**مستند سازی و توضیحات مربوط به کد Tour :**

* منظور از ویوی متحرک در این مستند مواردی مانند TextView,ImageView,LinearLayout و غیره است. هر چیزی که بخواهیم حرکت بدهیم یا اندازه و وضوح ان تغییر یابد.
* نمونه کد جدید از برنامه ی تور Grofers شبیه سازی شده.

فهرست :

|  |  |
| --- | --- |
| 1 | **متد ها و کلاس هایی که باید با ان آشنا باشید** |
| 2 | **توضیح اولیه در باره ی پروژه** |
| 3 | **توضیح انیمیشن و حرکت یک ویو از یک صفحه به صفحه ی دیگر** |
| 4 | **مراحل ایجاد و حرکت یک ویو به صورت انیمیشن مرحله به مرحله** |
| 5 | **توضیح positionOffset و position با تصویر** |
| 6 | **چگونه مشکل اندازه های مختلف دستگاه های اندرویدی را برطرف کنیم** |
| 7 | **استفاده از کد نمونه TOUR و ایجاد پروژه جدید** |
| 8 | **توضیح در باره ی page دارای listview با حرکت و انیمیشن از پیش تعریف شده ایتم ها و نیز ارتباط تعاملی کاربر با لیست ویو** |
| 9 | **زمانبندی کار جدید** |

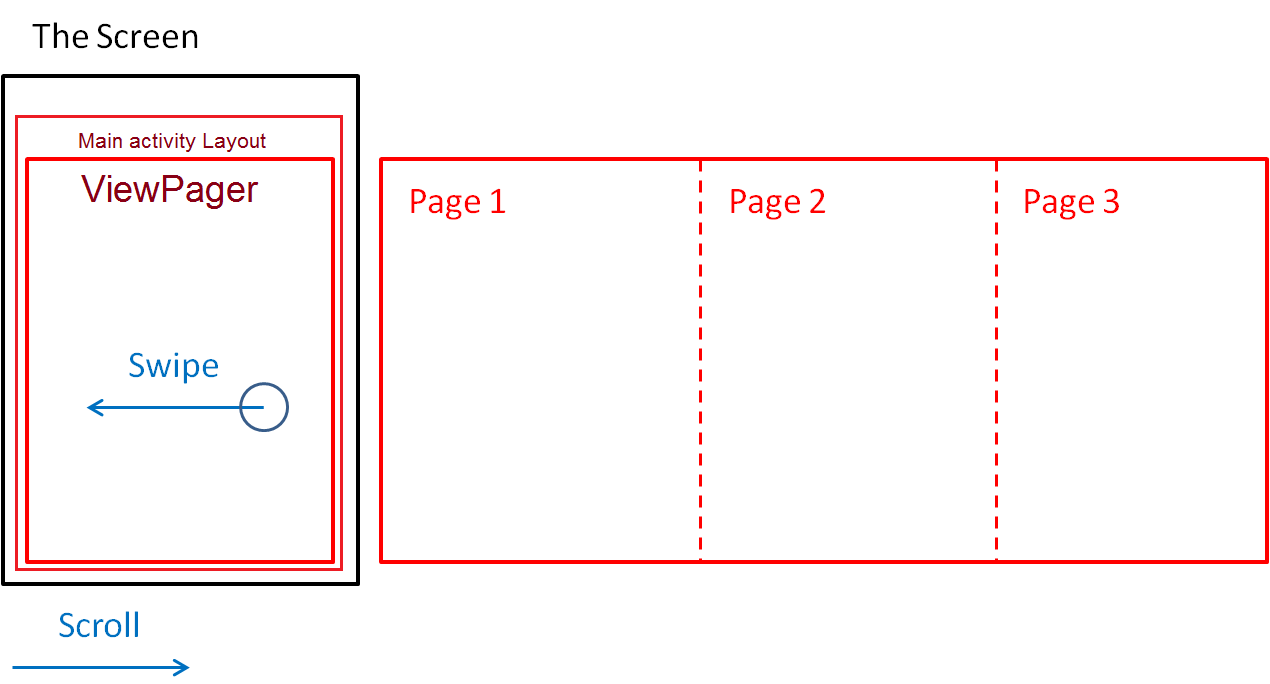
1. **متد ها و کلاس هایی که باید با ان آشنا باشید :**

