





Welcome back by part II. Since we have a full script already were going to add stuff to it and make our game even more interesting.

I created a few ramps in the level and we will make the car able to run up to them and land on the other side back on the road again. For this I added a few lines to the zombie_car action. I got this scripting from forum member Malice.

```
action zombie_car()
{
    set(my,SHADOW | METAL);
    zcar = me;
    var speed = 25;
    c_setminmax(me);
    ANGLE temp_angles; <<<<<creating temp angles.
    my.skill20=0; <<< scan result for gravity
    my.skill21=0; <<< gravity speed
    while(1)
    {
        while(game_enable ==1 && damage >0)
        {

            if(key_cul){
                my.pan = 6;
            }
            if(key_cur){
                my.pan = -6;
            }
            if(!key_cur && !key_cul){
                my.pan = 0;
            }
            my.skill20=c_trace(my.x, vector(my.x,my.y,my.z-500), IGNORE_ME | IGNORE_SPRITES | IGNORE_PASSABLE);<<<trac
            model to the ground

            temp_angles.tilt = 0;<<<<tilt angle
            temp_angles.roll = 0;<<<<<< roll angle
            temp_angles.pan = -my.pan;<<<<<< direction angle
            vec_rotate(normal, temp_angles);<<<<<<rotating angle
            temp_angles.tilt = -asin(normal.x);<<<< tilt on x axis
            temp_angles.roll = -asin(normal.y);<<<< tilt on y axis
            my.tilt += 0.1 * ang(temp_angles.tilt - my.tilt); <<<<< play with 0.1 - it sets the angle adjusting speed
            my.roll += 0.1 * ang(temp_angles.roll - my.roll); <<<<<<play with 0.1 - it sets the angle adjusting speed

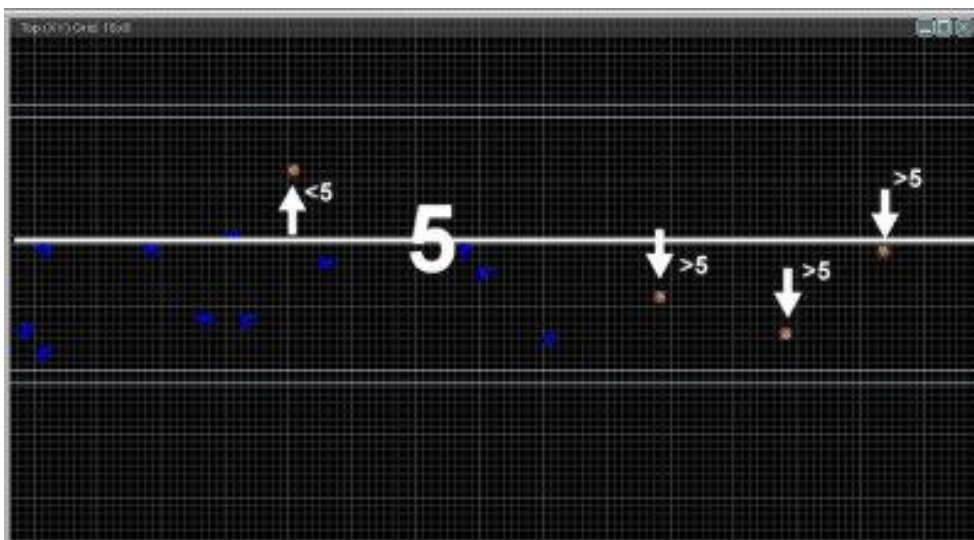
            if(my.skill20 >40) <<<< if result scan gravity is bigger then 40
            my.skill21=-10*time_step;<<<<<< go down with speed 20
            else<<<<<<<< if none of above
            my.skill21=0;<<<<<<<set gravity speed back to 0
            damage -=0.6*time_step;
            my.y += 25 * (key_cul - key_cur) * time_step;
            my.y = clamp(my.y,-220,215);
            c_move(my, vector(speed * time_step, 0, 0), vector(0,0,my.skill21), GLIDE| IGNORE_PASSABLE | IGNORE_SPRITES);

            camera.tilt = -90;
            camera.x = zcar.x+150;
            camera.z = zcar.z + 800;
            new_view.tilt = -10;
            new_view.pan = -180;
            new_view.y = zcar.y;
            new_view.x =zcar.x-80;
            new_view.z = zcar.z+130;
            wait (1);
            if(damage <=40){
                effect(effect_flame, 1, my.x, nullvector);
            }
            if(my.x >19441.801){
                set(zcar,INVISIBLE | PASSABLE);
            }
        }
    }
}
```



```
reset(zcar,SHADOW);
ent_create("zcar01.mdl", vector(zcar.x,zcar.y,zcar.z),race_out);
wait(1);
ent_remove(zcar);
return;
}
}
if(damage <1){
set(my,INVISIBLE | PASSABLE);
reset(my,SHADOW);
snd_play(explo2_snd,150,0);
ent_create("Explosion+16.tga", vector(zcar.x,zcar.y,zcar.z), sprite_played);
ent_create("dcar.mdl", vector(zcar.x,zcar.y,zcar.z),exploded);
wait(1);
ent_remove(me);
set(ga_pan,SHOW);
break;
}
if(game_enable ==0)
{
new_view.tilt = -10;
new_view.pan = -180;
new_view.y = zcar.y;
new_view.x =zcar.x-80;
new_view.z = zcar.z+130;
camera.tilt = -90;
camera.x = zcar.x+180;
camera.z = zcar.z + 800;
wait (1);
}
}
```

