDisabled, 29
 - GetVanityModeDisabled, 29
 - MoveOneToOne, 50
 - PCForce1stPerson, 53
 - PCForce3rdPerson, 53
 - PCGet3rdPerson, 53
 - ShowRestMenu, 67
- Status
 - Get, 28
 - GetArmorType, 28
 - GetAttacked, 28
 - GetBlightDisease, 28
 - GetCommonDisease, 28
 - GetDeadCount, 34
 - GetDisabled, 28
 - GetHealthGetRatio, 28
 - GetLocked, 28
 - GetPCCrimeLevel, 28
 - GetPCInJail, 28
 - GetPCJumping, 28
 - GetPCRunning, 28
 - GetPCSleep, 28
 - GetPCSneaking, 28
 - GetPCTraveling, 28
 - GetPCVisionBonus, 28
 - GetRace, 38
 - GetWaterLevel, 29
 - Mod, 46
 - ModPCCrimeLevel, 46
 - ModPCVisionBonus, 46
 - Resurrect, 60
 - Set, 62
 - SetLevel, 65
 - SetPCCrimeLevel, 62
 - SetPCVisionBonus, 62
 - SetWereWolfAcrobatics, 66
- Tribunal
 - AddToLevCreature, 11
 - AddToLevItem, 11
 - ClearForceJump, 19
 - ClearForceMoveJump, 19
 - ClearForceRun, 19
 - DisableLevitation, 20
 - EnableLevitation, 23
 - ForceJump, 28
 - ForceMoveJump, 28
 - ForceRun, 28
 - GetForceJump, 28
 - GetForceMoveJump, 28
 - GetForceRun, 28
 - GetLevel, 30

- GetPCJumping, 28
- GetPCRunning, 28
- GetPCSneaking, 28
- GetScale, 30
- GetSpellReadied, 29
- GetWaterLevel, 29
- GetWeaponDrawn, 29
- ModScale, 47
- ModWaterLevel, 46
- RemoveFromLevCreature, 58
- RemoveFromLevItem, 58
- SetDelete, 64
- SetLevel, 65
- SetScale, 63
- SetWaterLevel, 62

Wetter

- ChangeWeather, 17
- GetCurrentWeather, 33
- GetWindSpeed, 29
- ModRegion, 49

Zeit

- GetCurrentTime, 28
- GetSecondsPassed, 38

4 Anhang

4.1 Konstanten

In Morrowind-Script gibt es zwar keine Stringvariablen, aber eine Reihe von Stringkonstanten. Das sind fest vorgegebene Namen, die vom Spiel einen festen Wert zugewiesen bekommen haben. Die Inhalte dieser Konstanten lassen sich in Dialogen und Messageboxen anzeigen. Das bekannteste Beispiel ist der Name des Spielers, *PCName*.

Um eine Stringkonstante im Dialog anzuzeigen, muss der Name mit einem % angeführt (escaped) werden. In Messageboxen muss das ^ (Zirkumflex) verwendet werden.

Beispiel: %*PCName* (im Dialog) oder ^*PCName* (in einer Messagebox)

