ECT | Revista Cientifica Emprendimiento Científico Tecnologico

ISSN: 2810 – 8493

https://revista.ectperu.org.pe/index.php/ect/index

48 Hours of Augmented Reality: Surviving a Game Jam Without Visual Scripting

Jared Abraham Flores Guarneros
Computacion
Tec de Monterrey
Monterrey, Mexico
A01379868@exatec.tec.mx

Norman Gael Ávila Martinéz
Computacion
Tec de Monterrey
Monterrey, Mexico
A01379453@exatec.tec.mx

Carlos Gabriel Mora Madrigal
Computacion
Tec de Monterrey
Monterrey, Mexico
A01379575@exatec.tec.mx

Dr. Lorena B. Martinez E.
Computacion
Tec de Monterrey
Monterrey, Mexico
lorenamtze@tec.mx

***Abstract* — In this text, we explore the capabilities of the AR Foundation and its implementation through a coded approach for creating video games in Augmented Reality (AR). The rising popularity of visual scripting in game development offers a seemingly user-friendly interface for creators; however, it often comes at the cost of producing redundant or inefficient code, which can be particularly detrimental in the resource-constrained environment of mobile devices. By forgoing visual scripting in favor of direct coding, we propose a pathway not only to avoid these pitfalls but also to harness the full potential of AR development. This approach seeks to streamline the development process during the timeframe of a game jam, highlighting the challenges and triumphs of surviving 48 hours of AR game development without relying on visual scripting.**

Keywords—AR, Unity, AR Foundation, Code, GameJam , Visual Scripting

I. INTRODUCTION

In recent years, the video game industry has experienced rapid evolution driven by technological advances such as virtual reality and improvements in graphics. This evolution poses challenges to staying current, as developers must constantly adapt to deliver innovative and engaging experiences that make the most of new technologies, preventing their products from becoming obsolete.

Augmented Reality (AR) represents an enhanced and engaging version of the real world, by integrating with visual elements, sounds and sensory stimuli through holographic technology. AR offers a fusion between the digital and physical world, real-time interactions, as well as precise 3D identification of virtual and real objects.[1]

AR integration has been used in various games in recent years, but its popularity increased significantly with the release of Pokémon GO, developed by Niantic in collaboration with Gamefreak, becoming one of the most popular games of the last decade. This game transformed the perception of the industry, laying global foundations that could drive new AR projects, whether through smartphones or devices such as glasses. [2]

II. EXAMPLES OF VIDEO GAMES DEVELOPED WITH AR FOUNDATION

AR Foundation and Unity helped increment the number of games that combine augmented reality with immersive gaming experiences. Within the most known AR video games are:

Pokémon GO : This game allows players to catch Pokémon in the real world using AR. Developed in Unity, it uses ARCore on Android and ARKit on iOS.

Harry Potter : Wizards Unite : Players explore the magical world of Harry Potter in real life, discovering creatures and artifacts using AR. Developed in Unity, it uses AR Foundationto support a variety of devices.

Minecraft Earth : Inspired by Minecraft, this game allows players to build in the real world using AR. Developed in Unity, uses ARCore and ARKit .

Jurassic World Alive : Players hunt dinosaurs in the real world using AR. Developed in Unity, uses ARKit and ARCore .

The Walking Dead : Our World : Players fight zombies in the real world using AR. Developed in Unity, uses ARCore and ARKit .

III. AR Foundation: Fundamentals and Features

Unity's AR Foundation is a cross-platform framework

designed for developing augmented reality experiences. It allows creators to design and develop their AR apps once and then build them for Android or iOS devices without any additional settings. This functionality ensures a smooth and efficient transition between platforms, making it easy for developers to maximize their reach with minimal adaptation efforts.

A. Main features of AR Foundation.

- **Session**
  - Enables, disables and configures augmented reality on the target platform.
- **Device Tracking**
  - Monitors the position and rotation of the device in physical space.
- **Camera**
  - Renders images captured by the device's camera and estimates ambient lighting.
- **Plane Detection**
  - Detects and monitors flat surfaces.
- **Image Tracking**
  - Detects and tracks 2D images.
- **Object Tracking**
  - Detect and track 3D objects.
- **Body Tracking**
  - Detects and tracks human bodies.
- **Face Tracking**
  - Detects and tracks human faces.

