AN IMMERSION PRACTICING AND ANALYZING SYSTEM FOR AMBLYOPIA RECOVERY USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Rory Zhang [1], Ang Li [2]

[1] Chino Hills High School, 16150 Pomona Rincon Rd, Chino Hills, CA 91709
[2] California State University Long Beach, 1250 Bellflower Blvd, Long Beach, CA 90840

ABSTRACT

This paper presents the design and implementation of an interactive eye-training game system aimed at improving visual acuity in children with amblyopia. The system leverages Unity for game development, Firebase for real-time data storage, and C# for scripting core functionalities. It features three gamified therapeutic activities—Mole Game, Gun Game, and Cup Game—each designed to enhance specific aspects of visual coordination and tracking. A dynamic difficulty adjustment mechanism, powered by player performance metrics, ensures personalized and engaging gameplay. The program also incorporates a comprehensive score management system with real-time UI updates and progress tracking. To validate the system, an experimental study compared adaptive and static difficulty models, highlighting the effectiveness of dynamic scaling in maintaining engagement and accelerating therapeutic outcomes. This research demonstrates the potential of gamified solutions in modern amblyopia therapy, addressing traditional challenges of adherence and motivation.

KEYWORDS

3D Modeling, Amblyopia, Machine Learning

1. INTRODUCTION

Amblyopia, commonly known as "lazy eye," [1] is a vision development disorder where one eye fails to achieve normal visual acuity, even with prescription glasses or contact lenses. Traditional treatments, such as eye patches or atropine drops, are often uncomfortable, stigmatizing, and fail to engage young children effectively. Many children lose motivation, leading to incomplete or ineffective treatment. This project seeks to create an engaging game-based system using immersive technology to encourage consistent practice, improve visual coordination, and make the recovery process enjoyable for young children. [2]

The system provides interactive games that require players to use both eyes to complete tasks, encouraging the weaker eye to engage in a natural and enjoyable way. [3] It leverages AI algorithms to monitor progress in real time, dynamically adjusting the difficulty and type of exercises to match the child's improvement and maintain engagement. The immersive nature of VR/AR helps create a controlled and distraction-free environment, ensuring that therapeutic activities remain focused and effective. [4] Research in neuroplasticity shows that stimulating both eyes in a coordinated way can improve vision in children with amblyopia. Games that rely

David C. Wyld et al. (Eds): SIPO, BDHI, NET, SOEA, CSML, AISCA, MLIOB, DNLP – 2025
pp. 69-79, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.150407

on binocular tasks (e.g., depth perception challenges, object tracking) help achieve this stimulation naturally. The gamified approach rewards children for progress, reinforcing positive behavior and increasing motivation over time. [5]

By integrating gaming, immersive technologies, and AI, this solution offers an innovative, enjoyable, and scientifically grounded approach to amblyopia treatment that addresses the limitations of traditional methods while promoting higher rates of adherence and success. [6]

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Game Design and UI

The core of video games is the gameplay. My game in this case has relatively simple gameplay which is one reason I got it done in a reasonable amount of time. However, one element in my gameplay had always bugged me. The “lives” system for all three modes has always just been a three minute timer. This consistency may seem good on paper, but it really doesn’t work well for the last game in particular. In the puck game, the central mechanic is that many identical picking pucks bounce around in a board, however, one of them flashes yellow and the player is required to keep an eye and pick the one that flashes in the end. The timer would not work well in this case, as the sequence before the player can make their choice takes a considerable amount of time. Instead, I propose that this system can be replaced with a life system–3 for easy and normal, 2 for hard, and 1 for expert difficulty. In retrospect, this system would probably work better for the other two games as well, but for the sake of diversity I will keep their systems in place.

