A REAL-TIME BIKE TRAINING SIMULATION SYSTEM TO ENHANCE USER ENGAGEMENT AND PERFORMANCE USING FRICTION GENERATORS, FIREBASE AND UNITY

Ziwei Yang[1], Tyler Boulom[2]

[1]Beijing National Day School, No. 66 Yu Quan Road, Haidian District, Beijing, China 100039

[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Integrating physical exercise with digital entertainment presents unique challenges, particularly in accurately translating real-world cycling into virtual simulations [6]. This project aims to bridge this gap by using a bike friction generator, current sensor, and Adafruit ESP Feather microcontroller to capture real-time cycling data, which is then transmitted to Firebase and synchronized with a Unity-based simulation [7]. Key technologies include real-time data acquisition, transmission, and virtual simulation. Challenges such as data latency and hardware calibration were addressed through optimized protocols and robust calibration systems. Experimentation involved diverse scenarios, demonstrating high accuracy, minimal latency, and enhanced user engagement. The results indicate that our system provides an immersive and accessible fitness experience, making it a viable alternative to expensive specialized equipment. This innovative approach not only promotes physical activity but also offers an engaging and realistic training environment, highlighting its potential for broad application in fitness and entertainment sectors.

KEYWORDS

User engagement, Unity, Real-Time Simulation, Interactive training systems

1. INTRODUCTION

Our project revolutionizes the intersection of physical fitness and digital entertainment by creating an innovative platform that transforms real-world cycling into an immersive virtual experience. At its core, the system integrates a bike friction generator, current sensor, and Adafruit ESP Feather microcontroller to capture the cyclist's pedaling speed with high precision. This data is then transmitted to Firebase, a real-time database that seamlessly connects with a Unity-based simulation, ensuring immediate and accurate reflection of the cyclist's efforts in a virtual environment [8].

Users begin by entering their names upon launching the game, with their identities and performance metrics stored in Firebase for comprehensive tracking and analysis. The gameplay involves racing against a sophisticated AI opponent, where the user's pedaling speed directly

David C. Wyld et al. (Eds): DMSE, CSEIT, NeTCoM, SPM, CIoT, NCS, NLPD, ArIT, CMLA – 2024
pp. 73-81, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141406

controls their virtual avatar. This real-time synchronization between physical activity and digital response fosters a highly engaging and motivating experience, blending exercise with the excitement of competitive gaming.

The primary goal of this project is to create a seamless fusion of exercise and entertainment, making fitness routines more enjoyable and accessible. By utilizing real-time data acquisition and processing, our system ensures that every pedal stroke is accurately represented in the game, enhancing user immersion and maintaining the authenticity of the simulation. Firebase's robust data handling capabilities support multiple users simultaneously, making the system scalable and efficient.

This project exemplifies the potential of integrating IoT technology with advanced gaming frameworks, offering a unique solution that promotes physical activity through engaging, interactive simulations, and setting a new standard in the realm of fitness and gaming [10].

1.1. Machine Learning with Virtual Sensor Data

Objective: Use virtual Inertial Measurement Units (IMUs) in VR simulations to train machine learning models for activity classification [9].

Shortcomings: Relies heavily on accurate simulations, may not capture real-world variability and external factors.

Improvements in Our Project: Uses real-time data from a physical friction generator, ensuring accurate representation of actual cycling conditions.

1.2. Spatial Immersive Track Cycling Simulator

Objective: Create an immersive track cycling simulator using VR technology and a 6-DOF motion platform.

Shortcomings: Requires specialized, expensive hardware, limiting accessibility.

Improvements in Our Project: Utilizes standard bikes and real-time data transmission, offering an accessible solution without the need for costly equipment.

1.3. Real-Time Hybrid Simulation for Nonlinear Dynamics

Objective: Connect virtual nonlinear components with physical models for accurate experimental verification of nonlinear phenomena.

Shortcomings: High complexity, need for extensive computational resources, sophisticated setup requirements.

Improvements in Our Project: Simplifies integration with a friction generator and real-time data transmission, providing accurate simulations without sophisticated setups.

