پیاده سازی سیستم ذخیره و بارگذاری

در سه سطح ساده ، متوسط و پیشرفته

در موتور Unity3D

نویسنده: محمدرضا آرشیان

www.Silverman.ir

www.Persian-Designers.com

سلام و عرض ادب خدمت شما دوستان خوبم.

در این مقاله و آموزش می خواهیم به بررسی یک سیستم لود و سیو در سه مقطع "ساده" و "متوسط" و "پیشرفته" در موتور یونیتی بپردازیم.

برای خواندن این مقاله شما باید پیش زمینه های ذهنی ای از مقدمات برنامه نویسی یونیتی، کار با رشته ها و کار با روابط گرافیکی (GUI) داشته باشید.

بسیار خوب بهتر شروع کنیم!

باید خدمتتون عرض کنم که اساس کار تمام فایل های ذخیره و بارگذاری در یک چیز هست: ایجاد یک فایل در منطقه ای از سیستم بازیکن و نوشتن اطلاعات دلخواه در اونه. بعوان مثلا شما در اکثر بازی ها مشاهده می کنید که فایلی در پوشه ی my documents شما ایجاد میشه و محتویاتی در اون هست که معمولا ما فرمت های عادی برنامه های مختلف فرق داره. فرمت هایی مثل sav یا ss و ...

این ها همان فایل هایی هستند که در آنها مشخصات دلخواه ذخیره شده اند. در گام اول شما باید بدانید که چه اطلاعاتی رو می خواهید ذخیره بکنید. مثلا مختصات بازیکن، شماره ی مرحله، مقدار موجود در داخل یک متغیر و ... ولی باید این نکته رو مد نظر داشته باشید که تنها مقادیری بعد از خروج از بازی قابل بازیابی خواهند بود که در فایلی که ذخیره می کنید، نوشته شده باشد. مثلا شما مختصات بازیکن را ذخیره کرده و موقع شروع مجدد لود می کنید. در این هنگام مثلا دری که قبلا بازیکن آن را باز کرده به حالت پیشفرض خود یعنی بسته رد می اید و برای اینکه در همچنان باز باشد باید متغیری برای در را در فایل سیو ذخیره کنید.

حالا با آمادگی ذهنی ای که پیدا کردید شروع به ساخت یک فایل سیو و بارگذاری مجدد آن می پردازیم.

- **ساخت یک فایل ذخیره و بارگذاری آن (سطح ساده):**

در این سطح ما فقط فایلی به نام save.sv را در محل اجرای بازی و با محتویات شماره ی مرحله ذخیره می کنیم.

خوب در ابتدا چون ما می خواهیم از امکانی به نام System.IO در ذخیره کردن استفاده کنیم پس باید آن را وارد بازی بکنیم.

پس در ابتدای کد های خود، پیش از هرگونه متغیر و یا کدی این را می نویسیم

برای جاوا:

```
import System.IO;
```

برای #C:

```
using System.IO;
```

حالا میخواهیم فایلی به نام save.sv را در محل اجرای بازی ایجاد کنیم. برای این کار از فرمانی به نام StreamWriter استفاده می کنیم ک هزیر مجموعه ی System.ioهست. این دستور فایلی را به نام دلخواه و در مکانی دلخواه برای ما ذخیره می کند. بعنوان مثال کد زیر را در نظر بگیرید:

```
StreamWriter("c:/save.sv");
```

این خط فایلی به نام save.sv در شاخه ی اصلی درایو c: بازیکن ایجاد میکند. این فرمان قابلیت اسفاده از آدرس های مجازی محتلف را دارد. همینطور شما می توانید هر پسوند و یا هر اسمی که خواستید برای این فایل ذخیره انتخاب کنید. ولی حواستان به نام انتخابی باشد که با آن کار های زیادی داریم!

ما از این فرمان استفاده می کنیم و فایلی را در محل اجرای بازی ایجاد و برای کاهش خطوط برنامه حاصل را در متغیری به نام sr می ریزیم.:

```
var sw = new StreamWriter(Application.dataPath +"/save.sv");
```

حالا می خواهیم مشخص بکنیم که چه چیز هایی در این فایل ذخیره شوند. برای ذخیره سازی متن در داخل این فایل ایجاد شده از فرمان WriteLine استفاده میکنیم. هر نوع متغیر رشته ای، عددی و ... در این فایل قابلیت ذخیره سازی دارند. ولی حواستان باشد که ایت مختویات بصورت String ذخیره می شوند و باید برای استفاده به جای مختصات آنان را تبدیل کرد. ما نام مرحله را در این فایل ذخیره می کنیم:

```
sw.WriteLine(Application.loadedLevelName );
```

و می رسیم به مهمترین خط.. حواستان باشد که در پایان کار حتما باید فایل را ببندید گرنه فایلتان خراب میشود و قابلیت خواندن ندارد:

```
sw.Close();
```

ما یک دکمه هم تعریف می کنیم تا بتوانیم با آن سیو کنیم. شما بعدا قشنگ ترش کنید!

