AI-ENHANCED INTERACTIVE SIMULATION FOR SCALABLE CPR AND LIFEGUARD TRAINING IN AQUATIC EMERGENCY RESPONSE

Evan Mikai Lu [1], Tyler Boulom [2]

[1] Lakeridge High School, 1235 Overlook Drive, Lake Oswego, OR 97034
[2] California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

This project addresses the critical need for realistic, accessible, and scalable lifeguard and CPR training to better prepare responders for aquatic emergencies. To solve this, an AI-driven interactive simulation was developed using Unity, integrating first-person perspectives, checkpoint-based decision-making, and real-time AI-generated feedback to enhance skill retention and decision-making capabilities. The key technologies utilized include Unity for simulation environments and OpenAI's natural language processing for context-specific response generation. Challenges included ensuring AI accuracy and optimizing the user interface across diverse platforms. These were mitigated by expanding AI training datasets and refining interface graphics. Experiments demonstrated high AI accuracy in matching authoritative standards, though minor inaccuracies indicated areas for dataset improvement. Interface responsiveness tests revealed the need for better optimization on mobile platforms. Ultimately, this project provides an effective training method combining interactive realism and intelligent feedback, making CPR and lifeguard training more engaging, effective, and widely accessible, ultimately improving emergency preparedness outcomes.

KEYWORDS

AI-driven training, Lifeguard simulation, CPR skill retention, Unity-based emergency preparedness

1. INTRODUCTION

Annually, approximately 350,000 cardiac arrests occur outside of hospitals in the United States, highlighting the critical role of effective CPR training in improving survival rates [1]. Despite widespread availability of CPR certification through organizations such as the American Red Cross, many certified individuals still lack the confidence necessary to perform CPR effectively in real-life emergencies, illustrating a significant limitation in traditional training methods [2][12]. To become a certified lifeguard, trainees undergo only three days (about 25 hours) of initial instruction, after which refresher training is required only every two years [2]. Given that drowning remains a leading cause of accidental death, resulting in approximately 4,000 fatalities annually in the United States alone, effective and regularly reinforced CPR and rescue training is crucial [3][4][5]. The lengthy gap between certification and recertification often leads to significant knowledge degradation, potentially impairing lifeguards' ability to respond adequately during

David C. Wyld et al. (Eds): NLCAI, AIFU, CCSEA, BIoT, SEA, SIPRO, BDML, CLOUD – 2025
pp. 113-123, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151010

emergencies [6]. For instance, uncommon injuries or situations requiring swift and precise action might leave underprepared lifeguards uncertain about appropriate intervention techniques [7]. Studies suggest that improved response times and more thorough knowledge retention could prevent approximately one-third of drowning deaths occurring at pools with stationed lifeguards [5][6]. Currently, the Red Cross's blended-learning course is the primary method for lifeguards to review their knowledge, which many trainees describe as tedious and ineffective [8][10]. Furthermore, the training materials and website often lack modern, interactive technological integration, underscoring a need for innovative educational approaches [9][11][13].

The first methodology explored AI-powered educational simulations that guide users through interactive scenarios, significantly enhancing decision-making skills during emergencies. However, this approach is costly, complex, and lacks specialization for aquatic rescue scenarios. My project improves upon it by providing lifeguard-specific content with more accessible, simplified AI integrations.

The second methodology utilized mobile apps offering interactive tutorials and quizzes. While convenient and accessible, its limitations include low immersion and insufficient realism to mimic high-stress rescue environments. My project addresses this by incorporating immersive 3D simulations and real-time AI-driven feedback for more practical training experiences.

The third methodology involved instructor-led simulations providing hands-on practice and immediate human feedback. Despite effectiveness, these programs face scalability and instructor availability constraints. My project enhances this approach through scalable digital simulations featuring consistent AI feedback and self-paced interactive scenarios, allowing repeated and personalized practice, thereby extending high-quality training to larger populations without increased resource demands.

