

Wright State University

CORE Scholar

[Browse all Theses and Dissertations](#)

[Theses and Dissertations](#)

2022

Development of Enhanced User Interaction and User Experience for Supporting Serious Role-Playing Games in a Healthcare Setting

Mark Lee Alow
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Alow, Mark Lee, "Development of Enhanced User Interaction and User Experience for Supporting Serious Role-Playing Games in a Healthcare Setting" (2022). *Browse all Theses and Dissertations*. 2657.
https://corescholar.libraries.wright.edu/etd_all/2657

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

DEVELOPMENT OF ENHANCED USER INTERACTION AND USER
EXPERIENCE FOR SUPPORTING SERIOUS ROLE-PLAYING GAMES IN A
HEALTHCARE SETTING

A Thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science

by

MARK LEE ALLOW

B.S.B., Wright State University, 2019

2022

Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

December 8, 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Mark Lee Alow ENTITLED Development of Enhanced User Interaction and User Experience for Supporting Serious Role-Playing Games in a Healthcare Setting BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Ashutosh Shivakumar, Ph.D.
Thesis Director

Thomas Wischgoll, Ph.D.
Interim Chair, Department of
Computer Science and Engineering

Committee on Final Examination:

Ashutosh Shivakumar, Ph.D.

Yong Pei, Ph.D. (Co-Advisor)

Thomas Wischgoll, Ph.D.

Paul J. Hershberger, Ph.D.

Shu Schiller, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Alow, Mark Lee. M.S., Department of Computer Science and Engineering, Wright State University, 2022. User Experience and Data Collection for Supporting Serious Role-Playing Games in a Healthcare Setting.

Education about implicit bias in clinical settings is essential for improving the quality of healthcare for underrepresented groups. Such a learning experience can be delivered in the form of a serious game simulation. WrightLIFE (Lifelike Immersion for Equity) is a project that combines two serious game simulations, with each addressing the group that faces implicit bias. These groups are individuals that identify as LGBTQIA+ and people with autism spectrum disorder (ASD). The project presents healthcare providers with a training tool that puts them in the roles of the patient and a medical specialist and immerses them in social and clinical settings. WrightLIFE games are distributed on both mobile and desktop devices and go through the entire cycle of providing healthcare professionals with experiential learning, which starts with defining the goals of the simulation and ends with collecting feedback.

In this thesis work, cross-platform software frameworks like the Unity Engine have been used to develop survey scenes to comprehensively document users' pre- and post-simulation experience and attitudes towards implicit bias. Life course scenes were designed to convey an enhanced user experience that bridges the socio-technical gap between the real and virtual worlds. By applying existing user-experience design methodologies to design the survey scenes and life course scenes, it was possible to create an immersive experiential-learning assessment tool that has the potential to deliver data-driven and targeted learning.

TABLE OF CONTENTS

1 INTRODUCTION	1
1.1 Background.....	2
1.1.1 Serious Games and Experiential Learning.....	2
1.1.2 Serious Games in Medicine	3
1.1.3 Technology-Based Experiential Learning	5
1.2 Solution.....	9
2 SYSTEM DESIGN	12
2.1 Data Collection scenes.....	12
2.1.1 Technology Stack.....	14
2.1.2 Scene Structure	16
2.1.3 Functions and Data Structures	17
2.1.4 User Interface Implementation	30
2.2 Life Course Game.....	37
2.2.1 Tools	39
2.2.2 Implementation	40
3 RESULTS	49
3.1 Data Collection Scenes	49
3.2 Life Course Game.....	54
4 DISCUSSION	57
4.1 Interpretation.....	57
4.2 Challenges.....	58
4.3 Future Work	59

5 CONCLUSION.....	61
REFERENCES	62

LIST OF FIGURES

Figure 1: Function Definition and Outer Conditional Statement	19
Figure 2: Converting UI Dropdown Options to Strings Code Snippet.....	20
Figure 3: Getting Current Time and Date Code Snippet	21
Figure 4: Validation of Question 1 Code Snippet.....	22
Figure 5: Validation of Question 2 Code Snippet.....	23
Figure 6: Validation of Question 3 Code Snippet.....	23
Figure 7: Validation of Question 4 Code Snippet.....	24
Figure 8: Validation of Question 5 Code Snippet.....	24
Figure 9: Validation of Question 6 Code Snippet.....	24
Figure 10: Validation of Question 7 Code Snippet.....	25
Figure 11: Validation of Question 8 Code Snippet.....	25
Figure 12: Question 8 Extra Text Field Code Snippet.....	26
Figure 13: Validation Modal Window Code Snippet	26
Figure 14: Submitting the Collected Information Code Snippet	27
Figure 15: AnswerData Object Example	27
Figure 16: Submission to the Database	28
Figure 17: Loading the Next Scene Code Snippet.....	28
Figure 18: Interactive Text Area Script Code Snippet.....	29
Figure 19: Interactive Text Area UI Example	30
Figure 20: UI Hierarchy Tree	31
Figure 21: Toggle Group Color States.....	31

Figure 22: Validation Modal Window Example.....	32
Figure 23: Dropdown UI Example (Left) With its Value Settings (Right)	33
Figure 24: Active Checkbox Group Example.....	33
Figure 25: Binding UI to Script in Demographic (Left), Final Assessment (Right) Scenes .	34
Figure 26: Connecting the “Next” Button to a Script Function	35
Figure 27: Testing on Popular Viewports (Left) and More Viewport Examples (Right).....	36
Figure 28: Adjusting Build Settings for a Different Platform Type	36
Figure 29: Loading the Next Scene Code Snippet.....	37
Figure 30: ASD Life course First version - Ashley Example.....	41
Figure 31: ASD Life course First version - Sarah Example	41
Figure 32: Emoji Slider and Anxiety Meter	42
Figure 33: Psychological Safety Meter.....	43
Figure 34: LGBTQIA+ Life course - Chris Character Examples.....	44
Figure 35: ASD Life course Example - Removing the Main Character from the Scene.....	45
Figure 36: ASD Life course Example - First Version, Altered Version, and Final Version .	45
Figure 37: AutoFlip Script	46
Figure 38: LGBTQIA+ Life course - Photo Album Format Example.....	47
Figure 39: ASD Life Course - Photo Album Format Example	47
Figure 40: ASD Life Course - Cover	48
Figure 41: LGBTQIA+/ASD Demographic Scene: Introduction.....	49
Figure 42: LGBTQIA+/ASD Demographic Scene: Questionnaire	50
Figure 43: LGBTQIA+/ASD First/Second Assessment Scene Example	51
Figure 44: LGBTQIA+/ASD Final Assessment Scene (1).....	52

Figure 45: LGBTQIA+/ASD Final Assessment Scene (2)	53
Figure 46: ASD Life course - Age 7	54
Figure 47: LGBTQIA+ Life course - Age 11	54
Figure 48: ASD Life course - Age 11	55
Figure 49: LGBTQIA+ Life course - Age 25	55
Figure 50: ASD Life course - Age 18	55
Figure 51: LGBTQIA+ Life course - Age 47	56
Figure 52: Life course Game Unity UI and Page Flip Animation	56

ACKNOWLEDGMENT

I want to express my sincerest gratitude to my academic advisor, Dr. Ashutosh Shivakumar, and to my co-advisor Dr. Yong Pei. Dr. Shivakumar's and Dr. Pei's encouragement and expertise were essential for me to complete my degree. Their meticulous supervision of the "Wright LIFE (Lifelike Immersion for Equity)" project, which I was a part of, made it possible for me to conceptualize and develop my thesis work.

I would like to thank Dr. Thomas Wischgoll and Dr. Paul J. Hershberger, who graciously agreed to be on the thesis committee. They were providing me and other team members with the essential expertise that made the successful implementation of the project possible. I also want to express my best regards to all the team members that I have worked with throughout the project. Everyone on the team showed their utmost support in the tasks that we needed to complete.

I want to thank the Ohio Department of Medicaid and Ohio Colleges of Medicine Government Resource Center for funding my thesis research as a part of The Medicaid Equity Simulation (MES) Project and the Department of Computer Science and Engineering at Wright State University for supporting me as a student during my time in the Computer Science degree program.

Last but not least, I want to express my endless appreciation to my mother, Elena Johnson, and my stepfather Bruce Johnson for always being there for me, and persistently supporting me in any hardships that I had to face.

1 INTRODUCTION

This thesis focuses on the aspects of Serious Role-Playing Games for Healthcare Applications, and their use in eliminating bias. The thesis will reflect on how a quality user experience, created through the implementation of design and front-end development skills, can significantly enhance the results of learning. That user experience will be integrated into a cross-platform Serious Game application, intended for healthcare providers. This thesis will find a solution to seeking an effective way of creating and presenting content to these providers, on best ways of gathering their feedback, and on how that feedback will be used in the future. Important concepts and aspects of User Interface (UI) design, User Experience (UX) design, and Front-End development will be utilized while creating that experience.

Implicit bias in clinical settings is a problem with severe effects on patients, especially on individuals belonging to underrepresented communities. Experiential learning offers a way to enhance the skill-based training of different occupations, including medical professions. This thesis attempts to combine the advantages of experiential learning with modern-day cross-platform technologies, such as the Unity Engine, to develop an application that can transfer the living experiences of people from underrepresented communities. Such depictions of hardships, particularly within the clinical setting, are shown to medical providers to create a better understanding of the issues that the affected individuals have to go through. After being immersed in a such simulation, healthcare professionals will have a better understanding of the importance of implicit bias. This immersion allows for creating enhanced learning experiences, formally defined under the term “Experiential Learning”.

1.1 Background

1.1.1 Serious Games and Experiential Learning

Serious games are applications that combine active learning capabilities with the creative elements of conventional video games. They do not have entertainment aspects but focus on delivering information and educational value to the user that utilizes them. The player is placed in a virtual environment with a specific goal and is then motivated to make decisions to progress through the game. Serious games are used to provide skills-based training in the areas such as education, well-being, advertisement, corporate training, and healthcare. This allows for creating enhanced learning experiences, formally defined under the term “Experiential Learning”. Experiential Learning is defined as a “philosophy and methodology in which educators purposefully engage with students in direct experience and focused reflection in order to increase knowledge, develop skills, and clarify values” [1]. Such learning implies understanding information through actions, or “learning by doing”, which is what serious games are focusing on [1]. Modern serious games originated back in the 1970s and have since then found many practical applications [3]. There are many examples of Serious Games that made a significant contribution to a certain industry. One such example could be Flight Simulators. Flight simulator serious games now form the backbone of airline training and have made a major contribution to aviation safety [4]. Without such simulators, pilots would have to risk their lives every time they need to practice [4]. There exist serious games that help people with certain health conditions. Some serious games are used as a nonpharmacological intervention tool for improving cognitive skills [2]. Such games are trying to focus on observing players’ behavioral symptoms by conducting certain in-game interactions. Serious games can be a mental stimulant that improves brain health and memory [2].

1.1.2 Serious Games in Medicine

In healthcare, serious games help train doctors and medical professionals. They can teach specific practical skills, related to medical treatment or surgeries, and more comprehensive skills, such as aspects of interactions between healthcare professionals and clients [5]. Special variations of serious games found application in the fields of surgery, first aid, anesthesia, and cardiopulmonary resuscitation [6]. Serious games can improve medical students' skills in decision-making, cooperation, and practical operation ability [6]. Moreover, with the precedent of the COVID pandemic, serious games find use in telemedicine and distance education. The approach of remote learning helps avoid contagion, at the same time monitoring indications of rehabilitation and education [6].

Bias is a major issue in the healthcare setting. Bias is defined as a negative evaluation of members from one group towards another group [7]. Explicit bias, which usually constitutes open disdain towards groups, is rare nowadays, and the person who is biased is always aware of it. On the other hand, there exists an implicit bias, that a person is unaware of. An implicit bias is a type of bias that “does not require the perceiver to endorse it or devote attention to its expression” and “can be activated quickly and unknowingly by situational cues” [7]. Implicit bias may make a person think that they treat everyone equally, but such a person has subconscious negative associations about certain people. It is problematic to understand implicit bias since it cannot be knowingly expressed. Numerous methods were developed to measure such bias, with most of them focusing on situational cues, such as evaluating reactions of the participant to other person's skin color and accent [7]. In the healthcare field, implicit bias exists among all professions [7]. In a journal article, a correlation was found between bias and poor service in hospitals and pharmacies

[8]. Results of Implicit Association Tests (IATs) showed a widespread bias among the entire healthcare industry, with a particular emphasis on stronger negative perceptions and more hostile demeanor towards people of Latino or African American descent, especially if they are of a lower socio-economic status [8]. When a healthcare provider expresses implicit bias towards minority individuals, it leads to a lack of trust and commitment from the patient [7]. Moreover, a patient can then express their own implicit bias during the encounter, which even further reduces the effectiveness of medical treatment [7].

Deleterious effects of implicit bias in healthcare are felt by individuals who identify with the LGBTQIA+ community [13] and individuals with autism spectrum disorder (ASD) [9]. In a healthcare setting, medical stigma is a critical concern for LGBTQIA+ individuals [13]. LGBTQIA+ people report encountering bias towards their sexual and gender identities coming from their healthcare providers. This issue is persistent around the United States, especially in the southern states [15]. Moreover, experiences related to homophobia have been connected to substance abuse, suicide attempts, and increased anxiety in the LGBTQIA+ society [15]. Such exposure also interferes with LGBTQIA+ individuals' ability to create and maintain long-term relationships [16]. LGBTQIA+ group also faces a significantly higher risk of being HIV-positive [16]. In the healthcare field, there exists a significant problem for LGBTQIA+ individuals to seek appropriate mental health treatment [16]. Due to the bias, presented in the form of negative stereotypes about LGBTQIA+ individuals, such patients face significant issues in physical and mental health outcomes [14]. Compared to heterosexual patients, LGBTQIA+ patients have higher rates of obesity, asthma, and some types of cancer [14]. Bias and discrimination coming from the providers are identified as significant contributing factors to the lesser utilization of healthcare by LGBTQIA+ individuals [14].

Autism is a neurodevelopmental disorder where individuals are characterized by profound difficulties in communicating and perceiving social cues [9]. Many people with autism are stereotyped as socially awkward and are often faced with negative attitudes and significant hardships [10]. Adults with ASD reported issues with job search, lower quality of healthcare, and significant health issues that include anxiety and meltdowns. Moreover, autistic people are likely to have shorter than average lifespans and major chronic health conditions. Adults with ASD receive poorer healthcare, which also comes with higher expenditures compared to non-autistic adults [11]. They also face a higher likelihood of being hospitalized, having to go to therapy visits, and being prescribed drugs. Autistic patients reported difficulties communicating with their care providers. Furthermore, many healthcare professionals mentioned that they are unaware of how to interact with autistic patients and expressed a desire for information and training [12].

