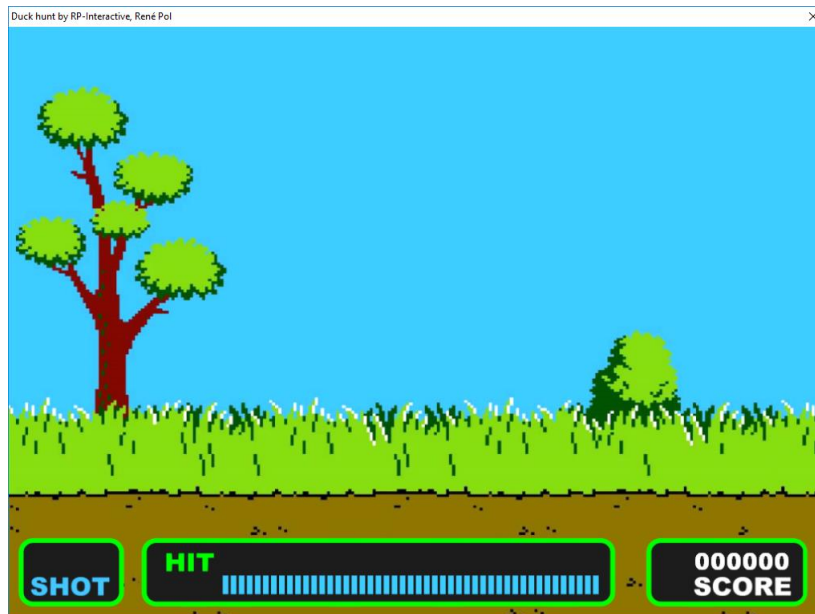


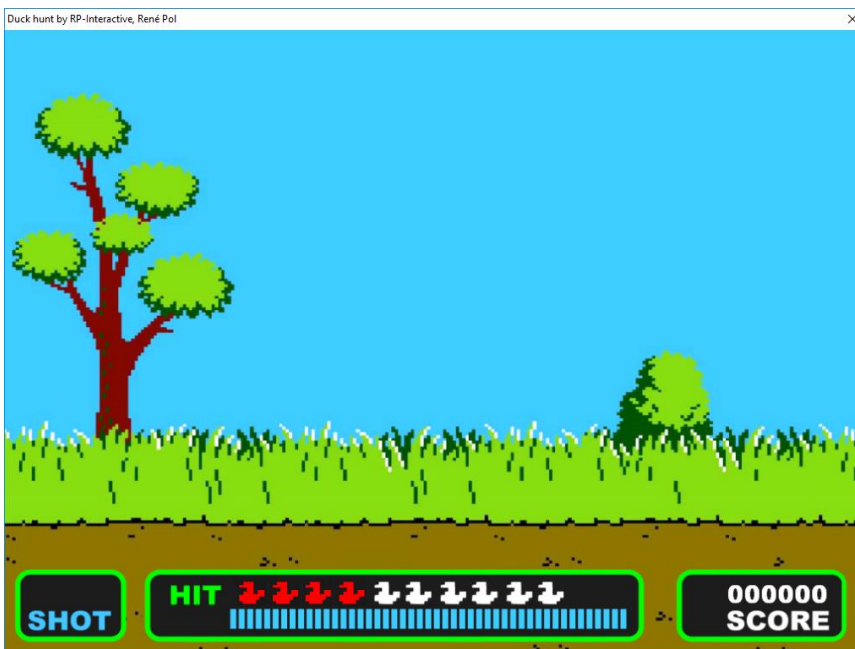
Recreating Duck hunt in Lite C Part I

Written by : René Pol

Just for fun I will try to recreate the old 8 bit NES game Duck hunt. Sprites are made so I am ready to begin. The level is nothing more than a picture in BMP format. I created this at size 1024x768. I just placed it in my level added a camera position to it and made sure it covered the total view. When I run the level it creates the perfect 8 bit outdoor world. Next I will start with the easy stuff adding a variable that holds the score. Also I created a timer bar. Not separate but first as part of the complete background picture. And there we have it, our game environment is done.



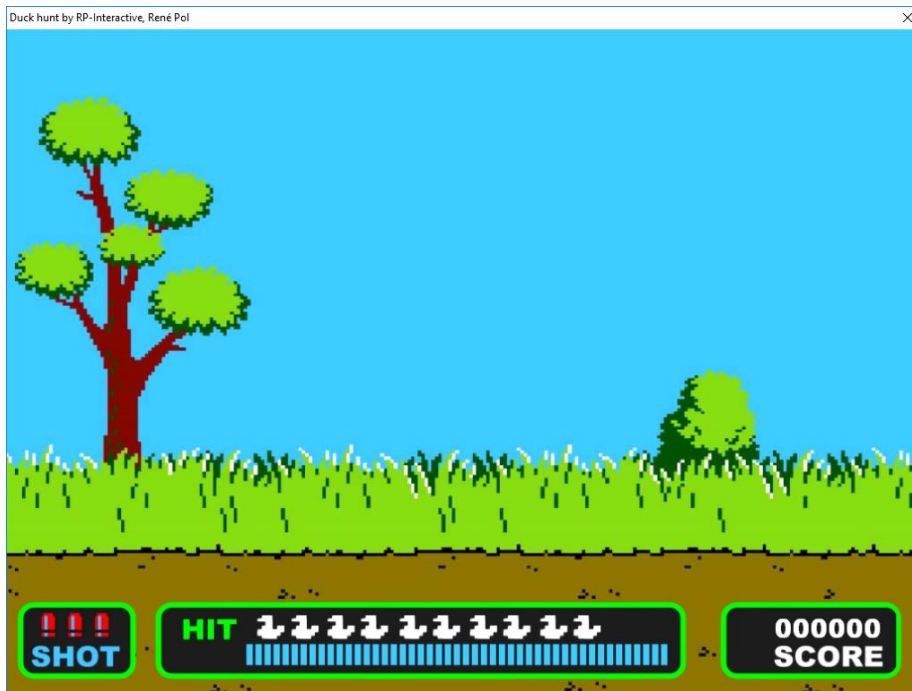
To show the amount of ducks hit I will use separate hit duck pictures. If one duck is shot picture 1 should appear, 2 ducks picture 2 and so on.



