

Die Grundlagen des Scriptens

Schritt 1: *Der SA-MP Client und die Serversoftware*

Download @ GTAVision:

- [SA-MP 0.2X Client](#)
- [SA-MP 0.2X Windows-Server](#)
- [SA-MP 0.2X Linux-Server](#)

Den Clienten, falls ihr ihn noch nicht habt, über die .exe Datei installieren. Das Serverpaket besteht aus einem Archiv, indem alle notwendigen Daten vorhanden sind. Dieses entpackt ihr in euren GTA San Andreas Ordner.

Schritt 2: *Installation sowie Konfiguration des Servers*

Der Server lässt sich über die Anwendung samp-server.exe bzw. mit samp02Xsvr auf einem Linuxsystem starten. Der Kern der Sache liegt aber in der server.cfg Datei - in der ihr alle notwendigen Daten für euren Server eintragen müsst. Im folgenden ein Auszug sowie eine kurze Erläuterung der Parameter.

```
echo Executing Server Config...
lanmode 0
rcon_password changeme
maxplayers 32
port 7777
hostname Unnamed 0.2X Server
gamemode0 lvdm 1
filterscripts adminspect vacations
announce 1
query 1
weburl www.sa-mp.com
anticheat 0
```

- *lanmode* ~ 0 bedeutet ihr spielt über das Internet, 1 heißt ihr spielt über LAN.
- *maxplayers* ~ hier stellt ihr ein, wieviele Spieler auf euren Server kommen dürfen.
- *port* ~ hier stellt ihr ein, über welchen Port der Server laufen soll. (Muss in eurer Routerkonfiguration geöffnet sein)
- *hostname* ~ hier gebt ihr euren Servernamen ein.

- *gamemode0* ~ hier stellt ihr ein, welche GameModes auf eurem Server laufen sollen. Am besten ihr macht nur eine Zeile, also "gamemode0 amxname 1".
- *rcon_password* ~ hier müsst ihr unbedingt euer Adminpasswort angeben, damit ihr euch später im Server einloggen könnt.
- *filterscripts* ~ hier die .amx Namen der Filterscripts eingeben, getrennt mit einem Leerzeichen (Filterscripts müssen sich im Ordner "filterscripts" befinden)
- *announce* ~ entweder true (1) oder false (0) als Parameter angeben. Bestimmt, ob der Server in der Internetliste angezeigt werden soll oder nicht.
- *query* ~ entweder true (1) oder false (0) als Parameter angeben. Bestimmt, ob Serverdaten in SA-MP angezeigt werden sollen oder nicht (dazu zählt auch der Ping und somit der Online-Status)
- *anticheat* ~ entweder true (1) oder false (0) als Parameter angeben. Bestimmt, ob SAC aktiviert werden soll oder nicht. Mit SAC werden gebannte Spieler auf diesem Server auf eine Blacklist gesetzt, sodass sie auf allen anderen Servern die auch SAC aktiviert haben auch gebannt sind.

WICHTIG: Das Anti-Cheat System SAC wird nicht mehr weitergeführt. Ihr müsst false (0) angeben!

- *mapname *optional** ~ hier könnt ihr euren Kartennamen angeben, Standard ist "San Andreas".
- *weburl *optional** ~ hier könnt ihr eine Website eingeben, falls vorhanden.

Wenn ihr die Datei verändert habt, abspeichern und fertig ist die Serverkonfiguration. Falls ihr nur einen vorgefertigten GameMode starten wollt, könnt ihr das Tutorial an dieser Stelle beenden!

