https://doi.org/10.24867/GRID-2024-p20 Professional paper

DEVELOPMENT OF A MOBILE AUGMENTED REALITY APPLICATION FOR INTERIOR DESIGN USING UNITY ENGINE

Stefan Đurđević, Iva Juretić, Luka Vukanić
University of Novi Sad, Faculty of Technical Sciences, Department of Graphic Engineering and Design, Novi Sad, Serbia

Abstract: *This paper details the creation of a mobile Augmented Reality (AR) application to enhance interior design processes by allowing users to place digital furniture within their physical spaces. Developed using the Unity Engine, the application leverages C# programming to provide a seamless and interactive user experience. The user interface, designed in Adobe XD, ensures functionality and aesthetic appeal, guiding users through intuitive screens and interactive elements. Three-dimensional (3D) models are integral to the application, created through manual modelling in Blender and photogrammetry using Polycam. Blender's advanced capabilities enable detailed and accurate 3D models, while Polycam's photogrammetry converts real-world objects into digital models, enhancing realism and diversity. Real-time functionality is supported by integrating Google Firebase for scalable and reliable backend services, ensuring smooth data synchronization and user-generated content storage. AR technology overlays digital information onto the physical world in real time, enhancing users' perception and decision-making in interior design. Users can visualize different arrangements and make informed choices by virtually placing and manipulating furniture, reducing uncertainty and increasing satisfaction with their design decisions. This paper also explores the historical context and advancements in AR technology, highlighting its transformative impact on various industries. The development process here combines programming expertise, creative design, and technical knowledge of 3D modelling and backend technologies, offering valuable insights and practical advice for future AR application developers.*

Key words: Augmented Reality, Unity Engine, 3D Modelling, Interior Design, Google Firebase

1. INTRODUCTION

Augmented Reality (AR) has rapidly emerged as a transformative technology, bridging the gap between the physical and digital worlds. By overlaying digital content onto real environments, AR has found applications across various industries, including entertainment, education, retail, and design. AR provides a powerful tool for visualizing and experimenting with design concepts in real-time in interior design. This technology allows users to place and manipulate digital furniture within their living spaces, offering a more interactive and immersive design experience (Rhonda et al., 2024).
This paper presents the development of a mobile AR application specifically designed for interior design, utilizing the Unity Engine as the primary development platform. The goal of the application is to enable users to visualize furniture layouts in their homes, helping them make more informed decisions during the design process. The app integrates several technologies, including C# programming for interactive features, Adobe XD for user interface design, Blender and Polycam for 3D model generation, and Google Firebase for real-time backend services. Together, these tools create an intuitive and visually appealing solution, enhancing user engagement and decision-making in interior design.

1.1 Augmented Reality (AR)

Augmented Reality (AR) is a technology that blends digital content with the real world, allowing users to interact with virtual elements in real time. Unlike virtual reality, which immerses users in a wholly digital environment, AR enhances the physical world by overlaying information, graphics, and 3D models onto real surroundings. This unique capability has made AR a versatile tool across a variety of industries, offering new ways for users to engage with both digital and physical spaces (Cipresso et al., 2018).
The concept of AR has roots dating back to 1967 when Ivan Sutherland, a Harvard professor, and his student Bob Sproull developed the first head-mounted display system, famously known as "The Sword of Damocles." This early device allowed users to experience computer-generated graphics overlaid onto their real-world view, marking the inception of augmented reality. However, the term "Augmented Reality" was officially coined in 1990 by Tom Caudell, a researcher at Boeing, who was working on a system to assist workers in airplane construction by overlaying digital information on their tasks. This

system allowed workers to visualize complex assembly instructions, laying the groundwork for future AR developments (Colocation America, 2018; Mealy, 2018).
AR technology has evolved significantly, becoming commercially available and widely adopted, especially through mobile devices. With the integration of AR into smartphones and tablets, the technology has become accessible to millions of users worldwide. Companies like Apple and Google have made major strides in AR development with the creation of ARKit for iOS and ARCore for Android, providing developers with powerful tools to create immersive AR experiences. Additionally, platforms like Vuforia, Easy AR, and Snapchat Lens Studio further expand the possibilities of AR development, enabling a wide range of applications in gaming, retail, education, and interior design. The mass adoption of AR has been further fuelled by advancements in hardware, such as AR glasses and headsets, which continue to push the boundaries of how AR can be integrated into everyday life (TechTarget, 2023).