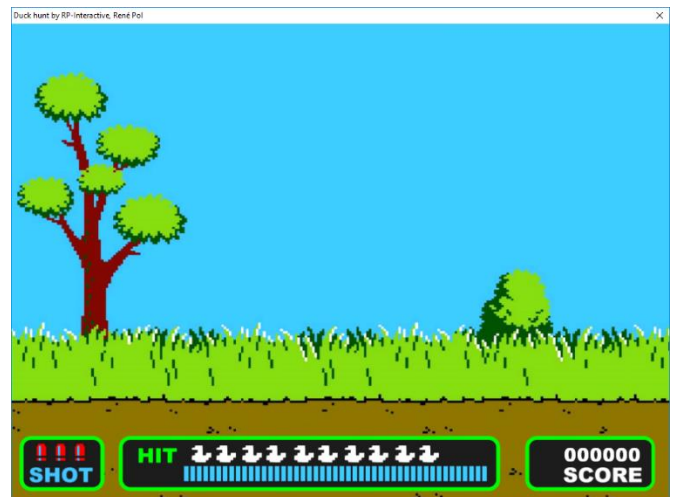
It's beginning to take shape isn't it ? In the original version you have 3 shots before reloading. Let's add the bullet count graphics. So start with 3 bullets and each shot will make one bullet disappear. When it hits 0 it's time to reload with the click of the right mouse button.

Recreating Duck hunt in Lite C Part I

Written by : René Pol



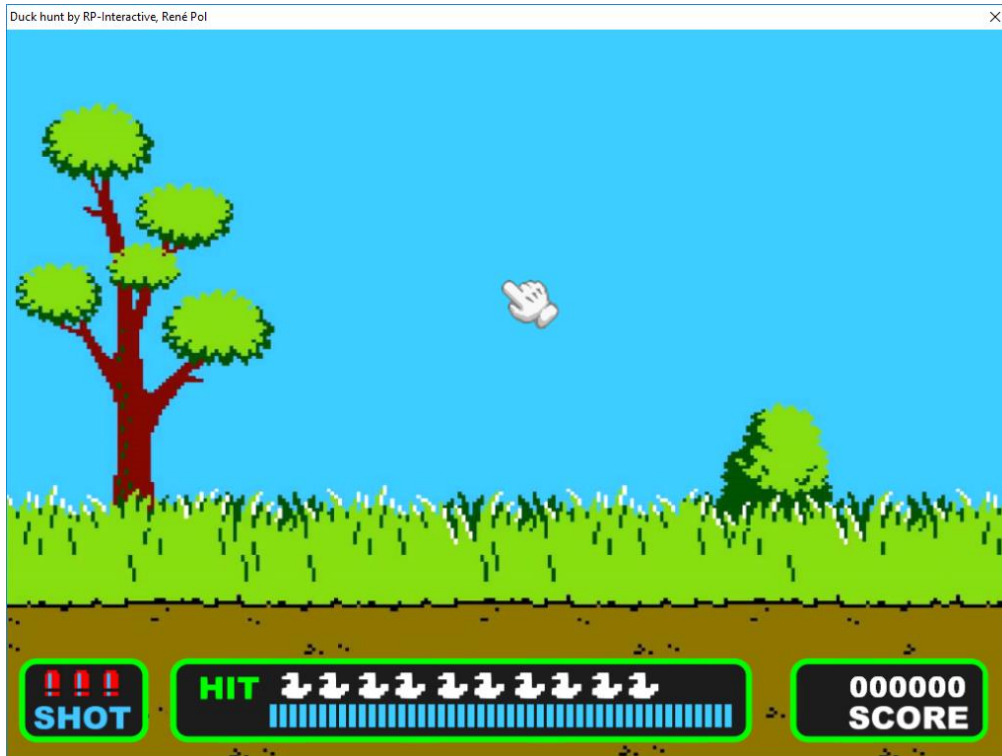
Next I will create a simple shoot function. Every time I hit the left mouse button a shot should be fired and a bullet should disappear. When zero is hit, a right click on the mouse should reload and return 3 bullets. So for this I need a variable bullets that checks which bullet picture should be on or off. I added sound effects to the shooting and reloading of the gun.



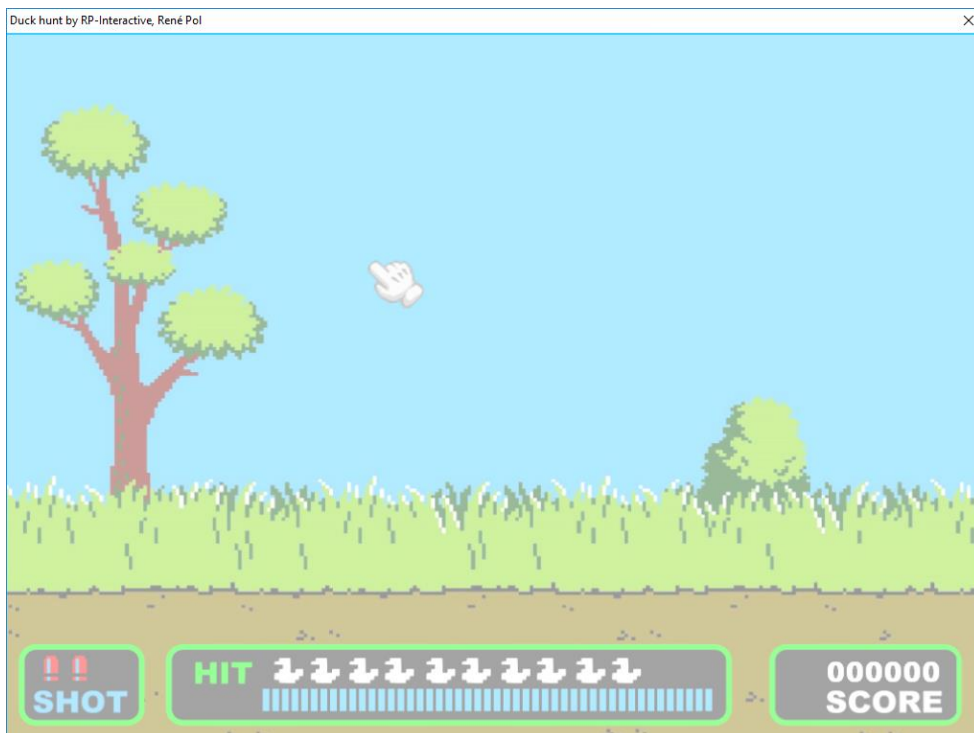
Recreating Duck hunt in Lite C Part I

Written by : René Pol

We do not use an actual light gun and since the ducks are clickable sprites we can use a simple mouse cursor to do the shooting. So I added a visible mouse pointer.



