Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 512-531

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Development of Fully Immersive Virtual Reality Job Simulator

Muhammad Asyraaf Syahmi Rosmawee, Mohd Norasri Ismail[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2023.04.02.029
Received 17 July 2023; Accepted 08 November 2023; Available online 30 November 2023

Abstract: Virtual Reality (VR) is a technology that is computer-generated to enable interactions between users and the virtual environment. The cost of development and exclusivity of VR equipment for VR training has led to less VR job simulators games in the market. The game Inside: VR Job Simulator is designed and develop using the simulation based Virtual Reality approach to produce a fully immersive job simulation game. The users targeted for playing this game will be teenagers from age 13 to 18 years old. The game development life cycle (GDLC) is used to develop this project. The functional testing is conducted, and the usability testing has resulted a score of 82.63 which is within the acceptable range based on the System Usability Scale (SUS). Therefore, the game could be able to assist the target users in trying to familiarize themselves and learn about the designated work environment and their tasks.

Keywords: Virtual Reality, job simulation, simulation based Virtual Reality

1. Introduction

Virtual reality (VR) is a technology that is computer generated to enable interactions between virtual environments and the users [1]. VR systems could be classified into 3 things which are non-immersive, semi-immersive and immersive which is the type of components, or the interfaces utilized in the system [2]. This system helps to create an immersive experience for the user as it gives immediate feedback based on user interaction with VR devices [13]. Simulator which is a piece of equipment that is designed to represent real conditions is used to provide a real-like experience to the user [3]. The helps of Virtual Reality technology have helped to evolve the simulator technology to become more advanced as it helps us to experience all the interactions in the simulator-based game.

Simulation methods have led to the development of a job simulation game as it could help the user to immerse himself and experience the job firsthand. Feats such as VR training has appeared from the job simulation concepts and it could reduce the risks for the training of new people in the industry such

*Corresponding author: norasri@uthm.edu.my

as the doctor training [4]. Therefore, the combination of these simulation and virtual technologies has led to many VR simulation games that could help to train people before their site experience [4].

However, due to ongoing research and development, many job simulators in the market provides limited interactivity in its game design [5]. The limited interaction of game and sandbox nature of some VR games could be seen as not helping the user to fully learn on the job tasks and equipment [5]. Thus, the project is proposed to provide a more interactive and facts-based game that could provide information and knowledge to the target user.

The objectives of this project are to design Inside: VR Job Simulator based on Virtual Reality approach, to develop a fully immersive job simulation game by implementing simulation based virtual reality technology and to perform functional testing and user acceptance test on the developed application to the target user. The target users of this game are teenagers ranging from the age 13 to 18 years old. The subject matter expert (SME) will be the doctor and game developer that work in the industry. The language that will be used is the English language.

The game will also be complemented with basic storyline narrative. The job simulation in the form of interaction with environment and equipment could be seen from the implemented task and objectives that user should clear to progress the story and the game. The game will use Game Development Life Cycle as its development methodology. Users of the game will only be able to interact with the objects in the limited area of the gameplay. Therefore, the game should provide users with the ability to interact with the displayed 3D model when user hover through the game. Next, all menus and buttons are to function properly during the user gameplay. The game camera should also follow the user's head movements.

The rest of the paper is as follows: Section 2 covers the domain of the study, the technology used, and the result of the comparative analysis. Section 3 describes the Game Development Life Cycle (GDLC) methodology that is chosen to apply in this project, as well as the output of the analysis and design phases of this project. Furthermore, Section 4 shows the results and discussions, and Section 5 states the conclusion of the project.

2. Related Work

In this section, the study domain, technology used, and the comparative analysis results are discussed.

2.1 Domain Background

Job simulation is one of the components in the simulation game genres. Job simulation could be defined as completing tasks such as professional tasks in the industry but in a non-employment context [6]. Therefore, job simulation is used to stimulate a work environment for the end user as they could experience the job firsthand without having to present physically in the designated place to complete the task. In the present days, many job simulations are done physically due to reasons such as the cost needed for a company to provide job simulation which is also known as VR training. Headset cost, XRS cost and many more contribute to the current cost problem in VR training [7]. These led to many developers begin to develop many VR simulations that could be accessed easily by user. However, due to limited budgets and not fully catering to the professional setting of the jobs, many VR games in the current market are less professional if compared to the proper traditional training of a job in the industry [6].

2.2 Related Approach

Two approaches are used in developing the project. Firstly, an immersive system, also known as fully immersive technology, is used in this project. These technologies require the user to use head mounted display (HMD) which enables user head's movements tracking and data glove is the main equipment for this system [9] . This system has allowed users to fully immerse themselves into the 3D environment as they could interact with the 3D objects. However, many companies need to spend money to make a proper program to train their employees [6]. Therefore, these problems have improved over the years with the more accessible options developed such as the Meta Quest 2 equipment. Thus, more fully

immersive games become more accessible to the user. Games such as Job Simulator could be seen as it uses the fully immersive system approach in their gameplay. The use of physics in the object such as plates breaking, grabbable object and many more contributes to the fully immersive experience to be realized [10].

Next, the simulation based Virtual Reality approach is also used in this project. Simulation tasks and exercises seen in doctor training could help to improve doctor skills [4]. This is because this approach could help to reduce the gap between theoretical knowledge and practical skills of the doctors. For example, the training facilities that provide only VR equipment to the doctors and pilots could help to reduce this gap. However, problems such as inconsistency between game and real-life industry work could affect the students as they could practice wrong information if the VR training does not provide an accurate information. Thus, a technically accurate simulation should be applied to the design as it will help the students to develop and hone their skill [4]. Games such as Surgeon Simulator use this approach in its game design. However, due to its sandbox nature of the game and inaccurate information could raise the problems such as students practicing incorrect information and skills [11].

2.3 Comparative Analysis

In this section, a comparison has been made between the existing applications such as Job Simulator [10], Surgeon Simulator VR: Meet The Medic [11], Electrical Safety VR Training [12] and the proposed application. These games will be compared based on 8 features such as supported platform, language, price, play area, input, jobs included, strengths and limitations. Figure 1(a), Figure 1(b), and Figure 1(c) show the interface of three existing games.