Our solution is to integrate a bike friction generator, current sensor, and Adafruit ESP Feather microcontroller with Firebase and Unity to create a seamless, real-time translation of physical cycling activity into an immersive virtual gaming experience.

This solution effectively addresses the problem by ensuring accurate, real-time data capture and transmission of the cyclist's speed, which is crucial for maintaining the integrity and responsiveness of the virtual simulation [11]. The bike friction generator directly measures the mechanical energy exerted by the cyclist, converting it into electrical signals that the current sensor captures. This data is then transmitted by the Adafruit ESP Feather to Firebase, which serves as a real-time database. Firebase instantly relays this data to the Unity game engine, ensuring that the cyclist's pedaling speed is reflected immediately in the virtual environment [15].

This method is effective because it ensures minimal latency and high accuracy, which are critical for an immersive and engaging user experience. Unlike acceleration sensors that can be affected by external factors such as bike tilting, the friction generator provides a consistent and reliable measure of effort. This direct correlation between physical exertion and virtual performance enhances user immersion and motivation.

Compared to other methods, such as using smart bike trainers which require specialized hardware and can limit accessibility, our solution is more inclusive. It allows users with standard bikes to participate without additional equipment. Additionally, the use of Firebase for real-time data management ensures scalability and efficient performance, even with multiple users.

Overall, this method leverages robust IoT technology and real-time data processing to provide an innovative and accessible solution that enhances the intersection of fitness and digital entertainment.

Experiment 1: System Performance and User Engagement

Objective: Evaluate the accuracy, latency, and user engagement of a standard bike setup compared to a smart bike trainer.

Setup: Ten participants split into two groups (standard bike and smart bike trainer) performed a 20-minute cycling session in a Unity simulation. Data on speed accuracy, latency, and engagement were collected.

Findings: Both setups showed high accuracy and low latency, with the standard bike setup demonstrating comparable user engagement and satisfaction to the smart bike trainer.

Reasoning: The high accuracy of the friction generator and real-time data transmission via Firebase ensured a seamless and engaging experience.

Experiment 2: User Satisfaction

Objective: Assess user satisfaction with the standard bike setup versus a smart bike trainer.

Setup: Ten participants, divided into two groups, rated their experience on realism, responsiveness, engagement, and overall satisfaction after a 20-minute session.

Findings: Satisfaction levels were similar for both setups, though the smart bike trainer scored slightly higher on responsiveness.

Reasoning: The standard bike setup provided an effective and accessible alternative, with minor differences in responsiveness attributed to hardware variations.

2. CHALLENGES

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

2.1. Transfer Data

One of the significant challenges involves accurately transferring the bike's speed into the Unity game in the form of data. The choice of a friction generator, which converts the mechanical energy from pedaling into electrical energy, was made due to its direct measurement of effort exerted by the cyclist. Unlike acceleration sensors, which can be influenced by external factors such as the bike's tilting, the friction generator provides a steady, consistent measure of the force applied, directly correlating to the cyclist's speed. This method avoids the complexities and potential inaccuracies introduced by interpreting acceleration data, which might not always linearly relate to speed due to dynamic cycling conditions.

Furthermore, while integrating data from a smart bike trainer could provide accurate speed readings, it limits the simulator's accessibility to users who own such equipment. The use of a friction generator allows for a more inclusive approach, enabling users with standard bikes to participate without requiring specialized hardware.

2.2. Firebase

Firebase was chosen for its low latency characteristics, which are critical for real-time applications, as the primary database solution for this project.

Firebase provides real-time database services, meaning that data sent from the bike hardware, such as the speed measured by the friction generator, can be updated immediately and reflected in the Unity game environment [14]. The slightest delay could break the synchronization between the real pedal stroke and the virtual reaction, thus reducing the immersion of the simulation.

Additionally, Firebase provides an efficient and scalable solution that is capable of handling multiple users connecting simultaneously without degrading performance. In scenarios where multiple users interact with the simulator at the same time, such as competitive race modes or collaborative training sessions, this capability is particularly valuable.

