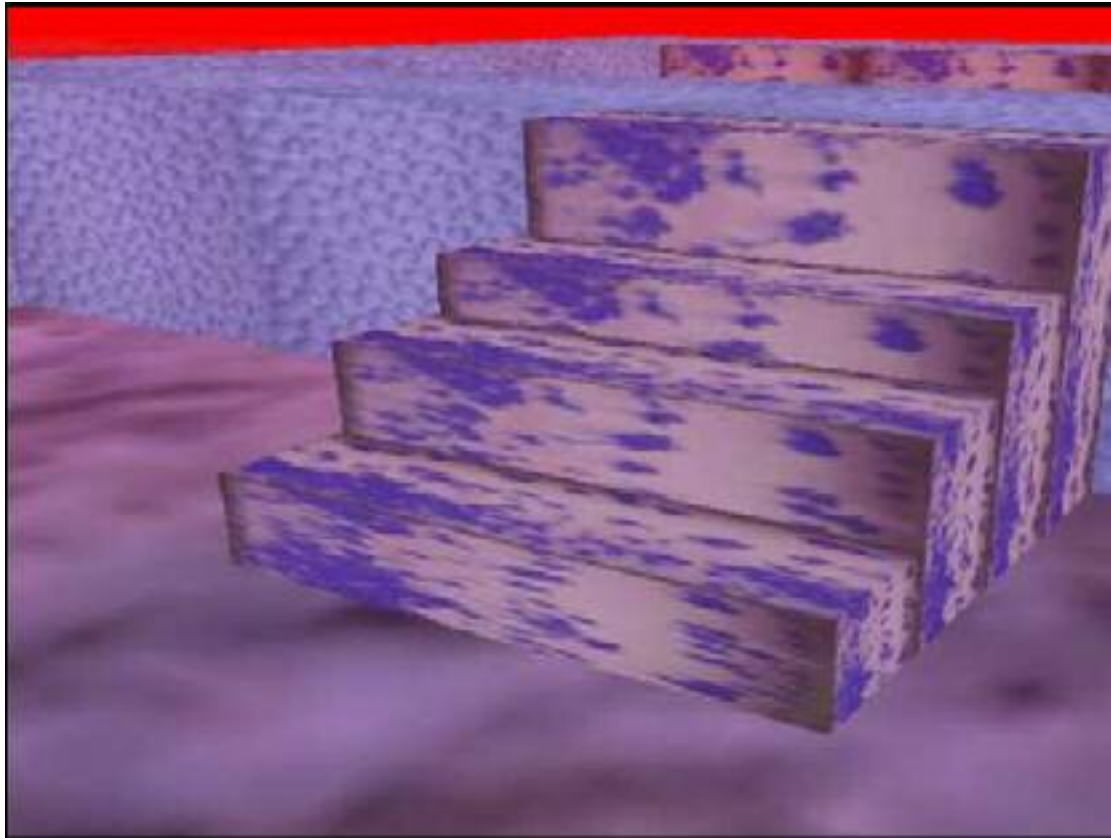


آموزش های سه بعدی GM6

جلسه هفتم: پلکان و سرایشی ها



نویسنده: Prince Of Persia

مطالبی که در این جلسه می آموزید:

- تنظیم اشیا
- تنظیم اسکریپت ها برای پلکان
- برخورد با بلوک ها
- راه رفتن بر روی سرایشی
- تغییر حالت بین سرایشی و پلکان

P30World

تنظیم اشیا

این جلسه در رابطه با پلکان و سرایشی ها است. ما نشان می دهیم که در واقع تفاوتی چندانی بین این دو وجود ندارد. تنها تفاوتی که بین این دو وجود دارد ظاهر آنها و نه طریقه کار آنها می باشد. برای اینکه ببینید که چگونه اینکار را انجام می دهیم بهتر است نگاهی به اشیا و اسکریپت های بکار رفته بیندازیم.

ما از چهار آبجکت استفاده می کنیم : یک دوربین (obj_Camera) ، یک سطح (obj_Floor) ، یک بلوک (obj_Block) و یک پلکان (obj_Ramp) .

دوربین برای مشاهده صحنه که یک سیاره بیگانه است بکار می رود. همچنین دوربین نقش کاراکتر ما در دید اول شخص را بازی می کند. سطح برای ما نقش زمین را که در ارتفاع صفر ($z=0$) قرار دارد بازی می کند. ما همچنین به آبجکت بلوک برای راه رفتن بر روی آنها و آبجکت پلکان برای بالا رفتن از بلوک ها نیاز داریم.

تنظیم اسکریپت ها برای پلکان

همانگونه که گفتیم ما همچنین از اسکریپت ها برای اینکار استفاده می کنیم. اجازه دهید نگاهی به آنها بیندازیم و کاری که انجام می دهند را به نظاره بنشینیم.