1.1.3 Technology-Based Experiential Learning

To address the implicit bias of healthcare providers toward LGBTQIA+ and ASD communities, a serious game called “Wright LIFE (Lifelike Immersion for Equity) Project”, has been developed by utilizing a cross-platform software framework called Unity and distributed on popular mobile computing platforms like iOS and Android. This project was developed in collaboration with the Ohio Department of Medicaid. WrightLIFE attempts to transfer the life experiences of individuals from the LGBTQIA+ and ASD communities to the healthcare provider, the player of the game. This process is done by mimicking real-life scenarios involving two individuals Charles, a 60-year-old African American belonging to the LGBTQIA+ community, and Ashley, an 18-year-old female with ASD. In the game, these individuals are attending a

doctor's appointment to address their health situations. The player (the provider) is required to make choices to maximize psychological safety while progressing through the depicted events.

One of the paramount tasks in WrightLIFE was creating the experience for the healthcare provider that will immerse them into the application's educational process and will convey the information effectively and interactively. This thesis will focus on the process of creating an immersive user experience inside the WrightLIFE simulations. User Experience (abbreviated as UX) defines the perceptions and feelings of the user when they are interfacing with a certain system. This system can include any type of interaction between a human and a device, which is usually controlled by a user interface (abbreviated as UI). Developing a quality user experience involves multiple steps. First, there is a need to gather the information about personas that will use the app. After that, the appropriate design of the application must be done to properly address the problem relevant to these personas. Then, the initial prototype of the app design should be created and later reviewed. Creating a prototype can involve building user flow diagrams, wireframes of pages and scenes, and sketched implementations of provided scenarios. After the prototype is reviewed, it needs to go through adjustment and testing iterations. After continuous testing, additional elements such as proper styling should be added to the application. It will eventually become a product that can be released to the intended audience. To gather appropriate feedback, the product should have relevant interactive functionality that allows for collecting responses from the users [17].

The correct use of UX can drastically improve software products' marketability and is considered one of the most important aspects of the software development process [18]. A good user experience must decrease the complexity of software, at the same time providing an enjoyable environment for the user. Properly implemented UX design has found its applications in many

areas. One such example can be designing user interfaces for older users. Due to older adults' lack of prior experience with user interfaces in their younger years, they must be properly accommodated with interfaces that have clear and simple navigation [18]. A correctly designed interface can significantly increase the outreach of the application to all age groups. Another age group that needs UI designed specifically for them are children. In the applications that are targeting the youngest age group, it is important to have an interface that has highly visual menus and icons, which presents all information at once, rather than requiring children to scroll through the information, and that has easily identifiable elements, such as large clickable buttons with easy-to-read text labels [18]. Designing UI for people with physical or mental limitations also requires special attention. The correct use of colors, avoiding using similar colors next to each other, using large legible fonts, adding speech recognition elements, and avoiding features that can cause stress is essential to keep that population group interested in the application [18].

One example of a meaningful UX implemented in a Serious Game setting can be found in the "S-Cube" game [19]. This game is focused on developing the soft skills of social enterprise employees. It urges players to use their avatars to communicate and secure sponsorship pledges. Players can control their avatar's gestures, emotions, and body movements. All these actions are performed by pressing buttons in the UI. This game's development introduces important aspects of the gaming experience that should be noted in the development of a quality UX. The game needs to present a player with a "challenge", that adds certain gameplay elements, which focus on increasing player immersion in the simulation [19]. Such a challenge allows for more effective learning. Another element that the game needs to focus on is "competence", which assures that the player can perform game tasks successfully, without facing any issues that may divert them from completing the game. Another important part of a good gaming experience is "flow", which

defines how well the experience of the game will be absorbed by the player [19]. Flow directly corresponds with the learning outcomes that the player will receive. However, before the flow is implemented, “immersion” comes into place. Immersion, which is a precondition to the “flow”, defines characteristics that make the game enjoyable to the player [19]. Such enjoyment will ensure the memorability of the game’s content. However, too much immersion can be harmful to achieving correct learning outcomes, so the level of detail in the scenes needs to be kept from being too overwhelming. Another important feature that needs to be present in the game is “affect” [19]. Affect covers emotions that the player will feel during the gaming process. Such emotions should focus on enhancing the player's experience and invoking detrimental emotions should be avoided. A feature that comes out from “affect” but is more difficult to control within the game’s development is “tension” [19]. Tension can lead to strong emotional responses, such as excitement, but can also lead to a strong negative response, such as anger or frustration. The level of tension in the game needs to be carefully controlled to avoid detracting users from the gaming experience [19].

The characteristics that connect the gaming experience with the learning experience are also essential to keep track of. Four noteworthy aspects of an effective and clear learning experience with the Serious Game can be highlighted. They are “adaptivity”, “usability”, “extensibility”, and “fidelity”. Adaptivity suggests that the game should be correctly adjusted towards the goals and needs of the intended player audience, such as presenting a player with hints and tips about the presented content. Usability implies that all users of the game should have a positive gaming experience, which comes from the ease of use of the interface and user controls. Extensibility evaluates if the elements of the game sufficiently support the achievement of defined

learning goals. Fidelity implies that there should be a prominent level of realism in the gaming environment to present the player with the best possible learning experience [19].

UX Design can use context-aware technologies to create a personalized experience for a specific group of users. Such experiences can be defined by either explicit or implicit human-computer interaction. This UX approach is known as Contextual Design. Explicit interaction implies direct manipulation of the UI, while implicit interaction learns to understand users' preferences and presents the best experience based on acquired information [20]. For the WrightLIFE project, it is essential to understand what stages of interaction the user needs to go through to get the best learning experience out of the application. The purpose of good UX is to find the most effective way to employ the user's perceptual and cognitive abilities and to receive the most accurate feedback. When the healthcare provider starts the WrightLIFE application, they need to have a goal in mind, which is education about implicit bias, and analysis of how to best cope with it [21]. After the goal is set, the user needs to be presented with the intention of the app - or with some UI that describes the app's features. Later, the provider should translate that intention into the functions implemented in the interface [21]. After that translation, the user must execute the actions specified by the app using appropriate input devices [21]. Once the action is executed, the app should then be presented with the confirmation that its actions gave a certain result [21]. After completing all actions, they need to be evaluated, and it should generate insight into the received experience in the mind of the user [21]. In the end, the interpretation of the outcome should be done by the user, and the final feedback should be sent by the user [21].

1.2 Solution

The WrightLIFE project attempts to enhance awareness of implicit bias toward LGBTQIA+ and ASD individuals among nurses, doctors, and healthcare practitioners. The goal of the WrightLIFE project was to deliver a practical solution to educating healthcare providers about implicit bias. The WrightLIFE simulation attempts to minimize the impact of implicit bias by utilizing its features and functionality. The interactive elements of the game attempt to achieve two significant outcomes: bringing the user experience as close as possible to a real-life scenario and receiving real feedback from the provider for further analysis. Both simulations don't just show the doctor's visits but present a wider picture that details all events that happened to the characters during the day of the appointment, starting with their commute to the clinic, and ending with their feelings after the appointment was concluded. Each step of the simulation is conducted from a first-person perspective, shows each patient's specific concerns and insecurities, and depicts their reflections on the events that occur to them.

The point of the serious experiential game is to transfer experience from the game to a player. User experience is a defining tool in such a transfer. The personas of the WrightLIFE project were defined as LGBTQIA+ and ASD individuals, and the scripts for the appropriate characters were developed by the medical team. While the animated scenes developed by other team members represented the characters themselves, there was a need to develop scenes that will immerse the player deeper into the characters' experiences and will collect the appropriate feedback during the playthrough and at the end of the simulations. To achieve immersion and feedback collection, it was decided to create the life course game and data collection Scenes. The life course game was a part of the simulation that intended to cover the story of how the simulations' main characters struggled through societal rejection and went through hardships throughout their entire life. There was a need to build a visual representation of the scenes of

rejection, neglect, and anxiety that were commonplace in the lives of Ashley and Charles. Additional characters of Sarah, who was Ashley's friend, and Chris, who was a brother of Charles, were proposed to create even more impact by juxtaposition to the primary characters. The solution was to create a striking and memorable visual representation of the past events described in the script and to make appropriate interactive UI elements that might add to the immersion.

The data collection scenes are intended to collect information about the provider who will play through the simulations, and then their respective feedback at different stages of simulations. From the UX standpoint, such scenes needed to be developed in a clear way that will be extensible (will support learning goals) and adaptable (correctly adjusted to the goals of the game). The data collection scenes needed to include validation functionality that will ensure the correctness of gathered information, which is essential for further research. Four separate data collection scenes were deemed as a good solution for feedback gathering. These scenes were planned to be present at four different stages of each simulation: the demographic survey in the beginning, and assessment questionnaires before the hospital visit, after the visit, and at the end of the entire game. The data collection scenes were intended to gather the demographic information of the player and their perceptions of the experiences received during the simulations' playthrough. The combination of the life course game and data collection scenes was intended to be a great tool to deliver the best possible user experience in the application.

2 SYSTEM DESIGN

With the scripts for the life course game and data collection scenes being developed by professionals from the medical team, the concept of validation and visual representation had to be developed through a strenuous iterative process. Such a process included many rounds of reviews, and in the end, resulted in the creation of a quality user experience. During the implementation, it was important to follow the best practices of user experience design. Moreover, with the front-end development of the data collection scenes, it was essential to keep the codebase organized and to track and correct all possible errors.

2.1 Data Collection Scenes

The goal of all data collection scenes was to effectively accumulate reflections of the players' experience during their playthrough of the serious game. Such scenes had to be clear and interactive and had to include the functionality to validate the user's input. Each question was presented to the player in the form of a text box with the question in it. The data was gathered from the player using interactive polling elements such as checkboxes, text input fields, and dropdown selection game objects. The player then confirmed the completion of filling out all scene elements by pressing the "Next" button at the bottom of the UI's layout. Each data collection scene had an underlying script that was serving as a basis for the UI. The scripts contained functions and data structures that were defining the validation functionality and the visual style of the scenes and later collected the validated information from each question into an object that was submitted to the database.

The demographic scene, being the first in the entire simulation, starts by providing the player with information about the serious game, and then gathers their consent, which confirms that the player agrees with the game's guidelines. After consent is given, the scene collects information about the player's location, the sphere of practice, experience, gender, race, and ethnicity. The demographic questions were presented as a series of text labels with interactive input elements located on their right side. The scene presented eight questions to the user, and the following types of interactive input: free-form field for question (1), multiple-selection checkbox groups for questions (3), (8), drop-down lists for questions (2), (5), (6), (7), and a single-selection checkbox group for the question (4). Additionally, questions (2), (3), (6), and (8) presented the player with appearing input fields if certain options were selected.

The assessment surveys consist of three questionnaires that aim to gather subjective perceptions of the player. The check marks in the surveys represent a range, with the description of the range being added on the first and last checkmark text boxes. The first two surveys are identical and consist of eleven questions. The first survey is displayed before the character's visit to the hospital, while the second survey comes after that visit. The surveys ask the participant about the amount of anxiety, frustration, and compassion they felt towards the patient depicted in the game. They then ask the player about their opinion on how they would have behaved if the patient in the simulation was a real person and if they had a real-life appointment with them. The surveys then proceed into questioning the player's attitudes towards prejudice and bias and ask if they have any biases themselves. The further questions ask the healthcare professionals about their experiences with ASD / LGBTQIA+ patients. The final assessment survey appeared at the end of the simulations and contained five additional questions. One of them had a conditional requirement of fetching user input in case certain options were selected. The player was required to read the

learning objectives above the question, and the question itself asked if these learning objectives were met in the simulation. The answer field, as in other questions, contained a range of five checkboxes that went from “Strongly Agree” to “Strongly Disagree”. If the player selected checkboxes closest to the “Strongly Disagree” side, this would have triggered the appearing container that had a free-fill text field. The layout below was moved accordingly.

2.1.1 Technology Stack

A technology stack is a description of languages and frameworks that are used together to build an application. Defining a stack has become essential for the proper implementation of any software project [22]. The stack used in creating WrightLIFE’s data collection scenes included the Unity3D game engine, a C-Sharp (C#) programming language that was implemented into scripts using the Visual Studio Code editor, and Git for version control.

1. Unity3D Engine

Unity is a game engine that allows for the development of game applications for a variety of platforms and devices [23]. Unity was the most crucial tool in the entire project. It allowed the creation of all the scenes for the serious game and then later allowed combining them into the builds, with testing these builds on desktop and mobile platforms. The Unity Engine primarily comprised two areas - the 2D and 3D asset editor, and the script editor. All the assets in the engine are defined in the script, and then their subsequent behavior is controlled using the functions and structures of the script. After the changes are made in the script or in the asset editor panel, the engine needs to compile them into a preliminary game preview and then be allowed to play through the resulting game application. Unity also shows all the warnings and errors that occurred during

the compilation process. If the error is critical, the compilation will not finish. After the appropriate edits are made and the application compiles without errors, it is important to commit the changes into a snapshot, which is later saved on a cloud repository. The game entities are separated into scenes, with each scene defining a self-contained entity of the application.

2. Visual Studio Code

Visual Studio Code is a multi-platform code editor that can support a wide variety of programming languages that are either built into it (JavaScript, TypeScript, Node.js), or added into it through extensions, which was the case for the WrightLIFE project's C# (C-Sharp) [24]. Visual Studio Code allows writing and editing C# scripts and provides helpful features, such as syntax highlighting and error analysis. It also allows for a variety of supplementary features that can be added through extensions, such as language shortcuts or direct interaction with Git.

3. Git

Git is an open-source software that provides a version control system for tracking changes in files, with later coordinating these changes among all team members [25]. The Git repositories for this project were hosted on a popular Git hosting - GitHub. Using Git, it is possible to gather changes made to the Unity application and commit them into a snapshot. This "commit" is first done locally, and then can be "pushed" into a remote cloud repository on GitHub. The changes can be made asynchronously by different team members, and later combined in the cloud using the feature called a "Pull request". During such requests, the conflicting changes are reviewed, and the final version of the changes is then pushed into the branch of a repository. One repository can have many branches, and they usually constitute different variants of the same project. The "Main"

branch can be responsible for a stable release-ready version of the project, while the “Feature” branches can be used to make changes to the application without directly passing errors into the stable version. These branches are then either merged into the “Main” branch or amended in case errors are found. Any secondary branch can also be discarded without the fears of deleting the stable version of a project. The stable version from the “Main” branch can then be copied into a new “Feature” branch and used to create additional changes.

2.1.2 Scene Structure

All data collection scenes have similar script functionality, with some notable differences. The script functionality of all scenes can be generalized to two large areas - Validation and Data Submission. These areas will be discussed in detail, along with providing examples of functions and data structures from the scripts. After the scripts’ overview, the process of UI creation and connection will be presented.