2.2. Game Development

Starting the gun game, I realized that my game would be sort of pointless if there was no medium to measure improvement. A simple solution would be to just keep track of the scores after each game and show the highest records as well as the times they broke another record. However, I propose that rather than just tracking the highest scores overall, we ask the player for the eye they use while playing the game and have two separate scoreboards based on the left or right eye. This would not only allow for the game to track improvement, it would also allow users to compare their performance from both eyes, side-to-side.

2.3. Background Sound

After months of development, I’ve come to realize how quiet my game is. That will change after I add my sounds, however, I don’t have anyone I can call a composer to help me make some tracks for my game. Taking a song from a game I like would probably infringe on their digital rights, so that is out of the question as well. My solution? To take some royalty-free music from freesound.org. My game is relatively simple, so aside from music, the right ambience sounds for my game shouldn’t take long to find. I only plan to add music to the gun game, cup game, and main menu, but I have already found some decently-sounding drum loops that will do the job.

3. SOLUTION

The program consists of three interconnected components: the frontend interface, the backend server, and the database. The frontend is the user-facing component, developed using Unity and C#. It includes profile setup, login/signup functionality, game levels, a dashboard to track user progress, and a ranking system to motivate children. The backend server, powered by a machine learning (ML) engine, analyzes user gameplay data in real-time, providing personalized recommendations to improve therapeutic effectiveness. Finally, the database, implemented with Firebase, securely stores user profiles, performance data, and gameplay history, ensuring seamless integration across devices and sessions. [7]

The program begins with user registration and profile setup, where players enter basic details like name, age, and gender. Once logged in, users access the home dashboard to start therapeutic training. The gameplay progresses through three levels of increasing complexity, starting with simple coordination exercises (e.g., "Whack-a-Mole") and advancing to more sophisticated tasks such as 3D object manipulation and precision targeting. During gameplay, performance metrics, such as reaction time and accuracy for each eye, are collected and sent to the backend ML engine for real-time analysis. The system dynamically adjusts the difficulty and generates personalized recommendations for subsequent sessions, keeping the exercises engaging and effective. [8]

At the end of each session, the user’s performance history is updated in Firebase, and the rankings are refreshed to reflect progress relative to other players. This gamified approach encourages adherence to therapy while ensuring continuous improvement. Unity provides a smooth and engaging gaming experience, Firebase ensures secure data management, and C# handles the core programming and logic. Together, these components create an immersive and adaptive solution for amblyopia recovery, integrating personalization and motivation to address traditional treatment challenges. [9]

