

به نام خدا



دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر

سیستم‌های نهفته‌ی بی‌درنگ

تمرین بازی پینگ پونگ با کنترل‌های حرکتی

(Motion Controlled Pong Game)

با هدف آشنایی با سیستم عامل اندروید و استفاده از سنسورهای تلفن همراه

اساتید:

دکتر مهدی کارگهی، دکتر مهدی مدرسی

نیمسال دوم ۱۴۰۱-۱۴۰۲

۱- مقدمه

امروزه تلفن های همراه هوشمند پیشرفت چشمگیری کرده اند و دارای قدرت پردازشی بالا، تنوعی از سنسورهای مختلف با دقت بالا و پشتیبانی از چندین شبکه ارتباطی مختلف مانند WiFi، Bluetooth و غیره هستند. بعلاوه، تمام این امکانات در ابعاد پایین و وزن کم در اختیار کاربران قرار گرفته اند. حال سؤال اساسی اینجاست که چطور می توان از این امکانات که همیشه همراه است، استفاده کرد؟ آیا می توان از یک تلفن همراه در کاربردهای صنعتی و تجاری به جای board های مرسوم استفاده نمود؟ چه محدودیتهایی برای این کار وجود دارد؟ به دنبال پاسخی برای سوالات فوق هستیم.

در این تمرین قرار است با سیستم عامل اندروید و امکاناتی که جهت تعامل با سنسورهای موبایل (این [لینک](#) را مطالعه کنید) در اختیار ما قرار می دهد آشنا شویم. همچنین با برخی محدودیت هایی که سیستم عامل جهت استفاده از سنسور ها و ارتباط با سخت افزار لایه پایین ایجاد می کند، آشنا خواهیم شد.

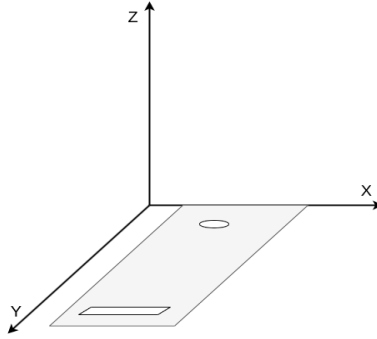
۲- شرح تمرین

در این تمرین هدف این است که یک بازی شبیه به بازی Pong بسازیم که محیط آن شامل یک توپ و یک راکت است. این بازی به صورت تک کاربره طراحی شده و با استفاده از سنسورهای شتاب سنج (accelerometer) وژیروسکوپ (gyroscope) در تلفن هوشمند با سیستم عامل اندروید کنترل می شود.



شکل ۱- نحوه قرارگیری دستگاه بر روی میز و تصویر محیط تمرین

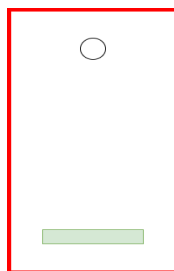
همان طوری که در شکل بالا مشخص شده است، برای شروع و انجام بازی دستگاه بر روی یک سطح صاف مانند میز قرار دارد. محیط بازی روی میز به طول ۵۰ سانتی متر است؛ دستگاه ۲۵ سانتی متر به سمت راست و ۲۵ سانتی متر به سمت چپ اجازه حرکت روی میز را خواهد داشت. در هر جهت که دستگاه را حرکت دهیم، راکت نیز در همان جهت حرکت خواهد کرد. همچنین، در صورتی که دستگاه را حول محور عمود بر آن بچرخانیم، راکت نیز به صورت مشابه خواهد چرخید. به شکل زیر توجه کنید.



شکل ۲- نحوه قرار گیری دستگاه در مختصات سه بعدی

به طور کلی دو موجودیت راکت و توپ پیاده‌سازی می شوند که دارای مشخصات زیر هستند:

۱. هنگام شروع بازی دستگاه روی میز و راکت در وسط عرض صفحه نمایش دستگاه قرار دارد (از نظر ارتفاع، راکت را در $\frac{1}{4}$ پایینی صفحه نمایش قرار دهید). طول راکت به اندازه $\frac{1}{3}$ عرض صفحه نمایش است. برای محیط بازی روی میز ۵۰ سانتی‌متر در نظر گرفته شده است که این مقدار می بایست به عرض صفحه نمایش دستگاه شما نگاشت شود. کنترل راکت با حرکت دستگاه در جهت محور X و با کمک سنسور شتاب‌سنج انجام می‌شود. در واقع هدف این است که موقعیت راکت نسبت به تغییر موقعیت دستگاه در طول این ۵۰ سانتی متر تغییر کند. به عنوان مثال، هنگام شروع بازی دستگاه ۲۵ سانتی‌متر به سمت راست و ۲۵ سانتی‌متر به سمت چپ اجازه حرکت دارد. وقتی دستگاه را ۲۵ سانتی‌متر به سمت راست محور X حرکت دهیم، راکت نیز در صفحه نمایش به سمت راست حرکت کرده، تا جایی که به طور کامل از سمت راست صفحه خارج شود. بدیهی است که حرکت بیش از ۲۵ سانتی‌متر تاثیری روی راکتی که به طور کامل از صفحه خارج شده است، ندارد. در این حالت باید دستگاه را به اندازه ۵۰ سانتی متر به سمت چپ محور X حرکت دهیم تا راکت به طور کامل از سمت چپ صفحه خارج شود.
۲. زاویه قرارگیری راکت باید توسط مقادیر سنسور ژيروسکوپ در محور Z حساب شود. این زاویه باید به دقت محاسبه گردد. این به این معنی است که اگر دستگاه ۱۰ درجه در محور Z ژيروسکوپ دوران داشت، راکت نیز باید در صفحه دستگاه ۱۰ درجه دوران کند.
۳. توپ در ابتدای بازی از بالا به سمت پایین سقوط آزاد می‌کند و در برخورد با راکت برمی‌گردد. زاویه این بازگشت باید با توجه به زاویه قرارگیری راکت انجام گیرد و به صورتی باشد که در بازی واقعی از توپ پینگ‌پونگ انتظار داریم.
۴. توپ در صفحه بین سه دیواره قرار دارد که این دیواره‌ها با رنگ قرمز در شکل زیر مشخص شده‌اند. دیواره‌ها به گونه ای هستند که توپ موقع برخورد با آن‌ها بازتاب یابد. برای جزئیات این کار به بخش پیاده‌سازی رجوع کنید.
۵. نیاز است که دکمه ریست در این تمرین تعبیه شود. در صورت ضربه زدن روی راکت باید محل توپ و راکت ریست شده و بازی به حالت اولیه خود بازگردند.



شکل ۳- دیواره‌ها در محیط بازی

۳- پیاده سازی

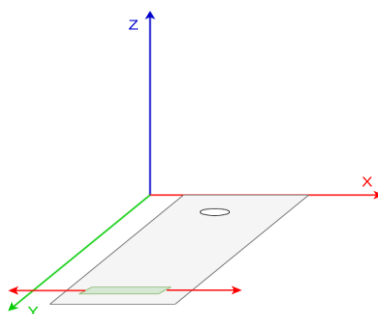
برای پیاده‌سازی این تمرین در ابتدا روشی را برای رسم اجزای دو بعدی روی صفحه انتخاب کنید. انتخاب کتابخانه دو بعدی بر عهده شما می‌باشد، اما دو گزینه زیر در سیستم عامل به صورت پیش فرض وجود دارند که می‌توانید از آن‌ها انتخاب نمایید:¹

- [Canvas](#)
- [OpenGL](#)

استفاده از سنسورها در اندروید با استفاده از [SensorManager](#) امکان پذیر است. همچنین می‌توانید مبحث [Motion Sensors](#) را نیز در راهنمای اندروید مطالعه کنید.

۳-۱ کنترل راکت

طراحی راکت به صورت یک مستطیل در فضای صفحه بازی می‌باشد که این مستطیل باید در صفحه طبق شکل زیر پیاده‌سازی شود.



شکل ۴- حرکت راکت در محور x در مختصات سه بعدی

¹ در صورتی که قصد دارید از این لیست استفاده کنید، پیشنهاد می‌شود از Canvas استفاده نمایید. در صورتی که با OpenGL تجربه قبلی ندارید بهتر است از به کارگیری آن خودداری کنید یا اینکه یک کتابخانه که توابع OpenGL را به صورت Abstraction در اختیار شما قرار می‌دهد استفاده کنید.