Figure 1(a): Job Simulator

Figure 1(b): Surgeon Simulator VR: Meet The Medic

Figure1(c): Electrical Safety VR Training

Table 1: Comparison between existing applications and proposed applications

Features	Job Simulator	Surgeon Simulator VR: Meet The Medic	Electrical Safety VR Training	Inside: VR Job Simulator
Supported platform	HTC Vive, PlayStation VR, Oculus Quest, Valve Index, Windows Mixed Reality	Valve Index, HTC Vive	Valve Index, HTC Vive, Oculus Rift	Oculus Quest 2
Language	English for the interface, full audio, and subtitles. French, German, Japanese, Korean, and Spanish for subtitles	English	English, Russian	English
Play Area	Room-Scale	Standing, Room-Scale	Standing, Room-Scale	Standing, Room-Scale
Input	Tracked Motion Controllers	Tracked Motion Controllers	Tracked Motion Controllers	Tracked Motion Controllers
Jobs Included	Gourmet Chef, Office Worker, Convenience Store Clerk, Automotive Mechanic	Surgeon	Electrical Engineering	Game Developer, Doctor

Features	Job Simulator	Surgeon Simulator VR: Meet The Medic	Electrical Safety VR Training	Inside: VR Job Simulator
Gameplay	Step by step task based on jobs with combination of realistic and non-realistic task.	Free to roam in the medical room and interact with equipment.	Step by step task based on instructions given.	Step by step task based on instruction given complement with some storyline.
Strengths	- Provides multiple language for player - Provides multiple jobs that player could experience - Allows user to interact freely with object	- Free to download - Player could use any equipment for surgery	- Free to download - Offers realistic and technically accurate facts - Hazard detection, hazard alert and informing contact of hazard to player	- Free to download - Provides multiple jobs that player could experience - Provides storyline and game lore - Offers realistic and technically accurate facts
Limitations	- The sandbox nature of the game making the game less suitable for expert - Does not provide proper facts and information	- Does not provide proper work simulation of a surgeon - Does not provide proper facts and information	- No game lore or storyline - Only focuses on VR training	- Does not include complex task due to incorporating storyline with VR job simulation - There is no sandbox or practice module for the user to play after finishing the game

Based on Table 2.6, the supported platform for these 4 different games supports different platforms. Job Simulator supports most of the VR headsets while the other games only support few of VR headsets. English is the language used by these 4 games, but Job Simulator provides 6 different languages for subtitle and Electrical Safety VR Training also supports Russian language. Majority of these games play area covers standing and room-scale except Job Simulator that covers only Room-Scale. All games use Tracked Motion Controllers as the input for the user while playing. Jobs included in these games are gourmet chef, office worker, convenience store clerk and automotive mechanic for the Job Simulator, surgeon for Surgeon Simulator VR: Meet The Medic, electrical engineering for Electrical Safety VR Training. The proposed game which is Inside: VR Job Simulator will include jobs such as game developer and doctor as the jobs for the gameplay.

3. Methodology/Framework

The proposed game, Inside: VR Job Simulator is a game that is designed to give work experience to the user. Thus, the Game Development Life Cycle (GDLC) is used to develop the proposed game. Due to game nature of not purely using systematic process of engineering to produce a creative project containing art, music and many more, common software development life cycle (SDLC) is not suitable to be used [8]. Therefore, the 6 phases of GDLC could be seen from Figure 2 is use in this project

development due to its flexibility provided to the developer to make changes while creating the game. The production cycle are repeated 2 times to produce a better-quality product.

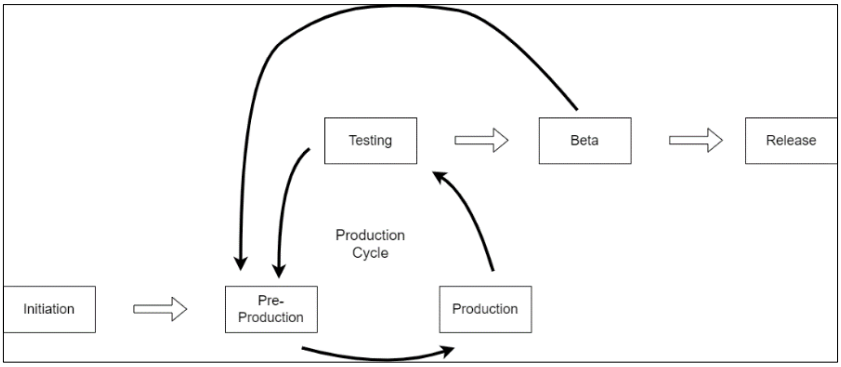

Figure 2: GDLC methodology

3.1 Initiation

This phase will focus on the idea and concepts of the game intended to be created. The proposal that contains game title, project background, problem statement, objective, scope, expected outcome, project significance and project planning is created in the proposal. The information is gathered by interviewing two of the subject matter experts (SME) of this game, which is the doctor and game developer. A physical and online interview is conducted with Dr. Nik Rosenorleyaney bte Nik Ibrahim who work at Poliklinik Pakar Marissa, Bandar Baru Bangi, Selangor, and Mr. Jayagaren A/L Paramasivam who work as the Chief Technology Officer at Teczo Sdn Bhd, Taman Teknologi Malaysia, Bukit Jalil, Kuala Lumpur respectively. The user analysis, functional and non-functional requirements are also included as shown from Table 2, Table 3, and Table 4.

Table 2: User Analysis

Stakeholder Category	Role in Product	Design Implications	Actions Needed
Subject Matter Expert	Content consultant expert in the doctor job in the industry	Easy to navigate	• Create simple and understandable navigational structure. • Easy and clear to understand instruction must be added.
		Content must be based on medical procedure	Refer to the latest medical procedure as a guideline for the game content.
		Realistic VR Interaction	• Implement physics to the game objects. • Model game prop and assets based on the real equipment.
		Easy to follow game instructions	Make a simple and clear step by step instruction in designing the game level.
		Easy to use	• Avoid using unnecessary menu and icon to reduce confusion. • Implement an easy to understands control for first time VR users.

Stakeholder Category	Role in Product	Design Implications	Actions Needed
Subject Matter Expert	Content consultant expert in game developer job in the industry	Content must be based on game development methodology	Refer to the latest development methodology used by the recent projects.
		Low latency, good graphics, and high fps	• Read forum, website, and watch tutorial for the settings configuration. • Edit player settings in Unity until it gives the low latency, good graphics, and high fps.
		Realistic VR Interaction	• Implement physics to the game objects. • Model game prop and assets based on the real equipment.
		Provide option in character movement	• Enable movement option such as continuous move, snap turn, and teleportation. • Allow the function for the user to choose which type of movement they preferred.
		Easy to follow game instructions	• Make a simple and clear step by step instruction in designing the game level.

Table 3: Functional Requirement

Functional Requirement	Description
Autonomous System Activity	• After the users open the game, the interface and prop in the main menu should be displayed automatically. • The object state can be change automatically such as object can be pressed when user interact with it. • The animation of the story should be played automatically when user press the play button for the new game playthrough. • Floating menu will appear for the pause menu when user touch the controller button. • The environments asset, camera, light and game object asset should be generated automatically for each job. • Every task will be checked out after player completed the task while playing the game. • Equipment physics can be shown when user interact with the equipment while playing the game. • Text and noninteractive object can be seen when playing the game to indicate the current action is not the supposed action for the current task.
User interaction	• Users should be able to move their hands based on user hand movement. • The game should allow the user to interact with the assigned equipment to complete the task. • The users should be able to understand the game controls while following the game instructions.

