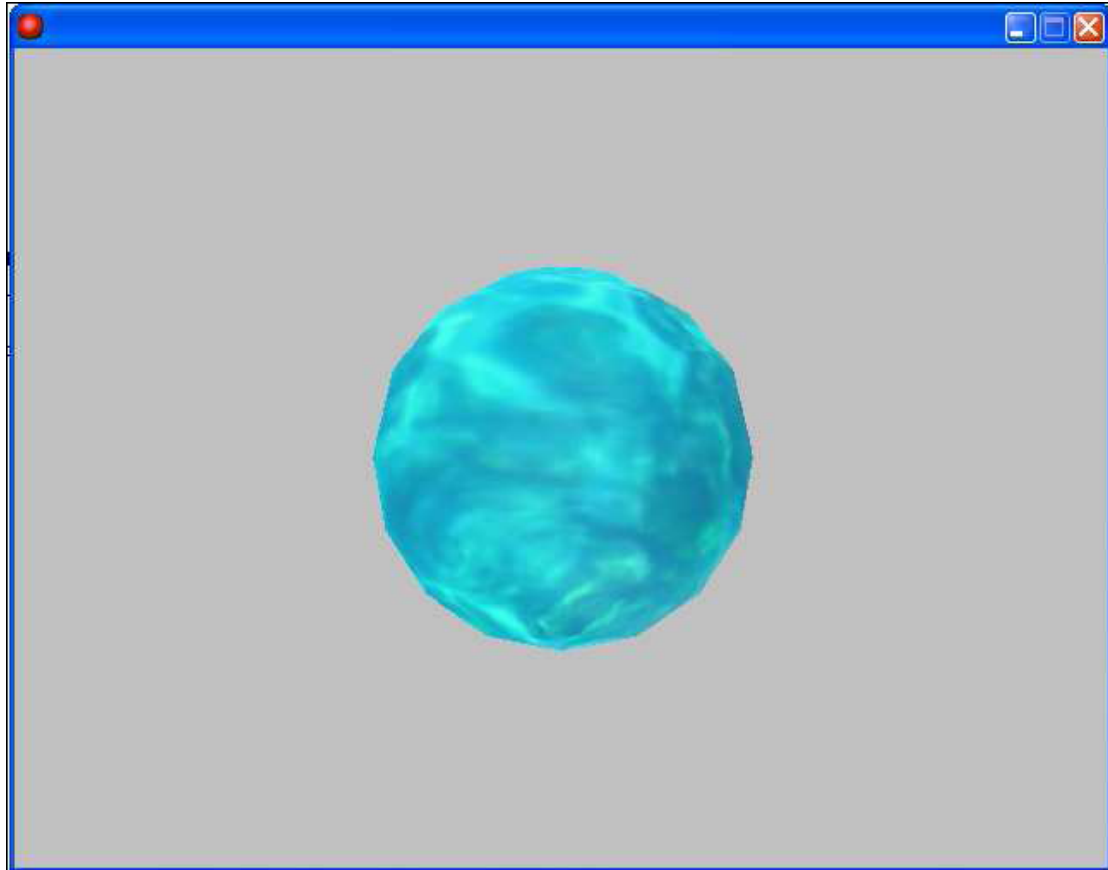


# آموزش های سه بعدی GM6

## جلسه دوم: مختصات Z



نویسنده: Prince Of Persia

مطالبی که در این جلسه می آموزید :

- طراحی وابسته اشیا
- قرار دادن دوربین بر روی یک آبجکت
- وارد و خارج نمودن اسکرینیت ها
- تعریف مختصات Z
- اضافه کردن حرکت در راستای محور Z

## طراحی وابسته اشیا

این آموزش ادامه آموزش قبلی می باشد ، بنابراین اول آموزش قبلی را مطالعه نمایید . مانند مثال قبلی ما می خواهیم شی را برای مشاهده در دنیای سه بعدی ایجاد نماییم. این بار ما نمی خواهیم که دوباره یک بلوک ایجاد کنیم بلکه یک شی ساده مثل توپ را ایجاد خواهیم کرد. بنابر آنچه شما در قسمت قبل یاد گرفته اید یک room با نام (roo\_Tutorial) و یک تصویر پس زمینه با نام (bac\_Ball) و یک آبجکت با نام (obj\_Ball) و یک اسکریپت با نام (scr\_Ball) ایجاد کنید که در اسکریپت توپ کد زیر را که توپ را ترسیم می کند وارد نمایید:

```
//draw the bubble
d3d_draw_ellipsoid(x-16,y-16,z-16, x+16,y+16,z+16,
background_get_texture(bac_Ball),1,1,16);
```

یک کره یا توپ در مکان نسبی  $(x,y,z)$  با این کد ترسیم می شود. ما اکنون باید کارایی مقدار Z را با امتحان آن در یابیم. این کره با موقعیت نسبی مرکز خود ترسیم می شود. مرکز کره در نقطه  $(x,y,z)$  می باشد ولی این کره از یک سمت با موقعیت  $(x-16,y-16,z-16)$  تا سمت دیگر با موقعیت  $(x+16,y+16,z+16)$  ترسیم می شود.

ترسیم یک کره نسبت به مقدار  $x$  ،  $y$  و  $z$  به این معنی می باشد که ما می توانیم مکان آن را تغییر دهیم. اگر شما با طراحی بازیهای دو بعدی با GM آشنایی داشته باشید معانی مقدارهای  $x$  و  $y$  را درک می کنید. مقدار  $z$  تنها یک مقدار دیگر مانند  $x$  و  $y$  تنها در جهت دیگری می باشد. شما با کار کردن با آن تاثیر آن را درک می کنید. در بخش های بعدی ما مقادیر  $x$  و  $y$  و  $z$  را تغییر می دهیم تا نتیجه را مشاهده کنیم.