اسکریپت scr_Start3D مانند دفعات پیشین حالت سه بعدی را تنظیم می کند. اسکریپت scr_Camera نیز آشنا بنظر می رسد. کد آن بصورت زیر است :

```
//draw what the camera sees
xt=x+cos(direction*pi/180);
yt=y-sin(direction*pi/180);
zt=z+zdirection;
d3d_set_projection(x,y,z+64, xt,yt,zt+64, 0,0,1); //64 is character's
height from z (feet)
```

دقت نمایید که از سینوس برای چرخش دوربین استفاده نکردیم در حالی که هنوز چرخش کار می کند.

اسکریپت scr_DeclareZ و اسکریپت scr_Mouselook نیز برای شما چیز جدیدی نیست. اسکریپت scr_Block کار ساده طراحی بلوک را بر عهده دارد و اسکریپت scr_Floor نیز طراحی زمین را انجام می دهد. تا کنون مطلب جدید مشاهده نکرده اید، درست است؟

برخورد با بلوک ها

ما نمی خواهیم که کاراکتر یا دوربین ما به درون بلوک ها برود و همچنین می خواهیم که برخورد و تکانی را هنگامی که بر روی بلوک ها قرار نگرفته ایم داشته باشیم. ما اینکار را با استفاده از اسکریپت scr_CameraBump انجام می دهیم :

```
//camera bumps into solid things
x=xprevious;

y=yprevious;
speed=0;
```

هنگامی که آبجکت دوربین (obj_Camera) با یک بلوک برخورد می کند اسکریپت scr_BlockCheck اجرا می کند. ما این اسکریپت را در درون رویداد Collision آبجکت دوربین اجرا می کنیم. اسکریپت scr_BlockCheck بصورت زیر است :

```
//check for bumping against blocks
if collision_circle(x,y,2,obj_Block,false,false) then
{
near=instance_nearest(x,y,obj_Block);
if near.z>= z then script_execute(scr_CameraBump);
}
else
{zgravity=1;}
```

در ابتدا ما بررسی می کنیم که آیا کاراکتر با بلوک برخورد داشته است. اگر پاسخ منفی بود ، جاذبه فعال می شود (خط آخر کد را مشاهده کنید) . اگر کاراکتر با بلوک تماس داشته باشد ما نزدیکترین بلوک را پیدا می کنیم و بررسی می کنیم که آیا مقدار z آن بزرگتر یا مساوی با مقدار z کاراکتر است. اگر برابر بود ما اسکریپت scr_CameraBump را با استفاده از دستور script_execute اجرا می کنیم.

این کد بدان معنی است که ما قادر خواهیم بود بر روی بلوک ها حرکت کنیم ولی اگر روی آنها نباشیم سقوط می کنیم.