Table 4: Non-functional requirements

Non-functional Requirement	Description
Performance	• The game should be able to respond to every process within 2 second for all users. • The game should be able to load its interface and object for every level within 10 seconds.
Implementation	• The game should be able to operate on Meta Quest 2 device if it uses the version 47 and above.
Usability	• Users should be able to learn and understand the game as it uses simple interface design. • Users should be able to remember the function of each controller button and menu after playing the game for some time.
Legal	• Users should not be able to modify the content of the game.
Cultural	• English language use for menu, buttons and many more.

3.2 Pre-Production

In the pre-production phase, literature review of the current apps is reviewed. Then, the design of storyboard, gameplay, game lore, user interface design and user experience design are designed in this phase. These elements are also included in the game design document which is one of the guidelines used to produce a game. The first iteration of this phase will focus on its basic foundations and structure and to be improved in the next iteration. The game flowchart shown in Figure 3 and navigation structure shown in Figure 4 are designed in this phase. The elements of game design documents such as gameplay design, game level design and the button design will be shown in Figure 5, Figure 6 and Table 5 respectively.

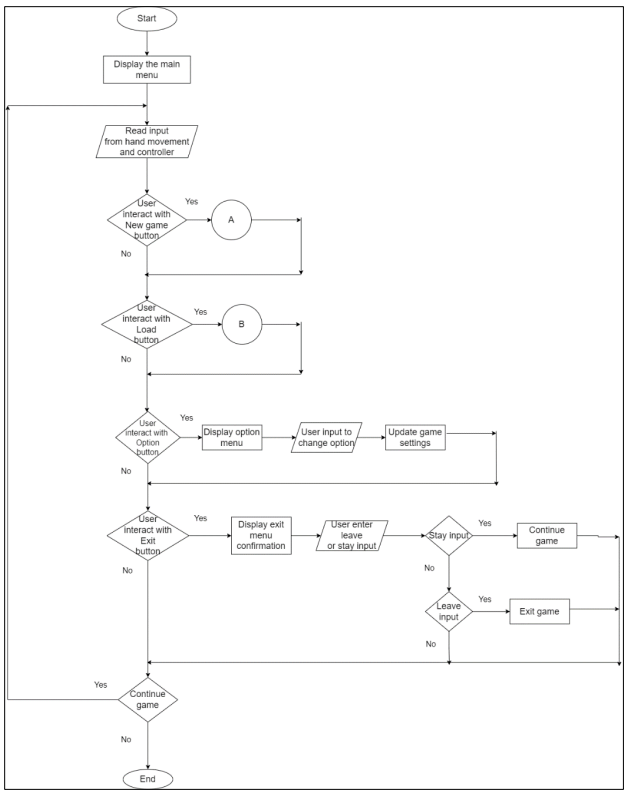

Figure 3(a): Flowchart

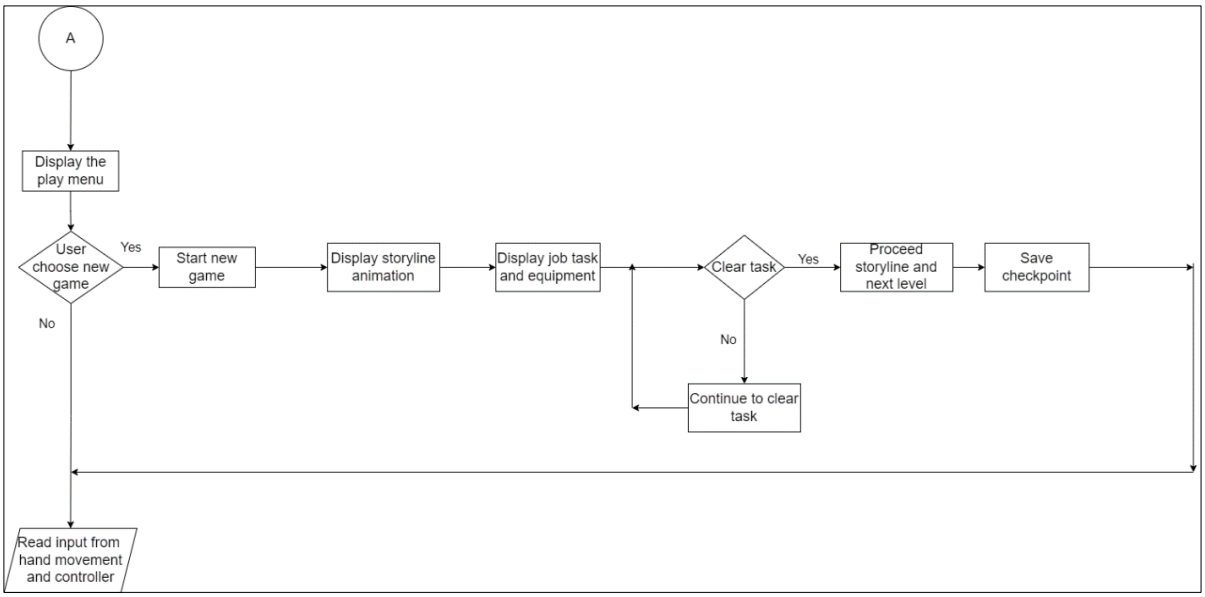

Figure 3(b): Connector A of Flowchart

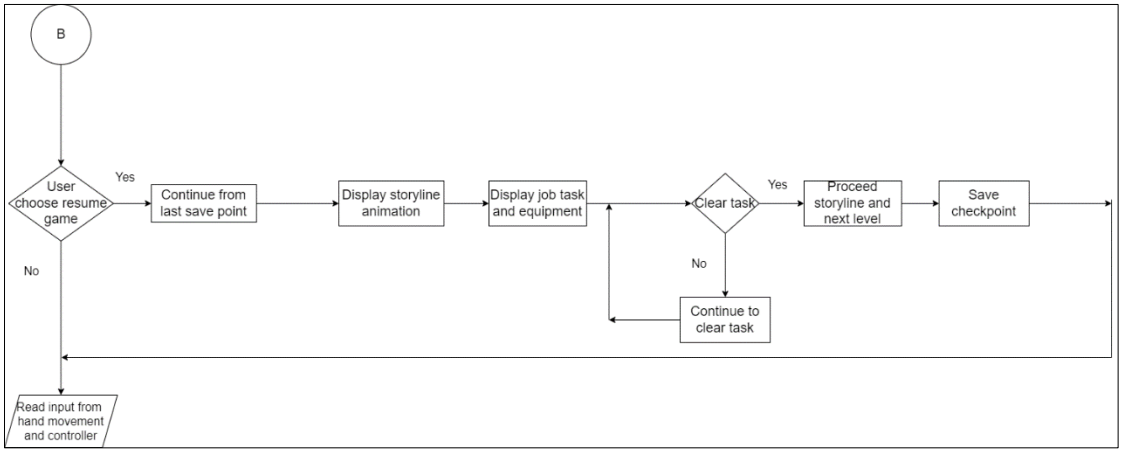

Figure 3(c): Connector B of the Flowchart

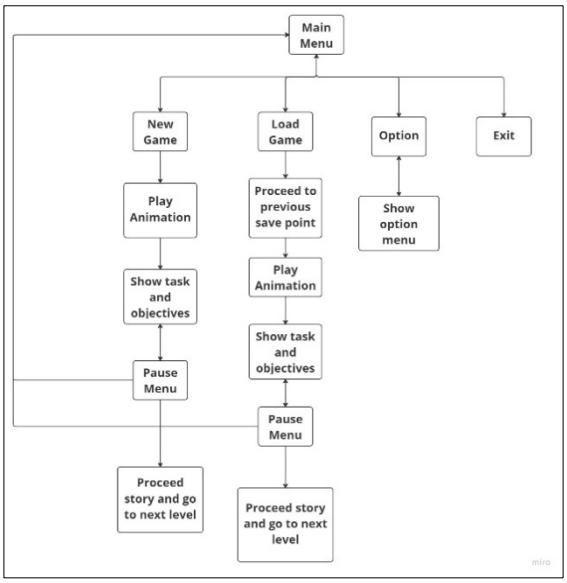

Figure 4: Navigation Structure

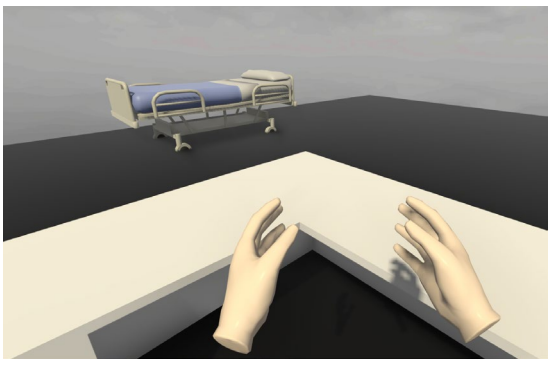

Figure 5: Gameplay Style

