

آموزش

اسکریپت اکشن از سطح صفر

Learning ActionScript 3.0



گردآوری:

سید محمد جواد نیکوکار



درس سی و یکم:

یکی از اساسی ترین مهارتهایی که تو AS3 باید یاد گرفته بشه اینه که چطور بتونیم فیلم (Movie) فلشمون رو کنترل کنیم.

درسهای اول رو به یاد بیارین. اگه یادتون باشه گفتیم می تونیم که به وسیله کدنویسی یه فریم های یه فیلم رو کنترل کنیم و مثلا حرکت

آهسته برای فیلم ایجاد کنیم و ...

خوب اینا مستلزم اینه که به وسیله AS3 بتونیم تایم لاین فلش و فریم های اون رو کنترل کنیم. حالا تایم لاین می تونه تایم لاین اصلی فلشمون باشه. یا تایم لاین یکی از مووی کلیپها. **توجه داشته باشین که هر مووی کلیپ از تعدادی (یا حداقل یک) فریم تشکیل شده.**

بذارین از یه مثال ساده شروع کنیم. می خوایم یه برنامه بنویسیم که با کلیدهای کنترلی تایم لاین اصلی رو کنترل کنه. مثلا یه فیلم ساده بسازیم و بتونیم با فشردن کلید stop فیلم رو متوقف کنیم و با فشردن دکمه play دوباره فیلم شروع به پخش شدن کنه.(نکته: منظور ما اینجا از فیلم اون فیلم که تو تلویزیون میبینید نیست ها!!! یعنی فیلم انیمشن)

مثال زیر رو ببینید تا همه چیز دستتون بیاد :



همون طور که تو عکس میبینید ما یه فیلم داریم که در حال پخشه. و یه سری کلید که می تونیم باهاشون فیلم رو کنترل کنیم.

البته دو تا کلید دیگه هم اضافه کردم تحت عنوان gotoAndPlay و gotoAndStop اینا همون طور که از نامشون پیداست فیلم رو به یه فریم خاص می پرنونن! و به ترتین فیلم رو از اون خاص فریم شروع به پخش (gotoAndPlay) و در اون فریم متوقف می کنن (gotoAndStop).

شاید از خودتون بپرسین که : خوی چه دلیلی داره ما از بپریم به یه فریم و از اونجا فیلم رو پخش کنیم یا متوقفش کنیم. در جواب می شه گفت که : فکر کنین که یه فیلم آموزشی برای درس فیزیک ساختین که ۳ فصل اول کتاب فیزیک رو شامل میشه.

آیا یه دانشجویی که دو فصل اول رو بلده و فقط می خواد فصل سه رو یاد بگیره مجبوره تمام فیلم رو ببینه و وقت گهربارش رو هدر بده تا برسه به فصل ۳؟ نه خداییش این انصافه؟

کد زیر رو به فریم اول اضافه می کنیم و برنامه رو اجرا می کنیم. 😊

```
stopBtn.addEventListener(MouseEvent.CLICK, onStopClick, false, 0, true);
playBtn.addEventListener(MouseEvent.CLICK, onPlayClick, false, 0, true);
gotoPlayBtn.addEventListener(MouseEvent.CLICK, onGotoPlayClick, false, 0, true);
gotoStopBtn.addEventListener(MouseEvent.CLICK, onGotoStopClick, false, 0, true);

function onStopClick(evt:MouseEvent):void {
    stop();
}
function onPlayClick(evt:MouseEvent):void {
    play();
}
function onGotoPlayClick(evt:MouseEvent):void {
    gotoAndPlay(1);
}
function onGotoStopClick(evt:MouseEvent):void {
    gotoAndStop(1);
}
```

در مورد کد فکر نکنم توضیحی لازم باشد. اگر `stopBtn` کلیک بشه که تابع `onStopClick` اجرا میشه و فیلم متوقف میشه و به این ترتیب برای کلیدهای دیگه هم همین اتفاق البته با رفتار متفاوت از سوی اونا (کلیدها) می افته.

نکته: اینجا ما با `stop()` و `Play()` و ... تایم لاین اصلی رو متوقف یا به فریم خاص می بریم. بعدا یاد خواهیم گرفت که چجوری یه مووی کلیپ رو به یه فریم خاص ببریم ، `play` کنیم و یا `stop` و

بذارین یه کار خوشکل کوچولوی دیگه هم روی برنامه انجام بدیم. کد زیر رو به انتهای برنامه اضافه کنین.

```
trace("This movie has " + totalFrames + " frames.");
trace(currentFrame);
```

خط اول می گه که تعداد فریم های فیلم رو تو خروجی نشون بده (خروجی تست!!!) . و خط دوم می گه که فریم جاری رو نشون بده. البته اگر باهوش باشین می فهین که خط دوم رو باید در هر تابع برنامه کپی کنین تا معنیدار باشه.