اکنون باید کد شما به زبان جاوا بصورت زیر باشد:

```
import System.IO;

      function save(){
            var    swRBell    =    new    StreamWriter(Application.dataPath
+"/silverdata.sd");
      swRBell.WriteLine(Application.loadedLevelName );
      swRBell.Close();
}

/////kod marbut be GUI va hich tasiri dar save nadarad

function OnGUI(){
      if(GUI.Button(Rect(0,0,300,200),"Save")) save();
}
```

- **ساخت سیستم بارگذاری:**

خوب بهتره یک *function* **دیگه تعریف کنید مثلا به نام** *load*

حواندن یک فایل با استفاده از دستور StreamReader **انجام خواهد شد. دقیقا عملیات مانند** StreamWriter **خواهد بود و** StreamReader **همان خصوصیات آدرس دهی و .. را به شما خواهد داد. ما نام فایل را داده و آن را در متغیری می خوانیم:**

```
var sr = new StreamReader(Application.dataPath + "\\save.sv");
```

سپس با فرمان ReadLine **تک تک خطوط ذخیره شده خوانده میشود. خوب من معمولا عادت دارم خطوط رو داخل یک متغیر دخیره کنم تا کارم ساده تر بشه ... پس:**

```
Var line : String = sr.ReadLine();
```

و مانند قبل یادمان باشد که حتما فایل را ببندیم. در غیر اینصورت برنامه با ارور مواجه شده و مجتویات فایل خوانده نمی شود:

```
sr.Close();
```

هدف ما بارگزاری و لود مرحله ی جدید است. لذا خواهیم داشت:

```
Application.LoadLevel(line);
```

لارم به ذکره که فرمان *LoadLevel* **نیاز به متغیری** *string* **دارد و ما از همین رو نیازی به تبدیل خطوط به** *float* **یا** *int* **نداشتیم.**

کد قسمت لود شما باید بصورت زیر باشد:

```
function load () {
      sr = new StreamReader(Application.dataPath + "\\silverdata.sd");
      line = sr.ReadLine();
      sr.Close();
      Application.LoadLevel(line);
}
```

دقت کنید که من این کد را زیر کد save() قرار دادم. به همین خاطر دیگر import System.IO; رو ننوشتم ولی اگه می خواهید این رو در یک کد جاوا جدید قرار بدید باید این خط رو ذکر کنید.

کد نهایی شما که هم saveرا داشته باشد و هم load را اینست:

```
      function save(){
            var    swRBell    =    new    StreamWriter(Application.dataPath
+"/silverdata.sd");
      swRBell.WriteLine(Application.loadedLevelName );
      swRBell.Close();
}



function load () {
      sr = new StreamReader(Application.dataPath + "\\silverdata.sd");
      line = sr.ReadLine();
      sr.Close();
      Application.LoadLevel(line);
}


/////kod marbut be GUI va hich tasiri dar save nadarad

function OnGUI(){
      if(GUI.Button(Rect(0,0,300,200),"Save")) save();
      if(GUI.Button(Rect(0,300,300,200),"Load")) load();

}
```

بسیار خوب این هم یکی سیستم لود ساده. در ادامه به مانور بیشتر روی این کد ها خواهیم پرداخت

-سیستم ذخیره (سطح متوسط):

خوب ما می خواهیم دوتا قسمت رو ذخیره کنیم. مثلا هم نام باریکن و هم مرحله ای که در اون هست. خوب در این صورت کد ذخیره و کد لود ما در دوتا خط تفاوت می کنن:

کد ذخیره :

```
sr.WriteLine(player_name +"#"+Application.LoadedLevelName);
```

با فرض اینکه نام پلیر شما در تغیری به نام player_nameذخیره شده باشد. بله اشتباه نکنید. اون # حتما برای این قسمت لازمه!

کد لود:

```
var player_name : String = line.Substring(0,line.IndexOf("#"));
var loaded_level : String =
line.Substring(line.IndexOf("#")+1,line.length-player_name.length-1);
print(player_name);
print(loaded_level);
```

ما با فرمان Substring می تونیم متن رو قسمت قسمت بکنیم. الان در کد بالا در خط اول ما از کاراکتر ۰ به طول شماره ی کاراکتر # جدا کردیم و در خط دوم ما از شماره ی کاراکتر # به طول مقدار کل رشته منهی طول رشته ی اسم جدا کردیم. در آخر هم حاصل ها رو نمایش دادیم..IndexOf کارش پیدا کردن کاراکتر مورد نظر و برگردوندن شماره ی اون در جملست.