From Figure 5, the gameplay style of the game could be seen as a VR game that uses the First-Person gameplay style. This type of game uses the in-game character eye view for the player to see the surroundings of the game world.

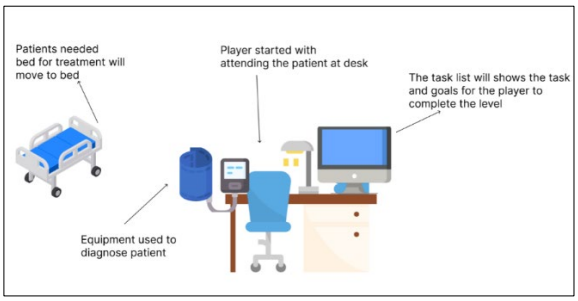

Figure 6(a): Gameplay Design for Doctor

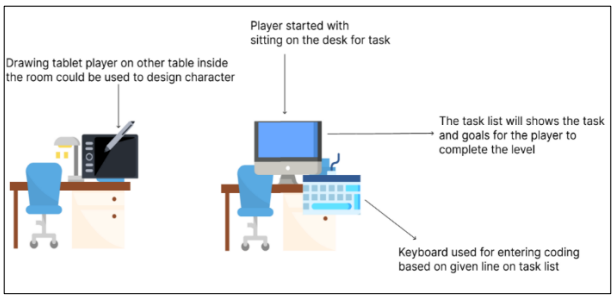

Figure 6(b): Gameplay Design for Game Developer

Table 5: Button Design

Button	Function Description	Button	Function Description
New Game	• This button will navigate user to the gameplay menu.	Exit	• This button helps the user to exit the game.
Load Game	• This button will continue the user to their last save checkpoint in the game.	Continue	• This button allows the user to close the menu and continue the game.
Option	• This button will show the option that user could change to their preferences.	Main Menu	• This button helps the user to return to the main menu interface.

3.3 Production

In the production phase, the design of game characters, models and many more are implemented into the game development software. The models are designed with 3D modeling software such as Blender. Source code is then added to make the game interact with the user. The process of balancing the game is one of the main focusses as this step helps to improve the overall performance of the game by adding new features, fixing bugs and many more.

3.3.1 3D modelling of props and equipment

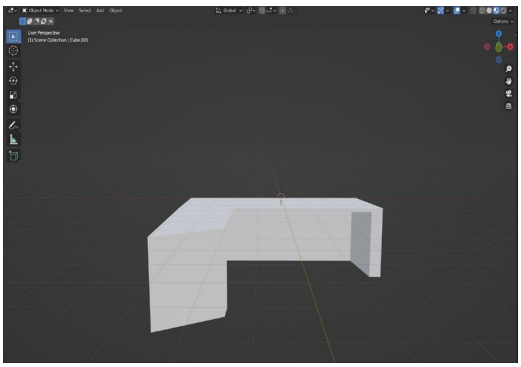

Figure 7(a): 3D Modelling of Desks Props

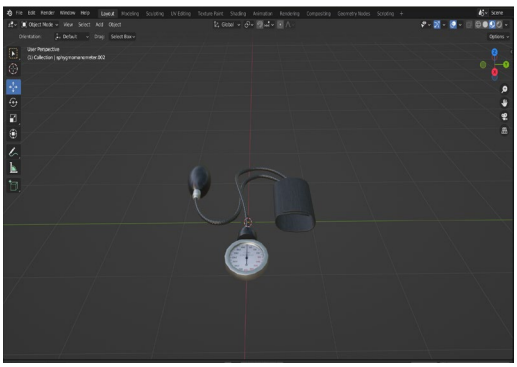

Figure 7(b): 3D Modelling of Sphygmomanometer

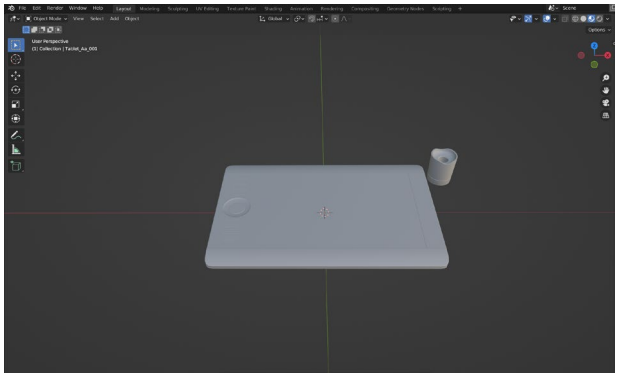

Figure 7(c): 3D Modelling of Drawing Tablet

From Figure 7(a), 7(b) and 7(c), the 3D model of the props and equipment were designed using the blender software. These figures show the examples of the basic props used by both jobs and specific equipment used by the specific jobs' tasks.