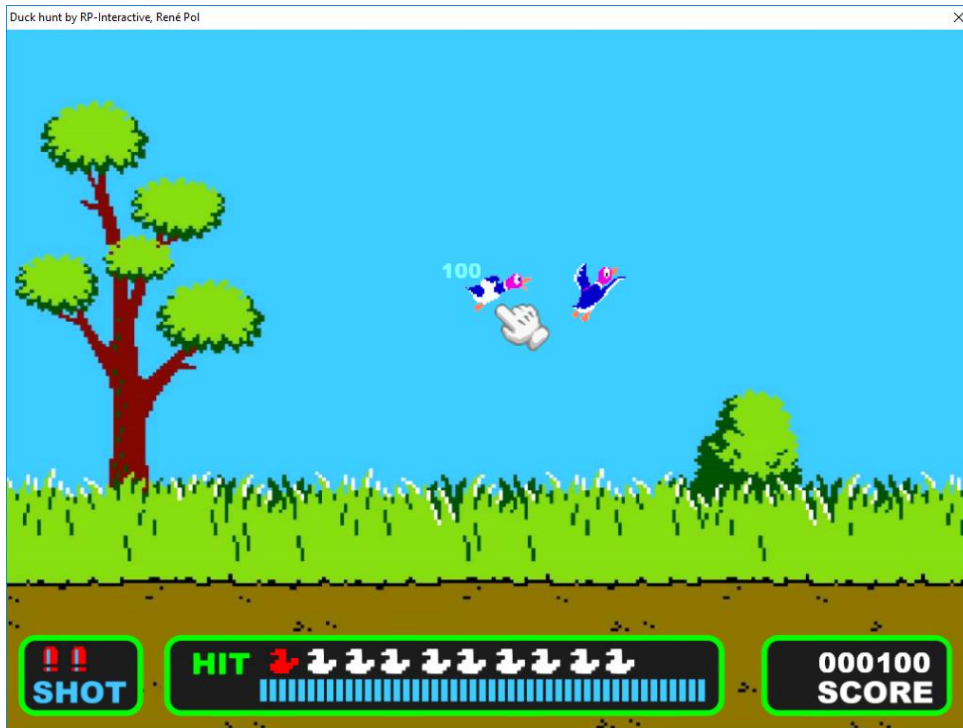
To give it the old style look I created a plain white panel that will flash when shot. It gives the old effect of the light gun.



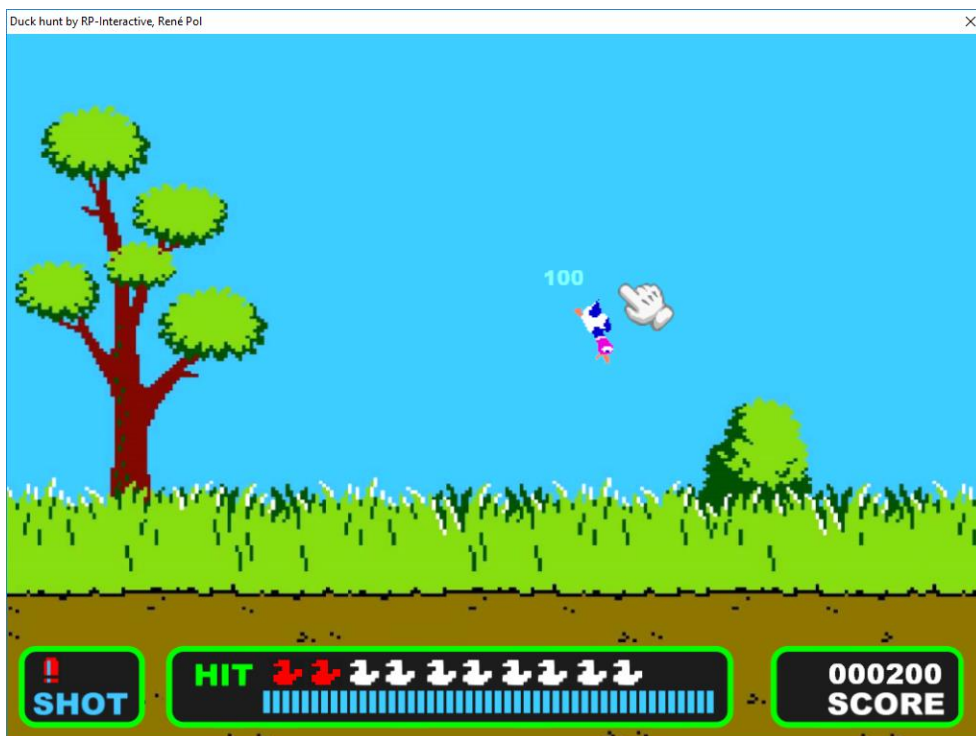
Recreating Duck hunt in Lite C Part I

Written by : René Pol

To create and test hitting a duck we create a sprite and make it clickable. When shot the duck will give 100 points to the score and adds a hit duck picture.



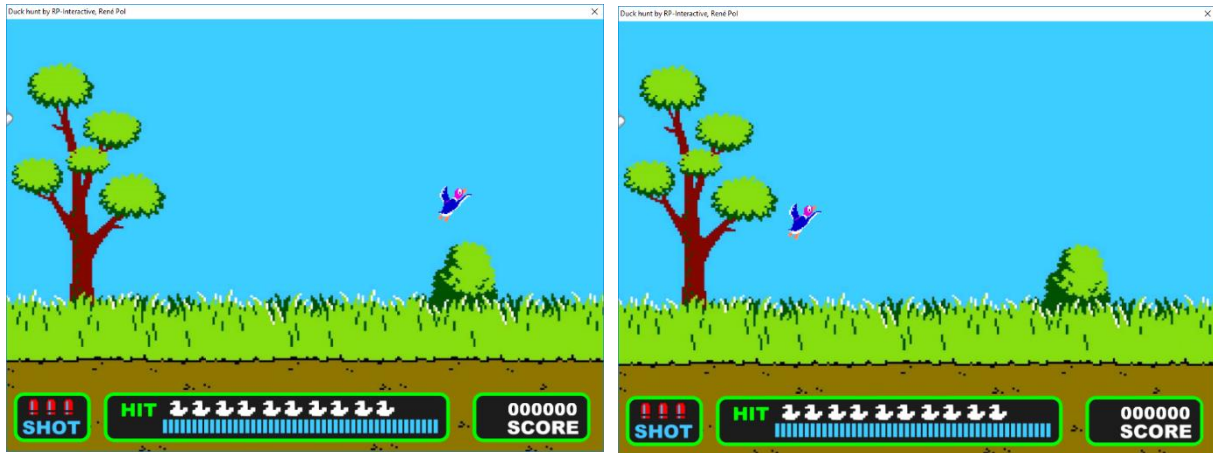
So now the ducks can be shot. We make the shot duck fall down and disappear right above the grass so it looks like it's falling behind it. I added a falling and hitting the ground sound to it.



