MemLib Doku
Author: RobbiTheFox

Abhängigkeiten
MemLib funktioniert in allen Versionen der Gold Edition.
`SP EXT` oder `MP` wird *nicht* benötigt, wird aber zur Verbesserung der performance verwendet falls vorhanden. Gleiches gilt für den `S5Hook`.
An Kompatibilität mit den Versionen der History Edition wird gearbeitet.

Verwendung
Um die MemLib zu laden, muss einmalig der Befehl

```
Script.Load("maps\\user\\EMS\\tools\\s5CommunityLib\\lib\\MemLib\\MemLib.lua")
```

oder

```
mcbPacker.require("s5CommunityLib/Lib/MemLib/MemLib")
```

aufgerufen werden.
Für die Verwendung von `mcbPacker` siehe `s5CommunityLib\packer\devLoad.lua`

Weitere Module können danach mit

```
MemLib.Load(_Module1, _Module2 … _ModuleN)
```

geladen werden.
Abhängigkeiten werden hierbei automatisch mitgeladen.

Beispiel

```
MemLib.Load("Entity", "Widget")
```

Lädt das Module für allgemeine entities und für widgets.
Der Modulname entspricht immer dem Dateinamen.

Anleitung

`MemLib.GetMemory(_Address)`

Diese Funktion arbeitet genau wie `CUtilMemory.GetMemory` oder `S5Hook.GetRawMem` und gibt ein Objekt zurück, das einen Pointer auf eine Adresse darstellt.
Dieser Pointer kann mit `[_Offset]` derefferenziert werden, und gibt einen neuen Pointer auf die derefferenzierte Adresse zurück.
Mit `:GetInt()`, `:SetInt(_Value)`, `:GetFloat()` und `:SetFloat(_Value)` kann der Wert an dieser Adresse gelesen und geschrieben werden.
Es ist auch möglich, einzelne bytes zu lesen und zu schreiben.
Hierfür wird `:GetByte(_Offset)`, `:SetByte(_Offset, _Value)` verwendet, wobei `_Offset` in bytes angegeben wird.
Den Pointer selbst zu verschieben, ohne ihn zu derefferenzieren, ist mit `:Offset(_Offset)` möglich. In dem Fall wird `_Offset` auf die interne Adresse des Pointers addiert.

Vorsicht

Im gegensatz zu fast allen anderen in der `MemLib` verfügbaren Funktionen, ist `GetMemory` nicht asserted!
Fehlerhafte Eingaben können zu unerwartetem Verhalten bis hin zum Absturz des Spiels führen.

`MemLib.EntityIterator.Iterator(...)`

Der EntityIterator funktioniert ebenfalls genau wie seine Gegenstücke `CEntityIterator.Iterator` oder `S5Hook.EntityIterator`

Beispiel

```
for id in
MemLib.EntityIterator.Iterator(MemLib.EntityIterator.IsBuilding())
do
      ...
end
```

Iteriert über alle Gebäude auf der Karte.

Mit der `MemLib` ist es möglich, eigene Filter zu schreiben. Diese müssen einer bestimmten Form folgen. Wichtig ist, dass die zurückgegebene Funktion einen Parameter verlängt, nämlich die `_EntityId` und einen boolean als return ausgibt.

```
functin MyFilter(_MyParams)
      return function(_EntityId)
            return MyConditions(_EntityId, _MyParams)
      end
end
```

`MemLib.Widget.Create(_WidgetConfig)`

Erstellt ein neues widget. `_WidgetConfig` ist sehr stark an die xml Definition von widgets angelehnt.
Ein Beispiel für einen GfxButton stellt `MyGfxButtonWidget.lua` dar und kann 1:1 verwendet werden.