1.2 Development of AR applications for Interior Design

The integration of Augmented Reality (AR) into interior design has transformed how users approach space planning and decoration. AR applications allow users to visualize furniture and decor in real-time, within their actual living environments, providing a practical and immersive solution to traditional design challenges. By overlaying 3D furniture, appliances, and decor models onto physical spaces, users can experiment with various layouts, styles, and arrangements, understanding how their choices will look and feel before making any physical changes (Medium, 2024).
The ability to manipulate virtual objects within a real environment helps users make more informed decisions, eliminating much of the uncertainty associated with interior design. This technology also facilitates collaboration between clients and designers, as both parties can visualize potential design solutions together, ensuring that the end result meets expectations. The rise of AR in interior design has empowered users to take a more hands-on approach to their projects, giving them greater control and flexibility in shaping their spaces.
As AR technology continues to evolve, its applications in interior design are expected to become even more sophisticated. This will allow for more advanced interactions, enhanced realism, and broader accessibility across different platforms. Whether for home renovations, office space planning, or retail store layouts, AR offers an innovative tool that enhances creativity, efficiency, and user satisfaction in the design process.

1.3 Mobile AR application Development using Unity Engine

The Unity Engine has become a popular choice for developing mobile AR applications due to its versatility, powerful feature set, and cross-platform capabilities. Unity offers a robust development environment that simplifies the creation of interactive and immersive AR experiences. With built-in support for ARKit and ARCore, Unity allows developers to seamlessly integrate augmented reality functionalities into both iOS and Android devices, making it ideal for creating applications that can reach a wide audience (Daily.dev, 2024).
Unity enables real-time 3D rendering in mobile AR application development, providing smooth and responsive user interactions. Developers can use Unity's scripting language, C#, to manage the application's logic, such as object placement, scaling, and user interaction. Additionally, Unity's asset pipeline supports importing 3D models, textures, and animations from external tools like Blender, making integrating high-quality graphics into the application easier.
Another key advantage of Unity is its comprehensive toolset, which includes debugging, performance optimization, and testing features. This ensures that AR applications perform efficiently on mobile devices with varying hardware capabilities. Unity also offers extensive documentation and a wide array of plugins, making it a flexible and developer-friendly platform for creating innovative and user-centric AR applications.

1.4 User Interface and User Experience Design

In the development of AR applications, User Interface (UI) and User Experience (UX) design play a crucial role in ensuring the application is both functional and intuitive. A well-designed UI guides users through the application seamlessly, allowing them to interact with AR features without confusion or difficulty. In AR interior design applications, the interface must be simple and clear, allowing users to easily navigate through furniture options, placement tools, and customization features.

UX design focuses on how users feel when interacting with the application, aiming to provide an engaging and satisfying experience. For AR applications, this means creating a smooth flow from one task to another, minimizing the complexity of interactions, and ensuring that visual elements and controls are easy to understand. The AR experience must be immersive without overwhelming the user, and interactions such as tapping, dragging, and rotating objects should feel natural and responsive.
The design process typically involves prototyping and testing through tools like Adobe XD or Figma, where designers can experiment with different layouts, interactions, and design elements. User feedback is essential during this phase, as it helps refine the interface to meet the needs and expectations of the target audience. An effective UI/UX design in AR applications ensures that users can fully engage with the technology, making their interaction with the virtual elements intuitive, enjoyable, and productive (Interaction Design Foundation, 2016).

1.5 3D modelling for AR applications

3D modelling is a critical component in the development of AR applications, as it creates the virtual objects that users interact with in their real-world environment. High-quality 3D models enhance the realism and effectiveness of AR experiences, making them more engaging and useful. For AR applications, especially in fields like interior design, the accuracy and detail of these models are essential for providing a believable and immersive experience (RebusFarm, 2023).
The process of creating 3D models for AR involves various techniques and tools. Manual modelling using software such as Blender allows for detailed and customized object creation, enabling designers to produce intricate and high-fidelity representations of furniture and decor. Blender's powerful features support the creation of complex geometries, textures, and animations, which are crucial for ensuring that virtual objects look and behave realistically in an augmented environment.
Additionally, photogrammetry techniques, employed through applications like Polycam, enable the conversion of real-world objects into digital 3D models by capturing and analysing multiple photographic images. This method enhances the diversity and authenticity of the 3D assets, allowing users to place realistic representations of actual items into their virtual spaces. Combining these modelling techniques results in a rich and varied set of digital assets, significantly improving the quality and utility of AR applications.

