

بِسْمِ اللَّهِ

# Real-Time Embedded Systems

## Assignment III

Kiarash Azarnia  
Mohammad Mahdi Islami  
Mohammad Ali Adelimanesh

## توضیحات کلی

برای تعیین مکان جدید گوی ها ابتدا پس از به دست آوردن داده ها از سنسور ها باید شتاب و سرعت و در نهایت به مکان جدید گوی برسیم با استفاده از فرمول های سینماتیک داریم:

$$F = ma$$

$$V = at + V_0$$

$$X = 0.5at^2 + V_0t + X_0$$

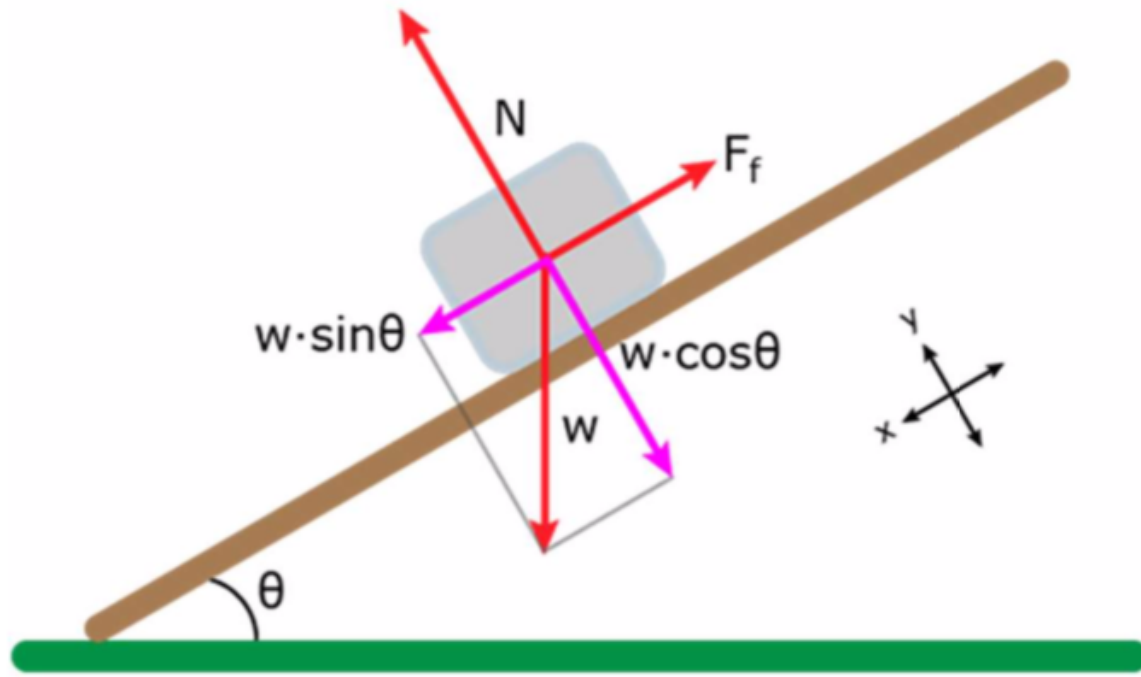
و فرمول سطح شیبدار برای محاسبه ی شتاب:

$$ma = mgsina - F_{fk} = mgsina - \underline{N \cdot \mu_k} = mgsina - mg\cos a \mu_k$$

در مواقع برخورد نیز طبق اصل برخورد کشسانی دو جسم سرعت ها تعیین میشود:

$$V_1 = u_1 + 2u_2(m_1 - m_2) / (m_1 + m_2)$$

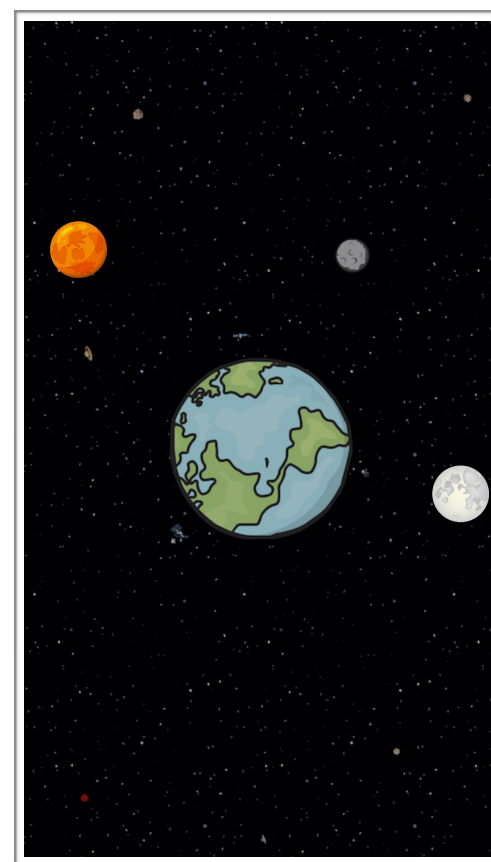
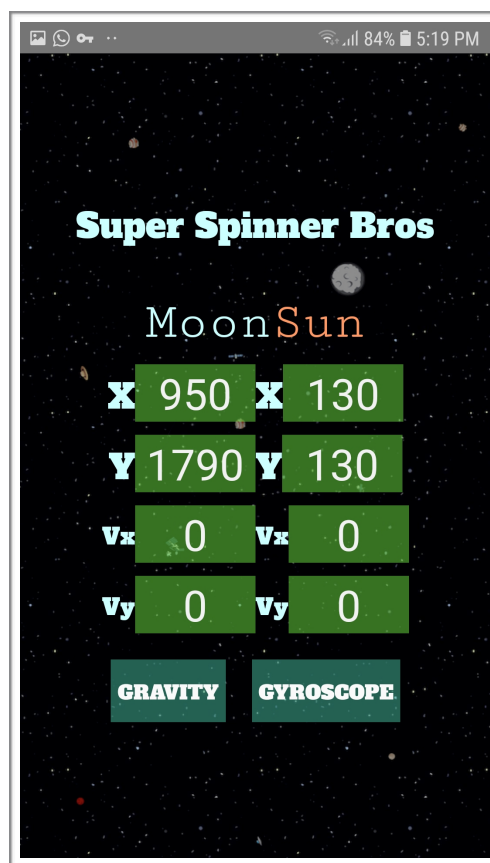
$$V_2 = u_2 + 2u_1(m_1 - m_2) / (m_1 + m_2)$$