اگه انتهای برنامه کپی کنین چی میشه؟ اگه تو تمام تابع ها بذارینش چی؟

Frame Labels (نشانه گذاری فریم ها):

لیبل گذاری یا همون نشونه گذاری فریم ها شاید در نظر اول کار بیهوده ای به نظر بیاد. چرا باید روی یه فریم اسم یا نشانه بذاریم؟ اصلا این کار چه فایده ای داره؟

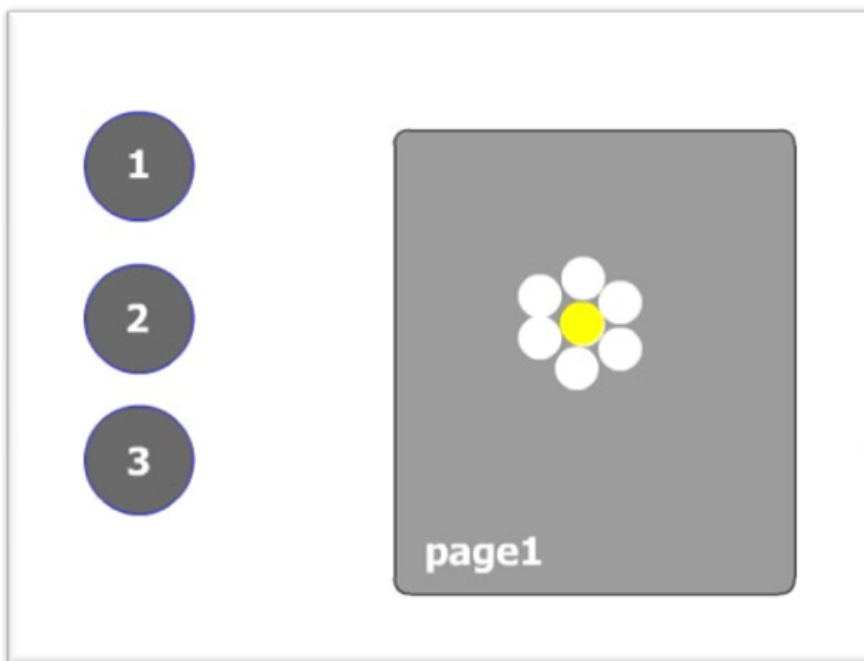
مثال قبلی در مورد لزوم استفاده از `gotoAndPlay` رو که یادتونه؟ خوب حالا شما فکر کنین تو فریم اول گفتین اگه روی کلید "فصل سوم" کلیک شد برو به فریم ۱۳۰۰ (شماره فریمی که فصل سوم از اون شروع میشه). تا اینجا هیچ مشکلی وجود نداره.

شما فیلم رو به استادتون نشون می دین تا در صورت تایید به دانشجویها تحویل بدین. استادتون بعد از دیدن فیلم گیر میده که : ۲۵ فریم فیلم اضافه و باید پاک بشه. و شما هم میگین اشکالی نداره! و سریعا موس رو تو دستتون می گیرین و اون ۲۵ فریم رو انتخاب می کنین و اونا رو حذف می کنین. چه اتفاقی می افته؟

خیلی سادست: دیگه اول فصل سوم فریم شماره ۱۳۰۰ نیست!!! پس کاربر با کلیک روی دکمه "فصل سوم" به اول فصل سوم نخواهد رفت. حالا این که مشکلی نیست نهایتا به تغییر یه عدد تو فریم اول حل میشه. فکرش رو بکنین که شما تو یه پروژه بزرگ ۲۳۴۵۶۳۷۴۶۷۵۶ تا آدرس دادین اون وقت که واویلا!!!!!! 🤪

راه حل چیه؟ خوب مثل یه بچه خوب، از همون اول فریمی رو که می خوایم با کلیک روش بریم نام گذاری میکنیم و به جای عدد فریم در کد نام فریم رو می نویسیم. حالا هر تغییری در تعداد فریم ها ایجاد بشه دیگه هیچ اتفاقی نمی افته

مثال زیر رو در پیوست قرار دادم اون رو بگیرین و بازش کنین و کارایی اون رو ببینید.



