AN IMMERSION SAILING EXPERIENCE AND SIMULATION FEEDBACK SYSTEM FOR DISABLED PEOPLE USING ARTIFICIAL INTELLIGENCE AND VIRTUAL REALITY

HoiNi Yeung[1], Ang Li[2]

[1]Ulink College of Shanghai, Shanghai, China
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Sailing requires specific practice conditions, such as access to open water and expensive equipment, limiting its accessibility for many. To overcome these barriers, I developed a virtual reality (VR) sailing simulator that allows users to practice sailing techniques anywhere, regardless of weather or location [5]. The simulator is built using Unity for realistic game environments, Visual Studio for robust coding, and VR technology for immersive interaction[6]. Key challenges included accurately modeling the physics of wind dynamics and sail adjustments and ensuring realistic responses to user inputs. These challenges were addressed through detailed data analysis and iterative refinement. The simulator was tested across various scenarios to replicate real-world conditions, demonstrating its effectiveness in improving users' sailing skills. This VR sailing simulator offers a cost-effective, accessible training solution, enabling both beginners and experienced sailors to practice and enhance their skills in a safe, controlled environment, making it an essential tool for sailing education and practice.

KEYWORDS

Unity, Simulation, Sailing Training, Virtual Reality

1. INTRODUCTION

Sailing, unlike many other sports, presents unique challenges due to its strict practice requirements. Unlike sports that need only basic equipment and accessible indoor spaces, sailing requires a large body of water and a sailboat [7]. This poses a significant barrier, as not everyone has easy access to the sea, and urban lakes often lack the consistent wind conditions necessary for effective practice. The presence of tall buildings around lakes can further disrupt wild patterns, creating unpredictable conditions unsuitable for training.

Moreover, the cost of a sailboat adds a substantial financial hurdle, making sailing less accessible than sports with more affordable equipment like balls or paddles. Beyond these logistical and financial barriers, sailing demands specific environmental conditions to ensure safe and effective practice, especially for boats like the Laser, which rely entirely on wind power [8]. Inadequate wind renders the boat immobile, while excessive wind poses safety risks and potential damage to the boat.

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 43-52, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141705

From my experience and that of my peers, we often face these limitations, leading to missed opportunities for practice. This has inspired the idea of developing a sailing simulator, enabling enthusiasts and professionals to practice sailing at home regardless of weather or location. Such a simulator would lower entry barriers, making the sport more accessible and appealing to newcomers while providing seasoned sailors with a tool to maintain their skills in any condition.

Physics-Based Watercraft Simulator in VR: This methodology uses hydrodynamic data and Unreal Engine to simulate realistic watercraft behavior in VR. It aims to enhance training by accurately replicating the effects of waves on vessels. However, it requires high computational resources and lacks comprehensive modeling of wind dynamics and other sailing-specific factors. Your project improves this by integrating wind dynamics and sail interactions for a more holistic sailing simulation.

Virtual Reality Scene Construction for Human-Machine Interaction: This approach creates immersive 3D simulations for maritime operations, focusing on visual and interactive realism [9]. It effectively enhances user experience but is limited by high setup costs and does not adequately simulate physical forces like wind and water interactions. Your project addresses this by incorporating realistic physical modeling, ensuring a more accurate representation of sailing mechanics.

Simulation of Real Water in 3D Animation: This technique uses advanced graphics to create visually stunning water effects but prioritizes aesthetics over physical accuracy, neglecting key hydrodynamic forces. Your project improves upon this by balancing visual realism with accurate physics simulations, offering a comprehensive training tool that better reflects real-world sailing conditions.

To overcome the barriers of traditional sailing practice, this project proposes the development of a Virtual Reality (VR) sailing simulation game. A virtual environment allows for precise control over conditions that are often unpredictable in real life, such as wind and weather. By integrating realistic models of the sky, sea, and sailboat, users can experience lifelike sailing scenarios. Players interact directly with the boat’s equipment, where adjustments to the sails and rudder realistically affect the boat's speed and direction, providing an authentic practice experience anytime and anywhere.

