****

**Software Engineering Department**

**Braude College**

**Capstone Project Phase B**

**MathStARz**

**Bridging the Gap Between Math and Reality**

**By:**

**Adar Cohen - Adarcohen1@gmail.com**

**Yaniv Blitzman - yaniv.blitzman@gmail.com**

**Advisors:**

**Dr. Nataly Levi**

**Dr. Naomi Unkelos Shpigel**

**Project Code:**

**25-1-D-14**

**Link to Github:**

[**Github**](https://github.com/AdarCohen1/final-project-)

**Table of content**

[**Abstract 3**](#_p5voz2n1v39h)

[**1. Introduction 4**](#_lj43x94n86c4)

[**2. Background and Related Work 5**](#_tjfx8abvzy40)

[**2.1 Mathematics Education Challenges 5**](#_e929d6ls12x)

[**2.2 Augmented Reality 5**](#_8z5y8we09xtl)

[**2.2.1 Augmented Reality in Education 5**](#_9mgxcfe8knuk)

[**2.2.2 AR for Learning Mathematical Functions 6**](#_252q4e45d75f)

[**2.2.3 Enhancing Geometry and Spatial Reasoning 6**](#_1sv2t8m2sfmw)

[**2.3 Gamification in Educational 6**](#_4s56f51uevgv)

[**2.3.1 Gamified AR Applications 6**](#_dwnm5cqxp4tp)

[**2.4 Development of AR Applications 7**](#_in0t1qeuy3v)

[**2.5 Future Directions in AR and Gamified Learning 8**](#_j8gkaru1q53u)

[**3. AR Game Design and Functionality 8**](#_k9uu313hsl1m)

[**3.1 MathStarz AR Educational Game 8**](#_jcimfalwg8ro)

[**3.2 Development and Implementation Process 9**](#_l5gxh3e0uvm0)

[**3.2.1 Identifying Learning Challenges through Educator Insight 9**](#_n1ynf0jroebb)

[**3.2.2 Designing with Research-Based Insight 9**](#_7d0eyey4sie2)

[**3.2.3 Requirements 10**](#_9bqc1ogcw5f0)

[**4 Diagrams 11**](#_cmn83v4qoxzi)

[**4.1 System Architecture. 11**](#_u8f7easd4tdt)

[**4.2 Workflow diagram 13**](#_ozkv9i9fwhmr)

[**4.3 Use Case diagram 15**](#_qaahbswtpsng)

[**4.4 Activity diagrams 20**](#_13bd7ykb3an0)

[**4.5 Package diagram 23**](#_h9ogfmkikh4m)

[**5. Testing 23**](#_9kd660f1ipdr)

[**6. Development Process 25**](#_kmuak8uf7ltz)

[**6.1 System Logic and Backend Flow 25**](#_g9yx8z2o7sty)

[**6.1.1 User Management and Authentication 25**](#_1h2l297aiy4n)

[**6.1.2 Puzzle Progress and Game Interaction 26**](#_chfa6b5mf2gn)

[**6.1.3 Question Handling 26**](#_7pu574aevaz)

[**6.1.4 Leaderboard Functionality 26**](#_kz7hmgdfyt8d)

[**6.2 Challenges 27**](#_2f2vq0qvu216)

[**6.2.1. Challenges and solutions 28**](#_vww5a7brz6mz)

[**7. Conclusions 29**](#_bbtnjbr13wh8)

[**8. User Documentation 30**](#_f7al507ckulk)

[**8.1 User Guide 30**](#_r097n9p1zudp)

[**8.2 Maintenance Guide 39**](#_gc0pybu87zsf)

[**8.2.1 Overview 39**](#_1j8eb0youjjf)

[**8.2.2 MainMenu 40**](#_2kfnkmz9xe3l)

[**8.2.3 Gameplay Flow 41**](#_68hi4hmoh70n)

[**8.2.4 Answer questions 43**](#_hkta9hwwd6jw)

[**8.2.5 Scanning 3D object 46**](#_cgbcfgpot4wj)

[**8.2.6 Settings 48**](#_esszd9x2qp89)

[**8.2.7 Leaderboards 49**](#_we0y2q2n68j8)

[**8.2.8 Student & Teacher Registration 50**](#_m8zijvbkfyj4)

[**8.2.9 Student data 51**](#_403c14rr5yls)

[**8.2.10 Language controller 52**](#_m6u0i4r1z879)

[**9 Results & conclusions 53**](#_kmz95fpe8i3n)

[**10. References 54**](#_jubotmm35the)

### 

### **Abstract**

Mathematics education often faces the challenge of engaging middle and high school students in understanding abstract concepts, such as geometric theorems and mathematical functions. Traditional teaching methods can struggle to captivate students, relying heavily on static explanations and rote memorization. To address these challenges and create a more dynamic learning experience, this project introduces MathStARz, an innovative augmented reality (AR) mobile application.

MathStARz leverages cutting-edge AR technology to transform the way students interact with mathematics. By integrating 3D interactive models, gamification elements, and multimedia content, the application enables students to explore mathematical concepts in a highly engaging and immersive environment. From visualizing geometric shapes to solving interactive math challenges, MathStARz fosters critical thinking and enhances students' spatial reasoning skills.

At the core of MathStARz are two essential components: AR-based visualization of math concepts and gamified learning pathways. These features empower students to understand complex ideas through hands-on interaction while earning rewards to maintain motivation. Driven by collaboration between educators and developers, MathStARz aims to revolutionize math education, transitioning from passive learning to active exploration. By harnessing the potential of AR, this project seeks to inspire a new generation of learners, equipping them with the skills and confidence to excel in mathematics.

**Keywords**

Augmented Reality (AR), Interactive Learning, Learning Motivation, Mathematics Education, Student Achievement, Virtual Interaction, Educational Technology, Game-Based Learning, Visualization.

### **1. Introduction**

The increasing integration of technology into education has revolutionized traditional teaching methods, introducing tools that enhance student engagement and learning outcomes. Among these technologies, Augmented Reality (AR) has emerged as a significant innovation, offering immersive, interactive experiences that bridge the gap between abstract concepts and material understanding [1]. Studies highlight the potential of AR-based learning environments to facilitate the comprehension of complex subjects like mathematics by combining digital and physical elements [3].

Mathematics is a critical area of education, essential for developing logical reasoning and problem-solving skills [1]. However, students often perceive it as abstract and challenging, particularly in topics such as functions, geometry, and probability [2]. Traditional methods rely on static tools such as textbooks and scientific calculators, which may not sufficiently address students’ difficulties in visualizing and interpreting mathematical concepts [3]. Consequently, researchers advocate for innovative approaches that make mathematics more engaging and accessible [4].

Game-based learning has shown considerable promise in addressing these challenges. It emphasizes active participation, immediate feedback, and problem-solving, creating an engaging learning environment. Comparative studies reveal that game-based learning enhances students' critical thinking, motivation, and academic performance more effectively than traditional methods [4]. Similarly, AR technologies enable students to visualize and manipulate mathematical objects in 3D, making abstract concepts more comprehensible and improving spatial intelligence [3].

This project combines the strengths of game-based learning and AR to develop an educational game aimed at enhancing middle and high school students’ math skills. The game integrates interactive 3D simulations with problem-based scenarios, allowing students to explore and apply mathematical principles in a virtual environment [1]. Research on AR-based math learning applications indicates notable advancements in students’ comprehension of functions, probabilities, and geometric relationships, along with increased enthusiasm and active participation [1].

The focus on interactive learning is supported by research demonstrating the importance of student-centered approaches. By encouraging exploration and experimentation, AR-based games promote a constructivist learning environment where students actively construct knowledge [1]. Furthermore, the gamification of math concepts aligns with educational goals to make learning enjoyable and meaningful, addressing the long-standing issue of math anxiety and lack of involvement among students [4]. By making abstract concepts tangible and learning experiences interactive, this project aims to empower students with the skills and confidence to excel in mathematics while fostering a lifelong appreciation for the subject.

The structure of this document is designed to provide a comprehensive understanding of the project. In **Section 2**, we present a literature review, discussing the challenges in mathematics education, the integration of augmented reality (AR) in learning environments, and the role of gamification in enhancing student engagement. **Section 3** outlines the expected achievements of the project, highlighting the educational goals we aim to accomplish through the development of MathStARz. In **Section 4**, we describe the engineering process, covering the methodologies, system architecture, and design decisions that shaped the application. **Section 5** includes the references that support our research, while **Section 6** provides an appendix with supplementary materials, such as interview transcripts and additional data that contributed to the development of the project.

### **2. Background and Related Work**

### **2.1 Mathematics Education Challenges**

Mathematics education has consistently grappled with the difficulty of connecting abstract concepts to real-world understanding, particularly for middle and high school students. Numerous learners find themselves struggling with topics such as algebra, geometry, and functions, primarily because these subjects can seem overly abstract [1]. This often results in a lack of engagement and diminished motivation among students. In an effort to combat these challenges, educational technologies have introduced innovative tools that promote visualization, interaction, and increased engagement, with Augmented Reality (AR) standing out as a significant solution [2].

### **2.2 Augmented Reality**

Augmented Reality (AR) is an innovative technology that blends digital information with the physical world, creating interactive environments where virtual objects coexist with real-world elements. In educational settings, AR transforms traditional learning methods by providing immersive, hands-on experiences that enhance student engagement and comprehension. Through dynamic visualizations and interactive models, AR allows learners to explore complex concepts, making abstract ideas more tangible and accessible. This technology fosters curiosity, critical thinking, and active participation, positioning AR as a powerful tool for modern education, particularly in subjects that benefit from spatial and conceptual visualization, such as mathematics.

### **2.2.1 Augmented Reality in Education**

Augmented Reality (AR) integrates the real world with virtual elements, superimposing digital information onto physical environments to create engaging and interactive experiences [3]. This technology is particularly beneficial in educational settings, where it enables students to visualize and work with concepts that might otherwise be difficult to understand [3].

### **2.2.2 AR for Learning Mathematical Functions**

AR has shown great promise in teaching mathematical functions, allowing students to visualize the relationships between different variables. For instance, AR applications empower learners to explore linear, quadratic, and trigonometric functions by adjusting parameters in real-time and observing the corresponding changes [1]. These interactive tools help make abstract concepts more tangible, improving both comprehension and retention. One noteworthy example involved students manipulating graphs through AR to see how variations in coefficients affected the shape and behaviors of functions, leading to notable gains in their understanding [2].

### **2.2.3 Enhancing Geometry and Spatial Reasoning**

The fields of geometry and spatial reasoning particularly benefit from the capabilities of AR. This technology allows students to construct, rotate, and analyze three-dimensional shapes, fostering a deeper understanding of spatial relationships [3]. For example, learners can investigate geometric solids, like prisms and cones, from various angles and even navigate through virtual representations [2]. This method effectively bridges the gap between theoretical concepts and their practical applications in the real world [3].

### **2.3 Gamification in Educational**

AR gamification incorporates elements typically found in games—like points, badges, leaderboards, and challenges—into non-gaming contexts, boosting engagement and motivation [8]. When integrated into math education, these techniques simplify abstract concepts, encourage active learning, and sustain student interest over time [8]. Gamified AR environments further enhance the learning experience by offering immediate feedback, fostering a more interactive and personalized approach for students [3]. Additionally, leaderboards introduce a healthy sense of competition, motivating students to improve their performance and remain engaged [4]. These tools are particularly effective in STEM education, where they make abstract concepts more tangible, simplify complex ideas, and promote deeper understanding [3].

### **2.3.1 Gamified AR Applications**

Math games that utilize AR often integrate gamification to spark active participation and keep students interested over time [6]. For instance, a game might involve solving equations to progress through various levels, rewarding correct answers with points or unlocking additional perks. A notable example describes an AR math game where students solve mathematical problems in interactive, real-world-inspired settings. This gamified strategy not only makes learning enjoyable but also sharpens critical thinking and problem-solving abilities [2].

Such applications also provide tools to boost engagement and improve learning outcomes. Gamification elements like challenges, rewards, and dynamic feedback encourage critical thinking and problem-solving skills [4]. However, careful consideration must be given to designing these systems to ensure that gaming mechanics align with educational objectives, maintaining a balance between fun and meaningful learning [4].

### **2.4 Development of AR Applications**

When developing educational AR applications, selecting the appropriate platforms and tools is crucial. Among the prominent choices are Unity with ARCore, Vuforia, and WebGL, each offering distinct advantages.

Unity, combined with ARCore, serves as a robust framework for creating interactive 3D experiences [5]. Unity 3D has a handy module on digital scenes, 3D objects, and programming logic, making it ideal for AR applications [5]. While Unity excels in building dynamic environments, ARCore enhances these applications with advanced AR capabilities tailored for Android devices [2] (see figure 1).

For applications that require precise image recognition and object tracking, Vuforia stands out. Its strong computer vision capabilities make it an excellent choice for AR projects focused on physical interaction accuracy [3]. Vuforia’s advanced image recognition features also support educational AR applications, enhancing interactive learning experiences [7].

Alternatively, WebGL offers the flexibility to develop AR experiences that are accessible directly through web browsers. This eliminates the need for specialized hardware and ensures compatibility across various platforms [2], [3].

Ultimately, the selection of a platform depends on key factors such as device compatibility, the specific features required, and the intended user experience.

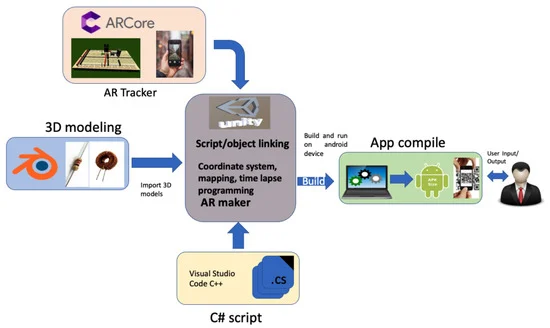


Figure 1- AR application developed in Unity using ARCore [9].

### 

### **2.5 Future Directions in AR and Gamified Learning**

Augmented Reality (AR) and gamification have shown incredible potential, but there are still hurdles to overcome. Issues like limited access to devices and challenges in software development can hinder the widespread adoption of these technologies. Moreover, there is a pressing need to integrate AR tools effectively into current curricula and to train educators so they can utilize these innovations proficiently [5]. Research efforts should prioritize developing scalable and cost-effective solutions that utilize AR to foster inclusive and accessible learning environments [7]. As AR technology continues to advance, we can look forward to exciting possibilities in transforming mathematics education. By fusing gamification with AR, teachers can create immersive, engaging, and impactful learning experiences that motivate students to delve deeper into mathematics and excel in their studies [4], [5], [7].

### **3. AR Game Design and Functionality**

#### **3.1 MathStarz AR Educational Game**

Our project, *MathStarz AR*, is an educational application developed in Unity using Vuforia, designed to transform the way students interact with geometry through Augmented Reality (AR). Aimed primarily at high school learners, the application merges physical interaction with digital content to create an engaging and immersive math learning experience.

