

mod programming voor quake III, deel 3

EEN STATIONAIR WAPEN

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

In Quake III Arena kan een speler gewoonlijk rond rennen en springen met wapens. Deze mod zal echter een wapen toevoegen dat vast aan de grond staat genageld.

De speler kan achter het stationaire wapen gaan staan en het met de "use item" toets in gebruik nemen. Vervolgens kan de speler zich niet meer verplaatsen totdat het stationaire wapen weer wordt losgelaten door nogmaals op de "use item" toets te drukken. Zolang de speler het wapen vast heeft kan met de "attack" toets worden geschoten. Daarbij hoeft de speler zich niet druk te maken over ammunitie omdat die nooit opraakt.

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 game/g_spawn.c direct boven:

```
spawn_t spawns[] = {
```

wordt het volgende toegevoegd:

```
void SP_func_gun( gentity_t *ent ) {
    trap_SetBrushModel( ent, ent->model );
    ent->svflags = SVF_USE_CURRENT_ORIGIN;
    ent->eType = ET_MOVER;
    ent->s.pos.trType = TR_STATIONARY;
    VectorCopy( ent->s.origin, ent->s.pos.trBase );
    VectorCopy( ent->s.origin, ent->r.currentOrigin );
    trap_LinkEntity( ent );
}
```

Direct na dezelfde regel wordt het volgende toegevoegd:

```
{ "func_gun", SP_func_gun,
```

Deze code zorgt ervoor dat het stationaire wapen correct in het level wordt geplaatst.

Bij de pmove->pm_flags in het bestand game/bg_public.h wordt het volgende toegevoegd:

```
#define PMF_STATIONARYGUN 32768 //
using stationary gun
```

In het bestand game/g_active.c in de functie G_TouchTriggers direct na:

```
hit = &g_entities[touch[i]];
```

wordt het volgende toegevoegd:

```
if ( ent->client->pers.cmd.buttons & BUTTON_USE_HOLDABLE ) {
    if ( !strcmp( hit->classname, "func_gun" ) ) {
        if ( hit->timestamp < level.time ) {
            if ( hit->count ) {
                if ( hit->enemy == ent ) {
                    ent->client->ps.pm_flags &= ~PMF_STATIONARYGUN;
                    hit->count = 0;
                    hit->timestamp = level.time + 300;
                    VectorClear( hit->s.apos.trBase );
                    ent->client->ps.generic1 = 0;
                    ent->flags &= ~FL_NO_KNOCKBACK;
                } else {
                    trap_EntityContact( mins, maxs, hit ) {
                        hit->enemy = ent;
                        hit->count = level.time;
                        hit->timestamp = level.time + 300;
                        ent->client->ps.pm_flags |= PMF_STATIONARYGUN;
                        ent->client->ps.generic1 = hit - g_entities;
                        ent->client->ps.weapon = WP_NONE;
                        ent->flags |= FL_NO_KNOCKBACK;
                    }
                }
            }
        }
    }
}
```

Deze code zorgt ervoor dat de speler het stationaire wapen met de "use item" toets kan gebruiken.

Aan het einde van de functie ClientThink_real in het bestand game/g_active.c wordt het volgende toegevoegd:

```
if ( client->ps.pm_flags & PMF_STATIONARYGUN ) {
    gentity_t *gun;
    gun = &g_entities[ client->ps.generic1 ];
    VectorCopy( client->ps.viewangles, gun->s.apos.trBase );
    if ( client->ps.stats[STAT_HEALTH] <= 0 ) {
        client->ps.pm_flags &= ~PMF_STATIONARYGUN;
        gun->count = 0;
        VectorClear( gun->s.apos.trBase );
        client->ps.generic1 = 0;
    }
}
```

Deze code zorgt ervoor dat het stationaire wapen meedraait met de richting waarin de speler kijkt. Ook wordt het stationaire wapen weer vrij gegeven als de speler dood gaat.