3.3.2 Implementing the 3D modelling into Unity

Figure 8: Main Menu Interface Design

Figure 9: Doctor Job Workplace Design

Figure 10: Game Developer Workplace Design

Interface design, prop design and button design are then implemented into the Unity software as seen from Figure 8, Figure 9, and Figure 10. The design of the main menu, doctor job workplace design and game developer workplace design are shown in Figure 8, Figure 9 and Figure 10 respectively.

3.3.3 Implementation of source code

The implementation of source code using the C# language was added for interaction of player with the prop and equipment in the virtual environment. Source code such as XR Origin, XR direct interaction and many more are added to make an immersive VR experience. Table 6 shows the C# script and its function in Unity.

Table 6: C# script implementation

Functions	C# Scripts	Description

XR Origin	namespace Unity.XR.CoreUtils
	{
	
	[AddComponentMenu("XR/XR Origin")]
	[DisallowMultipleComponent]
	public class XROrigin : MonoBehaviour
	{
	[SerializeField]
	[Tooltip("The Camera to associate with the XR device.")]
	Camera m_Camera;
	cref="UnityEngine.InputSystem.XR.TrackedPoseDriver"/> component to the <see cref="Camera"/>
	public Camera Camera
	{
	get => m_Camera;
	set => m_Camera = value; }
		XROrigin script enables to detect initial position of the player in the form of XR Origin Game Object, camera floor offset and tracking origin mode that detects floor layer for locomotion system. XROrigin also allows the addition of another XR function and script to be declared and used in the Virtual Environment.

Functions	C# Scripts	Description

XR Controller (Action based controller)	public partial class ActionBasedController : XRBaseController
	{
	[SerializeField]
	InputActionProperty m_PositionAction;
	
	public InputActionProperty positionAction
	{
	get => m_PositionAction;
	set => SetInputActionProperty(ref m_PositionAction, value);
	}
	
	[SerializeField]
	InputActionProperty m_RotationAction;
	/// <summary>
	/// The Input System action to use for Rotation Tracking for this GameObject. Must be a <see cref="QuaternionControl"/> Control.
	/// </summary>
	public InputActionProperty rotationAction
	{
	get => m_RotationAction;
	set => SetInputActionProperty(ref m_RotationAction, value);
	}
	
	[SerializeField]
	InputActionProperty m_TrackingStateAction;
	/// <summary>
	/// The Input System action to get the Tracking State when updating this GameObject position and rotation;
	/// falls back to the tracked device's tracking state that drives the position or rotation action when not set.
	
	public InputActionProperty trackingStateAction
	{
		XR Controller script that enables InputActionProperty to be used in the game. Thus, enabling the ability to assign the defined input action with the button and controller action. This script is also has been predefined when downloading the XR Interaction Toolkit.

Functions	C# Scripts	Description

	get => m_TrackingStateAction;
	set => SetInputActionProperty(ref m_TrackingStateAction, value);
	}
	}	

XR Direct Interaction (Grab Object)	protected override void Awake()
	{
	base.Awake();
	m_TriggerContactMonitor.interactionManager = interactionManager;
	m_UpdateCollidersAfterTriggerStay = UpdateCollidersAfterOnTriggerStay();
	
	ValidateTriggerCollider();
	}
	
	protected override void OnEnable()
	{
	base.OnEnable();
		XR Direct Interaction is also one of the predefined scripts that could be used. The collider added in the hand model and the collider added in the object helped this script to trigger the OnTriggerEnter function. Therefore, making the ability to grab object and direct interaction to take place.

Functions	C# Scripts	Description

XR Direct Interaction (Grab Object)	m_TriggerContactMonitor.contactAdded += OnContactAdded;
	m_TriggerContactMonitor.contactRemoved += OnContactRemoved;
	ResetCollidersAndValidTargets();
	
	StartCoroutine(m_UpdateCollidersAfterTriggerStay);
	}
	
	protected override void OnDisable()
	{
	base.OnDisable();
	m_TriggerContactMonitor.contactAdded -= OnContactAdded;
	m_TriggerContactMonitor.contactRemoved -= OnContactRemoved;
	ResetCollidersAndValidTargets();
	
	StopCoroutine(m_UpdateCollidersAfterTriggerStay);
	}
	protected void OnTriggerEnter(Collider other)
	{
	m_TriggerContactMonitor.AddCollider(other);
	}	

Script/Source code answer checking	using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.UI; using TMPro; using Microsoft.MixedReality.Toolkit.Experimental.UI; public class TypeKeyboard : MonoBehaviour { private TMP_InputField inputField;	The script is used to make a specific string answer that the user needed to input using the in-game keyboard. The input answer is compared with the assigned value of string. If a specific answer is inputted, the other game object will become

Functions	C# Scripts	Description

	private string Answer = "anim.play(\"DoorMove\",0,0.0f);";
	
	public GameObject img1;
	public GameObject input1;
	public GameObject playBtn1;
	public GameObject text1;
	
	public GameObject img2;
	public GameObject input2;
	public GameObject playBtn2;
	public GameObject text2;
	
	void Start()
	{
	inputField = GetComponent<TMP_InputField>();
	}
	
	public void Word(string ansText)
	{
	inputField.text += ansText;
	}
		active to signal the success of the input answer. This script is used in the coding task in the game developer jobs.

Functions	C# Scripts	Description

Script/Source code answer checking	public void Enter()
	{
	if (inputField.text == Answer)
	{
	inputField.text = "Correct";
	
	img1.SetActive(false);
	input1.SetActive(false);
	playBtn1.SetActive(false);
	text1.SetActive(false);
	
	img2.SetActive(true);
	input2.SetActive(true);
	playBtn2.SetActive(true);
	text2.SetActive(true);
	}
	
	else
	{
	inputField.text = "INVALID";
	}
	}
	
	public void Clear()
	{
	inputField.text = "";
	}
	}	

3.4 Testing

In the testing phase, the game usability and playability were tested. The tester consists of other project game developers who played the game to identify the functionality of the features and its difficulty is

suitable to be played. The tester played the VR game and identified problems regarding the overall function of the game. The tester informed the developer to fix the bugs, loopholes and many more.