I created an traffic cone. It can be placed on the road and be assigned with the traffic cone action script. It uses the familiar emask with a connected event. It should be clear to you how it works as it is used in part I. When the cone is hit it does damage and it is pushed out of the level while going up into the air. Depending if its distance is bigger or smaller than the Y coordinate 5 it will move to the left or to the right. You can find this Y coordinate in WED.





First we create the event function.

```
function cone_impact()<<<< name of the function
{
snd_play(conehit_snd,100,0);<<<<play soundfile
set(my,PASSABLE);<<<<set the cone passable
ent_create("damage.png", vector(my.x,my.y,my.z+50), score_up);<<<<create the damage panel with the function score_up
damage -=10;<<<<subtract 10 of damage variable
while(1)<<<<<<while all above is done
{
if(my.y >5){<<<<<<when the cone has distance bigger then coordinate level y 5
my.z +=5*time_step;<<<<move up with a speed of 5
my.x +=35*time_step;<<<<move forward with a speed of 35
my.y +=8*time_step;<<<<move to the right with speed of 8
my.roll +=10*time_step;<<<<make model roll with speed of 10
wait(1); wait one frame
}
if(my.z >850){<<<<<<if model reaches this height
ent_remove(me);<<<<<<remove model
break;<<<<<< break function
}
my.z +=5*time_step;<<<<move up with a speed of 5
my.x +=35*time_step;<<<<move forward with a speed of 35
my.y -=8*time_step;<<<<move to the left with speed of 8
my.roll +=10*time_step;<<<<make model roll with speed of 10
wait(1); wait one frame
}
if(my.z >850){<<<<<<if model reaches this height
ent_remove(me);<<<<<<remove model
break;<<<<<< break function
}
}
}
```

Then we create the simple action for the cone holding the emask and its event.

```
action traffic_cone()<<<<< name of the action
{
set(my,SHADOW | METAL);<<<<<<give model shadow and metal look
my.emask |= ENABLE_ENTITY | ENABLE_IMPACT;<<<<<<use emask entity and impact
my.event= cone_impact;<<<<<<event launches the cone impact function
}
```

You might have noticed that the event uses a new sound file. This one is defined with the rest of the other sound files like we always do.

```
SOUND* conehit_snd = "conehit.wav";
```

You can alter the code to your wishes and use it for any kind of obstacle model you want to use. If you want you can even made the zombies being thrown into the air this way.



