# بسم الله الرحمن الرحیم

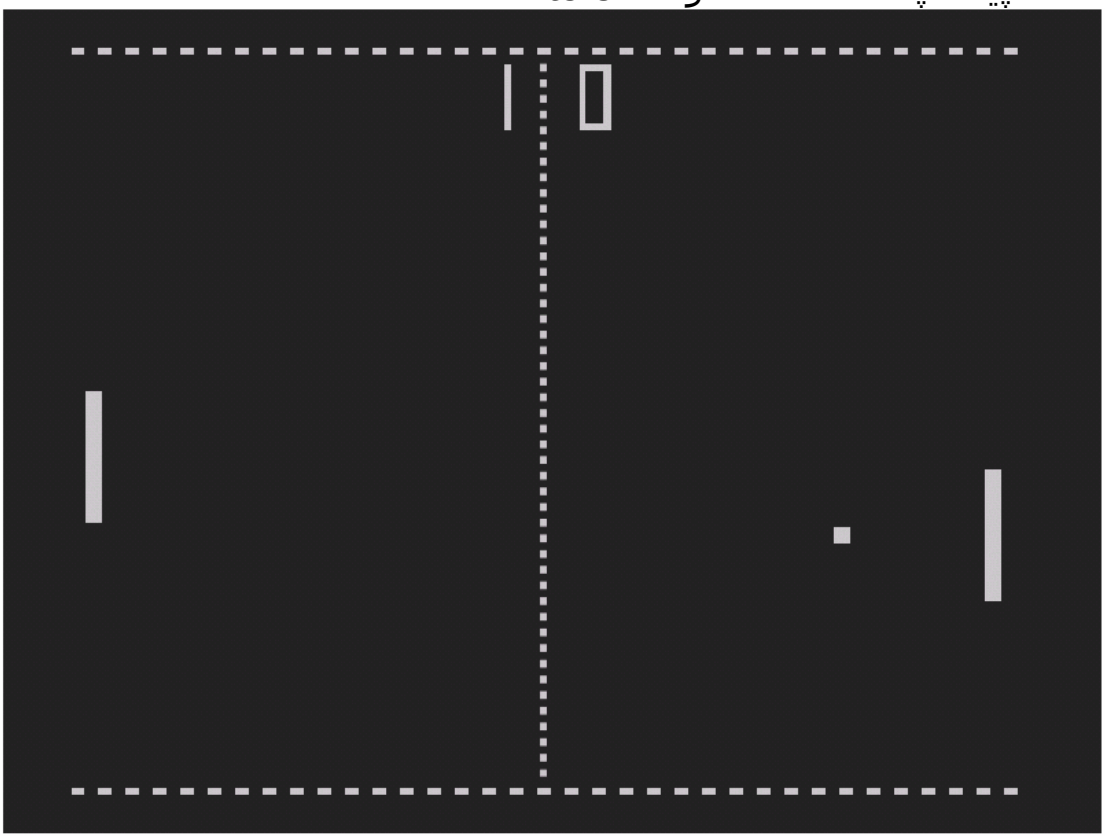
# موضوع مقاله: آموزش برنامه نویسی به زبان LITE C


## مترجم: محمد غضنفری (MAMAD GHA)

Mamad5959@yahoo.com


## بخش دهم: اشاره گرها

یک اشاره گر، یک منبع را به شی ذخیره می کند. تصویر زیر یک بازی ساده پینگ پنگ است که توسط Lite C ساخته شده است.



