INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN PROGRESSIVE EDUCATION & DEVELOPMENT

OPEN ACCESS JOURNAL

Vol 15, Issue 1, (2026) E-ISSN: 2226-6348

Developing CircuitGo: A Game-Based Driving Simulator for Driver Education under Kurikulum Pendidikan Pemandu 02

Haziq Amani Bin Redzuan, Sharifalillah Nordin, Rozianawaty Osman, Azliza Mohd Ali

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Email: haziqamani579@gmail.com, sharifalillah@uitm.edu.my, rozianawaty@uitm.edu.my, azliza794@uitm.edu.my

DOI Link: http://dx.doi.org/10.6007/IJARPED/v15-i1/27830

Published Online: 24 March 2026

Abstract

Kurikulum Pendidikan Pemandu 02 (KPP 02) has been the basic module for driving education that focuses on skill based practical training in preparation for Jabatan Pengangkutan Jalan (JPJ)driving test in Malaysia. Yet, the traditional training method relies largely on a few physical practice sessions at driving institutes that potentially increase learner anxiety while also reducing exposure to multiple attempts at skill rehearsal and limiting familiarization of learners with the assessment examination circuit pre-assessment. However, the application of GBL frameworks in driving simulators is limited and does not align with the Malaysian KPP 02 curriculum as structured GBL approaches. In this work, we present CircuitGo that is a game-based driving simulator as a complement to the KPP 02 circuit module providing safe and interactive digital training environment. In terms of methodology, the study uses an adapted Agile development framework, mixed with principles of GBL instructional design to allow for iterative prototyping and user-centered refinement of the simulator. The system features a 3D environment to model the Shahbandar training circuit, embedding five key KPP 02 modules: Hill Start, S-Course, Z-Course, Side Parking and Three-Point Turn. Interactive tutorials, performance feedback, scoring systems, and tracking were also part of the GBL features. Evaluation of the system involved functionality and usability testing with 10 participants. Metrics were based on accuracy in completing tasks, the responsiveness of the controls, the ease of understanding the interface, and the perceived usefulness of the system for educational purposes. Evaluation results showed that all the modules worked as per the operational standards of JPJ, and the usability of the simulator is high. The results show that CircuitGo has the potential to improve the preparedness of learners, decrease their anxiety, and close the knowledge-skill gap. Future works include the mobile version, VR, and AI-based dynamic simulation of traffic.

Keywords: Game-Based Learning, Driving Simulator, KPP 02, Unity 3D, Driver Education, Serious Games

Introduction

Driver education is a very important part of making the roads safer and making sure that new drivers have the skills and knowledge they need to drive safely and well. The Road Transport Department (JPJ) in Malaysia has set up a structured program called Kurikulum Pendidikan Pemandu (KPP) to teach people how to drive. The KPP program has three parts: KPP 01, which teaches theory; KPP 02, which teaches how to drive on a circuit; and KPP 03, which teaches how to drive on the road. KPP 02 is an important part of Malaysia's driver education program because it teaches new drivers basic skills like how to start on a hill, park on the side of the road, and make a three-point turn in a driving circuit. These basic driving skills are very important for new drivers to learn in order to pass a driving test and get a temporary license. The KPP syllabus uses a systematic approach, but the way KPP 02 training is currently taught relies heavily on limited physical practice opportunities at driving schools. This could mean that learner drivers don't get enough chances to practice the hard parts over and over again before taking the real driving test. Earlier research have shown that a lot of learner drivers in Malaysia are very anxious and stressed during their practical driving tests (Lee, 2024; Sanjay, 2021). This is because they aren't used to the layout of the circuit and don't have many chances to practice. This could hurt their performance and make it more likely that they will fail practical tests.

