

Design Specification

VR Medical Training Tool



Version 1.0

Jeremy Cavallo, Luka Cvetko, Slav Ivaskiv, Robert Tedeschi

Version 1.0

Jeremy Cavallo, Luka Cvetko, Slav Ivaskiv, Robert Tedeschi

Table of Contents

[2 Introduction and Purpose 2](#_Toc130988249)

[3 General Overview, Design Guidelines, and Approach 3](#_Toc1959883121)

[3.1 Assumptions and Constraints 4](#_Toc860336821)

[3.1.1 Assumptions 4](#_Toc2032825159)

[3.1.2 Constraints 4](#_Toc1618423505)

[4 Architecture Design 5](#_Toc950569551)

[4.1 Hardware Architecture 6](#_Toc127752276)

[4.2 Software Architecture 6](#_Toc1861605358)

[4.3 Security Architecture 6](#_Toc441440860)

[4.4 Communication Architecture 6](#_Toc1968679937)

[4.5 Performance 7](#_Toc501899546)

[5 System Design 7](#_Toc1186176634)

[5.1 Use Cases 8](#_Toc1306225924)

[5.2 Sequence Diagram 28](#_Toc2114842260)

[5.3 Data Flow Diagram 31](#_Toc1256702890)

[5.4 Use Case Diagram 31](#_Toc893885203)

[5.5 Database Design 31](#_Toc1701266477)

[5.6 Class Diagram 31](#_Toc1138085449)

[5.7 Application Program Interfaces 31](#_Toc48816884)

[5.8 User Interface Design 31](#_Toc1307494699)

[6 Appendix A – References 31](#_Toc508873849)

[7 Appendix B 31](#_Toc274564339)

# Introduction and Purpose

This document is intended to outline the exact steps we will take to implement all of the features we defined in the Software Requirement Specification. This includes defining the hardware, software, security, and communication architectures, as well as the system design itself. The system design consists of various diagrams outlining how different parts of the system interact, how data is passed through the software, use cases for the software, as well as further design considerations. This document serves as a high-level overview of each piece of code needed to implement the features for our software.

# General Overview, Design Guidelines, and Approach

## Assumptions and Constraints

### Assumptions

For the duration of the project, we are assuming that the XR Interaction Toolkit version we have utilized will remain stable and there will be no breaking changes introduced to it during the life cycle of the project that will affect our development. The XR Interaction Toolkit is provided and maintained by Unity, and we will be utilizing it heavily to capture user interactions for our tool. We are also assuming that the XR Interaction Toolkit provides all necessary data about the user’s inputs, specifically relating to the location and velocity of the user’s controllers.

We are also assuming that all equipment required for development, Testing, and demos will be available and in working condition throughout the project’s life cycle.

The statements offered in this document depend heavily on the timeframe of this project. The scope and specifications of the project will not change after the project has started, and the project will be in working condition by the beginning of April 2023. However, with the hard project deadline, certain requirements may be removed, and the priority of requirements may change. As the due date approaches, we will adjust our priorities to ensure this medical Training tool meets the minimum requirements to meet the needs of the user.

All people involved in the project will remain for the duration of the project to ensure the integrity and steady development pace.

All team members possess sufficient skills for the project. However, work quality may differ based on the difference in experience and Training. Also, team members might not be in their best condition throughout the duration of project, which might lead to performance gaps and violate the assumption of efficient work.

### Constraints

#### General Constraints

General constraints for this Training tool are around VR development tools and VR headset design limitations. VR headsets and controllers have a limited number of buttons. Trying to implement per-finger movement with different combinations of buttons will be confusing to users, increase learning curve, and defeat a purpose of simple controls that are appropriate for VR. There is also no precise implementation of hand tracking, which means there is a limit to the number of fine-motor movements that can be simulated in the scenarios.

We also need to choose a development platform that is supported in the target VR headsets. That platform should be mature enough, and is still being maintained, to eliminate potential problems that could affect and complicate the development process.

#### Design Constraints

Due to the design and implementation of VR controllers, it is impossible to implement some actions that require specific conditions or fine motor skills of human hands. For example, bandage wrapping, and head trauma scenario would require precise interactions with accuracy of human hands like opening a small bandage packaging, unwrapping bandage, or wrapping bandage around body part. Also, with VR controllers being basically in the air, it is also impossible to implement physical resistance like in CPR procedure.

Next are the limitations of XR Interaction Toolkit which is used for VR development in Unity Engine. Oculus Quest supports hand tracking without controllers which could alleviate the issues mentioned in previous paragraph. However, this functionality is not implemented in XR Interaction Toolkit. These limitations call for alternative implementations or simplified versions of certain interactions.

Processing power of Meta VR headsets, which are target platform for the application, should also be accounted for in the project. Graphical options such as complexity of the environment and 3D model resolutions need to be limited to accommodate for smooth and stable performance.

Considering the time constraint within which the project needs to be completed and set deadlines, there is a limit for scope of certain disciplines, such as what can be researched, implemented, and accomplished.

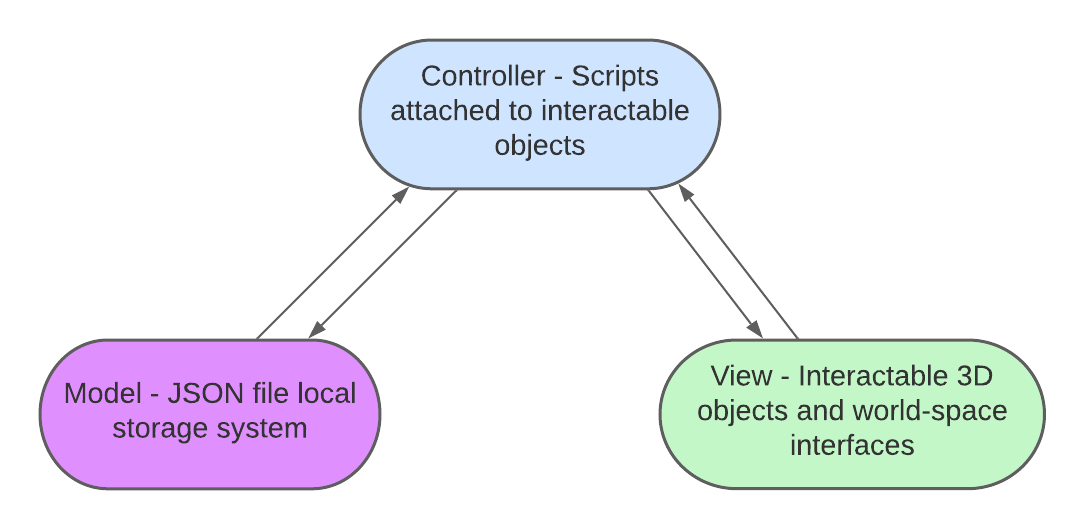
# Architecture Design

## Hardware Architecture

Non-applicable

## Software Architecture

This software will implement an MVC architecture, which fits the game development environment. Users will interact with 3D models within our scene, as well as world-space UI elements. These combine to make up the presentation (or “view”) layer of the MVC architecture. Attached to each of these interactable components are scripts that are triggered when the user interacts in a certain way. These act as the controller layer in the MVC model since they control what happens in the scene. Finally, the model layer for our application is a JSON storage system, which will be updated incrementally by the controllers to store the user’s progress for each scene. A visual of the software architecture is provided below.



## Security Architecture

Since our software is entirely offline and will not store sensitive information, we have few security concerns. However, we must ensure that users cannot tamper with their save data, which could give them access to test scenarios prior to completing Training scenarios or allow them to complete scenarios without going through them. Therefore, we will encrypt the local JSON file so that users can’t access it and change it.

## Communication Architecture

As stated in section 4.1, we must capture the button presses, physical location, and velocity of the VR controllers. The protocol used for communication between our software and the controllers is defined by the XR Interaction Toolkit, which is a library provided by Unity. Data from the controllers is passed to our app through this toolkit. We can simply define which kinds of interactions we would like to capture through the scripts provided in this library, and these scripts will communicate automatically with the controllers. For example, if we would like the user to be able to grab an object in a scene, we can attach a *XR Grab Interactable* script to that object, which will detect when the user grabs the object.