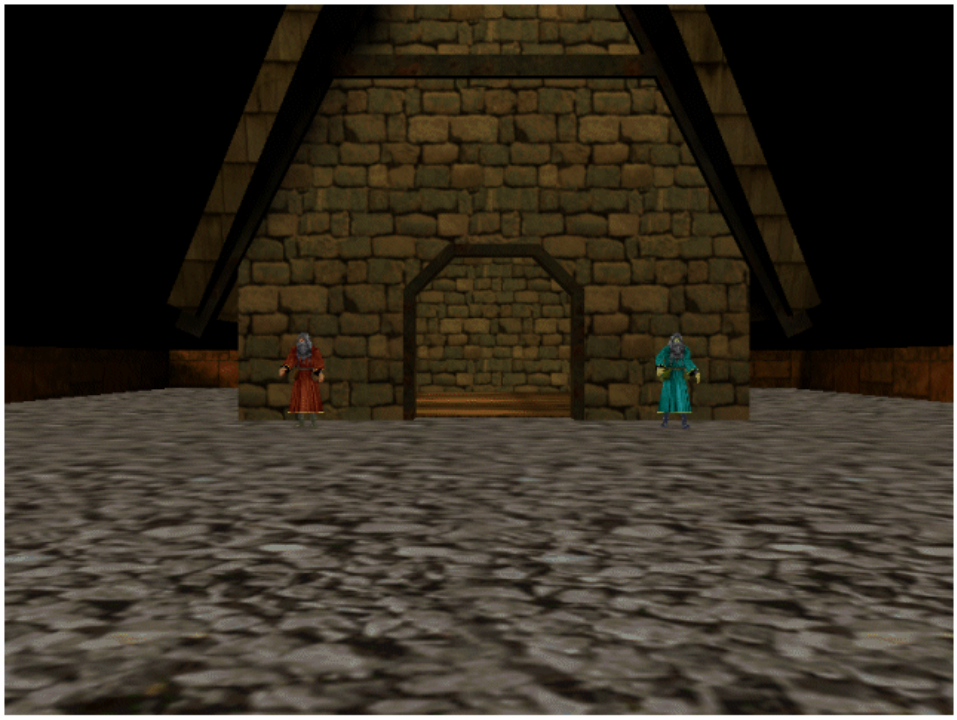
خوب اکنون شما می خواهید که راکت سمت چپ به دست بازیکن کنترل گردد و راکت سمت راست به دست رایانه. سوال و مسئله ما اینجاست که چگونه کد نویسی کنیم که کدام بازیکن است و کدام رایانه؟
جواب ساده است ما باید یک اشاره گر بسازیم که مرجعی برای اشیای دیگر است. اکنون به تعریف یک اشاره گر در پایین نگاهی بیاندازید.

**ENTITY* the_player;**
یا

**STRING* my_name;**
یا
**PANEL* game_over;**
یا

**BMAP* health_picture;**
یا

**var* my_number;**
خوب ما اکنون نگاهی به تعریف یک اشاره گر می اندازیم که بدون ذکر نام شی تعریف گردیده اند. ما به سادگی و با اضافه کردن تنها یک ستاره * به نوع شی می توانیم اشاره گر را برای استفاده آماده کنیم. ما می توانیم برای اشیا، رشته ها، پانلها و... یک اشاره گر تعریف کنیم. و همچنین ما می توانیم به این اشاره گرها یک مقدار بدهیم اگر با این اشاره گرها بخواهیم کار کنیم.
من می دونم که شما الان یک مقدار گیج شده اید پس بهتره بریم سر وقت تمرین و کار با نمونه ها C-LITE را راه بیاندازید و فایل **script11** واقع در پوشه **workshop11** را باز کنید. اکنون کدها را اجرا کنید. در صفحه شما یک خانه را به همراه دو کرکتر می بینید.



این دو تا جادوگر هم شکل هستند و تنها رنگ لباس شان با هم فرق دارد یکی از آنها قرمز هست و اون یکی آبی!
خوب ما اکنون می خواهیم موقعیت یکی از آنها را تغییر دهیم. اکنون بر روی کلید Tab از روی صفحه کلید کلیک کنید و سپس عبارت زیر را تایپ کنید و سپس بر روی کلید Enter از روی صفحه کلید کلیک کنید:

**wizard.x = 300;**

همانطور که می بینید جادوگر قرمز جلو می آید به دوربین نزدیک می شود.

حال این سوال مطرح می شود که چرا جادوگر قرمز به جلو آمد؟ چرا جادوگر آبی جلو نیامد؟ چرا ما نمی توانیم هر دو جادوگر را در یک زمان و به صورت یکسان حرکت دهیم؟ و اصلا چرا خانه و همه چیز به جلو نیامد و تنها این جادوگر قرمز به جلو آمد؟
من می ترسم که نتونم حریف سوالات بشوم! اما تنها یک جواب برای همه سوالات شما خواهم داد و اونم اینه که به این علت که من برای جادوگر قرمز یک اشاره گر تعریف کردم.
می بینید ما داریم در یک جهان بی رحم زندگی می کنیم. این رایانه های کثیف نمی توانند به ما علاقه مند باشند!!! من می دونم که این مدل جادوگر چگونه فکر می کند اما کامپیوتر نمی تواند به آنها علاقه مند باشد. اگر شما به رایانه تان بگویید که مدل جادوگر قرمز را حرکت بدهد رایانه با صدای بلند می خندد و می گوید که من نمی دونم چی داری می گی و نمی توانم به چیزی علاقه مند باشم و اصلا نمی دونم که این مدل جادوگر قرمز رنگی که می گی کجاست من مانند یک موجود بی شعور