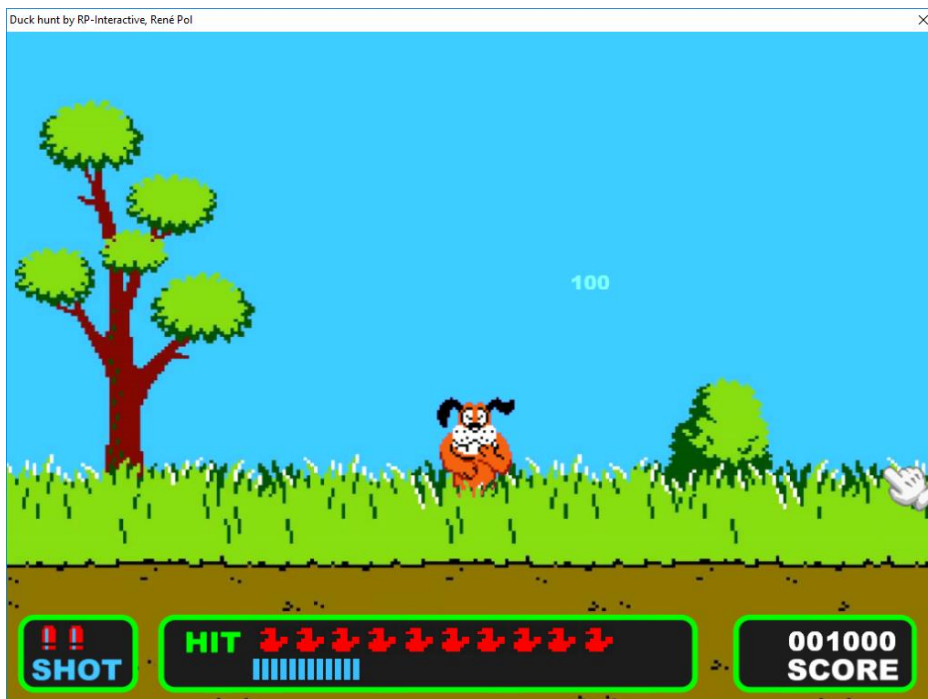
Recreating Duck hunt in Lite C Part I

Written by : René Pol

Almost done just adding a random duck spawner that starts ducks flying at random places. Also I made the ducks move for now just into one direction. You can add different duck placements and actions to them later. They all move now from left to right, going up and start from random places.



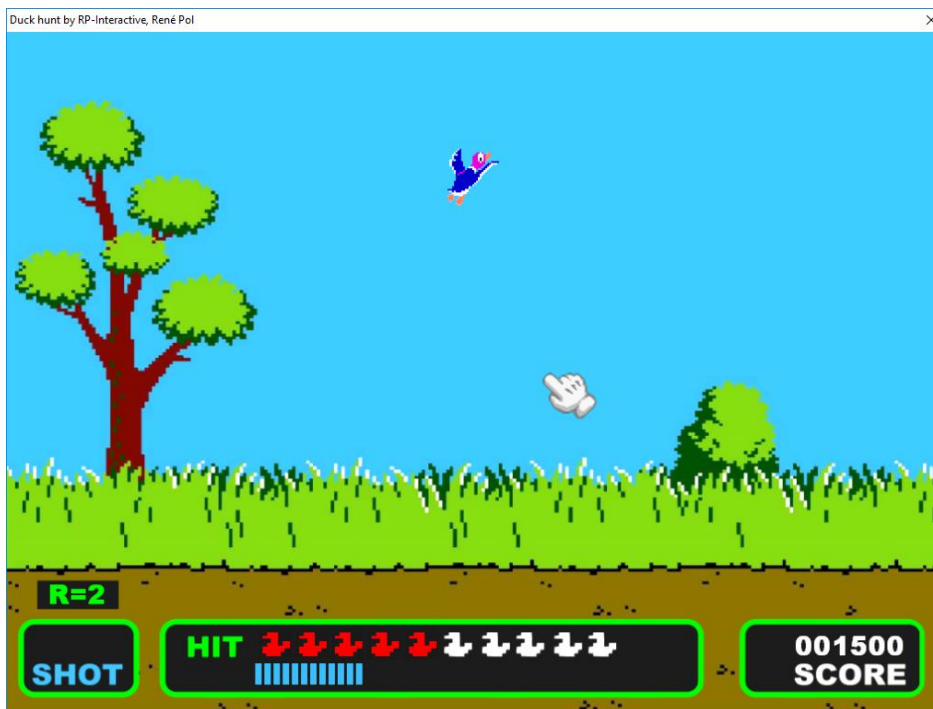
Each time a duck escapes the power bar will go down and a new duck is spawned. I took out the power bar and made this a separate graphic. To prevent a duck being shot while having no bullets we make sure a duck can only be hit if the number of bullets is higher than 0. I made the dog sprite appear by simply changing the background sprite where the dog is visible on.



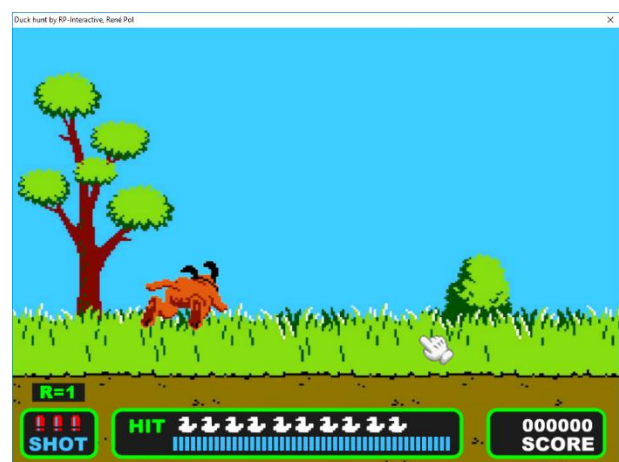
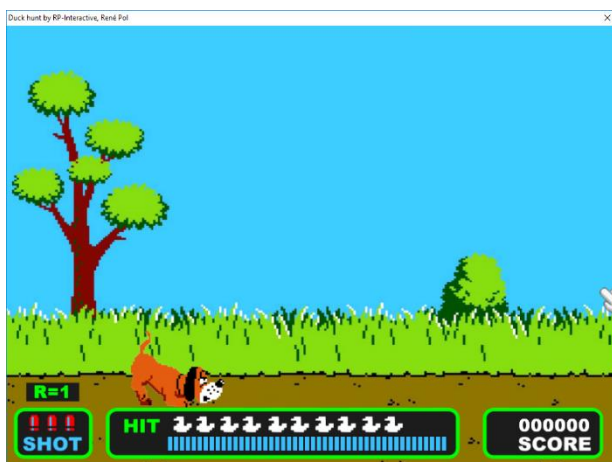
Recreating Duck hunt in Lite C Part I

Written by : René Pol

To make it all look more like the original I added a stage/round/level number. After each round it goes up by one and I can make the ducks speed up each new round or give them other actions. The easy part is no other level is needed. All takes place in the same stage. The background changes only when needed.



