

آموزش های سه بعدی GM6

جلسه نهم: ساخت آسانسورها



نویسنده: Prince Of Persia

مطالبی که در این جلسه یاد می گیرید :

- در رابطه با نوع بالابر
- حرکت دادن آسانسور
- حمل نمودن کاراکتر
- به حساب آوردن ارتفاع کاراکتر
- گشت زدن در اطراف

P30World

در رابطه با نوع بالابر

در آموزش قبلی ما دیدیم که چگونه کاراکتر می تواند بوسیله پرش از سطحی به سطح دیگر بالا رود. یک راه دیگر برای تغییر ارتفاع کاراکتر یک آسانسور می باشد. معمولا دو نوع آسانسور وجود دارد.

نوع اول آسانسورها نوع بسته می باشد: درهای آسانسور باز می شوند، کاراکتر وارد می شود درها بسته می شود و بعد از چند ثانیه درها باز می شوند و کاراکتر از آسانسور در طبقه دیگری خارج می شود. در این نوع از آسانسور محیط اطراف به چشم بازیکن نمی آید و بدلیل اینکه هیچ حرکتی برای بازیکن پدیدار نیست نوع جالبی در حالت سه بعدی نمی باشد.

از دید برنامه نویسان بازی ها پرش از اتاقی به اتاق دیگری راه بسیار ساده ای است. تنها زمانی برای اینکه room جدید در حافظه کامپیوتر بارگذاری شود مورد نیاز است. در واقع هیچ نوع استفاده از صنعت سه بعدی و مقدار ارتفاع یا z در این نوع استفاده نمی شود.

نوع دوم از آسانسورها نوع باز می باشد. بازیکن بر روی سطحی قدم می گذارد و هنگامی که آسانسور بالا می رود دنیای اطراف آسانسور برای بازیکن قابل مشاهده هستند.

حرکت دادن آسانسور

یک آسانسور در واقع تفاوت زیادی با یک کاشی platform ندارد. تنها تفاوتی که وجود دارد تغییر در ارتفاع آسانسور می باشد. آسانسور با استفاده از اسکریپت scr_ElevatorMove واقع در رویداد step آبجکت والد Par_Elevator حرکت می کند که در زیر کد آن آمده است:

```
//move elevator
z+=zspeed;
if z>1024 then z=-32;
```

مقدار zspeed سرعت آسانسور را مشخص می کند. آخرین خط تضمین می کند که هنگامی که آسانسور به بالا می رسد به پایین ترین مکان باز گردد. مقدار آن از زیر صفر شروع می شود زیرا باید کاراکتر را بر روی خود سوار نماید و کاراکتر بر روی زمین یا نقطه صفر قرار دارد.

حمل نمودن کاراکتر

در فایل سورسی که به همراه این آموزش آمده شما مشاهده می کنید که ما یک سطح آسانسور را ایجاد کردیم که می تواند کاراکتر را حمل نماید و آن را به بالاترین سطح منتقل کند. کد مهم اینکار در اسکریپت scr_MoveZ وجود دارد که در رویداد Step آبجکت دوربین obj_Camera قرار داده شده است. کد آن بصورت زیر است:

```
//check for elevator
if collision_point(x,y,chi_Elevator,false,false) then
{
znear=instance_nearest(x,y,chi_Elevator).z;
if z>znear then {z=znear+64;zspeed=1;zgravity=0}
if z<znear then {zgravity=.1;z+=zspeed;zspeed-=zgravity;}
}
else
{
zgravity=.1;z+=zspeed;zspeed-=zgravity;
}
```

```
//land on ground
if z<0 then {z=0;zspeed=0;zgravity=0;}
```

مقدار z کاراکتر با مقدار z نزدیکترین آبجکت آسانسور (chi_Elevator ، زیرا متد والد-فرزند در این مثال استفاده شده است) مقایسه می شود. تنظیمات بر اساس $zgravity$ ، $zspeed$ و غیره تغییر می کنند. این کد همچنین فرود آمدن کاراکتر را نیز کنترل می کند.

به حساب آوردن ارتفاع کاراکتر

دقت نمایید که آبجکت کاراکتر در فایل سورس وجود ندارد. آبجکت دوربین به عنوان کاراکتر برای ما عمل می کند. این بدلیل این است که ما از دید اول شخص یا first person استفاده کردیم که در آن خود کاراکتر مشاهده نمی شود.

آبجکت دوربین نسبت به مکان واقعی خود در ارتفاع (مقدار z) بالاتر ترسیم می شود. شما می توانید فکر کنید که مانند اینست که پاهای کاراکتر بر روی آسانسور است اما چشمان کاراکتر برای احساس واقع نمایی بالاتر قرار گرفته است. ما در واقع ارتفاع کاراکتر را به محاسبه خود اضافه می کنیم. دوربین بالاتر از مقدار z همانطور که شما در اسکریپت scr_Camera می بینید قرار می گیرد :

```
//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);
```

دو مقدار $z+64$ و $zt+64$ باعث می شود که دید دوربین بالاتر از ارتفاع واقعی دوربین قرار بگیرد و احساس چشم کاراکتر را به شما بدهد.

گشت زدن در اطراف

مقادیر xt ، yt و zt در کد بالا می تواند افکت گشت زدن را به شما در دید اول شخص بدهد. ما با استفاده از اسکریپت زیر افکت نگاه کردن به اطراف با ماوس را نیز ایجاد کرده ایم :

```
//MOUSELOOK
```

```
//calculate motion
change_x=(display_mouse_get_x()-512)/16
change_y=(display_mouse_get_y()-384)/16
```

```
//apply treshold
if (change_x<.1) {change_x=round(change_x)}
```

```
if (change_y<.001) {change_y=round(change_y)}
```

```
//move cam
obj_Camera.direction-=change_x
obj_Camera.zdirection-=change_y/64
```

```
//move mouse back  
display_mouse_set(display_mouse_get_x()-  
change_x*4,display_mouse_get_y()-change_y*6)
```

در کلام ساده این کد مکان نشانگر ماوس را بدست می آورد و از آن برای تنظیم جهت (چپ و راست) و zdirection (جهت بالا و پایین) دوربین استفاده می کند. به هر حال قسمت apply threshold از کد دوربین از شناور بودن و معلق بودن زیاد محافظت می نماید.

همچنین ما یک آبجکت به نام obj_Flash ایجاد نمودیم که افکت درخشش خورشید را ایجاد می کند. تنها برای اینکه محیط بازی را کمی جذاب تر کنیم.