The app includes three primary components:  
**Interactive AR Game**: At the core of the application is an AR-powered game where users scan 3D model target shapes (e.g., a cube). Once the shape is recognized, a 3D environment is rendered in AR, featuring non-playable characters (NPCs) who present math questions related to the scanned shape. For each correct answer, students earn points and receive a piece of a 2D puzzle image. As students progress, they collect all the puzzle pieces to complete the image, adding a visual and rewarding element to the learning process.  
**Learning Progress Tracker**: To support educational goals, the app tracks each user’s progress—including points earned per shape, levels reached, and correct answers. This information helps students and teachers monitor performance and identify areas for improvement.  
**Gamification Features**: The experience is enhanced with gamification elements such as points, levels, collectible puzzle pieces (2D images), and an optional leaderboard. These features increase motivation and provide tangible rewards for learning and problem-solving.

An admin panel is available via Unity Cloud for managing app functionality:  
**User Management Tools**: Admin users can register and manage teacher accounts, while teacher users can register and manage their students. Each teacher can search for a student by ID and view detailed statistics, including the student’s score across different geometric shapes. This monitoring is made possible by storing user progress data in MongoDB Atlas, allowing real-time access and analysis of student performance.

#### **3.2 Development and Implementation Process**

The development of *MathStarz AR* was guided by an iterative, user-centered design strategy. To ensure that the application would meet both educational goals and user expectations, the process combined classroom insights, academic literature, and rapid prototyping. At every stage, we aimed to translate best practices from AR-based and game-based learning into practical, engaging features suitable for high school students.

#### **3.2.1 Identifying Learning Challenges through Educator Insight**

To better understand the real needs of our target audience, we conducted an interview with Ariel, an experienced high school math teacher. Ariel shared that one of the key challenges in teaching geometry is helping students visualize and understand the properties of 3D shapes using only static diagrams from textbooks. He noted that many students struggle to make the connection between theoretical formulas and real-world shapes, which often leads to disengagement.

#### **3.2.2 Designing with Research-Based Insight**

To ensure pedagogical effectiveness, we reviewed a range of studies on AR and digital game-based learning [6]. We defined a feature set aligned with the following goals:

* **Shape-Based Interaction**: Using Vuforia, students scan a physical 3D model (e.g., cube or pyramid), triggering an interactive AR learning session.
* **NPC-Based Questioning**: Inside the AR environment, students interact with NPCs that ask math questions specifically related to the scanned shape.
* **Puzzle Progression System**: Correct answers reward students with pieces of a 2D puzzle image. Completing the image provides a sense of progress and achievement.
* **Gamified Learning**: Points, and visual rewards are used to keep students engaged and intrinsically motivated, following Self-Determination Theory principles [10].

The system also includes role-based tools for teachers and administrators to register users, track performance across geometric categories, and manage educational content—all built on top of MongoDB Atlas for cloud data storage and retrieval.

#### **3.2.3 Requirements**

Most of the functional requirements were gathered from our target users (students and teachers), while the majority of the non-functional requirements were determined by us as developers.

##### **The main functional requirements of our system are:**

* The system should utilize AR technology to present math concepts in a 3D interactive environment.
* The system should include gamification elements such as points, badges, and leaderboards to motivate students.
* The system should support scanning physical objects (e.g., markers or cards) to unlock math challenges.
* The system should instruct users on how to align their device for proper AR tracking and interaction.
* The system should support playing audio explanations for math problems and solutions.
* The system should provide text-based explanations for math exercises.
* The system should provide a feedback form for users to submit their opinions or report issues.
* The system should display a structured menu of math topics and exercises for easy navigation.
* The system should allow users to control sound settings such as audio volume and background music.

##### **The main non-functional requirements of our system are:**

* **Accessibility**: English and Hebrew will be supported as primary languages.
* **Usability**: The main control buttons will include both text and icons for clarity.
* **Usability**: A fixed navigation menu will always be available for quick topic access.
* **Platform**: The application will be available for both Android that support ARCore.
* **Development Environment**: Unity will be used as the primary development environment, with additional tools like Blender for 3D modeling.
* **Documentation**: The system will be fully described in this work, including a developer manual and a user manual.
* **Usability**: Sound settings will include sliders ranging from 0 to 1 for volume control.
* **Maintainability**: All assets, including 3D models, should be modular and reusable for future updates.

#### 

#### 

#### **4 Diagrams**

#### **4.1 System Architecture** Our system architecture is designed with a three-layer structure — **Application Layer, Logic Layer, and Backend Layer** — enabling seamless **AR** interaction, real-time data processing, and scalable data management to support an engaging educational experience (see figure 2).

#### **The Application Layer**, this layer is built using Unity and integrated with the **Vuforia SDK** to deliver interactive augmented reality experiences. It manages all user inputs, visual content, and gameplay feedback. Key components include:

#### **AR Visualization**: Displays 2D and 3D geometric shapes in AR space based on scanned image targets.

#### **GUI**: Provides intuitive navigation through screens like login, main menu, leaderboard, settings, and in-game panels.

#### **Gamification Elements**: Includes scoring, puzzle collection, and leaderboards to enhance engagement and motivation.

#### **Question Module**: Presents math questions related to scanned geometry shapes or triggered by interaction with NPCs.

#### **Collection UI**: Shows the puzzle pieces the student has collected so far.

#### Students scan Model targets (cube, pyramid, eg.) to trigger AR content, interact with NPCs, and answer math-related questions to earn puzzle pieces and score points.

#### **The Logic Layer** serves as the core intelligence of the **Unity** application. It manages the game's internal behavior, flow, and decision-making based on user input. Once the **AR** world is activated, this layer handles scene transitions, puzzle logic, and user progress tracking. It retrieves questions from the backend, determines the current puzzle state, and manages the interaction logic between the player and the NPCs. Real-time events such as scanning, answering, and dragging puzzle pieces are processed here. The Logic Layer is also responsible for calculating scores, updating the user interface based on the player's progress, and ensuring that the gameplay responds dynamically to the student’s performance. It ensures that collected puzzle pieces, completed questions, and scores are synchronized with the backend to maintain consistency across sessions.

#### **The Backend Layer** is implemented using **FastAPI** and serves as the central server-side system, deployed on **Render**. It handles all communication between the **Unity** client and the database. This layer is responsible for authenticating users and managing their roles as students, teachers, or admins. It processes user registration requests, ensuring that admins can create teacher accounts and teachers can register new students. When the game requests questions or puzzle data, the backend delivers content based on the player’s current progress and configuration. It also receives updates from Unity regarding collected puzzle pieces and scores, which it stores persistently in **MongoDB**. This allows teachers to monitor student progress through dashboards that reflect shape-specific performance and total points. The backend thus maintains data integrity, ensures secure access, and enables dynamic content delivery to the **AR** application.

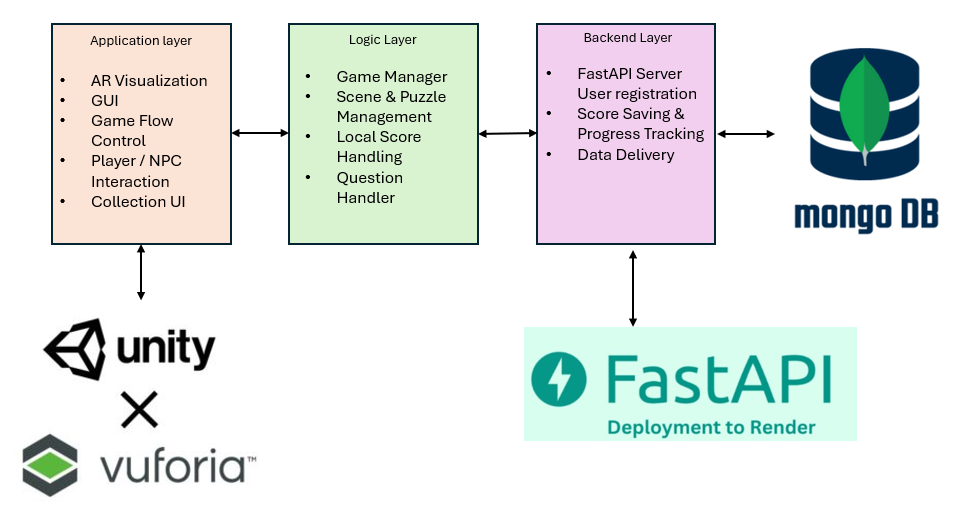


Figure 2 - System Architecture

### 

### **4.2 Workflow diagram**

**Phase A:**

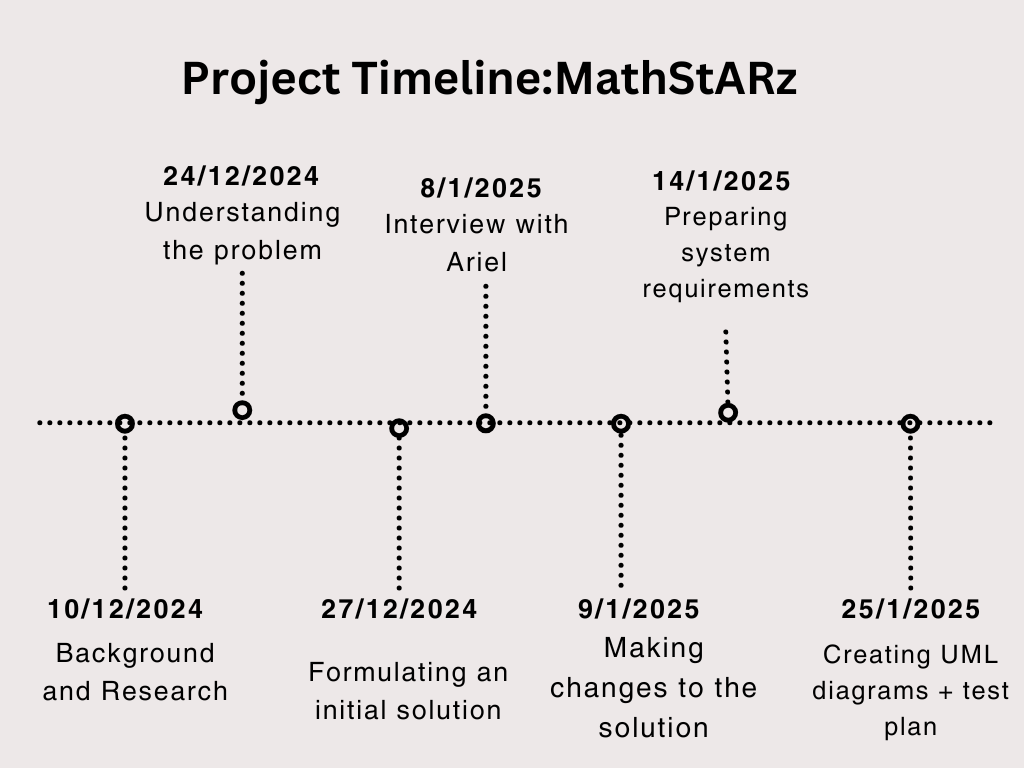


Figure 3 - workflow diagram phase A

**Phase B:**

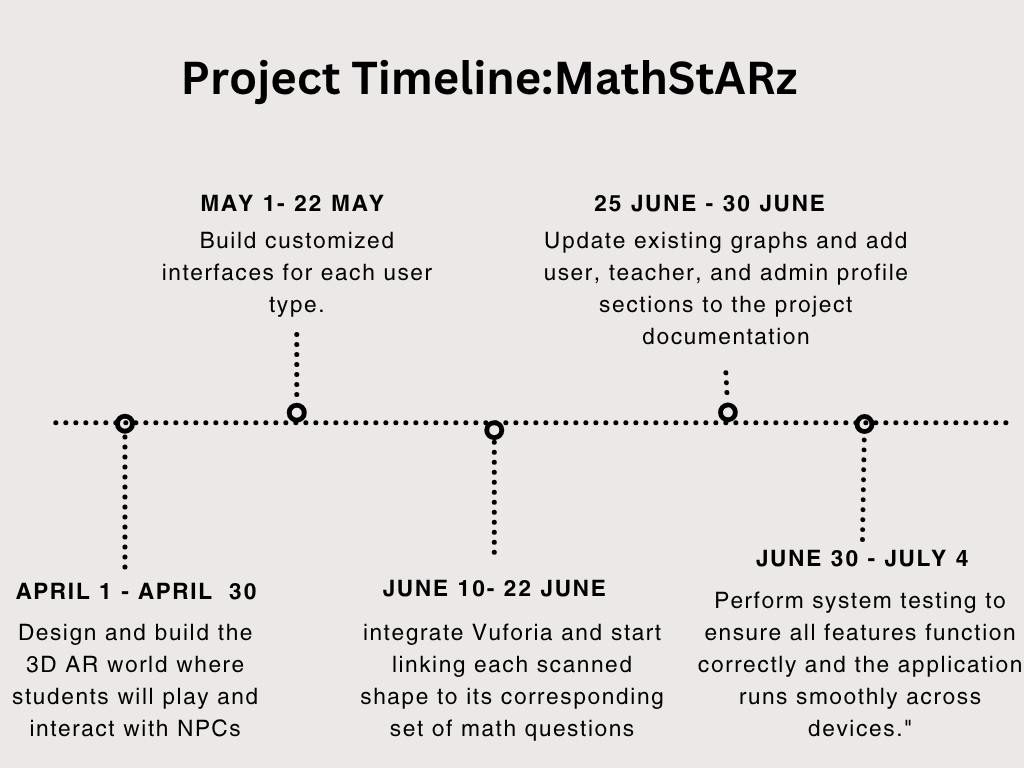
****

Figure 4 - workflow diagram phase B

### **4.3 Use Case diagram**

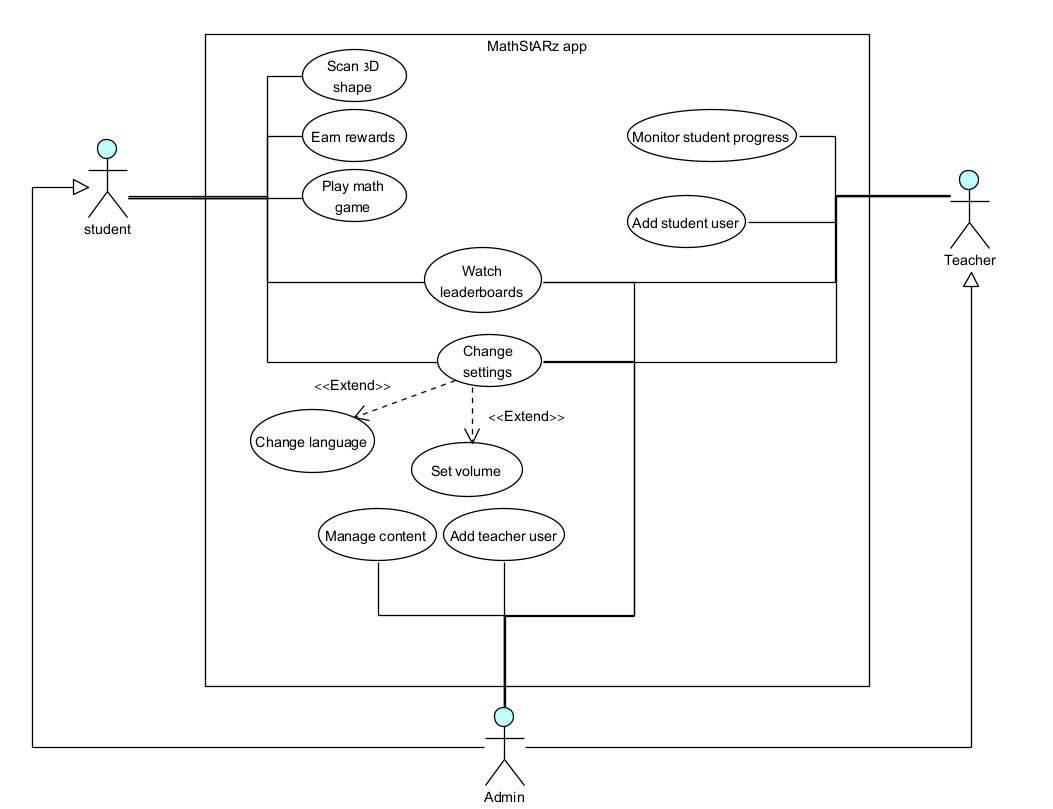


Figure 5 - Use Case diagram

| **Use case** | **Play math game** |
| --- | --- |
| **Description** | Start the AR game and solve geometry questions. |
| **Actors** | Student |
| **Triggers** | Students click 'New Game'. |
| **Successful Scenario** | 1. The student clicks the 'New Game' button.  2. The camera opens and scans the image target.  3. The AR world becomes visible and NPCs appear.  4. The student moves and interacts with an NPC.  5. The system displays a geometry question to solve. |
| **Alternative Scenario** | - |

