

توسعه برنامه های

موبایل

جلسه هفتم مجازی

بخش سوم

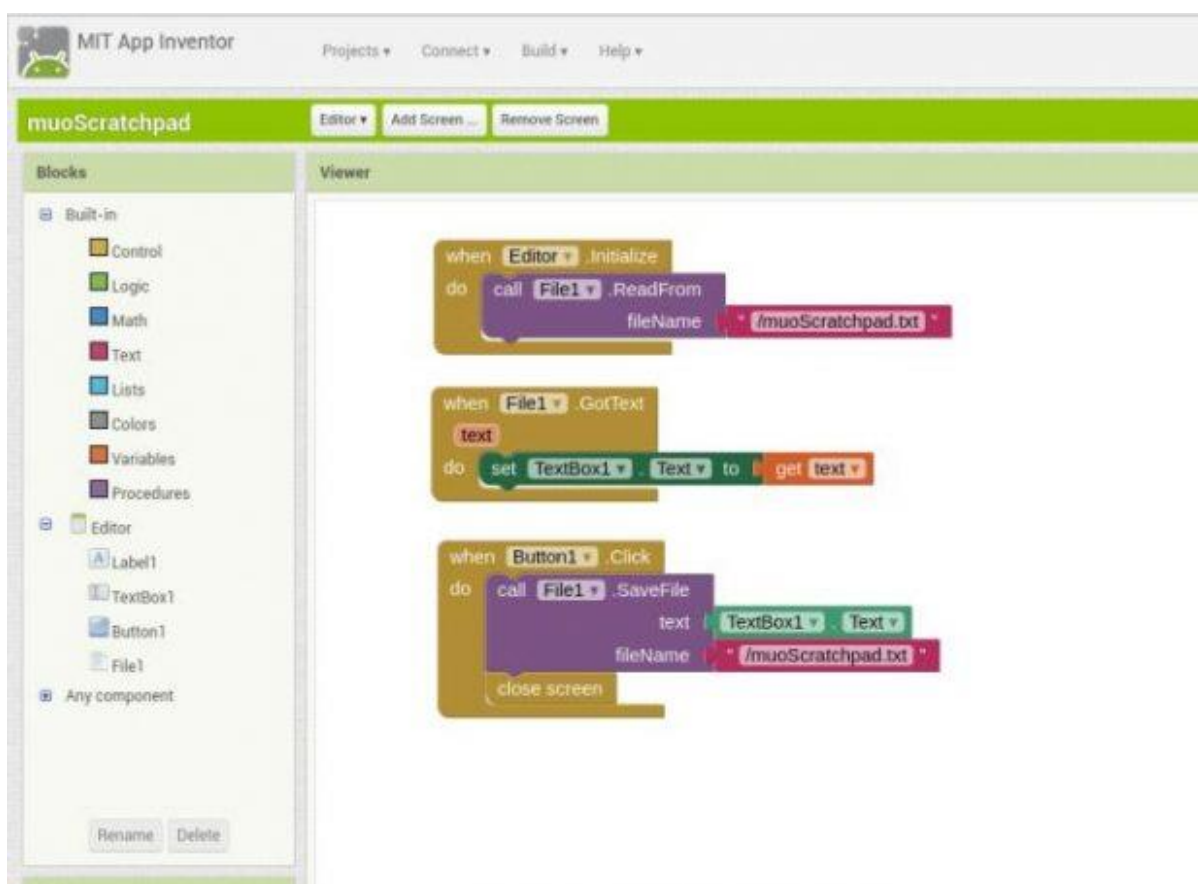
سحر صادقی

آموزش ساخت برنامه اندروید به زبانی ساده؛

ساخت برنامه اندروید همیشه نیاز به فراگیری یک زبان برنامه نویسی ندارد. برای آموزش برنامه نویسی اندروید می توان از ابزارهای آنلاین استفاده کرد.

در **قسمت اول** آموزش برنامه نویسی اندروید به دلایل یادگیری روش ایجاد یک اپلیکیشن اندرویدی اشاره کرده و ابزار آنلاین App Inventor که توسط محققان دانشگاه MIT توسعه پیدا کرده است را معرفی کردیم. در **قسمت دوم** مقدماتی از نحوه استفاده از ابزار App Inventor و چگونگی نصب و راه اندازی شبیه ساز آن را ذکر کردیم. اکنون که نخستین لایه بندی را در ساخت اپلیکیشن اندرویدی خود انجام داده ایم، بهتر است بیشتر به توسعه قابلیت های این ابزار نوت برداری بپردازیم.