3.5 Beta

In the beta phase, the third party which is the target user will then test the game. The testing criteria will focus on usability of the game and make sure the objective of the game is achieved. The close beta types of selection are selected as target user of this project is teenagers aged from 13 to 18 years old. Students in Form 1 to Form 5 from Sekolah Menengah Kebangsaan Mantin, Negeri Sembilan are selected to test the game. Beta testing using the System Usability Scale (SUS) is conducted physically in the school. After the close beta testing takes place, bug reports and user feedback are analyzed for the developer to make final adjustments to the game.

3.6 Release

In the release phase, the game is released to the public for them to download. The project documentation will be in the terms of the final report. Post-mortems, planning for maintenance and game expansion are discussed to improve the game for future updates.

4. Results and Discussion

This section shows the data and analysis of functional testing and user acceptance testing in the form System Usability Testing (SUS) model.

4.1 Functional testing

Several game developers that work on different projects tested physically using Oculus Quest 2 and played the VR game. The button functionality, game object display, animation, game functionality and many more are tested. Table 7 discusses the test conducted, expected result of each test, actual result of the test and the corrective action to improve the overall functionality of the VR game.

Table 7: Functional testing

Test	Expected Result	Actual Result	Corrective Action
New game button	Navigate to the in-game scene	Works well as expected	Not needed
Load game button	Continue game in the last save point and scene	Works well as expected	Not needed
Option button	Display option menu	Works well as expected	Not needed
Exit button	Exit the game	Works well as expected	Not needed
Interface and prop display	Interface and prop display automatically every scene	Works well as expected	Not needed
Pause floating menu	The menu displays every time player press button	Works well as expected	Not needed
Story animation	Story animation played automatically	Animation played automatically but there is bug in character movement	The character animation is tweak and reimport into the character
Task checked out	Every task checked out automatically when finish the task	Works well as expected	Not needed
Environment asset, camera, and light	Environment asset, camera and light generated automatically for each scene	Works well as expected	Not needed
Object interaction	The physics of the object can be seen and could be interacted	Works well as expected	Not needed

Object position in hand	Object place naturally in hand when grab	Some objects do not properly attach with the hand	Assign attach transform with the specific position and rotation
Game Control	Precise and detail game control instructions to the user	Works well as expected	Not needed

4.2 User Acceptance Testing

System Usability Scale (SUS) is used in this testing. SUS is one of the models used in measuring the usability of a system, application, or games [14]. In the form of Google Form questionnaire, the SUS questionnaire is distributed to the students after they have done testing the VR game physically. It has 10 items with 5 response options from strongly disagree to strongly agree [15]. A total of 20 responses were collected and analyzed. Figure 11 and Figure 12 show the analysis of user acceptance level from question 1 to question 10.

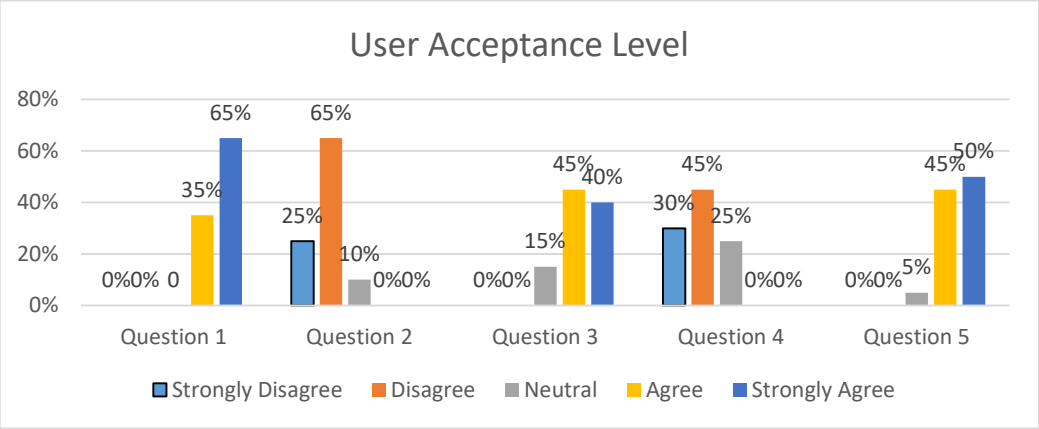

Figure 11: Analysis of User Acceptance Level Question 1 to Question 5

Question 1: I would like to play the VR game frequently.
Question 2: I found the VR game unnecessarily complex.
Question 3: I thought the VR game was easy to use.
Question 4: I think that I would need the support of a technical person to be able to play this VR game.
Question 5: I found the various functions and interactions in this VR game were well integrated.

Based on Figure 11, an average of 65% of respondents disagreed, 25% respondents strongly disagreed and 10% respondents neutral about the complexity of the VR game. This shows that while there is some complexity in the VR game, it does not hinder them from learning and performing the job tasks. The average of 45% of respondents agreed, 40% respondents strongly agree and 15% respondents neutral with the thought that the VR game was easy to use. This is because the target user consisted mostly of new users and inexperience in using Virtual Reality headset. Therefore, making it to be daunting at first but they could easily learn how to use and play the VR game.

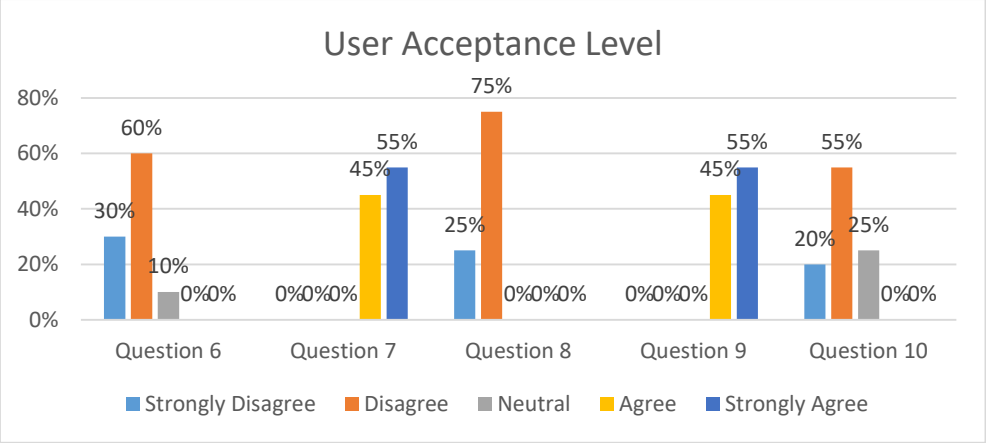

Figure 12: Analysis of User Acceptance Level Question 6 to Question 10

Question 6: I thought there was too much inconsistency in this VR game.
Question 7: I would imagine at most people would learn to use this VR game very quickly.
Question 8: I found the VR game very hard to handle or manage.
Question 9: I felt very confident playing this VR game.
Question 10: I needed to learn a lot of things before I could play this VR game.

