Tutorial: Script - Timer/Indicating

Inhaltsverzeichnis

1 HUGE THANKS TO thor2950 FOR THE TRANSLATION!

2 Principle

Since the SimStep-procedure is executed on every frame, it needs to be executed as fast as possible and commands like "delay" or "sleep" are prohibited. This also applies to infinite loops, such as while(true).

To program a timer, you need to "mark" the time point at which the timer was initiated and then check if the time has run out. If this is the case, the timer is the deactivated or reset.

3 Implementation

This could look something like this in a script:

Code

```
const
TIME_S = 2; // Seconds
var
Timer: single; // If the value is negative, then the timer is inactive.
...
function RunTimer: boolean;
begin
result := false;
if Timer > 0 then
begin
Timer := Timer - Timegap; // Timegap is a system variable, which contains the amount
of time in seconds that has passed since the last call of SimStep
if Timer <= 0 then
result := true;
end;
end;
procedure SimStep;
begin
...
// If the timer should be started at some point by some event:
Timer := TIME_S;
...
// The timer is "processed"
if RunTimer then
```

```
33. begin
34. // Do whatever needs to be done after timer runout
35. // If the timer should be continuous:
36. Timer := TIME_S;
37. end;
38. ...
39. end;
40. procedure Initialize;
41. begin
42. ...
43. Timer := -1; // Set once to a negative value so the timer doesn't start
44. // OR
45. Timer := TIME_S; // start the timer directly, by setting the duration
46. ...
47. end;
```

Alles anzeigen