Innovative & Immersive Fitness Experience

Project Report

Created For

Sense | Think | Act

Company's Contact: Farjana Salahudin | Khalid Al Ali

Advising Professor: Selma Limam Mansar

Course: Information System Consulting Project | Spring 2023

Team Members: Ahmed Issaoui | Jewel Dsouza | Mariam Al Thani | Raghad Abunabaa

Date: Thursday 13th April 2023

Carnegie Mellon University Qatar | STA Sense | Think | Act

Table of Contents

1. Executive Summary

1.1. Abstract

People often regard being active as a monotonous commitment that they must push themselves to complete as conventional techniques of being active do not intrigue and engage individuals, resulting in lack of enthusiasm and determination. The pleasure of staying active resides not merely in the final result, but also in the experience of becoming a better version of oneself. Therefore, we developed an innovative and immersive fitness experience utilizing advanced technology and creative designs using Python and TouchDesigner to create a distinctive workout experience. This experience primarily strives to boost engagement, offer a diverse range of exercises and settings, promote social connections and most importantly, urge healthy habits, as Qatar has recognized the widespread prevalence of sedentary lifestyles, obesity, and diabetes among its entire population.

1.2. Project Opportunity

STA seeks to further expand the scope of its creative department and promote immersive experiences with a social or public message in Qatar. It also aims to increase the organization's name and solution portfolio to a greater extent. Therefore, capitalizing on the opportunity to launch and ultimately increase awareness of the concept of immersive experiences in Qatar would offer significant benefits to and values to STA. This is mainly because delivering these new technologies and experiences to the still undernourished consumer market in Qatar will draw the market's attention to STA. In addition, one of the main findings of our initial research on market analysis concluded that fitness as a concept will be our stand-out message for us to promote. This is primarily due to the rampant spread of sedentary lifestyles, obesity, and diabetes across the entire population in Qatar. Therefore, we believe that an immersive experience tailored to Qatar's context, combined with a game to encourage users to minimize the use of handheld technologies, is what Qatar requires and ought to pursue.

1.3. Project Solution and Outcomes

To accommodate the company's needs, we developed a fun-based interactive experience that centers around the public message of encouraging people to become more active. The primary objective of this experience is to draw the interest of the audience to the unique means of preserving their health while also educating them about their health. Our solution is designed for anyone over the age of eight, and it enables everyone to become more conscious of their health and the techniques of remaining active.

The system will function by detecting peoples movements, through a camera, corresponding to obstacles and indicators, that is connected to a large screen, where users will have to avoid obstacles using exercise movements such as jumping to the right, left, a jumping jack and a crouch.

The experience will eventually display a leaderboard of people who have the highest score. By the end, we developed a functional Minimum Viable Product (MVP).

We anticipate that the system will adequately capitalize on the opportunity presented in the Qatari market for interactive experiences because:

- the public message is one that is not limited to a certain group of users,
- the experience does not require direct interaction with any technology,
- and the message is one that is applicable to Qatar's context.

The expected outcome of our technological solution is greater public awareness of the importance of being active, the range of opportunities, and technological advancements in the field of interactive experiences that are already available in the country.

1.4.Student Team

The student team, consisting of educated and driven individuals, were relocated together in order to tackle a challenging project, using their varied skills and experiences to develop an innovative solution. The following section includes brief information about each student member:

Mariam Al Thani is a third-year Information Systems student who is expected to graduate in May 2024. Mariam served as Project and Quality Assurance Manager. Mariam is passionate about using technology to create an impact on society and is dedicated to creating solutions that promote accessibility for all.

Ahmed Issaoui is a third year Information Systems student who is expected to graduate in May 2024. Ahmed served as the Chief Development Officer & Communicator. He strives to embark on a career related to data science, utilizing his skills to uncover creative solutions.

Jewel Dsouza is a fourth-year Information Systems student whose expected graduation is in May 2023. Jewel served as Chief Design Officer during the project. Jewel's professional goals are centered on applied informatics and is keen to use her expertise to bring value to the industry.

Raghad Abunabaa is a fourth-year Business Administration student with a minor in Information System whose expected graduation is in May 2023. Raghad served as Quality Assurance Manager. Raghad is eager to pursue a career in consulting to drive positive change.

2. Community Partner Background

"STA has deep experience developing mission critical hardware and software integrated solutions for government and military agencies. STA brings to bear many years of proven expertise in capturing and analyzing highly sensitive data, providing clients with state-of-the-art, efficient and rapid actionable intelligence." (STA, 2020)

Programs

STA has expertise in data fusion and mission-critical deployments, offering products and services such as aerial technologies, early warning systems, shoe scanning systems, automation, digitization in the oil and gas industry, and the creation of seamless critical hardware and software integration solutions for government and military organizations.

Organization Structure

STA is a startup company with five full-time employees, including the founder and CEO, a technological development specialist, part-time employees, and interns. They have served around seven clients in Qatar. The company is seeking to grow and diversify in other areas by expanding into a creative division, which is a unique contribution developed by Farjana Salahudin, a technological development specialist. Salahudin is now planning initiatives with the company's CEO to grow the organization. Thus, Farjana Salahudin was the key point of contact throughout the project's different phases.

Technology Infrastructure and Management

STA has a technological infrastructure with standard hardware and software with resources as required. This was irrelevant to our endeavor as creating an immersive experience is part of the newly formed division, so no collaboration or coordination with other organizational departments was necessary for project development.

3. About the Team

Mariam Al Thani is an Information Systems student with a minor in Professional Writing. She is proficient in various writing styles and can adapt to technical situations. Mariam played a dual role as Project Manager and Quality Assurance Manager. She contributed to the project by designing environments on TouchDesigner, developing experience features using Python, and ensuring that the team stayed on track throughout the different phases.

Ahmed Issaoui is an Information Systems student with extensive experience in problem-solving, critical thinking, and coding, especially in the fields of data science and artificial intelligence. He was the Chief Development Officer & Community Partner Communicator and

contributed significantly to the project by managing and organizing the software development, code framework, and technological developments on Python.

Jewel Dsouza is an Information Systems student who has developed design, research, and organization skills. She served as the Chief Design Officer, leveraging her past experience in VR development to create two environments, on TouchDesigner, for the project. Additionally, she made sure to record important minutes during team meetings.

Raghad Abunabaa is a Business Administration student with a minor in Information Systems. Through her past experiences, she has developed a wide range of skills, including communication and critical and analytical thinking. Raghad also served as a Quality Assurance Manager, managing documentation, training material, and screencasts, ensuring top project quality.

Through Ahmeds' extensive technical experience, Mariam's solid background in design and programming, Jewel's design-oriented skills, and Raghad's detail-oriented and professional skills, our skills complemented each other and the project well.

4. Community Partner Project Description

4.1. Project Opportunity

Value in Opportunity Pursual

STA aims to expand its creative portfolio by developing a touchless interactive experience that combines multiple sources of effects that remains rare in the country by having it function without users directly interacting with the technology, making interactiveness the selling point. This move is expected to catch the market's attention and showcase STA's technological adeptness, and understanding of the market. This will allow STA to invest in the new up-and-coming interactive technology and experiences market in Qatar (National Museum of Qatar, 2022; The Peninsula, 2016), leading to a wider range of projects under their belt.

Furthermore, the novelty of the touchless immersive experience is likely to attract potential investors and partners, while the addition of a social or public message shows other stakeholders a possible solution that can come out of working with STA.

Context & Experience Message

STA sought our input on a relevant public message for developing an experience. We suggested a fitness-centered experience, which aligned with Qatar's context and is a field yet delved in by STA. Qatar has recognized the prevalence of sedentary lifestyles, obesity, and diabetes in its population (Al-Thani et al., 2018; Chrismas et al., 2019; Cheema et al., 2022) and

has launched several initiatives to address these issues (Aspetar Orthopaedic and Sports Medicine Hospital, 2021; Weill Cornell Medicine, 2022). Thus, promoting fitness seemed appropriate.

We proposed an interactive, game-based experience with a competitive element through a leaderboard. This combination aimed to motivate users to minimize negative health impacts related to sedentary behavior and increased screen and technological use (Damato et al., 2022; Hu et al., 2021; Roman-Juan et al., 2022). The experience's message was specific to Qatar's context and past and current initiatives, making it more relevant to the target audience.

What was expected from us was a solution that aligned with STA's plans to expand its creative division, diversify its product portfolio, and create potential partnerships with external companies to develop interactive experiences. Subsequently allowing STA the ability to act upon the increasing demand and market for interactive experiences in Qatar.

We initially considered three potential projects.

Option 1

Our first consideration was an interactive experience on sustainability and pollution awareness where a world map displayed on a large screen would change colors and noises based on pollution levels, using motion detection through cameras. While the message aligned with Qatar's 2030 vision (Meza et al., 2019; Furlan & AL-Mohannadi, 2020), the project required consistent pollution data updates and complex technical development. We ultimately decided not to pursue it.

Option 2

An interactive experience that aimed to promote positivity and gratitude and would involve speech detection through microphones and a large screen that would change the ambiance based on the user's message. This experience would make a wide range of users more aware of the potential impact their words can have on themselves and others. However, the lack of resources and anticipated learning curve hindered our pursual of a speech-detection-based system. While the concept reflected STA in a brighter light and aligned well with Qatar's values, the implementation posed significant anticipated challenges.

Proposed Option

We proposed an interactive game-based and competitive experience that encourages physical activity through motion detection with cameras and a large screen. In the experience, the user is prompted to complete a series of upper and lower body movements such as jumping jacks, squatting, and jumping left and right to dodge obstacles to maintain their lives and get higher scores. When the game ends, a leaderboard with the highest scores is presented on the screen. A performance indicator is present while the game is ongoing, displaying the user's score. A

movement indicator also informs the user what obstacle is coming up. Additionally, the experience is equipped with background music and a collection of environments that randomly alter depending on the game session.

To make the solution more appealing to a wider set of users, the difficulty level increases based on the score. Our target group of users is people over 8 years old, but this experience is playable for all.

The company's interest in expanding their portfolio and showcasing technological advancements further supports the pursuit of this concept with a wide range of target users.

This project was chosen for its strong narrative, feasibility, and potential user interest.

Strength of the narrative

The project's narrative is strong due to its clear and simple public message and the fun competitive experience that is expected to generate user interest and participation.

Thus, our vision statement for this project would be to *encourage people in Qatar to become more active through multiple innovative immersive technologies.*

5. Project Solution and Outcomes

5.1. Project Outcomes

The immersive experience developed for STA is unique in nature as it is an area that STA, nor anyone in Qatar, has explicitly dwelled into before. In order to deliver the immersive fitness experience, the main tools used to develop and design our experience was Python and TouchDesigner. Through the use of Python, a key part of the game mechanics was able to be developed, which was the motion detection. In addition to that, other game features that were added were the obstacles and collision zones (Refer to Appendix C) that allowed the user to not only interact with the game, but also remain engaged. TouchDesigner was an additional tool that was integral in ensuring that our experience was immersive. It allowed us to create an infinite space and 3D dimensional backgrounds of three landscapes showcasing different times of day. Through the use of Python and TouchDesigner, the client has many additional capabilities that can be implemented if they wish to do so, in regards to game mechanics and the design of the experience environment.

Based on our choice of using Python and TouchDesigner as our software tools, the three outcomes or parts of our experience can be divided into the following:

1. **Warm-up Tutorial**: Through the warm-up tutorial in our experience, users will be able to follow a set of onscreen instructions in order to learn the different movements necessary to

avoid each type of obstacle. Only once the user has successfully completed the tutorial will they be able to proceed to the main game.(Refer to Appendix C)

2. **Main Game**: At this stage, the user will be able to earn points the longer they are able to stay in the game and avoid obstacles. The user starts off with three lives which are lost each time an obstacle is hit. (Refer to Appendix C)

3. **Leaderboard**: The experience concludes by showcasing a leaderboard of the top scores. If the user playing has earned a high enough score, the leaderboard is updated with his username and score.

In order to use Python for motion detection, Ahmed, our Chief Development Officer, made use of additional open source content, such as Mediapipe and Pygame libraries, to ensure that our experience is able to both detect movement, and have the capabilities to have game mechanics added onto it. Our experience generates user content while the game is running, which are the top user scores stored in the leaderboard. This data is stored in a CSV file, and displayed at the end of the game. This is a sustainable way to showcase the data, as it does not require a lot of storage and therefore can have multiple people pass by and immerse themselves in the experience.

In preparation to receive the immersive experience, STA has disclosed and finalized the legalities and the ownership of the system through an Intellectual Property Agreement (Refer to Appendix D) that highlights that all intellectual property belongs to STA. Additionally, throughout our weekly Community Partner meetings, our client has played a major role in understanding the reasoning behind our software and hardware selection, in addition to the algorithms being used to develop and deliver the experience.

Prior to the development of the project, most of the methods used to remain fit were stuck to traditional methods of exercising. Through the development of our project for STA, we aimed to deliver a fitness experience using technology to immerse a broader audience of eight years old and above. Previous fitness experiences using technology would require the user to directly interact with the technology. However, our fitness experience makes fitness more interactive and engaging through the use of a game in which the user does not need to physically interact with technology.

5.2. Solution Description

5.2.1. Users and Requirements

In our project, user segmentation is not required as the functionality and experience will remain consistent across all users. Therefore, we do not have different user groups.

Functional requirements

Requirement	Description	Priority	New functionality?	Implemented? / Justification	
GM1	The game should have a character that should be in a running motion, facing obstacles throughout the game	Must	Must	Completed / Altered	Initially we had a character with a running motion but after conducting Usability testing and communicating with our client and advisors, we opted for a more reflective character instead. So, this requirement is no longer applicable or appropriate for our experience.
GM2	The game should include a variety of obstacles that can be avoided using movements corresponding to up, right, left, and down inputs.	Must	Original	Completed	
GM3	The user should be able to view an indication of the movement they should do to avoid	Should	Original	Completed	

	the upcoming obstacle				
GM4	The score should increase by one each second during the game	Must	Original	Completed	
GM5	The score should increase by five each time an obstacle is avoided	Must	Original	Completed	
GM6	The score should increase by an additional five points for each subsequent obstacle avoided within 1 second of the previous one	Must	Original	Won't	This requirement is not implementable because after further discussions with the client and realizing the experience and it's status, we opted for a more straightforward score mechanism to ensure user understanding and agreed that this should no longer be a priority or under the 'must' requirements and instead should be a 'could'.

GM7	The character should have three lives	Must	Original	Completed	
GM8	A life should be consumed once the character touches an obstacle	Must	Original	Completed	
GM9	The game should end once all lives are consumed	Must	Original	Completed	
GM10	A user should be able to restart the game once it is ended	Must	Original	Completed	
GM11	A user should be able to pause and resume the game at any time.	Should	Original	Won't	After discussing between the members, our client, and gaining her approval - we decided that this requirement is not implementable as it will take away from the fun and immersiveness of the experience and will result in a more complicated system which will affect the

					overall quality of our development.
GM12	A user should be able to make the character jump by performing a jumping jack, move the character to the left, move the character to the right, and move the character down by performing a squat.	Must	Original	Completed	
GM13	A user should be able to view their score and track their progress throughout the game	Must	Original	Completed	
GF1	A user should be able to learn how to control the character's movement	Must	Original	Completed	

	through a tutorial				
GF2	A user should be able to go through a warm-up session before starting the game	Should	Original	Completed	
GF3	A user should be able to view an indication of how correct the movement they are doing compared to the expected movement	Should	Original	Won't	We instead opted for a textual indicator rather than a visual one. Additionally,due to concerns about the ability to complete a 'correctness' functionality in time, this requirement was not implementable.
GF4	A user should be able to view a leaderboard that shows the top scores of all players	Must	Original	Completed	
GF6	A user should be able to select different difficulty levels and environments	Should	Original	Completed / Altered	This requirement was modified after discussions with the client and is now presented through removal of the selection option - the difficulty

					automatically increases and environments are randomly presented with every game session or restart.
GF7	A user should be able to collect coins, power-ups, and other items while running.	Could	Original	Won't	This requirement is not implementable as, due to complications and extensive conversations with the CP on the presentation of the obstacle, character, and environment, no time was left.
GF8	A user should be able to use the collected items to buy upgrades and unlock new characters, levels, and environments.	Could	Original	Won't	Upon further development of the project, we identified that this requirement conflicts with Non-functional requiremen S3 as it would mean that other attributes about the user should be stored, likely in a database. This is a big aspect of our project that we did not initially prioritize and thus have concerns on the time constraints, making it not implementable.

GF9	A user should be able to control the game's sound and music volume	Should	Original	Won't	After discussing between the members, our client, and gaining her approval - we decided that this requirement is not implementable as it would result in a more complicated system which we did not have time for and thus would have affected the overall quality of our development.
CR1	A username should be randomly generated once a user starts the experience, so that their top score is stored.	Must	Original	Completed	

Non-functional requirements

Requirement	Description	Priority	New functionality?	Implemented? / Justification	
OP1	The system will run on 65" large screens connected with cameras for	Must	Original	Completed	

	motion detection.				
OP2	The system should be able to run on any computer with at least a 6GB GPU.	Should	Original	Completed	
OP3	The system should connect to the large screen in different environments.	Should	Original	Completed	
OP4	The system could operate in an enclosed room and open space.	Could	Original	Completed	
P1	The system will detect a person's movement in under 5 seconds.	Must	Original	Completed	
P2	The system will respond to a person's movement in under 5 seconds.	Must	Original	Completed	
P3	The system should be	Should	Original	In-Progress	This requirement has a status of 'in-progress' because

	playable at least 23 hours a day.				while, technically, it is completed and theoretically, the experience should be able to run forever, this has yet to be tested.
P4	To minimize energy consumption, the system could switch to 'sleep mode' if the motion detection has not been activated in 3 hours.	Could	Original	Won't	
P5	The system will not detect more than one users movement at a time.	Wont	Original	Completed	
S1	The system will only have edit access for technicians.	Must	Original	Completed	

5.2.2. Design

Architecture Design

The architectural pattern used in our solution is the blackboard pattern as it is the most suitable to our project due to the necessity of components working and responding closely together e.g. the

motion detection with the obstacle collision detection and score awarding. The different components of the pattern are applied to our project as follows:

- Blackboard - is the leaderboard (and the screen itself) as this is where the username and score are stored and displayed.
- Knowledge Source - is where the motion detection information from the player is stored and the leaderboard (or screen) are updated.
- Control Component - assesses the motion detection information and awards points or reduces lives accordingly.

Additionally, the knowledge source stores information on the environment, the obstacle images, and the text colors. This is later accessed by the control component as the experience develops.

In our project, there are no external interfaces to other systems.

Data Design

The data in our system is primarily real-time (e.g. motion-detection) and not stored, with the exception of the leaderboard of scores and associated usernames, which are stored on an external CSV file updated automatically during system operation. For further & visual representation, refer to the relational schema in the Appendix.

UI Design

Character

Our system has one user - the player. We decided on a genderless character to represent the large user base of varying ages, heights, and genders. The final character, Figure 1, is reflective and based on the user's body landmarks and was developed through usability testing and discussions with the client.

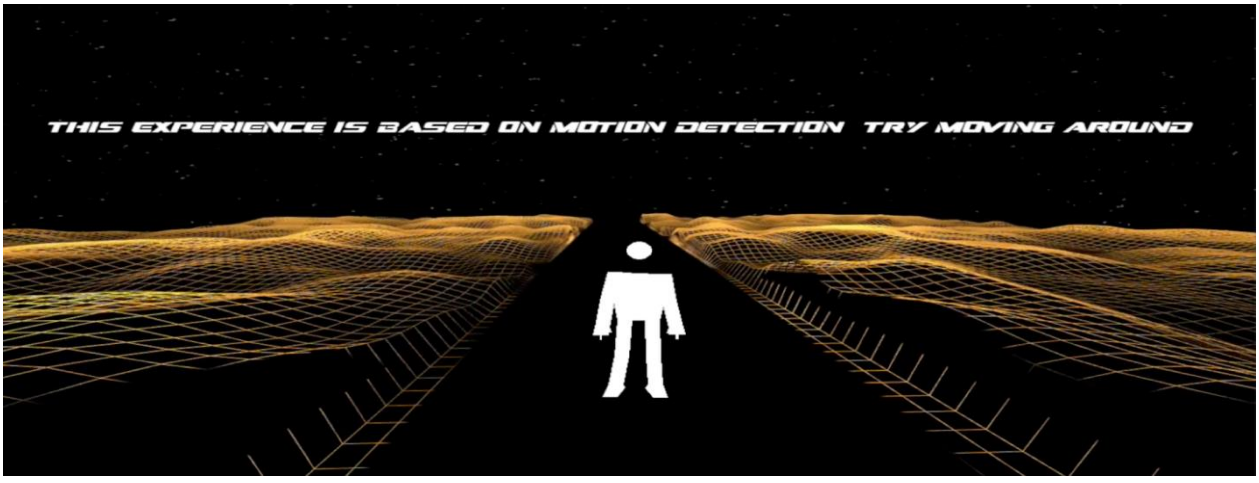


Figure 1: *The final reflective character in the experience*

Environments

Our system has three randomly presented environments that represent different regions of the world and times of day. The three environments are:

- Mountain - Day (Figure 2)
- Ocean - Sunset (Figure 3)
- Desert - Night (Figure 4)

The environments are abstract to suit the typical aesthetics of immersive experiences and are infinite space to enhance user immserveness. We also decided on having multiple and regional environments to maintain user interest after playing the game once, and to make it a unique selling point.