* کلاس ViewPager و خصوصیت های آن از جمله onPageScrolled()
* متد های setAlpha() و setTranslationX() و setTranslationY() و setScaleX() و setScaleY() که جزو خصوصیت های یک ویو برای حرکت یا کوچک و بزرگ شدن اندازه و ... است

1. **توضیح اولیه در باره ی پروژه:**

برنامه دارای یک اکتیویتی اصلی بوده که UI آن یک فایل xml می باشد با نام intro\_layout.xml. درون این فایل xml یک ViewPager قرار دارد که می تواند شامل چند صفحه xml دیگر باشد. که هر کدام از این xml ها یک صفحه از برنامه tour را تشکیل می دهد که با حرکت انگشت بر روی اسکرین دستگاه اندرویدی صفحه ی بعدی یا قبلی وارد صفحه نمایش می شود.

Viewpager وظیفه ی انتقال وحرکت page را بر عهده دارد. همچنین می توان از امکان حرکت سریعتر یا کند تر ویو ها در درون هر صفحه Viewpager به نسبت اندازه انتقال (PositionOffset) استفاده کرد اما در این پروژه حرکتی یا انیمیشنی داخل viewpager انجام نشده و فقط انتقال بین page ها مورد نظر بوده است.همچنین حرکت ها یی که به این روش انجام شود محدود به همان page ایی است که آن ویو در آن قرار دارد به دلیل محدودیت ما روش دیگری را انتخاب کردیم تا بتوانیم حرکت بین page ها را شبیه سازی کنیم.



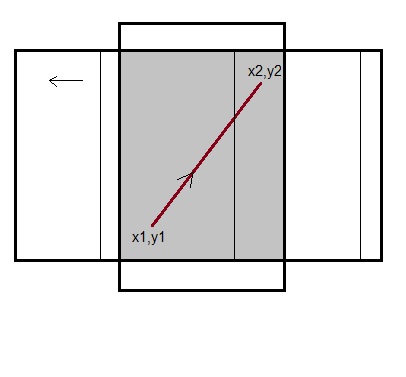
در کد زیر فایل intro\_layout.xml را مشاهده می کنید

<?xml version="1.0" encoding="utf-8"?>  
  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/rl\_bigest"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/backbg2">  
  
 <LinearLayout  
 ….>  
 <LinearLayout  
 ……>  
 <android.support.v4.view.ViewPager  
 android:id="@+id/viewpager"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent" />  
 </LinearLayout>  
 <LinearLayout  
 …..>  
 ………..  
 </LinearLayout>  
</RelativeLayout>

1. **توضیح انیمیشن و حرکت یک ویو از یک صفحه به صفحه ی دیگر**

نکته ی اصلی این پروژه این است که چگونه یک ویو یا چند ویو را از یک page به page دیگر انتقال بدهیم به صورتی که بتوان از این حرکت ها یک سناریو ساخت . همپنین حرکت های انجام شده متناسب با تغییر صفحه باشد.

همانطور که در بالا توصیح داده شد اگر ویو در ViewPager و در یک page از ان تعریف شود حرکت ان محدود به همان page میشود. برای حل این مشکل یک ویو به صورت آزاد و بدون طراحی اولیه xml بوسیله کد جاوا ایجاد می کنیم و ان را به یک layout که احتمالا بزرگترین لایه در intro\_layout.xml است اضافه کرده و حرکت ان را درهنگام انتقال بین page ها به گونه ایی تنظیم می کنیم که کاربر تصور می کند این ویو بین page ها در حال حرکت است اما در واقع این حرک­ت ها فقط محدود به x,y صفحه نمایش گوشی می باشد.

****

همانطور که در تصویر بالا مشاهده می کنید حرکت یک ویو کاملا محدود به حرکت عرض و طول screen در دستگاه اندرویدی است اما به دلیل این که page های ViewPager در حال حرکت هستند این طور به نظر می آید که ویو از یک page به page دیگر در حال حرکت است. در تصویر زیر نیز حرکت یک ویو در حین انتقال یک page از ViewPager نمایش داده شده است. یک نمونه انتقال pageو حرکت ویو متحرک را در تصویر زیر مشاهده می کنید.