1.6 Backend Services and Data Management

Effective backend services and data management are fundamental to the functionality and performance of AR applications. They ensure seamless interaction and reliable data handling. In AR applications, backend services support various critical functions, including real-time data synchronization, user-generated content storage, and overall application performance.
Google Firebase, a popular backend solution, provides a comprehensive suite of tools for managing these aspects. It offers real-time database capabilities that enable instantaneous updates and synchronization of data across multiple devices, ensuring that users experience a consistent and up-to-date AR environment. Firebase also handles user authentication, cloud storage, and analytics, allowing developers to manage user data and track application performance efficiently.
Furthermore, backend services support complex interactions within AR applications, such as saving user preferences, tracking design changes, and enabling collaborative features. By leveraging robust backend infrastructure, developers can ensure that the application runs smoothly, scales effectively with growing user numbers, and provides a reliable and engaging experience. Effective data management and backend services are essential for maintaining the integrity and efficiency of AR applications, allowing users to interact with digital content effortlessly while ensuring the application remains responsive and functional (MoldStud, 2024).

2. AR APPLICATION DEVELOPMENT

The application's development process includes several stages, from configuring the development environment using Unity Engine and integrating core functionalities to designing the user interface, creating 3D models, and linking the application to a real-time database. Each of these steps is essential for ensuring that the application functions smoothly and delivers an effective user experience.

2.1 Setting Up the AR Environment

Once a new project is created, it is crucial to integrate specific packages to support augmented reality (AR) application development. By accessing the Package Manager Window, developers can explore and manage the various available packages and sub-packages within the development environment.
The essential packages to be imported and installed include:

- AR Foundation,
- Google ARCore XR Plugin,
- Apple ARKit XR Plugin,
- XR Interaction Toolkit,
- Universal Render Pipeline, and
- TextMeshPro.

After integrating the required packages, the project is configured by setting specific values and specifications to ensure optimal performance on mobile devices. For Android, in the Player Settings window, it is essential to disable Multithreaded Rendering for compatibility with older devices. Auto-Graphics API should be turned off, and OpenGLES53 should be selected in the Rendering group. IL2CPP should be set as the Scripting Backend, ARM64 should be enabled, and the minimum supported Android version should be specified (Figure 1). For Apple devices, only the minimum iOS version needs to be defined. Once these configurations are complete, the Google ARCore and Apple ARKit packages through the XR Plugin Management tab must be activated (IBM, 2021).

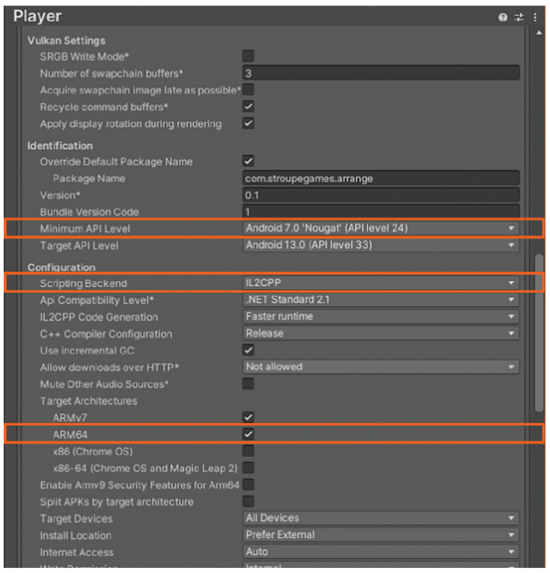


Figure 1: Player Settings

After configuring the key application parameters, the Universal Render Pipeline (URP) package is integrated. URP is a rendering system designed to provide high-quality graphics and optimal performance across both computers and mobile devices (Unity Technologies, 2024).
Once the application is optimized and the URP system implemented, a scene is created to establish the workspace for defining and applying necessary elements such as scripts and models.
A key aspect of AR applications is utilizing physical cameras, specifically the device's camera. AR Sessions and XR Origin are set up to define the application as an AR application, view the real world through the camera lens, and detect vertical and horizontal surfaces. The AR Plane Manager component is added to the XR Origin object to enable surface detection. Once the detected surfaces' appearance is defined through the AR Plane object and its material, and referenced in the AR Plane Manager component, the XR Origin, or the device's camera, displays these surfaces by generating a mesh of polygons on the recognized physical planes.

