گزارش کار پروژه دوم

آرین قره‌داغی ۸۱۰۱۹۷۵۵۷

امیرمهدی محمدیان ۸۱۰۱۹۷۶۵۱

محمد ملکی تبار ۸۱۰۱۹۷۵۹۲

علیرضا ابراهیمی ۸۱۰۱۹۷۴۴۱

معرفی

در این پروژه باید یک بازی راکت و توپ درست کنیم که از سنسورهای یک گوشی اندرویدی استفاده می‌کند برای جا به جایی و حرکت راکت. برای این کار ابتدا نیاز است که android studio را نصب کرده و android sdk را نصب کنیم. حال شروع به کد نویسی می‌کنیم.

نحوه پیاده سازی

ابتدا فایل های پروژه خود را ایجاد می‌کنیم. این فایل ها عبارت اند از:

- MainActivity

  در این فایل ما بازی را ایجاد کرده و دسترسی های مورد نیاز(سنسورها) را از کاربر درخواست می‌کنیم. همچنین یک سری فلگ ست می‌کنیم.

- MainThread

  در این فایل ما loop بازی خود را initialize می‌کنیم. با هربار اجرای این حلقه بازی را update می‌کنیم.

- GameView

  در این فایل ما سنسورهای مورد نیازمون رو تعریف می‌کنیم که یک سنسور accelerometer و یک سنسور خطی از همان نوع می‌باشد. همچنین زمین بازی را ایجاد می‌کنیم و از توپ و راکت یک اینستنس جدید ایجاد می‌کنیم. از سنسور ژیروسکوپ نیز می‌توان استفاده کرد.

- Ball

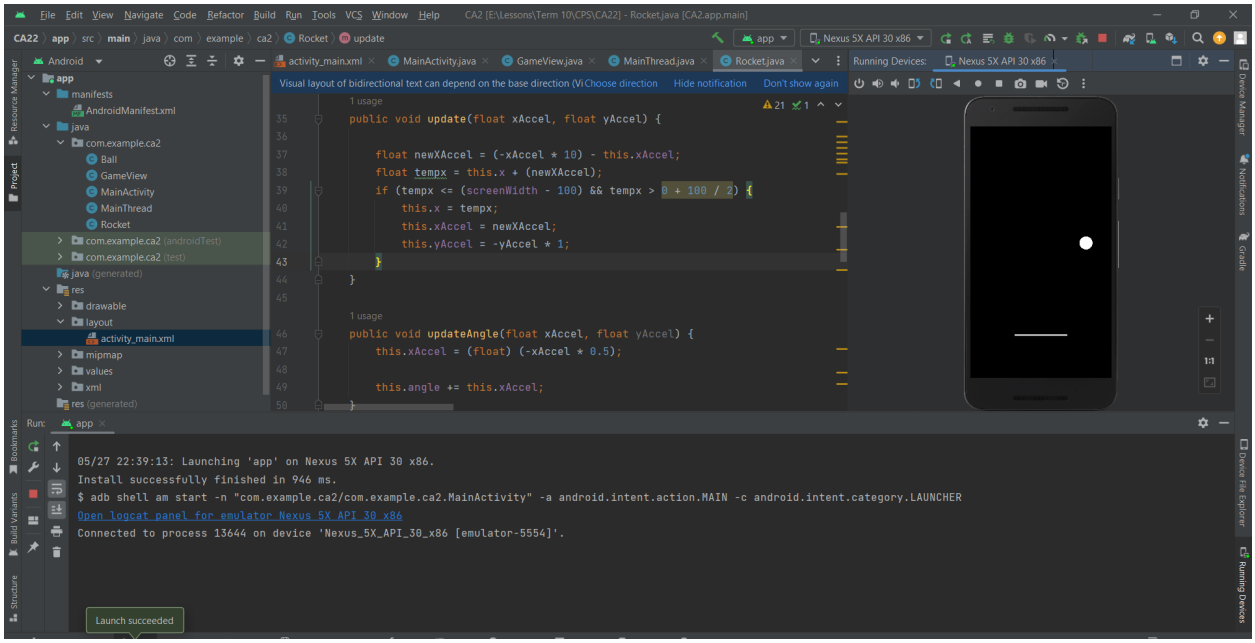
  در این فایل ما توپ و منطق مربوط به آن را ایجاد می‌کنیم.