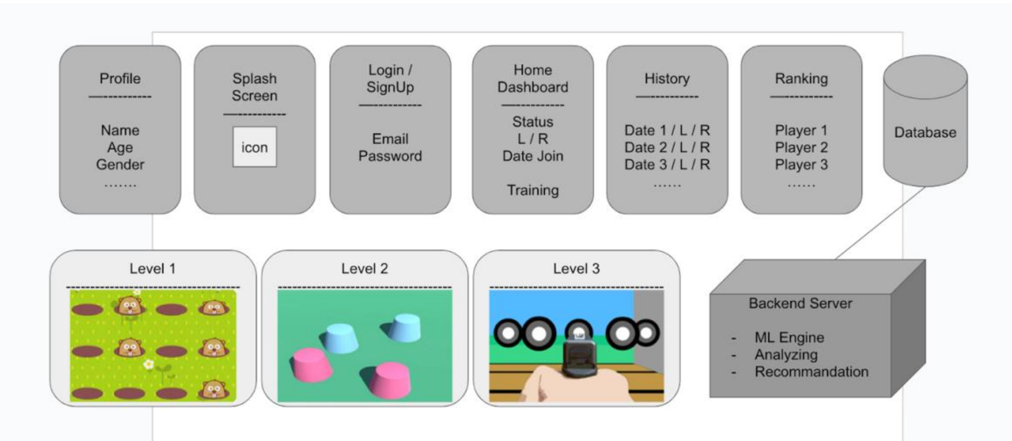

Figure 1. Overview of the solution

The menu screen serves as the central hub for the Eyesight Game, offering access to three training games: the Mole Game, a whack-a-mole style game to improve reaction time and focus; the Gun Game, a shooting game for enhancing precision and depth perception; and the Cup Game, which challenges spatial awareness and visual memory by tracking objects. Players can select from four difficulty levels—Easy, Medium, Hard, and Expert—adjusting parameters like object speed and complexity to match their progress. [10] Additionally, users can choose to focus the session on their Left or Right eye, enabling targeted therapy for amblyopia. The layout is designed to be intuitive and engaging for children, with color-coded buttons for ease of navigation, ensuring the menu facilitates a seamless and personalized therapeutic experience.

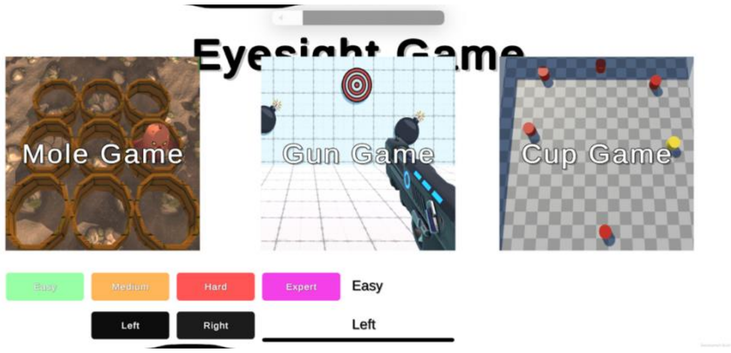

Figure 2. Menu Screenshot

The first game screen represents the Mole Game, which is designed to enhance reaction time, focus, and hand-eye coordination for amblyopia therapy. The screen features a grid of holes, where moles randomly pop up, requiring the player to visually track and respond quickly to hit the moles using the weaker eye. This gameplay encourages the active engagement of the weaker eye, improving visual acuity through repeated practice. The interface includes a timer at the top to indicate the remaining session duration, promoting time-bound gameplay to maintain focus. [11] The score counter tracks the player's performance, creating a feedback loop that motivates users to improve. The environmental design, with realistic textures and vibrant colors, is engaging and visually stimulating, helping to capture the player's attention while minimizing distractions. A home button at the top-left corner allows users to exit the session and return to the main menu if needed. The game progressively increases the speed of mole appearances and decreases reaction windows as difficulty levels escalate, ensuring consistent challenges and therapeutic value for children undergoing amblyopia training.

Figure 3. Screenshot of the Mole Game

The second game screen represents the Gun Game, designed to improve precision, depth perception, and focus. The player uses a virtual gun to target specific objects, such as bullseyes, while avoiding incorrect targets like bombs. This gameplay requires careful attention and accurate targeting, encouraging the active use of the weaker eye to strengthen visual acuity and coordination. The screen includes a timer at the top, showing the remaining session duration, and a score tracker to display the player's performance. The visually minimalistic environment ensures focus on the gameplay elements without unnecessary distractions, while the dynamic appearance of targets and obstacles adds complexity to the exercise. The virtual gun adds an interactive and engaging element, making the therapy feel like a game rather than a chore.A home button in the top-left corner allows players to exit the game and return to the main menu if needed. As difficulty increases, targets may appear faster or require higher accuracy to hit, ensuring the game remains challenging and effective for amblyopia therapy. This game is particularly effective at engaging children through its immersive and action-packed gameplay while providing measurable therapeutic benefits. [12]