کورم که باید همه چیز را با آب و تاب برایم توضیح بدهی. رایانه حتی نمی تواند رنگها را هم از هم تشخیص دهد. برای همین ما احتیاج به این داریم که برای هر موجود مان یک اشاره گر را تعریف کنیم تا هر وقت خواستیم به رایانه بگوییم که فلان موجود را می خواهیم فوری اشاره گر درون آن را به رایانه می گوییم و رایانه می فهمد که مدل کجاست و تغییرات را بر روی آن اعمال می کند.
اکنون بیایید یک نگاهی به کدهای فایل **script11** که خیلی مهم است بیاندازیم.

```
////////////////////////////////////////////////////////////////////////
ENTITY* wizard;
////////////////////////////////////////////////////////////////////////
function main()
{
 video_mode = 7;
 level_load ("work11.wmb");
}
action wizard_with_pointer()
{
 wizard = me;
 my.ambient = 100;
}
action wizard_simple()
{
 my.ambient = 100;
}
```

خدایا شکر بالاخره ما به یک اسکریپتی نگاه کردیم و آن برای مان پیچیده نبود!
خوب ما در ابتدای کد اندازه تصویر را مشخص کردیم و سپس یک مرحله را با نام **work11.wmb** بازخوانی کردیم و بعد از آن دو تا اکشن نوشتیم. که دو تا جادوگر را در ویرایشگر مرحله فراخوانی می کند. و در بالاترین قسمت ما یک اشاره گر را با نام **wizard** تعیین کردیم. بیایید یک نگاه نزدیکتر به این کد بیاندازیم.

**ENTITY* wizard;**

ما یک اشاره گر با نام **wizard** را برای یک موجود تعیین کردیم. در تعیین اسم برای اشاره گر خیلی مراقب باشید و تمام سعیتان بر این باشد که اسم اشاره گر با موجود کاملا همخوانی داشته باشد مثلاً اسم robot_12 نمی تواند برای یک موجود میمون کاربرد داشته باشد. اکنون بیاید اکشن **wizard_with_pointer** را امتحان کنیم.

```
action wizard_with_pointer()
{
   wizard = my;
   my.ambient = 100;
}
```

ما می دانیم که در ابتدا یک اشاره گر با نام wizard را تعریف کردیم درسته!
در اولین خط در اکشن ما یک اشاره گر را ضمیمه یک موجود کردیم.
درست در این خط: **wizard = my;** که my یک اشاره گر از پیش تعریف شده می باشد.
می خواهید اکنون جادوگر ما روشن شود! حدس می زنید چطور؟

اکشن **wizard_with_pointer** ضمیمه مدل جادوگر قرمز رنگ شده است. اکنون این جادوگر ما می توانید با استفاده از روش نقطه کنترل گردد. اکنون دیدید که چگونه خط **wizard.x = 300;** تنها جادوگر قرمز رنگ ما را تکان می دهد و تحت محور X به جلو می راند.
در خط دوم اکشن یک محدوده ای را برای جادوگر قرمز رنگ تعیین می کنیم. که از ۰ تا ۱۰۰ می باشد. که این محدوده نور اطراف جادوگر است که هر چه عدد به ۱۰۰ نزدیک تر باشد این محدوده نور بیشتر است البته اگر عددی بالاتر از ۱۰۰ بزنید طبق آزمایشی که من کردم مدل روشنتر نمی شود. برای درک بیشتر به تصویر زیر نگاه کنید.

این اکشن برای جادوگر آبی رنگ به کار می رود که چون ما قصد نداریم برای این جادوگر اشاره گر تعریف کنیم در نتیجه در اکشن این مدل ما از اشاره گر استفاده نمی کنیم و تنها از ambient برای روشن کردن مدل استفاده می کنیم.

```
action wizard_simple()
{
   my.ambient = 100;
}
```

بسیار خوب، در مورد اشاره گرهای از پیش تعریف شده زیاد نگران نباشد چون ما در انتهای این مقاله در موردش صحبت خواهیم کرد.
خوب اکنون در گام بعدی ما باید متوجه شویم که یک اشاره گر چگونه تعریف می شود؟

۱. ما در ابتدا در یک خط کوتاه یک اشاره گر واحد برای استفاده را تعریف می کنیم.

```
ENTITY* horse; // define an entity pointer
```

۲. اکنون ما توسط یک اکشن به اشاره گر می گوییم که چه کسی باید مالک این اشاره گر باشد و در حقیقت مکان مدل مان کجاست.

```
action players_horse()
{
   horse = my;
   ...
}
```