This VR solution is far more accessible and cost-effective than attempting to recreate ideal sailing conditions in reality. Unlike the substantial investment required for a sailboat and the constant upkeep needed for its maintenance, a VR setup is affordable and eliminates geographical and weather-related constraints. Most households already have a computer, and adding a VR headset is a relatively low-cost option compared to purchasing a sailboat.

Moreover, the time and effort required for real-world sailing—including transporting the boat, setting it up, and maintaining it—can be a significant deterrent, especially for those with limited access to suitable waters. In contrast, a VR sailing simulator allows immediate access to practice sessions, free from the logistical challenges and time constraints associated with traditional sailing. This innovative approach offers an efficient, realistic, and practical solution for both novice learners and experienced sailors looking to hone their skills.

In the first experiment, we tested if the boat speeds in the VR sailing simulator accurately reflect real-world conditions under different wind speeds, sail angles, and boat orientations [10]. We set up this experiment by having participants adjust sail angles and boat directions while the simulator recorded boat speeds. We compared these speeds to real-world Laser sailboat data and found that the simulator closely mimicked real-world speeds, confirming its realism.

In the second experiment, we examined if adjusting the sail angle in the simulator mirrored the physical effort required in real life. Participants used a gamepad to adjust the mainsail, and we measured the change in sail angle per swing. The results showed that the simulator underrepresented the effort needed to change sail angles, suggesting the need for calibration to match real-world sailing. Both experiments highlighted the importance of realistic simulations for effective training and learning experiences.

2. CHALLENGES

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

2.1. Accurately calculating

A major challenge in developing the sailing simulator is accurately calculating the sailboat's speed when players modify conditions. The boat's speed and direction depend on several forces, including lift, which allows sailing against the wind. Lift is generated by a pressure difference on either side of the sail, explained by Bernoulli's Principle. Understanding these forces involves analyzing how variables, such as sail angle, interact with constants, like sail area. By applying these principles and formulas, the simulator can realistically model the boat's response to different wind conditions, creating a more authentic and educational sailing experience.

2.2. Creating realistic wind behavior in the simulator

Creating realistic wind behavior in the simulator is challenging due to its natural complexity. Wind results from horizontal air movement, but simulating countless air particles is impractical. Instead, the simulator will use a 2D vector variable representing wind on the x and z axes, with values randomly generated to mimic natural variability [11]. This approach allows for dynamic changes in wind direction and force, essential for calculating the sailboat's response. By abstracting wind into invisible vectors without collision volume, the simulator effectively models wind's impact while maintaining computational efficiency and realism.

2.3. Creating seamless player interaction with the sailboat

Creating seamless player interaction with the sailboat in virtual reality is challenging. The simulator will use a VR Head-Mounted Display (HMD) and a gamepad for intuitive control. The HMD provides immersive visual and auditory feedback while tracking head movements, enabling actions like dodging the boom during tacks and gybes. The gamepad will manage crucial sailing controls, such as steering the rudder and adjusting the sail angle via the main sheet. Buttons will replicate grabbing and releasing actions, offering a realistic, hands-on sailing experience that combines sensory feedback with precise maneuvering controls.

3. SOLUTION

To construct this program, we used Unity to create the game scenarios, Visual Studio to write the code, and VR to allow the player to interact directly with the game. We decided to use Unity not only because of its clear interface, but also because of the ease of importing various game models. We decided to use Visual Studio Code because it contains many pre-set procedures, which reduces the time required and difficulty to write programs. We decided to use a VR device because the use of VR can sense movements, this allows the user to move their whole body while using the simulator, which achieves the goal of to physically practicing sailing. Our program is designed to be a VR sailing simulator. When opened, it will display a menu to prompt them to choose tutorial mode or practice mode. If practice mode is selected, the user will be brought into the winds of a 4-buoy course with changing wind directions at regular interval. There is no time limit for this mode, so it is used for free practice. If tutorial mode is selected, a tutorial will be displayed on the screen with a continuous explanation on how to sail, with corresponding images and diagrams. This allows the player to click "Continue" or "Back" on the tutorial screen to view the next or previous page. The included tutorial has a practice level every few pages so that the player can try out the newly learned skills right away. The practice levels only unlock some of the features of the sailboat but not all of them, allowing the player to focus on one skill at a time. During a practice level, the top left corner of the screen will show the tasks that need to be accomplished to pass the level and their progress, and once the task is completed, the player will automatically exit the level and continue to the next part of the tutorial.