Direct na de regel:

```
PM_UpdateViewAngles( pm->ps, &pm->cmd );
```

in de functie PmoveSingle in het bestand game/bg_pmove.c wordt het volgende toegevoegd:

```
if ( pmove->ps->pm_flags & PMF_STATIONARYGUN ) {
    pmove->cmd.forwardmove = 0;
    pmove->cmd.rightmove = 0;
    pmove->cmd.upmove = 0;
    if ( pmove->ps->viewangles[0] > 15 ) {
        pmove->ps->viewangles[0] = 15;
    } else if ( pmove->ps->viewangles[0] < -15 ) {
        pmove->ps->viewangles[0] = -15;
    }
    if ( pmove->ps->viewangles[1] > 35 ) {
        pmove->ps->viewangles[1] = 35;
    } else if ( pmove->ps->viewangles[1] < -35 ) {
        pmove->ps->viewangles[1] = -35;
    }
}
```

Deze code zorgt ervoor dat de speler aan het stationaire wapen vastkleeft en dat de speler

het stationaire wapen maar beperkt kan bewegen.

Direct na de volgende code:

```
if ( pm->ps->weaponTime > 0 ) {
    pm->ps->weaponTime -= pml.msec;
}
```

in PM_Weapon in het bestand game/bg_pmove.c wordt de volgende code toegevoegd:

```
if ( pm->ps->pm_flags & PMF_STATIONARYGUN ) {
    if ( pm->ps->weaponTime > 0 ) {
        return;
    }
    if ( ! ( pm->cmd.buttons & BUTTON_ATTACK ) ) {
        return;
    }
    PM_AddEvent( EV_FIRE_WEAPON );
    pm->ps->weaponTime += 100;
    return;
}
```

Direct boven de regel:

```
AngleVectors( ent->client->ps.viewangles, forward, right, up );
```

in de functie FireWeapon in het bestand game/g_weapon.c wordt het volgende toegevoegd:

```
if ( ent->client->ps.pm_flags & PMF_STATIONARYGUN ) {
    gentity_t *gun;
    gun = &g_entities[ ent->client->ps.generic1 ];
    AngleVectors( gun->s.apos.trBase, forward, right, up );
    VectorCopy( gun->s.pos.trBase, muzzle );
    VectorMA( muzzle, 50, forward, muzzle );
    SnapVector( muzzle );
    Bullet_Fire( ent, MACHINEGUN_SPREAD, MACHINEGUN_DAMAGE );
    Bullet_Fire( ent, MACHINEGUN_SPREAD, MACHINEGUN_DAMAGE );
    return;
}
```

Deze code zorgt ervoor dat het stationaire wapen twee kogels tegelijk afschiet.

Direct na de regel:

```
ent = &cent->currentState;
```

in de functie CG_FireWeapon in het bestand cgame/cg_weapon.c wordt het volgende toegevoegd:

```
if ( ent->generic1 ) {
    CG_MachineGunEjectBrass( &cg_entities[ ent->generic1 ] );
    trap_StartSound( NULL, ent->number, CHAN_WEAPON, cg_weapons[ WP_GAUNTLET ].flashSound[0] );
    return;
}
```

Deze code zorgt ervoor dat het stationaire wapen geluid maakt en lege patronen hulzen uitspuugt.

Plaats het bestand statgun.pk3 – te vinden op de RealGamer 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.

Type op de console:

```
/sv_pure 0
/map statgun
```

Loop vervolgens in de map naar het wapen toe en ga er achter tegenaan staan. Druk nu op de "use item" toets om het stationaire wapen in gebruik te nemen. Als de "attack" toets wordt ingedrukt zal het stationaire wapen schieten. Door een bot aan het spel toe te voegen kan het wapen even grondig worden getest. ■



DE INTERNET SITE

<http://www.quake3world.com> heeft een heleboel nuttige informatie voor iedereen die geïnteresseerd is in het zelf maken van maps (met bijvoorbeeld een stationair wapen).