## Performance

The performance aspect of the application is heavily reliant on the capabilities of target devices, specifically the ones that can be used standalone, such as Oculus Quest 2. Motion sickness is one of the side effects that can be caused by VR usage. Despite multiple causes for it, our goal is to target aspect that can be controlled by us, which is framerate. Our target is to hit a minimum of 60 frames per second at all times.

Considering that this is a Training simulator, short response time of application to user actions is crucial. It is worth noting that there are some aspects that are out of our control such as hardware malfunction or insufficient lighting that can affect some of the target devices. Despite that, during Testing we can utilize tools available in Unity Engine to ensure which can trace and measure response time. That should give us a close idea whether there are issues in that aspect. Another target is to take an event-based approach for interaction scenarios which will be triggered based on interactions. This way, overhead of expensive operations, such as waiting loops, can be minimized and issues that can be caused by them can be avoided.

# System Design

## Use Cases

|  |  |
| --- | --- |
| **ID**: UC1 | **Name**: Scene Selection |
| **Description**: This use case relates the user selecting one of the scenes. This use case will occur when the user is in the main menu scene of the Training tool. Possible scenes to select are the CPR scene, the bandage scene, the head trauma scene, and the Heimlich maneuver scene. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered whenever the user loads the app, since the main menu scene is the first scene loaded. | |
| **Preconditions**: The user opened the application and is in the main menu. User can see 4 available scenes, which are Bandage Scene, CPR Scene, Head Trauma Scene, and Heimlich Maneuver Scene | |
| **Postconditions**: The selected scene is loaded, and the user is teleported into that scene. | |
| **Normal** **Flow**:   1. The user opens the application 2. The user is loaded into the main menu scene 3. The user points controller on one of the scenes and clicks controller button to select the scene 4. The user is teleported to the corresponding scene | |
| **Alternative** **Flows**:   1. The user completes a scene, where they can click a button to navigate to the main menu 2. Once they travel back, the main menu scene should continue the normal flow path from step 3 | |
| **Exceptions**: None | |
| **Assumptions**: Application opens and main menu scene loads successfully | |

|  |  |
| --- | --- |
| **ID**: UC2 | **Name**: Scenario Mode Selection |
| **Description**: This use case relates the user selecting one of the modes inside scene. This use case will occur when the user is in the Head Trauma or CPR scenes of the Training tool. Available modes to select are Training and Testing. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered whenever the user loads either the head trauma or CPR scene. | |
| **Preconditions**: The user selected a scene in Main Menu (UC2) and is loaded into scene | |
| **Postconditions**: The selected mode is loaded, and the user is teleported into that mode. | |
| **Normal** **Flow**:   1. The user is prompted to select one of the available mode options, Training or Testing 2. User points controller and selects corresponding button to start selected mode 3. Corresponding mode is started for the user | |
| **Alternative** **Flows**:   1. The user completes a mode, where they can select to 2. Navigate to the main menu 3. Repeat current mode 4. Select different mode | |
| **Exceptions:**   1. User has not completed Training mode and attempts to click Testing Mode button 2. System provides warning message that user has to complete Training Mode first | |
| **Assumptions**: Selected mode loads successfully | |

|  |  |
| --- | --- |
| **ID**: UC3 | **Name**: Sub-Scenario Selection |
| **Description**: This use case relates to the user selecting one of the sub-scenarios. Both the bandage and Heimlich maneuver scenes have multiple sub-scenarios to select between. When the scene is first loaded, there will be a UI where the user can select between these different sub-scenarios. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered whenever the user loads either the bandage scene or the Heimlich maneuver scene. | |
| **Preconditions**: The user selected a Bandage or Heimlich Maneuver scene in Main Menu (UC2) and is loaded into selected scene | |
| **Postconditions**: The selected sub-scenario should begin once the user makes a selection. | |
| **Normal** **Flow**:  Bandage Scene:   1. User navigates to bandage scene from main menu 2. User views UI and selects from two options:    1. Small Cuts and Scrapes    2. If the user selects Large Wounds with Heavy Bleeding 3. After user selects one of the 2 options, they will be prompted with the selection of modes, Testing or Training 4. After user selects mode, corresponding mode will load   Heimlich Maneuver Scene:   1. User navigates to Heimlich maneuver scene from main menu 2. User views UI and selects from three options:    1. Heimlich maneuver: adult    2. Heimlich maneuver: infant    3. Heimlich maneuver: self 3. After user selects one of the 3 options, they will be prompted with the selection of modes, Testing or Training 4. After user selects mode, corresponding mode will load | |
| **Alternative** **Flows**:   1. User completes Testing or Training mode of selected scene, which provides an option to return to Main Menu 2. User returns to Main Menu 3. Normal flow will continue | |
| **Exceptions**:   1. User accidentally selects wrong sub-scenario 2. User selects the “back” button on the following UI menu, which will reload the initial sub-scenario selection menu 3. Flow continues down normal path 4. User selects a sub-scenario 5. User selects Testing, but Testing is not yet unlocked (the user must complete the corresponding Training scenario to unlock Testing) 6. Message is displayed saying the user must complete the Training scenario before Testing is unlocked 7. User selects Training, or navigates back to select different sub-scenario 8. Flow continues down normal path 9. User navigates to scene after previously completing a scenario within that scene 10. Scene should be reloaded as if they navigated from main menu, and flow should continue down the normal path. | |
| **Assumptions**: Bandage scene or Heimlich Maneuver scene is loaded successfully | |
| **ID**: UC4 | **Name**: Training: Small Cuts and Scrapes |
| **Description**: This use case encompasses the interactions necessary to complete the Small Cuts and Scrapes Training scenario. It outlines each item that should be interacted with, in the correct order, as well as what happens when the user selects an item out of order. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user selects the Small Cuts and Scrapes Training scenario from UC3. | |
| **Preconditions**: The user is in the bandage scene and selects the Small Cuts and Scrapes Training scenario from UC3. | |
| **Postconditions**: The user should receive a notification window about the completion of the Training scenario, and the user should be given a choice to start Testing mode or navigate back to the main menu. | |
| **Normal** **Flow**:   1. The user selects the Small Cuts and Scrapes Training scenario 2. The user interacts with the sink handle, as prompted by the doctor helper, audio queues, and the UI near the sink handle. This simulates the user washing their hands. 3. The user interacts with the door under the sink, which is necessary to grab the bandage for the next step 4. The user grabs the bandage from under the sink, as prompted 5. The user drops the bandage on the wound near the patient’s left leg, which simulates the application of pressure to the wound to stop bleeding 6. The user grabs an antibiotic substance off the sink counter, as prompted 7. The user drops the antibiotic substance near the wound, which simulates the application of the antibiotic to the wound 8. The user grabs a new bandage 9. The user drops the new bandage near the wound, which will simulate the application of the bandage to the wound 10. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Training. This menu will allow the user to navigate to the Testing scenario or to the main menu     1. If the user navigates to the Testing scenario, the Testing scenario should be loaded     2. If the user navigates to the main menu, the main menu should be loaded, and the user should be teleported to the main menu scene. | |
| **Alternative** **Flows**: | |
| **Exceptions**:   1. If user interacts with the wrong object or object from already completed step, auditory and visual feedback will be given to the user 2. The user will need to interact with the correct object to continue 3. Once the user interacts with the correct object, the flow will continue down the normal path | |
| **Assumptions**: Bandage scene is loaded successfully; Small Cuts and Scrapes Training scenario is loaded successfully. | |