از قسمت بالا و سمت چپ صفحه بر روی دکمه Blocks کلیک کنید. همانند نحوه نمایش Designer در اینجا نیز لایه بندی مشابهی دیده می شود، اما در عوض گزینه های متفاوتی در دسته بندی های متنوع در اختیار کاربر قرار می گیرند. این ها به جای کنترل های رابط کاربری، در واقع مفاهیمی از برنامه نویسی هستند، اما همانند هر نوع نمایش دیگر اکنون نیز می توان با استفاده از تکنیک کشیدن و رها کردن، قسمت های مختلف نشان داده شده را به عنوان بخشی از نرم افزار اندرویدی خود کنار هم قرار داد.



در جعبه سمت چپ صفحه آیتم‌هایی همچون کنترل (Control)، نوشته (Text) و متغیرها (Variables) در دسته‌بندی Built-in دیده می‌شود. بخش‌هایی که در این دسته‌بندی‌ها وجود دارند، براساس عملکرد از یکدیگر جدا شده‌اند. در واقع نام آن‌ها به خوبی عملکردشان را نشان می‌دهد. به عنوان نمونه آیتم ریاضی (Math) قادر به انجام اعمال محاسباتی است.

در پایین این قسمت لیستی از عناصر صفحه‌ها وجود دارد. بلاک‌هایی که در اینجا قرار دارند، عناصر صفحه‌های شما را تحت تأثیر قرار می‌دهند. به عنوان نمونه، کلیک کردن بر روی یکی از برجسب‌ها (Label)، بلاک‌هایی که می‌توانند نوشته آن برجسب را تغییر دهند را نشان می‌دهد. اما دکمه‌ها (Buttons) شامل بلاک‌هایی هستند که عملکرد بعد از کلیک شدن بر روی آن‌ها را تعریف می‌کنند.

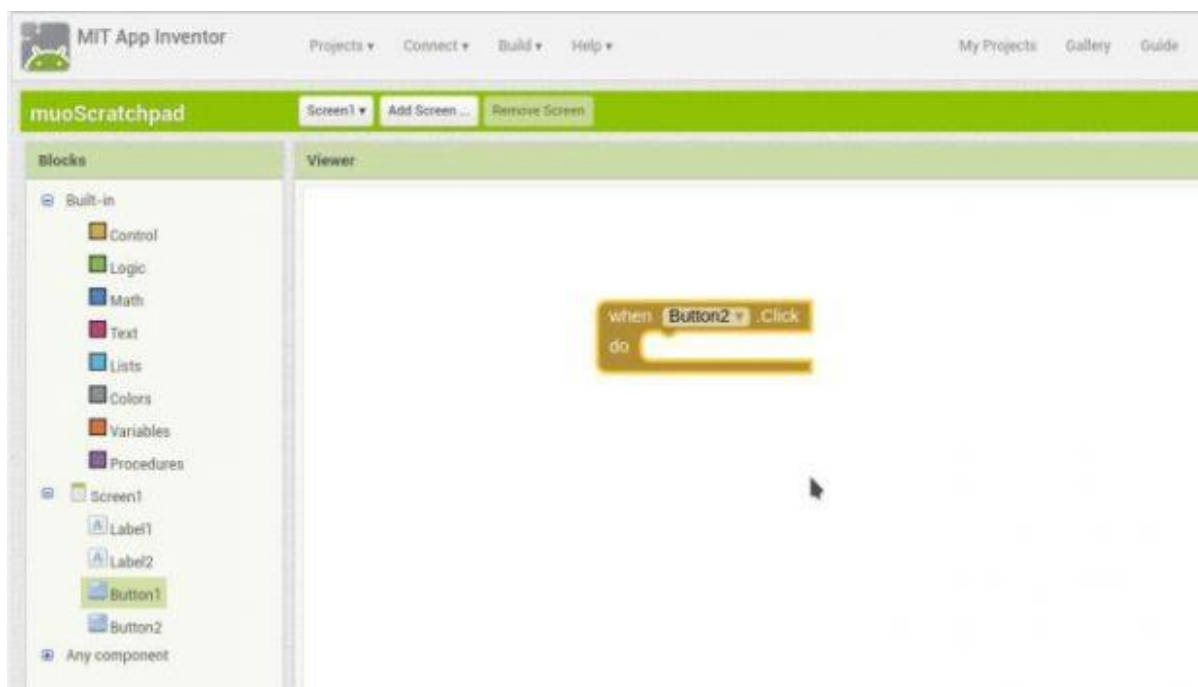
علاوه بر دسته‌بندی‌ها که براساس رنگ از یکدیگر جدا شده‌اند، هر بلاک دارای یک شکل خاص است که هدف و کاربرد آن را نشان می‌دهد. این شکل‌ها به صورت زیر تقسیم می‌شوند:

- می‌توان به آیتم‌های دارای شکاف بزرگ در وسط خود همچون if-then که در بالا نمایش داده شده است، به چشم یکی از رخ داده‌ها (events) کنترل نگاه کرد. وقتی بعد از ساخت برنامه اندروید اتفاقی در آن رخ می‌دهد، سایر دستوراتی که در شکاف این قسمت قرار دارند اجرا می‌شوند.
- بلاک‌های تخت دارای اتصال ممکن است statements یا expressions باشند. ابتدا توضیحات (statements) هستند که هم‌ارز فرمان‌ها محسوب می‌شوند. آیتم‌های در جریان بالا نشان‌دهنده آن‌ها هستند. در عکس بالا، بلاک‌های close application و make a list توضیح محسوب می‌شوند.
- حالت دوم عبارات (expressions) است که کمی با statements تفاوت دارد. به عنوان نمونه یک توضیح می‌گوید «این را بر روی عدد ۴۲ قرار بده». اما در یک عبارت گفته می‌شود «۲۲ را به ۲۰ اضافه کن و نتیجه را به من برگردان». در تصویر بالا، is in list یک عبارت است که صحیح یا غلط بودن را ارزیابی می‌کند. عبارات نیز بلاک‌های تخت و مسطح هستند، اما معمولاً در سمت چپ خود دارای یک تب بوده و در سمت راست یک بریدگی و شکاف دارند.
- در نهایت با متغیرها (values) سر و کار داریم که اعداد (مثلاً «۴۲» یا «۱۷»)، رشته‌ای از حروف (مثلاً «عبارت یک» یا «عبارت دو») و یا صحیح و غلط را شامل می‌شوند.

App Inventor یک ابزار کامل است و یادگیری نحوه صحیح استفاده از آن برای ساخت برنامه اندروید نیازمند آزمون و خطا است، پس، از کلیک کردن بر روی دکمه‌ها و قسمت‌های مختلف ترس نداشته باشید. اما همیشه کاربردهر دکمه و نتیجه کلیک کردن بر روی آن را به خاطر بسپارید.

در صفحه ابتدایی، دو آیتم وجود دارند (دکمه‌ها) که نیازمند توجه بیشتری هستند Button2. در زمان کلیک شدن بر روی آن اپلیکیشن را می‌بندد. از آنجایی که قرار است بعد از کلیک بر روی این دکمه فعل و انفعالاتی