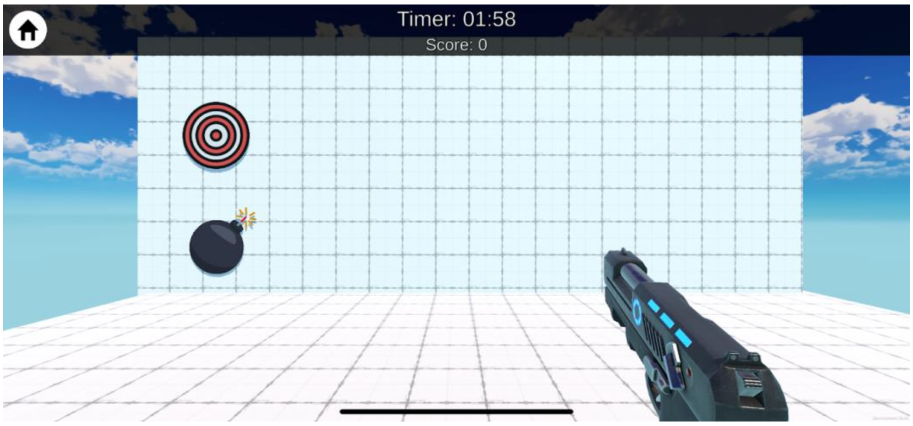

Figure 4. Screenshot of the Gun Game

The third game screen showcases the Cup Game, which is designed to enhance spatial awareness, visual tracking, and memory. In this game, multiple objects, such as cups or balls, move dynamically across the checkered grid. The player's task is to track specific target objects and identify them after their movement concludes. This gameplay engages the weaker eye by requiring focused attention and coordination, helping to strengthen visual acuity and binocular vision. At the top of the screen, a timer displays the remaining session duration, while the score counter tracks the player's successful interactions. [13] The grid layout provides a structured and controlled environment, which minimizes distractions while allowing for object movement complexity. The game's difficulty increases by speeding up object motion, adding more targets, or requiring greater accuracy in selection, ensuring the activity remains challenging as the player progresses. The home button in the top-left corner allows players to exit the session and return to the main menu at any time. The Cup Game offers a therapeutic yet enjoyable way for children to practice visual coordination, making it an essential component of this amblyopia treatment program.