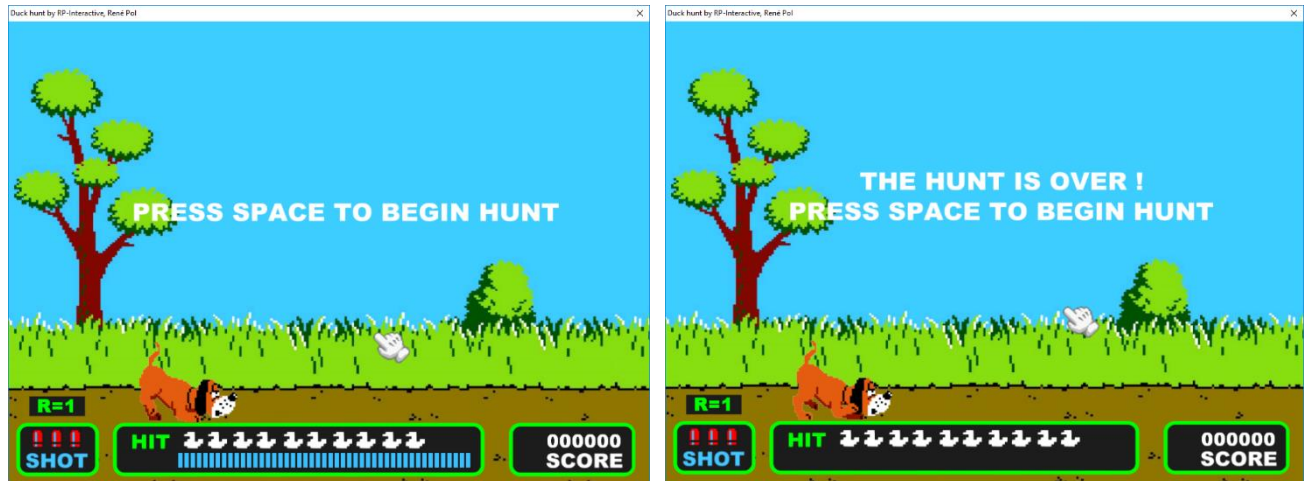
I creates a simple animated dog at game start. Basically it is 3 different pictures. 1 and 2 switch when game is not started yet so it looks like the dog is waiting. When game starts the dog picture 3 is shown and disappears. It looks like its jumping behind the grass !



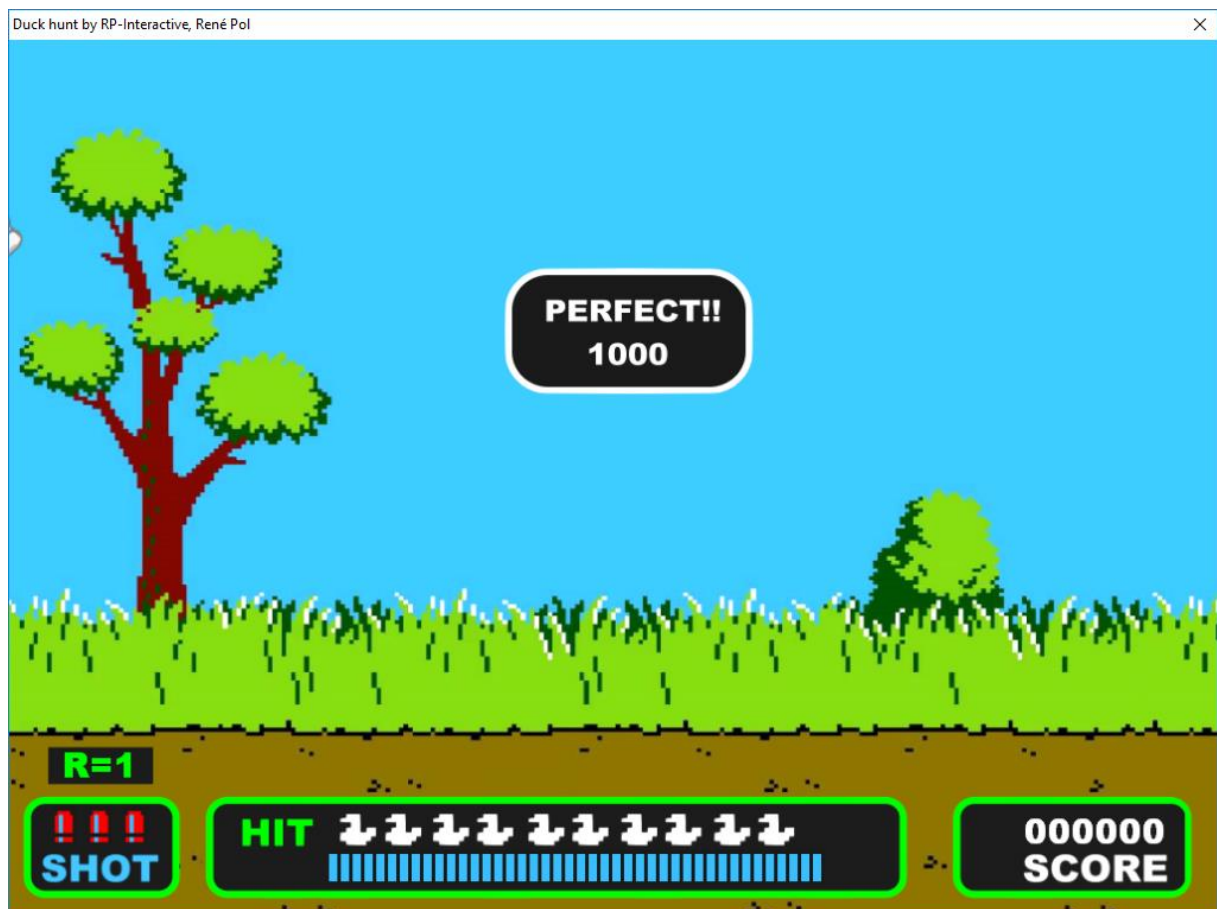
Recreating Duck hunt in Lite C Part I

Written by : René Pol

So all is working. I have level change and the game will work until the power bar hits zero then it should be game over. So I am adding a start text on screen and for game over I will create a function that shows a game over text on screen. I am almost there.