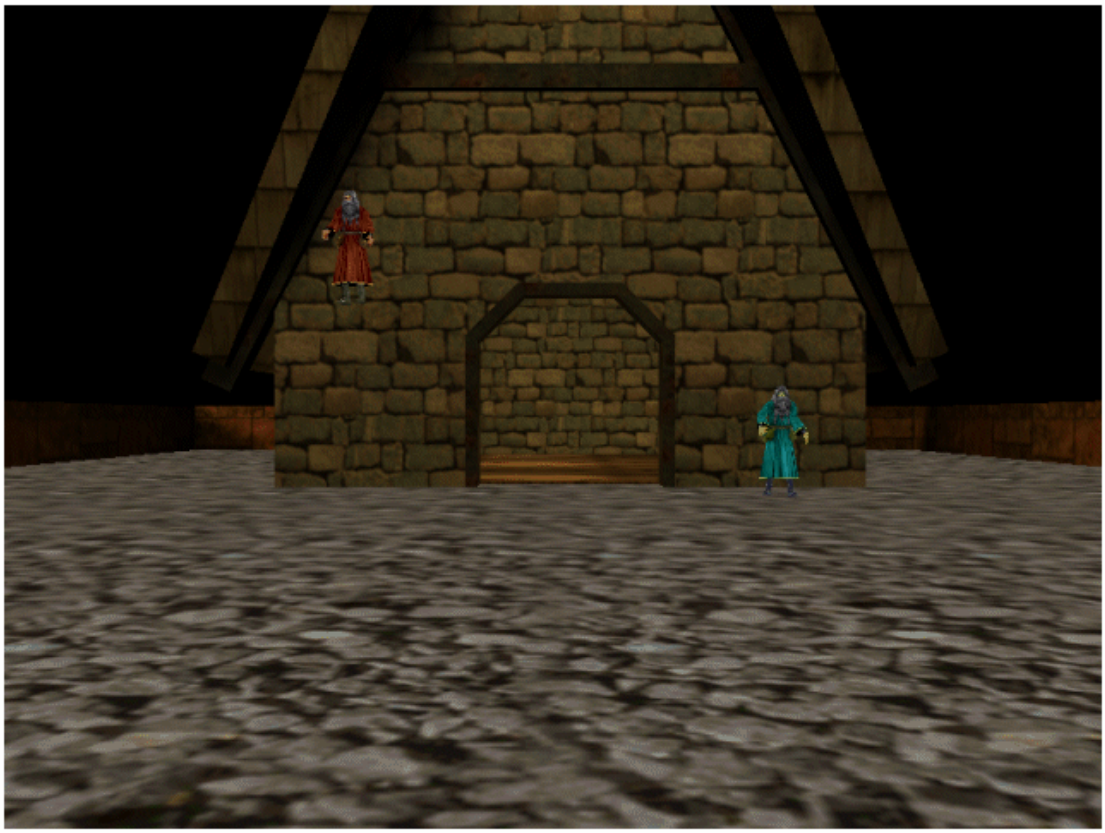
هم اکنون شما می توانید مدل تان را که اکشن **players_horse** برایش تعریف شه با استفاده از روش نقطه کنترل کنید.

```
object.property
```

روش نقطه همین طور که دقت بکنید به این صورت است که در سمت چپ و به جای **object** نام مدل مان را نوشته و سپس یک نقطه می گذاریم و در نهایت خصوصیتی را که می خواهیم برای شی تعریف کنیم به جای عبارت **property** می نویسیم.
اکنون بیایید یک بار دیگر با مرحله تستیمان بازی کنیم پس دوباره فایل **script11** واقع در پوشه **workshop11** را باز کنید. و کدها را اجرا کنید و سپس از طریق کلید Tab از روی صفحه کلید بخش فرمان را در پایین صفحه باز کرده و عبارت زیر را بنویسید.

**wizard.z = 100;**

اگر دقت بکنید در سم چپ ما عبارت **wizard** را می نویسیم که نام جادوگر مان می باشد و سپس یک نقطه گذاشته و عبارت **z = 100** را به معنای اینکه شی باید تحت محور z ۱۰۰ مرتبه حرکت کند را می نویسیم بنده فکر می کنم هم اکنون ما دیگر شیوه بسیار مهم و اساسی این زبان که همان روش یا شیوه نقطه می باشد را فرا گرفتید.
خوب بازیمان را اجرا می کنیم شما باید چیزی شبیه به تصویر زیر را ببینید.



همانطور که دیدید این جادوگر از نظر ارتفاع تغییر کرد همچنین شما می توانید با عدد دادن به **x**, **y**, **z**, **pan**, **tilt**, **roll**... بازی های بیشتری انجام دهید که انجام این بازیها را به خودتان واگذار می کنیم.