A key part of the simulator is the User Interface system. This system is integral to the simulator's design because it allows users to interact with the user interface, performing instructions such as the entering, exiting, and pausing of the game. This user Interface System relies on Unity Components and custom manager scripts to control 2D UI graphics on the main menu or tutorial pages [12].

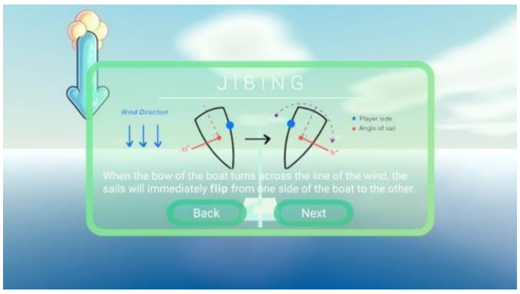

Figure 1. Screenshot of the main menu

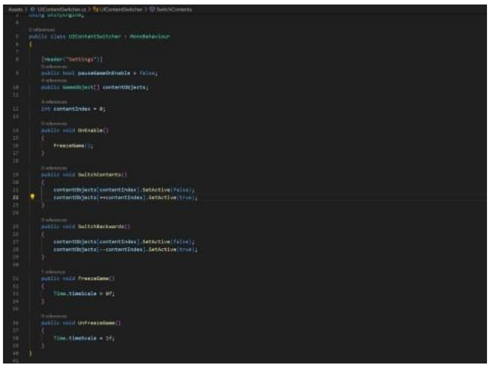

Figure 2. Screenshot of code 1

The program run on the user interface in the sailing simulator, when the user presses either the practice button or any of the buttons on the tutorial interface. Unity Components and custom manager scripts control 2D UI graphics for the main menu, tutorial pages, etc. Pause Game On Enable, Content Objects, and content Index are variables being made during this program. Pause Game On Enable is a boolean variable that represents the state of the game scene, it is used to pause the game scene when the user is on any of the user interface, and to continue the game scene when the user leaves the user interface. Content Objects is an array that stores a set ofgame objects that represents each slide of the tutorial. Content Index is an integer variable that initially set to 0, which is used to call the tutorial slides stored in the Content Object. When user opens any of the interface, procedure Freeze Game will run to set Time. time scale to 0 in order to freeze the game scene, while the user is on the interface. The Content Object with index 0, which stores the first slide of the tutorial, will then be displayed to the user's screen. If the user clicks "continue" button on the tutorial interface, then the procedure Switch Contents will run [15]. It will first deactivate the current Content Object displayed on the user's screen, increment the Content Index, and then display the Content Object with the new index on the screen, which storesthe next slide of the tutorial. If the user clicks "return" button on the tutorial interface, then the procedure Switch Backwards will run. It will first deactivate the current Content Object displayed on the user's screen, decrement the Content Index, and then display the Content Object with the new index on the screen, which stores the previous slide of the tutorial.

A key part of the simulator is the physical boat simulation. This system is integral to the simulator's design because it describes and calculates forces on the boat done by fluids that affects the state of the boat. The important Rigid Body component is added on the boat to make it interact with the physics system in Unity, giving the physics properties to boat that can make it behave realistically.