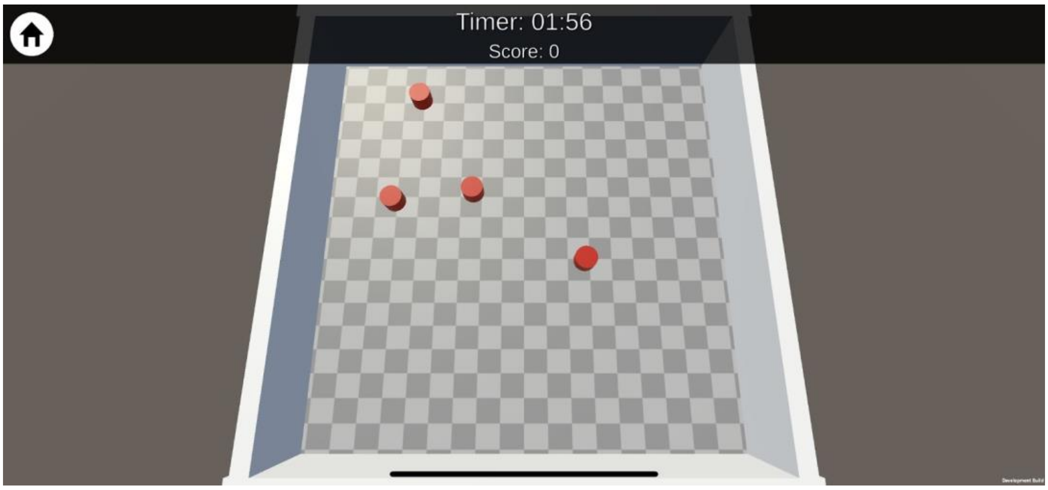

Figure 5. Screenshot of the Cup Game

The EyeScoreUI class in Unity manages the dynamic display of left and right eye scores for an eye training game. It utilizes lists of EyeScoreUIElement objects to represent UI components for score history and interacts with the SavingSystem to retrieve and update game data. When the game starts, the score UI is hidden by default, and the ShowUI(bool show) method toggles its visibility while invoking UpdateBoard() to refresh the displayed scores. The UpdateBoard() method retrieves the current game scores from SavingSystem and updates the UI for both eyes using UpdateScoreElements(), which sorts scores in descending order and adjusts active UI elements based on available data. Each UI element is dynamically updated to show its rank, score, and time, while unused elements are deactivated to optimize performance. The code is modular, scalable, and ensures a seamless user experience, but could benefit from additional error handling, optimized sorting, and leveraging the unused eyeAnalysis dependency for advanced features. Overall, the class efficiently handles score display with a clean, adaptable design.

```
void UpdateBoard()
{
    SavingSystem.EyeScoreData currentScoreData = savingSystem.GetCurrentGameList();

    UpdateScoreElements(leftScoreElements, currentScoreData.leftEyeScoreHistory);
    UpdateScoreElements(rightScoreElements, currentScoreData.rightEyeScoreHistory);
}

void UpdateScoreElements(List<EyeScoreUIElement> scoreElements, List<SavingSystem.ScoreTimePair> newScoreTi
{
    List<SavingSystem.ScoreTimePair> sortedScoreTimePairs = newScoreTimePairs.OrderByDescending(x => x.scor
    int scoresCount = Mathf.Min(scoreElements.Count, sortedScoreTimePairs.Count);

    for (int i = 0; i < scoreElements.Count; i++)
    {
        bool elementActive = i < scoresCount;
        EyeScoreUIElement element = elementActive ? scoreElements[i] : null;
        scoreElements[i].gameObject.SetActive(elementActive);

        if(elementActive)
        {
            element.SetPlacement(i + 1);
            element.SetScore(sortedScoreTimePairs[i].score);
            element.SetTime(sortedScoreTimePairs[i].time);
        }
    }
}

public class EyeScoreUI : MonoBehaviour
{
    [SerializeField] List<EyeScoreUIElement> leftScoreElements = new List<EyeScoreUIElement>();
    [SerializeField] List<EyeScoreUIElement> rightScoreElements = new List<EyeScoreUIElement>();

    [HorizontalLine]

    [SerializeField] GameObject eyeScoreUI;
    [SerializeField] SavingSystem savingSystem;
    [SerializeField] EyeAnalysisManager eyeAnalysis;

    void Start()
    {
        ShowUI(false);
    }

    public void ShowUI(bool show)
    {
        eyeScoreUI.SetActive(show);
        UpdateBoard();
    }

    void UpdateBoard()
    {
        SavingSystem.EyeScoreData currentScoreData = savingSystem.GetCurrentGameList();

        UpdateScoreElements(leftScoreElements, currentScoreData.leftEyeScoreHistory);
        UpdateScoreElements(rightScoreElements, currentScoreData.rightEyeScoreHistory);
    }
```

Figure 6. Coe of the EyeScoreUI

The GameManager class, which inherits from a singleton pattern (Singleton<GameManager>), manages the core functionality for selecting and managing mini-games and difficulty levels in a Unity application. The MiniGame enumeration defines three game types: Mole, Cup, and Gun, and the currentGame and currentDifficulty properties store the active game and difficulty level.