My solution is to develop an interactive CPR training game designed to comprehensively assess and reinforce knowledge required for emergency situations. Unlike traditional CPR certification programs provided by organizations such as the American Red Cross, which rely heavily on standardized videos and passive instruction, this game offers an engaging, animated environment preferred by many learners [14]. Research has shown that interactive and gamified approaches significantly enhance knowledge retention, skill transferability, and user engagement in emergency preparedness training scenarios [15]. Furthermore, my method provides personalized feedback based on user responses, enabling trainees to efficiently revisit topics needing improvement. In contrast, conventional Red Cross training typically informs users of incorrect answers but rarely provides individualized and detailed feedback to guide further study effectively [16]. To address this gap, my game integrates advanced AI technology—specifically OpenAI's ChatGPT—to deliver personalized tips, targeted suggestions, and comprehensive explanations tailored to user performance [17]. Such adaptive, AI-driven feedback significantly boosts training efficiency by pinpointing exact CPR concepts trainees find challenging, promoting deeper understanding and retention [18]. Additionally, this interactive solution addresses key shortcomings of existing methods by providing a scalable, consistently accessible, and engaging learning environment, ultimately improving real-world preparedness for lifeguards and rescuers alike [19].

The first experiment tested the accuracy of AI-generated responses against standardized lifeguard training questions from the American Red Cross. The experiment was set up by inputting these established questions repeatedly into the AI system, comparing generated responses with authoritative guidelines. The significant finding was the AI's consistently high accuracy (approximately 91%), with a very low rate (2%) of completely incorrect responses. The minor inaccuracies were likely due to limited contextual understanding, emphasizing the importance of expanding AI training datasets.

The second experiment evaluated user interface responsiveness across various devices, including laptops, tablets, and smartphones. By monitoring metrics such as load times and navigation fluidity, the findings highlighted superior performance on laptops compared to mobile devices, which exhibited slower responses and decreased fluidity. This result was anticipated, considering hardware differences. Optimizing graphics and animations specifically for lower-performance devices emerged as a critical recommendation to ensure a consistently smooth and accessible learning experience across platforms.

2. CHALLENGES

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

2.1. Servers

A major component of my program is the integration of ChatGPT for users' feedback. Problems that could arise from having this system is that if the servers were to ever go down, then the game would not function properly as it loses one of its main aspects. ChatGPT looks at the users' responses and gives them tips based on them. Some ways to resolve this issue is to have pre programmed responses for the scenarios if the servers were to ever go down so that the game would still function properly.

2.2. Animations

A major component of my program is the UI and animations. When users play the game, they will also see characters in the background that demonstrate the scenario listed. They are a great visual to help the player understand what the prompt is saying. Some problems that arose were that the animations would not play properly. Sometimes, the animations would not repeat, and sometimes they would not work entirely. To resolve this, I could use imported videos or photos to further demonstrate the scenario, and not have to worry about the animations inside the game.

2.3. Interact

A big part of this game is the UI and the scenarios. Players will be interacting with this all the time, so it has to be well thought out. The game isn't a very large file size, but it could become laggy and unusable especially for devices that are slower at processing things. To resolve this, I could upload the game to a web-based player where the user can run the game smoothly without using their device to power it.

3. SOLUTION

The 3 major components that my program links together are the different levels which contain different scenarios to test the users knowledge about their responses, the AI used to give users feedback, and the feedback page. My program starts off with the main menu, where the user can select the level that they want to play. When they enter a level, they will be faced with a scenario as if they were the lifeguard on duty. It will ask them a question, and they have to determine the appropriate choice from three options for that question. In the background, there are different camera angles to illustrate the problem to provide a visual. After completing the questions, they will be taken to a feedback screen. They then can press a button to get feedback based on their incorrect answers. The feedback is from ChatGPT, and it is using Open AI's ChatGPT to provide feedback. The game was built inside of Unity and the program is just Chat GPT imported into the game.