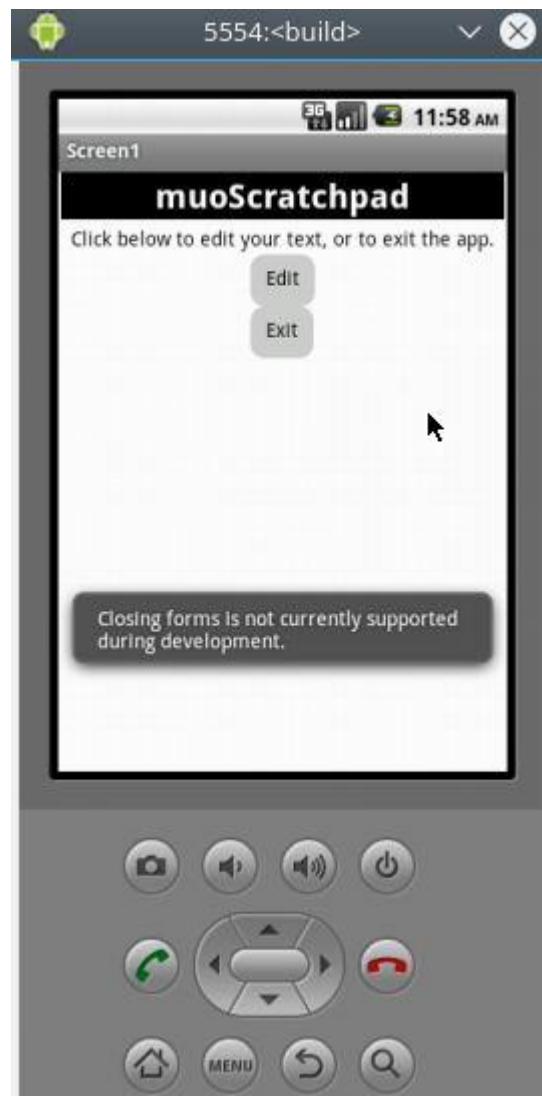
صورت بگیرد، با جستجو در بلاک دکمه‌ها، به دنبال عبارت when Button2.click می‌گردیم. این عبارت مشخص می‌کند وقتی بر روی این دکمه کلیک شد، یک کار خاص انجام شود. بعد از یافتن آیت، آن را با تکنیک کشیدن و رها کردن به درون صفحه بیندازید.



حال می‌خواهیم وقتی این دکمه لمس شد برنامه بسته شود. در قسمت Built-in و در زیرشاخه Control، بلاکی با عنوان close application را مشاهده می‌کنیم. آن را کشیده و به درون شکاف سفید رنگی که جلوی کلمه do قرار دارد بیندازید.



حالا وقتی این دکمه را لمس کنید، نرم افزار اندرویدی تان بسته خواهد شد. اجازه دهید یک مرتبه آن را در شبیه ساز تست کنیم. در ادامه با یک ارور مبنی بر عدم پشتیبانی از این عملکرد در محیط توسعه و ساخت برنامه اندروید مواجه می شویم که نشان دهنده کارکرد صحیح آن است.