در شکل بالا فلش‌های قرمز نشان دهنده آزادی راکت در محور X صفحه دستگاه است. برای اینکه تغییر مکان راکت در هر فریم از را حساب کنیم، می‌توانیم از معادلات سینماتیکی حاکم بر این راکت بهره بگیریم که به شکل ساده به صورت زیر هستند:

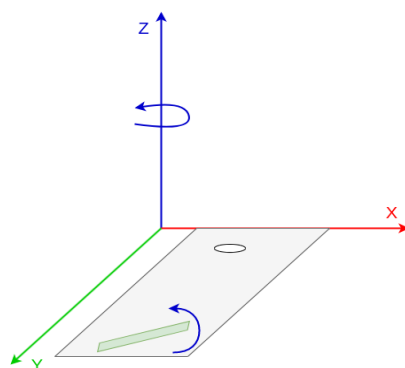
$$\Delta x = \frac{1}{2} a \Delta t^2$$

که در این معادله Δt نشان دهنده زمان بین هر دو فریم است. برای مثال اگر از ۳۰ فریم بر ثانیه استفاده کنیم (یعنی نرخ آپدیت هرگونه تغییر در صفحه دستگاه ۳۰ فریم در ثانیه باشد)، این مقدار حدود $\frac{1}{30}$ ثانیه می‌باشد. شتاب a توسط شتاب‌سنج به دست می‌آید^۲. نکته مهم این است که خروجی سنسور شتاب سنج بر حسب متر بر مجذور ثانیه است، در حالی که تغییر موقعیت راکت در صفحه نمایش بر حسب پیکسل انجام می‌گیرد. به عبارت دیگر، اگر تعداد پیکسل عرض صفحه نمایش دستگاه ۱۰۰۰ است و دستگاه ۲۵ سانتی‌متر به سمت راست محور X حرکت کند، آن گاه راکت باید ۵۰۰ پیکسل به سمت راست حرکت نماید. بنابراین، برای این نگاشت می‌بایست که جابجایی پیکسلی راکت را بر حسب شتاب متر بر مجذور ثانیه دستگاه بدست آورید. در صورتی که از canvas برای کشیدن اشکال بر روی صفحه استفاده می‌کنید، می‌توان از دو دستور زیر برای بدست آوردن طول و عرض صفحه دستگاه بر اساس پیکسل استفاده کرد.

`canvas.getWidth()`

`canvas.getHeight()`

مورد بعدی که باید به آن توجه کنیم؛ تغییر زاویه راکت در صفحه نسبت به زاویه قرارگیری دستگاه در محور Z ژيروسکوپ است. به شکل زیر توجه کنید.



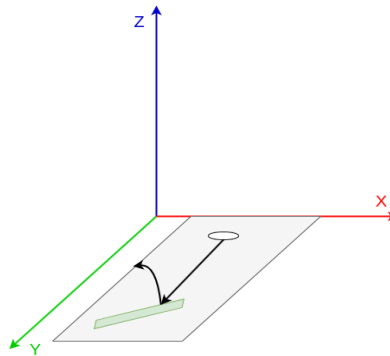
شکل ۵- چرخش راکت حول محور Z در مختصات سه‌بعدی

^۲ ممکن است برای کاهش نویز سمپل‌برداری به این نیاز داشته باشید که بر روی داده‌ها یک فیلتر اعمال کنید که می‌توانید به این [لینک](#) رجوع کنید و از دقیقه ۲۳ به بعد را مشاهده کنید. همچنین توضیحات تکمیلی در این [لینک](#) نیز برای پیاده‌سازی این فیلتر موجود می‌باشد.

در این شکل می‌بینیم که دستگاه، در محور Z ژيروسکوپ (که عمود به صفحه می‌باشد) دوران داشته است. این دوران با فلش آبی مشخص شده است. ژيروسکوپ سرعت زاویه‌ای را به صورت رادیان بر ثانیه به عنوان خروجی می‌دهد. با نمونه برداری از سنسور ژيروسکوپ در محور Z می‌توان پی برد که بین دو نقطه در زمان چقدر سرعت زاویه‌ای داشته‌ایم (معادل تغییر سرعت زاویه‌ای). به این صورت که با ضرب کردن سرعت زاویه‌ای در زمان، مقدار زاویه طی شده توسط دستگاه را محاسبه می‌شود و به این ترتیب متوجه می‌شویم که چقدر باید راکت را در صفحه بازی دوران دهیم. توجه کنید که مقدار زاویه دورانی که روی راکت اعمال می‌گردد، باید به صورت دقیق حساب شود. برای انجام این کار، نیاز است که نمونه برداری را به صورت صحیح انجام دهید. در صورت نیاز به کاهش نویز سمپل‌برداری به پاورقی شماره ۲ مراجعه کنید.