- **Point Clouds**
  - Detect and track key points in the environment.
- **Raycast**
  - Cast rays at detected objects to interact with them.
- **Anchors**
  - Allows you to fix arbitrary points in space that remain constant.
- **Meshing**
  - Generates meshes of the environment for a detailed representation of it.
- **Environment Probes**
  - Create cubemaps to reflect the surrounding environment.
- **Occlusion**
  - Allows you to hide augmented reality content behind physical objects and performs human segmentation.
- **Participants**
  - Monitor other devices in shared AR sessions.

[3]

The AR Foundation package acts as a set of interfaces for augmented reality functionality, without directly implementing any features. To use AR Foundation on a specific platform, it is necessary to additionally install a plugin package provided by the manufacturer of that platform. Unity offers official support for the following vendor plugins:

- **Google ARCore XR** Plugin for Android.
- **Apple ARKit XR** Plugin for iOS.
- **Apple VisionOS XR** Plugin for VisionOS.
- **OpenXR Plugin** for HoloLens 2.
- **Unity OpenXR** : Meta for Meta Quest.

These plugins enable the implementation of AR Foundation capabilities on their respective platforms, allowing developers to create cross-platform augmented reality applications with a common code base. [4]

Figure 1 outlines the key features of leading augmented reality platforms, including ARCore, ARKit, and OpenXR, among others. It serves as a vital tool for developers to assess the level of interactivity and specific functionalities each platform offers for their AR projects. By referring to this table, developers can make informed choices about the most suitable technologies for achieving their design goals, thereby enhancing the end-user experience across various devices.

Features	ARCore	ARKit		OpenXR	
	Android	IOS	VisionOS	HoloLens	Goal Quest
Session	Yes	Yes	Yes	Yes	Yes
Device tracking	Yes	Yes	Yes	Yes	Yes
Plane detection	Yes	Yes	Yes	Yes	Yes
Image tracking	Yes	Yes	Yes		
Object tracking		Yes			
Face tracking	Yes	Yes			
Body tracking		Yes			
Point clouds	Yes	Yes			
Raycasts	Yes	Yes		Yes	Yes
Anchors	Yes	Yes	Yes	Yes	Yes
Meshing		Yes	Yes	Yes	
Environment probes	Yes	Yes			
Occlusion	Yes	Yes			
Participants		Yes			

Fig. 1. Comparative Overview of Augmented Reality Platform Features

Drawing upon the insights and data from [5] [6] [7] [4] [8] [9] we constructed Figure 2, which offers a comprehensive comparison of key features across various augmented reality development platforms and tools, including AR Foundation, ARKit/ARCore, Vuforia, RealityKit, and Unreal Engine. This analysis encompasses factors such as multi-platform compatibility, image/object recognition capabilities, user-friendliness, cost implications, integration ease with Unity, device support range, availability of location-based AR features, graphical fidelity, and access to the latest technological advancements. The table can be used as a guide for developers to evaluate and select the most suitable AR development tool or platform that aligns with their project requirements and goals.

B. *Comparison to other AR development platforms*

Criterion	AR Foundation	ARKit / ARCore	Vuforia	Reality Kit	Unreal Engine
Compatibility Multi platform	Yes	No	Yes	No	Yes
Image / Object Recognition	Limited	Yes	Excellent	Limited	Limited
Features Advanced	Yes	Yes	Yes	Yes	Yes
Easy to use	high	high	Half	high	Half

Camera	Yes	Yes			Yes

Cost	Free	Free	Subscription / Lic .	Free	Free / Varies
Integration with Unity	Narrow	N/A	Good	Narrow	Good
Support for a Wide Range of Devices	Yes	Limited	Yes	Limited	Limited
Location-Based AR Features	No	No	Yes	No	No
High Fidelity Graphics	No	No	No	Yes	Yes
Access to Latest Features	Depends on Unity	Yes	Depends on license	Yes	Yes

Fig. 2. Comparative Analysis of Augmented Reality Development Platforms and Tools

By synthesizing the information from both Figure 1 and 2, we gained a comprehensive understanding of various AR platforms and tools available. After careful consideration, we ultimately decided to utilize AR Foundation in conjunction with Unity due to the numerous benefits it offers. However, one drawback we encountered was that most tutorials for AR Foundation were geared towards utilizing visual scripting, which wasn't our preferred method. Despite this challenge, we recognized the potential of AR Foundation and Unity to meet our project requirements, leveraging their advanced features and multi-platform compatibility to create immersive AR experiences.