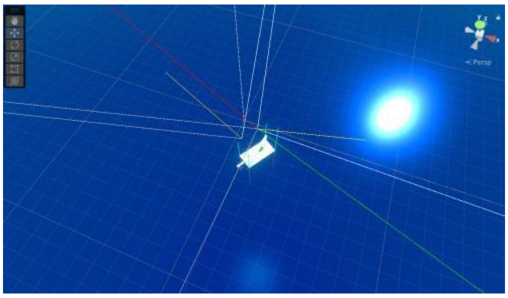

Figure 3. Screenshot of the Unity

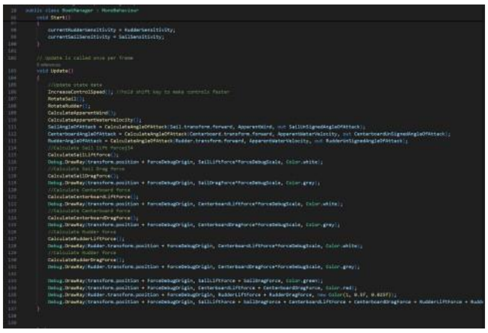

Figure 4. Screenshot of code 2

The program run one per frame when the user has entered the game scene. The important Rigid Body component is added to the boat to give it physics properties and to interact with the physics system in Unity. On top of the Unity Rigid body, a system for describing forces through a fluid were implemented. Wind Vector, Boat State Data, and Boat Forces are the main variables being made during this program. Wind Vectors is a 2D variable, it is used to represent the magnitude and direction of wind on a 2D plane. Boat State Data holds a set of variables, for example the Sail Angle Of Attack, which is used to calculate a series of forces that Wind Vector caused that effecton the sail. Boat Forces holds a set of variables that stores the values of the forces been calculated.

When the game has started, procedure Generate Wind will be called to generate a new variable stored in Current True Wind, which holds a x axis and z axis value. It will run and regenerate another value after a period of time. Then, procedure Rotate Sail and Rotate Rudder will be called to detect any adjustments made by the user to the state of the game object Sail and Rudder in the simulator and store their new position. Next, procedure Calculate Apparent Wind and Calculate Apparent Water Velocity will be called to calculate the Apparent Wind and Apparent WaterVelocity from Current True Wind and Boat Velocity, which will be used later in the calculation of forces on the boat. Then, function Calculating Angle Of Attack will be called 3 times with different parameters, including state of game objects, Apparent Wind and Apparent Water Velocity, to calculate variables Sail Angle Of Attack, Center board Angle Of Attack, and Rudder Angle Of Attack through real physics formulas. Lastly, 6 functions will be called each to calculate the value of lift and drag force on the 3 game objects: Sail, Rudder and Centerboard. Real data of angle of attack and coefficient of lift and drag was used in the formulas. Lastly, these calculations are applied to the game object sailboat, where these forces cancel each other out and produce a resultant force that moves the boat.

The VR component of the program is responsible for giving the user an immersive first person experience, and for interfacing the user with the program through motion controller input. The motion controllers allow the user to realistically interact with the rudder and the sail, while the XR camera tracking allows the user to look around by simply turning their head.

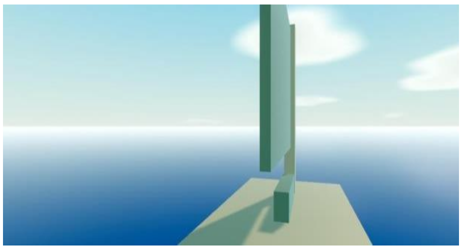

Figure 5. Screenshot of the game page

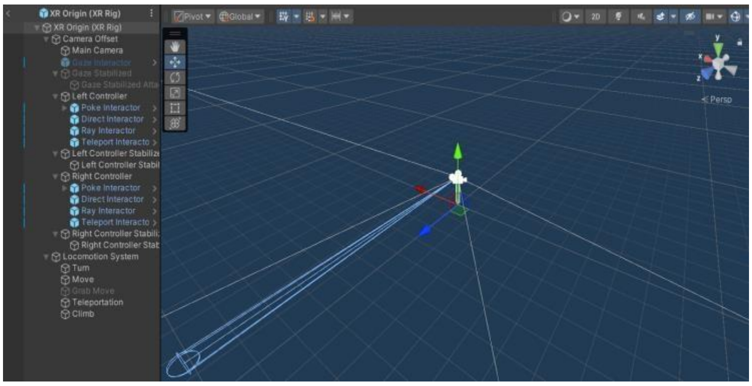

Figure 6. Screenshot of code 3

To implement VR in Unity, we used the Unity XR Toolkit package, which integrates multiplatform VR development tools into the game engine [14]. To set up the VR experience in our simulation, we replaced the typical unity camera with an XR Origin object, which is a configurable object that represents the VR player in the scene. We configured this XR origin to interface with the Meta Quest 2 VR headset. Once initialized into the scene, the XR origin is able to interface with the controls of the boat and properly track the position and rotation of the user’s VR headset.

