

The Dark Basic Swarm (قسمت اول)

آموزش ساخت بازی کامل Swarm

اسماعیل رادپور

eshagh@spymac.com

WWW.Persian-Designers.COM

این آموزش راهنمای قدم به قدمی است تا پایان ساخت یک بازی کامل کامپیوتری ، داستان یک حمله فضایی و تدبیری برای دفاع از زمین. پس اسم این بازی را SWARM می گذاریم!

مقدمه:

Dark Basic از DirectX برای پرداخت بازیهایی که شما از کامپیوترهای امروزی انتظار دارید استفاده می کند(که شامل بازیهای 3D هم می شود!). بعد از نصب Dark Basic شما باید زبان برنامه نویسی آن را یاد بگیرید. در این آموزش تا آنجا که امکانش بوده ، سعی شده تا مطالب برای همه سطحی قابل فهم باشد اما حداقل خواندن مقاله آشنایی با دارک بیسیک الزامی است. اصل این آموزش در Help نسخه های قدیمی دارک بیسیک کلاسیک موجود بوده ، اما به هر دلیل در نسخه ای که شما از آن استفاده می کنید فقط بازی آماده Swarm و کدهای آن موجود است. تا جایی که در توانم بوده ، بنا بر درکی که از آشنایی دوستان با این برنامه دارند ، مطالبی مضاف بر اصل آموزش هم به آن افزودم. اما در هر صورت:

این مقالات ترجمه سری آموزش های "The Dark Basic Swarm" است.

ساخت یک پروژه جدید:

فایل SWARMTUT.DBA را که همراه این جزوه دریافت کرده اید را در داخل ویرایشگر باز کنید. اکنون چند خط کد می بینید که اگر هم اجرایشان کنید ، مطلقاً هیچ اتفاقی نمی افتد. کدهایی که مشاهده می کنید را ملاحظات(remark) می نامند. این دسته سطرها فقط برای یادآوری به زبان انگلیسی(و نه زبان برنامه نویسی) برای برنامه نویس بکار می رود ، که در واقع برنامه را شرح میدهد.

اضافه کردن یک حلقه اصلی:

همه برنامه ها بطور کلی احتیاج به یک حلقه اصلی دارند(۱). یک حلقه اصلی از دو نقطه آغاز و پایان در قبل و بعد از جاییکه اشاره گر برنامه باید به شکل حلقه وار باشد (یعنی کدهایی را بطور مدام تا پایان بازی اجرا کند) قرار می گیرند.می توانید تصور کنید که اشاره گر برنامه یک پیکان نامرئی است و در نقطه ای قرار می گیرد که دستوری در حال اجرا شدن است و بعد از آن به خط بعدی می رود. وقتی این اشاره گر به نقطه پایان حلقه رسید ، برای ادامه چرخش به نقطه شروع حلقه بازمی گردد.

خطی که نوشته شده است **REM *HERE*** را انتخاب و آن را پاک کنید ، حالا به جای آن ، این عبارت را بنویسید: **HIDE MOUSE** اکنون کلید F4 را برای چک کردن برنامه فشار دهید. باید یک پیغام "Compilation Successful" در پایین صفحه دریافت کنید. کدها را از خطی که تغییر داده اید ، به شکل زیر اصلاح کنید:

```
rem Initialise program
HIDE MOUSE
```

```
rem Main game loop
DO
rem End of game loop
LOOP
```

کلید F5 را برای اجرای برنامه بزنید ، اگر با موفقیت اجرا شود ، مشاهده می کنید که اشاره گر ماوس رفته و سیاهی صفحه مانده!

برنامه تان اکنون در حال اجرای یک حلقه اصلی است. می توانید با کلید F12 به ویرایشگر باز گردید یا ESC را بزنید و روی Exit کلیک کنید.
برای وارد کردن مستقیم همه کدها می توانید فایل SWARM-01.DBA را باز کنید.

ساخت یک پرده پشت صحنه:

اگر همه کدهای این قسمت را می خواهید فایل SWARM-03.DBA را باز کنید و اگر می خواهید مرحله به مرحله با ما پیش روید ، فایل SWARM-02.DBA را باز کنید.
ساخت پرده پشت صحنه(Backdrop) به خاطر جذابیت بیشتر به بازیمن است که چند تا ستاره به جای صفحه سیاه کنونی قرار دهیم. تعدادی فایل های رسانه ای همراه این آموزش در پوشه های WAV ، OBJ و BMP قرار داده ایم. این پوشه ها محتوی فایل های گرافیکی و صوتی است که برای ساخت این بازی پیش بینی شده است.
در پوشه BMP فایلی به اسم STARS.BMP وجود دارد که تصویری است از ستاره ها. ابتدا نیاز داریم که این فایل را داخل حافظه موقت load کنیم و به عنوان یک IMAGE ذخیره کنیم.
اولین خط **REM *HERE*** را با کدهای زیر عوض کنید:

```
rem Load stars image  
LOAD IMAGE "bmp\stars.bmp",1
```

اگر برنامه را با فشردن کلید F4 چک کنید مطمئنا از Syntax Error ها خالی خواهد بود. Syntax Error هنگامی رخ می دهد که DB چیزی را که تایپ کرده اید ، شناسد که اغلب به خاطر غلط های املایی در دستورها رخ می دهد.
حالا که می خواهید ستاره هایتان را تماشا کنید خیلی ساده دو دستور دیگر به برنامه اضافه می کنیم.