*In many games (especially driving games) there are speed objects on the road. When the driver runs over them its speed will go higher for a certain amount of time.
As it turns out I came to discover it's not that hard to create.*

*We will have to create a new variable for this and a sprite that resembles the turbo pickup.
Then we need to add a few lines to the zombie_car action.*

First we add a new variable to the rest of our variables.

```
var turbo_speed = 0;
```

then the new action for the turbo speed.

```
action turbo_stripe()<<<<<name of action
{
set(my,PASSABLE);<<<<<<Set sprite passable
while(!zcar){wait(1);<<<<<<if pointer is not in game just wait
}
while(1)<<<<<while all of above
{
if (vec_dist (zcar.x, my.x) < 80){<<<<<< if distance between pointer and sprite is smaller than 80
turbo_speed = 1;<<<<<<set turbo speed variable to 1
ent_create("turbo.png", vector(my.x,my.y,my.z+50), score_up);<<<<<<Create turbo picture and use score up function
snd_play(excel_snd,100,0);<<<<<<play sound
wait(-5);<<<<<< wait minus 5 frames
turbo_speed = 0;<<<<<<set turbo speed variable back to 0
break;<<<<<<break the action
}
if (my.x < zcar.x-2000){<<<<<<if sprite distance is smaller than pointer - 20000
ent_remove(me);<<<<<<remove sprite
break;<<<<<<break action
}
wait(1);<<<<<<wait one frame
}
```



I created a nice turbo panel that shows up while getting turbo.

Finally we add some lines to the zombie_car action so when the turbo_speed variable is set to 1 it knows to speed up until it is back on 0 again.

```
action zombie_car()
{
set(my,SHADOW | METAL);
zcar = me;
var speed = 25;
c_setminmax(me);
ANGLE temp_angles; // make this a local ANGLE if you use several identical entities
my.skill20=0; //SCAN RESULT FOR GRAVITY
my.skill21=0; // GRAVITY SPEED
while(1)
{
while(game_enable ==1 && damage >0)
{
if(turbo_speed ==1){
speed = 35;
}
if(key_cul){
my.pan = 6;
}
if(key_cur){
my.pan = -6;
}
if(!key_cur && !key_cul){
my.pan = 0;
}
my.skill20=c_trace(my.x, vector(my.x,my.y,my.z-500), IGNORE_ME | IGNORE_SPRITES | IGNORE_PASSABLE);
temp_angles.tilt = 0;
temp_angles.roll = 0;
temp_angles.pan = -my.pan;
vec_rotate(normal, temp_angles);
temp_angles.tilt = -asin(normal.x);
temp_angles.roll = -asin(normal.y);
my.tilt += 0.1 * ang(temp_angles.tilt - my.tilt); // play with 0.1 - it sets the angle adjusting speed
my.roll += 0.1 * ang(temp_angles.roll - my.roll); // play with 0.1 - it sets the angle adjusting speed
if(my.skill20 >40) //CHANGED THIS TO 60
my.skill21=-10*time_step;
else
my.skill21=0;
damage -=0.6*time_step;
my.y += 25 * (key_cul - key_cur) * time_step;
my.y = clamp(my.y,-220,215);
c_move(my, vector(speed * time_step, 0, 0), vector(0,0,my.skill21), GLIDE| IGNORE_PASSABLE | IGNORE_SPRITES);
camera.tilt = -90;
camera.x = zcar.x+150;
camera.z = zcar.z + 800;
new_view.tilt = -10;
new_view.pan = -180;
new_view.y = zcar.y;
new_view.x =zcar.x-80;
new_view.z = zcar.z+130;
wait (1);
if(damage <=40){
effect(effect_flame, 1, my.x, nullvector);
}
if(my.x >19441.801){
set(zcar,INVISIBLE | PASSABLE);
reset(zcar,SHADOW);
ent_create("zcar01.mdl", vector(zcar.x,zcar.y,zcar.z),race_out);
wait(1);
ent_remove(zcar);
return;
}
}
```



```
if(damage <1){  
    set(my,INVISIBLE | PASSABLE);  
    reset(my,SHADOW);  
    snd_play(explo2_snd,150,0);  
    ent_create("Explosion+16.tga", vector(zcar.x,zcar.y,zcar.z), sprite_played);  
    ent_create("dcar.mdl", vector(zcar.x,zcar.y,zcar.z),exploded);  
    wait(1);  
    ent_remove(me);  
    set(ga_pan,SHOW);  
    break;  
}  
if(game_enable ==0)  
{  
    new_view.tilt = -10;  
    new_view.pan = -180;  
    new_view.y = zcar.y;  
    new_view.x =zcar.x-80;  
    new_view.z = zcar.z+130;  
    camera.tilt = -90;  
    camera.x = zcar.x+180;  
    camera.z = zcar.z + 800;  
    wait (1);  
}
```