Table 1 - Play math game

| **Use case** | **Earn rewards** |
| --- | --- |
| **Description** | ward the student for answering questions correctly. |
| **Actors** | Student |
| **Triggers** | Correct answer is submitted. |
| **Successful Scenario** | 1. The student submits a correct answer.  2. The system updates the score.  3. A puzzle piece is revealed in the collection panel.  4. Students receive positive feedback (visual/sound). |
| **Alternative Scenario** | - |

Table 2 - Earn rewards

| **Use case** | **Change Settings** |
| --- | --- |
| **Description** | Open application’s settings view to modify preferences |
| **Actors** | Student / teacher / admin |
| **Triggers** | "Settings" icon is clicked. |
| **Successful Scenario** | 1. The system opens the settings view with categories like audio, visuals, and language.  2. The student adjusts preferences, such as enabling or disabling background music or selecting a preferred language.  3. The system saves the changes and displays a confirmation message.  4. The student exits the settings view and returns to the previous screen. |
| **Alternative Scenario** | - |

Table 3 - Change settings

| **Use case** | **Watch leaderboards** |
| --- | --- |
| **Description** | View the top scores of students. |
| **Actors** | Student / teacher / admin |
| **Triggers** | "Leaderboards" icon is clicked. |
| **Successful S**  **cenario** | 1. The leaderboard panel opens.  2. The system retrieves the top 5 student scores from the backend.  3. The leaderboard is displayed in descending order. |
| **Alternative Scenario** | - |

Table 4 - Watch leaderboards

| **Use case** | **Monitor student progress** |
| --- | --- |
| **Description** | Allow teachers to view student progress per shape and score. |
| **Actors** | Teacher |
| **Triggers** | The teacher selects a student ID from the dashboard. |
| **Successful Scenario** | 1. The teacher logs into the app and opens the student dashboard.  2. The system shows a list of students and their progress.  3. The teacher selects a student and views shape-specific performance and collected pieces. |
| **Alternative Scenario** | - |

Table 5 - Monitor student progress

| **Use case** | **Sign student user** |
| --- | --- |
| **Description** | Register a new student user. |
| **Actors** | Teacher |
| **Triggers** | The teacher fills the registration form. |
| **Successful Scenario** | 1. The teacher accesses the registration screen.  2. Fills in student details (name, username, etc.). 3. The system validates and saves the new student record.  4. Confirmation is shown and students can now log in. |
| **Alternative Scenario** | - |

Table 6 - Sign student user

| **Use case** | **Sign teacher user** |
| --- | --- |
| **Description** | Register a new teacher user. |
| **Actors** | Admin |
| **Triggers** | Admin fills the registration form. |
| **Successful Scenario** | 1. Admin opens the 'Add Teacher' panel.  2. Admin enters teacher credentials and details.  3. The system creates the teacher account and saves it to the database.  4. A confirmation message is displayed. |
| **Alternative Scenario** | - |

Table 7 - Sign teacher user

| **Use case** | **Manage Content** |
| --- | --- |
| **Description** | Admin manages and updates geometry-related questions and answers used in the game. This feature is accessed through the Unity editor or admin development tools, not via the in-game UI. |
| **Actors** | Admin |
| **Triggers** | Admin runs a Unity editor tool or accesses a special content management scene during development. |
| **Successful Scenario** | 1. Admin opens the Unity project and navigates to relevant scenes or UI prefabs.  2. Admin updates UI elements (e.g., canvas layout, button logic, panels, localization text).  3. Admin verifies that question text fields are dynamically linked to the database.  4. Admin ensures that score and puzzle data bind correctly to backend responses.  5. Admin saves changes and runs tests in Unity to validate behavior. |
| **Alternative Scenario** | - If UI elements are misaligned or missing bindings, Unity logs visual or console errors.  - If the database connection is misconfigured, in-game data fails to load and must be re-linked. |

Table 8 - Manage content

### 

| **Use case** | **Scan 3D shape** |
| --- | --- |
| **Description** | Detects and recognizes a 3D shape using the device camera to trigger related math activities. |
| **Actors** | Student |
| **Triggers** | The student points the device camera at a supported physical shape. |
| **Successful Scenario** | 1. The student opens the scan feature.  2. The camera detects and recognizes the 3D shape using Vuforia.  3. The system activates the corresponding AR world object.  4.The math game related to the shape is initialized. |
| **Alternative Scenario** | - |

### Table 9 - Scan 3D shape

| **Use case** | **Change volume** |
| --- | --- |
| **Description** | Adjust the application’s sound level through the settings menu. |
| **Actors** | Student / Teacher / Admin |
| **Triggers** | The user navigates to the settings panel and selects the volume option. |
| **Successful Scenario** | 1. The user opens the settings panel.  2. The user selects "Set volume".  3. The system updates the sound level immediately and provides audio feedback if necessary. |
| **Alternative Scenario** | - |

Table 10 - Change volume

| **Use case** | **Change language** |
| --- | --- |
| **Description** | Switch the application’s language between supported options (English/Hebrew). |
| **Actors** | Student / Teacher / Admin |
| **Triggers** | The user navigates to the settings panel and selects the language option. |
| **Successful Scenario** | 1. The user opens the settings panel.  2. The user selects "Change language".  3. The system updates all UI texts to the selected language immediately. |
| **Alternative Scenario** | - |

Table 11 - Change language

### 

### **4.4 Activity diagrams**

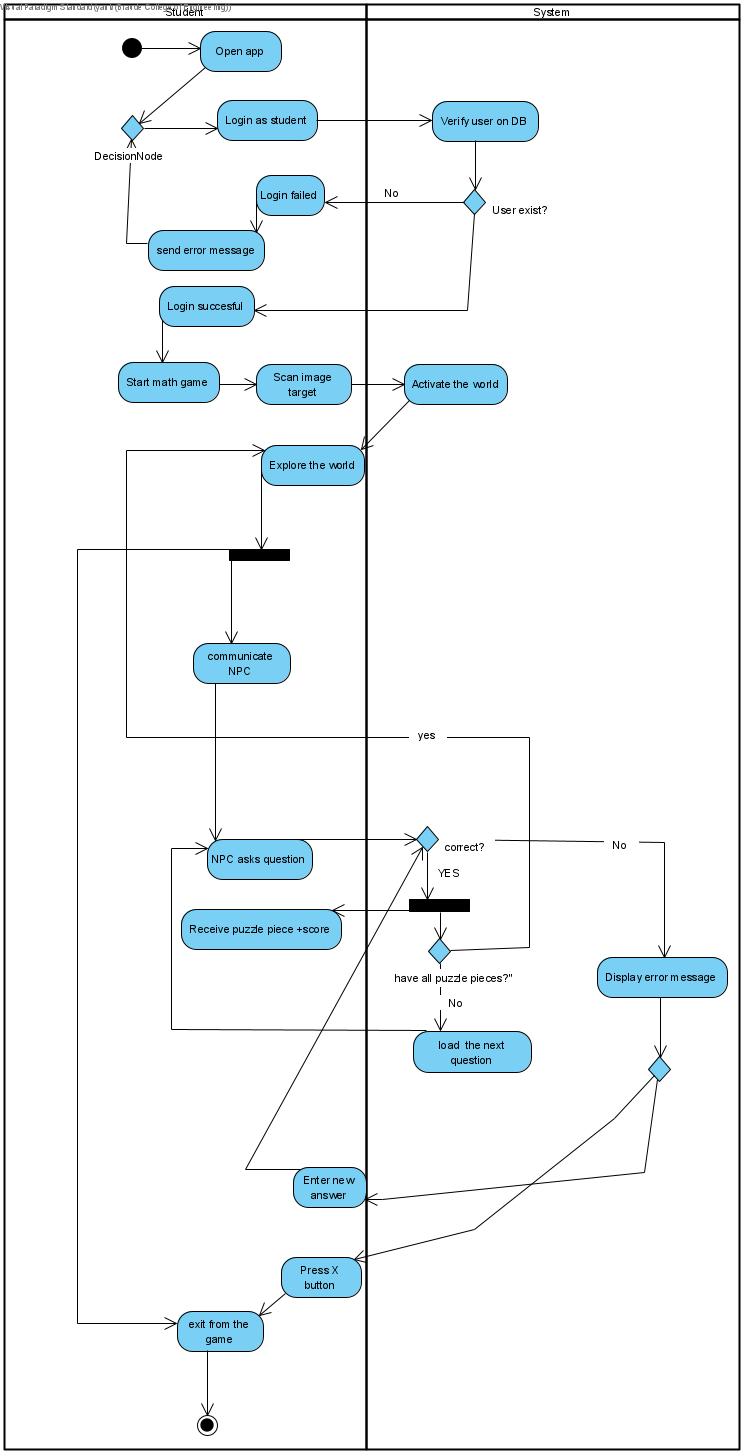


Figure 6 - Student Activity diagram

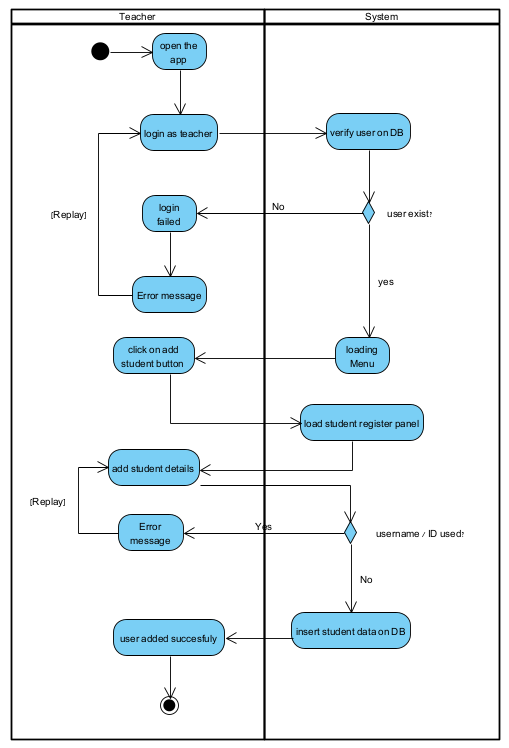


Figure 7 - Teacher activity diagram

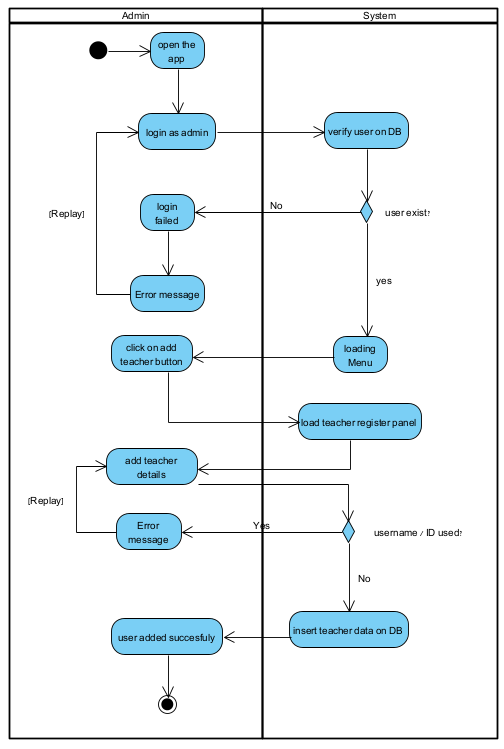


Figure 8 - Admin activity diagram

### **4.5 Package diagram**

**This package diagram illustrates the architecture of our AR-based educational application developed using Unity and Vuforia.** The user interacts with the system through a mobile device by scanning real-world image targets and engaging with on-screen elements. Unity is responsible for rendering the interface, managing player input, and orchestrating gameplay logic through dedicated script managers. Vuforia handles augmented reality functionality, including image recognition and activation of 3D puzzle environments. The system communicates with a FastAPI backend to handle authentication, serve geometry-related questions, update puzzle progress, and store user scores. Admins and teachers interact with the backend to manage users and monitor student performance. All in-game logic relies on modular data models to ensure consistent synchronization between the frontend and backend. This layered structure promotes scalability, maintainability, and a dynamic, real-time learning experience (see figure 9).