1. Validation

Form validation is usually associated with web-based resources to prevent the leak of sensitive information. In fact, the invalidated input is in the list of the top ten critical vulnerabilities of web resources [26]. In the case of serious games, the form validation’s function is not providing security but ensuring that the user will provide data that can be valuable for future research. Such data should be correctly entered in the input fields, must be legible, and needs to be verifiable. The validation was implemented using a large script structure that contained separate functions to validate each of the eight questions separately. These functions were invoked by the umbrella script that was called on the “Next” button click. In case any of the questions had input errors that

could constitute not selecting any options or having the wrong type of input, the “onClick” script triggered the modal window to appear and passed the error message from the validation code block of the corresponding question.

2. Data Submission

After the data was validated, it needed to be received by the application to further forward it to medical professionals that can later conduct appropriate research. In the WrightLIFE project, the data from the surveys were collected into a database. The Wright State’s Computer Science Department provided the WrightLIFE project with a dedicated server, which was defined by the “WrightServer” object that posted the information into the University’s internal database. The user feedback collected using the surveys was put into the resulting spreadsheet and was later used by other team members to visualize the data. Then, appropriate software, called Tableau, was used to create a visual representation of participants’ responses. Wright State’s database was utilizing the REST API to implement its functionality. REST API is an architectural style for manipulating information that utilizes HTTPS requests [27]. It is widely used in software development, especially in web development. In the case of WrightLIFE, the database was stored in a cloud, so using that API was reasonable. REST API uses GET, PUT, POST, and DELETE request structures to refer to appropriate operations [27]. In the case of WrightLIFE’s script, it primarily focuses on making a POST request into the database. This request takes the data object’s fields and then sends them to the Wright State University’s database.

2.1.3 Functions and Data Structures

WrightLIFE serious game contains many scenes and associated scripts that are controlling the scenes' behavior. The goal of this thesis is to observe the way of data collection and data submission in the WrightLIFE project. The demographic script contains the most functionality and will be used as a primary example of functions that are used for data validation and data submission.

Class and variable declaration - The script is created inside of the demographic scene's class and inherits the "MonoBehaviour" class. It is a base class of the Unity Engine, from which every other class derives [28]. It is responsible for the correct functioning of scripts and Unity UI. The variables that will be later used in the script are declared in the outer scope of the class. These variables will define objects that will later be presented as components in the Unity UI. The declared variable types will be described in the order they appear in the script. A "GameObject" type can represent anything that can exist within the scene [29]. A "Dictionary" type is a data structure that stores information in a key-value pair syntax. The values should usually represent the same type. For instance, a key will be an integer, while the value will be a string. In any dictionary, every new key always needs to be unique. [30]. A "Dropdown" is a Unity class that allows the creation of an object with a list of multiple options in it. The user can then choose one out of those options, and it can be further processed [31]. A "Toggle" Unity class defines a component that is represented by a checkbox that can be switched on and off by the player [32]. A "Toggle Group" - declared as "Toggle[]" - is a UI controller used to modify the behavior of a set of Toggles. It adds a constraint that does not allow selecting more than one toggle checkbox at a time in the defined group [33]. A Text class provides the user with a non-interactive text label [34]. However, in WrightLIFE games, the Text objects are used to collect certain input from the user. Custom "onValueChanged" functions are used to implement such functionality. In other

applications, the “InputField” object can be used instead to collect such input. A “DateTime” is a C# class that can instantiate an object that will contain information about a certain date or time. Such objects will have many methods available to manipulate and print dates and times in different ways [35]. Lastly, all counters used in the scripts are declared as integers (“int”).

1. Validation functionality

To proceed from the introduction panel into the demographic scene’s validation panel, the user is required to accept the terms and conditions of the application defined in the text area. That is done by checking the box below the introduction part of the scene. The scene then allows proceeding further down by activating the “demographicInfo” game object. The scene’s validation was implemented using a large parent function “onNextButtonClick()”, which included many function-like structures that checked the validation of questions (1) through (8). The validation only started when the demographic scene panel was set as active, after the user’s consent was received through the checkbox input, which set the “isConsentGiven” boolean variable as “true”, as shown in the Figure (1).

```
public void onNextButtonClick()
{
    if (isConsentGiven)
    {
        ...
        demographicInfo.SetActive(true);
        ...
    }
}
```

Figure 1: Function Definition and Outer Conditional Statement

Inside the “onNextButtonClick()” validation function, multiple variables were defined, such as counters for toggles, the boolean validation status checked, and the “answerData”

dictionary object to collect the validated information. Further on, it was necessary to receive values from the dropdown elements and convert them into strings for the proper submission to the database. As shown in the Figure (2), this is done by creating a list of values from the dropdown menu and then selecting the appropriate option from that list, which is then converted to a string.

```
professionStr = professionDropdown.options.Select(option =>
option.text).ToList()[professionDropdown.value];
hispanicStr = hispanicDropdown.options.Select(option =>
option.text).ToList()[hispanicDropdown.value];
experienceStr = experienceDropdown.options.Select(option =>
option.text).ToList()[experienceDropdown.value];
genderStr = genderDropdown.options.Select(option =>
option.text).ToList()[genderDropdown.value];
otherPacificStr =
otherPacificToggle.GetComponentInChildren<Text>().text;
otherAsianStr = otherAsianToggle.GetComponentInChildren<Text>().text;
someOtherStr = someOtherToggle.GetComponentInChildren<Text>().text;
```

Figure 2: Converting UI Dropdown Options to Strings Code Snippet

The first information that is required to be received by the database is the date and time of the information's submission. The date and time are generated using the "TimeZoneInfo" class that allows fetching the current time and date in the EST time zone. This information is then added to the "answerData" object. The try-catch block is handling the exceptions and will trigger the error in case the time zone is not found. Such functionality is shown in the Figure (3)

```
try {
    currentTimeEST =
    TimeZoneInfo.ConvertTimeFromUtc(DateTime.UtcNow,
    TimeZoneInfo.FindSystemTimeZoneById("Eastern Standard Time"));
    currentDateEST = currentTimeEST.ToString("MM/dd/yyyy");
    currentTimeEST = currentTimeEST.ToString("hh:mm:ss tt");
    answerData.Add("Date (MM/dd/yyyy)", currentDateEST);
    answerData.Add("Time (hh:mm:ss)", currentTimeEST); }
catch (TimeZoneNotFoundException e)
{
    Debug.Log("Time zone not found: " + e);
}
```

Figure 3: Getting Current Time and Date Code Snippet

After fetching the time and date, the simulation's script checks the first question. The question asks the user to enter their zip code. This validation block checks the correctness of the zip by utilizing the "IsUSZipCode" built-in class. If the zip code is incorrect, the error message prompt is sent to the validation window. The prompt asks the user to enter the zip code which is exactly five numbers. When the correct input is received, the validation script increases the number of correctly answered questions ("totalNumOfQuesAnswered++") and proceeds. This variable's incrementing is done in every other validation block in the case of a correct input. That function is presented in the Figure (4).

```

if(isValidationDone){
    if(!zipText.transform.Find("Text").
GetComponent<Text>().text.Trim().Equals(""))
    && Globals.IsUSZipCode(zipText.transform.Find("Text")
.GetComponent<Text>().text))
    {
        answerData.Add("zipText",
zipText.transform.Find("Text").GetComponent<Text>().text);
        totalNumOfQuesAnswered++;
    }
    else
    {
        zipErrorMsg += "(1) The zip code has to be exactly 5 numbers.\n";
        unansweredQuestionMessagePrompt();
    }
}

```

Figure 4: Validation of Question 1 Code Snippet

The second question, shown in the Figure (5), checks the correctness of the profession dropdown. In case the user chooses “Physician” or “Other” as their profession from the dropdown options, a mandatory text field appears and needs to be filled in by the user. In the field, the user must specify the exact profession that they have. The validator then finds the text value of the selected dropdown option by utilizing the “GetComponent()” method with the generic type definition of “<Text>”. This information is then submitted to the “answerData” object along with the chosen dropdown value that is converted to a string.

```

        if (!professionStr.Equals("Other") &&
            !professionStr.Equals("Physician"))
        {
            answerData.Add("profession", professionStr);
            totalNumOfQuesAnswered++;
        }
        else if (professionStr.Equals("Other") &&
            !professionOtherText.transform.Find("DescriptionText")
                .GetComponent<Text>().text.Trim().Equals(""))
        {
            answerData.Add("profession", professionStr);
            answerData.Add("professionOther",
                professionOtherText.transform.Find("DescriptionText")
                    .GetComponent<Text>().text);
            totalNumOfQuesAnswered++;
        }
    }

```

Figure 5: Validation of Question 2 Code Snippet

The third question, shown in the Figure (6), asks the user about the player’s work setting. The contextual value of the selected checkbox is found by using the “GetComponentInChildren()” method that searches for the “text” parameter value. The player’s selection is then attached to the “answerData” object.

```

    if (settingToggles[i].GetComponent<Toggle>().isOn)
    {
        answerData.Add("work-in" + (i + 1),
            settingToggles[i].GetComponentInChildren<Text>().text);
        answeredSettingTogglesCount++;
    }

```

Figure 6: Validation of Question 3 Code Snippet

The fourth question, shown in the Figure (7), has the same logic as the third question. It asks a “yes” or “no” question about a player’s licensing. It then collects the text field information into the “answerData” object.

```

if (licensedMedProToggles[i].GetComponent<Toggle>().isOn)
{
    answerData.Add("licensedMedpro",
        licensedMedProToggles[i].GetComponentInChildren<Text>().text);
    answeredlicensedMedProTogglesCount++;
} }

```

Figure 7: Validation of Question 4 Code Snippet

The fifth question, shown in the Figure (8), collects the dropdown option value that has the information about the years of experience, adds to the number of correctly answered questions, and then proceeds into further validation blocks.

```

if (experienceDropdown.value != 0 && isValidatationDone)
{
    answerData.Add("experience", experienceStr);
    totalNumOfQuesAnswered++;
}

```

Figure 8: Validation of Question 5 Code Snippet

The sixth question’s functions, shown in the Figure (9), work in the same manner as the fifth, except for presenting the player with the optional text field in case they choose the “Other” dropdown option. The player can then choose to fill the text field value in that optional field. The “DescriptionText” component is then found by the “transform.Find()” method and its “text” value is then added to the “answerData” object with the “genderOther” key.

```

answerData.Add("gender", genderStr);
if (genderStr.Equals("Other") &&
    !genderOtherText.transform.Find("DescriptionText")
    .GetComponent<Text>().text.Trim().Equals(""))
{
    answerData.Add("genderOther",
        genderOtherText.transform.Find("DescriptionText")
        .GetComponent<Text>().text);
}
totalNumOfQuesAnswered++;

```

Figure 9: Validation of Question 6 Code Snippet

The seventh question, shown in the Figure (10), also asks to select the value from the dropdown menu. If the player selects the option that reads "Yes, another Hispanic, Latino, or

Spanish origin:", it will ask for mandatory submission of additional information into the text field that will be shown. This text value is then trimmed and received by a “transform.Find()” function. Such information is then added to the “answerData” object along with the dropdown option selection.

```
if (!hispanicStr.Equals("Yes, another Hispanic, Latino, or Spanish origin:"))
{
    answerData.Add("hispanic", hispanicStr);
    totalNumOfQuesAnswered++;
}
else if (hispanicStr.Equals("Yes, another Hispanic, Latino, or Spanish
origin:")
&& !hispanicOtherText.transform.Find("DescriptionText")
.GetComponent<Text>().text.Trim().Equals(""))
{
    answerData.Add("hispanic", hispanicStr);
    answerData.Add("hispanicOther",
hispanicOtherText.transform.Find("DescriptionText")
.GetComponent<Text>().text);
    totalNumOfQuesAnswered++; }
}
```

Figure 10: Validation of Question 7 Code Snippet

The eighth question, shown in the Figure (11), works with the toggle group. It gets the text value of the selected toggle and submits it to the “answerData” object.

```
for (int i = 0; i < ethnicityToggles.Length; i++)
{
    if (ethnicityToggles[i].GetComponent<Toggle>().isOn)
    {
        answerData.Add("ethnicity" + (i + 1),
ethnicityToggles[i].GetComponentInChildren<Text>().text);
        answeredEthnicityTogglesCount++;
    }
}
```

Figure 11: Validation of Question 8 Code Snippet

In a part of question eight’s validation, presented in the Figure (12), some toggles in the group allow users to enter optional specifications of their exact ethnicity. When the user checks

such a box, the text field appears near it. If the text is entered in that field, it will be found by the “transform.Find()” method, will be trimmed off all spaces by the “Trim()” method, and will be added to the “answerData” object.

```
if(!otherAsianText.transform.Find("DescriptionText")
.GetComponent<Text>().text.Trim().Equals(""))
{
    answerData.Add("otherAsianText",
otherAsianText.transform.Find("DescriptionText")
.GetComponent<Text>().text);
}
```

Figure 12: Question 8 Extra Text Field Code Snippet

When the error is made in the input, the validation function will trigger the modal window to appear, as shown in the Figure (13). The modal window will have a prompt that will tell the user which question they made a mistake in. In the case of a mistake in the correctness of the zip code in the first question, the modal window will show a custom prompt with the error message from question one’s validation block. The “isValidationDone” boolean variable will be set to “false”, to prevent accidental data submission. It is then again said to be “true” by the outer function after the modal window is closed by pressing the “Okay” button.

```
if(zipErrorMsg != null && totalNumOfQuesAnswered == 0)
{
    validationText.text = zipErrorMsg.Trim();
}
else
{
    validationText.text = "Please Answer Question "
+ (totalNumOfQuesAnswered + 1);
}
```

Figure 13: Validation Modal Window Code Snippet

After the information is validated, it needs to be submitted in the database, as presented in the Figure (14). It is first logged in the Unity console, and then two conditions are checked: whether

the number of answered questions is equal to the total questions in the scene and if the “isValidationDone” boolean is still “true”, which confirms that there are no errors in the validation. The data is then submitted to the database by creating a coroutine that uploads the answerData object to a new “WrightServer()” database instance.

```
if ((totalNumOfQuesAnswered == TOTAL_QUESTIONS) && isValidationDone)
{
    Debug.Log("upload data");
    try
    {
        // Upload the data object to the server
        StartCoroutine(new WrightServer().upload(answerData));
        thankYouPanel.SetActive(true);
        questionsPanel.SetActive(false);
    }
}
```

Figure 14: Submitting the Collected Information Code Snippet

Using the “foreach” loop, the “answerData” object is logged to the Unity console before being submitted, as shown in the Figure (15). If validation of every question goes without errors, the keys and values created in the validation functions are displayed.