And there you have it speed under your control. As the zombie_car action uses its own speed variable it's easy to make it change speed during game.





To make this game a real challenge we will create a high score that players can try to beat. This simple example was created by Kartoffel. It's not the best or safest way to do it but hey it works so let's insert it.

We create a new script under the variables part.

```
#define HIGH_SCORE_SAVE_FILE "hs.score" <<<<<define the highscore file called hs.score
TEXT* input_file = { strings = 1; } <<<< used to load saved information, only one string needed
function load_highscore()<<<<<<<<<name of the function
{
txt_load(input_file, HIGH_SCORE_SAVE_FILE);<<<<<<<<load in the hs_score
wait(1); <<<<<<<<wait one frame to load the text file - important!
hscore_count = str_to_float( (input_file.pstring)[0] ); <<<<<<<< (<text>.pstring) [ line, beginning with 0! ] -> convert to float
}
function save_highscore()<<<<<<<<name of function
{
var file_handle = file_open_write(HIGH_SCORE_SAVE_FILE);<<<<<<<<save the file
wait(1); <<<<<<<<wait to open the file
file_str_write(file_handle, str_for_float(NULL, (double)hscore_count)); <<<<<<<<convert the highscore to a string -> write into file
file_close(file_handle);<<<<<<<<close the file
}
```

Since this script uses a variable called hscore_count we have to create it like we learned before. So we insert this script right under the previous made variables.

```
var hscore_count =0;<<<<<<<<name of variable set on 0
FONT* fnt10_pan = "Ariel Black#16bi";<<<<<using font ariel black sie 16 bold and italic
PANEL* pan_hscore_count = {digits=865,700,"HighScore:%08.0f",fnt10_pan,1,hscore_count;<<<<<create panel and position
layer = 10;<<<<<<<set it on layer 10
flags = SHOW;green=255; blue=0; red=255;<<<<<make it visible in colour yellow.
}
```

So in order to make it work we need to call it when its game over. The score will be compared to the high score. When it's higher than the high score the new high score is saved. When the score is lower nothing happens.

We make some changes in the game over script we made before.

```
function game_over()
{
if(score > hscore_count){<<<<<if score is bigger then hscore_count variable
hscore_count = score;<<<<<then score is hscore_count
save_highscore();<<<<<<<<save highscore
}
wait(-5);
set(black_pan,SHOW | TRANSLUCENT);
black_pan.alpha = 0;
while(black_pan.alpha <100){
black_pan.alpha +=5*time_step;
wait(1);
}
wait(-1);
mouse_mode =3;
set(menu_pan,SHOW);
game_enable = 0;
level_load("Z-roads.WMB");
score = 0;
zkills = 0;
damage=279;
reset(ga_pan,SHOW);
while(black_pan.alpha >0){
```




```
black_pan.alpha -=5*time_step;  
wait(1);  
}  
reset(black_pan,SHOW|TRANSLUCENT);  
}
```