Schritt 3: Der PAWNO-Script Editor

Den Editor findet ihr im Ordner "pawno", ist nicht zu verfehlen. Nun seht ihr rechts eine Liste mit Befehlen, die bis jetzt verfügbar sind. Auf einige von ihnen werden wir später noch einmal zurückkommen. Du kannst nun vorgefertigte Scripte öffnen (.pwn), im Ordner "gamemodes" sind einige. Du kannst auch ein Template (Vorlage) eines Scriptes öffnen. Scripte kannst du speichern und compilieren. Compilieren bedeutet, dass eure Scripte in das .amx Format konvertiert werden. Nur mit diesen compilierten Dateien könnt ihr eure GameModes zum Laufen bringen, die compilierten Versionen

der vorgefertigten Scripte sind im Ordner "gamemodes". Beachtet, dass die Dateien immer in diesem Ordner bleiben müssen, weil der Server sie sonst nicht findet. Wenn ihr ein Script compilt habt, ist die .amx Datei im Ordner "gamemodes".

Schritt 4: *Der LVDMode*

Als Beispiel für ein Script nehmen wir den bekannten LVDMode. Wie oben beschrieben, die Datei "lvdm.pwn" öffnen. Nun seht ihr ein fertiges Script im Editor. Ich werde anfangen, von oben bis unten alles was zum Basiswissen gehört zu erklären.

```
#include <a_samp>
```

Ein "#include" fügt dem Script eine Datei hinzu, die den Kern der PAWN Language ergänzen, diese Dateien befinden sich im Ordner "pawno/include". "a_samp" wird bei jedem Script benötigt, diese Datei enthält alle wichtigen Deklarationen für ein SA:MP Script.

```
#define COLOR_GREY 0xAFAFAFAA
```

Ein "#define" definiert einen Wert durch eine Kennzeichnung, die später im Script benötigt wird. Der Ausdruck "COLOR_GREY" wurde durch den daneben stehenden RGBA Farben Code definiert. Um z.B. das Startgeld in diesem Script zu verändern, müsst ihr einfach den Wert bei "PocketMoney" verändern.

```
new Float:gRandomPlayerSpawns[23][3] = {
{1958.3783,1343.1572,15.3746},
{2199.6531,1393.3678,10.8203},
{2483.5977,1222.0825,10.8203},
{2637.2712,1129.2743,11.1797},
{2000.0106,1521.1111,17.0625},
{2024.8190,1917.9425,12.3386},
{2261.9048,2035.9547,10.8203},
{2262.0986,2398.6572,10.8203},
{2244.2566,2523.7280,10.8203},
{2335.3228,2786.4478,10.8203},
{2150.0186,2734.2297,11.1763},
{2158.0811,2797.5488,10.8203},
{1969.8301,2722.8564,10.8203},
{1652.0555,2709.4072,10.8265},
{1564.0052,2756.9463,10.8203},
{1271.5452,2554.0227,10.8203},
{1441.5894,2567.9099,10.8203},
{1480.6473,2213.5718,11.0234},
{1400.5906,2225.6960,11.0234},
{1598.8419,2221.5676,11.0625},
{1318.7759,1251.3580,10.8203},
{1558.0731,1007.8292,10.8125},
```

```
//{-857.0551,1536.6832,22.5870}, Out of Town Spawns
//{817.3494,856.5039,12.7891},
//{116.9315,1110.1823,13.6094},
//{-18.8529,1176.0159,19.5634},
//{-315.0575,1774.0636,43.6406},
{1705.2347,1025.6808,10.8203}
};
```

Ein "new" erstellt eine neue Variable, in der wie hier Spawn-Punkte vorkommen können. Wenn ihr die zufälligen Start-Punkte eurer Spieler in diesem Mode verändern wollt, müsst ihr im Debug-Modus die Koordinaten herausfinden und hier einfügen. Dies wären die X, Y und Z Werte, die nacheinander in einem Datensatz angereiht sind. Ein // lässt die Zeilen im Spiel nicht wirksam werden, diese werden garnicht erst mit compiliert, sie sind sozusagen nur ein Vermerk. Wenn ihr Daten hinzufügen wollt, müsst ihr oben im [23] die Zeilenanzahl angeben (Also die Menge der Datensätze).

```
print("n-----");
print(" Running LVDM ~MoneyGrubn");
print("      Coded By");
print("      Jax");
print("-----n");
```