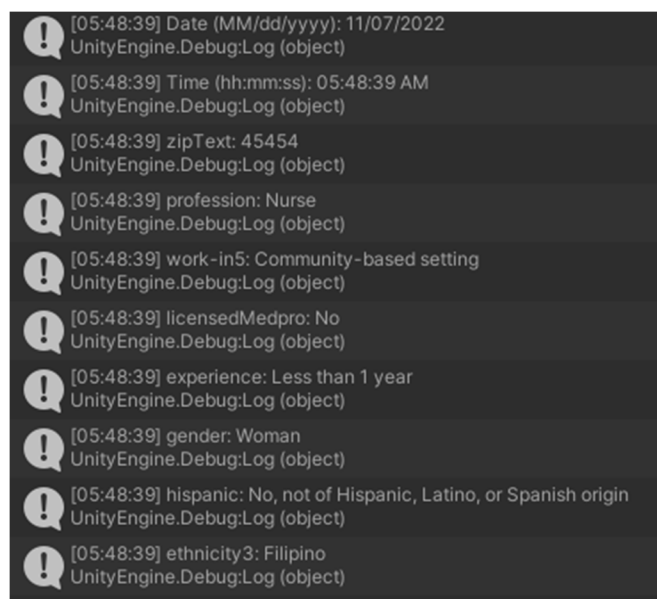


Figure 15: AnswerData Object Example

The successful submission to the database is then confirmed by the Unity console. After the database responds to the app, it returns the type of REST API request (in this case, “POST”), and the success message, as shown in the Figure (16).

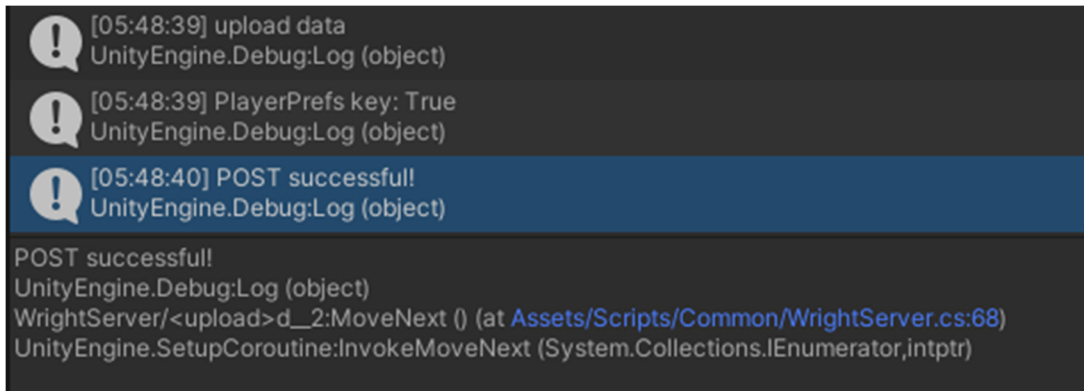


Figure 16: Submission to the Database

After the validation and data submission are complete, the application shows the “Thank you” panel with a small assisting text and the “Next” button that allows proceeding into the next scene. This button calls the function that loads the subsequent scene window, as shown in the Figure (17).

```
PlayerPrefs.SetString("LoadScene", "Erica_LGBTQ_Intro");
PlayerPrefs.SetString("UnloadScene", "DemographicScene");
Globals.levelLoaderEnum = Globals.LevelLoaderEnum.MAConvo;
SceneManager.LoadScene("Erica_LGBTQ_Intro", LoadSceneMode.Single);
```

Figure 17: Loading the Next Scene Code Snippet

2. Assessment scenes

The first, second, and final assessment scenes use similar structures and functions to do their validation. The process is less elaborate and only includes validating the checkbox groups. The checkbox groups all have five checkboxes with text labels near them, that enumerate the boxes from “1” to “5”. Only one checkbox can be checked at a time in a defined checkbox group. One

notable example from the assessment scenes is the interactive modal window that appears in question 15(a) of the final assessment, presented in the Figures (18, 19). That question has an additional textbox that requires the user to provide explanations, in case the user chose checkboxes “1”, “2”, or “3”. The functionality is implemented using two similar functions. They make the optional text area appear and disappear and move all questions below by a certain number of pixels. That amount had to be tested on all platforms, and the optimal number was found to be between 140 and 150.

```
Vector3 startPosSixteen = questionSixteen.transform.position;
Vector3 startPosSeventeen = questionSeventeen.transform.position;
Vector3 startPosNextButton = nextButton.transform.position;
// Debug.Log("Q15A active? " + questionFifteenA.activeSelf);
if(!questionFifteenA.activeSelf){
    questionSixteen.transform.position = new Vector3(startPosSixteen.x,
    startPosSixteen.y-150, startPosSixteen.z);
    questionSeventeen.transform.position = new Vector3(startPosSeventeen.x,
    startPosSeventeen.y-150, startPosSeventeen.z);
    nextButton.transform.position = new Vector3(startPosNextButton.x,
    startPosNextButton.y-140, startPosNextButton.z);
    questionFifteenA.SetActive(true);
} else {
    questionFifteenA.SetActive(true);
}
```

Figure 18: Interactive Text Area Script Code Snippet

This simulation contained the following learning objectives:

- Gain appreciation of the challenges faced by persons identifying as LGBTQIA+.
- Recognize how social determinants of health can exacerbate challenges faced by persons identifying as LGBTQIA+.
- Identify how implicit biases can affect the experience of persons identifying as LGBTQIA+ in the healthcare setting.
- Recognize that understanding an individual's unique life experiences can decrease vulnerability to the impact of implicit biases and increase compassion for the individual.

15. I feel that this simulation met all of the learning objectives listed above.

1. Strongly agree 2 3 4 5. Strongly disagree

15(a). Which learning objective(s) do you feel were not adequately addressed in this simulation experience and why?

Enter text...(Optional)

Figure 19: Interactive Text Area UI Example

2.1.4 User Interface Implementation

Data collection scenes needed a UI that was legible and easily understandable, and at the same time effective in collecting users' information. As shown in the Figure (20), the UI elements' parent object was defined by a canvas that included the question panel. That panel consisted of text boxes, toggle check marks, and dropdown objects. Outside that panel, there was a footer that had information about the Ohio Department of Medicaid and Wright State University. The footer was fixed at the bottom and was not scrollable. An essential element that controlled the entire flow of the scene was the "UIScript" game object. This UI element contained bindings of all interactive element groups to the corresponding game object variables that were created in the script. The UI also contained an important hidden element - the validation modal window. It only appeared when the validation script encountered an error in the user's input.

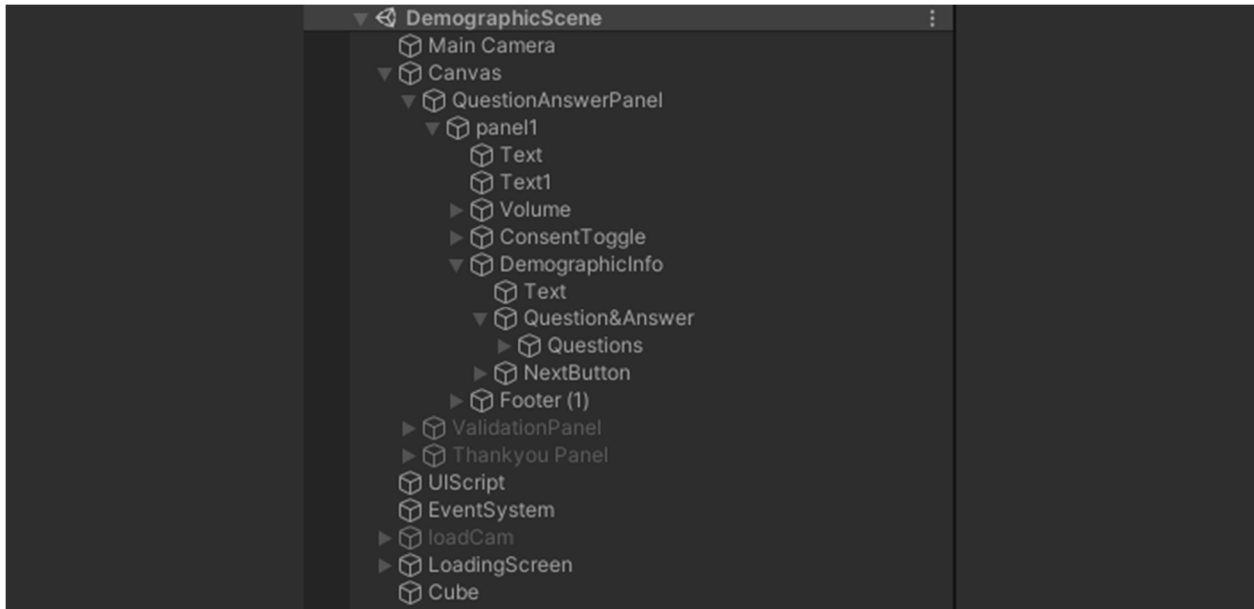


Figure 20: UI Hierarchy Tree

As mentioned before, the UI elements had three states - the default state, which presented the backgrounds of the elements with variations of white colors, the selected state, shown by a light-green background, and the error state, shown by a red-colored background. These states were also represented in the Unity UI, as shown in the Figure (21).

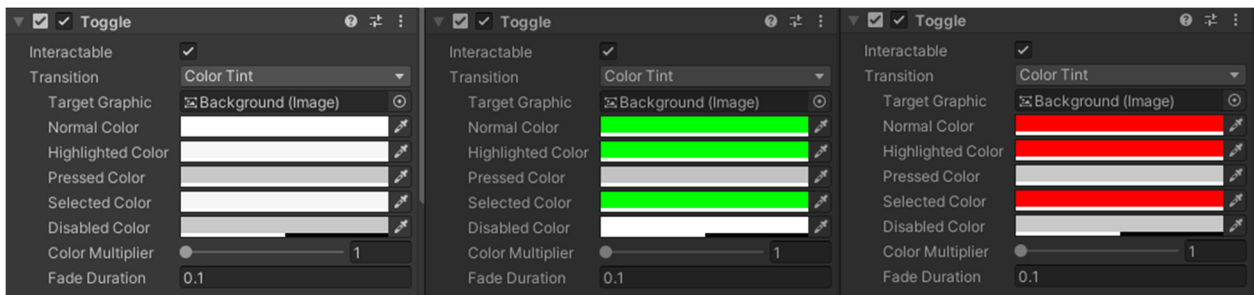


Figure 21: Toggle Group Color States

When the player pressed the “Next” button, the main validation script was triggered, and it invoked the separate validation functions for each of the eight questions in a subsequent manner. As soon as it encountered an error, the execution of further validation blocks stopped, and the error text from the function that triggered the error was passed into the modal window. That modal

window then became visible in the UI with the passed error message text. The text indicated the question number and either the description of the violated condition or the request to answer that question if no option was selected, as shown in the Figure (22).

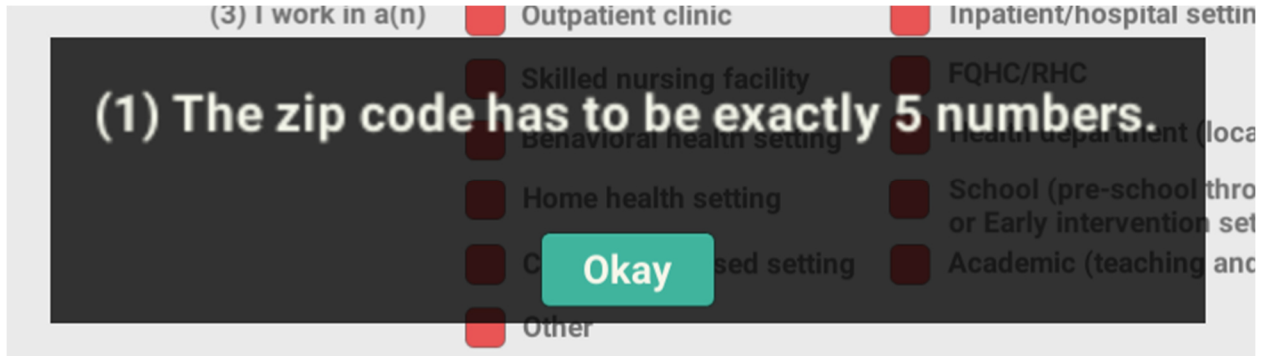


Figure 22: Validation Modal Window Example

In the case of dropdown elements, the user had to choose from the predefined values, with certain values also having a free-form text box appearing that required additional input. For instance, if the user selected “Physician” as their specialty, the script would make a text box near the dropdown menu visible and will only proceed if the user enters what type of physician they are, as shown in the Figure (23).

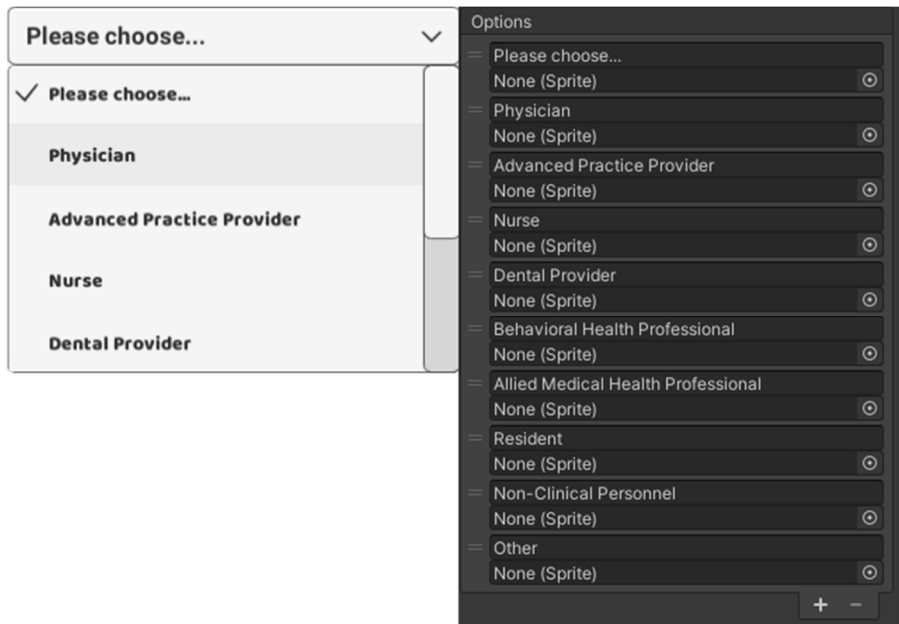


Figure 23: Dropdown UI Example (Left) With its Value Settings (Right)

Every checkbox was assigned to a group that corresponded to the question. When the user selected one checkbox, all other checkboxes in that group were deselected in the script, as shown in the Figure (24).

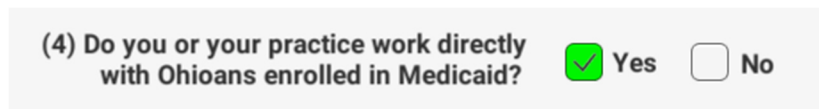


Figure 24: Active Checkbox Group Example

1. Connection of UI to the Script

The UI in the demographic script and assessment surveys required connecting its components with the game objects defined in the corresponding scripts, as shown in the Figure (25). Each interactive UI element had to be bound with the function in the validation script. Such binding was defined by the “On value changed” event handler, defined in the Unity API. Sometimes such binding required adding two separate functions from the script. This had to be done for the elements that had a double intent, such as some ethnicity toggles. For instance, the

“Other Pacific Islander”, “Other Asian”, and “Some Other Race” toggles at the same time belonged to the ethnicity toggle group and were used as toggles for the appearance of the free-form input fields. They appeared to let the user type any optional information about their ethnicity. The “Next” button was also connected through the “On Click” handler by using a similar procedure, as presented in the Figure (26).