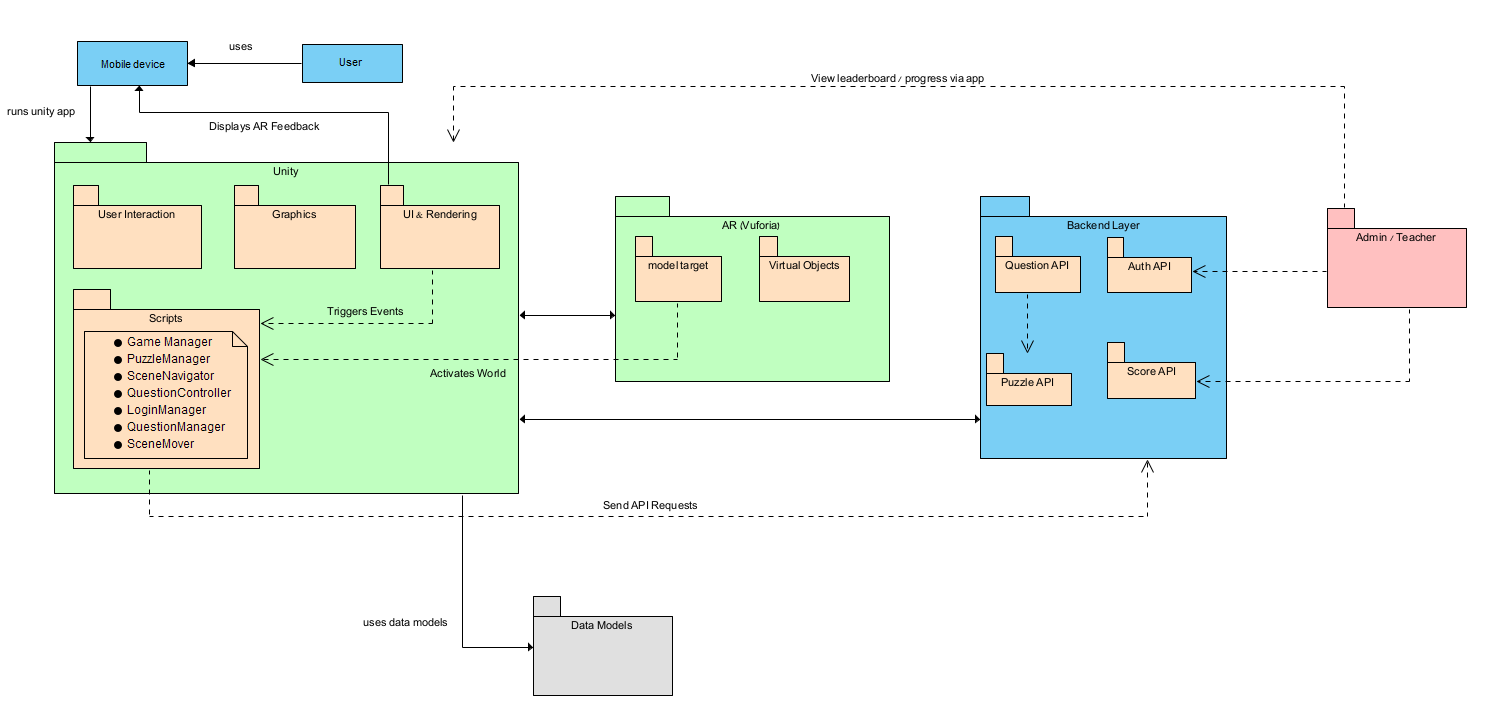


Figure 9 - package diagram

### **5. Testing**

As development of the MathStarz AR application progressed, we conducted formal **acceptance testing** to ensure the system met its functional and non-functional requirements. This process included evaluating key components such as user interaction flow, sound settings, gamification features, and AR functionality. Each requirement was mapped to specific test cases, and the outcomes were recorded to validate the system’s readiness for use in an educational setting.

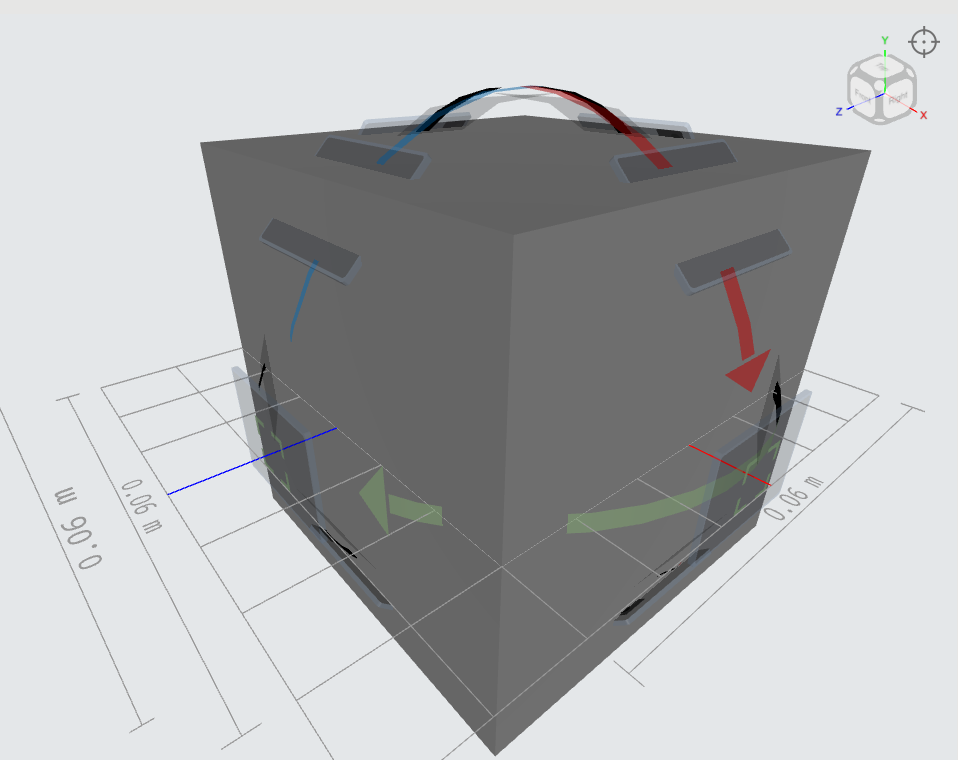
Table 12 presents the results of these acceptance tests, highlighting which requirements were successfully met and identifying areas where further improvement was needed. This systematic approach allowed us to prioritize final adjustments such as refining interaction timing, improving instruction clarity, and enhancing visual feedback to deliver a more intuitive and engaging learning experience.

| **ID** | **Feature** | **Description** | **Excepted result** | **Status** |
| --- | --- | --- | --- | --- |
| **1** | **Register Student / Teacher** | Teacher / Admin adds a new student to the system | Student / teacher user added to the users collection on MongoDB. | Completed |
| **2** | **Search by Student ID** | The teacher searches for a student by ID. | Student’s progress and scores per shape are displayed. | Completed |
| **3** | **Scan 3D Model** | Students scan a physical 3D shape (e.g., cube). | The AR environment loads correctly and responds to the shape. | Completed |
| **4** | **Interact with NPC** | Student touch click on NPC. | A dialog box appears and communicates with the student. | Completed |
| **5** | **Receive Math Question** | The NPC asks the user a question. | Question appears on the UI dialog box. | Completed |
| **6** | **Answer correctly and get rewards** | Students answer the correct answer to the question. | A visual feedback will appear, the student gets 10 score points and a puzzle piece. | Completed |
| **7** | **Answer wrong answer** | Students answer the wrong answer to the question. | A visual feedback will appear, and the student needs to fix his question. | Completed |
| **8** | **Click on leaderboards button** | Users click on the leaderboards button. | The Leaderboard table appears. | Completed |
| **9** | **Change volume** | Users change the volume value. | Volume value changed. | Completed |
| **10** | **Change language** | Users change language. | System language changed. | Completed |
| **11** | **User login** | User login to his user. | Move to the user type menu. | Completed |
| **12** | **Complete puzzle** | The NPC interacts with the user and tells him he completed the puzzle. | A visual text box appears and tells the user he completed the puzzle. | Completed |

Table 12 - Acceptance test table

### **6. Development Process**

We developed an AR-based educational puzzle game using Unity and the Vuforia SDK as the frontend, with a FastAPI backend connected to a MongoDB Atlas database. The system was built with a three-layer architecture: Application Layer, Logic Layer, and Backend Layer. We began by gathering requirements and designing the system structure and user flows. The frontend was built in Unity, featuring an interactive AR environment triggered by scanning image targets. As part of the AR experience, we also trained 3D objects models using Vuforia’s Target Management System (MTG) to enable real-world object recognition and interaction (see figure 10). The user interface was designed to support three user roles: Student, Teacher, and Admin, each with different permissions and functionality. The backend, built with FastAPI and deployed on Render, handles authentication, question management, puzzle tracking, and score storage. We used RESTful APIs for communication between Unity and the server. The development was iterative, with regular testing and integration of each feature before deployment.



-Figure 10 - Vuforia’s MTG guide view full 360°

### **6.1 System Logic and Backend Flow**

The backend of the **MathStARz AR** system was developed using **FastAPI** (Python) deployed on Render and serves as the core engine for handling user authentication, progress tracking, question management, and game logic. All data is stored and managed using **MongoDB Atlas**, and the backend communicates with the Unity-based frontend via a set of structured **RESTful APIs**.

The system is designed with clear logic flows that ensure correct decision-making, robust error handling, and role-based access for students, teachers, and admins.

### **6.1.1 User Management and Authentication**

The system allows users to register and log in with a username and password. Upon a successful login, the user's isLoggedIn status is updated in the database. Teachers and admins have extended permissions compared to students, allowing them to register users and retrieve student data.

* POST /users/register: Registers a new user after checking for username uniqueness.
* POST /users/login: Verifies credentials and marks the user as logged in.
* POST /users/logout: Sets isLoggedIn to False.
* GET /users/is-logged-in: Checks a user’s login status.
* GET /users/exists: Checks whether a given username already exists.
* GET /users: Retrieves user data by ID or username.
* POST /users/update: Updates full user profile.

### **6.1.2 Puzzle Progress and Game Interaction**

As students progress through the game by scanning shapes and answering questions, their puzzle completion status is updated and stored. Each puzzle represents 4 questions of the shape, and correct answers reveal puzzle pieces (as 2D images).

* POST /puzzles/update: Updates a student's puzzle progress after answering a question.
* GET /puzzles/get: Retrieves the progress for a specific puzzle based on userId and puzzleId.
* GET /puzzles/user: Retrieves all puzzle progress records for a specific student.

### **6.1.3 Question Handling**

Questions are assigned dynamically based on the shape scanned in the AR experience. Each question is tied to a unique ID and retrieved from the backend.

* GET /questions/{qid}: Retrieves a question by its ID using the internal logic function handle\_get\_question().

### **6.1.4 Leaderboard Functionality**

To encourage motivation and competitiveness, the system includes a leaderboard that displays the top students based on total score.

* GET /users/leaderboard: Retrieves the top 5 students ranked by score.

#### **Architecture Overview**

The backend follows a modular structure with clean separation between:

* **Routing layer** (API endpoints),
* **Logic layer** (functions for user validation, scoring, etc.),
* **Database access layer** (queries to MongoDB).

This structure promotes maintainability, scalability, and clear logic flow. All routes are stateless, and communication with Unity is handled through JSON-based HTTP requests and responses.

### **6.2 Challenges**

**Challenges and How We Solved Them:**

One major challenge in developing the AR game was maintaining stable placement of virtual objects. Initially, objects followed the camera instead of staying anchored in the real world. We solved this by organizing the Unity hierarchy correctly and using world-relative positioning for spawned objects. Another key challenge was synchronizing the game state between Unity and the backend, especially for puzzle piece collection and score tracking. We implemented a reliable data model where user progress is uploaded to the database after each interaction and validated on the server. The game fetches questions directly from the database, allowing for real-time updates and flexibility. Additionally, we developed a dynamic question loading system that adapts to the user's progress and supports teacher-submitted questions.

### **Key Decisions:**

We decided to:

* Use **FastAPI** for its simplicity and performance in building RESTful APIs.
* Store data in **MongoDB Atlas** to support flexible schema and scalable cloud access.
* Separate the Unity logic into a **three-layer structure** to keep the architecture modular and maintainable.
* Implement **Vuforia** for real-time AR recognition to trigger game states.

### **Tool we used:**

* **Unity + Vuforia SDK** – AR and game environment.
* **Visual Studio code** – C# scripting in Unity.
* **FastAPI + Uvicorn** – Backend development.
* **MongoDB Atlas** – Cloud-based NoSQL database.
* **Render** – Backend deployment.

### **6.2.1. Challenges and solutions**

**Challenge:**One of the main engineering challenges was ensuring that virtual objects, such as puzzle elements and NPCs, remained anchored in the real world after scanning a model target. Initially, objects followed the camera, breaking immersion.

**Solution:**We resolved this by detaching the spawned objects from the AR camera hierarchy and anchoring them to a world-relative position using Vuforia's ObserverBehaviour. This ensured that objects remained stable and correctly aligned in the physical space after detection.

**Challenge:**

Our system uses 2D puzzle images sliced into four equal square pieces, and each student collects pieces by answering questions correctly. The challenge was designing a scalable way to track which students collected which pieces — especially as the number of users and puzzles grows — while ensuring data consistency and fast retrieval for the UI.

**Solution:**  
 Instead of embedding puzzle data directly under each user, we created a separate collection in the database where each document includes:

* userId – the unique identifier of the student.
* puzzleId – the identifier of the puzzle.
* piecesCollected – an int value that indicates how many puzzle pieces are already collected.

### **7. Conclusions**

As we reached the final stages of the project, we took time to reflect on the development process and the outcomes we achieved. One of the key takeaways was the importance of making design choices that aligned with real-world constraints. Since the game was intended for use in students’ homes, we faced an early challenge in choosing an image target that would be accessible to all users. After experimenting with barcode detection and encountering limitations in object stability, we pivoted to using a QR code as a custom Vuforia image target—a decision that significantly improved object anchoring and gameplay consistency.

Throughout the development, we encountered technical and architectural challenges, such as ensuring consistent puzzle progress tracking, managing asynchronous communication with the backend, and maintaining user-friendly interaction in an AR environment. Addressing these required careful backend structuring and real-time synchronization logic, which we implemented using modular scripts in Unity and well-structured API endpoints in FastAPI. These decisions helped us maintain a clean and scalable architecture across both the frontend and backend.

Moreover, one of the strongest aspects of our approach was our commitment to a layered and modular design. By separating the system into distinct layers—Application, Logic, and Backend—we were able to test, update, and expand each part of the project independently. This allowed for better debugging, team collaboration, and future scalability.

In retrospect, one area we would improve is the timing of our technology validation. Had we performed earlier tests on AR tracking behavior and image target performance, we could have avoided initial setbacks. Still, these challenges provided valuable learning opportunities and helped us grow as developers.

Ultimately, this project demonstrated the importance of flexible planning, thoughtful architecture, and adaptability in the face of unexpected issues. Our ability to pivot when needed, collaborate effectively, and build on a solid codebase played a crucial role in successfully delivering an engaging and educational AR experience for students.