The use of Technology-Enhanced Learning (TEL) as a medium for skill development, a notion that has been largely accepted, is a trend that is being further fueled by the "Dasar e-Pembelajaran Negara 2.0" (National e-Learning Master Plan 2.0) initiative by the government of Malaysia. Within the context of GBL, it is seen as an extremely effective learning strategy, where a variety of challenges, scores, and feedback mechanisms are being used to maximize student motivation, engagement, and knowledge retention (Hulme et al., 2021). Within the context of learning how to drive, GBL allows an individual to practice a variety of complex maneuvers in a safe, risk-free environment, which is crucial for learning the skills needed for driving. The viability of a virtual learning environment as a reliable alternative to a physical learning environment was also demonstrated during the COVID-19 pandemic (Choo et al., 2022). Furthermore, it is seen as a simulation training strategy that not only allows for better driving skills but also results in a reduction of accidents in the long term (Bruck et al., 2021).
There are a lot of digital tools that can be used to help people learn how to drive in Malaysia, but most of them are more focused on testing and learning theories, like KPP 01 preparation using a quiz format. MySekolah Memandu and KPP Test Malaysia 2021 are two of these tools. They have multiple-choice questions, but they cannot give you a realistic experience of driving on the KPP 02 circuit. But most people who are learning to drive can't afford professional simulators that give them a real learning experience.

To fill the current gap in knowledge, a new study is suggesting a new game-based driving simulator called CircuitGo. This simulator is meant to be used as a supplement to the KPP 02 driving circuit training module. The simulator is a 3D version of the Shahbandar driving circuit that includes KPP 02 training exercises like Hill Start, S-Course, Z-Course, Side Parking, and Three-Point Turn. The study is founded on the concept of game-based learning and the utilization of a simulator to enhance the engagement and repetitiveness of the learning experience for students.

The novelty and contribution of this research resides in the creation of a curriculum-specific simulation meticulously designed to replicate the Shahbandar driving circuit, providing an accessible intermediary between low-fidelity quiz applications and expensive professional equipment. This study significantly advances the social sciences by presenting an empirical model illustrating how Game-Based Learning (GBL) can alleviate performance anxiety in high-stakes educational contexts. This research illustrates how digital intervention can mitigate the socio-economic disparity in driver education by providing a replicable "sandbox" for practice, thereby ensuring that proficiency is not constrained by inadequate access to physical training facilities.

Therefore, this study aims to develop and evaluate a game-based driving simulator that supports the practical training requirements of the KPP 02 curriculum. Specifically, the objectives of this study are:

- To identify and integrate suitable Game-Based Learning (GBL) principles for the development of a driving simulation environment that supports skill acquisition in driver education.
- To design and develop CircuitGo, a 3D game-based driving simulator that replicates the key driving modules in the Malaysian KPP 02 training circuit, including Hill Start, S-Course, Z-Course, Side Parking, and Three-Point Turn.
- To evaluate the functionality and usability of the CircuitGo simulator through user testing in order to assess its effectiveness as a supplementary learning tool for improving learner preparedness and confidence in practical driving training.

Methods

To develop CircuitGo, a driving simulation program, a structured, iterative methodology was used, and a process that allowed for an efficient and effective way to achieve project goals. The project will use an adapted version of the Agile development model with a focus on four core iterative phases: requirements analysis, design, development, and testing. The Agile methodology was chosen due to its flexibility, ability to provide continuous feedback, and suitability for a software project like serious games, which can have changing or evolving requirements.

Requirement Analysis Phase

To begin the project, a broad-based inquiry was conducted to create the foundation needed to develop the simulator in accordance with the first project objective. The inquiry consisted of a literature review of existing academic and professional resources in four essential areas: the structure and requirements of the KPP driver education curriculum in Malaysia, the effectiveness of game-based learning (GBL) and applicable pedagogical theories regarding skill acquisition, the ability of modern 3D game engines (specifically, Unity, Unreal and Godot) to meet technical requirements, and an assessment of existing driver education simulation applications within Malaysia. Along with this review of the literature, an analysis of learner drivers' problems in practice, as outlined in the news media and by driver education practitioners, was conducted. The synthesis of this data resulted in the formal statement of the project's problem, objectives, scope, and significance; this phase culminated with the determination of GBL as the centrepiece of the pedagogical approach and Unity as the primary game engine to be used for the development of the driving simulator. The primary reasons for selecting Unity included its combination of realistic physics, the ability to be deployed on multiple platforms, the strong support of assets, and the relative ease of developing educational applications.