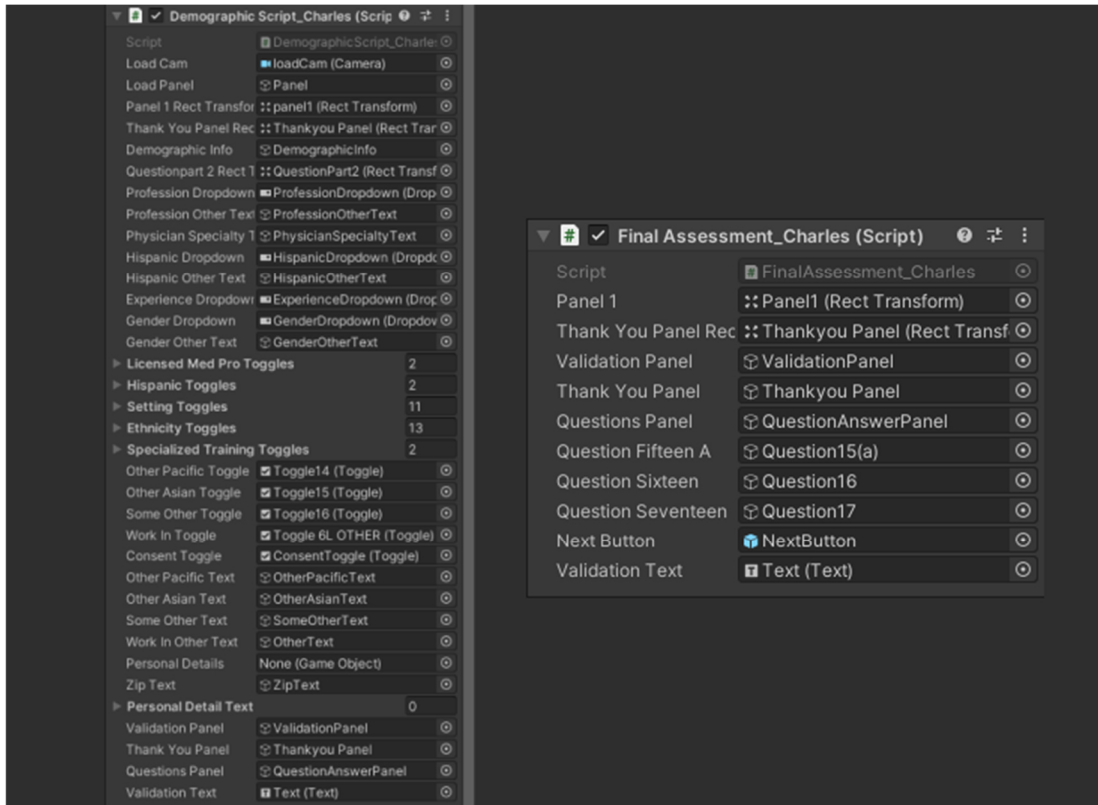


Figure 25: Binding UI to Script in Demographic (Left), Final Assessment (Right) Scenes

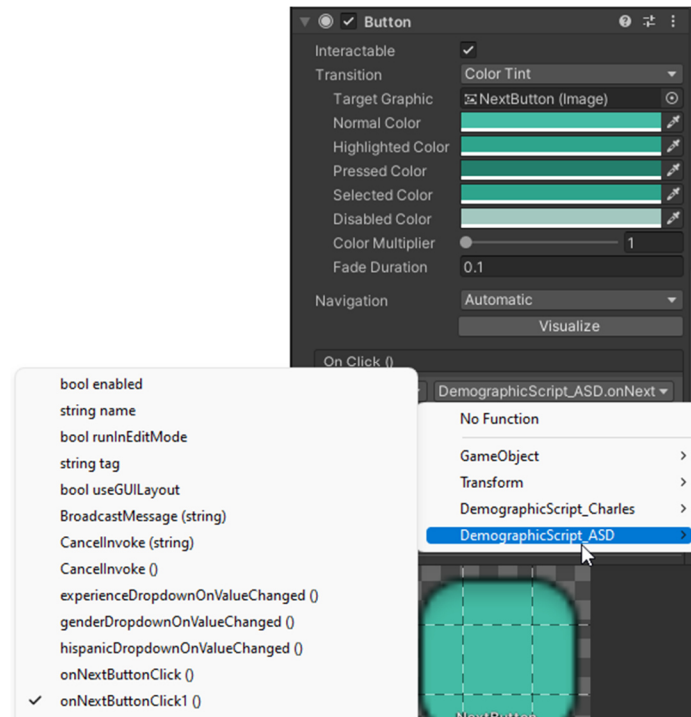


Figure 26: Connecting the “Next” Button to a Script Function

2. Testing

After the development of the UI and scripts, the data collection scenes needed to be thoroughly tested. The testing process included multiple important steps. The first step was to check the validation functionality. That required continuous and repetitive testing of all questions in every scene. Each question was first tested separately from others, and the correct performance of its corresponding function was confirmed. Then, all questions were tested together, and the functionality of the on-click script was observed. That script needed to correctly gather all information into an object, and then that object needed to be correctly submitted into the spreadsheet of the database.

The second step was to check the scenes on all possible viewport sizes, and on all possible platforms. That required playing through all scenes using different resolutions and switching the platforms in the Unity Engine to check for any changes that occurred. That process revealed

multiple layout issues that needed to be addressed on the UI side. After testing all functionalities, the scenes were inserted into the larger serious game application. The examples of screen ratios are shown in the Figure (27). An example of adjusting settings for another platform type is shown in the Figure (28).

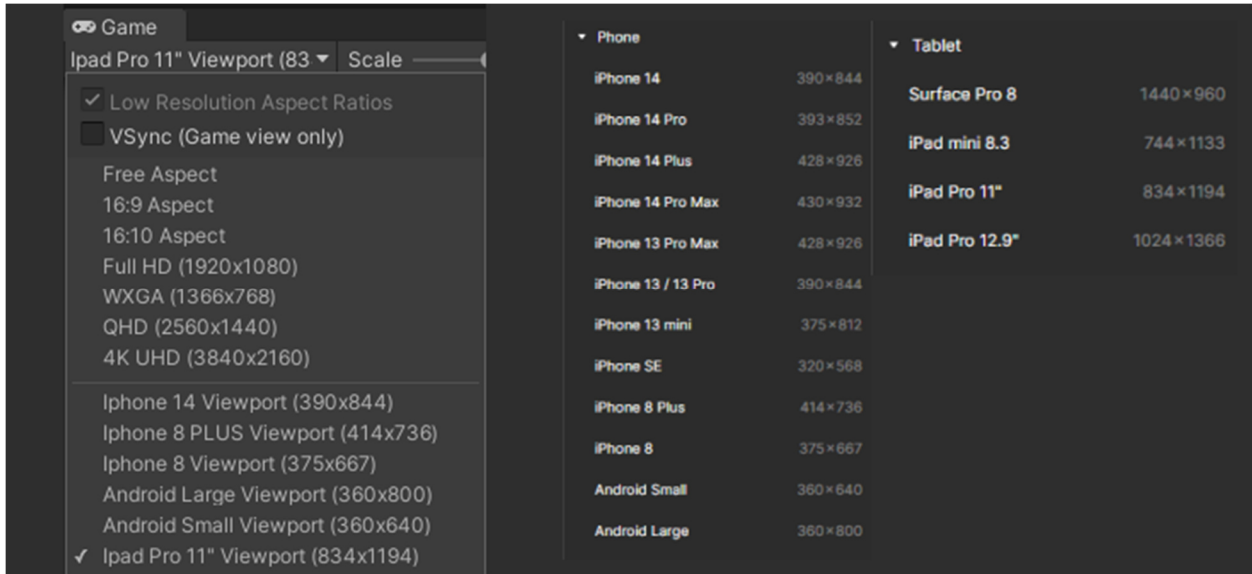


Figure 27: Testing on Popular Viewports (Left) and More Viewport Examples (Right)

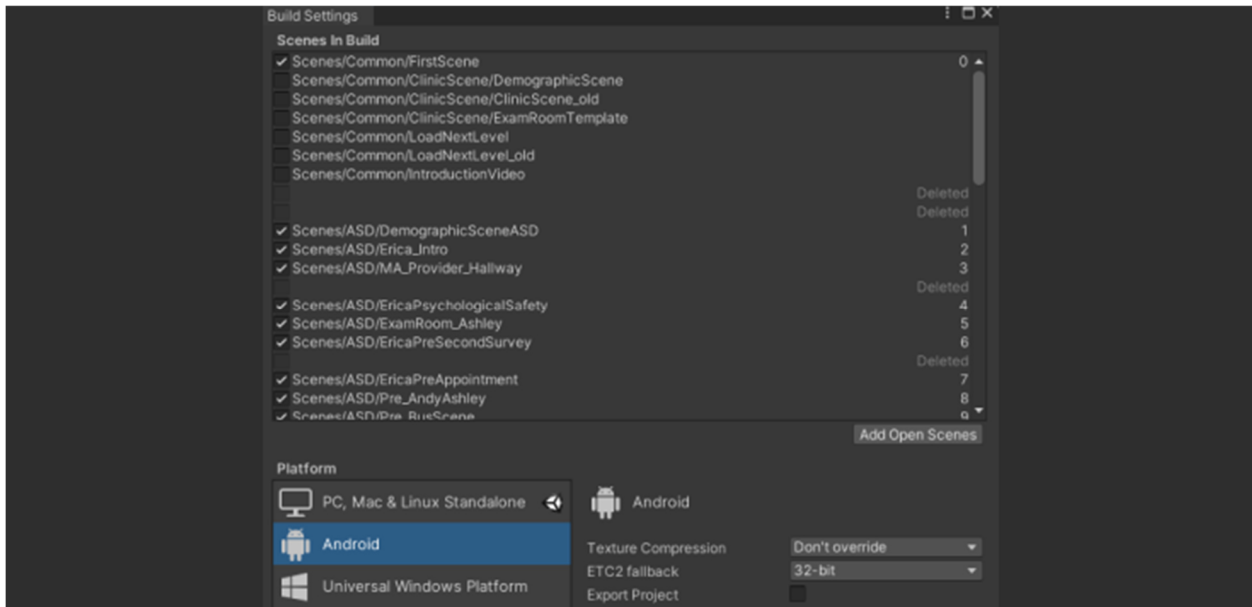


Figure 28: Adjusting Build Settings for a Different Platform Type

The third step was to check all scenes as they were already inserted in the “build” of the application. This was crucial because every scene in the WrightLIFE project was connected to the previous scene through the scripts, and if any of the scenes in that “chain” was broken, the further scenes would not load. An example of loading the next scene is shown in the Figure (29). Moreover, if the scene loader function was not correctly defined, that would also crash the entire application. After this step, the testing was done, and the application was ready to be built into all devices and dispersed.

```
public void OnThankYouPanelCloseButtonClick()
{
    PlayerPrefs.SetString("LoadScene", "ThankYouScene");
    PlayerPrefs.SetString("UnloadScene", "FinalAssesment");
    //Globals.levelLoaderEnum = Globals.LevelLoaderEnum.MAConvo;
    SceneManager.LoadScene("ThankYouScene", LoadSceneMode.Single);
}
```

Figure 29: Loading the Next Scene Code Snippet

2.2 Life Course Game

A significant part of the simulation consists of the life course game. The life course depicts events that happened to the game character prior to the doctor’s appointment. The life course game’s primary purpose is to let the player “dive” into the past of the main game character. It presents the player with the chronological life story of the patient, which starts at an early age and continues into the present day. The master script for the life course game was developed by a team of medical professionals. The approach and illustrations used in the life course were also thoroughly discussed by the medical team. The development of the life course, though not heavy

on the codebase, shows how proper user experience can be achieved through many rounds of development, reviews, and testing.

1. Previous life course game

In the serious game that was developed previously under the same umbrella of projects, the life course game was created using a 3D engine and depicted a character running on the road, with the dice appearing on the screen when the character stopped. The player could roll the dice and the number on the dice depicted how far the character would run. The character movement played a purely decorative role, while the entire content of the life course was presented in the appearing modal window in a text format. The modal window appeared every time before the player was asked to roll the dice again. The life course is a significant part of a simulation, with almost one-third of the estimated play time dedicated to it. There was a consensus among the medical team that previously created user experience did not fully immerse the player into the characters' life experiences. The agreement was that the 3D character and their movement was not triggering the player's feelings, and the text on the screen was the only instrument to connect with the player. The text content was not a sufficient medium for conveying the situations that were faced by the in-game characters in their past. The decision was made to create a more memorable user experience for the healthcare providers, which will be visually striking and will invoke certain emotions that will act as a necessary supplement to the simulation's learning experience. Additionally, it was decided that each life course game will juxtapose the patient character with another character that did not have the same social determinants of health that the primary character experienced. This comparison was intended to illustrate how social determinants of health and discriminatory experiences can have enduring consequences.

2.2.1 Tools

Creating the UI and design for the life course game involved the use of multiple graphic editing tools and an image stock library. Its implementation also required adding all illustrations into the Unity Engine and keeping proper versioning, using the tools described above (Unity3D Engine, GitHub, Visual Studio Code).

1. Adobe Illustrator

Adobe Illustrator is a vector-based software that is focused on the creation of artwork and designs using many sophisticated drawing tools [36]. With Illustrator, it was possible to create vector graphics and manipulate assets downloaded from the stock library. To create the required user experience, hundreds of assets needed to be created and organized into dozens of artboards. Assets like character models, backgrounds, and UI elements were all initially developed in Illustrator, and were then advanced into other editing software for further use.

2. Figma

Figma is a cloud-based multi-platform UI/UX design application that is used for interface design and prototyping [37]. It allows quickly delivering large amounts of assets for review, manipulating, and changing dozens of asset frames simultaneously, and generating assets for presentation in multiple aspect ratios. This tool proved to be most useful to combine the vector assets created in Adobe Illustrator and present them to the medical team in different variations. Figma's export features allow defining the dynamically updating folder hierarchy for all assets, to integrate them later quickly into the Unity Engine's UI.

3. Freepik

Freepik is a large stock asset library that is currently owned by Google [38]. It was chosen to be a primary tool for searching stock vector graphics for them to be later used in creating visual design and UI elements of the simulation. Other stock library websites, such as Shutterstock, were also utilized to a lesser extent.

