

به نام خدا

با سلام.

ما در این جزوه قصد داریم تا طریقه برنامه نویسی مقدماتی نرم افزار معروف 3D Game Studio را آموزش دهیم.

یکی از بازی های جدید ساخته شده توسط این نرم افزار ، بازی جامهای طلایی می باشد که یک بازی کاملاً فارسی است.

این بازی را می توانید از وبلاگ زیر تهیه نمایید.

www.Jamhayetalace.Parsianblog.Com



فایلهای برنامه نویسی شده این برنامه با پسوند Wdl و فایلهای مرامل با پسوند Wmp و فایلهای کامپایل شده مرامل با پسوند Wmb ذخیره می شوند.

توجه : شما قبل از فراگیری کد نویسی ، باید متماً با نرم افزارهای Model Editor و Level Editor آشنا شده باشید.

شما برای باز کردن فایلهای Wdl می توانید از محیط کد نویسی خود نرم افزار استفاده کرده و یا با استفاده از برنامه

Notepad این کار را انجام دهید.

برای شروع کد نویسی ابتدا تمامی کدهایی که در فایل مزبور مشاهده می کنید را پاک کنید و یا یک فایل جدید باز کرده و در آخر ، این فایل را با نام همان فایل Wdl ذخیره کنید.

ما در این جزوه ، نسخه ۶٫۱ از نرم افزار 3D Game Studio را آموزش می دهیم .

شما این نرم افزار را می توانید از سایت www.Persian-Designers.Com تهیه کنید.

در این نسخه از نرم افزار ، محیط کد نویسی بصورت جداگانه طراحی شده است.

پس به راحتی می توانید برنامه Script Editor را باز کرده و در آن به کد نویسی بپردازید.

فهرست

فصل ۱ صفحه ۳ (شروع برنامه نویسی)
فصل ۲ صفحه ۶ (کنترل توسط صفحه کلید)
فصل ۳ صفحه ۹ (بارگذاری انیمیشن)
فصل ۴ صفحه ۱۱ (ایجاد اشیاء در بازی)
فصل ۵ صفحه ۱۳ (حرکت اشیاء در هر زاویه ای)
فصل ۶ صفحه ۱۵ (دوربین)
فصل ۷ صفحه ۱۷ (رویدادها)
فصل ۸ صفحه ۱۹ (صفحه ها و متون)
فصل ۹ صفحه ۲۲ (پایان)

فصل ۱ : شروع برنامه نویسی

قبل از نوشتن هر کدی ابتدا باید تنظیمات صفحه نمایش را انجام دهیم :

```
Var video_mode = 6;  
Var video_depth = 16;
```

فقط اول رزولوشن صفحه را ۸۰۰*۶۰۰ قرار می دهد و فقط دوم بیت های نمایشی را به ۱۶ بیت تغییر می دهد. البته این مقادیر قابل تغییر هستند .

مالا برای بالا آمدن بازی باید متما در تمام بازی ها یک تابع اصلی به نام (main) وجود داشته باشد. تعریف این تابع بصورت زیر است :

```
Function main(){  
  
}
```

ما می توانیم هر چیزی را که قرار است ابتدای شروع بازی انجام شود ، در این تابع قرار دهیم. مثلا برای بارگذاری کردن مرحله اول بازی خود با نام Level1 عملیات زیر را انجام می دهیم :

```
Function main(){  
    Level_Load( ' Level1.wmb ' );  
}
```

تابع Level_Load() ، تابعی است که می تواند مراحل بازی را بارگذاری کند. بطوری که نام مرحله را همراه با علامت ' ، داخل این تابع قرار می دهیم.

با این کار مرحله ما بارگذاری شده و بازی ، اجرا می شود.

Var : برای تعریف متغیر ها استفاده می شود.

```
Var Counter;  
Var Counter = 0;
```

برای تعریف توابع بصورت زیر عمل می کنیم :

```
{ نام تابع تابع Function  
; دستورات  
}
```

در این آموزش به تدریج با بعضی از توابع این نرم افزار آشنا می شویم ولی باقی توابع را خود شما می توانید از Manual خود نرم افزار ، پیدا کند.