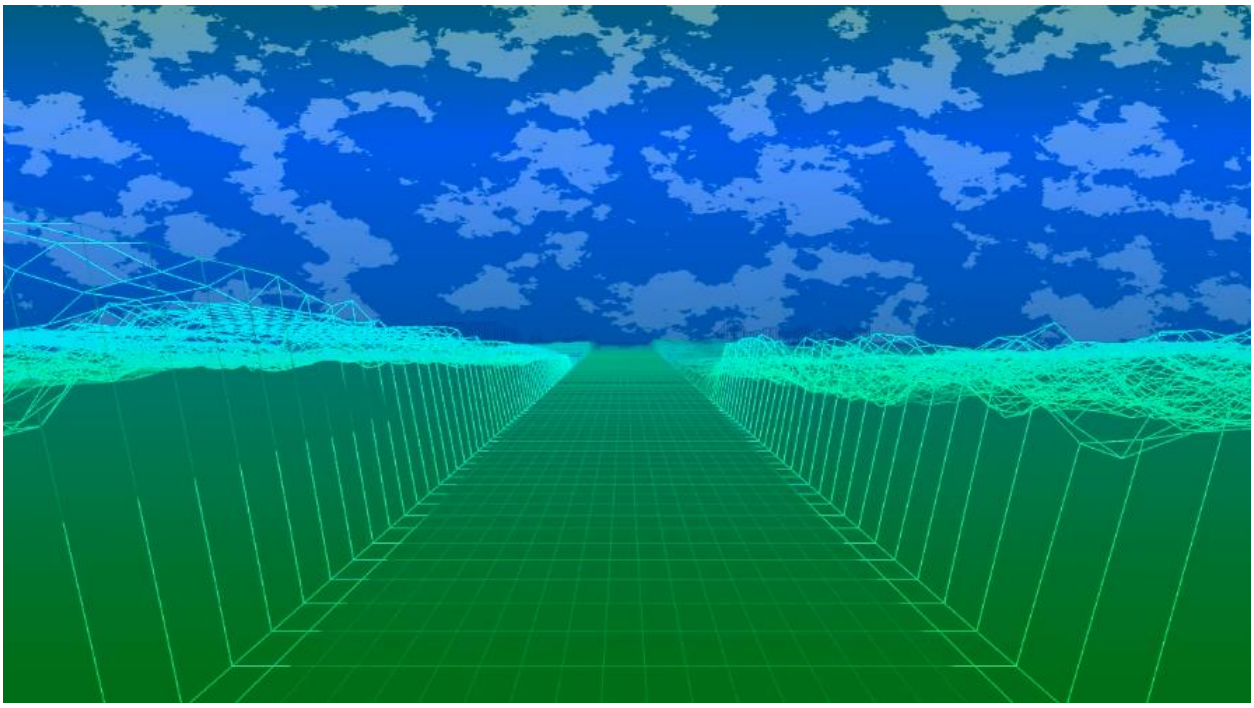


Figure 2: *A daytime mountain landscape, one of the random environments*

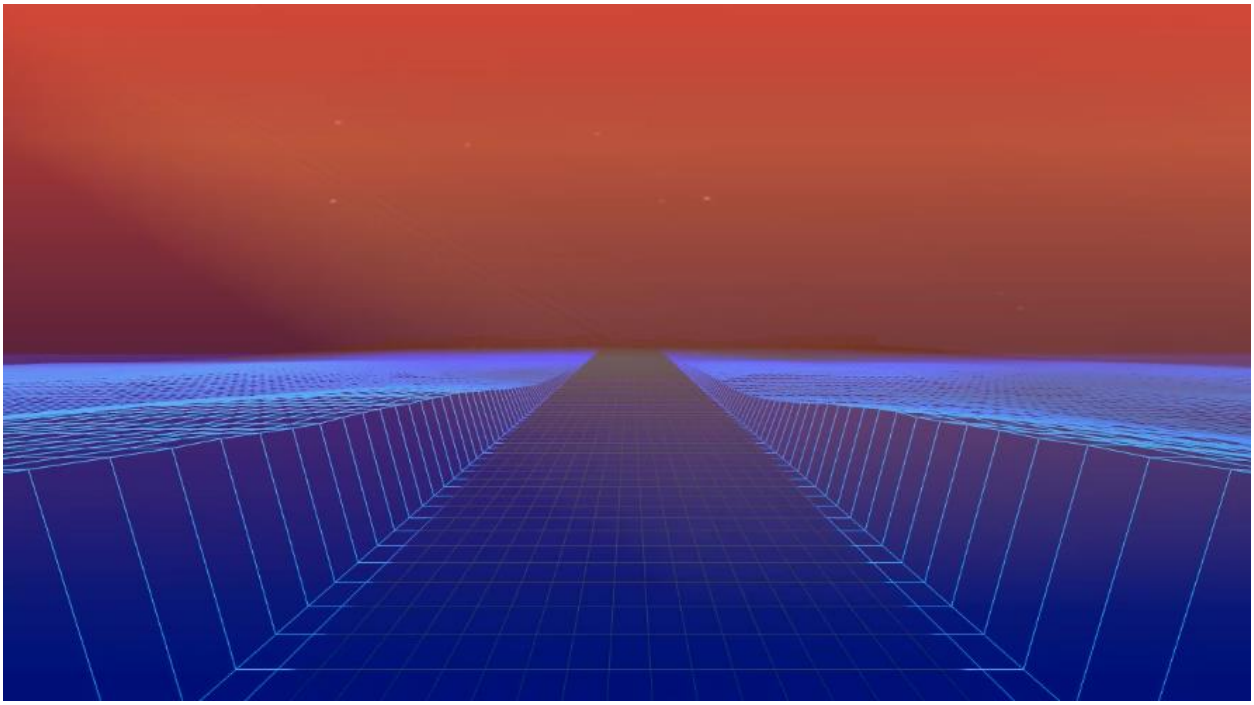


Figure 3: *A sunset ocean landscape, one of the random environments.*

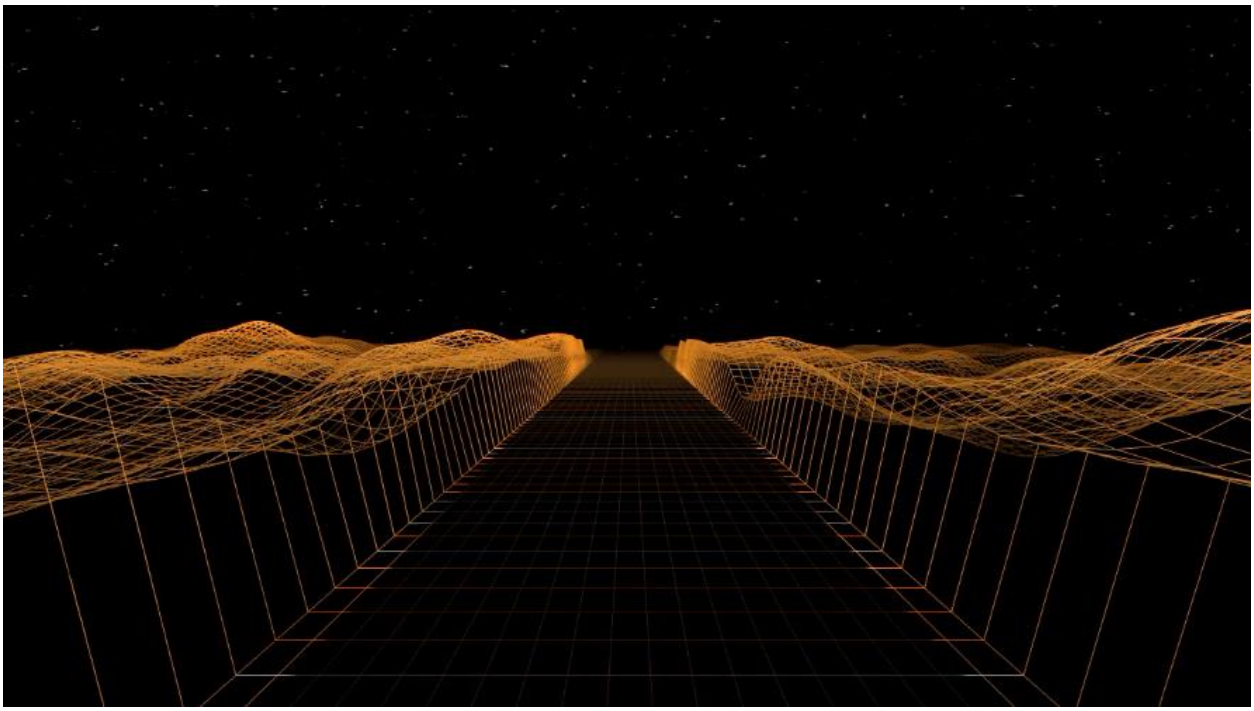


Figure 4: *A nighttime desert landscape, one of the random environments.*

Obstacles

Obstacles in the system are based on the exercises move, jump, and squat. Initially they were colored rectangles but further discussions with related parties and developments resulted in obstacles that are specific and image based, changing based on the environment (Figure 5-11).

Figure 5-11: *Collection of obstacles in the experience.*

Font

To suit the abstract aesthetic, we decided to use a digital and futuristic looking font.

Figure 12: *The font utilized in our experience.*

Physical

In the physical world, around the set-up of the experience, we decided to include a 'stand here' sticker on the floor to catch the users attention and to ensure they are standing in the right place. Additionally, we decided to include tape on the floor to indicate the borders of the screen, so when the user is playing, they know their limits easily.

Figure 13: *The stand here sticker placed on the floor where the experience will be set.*

5.2.3. Implementation

Python

To develop our solution, we utilized Python as the primary programming language, along with several technologies and tools to build different components, including motion detection, character animation, collision detection with obstacles, tutorial, leaderboard, and reset functionalities.

We leveraged Mediapipe, an open-source machine learning framework, to implement motion detection. This framework offers real-time hand, face, and pose recognition capabilities, enabling us to detect and track human movements accurately.

As for OpenCV, we utilized it to process the images of the obstacles in the game. Instead of using simple Pygame rectangles for the obstacles, we opted to display images, which offered a more engaging and visually appealing gameplay experience. OpenCV allowed us to manipulate the images` dimensions and resize them to give the illusion of 3-dimensional depth.

Furthermore, we also used OpenCV to process the frames in the video captured from the camera, which we fed to Mediapipe for motion detection. OpenCV provided us with a range of tools for video processing, such as video capture, and frame manipulation which were crucial in developing the motion detection component.

Overall, our use of OpenCV, in conjunction with Mediapipe and Pygame, enabled us to create a more immersive and interactive gaming experience with robust motion detection, collision detection, and obstacle display functionality.

To implement the game mechanics, we utilized Pygame, a cross-platform game development library that provides a comprehensive set of tools for building 2D games. This tool allowed us to introduce background music and create characters, obstacles, and game logic, such as collision detection and scorekeeping.

Overall, our use of Python and these technologies and tools allowed us to create a robust and dynamic solution that offers motion detection, gaming mechanics, tutorial, and leaderboard functionality.

Figure 14*: Leaderboard and reset functionalities*

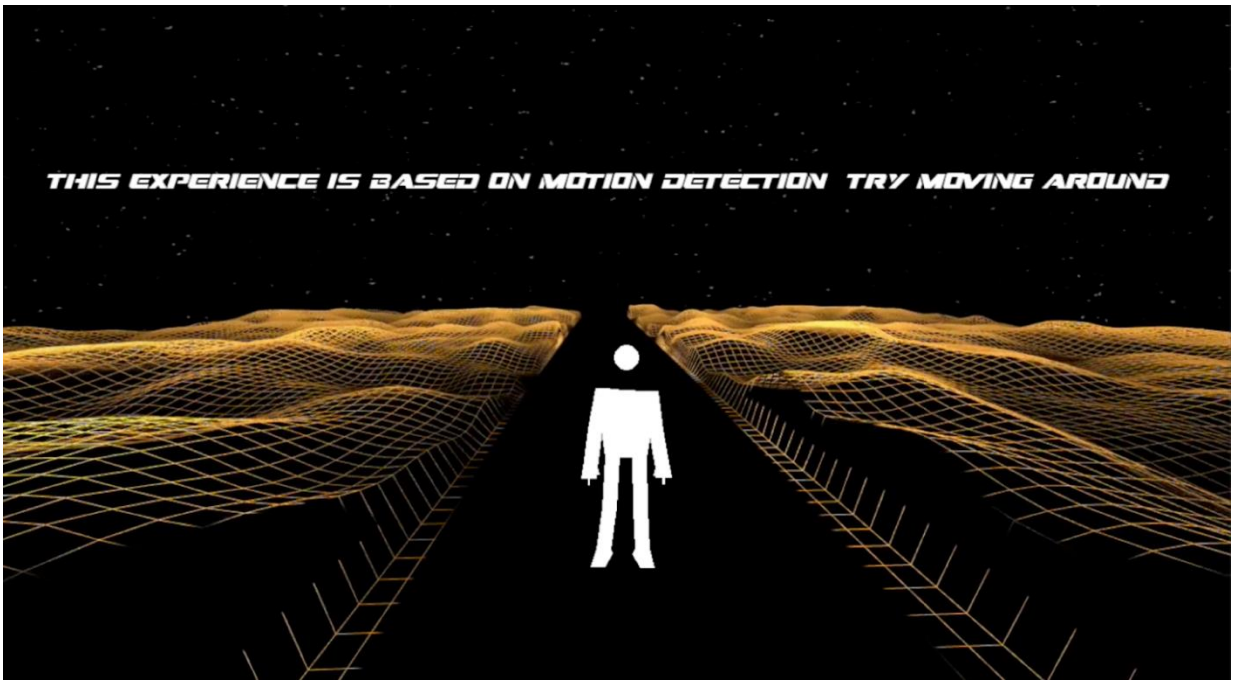


Figure 15: *Motion Detection functionality*

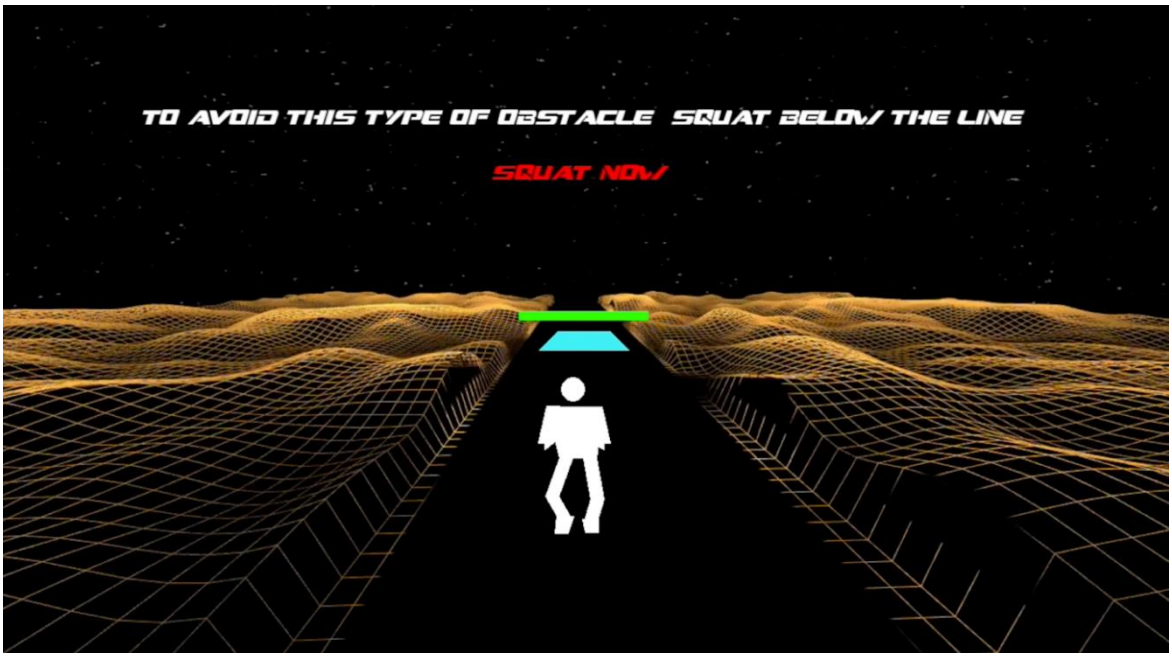


Figure 16: *Collision detection functionality*

TouchDesigner

The three environments (Figure 2-4) were created using a node-flow software, TouchDesigner, to expand its sophistication level. To integrate the environments with the game and immersiveness concept, the environments were designed to be infinite space, abstract, and with a path in the center.

5.3. Final Deliverables

This section presents an exhaustive outline of the work accomplished and the results achieved throughout the project. It serves as an important source of information for stakeholders who anticipate to proceed with the project in the future. The final deliverables include the source code, touchdesigner files, CSV file, background music used for the experience, three separate videos of the three different environments developed and multiple images of the different obstacles used for the experience. In addition, the final deliverable will also include a video demo of how to use the experience, testing methodologies and results, documentation and training material such as UX design and source code documentation, as well as an environment integration manual. The following table better describes the final deliverables included:

Item	**Description**
Source Code	This file contains the programming code developed to build the immersive fitness experience, all in Python. This includes constructing motion detection, collision detection, tutorials and a leaderboard.
TouchDesigner Files	The TouchDesigner files are divided into three components, each of which reflecting a distinct environment designed for the immersive fitness experience. These files are identified by the setting that they depict and were most likely developed with the TouchDesigner software, which is commonly used for creating real-time audiovisual content and graphical programming.
CSV File	The CSV file, which must be in the same directory as the source code, serves as a storage of the data required to generate the leaderboard displayed at the end of the experience. This file provides tabular data, with each row reflecting a player's random generated name and score.
Background Music File	The background music file has been used in the experience to make it more engaging for users.
Environments videos	The environment video clip is a three-part final product of TouchDesigner where the environments have been developed for the experience. These have been exported and looped to create the sense of an infinite space.

Images	The image files depict the numerous barriers used in each setting and are carefully selected depending on the exercise movement and environment design.
Video Demo	The video demo will show how to move around the virtual world, interact with the obstacles, and do the workout exercises step by step. It will also emphasize the system's numerous characteristics and functions while illustrating its simplicity of use and user advantages.
Testing methodologies and results	The system's testing procedures included manual testing and usability testing, which assisted in identifying bugs and usability concerns as well as gathering user input. Based on user input, the findings of these tests were utilized to improve the system, including addressing problems, improving usability, and introducing new features, leading to a more polished and user-friendly experience.
Documentation and training material	The documentation and training materials include an environment integration manual that describes the steps of incorporating the simulated setting into the experience, source code documentation containing comprehensive details about the code, and UX design documentation that outlines the user experience design and guidelines for the system. These resources are critical for properly comprehending and using the system, in addition to continuing advancement and upkeep.

All these final deliverables will be in one folder that will be shared and will be uploaded on both github, as well as on the shared folder with STA on Google Drive. In order to access the deliverables from the Google Drive:

1. Accept the invitation to contribute to the shared folder (already has been done):

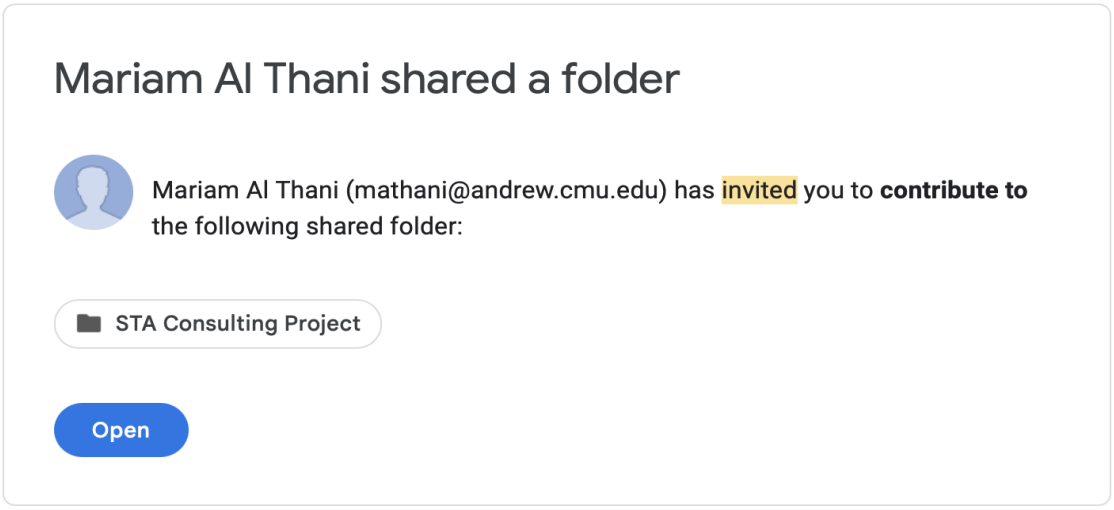


Figure 17: *Google Drive Invitation*

2. If already accepted the invitation, copy/click on the link below:

   https://drive.google.com/drive/folders/1zg5XNxymApSri4YrBlIjFmIwXtMXEP9H

3. Double click on the file named "Final Deliverables":

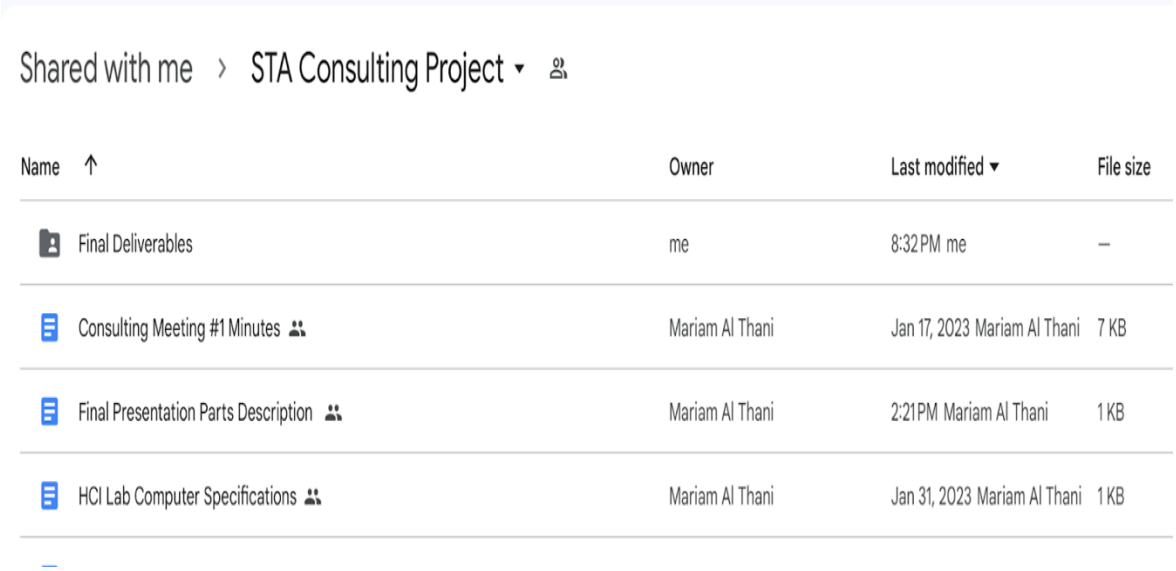


Figure 18: *Shared Folder Documents*

Following the above-mentioned steps, the user will be able to view all the final deliverables mentioned above.

To access the final deliverables on Github:

1. View the invitation that is sent via email:

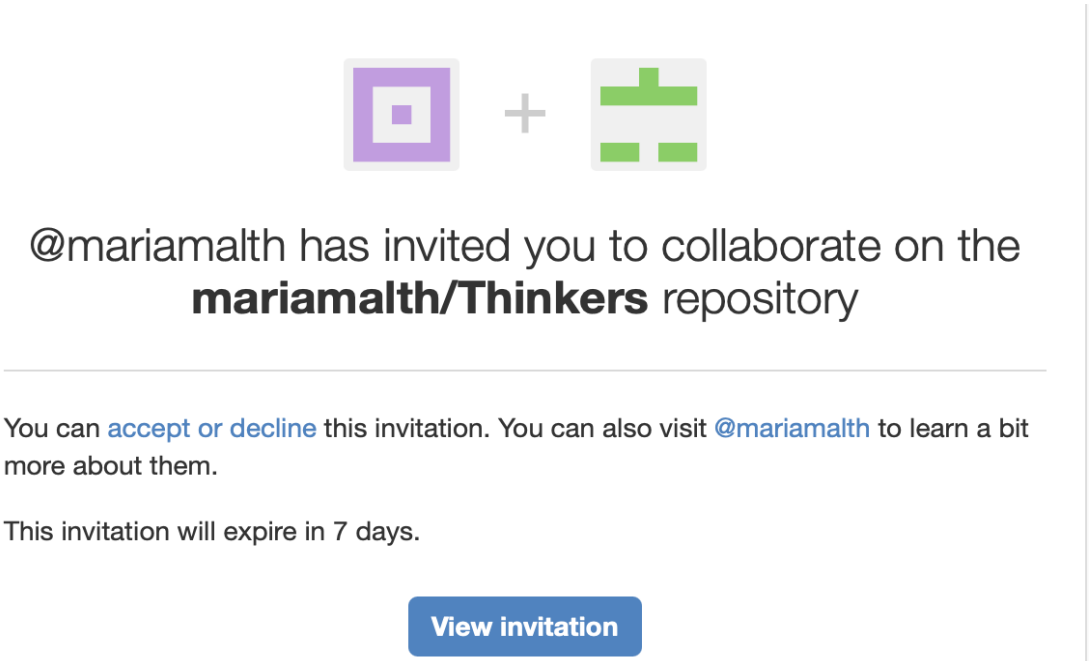


Figure 18: *GitHub Invitation*

Once the user accepts the invitation, they will be able to view all the files included on GitHub.

6. Testing

6.1. Software Testing

Due to the motion-detection aspect and lack of direct interaction between the user and the technology in our experience, we opted to have all of the testing conducted be manual testing to ensure the smoothest experience for users, by checking it first-hand, over and over again.

Our software testing focused on ensuring that our functional requirements, especially ensuring that the core game functionalities were not only functioning correctly for a variety of scenarios but sufficiently immune to user error, an aspect that intertwines with our usability testing.

The functionalities we focused on were:

- Correct motion-detection of user landmarks,
- Accurate reflection of these landmarks moving through the character,
- Accurate detection of obstacle collision for move, squat, and jumping obstacles.

- Consistent and correct updating through data storage mechanisms and subsequently, the game leaderboard.