4. Adobe Photoshop

The use of Adobe Photoshop was a great secondary tool for sketching certain examples that were used for the iteration reviews. Adobe Photoshop is a raster image editing software that allows the creation of graphic design and digital art [39]. Since all life course assets were developed using vector graphics, Photoshop acted as a supplementary software that allowed the quick organization of certain sketches of UI and designs.

2.2.2 Implementation

The life course game required the creation of a conceptual model that would have a significant positive impact on the player's experience. A conceptual model is a visual representation of a system. It shows how the elements should be presented, and how they should be interacted with, and discusses the effectiveness of these elements. In the educational setting, it additionally focuses on providing the user with high learnability and efficiency. Another factor that plays a role in it is the level of user satisfaction [40]. The conceptual model went from having multiple interactive elements presented in the UI, to being concise and heavily focused on the presented scripts.

The first iteration was developed by collecting hundreds of stock assets to work with, and then by using vector editing software to create appropriate illustrations. Such illustrations, shown in the Figures (30, 31), were presented in the form of a background image with the characters of Charles, Ashley, Sarah, and Chris in the foreground, along with the secondary characters.



Figure 30: ASD Life course First version - Ashley Example

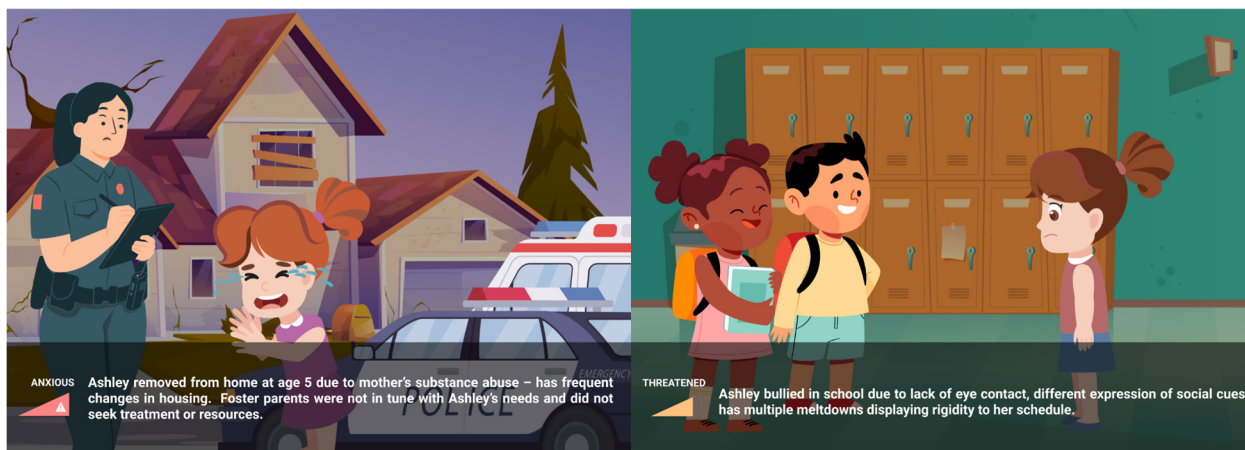


Figure 31: ASD Life course First version - Sarah Example

In addition to the illustrations, multiple interactive elements were proposed to be created. These elements were intended to enhance the educational experience. Multiple scoring and

measuring elements were proposed to represent the player with the emotional states of the characters, and with certain conclusions about the presented life course. The proposed elements, shown in Figures (32, 33) included emoji sliders that could show the “mood” of the character during every scene of the life course, the anxiety meter, which would have shown the emotional comfort of the character, and the psychological safety meter, which was an improved version of the anxiety meter. The emoji slider and anxiety meter were later discarded, and the psychological safety meter was later used in the animated parts of the simulation. The reasoning behind that was the distractions such elements created for the player since the goal was to focus the player’s attention exclusively on the visual contents.

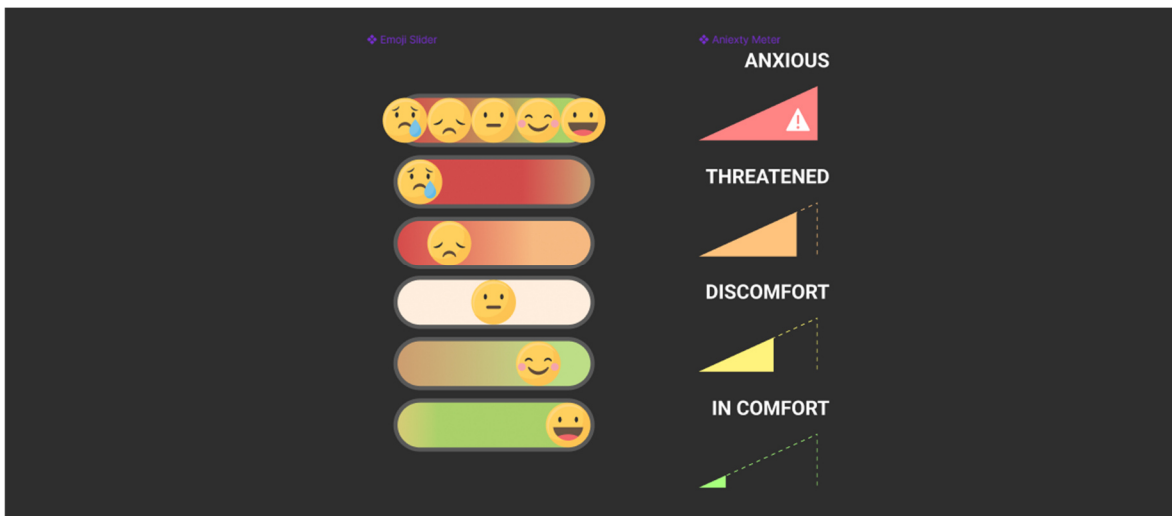


Figure 32: Emoji Slider and Anxiety Meter



Figure 33: Psychological Safety Meter

After multiple rounds of reviews, it was decided that all elements that increase the necessary attention span of the player should be removed from the life course portions of the simulations. These included all meters and proposed videos. Moreover, each illustration's background was minimized and placed into a circular mask, with the character placed outside and in front of that mask. These decisions were due to the playtime restrictions of the simulation. The subtitles were also removed and then replaced with a large legible text to the left or right of the illustrations. All presented characters were also redesigned to be presented in a similar stance and pose. The Figure (34) gives an example of such pose consistency by showing Chris from the LGBTQIA+ case. The life course now showed how characters consistently aged and how their life changed with time.

Parents side with Christopher as example of how a son should be. Christopher tries to be emotionally supportive to his brother.



Marries his college girlfriend.



Has two children. Second child has Down Syndrome.



Builds house in new suburban housing development.



Figure 34: LGBTQIA+ Life course - Chris Character Examples

During final reviews, many character assets were replaced with items that corresponded to the contents of the illustration. For instance, in the LGBTQIA+ case, Charles was diagnosed with asthma at age four. It was decided it to be a better idea to depict the vector model of a pill bottle rather than presenting a child version of Charles himself. Following the creation of necessary objects, it was also decided to decrease the vibrance and saturation settings to better represent the hardships that characters faced. The number of required assets was also reduced to fit them into the simulation's timespan. These changes were deemed to be the best possible implementation of a correct immersion of healthcare providers into the characters' past experiences. Figures (35, 36) represent the final alterations' process.

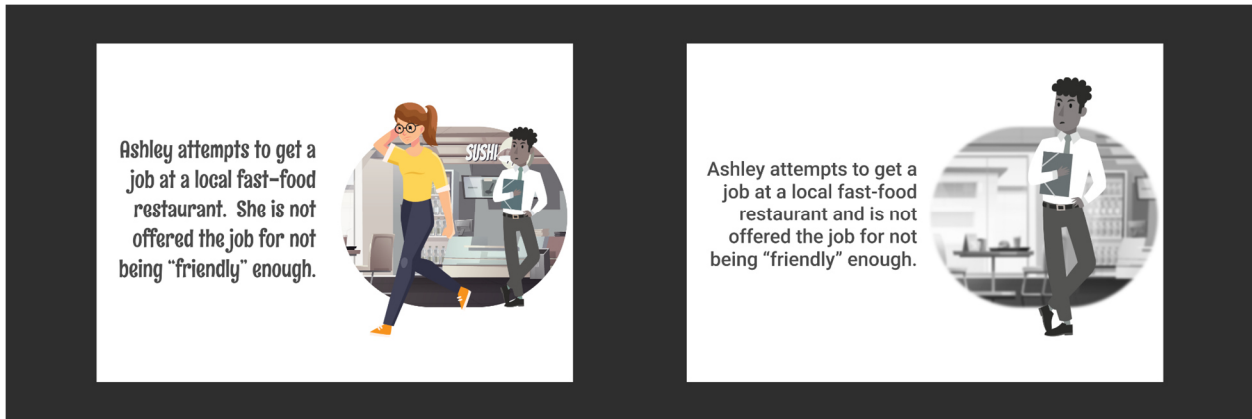


Figure 35: ASD Life Course Example - Removing the Main Character from the Scene

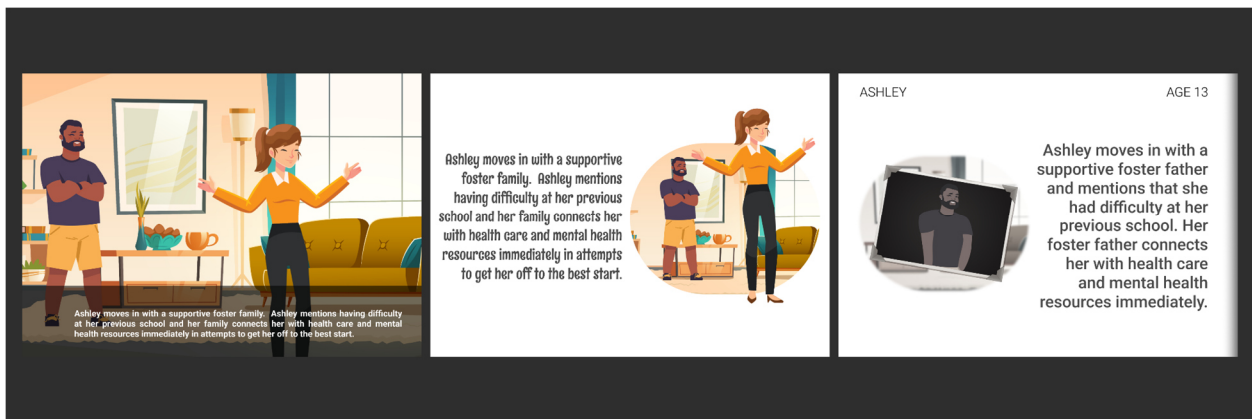


Figure 36: ASD Life Course Example - First Version, Altered Version, and Final Version

1. Implementation in Unity UI

The removal of all distracting elements led to having a simple UI with just “Previous” and “Next” buttons, which changed the illustration slides accordingly. It was additionally decided to implement the life course game in the form of a photo album, shown in the Figures (38, 39), to directly tie it to something that happened in the past. Thus, the appropriate script functionality had to be found to adhere to these requirements. The script, shown in the Figure (37) that provided the page-flipping animation effect was in the asset store and added to the scene [41]. After some adjustments and after adding the illustration assets to the scene, the script was able to present the

user with a convincing page-flipping animation that was connected to the buttons presented on the screen. Additionally, it was necessary to minimize the image sizes as they were added to the Unity scene to reduce the total size of the simulations. Such reductions were done by scaling down the illustrations and reducing their pixel-per-inch resolution.

```
AutoFlip.cs
C:\ASD_Simulation_desktop > ASD_Simulation_desktop > Assets > Book-Page Curl > scripts > AutoFlip.cs
1 using UnityEngine;
2 using System.Collections;
3 [RequireComponent(typeof(Book))]
4 public class AutoFlip : MonoBehaviour {
5     public FlipMode Mode;
6     public float PageFlipTime = 1;
7     public float TimeBetweenPages = 1;
8     public float DelayBeforeStarting = 0;
9     public bool AutoStartFlip=true;
10    public Book ControlledBook;
11    public int AnimationFramesCount = 40;
12    bool isFlipping = false;
13    // Use this for initialization
14    void Start () {
15        if (!ControlledBook)
16            ControlledBook = GetComponent<Book>();
17        if (AutoStartFlip)
18            StartFlipping();
19        ControlledBook.OnFlip.AddListener(new UnityEngine.Events.UnityAction(PageFlipped));
20    }
21    void PageFlipped()
22    {
23        isFlipping = false;
24    }
25    public void StartFlipping()
26    {
27        StartCoroutine(FlipToEnd());
28    }
29    public void FlipRightPage()
30    {
31        if (isFlipping) return;
32        if (ControlledBook.currentPage >= ControlledBook.TotalPageCount) return;
33        isFlipping = true;
34        float frameTime = PageFlipTime / AnimationFramesCount;
35        float xc = (ControlledBook.EndBottomRight.x + ControlledBook.EndBottomLeft.x) / 2;
36        float xl = ((ControlledBook.EndBottomRight.x - ControlledBook.EndBottomLeft.x) / 2) * 0.9f;
37        //float h = ControlledBook.Height * 0.5f;
38        float h = Mathf.Abs(ControlledBook.EndBottomRight.y) * 0.9f;
39        float dx = (xl)*2 / AnimationFramesCount;
40        StartCoroutine(FlipRTL(xc, xl, h, frameTime, dx));
41    }
}
```

Figure 37: AutoFlip Script

CHARLES

AGE 11



Charles is ridiculed by his parents at home because of the way he holds his hands. Charles is bullied and beat up at school because his classmates say he is gay and because of his weight.

AGE 11

CHRISTOPHER



Christopher is held up by his parents as an example of how Charles should be. Christopher is popular at school.

Figure 38: LGBTQIA+ Life Course - Photo Album Format Example

ASHLEY

AGE 11



Ashley goes to a new school, is placed in regular classes and has difficulty making friends. She often ends up in trouble and the guidance counselor recommends that she be tested for autism. Her foster family is unable to get her connected to the appropriate resources.

AGE 11

SARAH



Sarah ends up getting into special academic classes and attends space camp. Sarah begins playing soccer and her parents take her to a gym to practice.

Figure 39: ASD Life Course - Photo Album Format Example

Finally, there was a necessity to add the cover page for the created album-like life course UI. Such cover pages were made for both cases, with the Figure (40) showing an example of one. The cover is the initial slide presented in the book, and the book “opens” starting from it. After that part was finished, the life course was deemed complete.