4. EXPERIMENT

4.1. Experiment 1

Experiment A is to determine if the simulated boat speeds in the VR sailing simulator accurately reflect real-world conditions under varying wind speeds, sail angles, and boat orientations.This will ensure that players develop a realistic sense of boat speed changes, crucial for effective sailing skills.

This experiment aims to validate the accuracy of boat speeds in a VR sailing simulator. Ten participants will adjust the mainsail angle and boat direction relative to varying wind speeds (5, 10, and 15 knots). Each participant will test different combinations of sail angles and boat orientations to record the simulated boat speeds. The collected data will be compared with real-world Laser sailboat speeds under similar conditions to assess the simulator's realism. Accurate simulation is crucial for ensuring that players develop a realistic sense of speed changes, enhancing their sailing skills through the virtual environment.

Wind Speed (knots)	Sail Angle (relative)	Boat Direction (relative to wind)	Simulated Boat Speed (knots)	Real Boat Speed (knots)
5	0-30°	Upwind	2.5	2.95
5	0-30°	Beam Reach	3.93	4.03
5	0-30°	Broad Reach	1.62	1.28
5	0-30°	Downwind	1.23	1.6
5	30-60°	Upwind	3.4	3.61
5	30-60°	Beam Reach	1.08	1.55
5	30-60°	Broad Reach	4.33	4.04
5	30-60°	Downwind	1.73	1.41
5	60-90°	Upwind	2.22	2.24
5	60-90°	Beam Reach	2.73	2.52
5	60-90°	Broad Reach	3.45	3.09
5	60-90°	Downwind	2.17	2.03
10	0-30°	Upwind	5.1	5.39
10	0-30°	Beam Reach	2.8	2.81
10	0-30°	Broad Reach	6.33	5.88
10	0-30°	Downwind	6.47	6.14
10	30-60°	Upwind	1.59	2.03
10	30-60°	Beam Reach	9.69	10
10	30-60°	Broad Reach	3.74	3.34
10	30-60°	Downwind	7.16	7.1
10	60-90°	Upwind	2.1	2.09
10	60-90°	Beam Reach	1.31	1.72
10	60-90°	Broad Reach	3.33	3.49
10	60-90°	Downwind	3.81	3.83
15	0-30°	Upwind	8.65	8.34
15	0-30°	Beam Reach	14.57	14.85
15	0-30°	Broad Reach	14.15	14.55
15	0-30°	Downwind	9.37	9.79
15	30-60°	Upwind	2.24	1.93
15	30-60°	Beam Reach	1.63	1.46
15	30-60°	Broad Reach	6.44	6.21
15	30-60°	Downwind	12.6	12.46
15	60-90°	Upwind	4.93	4.98
15	60-90°	Beam Reach	2.97	3.28
15	60-90°	Broad Reach	2.04	2.53
15	60-90°	Downwind	11.81	11.51

Figure 7. Figure of experiment 1

The mean simulated boat speed is 4.87 knots, and the median is 3.43 knots. The lowest simulated speed is 1.08 knots, while the highest is 14.57 knots. For real boat speeds, the mean is 4.89 knots, with a median of 3.42 knots. The lowest real speed is 1.28 knots, and the highest is 14.85 knots. The simulated and real speeds are closely aligned, suggesting that the simulator accurately reflects real-world conditions. However, the slight discrepancies, especially in higher wind speeds, may surprise some. This could be due to the variability in real sailing conditions that are hard to perfectly model in a simulator. The biggest effect on results is the wind speed, which directly influences both simulated and real boat speeds. The alignment between simulated and real data suggests that wind speed settings and sail angle adjustments in the simulator are crucial for realistic training.

4.2. Experiment 2

Experiment B is to determine whether the sail angle adjustments in the VR sailing simulator accurately reflect the real-world effort and mechanics of adjusting the main rope on a sailboat. This ensures that players develop realistic expectations about the physical effort required to change the sail angle, avoiding misconceptions about sailing mechanics.