برای ایجاد یک Action یا عمل برای کراکتر خود باید بصورت زیر عمل کنید :

```
{ نام اکشن Action  
; دستورات  
}
```

برای استفاده از خصوصیات کراکتری که اکشن را به او نسبت داده اید ، می توانید از اشاره گر My استفاده کنید.

در زیر بعضی از خصوصیت های My را ملاحظه می کنید :

My . Pan	چرخش به دور خود
My . Tilt	چرخش به بالا و پایین
My . Roll	چرخش به چپ و راست
My . Flare	براق و بلوری کردن شیء
My . Bright	نورانی کردن شیء
My . Alpha	میزان ممو بودن شیء
My . Transparent	توانایی ممو شدن
My . Opacity	توانایی حذف کردن قسمت های مشکلی
My . x	x ممور
My . y	y ممور
My . z	z ممور
My . Event	مسیر عملکرد (فتارها)
...	

بقیه خصوصیت ها را نیز می توانید از Manual خود برنامه پیدا کنید.

استفاده از ملقه های تکرار بصورت زیر است :

```
While ( شرط ) {  
; دستورات  
}
```

حالا ما می خواهیم برنامه ای بنویسیم که توسط آن یک شیء به طور ممتد ، به دور خود بچرخد :

```
Var video_mode = 6;
Var video_depth = 16;

Action Object_Rotate {
    While ( 1 ) {
        My . Pan += 1;
        Wait ( 1 );
    }
}

Function main() {
    Level_Load ( ' Test_Level.wmb ' );
    Wait ( 1 );
}
```

در حلقه تکرار ما از عدد ۱ بعنوان شرط استفاده کردیم و دقت داشته باشید که اعداد جزو شرطهای همیشه درست هستند ؛ بنابراین این حلقه ، همیشه تکرار می شود.

استفاده از تابع Wait () باعث ایجاد یک توقف فیزی کوتاه برای دیدن تغییرات می شود. البته میزان توقف بستگی به عدد داخل تابع دارد.

شما می توانید این کد را ذخیره کرده و آن را امتحان کنید ؛ خواهید دید کراکتری که اکشن Object_Rotate را به آن نسبت داده اید ، به طور ممتد دور خود می چرخد.

تمرینات فصل ۱ :

- ۱) بازی طراحی کنید که در آن ، دو مکعب ، در جهت مخالف یکدیگر بچرخند.
- ۲) بازی طراحی کنید که در آن ، یک مکعب با سرعت فیزی کم بچرخد و به تدریج سرعت آن بالا برود.
- ۳) بازی طراحی کنید که در هنگام اجرا ، یک مکعب در یک زاویه تصادفی قرار بگیرد.
(با استفاده از تابع Random (Number))
- ۴) بازی طراحی کنید که ۳ توپ با مشخصات زیر قرار داشته باشند :
 - ۱- اولی به دور خود بچرخد.
 - ۲- دومی بصورت بلوری در محور Z به بالا و پایین برود.
 - ۳- سومی به دور خود بچرخد و وقتی ۹۰ درجه چرخد ، در جهت مخالف بچرخد.

فصل ۲ : کنترل توسط صفحه کلید

برای کنترل هر چیزی توسط صفحه کلید ، احتیاج به تشخیص فشردن شدن آن کلید وجود دارد. ما برای این کار در نرم افزار 3D Game Studio ، از دستورات زیر استفاده می کنیم.

یکی از این دستورات ، Key است.

برای استفاده از این دستور باید بصورت زیر عمل کنیم :

مثلا برای تشخیص فشردن کلید Space ، می نویسیم : Key_Space
و یا هر کلید دیگری ، چه مرفی و چه کنترلی .

این دستور ، مقدار On یا Off را بر می گرداند. اگر مقدار برگشتی On بود ، یعنی آن کلید فشار داده شده است و اگر Off بود ، یعنی آن کلید فشار داده نشده است.