دومین خط **REM *HERE*** را با کدهای زیر عوض کنید:

```
rem Create star backdrop  
BACKDROP ON  
TEXTURE BACKDROP 1
```

با اولین خط **BACKDROP ON** پرده پشت صحنه با رنگ پیش فرض آبی صفحه را پر می کند. دستور **TEXTURE BACKDROP** عکس ستاره ها را (که در ابتدا load کرده بودیم) با رنگ آبی جایگزین می کند.

اگر دستورهایمان را با استفاده از زیرروال ها(Subroutine) گروه بندی کنیم برنامه مان دلپذیر تر به نظر می آید.

دو تا دستور آخر را با کدهای زیر عوض کنید:

```
rem Create star backdrop  
GOSUB _create_backdrop
```

سومین خط **REM *HERE*** را با کدهای زیر عوض کنید:

```
rem Texture backdrop with stars image  
BACKDROP ON  
TEXTURE BACKDROP 1
```

مشاهده کردید که چگونه دستورات مربوط به backdrop را داخل زیرروالی به اسم **_create_backdrop** بردیم. دستور **GOSUB** به برنامه می گوید به نقطه آغاز زیرروال برود و محتویات آن را اجرا کند. نهایتا اشاره گر برنامه به دستور **RETURN** می رسد ، این دستور اشاره گر را به جایی برمیگرداند که از آن آمده بود. زیرروال ها و **GOSUB** روش بسیار خوبی برای سازمان دادن به برنامه تان است.

Scroll کردن پرده پشت صحنه:

اگر همه کدهای این قسمت را می خواهید فایل SWARM-05.DBA را باز کنید و اگر می خواهید مرحله به مرحله با ما پیش روید ، فایل SWARM-04.DBA را باز کنید.
حالا در محل بازی یک پرده پشت صحنه یا backdrop داریم که می خواهیم تحرکی به آن بدهیم ، تا احساسی به بازیکن دست دهد که گویی صفحه در حال حرکت است. به عبارتی می خواهیم backdrop را scroll کنیم.

خط **REM *HERE B*** را با کدهای زیر عوض کنید:

```
rem Control star backdrop  
GOSUB _control_backdrop
```

خط **REM *HERE D*** را با کدهای زیر عوض کنید:

```
rem Scroll star backdrop  
INC starscrolly#,1.0  
SCROLL BACKDROP 0,0-starscrolly#
```

همچنانکه مشاهده می کنید ما یک زیرروال جداگانه برای متحرک کردن backdrop به نام **control_backdrop** بکار بردیم. بوسیله قرار دادن **GOSUB** در میان حلقه اصلی مان می خواهیم بگوییم که این زیرروال پشت سر هم تکرار شود.

حرکت های بازی ها ، در هربار اجرا شدن کوتاه و مختصر صورت می گیرد ، چیزی که ما برای حرکت متوالی backdrop می خواهیم نیز یک حرکت کوتاه در میان حلقه بازی است – یعنی از تکرار متوالی حرکت های کوتاه ، یک حرکت صاف و یکدست بدست می آید.

برای نگهداری مقدار این تغییرحرکت های کوتاه پرده پشت صحنه ، از یک متغیر برای ذخیره موقعیت جاری scroll استفاده شده است. متغیر **starscrolly#** یک مقدار را نگهداری می کند که به ما می گوید چه مقدار backdrop باید حرکت کند. دستور **INC** متغیری را نسبت به مقداری که نگهداری می کند ، افزایش می دهد ، در مورد این متغیر ، مقدار ۱,۰ را در هربار اضافه می کند – مثلاً در دور اول حلقه بازی ، مقدار متغیر برابر ۱ است و در دور پنجم برابر ۵.

دستور **SCROLL BACKDROP** مقدار یک متغیر را گرفته و آن را در تنظیم موقعیت کنونی backdrop بکار گرفتیم. هر دو دستور **INC** و **SCROLL BACKDROP** دو پارامتر می گیرند. پارامترها مقدارهایی هستند که دستور (Command) مورد نظر برای انجام دادن کاری که به آن گفته می شود ، احتیاج دارند. در مورد دستور **INC** ، کامپایلر نیاز به شناخت یک متغیر برای افزایش پیدا کردن و یک مقدار برای افزایش دادن بوسیله آن ، دارد. در مورد دستور **SCROLL BACKDROP** ، کامپایلر برای انجام دستور احتیاج به شناخت مقدار افقی و عمودی scroll دارد. همانطور که مشاهده کردید ، ما مقدار scroll x را برابر صفر ، اما مقدار y آن را مبتنی بر متغیر **starscrolly#** تنظیم کردیم.