This experiment tests the accuracy of sail angle adjustments in a VR sailing simulator compared to real-world conditions. Five participants use a gamepad to simulate tightening or loosening the mainsail's rope by swinging the gamepad, while the simulator records the change in sail angle. The data collected is averaged to find the mean sail angle change per swing. This is then compared to the calculated ideal angle change based on real-world data to assess if the simulator accurately replicates the effort required to adjust the sail angle, ensuring a realistic training experience for users.

Participant	Swing Number	Simulated Sail Angle Change (degrees)	Realistic Sail Angle Change (degrees)
1	1	1.06	1.48
1	2	1.6	1.3
1	3	0.73	1.08
1	4	0.59	1.43
1	5	1.4	1.35
1	6	0.53	1.48
1	7	1.75	1.11
1	8	0.77	1.09
1	9	0.96	1.26
1	10	1.15	1.15
2	1	1.42	1.07
2	2	0.94	1.18
2	3	1.18	1.39
2	4	0.8	1.26
2	5	1.39	1.02
2	6	1.41	1.09
2	7	0.6	1.47
2	8	1.95	1.4
2	9	0.96	1.05
2	10	1.53	1.22
3	1	0.68	1.25
3	2	0.55	1.45
3	3	0.89	1.33
3	4	0.97	1.26
3	5	1.32	1.09
3	6	1.95	1.39
3	7	1.91	1.45
3	8	1.4	1.46
3	9	0.63	1.1
3	10	0.57	1.16

28	3	7	1.91	1.45
29	3	8	1.4	1.46
30	3	9	0.63	1.1
31	3	10	0.57	1.16
32	4	1	1.08	1.14
33	4	2	1.74	1.18
34	4	3	0.92	1.27
35	4	4	0.71	1.4
36	4	5	0.61	1.49
37	4	6	1.66	1.1
38	4	7	0.51	1.41
39	4	8	1.56	1.36
40	4	9	1.66	1.04
41	4	10	1.04	1.06
42	5	1	1.79	1.31
43	5	2	1	1.03
44	5	3	0.97	1.16
45	5	4	1.59	1.32
46	5	5	1.83	1.24
47	5	6	0.68	1.36
48	5	7	1.64	1.28
49	5	8	1.66	1.25
50	5	9	1.28	1.21
51	5	10	0.54	1.05

Figure 8. Figure of experiment 2

The mean simulated sail angle change is 1.16 degrees, with a median of 1.07 degrees. The lowest simulated change is 0.51 degrees, and the highest is 1.95 degrees. For realistic sail angle changes, the mean is 1.25 degrees, and the median is 1.26 degrees, with a minimum of 1.02 degrees and a maximum of 1.49 degrees. The data shows that simulated changes are slightly less than realistic changes, suggesting that the simulator may underrepresent the effort required to adjust the sail angle in reality. This discrepancy could be due to the simulator's sensitivity settings, which may need calibration to better reflect real-world conditions [1]. The biggest effect on results is the swing amplitude, which directly influences sail angle changes. Ensuring the simulator's sensitivity aligns with real-life data is crucial for accurate training outcomes.

5. RELATED WORK

The physics-based watercraft simulator in virtual reality utilizes the Unreal Engine to create a realistic simulation of watercraft interacting with near-shore waves. By integrating hydrodynamic data from the FUNWAVE model, the simulator accurately represents wave dynamics and their effects on watercraft, providing an immersive training environment for navigation and coastal operations. This approach enhances user understanding of wave impact on vessel control but requires significant computational resources due to large data sets and complex physics calculations, limiting real-time performance on standard systems [2].

The Virtual Reality Scene Construction Technology for Human-Machine Interaction on the Sea creates an immersive 3D simulation environment for maritime operations. Utilizing VR hardware and software, this technology simulates realistic sea conditions, allowing users to interact with virtual elements such as boat controls and respond to changing sea states. The methodology enhances training by improving situational awareness and decision-making skills in a controlled virtual environment. However, it has limitations such as high setup costs and a lack of detailed physical modeling for sailing dynamics [3].