|  |  |  |
| --- | --- | --- |
| **C:\Users\TGS\Desktop\a3.jpg** | **C:\Users\TGS\Desktop\a1.jpg** | **C:\Users\TGS\Desktop\a2.jpg** |
| C:\Users\TGS\Desktop\a7.jpg | C:\Users\TGS\Desktop\a6.jpg | C:\Users\TGS\Desktop\a5.jpg |

1. **مراحل ایجاد و حرکت یک ویو به صورت انیمیشن مرحله به مرحله :**

یک ImageView ایجاد کرده و طول و عرض ان را مشخص نموده و تصویر مناسب از Drawble را به ان الحاق می کنیم

ImageView movingObj1 = new ImageView(getApplicationContext());

movingObj1.setImageResource(R.drawable.*onboarding\_page\_1\_resource\_1*);

movingObj1.setLayoutParams(new RelativeLayout.LayoutParams(getXBasedOnNewScreen(movingObj1\_width), getYBasedOnNewScreen(movingObj1\_heigh)));

سپس لایه اصلی با نام rl\_bigest را که در xml ایجاد کردیم در جاوا نیز معرفی می کنیم و ویو ی متحرک را به ان اضافه می کنیم

rl\_bigest = (RelativeLayout) findViewById(R.id.*rl\_bigest*);  
  
rl\_bigest.addView(movingObj1);

حال باید مسیر حرکت را مشخص کنیم. (مجددا تاکید می شود x,y یک ویو محدود به طول و عرض screen صفحه نمایش است. )

برای جاگذاری یک ویو در X1,Y1 از متد setTransitionX() و setTransitionY() استفاده می شود.

movingObj1.setTranslationX(X1);  
movingObj1.setTranslationY(Y1);

یعنی ویوی movingObj1 در نقطه ی X1 , Y1 از screen قرار بگیرد .

قبل از این که توضیح حرکت یک ویو از X1,Y1 به X2,Y2 را اغاز کنیم باید با مفهموم نسبت انتقال اشنا شویم . یعنی تمام حرکت های ویو ها در داخل screen باید متناسب با حرکت صفحه در ViewPager تنظیم شوند. برای این کار ViewPager امکانی را در اختیار ما قرار می دهد تا بدانیم در هر لحظه از حرکت ، صفحه ی جاری در کجای screen قرار دارد .

mViewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {

@Override  
public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

....

});}

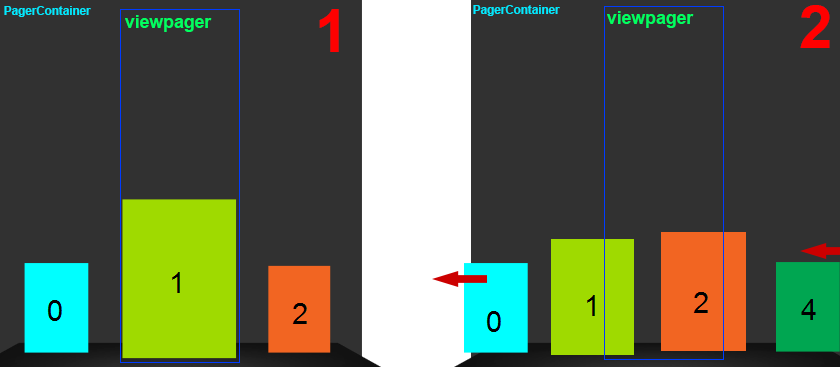
در متد onPageScrolled() به متغییری دسترسی دارید که می توانید با آن حرکت ویوی متحرک خود را با مقدار انتقال یک page به بیرون یا درون ViewPager تنظیم کنید. نام این متغییر positionOffset است که در بازه ی [0..1] قرار دارد . متغییر position نیز شماره ی page ایی است که در حال حاضر در حال نمایش یا انتقال است .

1. **توضیح positionOffset و position در تصویر زیر :**

در تصویر شماره 1 ، position برابر 1 وpositionOffset صفحه 1 هم برابر 0.0 است به محض حرکت صفحه ی 1 به سمت چپ و ورود صفحه 2 به screen مقدار positionOffset صفحه 1به تدریج افزایش یافته و زمانی که صفحه ی 2 کاملا در screen قرار بگیرد positionOffset صفحه 1 به 1.0 می رسد در این حالت positionOffset صفحه 2 برابر با 0.0 می شود.