Based on Figure 12, an average of 60% of respondents disagreed, 30% respondents strongly disagreed and 10% respondents neutral about there was too much inconsistency in the VR game. This shows that the overall of the VR game is consistent and not causing confusion to the target user when playing the VR game. The statistics in Figure 12 also shows that an average of 55% respondents strongly agree and 45% respondents agreed on the statement that most people would learn to use this VR game quickly. Thus, it could be seen that while the VR game and experience are new to the target user, they could learn quickly and understand the instructions in the VR game.

Next, the System Usability Scale is used to analyse the respondents score based on the positive statements and negative statements. The questionnaire contains five positive and five negative statements. Figure 13 shows the analysis of positive questions while Figure 14 shows the analysis of negative statements.

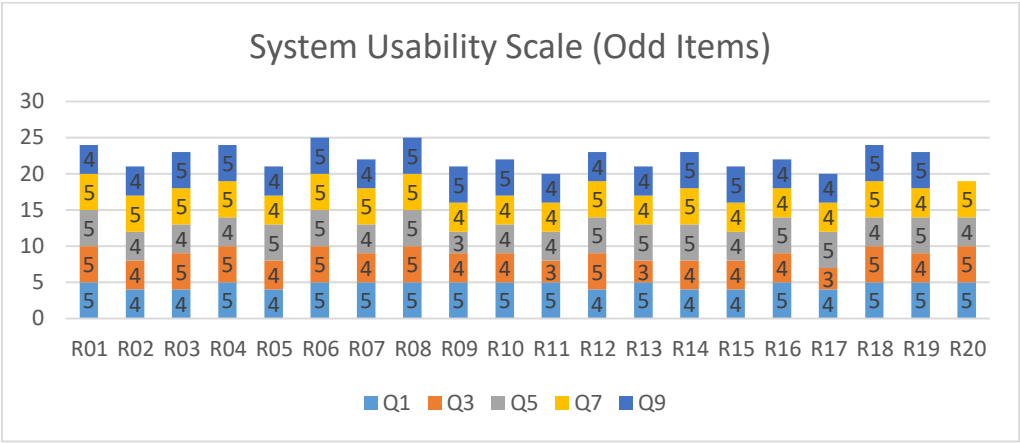

Figure 13: Analysis of Positive Questions

Question 1: I would like to play the VR game frequently.
Question 3: I thought the VR game was easy to use.
Question 5: I found the various functions and interactions in this VR game were well integrated.
Question 7: I would imagine at most people would learn to use this VR game very quickly.
Question 9: I felt very confident playing this VR game.

Based on Figure 13, many respondents gave 4 marks and above, which agrees with the positive statements in the questionnaire. Although some of the questions are given the score of 3 which is neutral, none of them disagrees with the positive outcome.

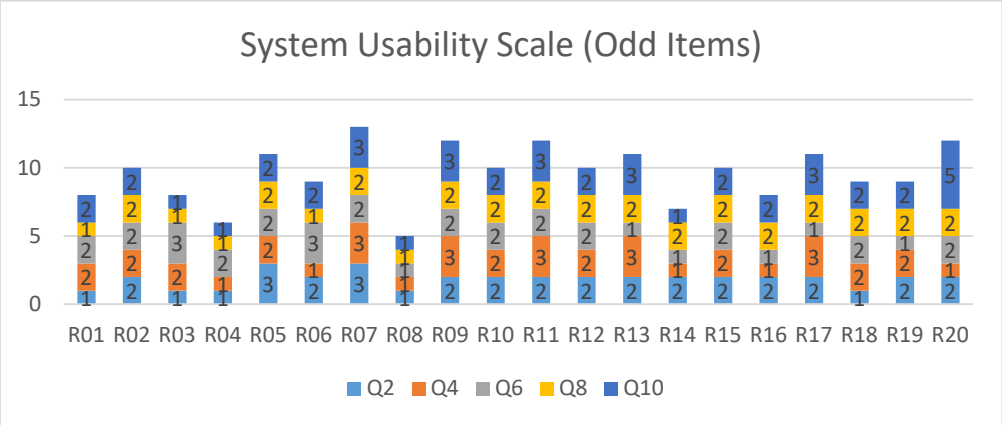

Figure 14: Analysis of Positive Questions

Question 2: I found the VR game unnecessarily complex. Question 4: I think that I would need the support of a technical person to be able to play this VR game. Question 6: I thought there was too much inconsistency in this VR game. Question 8: I found the VR game very hard to handle or manage. Question 10: I needed to learn a lot of things before I could play this VR game.

Based on Figure 14, all respondents gave a score between 1 to 3, which disagrees with the negative statements in the questionnaires. The result shows that most respondents do not have much problem playing the VR game and they could learn easily with the instructions given.

The total scores for each question from the user acceptance tests were shown in Table 8 in detail and the average score is calculated using the formula.

Table 8: Respondent's Score (User Acceptance Test)

Respondent	Item Score										Total Score
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
R01	5	1	5	2	5	2	5	1	4	2	90.0
R02	4	2	4	2	4	2	5	2	4	2	77.5
R03	4	1	5	2	4	3	5	1	5	1	87.5
R04	5	1	5	1	4	2	5	1	5	1	95.0
R05	4	3	4	2	5	2	4	2	4	2	75.0
R06	5	2	5	1	5	3	5	1	5	2	90.0
R07	5	3	4	3	4	2	5	2	4	3	72.5
R08	5	1	5	1	5	1	5	1	5	1	100.0
R09	5	2	4	3	3	2	4	2	5	3	72.5
R10	5	2	4	2	4	2	4	2	5	2	80.0
R11	5	2	3	3	4	2	4	2	4	3	70.0
R12	4	2	5	2	5	2	5	2	4	2	82.5
R13	5	2	3	3	5	1	4	2	4	3	75.0
R14	4	2	4	1	5	1	5	2	5	1	90.0
R15	4	2	4	2	4	2	4	2	5	2	77.5
R16	5	2	4	1	5	1	4	2	4	2	85.0
R17	4	2	3	3	5	1	4	2	4	3	72.5
R18	5	1	5	2	4	2	5	2	5	2	87.5

R19	5	2	4	2	5	1	4	2	5	2	85.0
R20	5	2	5	1	4	2	5	2	5	2	87.5
Average Score											82.63

Table 8 shows the total score of the usability value of Inside: VR Job Simulator game that has been analysed and evaluated. Next, the average score of usability value is calculated according to the given formula of the SUS test.