Design Phase

The design phase translated the original foundational information into detailed technical plans and architectural plans that fulfilled the second project objective. During this phase, we created a "conceptual" system architecture that defined how the simulator would operate. The system architecture is presented in Figure 1.

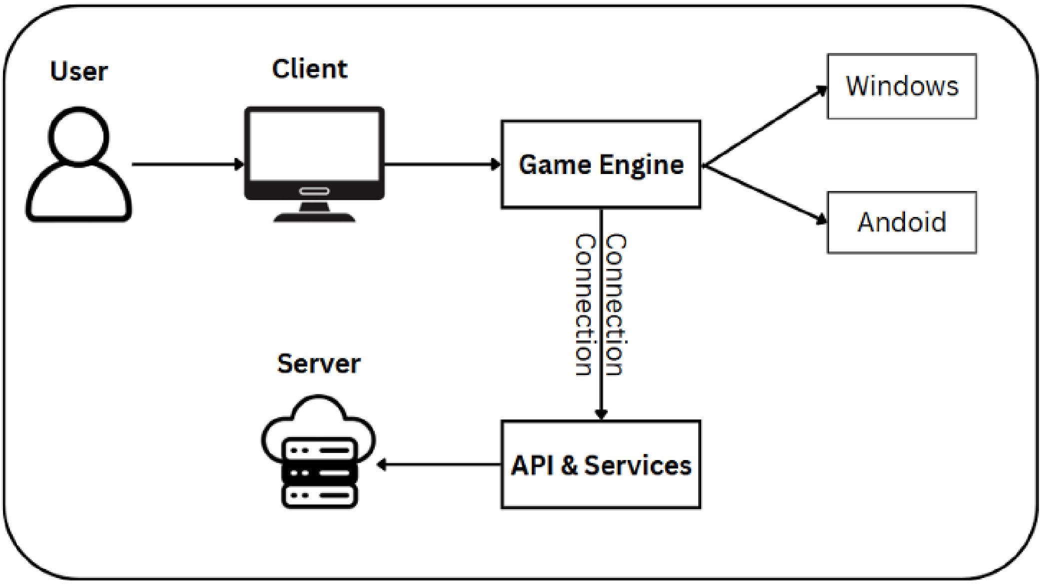

Figure 1 System Architecture

We defined the simulator's architecture as a client-server model, where players connect to the simulator through a web browser. The game's logic will be processed using the Unity3D game engine, while the game's data will be managed through backend services. To model user interactions, a Use Case Diagram was created using the "Player" as the primary actor. The Use Cases identified included: "View Instructions", "Select Driving Module", "Start Driving Simulator", "View Results", and "View Settings". Each Use Case was elaborated in a detailed specification table showing the Use Case's preconditions, postconditions, and normal/alternate flows. A storyboard (Figure 2) of hand-drawn sketches was created to show how the main user interface screens (Main Menu Screen, Instruction List Screen, Training Selection Screen, and Results Screen) are to be laid out and flow together. This allowed us to illustrate how the main User Interface screens will be displayed and to ensure designs would be intuitive and player-friendly before any coding was performed.

Development Phase

The development phase involved the practical construction of the CircuitGo simulator using the specifications defined in the previous stage. The hardware environment consisted of a development laptop (Lenovo Legion 5 with an Intel i5-12500H, 16GB RAM, NVIDIA RTX 3060 GPU) to handle 3D rendering and real-time simulation tasks. The core software stack included Windows 11 as the operating system, Unity 6000.0.43f1 LTS as the game engine, Blender for custom 3D modeling, and assets sourced from the Unity Asset Store. The deployment target was set for WebGL to ensure wide accessibility via modern web browsers without requiring software installation.