IV. Unity: Video Game Development Platform

Unity is a highly used game development platform that offers a wide variety of tools to create interactive experiences in 2D, 3D, and virtual and augmented reality (AR). It is known for its flexibility, ease of use, and its ability to create cross-platform games. Unity provides an intuitive editor that allows developers to design environments, create animations, program behaviors, and deploy their games on a variety of platforms, including PC, consoles, mobile devices, and the web . [10]

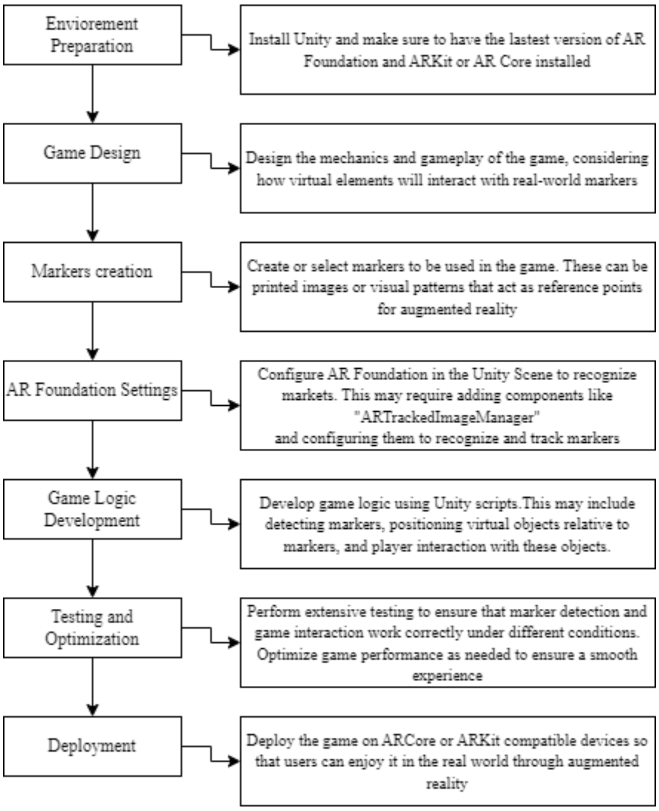
A. Advantages of using Unity for AR game development

Unity stands as a top choice for crafting AR games, thanks to its user-friendly nature and broad compatibility. With Unity, developers can effortlessly target various devices, spanning iOS and Android platforms, making their AR creations accessible to a wide audience. This versatility streamlines development, letting creators focus on crafting immersive experiences without fretting over compatibility issues. Plus, Unity's intuitive interface and extensive documentation make it easy for developers of all levels to dive into AR game development, regardless of their experience.

Moreover, Unity's vibrant community and strong graphics support add to its appeal. With a bustling ecosystem of developers and a treasure trove of tutorials, forums, and assets, Unity fosters collaboration and accelerates development. This community-driven approach encourages innovation, helping developers turn their AR gaming dreamsinto reality. Additionally, Unity's powerful graphics tools empower creators to build visually stunning AR environments that grab players' attention. And with AR Foundation seamlessly integrated into Unity, developers cancreate AR apps with ease, ensuring smooth performance across different devices. Overall, Unity's blend of accessibility, community support, graphics prowess, and ARFoundation integration makes it a go-to platform for captivating crafting AR gaming experiences.

V. Video Game Development with AR Foundation

A. Basic steps to start developing an AR game with ARFoundation.



Install Unity and make sure to have the lastest version of AR Foundation and ARKit or AR Core installed

Design the mechanics and gameplay of the game, considering how virtual elements will interact with real-world markers

Create or select markers to be used in the game. These can be printed images or visual patterns that act as reference points for augmented reality

Configure AR Foundation in the Unity Scene to recognize markets. This may require adding components like "ARTrackedImageManager" and configuring them to recognize and track markers

Develop game logic using Unity scripts.This may include detecting markers, positioning virtual objects relative to markers, and player interaction with these objects.