The formula used to obtain usability results based on SUS are:

Total score = (odd items + even items) x 2.5

$$\text{Average score} = \frac{\text{Total score}}{\text{Total respondents}}$$

Where:

Odd items (Q1, Q3, Q5, Q7, Q9) = 5 – contribution

Even items (Q2, Q4. Q6, Q8, Q10) = contribution – 1

Therefore,

Average score

$$= \frac{90+77.5+87.5+95+75+90+72.5+100+72.5+80+70+82.5+75+90+77.5+85+72.5+87.5+85+87.5}{20}$$

= 82.63

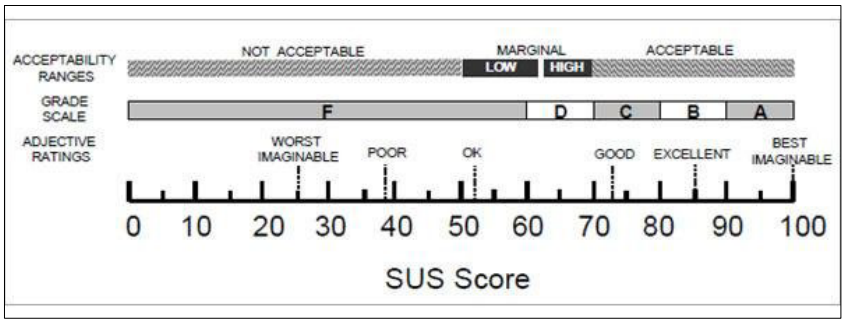

Figure 16: System Usability Scale (SUS)

Based on the SUS score scale in Figure 16, the average score of the usability value is 82.63 which in the range of Acceptable in the score scale. From the result, the VR game could be considered as success as it consider as usable and acceptable to the target user.

5. Conclusion

In conclusion, the Inside: VR Job Simulator game was successfully developed with two jobs namely doctor and game developer jobs. Next, the usability testing obtained the score of 82.63 in the SUS score which is within the acceptable range based on the System Usability Scale (SUS).

All three objectives set from the beginning of the project have been achieved. The first objective was achieved by designing the VR game based on user analysis and comparison between three existing applications. Next, the second objective was achieved when the game has been developed using Unity and XR interaction toolkit that allows the development and interaction of job simulation in Virtual Reality in Unity. Lastly, the third objective was achieved when the functional testing has been conducted and user acceptance results were collected from the target user and produce the SUS score of 82.63.

The Game Development Life Cycle (GDLC) methodology assisted this project to be completed on time. Additionally, the advantages and limitations of Inside: VR Job Simulator game are tabulated in Table 9.

Table 9: Advantages and Limitations of the Inside: VR Job Simulator game

Advantages	Limitations
Provides experience to the user to experience the real job tasks and equipment in the industry	Does not include more than two jobs
Consist of storyline that gives moral values to the target user	Limitation in task of the doctor jobs and game developer jobs
Consist of useful information about patient diseases and game developer tasks that mimics the real world	There is no sandbox or practice module for the user to play after finishing the game
Resulting in a high user acceptance level, where the target users agreed that the game is easy to use and understand	Does not include a complex task due to incorporation of storyline

Finally, for future work of this VR game, it is suggested to add more jobs so that user could experience multiple jobs in one game to information and insight of those jobs. Sandbox mode or practice module could be implemented so that it could also be used as practical training or test for the student studying these jobs in the game. More tasks and equipment could also be added to add more comprehensive experience to the target user when playing the game.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Y. Y. Liao, I. Hsuan Chen, Y. J. Lin, Y. Chen, and W. C. Hsu, "Effects of virtual reality-based physical and cognitive training on executive function and dual-task gait performance in older adults with mild cognitive impairment: A randomized control trial," *Front Aging Neurosci*, vol. 10, no. JUL, 2019, doi: 10.3389/fnagi.2019.00162.

[2] O. Bamodu and X. Ye, "Virtual Reality and Virtual Reality System Components," 2013.

[3] M. Fominykh and E. Prasolova-Forland, "Immersive job taste: A concept of demonstrating workplaces with virtual reality," in *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, Mar. 2019, pp. 1600–1605. doi: 10.1109/VR.2019.8798356.

[4] R. Sultanova and R. Sharaeva, "Virtual reality-based immersive simulation mechanics for invasive surgery training," in *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, Oct. 2019, vol. October-2019, pp. 924–928. doi: 10.1109/DeSE.2019.00171.

[5] S. Grassini, K. Laumann, and M. Rasmussen Skogstad, "The Use of Virtual Reality Alone Does Not Promote Training Performance (but Sense of Presence Does)," *Front Psychol*, vol. 11, Jul. 2020, doi: 10.3389/fpsyg.2020.01743.

[6] K. . Kozma, A. . Meyer-Miner, J. . Chio, S. Mak, A. El-Boraie, and D. Sealey, "Developing an Industry Job Simulation Program for Graduate and Postdoctoral Trainees in Life Sciences", *cjcd*, vol. 20, no. 2, pp. 84–93, Sep. 2021.

[7] A. Barto, "5 problems with virtual reality training they don't want you to know," *Roundtable Learning*, 04-May-2021. [Online]. Available: https://roundtablelearning.com/5-problems-with-virtual-reality-training-they-dont-want-you-to-know/. [Accessed: 10-Dec-2022].

[8] R. Ramadan and Y. Widyani, "Game development life cycle guidelines," in *2013 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, 2013, pp. 95–100. doi: 10.1109/ICACSIS.2013.6761558.

[9] A. Saeed Alqahtani, L. Foaud Daghestani, and L. Fattouh Ibrahim, "Environments and System Types of Virtual Reality Technology in STEM: A Survey," 2017. [Online]. Available: www.ijacsa.thesai.org

[10] (2016). Job Simulator (Version 2018.4.13.4816648) [Game]. Retrieved from Steam. https://store.steampowered.com/app/448280/Job_Simulator/

[11] (2016). Surgeon Simulator VR: Meet the Medic (Version 5.2.3.15978858) [Game]. Retrieved from Steam. https://store.steampowered.com/app/457420/Surgeon_Simulator_VR_Meet_The_Medic/

[12] (2022). Electrical Safety VR Training (Version 2020.1.17.10049454) [Game]. Retrieved from Steam. https://store.steampowered.com/app/2187560/Electrical_Safety_VR_Training/

[13] M. S. I. Abdul Fadil and N. Wahid, "Pembangunan Aplikasi m-Pembelajaran Secara Realiti Maya untuk Sains Tahun 4 ", *aitcs*, vol. 2, no. 2, pp. 650–667, Nov. 2021

[14] J. Brook, "SUS: a retrospective," J. Usability Studies, vol. 8, no. 2, pp. 29–40, 2013

[15] J. Sauro, "Measuring usability with the system usability scale (SUS)," MeasuringU, https://measuringu.com/sus/ (accessed Jun. 20, 2023).