شكل 1

$$ma = mgsina - F_{fk} = mgsina - \underline{N.u_k} = mgsina - mg\cos a u_k$$

نمایی از برنامه نوشته شده توسط گروه



1d110cb905dfbbdbff8e5546849883a8662ebdf8

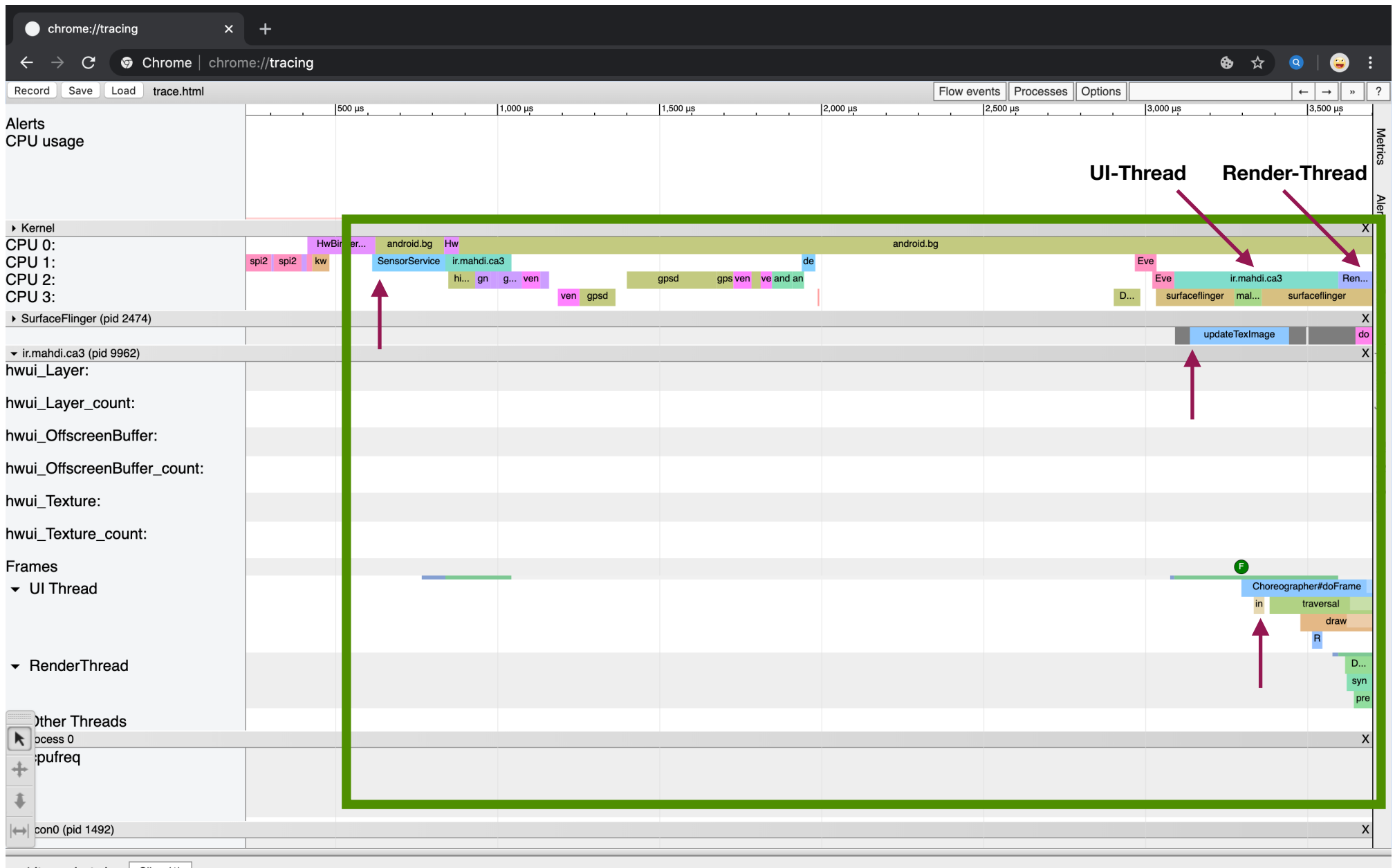
هش آخرین کامیت پروژه در گیت لب:

# Questions

## سوال اول:

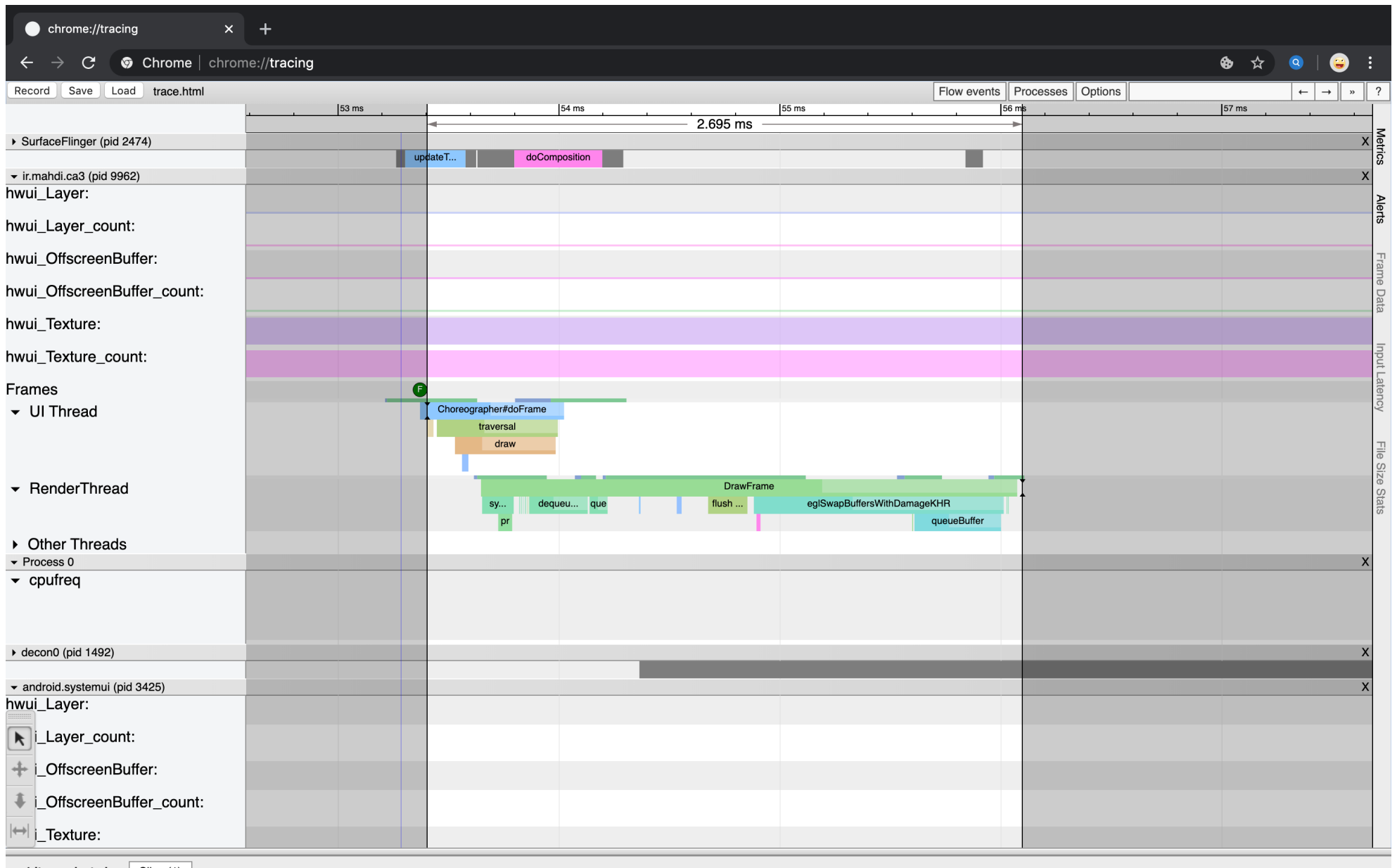
همان گونه که انتظار می رود پس از call کردن Sensor Service مقدار سنسور باید به عنوان ورودی به برنامه داده شود و سپس پس از پردازش هایی که برای تعیین شتاب و سرعت و در نهایت مکان گوی ها انجام می شود مورد استفاده قرار گیرد. سپس این مقدار با استفاده از متدهایی مانند setX() و setY() برای ایجاد تغییر در مکان گوی مورد استفاده قرار بگیرد و در نهایت cpu با استفاده از ابزارهای گرافیکی این تغییر را در قالب جابه جایی محل نمایش گوی اعمال می کند. در این حین از بین دو ریشه ی اصلی UI-Thread و Render-Thread جابه جا می شود.

به همین دلیل در خروجی systrace نیز می بینیم که پس از آنکه Sensor Service صدا زده می شود مقدار عملیات های CPU صورت می گیرد. و سپس به خروجی به عنوان ورودی به UI-Thread داده می شود.



## سوال دوم:

اگر درست تشخیص داده باشیم از زمان ورودی UI-Thread تا زمانی که Render-Thread به طور کامل عملیات ترسیم جدید را انجام دهد در حدود ۲.۶۹۵ میلی ثانیه به طول می انجامد.  
در کد اصلی ما برنامه هر ۱ میلی ثانیه یکبار صفحه آپدیت می شود.





## سوال سوم:

برای این منظور باید در نظر گرفت مدت زمانی که طول می کشد تا صفحه ی نمایش آپدیت شود چه مقدار است. اگر پریود خواندن دیتا را از سنسور کمتر از این مقدار قرار بدیم در واقع در حال خواندن داده های زیادی از محیط هستیم که به خیلی از آن ها احتیاجی نیست و در واقع به نوعی داریم منابع و داده ها و هزینه ها را هدر می دهیم.