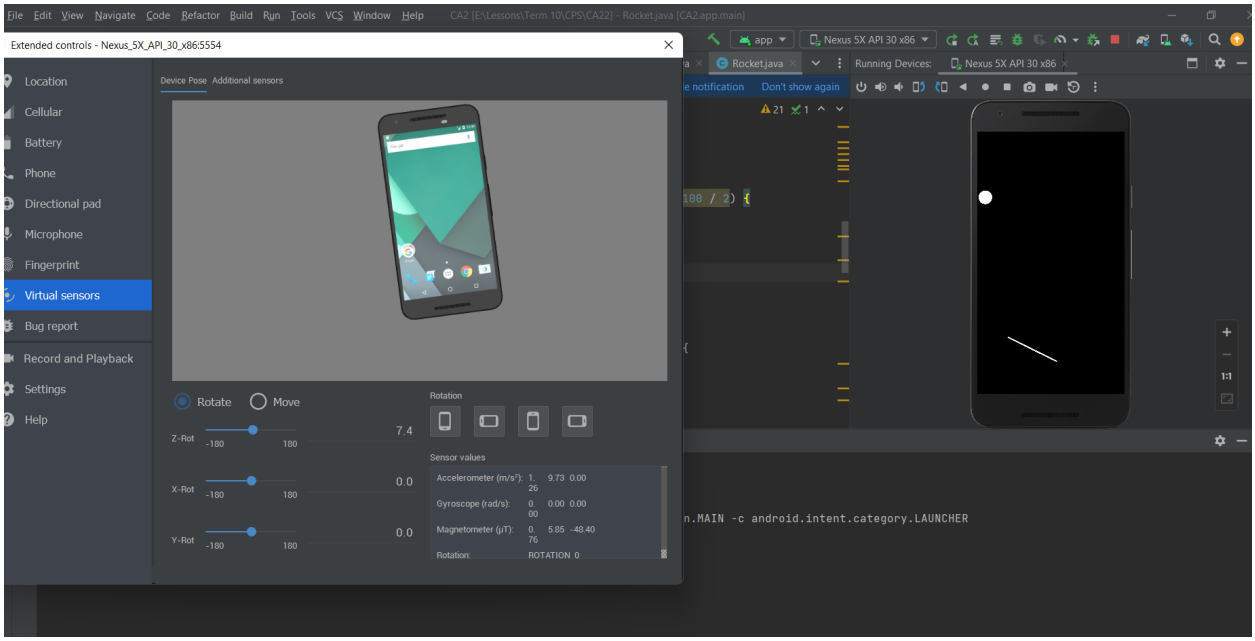
- Rocket

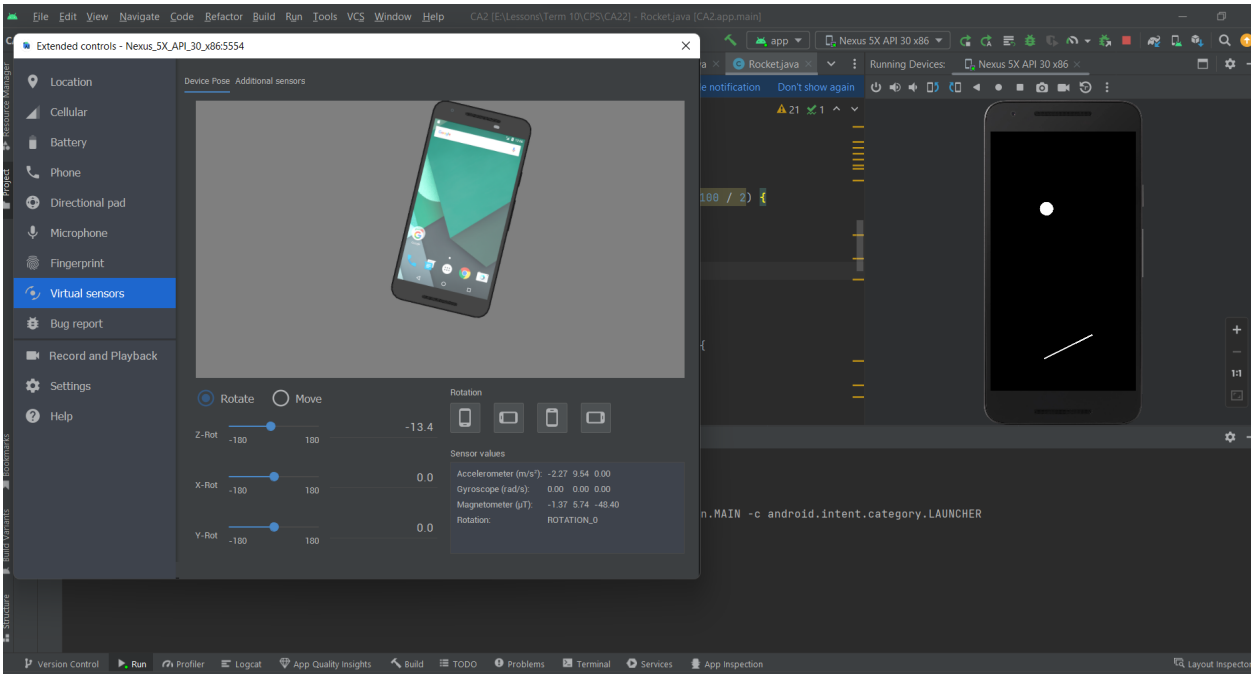
  در این فایل ما راکت و منطق مربوط به آن را ایجاد می‌کنیم.

