Zeile	Problem	Beschreibung & Lösur	g																
Diverse	Undefined V	Undefined Variat Die Variable last_orientation in mp_graphics4d_screenmode wird verwendet, ist aber nicht definiert. Lösung: Fügen Sie eine globale oder Modul-Level-Definition hinzu, z.B. static int last_orientation = 0; außerhalb der Funktionen.																	
Diverse	Typo mo_co	nst_l In mp_graphics4d_clip	_window wird mo_co	st_bool verwendet	Dieser Makro existi	ert nicht. Lösung: E	rsetzen Sie alle V	orkommen durch	das korrekte Mic	oPython-Makro: r	mp_const_bool.								
Zeile 259	Undefined V	ariat In mp_graphics4d_mo	ve_rel wird success v	erwendet, ohne del	lariert zu sein. Lösu	ng: Deklarieren Sie	die Variable lokal	bool success = :	self->gfx->MoveR	el(x_off, y_off);.									
polyline, polygo	n Falsche Fun	Diese Wrapper (mp_g a) Sie verwenden MP b) Die C-Funktion ben ktion c) Es fehlen lokale De	DEFINE_CONST_FU btigt self_in als erstes	JN_OBJ_4, obwohl Argument: STATIC	mp_obj_t mp_graph	ics4d_polyline(mp_	obj_t self_in, mp_	obj_t vx_obj,)	P_DEFINE_CONS (5 Argumente).	T_FUN_OBJ_5 vi	erwenden.								
Funktion	Problem	Beschreibung & Lösur	g																
mp_graphics4d mp_graphics4d		umer Die Makros MP_DEFI	NE_CONST_FUN_O	J_VAR_BETWEEN	sind mit 4, 5 definie	rt, was 4 bis 5 Ben	utzerargumenten	entspricht. Da die	Funktionen 4 Pfl	chtargumente (x1	/x, y1/x2, y2/y,	color) und 1 optio	nales (draw_fb) ha	en, sollte die Def	nition 5, 6 sein, un	n die korrekte Feh	lerbehandlung des	s Makros zu nutzer	١.
mp_graphics4d mp_graphics4d		umer Die optionalen Argum	ente werden mit (n_ar	gs > 5) geprüft. Bei	korrigierter FUN_OE	J Definition (5, 6) is	st das korrekt für d	das 6. Argument (args[5]). Bleibt ko	rrekt, wenn die M	akro-Definitione	en oben korrigiert	verden.						
mp_graphics4d mp_graphics4d		umer Der Check bool draw_	fb = (n_args > 7) ? m	_obj_is_true(args[6]) : true; ist fehlerha	ft. Bei 6 (Arc) bzw.	7 (ArcF) Pflichtarg	jumenten + 1 opti	ionales Argument	ist das maximale	n_args 7 bzw. 8	3. Die Prüfung mu	ss lauten: (n_args	N_MANDATOR	_ARGS) -> (n_arg	gs > 6) für arc und	I (n_args > 7) für a	rc_filled.	
mp_graphics4d	_ Duplizierte _	obj I Am Ende der Datei wi	d die Funktion mp_gr	aphics4d_circle def	niert, aber der Makr	o-Aufruf verwendet	mp_graphics4d_c	circle_filled_obj a	nstatt eines neuer	mp_graphics4d_	circle_obj. Lösu	ung: Definieren S	STATIC MP_DEF	INE_CONST_FU	_OBJ_VAR_BET	WEEN(mp_graph	ics4d_circle_obj, 5	5, 6, mp_graphics4	d_circle);

Nächste Schritte (inkl. Touch)

Ihr Ansatz, die Touch-Funktionalität in einer separaten Klasse zu kapseln, ist optimal. Der nächste logische Schritt wäre die Implementierung der Touch-Klasse, wie in den mp_touch4d_...-Headern vorbereitet, unter Berücksichtigung der folgenden Struktur:

Definieren der Touch Struct (mp_obj_touch_t).

Implementieren der Factory-Methode (mp_touch_make_new), die die C++ GraphicsTouch4D-Instanz (oder den Pointer zur Hauptinstanz Graphics4D zur Interaktion) hält.

Implementieren aller Touch-Methoden (initialize, getPoints, getX, etc.) als Methoden der Touch-Klasse.

Binden der Methoden an das mp_touch_locals_dict_table.

Beheben aller oben genannten kritischen Fehler.