# XorCrypt DLL

Version 1.02

Mit dieser DLL lassen sich Dateien mit einem Passwort verschlüsseln und zusammen in eine einzelne Datei packen. Die Daten werden nicht komprimiert wodurch diese DLL vor allem für ohnehin schon komprimierte Dateien wie JPG oder MP3 geeignet ist.

Hier findest du eine Liste aller Funktionen und wie sie im Game Maker verwendet werden. Importiere die Scripte der beiliegenden *XorCrypt.gml* um die unten stehenden Funktionen nutzen zu können.

## Format von Datei- und Ordnerangaben

Alle Datei- und Ordnerangaben, die in den unten stehenden Funktionen verlangt werden, müssen einen / (Slash) oder \\ (zwei Backslash) als Trennzeichen der Ordner verwenden. Ein einfacher \ (Backslash) darf nicht verwendet werden. Bei Ordnerangaben darf kein Trennzeichen (Slash) am Ende der Zeichenkette stehen. Ein relativer Pfad beginnt mit *./*. Der aktuelle Ordner wird mit *.* angesprochen.

## Einstellungen

`xor_crypt_init()`

> Muss einmal aufgerufen werden bevor die anderen Funktionen verwendet werden können.

`xor_crypt_free()`

> Sollte aufgerufen werden, wenn die DLL während der Laufzeit nicht mehr benötigt wird um Speicher frei zu geben.

`xor_crypt_password(`*pw*`)`

> Damit wird das Passwort *pw* für die Ver- und Entschlüsselung festgelegt. Voreingestellt ist eine leere Zeichenkette, die bewirkt, dass die Dateien nur kopiert und nicht verschlüsselt werden.

## Dateien verschlüsseln

`xor_encrypt_add(`*file*`)`

> Fügt die Datei *file* zu einer Liste hinzu welche später verschlüsselt und in eine Datei gepackt werden.

`xor_encrypt_dir(`*dir*`,`*subdir*`)`

Fügt alle Dateien in dem Ordner *dir* zur Liste hinzu. *subdir* gibt an ob auch alle Dateien in Unterordnern hinzugefügt werden sollen. Die Funktion gibt zurück, ob das Hinzufügen erfolgreich war.

`xor_encrypt_clear()`

Löscht die Liste der zu verschlüsselnden Dateien.

`xor_encrypt_start(`*`file`*`)`

Startet den Kodiervorgang und packt alle in der Liste vorhandenen Dateien in die Datei *file* und verschlüsselt sie mit dem festgelegten Passwort. Die Liste wird danach automatisch geleert. Die Funktion gibt zurück, ob das Archiv erfolgreich erstellt wurde.

Folgendes Beispiel verschlüsselt die beiden Dateien *image.jpg* und *music.mp3* mit dem Passwort *geheim* in die Datei *ressource.pak*:

```
xor_crypt_init();
xor_crypt_password("geheim");
xor_encrypt_add("image.jpg");
xor_encrypt_add("music.mp3");
xor_encrypt_start("ressource.pak");
```

## Dateien entschlüsseln

`xor_decrypt_file(`*`file`*`,`*`index`*`,`*`dir`*`)`

Entschlüsselt die Datei an Position *index* aus der Datei *file* in den Ordner *dir* mit dem festgelegten Passwort. Die erste Position ist 0. Die Funktion gibt zurück, ob die Datei erfolgreich entpackt wurde.

`xor_decrypt_range(`*`file`*`,`*`from`*`,`*`to`*`,`*`dir`*`)`

Entschlüsselt alle Dateien von Position *from* bis Position *to*, einschließlich der beiden Grenzen, aus der Datei *file* in den Ordner *dir* mit dem festgelegten Passwort. Die erste Position ist 0. Die Funktion gibt zurück, ob die Dateien erfolgreich entpackt wurden.

`xor_decrypt_all(`*`file`*`,`*`dir`*`)`

Entschlüsselt alle Dateien aus der Datei *file* in den Ordner *dir* mit dem festgelegten Passwort. Es wird zurückgegeben, ob alle Dateien erfolgreich entpackt wurden.

Folgendes Beispiel entpacked die Datei *music.mp3* aus dem oberen Beispiel in den aktuellen Ordner (in dem sich das Spiel befindet):

```
xor_crypt_password("geheim");
xor_decrypt_file("ressource.pak",1,".");
```

## Eigenschaften abrufen

`xor_crypt_check(`*`file`*`)`

Überprüft ob *file* eine verschlüsselte Datei und fehlerfrei ist und gibt in diesem Fall *true* zurück, anderenfalls *false*. Um ein Archiv sicher mit den obigen Funktionen zu entpacken, sollte mit dieser Funktion zuvor die zu entpackende Datei geprüft werden.

`xor_crypt_size(`*`file`*`)`

Gibt die Anzahl an Dateien wieder, die sich in der verschlüsselten Datei *file* befinden. Im Fehlerfall wird -1 zurückgegeben.

`xor_crypt_file_name(`*`file`*`,`*`index`*`)`

Gibt den Namen der Datei an Position *index* in der verschlüsselten Datei *file* wieder. Im Fehlerfall wird eine leere Zeichenkette zurückgegeben.

`xor_crypt_file_size(`*`file`*`,`*`index`*`)`

Gibt die Größe der Datei an Position *index* in der verschlüsselten Datei *file* in Bytes wieder. Gibt im Fehlerfall -1 zurück.