Last thing left to do is make sure that the high score is loaded on game start. So we add one line to our main function.

```
function main()  
{  
mouse_pointer=0;  
shadow_stencil = 2;  
set(black_pan,SHOW|TRANSLUCENT);  
set(logo_pan,SHOW);  
video_set(1024,768,32,0);  
video_window(vector(0,0,0),vector(1024,768,0), 16, "Z-Roads by RP-Interactive, René Pol");  
level_load("Z-roads.WMB");  
load_highscore();<<<<<loading the hiscore file  
black_pan.alpha = 100;  
while(black_pan.alpha >0){  
black_pan.alpha -=3*time_step;  
wait(1);  
}  
wait(-1);  
soundtrack_handle = media_loop("ingame.mp3", NULL, 200);  
while(black_pan.alpha <100){  
black_pan.alpha +=3*time_step;  
wait(1);  
}  
wait(-1);  
reset(logo_pan,SHOW);  
mouse_map = cursor_png;  
mouse_mode =3;  
while(black_pan.alpha >0){  
black_pan.alpha -=3*time_step;  
wait(1);  
}  
reset(black_pan,SHOW|TRANSLUCENT);  
on_p = pause_game;  
}
```

I created a new high score graphic and sound file. So during game when a new high score is reached the panel will show (Created from player pointer) and the sound will be played.

First we define the sound file to be used like we did before :

```
SOUND* highscore_snd = "highscore.wav";
```

We add some new lines to the zombie_car action. I give it a new variable called h_score and set it on 0. When a new high score is reached it will show the high score graphic and plays the soundfile only once. It checks the variable h_score first its set on 0 so it plays the sound and shows the graphic. Then it sets the variable to 1 so it won't be repeated unless a new game and a new high score is made.



```
action zombie_car()
{
set(my,SHADOW | METAL);
zcar = me;
var speed = 25;
var h_score = 0;<<<<create a new variable for checking highscore
c_setminmax(me);
ANGLE temp_angles; // make this a local ANGLE if you use several identical entities
my.skill20 =0; //SCAN RESULT FOR GRAVITY
my.skill21=0; // GRAVITY SPEED
while(1)
{
while(game_enable ==1 && damage >0)
{
if(turbo_speed ==1){
speed = 35;
}
if(key_cul){
my.pan = 6;
}
if(key_cur){
my.pan = -6;
}
if(!key_cur && !key_cul){
my.pan = 0;
}
if(score > hscore_count && h_score ==0){<<<<<check if highscore is made
h_score =1;<<<<<set variable to 1
ent_create("nscore.png", vector(zcar.x,zcar.y,zcar.z+50), score_up);<<<<<create highscore picture with score up function
snd_play(highscore_snd,150,0);<<<<<play sound file
}
my.skill20=c_trace(my.x, vector(my.x,my.y,my.z-500), IGNORE_ME | IGNORE_SPRITES | IGNORE_PASSABLE);
temp_angles.tilt = 0;
temp_angles.roll = 0;
temp_angles.pan = -my.pan;
vec_rotate(normal, temp_angles);
temp_angles.tilt = -asin(normal.x);
temp_angles.roll = -asin(normal.y);
my.tilt += 0.1 * ang(temp_angles.tilt - my.tilt); // play with 0.1 - it sets the angle adjusting speed
my.roll += 0.1 * ang(temp_angles.roll - my.roll); // play with 0.1 - it sets the angle adjusting speed
if(my.skill20 >40) //CHANGED THIS TO 60
my.skill21=-10*time_step;
else
my.skill21=0;
damage -=0.6*time_step;
my.y += 25 * (key_cul - key_cur) * time_step;
my.y = clamp(my.y,-220,215);
c_move(my, vector(speed * time_step, 0, 0), vector(0,0,my.skill21), GLIDE| IGNORE_PASSABLE | IGNORE_SPRITES);
camera.tilt = -90;
camera.x = zcar.x+150;
camera.z = zcar.z + 800;
new_view.tilt = -10;
new_view.pan = -180;
new_view.y = zcar.y;
new_view.x =zcar.x-80;
new_view.z = zcar.z+130;
wait (1);
if(damage <=40){
effect(effect_flame, 1, my.x, nullvector);
}
if(my.x >19441.801){
set(zcar,INVISIBLE | PASSABLE);
reset(zcar,SHADOW);
ent_create("zcar01.mdl", vector(zcar.x,zcar.y,zcar.z),race_out);
wait(1);
ent_remove(zcar);
return;
}
```



```
}  
  
if(damage <1){  
    set(my,INVISIBLE | PASSABLE);  
    reset(my,SHADOW);  
    snd_play(explo2_snd,150,0);  
    ent_create("Explosion+16.tga", vector(zcar.x,zcar.y,zcar.z), sprite_played);  
    ent_create("dcar.mdl", vector(zcar.x,zcar.y,zcar.z),exploded);  
    wait(1);  
    ent_remove(me);  
    set(ga_pan,SHOW);  
    break;  
}  
if(game_enable ==0)  
{  
  
    new_view.tilt = -10;  
    new_view.pan = -180;  
    new_view.y = zcar.y;  
    new_view.x =zcar.x-80;  
    new_view.z = zcar.z+130;  
    camera.tilt = -90;  
    camera.x = zcar.x+180;  
    camera.z = zcar.z + 800;  
    wait (1);  
}  
}
```