Perform extensive testing to ensure that marker detection and game interaction work correctly under different conditions. Optimize game performance as needed to ensure a smooth experience

Deploy the game on ARCore or ARKit compatible devices so that users can enjoy it in the real world through augmented reality

B. Description of personal experience as an AR gamedeveloper using AR Foundation.

During the Global Game Jam, the team embarked on the challenge of creating an augmented reality (AR) game usingthe AR Foundation, motivated by the event's theme: "Make Me Laugh." With the aim of creating a fun and novel experience, we created a game titled "Ayudando Ando". [11]

C. Challenges faced during the development process.

During the 48 hours of the event, our team faced a series of challenges both technical and creative. We started by divinginto the tutorials provided by Unity with the goal of mastering the generation of objects in markers through the use of the Image Track Manager. This initial step was crucial, as it allowed us to instantiate prefabs *(A prefab is a pre-made object or group of objects in Unity that can be reused across different parts of a game. It simplifies development by allowing quick and easy placement of game elements)* within the real environment, thus establishing a solid foundation upon which we could build interactive levels and immerse players in a unique and immersive gaming experience.

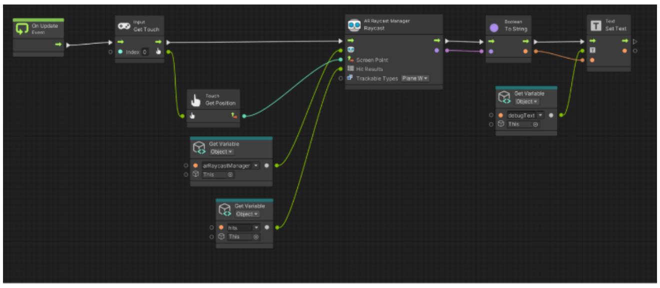
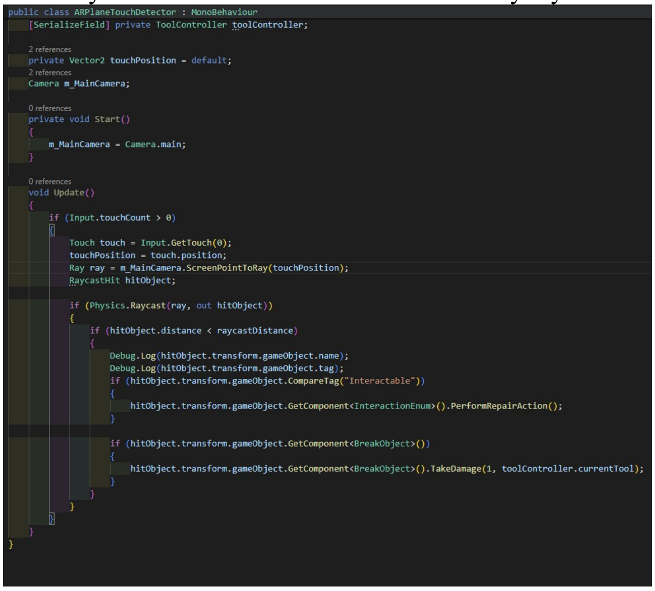


Fig. 3. AR solution in Visual Scripting

Despite the advantages of visual scripting, we soon ran into limitations in implementing additional functionality. Block programming was restrictive, and the lack of tutorials focused on direct coding prompted us to look for alternatives.

Our research led us to delve into the official documentation and follow specialized content creators, where we finally discovered that instead of the AR-specific ARRaycast, what we really needed was to use the traditional Unity raycast.



```
public class ARPlaneTouchDetector : MonoBehaviour
    [SerializeField] private ToolController toolController;

    2 references
    private Vector2 touchPosition = default;
    2 references
    Camera m_MainCamera;

    0 references
    private void Start()
    {
        m_MainCamera = Camera.main;
    }

    0 references
    void Update()
    {
        if (Input.touchCount > 0)
        {
            Touch touch = Input.GetTouch(0);
            touchPosition = touch.position;
            Ray ray = m_MainCamera.ScreenPointToRay(touchPosition);
            RaycastHit hitObject;

            if (Physics.Raycast(ray, out hitObject))
            {
                if (hitObject.distance < raycastDistance)
                {
                    Debug.Log(hitObject.transform.gameObject.name);
                    Debug.Log(hitObject.transform.gameObject.tag);
                    if (hitObject.transform.gameObject.CompareTag("Interactable"))
                    {
                        hitObject.transform.gameObject.GetComponent<InteractionEnum>().PerformRepairAction();
                    }

                    if (hitObject.transform.gameObject.GetComponent<BreakObject>())
                    {
                        hitObject.transform.gameObject.GetComponent<BreakObject>().TakeDamage(1, toolController.currentTool);
                    }
                }
            }
        }
    }
}
```

Fig. 4. Proposed solution for object detection in code

With the right code, we achieve seamless interaction with objects instantiated in the real world. This advancement meant a big step forward in our project, allowing us to continue development with a new perspective and tools.