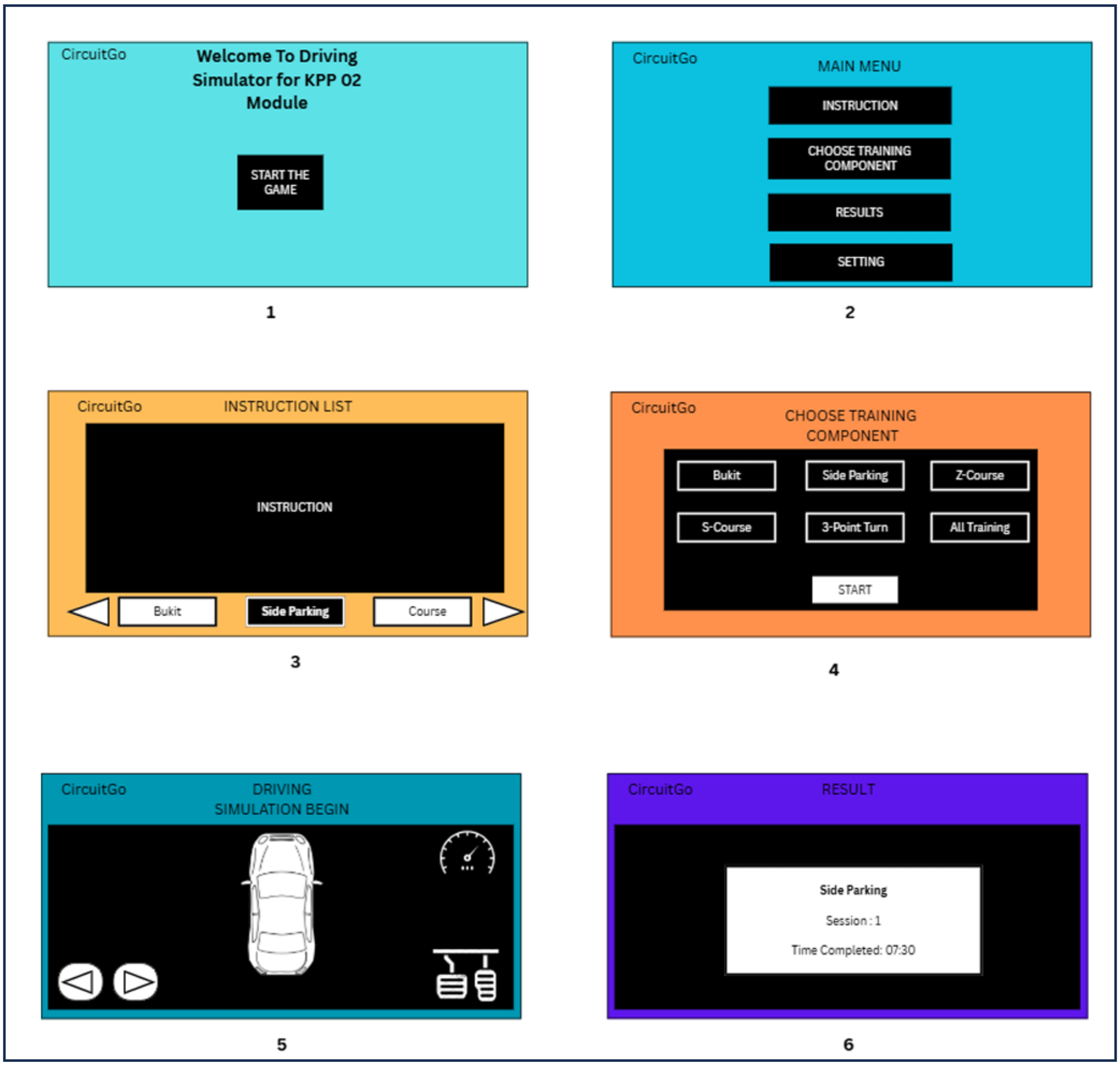

Figure 2 Storyboard

3D development was first to create an environment. The Shahbandar driver training course was then modelled carefully in Blender, showing all detailed physical features along with the corresponding physical dimensions of the Hill, The S-Course, The Z-Course, The Side Park area, and the Three-Point Turn area. An asset package containing a basic vehicle was purchased, imported from the Unity Asset store and added to physics in Unity as well as input systems in Unity using C# scripting to code all gameplay mechanics for vehicle control (acceleration, braking, steering and handbrake) and camera systems (first-person and third-person view) to module specific functionality such as the 5 second timer for Hill Hold functionality and collision detection for course dividers. The key features of GBL were then added as well such as interactive tutorials using video instructions for delivery, glowing checkpoints to assist in locating them, a scoring system for points scored, and immediate feedback to the player (pass or fail). Each KPP 02 was developed as an independent scene with individual associated rule-enforceable scripts.

Testing Phase

During the last phase, CircuitGo's functional and usability testing was the third goal/objective of the project. The testing was divided into two parts: Functional Testing and User Experience Testing. Functional testing includes a structured test case that verifies the execution of each of the core use cases, including user interface (UI) navigation, module selection, launch of the simulator, display of results, and access to settings.

Table 1
Functionality Test

TS-ID	Test Case	Pre-condition	Test Steps	Expected Results	Actual Results	Pass/Fail
TC1	View Instruction	Player is already started the game	1. From Main Menu, click "View Instruction."	Instructions page displays correctly.	Players can view multiple instruction modules of training component	Pass
TC2	Driving Component Selection	Player is already in training selection scene	1. Click "Select Driving Component." 2. Choose an option of driving component.	Selected component is confirmed.	Players can select any of the training module and confirm it with 'start' button	Pass
TC3	Driving Simulator Launch	A driving component is selected.	1. Click "Start Driving Simulator."	Simulator loads with selected component	The scene will load the car, environment, ui and start the countdown to start the module.	Pass