نمونه تغییرات positionOffset در یک حرکت صفحه در viewpager :

{0.0, 0.01, 0.02 … 0.1, 0.2 …, 0.5, 0.6, … , 0.9,0.91,0.99,1.0 }



حال اگر بخواهیم در صفحه ی اولposition=1 از X1,Y1 به X2,Y2 برویم به این روش زیر عمل می کنیم

if (position == 0 &&) {

if ( positionOffset=1 ) {

MovingObj1.setTranslationX((float) (X1));  
movingObj1.setTranslationY((float) (Y1));

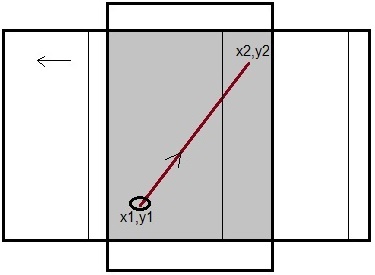
}

if ( positionOffset>0 ) {

MovingObj1.setTranslationX((float) (X1 + positionOffset\* (X2- X1)));  
movingObj1.setTranslationY((float) (Y1 + positionOffset\* (Y2- Y1)));

}  
}

این متد ویو را از موقعیت X1,Y1 با افزایش مقدار positionOffset تا 1.0 به X2,Y2 می رساند. یعنی در هر حرکت صفحه به چپ یا راست ویوی متحرک MovingObj1 مقداری به چپ یا راست می رود. با مقدارگذاری و محاسبه ی ساده ی ریاضی می توانید این حرکت را بهتر درک کنید .

****

1. **چگونه مشکل اندازه های مختلف دستگاه های اندرویدی را برطرف کنیم :**

برای حل این مورد یک اندازه ی استاندار در نظر میگیریم مانند LG G3

private int LG\_G3\_DIMENTIONS\_WIDTH = 1440;  
private int LG\_G3\_DIMENTIONS\_HEIGHT = 2392;

هر اندازه یا طول و عرض در دستگاه اندرویدی جدید را ضرب در نسبت این اندازه های از پیش تعریف شده بر طول و عرض می کنیم با متد های

getXBasedOnNewScreen(int oldX)

getYBasedOnNewScreen(int oldY)

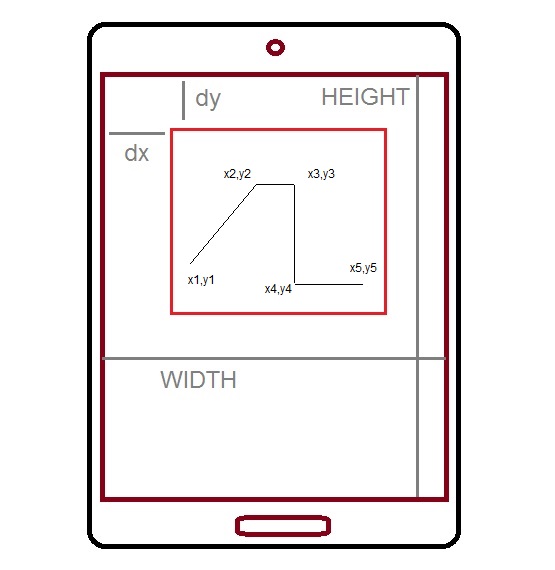
1. **برای استفاده از کد نمونه TOUR و ایجاد پروژه جدید باید مراحل زیر به ترتیب انجام شود:**
2. کپی کردن فایل های جاوا و xml و همچنین عکس ها از drawable پروژه کد نمونه به پروژه جدید – در صورت لزوم value.xml و color.xml نیز کپی شود .
3. مشخص کردن عکس های اصلی پس زمینه و قرار دادن آنها در فایل xml و محاسبه فاصله ی انها از بالا و از چپ صفحه screen
4. مشخص کردن نسبت های فاصله عرضی و طولی تصویر پس زمینه از سمت چپ و بالا ی SCREEN –

فاصله عرضی BACK\_IMAGE\_LEFT\_DISTANCE= dx/width

فاصله طولی BACK\_IMAGE\_TOP\_DISTANCE = dy/height

در طراحی فعلی این اندازه ها به صورت زیر است در صورتی که اندازه ی imageview تصاویر پس زمینه را نسبت به screen تغییر داده اید باید این مقادیر تغییر دهید.

private float BACK\_IMAGE\_TOP\_DISTANCE = 0.043f;  
private float BACK\_IMAGE\_LEFT\_DISTANCE = 0.2f;



