USER MANUAL FOR MY

PYTHON GAME MODULE

Onni Kolkka

Sisällysluettelo

[Overview of the project 1](#_Toc122197394)

[EngineType (metalcass) 2](#_Toc122197395)

[(**EngineType**) self.log(message, color1, clolor2, …) 2](#_Toc122197396)

[GameToTK (EngineTypeSingleton) 3](#_Toc122197397)

[Multip 3](#_Toc122197398)

# Overview of the project

The main restriction was to only use pythons builtins, so no external packages. Few other restrictions was that no Global variables, only constants. Also I assume no runtime package imports (pip and TCL runtime package imports) or writing code in other languages, as these would completely undermine every other restriction.

So the only libraries I was allowed to use were (according to this website <https://www.tutorialsteacher.com/python/python-builtin-modules>):

IPython, \_weakrefset, heapq, secrets, \_\_future\_\_, \_winapi, hmac, select, \_abc, abc, html, selectors, \_ast, aifc, http, setuptools, \_asyncio, antigravity, idlelib, shelve, \_bisect, argparse, imaplib, shlex, \_blake2, array, imghdr, shutil, \_bootlocale, ast, imp, signal, \_bz2, asynchat, importlib, simplegeneric, \_codecs, asyncio, ind, site, \_codecs\_cn, asyncore, inspect, six, \_codecs\_hk, atexit, io, smtpd, \_codecs\_iso2022, audioop, ipaddress, smtplib, \_codecs\_jp, autoreload, ipython\_genutilssndhdr, \_codecs\_kr, backcall, itertools, socket, \_codecs\_tw, base64, jedi, socketserver, \_collections, bdb, json, sqlite3, \_collections\_abcbinascii, keyword, sre\_compile, \_compat\_pickle, binhex, lib2to3, sre\_constants, \_compression, bisect, linecache, sre\_parse, \_contextvars, builtins, locale, ssl, \_csv, bz2, logging, stat, \_ctypes, cProfile, lzma, statistics, \_ctypes\_test, calendar, macpath, storemagic, \_datetime, cgi, mailbox, string, \_decimal, cgitb, mailcap, stringprep, \_distutils\_findvschunk, marshal, struct, \_dummy\_thread, cmath, math, subprocess, \_elementtree, cmd, mimetypes, sunau, \_functools, code, mmap, symbol, \_hashlib, codecs, modulefinder, sympyprinting, \_heapq, codeop, msilib, symtable, \_imp, collections, msvcrt, sys, \_io, colorama, multiprocessing, sysconfig, \_json, colorsys, netrc, tabnanny, \_locale, compileall, nntplib, tarfile, \_lsprof, concurrent, nt, telnetlib, \_lzma, configparser, ntpath, tempfile, \_markupbase, contextlib, nturl2path, test, \_md5, contextvars, numbers, tests, \_msi, copy, opcode, textwrap, \_multibytecodec, copyreg, operator, this, \_multiprocessingcrypt, optparse, threading, \_opcode, csv, os, time, \_operator, ctypes, parser, timeit, \_osx\_support, curses, parso, tkinter, \_overlapped, cythonmagic, pathlib, token, \_pickle, dataclasses, pdb, tokenize, \_py\_abc, datetime, pickle, trace, \_pydecimal, dbm, pickleshare, traceback, \_pyio, decimal, pickletools, tracemalloc, \_queue, decorator, pip, traitlets, \_random, difflib, pipes, tty, \_sha1, dis, pkg\_resources, turtle, \_sha256, distutils, pkgutil, turtledemo, \_sha3, doctest, platform, types, \_sha512, dummy\_threading, plistlib, typing, \_signal, easy\_install, poplib, unicodedata, \_sitebuiltins, email, posixpath, unittest, \_socket, encodings, pprint, urllib, \_sqlite3, ensurepip, profile, uu, \_sre, enum, prompt\_toolkit, uuid, \_ssl, errno, pstats, venv, \_stat, faulthandler, pty, warnings, \_string, filecmp, py\_compile, wave, \_strptime, fileinput, pyclbr, wcwidth, \_struct, fnmatch, pydoc, weakref, \_symtable, formatter, pydoc\_data, webbrowser, \_testbuffer, fractions, pyexpat, winreg, \_testcapi, ftplib, pygments, winsound, \_testconsole, functools, queue, wsgiref, \_testimportmultiplegc, quopri, xdrlib, \_testmultiphase, genericpath, random, xml, \_thread, getopt, re, xmlrpc, \_threading\_localgetpass, reprlib, xxsubtype, \_tkinter, gettext, rlcompleter, zipapp, \_tracemalloc, glob, rmagic, zipfile, \_warnings, gzip, runpy, zipimport, \_weakref, hashlib, sched, zlib,

I used only a fraction of those modules. I used following modules (may not be up to date)

higher-level modules:

tkinter, threading, multiprocessing, abc, functools, cProfile, pstats, unittest, logging, time, platform, math

lower-level modules:

sys, pathlib, winsound, socket, sha512, ctypes

# EngineType (metalcass)

Metaclass containing virtually everything. Implements useful stuff every class should have access to.

## (**EngineType**) self.log(message, color1, clolor2, …)

Used to print colorful log messages

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

## (**EngineType**) self.is\_windows

Builtin modules don’t have a lot of crossplatform solutions, especially for sound and mouse control.

I’m using modules called “winsound” and “ctypes.windll”, so I assume trying to access those with linux or mac won’t work.

That’s why I need this check before using them.

# GameToTK (EngineTypeSingleton)

Interface between the game and TK. Mostly used by inputs and rendering.

## Multip