

ARCHITECTURE OF MOBILE AR APPLICATIONS USING UNITY AND ARCORE FOR GAMIFIED ENVIRONMENTS

The rapid development of mobile augmented reality technologies has necessitated the formation of robust architectural approaches to the design of AR applications capable of ensuring stable performance, scalability, and a high level of interactivity in gamified environments. The complexity of gamified scenarios, which combine spatial tracking, real-world physical context, game logic, and user interaction, requires a carefully designed software architecture. In this context, the architecture of a mobile AR application cannot be limited to the conventional hierarchy of Unity scenes and objects. It must take into account the specific characteristics of ARCore as a system that processes sensor data in real time, the limited computational resources of mobile devices, and the need to support game mechanics such as progression, rewards, game states, and interaction with virtual objects embedded in the physical environment. Consequently, the architecture of an AR application can be conceptualized as a multi-layered system in which each layer performs a clearly defined function.

At the foundational level of the architecture lies the hardware-dependent layer, which includes the mobile device's sensors, such as the camera, gyroscope, accelerometer, and depth sensors. ARCore aggregates data from these sensors by performing complex visual-inertial odometry algorithms to determine the device's position and orientation in space. From an architectural perspective, it is essential to isolate this layer from the game logic, since changes in sensor data must be processed asynchronously and should not disrupt the stability of Unity's game loop [1].

The next level is represented by the ARCore service layer, which provides functionality for spatial tracking, plane detection, anchor recognition, and lighting estimation. From an architectural standpoint, the use of abstraction patterns is advisable, allowing Unity-level logic to interact not directly with the ARCore APIs but through an intermediate service layer. This approach increases system modularity and simplifies subsequent adaptation of the architecture to other AR platforms or future SDK updates.

A key architectural component of a gamified AR application is the layer responsible for scene and game state management. Unlike conventional mobile games, where scenes typically have clearly defined boundaries, AR applications operate within an open physical space. This necessitates the development of a state management system capable of responding to changes in the real environment, loss of tracking,

reinitialization of the ARCore session, and user actions. From an architectural perspective, the application of a finite state machine or the state pattern is an effective solution for managing the lifecycle of AR sessions and game scenarios [2].

In gamified AR applications, interaction extends beyond traditional screen-based input and includes spatial movement, gestures, and changes in viewpoint. This requires a multi-channel input processing system that integrates touch input, AR-based raycasting, and events generated by ARCore. Architecturally, it is effective to separate input handling into a data acquisition layer and an interpretation layer, which enables flexible modification of interaction mechanics without restructuring the entire system.

An important architectural element is the resource management system. Gamified AR applications often employ complex 3D models, animations, and shaders, which place significant load on the mobile device's GPU. Architectural solutions should therefore incorporate level-of-detail systems, object pooling, and adaptive resource loading based on the state of the AR session. This approach ensures an optimal balance between visual quality and application stability [3].

In conclusion, the architecture of mobile AR applications using Unity and ARCore for gamified environments represents a complex multi-layered system that integrates sensor technologies, spatial computing, game logic, and optimization mechanisms. Its effectiveness is determined not only by the correct use of ARCore and Unity tools but also by the depth of architectural design, which ensures scalability, stability, and technical reliability of gamified AR solutions. Ultimately, the architectural approach becomes a key factor in the successful implementation of complex AR applications in the contemporary mobile environment.

References

1. Lanham, M. (2018). *Learn ARCore: Fundamentals of Google ARCore*. Berkeley: Apress.
 2. Glover, J. (2018). *Unity 2018 Augmented Reality Projects: Build Four Immersive and Fun AR Applications Using ARKit, ARCore, and Vuforia*. Birmingham: Packt Publishing.
 3. Azuma, R.T. (2015). *Location-Based Mixed and Augmented Reality Storytelling*. Cham: Springer.
 4. Deterding, S., & Dixon, D. (2011). *From Game Design Elements to Gamefulness: Defining "Gamification"*. New York: ACM Press.
- Academic supervisor: Cand. Sc. (Technic), Ass. Professor. Suprun O.