TC4	Results display	The driving simulator session is completed	1. Click "View Results.	Performance metrics are displayed.	The new result will show after player completes the module. It will show whether players pass or fail the course	Pass
TC5	Settings Access	Player is already started the game	1. Click "View Setting."	Settings menu opens with configurable options.	Players can adjust the sound volume, control sensitivity and graphics setting.	Pass

Table 1 summarizes the test cases, steps, and results, confirming that all intended features operated correctly for instance, the Hill module correctly enforced the stationary hold, and the S-Course correctly failed the player upon divider collision.

After conducting usability testing, we ran this through the functional validation phase with 10 participants (novice and experienced drivers) who had either not seen the KPP 02 syllabus or had little knowledge about it. Each participant had to interact with the simulator and complete a brief 10-question Likert scale survey covering: Controls' responsiveness, realism, tutorial effectiveness, appropriateness of difficulty level, use of feedback, and interface clarity.

Table 2 provides an overview of the collective feedback received from the participants after completing the usability tests. The feedback was overwhelmingly positive for controls, engagement, and interface design, with constructive comments providing input/ideas to improve and fill gaps for future iterations: Additional more advanced instructions, inconsistencies with certain aspects of vehicle physics, etc.

Table 2
Usability test feedback

	Player Testing ID:					
Num	Questionnaire	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Driving controls are responsive.	50%	50%	-	-	-
2	The simulation accurately replicated real-world driving conditions	50%	30%	20%	-	-
3	The tutorial effectively taught me the core driving skills needed for the KPP-02 module.	50%	40%	-	-	-
4	I needed additional instructions to understand advanced maneuvers	-	-	40%	60%	-
5	I noticed inconsistencies in vehicle behavior or physics	-	-	20%	60%	20%
6	The difficulty level of scenarios matched real-world driving challenges.	10%	80%	10%	-	-
7	The feedback system helped me improve my driving.	30%	70%	-	-	-
8	I felt confident applying the skills learned in this simulator to real-world driving.	20%	60%	20%	-	-
9	The simulation kept me engaged and motivated to complete all training modules.	50%	50%	-	-	-
10	The interface was clear and easy to read during gameplay.	70%	30%	-	-	-