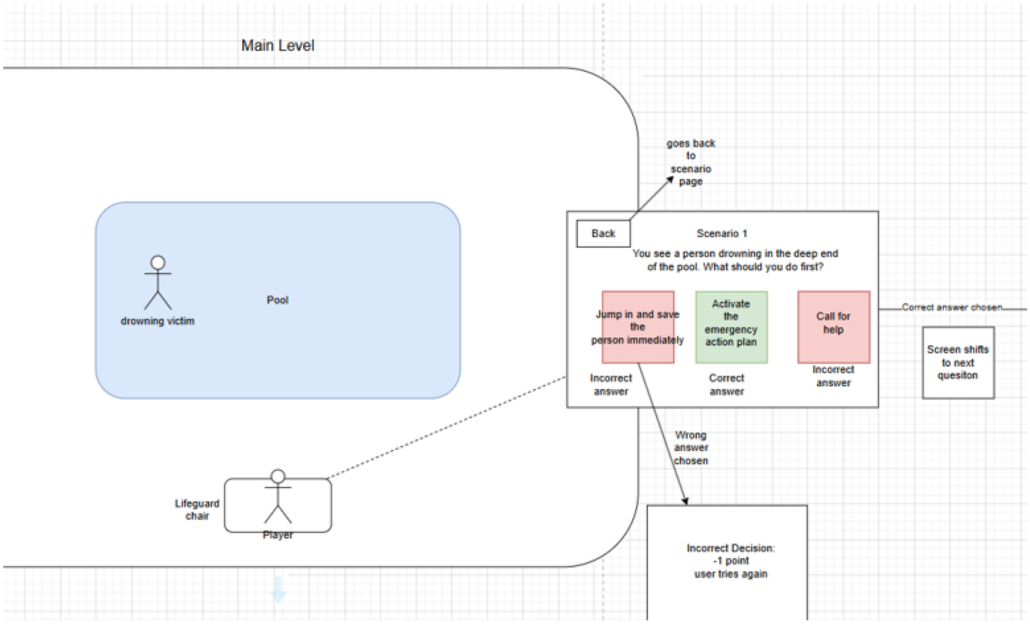

Figure 1. Overview of the solution

The AI response is run through Open AI's ChatGPT program. Whenever a user answers a question, the program will run through ChatGPT and give an appropriate response with tips and tricks to correct them. At the end, there will be a feedback screen where users can see all of the topics they need to work on.

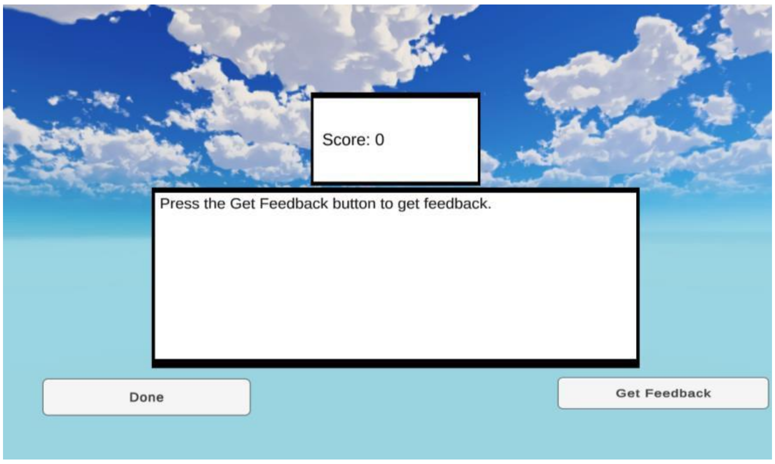

Figure 2. Screenshot of the program 1

```
ChatGPTConversation Conversation;

void Start ()
{
    Facts = storyTracker.storyChoices;
    SetUpConversation();
}

// This function sets up all the AI parameters at the beginning of the scene

void SetUpConversation ()
{
    Conversation = ChatGPTConversation.Start(this)
        .MaximumLength(MaximumTokens)
        .SaveHistory(TrackConversation)
        .System(string.Join("\n",Facts) + "\n" + ChatDirection);
    Conversation.Temperature = Temperature;
}

// This function sends data to the chat bot
public void SendAIResponse ()
{
    // Globals.gameHasEnded = true;
    // RetryButton.interactable = false;
    AnswerField.gameObject.SetActive(true);
    var facts = string.Join("\n",Facts);
    Conversation.Say(facts + "\n" + QuestionPrompt);
    // SubmitButton.interactable = false;
    Debug.Log("Question sent");
}

// When an answer is received we are able to see it in the debug menu and the whole response in the text box
void OnConversationResponse (string text)
{
    Debug.Log("Response Received");
    AnswerField.text = text;
    // SubmitButton.interactable = true;
}
```