Konstante	Bedeutung
Cell	Zellenname der Zelle in der sich der Spieler aufhält
Class	Klasse des aufrufenden NPCs
Faction	Fraktion des aufrufenden NPCs
Name	Name des aufrufenden NPCs
NextPCRank	Nächster Rang des Spielers in der Fraktion des aufrufenden NPCs
PCClass	Klasse des Spielers
PCName	Name des Spielers
PCRace	Rasse des Spielers
PCRank	Rang des Spielers in der Fraktion des aufrufenden NPCs
Race	Rasse des aufrufenden NPCs
Rank	Rang des aufrufenden NPCs in dessen Fraktion
ActionForward	Bezeichnung der eingestellten Vorwärtstaste
ActionBack	Bezeichnung der eingestellten Rückwärtstaste
ActionSlideLeft	Bezeichnung der eingestellten Linkstaste
ActionSlideRight	Bezeichnung der eingestellten Rechtstaste
ActionMenuMode	Bezeichnung der eingestellten Menutaste
ActionActivate	Bezeichnung der eingestellten Aktivierentaste
ActionUse	Bezeichnung der eingestellten Angriffstaste
ActionReadyItem	Bezeichnung der eingestellten Waffe ziehen - Taste
ActionReadyMagic	Bezeichnung der eingestellten Zaubertaste
ActionCrouch	Bezeichnung der eingestellten Schleichtaste
ActionRun	Bezeichnung der eingestellten Rennentaste
ActionToggleRun	Bezeichnung der eingestellten Dauerhaft Rennen - Taste
ActionJump	Bezeichnung der eingestellten Sprungtaste
ActionJournal	Bezeichnung der eingestellten Journaltaste
ActionRestMenu	Bezeichnung der eingestellten Rastentaste
ActionNextWeapon	Bezeichnung der eingestellten Nächste Waffe - Taste
ActionPrevWeapon	Bezeichnung der eingestellten Vorherige Waffe - Taste
ActionNextSpell	Bezeichnung der eingestellten Nächster Zauber - Taste
ActionPrevSpell	Bezeichnung der eingestellten Vorheriger Zauber - Taste

Anmerkung: Die Bezeichnungen aller Tasten sind in den GameSettings unter Menus in den Einstellungen *sKeyName_00* bis *sKeyName_FF* gespeichert und können dort entsprechend geändert werden.

4.2 Liste der magischen Effekte

Die Effektnummern werden für die Funktion *RemoveEffects* und die Effekt-IDs für die Funktion *GetEffect* benötigt.

Nummer	ID
0	sEffectWaterBreathing
1	sEffectSwiftSwim
2	sEffectWaterWalking
3	sEffectShield
4	sEffectFireShield
5	sEffectLightningShield
6	sEffectFrostShield
7	sEffectBurden
8	sEffectFeather
9	sEffectJump
10	sEffectLevitate
11	sEffectSlowFall
12	sEffectLock
13	sEffectOpen
14	sEffectFireDamage
15	sEffectShockDamage
16	sEffectFrostDamage
17	sEffectDrainAttribute
18	sEffectDrainHealth
19	sEffectDrainSpellpoints
20	sEffectDrainFatigue
21	sEffectDrainSkill
22	sEffectDamageAttribute
23	sEffectDamageHealth
24	sEffectDamageMagicka
25	sEffectDamageFatigue
26	sEffectDamageSkill
27	sEffectPoison
28	sEffectWeaknessToFire
29	sEffectWeaknessToFrost
30	sEffectWeaknessToShock
31	sEffectWeaknessToMagicka
32	sEffectWeaknessToCommonDisease
33	sEffectWeaknessToBlightDisease
34	sEffectWeaknessToCorprusDisease
35	sEffectWeaknessToPoison
36	sEffectWeaknessToNormalWeapons
37	sEffectDisintegrateWeapon
38	sEffectDisintegrateArmor
39	sEffectInvisibility
40	sEffectChameleon
41	sEffectLight
42	sEffectSanctuary
43	sEffectNightEye
44	sEffectCharm
45	sEffectParalyze
46	sEffectSilence
47	sEffectBlind
48	sEffectSound