|  |  |
| --- | --- |
| **ID**: UC5 | **Name**: Training: Large Wounds with Heavy Bleeding |
| **Description**: This use case encompasses the interactions necessary to complete the Large Wounds with Heavy Bleeding Training scenario. It outlines each item that should be interacted with, in the correct order, as well as what happens when the user selects an item out of order. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user selects the Large Wounds with Heavy Bleeding Training scenario from UC3. | |
| **Preconditions**: The user is in the bandage scene and selects the Large Wounds with Heavy Bleeding Training scenario from UC3. | |
| **Postconditions**: The user should receive a notification window about the completion of the Training scenario, and the user should be given a choice to start Testing or navigate back to the main menu. | |
| **Normal** **Flow**:   1. The user selects the Large Wounds with Heavy Bleeding Training scenario 2. The user interacts with the phone on the wall near the door by grabbing it, as prompted by a UI near the phone, the doctor character model, and auditory queues. 3. The user interacts with the wound, which will simulate clearing debris and clothing from around the wound to prepare for the next steps 4. The user opens the door under the sink, which is necessary to access the bandage for the next step 5. The user grabs the bandage, as prompted 6. The user drops the bandage near the wound, which simulates the user applying pressure to the wound to slow bleeding 7. The user interacts with the patient’s upper body, which will simulate the patient lying down, as prompted 8. The user grabs the bandage again, as prompted 9. The user drops the bandage near the wound, which will simulate applying a bandage to the wound 10. The user grabs the tourniquet, as prompted 11. The user drops the tourniquet near the wound, which simulates applying the tourniquet to slow bleeding 12. The user washes their hands after help arrives, as prompted. 13. Upon completion of the final step, this will trigger the appearance of the end of scene menu, which congratulates the user on completing Training. It also allows the user to select between beginning the Testing scenario and navigating back to the main menu     1. If the user chooses to begin the test, the test scenario for Large Wounds with Heavy Bleeding should be loaded     2. If the user chooses to navigate to the main menu, the main menu scene should be loaded. | |
| **Alternative** **Flows**: | |
| **Exceptions**:   1. If user interacts with the wrong object or object from already completed step, auditory and visual feedback will be given to the user 2. The user will need to interact with the correct object to continue 3. Once the user interacts with the correct object, the flow will continue down the normal path | |
| **Assumptions**: Bandage scene is loaded successfully; Small Cuts and Scrapes Training scenario is loaded successfully. | |

|  |  |
| --- | --- |
| **ID**: UC6 | **Name**: Testing: Small Cuts and Scrapes |
| **Description**: This use case encompasses the interactions necessary to complete the Small Cuts and Scrapes Testing scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user selects the Small Cuts and Scrapes Testing scenario from UC3. | |
| **Preconditions**: The user is in the bandage scene and selects Small Cuts and Scrapes Testing scenario from UC3. This also requires that the user has completed the Small Cuts and Scrapes Training scenario. | |
| **Postconditions**: If the user completes the scenario with zero mistakes, the user should be given credit for passing the test scenario. If the user completes the scenario with mistakes, the user should be given feedback on where the mistakes were made. The user should be able to re-try the Testing scenario if they failed. If they succeeded, they should be given the option to navigate back to either the bandage scenario sub-scenario selection menu or the main menu. | |
| **Normal** **Flow**:   1. The user selects the Small Cuts and Scrapes Training scenario 2. The user interacts with the sink handle – this simulates the user washing their hands 3. The user interacts with the door under the sink, which is necessary to grab the bandage for the next step 4. The user grabs the bandage from under the sink 5. The user drops the bandage on the wound near the patient’s left leg, which simulates the application of pressure to the wound to stop bleeding 6. The user grabs an antibiotic substance off the sink counter 7. The user drops the antibiotic substance near the wound, which simulates the application of the antibiotic to the wound 8. The user grabs a new bandage 9. The user drops the new bandage near the wound, which will simulate the application of the bandage to the wound 10. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Testing. This menu will give the user feedback on their performance in the Testing scenario.     1. If they passed, the user should be given the option to navigate back to the main menu or the bandage scene sub-scenario selection menu     2. If they failed, the user should be given the option to retry the test, navigate back to the bandage scene sub-scenario selection menu, or navigate back to the main menu | |
| **Alternative** **Flows**: | |
| **Exceptions**:   1. If user interacts with wrong object, incorrect action will be logged. Normal flow will resume when user interacts with correct object. 2. If user interacts with wrong object in any of the steps, they will be required to repeat Testing scenario in order to pass. | |
| **Assumptions**: Bandage scene is loaded successfully, Small Cuts and Scrapes Testing scenario is loaded successfully, and the user has completed Small Cuts and Scrapes Training scenario. | |

|  |  |
| --- | --- |
| **ID**: UC7 | **Name**: Testing: Large Wounds with Heavy Bleeding |
| **Description**: This use case encompasses the interactions necessary to complete the Large Wounds with Heavy Bleeding Testing scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user selects the Large Wounds with Heavy Bleeding Testing scenario from UC3. | |
| **Preconditions**: The user is in the bandage scene and selects Large Wounds with Heavy Bleeding Testing scenario from UC3. This also requires that the user has completed the Large Wounds with Heavy Bleeding Training scenario. | |
| **Postconditions**: If the user completes the scenario with zero mistakes, the user should be given credit for passing the test scenario. If the user completes the scenario with mistakes, the user should be given feedback on where the mistakes were made. The user should be able to re-try the Testing scenario if they fail. If they succeeded, they should be given the option to navigate back to either the bandage scenario sub-scenario selection menu or the main menu. | |
| **Normal** **Flow**:   1. The user selects the Large Wounds with Heavy Bleeding Training scenario 2. The user interacts with the phone on the wall near the door by grabbing it 3. The user interacts with the wound, which will simulate clearing debris and clothing from around the wound to prepare for the next steps 4. The user opens the door under the sink, which is necessary to access the bandage for the next step 5. The user grabs the bandage 6. The user drops the bandage near the wound, which simulates the user applying pressure to the wound to slow bleeding 7. The user interacts with the patient’s upper body, which will simulate the patient lying down 8. The user grabs the bandage again 9. The user drops the bandage near the wound, which will simulate applying a bandage to the wound 10. The user grabs the tourniquet 11. The user drops the tourniquet near the wound, which simulates applying the tourniquet to slow bleeding 12. The user washes their hands after help arrives 13. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Testing. This menu will give the user feedback on their performance in the Testing scenario.     1. If they passed, the user should be given the option to navigate back to the main menu or the bandage scene sub-scenario selection menu     2. If they failed, the user should be given the option to retry the test, navigate back to the bandage scene sub-scenario selection menu, or navigate back to the main menu | |
| **Alternative** **Flows**: | |
| **Exceptions**:   1. If user interacts with wrong object, incorrect action will be logged. Normal flow will resume when user interacts with correct object. 2. If user interacts with wrong object in any of the steps, they will be required to repeat Testing scenario in order to pass. | |
| **Assumptions**: Bandage scene is loaded successfully, Large Wounds with Heavy Bleeding Testing scenario is loaded successfully, and the user has completed the Large Wounds with Heavy Bleeding Training scenario. | |

|  |  |
| --- | --- |
| **ID**: UC8 | **Name**: Training: CPR |
| **Description**: This use case encompasses the interactions necessary to complete the CPR Training scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in CPR scene and selects Training mode from UC2. | |
| **Preconditions**:   1. The user was in the CPR scene 2. The user selected Training mode from UC2 | |
| **Postconditions**: The user should receive a notification window about the completion of the Training scenario, and the user should be given a choice to start Testing or navigate back to the main menu. | |
| **Normal Flow:**   1. The user selects the CPR Training scenario 2. The user selects Training mode 3. The user is prompted with UI and auditory queues to interact with the phone on the wall near the door by grabbing it 4. The user is prompted to navigate to patient and place patient on their back on hard surface 5. The user is prompted to kneel down near patient and place hands on patient’s chest 6. The user is prompted to start pressing on patient’s chest with a certain tempo 7. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Training. This menu will give the user feedback on their performance in the Training scenario:    1. If they passed, the user should be given the option to navigate back to the main menu or start Testing mode    2. If they failed, the user should be given the option to retry the test or navigate back to the main menu | |
| **Alternative Flows:** | |
| **Exceptions**:   1. If user interacts with the wrong object or object from already completed step, auditory and visual feedback will be given to the user 2. The user will need to interact with the correct object to continue 3. Once the user interacts with the correct object, the flow will continue down the normal path | |
| **Assumptions**: CPR scene is loaded successfully, and Training mode has loaded successfully | |