Results and Discussions

Implementation Overview

The project was originally based on the design blueprints of the simulator, which converted it into the final usable product. The development environment, using Unity 6000.0.43f1, worked well for developing the game's real-time 3D graphics and complex vehicle physics (through use of the NVIDIA PhysX engine) and the event-driven C# scripting logic for the game. The final design architecture of the system functioned according to its design: the core game was appointed as a WebGL build and maintained by a master Game Manager object (as part of the Unity Engine functionality), which was responsible for managing transitions in between game scenes, maintaining and enforcing rules, keeping track of the score, etc. The User Interface, built using both the Canvas and TextMeshPro User Interfaces of Unity, was developed separately of the Game Logic, allowing independent updates and maintenance. The result was the creation of a reliable and responsive application capable of being executed directly from a web page.

Core Game Development

The Shahbandar circuit was faithfully recreated within the 3D environment as illustrated in Figure 3, where custom course components were inserted into a dynamic terrain with generic urban assets for visual depth and spatial context. The vehicle handling model supports basic realistic physics for acceleration, braking, and turning. A significant feature is the dual camera system, allowing the player to toggle between first-person view (complete with real-time virtual mirrors) and third-person view (see Figure 4) to meet the different preferences of players or improve situational awareness when executing more complicated manoeuvres.

Figure 3 Shahbandar Training Circuit Environment

Figure 4 First-Person View vs Third-Person View

The simulator encompasses all five core KPP 02 modules, each with unique mechanics:

Figure 5 Hills Course Scene

Hills Course: Requires the player to stop on an incline and hold the vehicle stationary for five seconds before proceeding, simulating a hill-start (Figure 5).

Figure 6 S-Course and Z-Course Layout

S-Course & Z-Course: Navigate tight, winding paths delineated by dividers. Contact with any divider or reversing triggers an immediate failure, emphasizing steering precision and path-following (Figure 6).

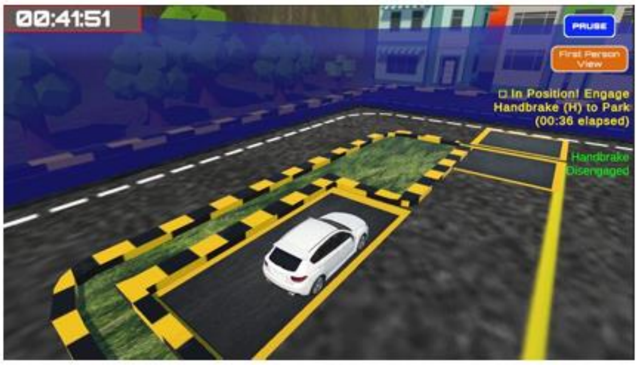

Figure 7 Side Parking Module

Side Parking Module: Players must align the vehicle within a designated bay and activate the handbrake (assigned to the 'H' key) to successfully complete the task within a five-minute time limit (Figure 7).

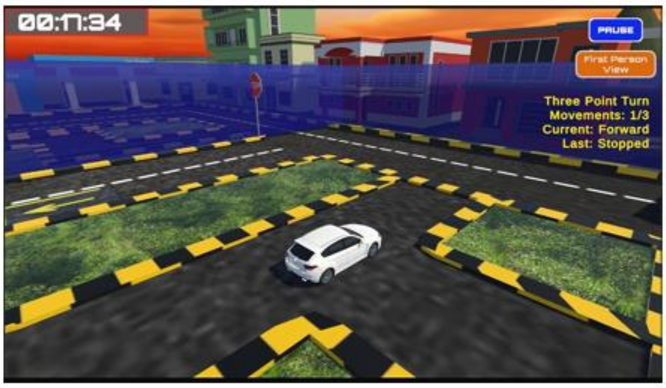

Figure 8 Three-Point Turn Area

Three-Point Turn: Conducted in a confined rectangular area, requiring the maneuverer to complete the maneuver in exactly three movements without touching the boundaries (Figure 8).

