Colleen Lemak

CSS 545

HW2-BasicStorage

(5 Points) Various approaches to storage management on your platform of choice

(5 Points) Pros AND cons of each approach for your project

When developing **Numbly**, several storage management approaches can be considered, each with distinct advantages and disadvantages.

Shared Preferences is ideal for storing simple user preferences, such as text and background colors. It's easy to implement and allows for quick access to saved settings, especially storing and loading user preferences. However, it is limited to simple key-value pairs and is not suitable for larger or more complex data structures, which could hinder scalability if the application expands.

Internal Storage provides a secure space for saving user-related files, such as game configurations. This ensures privacy, as files are accessible only by your app. However, a limitation is its restricted capacity, and the responsibility lies on the developer to manage the data lifecycle, which could become inconvenient as user-generated content grows.

External Storage allows for larger storage capacity, enabling users to save game assets or share scores easily. However, this approach poses security risks since data can be accessed by other applications, potentially compromising user privacy. Additionally, it requires management of permissions, which can complicate user experience and lead to data loss if users delete files inadvertently.

For structured data storage, **SQLite** enables efficient management of complex datasets through SQL queries. This is beneficial for tracking user scores and game history. However, the complexity of implementing SQLite requires managing schema and data migration, increasing development time and effort.

Another option is the **Room Persistence Library** which simplifies the use of SQLite by providing an abstraction layer, making data management more accessible in **Numbly**. Its integration with LiveData promotes a responsive UI, reflecting updates immediately. However, it introduces a learning curve and may add performance overhead due to its abstraction.