KMOD Developer Manual

"Kai's Modular Interface" (Formerly KMODG) is a Sandbox user-generated experience written in the GODOT engine. This contains the basics you need to create mods. And this comes with KMOD.

1.0 - Managing Scripts

This will show you how to begin your journey with creating Mods for KMOD.

KMOD uses Godot Engine for it's core scripting, with an external Mod API referencing multiple handlers & Overrides (since 0.0.3)

1.1 - Project Format

The basic KMOD project format is as follows:

```
KModProject ---
    KModProject.pck ---
        KModProject.gd
        (optional):
        README.txt
        docs
        etc ...
```

A DLC is like so:

```
KModProject.pck ---
        KModProject.tscn
        KModProject.gd
        (optional):
        README.txt
        docs
        etc ...
```

The reason for this is to prevent similar naming conventions (only one addon) (Bug fix)

1.2 - Player projects

There are very limited types of projects, However, they are ever so slowly expanding. you can use the FireOverride event:

```
var soundGun = load("res://gunsound.wav")

var OVERRIDES = "GunFire"

func FireOverride(audioplayer: AudioStreamPlayer):
    audioplayer.stream = soundGun
    audioplayer.play()
    audioplayer.stop()
```

Speed Overrides:

```
var OVERRIDES = "PlayerSpeed"

var speed_enhance = true

var _speed = 3
```

Gravity Overrides:

```
var OVERRIDES = "PhysGravity"

var gravity = 0.6
```

Jumping Power Overrides:

```
var OVERRIDES = "JumpPower"

var jump_power = 100
```

Sliding SPEED Overrides

```
var OVERRIDES = "SlidingSpeed"
var speed = 16
```

Sliding Event hook Overrides

```
func OnSlide(camera, player):
    print("Sliding!")
```

Which are checked by certain compartments of the Player (hence, why they're added into /KMOD/Player) as player scripts.

1.3 - KMOD SDK

From 0.0.9 on the KMOD SDK is now bundled in with your installation of KMOD, all mods are backwards compatible for now from 0.0.8, so you can use legacy mods

The KMOD SDK is the KMOD game packed into a Resource File which you can use to create mods and games with KMOD, kind of like half life.

DLCs are maps (Separate games) made in the KMOD universe.

It starts with a first person controller with guns, and variables.

Building a simple KMOD SDK DLC is like so:

```
extends Spatial

# Called when the node enters the scene tree for the first time.


func _ready():
    # Load the SDK
    ProjectSettings.load_resource_pack("res://kmodSDK.pck")

    # the path of the the Player object (for FPS)
    var player = load("res://Controls/FPS.tscn").instance()

    # Set the player transform to the player start node
    player.global_transform.origin = $player_start.global_transform.origin
    player.allow_guns = false

    # add the player to the map
    add_child(player)
```

Place this as the script of your scene (named your project), then build your project by going to Export... > Windows Desktop > Zip/PCK

Copy your map.pck (or your name) to the C:/KMOD/Maps/map.pck (or your name map) Then load it in the DLC loader.