Vol. 15, No. 1, 2026, E-ISSN: 2226-6348 © 2026

Figure 9 Glow Checkpoint at an Intersection

All Training (Integrated Mode): A comprehensive scene that sequentially combines all modules. It also introduces a dynamic traffic light system at an intersection, which uses sensors to detect vehicles and change signals, adding a layer of real-world traffic interaction (Figure 9).

Player Interface Design

The UI/UX flow was designed for intuitive navigation. Upon starting, players are greeted by a Welcome Scene leading to a clean Main Menu (Figure 10). From here, they can access the Instruction List (containing video tutorials), the Training List to select a specific module (Figure 11), view their performance History in the Results screen (Figure 12), or adjust Settings. The interface maintains consistency and clarity, using unambiguous icons and text to guide the player without overwhelming them, thereby supporting the self-paced learning objective.

Figure 10 Main Menu Scene

Figure 11 Instruction List Scene

Figure 12 Results Scene

Testing and Validation Outcomes

The functional test results, detailed in **Table 1**, confirmed that every module's rule set was correctly enforced by the underlying game logic. The pass/fail feedback panels (**Figure 13**) were triggered appropriately based on player performance, providing immediate and clear outcomes.

Figure 13 Passed and Failed Panels

The usability test feedback on Table 2 revealed a strong positive reception. Most participants (80-100%) agreed or strongly agreed that the driving controls were responsive, the interface was clear, the tutorial was effective for core skills, and the simulation was engaging. The feedback system was also rated highly for helping improve driving. These results validate the project's success in creating an accessible and motivating learning tool. However, the feedback also highlighted areas for growth. Approximately 60% of participants felt they needed additional instructions for advanced maneuvers, and 60% were neutral or noticed inconsistencies in vehicle physics, suggesting that enhancing the tutorial depth and refining the physical realism should be priorities for future development

Discussion of Development Challenges

This process of development was not incident-free, and that too has given many learning experiences. It was necessary to fine-tune the implementation of correct and accomplished sensor-based tracking to implement module rules. An example of that is the sensitivity/forgiveness balance of the collision detector of the S/Z-Course dividers, required to balance between sensitivity (to pick up errors) and forgiveness (to pick up precise near-misses). Equally, the programming behind the hill-hold timer and the activation of the handbrake in the parking module had to have conditional checks that were specific to activate only at the right vehicle states. The process of debugging was continuous, especially when dealing with interconnected systems, as was the case with the traffic light sensors in the integrated mode, where it was important to make sure of the order of script execution and the trigger conditions. These issues highlighted the difficulty of changing real-life test specifications of exams into strong digital logic and the significance of trial testing in simulation creation.

Conclusion

The research project met all three primary objectives. First, to identify a suitable game-based learning approach for driving simulation to improve Malaysian learner drivers in mastering the KPP 02 circuit module, GBL has been identified as the most appropriate pedagogical approach to use with a KPP 02 driving simulator because GBL has been shown to provide motivation, engagement, and safe skill acquisition or skill practice. The second objective, to develop a driving game simulator for the KPP 02 module using game-based learning, has been achieved; the simulator is a true representation of the Shahbandar training circuit and all of the critical training modules associated with teaching the KPP 02 within the simulator, and includes the features of GBL and has provided immediate feedback during use. Finally, to evaluate the functionalities and usability of the CircuitGo game. The simulator was evaluated for operational stability, educational usefulness, and user satisfaction through a structured functional and usability testing program.

