



1 Was ist LuaTileArranger?

LuaTileArranger (kurz: LuTA) ist ein Programm um Maps für Spiele oder Programme zu erstellen und zu verwalten.

2 Steuerung Allgemein:

[D-Pad]	...	...	Bewegt den Cursor
[A-Pad]	...	...	Auswahl im Speichern-Dialog
[Start]	...	...	Öffnet das Hauptmenü / Speichert bzw. Ladet
[Select]	...	...	ESC
[Cross]	...	...	Bestätigen/ Zeichnen
[Square]	...	...	Gedrückt halten um mit [D-Pad] den Cursor flüssig zu bewegen
[Triangle]	...	...	Radieren / Radierfarbe in der Brushauswahl setzen
[R-Trigger]	...	...	Öffnet die Blockauswahl

3 Menüs:

Die Navigation ist in allen Menüs gleich. Mit dem [D-Pad] kann man die Auswahl links bzw. rechts bewegen.



1. Aktiver Menüpunkt
2. Auswahl, bestätigen mit [Cross]

3.1 Das Hauptmenü:

Das Hauptmenü wird mit [Start] aufgerufen und besteht aus vier bereichen:



3.1.1 Werkzeuge:

	Stift	...	...	Zeichenwerkzeug. Betätigen mit [Cross] radieren mit [Triangle]
	Radierer	...	...	Radierwerkzeug. Betätigen mit [Cross]
	Füllen	...	...	Füllt alle(!) gleichen Blöcke. Betätigen mit [Cross]
	Pipette	...	...	Auswahl mit [X]. Wechselt sofort in den Stiftmodus

3.1.2 Funktionen:

	Tileauswahl	...	Wählt die Tiles für die einzelnen Ebenen aus.
	Raster	...	Schaltet den Hilfsraster An/ Aus
	Ebene	...	Schaltet durch die 3 Ebenen
	Sichtbarkeit	...	Sichtbarkeit einer Ebene An/ Aus
	USB	...	Schaltet die USB-Verbindung An/ Aus

3.1.3 Dateiverwaltung:

	Neu	...	Starte neues Projekt
	Öffnen	...	Öffnet ein Projekt
	Speichern	...	Speichert ein Projekt

3.1.4 Informationen:

Ram	...	...	Gibt den zur Verfügungstehenden RAM in % an
Power	...	...	Zeigt den Akkuzustand in % an
Layer	...	...	Gibt an auf welcher Ebene man sich gerade Befindet. Ist die Schrift [schwarz] so ist sie Sichtbar, ist sie [rot] so ist sie ausgeblendet
X/ Y Informationen	...		Gibt die momentane position wieder

3.2 Blockauswahlmenü:

Das Blockauswahlmenü wird durch drücken von [R-Trigger] aufgerufen.



1. Auswahl: Bestätigen mit [Cross] für Vordergrundfarbe, oder [Triangle] für Hintergrundfarbe
2. Vordergrundfarbe: Mit Stift und [Cross] auftragen
3. Hintergrundfarbe: Mit Radierer und [Cross] auftragen, oder mit Stift und [Triangle]



4 Dialoge:

4.1 Tileauswahl:

Hier wählen sie die Tiles für die Ebenen aus, und zwar immer der Reihenfolge! Also zuerst Ebene1, dann Ebene2 und schließlich Ebene3!

Eigene Tiles müssen(!) 69x69 Pixel haben und im Format *.png in den Ordner „ms0:/PSP/GAME/LuaTileArranger/Applications/files/tiles“ kopiert werden.

4.2 Projekt Öffnen:

Wählen sie aus der Liste das gewünschte Projekt. Die Projekte befinden sich unter „ms0:/PSP/GAME/LuaTileArranger/Applications/files/projects“ und liegen in *.map vor.

4.3 Projekt Speichern:

a	'	c	d	.	f	g	!	i
b			e			h		
j	-	l	m	n	o	p	q	
			del	space				
r	(	t	u	:	w	x	)	z
s			v			y		

Mit [A-Pad] wählen sie ein Feld und je nach Buchstaben den sie erreichen wollen [Triangle] -das wäre im 1. Feld [,] | [Circle] in diesem Falle für [c] | [Cross] für [b] und [Square] für [a].

Durch drücken und halten von [R-Trigger] gelangen sie zu den Großbuchstaben

Mit [L-Trigger] wechseln sie zwischen Buchstaben und Zahlen

Drücken von [D-Pad]-[Oben] ist Backspace

7 Einfügen von LuTA Maps in andere Lua-Programme

7.1 Grundlagen:

Eine *.map Datei von LuTA besteht aus 53 Zeilen welche jeweils mit 30 Buchstaben von a - p gefüllt sind. Das Auslesen der Map ist also nichts anderes als das ersetzen eines Buchstaben durch ein Bild.

7.2 Maps laden:

Kopiert die „ms0:/PSP/GAME/LuaTileArranger/Applications/files/luatilesave.lua“ datei in euren Scripfolder und fügt sie eurem Script (z.B.: in Index.lua) mit dem befehl **dofile("luatilesave.lua")** hinzu. Vergesst nicht die Tiles (Bilder) zu laden und kopiert noch folgendes in euer Script:

```
function loadmap(file)  
    map.line = table.load(file)  
    for y = 1, 53 do  
        for x = 1, 30 do  
            loadtemp = string.byte(map.line[y], x)  
            map[y][x]=string.char(loadtemp)  
        end  
    end  
end
```

also mit sowas wie **loadmap(maps/map01.map)** kann man dann wieder eine Map laden. Zu Beachten ist noch dass die Ebene1 von y1-17, die Ebene2 von y18-35 und die Ebene3 von y36-53 reicht, wobei y die String liniennummer ist!

7.3 Tiles zerlegen:

Funktioniert mit Folgender Funktion:

```
function drawTile(tile, x, y)  
local tileX = math.mod(tile, 4)  
local tileY = math.floor(tile / 4)  
screen:blit(16 * x, 16 * y, tiles, 17 * tileX + 1, 17 * tileY + 1, 16, 16)  
end
```

6 FAQ's

Gibt's noch keine, ist der 1. offizielle Release. Fragen einfach an

fetttersplatter@gmx.at