۳-۲ کنترل توپ

در ادامه باید توپ را در فضای بازی نیز شبیه سازی کنیم.



شکل ۶- سقوط آزاد توپ و برخورد آن به راکت در مختصات سه‌بعدی

در ابتدای بازی توپ از بالای صفحه نمایش شروع به سقوط آزاد می‌کند و در برخورد با راکت به سمت بالا پرتاب می‌شود. جهت بازگشت توپ وابسته به زاویه راکت است.

معادلات حاکم بر توپ مانند معادلات حاکم بر راکت است. برای محاسبه برخورد توپ بهتر است که علاوه بر مکان توپ در هر فریم، سرعت آن در هر فریم را نیز محاسبه کنیم. بدین منظور، سرعت توپ را به شکل یک بردار در نظر بگیرید. بردار سرعت توپ دو مؤلفه V_x (مؤلفه سرعت در جهت محور x) و V_y (مؤلفه سرعت در جهت محور y) دارد. در ابتدای بازی هر دو مؤلفه سرعت توپ صفر می‌باشند. در ابتدای شروع بازی، به این علت که توپ در حال سقوط آزاد است، فقط مؤلفه V_y آن تغییر می‌کند و مؤلفه V_x است صفر می‌ماند.

تغییرات سرعت توپ در جهت محور y (به خاطر وجود شتاب گرانش) در هر فریم را می‌توان با استفاده از فرمول زیر بدست آورد:

$$\Delta V_y = g \Delta t$$

در فرمول بالا شتاب گرانش (g) را به شکلی محاسبه کنید که بعد از گذشت یک ثانیه توپ یک سوم طول صفحه نمایش را بر اساس پیکسل را طی کند. این مقدار در هر فریم باید به سرعت در جهت محور y اضافه شود، زیرا توپ همیشه در حال سقوط می‌باشد.

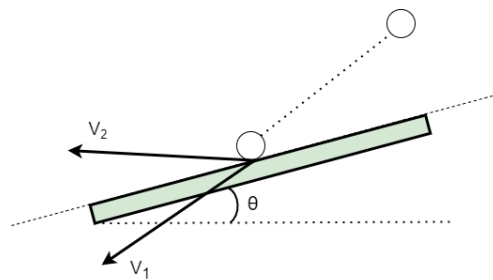
با استفاده از سرعت در هر فریم می‌توان تغییرات مکان توپ در فریم بعد را حساب کرد:

$$\Delta y = V_y \Delta t$$

$$\Delta x = V_x \Delta t$$

در ادامه، به علامت شتاب و سرعت توجه کنید زیرا جهت‌ها توسط علامت منفی یا مثبت مشخص می‌گردند.

مطابق صورت تمرین، قرار گیری دستگاه نسبت به محور Z برای ما مشخص می‌کند که موقعیت زاویه راکت چگونه بوده و در نتیجه توپ با چه زاویه‌ای بر می‌گردد. در هنگام برخورد توپ به راکت (یا به طور مشابه به دیوار) بردار سرعت باید همانند پرتو نور که از روی آینه بازتاب می‌یابد، بازتاب یابد. به شکل زیر توجه کنید.



شکل ۷- برخورد توپ با راکت (و به طور مشابه با دیوار)

در شکل بالا توپ به راکت برخورد کرده است. در هنگام برخورد توپ دارای بردار سرعت V_1 است. در لحظه بعد از برخورد بردار سرعت، تبدیل به بردار V_2 می‌شود. برای محاسبه بردار V_2 داریم:

$$V_{x2} = V_{x1} \cos 2\theta + V_{y1} \sin 2\theta$$

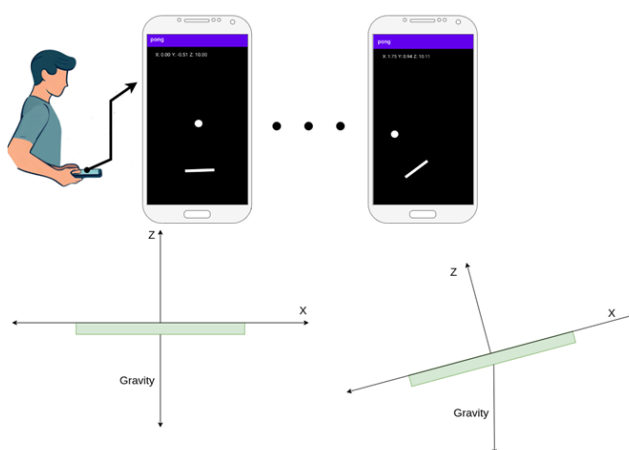
$$V_{y2} = -V_{x1} \sin 2\theta - V_{y1} \cos 2\theta$$

