The Deep Blue

Developer•Edition

The S.T.O.R.M. Developer News • Paris New-York • Issue 1 • 29/05/1995

STORM is dealing with parallax scrolling.

On consoles we will enjoy parallax scrolling. For now Three playfields are set up:

- The background is a 256 colors 320x400 BMP non compressed image and it's stored into VRAM . It uses 128Kb. And it's rolling twice lower left and right than the speed of the main playfield. (The name of the file is *LEVEL*__B.BMP).
- The main playfield: it's the former we used till now decor with transparency enabled. It is still stored using slices and is read on the fly directly from the CD-ROM as we are scrolling. (The name of the file is *LEVEL*___.DCR).
- The Foreground playfield: Just a small image stored into the VRAM as an uncompressed 256 320 x any height BMP file. (The name of the file is *LEVEL*__F.BMP).

For the moment, only Level.C and PSX.C has been modified for doing parallax scrolling. We expect to do it also in SATURN.C

In order to do parallax scrolling not-by-hand we have to integrate a new module in the level editor of MMDEDIT.
Unfortunately, unlike the first level, next levels are also doint vertical scrolling. This means that we have to develop also a vertical scrolling related routines. I will talk with the game designer to fix that thing.

Real world

Actors now act as living people

Real world has to deal with actual world behaviou. That is forces, power, energy and all that stuff. Unlike we did before, all actors will be moved depending on their own power and all external forces they are subject to.
So new members have been added to Tactor structure:

```
struct Tactor
{
... /* Old stuffs */

long   _mass;
long   _powerx,
       _powery,
       _powerz,
       _strengthx,
       _strengthy,
       _strengthz;

long   _timeoutx,
       _timeouty,
       _timeoutz;
}
```

- Speeds:

All speed was expressed as a pixel to move per game rate basis. But the big drawback is that we couldn't move any sprites with a very slow speed, say 1 pixel every second!
Now all speed are expressed as the number of pixels to move per second. The guy taking care about that is, I give it you, ACTDefBehaviour() default function. The way DefBehaviour() is doing this is:
1) Just calculate the speed in pixels/second.
2) Calculate the pixel increment for that frame taking care of frame duration. (Frame duration is hold by the FRAMETicks global variable). This is done for each direction (x, y, z).
Two cases can happend:
Either the pixel increment is null on that direction, then it just add the frameduration to the _timeout? member and do not change the _?pos member, or the pixel increment is not nulll, then it changes the _?pos member , doing an actual move on that direction, eventually it resets the _timeout? member.
All necessary actor behaviour source updates has been done in all actors.

- Mass:
  Every actor has now a mass. This is calculate as the number of kilobyte the bigest frame of an MMD's got. The mass must not be null, so if the size of an MMD is below one kb, the mass is set up to 1.
- Power:
  This is the force which enables an actor to move by itself. If an actor is self powered (_power? members not all null), then this means that the actor has an engine and can move by itself.
- Strenghts:
  These members hold any external forces they are stressing the actor. It is generally updated by other actors.
- Gravity:

The gravity is simulated using a gravity constant named GEOM_GRAVITY?. The force is computed using the formula:
Fx=0, Fy=_mass*GEOM_GRAVITYY, Fz=0.

– Written and Published on May 29th 1995 by Sdl –