We did a lot and have a nice complete game. Turbo speed, high scores, flying away obstacles. All is working just fine. Now one final thing I want to add is some oil pools on the road. If player runs into it, it loses control of the car for a while and does damage.

You should know how to use distance, pointers and sprites as we used them with the turbo speed. Let's do something similar for the oil on the road. I created a nice oil pool sprite.

We begin with the oil sprite action.

```
action oil_puddle()<<<< name of the action
{
set(my,PASSABLE,UNLITE);<<<<make it unlit and passable
while(!zcar){wait(1);<<<<<<when car is not in game just wait
}
while(1)<<<<while above
{
if (vec_dist (zcar.x, my.x) < 80){<<<<<<if distance between oil and car is smaller than 80
oil_p =1;<<<<<<set oil p variable to 1
snd_play(tires_snd,200,0);
wait(-3);<<<<<<wait minus 3 frames
oil_p = 0;<<<<<<set oil p variable back to 0
break;<<<<<<break the action
}
if (my.x < zcar.x-2000){<<<<<< if oil sprite is behind player 2000 distance
ent_remove(me);<<<<<<<remove sprite
break;<<<<<<break action
}
wait(1);<<<<<<wait one frame
}
```

A sound file is played that we defined as all the others.

```
SOUND* tires_snd = "tires.wav";
```

As you can see the oil uses a variable called oil_p so let's place this one with the other variables.

```
var oil_p = 0;
```