سپس باید منطق مربوط به حرکت توپ و برخورد آن با راکت را با توجه به فرمول های داده شده ایجاد کنیم. برای آپدیت و کشیدن صفحه بازی از canvas استفاده می‌کنیم. برای خواندن داده های سنسور ها نیز از sensor manager استفاده می‌کنیم.

خروجی برنامه







سوالات

2) پارامتری که ما برای سمپلینگ داریم sensor_delay_game است که یک سمپلینگ استاندارد تعریف شده توسط خود اندروید برای بازی ها می‌باشد که 20ms است. پارامترهایی دیگری نیز دارد مثل fastest, ui, normal. انتخاب بهترین سمپلینگ ریت بسته به تایم update بازی ما برای هر فریم می‌باشد باید سعی کنیم که سمپلینگ ما نزدیک زمان آپدیت صفحه بازی باشد تا بهترین نتیجه را بگیریم. اگر خیلی کم تر از تایم آپدیت صفحه بازی باشد عملا مقدار زیادی سمپلینگ ما نادیده گرفته می‌شود و مصرف انرژی بالایی خواهیم داشت برای خواندن سنسورها. برعکس اگر تایم سمپلینگ ما بیشتر از تایم آپدیت صفحه بازی باشد زمانی که صفحه آپدیت می‌شود دیتای جدیدی از سنسور ها خوانده نشده برای همین در بازی مشکل ایجاد می‌شود و حرکت راکت دیر تر از آپدیت شدن صفحه انجام می‌شود.

3) یکی از مزایا این است که پرفورمنس و اپتیمیزیشن بهتری خواهیم داشت ولی کد سخت تر و پیچیده تری خواهیم داشت چون با c/c++ باید کد را می‌نوشتیم و نیاز به memory management داشتیم.

4) سنسورهای هاردوری سنسورهایی هستند که به شکل سخت افزار داخل گوشی وجود دارند ولی سنسورهای سافتوری سنسورهایی هستند که بر پایه الگوریتم ها و داده های جمع شده توسط سنسورهای سخت افزاری می‌باشد. سنسورهای استفاده شده در این پروژه hardware-based هستند.

5) فرق این سنسورها در این است که نوع wake-up از نوع prompt می‌باشد و حتی اگر دستگاه روی اسلیپ باشد یا low-power نیز روشن باشد داده را به صورت real time خوانده و به دستگاه می‌دهند. وقتی

یک event رخ می‌دهد و سنسور آن را pick up میکند این قابلیت را دارد که پروسسور گوشی را روشن کند و داده را به اپلیکیشن مورد نیاز می‌دهد. سنسور های non-wake-up این قابلیت را ندارند و در صورت اسلیپ بودن دستگاه داده ای جمع آوری نمی‌کنند. دلیل انتخاب بین این دو سنسور مدیریت power می‌باشد و همچنین نیاز اپلیکیشن ما. استفاده از سنسور های نان ویک آپ ممکن است در حرکت راکت اختلال ایجاد کند اگر که دستگاه روی حالت save power باشد و همچنین اگر صفحه خاموش بشود دیگر داده ای از سمت سنسورها خوانده نخواهد شد و راکت حرکت نخواهد کرد.