This form of testing required extensive trial and error as we made real-time changes to the code and continued to test through various user scenarios (Appendix) until the functionality was deemed as functioning and bug-free. So, many of our testing sessions consisted of multiple team members, one testing the experience through being the user, one noting the fail and success cases, all theorizing on how to solve the problem or what is causing it, and one fixing the code.

Below is a table highlighting all our software testing activities and results.

TC_ID	Summary	Type	Automated	Tester	Test date	Results
001	Motion Detection	UT	Manual	Ahmed	19.02.23	Success
001	Motion Detection (changed character)	UT	Manual	Ahmed, Mariam	02.03.23	Success
001	Motion Detection (changed character)	UT	Manual	Ahmed, Raghad	19.03.23	Success
001	Motion Detection (changed character)	UT	Manual	Ahmed, Mariam	27.03.23	Success
002	Game start	UT	Manual	Ahmed, Jewel	19.02.23	Success
003	Obstacle avoidance	UT	Manual	Ahmed, Jewel	19.02.23	Success
004	Obstacle Collision	UT	Manual	Ahmed, Mariam	16.02.23	Success
004	Obstacle Collision	UT	Manual	Ahmed, Mariam	02.03.23	Success
004	Obstacle Collision	UT	Manual	Ahmed, Mariam	19.03.23	Failure
004	Obstacle Collision	UT	Manual	Ahmed, Mariam	10.04.23	Success

005	Username generation	UT	Manual	Mariam	27.02.23	Success
006	Motion detection integration	IT	Manual	Raghad, Ahmed	19.02.23	Success
006	Motion detection integration	IT	Manual	Raghad, Ahmed	19.03.23	Success
008	Tutorial	IT	Manual	Mariam, Ahmed	20.03.23	Success
009	Game reset	IT	Manual	Mariam, Ahmed	20.03.23	Success
008	Tutorial	IT	Manual	Mariam, Ahmed	10.04.23	Success
007	Immune to improper actions	ST	Manual	Raghad, Jewel	16.03.23	Success

While the majority of our testing sessions were successful from the start, we faced some major issues throughout testing and development. Our main issue was that, as we continued to develop our character and the obstacles, there were times where the system would fail to detect a character colliding with an obstacle. We dealt with this by integrating print statements throughout the section of code to identify the issue, highlighting the rectangles that are created from the obstacle and the character, and continuously trying each obstacle (move, jump, and crouch) to ensure they all can detect collisions (See Appendix).

Additionally, one constraint of having manual based testing was that, as we were going through the experience each time, it was incredibly time and energy consuming.

6.2. Usability Testing

For usability testing, we utilized the think-aloud protocol and decided to make our testing sessions in person and semi-moderated. Considering the context of our project, it was most logical to make these sessions in person as the experience needed to be set up with the cameras and large screen and involves interaction with the physical and digital space. As for moderation we believed it was especially imperative for our project that we directly witness how the user is responding to the experience, any subtle changes in expressions, reactions, or body language, allowing us to be able to notice a pattern more clearly. This also gave us the chance to see how users responded in situations where the motion detection is slow or inaccurate, if the situation arises.

In our testing sessions, we were monitoring how intuitive the system was and how comfortably the user can achieve the shared objective of playing the game. Specifically, we were looking at:

- the accuracy and responsiveness of the motion detection,
- the smoothness of the experience e.g. environment background, movement, obstacles,
- the visibility of the text with the background,
- the users' understanding of the game mechanics.

We conducted both usability and user acceptance testing in the same session.

Throughout the development of our experience, we conducted usability testing with four users in some of our different user groups (age, gender, height, interest in fitness). This includes our client (Farjana).

Name	**Description**
Faisal Mashhadi	• University Student • Male • Not greatly interested in sports
Nawaf Al Thani	• University Student • Male • Interested in sports & fitness
Christina Atat	• University Student • Female • Not greatly interested in sports
Farjana Salahuddin	• Technological Development Specialist

	• Female • Sports interest unknown

In these sessions, we gave the users the simple task of just starting and playing the game for as long as they can (until they lose) and speaking out loud their thought process. In later iterations of usability testing, users were also given the option to reset the experience after they lost.

As we were dually focusing on user acceptance, we aimed to have minimum interactions and interruptions from our end while the user was undergoing the experience. To mimic as realistic of an experience, we did not provide any guidelines either.

After the session, to gain insights explicitly from the users themselves, we asked them to share:

- How they felt playing the experience,
- What they liked most,
- What suggestions they have.

The following are the results from our usability testing and the actions we took:

User	**Results**	**Actions**
Faisal Mashhadi	• Thought it was an interesting experience, • Was fascinated with the motion-detection element, • Preferred a more reflective character over the running motion character (Figure 1) • Struggled in remembering which obstacles corresponded to which exercise.	• changed the character to a more reflective one, • included a portion in the tutorial that explains each obstacle and what action corresponds to it.

Nawaf Al Thani	• Enjoyed the experience. • Struggled in understanding the dotted character (Figure 20.4) represented him. • Thought the experience was not too intuitive and needed time understanding the concept.	• connected the landmarks between the reflective character to create a more understandable figure (figure 1), • expanded the tutorial length and slowed the tutorial pace.
Christina Atat	• Greatly enjoyed the experience. • Liked the reflective character. • Thought the character was too large and so could not see the obstacles clearly. • Struggled in remembering what each obstacle meant.	• made the character smaller, • included a portion in the tutorial that shows all the obstacle images for that respective environment.
Farjana Salahuddin	• Enjoyed the concept of the experience. • Thought the experience was moving too fast from the tutorial and did not understand what point in the experience she was on / what she needed to do. • Had difficulties with the clarity of the text and font. • Preferred a more abstract environment	• expanded the tutorial and slowed the pace of the tutorial, • changed the font color and choice to a clearer one, • changed the environment to a more abstract one.

Throughout project development, our UI design entirely changed, multiple times.

Experience Design

Figure 19 shows a low-fidelity storyboard of our experience and the major sections of it. Considering we highlighted no issues from the users on the actual sectioning of the experience, it remained the same (Welcome → Tutorial→ Game → Leaderboard).

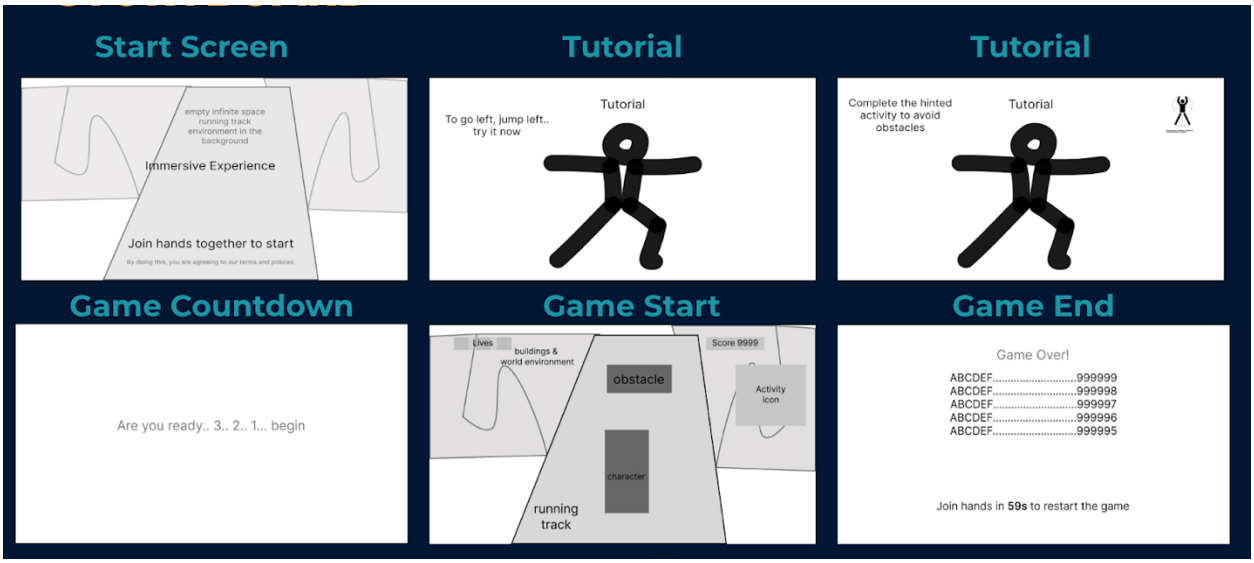


Figure 19: *A low-fidelity storyboard of our experience and the major sections of it.*

Character Design

Evidently, our character design underwent numerous iterations after discussions with the client and usability testing results. Initially, (Figure 20.1) we had a rectangle while we developed other functionalities of the system. Moving forward, we had static characters in a running motion (20.2). Aiming for a more reflective character, we had a stick figure which could move its arms(20.3). Even more reflective, we created a character which was a culmination of dots (20.4). Lastly, our final character (Figure 16) was the output of all iterations and usability feedback.

Figure 20.1: *Our first iteration of our character design - a rectangle.*

Figure 20.2: *Our second iteration of our character design - a 'gif' of a running motion character.*

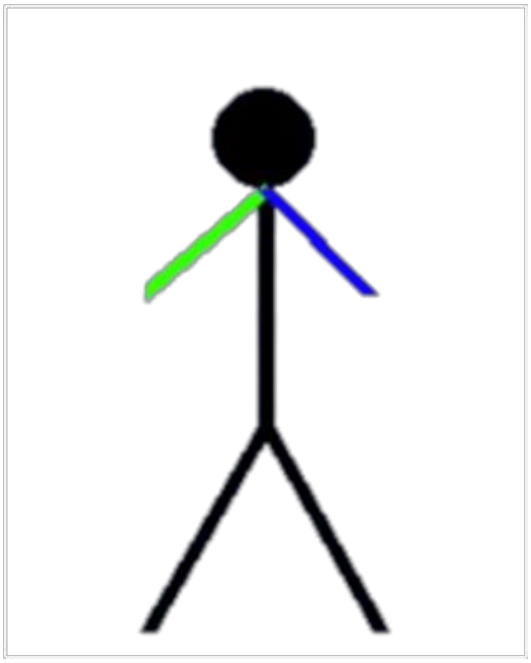


Figure 20.3: *Our third iteration of our character design - a stick figure with movable arms.*

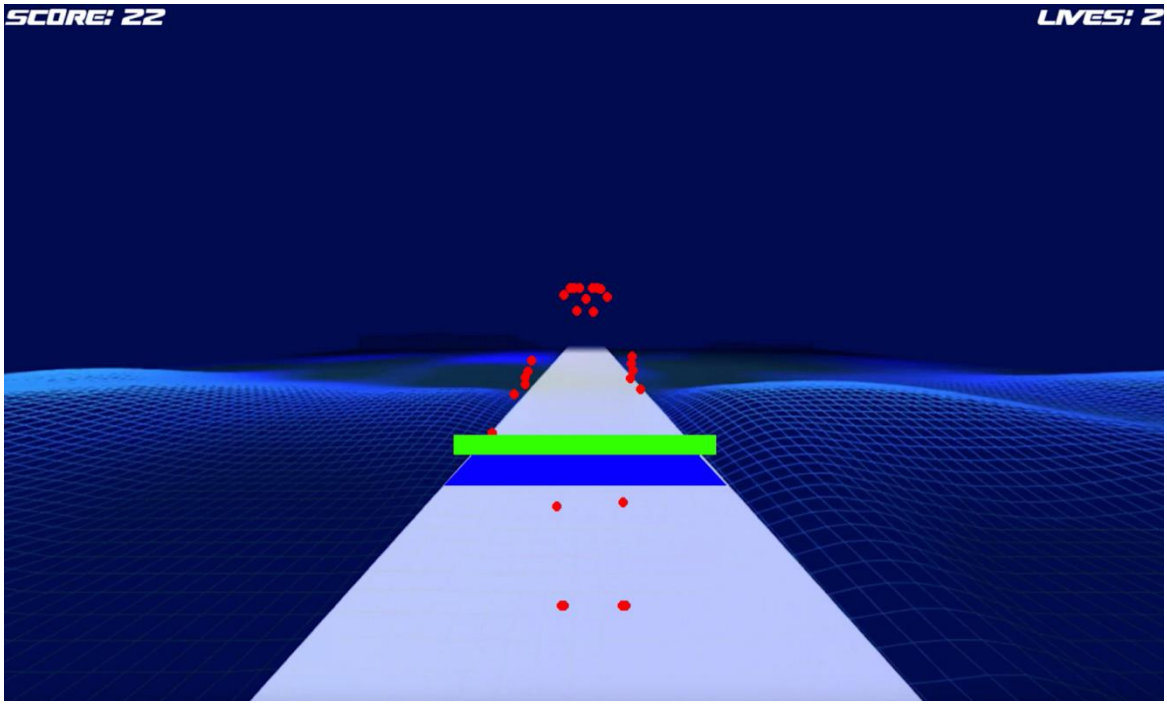


Figure 20.4: *Our penultimate iteration of our character design - a series of landmark dots.*

Environment Design

Initially, we planned to have the environment(s) represent different parts of Qatar with a path running through it (Figure 21). After discussions with our client on the choice of software and technical capabilities in the timeframe, we agreed that a more abstract environment would be suitable. So, we created an infinite space environment of moving buildings (that resembled the Old Doha Port) (Figure 22.1). Usability testing with our client and researching different potential environment designs led to an even more abstract environment (Figure 22.2). We continued to

develop our environments' designs, making them more sophisticated per our usability feedback from our client, especially with her experience in similar projects.

Figure 21: *Initial prototype of the experiences' environment design*.

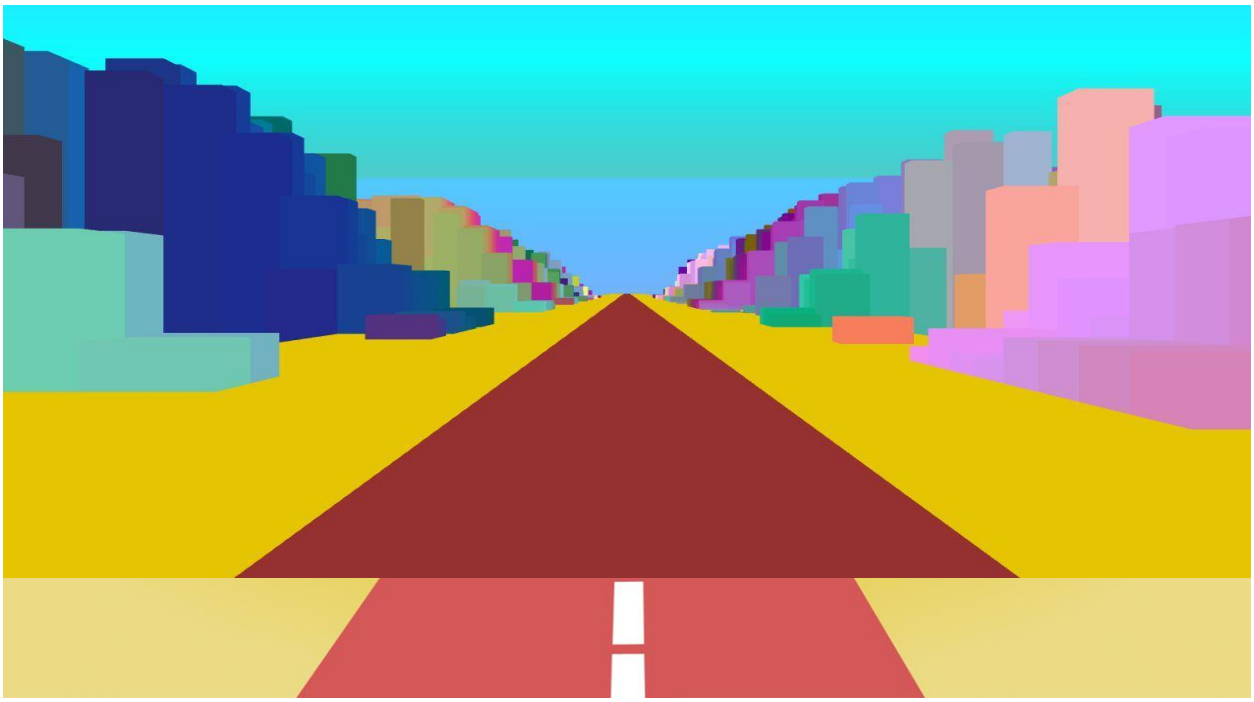


Figure 22.1: *Our first iteration of our environment design - infinite space with moving buildings.*

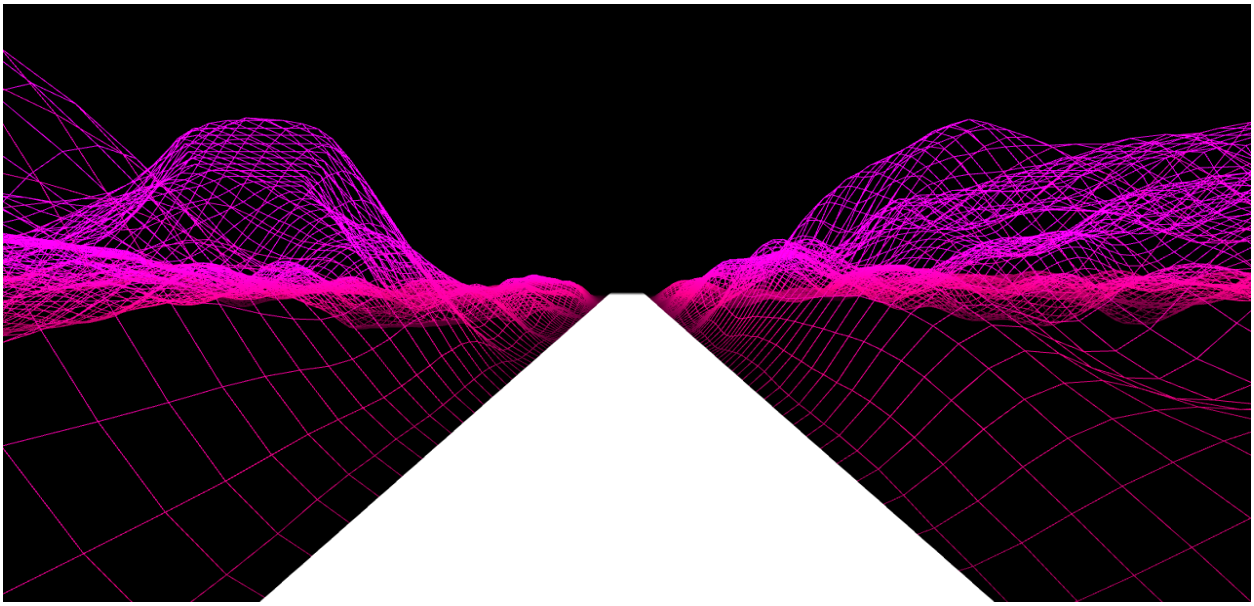


Figure 22.2: *Our second iteration of our environment design - an infinite space abstract line-based environment.*

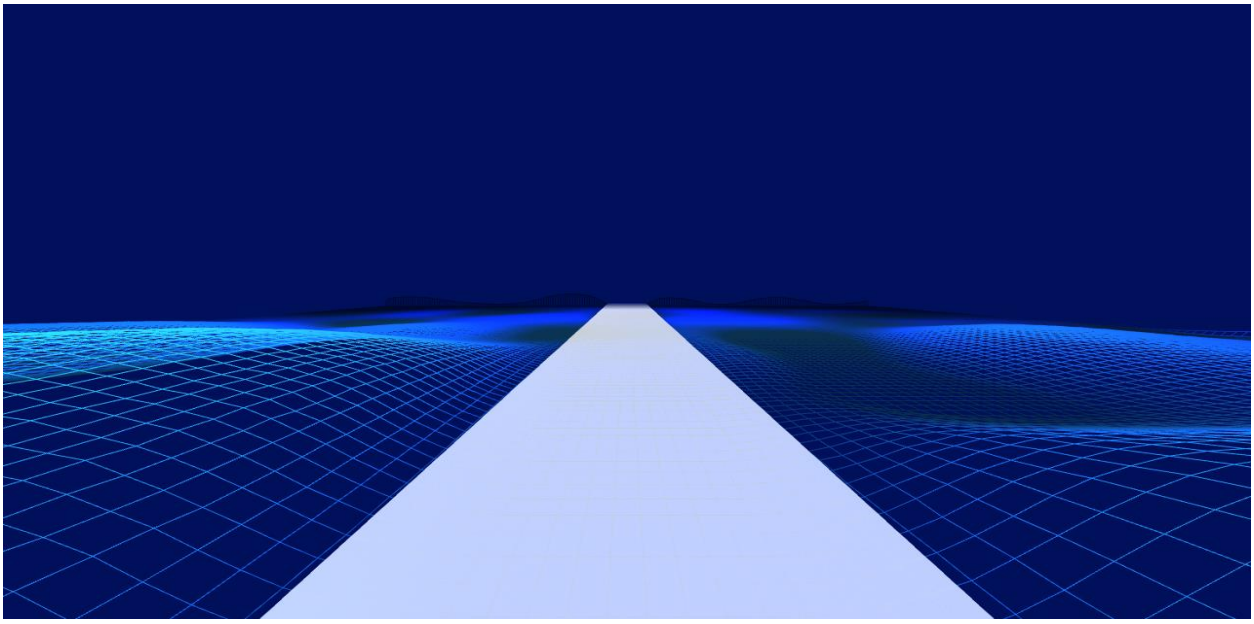


Figure 22.3: *Our penultimate iteration of one of our environment designs - an infinite space abstract line-based environment, made to look like an ocean.*

Obstacle Design

Throughout development, our obstacles remained relatively the same until the end where we developed them into our final iteration, image-based obstacles (figure 23), which was a complete change from our initial design. This change was fueled by the usability testing results.

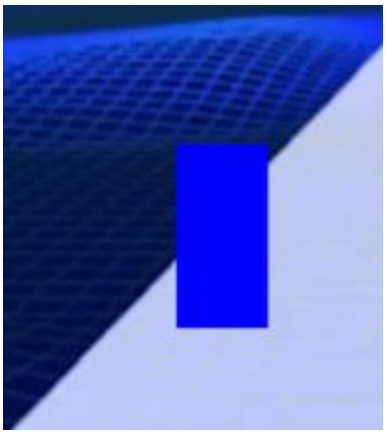


Figure 23.1: *Our first iteration of the move obstacle - colored rectangle.*

Figure 23.2: *Our first iteration of the jump obstacle - a colored rectangle.*

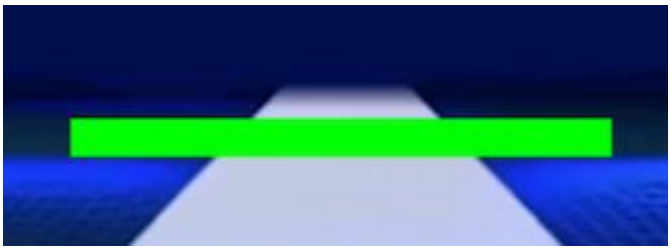


Figure 23.3: *Our first iteration of the squat obstacle - a colored rectangle.*

Font Design

Additionally, due to feedback from usability testing, we decided to change our font choice to one that is clearer (figure 24). Also, we conducted contrast checking of the text colors with the background to ensure our experience is clear, visible, and accessible to as many users as possible.

Figure 24: *Our first iteration of our font choice.*

For additional documentation and evidence on software and usability testing, see Appendix.

6.3. Limitations

Based on the proposals provided, there were several requirements that we were not able to implement.

- A user should be able to control the game's sound and music volume. However, due to time constraints, we had to prioritize other aspects of development and couldn't accommodate this requirement.
- A user should be able to use collected items to buy upgrades and unlock new characters, levels, and environments. However, this conflicted with non-functional requirements and would have required additional resources to implement, which we didn't have.
- A user should be able to collect coins, power-ups, and other items while running. However, due to complications in the presentation of obstacles, characters, and environments, we couldn't implement this feature.
- A user should be able to view an indication of how correct their movement is compared to the expected movement. However, we had concerns about completing this functionality within the given time frame, which was a limitation.
- A user should be able to pause and resume the game at any time. However, after discussions with the client and gaining her approval, we decided not to implement this requirement to ensure that the game remains fun and immersive.
- The score should increase by an additional five points for each subsequent obstacle avoided within one second of the previous one. However, after further discussions with the client and realization of the experience and its status, we opted for a simpler score mechanism, which was a limitation.