### قرار دادن دوربین بر روی یک آبجکت

بیا بیاید یک آبجکت به نام (obj\_Ball) را ایجاد کنیم که اسکریپت (scr\_Ball) را اجرا کند. ما می خواهیم این اسکریپت در رویداد Draw آبجکت توپ ما اجرا شود. مانند مثال قبل ما به یک دوربین برای دیدن آبجکت توپ خود احتیاج داریم. اسکریپتی به نام (scr\_Camera) ایجاد کنید و کد زیر را درون آن قرار دهید :

```
//draw what the camera sees
d3d_set_projection(0,0,0, obj_Ball.x,obj_Ball.y,obj_Ball.z, 0,0,1);
```

همانطور که می بینید دوربین در موقعیت  $(0,0,0)$  قرار داده شده است و به مسیر توپ ما  $(obj\_Ball.x,obj\_Ball.y,obj\_Ball.z)$  خیره شده است . اینکار باعث می شود دوربین هموار به مرکز توپ نگاه کند.

### وارد نمودن و صادر نمودن اسکریپت ها

ما به آبجکتی برای اجرای اسکریپت دوربین خود مانند مثال قبل احتیاج داریم. بنابراین ما یک آبجکت دوربین با نام (obj\_Camera) ایجاد می کنیم. در رویداد Draw این آبجکت ما اسکریپت دوربین (scr\_Camera) را اجرا می کنیم. شما باید تا اکنون با اینکار آشنا شده باشید. در رویداد Create آبجکت دوربین ما دقیقاً همان آبجکتی که حالت سه بعدی را آغاز می کرد اجرا می کنیم.

ما میتوانیم اینکار را با باز کردن مثال آموزش قبلی و باز کردن پوشه Scripts و باز کردن اسکریپت (scr\_Start3D) ، انتخاب و کپی کردن متن این اسکریپت و paste کردن در مثال جدید خود انجام دهیم. ما راه بهتری برای اینکار وجود دارد. مثال آموزش قبلی را باز نمایید و اسکریپت مورد نظر را انتخاب کنید اکنون بر روی آن راست کلیک کنید و گزینه "Export Selected Script .." را انتخاب نمایید. از این طریق شما می توانید اسکریپت خود هر جایی که می خواهید ذخیره کنید تا بعداً آن را مورد استفاده قرار دهید. بنابراین ما اینکار را انجام می دهیم و آن را تحت نام scr\_Start3D ذخیره کنید.

اجازه دهید به مثال فعلی خود برگردیم و اسکریپتی را که export کردیم را در مثال فعلی import کنیم. کافی است از منوی Scripts گزینه "Import Scripts" را انتخاب نمایید. این اسکریپت در پوشه Scripts نشان داده می شود و برای استفاده در برنامه فعلی آماده است.

### معرفی مختصات z

مانند مثال قبلی ما یک اسکریپت دیگر با نام scr\_Make ایجاد نمودیم که شامل کد زیر است :

```
//create ball object  
instance_create(64,64,obj_Ball);
```

این کد بعد از اسکریپت (scr\_Start3D) اجرا می شود. این اسکریپت باعث می شود که دوربین ما یک آبجکت توپ را برای ما ایجاد کند. اگر ما دوربین را در room قرار دهیم و برنامه را اجرا کنیم ما یک خطا دریافت خواهیم کرد که به ما می گوید z unknown variable . این خطا بخاطر اینست که Game Maker برای ساخت بازیهای دو بعدی طراحی شده است و مختصات z یک مقدار داخلی در GM نمی باشد. در بازیهای دو بعدی ما تنها به مقادیر x و y نیاز داریم و در بازیهای سه بعدی به مقدار سوم یعنی z نیز نیاز داریم.

ما باید متغیر z را تعریف کنیم زیرا برای برنامه ناشناخته است. بیایید یک اسکریپت برای اینکار ایجاد کنیم تا بتوانیم از آن در آبجکت های آتی خود استفاده کنیم. ما یک اسکریپت با نام scr\_DeclareZ ایجاد می کنیم که بعداً آن را در رویداد Creation آبجکت توپ قرار می دهیم. این اسکریپت کد زیر را برای تعریف z در بر دارد :

```
//declare z value  
z=0;
```

اگر اکنون ما اسکریپت را در آبجکت توپ به طریقی که بالا گفته شد قرار دهیم و برنامه را اجرا کنیم به خوبی کار می کند و ما یک توپ بزرگ آبی در وسط صفحه مشاهده می کنیم.

### اضافه نمودن حرکت در راستای محور z

بیاید آخرین اسکریپت خود را با نام scr\_Motion که از argument0 خود استفاده می کند که شامل مقدار حرکت یا سرعت می باشد را ایجاد کنیم. این اسکریپت به شکل زیر است :

```
//apply motion  
z+=argument0;
```

این اسکریپت باعث می شود که توپ در راستای محور z حرکت کند. ما قطعه کد زیر را در رویداد Step می نویسیم. ما اسکریپت جدایی را برای اینکار در نظر نمی گیریم و فقط به صورت piece of code استفاده می کنیم :

```
//apply motion  
script_execute(scr_Motion,1);
```

اسکریپت scr\_Motion اکنون با مقدار حرکت یا سرعت 1 اجرا می شود. اگر شما برنامه خود را اجرا کنید شما توپ را می بینید که در فضای دور کم کم ناپدید می شود.  
\*\*\*\*\*