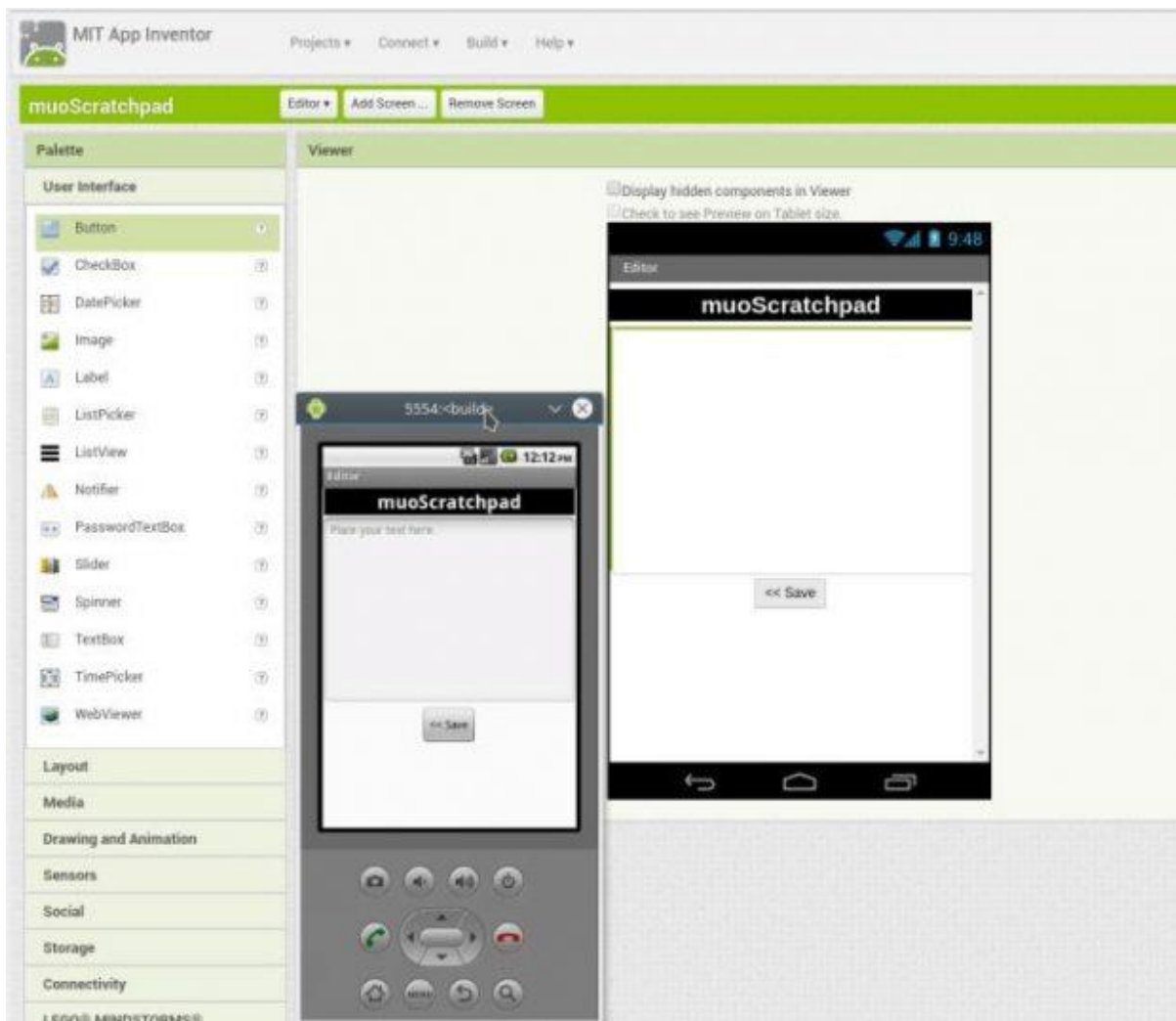


ساخت دومین صفحه (صفحه ویرایش)

اکنون تمرکز خود را بر روی دکمه نخست (Button1) قرار می دهیم. لمس این دکمه قرار است ویرایشگر برنامه ما را باز کند. پس ابتدا باید از وجود خود این ویرایشگر مطمئن شویم! به نحوه نمایش Designer بازگشته و یک