As for bugs and issues, our system has been thoroughly tested and verified to be completely free of any bugs, glitches, or issues.

7. Project Deployment and Migration

7.1. Conversion Strategies and Business Contingency Plans

During the handover of the project, STA will be provided with three key documents. These include the Source Code Documentation, Environment Integration Manual and the UX Design Documentation. These documentations, in addition to the code, will allow STA to install, configure and deploy the immersive fitness experience. In regards to deployment, it is recommended that The immersive experience be deployed using a pilot deployment strategy. Although this could result in a smaller reach in regards to the audience, the impact of a possible bug would not be widespread and can be detected at an earlier stage to be resolved. Currently, no bugs have been identified, yet it is still crucial that a new experience such as the one we have developed be deployed in a small setting first. A recommendation of where this experience can first be deployed is in Carnegie Mellon University to a student population. This is because the experience was developed on campus, making it easier to deploy and resolve bugs if they arise. In the case of any installation or deployment issues, the Source Code documentation contains an in-depth review of all the code allowing for the client to understand and fix any potential issues.

7.2. Preparing the Technology

Our Source Code Documentation aims to highlight how STA can install, develop and configure the immersive fitness experience. Since our immersive fitness experience has been built from scratch and not as development of a prior system, no data conversion or migration is required. The source code documentation highlights the software and their respective versions that the clients can access for free, or already have licensing for, such as TouchDesigner. After mentioning the relevant software, a comprehensive guide has also been included in the Source Documentation that highlights all the various libraries that need to additionally be downloaded to run the experience. The code will also be emailed to the client, allowing them to run it on their devices.

In terms of the technology infrastructure, the client has already been made aware of the specifications of the technology being used to deliver this experience through the table of the non-functional requirements. This includes details of the screen size and webcam needed.

7.3. Preparing the People

Due to the nature of our experience, all the people interacting with the experience will not require prior knowledge on how to use the system. The users are prepared within the experience itself through a warm-up and tutorial session. The tutorial will be used to familiarize the user with the system to ensure that they are able to understand the game obstacles and collision zones to be successful in the game. This goes hand-in-hand with the immersive aspect of the experience as it

is critical that the user spends their time engaged, rather than having to understand how the system works.

In addition to the warm-up tutorial, a “Stand Here” place marker and edge floor markers will be used to prepare the user prior to beginning the experience. This allows the user to know where to stand and also the bounds of the areas to stay within.

8. Documentation and Training Material

As part of preparing our developers, we have prepared the required training materials, including:

1. ***User Experience Design Documentation***

The User Experience (UX) designer document presents an in-depth reference to the UX design and development process for our immersive fitness experience. This document aims to provide a glimpse into how we have designed and structured the UX to be simple, effective, and entertaining for our users. The audience to be served for this document comprises the developers, stakeholders, and any individual engaged in the conceptualization and creation of the experience, mainly in STA. Through this document, we intend to foster a common understanding of the UX and improve collaboration among members of the team. The following guide includes user research, personas, storyboards, flow of the user experience, visual design of the character, environment and fonts used, and finally the results of user testing.

2. ***Environment Integration Manual***

This document serves as an encompassing manual on the integration between the environments and the remaining mechanisms of the system that makes up the immersive fitness experience. From this manual, you will be able to understand the relationship between the environment background and the Python-created aspects of the system, how to change the background of the experience through the Python code, and how to alter the environment's presentation.

3. ***Source Code Documentation***

The purpose of this document is to document and explain all the code about the immersive fitness experience created for STA. From this document, one can learn what the code does, how to install, configure, and maintain it, opportunities for updating and future development, and the testing conducted to ensure a working, functional, and smooth experience. In addition to the aforementioned information, the author's (initial developers) information is provided at the end, and the Python file contains consistent comments throughout, describing and explaining the purpose of code snippets and their relation to the experiences’ functionality and presentation.

Please refer to the last section of the appendix for the documentation documents.

9. Challenges and Opportunities

In the initial stages of the development of the immersive experience, one of the key challenges that was faced was the integration of both TouchDesigner and Python. This was because we wanted to ensure that the motion detection and game mechanics were highly functional, without compromising on the design. This was overcome by exporting the TouchDesigners backgrounds and integrating it into Python, which is further discussed in the Environment Integration Manual. To sustain the progress, STA can continue to diversify the design and backgrounds to better suit where they will deploy the solution or customize it to the needs of the clients they will provide it to.

Other challenges that we faced included the collision zones and the detection of more than one user at a time if they're within the same frame. In regards to the collision zones, this challenge was also overcome through rounds of testing and determining the relevant body landmarks outlined in the Source Code Documentation. Finally, detecting only one user at a time is something that can continue to be developed in the future. With the current pilot recommended deployment strategy, having one user stand within the frame will ensure that they are able to effectively play the game.

In order to further progress the developed solution, a variety of different features can also be introduced and integrated into the immersive fitness experience. In regards to the game mechanics, a range of different fitness movements can be introduced that go beyond jumping, or crouching. With the foundation of the motion detection already integrated, the source code documentation can be used as a way to determine and identify additional movements.

The immersive experience can also be customized if STA wishes to do so for a client. For example, if STA would like to change the backgrounds, they can create one via TouchDesigner and refer to the Environment Integration Manual to see how the background can be exported and integrated into the Python. In addition to changing the backgrounds, the experience can also be customized by changing the types of obstacles or characters in the game. Instructions on how to do so can be found and referred to in the UX design documentation.

The immersive fitness experience developed in this project is one that is not only a new concept for STA, but one that is new to the Qatari market. Additionally, STA prides itself in providing "bespoke solutions," and therefore the developed experience aims to tackle key issues of fitness through an innovative, immersive and inventive way through the use of motion detection. Being the first of its kind immersive experience in Qatar, STA can be expected to gain more exposure to their innovative solutions and public engagement with their experiences. This would open up the doors to more investments and also a broader range of clients than they currently serve. Therefore, for STA, this immersive experience can be expected to generate a higher ROI and pave the way for more innovative and immersive experiences that further expand the horizons of fitness.

10. Acknowledgments

This project would not have been possible without the continuous support from Carnegie Mellon University Qatar and its faculty. We would especially like to thank the teaching team including Professor Selma & Aazam , TA Mohammed, and all the CAs for all of the advice and assistance they have provided us throughout the stages of project development.

We would also like to show appreciation to STA, namely our Community Partner, Farjana Salahuddin, for her dedication throughout working with us, her flexibility, and her mentoring.

Lastly, we would like to extend our gratitude to each of our members for their resolve and perseverance in completing this project and creating this experience.

11. References

Al-Thani, Al-Thani, A., Alyafei, S., Al-Kuwari, M. G., Al-Chetachi, W., Khalifa, S. E., Ibrahim, I., Sayegh, S., Vinodson, B., & Akram, H. (2018). Prevalence of physical activity and sedentary-related behaviors among adolescents: data from the Qatar National School Survey. Public Health (London), 160, 150–155. https://doi.org/10.1016/j.puhe.2018.03.019

Aspetar Orthopaedic and Sports Medicine Hospital. (2021). Qatar National Physical Activity Guidelines. Aspetar. https://www.aspetar.com/AspetarFILEUPLOAD/UploadCenter/637736948034432931_QATAR%20NATIONAL%20PHYSICAL%20ACTIVITY%20GUIDELINES_ENGLISH.pdf

Cheema, Abraham, A., El-Nahas, K. G., Abou-Amona, R., Al-Hamaq, A. O., Maisonneuve, P., Chaabna, K., Lowenfels, A. B., & Mamtani, R. (2022). Assessment of Overweight, Obesity, Central Obesity, and Type 2 Diabetes among Adolescents in Qatar: A Cross-Sectional Study. International Journal of Environmental Research and Public Health, 19(21), 14601–. https://doi.org/10.3390/ijerph192114601

Chrismas, Majed, L., & Kneffel, Z. (2019). Physical fitness and physical self-concept of male and female young adults in Qatar. PloS One, 14(10), e0223359–e0223359. https://doi.org/10.1371/journal.pone.0223359

Damato, Tebar, W. R., Oliveira, C. B. S., Saraiva, B. T. C., Morelhao, P. K., Ritti-Dias, R. M., & Christofaro, D. G. D. (2022). Relationship of sleep quality with screen-based sedentary time and physical activity in adolescents — the moderating effect of body mass index. Sleep & Breathing, 26(4), 1809–1816. https://doi.org/10.1007/s11325-021-02519-5

Furlan, R., & AL-Mohannadi, A. (2020). An Urban Regeneration Planning Scheme for the Souq Waqif Heritage Site of Doha. Sustainability, 12(19), 7927. https://doi.org/10.3390/su12197927

Hu, Zheng, H., & Lu, C. (2021). The Association Between Sedentary Screen Time, Non-screen-based Sedentary Time, and Overweight in Chinese Preschool Children: A Cross-Sectional Study. Frontiers in Pediatrics, 9, 767608–767608. https://doi.org/10.3389/fped.2021.767608

Machado, R. (2021, October 20). Motion Capture In Games: An Overview - Sooner Esports. Sooner Esports. Retrieved January 21, 2023, from https://www.sooneresports.org/2021/10/20/motion-capture-in-games-an-overvie w/

Meza Talavera, Al-Ghamdi, S., & Koç, M. (2019). Sustainability in Mega-Events: Beyond Qatar 2022. Sustainability (Basel, Switzerland), 11(22), 6407–. https://doi.org/10.3390/su11226407

National Museum of Qatar. (2022). Interactive Learning Spaces. NMOQ. https://nmoq.org.qa/en/learn/interactive-learning-spaces/

Roman-Juan, Roy, R., Jensen, M. P., & Miró, J. (2022). The explanatory role of sedentary screen time and obesity in the increase of chronic back pain amongst European adolescents: The HBSC study 2002–2014. European Journal of Pain, 26(8), 1781–1789. https://doi.org/10.1002/ejp.2003

STA. (2020, December 9). Home. STA Enterprises. Retrieved January 21, 2023, from https://www.staenterprises.com/

The Peninsula. (2016, February 15). QNB launches Interactive Video Teller Machine at Al Sadd branch. https://thepeninsulaqatar.com/article/15/02/2016/QNB-launches-Interactive-Video-Teller-Machine-at-Al-Sadd-branch

Weil Cornell Medicine. (2022, January 24). Sahtak Awalan – Your Health First launches innovative Fitness Box campaign to boost health and wellbeing in Qatar. Latest News - Weill Cornell Medicine - Qatar. https://qatar-weill.cornell.edu/media-and-news/news/story/sahtak-awalan-your-health-first-launches-innovative-fitness-box-campaign-to-boost-health-and-wellbeing-in-qatar

12. Team CVs

Ahmed Issaoui

aissaoui@andrew.cmu.edu 70167107 doha

Professional Experience

Research Intern, *Qatar Computing Research Institute* 05/2022 – 07/2022
Achievements: Won second place in the poster presentation at QCRI
Using pre-trained hugging face models, I created a data augmentation pipeline for Arabic text. The pipeline consisted of text preprocessing (text Cleaning, tokenization, POS tagging, NER, ...) and text augmentation using mainly 5 techniques: Contextual word embedding replacement (BERT-based models), Classical word embedding replacement (Word2Vec-based models), text2text models, Generative models (GPT based) and Back Translation.

Education

Information Systems, *Carnegie Mellon University* 2024
Relevant Courses: Practical Data Science, Machine Learning in a Nutshell, Natural Language Processing with Python, Software Engineering, Application Design and Development, Regression Analysis Qatar

Projects

Tweets Gender Classification
Using tweeter API and web scraping, I scraped tweets from Twitter and scraped profile pictures associated with each tweet. Then I used face and gender detection to label each profile picture and its corresponding tweet to male or female. Finally, I trained a classification model to predict the gender of a person from his/her posted tweet.

Sentiment Analysis on African languages
Trained a model that determines the polarity of a tweet in the target African language (positive, negative, or neutral). If a tweet For message conveys both a positive and negative sentiment, whichever is the stronger sentiment is chosen.

Question-Answering System Using NLP
Built a rule-based model that generates n questions given a certain text and finds the answer to those questions in the given text.

Instagram Account Creation Automation
Using selenium, I scraped usernames and profile photos from randomly picked Instagram users. Then, using CV2 image recognition models, I filtered out all profile pictures that do not contain real human faces. Finally, using selenium I created a bot that automatically creates Instagram accounts using the collected data.

Skills

Python / Java / R / Ruby
AI / ML / NLP / DL
SQL / NOSQL
Data Analysis / Data Science
Agile / SCRUM
HTML & CSS

Languages

- English
- French
- Arabic

JEWEL DSOUZA

Mobile: +974 55630341 | **E-mail:** jew.dso@gmail.com | **LinkedIn:** linkedin.com/in/jewel-dsouza

Summary

A **dedicated**, **driven**, and **diligent** candidate seeking to utilize communication, organizational and problem-solving skills in a professional setting. Capable of working in a fast-paced environment and learning new skills.

Education

B.Sc. Information Systems | Carnegie Mellon University (2019-PRESENT)

Relevant Courses: Digital Transformation, Strategy and Management, I.T Risk Management and Auditing, International Management, Organizational Behavior, Information Systems Security

High School I.B Diploma | Doha British School (2019)

Technical Skills

Python | HTML | Java | Ruby | Ruby on Rails | CSS | JavaScript | SQL | Microsoft Office | Canva | Figma

Languages

Multilingual (English, Hindi, Konkani, and Spanish - Elementary level proficiency)

Work Experience

Course Assistant | Carnegie Mellon University | Aug 2020 – Mar 2022

- Assisted professors with classroom management and document coordination to maintain positive learning and working environment.
- Tested web assessments involving HTML, CSS, JavaScript, and Microsoft Access.
- Mentored and assisted students with term projects, case studies, and research papers.

Projects and Coursework

Factors Affecting the Adoption Of A.I Chatbots in the Qatari Airline Industry

- Analyzed and applied the Information System's success model on the adoption of artificially intelligent chatbots used in customer service for the Qatari airline industry.
- Proposed recommendations for the successful adoption of A.I chatbots for enhanced customer service and relations.

External I.T Audit Report: Nakilat

- Compiled and proposed an audit report for Nakilat, a maritime shipping company in Qatar, under the guidance of Dr. Savanid Vatanasakdakul.

Implementation of Emergency Management Systems in Natural Disasters

- Analyzed the implementation of Emergency Management Systems and technologies and the roles of relevant stakeholders.

Mariam Al Thani

+974 3307 3222 | mathani@andrew.cmu.edu
linkedIn.com/in/mariam-al-thani/

Education

Carnegie Mellon University | Class of 2024

B.Sc. Information Systems | QPA 3.54
Concentration: Digitalization
Minor: Professional Writing

Courses

Sustainability in the Digital Age
Business Process Modeling & Implementation
Information Systems Consulting Project
Digital Transformation Strategy & Management
Application Design & Development
Foundations of Software Engineering
Introduction to Data Structures
Special Topics in HCI
Information Graphics
Linguistic & Social Aspects of Immigration
Social & Political Philosophy
Independent Study in Rhetoric

Technical Skills

Python | Django | Ruby on Rails | Java
HTML | CSS | JS
SQL
Adobe XD | Adobe Photoshop | TouchDesigner
NVIVO
Signavio BPM

Projects

Information Systems Consulting Project

- Consulted with & proposed a product to a start-up company, STA, to create an immersive fitness experience that encourages physical and mental wellbeing in Qatar.
- Managed the project with a team of four students, communicating with the client throughout the stages of an Agile System Development Life Cycle.

Sustainability in the Digital Age

- Researched and designed an innovative technology-based solution to encourage waste management and proper recycling in CMU-Q.
- Presented the innovative sustainable solution to the Eco-Campus Waste Management Committee.

Business Process Modeling & Implementation

- Analyzed, redesigned, and refined CMU-Qs' current Procurement Process, in three months, using two process analysis methods.
- Automated and implemented CMUQs' improved Procurement Process through Enterprise Resource Planning with Tallyfy in four weeks.

Application Design & Development

- Designed, developed, and tested a web application for an online store using a Model-View-Controller pattern and Test-Driven Development.
- Created an interactive dashboard curated for two users based on the principles of user-centered design using Figma.

Experience

Course Assistant, CMU-Q | Jan - April 2023

- Tutored students in probability & statistics, gaining hands-on teaching skills.
- Created Excel sheets documenting students' attendance records, identifying and reporting discrepancies between written and digital records.

Eco-Campus Waste Management Committee, CMU-Q | Sep 2022 - Present

- Advertised reusable food containers, encouraging the CMU community to become more sustainable and waste-conscious.
- Tailored digital media to promote ongoing and future events, refining my graphic and information-presenting skills.
- Held waste management information and recruiting sessions, developing my public speaking and presentation skills.

Raghad Mazen Abunabaa

(+974) 5033 3032 | rmabunab@andrew.cmu.edu | Doha, Qatar
https://www.linkedin.com/in/raghad-abunabaa
Nationality: Jordanian and Brazilian

QUALIFICATIONS SUMMARY

- Demonstrative of strong interpersonal skills with the ability to work with people at all levels and from various backgrounds.
- Proven ability to work under change and pressure in fast paced environments.
- Proficient in Microsoft Office tools, as well as various programming languages including Python, Java, Ruby, HTML, and R.
- Proficient in analyzing and interpreting financial statements using Bloomberg and Eikon terminals.

EDUCATION

Carnegie Mellon University
Bachelor of Science in Business Administration — Expected Graduation: May 2023
Concentration: Finance & Business Analytics | Minor: Information System
Dean's Honor List: Fall 2020 | Fall 2021 | Spring 2022 | Fall 2022

COURSES

Corporate Finance | Mathematical Models for Consulting | International Finance | Business Presentation | Pricing Strategy | Operation Management | Fundamentals of Programming and Computer Science | Application Design and Development

EXPERIENCE

Finance Intern | **Rimads** | Telehealth Startup | Feb – Apr 2022

- Aided in the development of two revenue models, identifying the most viable business line to facilitate its expansion strategy.
- Created and presented reports and presentations for introducing a new business line, granting board members' approval.
- Performed monthly reconciliation of records for 5-years, resolving discrepancies and ensuring accurate financial information.

Business Analyst Intern | **C-Wallet** | FinTech Startup | Aug - Oct 2021

- Synthesized market research data, identifying market gaps and market penetration which were used to improve the app.
- Conducted market analysis over a 2-month period, reporting current market trends used to assess company's performance.
- Assisted in building and surveying 60+ customers, gaining insights of local FinTech industry and customer preferences.

Transaction Advisory Service Intern | **ECOVIS** | Accounting | Jun – Aug 2021

- Utilized public data to compile EV, EBITDA, and profitability margins, supporting the firm's financial model.
- Contributed to the development of a financial model, guiding owners to restructuring their debt of $120 million.
- Supported the derivation of a valuation, aiming to find an investor to complete their partial exit from the investment.

PROJECTS

Corporate Finance

- Analyzed annual reports of a public limited company using Bloomberg, measuring the value of business opportunities
- Derived a valuation model for a public company, determining whether it was a good target for merger and acquisition.

Business Communication

- Developed a business plan to grow a retail business, identifying strategies to achieve business objectives.
- Devised a strategic plan for the retail business, proposing a new direction for the company and support its expansion plan.

Application Design and Development

- Designed, developed, and tested a web application using Model-View-Controller pattern in software architecture.
- Applied the principles of user-centered design to build the back-and-front end of the web application using Figma.

ADDITIONAL

Languages: Arabic (native) | English (fluent) | Spanish (intermediate)
Interests: Volunteering | Travelling | Horseback Riding | Badminton
Engagement: Philanthropy Club (president) | Consulting Club (member) | Carnegie Mellon Business Association (member)

13. Appendices

Finalized Functional Requirements

A. Final System Requirements

The final requirements for our system and their classification. They are divided based on kind (functional or non-functional) and within their own kind based on categories.

Functional Requirements

Requirements that are crossed out were not implemented in the final solution. For further information refer to section 6.3, Limitations.

Group	Number	Functional Requirements	Classification
Gameplay Mechanics	GM1	The game should have a character that should be in a running motion, facing obstacles throughout the game	Must
	GM2	The game should include a variety of obstacles that can be avoided using movements corresponding to up, right, left, and down inputs.	Must
	GM3	The user should be able to view an indication of the movement they should do to avoid the upcoming obstacle	Should
	GM4	The score should increase by one each second during the game	Must
	GM5	The score should increase by five each time an obstacle is avoided	Must
	~~GM6~~	~~The score should increase by an additional five points for each subsequent obstacle avoided within 1 second of the previous one~~	~~Must~~
	GM7	The character should have three lives	Must
	GM8	A life should be consumed once the character touches an obstacle	Must
	GM9	The game should end once all lives are consumed	Must
	GM10	A user should be able to restart the game once it is ended	Must

	GM11	~~A user should be able to pause and resume the game at any time.~~	~~Should~~
	GM12	A user should be able to make the character jump by performing a jumping jack, move the character to the left, move the character to the right, and move the character down by performing a squat.	Must
	GM13	A user should be able to view their score and track their progress throughout the game	Must
Game Features	GF1	A user should be able to learn how to control the character's movement through a tutorial	Must
	GF2	A user should be able to go through a warm-up session before starting the game	Should
	GF3	~~A user should be able to view an indication of how correct the movement they are doing compared to the expected movement~~	~~Should~~
	GF4	A user should be able to view a leaderboard that shows the top scores of all players	Must
	GF6	A user should be able to select different difficulty levels and environments	Should
	GF7	A user should be able to collect coins, power-ups, and other items while running.	Could
	GF8	A user should be able to use the collected items to buy upgrades and unlock new characters, levels, and environments.	Could
	GF9	~~A user should be able to control the game's sound and music volume~~	~~Should~~
CRUD Operations	CR1	A username should be randomly generated once a user starts the experience, so that their top score is stored.	Must

Non-Functional Requirements

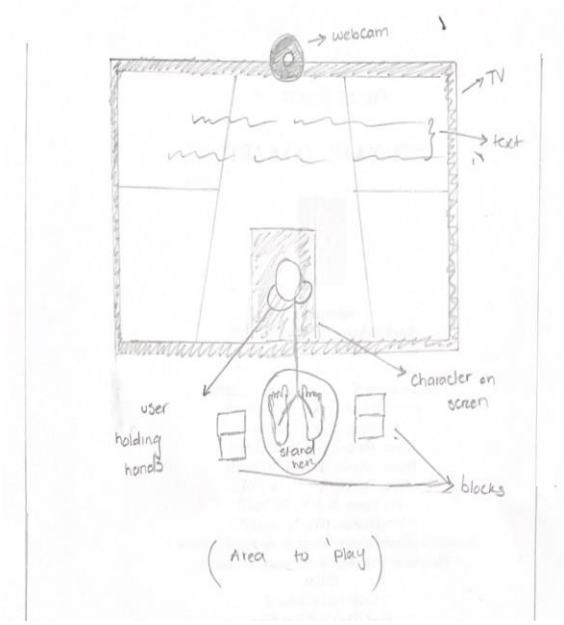
Group	Number	Requirement	Classification
Operational	OP1	The system will run on 65” large screens connected with cameras for motion detection.	Must
	OP2	The system should be able to run on any computer with at least a 6GB GPU.	Should
	OP3	The system should connect to the large screen in different environments.	Should
	OP4	The system could operate in an enclosed room and open space.	Could
Performance	P1	The system will detect a person's movement in under 5 seconds.	Must
	P2	The system will respond to a person's movement in under 5 seconds.	Must
	P3	The system should be playable at least 23 hours a day.	Should
	P4	To minimize energy consumption, the system could switch to ‘sleep mode’ if the motion detection has not been activated in 3 hours.	Could
	P5	The system will not detect more than one users movement at a time.	Wont
Security	S1	The system will only have edit access for technicians.	Must
	S2	The system will only be readable for users.	Must
	S3	The system should only store user scores to present on the leaderboard.	Should

	S4	Only the technicians could have access to records of past user scores.	Could
Cultural and Political	CP1	The system will not record any footage through the camera.	Must
	CP2	The system should explicitly highlight its accordance with Qatar's Data Protection & Privacy Laws.	Should
	CP3	All components of the system must be in compliance with commercial copyright usage laws, making it alterable for the company for future purposes.	Must
Usability	U1	The system should have an interface that allows 90% of users to start playing the game in less than 3 movements.	Should
	U2	80% of users could be able to understand the game's movements within the first play.	Could
	U3	The system should accurately recognize a persons movement 80% of the time.	Should
	U4	The systems movement mechanics should be understandable for all users in less than 10 minutes.	Should

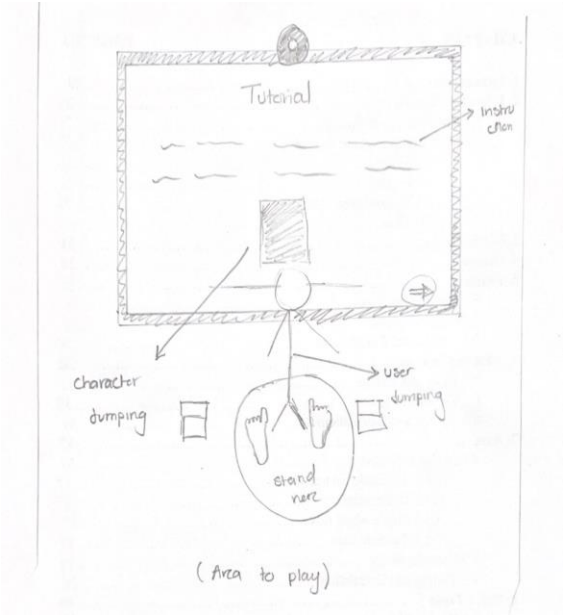
User Experience Development

B. Initial Experience Storyboard of the Physical Space

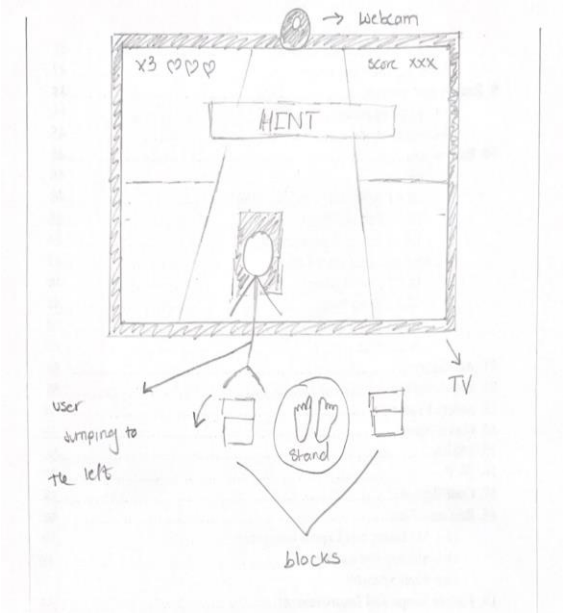
The initial drafted vision of what the experience would look like in the ‘real’ world based on discussions with our client and the team.