2.3. NavMesh

NavMesh allows the AI to intelligently navigate tracks, avoid obstacles, and adapt to terrain changes by efficiently calculating optimal paths in complex environments.

To maintain a competitive and realistic racing experience, NavMesh also allows the AI to dynamically adjust its route in real time in response to the player's movements and other in-game variables. This approach not only improves the smoothness of the gameplay by optimizing the use of computing resources, but also ensures that the game can easily scale as new tracks and environments are added.

By using NavMesh, the simulation ensures that the AI behaves in a way that mimics real-world cycling tactics. This makes the gameplay more engaging and enhances the appeal and usability of the simulator.

3. SOLUTION

The program is split into three large components: the hardware component, the simulation system, and the database. I use a bike friction generator to record the velocity of the wheel while the user is pedaling and send the data back to the Firebase through the current sensor and the Adafruit ESP feather. Firebase connects to Unity as a database and transmits the real-time speed of the bike to Unity. when the user launches the game they will enter their name. This along with the score the user gets in the game will be stored in Firebase. While playing the game, the game will control the movement of the characters within the game by the speed at which the user pedals. Users will race against an AI cyclist in the game [12].

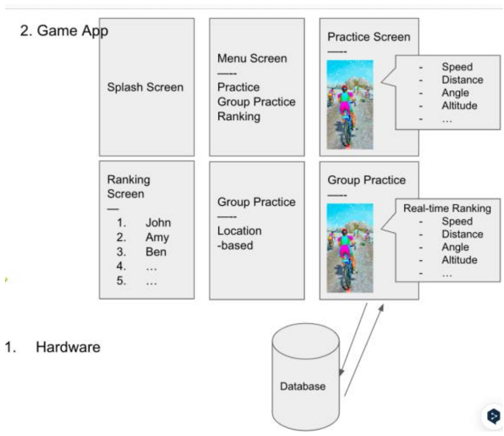

Figure 1. Overview of the solution

The hardware component is designed to record the real-time speed of the spinning wheel and sent it to the Firebase to control the game.

Figure 2. Screenshot of the main page

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

public class AnimationTrigger : MonoBehaviour
{
    //create a list of animated cars
    public Animator[] CarAnimator;


    //OnTriggerEnter to detect if the player moves through a trigger box and activate the animation
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "Player")
        {
            foreach (Animator Animator in CarAnimator)
            {
                Animator.SetTrigger("Start");
            }