صفحه جدید با برچسب مشابه صفحه نخست ایجاد می کنیم.

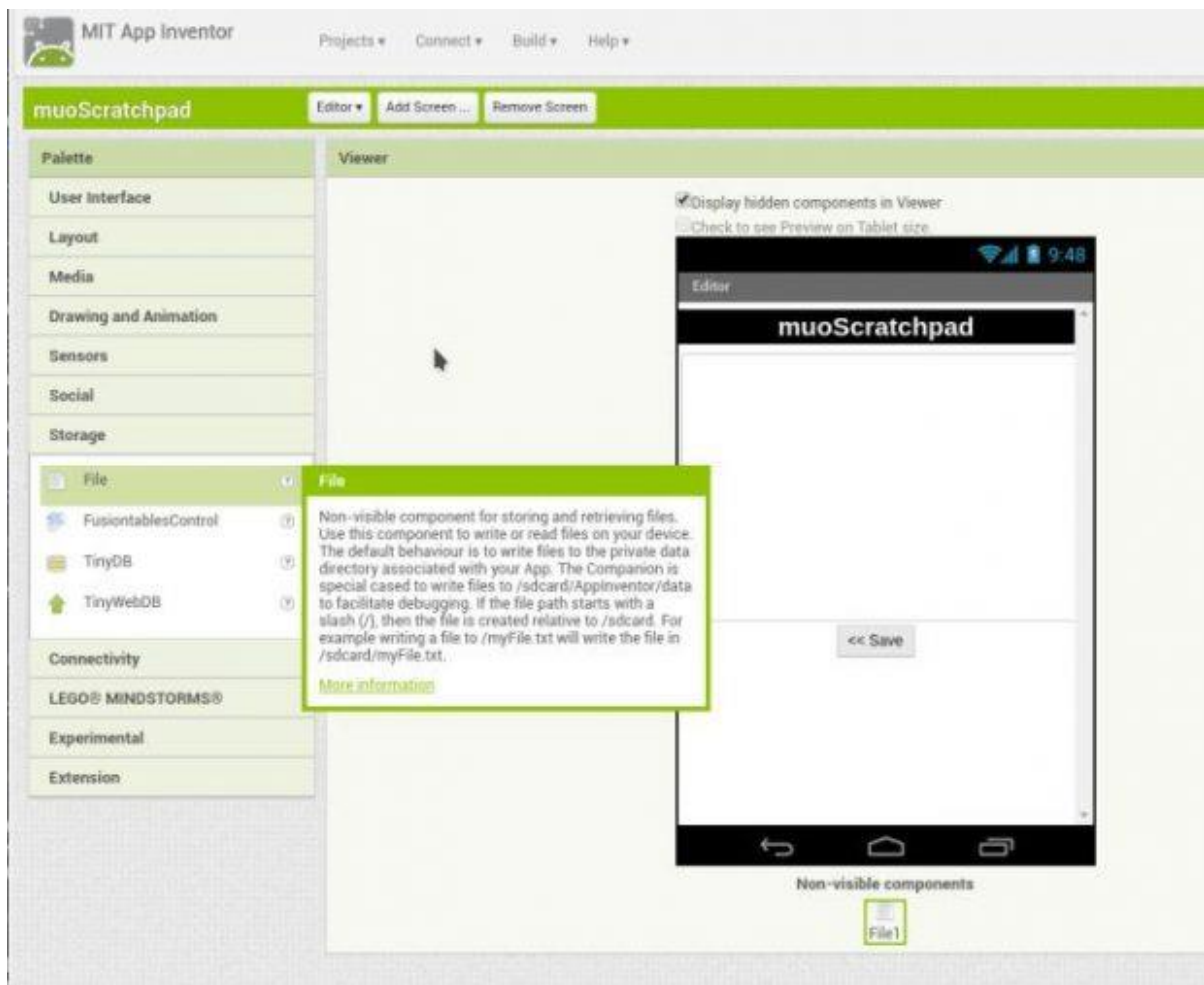
همچنین باید یک TextBox ایجاد کنیم. در بخش گزینه های آن، عرض (Width) را بر روی حالت fill parent قرار داده و ارتفاع (Height) را بر روی ۵۰ درصد تنظیم کنید. همچنین Multiline باید فعال باشد.