### 

### **8. User Documentation**

### **8.1 User Guide**

**8.1.1 player Settings Screens Overview**

**01:Login Screen**



**Steps to log into the game:**

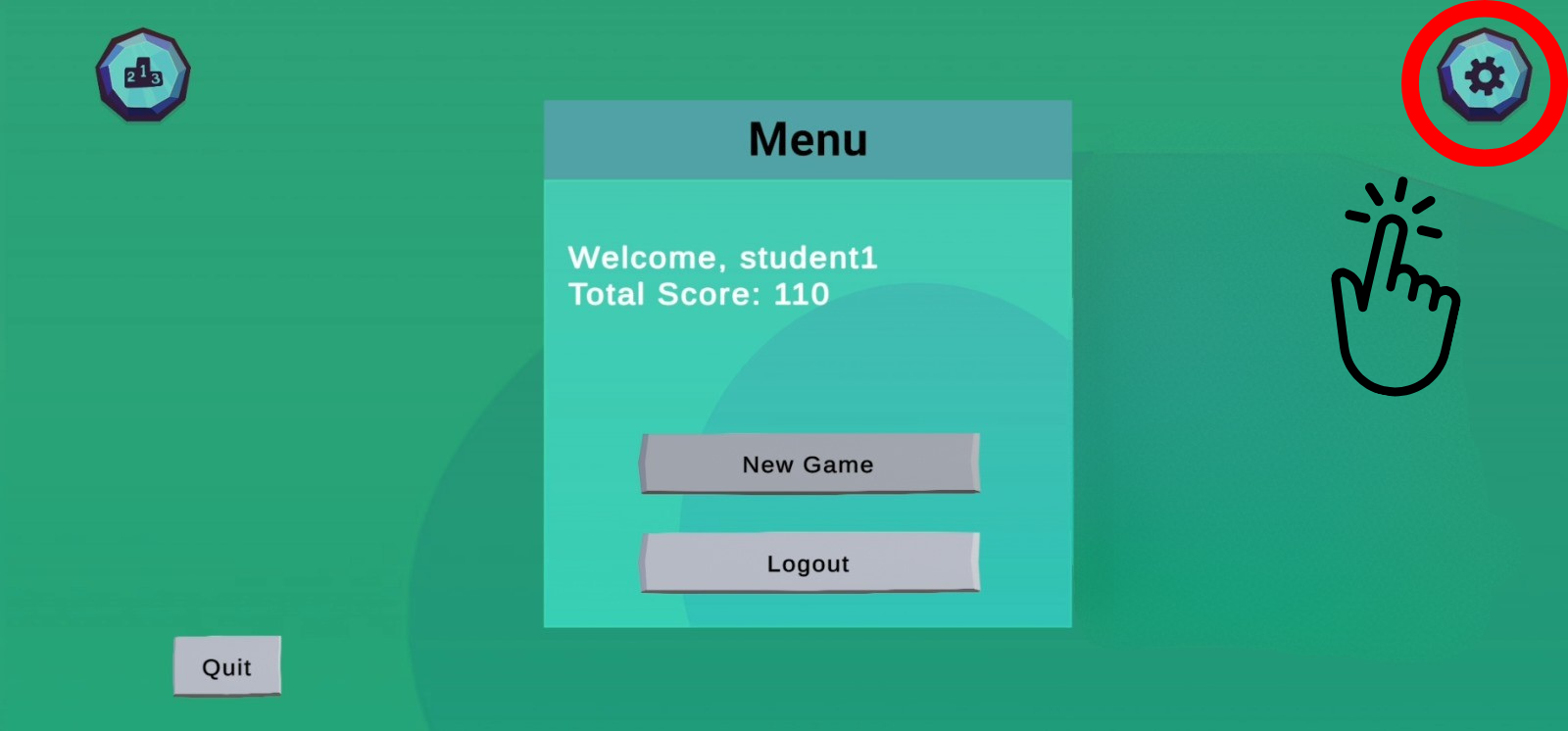
Enter your username in the top field.

Enter your password in the field below.

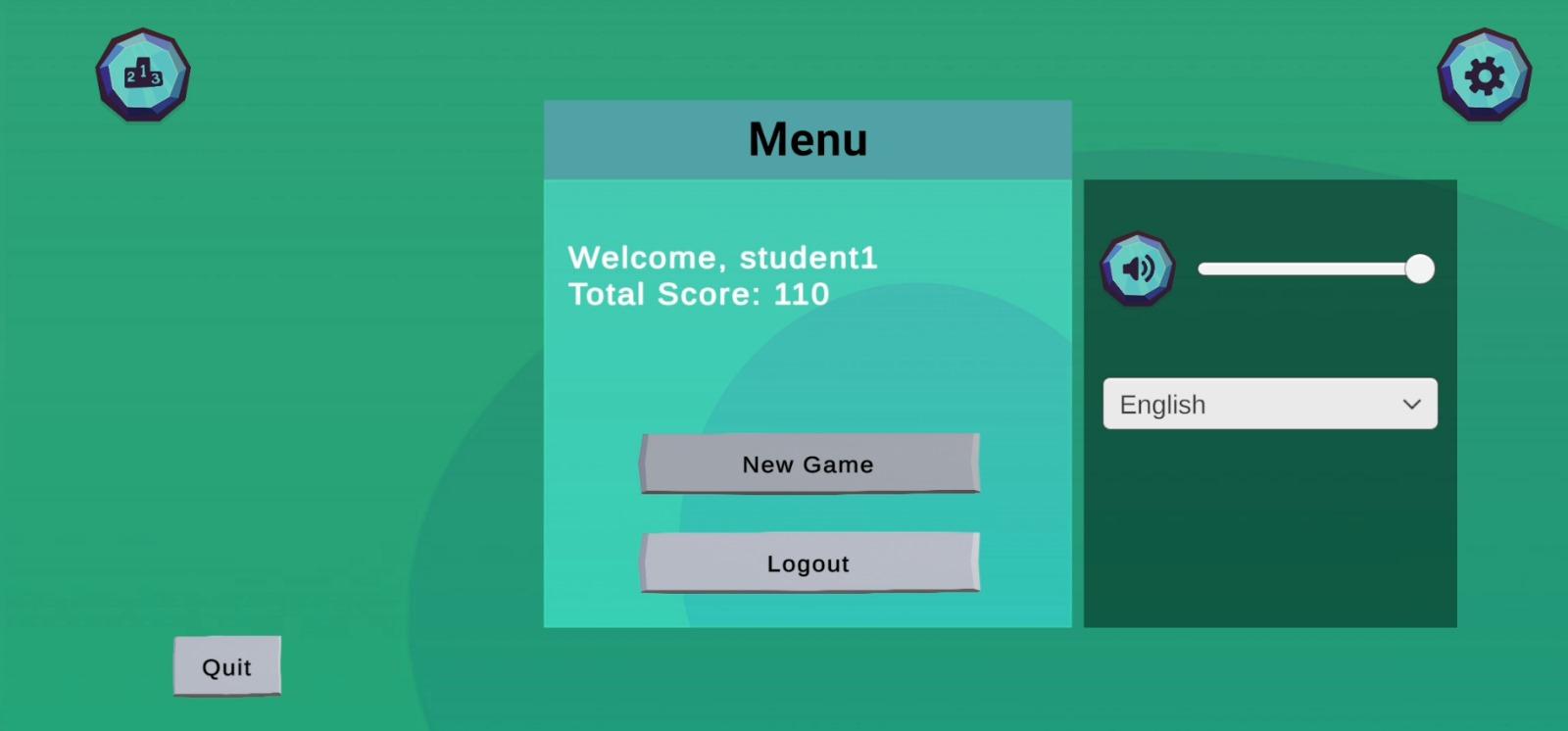
Tap the “Login” button to access the game.

To exit the game, tap the “Exit” button at the bottom left.

**02:Settings**



Located on the right side of the screen.press on it.

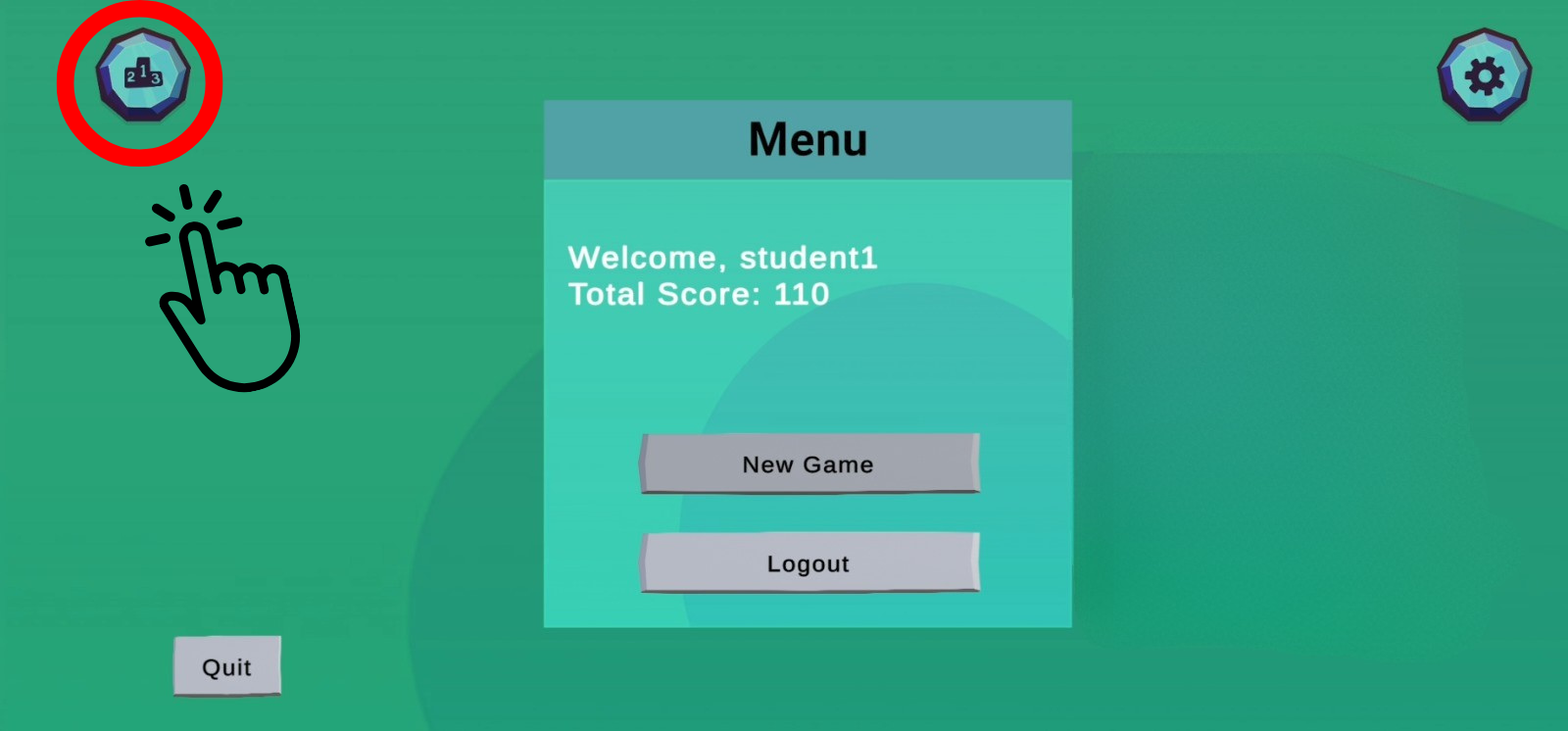


Includes:

Volume Slider – Adjust the game sound.

Language Dropdown – Select your preferred language (English, Hebrew).

**03:Leaderboard**



By pressing on the trophy icon at the top-left, the Leaderboard opens.



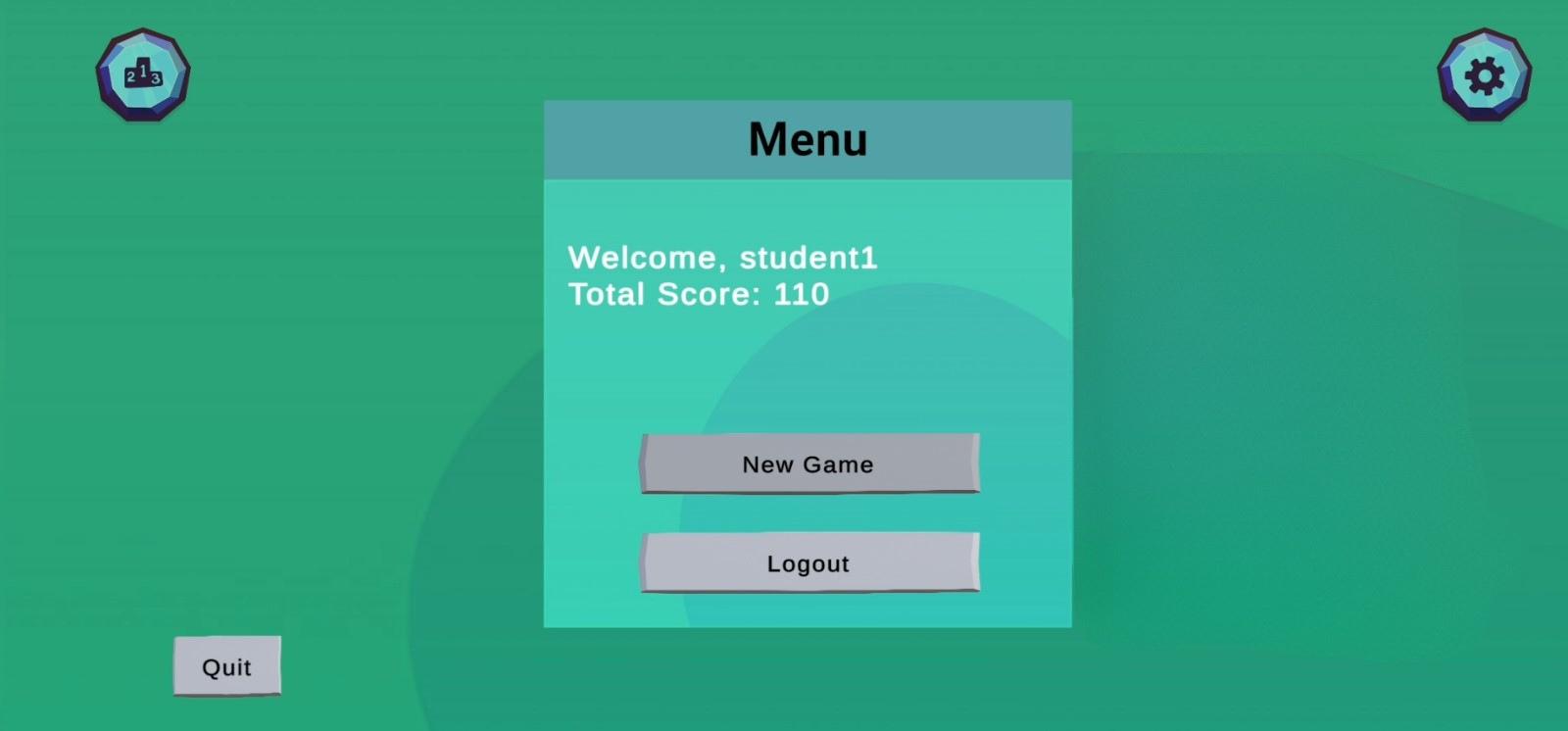
Displays the top 5 players ranked by score.

Each entry shows: Position,Username,Username.

Use this to see how you rank among other players.

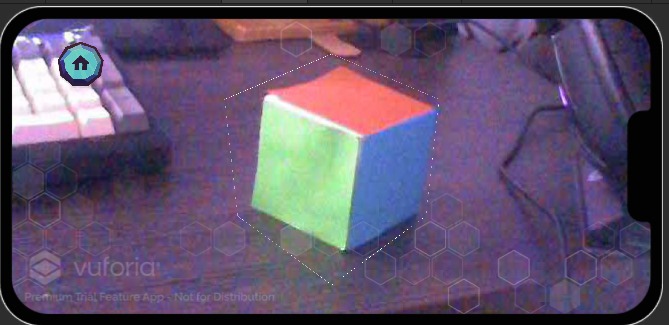
**8.1.2 Game Instructions for Players**

**01:Menu Screen**

****

From the main menu, press the “New Game” button to begin your adventure.

