

func_vehicle

How to implement func_vehicle 0.1 in progs106

Add the 6 new files into **progs.src** between subs.qc and fight.qc, looking like this:

```
...
defs.qc
subs.qc

maths.qc
phys_defs.qc
phys_matrix.qc
phys_exec.qc

func_vehicle.qc
func_vehicle_builtins.qc

fight.qc
...
```

Currently the code won't compile without a small change, so start by opening **client.qc** and near the top, remove this line:

```
...
void (vector org, entity death_owner) spawn_tdeath;

float modelindex_eyes, modelindex_player;

/*
...
```

We need to add the same line to the bottom of **defs.qc** instead:

```
...
float(entity targ, entity inflictor) CanDamage;

float    modelindex_eyes, modelindex_player;
```

Now we have the vehicle stuff included in the game, next we need to add interactions between the vehicle(s) and the rest.

First we want to add some functionality to **client.qc** for driving and entering/exiting the vehicle. In `PutClientInServer`, right below the `setmodel` stuff, add this:

```
...
    setmodel (self, "progs/player.mdl");
    modelindex_player = self.modelindex;

    setmodel (self, "progs/null.spr");
    modelindex_null = self.modelindex;

    stuffcmd(self, "chase_active 0\n");
    stuffcmd(self, "crosshair 1\n");

    setsize (self, VEC_HULL_MIN, VEC_HULL_MAX);
...
```

This is for allowing the player model to be invisible when seated, and resets the chasecam if a player stops playing while seated in a vehicle.

Still in **client.qc**, move down to `PlayerPostThink`, and add this at the bottom, under `CheckPowerups`:

```
...
    CheckPowerups ();

    TryUpdateVehicle();
};
...
```

This handles the driving when seated in a vehicle.

When the player dies, they need to get tossed out from the vehicle. Open up **player.qc** and navigate to `PlayerDie` to add this:

```
...
void() PlayerDie =
{
    local float i;

    if (self.driving)
        Car_AttachToVehicle(self, self.driving); // will jump out from
vehicle immediately

    self.items = self.items - (self.items & IT_INVISIBILITY);
...
}
```

Next up we need to add a new impulse so players can enter and exit nearby vehicles. Open up **weapons.qc** and navigate to ImpulseCommands to add the following:

```
...
    if (self.impulse == 255)
        QuadCheat ();
    if (self.impulse == 50)
        Car_CheckAttach ();

    self.impulse = 0;
...
```

You've successfully added the basic functionality for entering, exiting and driving vehicles! Keep going if you also want to add the ability for monsters to detect you inside a vehicle and shoot at you.

In **ai.qc**, put this into visible:

```
...
    if (trace_inopen && trace_inwater)
        return FALSE; // sight line crossed contents

    if (trace_fraction < 1 && trace_ent.owner.driving == targ)
        return TRUE;

    if (trace_fraction == 1)
...

```

This way monsters can detect players in cars.

In **combat.qc**, add this to T_RadiusDamage:

```
...
    if (points > 0)
    {
        vector dir = normalize(org - inflictor.origin);
        Phys_Force(head, 1000, dir * points, org, dir);
        if (CanDamage (head, inflictor))
        { // shambler takes half damage from all explosions
...

```

Cars can now get pushed around by explosions.

In **fight.qc** we need to change each *CheckAttack function to support monsters fighting vehicles. This means that each case of:

```
if (trace_ent != targ)
    return FALSE; // don't have a clear shot
```

Needs to change into:

```
if (trace_ent != targ && trace_ent.owner.driving != targ)
    return FALSE; // don't have a clear shot
```

There's 4 different cases.

If you want shambler lightning to do damage, open up **shambler.qc** and find the CastLightning function and change the following:

```
...
    WriteCoord (MSG_BROADCAST, trace_endpos_y);
    WriteCoord (MSG_BROADCAST, trace_endpos_z);

    LightningDamage (org, trace_endpos + dir*4, self, 10);
};
...
```

And you're done!

/ Marko "Polo" Permanto