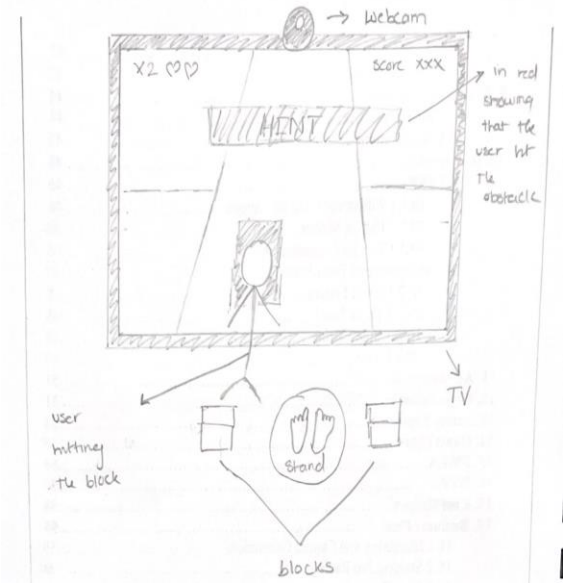
Starting the game



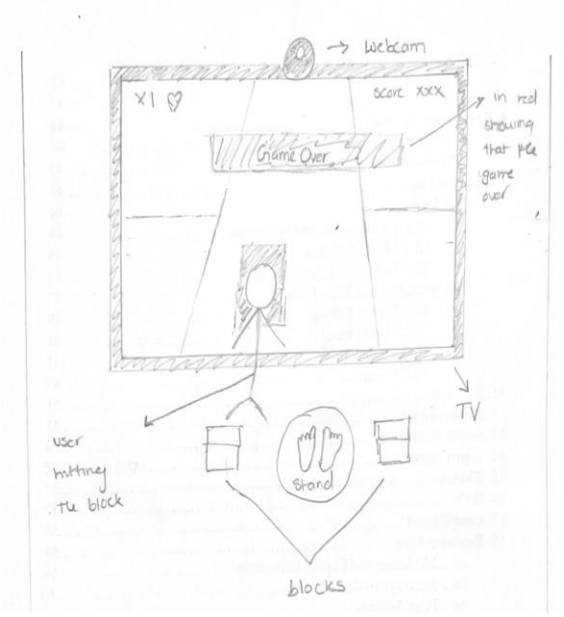
Tutorial Demonstration



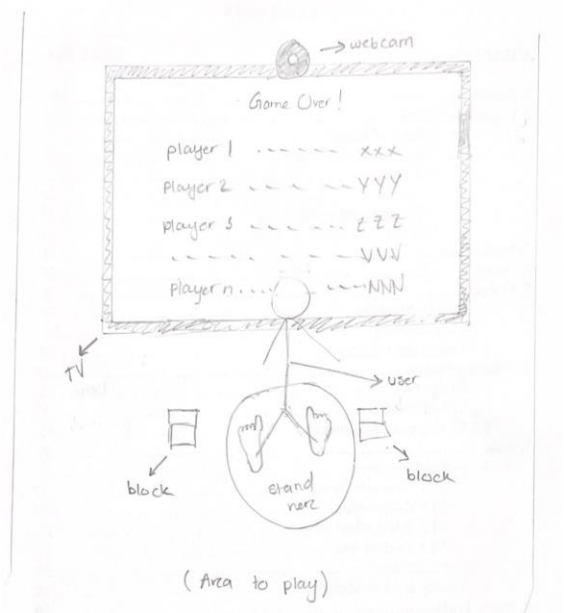
Engaging with the game



Hitting an Obstacle



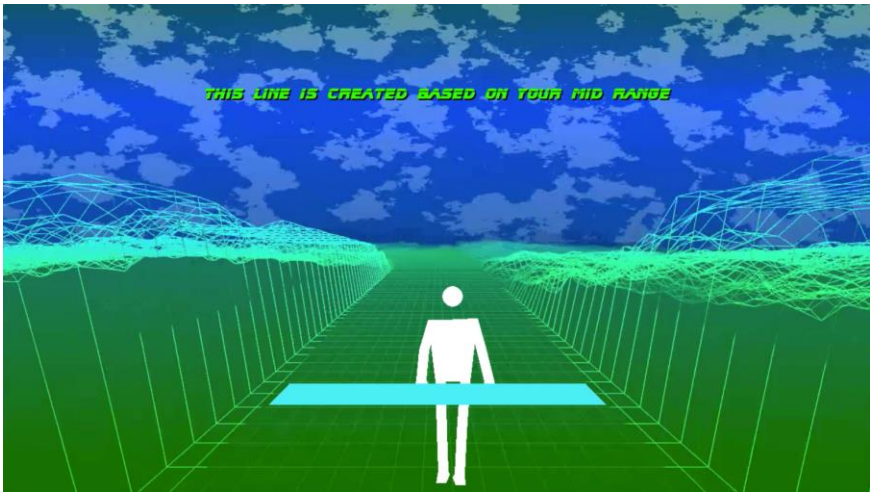
Hitting an obstacle with one life left



Leaderboard

C. Additional visuals of the Solution

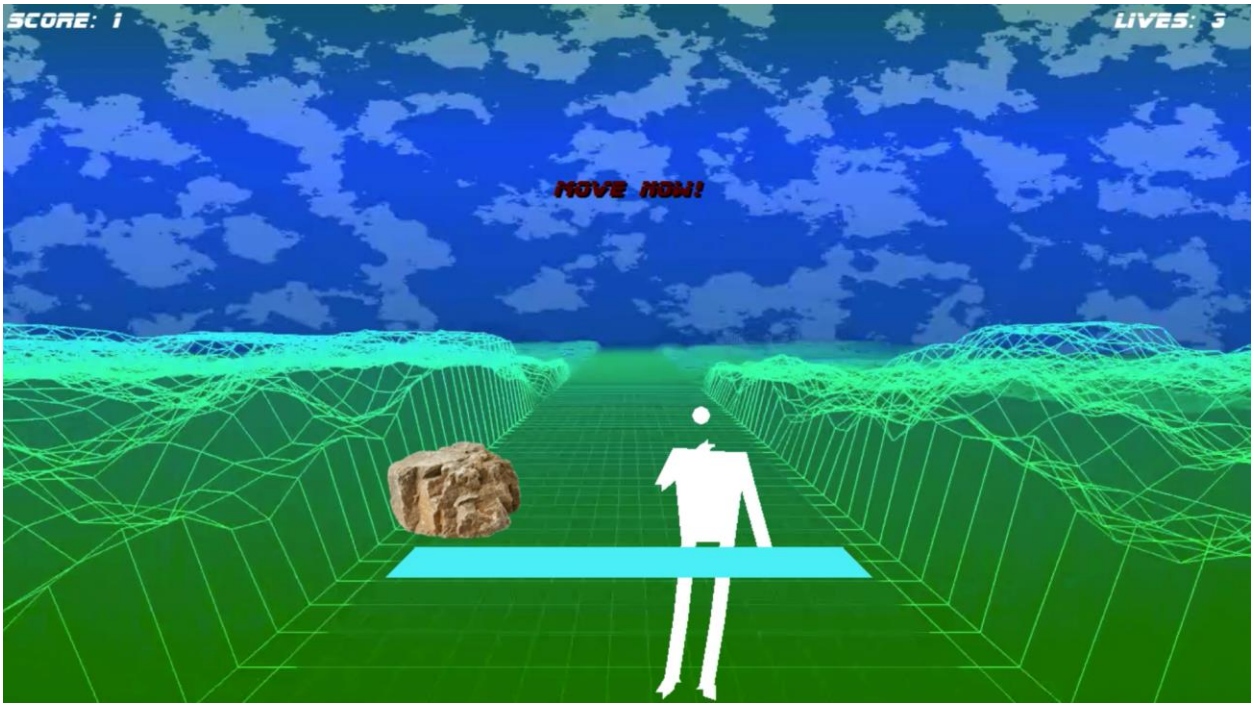
In addition to visuals provided throughout the report, below are some additional visuals with aspects of the experience in its final state.



Part of the tutorial - Explaining the different collision zones and how they are created.

Part of the tutorial - Explaining the different exercises in the experience.

Part of the tutorial - Explaining the different obstacles and how to avoid them, and a movement indicator.



Part of the experience - The score, lives, collision zone, obstacle, character, and movement indicator all together.

D. Intellectual Property Agreement

The intellectual property agreement our team signed at the beginning of project development and client consultation in order to ensure both parties are on the same page on the ownership of the final deliverables.

STA QSTP LLC
Office 382, Innovation Center
Qatar Science & Technology Park
Tel +974 5044 8925

STA
Sense | Think | Act

INTELLECTUAL PROPERTY AGREEMENT

The student team comprising of Ahmed Issaoui, Jewel Dsouza, Mariam Al Thani & Raghad Abunabaa hereby agrees that all intellectual property, including but not limited to all deliverables, assets, creative ideas, visuals, code, documents, data, video recordings, audio recordings, electronic recordings, and other materials that the team develops at any time as a result for the Carnegie Mellon University course 67-373 Information Systems Consulting Project shall belong to STA QSTP LLC ("STA"), together with any associated improvements, technology, designs, ideas, processes, techniques, know-how and data, whether or not patentable, patents, copyrights, trademarks and trade secrets (collectively, the "Work Product") shall be the sole property of STA. STA shall have an exclusive, perpetual, irrevocable, transferable, unlimited, royalty-free, worldwide license to make, have made, use, reproduce, market, modify, make derivative works from, publicly perform, publicly display, offer to sell or sell the Work Product

The student team however may ask for permission from STA to re-use the Work Product including assets, ideas, techniques and other intellectual property developed during the project, if they wish to use them in the future.

Testing Methodologies and Results

E. Full Testing Plans

The full testing plans that we referred to in section 3. This included information on all the different kinds of testing, what components of the experience they relate to, which requirements they refer to, who developed it, and who was assigned to test it. We completed each of these and more, sometimes multiple times.

1. Unit Testing

a. Motion Detection

Test Suite ID	TS001_COMPONENT_MOTION_DETECTION
Test Case ID	TC001
Test Case Summary	To verify that the character behaves in line with the user's movement.
Related Component(s) and developers	COMPONENT U2 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM2. Nonfunctional requirement U2.
Prerequisites	A functional webcam is available. webcam connected to the camera.
Test Procedure	To test motion detection, we will follow the below procedures: 1. Connect the camera to the tv. 2. Open the game. 3. Stand in the middle of the screen.

	4. Join both hands to start the game. 5. Move to the left 6. Move to the right 7. Complete a jump 8. Complete a jumping jack 9. complete a squat
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. According to the testing procedures above, we expect the character to move the left, back to the center when they move right. In addition, we expect the rectangle not to react to a normal jump, but move up when the user does a complete jumping jack. Moreover, we expect the rectangle to get halved when the user completes a crouch.
Tester name	Raghad Abunabaa, Ahmed Issaoui
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

b. Use Scenario Testing

Test Suite ID	TS002_COMPONENT_USE_SCENARIO
Test Case ID	TC002

Test Case Summary	To verify that the game starts only when the user joins hands
Related Component(s) and developers	COMPONENT P001 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM10.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users' movements.
Test Procedure	To test starting the game, we will follow the below procedures: 1. Connect the camera to the tv. 2. Open the game. 3. Stand in the middle of the screen. 4. Keep both hands away from each other. 5. Make both hands touch each other. 6. Join both hands.
Test Data	User Movements: hands movement Character Movements: hands movement
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect different body landmarks. We expect the game not to start when the user stands in front of the camera or when the hands are separated or are only touching each other. However, we expect the game to start when the camera detects that the user has both hands joined together.

Tester name	Jewel Dsouza, Mariam Al Thani & Raghad Abunabaa
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

Test Suite ID	TS002_COMPONENT_USE_SCENARIO
Test Case ID	TC003
Test Case Summary	To verify that the game has obstacles that can be avoided by moving
Related Component(s) and developers	COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM1. Functional requirement GM2. Functional requirement GM7.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users’ movements.
Test Procedure	To test avoiding obstacles, we will follow the below procedures: 1. Connect the camera to the tv. 2. Open the game. 3. Stand in the middle of the screen.

	4. Join hands to start the game. 5. Observe if there are obstacles throughout the game. 6. Move away from the obstacle by jumping to the left, right, up or down depending on where the obstacle is.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. In addition, we expect obstacles to be randomly placed throughout the experience. We expect the character on the screen to move to the side where the user has moved, away from the obstacle. This means that we expect the number of lives to stay the same.
Tester name	Jewel Dsouza & Ahmed Issaoui
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

Test Suite ID	TS002_COMPONENT_USE_SCENARIO
Test Case ID	TC004
Test Case Summary	To verify that lives in the game are responsive to hitting obstacles.
Related Component(s) and developers	COMPONENT P002 (Developed by Ahmed Issaoui)

Related requirement	Functional requirement GM1. Functional requirement GM2. Functional requirement GM7. Functional requirement GM8.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users' movements.
Test Procedure	To test lives responsiveness, we will follow the below procedures: 1. Connect the camera to the tv. 2. Open the game. 3. Stand in the middle of the screen. 4. Join hands to start the game. 5. Observe if there are obstacles throughout the game. 6. Move away from the obstacle by jumping to the left, right, up or down depending on where the obstacle is. 7. Stay/move to the side where the obstacle is at.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. In addition, we expect obstacles to be randomly placed throughout the experience. We expect the character on the screen to move to the side where the user

	has moved, away from the obstacle. This means that we expect the number of lives to stay the same. Lastly, when the user stays on the same side of the obstacle or moves towards the obstacle and hits it, we expect the number of lives to decrease by one.
Tester name	Mariam Al Thani & Ahmed Issaoui
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

Test Suite ID	TS002_COMPONENT_USE_SCENARIO
Test Case ID	TC004
Test Case Summary	To verify that the game ends once the user consumes all their lives.
Related Component(s) and developers	COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM7. Functional requirement GM8. Functional requirement GM9.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users’ movements. The game successfully detects collisions with the obstacle.

Test Procedure	To test ending the game, we will follow the below procedures: 1. Connect the camera to the tv. 2. Open the game. 3. Stand in the middle of the screen. 4. Join hands to start the game. 5. Move to the side of the obstacle three times consecutively.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. In addition, we expect obstacles to be randomly placed throughout the experience. We expect the character on the screen to move to the same side of the obstacle. In addition, we expect that each time the user hits an obstacle, one life will be consumed for the first two obstacles. However, when the character hits the user for the third time, we expect the game to end by showing that the game is over.
Tester name	Raghad Abunabaa & Ahmed Issaoui
Test Environment	Screen: A 65" SONY TV screen connected with a web camera. OS: Mac OS & Windows

c. *Data Flow Testing*

Test Suite ID	TS003_COMPONENT_DATA_FLOW
Test Case ID	TC005

Test Case Summary	To verify that the game generates a random username every game.
Related Component(s) and developers	COMPONENT S003 (Developed by Mariam Al Thani)
Related requirement	Functional requirement GF4.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users' movements. The game successfully detects collisions with the obstacle.
Test Procedure	To test name generation, we will follow the below procedures: 6. Connect the camera to the tv. 7. Open the game. 8. Stand in the middle of the screen. 9. Join hands to start the game. 10. Move to the side of the obstacle three times consecutively. 11. Repeat the above steps three times.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. In addition, we expect obstacles to be randomly placed throughout the experience. We expect the character on the screen to move to the same side of the obstacle. In addition, we expect that each time the user hits an obstacle,

	one life will be consumed for the first two obstacles. However, when the character hits the user for the third time, we expect the game to end by showing that the game is over. By checking the data that is stored on excel, we expect to observe three different names stored with the date in which the user played the game.
Tester name	Jewel Dsouza & Ahmed Issaoui
Test Environment	Screen: A 65" SONY TV screen connected with a web camera. OS: Mac OS & Windows

2. Integration Testing

2.1. Motion Detection

Test Suite ID	TS004_INTEGRATION_MOTION_DETECTION
Test Case ID	TC006
Test Case Summary	To verify that the output of the motion detection unit is successfully taken as an input in the game
Related Component(s) and developers	COMPONENT U2 (Developed by Ahmed Issaoui) COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM12.
Prerequisites	A functional webcam is available. webcam connected to the camera. Motion detection accurately recognizes users' movements.

	The game has a character that is able to move left, right, up, and down
Test Procedure	To test the integration between the motion detection unit and the game , we will follow the below procedures: 12. Connect the camera to the tv. 13. Open the game. 14. Stand in the middle of the screen. 15. Join hands to start the game. 16. Perform all the movements (jump left, jump right, jumping jack, squat)
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After connecting the webcam to the tv and opening the game, we expect the camera to detect joined hands to start the game. We also expect the character to be in the middle of the screen. And finally, we expect the character to move corresponding to the user movement
Tester name	Raghad Abunabaa & Ahmed Issaoui
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

2.2. System Testing

2.3. Requirements Testing

Test Suite ID	TS001_SYSTEM_REQUIREMENTS
Test Case ID	TC007
Test Case Summary	To verify that the system is immune to improper actions attempted by the user when interacting with the experience.
Related Component(s) and developers	COMPONENT S002 (Developed by Ahmed Issaoui, Mariam Al Thani) COMPONENT U2 (Developed by Ahmed Issaoui) COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM7 Functional requirement GM8 Functional requirement GM10 Functional requirement GM12.
Prerequisites	A functioning web camera and screen set up is available. The motion detection is accurately recognizing users' movements. The system and background environment are smoothly running. The character is appropriately responding to the user's movements.
Test Procedure	To test that the system is immune to improper actions, we will follow the below procedures: 1. Run the system on the screen and camera set up. 2. Stand in the middle of the screen. 3. Click on the television screen.

	4. Join hands to start the game. 5. Perform multiple movements consecutively at a fast pace. 6. Step away from the screen and cameras' range. 7. Introduce another user to the screen. 8. Both users try to perform all the movements (jump left, jump right, jumping jack, squat) at the same time.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After running the system and standing in the middle of the screen, we expect the users' movements to be detected. After clicking on the television screen, we expect no outcome as the experience is only readable. After joining hands to start the game we expect the users' movements to be reflected by the character on the screen. When the user performs multiple movements at a fast pace, we expect the system to keep up and detect and reflect these movements on the screen. When multiple users try to perform any movements, we expect the system to only detect one user and not reflect movements performed by other users.
Tester name	Raghad Abunabaa & Jewel Dsouza
Test Environment	Screen: A 65" SONY TV screen connected with a web camera. OS: Mac OS & Windows

3. Usability Testing

3.1. Usability Testing Plan

Our usability testing plans intertwine with our user acceptance testing plans as both forms of testing will be done in the same session - the main difference being what we are looking for while the user

completes the task and undergoes the experience. For usability testing, we will be monitoring how intuitive the system is and how comfortably they can achieve the objective of playing the game. Through these sessions, our main focus will be on the anticipated error/fault of inaccurate or delayed motion detection so, this aspect and how it interferes with users' experience will be our focal point. The non-functional requirements associated with our focus is U1.

For user acceptance testing, we will be looking at how smoothly users can actually interact with the experience and complete the objective - focusing more on the functionality of the system and how the motion detection works and presents. Ideally, in these testing sessions, evidence of good UI/UX would present through little to no interference from our end to explain anything to the user as the motion detection will be accurate, quick, and smooth enough for the user to immediately understand the experience. The non-functional requirements associated with our focus are P1, P2, and P5.

All of our testing sessions will be conducted in person and will be moderated. Considering the context of our project, it is most logical to make these sessions in person as the experience needs to be set up and involves interaction with the physical and digital space. We decided to pursue the route of moderated usability testing sessions as we believe it is especially imperative for our project that we directly witness how the user is responding to the experience, any subtle changes in expressions, reactions, or body language, allowing us to be able to notice a pattern more clearly. This also gives us the chance to see how users will respond in a situation where the motion detection is slow or inaccurate, if the situation arises.

Additionally, we have the opportunity to make the testing sessions semi-moderated in the sense that team member(s) will be present and can and will give the user a general task of starting and playing the game but will minimize interaction and interruptions while the user is completing the task assigned to them. This would be similar to the think-aloud protocol. Lastly, moderated sessions are also useful for our project since the intuitiveness of the UX is incredibly important for an immersive experience that relies on the user's continuous interest to be used - in the end, users do not have to use our experience or play it, they should want to.