نکته : دستورات فارچ از ملقه های تکرار ، فقط یک بار اجرا می شوند.

Key_Any : این متغیر ، به ما می گوید که آیا کلیدی فشار داده شده است یا خیر.

اگر این متغیر ، مقدار ۰ را برگرداند ، یعنی هیچ کلیدی فشار داده نشده است ولی اگر مقدار ۱ را برگرداند یعنی کلیدی فشار داده شده است.

(KeyAscii) Key_Pressed : با این تابع می توانید متوجه شوید که آیا کلیدی که کد اسکی آن را درون این تابع نوشته اید ، فشار داده شده است یا خیر .
این تابع نیز همانند متغیر Key ، مقادیر On و Off را بر می گرداند.

این موارد ، پر کاربرد ترین متغیر ها و توابعی هستند که مورد استفاده قرار می گیرند.
اگر مایل بودید می توانید موارد دیگر را از Manual نرم افزار ، پیدا کنید.

ما بعنوان مثال برنامه کوچکی می نویسیم که در آن با فشردن کلید Enter ، بازی تمام شود.

```
Var video_mode = 6;
Var video_depth = 16;

Function Test_Control() {
    While (1) {
        If ( key_enter == on ) {
            Exit ;
        }
        Wait (1);
    }
}

Function main () {
    Level_load ( ' Test.wmb ' );
    Wait (1);
}
```

تابع Exit ، باعث خروج از بازی می شود .

در این مثال شما ساختار دستور If را نیز یاد گرفتید. شما از If ، برای دستورات شرطی استفاده خواهید کرد.

```
If ( شرط ) {
    دستورات ;
}
```

```
If ( شرط ) {
    دستورات ;
} else {
    دستورات ;
}
```

در قسمت Else ، شما باید دستوراتی را بنویسید که در صورت غلط بودن شرط اول ، اجرا شوند.

علامات مورد استفاده در دستورها شرطی :

==	مساوی
!=	نا مساوی
	یا
&&	و

تمرین ها فصل ۲ :

- (۱) بازی طراحی کنید که در آن ، یک توپ وجود داشته باشید که در صورت فشردن کلید Enter ، توپ به پرفش در آید و در صورت فشردن کلید Space ، توپ بایستد.
- (۲) بازی طراحی کنید که در آن یک مکعب وجود داشته باشد که در جهت محور x ها ، به جلو و عقب رفته و در صورت فشردن هر کلیدی بجز کلید Enter بایستد و با فشردن کلید Enter ، بازی تمام شود.
- (۳) بازی طراحی کنید که کراکتر بازی توسط کلیدهای جهت دار به چهار جهت اصلی ، حرکت داده شود.
- (کد کلید های پپ ، راست ، بالا و پایین به ترتیب = Key_Cud , Key_Cuu , Key_Cur , Key_Cul)
- (۴) بازی طراحی کنید که ۳ توپ در آن وجود داشته باشد و با فشردن کلیدهای ۱ تا ۳ ، توپ های مربوطه ، به سمت بالا حرکت کرده و در صورت رها کردن آن کلید ، توپ مزبور ، به ممل اول خود بازگردد.

فصل ۳ : بارگذاری انیمیشن

منظور از بارگذاری انیمیشن ، استفاده از انیمیشنی است که شما آن را برای کراکتر خود ، در Model Editor ، طراحی کرده اید.

شما برای طراحی حرکات ، بسته به نوع حرکت ، نام هایی را انتخاب می کنید.
مثلا Walk و یا Run .

حالا ما می خواهیم که این انیمیشن ها را در بازی ، مورد استفاده قرار دهیم.

برای استفاده از انیمیشن کراکتر ، از دستور Ent_Animate استفاده می کنیم.
در غیر این صورت نام اشاره گر مورد نظر را قرار می دهیم. (در فصلهای بعد با اشاره گرها آشنا می شوید.)
; (حالت حرکت , درصد حرکت , نام حرکت , نام موجودیت) Ent_animate

اگر بخواهم کراکتر مربوط به اکشن را حرکت دهیم ، بجای نام موجودیت ، از My استفاده می کنید.
نام حرکت ، بسته به حرکات موجود در انیمیشن کراکتر ، نام همان فریم ها را می گیرد.
مثلا " Walk " .