49	sEffectCalmHumanoid
50	sEffectCalmCreature
51	sEffectFrenzyHumanoid
52	sEffectFrenzyCreature
53	sEffectDemoralizeHumanoid
54	sEffectDemoralizeCreature
55	sEffectRallyHumanoid
56	sEffectRallyCreature
57	sEffectDispel
58	sEffectSoultrap
59	sEffectTelekinesis
60	sEffectMark
61	sEffectRecall
62	sEffectDivineIntervention
63	sEffectAlmsiviIntervention
64	sEffectDetectAnimal
65	sEffectDetectEnchantment
66	sEffectDetectKey
67	sEffectSpellAbsorption
68	sEffectReflect
69	sEffectCureCommonDisease
70	sEffectCureBlightDisease
71	sEffectCureCorprusDisease
72	sEffectCurePoison
73	sEffectCureParalyzation
74	sEffectRestoreAttribute
75	sEffectRestoreHealth
76	sEffectRestoreSpellPoints
77	sEffectRestoreFatigue
78	sEffectRestoreSkill
79	sEffectFortifyAttribute
80	sEffectFortifyHealth
81	sEffectFortifySpellPoints
82	sEffectFortifyFatigue
83	sEffectFortifySkill
84	sEffectFortifyMagickaMultiplier
85	sEffectAbsorbAttribute
86	sEffectAbsorbHealth
87	sEffectAbsorbSpellPoints
88	sEffectAbsorbFatigue
89	sEffectAbsorbSkill
90	sEffectResistFire
91	sEffectResistFrost
92	sEffectResistShock
93	sEffectResistMagicka
94	sEffectResistCommonDisease
95	sEffectResistBlightDisease
96	sEffectResistCorprusDisease
97	sEffectResistPoison
98	sEffectResistNormalWeapons
99	sEffectResistParalysis
100	sEffectRemoveCurse
101	sEffectTurnUndead
102	sEffectSummonScamp
103	sEffectSummonClannfear
104	sEffectSummonDaedroth
105	sEffectSummonDremora

106	sEffectSummonAncestralGhost	
107	sEffectSummonSkeletalMinion	
108	sEffectSummonLeastBonewalker	
109	sEffectSummonGreaterBonewalker	
100	sEffectSummonBonelord	
111	sEffectSummonWingedTwilight	
112	sEffectSummonHunger	
113	sEffectSummonGoldensaint	
114	sEffectSummonFlameAtronach	
115	sEffectSummonFrostAtronach	
116	sEffectSummonStormAtronach	
117	sEffectFortifyAttackBonus	
118	sEffectCommandCreatures	
119	sEffectCommandHumanoids	
120	sEffectBoundDagger	
121	sEffectBoundLongsword	
122	sEffectBoundMace	
123	sEffectBoundBattleAxe	
124	sEffectBoundSpear	
125	sEffectBoundLongbow	
126	sEffectExtraSpell	unbelegt
127	sEffectBoundCuirass	
128	sEffectBoundHelm	
129	sEffectBoundBoots	
130	sEffectBoundShield	
131	sEffectBoundGloves	
132	sEffectCorprus	
133	sEffectVampirism	
134	sEffectSummonCenturionSphere	
135	sEffectSunDamage	
136	sEffectStuntedMagicka	
137	sEffectSummonFabricant	(Tribunal) Mechanoiden beschwören
138	sEffectSummonCreature01	(Bloodmoon) Wolf beschwören
139	sEffectSummonCreature02	(Bloodmoon) Bär beschwören
140	sEffectSummonCreature03	(Bloodmoon) Knochenwolf beschwören
141	sEffectSummonCreature04	(Bloodmoon) unbelegt
142	sEffectSummonCreature05	(Bloodmoon) unbelegt

4.3 Inventarscripte

Neben Scriptfunktionen besitzt das Spiel noch eine weitere Möglichkeit bestimmte Inventaraktionen des Spielers zu detektieren. Dazu müssen bestimmte lokale Variablen in einem Script deklariert werden. Morrowind setzt diese Variablen bei bestimmten Aktion automatisch. In der Hilfe werden diese als *Inventory Scripts* bezeichnet. Weiterhin gibt es einige ähnliche Variablen, die bestimmte Aktionen verhindern oder auslösen, wenn man sie zuvor in einem Script gesetzt hat. Diese Variablen werden hier nun vorgestellt.

Zunächst einmal ein Beispiel: Wenn man detektieren möchte, wann der Spieler einen bestimmten Gegenstand ausrüstet, ist dies bei Waffen oder Rüstungen mit ein wenig Aufwand noch möglich, aber spätestens bei einem Buch oder einem Trank versagen die bekannten Scriptfunktionen. So kann man herausbekommen, ob ein bestimmter Gegenstand vom Spieler ausgerüstet wird:

Dieser Gegenstand muss mit einem Script versehen sein und dies muss in folgender Art aufgebaut sein:

```
begin kf_beispiel1_script

short OnPCEquip    ;wird automatisch auf 1 gesetzt, wenn der Spieler den Gegenstand ausrüstet

if ( OnPCEquip == 1 )
    MessageBox, "Ihr hört Euer Schwert in Eurem Geist: 'Füttert mir Seelen!'"
    set OnPCEquip to 0
endif

end
```