*If the variable is set on 1 the car will spin and lose control until it is set back to 0.
So time again to add new lines to our zombie_car script. I added a spinning animation to the car and one simple drive frame. They will be used in the script.*



```
action zombie_car()
{
set(my,SHADOW | METAL);
zcar = me;
var speed = 25;
var h_score = 0;
var spin_percentage;<<<<<set variable for spinning animation
var drive_percentage;<<<<<set variable for driving animation
c_setminmax(me);
ANGLE temp_angles; // make this a local ANGLE if you use several identical entities
my.skill20=0; //SCAN RESULT FOR GRAVITY
my.skill21=0; // GRAVITY SPEED
while(1)
{
while(game_enable ==1 && damage >0)
{
if(turbo_speed ==1){
speed = 35;
}
if(key_cul){
my.pan = 6;
}
if(key_cur){
my.pan = -6;
}
if(!key_cur && !key_cul){
my.pan = 0;
}
while(oil_p ==1){<<<<<when oil_p variable is set to 1
spin_percentage %= 100;<<<<use 100% of spinning animation frames
spin_percentage += 20 * time_step;<<<<play animation at speed 20
ent_animate(my, "spin", spin_percentage, ANM_CYCLE);<<<<<repeat the spinning animation

/////////attach the camera and view so it stays with the car the same for the gravity stuff/////////

camera.tilt = -90;
camera.x = zcar.x+150;
camera.z = zcar.z + 800;
new_view.tilt = -10;
new_view.pan = -180;
new_view.y = zcar.y;
new_view.x =zcar.x-80;
new_view.z = zcar.z+130;
effect(effect_flame, 1, my.x-20, nullvector);
my.skill20=c_trace(my.x, vector(my.x,my.y,my.z-500), IGNORE_ME | IGNORE_SPRITES | IGNORE_PASSABLE);
temp_angles.tilt = 0;
temp_angles.roll = 0;
temp_angles.pan = -my.pan;
vec_rotate(normal, temp_angles);
temp_angles.tilt = -asin(normal.x);
temp_angles.roll = -asin(normal.y);
my.tilt += 0.1 * ang(temp_angles.tilt - my.tilt); // play with 0.1 - it sets the angle adjusting speed
my.roll += 0.1 * ang(temp_angles.roll - my.roll); // play with 0.1 - it sets the angle adjusting speed
if(my.skill20 >40) //CHANGED THIS TO 60
my.skill21=-7*time_step;
else
my.skill21=0;
damage -=0.6*time_step;
c_move(my, vector(speed * time_step, 0, 0), vector(0,0,my.skill21), GLIDE| IGNORE_PASSABLE | IGNORE_SPRITES);
wait(1);
}
if(score > hscore_count && h_score ==0){
h_score =1;
ent_create("nscore.png", vector(zcar.x,zcar.y,zcar.z+50), score_up);
snd_play(highscore_snd,150,0);
}
```



```
}  
ent_animate(my, "drive", drive_percentage, ANM_CYCLE);  
my.skill20=c_trace(my.x, vector(my.x,my.y,my.z-500), IGNORE_ME | IGNORE_SPRITES | IGNORE_PASSABLE);  
temp_angles.tilt = 0;  
temp_angles.roll = 0;  
  
temp_angles.pan = -my.pan;  
vec_rotate(normal, temp_angles);  
temp_angles.tilt = -asin(normal.x);  
temp_angles.roll = -asin(normal.y);  
my.tilt += 0.1 * ang(temp_angles.tilt - my.tilt); // play with 0.1 - it sets the angle adjusting speed  
my.roll += 0.1 * ang(temp_angles.roll - my.roll); // play with 0.1 - it sets the angle adjusting speed  
if(my.skill20 > 40) //CHANGED THIS TO 60  
my.skill21=7*time_step;  
else  
my.skill21=0;  
damage -=0.6*time_step;  
my.y += 25 * (key_cul - key_cur) * time_step;  
my.y = clamp(my.y, -220, 215);  
c_move(my, vector(speed * time_step, 0, 0), vector(0,0,my.skill21), GLIDE | IGNORE_PASSABLE | IGNORE_SPRITES);  
camera.tilt = -90;  
camera.x = zcar.x+150;  
camera.z = zcar.z + 800;  
new_view.tilt = -10;  
new_view.pan = -180;  
new_view.y = zcar.y;  
new_view.x = zcar.x-80;  
new_view.z = zcar.z+130;  
wait(1);  
if(damage <=40){  
effect(effect_flame, 1, my.x, nullvector);  
}  
if(my.x > 19441.801){  
set(zcar, INVISIBLE | PASSABLE);  
reset(zcar, SHADOW);  
ent_create("zcar01.mdl", vector(zcar.x, zcar.y, zcar.z), race_out);  
wait(1);  
ent_remove(zcar);  
return;  
}  
if(damage < 1){  
set(my, INVISIBLE | PASSABLE);  
reset(my, SHADOW);  
snd_play(explo2_snd, 150, 0);  
ent_create("Explosion+16.tga", vector(zcar.x, zcar.y, zcar.z), sprite_played);  
ent_create("dcar.mdl", vector(zcar.x, zcar.y, zcar.z), exploded);  
wait(1);  
ent_remove(me);  
set(ga_pan, SHOW);  
break;  
}  
if(game_enable == 0)  
{  
new_view.tilt = -10;  
new_view.pan = -180;  
new_view.y = zcar.y;  
new_view.x = zcar.x-80;  
new_view.z = zcar.z+130;  
camera.tilt = -90;  
camera.x = zcar.x+180;  
camera.z = zcar.z + 800;  
wait(1);  
}  
}
```