دقت کنید که نام حرکت را باید در داخل علامت " ، قرار دهیم.
درصد حرکت ، مقدار و سرعت انیمیشن ما را تعیین می کند.
مثلا ما می توانیم از متغیری استفاده کنیم که مقدار آن به طور ممتد ، اضافه شود.
بجای حالت حرکت نیز می توانیم از دستورهایی مانند anm_cycle استفاده کنیم.
این دستور ، انیمیشن را به صورت ممتد اجرا می کند ، مثلا برای حرکت راه رفتن.

برای مثال ، ما اکشنی می نویسیم که در آن کراکتر با سرعتی ثابت ، انیمیشن راه رفتن را اجرا کند.

```
Var speed = 0;  
Action player_1 {  
    While ( 1 ) {  
        Speed += 8 * time;  
        Ent_animate ( my , "walk" , speed , anm_cycle );  
        Wait ( 1 );  
    }  
}
```

Time ، باعث هماهنگی در مقداردهی ، در CPU های مختلف می شود و زمان آفرین فریم را بر می گرداند.

تمرین های فصل ۳ :

- ۱) بازی طراحی کنید که کراکتر ، در ممور x ها بصورت راه رفتن ، حرکت کند.
 - ۲) بازی طراحی کنید که کراکتر در صورت فشردن کلید Space ، بدود.
 - ۳) بازی طراحی کنید که کراکتر در یک خط به جلو راه برود و دوباره برگردد و این عمل را تا فشردن کلید Enter ، تکرار کند.
 - ۴) بازی طراحی کنید که در آن کراکتر به طور ممتد راه برود و در صورت فشردن کلید D ، کراکتر ، بمیرد.
- (برای انجام یک انیمیشن برای یک بار ، در صد حرکت را از ۱ تا ۱۰۰ در نظر بگیرید و فریم ۱۰۰ ، آفرین فریم می باشد و انیمیشن را در همان جا متوقف کنید.)

فصل ۴ : ایجاد اشیاء در بازی

فیلی مواقع ، پیش می آید که می فوایم ، شیئی را در طول بازی ایجاد کنیم .
مثلا تیری که از یک تفنگ شلیک می شود یا قطرات خون یا باران یا برف و غیره بصورت شیء .

برای ساختن اشیاء در بازی از دستور ent_create استفاده می کنیم.

; (تابع کنترل کننده , محل ساخته شدن , مسیر مدل) Ent_create

برای مدل ، باید از فایل مدل های ساخته شده استفاده کنیم و آنها را خارج از اکشن به صورت زیر
تعریف کنیم :

String Test_Model = <T.mdl> ;

سپس بجای مسیر مدل ، نام آن را بنویسید .
مثلا نام مدل تعریف شده ما ، Test_Model می باشد.

در قسمت محل ساخته شدن ، باید برداری شامل سه محور x , y , z را قرار دهیم.
برای ایجاد یک بردار از دستور زیر استفاده می کنیم :

Vector (x , y , x);

بجای استفاده مستقیم از اعداد ، می توانیم از متغیرهای حاوی این اطلاعات نیز استفاده کنیم.

تابع کنترل کننده هم ، نام تابعی است که شما آن را برای کنترل شیء ساخته شده ، ایجاد می کنید.
اگر نمی فوایید که تابعی ، شیء شما را کنترل کند ، می توانید از کلمه کلیدی Null به معنای فالی
استفاده کنید.

ما در ساخت اشیاء جدید از اشاره گر های جدید مانند My نیز استفاده می کنیم.
My همیشه به کراکتری اشاره می کند که اکشن به آن نسبت داده می شود.
مالا ما می فوایم اشاره گری با نام دلفواه خود ایجاد کنیم که شیء ساخته شده را با آن صدا کرده و
فصوصیاتش را در دسترس داشته باشیم.
برای تعریف یک اشاره گر جدید ، بصورت زیر عمل می کنیم :

; نام اشاره گر Entity*

شما از این نام ، می توانید در اکشن ها و توابعی که مربوط به این شیء نیستند نیز استفاده کنید و
فصوصیات آن را همواره در دسترس داشته باشید.

