

Zerstörbare Landschaft

by xiebel

Ok, hier also ne Möglichkeit wie man sehr einfach Zerstörbare Landschaften macht.

Was wir brauchen:

- Ein Landschaftssprite namens "spr_landscape"
- Ein "Loch"-Sprite namens "spr_hole"
- Ein Objekt namens "obj_landscape" mit spr_landscape als sprite
- Ein Objekt namens "obj_hole" mit spr_hole als sprite

Das Landschaftsobjekt muss Transparent sein, und die Transparente farbe sollte im standart-GM-transparent-grün sein (64,128,0), und wenn man doch eine andere will muss man halt paar Sachen im Code umändern

Das Loch-sprite soll ein einfacher Kreis sein, der die transparente Farbe von oben hat d.h. die eigene Transparente Farbe muss anders sein.

obj_landscape

Create:

```
global.landscape=spr_landscape; //wir legen spr_landscape als Anfangs-landschafts-sprite fest
```

press <Space>:

```
instance_create(mouse_x,mouse_y,obj_hole); // ein Loch-Objekt wird an Mausposition erstellt
```

So, jetzt haben wir die Landschaft, die wir auch in den room setzen. Beim Druck auf die Leertaste soll an der Mausposition ein Loch in der Landschaft erscheinen, bisher wird aber nur eine obj_hole-Instanz an der Mausposition erstellt.

obj_hole

collision obj_landscape:

```
change_landscape(x,y,obj_hole); //Bei einer Kollision von Loch und Gelände wird ein Script ausgeführt was wir später noch einbauen werden.
```

Create:

```
event_perform(ev_collision,obj_landscape); //das gerade definierte Kollisions-event mit obj_landscape wird ausgeführt.  
instance_destroy(); //Gleich danach die Instanz zerstört.
```

So, jetzt kommen wir zum Kernstück des ganzen, dem Skript das die Landschaft verändert:

Skript: change_landscape

```
//argument0: x-position of sprite, which is added to landscape
//argument1: y-position ...
//argument2: index of sprite
var _width, _height;
_width=sprite_get_width(global.landscape);
_height=sprite_get_height(global.landscape);
brush_color=make_color(64,128,0);
pen_color=make_color(64,128,0);
draw_rectangle(0,0,_width,_height);
draw_sprite(global.landscape,0,0,0);
draw_sprite(argument2,0,argument0,argument1);
sprite_delete(global.landscape);
global.landscape=sprite_create_from_screen(0,0,_width-1,_height-1,true,true,true,false,0,0);
obj_landscape.sprite_index=global.landscape;
```

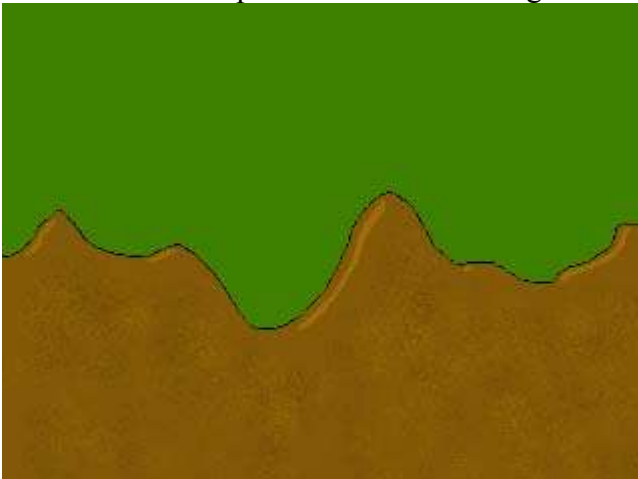
Wenn wir jetzt testen müsste es Einwandfrei funktionieren, aber was genau macht das Skript?
Da es nicht im draw-event ist wird, obwohl wir zeichnen, nichts davon auf dem Bildschirm angezeigt.

Zuerst wird die Farbe der Zeichenwerkzeuge auf die Transparente Farbe gesetzt (das hier ändern wenn man eine andere Transparente Farbe genommen hat).

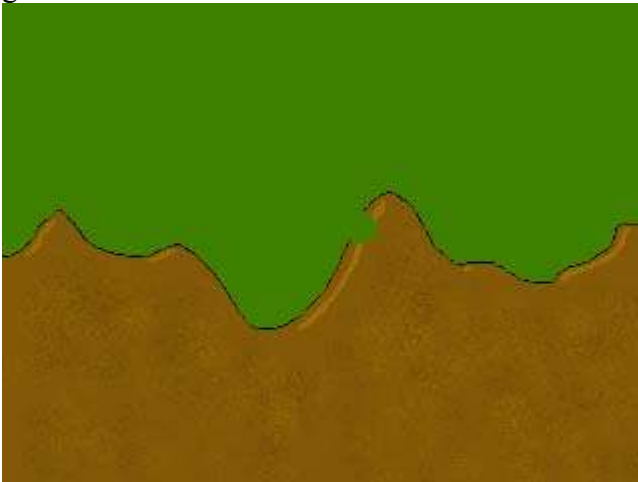
Dann wird der komplette Landschaftssprite mit grün(Transparente Farbe) übermalt. (damit Hintergrund und andere Objekte nicht im Bild sind)



Das Landschafts-sprite nochmal darüber gezeichnet.



Jetzt das Loch in der transparenten Farbe der Landschaft darüber an der Position des obj_hole's gezeichnet. Damit sieht es so aus als wäre ein Loch in der Landschaft.



Es soll aber nicht nur so aussehen, wir wollen ja das das Loch wirklich IM Landschaftssprite ist, damit man da auch durchgehen kann wenn die Landschaft solid ist.

Deshalb wird der alte Landschaftssprite aus dem Speicher gelöscht und ein neuer von dem was jetzt NICHT auf dem Bildschirm zu sehen ist (da ja nicht in draw event) per `sprite_add_from_screen` gemacht.

Sieht auch schön aus wenn man statt dem runden Loch-sprite einen strukturierten nimmt ^^

Dieses Tutorial wurde heruntergeladen von: gmaker.de – The German Game Maker's Headquarter