این بخش قرار است محتوای ما را در دل خود نگه دارد. به یک دکمه دیگر نیز با برچسب Save << نیاز داریم. بعد از ایجاد این موارد، تغییرات اعمال شده را در شبیه‌ساز تست کنید.



پیش از ادامه سایر مراحل ساخت نرم‌افزار اندروید، باید یک نکته مهم را یادآور شویم. قرار است محتواهای درون TextBox را یکی پس از دیگری انباشته کنیم که نیاز به ذخیره‌سازی داده‌ها را نشان می‌دهد. چندین راهکار برای حل این مسئله وجود دارد.

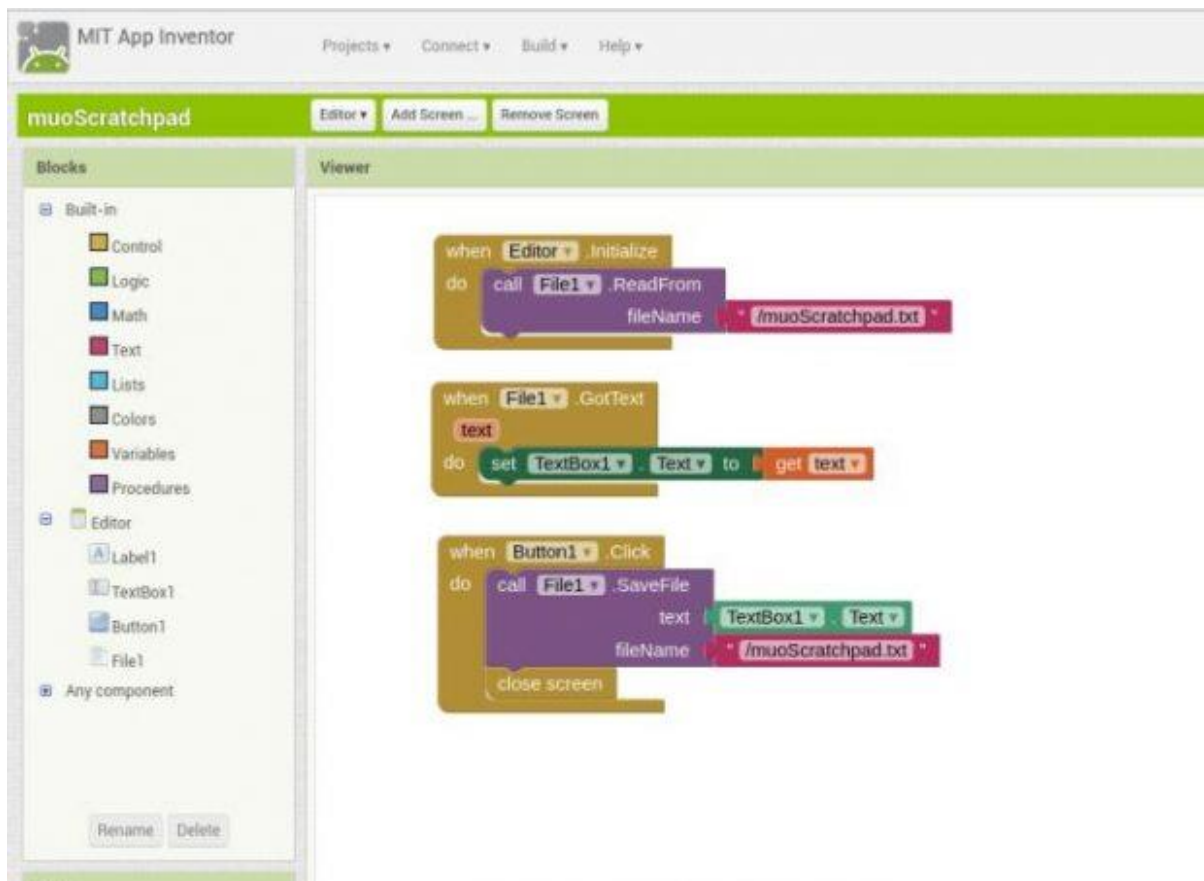
سراسرترین مورد File است. از آنجایی که ما فقط یک متن ساده می‌خواهیم، این نوع کاملاً کافی خواهد بود. وقتی این مورد را در Viewer قرار دهید، متوجه عدم نمایش آن می‌شوید، چرا که File یک کامپوننت غیر قابل مشاهده است. البته دلیلی نیز برای نمایش آن به کاربر وجود ندارد. این بخش تمامی وظایف خود را در پس زمینه انجام داده و محتواهای اپلیکیشن را در قالب یک فایل در حافظه دستگاه ذخیره می‌کند. اما در هر صورت اگر دوست دارید این آیتم قابل مشاهده باشد، کافی است تیک کنار Display hidden components را در Viewer بزنید.



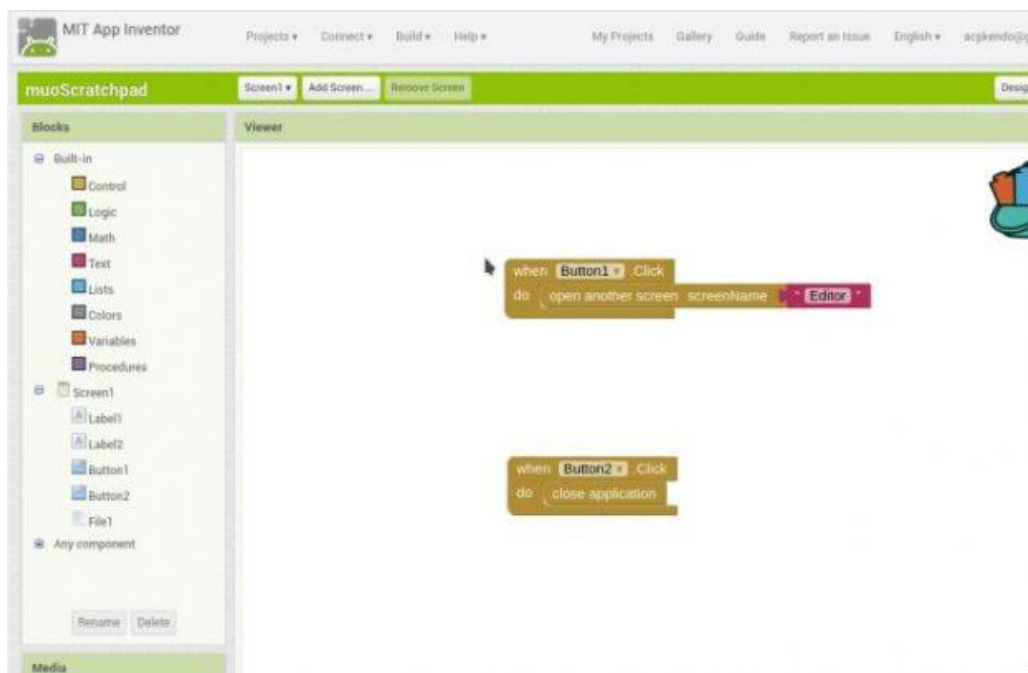
به حالت نمایش Blocks برگردید تا برای پیشبرد ساخت برنامه اندروید خود چند دستور به آن اضافه کنیم. تنها عملکرد مورد نیاز، تعیین شیوه رفتار اپلیکیشن بعد از لمس دکمه << Save است. پس به بلاک when Button1.click نیاز است. در این قسمت توانایی‌های App Inventor به خوبی نمایان می‌شوند!