2.2 Basic Furniture Manipulation Functions Implementation

The fundamental functions for manipulating furniture or scene objects that are necessary for this type of application include:

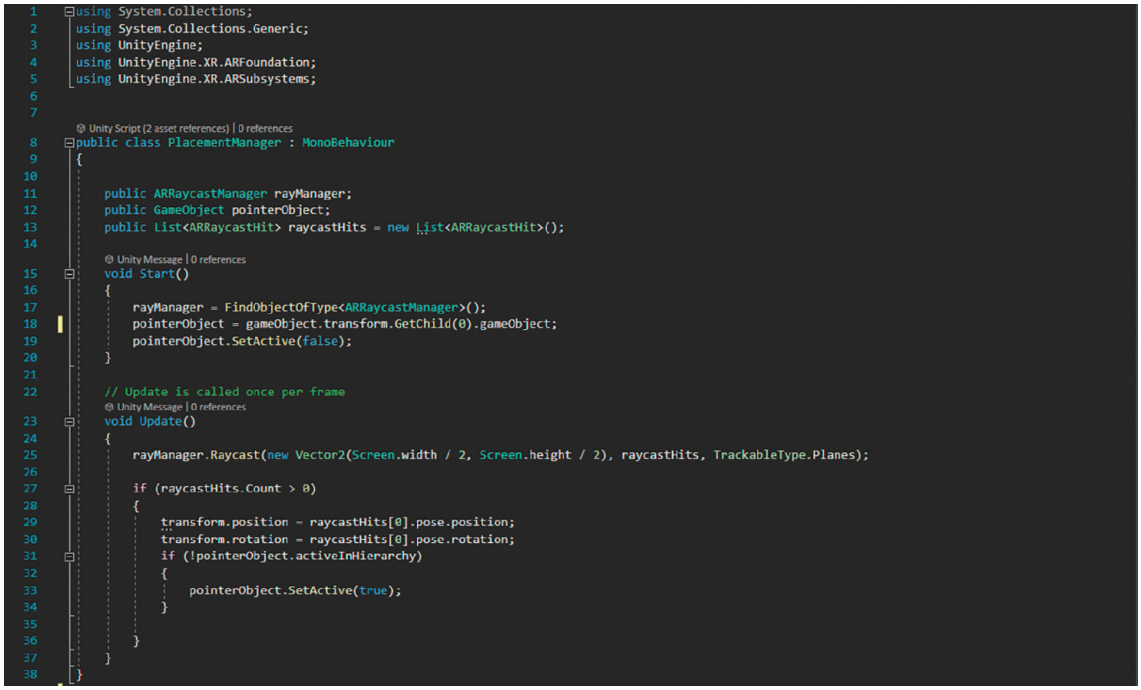
- instantiating objects in the scene,
- moving instantiated objects and
- rotating instantiated objects.

Instantiating objects involves creating copies of predefined models using C# scripts and the Instantiate() method. The placement of these objects is determined using the Raycast technique, which involves projecting directed rays to gather information about the impacted surfaces or previously placed furniture in the scene (Unity Technologies, 2024). The AR Raycast Manager component is attached to the XR Origin object, making the device's camera the source of these rays. The script ensures continuous raycasting towards the centre of the device's screen. When the rays hit detected horizontal surfaces, an identifier is displayed, marking where the selected furniture will be placed. The script responsible for casting a ray to place furniture instances at the intersection point of the ray and the detected surface is called the PlacementManager script and is illustrated in Figure 2.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;


Unity Script (2 asset references) | 0 references
public class PlacementManager : MonoBehaviour
{

    public ARRaycastManager rayManager;
    public GameObject pointerObject;
    public List<ARRaycastHit> raycastHits = new List<ARRaycastHit>();

    Unity Message | 0 references
    void Start()
    {
        rayManager = FindObjectOfType<ARRaycastManager>();
        pointerObject = gameObject.transform.GetChild(0).gameObject;
        pointerObject.SetActive(false);
    }