اینم کدش :

```
pages.stop();

one.addEventListener(MouseEvent.CLICK, onOneClick, false, 0, true);
two.addEventListener(MouseEvent.CLICK, onTwoClick, false, 0, true);
three.addEventListener(MouseEvent.CLICK, onThreeClick, false, 0,
true);

function onOneClick(evt:MouseEvent):void {
    pages.gotoAndStop("page1");
}
function onTwoClick(evt:MouseEvent):void {
    pages.gotoAndStop("page2");
}
function onThreeClick(evt:MouseEvent):void {
    pages.gotoAndStop("page3");
}
```

می بینید که توی کد برای رفتن به یه فریم خاص از نام اون فریم استفاده شده. می بینید که برای فرستاده مووی کلیپ به فریم خاص فقط اسم مووی کلیپ رو اول دستور `play` یا `stop` یا `gotoAndPlay` و... اضافه می کنیم.

نکته: فکر نکنم لازم باشه که بگم برای اضافه کردن `label` باید فریم رو انتخاب و از پنجره `properties` فیلد `FrameLabel` رو مقداردهی کنید!

برای درس بعد یه ویژگی جدید در فریم لیبل رو که با `AS3` معرفی شده رو بررسی خواهیم کرد.

درس سی و دوم:



کار با لیبل فریم ها - پیشرفته:

همونطور که جلسه قبل قول داده بودم قرار شد ویژگی جدید و جالبی رو که با AS3 برای کار با فریم لیبل معرفی شد رو توضیح بدم.

همونطور که می دونید scene ها توی فلش ابزاری هستند برای مدیریت کردن تایم لاین . با استفاده از scene ها می تونین تایم لاین رو به تیکه ها کوچیک تر تقسیم کنین. فکرش بکنین شما یه تایم لاین برا یه انیمیشن نسبتا طولانی دارین. خوب چه بهتر که بتونین فیلمتون رو به تیکه های کوچیک تر تقسیم کنین تا کار روی اونا و تستوشون راحت تر بشه. در حقیقت شما تایم لاین رو دسته دسته می بینین ولی فلش scene ها رو پشت سر هم و به صورت یه تایم لاین واحد می بینه.

نکته: اگه با scene ها کار نکردین و توضیحات بالا هم بهتون نچسبید می تونین از help خود فلش استفاده کنین

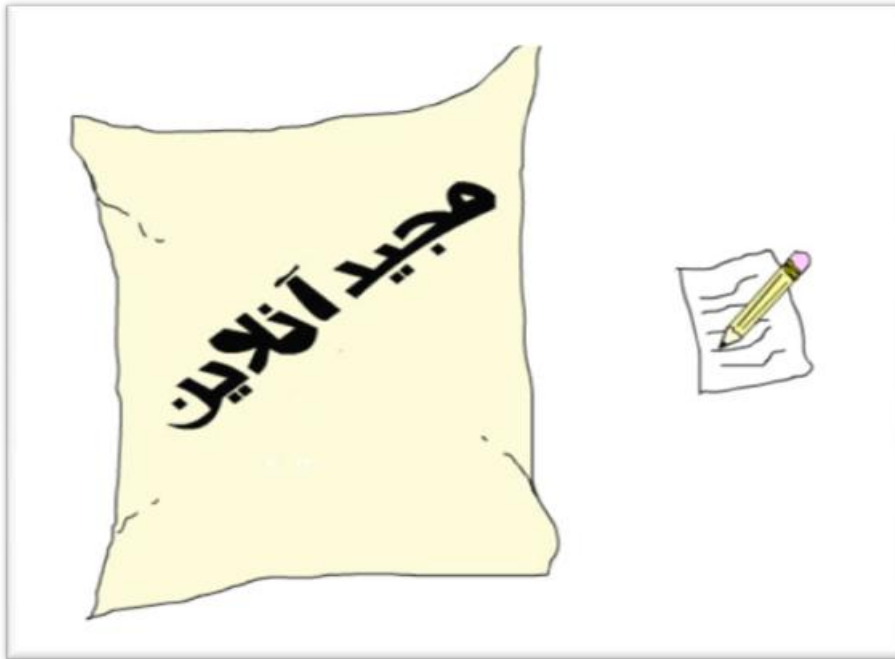
حالا AS3 چیکار کرده؟

اومده یه آرایه در اختیار کاربر قرار داده که لیبل فریم ها و خصوصیات اونا توش قرار داره.