Dieser Text wird in eurer samp-server.exe Datei angezeigt.

```
public OnPlayerConnect(playerid)
{
    GameTextForPlayer(playerid,"~w~SA-MP: ~r~Las Venturas ~g~MoneyGrub",5000,5);
    SendPlayerFormattedText(playerid, "Welcome to Las Venturas MoneyGrub, For help ty
pe /help.", 0);
    gActivePlayers[playerid]++;
    gLastGaveCash[playerid] = GetTickCount();
    return 1;
}
```

Diese Funktion (public-type) wird dann aufgerufen, wenn ein Spieler den Server betritt "playerid" ist die Variable, in der die Spieler-ID vorkommt. In "GameTextForPlayer" könnt ihr z.B. euren Text schreiben, der beim Connecten dem Spieler gezeigt wird, der den Server betritt. Generell gilt, dass eine Funktion mit mehr als einer Anweisung mit { geöffnet und mit } geschlossen wird.

```
public OnPlayerCommandText(playerid, cmdtext[])
{
    cmd = strtok(cmdtext, idx);

    if(strcmp(cmd, "/help", true) == 0) {
        SendPlayerFormattedText(playerid,"Las Venturas Deathmatch: Money Grub Coded
```

```

By Jax and the SA-MP Team.",0);
    SendPlayerFormattedText(playerid,"Type: /objective : to find out what to do in this
gamemode.",0);
    SendPlayerFormattedText(playerid,"Type: /givecash [playerid] [money-
amount] to send money to other players.",0);
    SendPlayerFormattedText(playerid,"Type: /tips : to see some tips from the creator
of the gamemode.", 0);
    return 1;
}
}

```

Einer der wichtigsten Funktionen (wahrscheinlich die wichtigste) im Spiel, "OnPlayerCommandText". Wenn z.B. /help eingegeben wird, wird der Text angezeigt, der in "SendPlayerFormattedText" vorkommt. Nun kannst du eigene Commands erstellen, kopier dieses ein Beispiel, ersetze das /help durch deinen Befehl und den entsprechenden Inhalt von "SendPlayerFormattedText" durch deinen Text.

```

public OnGameModeInit()
{
    SetGameModeText("Ventura's DM~MG");

    ShowPlayerMarkers(1);
    ShowNameTags(1);

    AddPlayerClass(280,1958.3783,1343.1572,15.3746,270.1425,0,0,24,300,-
1,-1);
    AddPlayerClass(281,1958.3783,1343.1572,15.3746,270.1425,0,0,24,300,-
1,-1);
    AddPlayerClass(282,1958.3783,1343.1572,15.3746,270.1425,0,0,24,300,-
1,-1);
    AddPlayerClass(283,1958.3783,1343.1572,15.3746,269.1425,0,0,24,300,-
1,-1);

    AddStaticVehicle(550,1555.2734,2750.5261,10.6388,91.7773,62,62);
    AddStaticVehicle(535,1455.9305,2878.5288,10.5837,181.0987,118,1);
    AddStaticVehicle(477,1537.8425,2578.0525,10.5662,0.0650,121,1);
    AddStaticVehicle(451,1433.1594,2607.3762,10.3781,88.0013,16,16);
    AddStaticVehicle(603,2223.5898,1288.1464,10.5104,182.0297,18,1);
    AddStaticVehicle(558,2451.6707,1207.1179,10.4510,179.8960,24,1);
    AddStaticVehicle(550,2461.7253,1357.9705,10.6389,180.2927,62,62);

    return 1;
}

```

Diese Funktion wird direkt aktiviert, wenn ihr euren Server mit samp-server.exe bzw. samp02Xsvr startet. Sie beinhaltet alle Spielfiguren, Autos, Pickups und vieles andere. In diesem Mode sind bereits alle möglichen Spielfiguren eingefügt, also kann man lediglich welche entfernen.

```
AddPlayerClass(classid,x-wert,y-wert,z-wert,a-  
wert,waffe1,munition1,waffe2,munition2,waffe3,munition3);
```

Die Daten können im Debug-Modus erhalten werden. Eine Liste der Waffen-ID's findet ihr auf wiki.sa-mp.com. Kommen wir zu einem begehrten Punkt, dem Hinzufügen von Fahrzeugen. Dabei sind die beiden Regeln zu beachten:

