Wegfindung mit den „mp_grid“ - Funktionen

von Fabian Nicolay a.k. „Glowing Orb“
Version 1.0
Benötigte GM-Version: 5.3a registriert, 6 registriert oder 7 Pro
Bearbeitet von: Cygnus und MewX

In diesem Tutorial geht es darum, wie man die „mp_grid“ – Funktionen zur Wegfindung verwenden kann.
Dazu werden wir als Beispiel ein rundenbasierendes Maze-Game erstellen, in dem der Spieler das Ziel erreichen muss, während drei Gegner versuchen ihn zu fangen.

Hierfür erstellen wir erstmal fünf Objekte inklusive Sprites (32x32 Pixel):

- obj_Wand
- obj_Gegner
- obj_Spieler
- obj_Ziel
- obj_Controller (nicht sichtbar!)

Nun bauen wir uns damit ein kleines Labyrinth, bestehend aus je einem obj_Spieler, obj_Ziel, obj_Controller, ein paar obj_Gegner und vielen obj_Wand (siehe Screenshot).

Dann Scripten wir erstmal die Bewegung für den Spieler und rufen sie vom Step Event des **obj_Spieler** auf.

scr_SpielerBewegung:

```
{
if (global.SpielerZug)
{
    switch (keyboard_key)
    {
        case vk_left: if (place_free(x-32,y)){x-=32;global.SpielerZug = false};
            break;
        case vk_right: if (place_free(x+32,y)){x+=32;global.SpielerZug= false};
            break;
        case vk_up: if (place_free(x,y-32)){y-=32;global.SpielerZug = false};
            break;
        case vk_down: if (place_free(x,y+32)){y+=32;global.SpielerZug = false};
            break;
    }
}

}
```

Außerdem scripten wir gleich noch etwas für das Room-Start Event des Controllerobjekts:

scr_RoomStart:

```
{
global.SpielerZug = true;
global.GegnerZug = false;
}
```

Später werden wir noch mehr dazuscripten.
Bevor es jetzt zum spannenden Teil geht, sollten wir das Spiel mal testen.
Wenn alles richtig war, müsstens wir uns jetzt genau einen Schritt fortbewegen können.

Jetzt öffnen wir erneut **scr_RoomStart** und fügen folgende Zeile ein:

```
global.Netz = mp_grid_create(0,0,20,15,32,32);
```

Damit erstellen wir ein Netz, bestehend aus 20x15 Feldern der Größe 32x32, dessen Koordinaten 0,0 sind.
Dieses werden wir für sämtliche mp_grid-Funktionen benötigen. Die Globale global.Netz brauchen wir, um auf das eben erstellte Netz zuzugreifen.

Zitat:
mp_grid_create(left,top,hcells,vcells,cellwidth,cellheight) *Diese Funktion erstellt das Gitter. Sie gibt einen Index zurück, der bei allen anderen Aufrufen verwendet werden muß. Du kannst mehrfache Gitter-Strukturen zur selben Zeit erstellen und verwalten. "left" und "top" geben die Position der linken oberen Ecke des Gitters an. "hcells" und "vcells" geben die Anzahl der horizontalen und vertikalen Zellen an. Schließlich geben "cellwidth" und "cellheight" die Größe der Zellen an.*

Nun erstellen wir ein Script für das Step-Event des obj_Controller:

scr_GridAktualisierung:

```
{
if (!global.SpielerZug)
{
    mp_grid_clear_all(global.Netz);
    mp_grid_add_instances(global.Netz,obj_Wand,0);
    global.GegnerZug = true;
}

}
```

Mit mp_grid_clear_all machen wir das gesamte Netz frei.
Mit mp_grid_add_instances wird jedes Feld, auf dem ein obj_Wand liegt blockiert.
Die beiden mp_grid-Funktionen hätten wir natürlich auch in ein Room_Start – Event packen können, allerdings ist es von Vorteil, dies wie hier zu machen, falls man auch zerstörbare oder bewegliche Wände einbauen will.

Zitat:

mp_grid_clear_all(id) *Markiert alle Zellen des Gitters als frei.*
mp_grid_add_instances(id,obj,prec) *Markiert alle Zellen die eine Instanz des angegebenen Objektes schneiden als verboten. Du kannst auch eine individuelle Instanz verwenden, indem du "obj" die id der Instanz zuweist. Ferner kannst du das Schlüsselwort "all" angeben, um alle Instanzen aller Objekte anzugeben. "prec" gibt an, ob präzise Kollisionsprüfung verwendet werden soll (funktioniert nur, wenn präzise Prüfung für das von der Instanz verwendete sprite aktiviert ist).*

Jetzt noch ein Script für das StepEvent der Gegner (**obj_Gegner**):

```
scr_GegnerBewegung:
{
if (global.GegnerZug)
{
    mp_grid_path(global.Netz,Pfad,x,y,obj_Spieler.x,obj_Spieler.y,0);
    path_start(Pfad,4,0,0);
    if (!InBewegung) {alarm[0] = 4};
    InBewegung = true
}
else
    path_end();
}
```

Mit mp_grid_path veranlasst man das Programm einem Pfad durch die nicht blockierten Felder des Netzes zu generieren.