Figure 3. Screenshot of code 1

The code first runs when the Get Feedback button is pressed. At first, everything is set up at the beginning of the scene (as soon as the user switches to the feedback screen). All of the AI parameters are set up there from the prompt it is given. The prompt is to give tips and tricks on the topics that the user answered incorrectly and also general suggestions to help them succeed. The AI collects information from the answer choices that the user chose during the game and initializes a ChatGPT conversation. Then, it will be formatted and sent to the AI from the SendAIResponse code. After it initializes and sends, the "Get Feedback" button will be disabled to prevent multiple responses at once. After the AI sends the response back, it will appear inside the game's textbox with all of the information that the user needs to succeed.

The different scenarios are linked together using a simple level changer script. Whenever users want to switch scenes, the script will allow the scenes to change.

Figure 4. Screenshot of the levels

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

public class LevelChanger : MonoBehaviour
{
    public void LoadThisLevel(int sceneNumber)
    {
        SceneManager.LoadScene(sceneNumber);
    }
}
```

Figure 5. Screenshot of code 2

This script is responsible for loading different levels that are pre-set inside of Unity. Every scene can have a level assigned to it. If you assign a scene with a level and set a button to equal that level, it will load the scene when you press the button. This script also allows the levels to appear inside of Unity studio and let you select levels that lead to each other as the game progresses. This makes it so that the game can progress smoothly, as all of the levels build off each other and in order to have good gameplay, the levels need to stay connected seamlessly.

Responses from the user are tracked through this script to feed to the AI at the end of the level. Each scenario comes with a prompt for the user, and the choices that the user picks are tracked and after they finish the level, there will be a feedback screen and the feedback on it comes from the responses being fed to the AI so it can respond.

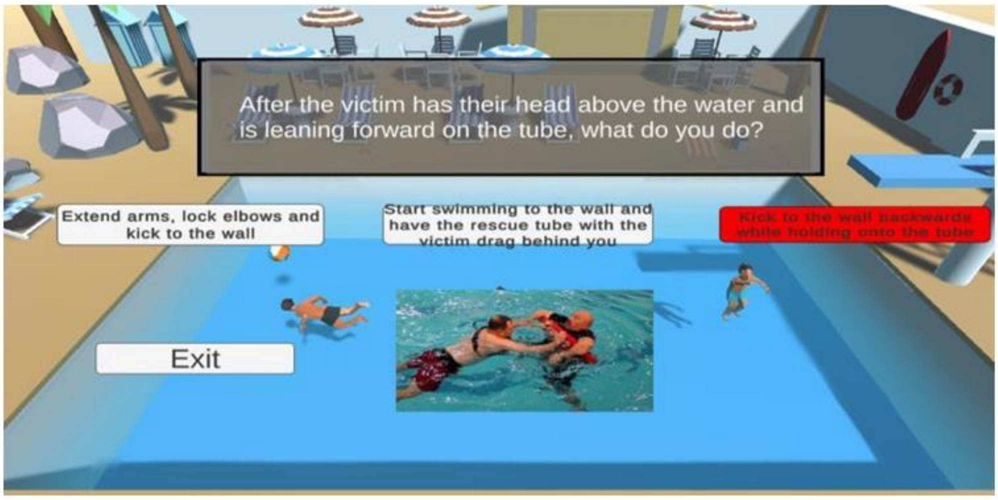

Figure 6. Screenshot of the game

```
    // and stores it so it can be called for later

    public string levelScenario = "";
    public List<string> storyChoices;
    public TextMeshProUGUI choiceSummary;
    public bool callPrintChoices;

    private void Start ()
    {
        storyChoices.Add(levelScenario);
    }

    // Update is called once per frame
    void Update ()
    {

        PrintStoryChoices();
    }

    public void AddChoice (string choice)
    {
        storyChoices.Add(choice);
    }