Alle verfügbaren Inventarscripte dieser Art sind in dieser Tabelle aufgeführt:

Variable	Funktionsweise
PCSkipEquip	Diese Variable muss per Hand auf 1 gesetzt werden. Solange sie den Wert 1 hat, kann der mit dem Script belegte Gegenstand nicht ausgerüstet werden. Trotzdem wird bei dem Versuch den Gegenstand zu benutzen die Variable <i>OnPCEquip</i> auf 1 gesetzt. Da der Gegenstand nie ausgerüstet wurde, muss <i>OnPCEquip</i> per Hand auf 0 gesetzt werden. Um ein Ausrüsten des Gegenstandes wieder zu ermöglichen muss <i>PCSkipEquip</i> wieder auf 0 gesetzt werden. Siehe hierzu auch das Beispielscript zum versiegelten Buch.
OnPCAdd	Wird auf 1 gesetzt, wenn der Spieler den Gegenstand seinem Inventar hinzufügt.
OnPCDrop	Wird auf 1 gesetzt, wenn der Spieler den Gegenstand aus seinem Inventar ablegt.
OnPCEquip	Wird auf 1 gesetzt, wenn der Spieler den Gegenstand ausrüstet (bzw. auf sein Charakterbild im Inventar zieht). Die Variable wird beim ablegen automatisch auf 0 gesetzt, kann aber je nach Situation auch per Hand zurückgesetzt werden (siehe oben).
OnPCRepair	Liefert eine 1, wenn der Gegenstand repariert wird. Genauer gesagt ist es egal, ob der Spieler den Gegenstand selbst repariert, nur versucht ihn zu reparieren (also die Reparatur misslingt) oder ob der Spieler einen NPC dafür bezahlt den Gegenstand zu reparieren.
OnPCSoulGemUse	Liefert eine 1, wenn der Spieler den Seelenstein verbraucht.
OnRepair	Funktion defekt!

Die andere Art von Inventarscripten wird nicht benutzt um Aktionen des Spielers zu entdecken, sondern um das Verhalten von NPCs und Kreaturen zu steuern. Entsprechend müssen diese Variablen in Scripten, die auf dem gewünschten Actor liegen deklariert werden. Folgende sind verfügbar:

Variable	Funktionsweise
Companion	(Tribunal) Diese Variable muss per Hand auf 1 gesetzt werden. Solange sie den Wert 1 hat, wird der mit dem Script belegte Actor, dem Spieler folgen und für ihn kämpfen. Zusätzlich ist über das Dialogmenu die Option Teilen verfügbar, mit dem der Spieler auf das Inventar des Actors zugreifen kann. Um den Effekt zu beenden, muss die Variable wieder auf 0 gesetzt werden.
minimumProfit	(Tribunal) (float) Diese Variable gibt die Differenz zwischen dem aktuellen Geldwert aller Gegenstände im Inventar des Actors und dem Geldwert aller Gegenstände im Inventar des Actors zu dem Zeitpunkt, als die <i>Companion</i> -Variable auf 1 gesetzt wurde zurück.
NoFlee	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, wird der Actor keine Voices aus dem Flee-Thema mehr von sich geben.
NoHello	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, wird der Actor keine Voices aus dem Hello-Thema mehr von sich geben.
NoIdle	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, wird der Actor keine Voices aus dem Idle-Thema mehr von sich geben.
NoIntruder	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, wird der Actor keine Voices aus dem Intruder-Thema mehr von sich geben.
NoLore	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, verfügt der Actor nur noch über Dialogoptionen, die als ID seine ID angegeben haben. Alle Antworten ohne festgelegte ID sind nicht mehr verfügbar.
NoThief	Diese Variable muss nur deklariert werden. Sobald sie im Script steht, wird der Actor keine Voices aus dem Thief-Thema mehr von sich geben.
StayOutside	(Bloodmoon) Diese Variable muss per Hand auf 1 gesetzt werden. Solange der Actor, der dieses Script trägt dem Spieler folgt, wird er keine Interiorbereiche betreten. Sobald der Spieler wieder in den Exteriorbereich zurückkehrt, folgt der Actor dem Spieler wieder automatisch.

