Chapter 12
Class Model for Mobile Game Architecture

Leon Cana
Rochester Institute of Technology, Kosovo

Art Jashari
Rochester Institute of Technology, Kosovo

Rafaela Marku
Rochester Institute of Technology, Kosovo

ABSTRACT

Class models are an essential part of mobile game architecture, providing a framework for organizing and structuring the code. These models help developers to understand the relationships and dependencies between different objects in the game, allowing for more efficient and effective code development. The use of class models is likely to continue to evolve and improve in the future, as mobile games become more complex and sophisticated. As AI and ML technologies become more advanced, developers will need to create new classes and models to represent these technologies in mobile games. Additionally, the use of cloud computing and other distributed technologies is likely to become more prevalent, requiring the creation of new classes and models to represent these technologies in the game. Overall, class models will play a critical role in enabling developers to create more engaging and immersive mobile gameplay experiences.

INTRODUCTION

A class model is a type of diagram that is used in object-oriented modeling and design. It represents the classes and their relationships in a system, and it is used to visualize the structure and behavior of the system. Class models are an important tool in software engineering, and they are used to help developers understand and design the architecture of a software system. Class models usually include the following elements:

Classes: Classes are the fundamental building blocks of a class model. They represent the objects or entities in the system, and they are defined by their attributes, operations, and relationships with other classes.

Attributes: Attributes are the characteristics or properties of a class, and they describe the data or information that is associated with the class. Examples of attributes include name, age, gender, address, and phone number.

Operations: Operations are the actions or behaviors that a class can perform, and they define the functionality of the class. Examples of operations include calculate, display, update, and delete.

Relationships: Relationships are the connections or associations between classes, and they describe how the classes interact with each other. Examples of relationships include inheritance, aggregation, composition, and association.

Class models are often depicted using UML (Unified Modeling Language) notation, which is a standardized visual language for representing software systems. UML class diagrams are used to represent the structure of a system, and they show the classes, attributes, operations, and relationships of the system. Here is an example of a UML class diagram that represents a simple system of bank managing system information:

In this class diagram, the Customer class is the main class, and it has attributes such as name, address, and phone number. The Customer class also has operations such as OpenAccount, CloseAccount, and RequestCard. The Customer class has a relationship with the Loan class, which represents the type of loans that the customer has made. The Loan class has attributes such as type and accountId. Class models are an important tool in software engineering, and they are used to help developers understand and design the architecture of a software system. By representing the classes and their relationships in a visual diagram, class models provide a clear and concise representation of the system, and they help developers to identify and resolve potential issues or problems with the system.

The role of class models in software engineering is to represent the structure and behavior of a software system. Class models are a type of diagram that is used in object-oriented modeling and design, and they are an important tool for understanding and designing the architecture of a software system. The role of class

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/class-model-for-mobile-game-architecture/322072

Related Content

Mobile Code and Security Issues
E. S. Samundeeswariand F. Mary Magdalene Jane (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications (pp. 2568-2582).*
www.irma-international.org/chapter/mobile-code-security-issues/26678

Cache Invalidation in a Mobile Environment
S. Lim (2007). *Encyclopedia of Mobile Computing and Commerce (pp. 102-107).*
www.irma-international.org/chapter/cache-invalidation-mobile-environment/17060

The Value of Mobile Communication for Social Belonging: Mobile Apps and the Impact on Social Interaction
Sara Henriquesand Manuel Jose Damasio (2016). *International Journal of Handheld Computing Research (pp. 44-58).*
www.irma-international.org/article/the-value-of-mobile-communication-for-social-belonging/167834

Networked Wireless Sensors, Active RFID, and Handheld Devices for Modern Car Park Management: WSN, RFID, and Mob Devs for Car Park Management
Djamel Djenouri, ElMouatezbillah Karbab, Sahar Boulkabouland Antoine Bagula (2015). *International Journal of Handheld Computing Research (pp. 33-45).*
www.irma-international.org/article/networked-wireless-sensors-active-rfid-and-handheld-devices-for-modern-car-park-management/144335

Face Recognition System using Discrete Cosine Transform combined with MLP and RBF Neural Networks
Fatma Zohra Chelaliand Amar Djeradi (2012). *International Journal of Mobile Computing and Multimedia Communications (pp. 11-35).*
www.irma-international.org/article/face-recognition-system-using-discrete/73718