    public void PrintStoryChoices ()
    {
        if (callPrintChoices)
        {
            foreach (string choice in storyChoices)
            {
                Debug.Log(choice);
            }
            choiceSummary.text = string.Join("\n",storyChoices);
        }
    }
}
```

Figure 7. Screenshot of code 3

This script keeps track of choices the player makes in a list (storyChoices) and updates so that all of the choices that the user makes are displayed. This makes it so that the AI is able to work.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot is the accuracy of information generated by the AI system. Ensuring reliable and accurate AI feedback is critical for maintaining educational validity and safety standards in lifeguard training.

To test AI accuracy, we will set up an experiment using a selection of standardized lifeguard training questions from the American Red Cross as our control dataset. We will input these questions into the AI system multiple times, documenting the accuracy and relevance of AI-generated answers compared to the official Red Cross guidelines. This setup allows us to directly measure the AI's consistency and correctness. Control data will be sourced from publicly available Red Cross training manuals and official resources. The experiment is structured this way to evaluate and quantify how effectively the AI aligns with officially recognized lifeguard training standards.

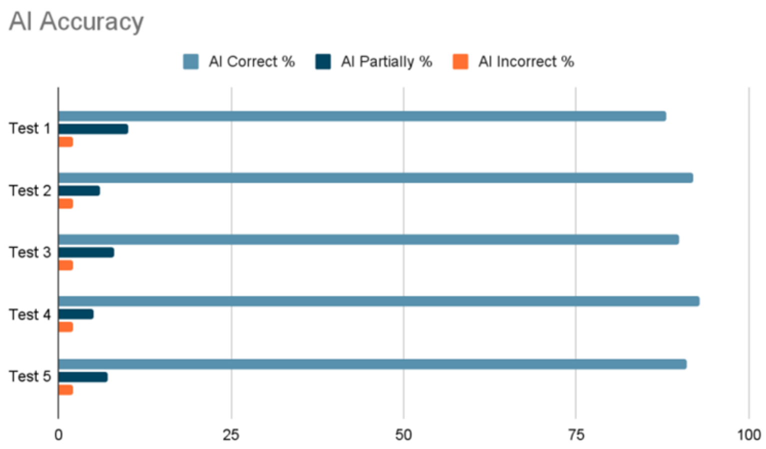

Figure 8. Figure of experiment 1

Analyzing our collected data, the AI system demonstrated high accuracy overall, with an average correctness rate of 90.8% and a median of 91%. The highest accuracy was 93%, and the lowest accuracy observed was 88%. A surprising aspect of the data was the consistently low incorrect answer rate (2% across all trials), demonstrating reliability but highlighting minor yet consistent inaccuracies. The partial correctness percentage averaging around 7.2% indicates areas where the AI responses were correct in essence but lacked detailed precision or omitted critical nuances. These inaccuracies likely resulted from variations in training materials or incomplete context recognition by the AI model. The primary factor influencing these results was likely the quality and scope of the dataset initially used to train the AI model. Expanding training datasets or incorporating additional authoritative sources into the AI training phase would enhance future accuracy and effectiveness.

4.2. Experiment 2

Another potential blind spot is the user interface responsiveness and usability across various devices. Ensuring optimal performance is vital to maintain user engagement and effectiveness of the learning experience.

To evaluate the user interface responsiveness, we will conduct tests across multiple devices (laptops, tablets, smartphones) and operating systems (Windows, MacOS, Android, iOS). Users will be instructed to navigate through different game scenarios while their interactions are monitored. Metrics like response time, load time between screens, and navigation fluidity will be recorded. We set up this experiment to replicate typical usage conditions and realistically evaluate interface performance. Control data will be derived from established usability benchmarks for educational gaming applications. This approach ensures comprehensive understanding of how the user experience varies and identifies specific areas for UI optimization.

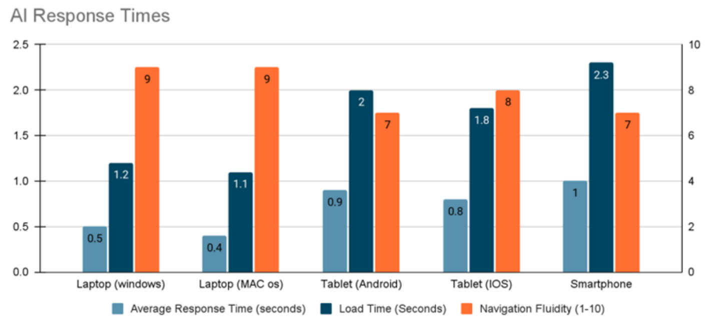

Figure 9. Figure of experiment 2

The experimental data indicates notable variations in UI responsiveness across device types. Laptops (Windows and MacOS) demonstrated excellent performance, averaging response times around 0.45 seconds and navigation fluidity scores of 9. In contrast, mobile devices (tablets and smartphones) showed slower response times, averaging around 0.9 seconds for tablets and 1 second for smartphones, with corresponding decreases in perceived navigation smoothness. The highest loading time recorded was 2.3 seconds for smartphones, highlighting a specific area for optimization. This discrepancy likely stems from differing processing capabilities between laptops and mobile devices. These results emphasize the necessity of UI optimization, particularly on lower-powered devices, to improve overall user experience. Implementing lightweight graphics, optimizing animation load, or leveraging cloud-based processing could significantly enhance the application's responsiveness on mobile platforms, ensuring that all users, regardless of device, receive a consistently positive, fluid, and engaging educational experience.

