

```

// R_FatNumForName (SKYFLATNAME)
// If no world levels, not exit
// If (gamemode == commercial)
// (W_CheckNumForName("map31") < 0)
// secrettext = false;
// secrettext = true;
// gameaction = ga_completed;
// skip the description field
// memtest (check, 0, sizeof(check));
// sprint (check, "version %d", VERSION);
// if (strcmp (base_p, check))
// return; // bad version
// save_p += VERSIONSIZE;

// This was quite messy with SPECIAL and commented parts,
// supposedly backs to make the latest edition work.
// It might not work properly.
// episode < 1;
// episode = 1;
// if (gamemode == retail)
// {
// episode < 4;
// speed = 4;
// else if (gamemode == shareware)
// {
// episode < 3;
// episode = 1; // only start episode 1 on shareware
// }
// else
// {
// episode < 3;
// episode = 3;
// }
// if (map < 1)
// map = 1;
// if (map > 9)
// {
// if (gamemode != commercial)
// map = 9;
// M_ClearRandom ();
// }
// if (skill == sk_nightmare || respawnparm)
// {
// respawnmonsters = true;
// else
// respawnmonsters = false;
// if (respawn || skill == sk_nightmare && gameskill != sk_nightmare)
// {
// for (i=S_SARG_RUN1; i<=S_SARG_PAIN2; i++)
// mobinfo[MT_BRUISERSHOT].speed = 20*FRACUNIT;
// mobinfo[MT_TROOPSHOT].speed = 20*FRACUNIT;
// }
// else if (skill == sk_nightmare && gameskill == sk_nightmare)
// {
// for (i=S_SARG_RUN1; i<=S_SARG_PAIN2; i++)
// statesh.tics <<= 1;
// mobinfo[MT_BRUISERSHOT].speed = 15*FRACUNIT;
// mobinfo[MT_TROOPSHOT].speed = 10*FRACUNIT;
// }
// }
// force players to be initialized upon first level load
// for (i=0; i<MAXPLAYERS; i++)
// players[i].playerstate = PST_REBORN;
// usergame = true; // will be set false if a demo
// paused = false;
// demoplayback = false;
// automapactive = false;
// viewactive = true;
// gamemode = episode;
// gamemap = map;
// gameskill = skill;
// viewactive = true;

```

Lambdas and LINQ

Rasmus Lystrøm
External Associate Professor
ITU
rnie@itu.dk

Agenda

Anonymous methods

Anonymous types

Delegates

Lambda expressions

Extension methods

LINQ

Polymorphism