3.2. User Categories and Tasks

In our immersive experience, there is one main user - the player. But, this means that anyone can be this 'user'. So, we will focus on different types of users or user groups. For example, some categories we will certainly involve in our usability testing plans including people of different ages, heights, genders, professions, cultures, etc. to ensure that our experience is functional, usable, interesting, and accessible to all, or at least a wide range of, users. This is the ideal and applicable route for us to follow as, our most anticipated error relates to motion detection which is directly

connected to the camera sensors and what it detects. By this, we mean that, it is important to ensure that the motion detection will respond with the same level of accuracy, speed, smoothness, etc. for all types of users because the overall usability of our experience will greatly be reduced if we only make sure the motion detection is accurate and has a short response time for one kind of user.

3.3. Number of Users and Types

In our UI testing, we will aim to include 3-5 users from each category. Based on the aforementioned user groups, users we will involve include, but are not limited to, students, professors, children (of different ages), and older faculty and staff. When considering the users we will involve in our testing sessions, we have to consider the need to physically set up the experience which will make it difficult to conduct this testing in any public space. Thus, we anticipate that these testing sessions will take place on the university campus - in the HCI lab and in any classrooms or open spaces with a large screen we can connect to. As for the decision to test 3-5 users from each category, we agreed that this is a feasible enough range to provide us with enough assurance through testing of the errors we anticipate and if and how they are manifesting with this category of user. So, if we included five children, some of them would be different genders, different heights, and different ages - and we would receive all these different responses and ideally find the same level of usability throughout the category. Furthermore, if it is feasible to do so at the time, and we find a usability issue arising specifically within one category of users, we plan on expanding our testing for that category to further research the issue as we progress in improving the usability to ensure this problem does not persist.

3.4. User Task and List of Procedures

Since the experience is one where all users will generally have one ‘task’, in our sessions,we will:

1. Ask users to think out loud their thought process throughout the experience,
2. Ask users to position themselves in front of the screen,
3. Ask users to start the game,
4. Ask users to play the game and try to achieve the highest score.

Automatically, once the user loses all of their lives, the game will end.

We have chosen to proceed with the task of users starting and playing the game to get the highest score for all of our sessions because it is as similar as we can get the session to how users will experience the final solution. All the information we provide in the task will eventually be presented in the experience itself (explicitly or implicitly). In our final experience, users will simply know that this is what the experience is for and just undergo it, there is no opportunity or ability for

them to do anything else - even if they quit midgame or do not join their hands at the beginning, or so and so, nothing will happen - the experience will just not continue and revert to the start screen.

For our usability testing plans of our experience thus far, we will have nearly all the game mechanics of the experience functional or available for the user. This means that, when the users undergo the testing session, they will be tasked and presented with an experience that is, function-wise, realistically close to the final experience. This connects with our ability to sufficiently test the usability of the experience as the user will have a wider range of actions they will be able to perform and features to look at e.g. the lives, what the obstacles look like, the character, etc.

3.5. Security Testing

Test Suite ID	TS001_SYSTEM_SECURITY
Test Case ID	TC008
Test Case Summary	To verify that the system cannot be inappropriately accessed by users without authority.
Related Component(s) and developers	COMPONENT S002 (Developed by Ahmed Issaoui, Mariam Al Thani) COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM10 Functional requirement GF4
Prerequisites	A functioning web camera and screen set up is available. The system and background environment are smoothly running.
Test Procedure	To test that the systems' security is not compromised in attempts at unauthorized access, we will follow the below procedures: 1. Run the system on the screen and camera set up.

	2. Stand in the middle of the screen. 3. Tap screen to start the game. 4. Join hands to start the game. 5. End the game. 6. Tap screen to restart the game. 7. Hover over the screen to go through the leaderboard.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat, Tapping
Expected Result	After running the system and standing in the middle of the screen, we expect the users' movements to be detected. After clicking on the television screen, we expect no outcome as the experience is only readable. We expect that, as the system is only readable to users, tapping on the screen to start the game will elicit no response from the system.After joining hands to start the game we expect the users' movements to be reflected by the character on the screen. When ending the game we expect that the system will present the leaderboard screen. When the user taps on the screen to restart the game we expect the system to elicit no response. This is the same expectation for when the user hovers through the screen to scroll through the leaderboard.
Tester name	Jewel Dsouza
Test Environment	Screen: A 65" SONY TV screen connected with a web camera. OS: Mac OS & Windows

3.6. Performance Testing

Test Suite ID	TS001_SYSTEM_PERFORMANCE
Test Case ID	TC009
Test Case Summary	To verify that the system can withstand high traffic from users and can respond accordingly.
Related Component(s) and developers	COMPONENT U2 (Developed by Ahmed Issaoui) COMPONENT P002 (Developed by Ahmed Issaoui)
Related requirement	Functional requirement GM1 Functional requirement GM2 Functional requirement GM8 Functional requirement GM10 Functional requirement GM12.
Prerequisites	A functioning web camera and screen set up is available. The motion detection is accurately recognizing users' movements. The system and background environment are smoothly running. The character is appropriately responding to the user's movements.
Test Procedure	To test that the system can respond in high traffic situations, we will follow the below procedures: 8. Run the system on the screen and camera set up. 9. Stand in the middle of the screen. 10. Join hands to start the game. 11. Move the whole body and or limbs around the area in a random

	manner. 12. Wave hands and arms while moving left, right, up, and down. 13. Perform multiple movements (jumping jack, squat) consecutively at a fast pace. 14. Purposely repeatedly miss obstacles (to lose lives). 15. Introduce multiple users to the screen. 16. All users try to perform all the movements (jump left, jump right, jumping jack, squat) at the same time.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	After running the system and standing in the middle of the screen, we expect the users' movements to be detected. After clicking on the television screen, we expect no outcome as the experience is only readable. After joining hands to start the game we expect the users' movements to be reflected by the character on the screen. We also expect that the system will be able to perform under situations where immense changes in the motion detected data are input into the system by a user moving randomly across the area. We also expect the system to be able to handle similar situations when the user is moving their hands and legs at the same time in a fast manner. In both of these cases, we expect the system to accurately perform and respond to the persons movements by reflecting the changes in the character.When the user performs multiple movements at a fast pace, we expect the system to keep up and detect and reflect these movements on the screen. When users repeatedly hit obstacles and thus lose lives quickly, we expect the system to perform smoothly and function as normal in these cases by removing one life each time the user hits an obstacle and ending the game if they have lost all their lives. When multiple users try to perform any movements, we expect the system to only detect one user and not reflect movements performed by other users.

Tester name	Raghad Abunabaa, Jewel Dsouza, Mariam Al Thani
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

4. Acceptance Testing

Test Suite ID	TS001_ACCEPTANCE
Test Case ID	TC0010
Test Case Summary	To verify the user is able to learn how to control the character’s movement and avoid obstacles to be able to play the game.
Related Component(s) and developers	Component GF1 (Developed by)
Related requirement	Functional Requirement GM12. Functional Requirement GF1. Functional Requirement GF2. Functional Requirement GF3.
Prerequisites	The warm-up tutorial should be developed and available in the game.
Test Procedure	To test that the user is able to control the character movements, we will:

	1. Run the game on the screen. 2. Make the user stand at the center of the screen. 3. The system will prompt the user to join hands. 4. User will begin the warm up. 5. Verify through observation whether the user is able to control the character through the warm up by following the onscreen instructions.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	The expected result is that the user should be able to move the character left and right. Additionally, the user should also be able to make the character crouch, and jump by doing a jumping jack. The successful result is that the user will be able to imitate each movement from the tutorial.
Tester name	Raghan Abunabaa and Mariam Al Thani
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

Test Suite ID	TS002_ACCEPTANCE
Test Case ID	TC011
Test Case Summary	To verify that the user is able to run the end to end scenario of that game by themselves without third party assistance.

Related Component(s) and developers	Component GF1 (Developed by Ahmed Issaoui)
Related requirement	Functional Requirement GM1, GM2, GM4, GM5, GM6, GM7, GM8, GM9, GM10, GM11, GM12. Functional Requirement GF1, GF2, GF4, G7. Functional Requirement CR1.
Prerequisites	The game should be up and running. Users require no prior knowledge of the game and its mechanics.
Test Procedure	To test that the user is able to run the end to end scenario of the game, we will: 1. Run the system on the screen with the camera. 2. Have the user stand at the center of the screen. 3. Leave the user to follow the prompt on the screen, without verbal guidance. 4. Allow the user to follow the warm tutorial and begin the game. 5. Allow the user to begin the game and play until the three lives are over.
Test Data	User Movements: Left jump, Right jump, Jumping Jack, Squat Character Movements: Left jump, Right jump, Jumping Jack, Squat
Expected Result	The expected result is that the user should be able to begin playing the game until all three lives are lost. Once the lives are lost, the user should be prompted to restart, or end the game. If the user has received higher scores than the leaderboard, the leaderboard will be updated with the username and the respective score. As a result, the user should be able to go through the testing procedure without verbal assistance

	from the tester. If the user does so, they will be able to play the game from end to end and understand the mechanics. Therefore, this would result in a pass.
Tester name	Jewel Dsouza and Mariam Al Thani
Test Environment	Screen: A 65” SONY TV screen connected with a web camera. OS: Mac OS & Windows

F. Evidence of Testing

Additional images to support evidence of our testing plans. For manual testing, we provide evidence through print statements in the code and multiple commits at the same time to show real-time fixing while testing. For usability testing, we show pictures of some of the think-aloud protocols and the results of the contrast checking for all of the environments.

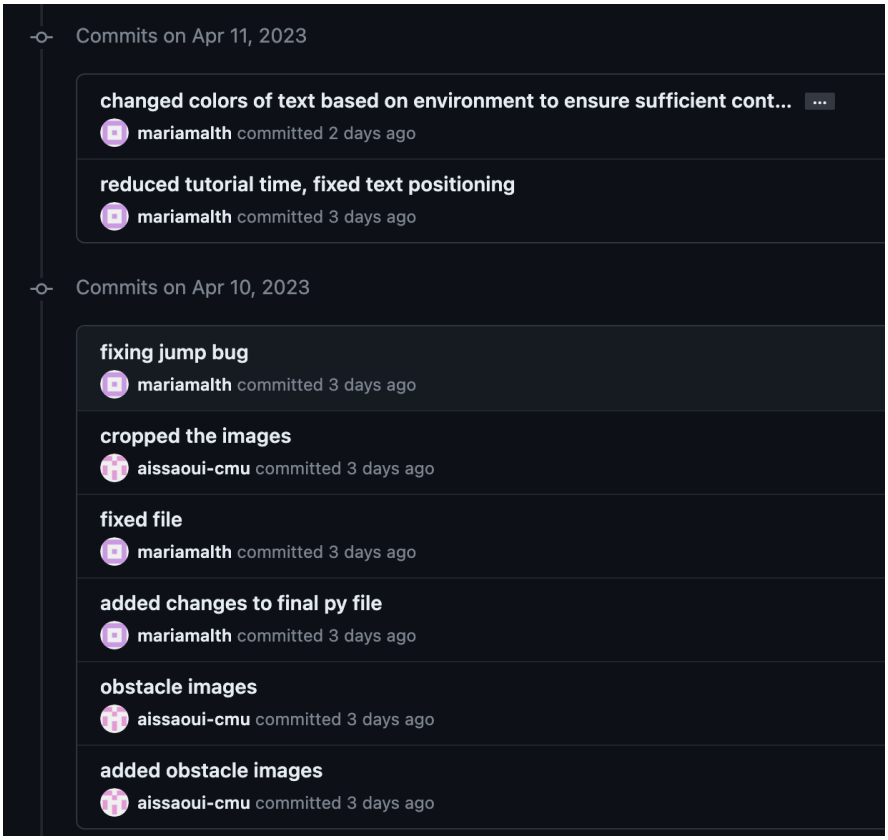
Software testing

```
# colision handling
hit_obstacle = False
if player.landmarks:
    for obstacle in obstacles:
        pass
    print(f"{obstacle.style} , {obstacle.y},{y1},{y3}")
    if obstacle.y > y1 and obstacle.y < y3:
        print("y1 good")
        for x, y in player.landmarks:
            temp_rect = pygame.Rect(x, y, 1, 1)
            if temp_rect.colliderect(obstacle.get_rect()):
                    print("coll")
                    hit_obstacle = True
                    obstacles.empty()
                    break
        if hit_obstacle == False:
                obstacles_avoided+=1
        else:
            #reset variable for next obstacle
            hit_obstacle = False
        print(str(hit_obstacle) + " " + style)
```

Print statements in the code to locate source of obstacles not colliding issue

```
tutorial_hands_joined: False
 tutorial_point: not started
 tutorial_completed: False
 game started: False
 player lives: 3
 player username: VCGEX
```

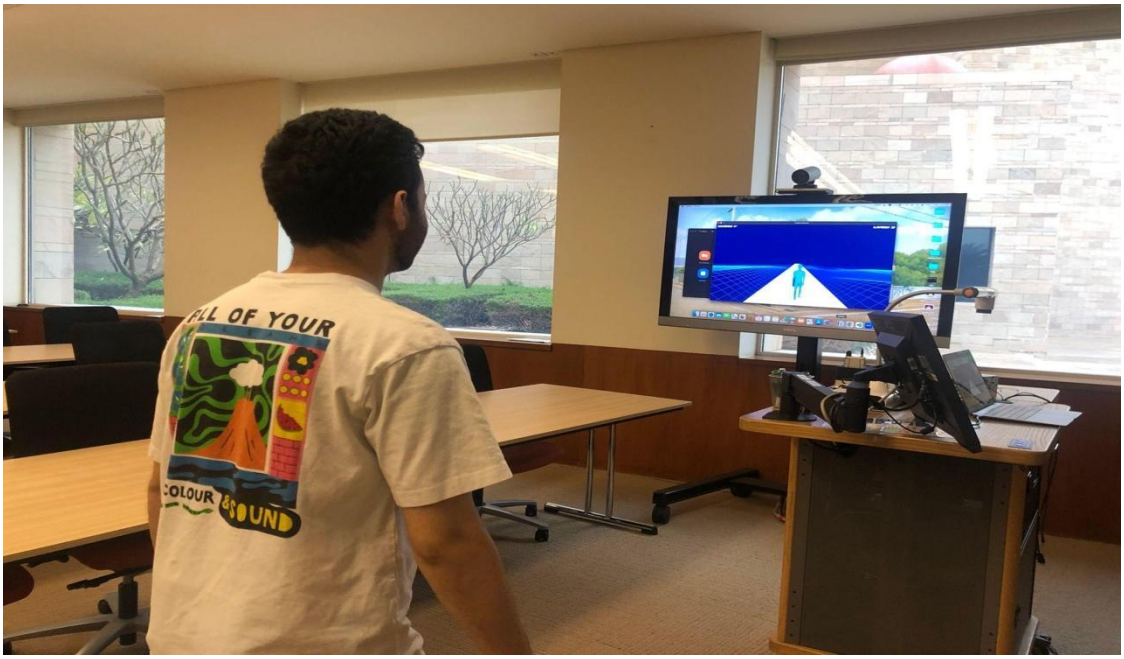
Print statements in the terminal to highlight state of session throughout



Series of commits one after another to highlight real-time fixing of issues while testing.

Usability Testing

Think-aloud protocols



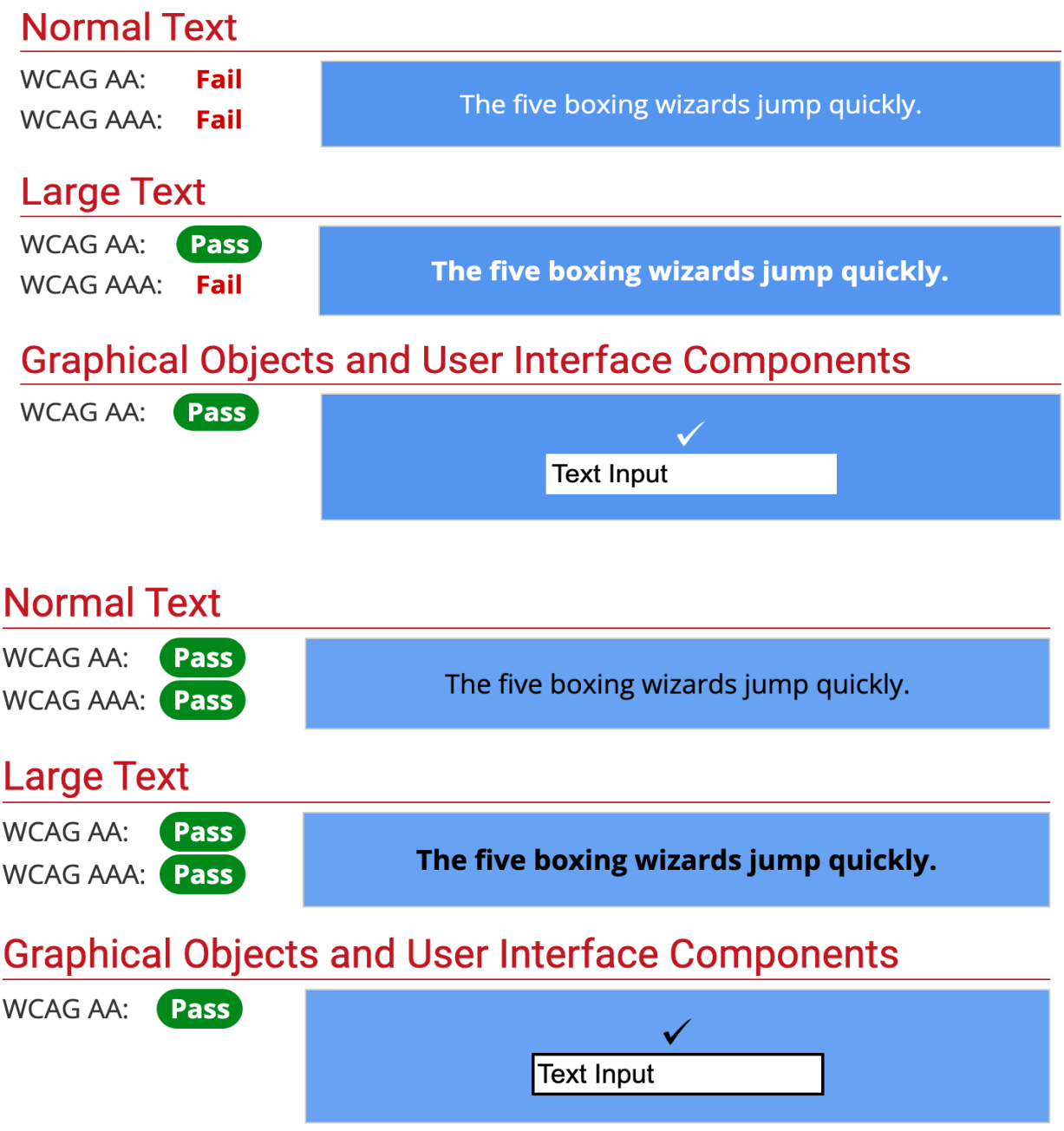
One user undergoing the think-aloud protocol.

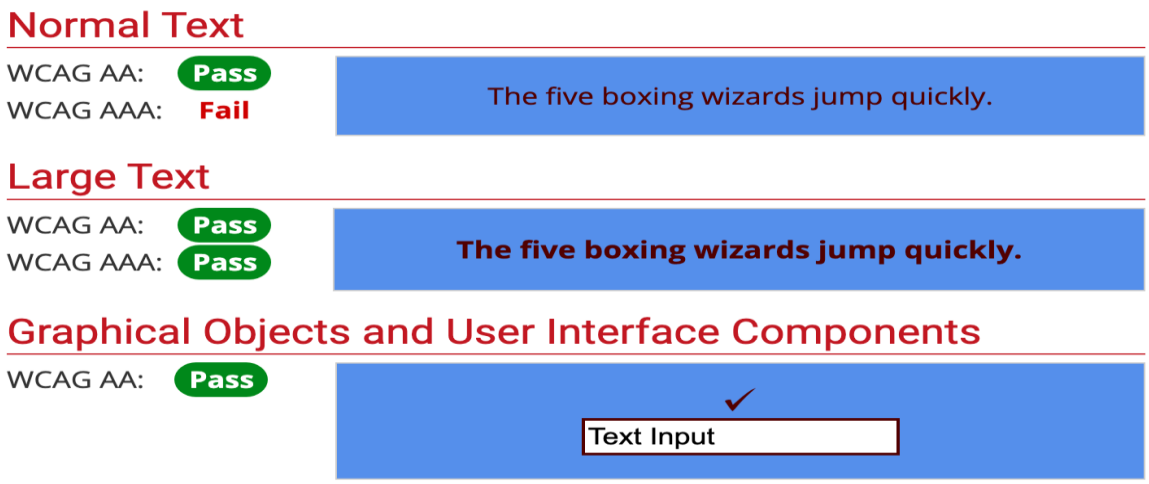
Another user (hidden for privacy) undergoing the think-aloud protocol.

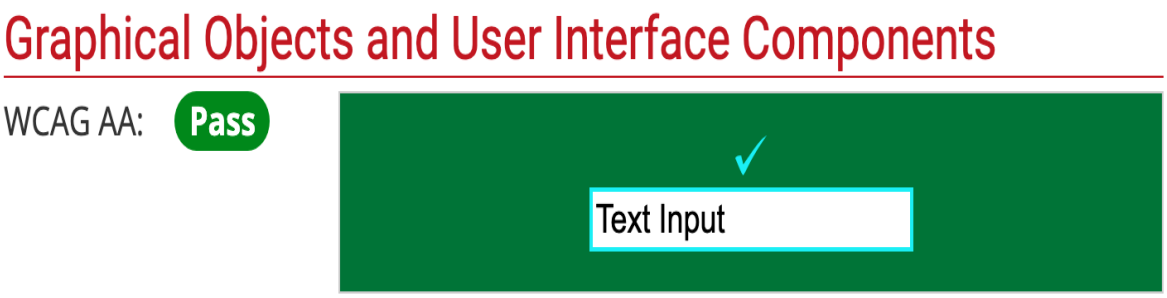
Contrast checking between text colors and environment backgrounds.

Note: all text in the experience is defined (by us) as being large text.

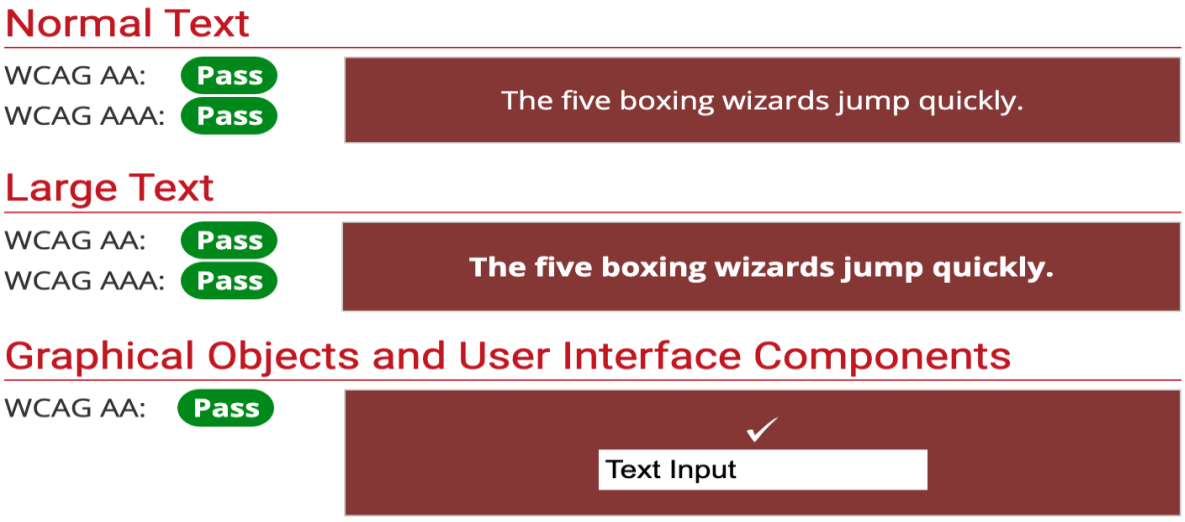
Mountainscape




Normal Text
WCAG AA: Pass
WCAG AAA: Fail
The five boxing wizards jump quickly.
Large Text
WCAG AA: Pass
WCAG AAA: Pass
The five boxing wizards jump quickly.
Graphical Objects and User Interface Components
WCAG AA: Pass
Text Input


Graphical Objects and User Interface Components
WCAG AA: Pass
Text Input

Oceanscape


Normal Text
WCAG AA: Pass
WCAG AAA: Pass
The five boxing wizards jump quickly.
Large Text
WCAG AA: Pass
WCAG AAA: Pass
The five boxing wizards jump quickly.
Graphical Objects and User Interface Components
WCAG AA: Pass
Text Input

Normal Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

Large Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: Pass

Normal Text

WCAG AA: Pass
WCAG AAA: Fail

The five boxing wizards jump quickly.