**02:Scan a Physical Object**



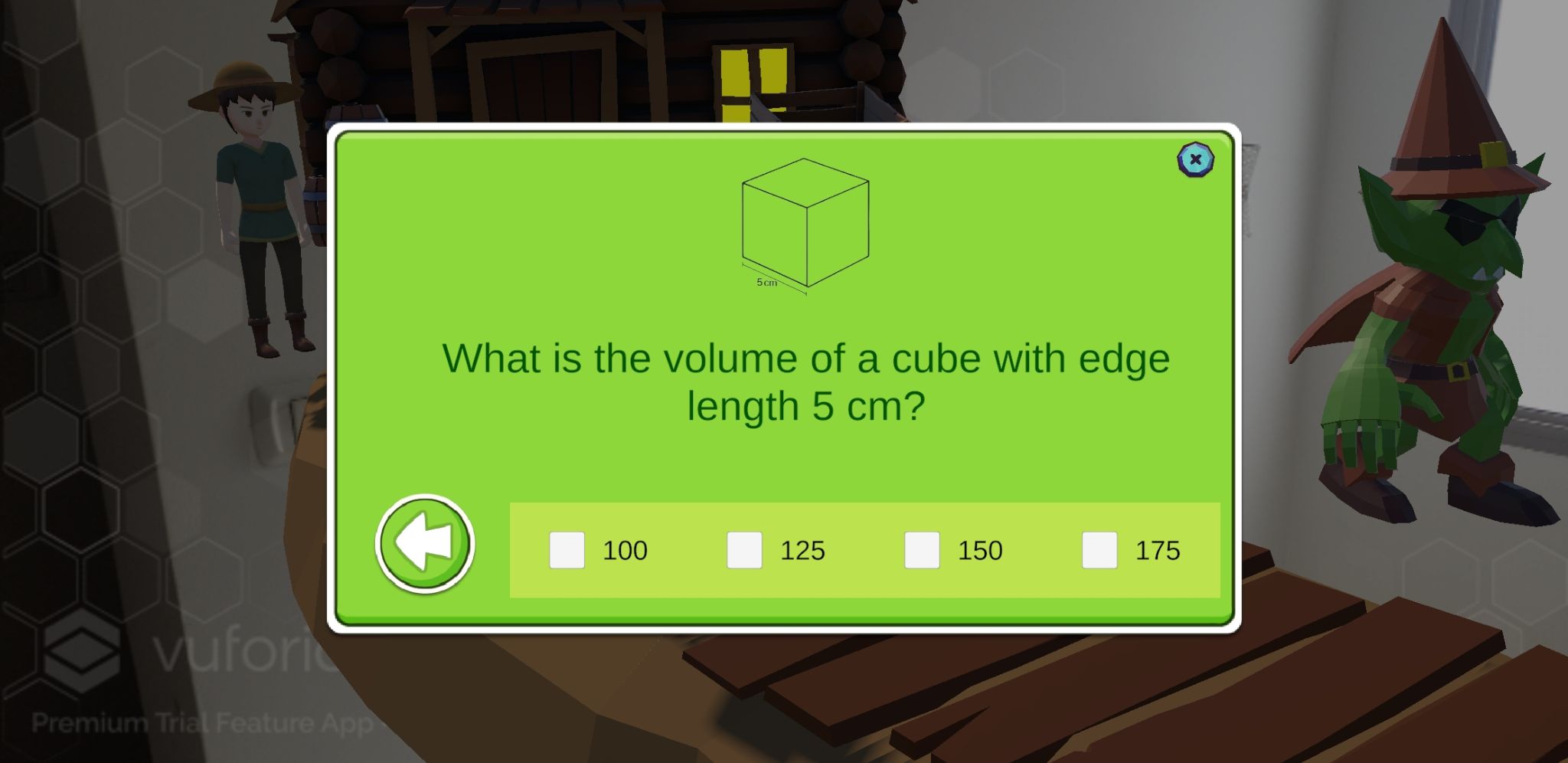
Use your device’s camera to scan a real-world object (like a cube or pyramid). The game will recognize the object and transport you into a matching 3D environment with related math questions.

**03:Enter the Shape World**



You’ll enter a virtual world that reflects the scanned shape.There, you'll meet an NPC, when you touch click the NPC, you will start a conversation with him that will lead to answering questions.

**04:Answer Questions About the Shape**

The NPC will ask you geometry questions about the shape you scanned.

Each question is a multiple-choice or an open question.

**05:Earn Puzzle Pieces**

****

For every correct answer, you will earn a puzzle piece.

Collect all the pieces to complete the puzzle and progress in the game!

**8.1.3 Settings screen for teacher**

**01:Management Screen**

****

Available Options:

Add Student

Allows the teacher to register a new student in the system by filling in their personal details.

View Student Progress

Displays each student’s performance, including scores and puzzle progress by shape.

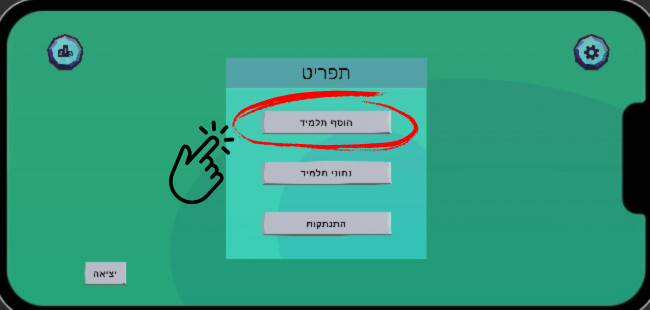
Logout Button

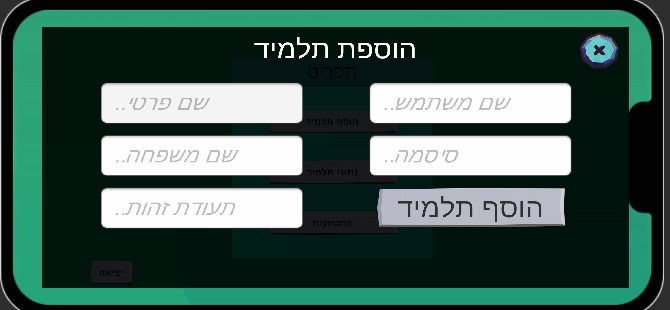
Pressing the Logout button will return the user to the Login screen.

Quit Button

Pressing the Quit button will close the application completely.

**02:Add Student Screen**

****

****

This screen allows teachers to register a new student in the system.

Required Fields (all in english only):

First Name – Enter the student's first name.

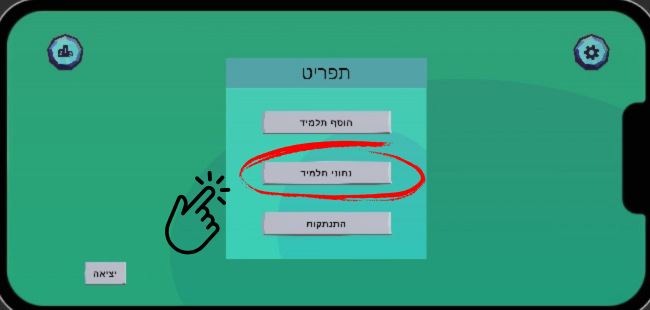
Last Name – Enter the student's last name.

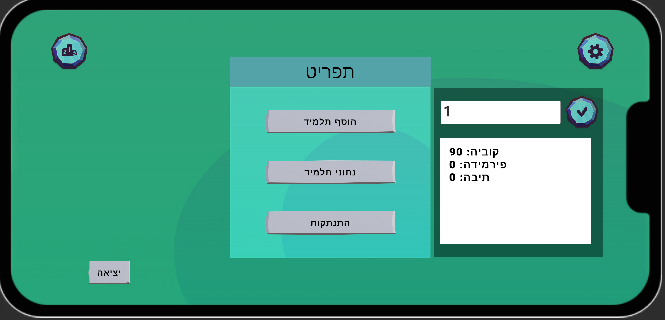
Username – Choose a unique username for the student.

Password – Create a secure password.

Student ID – Enter the student’s unique identification number.

**03:View Student Progress**

****

****

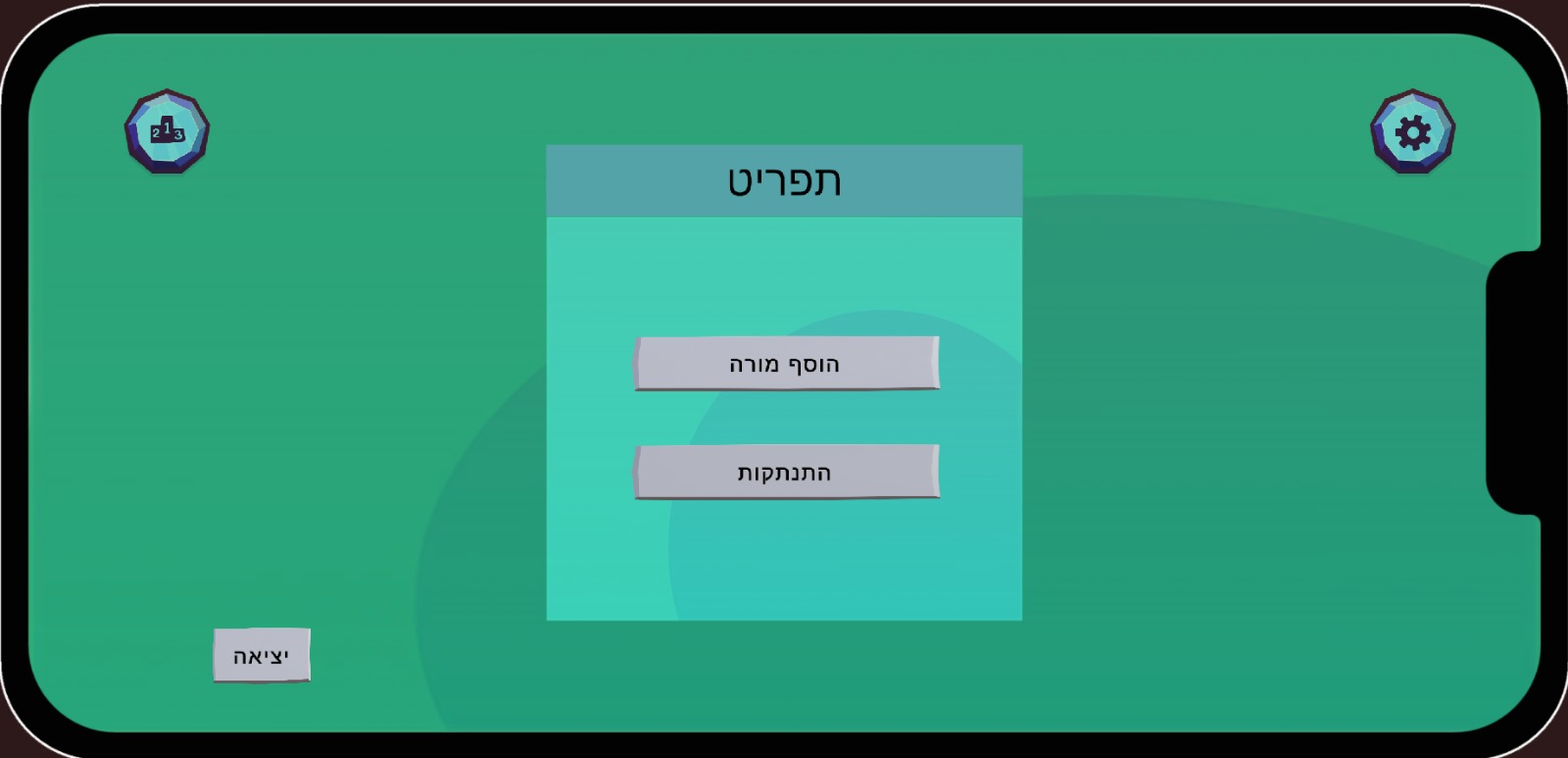
1.Enter the Student ID in the text field at the top-right.

2.Press the **(V)** check icon to confirm.

3.A list of shapes and corresponding points will appear, for example:

Pyramid: 0, Rectangular\_prism: 0,Cube: 90.

**8.1.4 Settings screen for Admin**

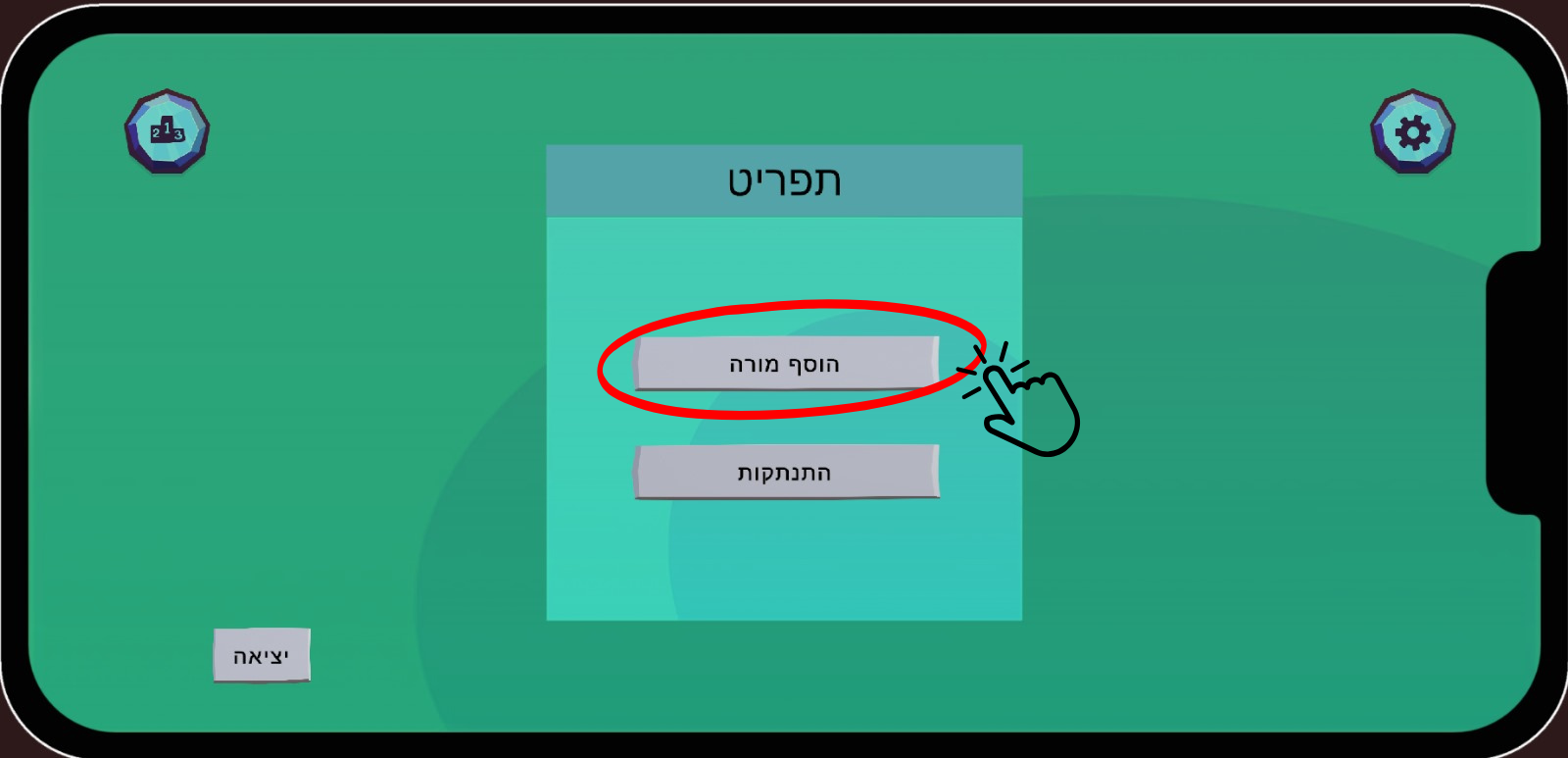
****

**Available Options:**

Add Teacher-Register a new teacher account by entering their details.