D. *Lessons learned and advice for other developers interested in working with the AR Foundation.*

Despite the challenges encountered, this experience allowed us to learn valuable lessons about the importance of perseverance and adaptability in AR game development. We've found that experimentation and actively seeking alternative solutions are critical when working with emerging technologies like AR Foundation.

Additionally, we emphasize the importance of extensive planning and research before embarking on a project of this nature. A good understanding of the fundamental principles of augmented reality (*a combination of digital and physical worlds, interactions performed in real time, and precise 3D identification of virtual and real objects* [1]) and how AR Foundation works can greatly facilitate the development process and reduce the likelihood of facing significant technical obstacles.

VI. Challenges and Future of AR Video Game Development

As a team, we're constantly seeking out new tutorials, guides, and best practices to enhance our skills and expand our knowledge base in augmented reality. This proactive approach ensures that we're well-equipped to face the challenges and opportunities that come our way, including the integration of cutting-edge technologies like the Apple's Vision Pro. We excitedly anticipate the integration of AR Foundation with Vision Pro, as it promises to open up exciting possibilities for learning and potentially creating even more immersive games that push the boundaries of what's possible in augmented reality.

VII. Conclusions

The development of augmented reality (AR) games with Unity's AR Foundation marks an exciting milestone in interactive entertainment, merging the digital and physical worlds through technology and creativity. Despite the technical challenges and the quest for innovation, it offers enormous possibilities to transform our interactions with technology. However, it faces a shortage of documentation and pure code examples; most tutorials lean towards visual scripting, which can be an obstacle for those who prefer direct development in code. This aspect underlines the importance of expanding resources for a more technical implementation, allowing developers to fully exploit the potential of AR Foundation.

Acknowledgment

We would like to thank the REDINDVJ for the publication opportunity, as well as the IIDIV and 48 to play for the support during our participation in the Global Game Jam.

VIII. References

[1] Microsoft, "What is augmented reality or AR?" Microsoft, [Online]. Available: https://dynamics.microsoft.com/es-mx/mixed-reality/guides/what-is-augmented-reality-ar/. [Last access: February 24, 2024].

[2 Quasar Dynamics, "Augmented Reality in Video

] Games," Quasar Dynamics, [Online]. Available: https://quasardynamics.com/reality-aumentada-los- videojuegos/. [Last access: February 24, 2024].

[3] AJSR Tanisha Chaudhry, "AR Foundation for Augmented Reality in Unity," *International Journal of*

Advances in Engineering and Management, vol. 3, no. 1, pp. 662-667, 2021.

[4] Unity, "AR Foundation," Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/index.html. [Last access: February 24, 2024].

[5] Apple Inc, "Apple ARKit Documentation," Apple Inc, 2022. [Online]. Available: https://developer.apple.com/documentation/arkit. [Last access: February 26, 2024].

[6] Apple Inc, "RealityKit," Apple Inc, 2024. [Online]. Available: https://developer.apple.com/documentation/realitykit. [Last access: February 26, 2024].

[7] Google, "Overview of ARCore and supported development environments," Google, 2022. [Online]. Available: https://developers.google.com/ar/develop?hl=es-419. [Last access: February 26, 2024].

[8] Vuforia, "Vuforia Engine Library," 2024. [Online]. Available: https://developer.vuforia.com/library/. [Last access: February 26, 2024].

[9] Epic Games, "Developing for Handheld Augmented Reality Experiences," Unreal Engine, 2024. [Online]. Available: https://docs.unrealengine.com/5.3/en-US/developing-for-handheld-augmented-reality-experiences-in-unreal-engine/. [Last access: February 26, 2024].

[10] Unity, "Unity Documentation," 2024. [Online]. Available: https://docs.unity.com. [Last access: March 4, 2024].

[11] NACMRQ Jared Flores, "Ayudando Ando," Global Game Jam, January 2024. [Online]. Available: https://globalgamejam.org/games/2024/ayudando-ando-7. [Last access: March 4, 2024].