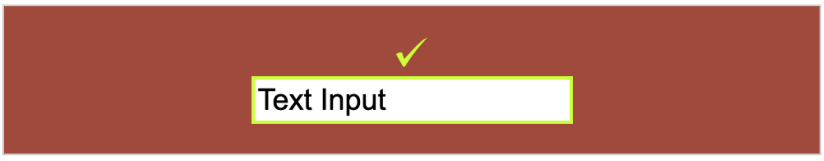
Large Text

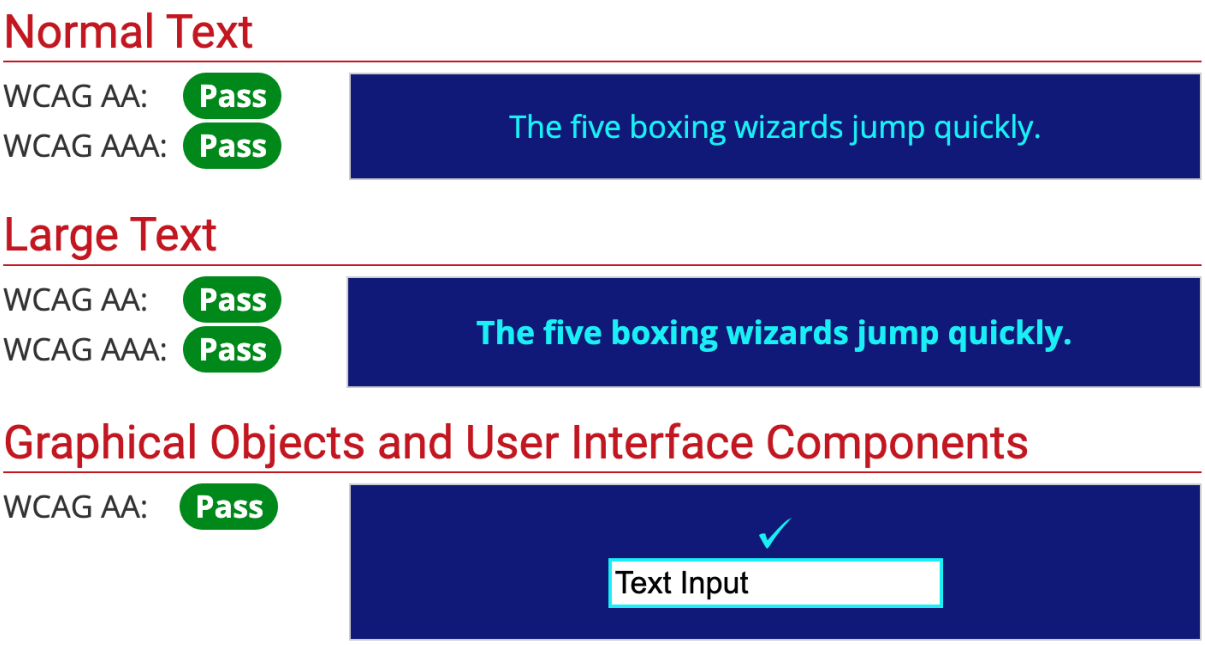
WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

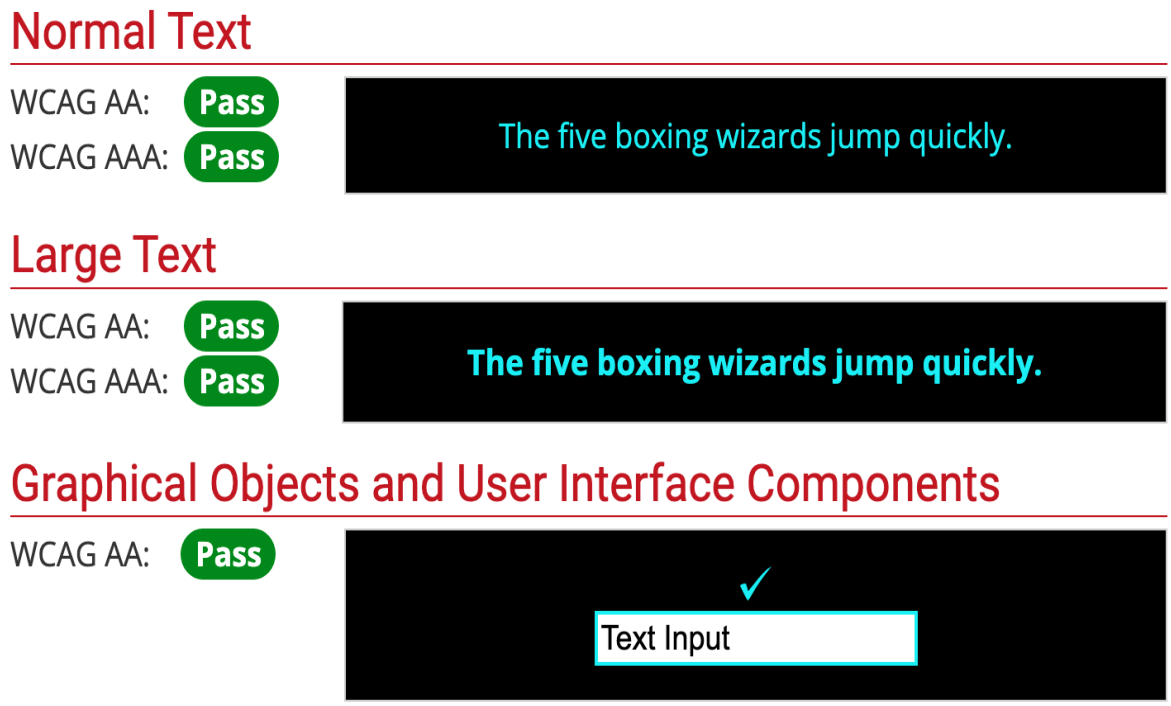
Graphical Objects and User Interface Components

WCAG AA: Pass





Desertscape



Normal Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

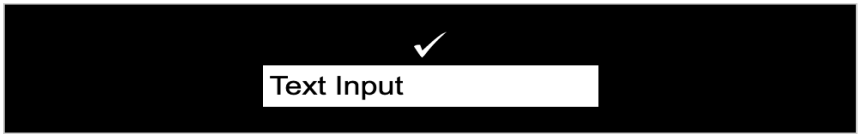
Large Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: Pass



Normal Text

WCAG AA: Pass
WCAG AAA: Fail

The five boxing wizards jump quickly.

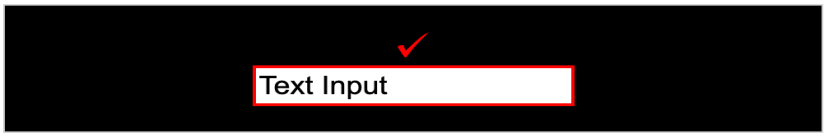
Large Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: Pass



Normal Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

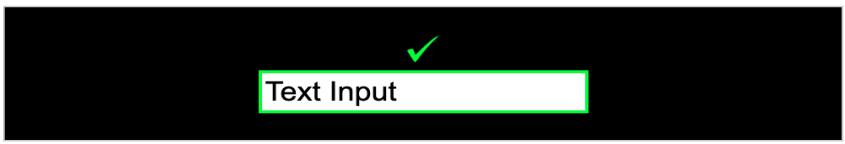
Large Text

WCAG AA: Pass
WCAG AAA: Pass

The five boxing wizards jump quickly.

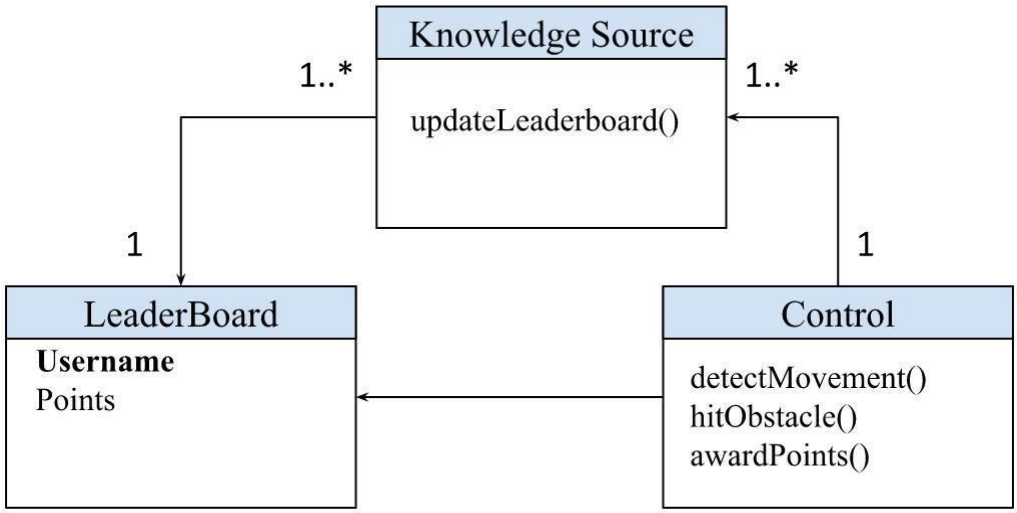
Graphical Objects and User Interface Components

WCAG AA: Pass



G. Database Relational Schema

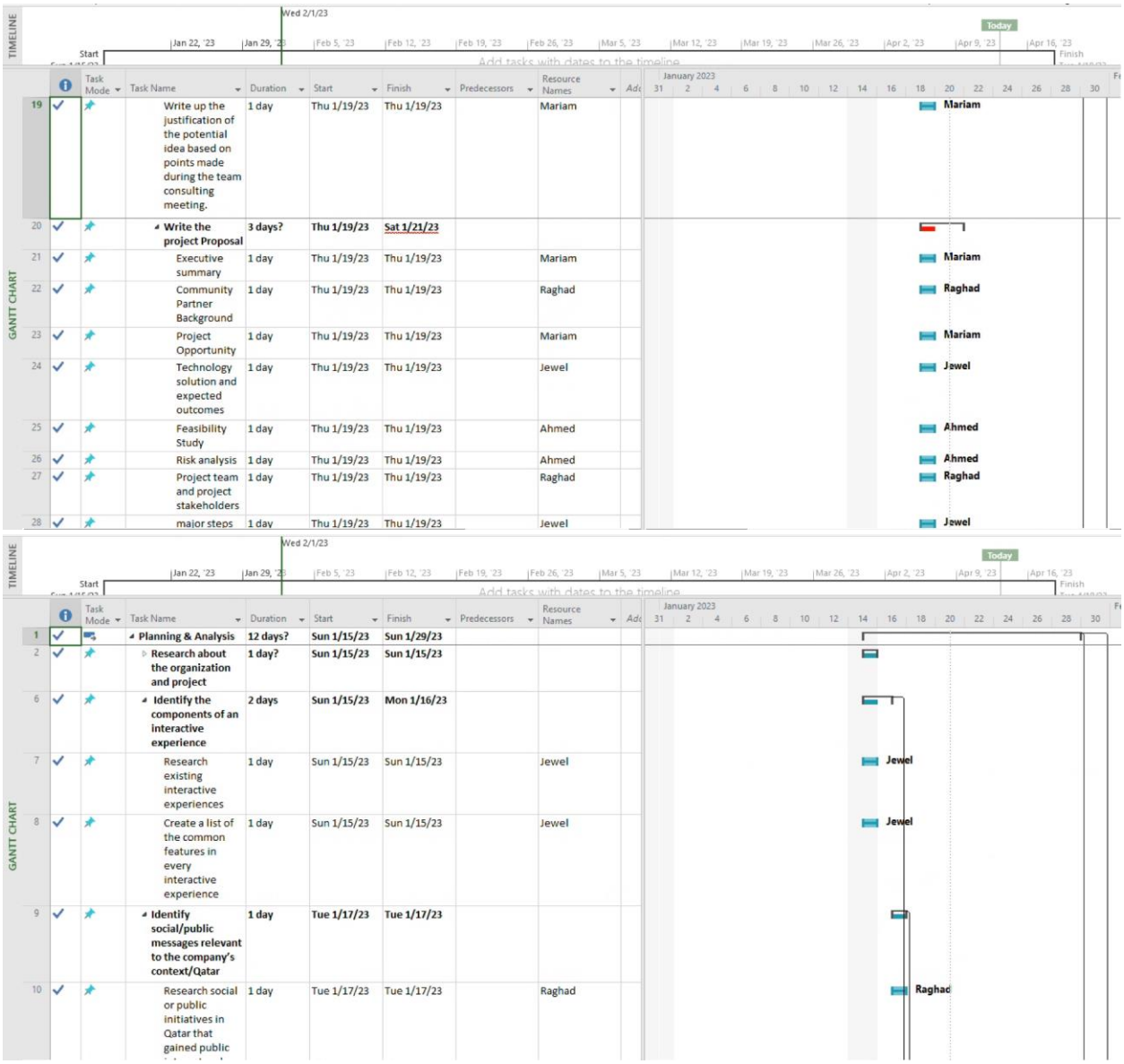
The relational schema of our system that highlights the application of the blackboard pattern and the data stored from the experience.



Project Schedule and Timeline

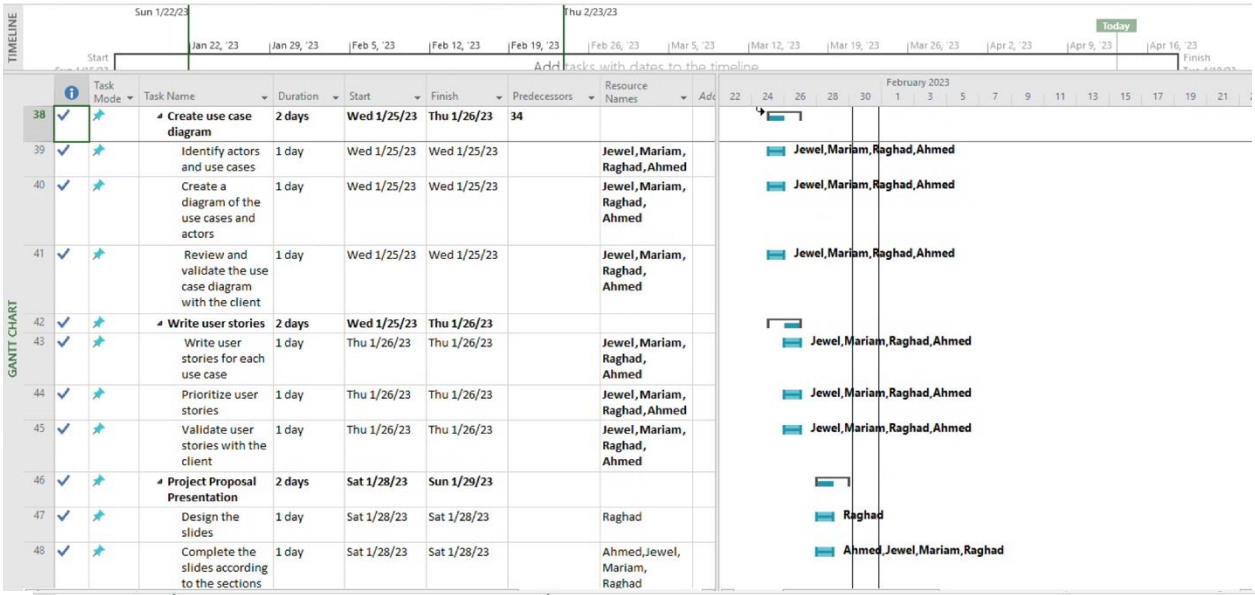
H. Project Gantt Chart

The completed Gantt chart that contains all of our project phases, tasks, schedule, and assignments.

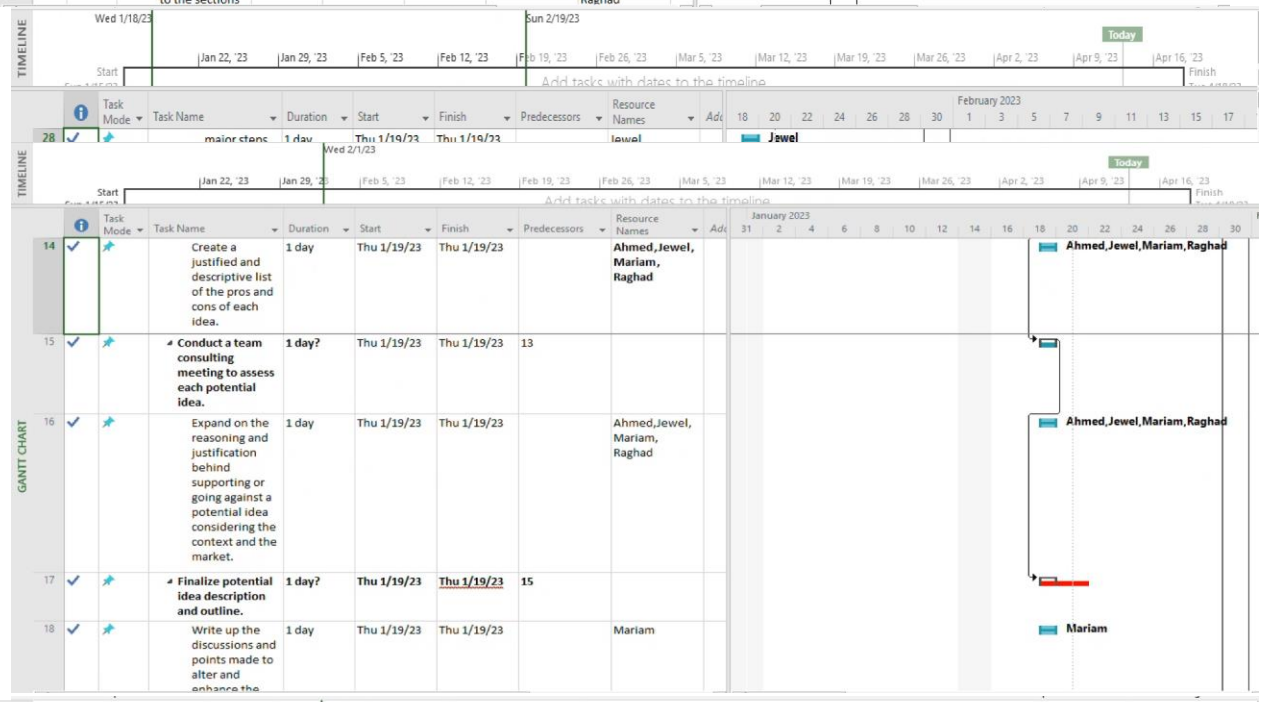


	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
19		Write up the justification of the potential idea based on points made during the team consulting meeting.	1 day	Thu 1/19/23	Thu 1/19/23		Mariam
20		**Write the project Proposal**	**3 days?**	**Thu 1/19/23**	**Sat 1/21/23**		
21		Executive summary	1 day	Thu 1/19/23	Thu 1/19/23		Mariam
22		Community Partner Background	1 day	Thu 1/19/23	Thu 1/19/23		Raghad
23		Project Opportunity	1 day	Thu 1/19/23	Thu 1/19/23		Mariam
24		Technology solution and expected outcomes	1 day	Thu 1/19/23	Thu 1/19/23		Jewel
25		Feasibility Study	1 day	Thu 1/19/23	Thu 1/19/23		Ahmed
26		Risk analysis	1 day	Thu 1/19/23	Thu 1/19/23		Ahmed
27		Project team and project stakeholders	1 day	Thu 1/19/23	Thu 1/19/23		Raghad
28		major steps	1 day	Thu 1/19/23	Thu 1/19/23		Jewel

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Planning & Analysis**	**12 days?**	**Sun 1/15/23**	**Sun 1/29/23**		
2		**Research about the organization and project**	**1 day?**	**Sun 1/15/23**	**Sun 1/15/23**		
6		**Identify the components of an interactive experience**	**2 days**	**Sun 1/15/23**	**Mon 1/16/23**		
7		Research existing interactive experiences	1 day	Sun 1/15/23	Sun 1/15/23		Jewel
8		Create a list of the common features in every interactive experience	1 day	Sun 1/15/23	Sun 1/15/23		Jewel
9		**Identify social/public messages relevant to the company's context/Qatar**	**1 day**	**Tue 1/17/23**	**Tue 1/17/23**		
10		Research social or public initiatives in Qatar that gained public	1 day	Tue 1/17/23	Tue 1/17/23		Raghad

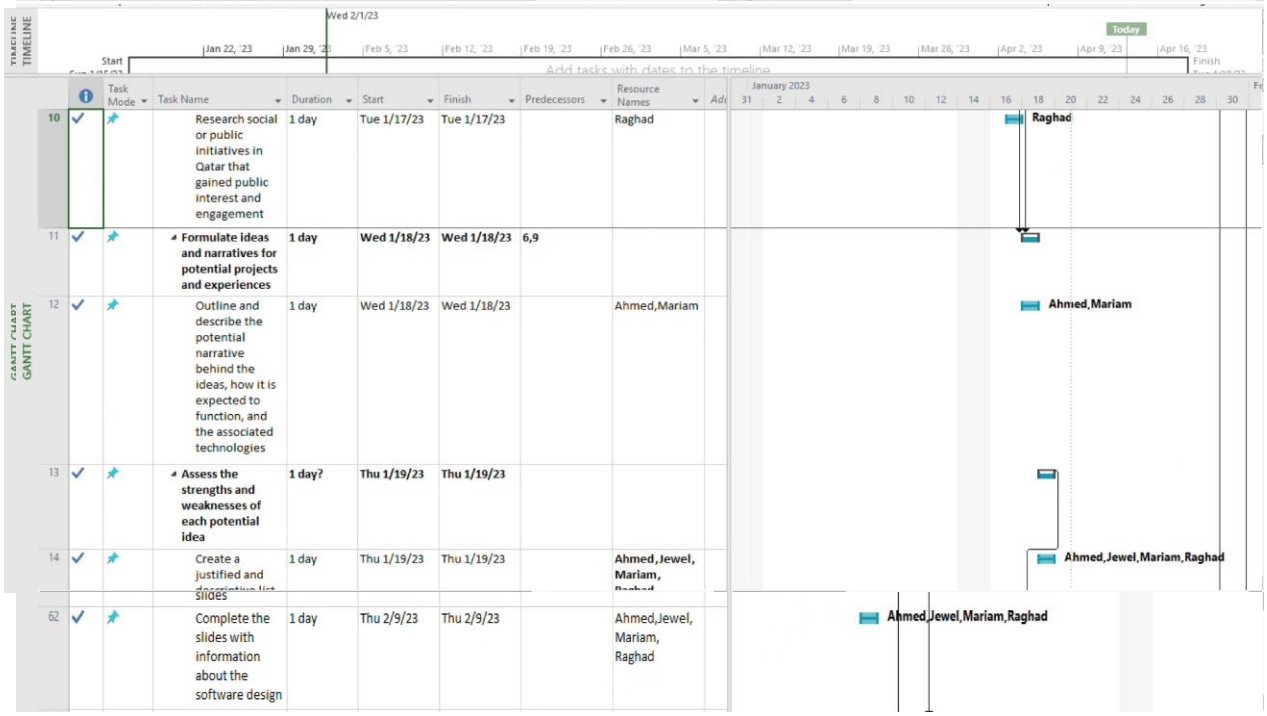


	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
38	✓	**Create use case diagram**	**2 days**	**Wed 1/25/23**	**Thu 1/26/23**	**34**	
39	✓	Identify actors and use cases	1 day	Wed 1/25/23	Wed 1/25/23		**Jewel,Mariam, Raghad,Ahmed**
40	✓	Create a diagram of the use cases and actors	1 day	Wed 1/25/23	Wed 1/25/23		**Jewel,Mariam, Raghad, Ahmed**
41	✓	Review and validate the use case diagram with the client	1 day	Wed 1/25/23	Wed 1/25/23		**Jewel,Mariam, Raghad, Ahmed**
42	✓	**Write user stories**	**2 days**	**Wed 1/25/23**	**Thu 1/26/23**		
43	✓	Write user stories for each use case	1 day	Thu 1/26/23	Thu 1/26/23		**Jewel,Mariam, Raghad, Ahmed**
44	✓	Prioritize user stories	1 day	Thu 1/26/23	Thu 1/26/23		**Jewel,Mariam, Raghad,Ahmed**
45	✓	Validate user stories with the client	1 day	Thu 1/26/23	Thu 1/26/23		**Jewel,Mariam, Raghad, Ahmed**
46	✓	**Project Proposal Presentation**	**2 days**	**Sat 1/28/23**	**Sun 1/29/23**		
47	✓	Design the slides	1 day	Sat 1/28/23	Sat 1/28/23		Raghad
48	✓	Complete the slides according to the sections	1 day	Sat 1/28/23	Sat 1/28/23		Ahmed,Jewel, Mariam, Raghad