The last thing I will add is a bonus score. I create a variable that counts the shots. If the number of shots is the same as the ducks shot you earn a 1000 points more.



Recreating Duck hunt in Lite C Part I

Written by : René Pol

And it's done. A full duck hunt game created in a total of 6 hours. I might do a part II that will add some more features to it, but for now there is enough to explore and learn from. It was fun for me as a non-expert programmer to recreate this game. My advice is try yourself to duplicate small simple games. As they look simple it involves knowing how to script it and the simplest things sometimes turn out to be the hardest to figure out.

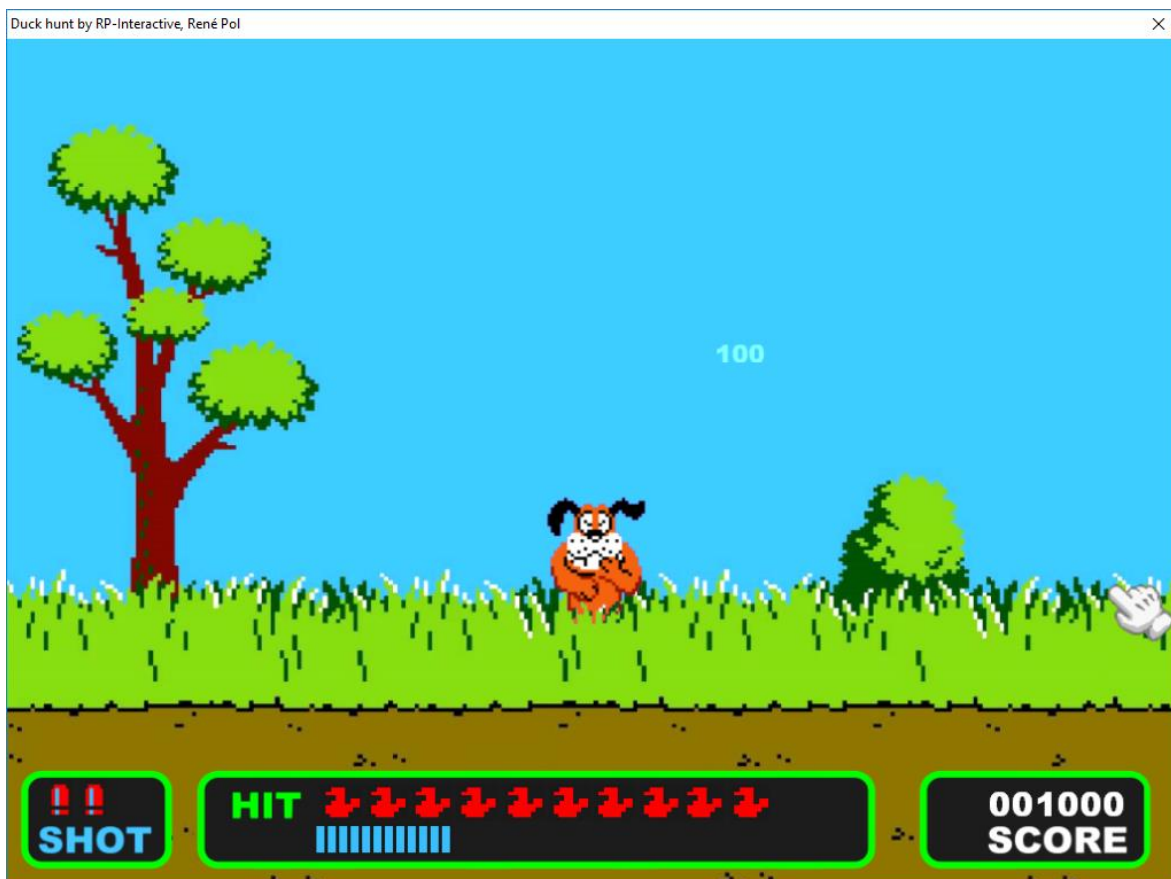
You will find step by step information how this game works right after this page. You can use these basics to create any kind of games. Imagine you could shoot ufo's this way or enemy aliens.

If you enjoyed this workshop drop me a line so I know people appreciate it.

renepol@casema.nl or have a look at my site: <http://www.rp-interactive.nl>

Good luck and happy game creation !

René Pol aka Realspawn



Recreating Duck hunt in Lite C Part I

Written by : René Pol

So you watched me creating a game and it's script. Now you want to learn and understand how all works. Let's go step by step through the game so you see all the script parts working together and you see how it all works. It's always my goal to try to make others learn and understand. If I understand it every one should right ?

The game starts with the main function. **624-636**.

It makes the game start in a window at resolution 1024x768 with a colour dept. of 32bit. It has a title bar with a close icon and a title. The level loaded is called duckhunt.WMB (WMB is the format 3d game studio A8 is using). A mouse pointer (Cursor is used) called pointer_png and mouse_mode 4 is activated. When left mouse button is clicked it will start the fire_bullet function. When right mouse button is clicked it will start the reload_bullets function. When spacebar is pressed the game will start. Another function called doggy_dog starts right away.

```
function main()
{
    video_set(1024,768,32,0);
    video_window(vector(0,0,0),vector(1024,768,0), 16+32, "Duck hunt by RP-Interactive, René Pol");
    level_load("duckhunt.WMB");
    mouse_mode = 4;
    mouse_map = pointer_png;
    on_mouse_left = fire_bullets;
    on_mouse_right = reload_bullets;
    on_space = begin_game;
    doggy_dog();
}
```

If you want to know about the different mouse_modes to use check the manual. Do a search in there for mouse_mode. You will find all clear examples of the different modes.