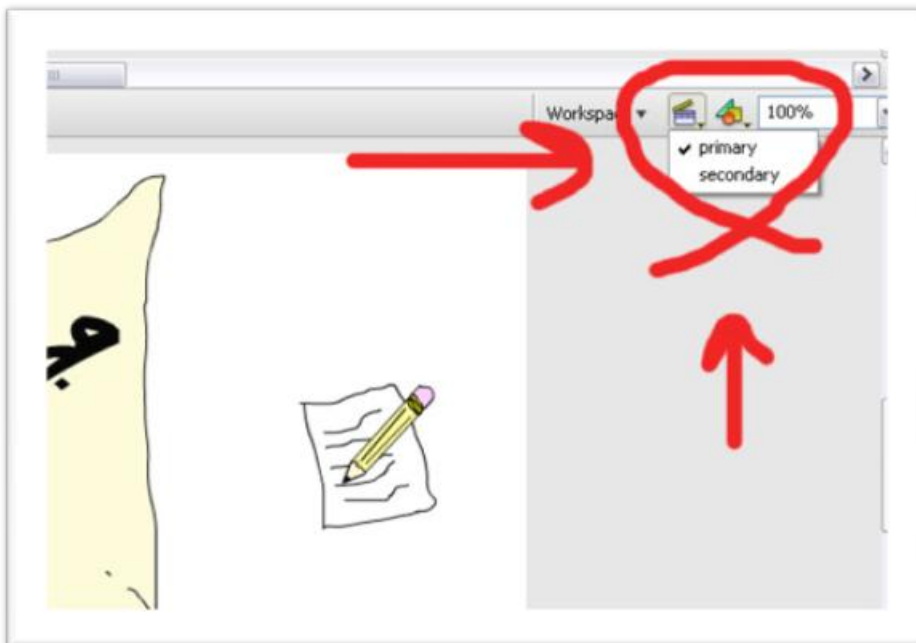
و یه آرایه دیگه از scene ها که اسم اونا به همراه تعداد فریم هاشون در این آرایه قرار داره.

خوب اینا رو ساده نگیرین اینا امکانات خیلی خوبی هستن . مخصوصا برای آدما شلخته ای مثل من تا فیلمشون رو خیلی منظم و باحال مدیریت کنن.

این مثال رو با هم ببینیم تا ببینیم چی میشه:



اگه فایل نمونه رو باز کنین و ببینین . متوجه می شین که تایم لاین دو تا scene داره به نام های primary و secondary



تو فریم اول کد زیر رو می بینیم

```
stop();

trace(scenes[0].name);
trace(scenes[0].numFrames);
trace(scenes[0].labels[0].name);
trace(scenes[0].labels[0].frame);

pages.stop();

output.addEventListener(MouseEvent.CLICK, onOutputClick, false, 0, true);

function onOutputClick(evt:MouseEvent):void {
    trace("tedade scene ha = " + scenes.length + " scenes.");
    trace("scene jari = '" + currentScene.name + "'.");
    trace("dar in scene " + currentScene.numFrames + " frame ,");
    trace(" va " + currentScene.labels.length + " label vojood darad. ");
    trace("labele aval az scene dovam '" + scenes[1].labels[0].name + " nam darad,");
    trace(" ke in label dar frame " + scenes[1].labels[0].frame + " gharar darad.");
    trace("Movie clip 'pages' " + pages.currentLabels.length + " label darad.");
    trace("va labele akhare an '" + pages.currentLabels.pop().name + "' nam darad.");
}
```

همونطور که تو کد می بینیم بعد از خط اول که تایم لاین رو stop می کنه. چهار دستور بعدی به ترتیب:

نام اولین scene - تعداد فریم های اولین scene - نام لیبل اول scene اول و شماره فریم لیبل اول از scene اول رو به پنجره خروجی می بره.

در خط بعد مووی کلیپ pages رو Stop می کنیم که در فریم اول خودش بمونه.

در خط بعد listener مربوط به کلید output رو تعریف می کنیم. تا با کلیک روی اون تابع onOutputClick اجرا بشه.

و اما کار تابع onOutputClick چیه؟

انتظار ندارین با توضیحات فینگلیشی که توی کدش نوشتم دیگه خط خط براتون توضیح بدم؟

اینا تعدادی از توابع کاربردی مربوط به این بخش هستن. حالا دیگه کار کشیدن از این توابع و امکانات به خلاقیت خودتون بستگی داره و اینکه کجاها بهش احساس نیاز کنین.