ابتدا محتوای TextBox را با گرفتن بلاک call File1.saveFile ذخیره کرده و در درون آن نوشته دلخواه خود را قرار می‌دهیم. برای این کار باید از TextBox1.text که وظیفه آن بازگرداندن محتواها است استفاده کرد. همچنین باید برای ذخیره‌سازی این نوشته یک فایل ایجاد کنیم. کافی است مسیر و نام فایل Text Block را مشخص کنید. اگر این فایل وجود نداشته باشد، اپلیکیشن آن را ایجاد خواهد کرد.

اکنون باید صفحه را برای بارگذاری محتواهای این فایل در زمان باز شدن آن تنظیم کنیم. برای این کار از بلاک when Editor.initialize در Editor استفاده کنید. با کمک آیتم call File1.ReadFrom به فایل خود اشاره می‌کنیم. می‌توان نتیجه خواندن فایل متنی را با استفاده از > when File1.GotText File ثبت کرده و محتوای آن را به کمک بلاک > set TextBox.Text to TextBox ارجاع داد. سپس باید مقدار get text را به آن داد. در نهایت و بعد از ذخیره، به یک کلیک بر روی Button1 نیاز داریم تا ما را به صفحه اصلی بازگرداند. برای این کار از یک بلاک close screen استفاده می‌کنیم.



آخرین گام، بازگشت به صفحه اصلی و تنظیم نخستین دکمه است. در واقع می‌خواهیم این دکمه ما را به صفحه ویرایشگر منتقل کند. با استفاده از بلاک `Control > open another screen` این کار همچون آب خوردن خواهد بود! البته در جلوی آن باید به Editor اشاره کنید.



قدم بعدی در ساخت برنامه اندروید

حالا که یک برنامه قابل اجرا ایجاد کرده‌اید، باید آن را توسعه بدهید. همان‌طور که در قسمت اول آموزش ساخت برنامه اندروید به آن اشاره کردیم، بعد از ایجاد اپلیکیشن نوت‌برداری ساده خود به کمک App Inventor آن را در اندروید استودیو گسترش خواهیم داد. گستره‌ای از دستورات زبان برنامه‌نویسی اندروید را در اختیار کاربر قرار می‌دهد، اما بهتر است برای افزایش کاربرد صفحه‌های ساده‌ای که ایجاد کرده‌ایم، از ابزارهای حرفه‌ای‌تر کمک بگیریم. می‌توان به این صفحه‌ها قابلیت‌هایی همچون پخش فایل چندرسانه‌ای، ارسال متن و یا حتی یک نمایش وب زنده را اضافه کرد.

یکی از اولین قدم‌هایی که به ذهن می‌آید، افزودن قابلیت انتخاب از میان چندین فایل است. اما بایک جستجوی ساده در گوگل متوجه می‌شوید این قابلیت به چند ترفند حرفه‌ای نیاز دارد. برای افزودن آن و همچنین سایر قابلیت‌ها، از اندروید استودیو استفاده کرده و از زبان برنامه‌نویسی جاوا کمک می‌گیریم. در قسمت بعد، کار توسعه اپلیکیشن ساده خود در اندروید استودیو را پیش می‌گیریم.