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        rayManager.Raycast(new Vector2(Screen.width / 2, Screen.height / 2), raycastHits, TrackableType.Planes);

        if (raycastHits.Count > 0)
        {
            transform.position = raycastHits[0].pose.position;
            transform.rotation = raycastHits[0].pose.rotation;
            if (!pointerObject.activeInHierarchy)
            {
                pointerObject.SetActive(true);
            }

        }
    }
}
```

Figure 2: PlacementManager script

Moving objects works similarly: rays are cast towards the touch point when the screen is touched. If a ray hits and identifies a previously instantiated object, the object is anchored to the touch point until the screen is released. This functionality is defined in the SelectObjects script (Figure 3).



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;


Unity Script (2 asset references) | 0 references
public class PlacementManager : MonoBehaviour
{

    public ARRaycastManager rayManager;
    public GameObject pointerObject;
    public List<ARRaycastHit> raycastHits = new List<ARRaycastHit>();

    Unity Message | 0 references
    void Start()
    {
        rayManager = FindObjectOfType<ARRaycastManager>();
        pointerObject = gameObject.transform.GetChild(0).gameObject;
        pointerObject.SetActive(false);
    }

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        rayManager.Raycast(new Vector2(Screen.width / 2, Screen.height / 2), raycastHits, TrackableType.Planes);

        if (raycastHits.Count > 0)
        {
            transform.position = raycastHits[0].pose.position;
            transform.rotation = raycastHits[0].pose.rotation;
            if (!pointerObject.activeInHierarchy)
            {
                pointerObject.SetActive(true);
            }

        }
    }
}
```

Figure 3: SelectObjects script

Rotation is managed based on the relative position of two fingers touching the screen and occurs exclusively around the vertical axis of the object.

2.3 Design and Implementation of the User Interface

User Interface (UI) design involves visually creating the layout and appearance of software or devices that use screens for user interaction (BrowserStack, 2023). This design process is facilitated by Adobe XD, which provides a comprehensive set of tools. A segment of the User Interface is displayed in Figure 4. The visual components of the application include:

- the welcome and loading screen,
- the main application screen,
- the product category selection menu,
- the submenu for product categories and the display of products in each category,
- the individual product details screen,
- the screen for placing products in the physical environment including manipulation features and
- the settings menu.

After exporting the graphical elements from Adobe XD, they are imported into Unity Engine by dragging them into the project window and setting their texture type to Sprite (2D and UI). These elements are integrated within the Canvas object in Unity Engine, which encompasses the entire application's UI. All menus, buttons, images, and text elements within the Canvas are defined along with their respective functions and methods.

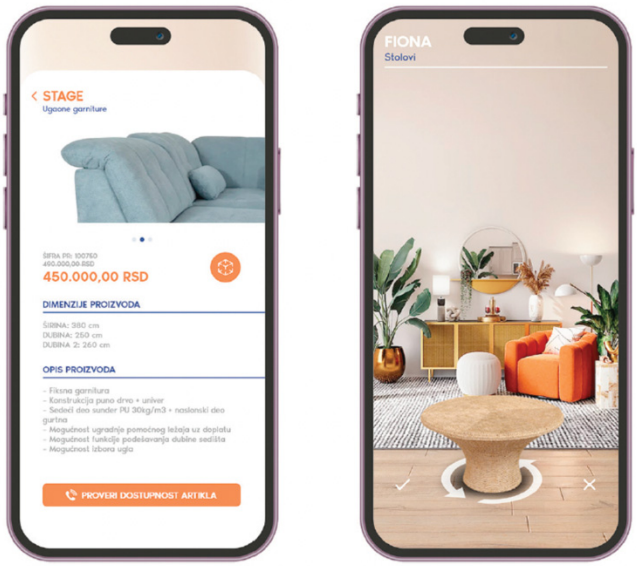


Figure 4: AR application User Interface

Unity Engine also supports the import of external fonts, which are processed using the TextMeshPro package to create a font atlas, enabling the use of custom fonts.

2.4 Creation and Preparation of 3D Models

In this research, the creation and design of three-dimensional models are accomplished through two main methods:

- manual modelling using Blender and
- photogrammetry with Polycam.