در معادله بالا اندیس ۱ به معنای سرعت توپ قبل از برخورد است و اندیس ۲ به معنای سرعت توپ بعد از برخورد است. بعد از انجام برخورد، دو مولفه x و y سرعت توسط فرمول‌های بالا عوض می‌شوند.

توجه کنید که برای کنترل توپ محدودیتی در روش استفاده از معادلات فیزیکی بالا وجود ندارد. می‌توانید از فرم دیگری از معادلات سینماتیکی استفاده کرده و یا معادلات تکانه را به دقت پیاده‌سازی نمایید.

برای تحویل این تمرین، کد سورس برنامه و یک خروجی apk تحویل خواهید داد که در آن بازی با توضیحات بالا ساخته شده باشد.

نمره امتیازی برای پیاده سازی: در این حالت دستگاه روی میز قرار ندارد و در دست های کاربر موازی با سطح زمین گرفته شده است. برای حرکت دادن راکت در محور X صفحه نمایش از شتاب گرانش استفاده نماییم. برای این کار می توانیم از پخش شدن بردار شتاب گرانش در سه محور X و Y و Z سنسور شتاب سنج استفاده کنید و شتاب گرانشی را که در راستای محور X شتاب سنج است، به راکت برای تغییر موقعیت در محور X صفحه نمایش وارد نماییم. همچنین، هنگامی که توپ به راکت برخورد می کند، در صورتی که ما هنگام برخورد توپ به راکت در جهت Z محیط یک شتاب ناگهانی به سمت بالا بدهیم، توپ مسافت بیشتری را در صفحه طی خواهد کرد. این قید راکت پینگ پونگ واقعی را شبیه سازی می کند که در آن نیروی وارد شده به توپ هنگام برخورد، بر سرعت توپ اثر خواهد گذاشت. چرخش راکت در این حالت مانند حالت اجباری و با کمک سنسورژیروسکوپ خواهد بود.



شکل ۸- نحوه در دست نگه داشتن دستگاه و اعمال شتاب گرانشی در پیاده سازی امتیازی

۴- ابزار Profile and Trace

ابزاری که شرکت گوگل برای انجام عمل Profiling و Tracing در اختیار برنامه نویسان قرار داده است، [Perfetto](#) نام دارد. این ابزار به ما کمک می کند تا بتوانیم از برنامه ای که نوشتیم، گزارش تهیه کنیم. این ابزار با نمونه برداری از سیستم و نشان دادن این داده ها به صورت گرافیکی به ما کمک می کند که عملکرد برنامه را تجزیه و تحلیل کنیم. برای مثال، در صورتی که بخواهیم تعداد Context Switch را در زمان بند بدانیم، باید به پارامترهایی که زمان بند از آن ها نگهداری می کند دسترسی پیدا کنیم. ابزارهای بر پایه trace به این پارامترها دسترسی پیدا می کنند و اطلاعات مورد نیاز را برای ما فراهم می نمایند.



شکل ۹- تصویری از محیط ابزار Perfetto

برای آشنایی بیشتر با مفاهیم این ابزار می‌توانید از [راهنمای ۱](#) و [راهنمای ۲](#) استفاده کنید. برای مثال، طبق دستورالعمل این ابزار برای اندروید با ورژن پایین تر از ۹ به شکل زیر عمل می‌کنیم:

```
curl -O
https://raw.githubusercontent.com/google/perfetto/master/tools/record_android_trace
chmod +x record_android_trace

# See ./record_android_trace --help for more
./record_android_trace -o trace_file.perfetto-trace -t 10s -b 32mb \
sched freq idle am wm gfx view binder_driver hal dalvik camera input res memory

# Or use the configuration https://perfetto.dev/docs/concepts/config
./record_android_trace -c config.pbtx -o trace_file.perfetto-trace
```

این دستور به ما کمک می‌کند که یک Trace از سیستم را دریافت کرده و در سیستم ذخیره کنیم.