اکنون به جاهای خوبی رسیدیم. شما توسط اکشن ها و توابع دیگری می توانید برخی خواص دیگر جادوگر قرمز را تغییر دهید. اکنون تابع زیر را در بالاتر از تابع main بنویسید.

```
function move_up()
{
   wizard.z += 5; // add 5 quants to the height of the wizard each time
when this function is called
}
```

سپس خط کد زیر را به درون تابع main اضافه کنید.

```
on_u = move_up;
```

اکنون کدهای شما باید به شکل زیر باشد.

```
//////////////////////////////////////////////////////
ENTITY* wizard;
//////////////////////////////////////////////////////
function move_up()
{
  wizard.z += 5;
}
function main()
{
  video_mode = 7;
 level_load ("work11.wmb");
  on_u = move_up;
}
action wizard_with_pointer()
{
 wizard = me;
 my.ambient = 100;
}
action wizard_simple()
{
 my.ambient = 100;
}
```

اکنون شما می توانید تابعی با نام **move_up** را ببینید. که ارتفاع جادوگر را (محور Z) ۵ واحد افزایش می دهد. اگر ارتفاع نخست مدل مان ۱۵ کوئنت quant (واحدی شبیه اینچ که در این موتور از آن استفاده می شود) این تابع ارتفاع را ۲۰ کوئنت quant می کند.
اکنون نگاهی به کد زیر بکنید.

```
on_u = move_up;
```

خوب، **on_u** این توانایی را دارد که هنگامی که بازیکن کلید u را در بازی فشار می دهد اکشن مورد نظر شما اجرا گردد که آن اکشن بعد از علامت مساوی= نوشته می شود. که ما در اینجا تابع **move_up()** را برای اجرا بعد از علامت مساوی= می گذاریم.

**نکته مهم: در اینجا استثناً بعد از نوشتن نام تابع دو علامت پرانتز () را نگذارید. و نکته بعدی هم این که چون ما بعد از on_ از حرف یو کوچک u استفاده کردیم در نتیجه بازیکن هم باید هنگام فشردن کلید u دقت کند که دکمه Caps Lock خاموش باشد.**

در نهایت هر زمان که شما دکمه u را از روی صفحه کلید بزنید تابع **move_up()** اجرا می گردد که طبق این تابع مدل جادوگر باید ۵ واحد به بالا افزایش پیدا کند در حقیقت هر زمان که شما دکمه u را فشار دهید جادوگر در بازی به سمت بالا می رود.
در نهایت اینکه my یک اشاره گر کلی می باشد. که می توانید به هر چیزی نسبت داده شود و از طریق روش نقطه آن را کنترل کرد. این اشاره گر می تواند برای هر چیزی تعریف شود مثلا برای یک اسپریت یا برای یک موجود wmp و ...
مثلا شما می توانید در اکشن جادوگر آبی رنگ که این کد می باشد.

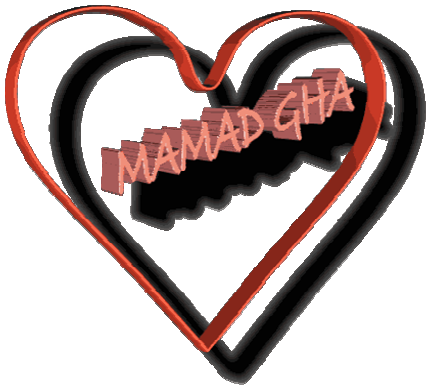
```
action wizard_simple()
{
   my.ambient = 100;
}
```

کد زیر را به این اکشن اضافه کنید و آنوقت بازی را اجرا کنید و خودتان نتیجه را ببینید.

```
my.z = 100;
```

خوب دوستان ما به انتهای این قسمت رسیدیم و امیدوارم که با تمرین بتوانید خودتان را آماده تر از قبل بکنید. در مبحث بعد ما به بخش بسیار مهم عبارات شرطی خواهیم رسید پس خودتان را برای قسمت بعدی آماده کنید.

**مترجم: محمد غضنفری (MAMAD GHA)**
**Mamad5959@yahoo.com**




## منبع:

**مجله رسمی موتور AKCNEX**
**Acknex User Magazine (aum)**
**http://aum.conitec.net**

**و برخی اطلاعاتی که از قبل داشتم.**

هرگونه کپی برداری از این اثر تنها با **ذکر منبع** و **نام ناشر و مترجم** مجاز می باشد.

## ناشر:

**http://futureworld.ir**