چون ما می خواهیم backdrop به سمت پایین حرکت کند نه بالا ، مقدار را بوسیله کم کردن از صفر ، منفی کردیم (با استفاده از فرمول **0-starscrolly#**) حالا نیاز داریم تا حرکت را صاف و یکدست کنیم. این کار را بوسیله کنترل بروز شدن صفحه می توانیم انجام دهیم.

بجای خط **REM *HERE A*** این عبارت را قرار دهید : **SYNC ON**

بجای خط **REM *HERE C*** این عبارت را قرار دهید : **SYNC**

این دو عبارت به کامپایلر گوش زد می کنند که فقط هنگامی که به پایان حلقه اصلی رسید ، صحنه را بروز (update) کند. بطور پیش فرض کامپایلر شما بعد از هر دستور صحنه را بروز می کند که این باعث کم شدن سرعت و زیاد شدن پیچیدگی بازی می شود.

منتظر قسمت بعدی باشید...

پانوشت ها _____
۱- برای اطلاعات بیشتر، رجوع شود به مقاله حلقه اصلی بازی واقع در سایت [PD](#) ، قسمت دریافت فایل و بخش مقالات متفرقه.

کلیه حقوق این مقاله متعلق به نویسنده و سایت Persian Designers می باشد

استفاده از مطالب این مقاله در صورت ذکر ماخذ ، بلا مانع است

سایت طراحان ایرانی با هدف آموزش ساخت بازیهای کامپیوتری به زبان فارسی طراحی شده است و تا کنون مقالات متعددی در زمینه های مختلف برنامه نویسی و ساخت بازی در آن قرار گرفته است. مدیریت سایت از تمامی عزیزان علاقمند به بازی های کامپیوتری ، برنامه نویسان ، طراحان و سایر کسانی که به نحوی با بازی ها در ارتباطند ، دعوت به همکاری به عمل می آورد تا بدینوسیله یک پایگاه علمی و موثق در زمینه صنعت ساخت بازیهای کامپیوتری در ایران ایجاد گردد.

در ضمن بسیاری از نرم افزار های ساخت بازی های کامپیوتری که امروزه در سطح وسیع مورد استفاده قرار میگیرند ، در سایت جمع آوری شده است و با مبلغ بسیار ناچیزی در اختیار علاقمندان به طراحی بازی های کامپیوتری قرار داده شده است. استفاده از این نرم افزار ها در آغاز کار و به منظور آشنا شدن با اصول اولیه در طراحی بازیها بسیار موثر و مفید بوده و شما میتوانید تا با چند جستجوی ساده در این زمینه ، به صحت موضوع پی ببرید. لیست زیر برخی از نرم افزار هایی هستند که توسط فروشگاه الکترونیکی سایت به مشتاقان عرضه میشوند :

- Game Maker Version 5.0 – 5.1 – 5.2 – 5.3 – 6.0 (Registered) •
- The Game Factory (Home – Professional) (registered) •
- Xtereme 3D 1.0 •
- King Space 3D •
- Genesis 3D V1.6 •
- 3D Game Studio 5.12 •
- 3D State (Morfit 3D) (Registered) •
- Blender 3D •
- Q3d (Unregistered) •
- Alice 3D •
- True Vision 3D V6.2 •
- DirectX 9.0 Complete SDK (Software Development Kit Package) •

و موتور های سورس باز ۳ بعدی زیر :

- Jolt3D (Third Person Maker) Demo & Source •
- Boom3D (Demo & Source) •
- PushTheLimits 3D Engine •
- Mystica 3D •
- Arfnold 3D (Source & Bin) •
- TerraCresta 3D (Demo & Source) •
- XEngine (Sample & Source) •
- Muli3D (Sample & Source) •
- Apocalyx (Source) •
- Graden (Source) •
- DXQuake (Source) •
- 6DX •
- CHAI 3D (Source) •
- Axiom 3D •
- syBR (Source) •
- iRender 3D (SDK) •
- Cube 3D (Source & SDK) •
- Q Engine •
- Hawk 3D Engine (Source & Bin) •
- Neo Engine (Tools & Source & Tutorials) •
- Aurora (installer & Tutorials) •
- Soya 3D •
- DexVT (Source) •
- Jet 3D (Source & Bin) •
- Traktor 3D SDK (Source & Bin) •
- NemoX (Installer) •
- Unreal 2 (SDK) •
- Irrlicht 3D (SDK & Source) •

و کامپایلر های

- Visual Basic V6.0 •
- Visual C++ V6.0 •

لینک فروشگاه الکترونیکی سایت طراحان ایرانی :

WWW.Persian-Designers.COM/index.php?pid=1