

Game2DProject

GameObject

Ein GameObject ist ein besonderes Objekt. Es beinhaltet Bilder (nur vom Format *.gif) und Animationen. Ein GameObject wird von der Map in den Arbeitsspeicher geladen. Es kann nicht einfach dargestellt werden. Hierzu benötigt man ein MapObject.

ImageObject

Der Unterschied zwischen einem ImageObject und einem GameObject ist der, dass das ImageObject nur maximal ein Bild zur Verfügung stellt. Es kann außerdem keine Animationen speichern. Um ein ImageObject zu erstellen, sagt man ihm, wo sich das zu ladende Bild befindet.

MapObject

Ein MapObject definiert ein GameObject an einer Stelle in der Map. Es ist der Container für das GameObject. Neben dem Namen werden folgende Informationen benötigt um ein MapObject zu erstellen: Das GameObject, eine X-Position, eine Y-Position, und die Information, wie das MapObject gezeichnet werden soll.

Hierzu gibt es 4 Möglichkeiten ein MapObject zu zeichnen: AlwaysOnBottom, AlwaysOnTop, DependOnMapY, Invisible.

Map

Die Map ist ein Bauplan für MapObjecte. Sie definiert, wo die MapObjecte platziert sind.

Doch zuerst muss die Map wissen, wie groß sie sein soll und wie gescrollt werden soll.

Im ersten Block werden diese Informationen definiert.

Neben der Funktion als Bauplan enthält sie ein Script.

Man unterscheidet zwischen zwei Arten von Scripten: Das Init-Script und das While-Script.

Das Init-Script wird beim Laden der Map ausgeführt. Es enthält Befehle, die der Map sagen, welche GameObjecte oder MapObjecte erstellt werden sollen. Außerdem werden noch die Map-Variablen festgelegt.

Das While-Script wird während das Spiel läuft ausgeführt. Es kann Bedingungen und Methoden enthalten.

Das Map-Script

Wie schon oben erwähnt, braucht man Methoden, um der Map zu sagen, was sie tun soll.

Hier nun eine kleine Einführung in die Script-Sprache.

Wir wollen ein Spiel entwickeln, dass ein Bild beim Tastendruck 'nach Links' das Bild nach Links verschiebt.

Zuerst müssen wir in dem Initblock das ImageObject und das MapObjecte definieren.

```
loadImageObject[ string name, string filename ];
```

Diese Zeile veranlasst, dass ein bild geladen wird. Die Zeile könnte dann so aussehen:

```
loadImageObject[io_Baum,images\baum.jpg];
```

Du musst also eine Bild mit dem Namen 'baum.jpg' haben, das im Ordner 'images' abgelegt ist. Die Map muss sich im Ordner befinden, in dem der Ordner 'images' enthalten ist, d.h. der pfad sollte immer relativ zur Map gewählt werden.

Als nächstes brauchen wir das MapObject. Es wird so geladen:

```
loadMapObject[ string name, string nameOfImageObject, int x, int y, DrawOption drawOption];
```

Wieder vergibt man einen Namen. Aber nicht nur das. Ein MapObject benötigt außerdem die Information, welches ImageObject verwendet werden soll. Danach muss man die Koordinaten festlegen. Zuletzt sagt man, wie das MapObject gezeichnet werden soll. Die Zeile könnte also so aussehen:

```
loadMapObject[mo_Baum,io_Baum,0,0,AlwaysOnBottom];
```

Nach den zwei Zeilen, kann man die Map spielen. Man wird das Bild auf der Position (0|0) sehen.

Nun wollen wir aber die Bewegung festlegen.

Zuerst müssen wir erfragen, ob die Tasten überhaupt gedrückt sind.

Das tun wir im While-Script. Wir schreiben als erstes eine Bedingung.

```
if( bedingung )  
{  
  
}
```

Die Bedingung nennt sich 'control[string key]'. Wir geben also an:

```
if( control[Left] )  
{  
  
}
```

'Left' steht für die linke Pfeiltaste. Wenn sie gedrückt wird springt das Script in den Block hinein.

Jetzt müssen wir das MapObject verschieben. Ein MapObject wird immer mit folgender Syntax angesprochen.

```
mapObject( string name ).
```

Nach dem Punkt folgt die Methode, die man aufrufen möchte.

Die Methode lautet: 'moveX[int step]'. Mit ihr wird das MapObject auf der X-Achse um eine Anzahl von step nach Links bzw. nach Rechts bewegt. Die Zeile lautet also wie folgt:

```
if( control[Left] )
{
    mapObject(mo_Baum).moveX[-2];
}
```

Wenn du jetzt das Spiel spielst, wirst du merken, dass beim Tastendruck auf Links, der Baum sich nach links bewegt.

Hier noch mal der vollständige Code:

```
map
{
    300;
    300;
    0;
    0;
}

init
{
    loadImageObject[io_Baum,images\baum.jpg];
    loadMapObject[mo_Baum,io_Baum,0,0,AlwaysOnBottom];
}

while
{
    if( control[Left] )
    {
        mapObject(mo_Baum).moveX[-2];
    }
}
```