Settings-Access the application’s general settings such as language and sound.

**01:Add Teacher**

****

****

**This screen allows the admin to register a new teacher account.**

**Required Fields:**

First Name – Enter the teacher’s first name.

Last Name – Enter the teacher’s last name.

Username – Create a unique username for the teacher.

Password – Set a secure password.

ID Number – Enter the teacher’s identification number.

Tap the Add Teacher button to complete the registration process.

### **8.2 Maintenance Guide**

### **8.2.1 Overview**

This maintenance guide outlines how to manage, update, and extend the current codebase to ensure smooth functionality and long-term stability. The project is built using **Unity version 2022.3.58f1** and utilizes the **Vuforia Engine SDK** for augmented reality features. All required assets and dependencies are already included in the project, but it is advised to periodically check for updates to Unity and Vuforia to maintain compatibility and performance.

The project is structured using a **Model-View-Controller (MVC)** design approach, with additional support from global **Manager** scripts:

* **View** components are Unity-based UI elements and visual GameObjects.
* **Controllers** are MonoBehaviour scripts attached to scene objects to handle user interaction.
* **Models** represent core logic and data classes that operate independently of the scene.
* **Managers** are singleton classes responsible for handling global behavior like language settings, scene management, login sessions, and configuration.

Scripts are organized into clear directories:

* Scripts/Managers: Global control scripts such as LanguageManager, LoginManager, PuzzleManager, QuestionController and SettingsController.
* Scripts/Game: Core data models like UserData and QuestionData.
* Scripts/: General gameplay and UI logic including PlayerBehaviour, QuestionFetcher, and more.

Maintainers should be familiar with the role of each component and follow the MVC pattern when implementing new features or debugging existing functionality. Consistency in naming, modular development, and clear separation between view logic and backend logic are essential for maintaining a scalable and readable codebase.

**Main Scripts**

**Managers:** LanguageManager  PuzzleManager LoginManager

SceneMover  
 SettingsController  
 MainMenu

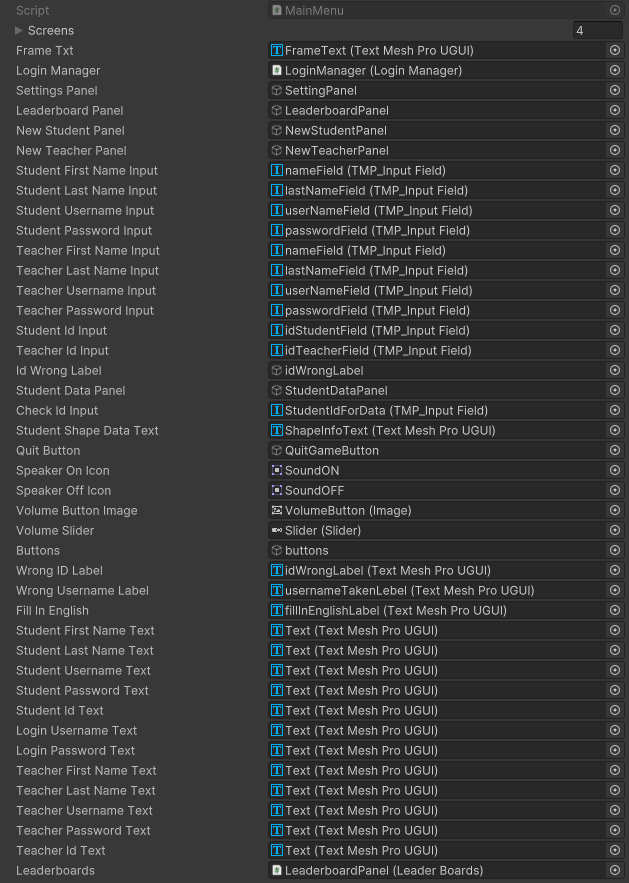
**Controllers & Models:** PlayerBehaviour  
 NPCBehaviour  
 QuestionFetcher  
 QuestionController  
 Puzzle  
 PuzzleCollectionLoader  
 QuestionItem  
 PuzzleUI  
 LeaderBoards  
 LeaderboardItem  
 DialogBox  
 LocalizedText  
 PlayerFollowCamera

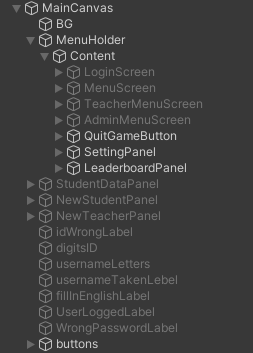
**Data Models:** UserData  
 QuestionData  
 GameData

**Helpers:** ARAudio  
 MyTargetHandler

### **8.2.2 MainMenu**

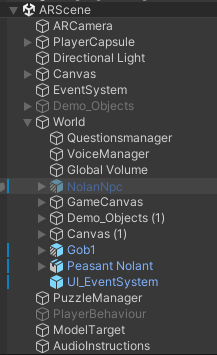
The main screen in this project is organized under the MainCanvas object, where all UI elements are grouped within the MenuHolder > Content structure. The controller that manages navigation and UI logic for this screen is called MainMenu. It is responsible for switching between different user panels (e.g., student, teacher, admin), showing the leaderboard and settings panels, and initializing the correct screen based on the user's role. Screen transitions are handled by enabling and disabling GameObjects such as LoginScreen, MenuScreen, TeacherMenuScreen, and others. Additionally, MainMenu connects UI elements such as input fields, labels, and buttons to logic for user registration, login, and volume control. The class also listens for language changes to update text direction and localized labels accordingly. Interaction with other managers (like LoginManager and LanguageManager) is coordinated through serialized references.



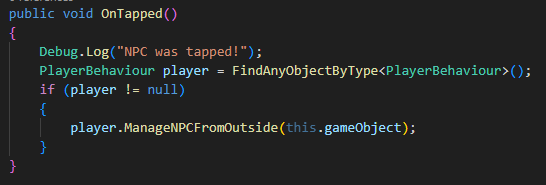


### **8.2.3 Gameplay Flow**

This section describes the flow of interaction from the main menu to the core AR gameplay. The MainMenu script is responsible for managing the user interface, including login, registration, and role-based navigation (student, teacher, admin). Upon successful login, the appropriate panel is displayed using the ManageScreens() method. Selecting **"New Game"** transitions the player to the AR scene (ARScene), where the PlayerBehaviour script becomes active. Players begin by scanning a hand-drawn shape, which unlocks the associated puzzle world. Within this world, players can approach and tap NPCs to initiate interaction. Each NPC functions as a **stage**, offering a set of 4 questions. As players answer correctly, they earn both **puzzle pieces** (1 per question) and **score points**. Once all 4 questions are answered, the stage is complete and the full puzzle is revealed. The player can then either proceed to the next NPC to begin a new stage or exit the game. This loop — login → new game → scan shape → stage-based NPC interaction → answer 4 questions → complete puzzle and earn points — defines the structured progression of the game.



-AR scene hierarchy



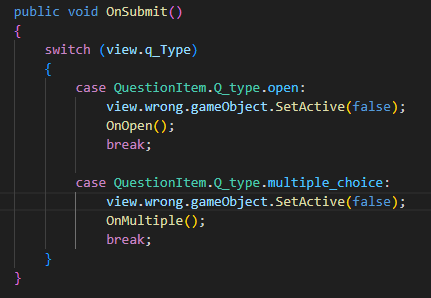
-The OnTapped() method in the NPCbehaviour script, which triggers NPC interaction by calling ManageNPCFromOutside() in the PlayerBehaviour script when the player taps on an NPC.

### **8.2.4 Answer questions**

At this phase of the game, the player interacts with an NPC and is presented with a series of four questions. These questions appear either as open-ended input or multiple-choice selections. The interface includes a visible area for the question, a section for answering, and feedback that confirms whether the answer is correct or incorrect.

Once the player selects or types an answer and submits it, the system checks if the answer is valid. If correct, a puzzle piece is rewarded and immediately shown on screen. Behind the scenes, this updates the player's progress, both visually on the puzzle and internally in memory. After all four correct answers, the puzzle is completed and the player can move to the next NPC.

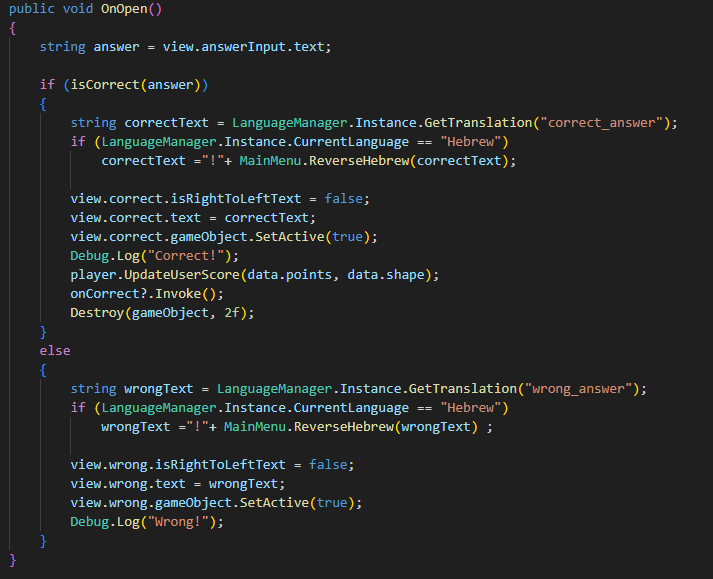
Puzzle pieces are saved per user and puzzle type, and progress is sent to the server to be stored. The user earns score points with every correct answer, which are also tracked by shape. The system includes a mechanism to update both the puzzle display and the user's total score dynamically, reflecting the educational progress made.



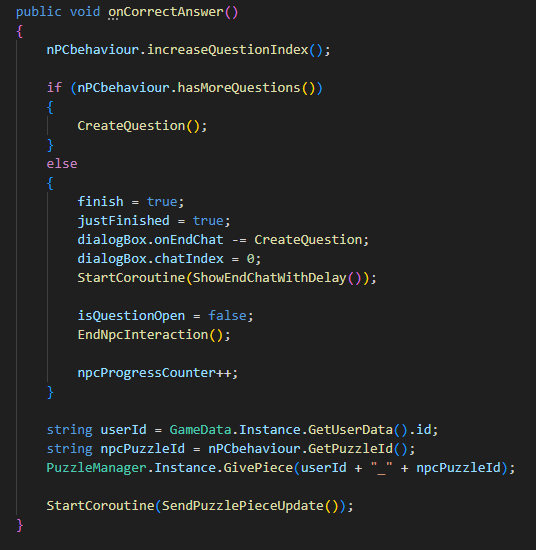
-Handles the player's answer submission and routes to the appropriate check method.



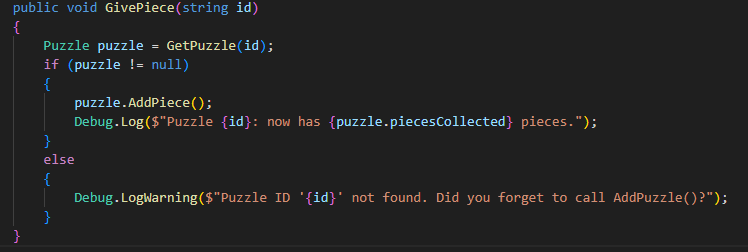
* Checks selected answers in multiple-choice questions and gives feedback.



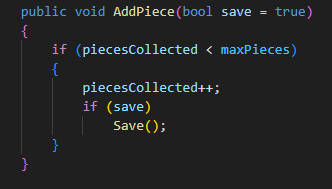
-Validates player input in open-ended questions and handles results.



-Called when an answer is correct; updates puzzle progress, score, and advances the game.



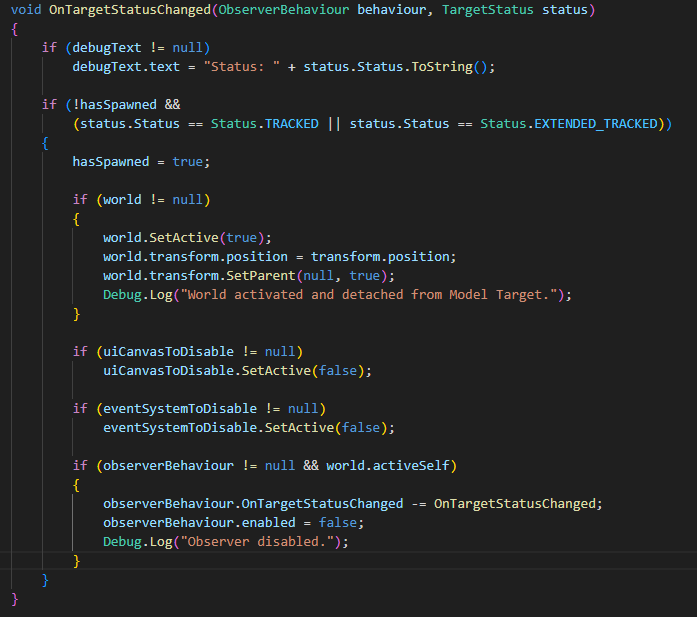
-Rewards the player with a puzzle piece and tracks which pieces are collected.



-Updates the internal puzzle data to reflect that a new piece was collected and saves it.

### **8.2.5 Scanning 3D object**

The scanning procedure centers on recognizing a simple, hand-made 3D object—typically a cube or pyramid—using the device’s camera. This is done through Vuforia’s Model Target feature, which identifies the object based on its polygonal structure, using a database trained on that specific shape (e.g., Pyramid\_model\_target). Once detected and tracked, the object becomes the world center (0,0,0), serving as a fixed anchor point for all AR elements. This configuration ensures that NPCs, puzzles, and other components are positioned consistently, whether placed manually in Unity or generated at runtime. Upon a successful scan, the instructional UI is hidden and the corresponding puzzle world is activated. This moment marks the beginning of the player’s interactive learning experience, seamlessly bridging the physical object with its virtual counterpart.



-The OnTargetStatusChanged method, which activates the game world and disables UI elements once the model target is detected and tracked by Vuforia, then detaches the world object and disables the observer to prevent repeated triggers.