1. مشخص کردن تصاویر متحرک و تعداد و اندازه ی انها.
2. مشخص کردن نقاطی که به ترتیب ویوِ متحرک در آنها قرار می گیرد. (نقاط مشخص شده باید مکان انها نسبت به طول و عرض تصویر پس زمینه مشخص شود)
3. قرار دادن این ویو ها در صفحه و حرکت دادن به انها در هر کدام از انتقال صفحات که لازم است.

initializations();

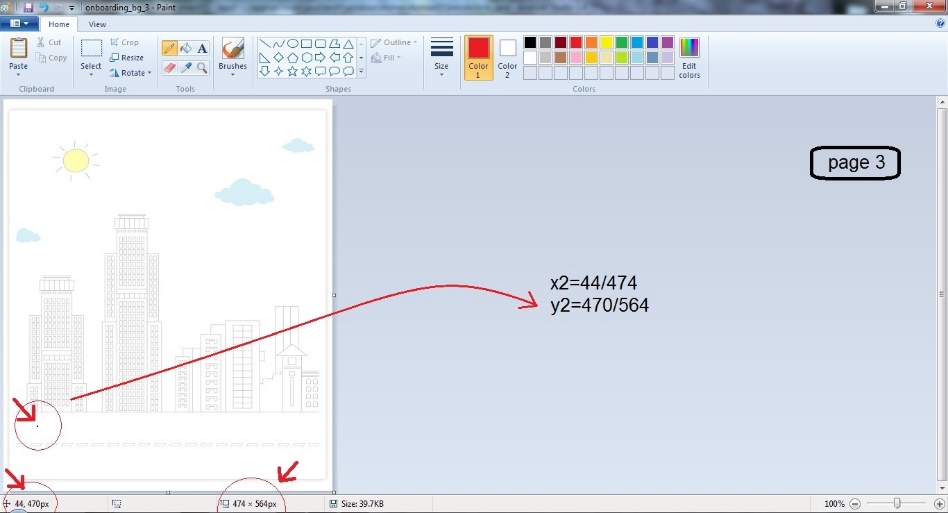
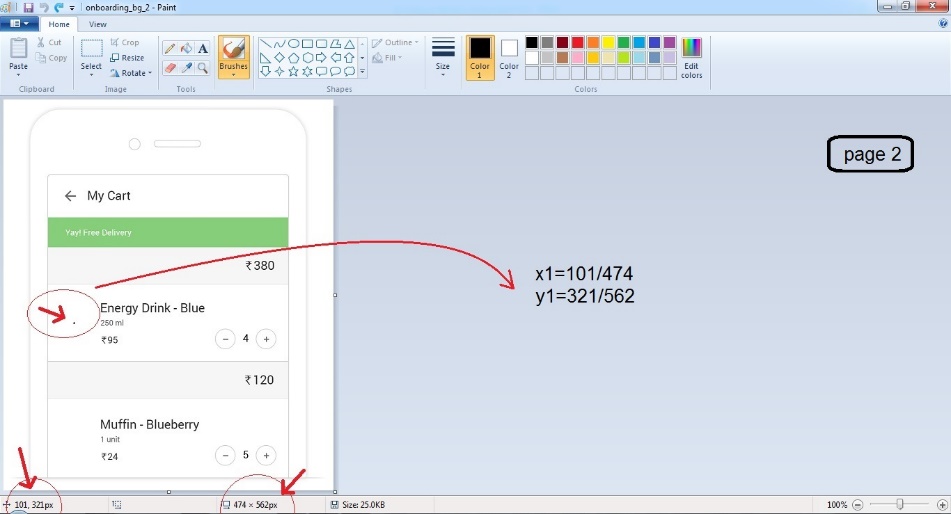
در این متد اندازه ی ویوهای متحرک و تعریف ویو و مشخص کردن تصویر انجام می شود . به همراه اضافه کردن ویو به لایه اصلی و نیز مشخص کردن point های ویو های متحرک.

positionsInitializationOFAnimations();

در این متد محاسبات point های مختلف ویوی ها متحرک نسبت به دستگاه اندرویدی جدید محاسبه و مقدار دهی تولیه می شود.