برای پاسخ کامل به سوالات بخش بعدی، نیاز به تنظیم ابزار Perfetto مطابق با راهنمای بخش **Config** است. همچنین، برای استفاده از ابزارهای Profiling برای Heap و Call Stack نیاز است که اندروید شما از نسخه ۱۱ جدیدتر باشد.

۵- سوالات

- (۱) با استفاده از ابزار Perfetto از دستگاه داده جمع آوری کنید و عملیات Trace و Profile را انجام دهید. طبق راهنمای این ابزار و بعد از جمع آوری داده به سوالات زیر پاسخ دهید.
- از وقتی که درخواست خواندن داده به یک سنسور داده شده تا گرفتن داده چه اتفاقاتی در سطح سیستم عامل افتاده است؟ توضیح خود را با خروجی Perfetto توضیح داده و توجیه کنید.
 - آیا در فراخوانی های سیستمی، تعارضی (انتظار مشغول یک Thread تا Thread دیگر کار خود را تمام کند) بین استفاده از کتابخانه مربوط به گرافیک و بروزرسانی سنسورها وجود دارد؟ پاسخ خود را چگونه توجیه می کنید؟
 - بیشترین بار پردازش اشغال شده برای CPU، مربوط به سنسورها است یا پردازش های گرافیکی؟
 - از زمان فراخوانی دستور مربوط، چه مدت زمانی طول می کشد تا تغییرات مکانی و زاویه ای راکت (بر اساس مقادیری که از سنسورها خوانده شده است) روی صفحه نمایش ظاهر شود؟ (تصویر واضح از Perfetto قرار داده شود)
- (۲) بهترین دوره تناوب برای خواندن مقادیر سنسور شتاب سنج و ژيروسکوپ چه مقدار است؟ با استدلال توجیه شود.
- (۳) اگر از Android NDK بجای Android SDK استفاده می شود، تمرین شما چه مزایا و معایبی خواهد داشت؟
- (۴) در مورد سنسورهای hardware-based و software-based تحقیق نمایید و هر یک را تشریح نمایید. هر کدام از سنسورهای مورد استفاده در این تمرین در کدام دسته قرار می گیرند؟
- (۵) چه تفاوتی بین تعریف سنسور به صورت wake-up و non-wake-up وجود دارد؟ ضمن تشریح مزایا و معایب هر کدام، مشخص کنید که انجام این کار تاثیری بر نحوه دریافت بروزرسانی سنسورها و نتیجه کنترل راکت دارد؟

۶- نکات مهم

- برای پیاده سازی این تمرین تنها می توانید از زبانهای Kotlin ، Java و C++ استفاده نمایید.
- گزارش کار باید کامل و شامل موارد زیر باشد:
 - نحوه شکست کار بین اعضا
 - مفروضات
 - طراحی مفهومی و ساختار برنامه اندرویدی
 - مشخصات سکوی نرم افزاری و سخت افزاری در پیاده سازی
 - ابزارهای مربوط و کتابخانه های مورد استفاده

- تنظیمات آزمایش‌ها و سناریوهای تست
 - تصاویر مربوط به خروجی برنامه و تصاویر Perfetto
 - گزارش نتایج و پاسخ به سوالات مطرح شده
 - مراجع
- علاوه بر کدها، فایل apk مربوط به پیاده‌سازی را نیز آپلود کنید. دقت کنید که فایل apk شما باید سازگار با اندرویدهای ۶ به بالا باشد.
 - کد برنامه شما باید روی گوشی واقعی تست شده باشد (شبیه‌ساز اندروید کافی نیست).
 - برای نصب Android Studio و مطالعه منابع به V_P_N نیاز خواهید داشت. همچنین می‌توانید از ابزار هایی مانند [شکن](#) و ابزار [۴۰۳](#) استفاده کنید.
 - هرگونه شباهت در کدها و گزارش‌ها به عنوان تقلب در نظر گرفته خواهد شد.
 - تسلط به تمام بخش‌های تمرین از راه اندازی ابزارها تا کدهای نوشته شده الزامی است و در تحویل، از بخش‌های مختلف سوال پرسیده خواهد شد.
 - در صورتیکه هیچ تجربه‌ای در نوشته برنامه اندروید ندارید، از [این لینک](#) استفاده کنید.
 - برای آشنایی با محیط Android Studio می‌توانید [این صفحه](#) را ببینید.

موفق باشید