For instance, in Blender, the design of a Himalayan salt lamp begins with a basic 3D cube. This cube is modified using the Subdivision Surface modifier to create a more detailed polygon mesh, and the Displace modifier is applied to shape the mesh into an irregular, natural form (Figure 5a). The Decimate Geometry function is used to optimize the polygon mesh. The lamp's base and material are also created based on a texture image of the Himalayan salt lamp, resulting in the 3D model depicted in Figure 5b.

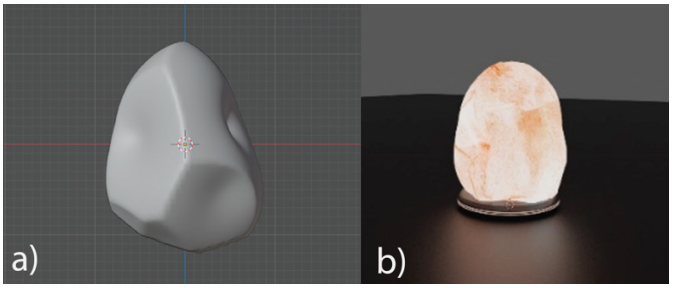


Figure 5: Himalayan salt lamp, a) manual 3D modelling after modifier applied and b) the final result

When exporting a 3D model from Blender, it is crucial to include the model and its applied textures and materials.
For photogrammetry-based 3D model creation, the real-world object- such as a coffee table in this case-is photographed from multiple angles. These photos are then processed using Polycam, resulting in a 3D model that may exhibit significant distortions and may not meet the standards for an accurate representation (Figure 6). Users expect a faithful digital reproduction when selecting and viewing items for purchase within the application.

Figure 6: Coffee table photographs and final 3D model

2.5 Integration of Firebase Database and Storage

The application only loads data and models that the user selects to optimize memory usage. The application employs the Firestore Database to store essential product information and Firebase Storage to handle images and furniture models. Before these services can be used, Firebase must be integrated into the Unity project, and a database and storage must be set up on the Firebase platform, linking the project to Firebase services through the Firebase console.
Once integration is complete, the database and storage are organized into the following hierarchy:

- all products,
- room,
- product type,
- individual product,
- product details.

Product data, models, and images are retrieved in real time as the user selects items, utilizing predefined methods from the Firebase package.

3. RESULTS

This research paper details the development of a mobile augmented reality application designed to enable users to place digital furniture within their physical space. The application was built using Unity Engine, with 3D models created in Blender and Polycam. Google Firebase was employed for backend services.
The outcome is a fully functional AR application that allows users to virtually position and adjust furniture in real-world settings using mobile devices. Additionally, the project includes a comprehensive workspace with all the required packages and configurations for developing this type of application (Figure 7).

Figure 7: Display of the application on the device

The key stages in developing the application involve setting up the development environment, implementing features for placing and moving furniture, designing the user interface, creating 3D models, and integrating with Firebase for data storage.
This application provides a user-friendly and enjoyable method for planning and visualizing interior design. It allows users to see how specific furniture items would look in their own space, assisting them in making informed purchase decisions.

4. DISCUSSION

The development of Augmented Reality (AR) applications is a multifaceted endeavour that integrates various disciplines, including programming, creative design, 3D modelling, and backend management. This discussion examines the critical aspects of AR application development, with a focus on the AR application, which allows users to visualize and manipulate furniture in their physical spaces.
One of the primary challenges in AR development is configuring the development environment effectively. For AR application integrating essential AR packages within the Unity Engine was pivotal. The inclusion of AR Foundation, Google ARCore XR Plugin, and Apple ARKit XR Plugin provided the necessary tools for creating a seamless AR experience. These packages facilitated interaction with real-world environments and ensured cross-device compatibility. The successful implementation of these technologies highlights the importance of selecting appropriate development tools and configuring them correctly to build a robust AR application.
The technical complexity involved in the application's core functionalities-such as furniture placement, movement, and rotation-underscores the sophisticated nature of AR development. Utilizing Raycast techniques for precise object placement exemplifies the detailed understanding required in spatial computing and user interaction. The implementation of these features using C# scripting illustrates the significant effort needed to create a responsive and intuitive AR experience.
Another crucial element is designing an engaging user interface (UI). The use of Adobe XD to create visually appealing screens and integrate these designs into Unity Engine through the Canvas system demonstrates the role of UI in enhancing user experience. Thoughtful design of interaction points, including welcome screens, product selection menus, and settings, ensures easy navigation and effective user engagement. The incorporation of external fonts and graphical elements further refines the application's appearance, emphasizing the importance of a polished and professional design.
The creation of 3D models, whether through manual modelling in Blender or photogrammetry using Polycam, adds another layer of complexity. The article emphasizes the significance of accurate and realistic 3D models in enriching the AR experience. Detailed discussions on modelling techniques, including texture mapping and optimization, highlight the skill required to produce high-quality models that enhance the application's realism and functionality.
Backend services and data management are critical for managing real-time data and user-generated content. The integration of Firebase services allows for efficient data storage and retrieval, ensuring that the application remains responsive and reliable. The hierarchical organization of product data within the Firestore Database and Firebase Storage exemplifies the structured approach necessary for effective data management.

