**Assignment 3 Dice Roller**

COMP2430-Mobile Computing Technologies

Department of Computer Science

Winter 2025

**Group 12**

**Group Members:-**

1. Sanskriti Dabhade (1225131)
2. Sweety Das (1238456)
3. Kirtan Sukhadiya (1222983)
4. Julia Petterle (1185004)

**Introduction**

This report presents our group’s submission for Assignment 3 in COMP2430. The assignment required us to create an interactive Dice Roller app in Kotlin, which utilizes sensory data to determine the dice value. The objective was to familiarize ourselves with Button and callback functions in Kotlin, along with integrating Android sensors. We extended our Assignment 2 app to incorporate the functionality of using the light sensor data to dynamically update the dice value. The dice value is calculated as one plus the integer part of the current light level modulo six. This report details the implementation, challenges faced, and results of the application.

**Objectives**

The primary objective of this assignment was to develop an interactive Dice Roller app using Kotlin and Jetpack Compose. The app needed to demonstrate proficiency in using Android sensors and callback functions while extending our previous assignment’s codebase. Specific objectives included:

1. Implementing a dice roller app using sensory data.
2. Utilizing the light sensor to determine dice values dynamically.
3. Integrating button and callback functionalities for user interaction.
4. Ensuring that the app runs correctly on both an emulator and a physical device.

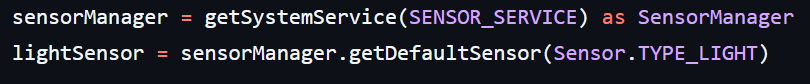
**Implementation**

The implementation involved reusing the sensor management code from Assignment 2 while modifying it to support the Dice Roller functionality. The app listens to changes in the ambient light level using the light sensor and calculates the dice value as (1 + light level % 6). The app displays the corresponding dice image based on the calculated value. A button is also provided to roll the dice manually. The button functionality was updated to correctly update the dice result when clicked by associating a proper callback function to change the state.

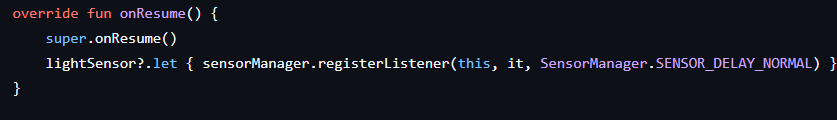
**Code Explanation**

1. Sensor Initialization and Management:

* We initialize the SensorManager and light sensor in the MainActivity:

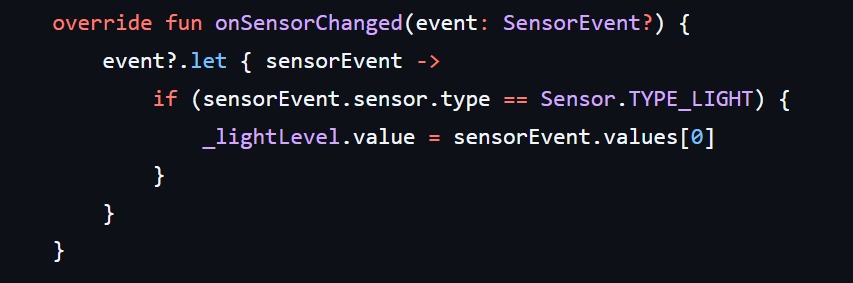


* The sensor is registered in the onResume() method to receive updates:



1. Light Sensor Data Handling:

* The onSensorChanged() method captures light sensor changes and updates the light level:



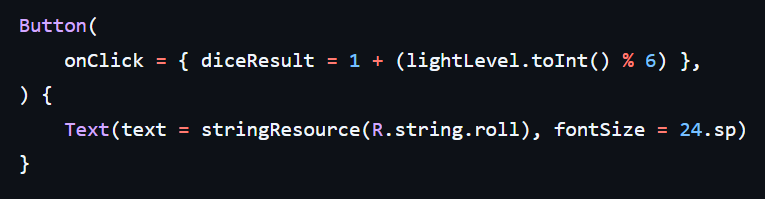
1. Dice Roll Logic:

* The dice result is calculated based on the light level using the following line:



1. Button Functionality:

* The button triggers a manual dice roll by updating the dice result:



**Challenges and Resolutions**

1. Sensor Data Handling:  
   * Handling varying light levels was challenging. We used the integer part of the light level to ensure consistent dice values.
2. Button and UI Updates:  
   * Initially, the button did not change the dice result. The issue was resolved by properly linking the button’s onClick event to update the dice value state.

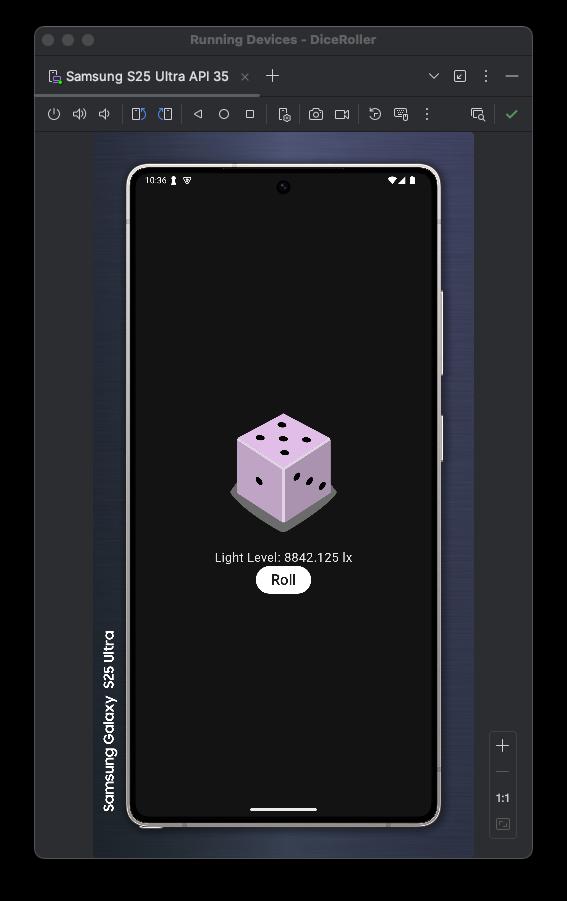
**Conclusion**

This assignment enhanced our understanding of integrating Android sensors with UI components using Kotlin. The Dice Roller app successfully demonstrates interactive functionality using sensory data. We tested the app on both an emulator and a physical device to verify its correctness.

**Running the Application**

1. Open the project in Android Studio.
2. Connect an Android device or start an emulator.
3. Run the app and observe the dice change as the light level varies.
4. Use the button to manually roll the dice.

**Application Output (Screenshot)**



*Screen Recording Link:* [MCTAssignment3VIDEO-OutputRecording.mp4](https://drive.google.com/file/d/1uJ4Mee9fX73nDZNFol3WvmDXcWw1MDdr/view?usp=sharing)

*Presentation Video Link:* [MCTAssignment3VIDEO.mov](https://drive.google.com/file/d/1qwk1WxdcFeOpC5-qdApeUbaZGaS71wx_/view?usp=sharing)