برای قرار دادن شیء جدید خود در اشاره گر ، فقط کافیست که نام اشاره گر را مساوی دستور
ent_create قرار دهید.

مثلا :

```
String B = <bomb.mdl>;  
Entity* Test_Ent;  
  
Action Bomb {  
    Test_Ent = Ent_create ( B,my.x,Null );  
}
```

ما در این مثال بجای بردار جدید ، از بردار اشاره گر My استفاده کردیم.
اگر در محل استفاده از بردار ، جلوی نام اشاره گر ، از محور x استفاده کنید ، منظور تمام محورهای
x,y,z است.
در این مثال یک گلوله دقیقا در محل خود کراکتر ، ایجاد می شود.
وقتی در تابعی که برای کنترل شیء قرار می دهید ، از اشاره گر My استفاده کنید ، منظور همان شیء
جدید است.

تمرین های فصل ۴ :

- ۱) بازی طراحی کنید که در این بازی یک تفنگ وجود داشته باشد و با زدن هر بار کلید Space ،
یک تیر از تفنگ فارچ شود و در محور x ها حرکت کند.
- ۲) بازی طراحی کنید که از یک توپ ، توپ های هم شکل تولید شوند و هر کدام یک وضوح
تصادفی داشته باشند.
(منظور از وضوح ، همان Alpha است. دقت کنید برای استفاده از Alpha ، متما Transparent شیء ،
باید On باشد.)
- ۳) بازی طراحی کنید که یک دایره کوچک در صفحه موجود باشد و با زدن کلید های ۱ تا ۳ ، به
ترتیب ، مدلهایی از دایره ، مکعب و مخروط تولید شده و به سمت بالا حرکت کنند و به
مرور از میزان وضوح آنها کاسته شود تا کاملا محو شوند و بعد از محو شدن ، از بین
بروند.
(برای حذف کردن اشیاء ، می توانید از تابع Ent_remove() استفاده کنید به این ترتیب که داخل تابع ،
نام اشاره گر شیء مورد نظر را قرار دهید.)

فصل ۵ : حرکت اشیاء در هر زاویه ای

تا اینجا یاد گرفتیم که اشیاء را در محورهای x, y حرکت دهیم. در این فصل خواهیم آموخت که چگونه اشیاء را در هر جهتی حرکت دهیم.

برای حرکت در بین محورهای x, y باید به Pan شیء توجه کنیم. شیء ما، هر زاویه ای که داشت، از روی آن به جلو، عقب و چپ و راست حرکت می کند. مثل بازیکن، در یک بازی که کراکتر با استفاده از جهتهای چپ و راست، جهت خود را مشخص کرده و سپس توسط جهتهای جلو و عقب، به سمت جلو یا عقب حرکت می کنید.

ما با استفاده از این دستور، می توانیم علاوه بر حرکت کراکتر، از عبور اشیاء از سطوح نیز جلوگیری کنیم.

برای حرکت دادن اشیاء میتوانیم از دستورهای `c_move` و `ent_move` استفاده کنیم که ما در این فصل طرز استفاده از دستور `ent_move` را شرح خواهیم داد.

ابتدا سافتکار دستور `ent_move` را بررسی می کنیم :

`Ent_move` (فاصله قطعی , فاصله نسبی) ;

شما می توانید به هر صورتی که مایلید، از این دستور استفاده کنید ولی چون ما می خواهیم که کراکتر را از روی Pan، حرکت دهیم باید آن را با فاصله های کوتاه حرکت دهیم. برای این کار باید فاصله نسبی را مموری در نظر بگیریم که محور x آن، یک مقدار ثابت دارد و فاصله قطعی را بردار صفر در نظر میگیریم تا به فاصله نسبی از بردار صفر حرکت کند. در مثال بهتر متوجه می شوید :

`Ent_move` (`vector (2,0,0)` , `nullvector`) ;

در این مثال کراکتر ما در هر بار اجرای این دستور، از بردار صفر، به اندازه ۲ بر روی محور x ها حرکت می کند ولی این بار محور x ها، همان محور x هایی نیست که در قبل دیدید. این بار کراکتر در هر زاویه ای باشد، چون ما از بردار صفر، کراکتر را حرکت می دهیم، همیشه جهت روبرو، محور x است.