|  |  |
| --- | --- |
| **ID**: UC9 | **Name**: Testing: CPR |
| **Description**: This use case encompasses the interactions necessary to complete the CPR Testing scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in CPR scene and selects Testing mode from UC2. It is also required that the user completes CPR Training beforehand. | |
| **Preconditions**:   1. The user has completed Training mode 2. The user was in the CPR scene 3. The user selected Testing mode from UC2 | |
| **Postconditions**: If the user completes the scenario with zero mistakes, the user should be given credit for passing the Training scenario. If the user completes the scenario with mistakes, the user should be given feedback on where the mistakes were made. The user should be able to re-try the Testing scenario if they fail. If they succeeded, they should be given the option to navigate back to either the CPR mode selection menu or the main menu. | |
| **Normal Flow**:   1. The user selects CPR scenario 2. The user selects Testing mode 3. The user interacts with phone on the wall by grabbing it 4. The user navigates to patient and places patient on their back on hard surface 5. User kneels near patient and places hands on patient’s chest 6. User starts pressing on patient's chest with a certain tempo 7. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Testing. This menu will give the user feedback on their performance in the Testing scenario:    1. If they passed, the user should be given the option to navigate back to the main menu or CPR scene mode selection    2. If they failed, the user should be given the option to retry the test, navigate back to the CPR scene mode selection, or navigate back to the main menu | |
| **Alternative Flows:** | |
| **Exceptions**:   1. If user interacts with wrong object, incorrect action will be logged. Normal flow will resume when user interacts with correct object. 2. If user interacts with wrong object in any of the steps, they will be required to repeat Testing scenario in order to pass. | |
| **Assumptions**:   1. User has completed Training scenario 2. CPR scene is loaded successfully 3. Testing mode has loaded successfully | |

|  |  |
| --- | --- |
| **ID**: UC10 | **Name**: Training: Head Trauma |
| **Description**: This use case encompasses the interactions necessary to complete the head trauma Training scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in the head trauma scene and selects Training from UC2 | |
| **Preconditions**: The user is in the head trauma scene and selects Training from UC2. | |
| **Postconditions**: The user should receive a notification window about the completion of the Training scenario, and the user should be given a choice to start Testing or navigate back to the main menu. | |
| **Normal Flow:**   1. User selects head trauma scenario 2. User selects Training 3. User interacts with phone on the wall by grabbing it 4. The user navigates to patient and keep them still 5. The user navigates to the sink and grab a bandage by opening the sink door and grabbing the bandage 6. The user takes bandage to the patient and applies it to the wound on the head of the patient 7. User then monitors the patient until help arrives 8. Upon completion of this Training the user will be credited with completion of the Training scenario and given a choice of starting the Testing or going back to main menu | |
| **Alternate Flows:** | |
| **Exceptions**:   1. If user interacts with the wrong object or object from already completed step, auditory and visual feedback will be given to the user 2. The user will need to interact with the correct object to continue 3. Once the user interacts with the correct object, the flow will continue down the normal path | |
| **Assumptions**: Head trauma scene loads correctly. | |

|  |  |
| --- | --- |
| **ID**: UC11 | **Name**: Testing: Head Trauma |
| **Description**: This use case encompasses the interactions necessary to complete the head trauma Testing scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in the head trauma scene and selects Testing from UC2 | |
| **Preconditions**: The user is in the head trauma scene and selects Testing from UC2. | |
| **Postconditions**: If the user completes the scenario with zero mistakes, the user should be given credit for passing the Training scenario. If the user completes the scenario with mistakes, the user should be given feedback on where the mistakes were made. The user should be able to re-try the Testing scenario if they fail. If they succeeded, they should be given the option to navigate back to either the head trauma mode selection menu or the main menu. | |
| **Normal Flow:**   1. User selects head trauma scenario 2. User selects Testing 3. User interacts with phone on the wall by grabbing it 4. The user navigates to patient and keep them still 5. The user navigates to the sink and grabs a bandage by opening the sink door and grabbing the bandage 6. The user takes bandage to the patient and applies it to the wound on the head of the patient 7. User then monitors the patient until help arrives 8. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Testing. This menu will give the user feedback on their performance in the Testing scenario:    1. If they passed, the user should be given the option to navigate back to the main menu or head trauma scene mode selection    2. If they failed, the user should be given the option to retry the test, navigate back to the head trauma scene mode selection, or navigate back to the main menu | |
| **Alternate Flows:** | |
| **Exceptions**:   1. If user interacts with wrong object, incorrect action will be logged. Normal flow will resume when user interacts with correct object. 2. If user interacts with wrong object in any of the steps, they will be required to repeat Testing scenario in order to pass. | |
| **Assumptions**: Head trauma scene loads correctly | |

|  |  |
| --- | --- |
| **ID**: UC12 | **Name**: Training: Heimlich Maneuver, Adult |
| **Description**: This use case encompasses the interactions necessary to complete the Heimlich maneuver adult Training scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in Heimlich Maneuver scene and selects Training mode and Adult from UC3. | |
| **Preconditions**:   1. The user was in the Heimlich Maneuver scene 2. The user selected adult scene from UC3 3. The user selected Training mode for adult scene from UC3 | |
| **Postconditions**: The user should receive a notification window about the completion of the Training scenario, and the user should be given a choice to start Testing or navigate back to the main menu. | |
| **Normal Flow:**   1. The user selects Heimlich Maneuver adult Training scenario 2. The user sees a person struggling to breathe, indicated with an arrow 3. Pop-up opens with the information that tells the user to see if the person can talk, by interacting with them. 4. After the interaction the confirmation that the person is indeed choking will pop up, and steps will be outlined on how to help them. 5. First the user will get behind the person and interact by placing their hand on the person's back 5 times, the area will be highlighted with a red marking, indicated with a counter which displays the number of touches. 6. The point of view will then change from first person, to an over the shoulder camera view where the user will be prompted to press down 5 times. 7. These steps then repeat randomly 1 to 3 times, until an animation plays that indicates the object has been dislodged out of the person's system. 8. After the last step, the end of scenario menu will be displayed, there three options will be given; continue to test, return to main menu, and repeat Training. | |
| **Alternate Flows:** | |
| **Exceptions**:   1. If user interacts with the wrong object or object from already completed step, auditory and visual feedback will be given to the user 2. The user will need to interact with the correct object to continue 3. Once the user interacts with the correct object, the flow will continue down the normal path | |
| **Assumptions**: Heimlich Maneuver adult Training scene loaded correctly. | |

|  |  |
| --- | --- |
| **ID**: UC13 | **Name**: Testing: Heimlich Maneuver, Adult |
| **Description**: This use case encompasses the interactions necessary to complete the Heimlich maneuver adult Testing scenario. It outlines each item that should be interacted with, in the correct order, what happens when the user selects an item out of order, and how correct and incorrect actions are calculated. | |
| **Primary** **Actors**: User | **Secondary** **Actors**: N/A |
| **Trigger**: This use case will be triggered when the user is in Heimlich Maneuver scene and selects adult Testing mode from UC3. It is also required that the user completes Heimlich Maneuver Training adult beforehand. | |
| **Preconditions**:   1. The user was in Heimlich Maneuver scene 2. The user selected Testing adult scenario from UC3 3. The user has finished Training adult scenario from UC3 | |
| **Postconditions**: If the user completes the scenario with no mistakes, the user should be given credit for passing the Training scenario in the form of a pop-up indicating success. If the user completes the scenario with some mistakes, the user should be given feedback on where the mistakes were made in the form of a log file. The user should be able to re-try the Testing scenario if they fail and go back to the Training option. If they succeeded, they should be given the option to navigate back to either the Heimlich Maneuver mode selection menu or the main menu. | |
| **Normal Flow:**   1. The user selects Heimlich Maneuver adult Testing scenario 2. The user sees a person struggling to breathe. 3. User interacts with person. 4. After the interaction the confirmation that the person is indeed choking will pop up, and a timer will start running. 5. First the user will get behind the person and interact by placing their hand on the person's back 5 times on highlighted area. 6. The point of view will then change from first person, to an over the shoulder camera view where the user will press down 5 times on highlighted area. 7. These steps then repeat randomly 1 to 3 times, until an animation plays that indicates the object has been dislodged out of the person's system. 8. After the last step of the scenario, the end of scenario menu will appear, telling the user they completed Testing. This menu will give the user feedback on their performance in the Testing scenario:    1. If they passed, the user should be given the option to navigate back to the main menu or Heimlich Maneuver scene mode selection    2. If they failed, the user should be given the option to retry the test, navigate back to the Heimlich Maneuver scene mode selection, or navigate back to the main menu. Here they will also see a list of mistakes on the results screen. | |
| **Alternative Flows:** | |
| **Exceptions**:   1. If user interacts with wrong object, incorrect action will be logged. Normal flow will resume when user interacts with correct object. 2. If user interacts with wrong object in any of the steps, they will be required to repeat Testing scenario in order to pass. 3. If the timer runs out, the user will fail the test immediately. | |
| **Assumptions**: Heimlich Maneuver scene is loaded successfully, and Testing mode has loaded successfully, and user has completed Training scenario | |