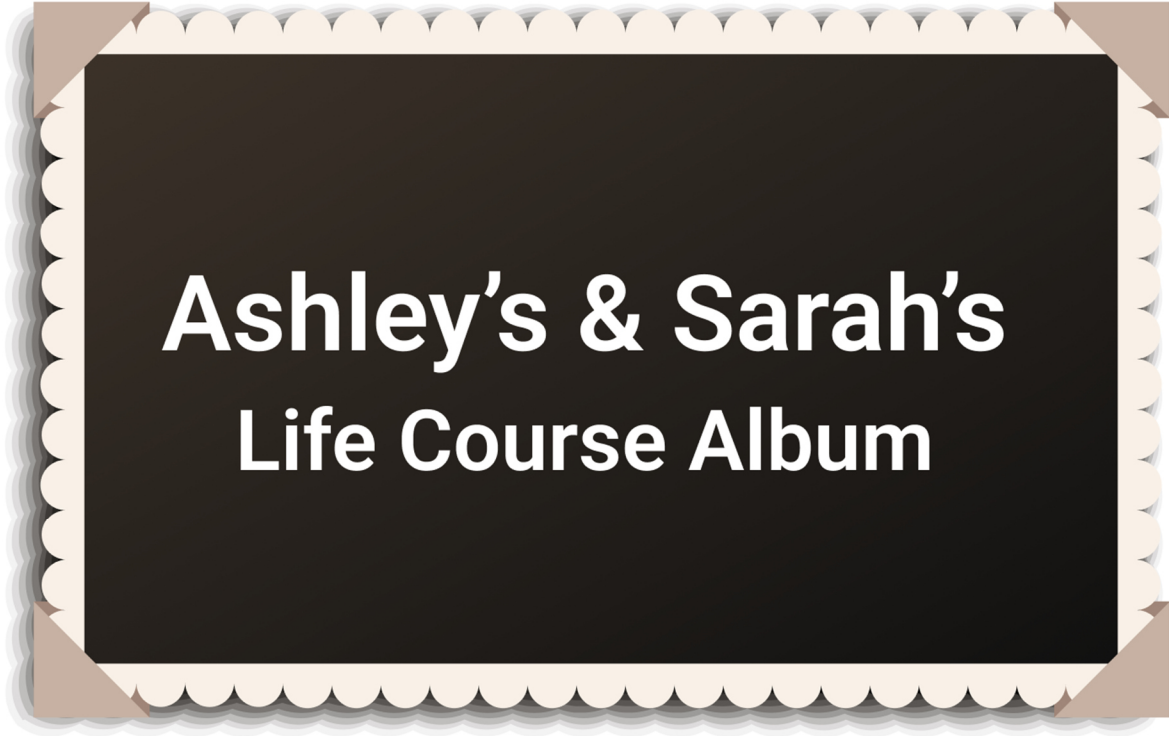


Figure 40: ASD Life course – Cover

3 RESULTS

The development process, as presented in Section 2.2, resulted in the creation of data collection scenes and life course scenes that contributed to producing an effective educational tool that can help eliminate implicit bias towards LGBTQIA+ and ASD patients. Screenshots that depict data collection scenes and a final version of a life course game are presented in the Figures (41 - 52).

3.1 Data Collection Scenes

Welcome to Wright State University's Virtual Patient Project!

The 30-minute interactive simulation focuses on Charles, a 60-year-old gay man with an appointment at a community health center. For a portion of the simulation, you will be in the role of the patient, Charles, and for other portions, you will be in the role of Charles' primary health care provider.

The purpose of the training is to decrease the impact of implicit biases in health care to help reduce health disparities, improve patient care, and improve health outcomes.

To participate, you must agree to the following data collection and privacy statement:

The Medicaid Care Experience Simulation (MCarES) project is funded by the Ohio Department of Medicaid and administered by the Ohio Colleges of Medicine Government Resource Center. The views expressed in this training are solely those of the authors and do not represent the views of the state of Ohio or federal Medicaid programs.

This virtual reality simulation contains content that some people may find triggering. If you believe that you will find the material triggering, you may choose not to participate in it or you can withdraw at any time.

In addition to some demographic information, we will collect information on your actions and decisions during the experience as well as your responses to the brief pre-, mid- and post-experience surveys included in the simulation.

Data are anonymized and stored securely. We respect your trust and protect your privacy, therefore the data we collect will never be sold or shared with third parties outside of the Ohio Department of Medicaid.

Your participation is voluntary. You may choose to stop at any time. Your responses will remain confidential. All data will be reported in aggregate and your responses will have no impact on your relationship with the Ohio Department of Medicaid, universities/academic medical centers in Ohio, or any other entity.

Please turn the volume up and ringer on for the entire simulation!

By checking the box below, you agree with and consent to this data collection policy. I consent to the application use and data collection policy

In the state of Ohio, per Section 4743.10: Medical Practitioner Conscience Clause, a medical practitioner, health care institution, or health care payer has the freedom to decline to perform, participate in, or pay for any health care service which violates the practitioner's, institution's, or payer's conscience as informed by the moral, ethical, or religious beliefs or principles held by the practitioner, institution, or payer. This law was signed on July 1, 2021 and made effective on September 30, 2021.

Figure 41: LGBTQIA+/ASD Demographic Scene: Introduction

Demographic Questions

(1) What is the ZIP Code where you work most of the time?

(2) I am a(n)...

(3) I work in a(n) (Select all that apply)

<input type="checkbox"/> Outpatient clinic	<input type="checkbox"/> Inpatient/hospital setting
<input type="checkbox"/> Skilled nursing facility	<input type="checkbox"/> FQHC/RHC
<input type="checkbox"/> Behavioral health setting	<input type="checkbox"/> Health department (local, county, or state)
<input type="checkbox"/> Home health setting	<input type="checkbox"/> School (pre-school through high school) or Early intervention setting
<input type="checkbox"/> Community-based setting	<input type="checkbox"/> Academic (teaching and/or research)
<input type="checkbox"/> Other	

(4) Do you or your practice work directly with Ohioans enrolled in Medicaid? Yes No

(5) How many years have you worked in healthcare?

(6) What is your gender identity?

(7) Are you of Hispanic, Latino, or Spanish origin?

(8) What is your race/ethnicity? (Select all that apply)

<input type="checkbox"/> White	<input type="checkbox"/> Black or African American
<input type="checkbox"/> Chinese	<input type="checkbox"/> American Indian or Alaska Native
<input type="checkbox"/> Filipino	<input type="checkbox"/> Asian Indian
<input type="checkbox"/> Vietnamese	<input type="checkbox"/> Korean
<input type="checkbox"/> Japanese	<input type="checkbox"/> Middle Eastern or North African
<input type="checkbox"/> Filipino	<input type="checkbox"/> Other Pacific Islander
<input type="checkbox"/> Guamanian or Chamorro	<input type="checkbox"/> Other Asian
<input type="checkbox"/> Samoan	<input type="checkbox"/> Some other race

[Next](#)










Figure 42: LGBTQIA+/ASD Demographic Scene: Questionnaire

First Assessment Survey

1. With respect to having this individual as my next patient, the amount of ANXIETY I feel is:

1. Low 2 3 4 5. High

2. With respect to having this individual as my next patient, the amount of FRUSTRATION that I feel is:

1. Low 2 3 4 5. High

3. With respect to having this individual as my next patient, the amount of COMPASSION that I feel is:

1. Low 2 3 4 5. High

4. I expect that my encounter with this patient will be:

1. Easy 2 3 4 5. Difficult

5. I believe that this patient is largely responsible for being in their current circumstances.

1. Strongly disagree 2 3 4 5. Strongly agree

6. I believe that the circumstances in which this patient finds themselves are largely beyond their control.

1. Strongly disagree 2 3 4 5. Strongly agree

7. I attempt to act in nonprejudiced ways toward patients like this because it is personally important to me.

Figure 43: LGBTQIA+/ASD First/Second Assessment Scene Example

Final Assessment Survey

1. With respect to having this individual as my next patient, the amount of ANXIETY I feel is:

1. Low 2 3 4 5. High

2. With respect to having this individual as my next patient, the amount of FRUSTRATION I feel is:

1. Low 2 3 4 5. High

3. With respect to having this individual as my next patient, the amount of COMPASSION I feel is:

1. Low 2 3 4 5. High

4. I expect that my encounter with this patient will be:

1. Easy 2 3 4 5. Difficult

5. I believe that this patient is largely responsible for being in their current circumstances.

1. Strongly disagree 2 3 4 5. Strongly agree

6. I believe that the circumstances in which this patient finds themselves are largely beyond their control.

1. Strongly disagree 2 3 4 5. Strongly agree

7. I attempt to act in nonprejudiced ways toward patients like this because it is personally important to me.

1. Never 2 3 4 5. Always

8. I consider discrimination to be a serious social problem.

1. Strongly disagree 2 3 4 5. Strongly agree

9. I believe that I have biases toward certain types of people/ patients.

1. Strongly disagree 2 3 4 5. Strongly agree

10. Being LGBTQIA+ is a natural expression of sexuality/gender identity.

1. Strongly disagree 2 3 4 5. Strongly agree

11. I am comfortable providing health care to individuals identifying as LGBTQIA+.

1. Strongly disagree 2 3 4 5. Strongly agree

12. As a result of this simulation experience, I would be comfortable interacting with a patient/client similar to the one in the simulation scenario in my clinical or non-clinical role in the future.

1. Strongly agree 2 3 4 5. Strongly disagree

Figure 44: LGBTQIA+/ASD Final Assessment Scene (1)

13. This simulation experience equipped me with new knowledge and resources to apply in my clinical/non-clinical practice to improve the care experience and reduce health disparities for my patients/clients.

1. Strongly agree 2 3 4 5. Strongly disagree

14. I will apply at least one new thing that I learned from this simulation experience in my clinical/non-clinical practice.

1. Strongly agree 2 3 4 5. Strongly disagree

This simulation contained the following learning objectives:

- Gain appreciation of the challenges faced by persons identifying as LGBTQIA+.
- Recognize how social determinants of health can exacerbate challenges faced by persons identifying as LGBTQIA+.
- Identify how implicit biases can affect the experience of persons identifying as LGBTQIA+ in the healthcare setting.
- Recognize that understanding an individual's unique life experiences can decrease vulnerability to the impact of implicit biases and increase compassion for the individual.

15. I feel that this simulation met all of the learning objectives listed above.

1. Strongly agree 2 3 4 5. Strongly disagree

15(a). Which learning objective(s) do you feel were not adequately addressed in this simulation experience and why?

Enter text...(Optional)

16. How has your knowledge and/or attitudes about LGBTQIA+ persons changed after participating in this simulation? Please describe if this simulation bridged any gaps in knowledge and/or revealed any incorrect assumptions about this patient population that you may have had.

Enter text...(Optional)

17. Please describe how this simulation may be relevant to your clinical or non-clinical practice and areas you would like to explore for further development.

Enter text...(Optional)



Figure 45: LGBTQIA+/ASD Final Assessment Scene (2)

3.2 Life Course Game

ASHLEY

AGE 7



Ashley is bullied in school due to her lack of eye contact, her different responses to social cues, and she has multiple meltdowns.

AGE 7

SARAH



Sarah continues to acclimate well to school.

Figure 46: ASD Life course - Age 7

CHARLES

AGE 11



Charles is ridiculed by his parents at home because of the way he holds his hands. Charles is bullied and beat up at school because his classmates say he is gay and because of his weight.

AGE 11

CHRISTOPHER



Christopher is held up by his parents as an example of how Charles should be. Christopher is popular at school.

Figure 47: LGBTQIA+ Life course - Age 11

ASHLEY

AGE 11



Ashley goes to a new school, is placed in regular classes and has difficulty making friends. She often ends up in trouble and the guidance counselor recommends that she be tested for autism. Her foster family is unable to get her connected to the appropriate resources.

AGE 11

SARAH



Sarah ends up getting into special academic classes and attends space camp. Sarah begins playing soccer and her parents take her to a gym to practice.

Figure 48: ASD Life course - Age 11

CHARLES

AGE 25



Charles begins college but decides to drop out for a year due to being bullied by anti-gay students.

AGE 25

CHRISTOPHER



After graduating from college, Christopher has worked for an accounting firm for 3 years. He marries his college girlfriend.

Figure 49: LGBTQIA+ Life course - Age 25

ASHLEY

AGE 18



Ashley is accepted into a four-year university's computer science program, a goal that was identified in her IEP and transition plan. Ashley benefited from the support her foster family gave her and being linked to appropriate resources.

AGE 18

SARAH



Sarah is accepted into a four-year university with an undecided major and a full-tuition soccer scholarship.

Figure 50: ASD Life course - Age 18

CHARLES

AGE 47



Charles' PCP prescribes him a relatively new antidepressant, paroxetine, which contributes to additional weight gain.

AGE 47

CHRISTOPHER



Christopher has gained some weight. He has begun to exercise more seriously. He occasionally participates in 5K road races.

Figure 51: LGBTQIA+ Life course - Age 47

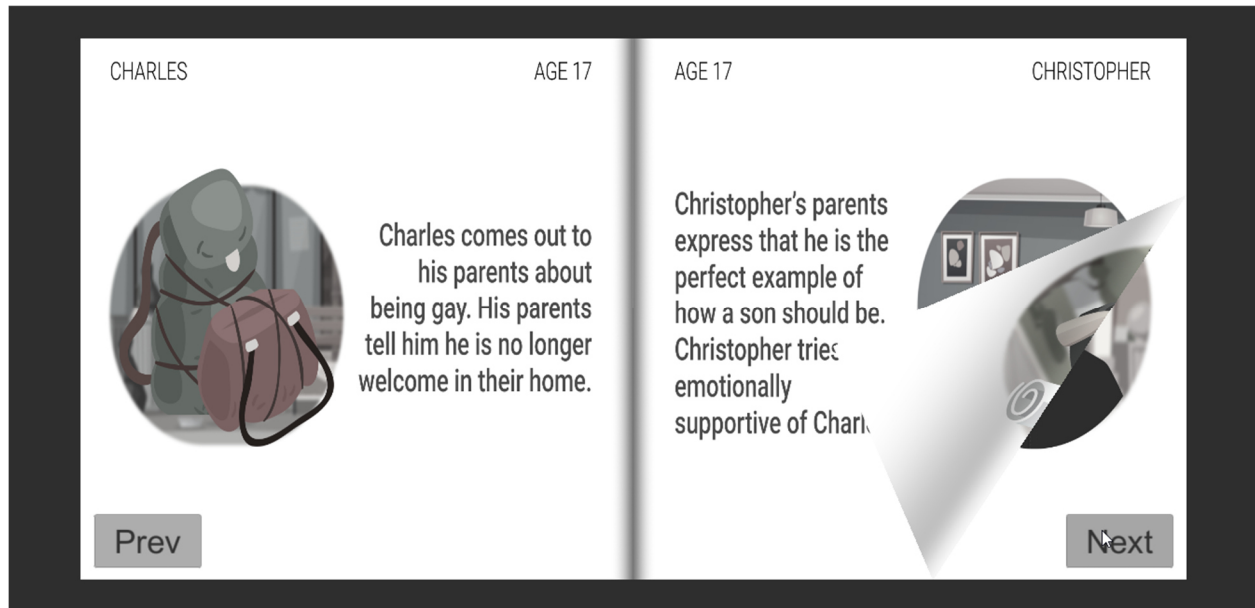


Figure 52: Life Course Game Unity UI and Page Flip Animation

4 DISCUSSION

4.1 Interpretation

The two developed serious game simulations presented experiential learning-based solutions to educating healthcare providers about the implicit bias issues among ASD and LGBTQIA+ communities. Both simulations addressed key concepts of a quality UI/UX.