`Nullvector` یک کلمه کلیدی به معنی بردار صفر است.

در مثال زیر ، جهت کراکتر با کلیدهای جهت دار چپ و راست تعیین شده و سپس با کلید جلو ، کراکتر به سمت جلو حرکت می کند.

```
Action Player_1 {
    Whilew ( 1 ){
        If ( key_cur == on ){
            My.pan -=1;
        }
        If ( key_cul == on ){
            My.pan +=1;
        }
        If ( key_cuu == on ){
            Ent_move ( vector ( 2,0,0 ) , nullvector );
        }
        Wait ( 1 );
    }
}
```

برای پرفش کراکتر می توانید از دستور روبرو نیز استفاده کنید که مقدار پرفش را برای هر دو جهت مشخص می کند :

```
My.pan += 4 * time * ( key_cul - key_cur ) ;
```

در اینجا ، عدد ۴ سرعت پرفش است که شما می توانید آن را به دلفواه تغییر دهید.

این دستور یک فرمول ریاضی فیزی ساده است.

وقتی که کلیدها فشرد می شوند ، مقدار ۱ یا On را بر می گردانند و اگر مقدار آنها ۱ باشد ، حاصلضرب آنها مخالف صفر شده و مقداری را به Pan ، اضافه می کنند.

تمرین ها فصل ۵ :

۱) بازی طراحی کنید که کراکتر بازی ، بتواند به هر جهتی حرکت کند.

۲) بازی طراحی کنید که کراکتر بازی یک توپ جنگی است که می تواند به هر جهتی حرکت

کرده و با فشار کلید Space ، تیر شلیک کند.

فصل ۴ : دوربین

در این فصل می‌فواهیم درباره دوربین و تنظیمات آن صحبت کنیم.

دوربین در بازی‌های کامپیوتری، نقش اساسی را ایفا می‌کند. حرکت، پرفش، زاویه و

دوربین می‌تواند حالتها و فرم‌ها و جایگاه‌های خاصی در هر بازی داشته باشد و آن بسته به نوع بازی و نظر سازنده بازی است.

ما در این مقاله، فرم‌های قرارگیری دوربین را شرح نمی‌دهیم بلکه فقط به توضیح تنظیمات و پیچونگی حرکت آن می‌پردازیم.

دوربین نیز مانند اشاره گر My که به کراکتر صامب اکشن، اشاره می‌کند، اشاره‌گری با نام Camera دارد.

Camera، دوربین اصلی بازی ماست. علاوه بر این دوربین، ما می‌توانیم دوربین‌های دیگری نیز در بازی داشته باشیم و یا تصویر را به طور همزمان از چند دوربین ببینیم.

بعضی از فصوصیات دوربین به شرح زیر است :

Camera . Layer	لایه نمایش تصویر
Camera . Pos_x	محل قرارگیری دوربین در محور ایکس
Camera . Pos_y	محل قرارگیری دوربین در محور ایگرگ
Camera . Pos_z	محل قرارگیری دوربین در محور زد
Camera . Arc	ممدوده تصویر برداری دوربین
Camera . Size_x	پهنای صفحه ای که تصویر دوربین در آن نمایش داده می‌شود
Camera . Size_y	بلندی صفحه ای که تصویر دوربین در آن نمایش داده می‌شود
Camera . Visible	نمایش داده شدن تصویر دوربین
...	

فصوصیات دیگر دوربین را می‌توانید در Manual نرم افزار، پیدا کنید.

