

mod programming voor quake III, deel 4

DE RADAR MOD

door Jan Paul van Waveren aka Mr.Elusive | Jan Paul werkt als programmeur voor ID software.

In Quake III Arena kunnen spelers niet door muren heen kijken. Echter met deze modificatie zal de positie van andere spelers in een beperkte radius wel zichtbaar worden. De radar geeft de posities van andere spelers weer als punten op het radar scherm.

Voor deze modificatie moeten in de Quake III Arena game source en client game source bestanden een aantal code regels worden toegevoegd.

In het bestand cgame/cg_local.h aan het einde van de cg_t structuur wordt het volgende toegevoegd:

```
char radarString[MAX_STRING_CHARS];
```

In hetzelfde bestand aan het einde van de cgMedia_t structuur wordt het volgende toegevoegd:

```
qhandle_t radarCircle;  
qhandle_t radarDot;
```

In het bestand cgame/cg_main.c aan het einde van de functie CG_RegisterGraphics wordt het volgende toegevoegd:

```
cgs.media.radarCircle = trap_R_RegisterShaderNoMip("gfx/radarcircle.tga");  
cgs.media.radarDot = trap_R_RegisterShaderNoMip("gfx/radardot.tga");
```

Deze code zorgt ervoor dat de plaatjes worden geladen die gebruikt worden bij het tekenen van het radar scherm.

In het bestand cgame/cg_draw.c boven de functie CG_Draw2D wordt de volgende functie toegevoegd:

```
void CG_DrawRadar( void ) {  
    int i, x, y, x2, y2;  
    float a, sina, cosa;  
    char *p;  
  
    CG_DrawPic( 480, 40, 140, 140, cgs.media.radarCircle );  
    a = (90.0f - cg.refdefViewAngles[YAW]) * (M_PI / 180.0f);  
    sina = sin(a);  
    cosa = cos(a);  
    p = cg.radarString;  
    i = 0;  
    while ( i < MAX_STRING_CHARS && p[i] ) {  
        x = y = 0;  
        while( i < MAX_STRING_CHARS && p[i] && p[i] != ';' ) {  
            x = x * 10 + p[i++] - '0';  
        }  
        if ( !p[i] ) {  
            break;  
        }  
        i++;  
        while( i < MAX_STRING_CHARS && p[i] && p[i] != ';' ) {  
            y = y * 10 + p[i++] - '0';  
        }  
        if ( p[i] ) {  
            i++;  
        }  
        x -= 1024;  
        y -= 1024;  
        if ( x * x + y * y < (1024 * 1024) ) {  
            x2 = (cosa * x - sina * y) * (70.0f / 1024.0f) + 70.0f;  
            y2 = (sina * x + cosa * y) * (-70.0f / 1024.0f) + 70.0f;  
            CG_DrawPic( 480 + x2 - 1, 40 + y2 - 1, 2, 2, cgs.media.radarDot );  
        }  
    }  
}
```

In hetzelfde bestand aan het eind van de functie CG_Draw2D wordt het volgende toegevoegd:

```
CG_DrawRadar();
```

Deze code tekent de radar op het scherm.

In het bestand cgame/cg_servercmd.c in de functie CG_ServerCommand direct boven de regel:

```
if ( !strcmp( cmd, "print" ) ) {
```

wordt het volgende toegevoegd:

```
if ( !strcmp( cmd, "radar" ) ) {  
    strcpy( cg.radarString, CG_Argv(1) );  
    return;  
}
```

Deze code zorgt ervoor dat de radar informatie voor de client wordt opgeslagen zodat deze informatie kan worden weergegeven.

In het bestand game/g_main.c direct boven de functie G_RunFrame wordt de volgende functie toegevoegd:

```
void G_UpdateRadar( gentity_t *r ) {  
    int i, l, x, y;  
    gentity_t *ent;  
    char radarString[MAX_STRING_CHARS];  
  
    l = 0;  
    radarString[0] = '\0';  
    ent = &g_entities[0];  
    for ( i = 0; i < MAX_CLIENTS; i++, ent++ ) {  
        if ( !ent->inuse || !ent->r.linked ) {  
            continue;  
        }  
        if ( !ent->client ) {  
            continue;  
        }  
        if ( ent->client->sess.sessionTeam == TEAM_SPECTATOR ) {  
            continue;  
        }  
        if ( ent == r ) {  
            continue;  
        }  
        x = ent->client->ps.origin[0] - r->client->ps.origin[0];  
        y = ent->client->ps.origin[1] - r->client->ps.origin[1];  
        if ( x * x + y * y < (1024 * 1024) ) {  
            x += 1024;  
            y += 1024;  
            l = strlen(radarString);  
            if ( radarString[l] ) {  
                Com_sprintf( &radarString[l], sizeof(radarString)-l, "%i,%i", x,y);  
            } else {  
                Com_sprintf( &radarString[l], sizeof(radarString)-l, "%i,%i", x,y);  
            }  
        }  
        trap_SendServerCommand( r - g_entities, va( "radar l"%s", radarString ) );  
    }  
}
```

In hetzelfde bestand in de functie G_RunFrame direct onder de regel:

```
G_RunClient( ent );
```

wordt het volgende toegevoegd:

```
G_UpdateRadar( ent );
```

Deze code zorgt ervoor dat de posities van andere spelers worden doorgegeven naar het radar scherm.

Plaats het bestand radar.pk3 van de cd in de mod folder. Compileer de code en plaats de virtuele machine bestanden in een sub-folder "vm" ook in de mod folder. Start nu Quake III Arena met de modificatie. Voeg een paar bots toe om de mod te testen.