یه مبحث کوچولو و کاربردی دیگه رو هم بگم و درس امروز رو تموم کنم.

کنترل سرعت رد شدن فریم ها - frame Rate :

گه بدونین تو AS2 من چقدر دنبال این قابلیت گشتم و نتونستم چیزی پیدا کنم. آخر سر هم از یه کاربر کاردرست فلش (البته خارجی ها!) شنیدم که AS2 این قابلیت رو نداره. کلی تو روحیه تاثیر گذاشت

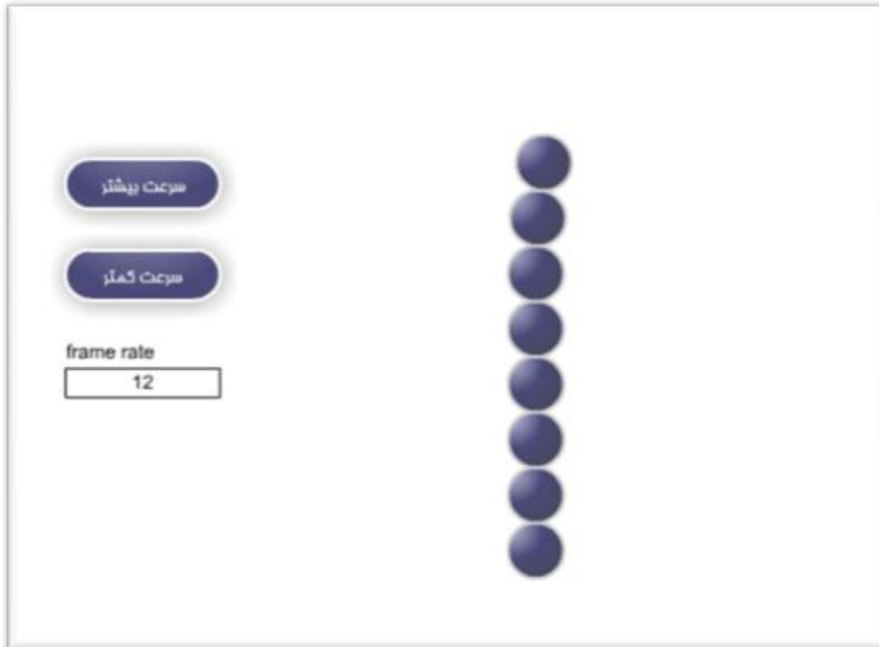
وقتی تو AS3 این قابلیت رو دیدم کلی حال کردم. ولی حیف! دیگه خیلی دیر بود... حالا برا شما می گم که دیگه این بلا سر شما نیاد!

همونطور که می دونین هر فیلم فلش یه property به نام frame rate داره که سرعت پخش فیلم رو تنظیم میکنه. تو AS2 شما فقط می تونستید برا کل فیلم یه سرعت ثابت داشته باشین.

اما این همیشه خوب نبود. مثلا من اون موقع می خواستم یه ماشین در حال حرکت داشته باشم که با فرشته شدن یه کلید ماشین گاز بخوره و سرعتش بیشتر و بیشتر بشه!!!

پس AS3 آستینش رو بالا زد و این قابلیت رو به برنامه نویسا داد تا از لذت بفرن و بتونن در حالت اجرا (run time) سرعت پخش فیلم رو تنظیم کنن.

مثال زیر رو ببینین.



ما می خوایم با فشردن کلیدهای افزایش سرعت و کاهش سرعت جابجایی توپ ها تغییر کنه.

این کد رو به فریم اول اضافه می کنیم.

```

info.text = stage.frameRate;

faster.addEventListener(MouseEvent.CLICK, onFasterClick, false, 0,
true);
slower.addEventListener(MouseEvent.CLICK, onSlowerClick, false, 0,
true);

function onFasterClick(evt:MouseEvent):void {
    stage.frameRate += 5;
    info.text = stage.frameRate;
}
function onSlowerClick(evt:MouseEvent):void {
    if (stage.frameRate > 5) {
        stage.frameRate -= 5;
    }
    info.text = stage.frameRate;
}

```

دیگه نیازی به توضیح نیست.... کلید سرعت بیشتر onFasterClick خاصیت frameRate مربوط به Stage رو ۵ تا اضافه و کلید سرعت کمتر 5 onSlowerClick تا از frameRate کم میکنه.

به همین سادگی!