استفاده از دوربین خیلی ساده است. شما می‌توانید با استفاده از دستورهای ریاضی، دوربین را وادار به انجام کارهای پیچیده ای بکنید.

برای تعریف دوربین های دیگر در محیط برنامه باید به شکل زیر عمل کنید :

```
{ نام دوربین View  
    ; خصوصیات دلفواه  
}
```

دقت کنید که در تعریف خصوصیات دوربین هایی که خودتان آنها را تعریف می کنید ، امتیابی به نوشتن نام دوربین نیست بلکه فقط نوشتن نام خصوصیت کافیست.
شما می توانید دوربین را در تمامی ممورها به حرکت درآورید .

ما در این فصل نمونه استفاده از دوربین را به عهده شما می گذاریم .

تمرین های فصل ۶ :

۱) دوربینی طراحی کنید که توانایی چرخش به دور کراکتر را داشته باشد.

(مانند سایر بازی های کامپیوتری)

۲) بازی طراحی کنید که فشار دادن کلید Space تصویر بین ۴ دوربین در ۴ گوشه اتاق ، جابجا شود.

فصل ۷ : رویدادها

رویدادها ، یکی از مهمترین و اساسی ترین نکات در برنامه نویسی بازی های کامپیوتری هستند . شما بوسیله رویدادها ، امکان تشفیص هر چیزی را دارید .
مثلا می توانید تشفیص دهید که شیئی به کراکتر نزدیک می شود و یا صدایی شنیده می شود و یا چیزی در محیط ، حرکت می کند و

برای استفاده از رویدادها ، شما باید امکان تشفیص آن رویداد را در کراکتر خود ، فعال کنید .
مثلا رویداد Impact یا برفورد ، تشفیص می دهد که شیئی به کراکتر برفورد کرده است یا خیر .

بعد از تشفیص رویداد ، شما باید برای انجام عملیات در رابطه با آن رویداد ، خصوصیت رویداد کراکتر را به تابعی انتقال دهید که بتواند بر اساس آن رویداد تصمیم گیری کند .
مثلا در مثال زیر ، وقتی کراکتر با شیئی برفورد کند ، بازی تمام می شود .

```
Function set_event () {  
    If ( event_type == event_impact ) {  
        Exit ;  
    }  
}  
  
Action Player_1 {  
    My . enable_impact = on ;  
    My . event = set_event ;  
}
```

در این دستور ساده ، احتمای به استفاده از event_type و event_impact نیست زیرا این شرط تشفیص می دهد که آیا رویداد رخ داده ، رویداد برفورد است یا رویداد دیگری است .
این شروط برای کراکتر هایی که از چندین رویداد استفاده می کنند ، ضروری است .
شما سایر رویدادها را می توانید در Manual نرم افزار ، ملاحظه کنید .

وقتی که رویداری رخ می دهد ، عامل برفورد کننده ، در اشاره گر you قرار می گیرد .
در کراکتر اصلی ، برای شناساندن بازیکن ، می توانید از اشاره گر Player بصورت زیر استفاده کنید :

```
Player = My ;
```

شما می توانید در شروط رویداد ها ، بررسی کنید که مثلاً عامل برفورد کننده ، بازیکن اصلی است :

If (event_type == event_impact && you == player)

تمرین های فصل ۷ :

۱) بازی طراحی کنید که در محیط بازی ۱۰ سکه وجود داشته باشد و کراکتر با برفورد به هر

سکه ، آن سکه از بین برود و یکی به امتیازاتش اضافه شود. بعد از جمع آوری هر ۱۰

سکه ، بازی تمام شود.

(این بازی را به میل خود ، جالب تر کرده و مانع و فطر نیز به بازی خود اضافه کنید.)

فصل ۸ : صفحه ها و متون

در تمام بازی ها ، شما منو ها و عکس ها و متونی را در اطراف صفحه نمایش می بینید. بعضی از آنها ، امتیاز شما ، بعضی وقت شما و ... را نمایش می دهند.