4.4 Nützliche globale Variablen

Morrowind besitzt viele globale Variablen. Einige davon liefern Werte des Spiels, die sonst nicht zu erfassen sind. Hier ist eine Tabelle mit nützlichen globalen Variablen zu finden (der Startwert gibt den Wert der Variable in einem neu gestarteten Spiel an):

Variable	Typ	Startwert	Funktionsweise
CharGenState	float	0.00	Gibt an, wie weit die Charaktererstellung fortgeschritten ist. Speichern, das Tagebuch und einige andere Dinge sind nur verfügbar, wenn diese Variable auf -1 steht.
CrimeGoldDiscount	short	0	Gibt den aus <i>GetPCCrimeLevel</i> abgeleiteten Geldbetrag an, den der Spieler bei der Diebesgilde zum Entfernen seines Kopfgeldes bezahlen muss. In der Standardeinstellung ist dies die Hälfte. Der Multiplikator wird im Gamesetting <i>fCrimeGoldDiscountMult</i> festgelegt.
CrimeGoldTurnIn	short	0	Gibt den aus <i>GetPCCrimeLevel</i> abgeleiteten Geldbetrag an, den der Spieler zum Entfernen seines Kopfgeldes bezahlen muss, wenn er sich den Wachen stellt. In der Standardeinstellung sind das 90%. Der Multiplikator wird im Gamesetting <i>fCrimeGoldTurnInMult</i> festgelegt.
Day	short	16	Gibt den Tag im aktuellen Monat an.
DaysPassed	short	1	(Tribunal) Gibt die Anzahl der bereits gespielten Tage an.
DestroyBlight	short	-1	Wird am Ende der Hauptquest auf -1 gesetzt (durch das Script <i>EndGame</i>).
DuelActive	short	0	Gibt an, ob in der Arena von Vivec ein Duell abläuft (wird in den entsprechenden Duellscripten gesetzt).
GameHour	float	9.00	Gibt die aktuelle Uhrzeit im Spiel an. Identisch zu <i>GetCurrentTime</i> .
Month	short	7	Gibt den aktuellen Monat an.
PCGold	long	0	Gibt an, wie viele Draken der Spieler in seinem Inventar hat.

PCHasCrimeGold	short	0	Dieser Wert ist 1, wenn der Spieler mindestens so viel Geld im Inventar besitzt wie der Wert den <i>GetPC-CrimeLevel</i> liefert.
PCHasCrimeGoldDiscount	short	0	Der Wert ist 1, wenn der Spieler mindestens so viel Geld im Inventar hat, wie in <i>CrimeGoldDiscount</i> gespeichert ist.
PCHasCrimeGoldTurnIn	short	0	Der Wert ist 1, wenn der Spieler mindestens so viel Geld im Inventar hat, wie in <i>CrimeGoldTurnIn</i> gespeichert ist.
PCRace	short	0	Gibt die Rasse des Spielers an. 0 steht für eine eigene Rasse, 1 für Argonier, 2 für Bretone, 3 für Dunkelelf, 4 für Hochelf, 5 für Imperialer, 6 für Khajiit, 7 für Nord, 8 für Ork, 9 für Rothwardon und 10 für Waldelf.
PCVampire	short	0	Liefert eine 1, wenn der Spieler Vampir ist und eine -1, wenn er geheilt wurde (wird von den entsprechenden Vampirsripten gesetzt).
PCWerewolf	short	0	(Bloodmoon) Liefert eine 1, wenn der Spieler Werwolf ist und eine -1, wenn er geheilt wurde (wird von den entsprechenden Werwolfscripten gesetzt).
Random100	short	0	Wird jeden Frame auf eine Zufallszahl zwischen 0 und 99 gesetzt.
TimeScale	float	30.00	Gibt das Verhältnis von Spielzeit zu Echtzeit an. Standardmäßig läuft die Zeit im Spiel 30mal so schnell wie in echt.
VamClan	short	0	Gibt an welchem Vampiercaln der Spieler angehört, wenn dieser ein Vampir ist. 1 steht für Aundae, 2 für Berne und 3 für Quarra.
Year	short	427	Gibt das aktuelle Jahr an.