5. CONCLUSIONS

The development of an AR application is a complex, interdisciplinary process that demands a blend of programming expertise, creative design, and technical knowledge in 3D modelling and backend technologies. This article provides a comprehensive overview of the development process, from initial concept through to implementation and optimization, offering valuable insights and practical advice for creating AR applications. By balancing technical performance with user experience, developers can achieve a successful and effective AR application. This approach is crucial for meeting the demands of modern applications and ensuring a satisfying user experience.

6. ACKNOWLEDGMENTS

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. 451-03-65/2024-03/200156) and the Faculty of Technical Sciences, University of Novi Sad through project "Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad" (No. 01-3394/1).

7. REFERENCES

BrowserStack. (2023) *What is User Interface (UI)? (Types & Features).* Available from: https://www.browserstack.com/guide/what-is-user-interface [Accessed 20th August 2024].

Cipresso, P., Giglioli, I. A. C., Raya, M. A. & Riva, G. (2018) The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature. *Frontiers in Psychology*. 9, 1-20. Available from: doi: 10.3389/fpsyg.2018.02086

Colocation America. (2018) *The History of Augmented Reality.* Available from: https://colocationamerica.com/blog/history-of-augmented-reality [Accessed 20th June 2024].

Daily.dev. (2024) *Unity vs Unreal Engine for VR/AR Development.* Available from: https://daily.dev/blog/unity-vs-unreal-engine-for-vrar-development [Accessed 1st February 2024].

IBM. (2021) *SDK vs. API: What's the difference?* Available from: https://www.ibm.com/think/topics/api-vs-sdk [Accessed 9th August 2024].

Interaction Design Foundation. (2016) *What is User Experience (UX) Design?* Available from: https://www.interaction-design.org/literature/topics/ux-design [Accessed 1st July 2024].

Mealy, P. (2018) *Virtual & Augmented Reality. Hoboken,* New Jersey, John Wiley & Sons, Inc.

Medium. (2024) *How is Augmented Reality Revolutionizing Interior Design?* Available from: https://subrahmanyam-plugxr.medium.com/how-is-augmented-reality-revolutionizing-interior-design-93de4ac8b5dd [Accessed 2nd July 2024].

MoldStud. (2023) *The Importance of Back-End Development in Augmented Reality Applications.* Available from: https://moldstud.com/articles/p-the-importance-of-back-end-development-in-augmented-reality-applications [Accessed 19th May 2024].

RebusFarm. (2023) *Augmented Reality and Virtual Reality in 3D Modeling.* Available from: https://rebusfarm.net/blog/augmented-reality-and-virtual-reality-in-3d-modeling [Accessed 10th May 2024].

Rhonda, H. & Eric, S. P. (2024) Bridging the digital and physical: The psychology of augmented reality. London, Thomas Telford Publishing, *Current Opinion in Psychology*. 58, 1-6. Available from: doi: 10.1016/j.copsyc.2024.101842

TechTarget. (2023) *What is Google Glass?* Available from: https://techtarget.com/iotagenda/definition/Google-Glass [Accessed 11th July 2024].

Unity Technologies. (2023) *AR Raycast Manager.* Available from: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/raycast-manager.html [Accessed 16th August 2024].

Unity Technologies. (2024) *Introduction to the Universal Render Pipeline.* Available from: https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@17.0/manual/index.html [Accessed 15th August 2024].