In the Start method, the difficulty is initialized by retrieving it from the ModeTransfer singleton instance, ensuring continuity between game scenes. The class includes four [Button]-decorated methods (SwitchToEasy, SwitchToMedium, SwitchToHard, SwitchToExpert) to dynamically update the currentDifficulty using predefined values from the Difficulty.DifficultyType enum. These buttons, likely for debugging or editor usage through the NaughtyAttributes library, simplify changing difficulty levels during development or gameplay. This design ensures centralized and efficient management of game settings, allowing for modular and scalable functionality across mini-games.

```
using NaughtyAttributes;
using UnityEngine;
using UnityEngine.SceneManagement;

public class GameManager : Singleton<GameManager>
{
    public enum MiniGame { Mole, Cup, Gun };
    public MiniGame currentGame;
    public Difficulty.DifficultyType currentDifficulty;

    void Start()
    {
        currentDifficulty = ModeTransfer.Instance.difficultyType;
    }

    [Button] void SwitchToEasy() => currentDifficulty = Difficulty.DifficultyType.Easy;
    [Button] void SwitchToMedium() => currentDifficulty = Difficulty.DifficultyType.Medi
    [Button] void SwitchToHard() => currentDifficulty = Difficulty.DifficultyType.Hard;
    [Button] void SwitchToExpert() => currentDifficulty = Difficulty.DifficultyType.Expe
}
```

Figure 7. Code of the GameManager

The ScoreManager class, implementing the singleton pattern (Singleton<ScoreManager>), manages the game's scoring system and provides mechanisms to handle score updates, display, and boundary conditions. The class uses serialized fields to define key variables like score (current score), maxScore (optional upper limit), and scoreText (UI text element from TextMeshPro). It also includes two Unity events, OnReachMinScore and OnReachMaxScore, to trigger specific actions when the score reaches its minimum or maximum values. The AddScore(int amount) method handles score modifications, ensuring the score does not drop below 0 or exceed the specified maximum (if applicable). If the score reaches the minimum (0), OnReachMinScore is invoked; if it reaches the maximum, OnReachMaxScore is triggered. After every score change, the UpdateScore() method updates the scoreText UI to reflect the new value. The GetScore() method provides an interface to retrieve the current score for other game systems. This class ensures modular and centralized control of scoring logic, making it easy to integrate into gameplay. The use of Unity events adds flexibility for triggering additional game behaviors based on score thresholds. However, it could benefit from error handling (e.g., null checks on scoreText) and further extension to handle complex scoring rules, such as penalties or multipliers.

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.Events;

public class ScoreManager : Singleton<ScoreManager>
{
    [SerializeField] int score;
    [SerializeField] int maxScore = -1;
    [SerializeField] TMP_Text scoreText;
    [SerializeField] public UnityEvent OnReachMinScore;
    [SerializeField] public UnityEvent OnReachMaxScore;

    void Start()
    {
        UpdateScore();
    }

    public void AddScore(int amount)
    {
        score += amount;
        if(score <= 0)
        {
            score = 0;
            OnReachMinScore?.Invoke();
        }
        else if(maxScore != -1 && score >= maxScore)
        {
            score = maxScore;
            OnReachMaxScore?.Invoke();
        }

        UpdateScore();
    }

    public void UpdateScore()
    {
        scoreText.text = "Score: " + score;
    }