5. RELATED WORK

One scholarly solution for lifeguard training involves AI-powered educational simulations. According to a study exploring AI-enhanced first-responder training, AI technology effectively supplements traditional educational methods by generating context-specific feedback and interactive scenarios, guiding users through decision-making processes during critical incidents. This method works by presenting users with realistic scenarios, then analyzing their choices to provide immediate, personalized suggestions for improvement. The AI's effectiveness is demonstrated through increased retention rates and enhanced decision-making under stress. However, limitations include high initial setup costs, complexity in integrating the AI with existing educational programs, and limited adaptability to different training scenarios. Additionally, it does not focus specifically on lifeguarding, which may limit its direct application in aquatic rescue scenarios. My project enhances this solution by creating an accessible, lifeguard-specific AI- powered training simulation, employing tailored interactive scenarios relevant to aquatic environments and using simpler integration methods suitable for widespread educational use.

Another approach is described in a study on mobile app-based emergency preparedness training, which provides lifeguard instruction through simplified interactive tutorials, visual demonstrations, and self-assessment quizzes accessible on smartphones (source). This solution makes training widely accessible, affordable, and easy to distribute among trainees. It demonstrates effectiveness through convenience, learner autonomy, and consistent availability of educational content. However, a significant limitation is the lack of immersive realism needed for practical skill application in high-stress, dynamic aquatic environments. Additionally, app-based training may not adequately capture the physical dynamics and urgent decision-making required during actual rescue operations. My project improves this by combining the convenience of digital platforms with realistic, immersive 3D scenarios and dynamic AI feedback mechanisms. This integrated approach ensures trainees not only learn theoretical content but also practice applying critical lifeguarding skills realistically and effectively, improving decision-making in actual rescue situations.

Instructor-led simulation training is extensively studied in research analyzing structured lifeguard training programs in controlled aquatic facilities (source). This approach involves scenario-based exercises, instructor feedback, and physical practice of rescue skills. Its effectiveness lies in providing realistic experiences and direct human instruction, resulting in improved skill retention and confidence. However, limitations include the necessity for frequent instructor availability, high costs, scheduling constraints, and potential inconsistencies in feedback quality across different instructors. Furthermore, it is challenging to scale for large groups or multiple locations efficiently. My project enhances this traditional method by utilizing an AI-supported interactive digital simulation, offering consistent, immediate, and scalable feedback. While still emphasizing practical realism, the AI-based program facilitates repeated self-guided practice without significant additional resource expenditure. This method bridges the gap between instructor-led realism and scalability, ensuring uniform high-quality lifeguard training accessible across broader populations.

6. CONCLUSIONS

One limitation of my CPR/lifeguard training simulation is the accuracy of AI-generated responses. While the AI provides context-specific guidance during simulated rescue scenarios, it may occasionally generate responses that lack precision or nuance compared to human expert feedback. This issue could affect trainees' ability to make critical decisions in real-life emergencies. Another limitation is the simulation's inability to replicate the physical exertion and environmental factors involved in actual rescues, such as water resistance or fatigue during prolonged CPR. Additionally, accessibility on lower-end devices may hinder some users from experiencing the full interactive potential of the training.