Once the 3D object (such as a pyramid or cube) is successfully scanned, the game world is activated using SetActive(true). From this point, the player is able to walk and rotate freely in the AR environment, interact with NPCs, and answer questions to progress through the game.

#### **Environment Creation**

The world is built using a combination of methods:

* **Placed in Unity Editor** Trees, rocks, flowers, the cabin, wooden paths, and NPCs were manually arranged in the Unity Editor to build a forest-themed environment.
* **Generated by Code** Additional objects (such as puzzles, triggers, or future gameplay elements) can be spawned at runtime based on player progress or puzzle data.
* **Prefabs with Runtime Changes** Basic prefabs (e.g., NPCs, puzzle pieces, or interactive zones) are prepared in the editor and may change their visuals or logic at runtime (e.g., displaying new puzzle pieces or switching dialogues).

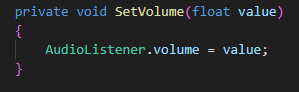


-example of the world object appears after scanning the 3D object.

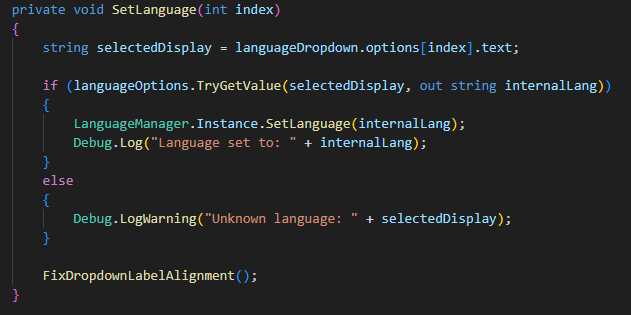
### **8.2.6 Settings**

The settings panel allows users to configure two main aspects of the game experience: **volume control** and **language selection**. The interface includes a slider for adjusting the global audio volume and a dropdown menu for choosing between English and Hebrew. The current language is handled via the LanguageManager, and changes are applied in real time. Hebrew text is displayed in reverse using a custom method to account for the right-to-left reading direction and ensure proper visual alignment in the dropdown.

When the user adjusts the volume slider, the AudioListener.volume property is updated immediately. Similarly, selecting a language from the dropdown updates the UI text and layout dynamically, including text alignment and right-to-left formatting for Hebrew. This responsive system ensures a personalized and accessible experience for users across different languages and preferences.



-control the volume value through the settings volume slider.



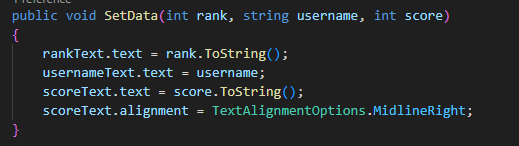
-Updates the game’s language based on the user’s dropdown selection.

It retrieves the selected value, maps it to an internal language key, and applies it using the language manager.

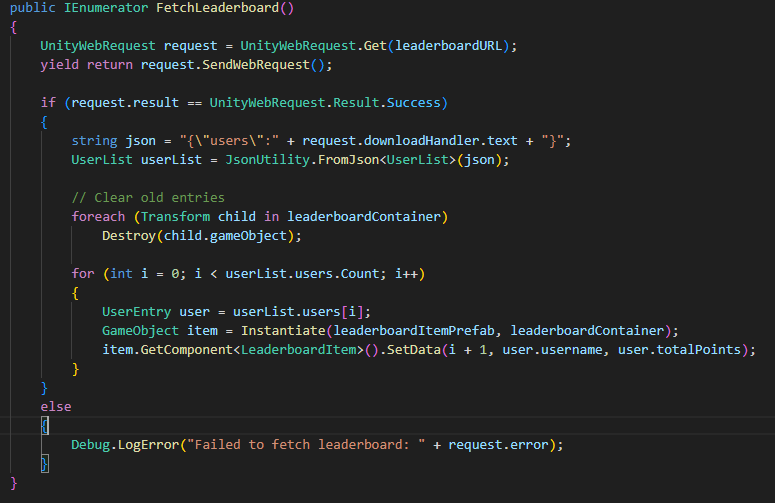
### **8.2.7 Leaderboards**

The leaderboard system allows players to view their ranking based on accumulated points. When the leaderboard panel is opened, the system sends a web request to a remote server that returns the top users and their total scores. The data is parsed and dynamically displayed in the interface using a prefab for each leaderboard row.

Each row shows the player’s rank, username, and total points. The list is updated in real-time and sorted in descending order by score. This feature encourages competition and motivates students to improve their performance through visible progression and comparison with peers.



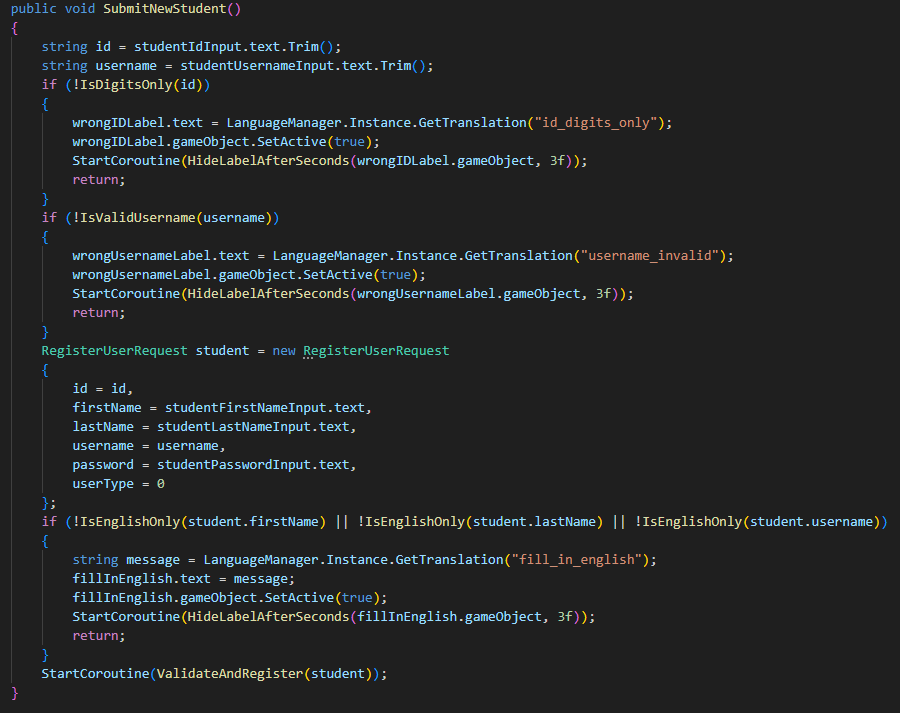
-Updates the UI elements of a leaderboard row by setting the player's rank, username, and score. It also aligns the score text to the right for better visual formatting.



-Updates the leaderboard panel by clearing old entries and populating it with the latest user rankings retrieved from the server. It ensures the panel always displays fresh and accurate score data when opened.

### **8.2.8 Student & Teacher Registration**

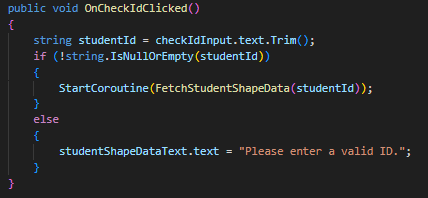
The registration system allows new users to sign up either as students or teachers. When opening the registration panel from the main menu, users are prompted to enter their personal details including first name, last name, ID, username, and password. The system includes validation steps to ensure the ID consists of digits only, the username is alphanumeric and not purely numeric, and all fields are filled using English characters. Upon submission, the system checks whether the ID or username already exists in the database by sending a request to the server. If all validations pass, the data is submitted to the backend and the user is registered successfully. This mechanism ensures unique accounts and clean input formatting, while providing feedback for any errors detected during the process.



-handles the registration process for new students by collecting input from the form, validating the ID, username, and character set, and then initiating a server request to complete the registration if all checks pass. It also displays appropriate error messages if the input is invalid (same method for teacher registration).

### **8.2.9 Student data**

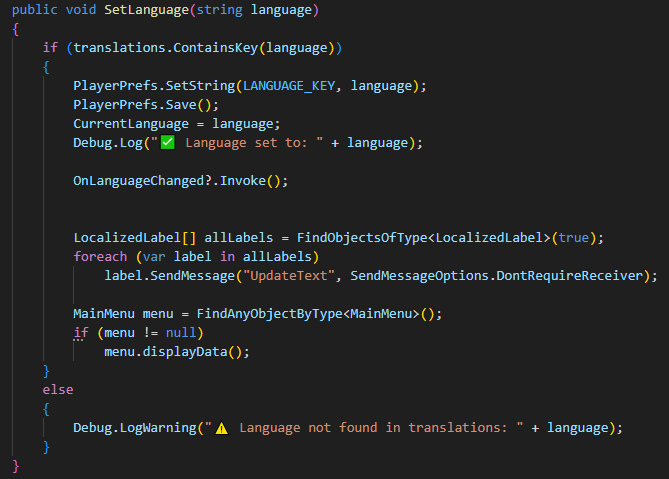
The student data panel allows teachers to enter a student's ID and retrieve detailed statistics about their progress. Once an ID is submitted, a request is sent to the server to fetch data including the total number of points and a breakdown of points earned by shape category (circle, square, triangle). If the student is found, the information is displayed in a formatted text panel, dynamically adjusted based on the selected language. The system also handles invalid or missing IDs by providing translated feedback messages. This feature gives teachers a quick and clear view of each student's learning performance.



- When the user clicks the "Check ID" button. It reads the entered student ID and, if valid, starts a coroutine to fetch and display the student’s shape-related performance data.

### **8.2.10 Language controller**

The language controller is managed by the LanguageManager script, which enables dynamic switching between English and Hebrew throughout the application. On startup, it loads translation data from a JSON file (translations.json) and applies the user's previously selected language stored in PlayerPrefs. The system supports real-time updates by raising a language change event and refreshing all UI labels tagged with LocalizedLabel. Texts are pulled using translation keys, ensuring consistency across scenes. Additionally, the system reverses Hebrew text as needed to support right-to-left display. This controller ensures the application is accessible and properly localized for users in both supported languages.



-sets the application's current language, saves the selection in PlayerPrefs, updates all localized UI elements, and refreshes the main menu display. If the specified language is not found in the translation data, a warning is logged.

#### **9 Results & conclusions**

The project successfully achieved its defined objectives, including the development of a functional AR-based educational game and the successful training and deployment of 3D model targets using Vuforia’s Target Management System (TMG). The system was able to recognize them consistently during testing. Once a shape is detected by the device’s camera, the world game object is activated, allowing the player to explore, interact with NPCs, and solve shape-related questions. This seamless transition from physical object to virtual environment is central to the game’s learning experience.

Key features of the system were implemented as planned and performed reliably. The Unity-based frontend, enhanced with Vuforia SDK, managed the AR interactions and in-game logic effectively. The backend, built with FastAPI and deployed on Render, supported essential functionality such as user authentication, dynamic question delivery, puzzle tracking, and score management. Communication between the frontend and backend was established using RESTful APIs, ensuring real-time data exchange and synchronization. The system also featured a dynamic question-loading mechanism that adapted content based on the player's progress and supported runtime content delivery from the database.

The application was structured according to a three-layer architecture—Application Layer, Logic Layer, and Backend Layer—which contributed to the maintainability and scalability of the system. Model recognition was accurate and consistent, and puzzle progression was updated and stored successfully after each correct answer. The modularity of the codebase and the clear separation of responsibilities between system components supported an efficient development process and laid the groundwork for future enhancements. Overall, the system met its performance, functionality, and educational design goals, demonstrating the effectiveness of combining AR and gamification for interactive math learning.

#### **9.1 Evaluation**

The system was evaluated using the System Usability Scale (SUS) questionnaire and qualitative user feedback. The SUS results indicated a high level of usability, with the majority of participants (86%) agreeing or strongly agreeing that they would like to use the system frequently. All participants disagreed that the system was too complex, difficult to navigate, or inconsistent, while 86% agreed it was easy to use. Additionally, all participants reported feeling confident while using the system and stated they would not require extensive learning before operating it. These findings reflect the system’s intuitive design and effective user experience.

Qualitative feedback further supported the positive SUS results. Most users expressed enjoyment and described the learning experience as engaging and different from traditional classroom methods. Representative quotes include: *“ The game makes learning more experiential”*, *“The game feels different from learning in the classroom”*, and *“The game introduces a different learning experience for students.”* Improvement suggestions were minimal, with only a few participants recommending the addition of more shapes and learning units, such as *“I would consider adding more shapes”* and *“Scanning more shapes, and adding an option for multiple learning units.”*

Overall, the evaluation demonstrated that the system provides a positive and effective learning experience, aligning with the project’s goal of combining AR and gamification to enhance interactive math education.

### **10. References**

1. Cerqueira, J., Sylla, C., Moura, J. M., & Ferreira, L. (2019). Learning Basic Mathematical Functions with Augmented Reality. In *ArtsIT 2018/DLI 2018, LNICST 265* (pp. 508–513). Springer.
2. Cai, S., Liu, E., Shen, Y., Liu, C., Li, S., & Shen, Y. (2020). Probability Learning in Mathematics Using Augmented Reality: Impact on Students’ Learning Gains and Attitudes. *Interactive Learning Environments, 28*(5), 560–573.
3. del Cerro Velázquez, F., & Morales Méndez, G. (2021). Application in Augmented Reality for Learning Mathematical Functions: A Study for the Development of Spatial Intelligence in Secondary Education Students. *Mathematics, 9*(369).
4. Hafeez, M. (2021). Effects of Game-Based Learning in Comparison with Traditional Learning to Provide Effective Learning Environment: A Comparative Review. *International Journal of Social Sciences & Educational Studies, 8*(4), 100–115.
5. Chao, W.-H., & Chang, R.-C. (2018). Using Augmented Reality to Enhance and Engage Students in Learning Mathematics. *Advances in Social Sciences Research Journal, 5*(12), 455–464.
6. Torres-Molina, R., Riofrío-Valdivieso, A., Bustamante-Orellana, C., & Ortega-Zamorano, F. (2019). Prediction of Learning Improvement in Mathematics through a Video Game using Neurocomputational Models. In Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019), pp. 554–559.
7. Kounlaxay, K., Shim, Y., Kang, S.-J., Kwak, H.-Y., & Kim, S. K. (2021). *Learning Media on Mathematical Education Based on Augmented Reality.* KSII Transactions on Internet and Information Systems, 15(3), 1015–1027​.
8. Unkelos-Shpigel, N., Berencwaig, B., & Kas, S. (2023). *Revise That Again: Are You Motivated?*
9. *López-Rosado, L., Soto, A., Ramírez, J. M., & Pérez, S. I. (2022). On the use of augmented reality to reinforce the learning of power electronics for beginners. Electronics, 11(3), 302.*
10. *Li, J., van der Spek, E., Hu, J., & Feijs, L. (2018). See Me Roar: On the over-positive, cross-cultural response to an AR game for math learning. In Proceedings of the International Conference on Serious Games (JCSG 2018)*