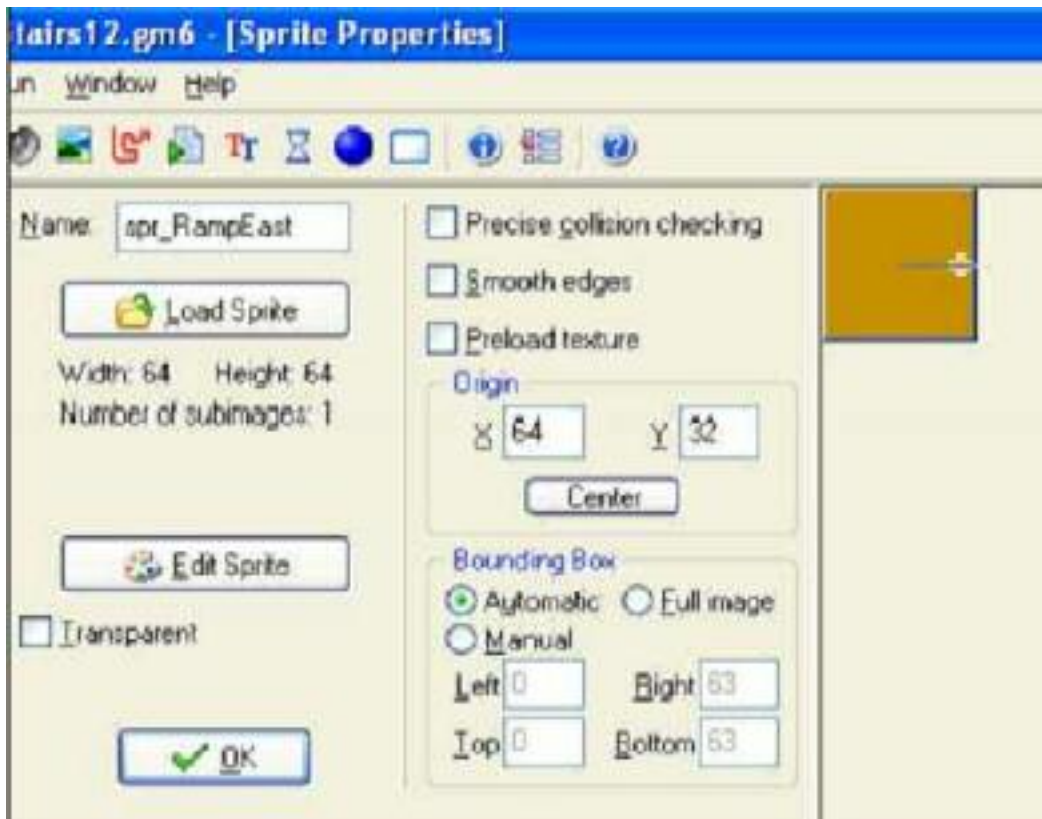
راه رفتن بر روی سراشیبی

در رویداد Step آبجکت دوربین ما دو اسکریپت داریم ، اولی scr_Move که کد زیر را در بر دارد :

```
//check for ramps
if collision_circle(x,y,2,obj_Ramp,false,false) then
{
zgravity=0;zspeed=0;
near=instance_nearest(x,y,obj_Ramp);
z=48-distance_to_point(near.x,near.y);
}
else
script_execute(scr_BlockCheck);
```

در ابتدا ما بررسی می کنیم که آیا سراشیبی وجود دارد. اگر نتیجه منفی بود ما اسکریپت scr_BlockCheck را دوباره اجرا می کنیم.

اگر سراشیبی موجود بود ما قادر خواهیم بود تا بوسیله تنظیم جاذبه z و zspeed به صفر بر روی آن حرکت نماییم و نزدیکتری آبجکت سراشیبی (obj_Ramp) را پیدا نموده و مقدار z کاراکتر خود را به ارتفاع کاراکتر منهای ۴۸ از فاصله تا نقطه (x,y) از نزدیکترین سراشیبی قرار می دهیم.



اگر شما نگاه دقیقتری به اسپرایت سراشیبی (spr_RampEast) بیاندازید می بینید که مبدا یا origin اسپرایت در کجا واقع شده است. ما از مبدا اسپرایت برای محاسبه ارتفاع کاراکتر استفاده می کنیم. ما می خواهیم که کاراکتر هنگامی که به بالای (مبدا) سراشیبی یا پلکان می رسد بیشترین مقدار z را داشته باشد. این اسپرایت spr_RampEast نامیده می شود چون بسمت مشرق یا east دیده می شود. در بازی ممکنست که شما بخواهید سراشیبی ها دیگری که رو به جهات دیگر نیز باشد ایجاد نمایید.

اسکرپت دوم که در رویداد Step آبجکت دوربین قرار دارد scr_MoveZ است. کدی که این اسکرپت در بر دارد بصورت زیر است :

```
//perform gravity
z+=zspeed;
zspeed-=zgravity;

if z<0 then {z=0;zspeed=0;zgravity=0;}
```

این کد اطمینان حاصل می کند که سرعت و جاذبه به طریقی کار می کنند که باعث شوند کاراکتر نتواند به زیر سطح زمین برود.

تغییر حالت بین سراشیبی و پلکان

کدی که در اسکرپت scr_Ramp وجود دارد بصورت زیر است :

```
//draw ramp
if stairs=0 then d3d_draw_floor(x-64,y-32,z, x,y+32,z+48,
background_get_texture(bac_Ramp),1,1)
else
//draw stairs
```

```

{
d3d_draw_block(x,y-32,z, x-64,y+32,z+12,
background_get_texture(bac_Ramp),1,1)
d3d_draw_block(x,y-32,z+12, x-48,y+32,z+24,
background_get_texture(bac_Ramp),1,1)
d3d_draw_block(x,y-32,z+24, x-32,y+32,z+36,
background_get_texture(bac_Ramp),1,1)
d3d_draw_block(x,y-32,z+36, x-16,y+32,z+48,
background_get_texture(bac_Ramp),1,1)
}

```

اولین خط بررسی می کند که آیا مقدار متغیر staris صفر است. اگر نتیجه مثبت بود ما تصویر یک سراشیبی را (با استفاده از d3d_draw_floor) ترسیم می کنیم. وگرنه یک توالی از مکعب ها یا بلوک ها که به نام پلکان معروف است ترسیم می کنیم.

تغییر حالت بین سراشیبی و پلکان در رویداد Mouse آبجکت سراشیبی (obj_Ramp) انجام می پذیرد. این رویداد کد سوییچ متغیر staris از صفر به یک را نگه داری می کند :

```

//change into stairs
stairs=abs(stairs)-1;

```

دقت نمایید که شما به حلقه if...then برای تغییر متغیر stairs به صفر و یک نیاز ندارید. کلیک دکمه چپ ماوس شکل سراشیبی و پلکان را تغییر می دهد. قاعده کلی کد هر دوی آنها یکسان است. همانگونه که به شما گفتم تنها تفاوت آنها شکل ظاهریشان و نه در شیوه کار با آنها می باشد.