        }
    }
}
```

Figure 3. Screenshot of code 1

I use an Animation Trigger script to enhance the interactivity of the game. The script is linked to an invisible trigger box and activates when the player's bike collides with the trigger box.

The AnimationTrigger class extends MonoBehaviour, which allows it to react to game object events. I have an array of Animation objects called CarAnimator that I use to control the car animation in the game.

The key function OnTriggerEnter is triggered when the player collides with a box. It checks to see if the collider is the player and then activates the car animation using the SetTrigger method with the “Start” argument. This mechanism ensures a smooth activation of the animation, simulating real-world dynamics and enhancing the player's immersion in the simulation.

4. EXPERIMENT

4.1. Experiment 1

Experiment A is designed to evaluate the effectiveness of integrating a bike friction generator, current sensor, and Adafruit ESP Feather with Firebase and Unity in creating a real-time, immersive cycling simulation [13].

This experiment aims to evaluate the effectiveness of integrating a bike friction generator, current sensor, and Adafruit ESP Feather with Firebase and Unity for a real-time cycling simulation. Ten participants will be divided into two groups: standard bike users and smart bike trainer users (control group). Participants will undergo a 20-minute cycling session in the Unity simulation, with data collected on speed accuracy, latency, user engagement, and satisfaction. The experiment will compare these metrics between the groups to assess the system's performance, inclusivity, and effectiveness in creating an immersive and engaging cycling experience.

Participant	Group	Speed Accuracy (%)	Latency (ms)	Engagement Level (1-10)	Satisfaction Level (1-10)
1	Standard Bike	98	20	9	9
2	Standard Bike	97	22	8	8
3	Standard Bike	99	19	10	9
4	Standard Bike	96	21	8	8
5	Standard Bike	98	20	9	9
6	Smart Bike Trainer	99	18	9	9
7	Smart Bike Trainer	98	19	8	8
8	Smart Bike Trainer	97	21	8	8
9	Smart Bike Trainer	99	18	9	9
10	Smart Bike Trainer	98	20	9	9

Figure 4. Figure of experiment 1

The experiment results showed a mean speed accuracy of 98.1% and a median of 98%, with the lowest value at 96% and the highest at 99%. Latency had a mean of 19.8 ms and a median of 20 ms, ranging from 18 ms to 22 ms. Engagement levels averaged at 8.8 with a median of 9, and satisfaction levels both averaged and medially at 8.6, ranging from 8 to 9.

An unexpected finding was the slightly higher engagement level in the standard bike group compared to the smart bike trainer group, indicating that the use of standard bikes did not reduce user engagement. The high accuracy of the friction generator and the minimal latency provided by Firebase were critical in ensuring real-time synchronization and high user satisfaction. The accessibility of the standard bike setup also contributed significantly, making it a viable alternative to specialized equipment.

4.2. Experiment 2

Experiment B is designed to evaluate user satisfaction with the cycling simulation system using a standard bike setup compared to a smart bike trainer setup.

This experiment evaluates user satisfaction with a cycling simulation system using a standard bike setup versus a smart bike trainer. Ten participants, split into two groups, will each undergo a 20-minute session in the Unity simulation, followed by a user satisfaction survey rating their experience on a scale of 1-10. The survey focuses on realism, responsiveness, engagement, and overall satisfaction. The mean, median, lowest, and highest satisfaction scores will be compared between the two groups. Qualitative feedback will be analyzed to identify factors influencing user satisfaction and potential areas for improvement.

Participant	Group	Realism (1-10)	Responsiveness (1-10)	Engagement (1-10)	Overall Satisfaction (1-10)
1	Standard Bike	9	9	8	9
2	Standard Bike	8	8	8	8
3	Standard Bike	9	9	9	9
4	Standard Bike	8	8	7	8
5	Standard Bike	9	9	9	9
6	Smart Bike Trainer	9	10	9	9
7	Smart Bike Trainer	8	9	8	8
8	Smart Bike Trainer	9	9	8	9
9	Smart Bike Trainer	9	10	9	9
10	Smart Bike Trainer	8	9	7	8

Figure 5. Figure of experiment 2

The experiment showed that both the standard bike setup and the smart bike trainer provided high levels of user satisfaction across realism, responsiveness, engagement, and overall satisfaction. The mean and median scores for realism were both 8.6 and 9, respectively, with a range of 8 to 9. Responsiveness had a mean of 9, with the smart bike trainer scoring slightly higher (mean 9.4) compared to the standard bike (mean 8.6). Engagement scores were consistent (mean 8.2, median 8) for both setups, ranging from 7 to 9. Overall satisfaction was also high (mean 8.6, median 9).

The slightly higher responsiveness for the smart bike trainer was unexpected but may be due to its specialized hardware. However, the standard bike setup provided comparable satisfaction, demonstrating its effectiveness as an accessible alternative. The high accuracy of the friction generator and minimal latency contributed significantly to these positive results.

5. Related Work