The data collection scenes shown in Figures (41 – 45) successfully follow the user experience design flow. Since they are preceded by the patient hospital visit scenes or with the scenes where the character is discussing the hospital visit, the goal and intention are already set for the player. The data collection scenes continue that user experience flow and provide the player with a functional interface that asks a user for an action. The validation functionality acts as a confirmation medium for the player, by informing the player about the correctness of their input. Most importantly, data collection scenes' results are then evaluated by the user when they press the “Next” button to submit them. The feedback is then received from the player and then forwarded for future analysis. Thus, the “goal - intention - function - action - confirmation - evaluation - feedback” flow is fully implemented in the simulation. After that information is thoroughly analyzed, it can be used for further research.

The life course game, shown in Figures (46 – 52) is effective in immersing the healthcare provider in the simulation's context. Its content provides the player with the ability to adapt to the presented environment by giving more context to the lives of the characters. As mentioned in Section 1.1.3, it also has high fidelity, or in other words, a high level of realism, since the simulation is delivering scenes of societal rejection, hardships, and anxiety [19]. It can be also said that the Life course game is extensible since it provides robust support for the learning goals of

the simulation. Thus, both Life course games adhere to essential UX design practices and should be a valuable contribution to eliminating bias.

4.2 Challenges

1. Cross-platform testing

The UI and Front-End development of the WrightLIFE project presented certain challenges. During the development process, it was essential to test the UI's appearance in different contexts. That required adjusting the UI for different device screen ratios and testing the scenes on different mobile device platforms. All Survey scenes and the Life course scene had to be tested on three platforms: iOS, Android, and PC. Every scene was additionally tested on tablet devices, specifically iPad. The feedback from testing was then reviewed and appropriate changes were made if the scene contents appeared incorrectly on the screen. Some UI elements in the survey scene did not align on certain mobile viewports and required manual readjustment using relative margins. Such elements included the buttons in the "Thank You" scene, the interactive pop-up field for one of the questions in the Final Assessment scene, and some other buttons.

2. Creating quality life course game illustrations

Since the game's characters were created in the 3D environment of the Unity Engine, there was a need to create life course character illustrations that are close in their appearance to the ones presented in the simulation. It was important to ensure that the player will automatically understand that the characters presented in the life course game are the same as in the other scenes of the simulation. This required finding a way to develop the assets that will immediately resemble their 3D versions. Multiple methods of asset creation were considered. Converting the 3D assets into

2D illustrations was regarded as an option, with the additional possibility of adding contours to the textures to make them look more “cartoon-like”. The issue with that approach was that it did not clearly express characters’ emotions due to the Unity Engine texture lighting issues and would not fit into the 2D backgrounds that were proposed to be created. Thus, the way of creating vector 2D assets was chosen. It required thorough editing of the vector shapes to create appropriate emotions and poses for the characters. It was also necessary to create unique clothing and hairstyles that represented the aging of Charles and Chris characters. Overall, the process of 2D character development required utilizing many tools of Adobe Illustrator and Adobe Photoshop.

4.3 Future Work

1. Animating the life course game

The current version of a Life course Game presents healthcare providers who are going to play the game with a memorable and impactful experience. However, there are multiple ways in which it can be improved in the future. As shown by numerous empirical studies, animation can have a significant advantage on the user’s learning experience. The animated parts of the life course were discarded during the development process because of the simulation playtime constraints. However, these animations were only intended to represent the lives of characters at a specific age. Recreating the entire life course game in the animated style could possibly even reduce the playtime needed for it and could provide an even better experience for the player. There is, however, one serious argument against it, which is that the player will lose interactivity by not being able to progress through the presented ages using the “Next” button if that animation is presented as one continuous video. The “Next” button, however, can still be implemented, by

possibly creating a video player-like functionality with the possibility of rewinding the animations and stopping them at a certain point.

2. Reaching new platforms and devices

The WrightLIFE project can be expanded into being used on emerging metaverse technologies. Today, the world is witnessing a significant growth of interest in virtual and augmented reality. Virtual Reality (VR) and Augmented Reality (AR) technology is proving their effectiveness in experiential learning. VR already has seen practical applications related to treating anxiety, phobias, and psychological disorders. Moreover, research was done on integrating VR and AR technologies into the creation of immersive learning environments [42]. Thus, it can be beneficial to make the WrightLIFE project compatible with VR goggles and haptic devices.

5 CONCLUSION

In this thesis, it was shown that by using a variety of technologies, a user experience (UX) that contributes to the elimination of implicit bias in the healthcare setting can be created and presented to healthcare professionals. By addressing all essential aspects of a quality UX, it was possible to make important experiential learning tools for the “Wright LIFE (Lifelike Immersion for Equity)” project’s serious game simulations.

The life course game provided healthcare specialists with a visually striking and immersive experience. Multiple graphic design tools were utilized during the development process, and additional tools were used for integrating the UX into the Unity Engine. The resulting user experience was an effective visual aid for the provider to fully understand the impact of bias on the game characters, which later helps getting more extensive feedback from them. The data collection survey scenes were essential for encouraging the healthcare provider to share their experiences throughout various stages of the simulation. These experiences were collected and sent to the database to be stored for future research. The scenes’ outcome fulfilled the essential rules of a quality user experience and addressed all parts of the “goal - intention - function - action - confirmation - evaluation - feedback” design flow.

The scripts and assets created during the development could serve as a learning tool for software engineers and user experience designers who will work on similar projects. With the demand for serious games increasing, the developed tools within the project’s desktop and mobile applications can become a noticeable example of experiential learning in form of a serious game.

REFERENCES

- [1] “Experiential learning: Center for innovative teaching and learning,” Northern Illinois University, 2012. [Online]. Available: <https://www.niu.edu/citl/resources/guides/instructional-guide/experiential-learning.shtml>. [Accessed: 29-Nov-2022].
- [2] A. Abd-alrazaq, M. Alajlani, D. Alhuwail, C. T. Toro, A. Giannicchi, A. Ahmed, A. Makhoulf, and M. Househ, “The effectiveness and safety of serious games for improving cognitive abilities among elderly people with cognitive impairment: Systematic Review and meta-analysis,” *JMIR Serious games*, vol. 10, no. 1, 2022.
- [3] Djaouti, Damien & Alvarez, Julian & Jessel, Jean-Pierre & Rampnoux, Olivier. (2011). *Origins of Serious games*. 10.1007/978-1-4471-2161-9_3.
- [4] Kuindersma, Esther & Field, Joris & van der Pal, Jelke. (2015). *Game-Based Training for Airline Pilots*.
- [5] F. Laamarti, M. Eid, and A. El Saddik, “An overview of serious games,” *International Journal of Computer Games Technology*, vol. 2014, pp. 1–15, 2014.
- [6] Y. Wang, “Application of Serious Games in Health Care: Scoping Review and Bibliometric Analysis.” *Frontiers in Public Health*, vol. 10, 2022. 10.3389/fpubh.2022.896974.
- [7] I. V. Blair, J. F. Steiner, and E. P. Havranek, “Unconscious (implicit) bias and health disparities: Where do we go from here?,” *The Permanente Journal*, vol. 15, no. 2, pp. 71–78, 2011.
- [8] “Biases in healthcare: Types, examples, overcoming bias,” *Medical News Today*, 30-Aug-2021. [Online]. Available: <https://www.medicalnewstoday.com/articles/biases-in-healthcare>. [Accessed: 10-Nov-2022].
- [9] Birmingham, Elina, *Implicit Social Biases in People with Autism*. *Psychological Science*, vol. 26, no. 11, 2015, pp. 1693–1705. 10.1177/0956797615595607.
- [10] S. Fontenot, “Study: Reducing biases about autism may increase social inclusion,” *News Center*. [Online]. Available: <https://news.utdallas.edu/health-medicine/autism-biases-study-2021>. [Accessed: 10-Nov-2022].
- [11] E. Weir, C. Allison, and S. Baron-Cohen, “Autistic adults have poorer quality healthcare and worse health based on self-report data,” *Molecular Autism*, vol. 13, no. 1, 2022.

- [12] Taylor, Caitlin, "From Bias to Behavior: Predicting Prejudice Towards Autistic Individuals" (2018). Undergraduate Honors Theses. William & Mary. Paper 1203. [Online]. Available: <https://scholarworks.wm.edu/honorstheses/1203>. [Accessed: 10-Nov-2022].
- [13] K. K. Strunk and W. Takewell, "LGBT bias and discrimination: Occurrence, outcomes, and the impact of Policy Change," 2017.
- [14] M. Morris, "Training to reduce LGBTQ-related bias among medical, nursing, and dental students and providers: a systematic review." *BMC Medical Education*, vol. 19, no. 1, 2019, doi: 10.1186/s12909-019-1727-3.
- [15] P. N. Halkitis, "Discrimination and homophobia fuel the HIV epidemic in gay and bisexual men," American Psychological Association. [Online]. Available: <https://www.apa.org/pi/aids/resources/exchange/2012/04/discrimination-homophobia>. [Accessed: 10-Nov-2022].
- [16] P. J. Smit, M. Brady, M. Carter, R. Fernandes, L. Lamore, M. Meulbroek, M. Ohayon, T. Platteau, P. Rehberg, J. K. Rockstroh, and M. Thompson, "HIV-related stigma within communities of gay men: A literature review," *AIDS Care*, vol. 24, no. 4, pp. 405–412, 2011.
- [17] "The importance of UX Design," Rocket55, 07-Nov-2019. [Online]. Available: <https://www.rocket55.com/lab-note/the-importance-of-user-experience/>. [Accessed: 10-Nov-2022].
- [18] A. Darejeh and D. Singh, "A review on user interface design principles to increase software usability for users with less computer literacy," *Journal of Computer Science*, vol. 9, no. 11, pp. 1443–1450, 2013.
- [19] J. Moizer, J. Lean, E. Dell'Aquila, P. Walsh, A. (A. Keary, D. O'Byrne, A. Di Ferdinando, O. Miglino, R. Friedrich, R. Asperges, and L. S. Sica, "An approach to evaluating the user experience of serious games," *Computers & Education*, vol. 136, pp. 141–151, 2019.
- [20] A. Smyk, "Contextual UX. Building relevant and customized experiences," UX Collective, 23-Oct-2014. [Online]. Available: <https://uxdesign.cc/contextual-ux-5ef5db68d869>. [Accessed: 10-Nov-2022].
- [21] R. Roth, "User interface and user experience (UI/UX) design," *Geographic Information Science & Technology Body of Knowledge*, vol. 2017, no. Q2, 2017.
- [22] "What is a technology stack? Tech Stacks explained," MongoDB. [Online]. Available: <https://www.mongodb.com/basics/technology-stack>. [Accessed: 10-Nov-2022].

- [23] “Unity Overview,” Unity. [Online]. Available: <https://docs.unity3d.com/520/Documentation/Manual/UnityOverview.html>. [Accessed: 10-Nov-2022].
- [24] Microsoft, “Documentation for visual studio code,” RSS, 03-Nov-2021. [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed: 10-Nov-2022].
- [25] M. Sostero, “Git better: Collaborative project management using Git and GitHub,” matteosostero.com, Mar. 2018.
- [26] M. Levis, M. Brady, and M. Helfert, “Website design quality and form input validation: An empirical study on Irish corporate websites,” *Journal of Service Science and Management*, vol. 01, no. 01, pp. 91–100, 2008.
- [27] A. S. Gillis, “What is Rest API (restful API)?,” *SearchAppArchitecture*, 22-Sep-2020. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>. [Accessed: 10-Nov-2022].
- [28] Unity Technologies, “Important Classes - MonoBehaviour,” Unity, 2021. [Online]. Available: <https://docs.unity3d.com/Manual/class-MonoBehaviour.html>. [Accessed: 10-Nov-2022].
- [29] Unity Technologies, “Important Classes - GameObject,” Unity, 2021. [Online]. Available: <https://docs.unity3d.com/Manual/class-GameObject.html>. [Accessed: 10-Nov-2022].
- [30] “Lists and dictionaries,” *Unity Learn*, 2022. [Online]. Available: <https://learn.unity.com/tutorial/lists-and-dictionaries#>. [Accessed: 10-Nov-2022].
- [31] “Dropdown: Unity UI,” *Unity UI | 1.0.0*, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-Dropdown.html>. [Accessed: 10-Nov-2022].
- [32] “Toggle: Unity UI,” *Unity UI | 1.0.0*, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-Toggle.html>. [Accessed: 10-Nov-2022].
- [33] “Toggle Group: Unity UI,” *Unity UI | 1.0.0*, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-ToggleGroup.html>. [Accessed: 10-Nov-2022].
- [34] “Text: Unity UI,” *Unity UI | 1.0.0*, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-Text.html>. [Accessed: 10-Nov-2022].

- [35] M. Mohapatra, "DateTime in C#," C# Corner, 10-Mar-2021. [Online]. Available: <https://www.c-sharpcorner.com/article/datetime-in-c-sharp/>. [Accessed: 10-Nov-2022].
- [36] "What is Adobe Illustrator," Training and Classes at American Graphics Institute, 16-Sep-2022. [Online]. Available: <https://www.agitraining.com/adobe/illustrator/classes/what-is-adobe-illustrator>. [Accessed: 10-Nov-2022].
- [37] B. Kopf, "The power of Figma as a design tool," Toptal Design Blog, 31-Jul-2018. [Online]. Available: <https://www.toptal.com/designers/ui/figma-design-tool>. [Accessed: 10-Nov-2022].
- [38] "Do you want to know about us? - Freepik company.," Freepik Company, 2022. [Online]. Available: https://www.freepikcompany.com/about_us. [Accessed: 10-Nov-2022].
- [39] "What is Adobe Photoshop? - definition from Techopedia," Techopedia.com, 2022. [Online]. Available: <https://www.techopedia.com/definition/32364/adobe-photoshop>. [Accessed: 10-Nov-2022].
- [40] A. Parush, "Conceptual Design for interactive systems: Designing for performance and User Experience," UXmatters, 07-Dec-2015. [Online]. Available: <https://www.uxmatters.com/mt/archives/2015/12/conceptual-design-for-interactive-systems-designing-for-performance-and-user-experience.php>. [Accessed: 10-Nov-2022].
- [41] A. Aldandarawy, "Book - page curl: Animation tools," Unity Asset Store, 17-Feb-2021. [Online]. Available: <https://assetstore.unity.com/packages/tools/animation/book-page-curl-55588>. [Accessed: 10-Nov-2022].
- [42] A. Paíno Ambrosio and M. I. Rodríguez Fidalgo, "Past, present and future of virtual reality: Analysis of its technological variables and definitions," Culture & History Digital Journal, vol. 9, no. 1, p. 010, 2020.