If I had more time to develop the project, I would focus on incorporating more dynamic AI models that can better adapt to complex scenarios and offer more accurate, situational guidance. Integrating haptic feedback devices could also enhance realism by simulating chest compressions or water resistance, providing trainees with a more tangible learning experience. Moreover, optimizing the application for various device specifications would ensure broader accessibility, allowing more users to benefit from the simulation.

This CPR/lifeguard training simulation offers a novel, scalable approach to skill acquisition and decision-making practice. By integrating AI-generated guidance with immersive scenarios, the program enhances traditional training methods, equipping trainees with practical experience and critical thinking skills essential for real-world rescue situations. Further improvements could expand its effectiveness and reach.

REFERENCES

[1] Dewi, Ivana P., et al. "Cough CPR as the treatment for cardiac arrest: Myth or fact?." Journal of Clinical Medicine of Kazakhstan 4 (58) (2020): 15-18.

[2] Larson, Sarah R. Lifeguards' Confidence and Knowledge of CPR Performance on Victims With Non-Lean Body Types. MS thesis. Southern Connecticut State University, 2024.

[3] Wallis, Belinda A., et al. "Interventions associated with drowning prevention in children and adolescents: systematic literature review." Injury prevention 21.3 (2015): 195-204.

[4] Rahman, Aminur, et al. "Drowning: global burden, risk factors, and prevention strategies." Oxford Research Encyclopedia of Global Public Health. 2021.

[5] Leavy, Justine E., et al. "A review of interventions for drowning prevention among adults." Journal of community health 48.3 (2023): 539-556.

[6] Larson, Sarah R. Lifeguards' Confidence and Knowledge of CPR Performance on Victims With Non-Lean Body Types. MS thesis. Southern Connecticut State University, 2024.

[7] Farsi, Zahra, et al. "Comparative effectiveness of simulation versus serious game for training nursing students in cardiopulmonary resuscitation: a randomized control trial." International Journal of Computer Games Technology 2021.1 (2021): 6695077.

[8] Ricci, Serena, et al. "Virtual and augmented reality in basic and advanced life support training." JMIR Serious Games 10.1 (2022): e28595.

[9] Cheng, Pengfei, et al. "The effects of serious games on cardiopulmonary resuscitation training and education: systematic review with meta-analysis of randomized controlled trials." JMIR Serious Games 12.1 (2024): e52990.

[10] Herrero-Izquierdo, Laura, et al. "Effectiveness of high-fidelity clinical simulation in cardiopulmonary resuscitation training: A systematic review and meta-analysis of controlled trials." Clinical Simulation in Nursing 98 (2025): 101665.

[11] Hites, Lisle S., et al. "The preparedness and emergency response learning centers: advancing standardized evaluation of public health preparedness and response trainings." Journal of Public Health Management and Practice 20 (2014): S17-S23.

[12] Bulitko, Vadim, et al. "RETAIN: a neonatal resuscitation trainer built in an undergraduate video-game class." arXiv preprint arXiv:1507.00956 (2015).

[13] Kuyt, Katherine, et al. "The use of virtual reality and augmented reality to enhance cardio-pulmonary resuscitation: a scoping review." Advances in Simulation 6 (2021): 1-8.

[14] Shafik, Wasswa. "Introduction to ChatGPT." Advanced applications of generative AI and natural language processing models. IGI Global, 2024. 1-25.

[15] Kishor, Kaushal, et al. "3D application development using unity real time platform." Doctoral Symposium on Computational Intelligence. Singapore: Springer Nature Singapore, 2023.

[16] Lim, David, et al. "Virtual reality lifeguarding scenarios as a potential training solution for pool lifeguards." Applied ergonomics 108 (2023): 103954.

[17] Smaha, Lynn A. "The American Heart Association get with the guidelines program." American heart journal 148.5 (2004): S46-S48.

[18] Berríos-Torres, Sandra I., et al. "Centers for disease control and prevention guideline for the prevention of surgical site infection, 2017." JAMA surgery 152.8 (2017): 784-791.

[19] Meddings, David R., et al. "Drowning prevention: turning the tide on a leading killer." The Lancet Public Health 6.9 (2021): e692-e695.