The mouse cursor to use in game is called a mouse_map I defined it in the script : **Line 59**. So the game knows what cursor to use.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

I created a function to shoot bullets and as you can see in the main it will activate every time I press the left mouse button. The script is located in Lines : **420-454**

```
function fire_bullets()
{
if(bullets >0 && game_enable ==1){
set(wit_pan,SHOW);
shot_count +=1;
bullets -=1;
snd_play(shot_snd,100,0);
wait(-0.1);
reset(wit_pan,SHOW);
}
else
{
wait(1);
}
if(bullets ==3){
set(bullet1_pan,SHOW);
set(bullet2_pan,SHOW);
set(bullet3_pan,SHOW);
}
if(bullets ==2){
set(bullet1_pan,SHOW);
set(bullet2_pan,SHOW);
reset(bullet3_pan,SHOW);
}
if(bullets ==1){
set(bullet1_pan,SHOW);
reset(bullet2_pan,SHOW);
reset(bullet3_pan,SHOW);
}
if(bullets ==0){
reset(bullet1_pan,SHOW);
reset(bullet2_pan,SHOW);
reset(bullet3_pan,SHOW);
}
}
```

The bullets can only be shot when the variable game_enable is equal to 1 and the variable bullets is bigger than 0. **Line 10 and 12** a picture will flash during a shot defined in **lines 90-97**

One will be added to the shot_count variable while one get subtracted from the variable bullets. **Lines 11 and 12.** A sound file will play which is defined in **line 65.**

Depending on the number of bullets the pictures of the bullets appear or disappear. These bullet pictures are defined in **lines 241-269.**

All clear could you find it all back in the script ? Then let's continue.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

In the main function we see another function gets called when the right mouse button is clicked. You can find this function in lines 406-416

```
function reload_bullets()
{
if(bullets <=0){
snd_play(reload_snd,100,0);
wait(-0.5);
bullets =3;
set(bullet1_pan,SHOW);
set(bullet2_pan,SHOW);
set(bullet3_pan,SHOW);
}
}
```

So when the right mouse button is clicked it checks if the bullets variable is smaller or equal to 0. If that's the case it plays the reload sound file defined in line 67. Then it sets the bullets variable back to 3 and shows all the bullet panels again.

On the press of the spacebar the game will begin.

When the game starts 1 panel is made visible right away. The text press space to begin hunt is defined : Line 222-229. On press spacebar the begin game function is activated Line 607-620.

```
function begin_game()
{
if(game_enable ==0){
power =471;
reset(press_pan,SHOW);
reset(hover_pan,SHOW);
snd_play(start_snd,100,0);
score =0;
game_enable =1;
rounds =1;
wait(-2);
duck_spawner();
level_switch();
}
}
```

It will only work when the game_enable variable is equal to 0 (line 10). It will remove the press spacebar panel and the game over panel if needed. It plays the start sound file defined in line : 70. Then it sets the score variable to 0 (lines 14-20) the game_enable variable is set to 1 so you can't start again and again. The rounds variable (lines 22-28) is set to 1. After 2 seconds 2 functions will start called duck_spawner and level_switch. The power bar is set to full size.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

Last part of the main function is a function that is started right away called `doggy_dog`.

Lines 274-294.

```
function doggy_dog()
{
while(game_enable ==0){
set(dog1_pan,SHOW);
wait(-0.5);
reset(dog1_pan,SHOW);
set(dog2_pan,SHOW);
wait(-0.5);
reset(dog2_pan,SHOW);
}
if(game_enable ==1){
set(dog3_pan,SHOW);
reset(dog1_pan,SHOW);
reset(dog2_pan,SHOW);
snd_play(bark_snd,100,0);
wait(-1);
reset(dog3_pan,SHOW);
}
}
```

This function uses 3 graphics defined in **lines 100-125.**

When the `game_enable` variable is equal to 0 it switches between picture 1 and 2. So it looks like the dog is moving. When `game_enable` variable is equal to 1 it resets the first 2 pictures and shows picture 3 before it disappears.

So that was not that hard right ? The main function starts up our game and functions are starting or can be called upon the push of a key.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

In the `begin_game` function we noticed that 2 other functions were started. Let's have a look at them. We begin with the `duck_spawner` function **Lines 688-720**

```
function duck_spawner()
{
    var pick_aduck = 5;
    random_seed = 0;
    wait(1);
    pick_aduck = integer(random(6));
    snd_play(kwak_snd,100,0);
    if(pick_aduck == 0 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,-10,-80), duck_fly);
        return;
    }
    if(pick_aduck == 1 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,-110,-80), duck_fly);
        return;
    }
    if(pick_aduck == 2 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,310,-80), duck_fly);
        return;
    }
    if(pick_aduck == 3 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,-210,-80), duck_fly);
        return;
    }
    if(pick_aduck == 4 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,110,-80), duck_fly);
        return;
    }
    if(pick_aduck == 5 && ducks_hitted < 10 && game_enable == 1){
        ent_create("duck.png", vector(0,200,-80), duck_fly);
        return;
    }
}
```

Every time this function is called it picks a random number from the variable `pick_aduck`. It will only work when the `ducks_hitted` variable (**Line 13**) is smaller than 10 and when the `game_enable` variable is set on 1. On each number it will create a duck with the duck fly action.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

Next is the level_switch function found in lines 559-582.

```
function level_switch()
{
while(1)
{
if(ducks_hitted ==10){
rounds +=1;
wait(-2);
ducks_hitted =0;
check_ducks();
wait(-1);
set(bullet1_pan,SHOW);
set(bullet2_pan,SHOW);
set(bullet3_pan,SHOW);
bullets = 3;
wait(1);
snd_play(start_snd,100,0);
shot_count =0;
wait(-1);
duck_spawner();
}
wait(1);
}
}
```

It keeps checking how many ducks are hit and when the ducks_hitted variable is equal to 10 it adds one to the rounds variable and sets the ducks_hitted back to 0. Then it make sure that the check_ducks function is executed. It sets back the bullet panels and restores the variable bullets back to 3. It plays the start sound file again, sets the shot_count variable back to 0 and activates the duck_spawner function again. This way the game knows every time 10 ducks are hit we continue to the next round.

Ok so the duck_spawner spawns random ducks with a fly action. This action can be found in lines 523-554

```
action duck_fly()
{
set(my,BRIGHT | PASSABLE);
my.lightrange = 200;
my.ambient = 200;
my.emask = ENABLE_CLICK;
my.event = duckshot_event;
while(1)
{
```

Recreating Duck hunt in Lite C Part I

Written by : René Pol

```
my.y -=10*time_step;
my.z +=10*time_step;
wait(1);
if(my.z > 350){
    ent_remove(me);
    power -=50;
    snd_play(power_snd,100,0);
    wait(-2);
    game_over();
    duck_spawner();
    break;
}
if(my.y < -500){
    ent_remove(me);
    power -=50;
    snd_play(power_snd,100,0);
    wait(-2);
    game_over();
    duck_spawner();
    break;
}
}
```

The action uses as emask that is enable click so it will be sensible for clicking on it. When clicked on it it activates the duckshot_event. While on screen it moves on the y and z axis but when a duck reaches the y-axis that is smaller than -500 or the z-axis bigger than 350 it will be removed. The power bar will lose 50 of its total and the power sound file is played. Defined in [line : 69](#).

The game_over function is checked and the duck_spawner function is reactivated.