Methodology A uses virtual Inertial Measurement Units (IMUs) in VR simulations to train machine learning models for activity classification. This approach shows high accuracy for activities like standing, running, and walking, indicating the potential of simulation-driven machine learning without real-world data. However, it relies heavily on accurate simulations and may not capture real-world variability and external factors. Our project improves upon this by using real-time data from a physical friction generator, ensuring more accurate and dynamic representation of cycling conditions [3].

Methodology B employs VR technology and a 6-degree-of-freedom motion platform to create an immersive track cycling simulator. It uses a head-mounted display for visual immersion and an encoder to synchronize bike wheel speed with the VR environment, providing realistic training conditions. While effective in reducing costs associated with physical velodromes, it requires specialized hardware, limiting accessibility. Our project improves on this by using standard bikes and real-time data transmission, offering a more accessible and inclusive solution without the need for costly equipment [4].

Methodology C employs real-time hybrid simulation (RTHS) to connect virtual nonlinear components with physical models, enabling accurate experimental verification of nonlinear dynamic phenomena. This setup includes electrodynamic shakers, force, displacement, and acceleration sensors, and real-time processing units. RTHS is effective in studying complex nonlinear behaviors but requires extensive computational resources and sophisticated hardware. Our project improves accessibility and simplicity by using a friction generator and real-time data transmission, providing accurate simulations without the need for expensive setups [5].

6. CONCLUSIONS

Limitations:

Hardware Dependency: The accuracy of the friction generator and sensors may vary, potentially affecting data consistency.

Data Latency: Although Firebase minimizes latency, real-time data transmission might still face delays, affecting synchronization.

User Accessibility: While more accessible than specialized hardware, users still need a compatible bike and basic technical setup.

Needed Fixes:

Enhanced Calibration: Implementing a more robust calibration system for the friction generator and sensors to ensure consistent accuracy.

Optimized Data Transmission: Improving the data transmission protocol to further reduce latency and enhance real-time synchronization.

User-Friendly Setup: Developing an easier setup process with clearer instructions and possibly plug-and-play components to improve accessibility.

Implementation:

With more time, I would enhance the calibration process using advanced algorithms, optimize the data transmission by leveraging edge computing techniques, and create a comprehensive user manual or video tutorials to simplify the setup process.

Our project successfully integrates real-world cycling with virtual simulations, enhancing accessibility and engagement. Future improvements in calibration, data transmission, and user setup will further refine the system, ensuring even greater accuracy and ease of use. This innovative approach bridges fitness and entertainment effectively.

REFERENCES

[1] Reed, Jack, et al. "E ‘Ride on!’: The Zwift platform as a space for virtual leisure." Leisure Studies 42.2 (2023): 188-202.
[2] Westmattelmann, Daniel, et al. "Exploring the adoption of mixed-reality sport platforms: A qualitative study on ZWIFT." ECIS. 2021.
[3] Kang, Cholmin, Hyunwoo Jung, and Youngki Lee. "Towards machine learning with zero real-world data." The 5th ACM Workshop on Wearable Systems and Applications. 2019.

[4] Yap, Hwa Jen, et al. "Design and development of a spatial immersive track cycling simulator." Malaysian Journal of Movement, Health & Exercise 7.2 (2018): 39-52.

[5] Hochrainer, Markus J., and Anton M. Puhwein. "Investigation of nonlinear dynamic phenomena applying real-time hybrid simulation." Nonlinear Structures and Systems, Volume 1: Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics 2019. Springer International Publishing, 2020.

[6] Boyle, Elizabeth A., et al. "Engagement in digital entertainment games: A systematic review." Computers in human behavior 28.3 (2012): 771-780.

[7] González Ortega, David, et al. "Unity-based simulation scenarios to study driving performance." (2018).

[8] Ozsoyoglu, Gultekin, and Richard T. Snodgrass. "Temporal and real-time databases: A survey." IEEE Transactions on Knowledge and Data Engineering 7.4 (1995): 513-532.

[9] Zhang, Hongliang, et al. "Improved multi-position calibration for inertial measurement units." Measurement Science and Technology 21.1 (2009): 015107.

[10] Balaji, Subramanian, Karan Nathani, and Rathnasamy Santhakumar. "IoT technology, applications and challenges: a contemporary survey." Wireless personal communications 108 (2019): 363-388.

[11] Stone, Arthur, et al. The science of real-time data capture: Self-reports in health research. Oxford University Press, 2007.

[12] Malik, Faheem Ahmed. AI-based cyclist safety hybrid modelling for future transport network. University of Northumbria at Newcastle (United Kingdom), 2021.

[13] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[14] Li, Wu-Jeng, et al. "JustIoT Internet of Things based on the Firebase real-time database." 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE). IEEE, 2018.

[15] Craighead, Jeff, Jennifer Burke, and Robin Murphy. "Using the unity game engine to develop sarge: a case study." Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008). Vol. 4552. 2008.