The simulation of real water in 3D animation employs advanced computer graphics techniques, such as reflection, refraction, and dynamic texture mapping, to create visually realistic water effects. Using 3ds Max and RealFlow software, this approach generates detailed animations of waves, splashes, and foam to enhance the realism of virtual environments, naval simulations, and games. While it excels in visual accuracy, this methodology focuses on aesthetics over physical realism, neglecting the hydrodynamic forces critical for sailing simulations [4].

6. CONCLUSIONS

One limitation of my project is the approximation in physics calculations. Due to the lack of precise formulas for calculating forces on a sailboat from the complex data graphs I found, I had to manually approximate these forces by redrawing the graphs, which may reduce the accuracy of the simulation. To improve this, obtaining more precise mathematical models or collaborating with experts to derive exact formulas would enhance the simulator's realism. Another limitation is the absence of a competitive mode, which is crucial for training on competitive sailing techniques like route planning and tactical maneuvers. Adding a competitive mode would allow users to practice these critical skills. If given more time, I would implement a multiplayer feature that simulates real race conditions and strategies, providing a comprehensive training environment that mirrors the competitive nature of Laser sailing, enhancing both the realism and educational value of the simulator [13].

In conclusion, my VR sailing simulator effectively combines visual realism with accurate physics to provide a comprehensive training tool. While it has limitations in physics calculations and lacks competitive modes, further improvements can enhance its accuracy and versatility, making it an invaluable resource for both novice and experienced sailors.

REFERENCES

[1] Darrah, Marjorie, et al. "Are virtual labs as effective as hands-on labs for undergraduate physics? Acomparative study at two major universities." Journal of science education and technology 23 (2014): 803-814.

[2] Tatli, Zeynep, and Alipasa Ayas. "Effect of a virtual chemistry laboratory on students' achievement." Journalof Educational Technology & Society 16.1 (2013): 159-170.

[3] Balamuralithara, Balakrishnan, and Peter Charles Woods. "Virtual laboratories in engineering education: Thesimulation lab and remote lab." Computer Applications in Engineering Education 17.1 (2009): 108-118.

[4] Yaron, David, et al. "The ChemCollective — virtual labs for introductory chemistry courses." Science328.5978 (2010): 584-585.

[5] Anthes, Christoph, et al. "State of the art of virtual reality technology." 2016 IEEE aerospace conference.IEEE, 2016.

[6] Jayaram, Sankar, et al. "Assessment of VR technology and its applications to engineering problems." J.Comput. Inf. Sci. Eng. 1.1 (2001): 72-83.

[7] Silva Junior, Andouglas Goncalves da, et al. "Towards a real-time embedded system for water monitoringinstalled in a robotic sailboat." Sensors 16.8 (2016): 1226.

[8] Barnes, Matthew A., et al. "Environmental conditions influence eDNA persistence in aquatic systems."Environmental science & technology 48.3 (2014): 1819-1827.

[9] Harris, Ruth A., and Steven M. Day. "Dynamic 3D simulations of earthquakes on en echelon faults."Geophysical Research Letters 26.14 (1999): 2089-2092.

[10] Walls, Justin, et al. "Assessment of upwind dinghy sailing performance using a virtual reality dinghy sailingsimulator." Journal of science and medicine in sport 1.2 (1998): 61-72.

[11] Laidlaw, David H., et al. "Comparing 2D vector field visualization methods: A user study." IEEE Transactions on Visualization and Computer Graphics 11.1 (2005): 59-70.

[12] Wu, Shin-Ting, and Marcelo de Gomensoro Malheiros. "Interative 3D geometric modelers with 2D UI."(2002).

[13] Vangelakoudi, A., I. Vogiatzis, and N. Geladas. "Anaerobic capacity, isometric endurance, and Laser sailingperformance." Journal of sports sciences 25.10 (2007): 1095-1100.

[14] Juránek, Vojtěch. "Virtual reality toolkit for the Unity game engine." Bakal árskáprace, Masarykova Univerzita, Fakulta informatiky (2021).

[15] Van Aalst, J. W., C. A. P. G. Van Der Mast, and T. T. Carey. "An interactive multimedia tutorial for userinterface design." Computers & Education 25.4 (1995): 227-233.