

Welcome back you curious coders. Issue number 3 is alive and you will create your own first mini game. Ready for it ? wait for it..... go !

Our first simple mini game.

We will use our level and the Sid model with its walking_sid action. The game will be very basic and simple but hey it's a good start and from there we can expand it step by step.

First new thing we will add is a script that will make a score visible on screen. This is done fairly simple. You can use any font you want.

```
////////////////////////////////////Scoring////////////////////////////////////  
  
var score =0;  
  
FONT* fnt1_pan = "Supercell-Magic#28b";  
PANEL* pan_score = {digits=170,50,"Score : %006.0f",fnt1_pan,1,score;  
layer = 1;  
flags = SHOW;green=255; blue=0; red=255;  
}
```

If you run the level now a score will be shown on screen and it starts at zero.
For scoring we will make our hero pick up random stars that will show up every time one is picked up. Each star will be adding 10 points to the score. For this we will make a new action for the star model that we place somewhere in reach in the level.



Here is the script used for stars to pick up. Assign it to the star model and try it.

```
action stars_pick()
{
set(my,BRIGHT | METAL | PASSABLE);
while(1)
{
my.pan -=5*time_step;
wait(1);
if (vec_dist (sid.x, my.x) < 20){
set(my,INVISIBLE);
my.event = NULL;
wait(1);
score +=10;
ent_remove(me);
break;
}
}
}
```

So when the player with the pointer Sid (Remember ?) comes closer than a distance of 20 the star will get invisible and gets removed. Also 10 will be added to the score variable we have created. In this example we collect stars but it could be anything. From the coins in super Mario to power pills in Pac-man.

It would be nice if a sound is playing when the star is caught. Well we know how to do that right ?

First we define the sound we will be using.

```
SOUND* stars_snd = "stars.wav";
```

Then all left is to make it play by adding an extra line in the action stars_pick.

```
action stars_pick()
{
set(my,BRIGHT | METAL | PASSABLE | SHADOW);
while(!sid){
wait(1);
}
while(1)
{
my.pan -=5*time_step;
wait(1);
if (vec_dist (sid.x, my.x) < 20){
set(my,INVISIBLE);
my.event = NULL;
wait(1);
score +=10;
snd_play(stars_snd,100,0);
ent_remove(me);
break;
}
}
}
```



Welcome to creating mini games with 3D Game studio A8.

Written by : René Pol 30-07-2017©

We have a problem. When the star is caught no new star appears. So time to make it happen. By using the distance on x and y axis on the floor part we create 2 variables so the star will always be appearing where our hero can catch them. After the star gets invisible it waits 2 seconds before creating a new star in a random position between the variables we created.

```
action stars_pick()
{
set(my,BRIGHT | METAL | PASSABLE | SHADOW);
var RandomX = random(213);
var RandomY = random(155);
while(!sid){
wait(1);
}
while(1)
{
my.pan -=5*time_step;
wait(1);
if (vec_dist (sid.x, my.x) < 20){
set(my,INVISIBLE);
my.event = NULL;
wait(1);
score +=10;
snd_play(stars_snd,100,0);
ent_remove(me);
wait(-2);
ent_create("star.mdl", vector(RandomX, RandomY, 44),stars_pick);
break;
}
}
}
```

Done !. Now we will create the full game by using our timer and the start + finish function. Make sure to remove the Sid model from the level as we will make it appear on game start. In order to prevent the game starting over and over I created a new variable.

```
var game_begin =0;
```

On B the start function will be called. It only works when the variable is set on 0. We make some changes in the begin_game function. Score is set back to 0 and the countdown back on 60.

```
function begin_game()
{
if(game_begin ==0){
score=0;
countdown_timer = 60
game_begin = 1;
wait(-1);
game_start();
}
}
```

In the game_start function we create the player model.

```
function game_start()
{
set(start_pan,SHOW);
snd_play(start_snd,100,0);
wait(-1);
ent_create("sid.mdl", vector(1.597,0, 19.507),walking_sid);
countdown();
reset(start_pan,SHOW);
}
```

Woahhhh we almost have a real game ! When the timer hits zero it's game over. We remove the sid model in the game_finish function by adding two more lines of script. To be able to restart the game the variable game_begin is set back to 0.

```
function game_finish()
{
set(finish_pan,SHOW);
snd_play(finish_snd,100,0);
set(sid,PASSABLE | INVISIBLE);
sid.event = NULL;
wait(1);
ent_remove(sid);
wait(-1);
reset(finish_pan,SHOW);
game_begin =0;
}
```

Amazing isn't it ? The game starts, the player moves and jumps around and catches stars and when the timer hits zero the game ends and can be restarted. This example is very basic and might look very simple but these lines of scripting are used in many many games. So take your time to get yourself familiar with all you learned in the first 3 issues. In next issue we will create some enemies and learn how to use a mouse cursor with buttons to start the game instead of using keys. You might have noticed how good it is to have the basic scripts and how easy they are to adjust to your wishes.

Happy game creation !
René Pol aka Realspawn.