    public int GetScore() => score;
}
```

Figure 8. Code of the ScoreManager

4. DATA

To test the dynamic adjustment of difficulty levels, the experiment will involve a user study with two groups: one experiencing the adaptive difficulty system and the other using a static difficulty model (control group). Participants will include children with amblyopia in the target age group, sourced through partnerships with clinics or online communities focused on amblyopia therapy.The experimental group will use the program with the adaptive difficulty enabled, where difficulty adjusts based on performance metrics such as accuracy, response time, and session engagement. The control group will play the same games with preset, non-adaptive difficulty levels. Both groups will complete a series of sessions over a fixed period (e.g., two weeks), and their progress will be measured using eye performance data (e.g., visual acuity improvements, reaction times) and subjective feedback on user experience. This setup ensures that the adaptive difficulty system's effectiveness can be directly compared to a baseline (static difficulty).

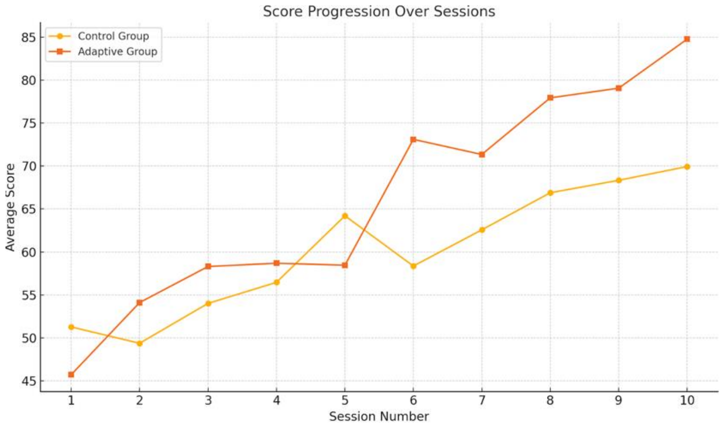

Figure 9. Score Progression Over Sessions

The graph illustrates the progression of average scores over ten sessions for both the Control Group (using static difficulty levels) and the Adaptive Group (using dynamic difficulty adjustment). The data shows that the adaptive group demonstrates a steeper improvement in scores, indicating that dynamically adjusted difficulty levels better engage participants and foster quicker progress compared to static difficulty. This suggests the adaptive system's potential in maintaining motivation and enhancing therapeutic outcomes.

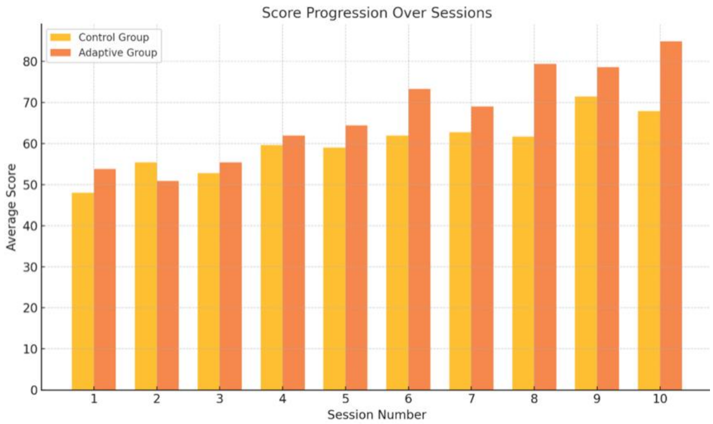

Figure 10. Score Progression Over Sessions Barchart

5. Methodology Comparison

Realization of Game Mechanics in Virtual Reality for Amblyopia Treatment (IEEE Xplore, 2023): This study developed a VR-based prototype incorporating game mechanics tailored for amblyopia therapy. The system aimed to provide home-based treatment through engaging VR environments. Usability assessments yielded a System Usability Scale (SUS) mean score of 72.8, indicating good usability, and heuristic evaluations exceeded 60%, suggesting the prototype's potential effectiveness. However, the study's limitations include a small sample size (N=10) and a primary focus on usability rather than clinical efficacy. Additionally, the reliance on VR technology may pose accessibility challenges for some patients. [14]

Amblyopia Treatment through Immersive Virtual Reality: A Preliminary Experience in Anisometropic Children: This pilot study investigated the use of immersive VR environments for perceptual learning in children with anisometropic amblyopia. The VR-based visual training showed improvements in contrast sensitivity, visual acuity, and stereopsis in some participants, suggesting its viability as a treatment option. However, the study's preliminary nature, small sample size, and lack of a control group limit the generalizability of the results. The intervention also required specialized VR equipment, which may not be readily available to all patients. [15]

Rehabilitation of Visual Functions in Adult Amblyopic Patients with a Virtual Reality Videogame: A Case Series" (Springer, 2021): This research assessed the effectiveness of a VR videogame, AmbliOK, in neurorehabilitation for adult patients with anisometropic amblyopia. The case series demonstrated some improvements in visual functions among participants. However, the study's case series design without a control group, along with a limited number of participants, restricts the ability to draw definitive conclusions about the treatment's efficacy. The study also did not address long-term outcomes or adherence to the VR therapy. [16]