در اول هر بازی هم شما منویی را مشاهده می کنید که توسط آن شما می توانید انتخاب خود را انجام دهید.

شما تمام این کارها را می توانید با استفاده از صفحه ها یا در اصطلاح Panel ها و متون یا Text ها ، انجام دهید.

Panel :

پنل ها ، صفحه هایی هستند که شما می توانید در آنها عکس و اعداد را قرار دهید و از آنها به عنوان قسمت پس زمینه یا Background اعداد و علائم خود در بازی استفاده کنید.

نمونه ایجاد یک پنل ، بصورت زیر است :

```
{ نام پنل Panel
    Layer = شماره لایه ;
    Pos_x = محور x ;
    Pos_y = محور y ;
    Window = x , y , پهنا , بلندی , عکس زمینه , عکس افقی عکس , مکان عمودی عکس ;
    Digits = x , y , طول عدد , قلم , مقدار ضریب ;
    Button = x , y , تابع خاموش , تابع روشن , عکس بالا , عکس خاموش , عکس روشن ;
    Flags = حالت های پنل ;
    ...
}
```

نوشتن لایه و محور x و محور y ، الزامی است.

دقت کنید که عکس زمینه متما باید بصورت Bitmap تعریف شود :

> مسیر عکس < = نام عکس Bmap

نوع عکس شما ، می تواند یکی از انواع Pcx ، Bmp ، Tga و ... باشد.

نوع رنگ بندی عکس نیز طبقه بندی شده است که بستگی به استفاده شما از عکس دارد.

برای نمایش دادن پنل ، می توانید از دستور زیر استفاده کنید :

visible = on . نام پنل

: Text

نمونه ایجاد متن :

```
{ نام متن Text
    Layer = شماره لایه ;
    Pos_x = x محور ;
    Pos_y = y محور ;
    Font = نام قلم ;
    Strings = تعداد متون ;
    String = ۱ متن , ۲ متن , ۳ متن , ... ;
    Alpha = میزان وضوح ;
    Flags = حالت های متن ;
    ...
}
```

در مورد گزینه های دیگر پنل و متن ، می توانید از Manual نرم افزار استفاده کنید.

در قسمت نام قلم ، باید نام قلمی که از قبل تعریف کرده اید را بنویسید :

اندازه قلم , حالت قلم , " نام قلم موجود در سیستم " = نام قلم Font

حالات قلم :

۰ - معمولی

۱ - کلفت

۲ - کج

۳ - کلفت و کج

برای نمایش دادن متن نیز می توانید به روش زیر عمل کنید :

visible = on . نام متن

تمرین های فصل ۸ :

۱) منو های زیر را به بازی طراحی شده در فصل قبل ، اضافه کنید :

۱- میزان امتیاز

۲- میزان زندگی

۳- شماره مرحله

۴- نام فودتان در پایان منو

فصل ۹ : پایان

شما با مطالعه کامل این جزوه و تمرین ، می توانید به راحتی ، بازی های جالب و زیبایی طراحی کنید.
ما در این جزوه سعی کردیم نکات اصلی در برنامه نویسی نرم افزار 3D Game Studio را به شما آموزش دهیم .

باقی نکات را خودتان باید با جستجو و مطالعه و تمرین ، بدست آورید.
شما می توانید با مطالعه این جزوه و استفاده از تالارهای سایت www.Persian-Designers.Com ،
در مدت زمان تقریباً کوتاهی ، بسیاری از نکات برنامه نویسی بازی های کامپیوتری را یاد بگیرید.

بعد از آن ، خود شما هستید که باید به دنبال ترفند ها و استفاده از ابزارهای گوناگون بروید.

امید داریم که این جزوه برای دوستان مبتدی و کسانی که تازه برنامه نویسی با این نرم افزار را شروع کرده اند ، مفید بوده باشد.

با تشکر و آرزوی موفقیت و پیروزی برای تمام شما عزیزان.

تهیه و تنظیم : شهرزاد علیممّدی
هر گونه تکثیر از این مقاله با ذکر منبع ، بلامانع است
www.Persian-Designers.Com