Zitat:

mp_grid_path(id,path,xstart,ystart,xgoal,ygoal,allowdiag) *Berechnet einen Pfad durch das Gitter. "path" muß einen existierenden Pfad angeben, der durch den Computerpfad ersetzt wird. "xstart" und "ystart" geben den Start und "xgoal" und "ygoal" das Ziel an. "allowdiag" gibt an, ob diagonale Bewegung zulässig ist statt nur horizontaler oder vertikaler. Die Funktion gibt wieder, ob sie erfolgreich einen Pfad gefunden hat. (Beachte, daß der Pfad unabhängig von der aktuellen Instanz ist; es ist ein Pfad durch das Gitter, nicht ein Pfad für eine besondere Instanz.)*

Jetzt noch eins für das Create Event der Gegner (**obj_Gegner**):

scr_GegnerIni:

```
{
Pfad = path_add();
InBewegung = false;
}
```

Hier muss ein Pfad erstellt werden, da die mp_grid_path Funktion immer einen schon bestehend Pfad verändert und keinen neuen erstellt.

Und für das Alarm[0] Event:

scr_GegnerStop:

```
{
global.GegnerZug = false;
global.SpielerZug = true;
InBewegung = false;
}
```

Zur Veranschaulichung des Netzes kann man noch folgendes Script in das Draw-Event des obj_Controller packen (natürlich muss man dann den Controller sichtbar machen:

scr_DrawNetz:

```
{
if keyboard_check(vk_alt) {mp_grid_draw(global.Netz)}
}
```

Die Funktion mp_grid_draw zeichnet – wer hätte das gedacht – das Netz.

Zitat:

mp_grid_draw(id) *Diese Funktion zeichnet das Gitter, wobei freie Zellen grün und verbotene rot gezeichnet werden. Diese Funktion ist sehr langsam und steht nur als Fehlersuchwerkzeug zur Verfügung.*

Und fertig!

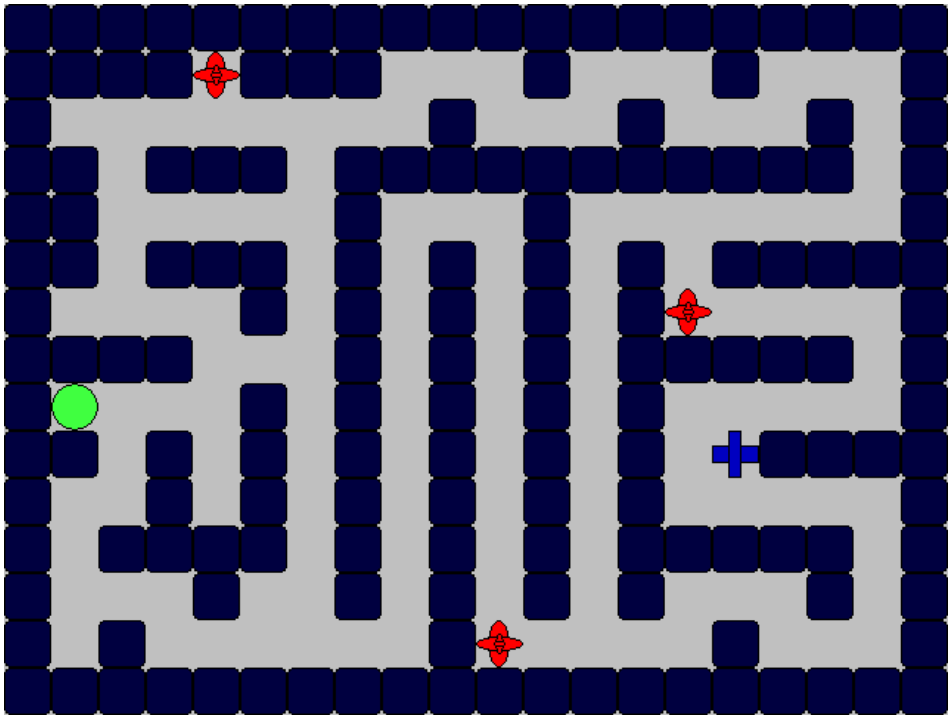
Zu guter Letzt...

Anbei ist ein Beispiel das ihr euch im Game Maker anschauen könnt.

Ihr könnt dieses Beispiel erweitern indem z.B....
...etwas passiert, wenn man das Ziel erreicht.
...etwas passiert, wenn man einen Gegner berührt.

Kleiner Bug:
- Der Pfad für die Gegnerbewegung wird leider nicht nur in der Mitte der Gänge, sondern auch an den Rändern angelegt.

So das war's auch schon für heute, euer G. O.



Dieses Tutorial wurde heruntergeladen bei:

www.gm-d.de – Deine deutschsprachige Game Maker Community