## Sequence Diagram

Calendar

Description automatically generated

Figure 1 Scenario Flow

Chart, box and whisker chart

Description automatically generated

Figure 2 Testing Scenarios

Calendar

Description automatically generated with medium confidence

Figure 3 Training: CPR

Timeline

Description automatically generated with medium confidence

Figure 4 Training: Small Cuts and Scrapes

## Data Flow Diagram

Diagram

Description automatically generated

Figure 4 VR Training Tool with Save State

## Use Case Diagram

Diagram

Description automatically generated

Figure 5 Use Case Diagram – General Flow

Diagram

Description automatically generated

Figure 6 Use Case Diagram - CPR Training with Save State

Diagram, engineering drawing

Description automatically generated

Figure 7 Use Case Diagram – Small Cuts and Scrapes Testing Mode

## Database Design

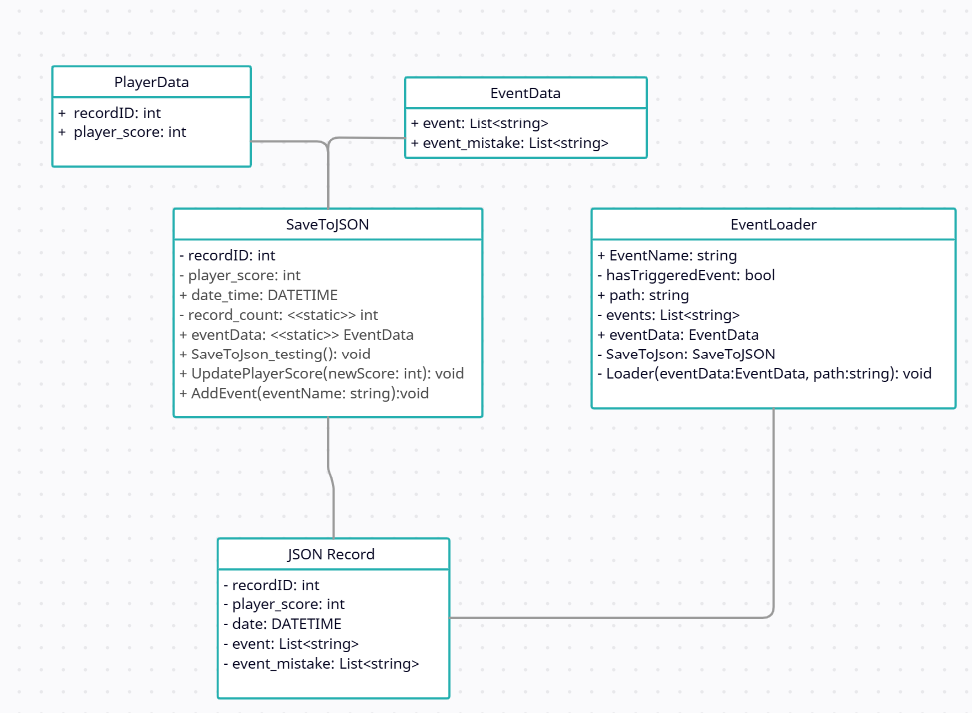


Figure 8 Database Diagram

## Class Diagram

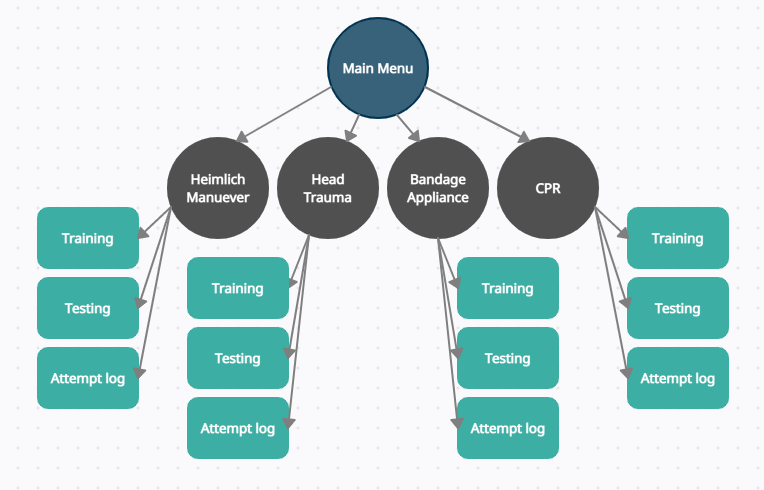


Figure 9 Class Diagram



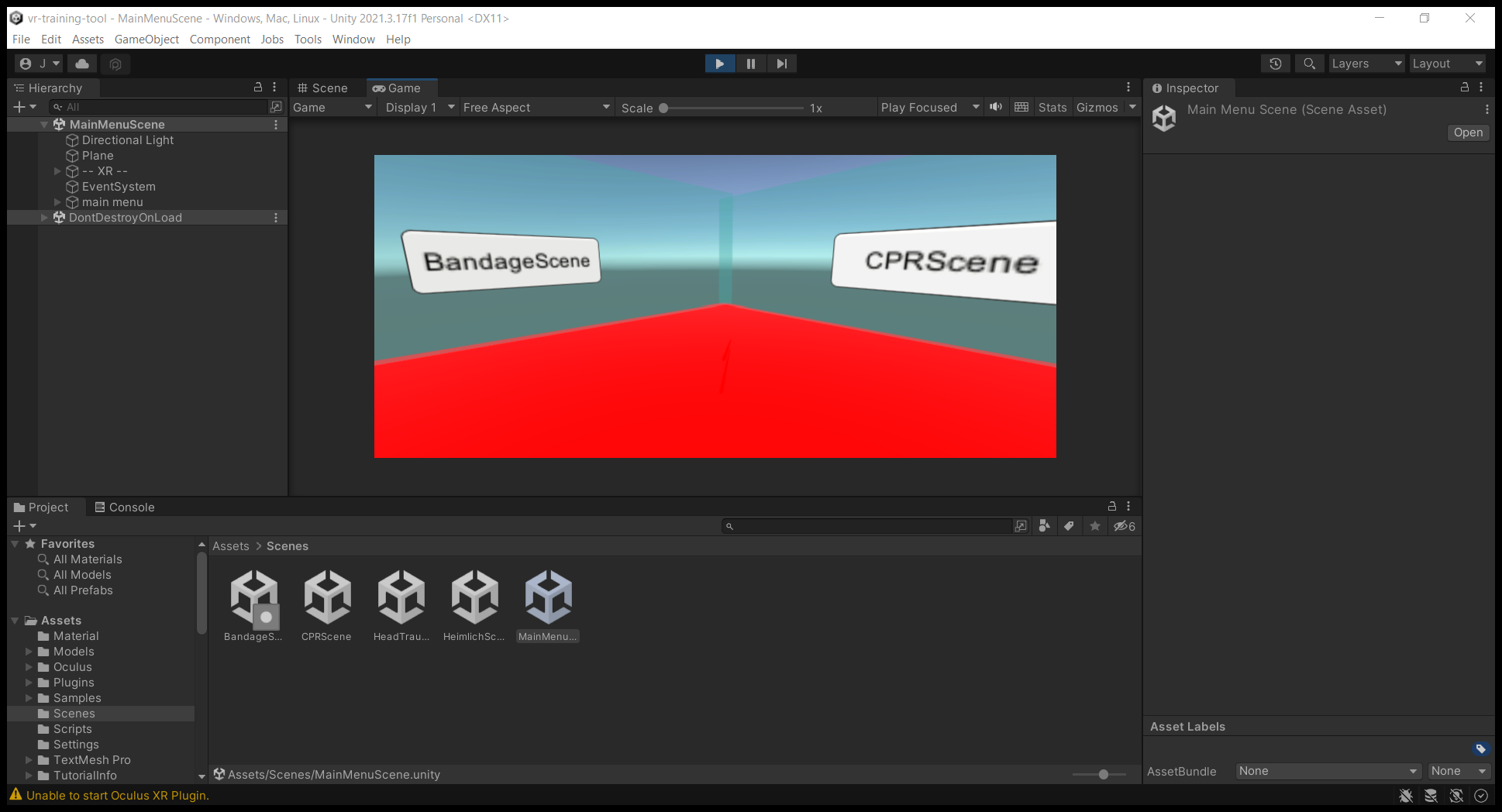
## Application Program Interfaces

Non-applicable

## User Interface Design

User interaction will be handled through both user interface elements and interactable objects in the VR settings. User interfaces will contain text explaining scenarios, cues for interactions (such as grabbing an object), and buttons for navigating between scenes. Interactable objects will be highlighted with a different color ray indicator while navigating the tool, as well as user interface cues.