برای این کار بهتر است تا بسته به منابعی که در اختیار داریم مانند سرعت CPU به منظور محاسبات و یا GPU به منظور میزان زمانی که آپدیت کردن صفحه ی نمایش به طول می انجامد این مقدار را تعیین کنیم تا به هنگام داده ها از سنسور خوانده شود طوری که نه داده ای را miss کنیم و نه داده ی زیادی به دست بیاوریم. بر فرض مثال اگر هر ۱ میکروثانیه داده ی سنسور را دریافت کنیم داده های به فاصله ی ۱ میکروثانیه اطلاعات به درد بخور جدیدی را در اختیار ما نمی گذارند. پس بهتر است با توجه به ویژگی های منابع و میزان زمان آپدیت صفحه این مقدار را تعیین کنیم.

## سوال چهارم:

### NDK:

- برای عملیات‌های شدید CPU این platform بهتر عمل میکند مانند VideoGame ها و پردازش‌های سیگنال.
- می‌توانستیم از زبان C و C++ استفاده کنیم.
- می‌توانستیم همزمان برای ios و Windows نیز اپلیکیشن را توسعه دهیم.

### SDK:

- به مجموعه ای غنی از کتابخانه ها دسترسی پیدا کردیم.
- مدیریت حافظه به صورت خودکار انجام می‌شود.
- portability دستگاه را با وجود معماری های مختلف پردازنده ها تضمین می‌کند.

## سوال پنجم:

### Hardware-Base Sensor:

سنسور سخت‌افزاری قطعه‌ی فیزیکی است که درون دستگاه موبایل قرار می‌گیرد و داده‌ی خود را از طریق اندازه‌گیری مستقیم ویژگی‌های مختلف محیط پیرامون به دست می‌آورد.

● مانند سنسور Gyroscope

### Software-Base Sensor:

سنسور نرم‌افزاری به نوعی کار سنسور سخت‌افزاری را شبیه‌سازی میکند و داده‌ی خود را از طریق شبیه‌سازی یک یا ترکیب داده‌های چند سنسور سخت‌افزاری بدست می‌آورد.

● مانند سنسور Gravity

## سوال ششم:

### Gravity:

سنسور گراویتی سنسور نرم‌افزاری است و مقدار نیروی جاذبه با واحد  $m/s^2$  در راستای محورهای XYZ را محاسبه می‌کند و در واقع  $gsina$  را به ما می‌دهد و تنها چیزی که باید محاسبه شود مقدار  $cosa$  است که از طریق فرمول  $\sin^2 a + \cos^2 a = 1$  محاسبه می‌شود. و در واقع در این قسمت محاسبه ی خاصی نیاز نیست.

### Gyroscope:

سنسور ژيروسکوپ سنسور سخت‌افزاری است و مقدار سرعت زاویه ای را با واحد  $rad/s$  حول محورهای XYZ محاسبه می‌کند که در اینجا برای قابل استفاده شدن این داده برای تعیین شتاب گوی باید در فریم‌های زمانی مشخص مقدار تغییر زاویه را بسنجیم و سپس با محاسبه سینوس و کوسینوس این زاویه مقدار شتاب و اصطکاک وارد بر گوی‌ها را محاسبه کنیم.

سپس با شبیه‌سازی زمان مقدار شتاب و سرعت و در نهایت مکان گوی‌ها را محاسبه می‌کنیم.  
فرمول اصلی محاسبه ی شتاب به این گونه است:

$$F = ma = mgsina - mgcosa.u_k$$

## سوال هفتم:

بله تفاوت وجود دارد. به دلیل سخت‌افزاری بودن سنسور ژيروسکوپ باید ابتدا در شرایط خاصی قرار بگیرد و آن این است که باید ابتدا در شیب برابر صفر قرار بگیرد و **stable** باشد و این جزو ویژگی‌های سنسور ژيروسکوپ است و الا در تشخیص جهات دچار اختلال می‌شود. اما در سنسور گراویتی تا حدی به دلیل نرم‌افزاری بودن لزومی ندارد ابتدا در شرایط خاصی قرار بگیرد و در شرایط شروع بر روی شیب نیز می‌تواند به درستی تشخیص بدهد.