The CircuitGo program has made a significant impact on the modernization of driver education in Malaysia. The program eliminates three major problems with traditional methods of educating people (i.e., learner anxiety, limited amount of practice time, and instrumenting the learning of driving skills on a driving circuit) through a digital training aid that is safe, cost-effective, and easy to access. The simulator is useful to two types of learners: first-time learner drivers, who can practice without fear of hurting someone or damaging equipment; and driving instructors, who can use this digital tool to demonstrate the correct way to perform a specific skill and monitor how all their students are progressing as they learn. By bridging the gap between the theoretical components of knowledge contained within KPP 01 and the practical applications contained within KPP 02, it is anticipated that there will be a greater number of individuals who will pass their driving tests on their first attempt and become more confident/capable new drivers.

Recommendations

To build upon the foundation established by this project, several opinions for future enhancement are recommended:

Platform Expansion: Optimize and port CircuitGo to Android and iOS platforms to dramatically increase accessibility, allowing learners to practice on ubiquitous smartphones and tablets.

Immersive Technology Integration: Incorporate Virtual Reality (VR) support to provide an unparalleled level of immersion and spatial awareness, closely mimicking the actual driver's field of view and head movements.

Enhanced Simulation Realism: Introduce dynamic and unpredictable elements to move beyond static circuit training. This could include AI-controlled traffic vehicles, pedestrian crossings, variable weather conditions (rain, night driving), and randomized minor road events to train hazard perception and adaptive driving skills.

Advanced Analytics and Personalization: Develop a more sophisticated backend analytics dashboard for instructors, offering detailed breakdowns of learner performance across attempts. Furthermore, implement an adaptive difficulty system that adjusts scenario complexity based on the learner's proficiency

Acknowledgment
The authors would like to thank the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, for all the given support.

References

Abdulrahaman, M. D., Faruk, N., Oloyede, A. A., Surajudeen-Bakinde, N. T., Olawoyin, L. A., Mejabi, O. V., Imam-Fulani, Y. O., Fahm, A. O., & Azeez, A. L. (2020). Multimedia tools in the teaching and learning processes: A systematic review. *Heliyon*, 6(11). https://doi.org/10.1016/j.heliyon.2020.e05312

Bruck, L., Haycock, B., & Emadi, A. (2021). A Review of Driving Simulation Technology and Applications. *IEEE Open Journal of Vehicular Technology*, 2, 1–16. https://doi.org/10.1109/OJVT.2020.3036582

Cavalcanti, A. P., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y. S., Gašević, D., & Mello, R. F. (2021). Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*, 2. https://doi.org/10.1016/j.caeai.2021.100027

Choo, Y. B., Yoke, S. K., Teng, L. A., & Ismail, U. N. (2022). A Systematic Literature Review on the Use of Digital Media among Malaysian Teachers and Students During the Covid-19 Pandemic. *International Journal of Academic Research in Progressive Education and Development*, 11(2). https://doi.org/10.6007/ijarped/v11-i2/14130

Jabatan Pengangkutan Jalan Malaysia. (2021). *Kurikulum Pendidikan Pemandu*.

Mustapha, A., Saad, N., Mustapha, M., Raji Egigogo, A., & Bala Umar, M. (2024). Enhancing Road Safety Education through AI-Driven Tools: The Impact of Driving Simulations for Accident Prevention.

Othman, Z., Zain, N. H. M., Ismail, I., Affandi, S. N. S. S. M., Noh, N. A. M., & Yasin, A. M. (2020). An educational game on the theories of driver education curriculum: An evaluation. *International Journal of Evaluation and Research in Education*, 9(4), 1088–1095. https://doi.org/10.11591/ijere.v9i4.20659

Sanjay. (2021, May 14). Outdated and woefully incomplete, Malaysia's driving curriculum needs a major update. *WapCar*. https://www.wapcar.my/news/outdated-and-woefully-incomplete-malaysias-driving-curriculum-needs-a-major-update-26739

Lee, S.Y., (2024, July 25). Study: M'sia 2nd Most Stressful Country For Those Learning How To Drive. *WeirdKaya*. https://weirdkaya.com/study-msia-2nd-most-stressful-country-for-those-learning-how-to-drive/