When the user first enters our application, they will be placed in a scene containing the main menu. This scene will contain a navigation menu, with buttons corresponding to the bandage, CPR, head trauma, and Heimlich maneuver scenes. Pressing these buttons will allow users to navigate to the corresponding scenarios.



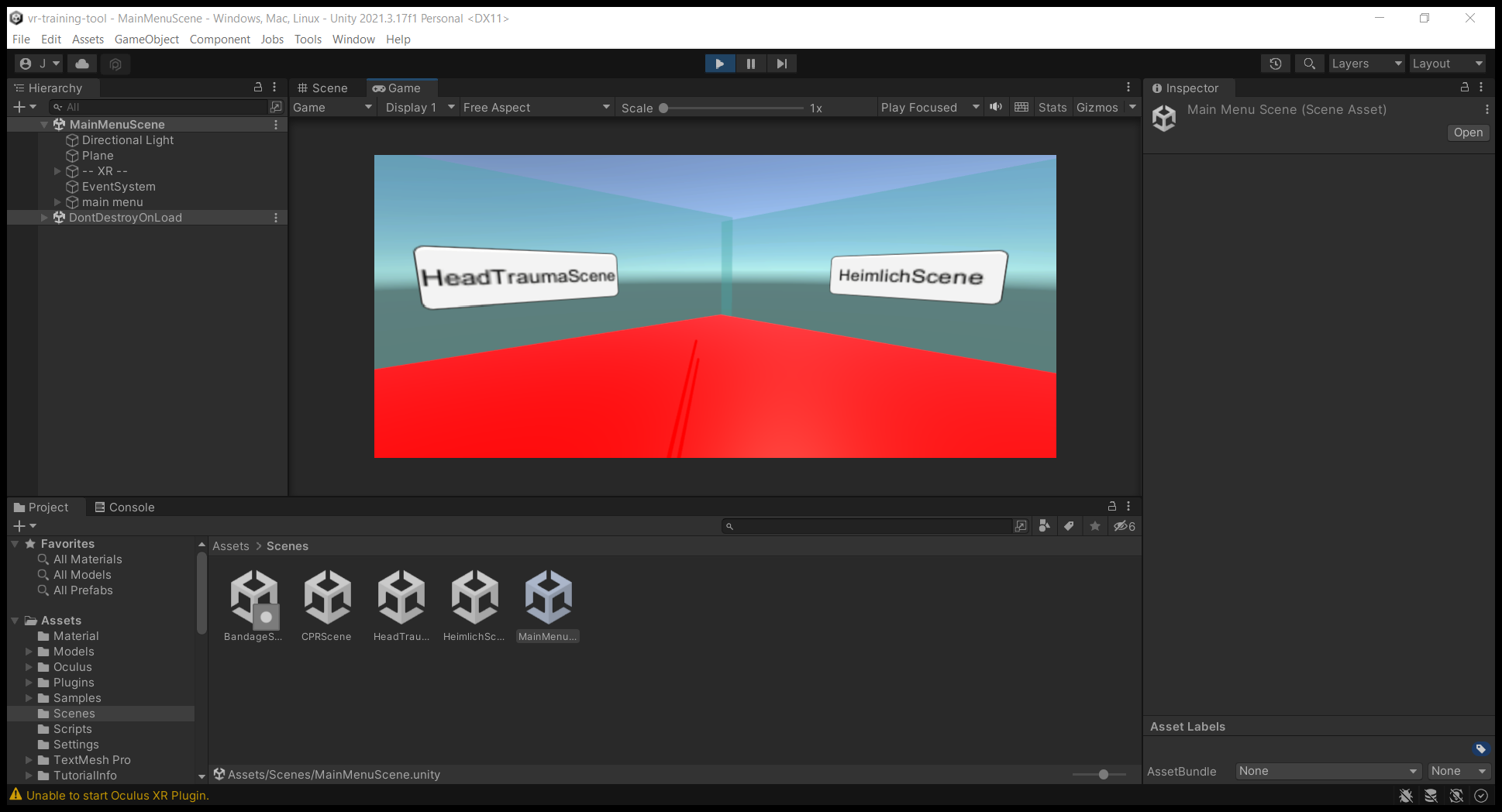


Figure 17: Main Menu User Interface with Selectable Scenarios

When the user navigates to a scenario, there will be information posted to a UI corresponding to the scenario. This information will explain the scenario and allow selection between the Training scenario and Testing scenario if it is unlocked. The bandage scene will have one extra user interface, which will ask the user to select between the Small Cuts and Scrapes and large wounds scenarios, which will each have sub-menus for Training and Testing scenarios.

Text

Description automatically generated

Figure 18: Bandage Scene UI for Information and Sub-Scenario Selection

Once users enter a Training or Testing scenario, user interaction will be tracked based on interactions with objects rather than specific user interface elements. These interactions include grabbing objects, dropping objects, moving the controllers to a specific position, or moving controllers at a specific velocity. These interactions are handled by the XR Interaction Toolkit in Unity and will trigger events to occur within the scenario.

