Assignment 3:
Snakey Snake

Must work in pairs of 2.

Don't use ChatGPT or AI.
Use it for looking up documentation; don't use it to code for you.
I say this not because it hurts me, but because it'll hurt you as a developer.
Job market is exceedingly difficult, so work hard so you can stand out.

Topics we will cover:
Git
Polymorphism
Interfaces

The Snake Game can be found here:
https://github.com/PacktPublishing/Learning-Java-by-Building-Android-Games-Second-Edition

Download and compile Chapter 17.

Objective:

- Meet the requirements for the assignment (Git, Polymorphism, Interfaces)
- Add a pause button to the game; clicking on it again resumes the game
    - Look in Pakt book for help
- Change the look of the game
    - Change the font and game background to something unique
    - Add your names to the top right corner of the app

Requirement 1: Git

- Demonstrate your knowledge of Git by hosting your project on a Git repository and utilizing version control throughout the entire project. Ensure to include a printed log of your Git commits in your report.
- You and your teammate must use the same GitHub Repository.

Requirement 2: Polymorphism and Interfaces

- Apply the fundamental concepts of polymorphism and interfaces learned in class to elevate the structure of your Snake game.

- This part focuses on practical implementation of polymorphism and interfaces.

- **Polymorphism:**
  - Static Polymorphism**:**
    - Use method overloading for relevant actions in your Snake game, whether it's for a constructor or actions your game object can perform.
    - (e.g., snake movement, apple spawning, collision interaction).

  - Dynamic Polymorphism**:**
    - Create a base class or interface representing a game object (e.g., GameObject).
    - Implement derived classes (e.g., Snake, Apple) that use method overriding.
    - Show how dynamic polymorphism gives more flexibility when managing game objects.
  - Think of how you can represent the objects in polymorphic ways.
    - Maybe creating a parent class for all the game objects that contain common methods?
    - Maybe creating polymorphic methods in your class to allow for different functionality based on the input parameters.
- **Interfaces:**
  - **Interface Hierarchy:**
    - Identify key functionalities in your Snake game (e.g., movable, drawable) and create corresponding interfaces.
    - Implement these interfaces in relevant game components.
  - **Abstract Classes vs Interfaces:**
    - Choose specific functionalities to be represented by abstract classes and others by interfaces.
    - Justify your choices based on the advantages of abstract classes and interfaces in different situations.
  - **Multiple Inheritance via Interfaces:**
    - Explore scenarios in your Snake game where multiple inheritance can be achieved using interfaces.

- Implement and explain how this enhances the modularity and extensibility of your code.

Submission Details:

Write a one to two page report discussing the changes that you made and why you made those changes. Include as much detail as possible, however, do not just write filler, your assignment will be graded on quality, more so than quantity. If you are struggling to write a full page of high quality description, you might want to revisit the lectures.

Consider the following points in your report:

1. **OOP Implementation:**
   - Briefly outline the integration of OOP concepts (Classes & Objects, Abstraction, Encapsulation, Interfaces, Polymorphism) in the Snake game.
2. **Code Smells Identification:**
   - Articulate specific issues associated with each code smell and their impact on code quality.
3. **Refactoring Choices and Techniques:**
   - Explain the rationale behind refactoring choices to address identified code smells.
4. **Integration of OOP Concepts with Refactoring:**
   - Briefly describe how OOP concepts from Part 1 seamlessly enhance the refactoring process in Part 2.
   - Highlight how these concepts contribute to improved code maintainability and readability.
5. **Git Integration**:
   - Explain how using Git and version control enhanced your development process.
   - Discuss how you used Git as a pair of developers.
6. **Challenges and Solutions:**
   - Summarize challenges faced during implementation and refactoring.
   - Provide succinct insights into the strategies used to overcome these challenges.

Also:

- A zip file containing your entire Android Studio Project

- Provide a screenshot of the Snake game running in the emulator to demonstrate successful implementation.

- SUBMIT YOUR REPORT AS A PDF FILE WITH YOUR NAMES