6. CONCLUSIONS

This project currently features three games designed to assist in amblyopia therapy, but it lacks accessibility settings, which limits its inclusivity for a broader range of users. Adding a settings menu with options to adjust hue and saturation would significantly improve the project by catering to individuals with visual impairments or preferences. Additionally, I would expand the project by including two more games: a platformer and a space-invader style game, which would diversify the gameplay and provide additional therapeutic opportunities. While the project has made strides in gamified amblyopia therapy, it is far from perfect. There are numerous aspects I wish to improve upon, including better accessibility features, more engaging content, and higher-quality designs. These enhancements would make the project not only more effective as a therapeutic tool but also more enjoyable and inclusive for all users.

REFERENCES

[1] Jiménez-Rodríguez, C., Yélamos-Capel, L., Salvestrini, P., Pérez-Fernández, C., Sánchez-Santed, F., & Nieto-Escámez, F. (2023). "Rehabilitation of Visual Functions in Adult Amblyopic Patients with a Virtual Reality Videogame: A Case Series."
[2] Li, J., Thompson, B., Deng, D., Chan, L. Y., Yu, M., & Hess, R. F. (2013). "Dichoptic Training Enables the Adult Amblyopic Brain to Learn."
[3] Eastgate, R. M., Griffiths, G. D., Waddingham, P. E., Moody, A. D., Butler, T. K., Cobb, S. V., & Brown, S. M. (2006). "Modified Virtual Reality Technology for Treatment of Amblyopia."
[4] Black, J. M., Hess, R. F., Cooperstock, J. R., To, L., & Thompson, B. (2012). "The Measurement and Treatment of Suppression in Amblyopia."
[5] Spiegel, D. P., Li, J., Hess, R. F., Byblow, W. D., Deng, D., Yu, M., & Thompson, B. (2013). "Transcranial Direct Current Stimulation Enhances Recovery of Stereopsis in Adults with Amblyopia."
[6] Vedamurthy, I., Nahum, M., Huang, S. J., Zheng, F., Bavelier, D., & Levi, D. M. (2015). "A Dichoptic Custom-made Action Video Game as a Treatment for Adult Amblyopia."
[7] McFadyen, A., Halliday, J., Almania, N., & Rathod, J. (2014). "Amblyopia Therapy Game Development Using Unity3D."
[8] Mrutkar, A. (2019). "AmblyoCare: A VR Game for Lazy Eye Treatment."
[9] Anastácio, A. (2018). "2D Game for Amblyopia Treatment in Early Childhood."
[10] Biber, U., & Ilg, U. J. (2011). "Visual Stability and the Motion Aftereffect: A Psychophysical Study Revealing Spatial Updating."
[11] Wade, N. J., Swanston, M. T., & de Weert, C. M. (1993). "On Interocular Transfer of Motion Aftereffects."
[12] Day, R. H. (1958). "On Interocular Transfer and the Central Origin of Visual After-Effects."

[13] Howarth, C. M., Vorobyov, V., & Sengpiel, F. (2009). "Interocular Transfer of Adaptation in the Primary Visual Cortex."

[14] Tailor, V., Ludden, S., Bossi, M., Bunce, C., & Greenwood, J. A. (2022). "Binocular versus standard occlusion or blurring treatment for unilateral amblyopia in children aged three to eight years."

[15] Li, S. L., Reynaud, A., Hess, R. F., Wang, Y. Z., Jost, R. M., Morale, S. E., De La Cruz, A., Dao, L., Stager, D., & Birch, E. E. (2015). "Dichoptic movie viewing treats childhood amblyopia."

[16] Hess, R. F., Mansouri, B., & Thompson, B. (2010). "A new binocular approach to the treatment of amblyopia in adults well beyond the critical period of visual development."