Graphical user interface, text, application

Description automatically generated

Figure 19: Outline of Steps for Small Cuts and Scrapes Scenario

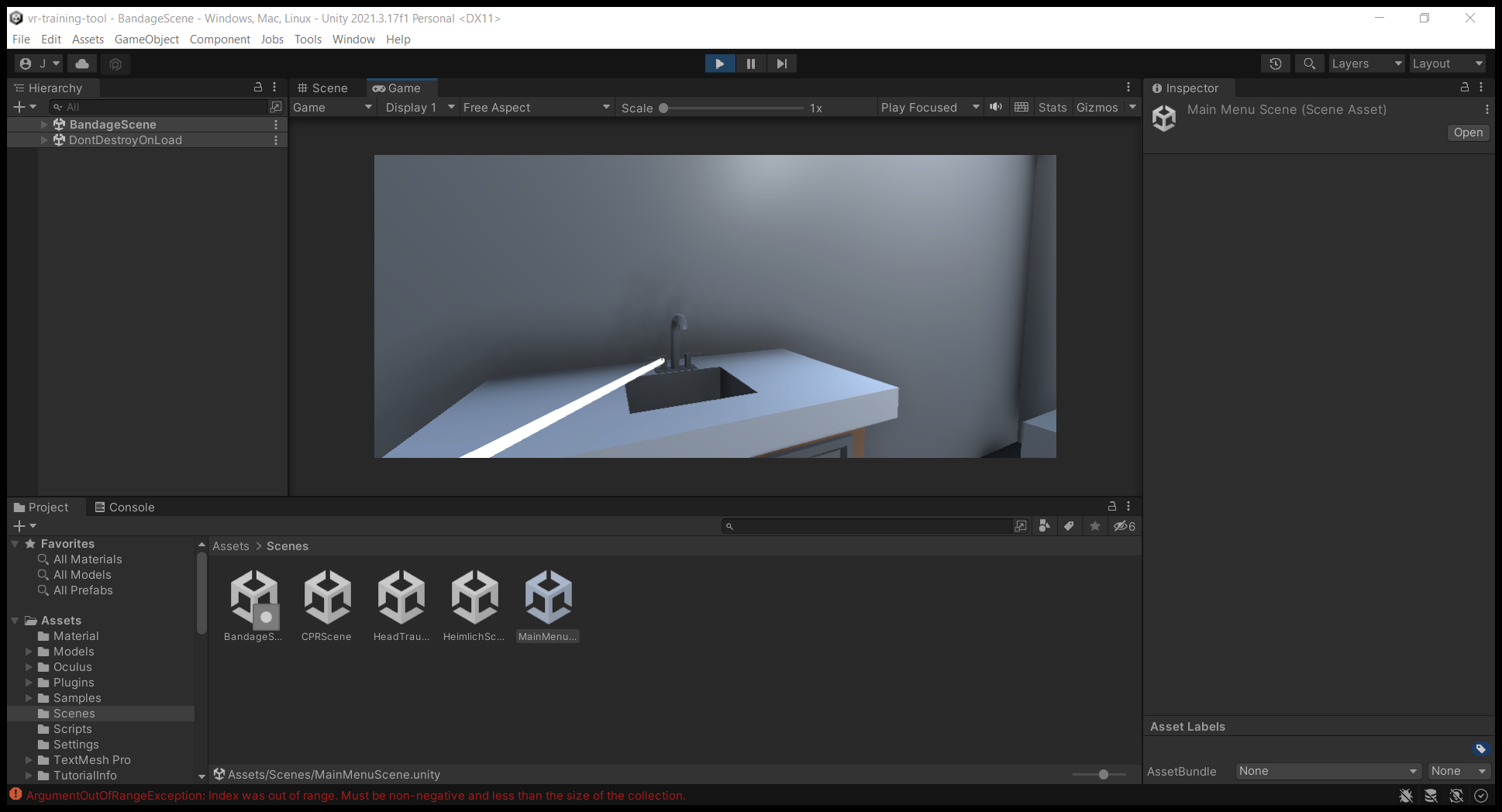


Figure 20: Interactable Sink Handle in Small Cuts and Scrapes Scenario

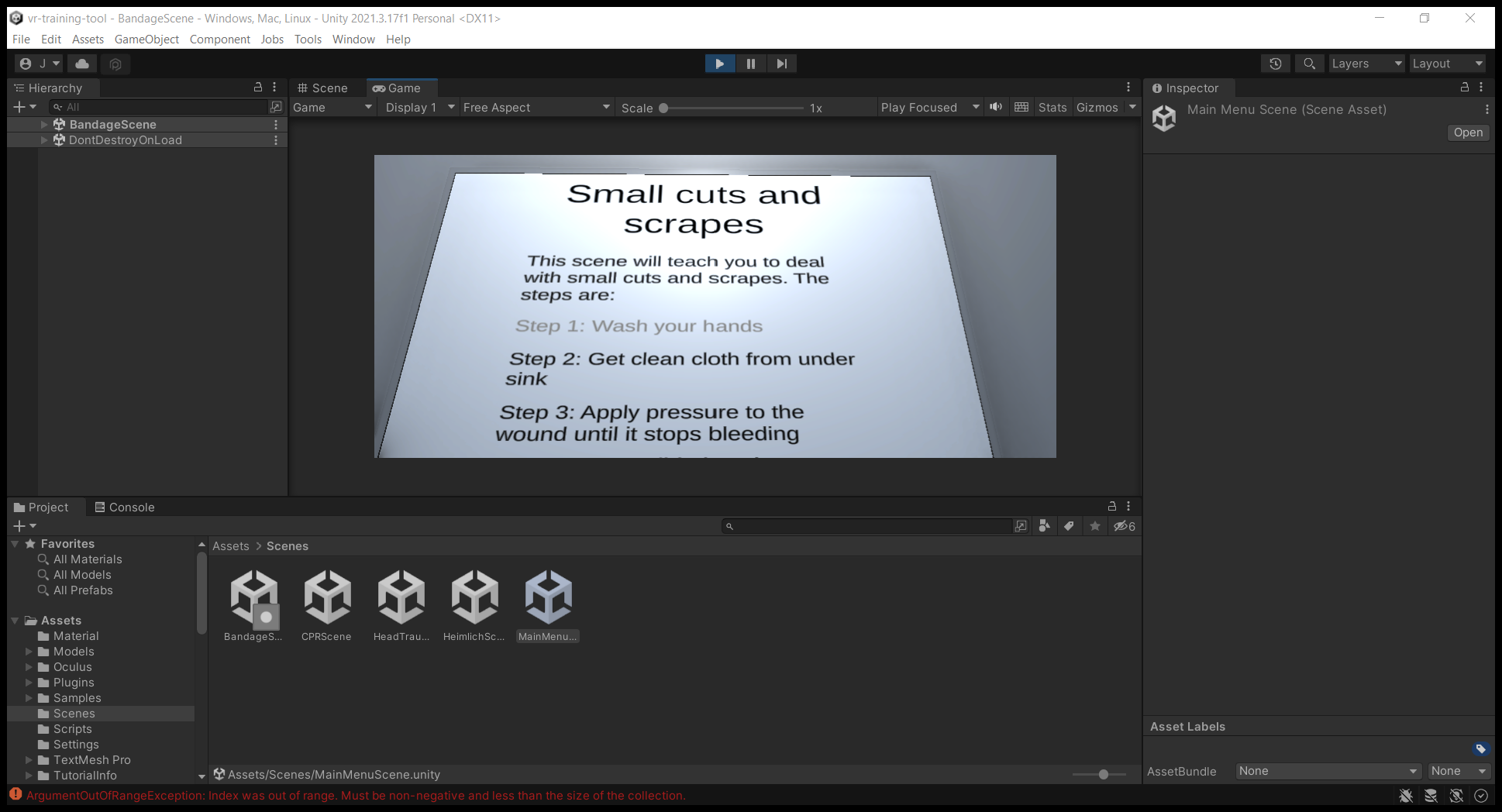


Figure 21: UI is Updated After Object Interaction

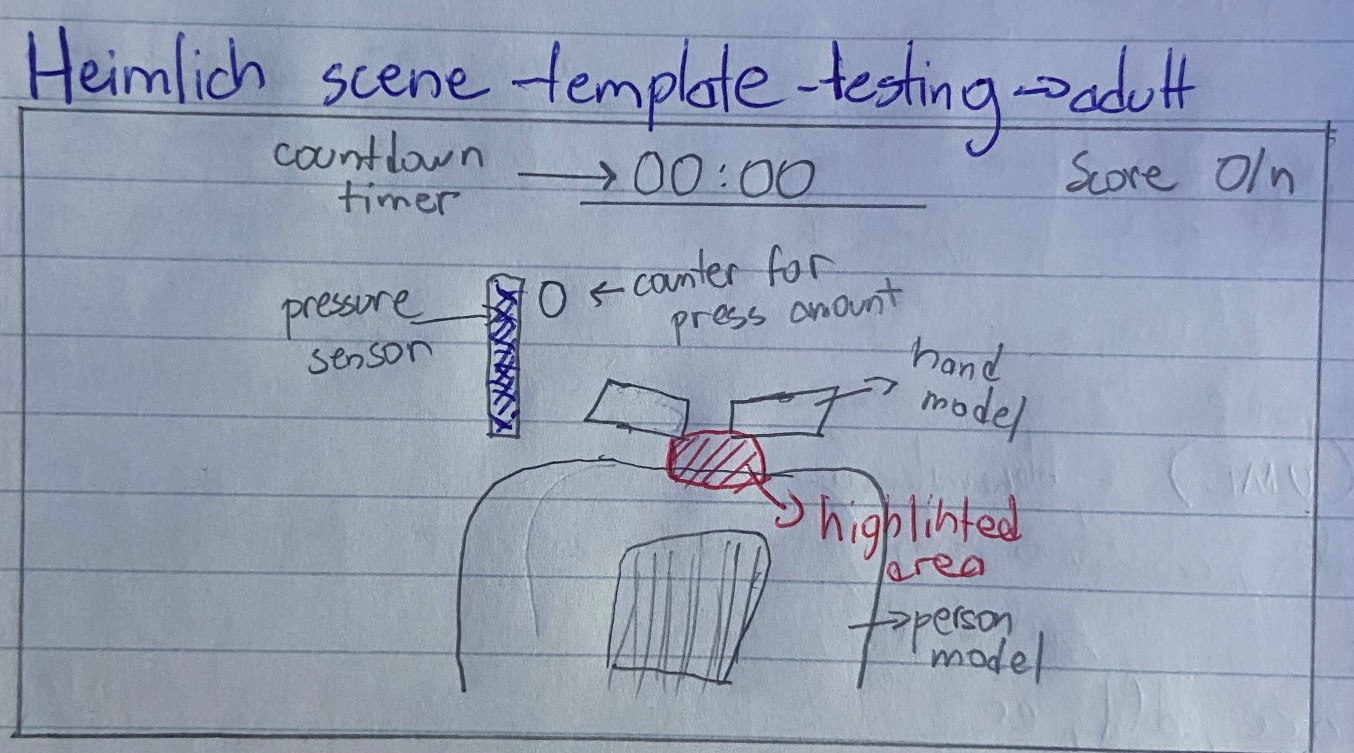


Figure 20. UI and scene design template for Heimlich Maneuver adult Testing scenario, present in the figure are the main components of the Testing procedures, which include the countdown timer and score along with interactable objects.

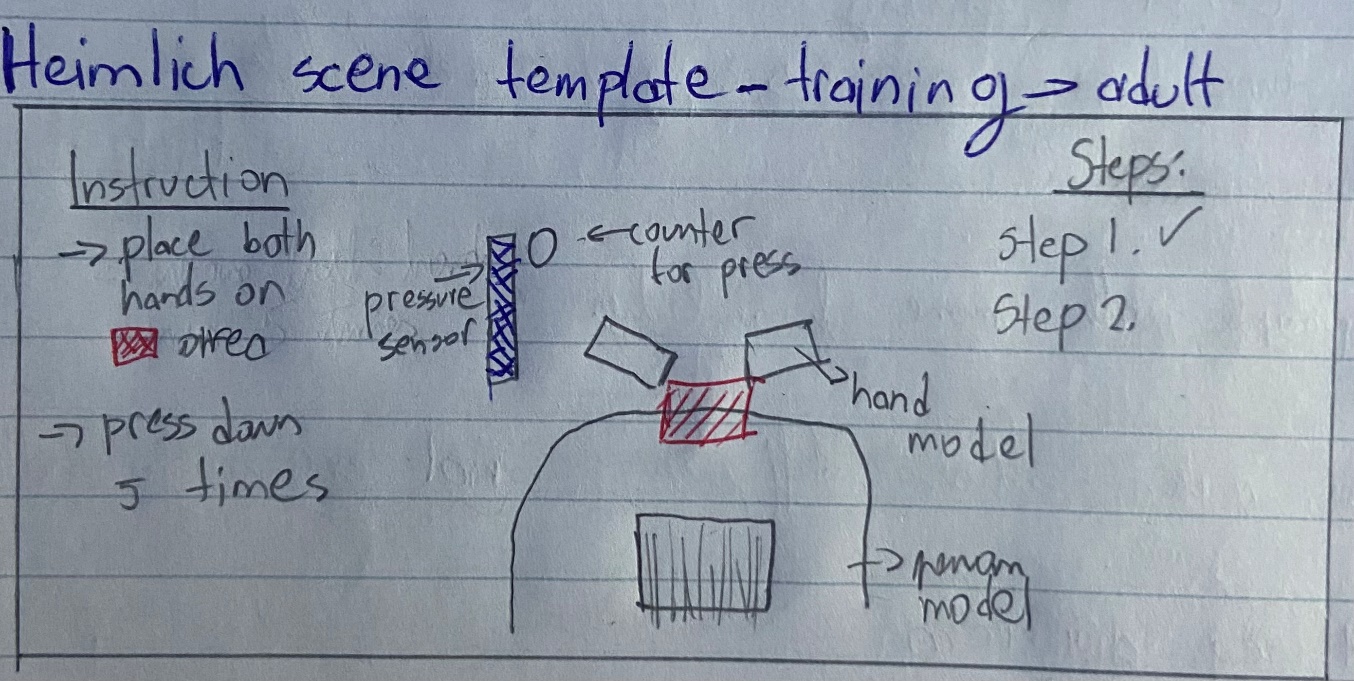


Figure 21. UI and scene design template for Heimlich Maneuver adult Training scenario, present in the figure are the main components of the