برای یافتن نقاط حرکت ساده ترین را استفاده از نرم افزار Paint در windows است.دقت شود نسبت فاصله ها مهم است نه فاصله با واحد پیکسل . حال با داشتن این مقادیر به همان روشی که در **بخش 4** توضیح داده شد می توانید مراحل انتقال ویو متحرک را انجام دهید .

x\_movingObj3\_init\_onCenterPic = (float) 101 / 474;  
y\_movingObj3\_init\_onCenterPic = (float) 321 / 562;  
x\_movingObj3\_post1\_onCenterPic = (float) 44 / 474;  
y\_movingObj3\_post1\_onCenterPic = (float) 470 / 564;



نکته مهم : نحوه ی حرکت و انیمیشن و کوچک یا بزرگ شدن و یا fade شدن و هر تغییر دیگر بر روی ویوی متحرک در هر مکان با کمی خلاقیت و حوصله قابل انجام است. برای مثال بهتر است ویو ها برای بار اول ناگهانی بر روی صفحه نمایش ظاهر نشوند بلکه بوسیله متغییر نسبت حرکت positionOffsetان را با ورود به صفحه از اندازه کوچک به بزرگ تبدیل کنید یا با متد زیر وضوح ان را بیشتر کنید.

movingObj3.setAlpha((float) (positionOffset));  
movingObj4.setAlpha((float) (positionOffset));

1. **توضیح در باره ی page دارای listview با حرکت و انیمیشن از پیش تعریف شده ایتم ها و نیز ارتباط تعاملی کاربر با لیست ویو**

در کد جاوا IntroFragment.java در متد setMenuVisibility در لحظه ی ورود به page شماره 4با یک سری محاسبات بررسی می شود که ایا ایتم اول لیست در بالای لسیت ویو نمایش داده شده است یا نه اگر پاسخ مثبت یود یا متد smoothScrollToPositionFromTop() لسیت ویو از اولیتم حرکت و به پایین می رود در غیر این صورت لیست به سمت بالا حرکت می کند . پارامتر سوم متد smoothScrollToPositionFromTop() مدت زمانی که این حرکت باید به طول بیانجامد را مشخص می کند. 3000 یعنی 3 ثانیه.

public void setMenuVisibility(final boolean visible) {  
 super.setMenuVisibility(visible);  
 if (visible && mPage == 4) {  
  
 if ((mListView.getCount() - 1) - mListView.getLastVisiblePosition() < 1) {  
 mListView.smoothScrollToPositionFromTop(0, mListView.getCount(), 3000);  
 } else {  
 mListView.smoothScrollToPositionFromTop(0, 0, 5000);  
 mListView.smoothScrollToPositionFromTop(mListView.getCount() - 1, 0, 3000);  
 }  
 }  
  
}

در متد

OnCreateView()

از IntroFragment می توان Listview را تعریف نموده و امکان کلیک بر روی ایتم ها و تعاملات کاربر با لیست را فراهم ساخت.

نکته ی مهم : مکان مرکز ImageView (ویو ی متحرک) برای جاگذاری ها و مسیر ها استفاده می شود .

x\_movingObj1\_init = (int) (-getXBasedOnNewScreen(movingObj1\_width) / 2 + getDimensionsOfSmartPhoneScreen().x \* (BACK\_IMAGE\_LEFT\_DISTANCE) + dimensOfBackGroundImage.width \* x\_movingObj1\_init\_onCenterPic);  
y\_movingObj1\_init = (int) (-getYBasedOnNewScreen(movingObj1\_heigh) / 2 + getDimensionsOfSmartPhoneScreen().y \* (BACK\_IMAGE\_TOP\_DISTANCE) + dimensOfBackGroundImage.height \* y\_movingObj1\_init\_onCenterPic);

کد زیر مربوط به این مورد می باشد.

(-getXBasedOnNewScreen(movingObj1\_width) / 2

(-getYBasedOnNewScreen(movingObj1\_heigh) / 2

پایان

**زمانبندی :**

**برای یادگیری این روش برای بار اول نیاز به یک روز تا یک روز و نصف زمان می باشد**

**برای پیاده سازی و استفاده از کد در پروژه Tour پس از یادگیری روش انجام کار ،نیم تا یک روز کار ی کافیست البته بستگی به تعداد انیمیشن ها و page های برنامه دارد در صورتی که چندین انیمیشن و چندین صفحه طراحی شود زمان بیشتری نیاز است.**