1. Nicht mehr als ~700 "AddStaticVehicle" Anweisungen
2. Nicht mehr als ~70 verschiedene Vehicles-Types (Erster Wert in "AddStaticVehicle")

```
AddStaticVehicle(vehicleid,x-wert,y-wert,z-wert,a-wert,farbe1,farbe2);
```

Die Daten können im Debug-Modus erhalten werden. Dies werde ich gleich noch näher erläutern.

```
SetGameModeText("Ventura's DM~MG");
```

Hier kannst du den Namen des GameModes der im SA:MP Client angezeigt wird angeben.

Schritt 5: *Die wichtigsten Befehle*

```
SendClientMessage(playerid, COLOR_GREEN, "Test");
```

playerid ~ An wen soll die Nachricht gesendet werden?

COLOR_GREEN ~ Farbe des Textes, durch ein #define zu definieren.

"Text" ~ Der Inhalt des zu sendenden Textes.

```
SendClientMessageToAll(COLOR_GREEN, "Test");
```

COLOR_GREEN ~ Farbe des Textes, durch ein #define zu definieren.

"Text" ~ Der Inhalt des zu sendenden Textes.

```
GameTextForPlayer(playerid, "~r~ H ~b~ E ~g~ H ~y~ E",1000,5);
```

playerid ~ An wen soll die Nachricht gesendet werden?

"~r~ H ~b~ E ~g~ H ~y~ E" ~ Der Inhalt des zu sendenden Textes.

~n~ Neue Zeile

~r~ Farbe Rot

~g~ Farbe Grün

~y~ Farbe Gelb

~b~ Farbe Blau

~l~ Farbe Schwarz

~p~ Farbe Pink

1000 ~ Zeit in ms wie lange der Text erscheinen soll

5 ~ Textstyle

Die verschiedenen Styles findet ihr hier: [General Reference \[sa-mp.com\]](http://General Reference [sa-mp.com])

```
GameTextForAll("~r~ H ~b~ E ~g~ H ~y~ E",1000,5);
```

siehe oben

IsPlayerAdmin(playerid)

playerid ~ Wer soll den Admin-Check durchmachen?

```
if (IsPlayerAdmin(playerid))
{
// Wenn der Spieler Admin ist, passiert das was hier steht.
}
```

SetPlayerHealth(playerid, 50.0);

playerid ~ Wer soll Leben verlieren/erhalten?

50.0 ~ HP-Wert, 100.0 ist maximum, 0.0 ist minimum. (Wobei der Spieler bei 0.0 stirbt)

Kick(playerid); - Ban(playerid);

playerid ~ Wer soll gekickt/gebannt werden?

GivePlayerMoney(playerid, 100);

playerid ~ Wer soll Geld erhalten/verlieren?

100 ~ Geldsumme die der Spieler erhält (negative Werte möglich)

ResetPlayerMoney(playerid);

playerid ~ Wer soll wieder 0\$ haben?

GivePlayerWeapon(playerid,28,300);

playerid ~ Wer soll eine Waffe erhalten?

28 ~ Waffenid der Waffe die der Spieler erhalten soll.

300 ~ Munition der Waffe die der Spieler erhalten soll.

Eine WaffenID Liste findet ihr auf wiki.sa-mp.com

Um eine Funktion für alle Spieler gültig zu machen, sollte es so aussehen:

```
for (new i; i < MAX_PLAYERS; i++)
{
if (IsPlayerConnected(i))
{
// Hier die Funktion schreiben. Beachtet, dass ein playerid nun durch ein i ersetzt werden muss.
SetPlayerHealth(i,0.0);
}
}
```

Ergänzungen zu diesem Tutorial:

- [if & if-else-Anweisungen](#)

Mit Erlaubnis des Autors wurde dieses Tutorial auf
<http://www.GTAvision.com> hochgeladen.

Weitere Tutorials rund um GTA findet ihr dort im Tutorialbereich.