Testing procedure, which include the steps marker which will be filled with events and instruction set for that specific event.

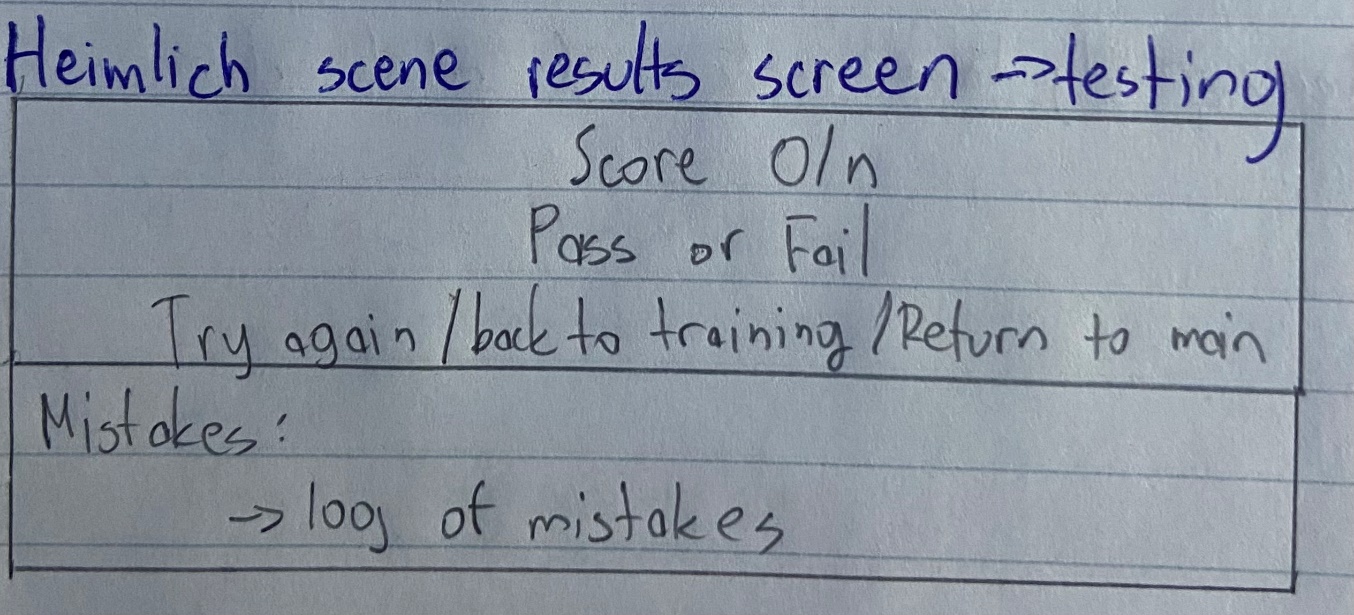


Figure 22. Heimlich Maneuver screen after test scenario is done, the components show the score calculated by the number of correct actions taken over the total amount of actions. Regarding this a pass or fail clause will be given. Then depending on if they pass or fail multiple options will be shown. At the bottom will be a list of mistakes which have occurred or if the timer has run out.

# Appendix A – References

Gordon, Whitson. “How to Reduce Motion Sickness in Virtual Reality.” *Wired*, Conde Nast, 22 Apr. 2021, https://www.wired.com/story/how-to-reduce-motion-sickness-virtual-reality/.

# Appendix B

[Whatever other extra info we need]