4.5 Beispielscripte

Zum Abschluss sind noch einige Beispielscripte aufgeführt. Diese sind natürlich weit von einer erschöpfenden Darstellung der Scriptmöglichkeiten entfernt, aber es sind einfache Implementierungen, der in den Foren am häufigsten angefragten Scripte.

4.5.1 Teleporteramulett

Als erstes Script möchte ich einen der Klassiker vorstellen: das Teleporteramulett. Dieses besitzt zwar nur einen einzigen Zielort, aber es sollte kein Problem sein, es auf weitere Zielorte zu erweitern.

```
begin kf_teleport_amulet01_script

short OnPCEquip
short state
short button

if ( state == 0 )
    if ( OnPCEquip == 1 )
        set state to 1
        set OnPCEquip to 0
        MessageBox, "Wollt Ihr nach Balmora zurückkehren?", "Ja", "Nein"
    endif
elseif ( state == 1 )
    set button to GetButtonPressed
    if ( button == 0 )
        set button to -1
        set state to 0
        Player->PositionCell, -24930, -12520, 1070, 270, "Balmora"
    elseif ( button == 1 )
        set button to -1
        set state to 0
    endif
endif
endif

end
```

4.5.2 Fallgitter

Wenn das folgendes Script auf ein Tor gelegt wird, kann das Tor durch Aktivieren gehoben werden. Nach einer gewissen Zeit senkt es sich wieder ab. Das Tor wird insgesamt um 210 Editoreinheiten gehoben, das ist die gleiche Höhe wie beim Tor in der Geisterpforte.

```
begin kf_fallgitter01_script

float timer
short state

if ( state == 0 )
    if ( OnActivate == 1 )
        set state to 1
    endif
elseif ( state == 1 )
    set timer to timer + GetSecondsPassed
    if ( timer <= 3 )
        MoveWorld, Z, 70
    elseif ( timer >= 8 )
        MoveWorld, Z, -70
        if ( timer >= 11 )
            set state to 0
            set timer to 0
        endif
    endif
endif

end
```

4.5.3 Rüstungspuppe

Eine Rüstungspuppe wird immer gerne gesehen, wenn es darum geht seine Rüstungen oder Kleidungen auszustellen. Man kann diese beliebig kompliziert aufbauen, eine einfache solche Puppe lässt sich aber mit wenigen Zeilen Code umsetzen. Dazu erschafft man zunächst einen neuen NPC und setzt einen Haken bei Corpse persistent. Wenn man ihn danach mit diesem Script ausstattet, funktioniert er wie eine einfache Rüstungspuppe:

```
begin kf_dummy01_script

SetUnarmored, 0 ;stellt sicher, dass der NPC die Rüstung auch anzieht
SetHealth, 0    ;tötet den NPC
SkipAnim        ;verhindert, dass die Leiche umfällt

end
```

4.5.4 Versiegeltes Buch

Bücher besitzen in Morrowind ein sehr ungewöhnliches Aktivierungssystem. Deshalb kann man nicht mit *OnActivate* herausfinden, ob ein Buch geöffnet wird. Dafür gibt es eine kompliziertere Struktur, die im folgenden Script gezeigt wird. Das damit ausgerüstete Buch, wird beim ersten Öffnen eine Messagebox zeigen, dass ein Siegel aufgebrochen wurde.

```
begin my_book01_script

short PCSkipEquip
short OnPCEquip
short state

if ( state == 0 )
    Set PCSkipEquip to 1
    Set state to 1
elseif ( state == 1 )
    if ( OnPCEquip == 1 )
        MessageBox, "Ihr habt das Siegel aufgebrochen."
        set OnPCEquip to 0
        set PCSkipEquip to 0
        Activate
        set state to 2
    endif
endif

end
```