خیلی خوب .. اینم نمایی از کل این دوتا کد به همراه آخرین تغییرات روش:

```
function OnGUI(){
      if(GUI.Button(Rect(0,0,300,200),"Save")) save2();
      if(GUI.Button(Rect(0,300,300,200),"Load")) load2();

}
function save2(){
var player_name="Silverman!!";
      var sw = new StreamWriter(Application.dataPath +"/silverdata.sd");
      sw.WriteLine(player_name +"#"+Application.loadedLevelName);
      sw.Close();
}

function load2 () {
      sr = new StreamReader(Application.dataPath + "\\silverdata.sd");
      line = sr.ReadLine();
      sr.Close();
      var player_name : String = line.Substring(0,line.IndexOf("#"));
var loaded_level : String =
line.Substring(line.IndexOf("#")+1,line.length-player_name.length-1);
      print(player_name);
      print(loaded_level);}
```

-سیستم ذخیره(سطح پیشرفته):

در این سطح می خواهیم ذخیره و بارگذاری رو بر اساس مختصاتشون انجام بدیم. کلیت کد مثل بالا هست:

```
sw.WriteLine (transform.position.x.ToString("#.00")+"#" +
transform.position.x.ToString("#.00") +"$"+
transform.position.x.ToString("#.00"));
```

خب یک توضیح راجع به بالا. ببینید به طور کلی مختصات ها بر اساس Vector3 هستند و سه خاصیت x , y , z رو باهم دارند. ولی چون توی این نوع سیستم مختصات دهی، پرانتز وجود داره نمی شه مختصات رو مستقیما در داخل فایل سیو گذاشت. مثلا شما نمی تونید این مختصات رو ذخیره کنید : (0,10,50) چون مشکلی موقع لود پیش میاد که همونجا بهتون میگم.

خوب پس ما اومدیم مختصات های x , y , z رو جدا جدا ذخیره کردیم.

می دونید که مختصات ها از نوع float هستند. یعنی تا چندین رقم اعشار رو قبول می کنن. ما می تونیم که مختصات ها رو به همون صورت ذخیره کنیم ولی واسه خوندن از توش واسمون دردسر میشه. پس با این قسمت از کد ToString("#.00"). کاری می کنیم که در هر صورت عددمون فقط دورقم اعشار داشته باشه. بعدشم دوباره مثه بالا شروع به قسمت بندی می کنیم:

```
var pos_x : float = float.Parse(line.Substring(0,line.IndexOf("#")));
var pos_y : float =
float.Parse(line.Substring(line.IndexOf("#")+1,line.IndexOf("s")-
line.IndexOf("#")-1));
var pos_z : float =
float.Parse(line.Substring(line.IndexOf("s")+1,line.Length-
line.IndexOf("s")-1));
transform.position=Vector3(pos_x,pos_y,pos_z);
```

همونطوری که گفتم مقدار ذخیره شده در فایلتون بصورت String درمیاد.. شما باید برای استفاده در سیستم مختصات این ها رو تبدیل کنید به مقادیر عددی int یا float. دستور float.Parse این کار رو برای ما به راحتی انجام میده. ولی این دستور هرگونه چیز غیر عددی مثل علائمو یا حروف رو تبدیل به عدد نمیکنه. به خاطر همینه که گفتم نمی تونید توش پرانتز به کار ببرید. درواقع دستور float.Parse در مقابل دستور ToString قرار داره.

کد کامل شما اینچنین خواهد شد:

```
/////kod marbut be GUI va hich tasiri dar save nadarad
```

```
function OnGUI(){
      if(GUI.Button(Rect(0,0,300,200),"Save")) save۳();
      if(GUI.Button(Rect(0,300,300,200),"Load")) load۳();

}

function save۳(){
var player_name="Silverman!!";
      var sw = new StreamWriter(Application.dataPath +"/silverdata.sd");
      sw.WriteLine       (transform.position.x.ToString("#.00")+"#"       +
transform.position.y.ToString("#.00")                                    +"s"+
transform.position.z.ToString("#.00"));
      sw.Close();
}


function load۳ () {
      sr = new StreamReader(Application.dataPath + "\\silverdata.sd");
      line = sr.ReadLine();
      sr.Close();
      var              pos_x             :             float             =
float.Parse(line.Substring(0,line.IndexOf("#")));
      var              pos_y             :             float             =
float.Parse(line.Substring(line.IndexOf("#")+1,line.IndexOf("s")-
line.IndexOf("#")-1));
      var              pos_z             :             float             =
float.Parse(line.Substring(line.IndexOf("s")+1,line.Length-
line.IndexOf("s")-1));
      transform.position=Vector3(pos_x,pos_y,pos_z);
print(pos_x +"  "+pos_y+"  "+pos_z);
print(line);
}
```

سیستمی که خدمتتون معرفی شد الزامی نداره که در قالب سیستم ذخیره و بارگذاری استفاده بشه. بلکه میتونه راهکاری باشه برای بسیاری از مشکلات و وسیله ای باشه برای برقراری ارتباط قسمت های مختلف بازی به هم.

آموزش من در همینجا به پایان میرسه. امیدوارم که از این به بعد شاهد پشیرفت چشمگیر و روزانه ی صنعت ساخت بازی در ایران باشیم.

لطفا هرگونه مشکلی رو یا در انجمن مطرح کنید و یا مستقیما با خود من درمیون بذارید.

به امید آینده ای روشن...

محمدرضا آرشیان – بهار ۹۰

www.Silverman.ir

www.persian-designers.com


ترسم کم بیایی آذر و تباشم
یا صاحب الزمان
عجل الله تعالی فرجه الشریف