When this duck is clicked you see the duckshot_event is started. You can find it in [lines : 506-521](#)

```
function duckshot_event()
{
    if (event_type == EVENT_CLICK && bullets >0)
    {
        ducks_hitted +=1;
        check_ducks();
        set(my,INVISIBLE);
        ent_create("hundred.png", vector(my.x-50,my.y+40,my.z+20),score_shot);
        ent_create("duckhit.png", vector(my.x,my.y,my.z),vanish_now);
        wait(1);
        ent_remove(me);
        score +=100;
        wait(-2);
        duck_spawner();
    }
}
```

Now this is what happens here. If the bullets variable is bigger than 0 (So you have bullets) the ducks_hitted variable gets one added. The check_ducks function is activated. It makes the duck disappear as it gets replaced by another duck picture that gets the vanish_now action. Also a picture

Recreating Duck hunt in Lite C Part I

Written by : René Pol

of 100 score will be created at its location with the score_shot function. The original gets removed and 100 is added to the score. Then the duck_spawner gets activated again. After the duck is shot 2 more functions will follow. One makes the new picture appear the other makes it fall down until it reach a certain Height where it will be removed.

Lines 489-499.

```
function vanish_now()
{
    set(my,PASSABLE);
    my.lightrange = 200;
    my.ambient = 200;
    wait(-0.4);
    set(my,INVISIBLE | PASSABLE);
    ent_create("fallduck.png", vector(my.x,my.y,my.z),fall_down);
    wait(1);
    ent_remove(me);
}
```

You see this creates a new picture, removes the old and gives the fall_down function to the new appeared duck picture. Lines 469-485

```
function fall_down()
{
    set(my, PASSABLE);
    my.lightrange = 200;
    my.ambient = 200;
    snd_play(down_snd,100,0);
    while(1)
    {
        my.z -=25*time_step;
        wait(1);
        if(my.z < -80){
            snd_play(duckcrash_snd,100,0);
            ent_remove(me);
            break;
        }
    }
}
```

At this point you should know that sounds and pictures used are always defined on top of your script. Functions and actions can be called up from anywhere in the script as long as it has a logic order.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

2 more functions to explain. The check_ducks function lines 300-401

```
function check_ducks()
{
    while(1)
    {
        if(ducks_hitted ==0){
            reset(duck1_pan,SHOW);
            reset(duck2_pan,SHOW);
            reset(duck3_pan,SHOW);
            reset(duck4_pan,SHOW);
            reset(duck5_pan,SHOW);
            reset(duck6_pan,SHOW);
            reset(duck7_pan,SHOW);
            reset(duck8_pan,SHOW);
            reset(duck9_pan,SHOW);
            reset(duck10_pan,SHOW);
        }
        if(ducks_hitted ==1){
            set(duck1_pan,SHOW);
        }
        if(ducks_hitted ==2){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
        }
        if(ducks_hitted ==3){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
            set(duck3_pan,SHOW);
        }
        if(ducks_hitted ==4){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
            set(duck3_pan,SHOW);
            set(duck4_pan,SHOW);
        }
        if(ducks_hitted ==5){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
            set(duck3_pan,SHOW);
            set(duck4_pan,SHOW);
            set(duck5_pan,SHOW);
        }
        if(ducks_hitted ==6){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
            set(duck3_pan,SHOW);
            set(duck4_pan,SHOW);
            set(duck5_pan,SHOW);
            set(duck6_pan,SHOW);
        }
        if(ducks_hitted ==7){
            set(duck1_pan,SHOW);
            set(duck2_pan,SHOW);
            set(duck3_pan,SHOW);
            set(duck4_pan,SHOW);
            set(duck5_pan,SHOW);
        }
    }
}
```

Recreating Duck hunt in Lite C Part I

Written by : René Pol

```
set(duck6_pan,SHOW);
set(duck7_pan,SHOW);
}
if(ducks_hitted ==8){
set(duck1_pan,SHOW);
set(duck2_pan,SHOW);
set(duck3_pan,SHOW);
set(duck4_pan,SHOW);
set(duck5_pan,SHOW);
set(duck6_pan,SHOW);
set(duck7_pan,SHOW);
set(duck8_pan,SHOW);
}
if(ducks_hitted ==9){
set(duck1_pan,SHOW);
set(duck2_pan,SHOW);
set(duck3_pan,SHOW);
set(duck4_pan,SHOW);
set(duck5_pan,SHOW);
set(duck6_pan,SHOW);
set(duck7_pan,SHOW);
set(duck8_pan,SHOW);
set(duck9_pan,SHOW);
}
if(ducks_hitted ==10){

set(duck1_pan,SHOW);
set(duck2_pan,SHOW);
set(duck3_pan,SHOW);
set(duck4_pan,SHOW);
set(duck5_pan,SHOW);
set(duck6_pan,SHOW);
set(duck7_pan,SHOW);
set(duck8_pan,SHOW);
set(duck9_pan,SHOW);
set(duck10_pan,SHOW);
snd_play(bark_snd,100,0);
snd_play(win_snd,100,0);
if(shot_count == ducks_hitted){
set(bonus_pan,SHOW);
score +=1000;
}
break;
}
wait(1);
}
```

This function checks the `ducks_hitted` variable. Depending of the number it shows the hit duck pictures that are defined in **lines 130-220** on their positions given.

When `ducks_hitted` variable is equal to the `shot_count` variable it will add 1000 to the score variable and shows the bonus panel.

Recreating Duck hunt in Lite C Part I

Written by : René Pol

The game_over function what does it do ? Lines 586-602

```
function game_over()
{
if(power <=0){
snd_play(gameover_snd,100,0);
set(hover_pan,SHOW);
set(press_pan,SHOW);
game_enable =0;
shot_count =0;
bullets = 3;
ducks_hitted =0;
doggy_dog();
check_ducks();
set(bullet1_pan,SHOW);
set(bullet2_pan,SHOW);
set(bullet3_pan,SHOW);
}
}
```

When the power bar is smaller or equal to 0 it plays the game over sound file. It sets the press and game over panel. It resets the game_enable variable back to 0 so the game can be restarted. It resets also the variables shot_count, bullets,ducks_hitted back to their original settings.

The doggy_dog function is activated again as well as the check_ducks function. The bullet graphics are all set again.

So one last thing that power bar how does it work ? Lines 35-46.

```
var power =471;
BMAP* test_map = "pbar.png";
PANEL* back_pan =
{
hbar = 270, 710, 471,test_map, 1, power;
layer = 55;
flags = VISIBLE | OVERLAY;
}
```

The graphic used for the power bar is 471 in width so we set a variable power on that number (Full bar). We place it at given position and make it visible. Every time somewhere in the script a number of the power is subtracted it will make the power bar go smaller. (power -=10; for example)

Any questions ? You know how to reach me.