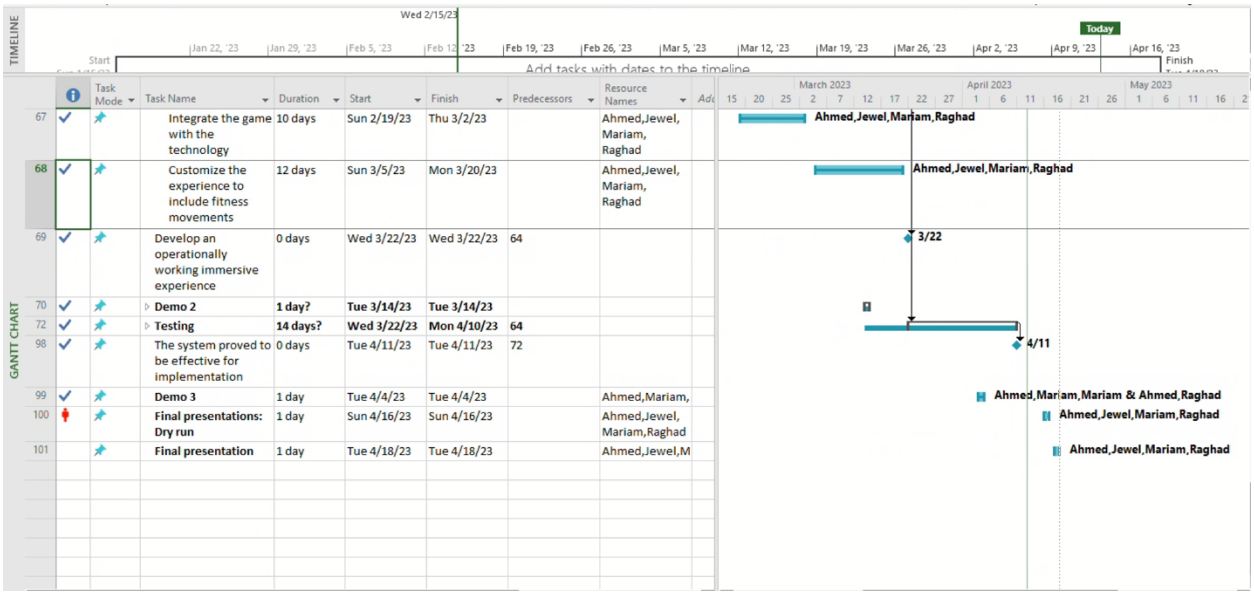


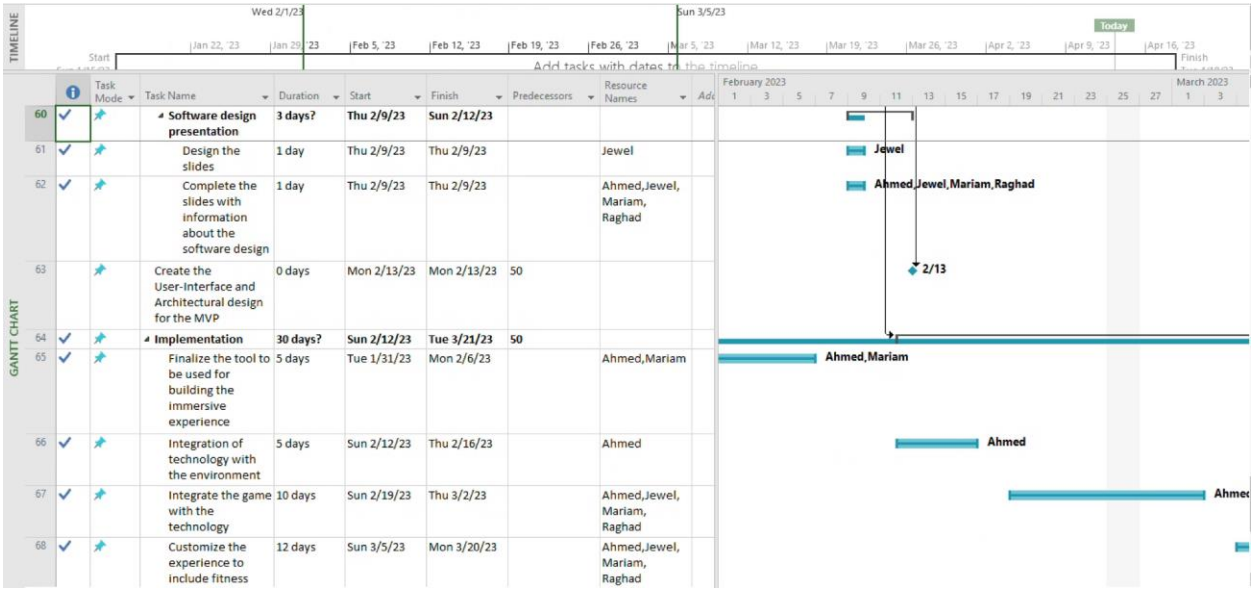
	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
28	✓	major steps	1 day	Thu 1/19/23	Thu 1/19/23		Jewel

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
14	✓	Create a justified and descriptive list of the pros and cons of each idea.	1 day	Thu 1/19/23	Thu 1/19/23		**Ahmed,Jewel, Mariam, Raghad**
15	✓	**Conduct a team consulting meeting to assess each potential idea.**	**1 day?**	Thu 1/19/23	Thu 1/19/23	13	
16	✓	Expand on the reasoning and justification behind supporting or going against a potential idea considering the context and the market.	1 day	Thu 1/19/23	Thu 1/19/23		Ahmed,Jewel, Mariam, Raghad
17	✓	**Finalize potential idea description and outline.**	**1 day?**	**Thu 1/19/23**	**Thu 1/19/23**	**15**	
18	✓	Write up the discussions and points made to alter and enhance the	1 day	Thu 1/19/23	Thu 1/19/23		Mariam



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
10	✓	Research social or public initiatives in Qatar that gained public interest and engagement	1 day	Tue 1/17/23	Tue 1/17/23		Raghad
11	✓	**Formulate ideas and narratives for potential projects and experiences**	**1 day**	**Wed 1/18/23**	**Wed 1/18/23**	**6,9**	
12	✓	Outline and describe the potential narrative behind the ideas, how it is expected to function, and the associated technologies	1 day	Wed 1/18/23	Wed 1/18/23		Ahmed,Mariam
13	✓	**Assess the strengths and weaknesses of each potential idea**	**1 day?**	**Thu 1/19/23**	**Thu 1/19/23**		
14	✓	Create a justified and descriptive list slides	1 day	Thu 1/19/23	Thu 1/19/23		**Ahmed,Jewel, Mariam,**
62	✓	Complete the slides with information about the software design	1 day	Thu 2/9/23	Thu 2/9/23		Ahmed,Jewel, Mariam, Raghad


TIMELINE
Wed 2/15/23
Today
Start
Add tasks with dates to the timeline
Finish
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
Predecessors
Resource Names
67 Integrate the game with the technology 10 days Sun 2/19/23 Thu 3/2/23 Ahmed,Jewel, Mariam, Raghad
68 Customize the experience to include fitness movements 12 days Sun 3/5/23 Mon 3/20/23 Ahmed,Jewel, Mariam, Raghad
69 Develop an operationally working immersive experience 0 days Wed 3/22/23 Wed 3/22/23 64
70 Demo 2 1 day? Tue 3/14/23 Tue 3/14/23
72 Testing 14 days? Wed 3/22/23 Mon 4/10/23 64
98 The system proved to be effective for implementation 0 days Tue 4/11/23 Tue 4/11/23 72
99 Demo 3 1 day Tue 4/4/23 Tue 4/4/23 Ahmed,Mariam,
100 Final presentations: Dry run 1 day Sun 4/16/23 Sun 4/16/23 Ahmed,Jewel, Mariam,Raghad
101 Final presentation 1 day Tue 4/18/23 Tue 4/18/23 Ahmed,Jewel,M
March 2023
April 2023
May 2023
Ahmed,Jewel,Mariam,Raghad
Ahmed,Jewel,Mariam,Raghad
3/22
4/11
Ahmed,Mariam,Mariam & Ahmed,Raghad
Ahmed,Jewel,Mariam,Raghad
Ahmed,Jewel,Mariam,Raghad


TIMELINE
Wed 2/1/23
Sun 3/5/23
Today
Start
Add tasks with dates to the timeline
Finish
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
Predecessors
Resource Names
60 Software design presentation 3 days? Thu 2/9/23 Sun 2/12/23
61 Design the slides 1 day Thu 2/9/23 Thu 2/9/23 Jewel
62 Complete the slides with information about the software design 1 day Thu 2/9/23 Thu 2/9/23 Ahmed,Jewel, Mariam, Raghad
63 Create the User-Interface and Architectural design for the MVP 0 days Mon 2/13/23 Mon 2/13/23 50
64 Implementation 30 days? Sun 2/12/23 Tue 3/21/23 50
65 Finalize the tool to be used for building the immersive experience 5 days Tue 1/31/23 Mon 2/6/23 Ahmed,Mariam
66 Integration of technology with the environment 5 days Sun 2/12/23 Thu 2/16/23 Ahmed
67 Integrate the game with the technology 10 days Sun 2/19/23 Thu 3/2/23 Ahmed,Jewel, Mariam, Raghad
68 Customize the experience to include fitness 12 days Sun 3/5/23 Mon 3/20/23 Ahmed,Jewel, Mariam, Raghad
February 2023
March 2023
Jewel
Ahmed,Jewel,Mariam,Raghad
2/13
Ahmed,Mariam
Ahmed

Documentation and Training Material

I. Documentation & Training Material

All of the final documentation and training material related to the project.

User Experience Design Documentation

Introduction

The User Experience (UX) designer document presents an in-depth reference to the UX design and development process for our immersive fitness experience. This document aims to provide a glimpse into how we have designed and structured the UX to be simple, effective, and entertaining for our users. The audience to be served for this document comprises the developers, stakeholders, and any individual engaged in the conceptualization and creation of the experience, mainly in STA. Through this document, we intend to foster a common understanding of the UX and improve collaboration among members of the team. The following guide includes user research, personas, storyboards, flow of the user experience, visual design of the character, environment and fonts used, and finally the results of user testing.

Research

I. Objectives

As part of our effort to develop a user-focused design, we conducted an extensive research to acquire insights concerning our audience, as well as the in-depth nature of the experience. The study sought to comprehend users' characteristics associated with the intended application of the experience, determine user requirements, goals and expectations, and learn about their preferences regarding the attributes and functions of the experience, as well as assess the ease of use while identifying any areas for enhancement.

II. Methods

We employed the following research approaches to attain our objectives:

- Market analysis: we carried out market research to uncover potential opportunities by identifying conceivable gaps in the market that could potentially be addressed with an innovative immersive experience. In addition, we investigated and examined previous immersive experiences in order to identify emerging techniques to distinguish ourselves and exceed users' expectations. Furthermore, we studied market trends in the market by analyzing users' practices, and upcoming technologies in the market.

- Product analysis: we performed product analysis to identify the experience's abilities, shortcomings, prospects, and challenges. We researched current immersive experiences in order to learn about fundamental features and abilities such as what it essentially accomplishes, its method of operation, and what positive effects it delivers to users. In

addition, we looked at various designs including its physical characteristics, structure and functionality. Furthermore, we evaluated the experience's quality considering its enduring nature and consistency, as well as its efficacy in conveying the message.

- Usability testing: we conducted usability tests to assess the user experience to ascertain that the experience is easy to navigate, successful and effective, along with identifying potential areas for improvement.

III. Results

Our research findings influenced our choices regarding design and assisted us in selecting the characteristics and functions that are going to provide the greatest impact to prospective users. Among the important results are:

- Qatar is notable for and has publicly acknowledged the widespread prevalence of unhealthy lifestyles, obesity, and diabetes among its whole population (Al-Thani et al., 2018; Chrismas et al., 2019; Cheema et al., 2022). Moreover, the country adopted a number of measures to address the problem (Aspetar Orthopaedic and Sports Medicine Hospital., 2021; Weill Cornell Medicine, 2022). These preliminary findings suggest fitness as a key message to convey.

- Immersive experiences have been successfully introduced in Qatar. The themes ascribed from already existing experiences spanned from culture to environment to global communication. However, according to our study, there has been no immersive fitness experience developed before in Qatar, making it a one-of-a-kind experience.

- Immersive fitness experiences established outside Qatar primarily involve a simulated setting that imitates a fictional environment in which the user perceives that they are present in the setting. In addition, the experience included an immersive exercise in which the users feel that the game is tailored to their needs as they have control over the actions of the character displayed in the environment. Finally, the experience also included real-time information on how the user performs throughout the game, measured in different metrics. These findings are the fundamental features identified in each immersive fitness experience and have been used to guide the development process.

- Concepts for immersive fitness environments aren't merely restricted to gaming scenarios or limitless expanses, but may also include real-world settings such as cities or towns. Game setting and infinite spaces, on the other hand, have been commonly augmented to give an entertaining yet challenging immersive experience. This is mostly because game settings, similar to video games, are capable of providing users with an enjoyable and participatory experience that includes goals and incentivizes users to exercise more. Meanwhile, endless settings allowed individuals to travel in any fashion they envisioned,

with obstacles surfacing arbitrarily in order to preserve the interest of users. By combining video game settings with abstract realistic environments and our technical abilities, we aim to create a unique selling point for users by making them feel that they are in a different world.

- Numerous contemporary immersive experiences entail showcasing users' motions or enabling them to make their way around inside the virtual world without any observable avatar or figure. Although this strategy might provide an illusion of autonomy and immersion, it likewise may leave the user experiencing a little emptiness or without emotional interest. Consequently, we have elected to include a character in the experience to provide depth and engagement. By allowing users to possess or engage with their digital figure, they can cultivate a deeper psychological attachment to the experience, as well as a higher feeling of control inside the virtual environment. However, it is critical to ensure the simplicity of the character to make it applicable to a wide range of audience and not too specific to not exclude other users.

- Recent immersive fitness experiences frequently call for two or more individuals to participate in the experience at a given time to boost their enthusiasm and involvement. However, owing to time limits, it was not viable to include multiple people in the workout. However, we created a new technique to preserve the competitiveness of the experience by introducing a leaderboard which highlights users' scores at the end of each session. This concept empowers users to compete against themselves, as well as others who have gone through the experience with better scores, establishing an atmosphere of competitiveness and urging them to utilize the experience repeatedly. By attempting to outperform their or others' prior scores, users are more inclined to participate in the experience consistently and motivate themselves to do better.

- Participants noted areas for improvement, which mostly include providing tutorials and hints to assist the user on the anticipated movements, as well as enhancements to the character to give the user the feeling that they are part of the game and have the ability to control the game.

IV. Implications

Immersive fitness experiences possess an opportunity to promote dedication to ongoing exercise regimens by making exercise more intriguing and exciting, which in turn positively impacts physical health. By offering an interactive and engaging atmosphere that simulates and challenges individuals, it can boost individual's motivation, as well as their participation in exercises. Therefore, In order to establish an ambiance that excites the senses and grabs the interest of users, the design should include components such as modern technology, dynamic lighting, and sound systems. In addition, exercise has been demonstrated to improve mental health, and the proposed experience can augment these benefits by offering a feeling of belonging, socialization

and supportive reinforcement. Therefore, the experience will have to incorporate elements that would allow them to interact with other users. Moreover, the potential of the immersive fitness experience to appeal to a wide range of audience could result in the rise of new business opportunities that have the opportunity to cause havoc in the conventional fitness sector through introducing novel and creative methods to exercise. Thus, the design should address the various needs and interests of potential customers to attract a wide audience. This might involve developing distinct themes or experiences for various ages, or levels of physical activity.

User Personas

Our immersive fitness experience seeks to deliver an innovative and enjoyable exercise session for everyone, irrespective of fitness level or expertise. While our platform may be utilized by a broad spectrum of audiences, we have only a single user group which provides consistent and effective functionality. The reason for this is because our platform is built on a certain set of functions that have been meticulously created to improve the user's exercise experience. These features are intended to be useful for all users, regardless of their own fitness expertise.

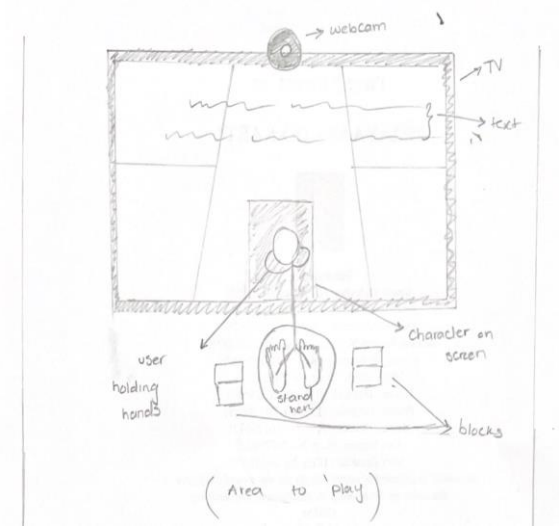
Although certain users may have unique objectives or habits, our experience strives to give a complete training experience that meets a wide range of fitness requirements. As opposed to categorizing our user base according to distinct goals or interests, we believe sustaining a single user group enables us to focus on providing the greatest possible exercise experience to all consumers.

Overall, we believe that our immersive fitness experience will present individuals with a difficult and entertaining exercise experience, regardless of their specific objectives or interests. Our experience is intended for a diverse variety of users, and we are dedicated to providing a consistent and successful training experience for everyone who utilizes it.

Storyboards

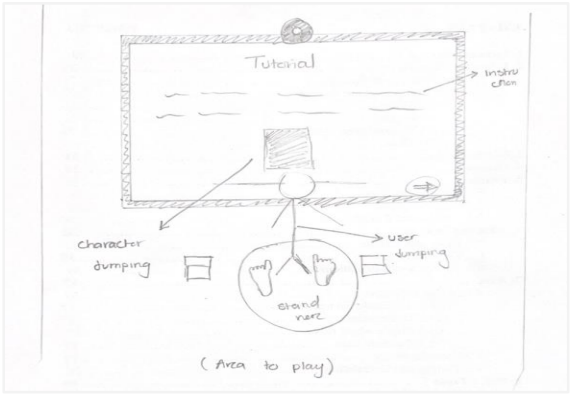
We designed a storyboard to depict an ordinary user experience in order to more effectively convey and communicate the experience, and to emphasize the essential components of the immersive fitness experience. The following information represents the storyboard.

I. Start Screen



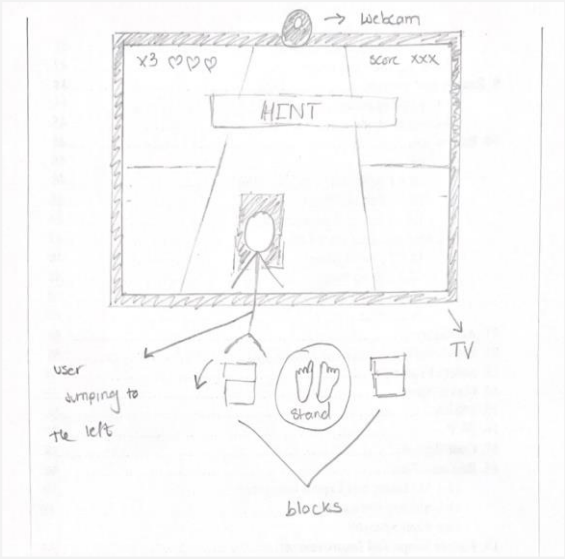
The immersive experience aims to provide users with a lively and participatory environment that stimulates them on various fronts. The digital environment has been configured to continue moving indefinitely, delivering a feeling of perpetual motion and velocity to the immersion of the experience. A message is also displayed on the screen, asking users to connect their hands, establishing an in-person connection between users and commencing the immersive experience. In addition, there will also be stickers on the floor to indicate to the user where they should be when they move to the left, right or center. This configuration delivers an experience that pushes users to come together and get active.

II. Tutorial



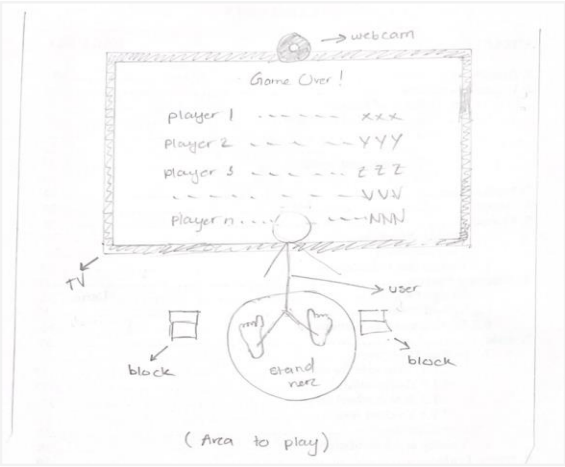
After joining hands, there will be a tutorial page to demonstrate how to carry out the moves to avoid the obstacles is a vital element of the configuration. The tutorial will aid users in avoiding accidents through making certain that they learn how to execute the activities properly and securely. It also guarantees that users make the greatest possible use of the experience by performing the actions correctly. Finally, the tutorial could also improve the whole experience by offering an explicit knowledge of the workout's objectives, allowing users to solely concentrate on attaining their fitness goals.

III. Game Start



The user is taken to a simulated environment that mimics the endless expanse seen on the start screen. The setting also comprises a figure that the user controls, as well as obstacles that pop up unpredictable as the game progresses. The user controls the character's motions, which must be accomplished in order to circumvent the obstacles and progress through the game. The setting also features the lives available to the user on the left side, whereas the scores are included on the right side of the screen. In addition, written instructions will be displayed on the screen, indicating what actions the player should perform next to escape the obstacles and finish the game. If the user hits an obstacle, they will lose a life. Otherwise, they will gain extra points for avoiding the obstacles.

IV. Game End



The experience will ultimately end when the user runs into three hurdles and has lost all of the three lives. At that moment, the setting switches to show a leaderboard with the game's highest ten rankings. This allows the user to assess their progress to that of others, as well as potentially seek the greatest score. In the meantime, a countdown timer is also displayed with the leaderboard, allowing users to view the final results and choose whether to keep playing by rejoining hands before the timer ends. Otherwise, the game will navigate back to the instructions where other users can begin the game by joining hands.

UX Designer Style Guide

The UX design style guide's main objective is to give a thorough list of guidelines in order to maintain uniformity and cohesiveness throughout all design aspects of the experience. The style guide is the foundation for the aesthetic and user-friendly layout of the experience, presenting

critical elements to the design process of the environment, obstacles, as well as the character, which are discussed below.

I. Environment Design

The ongoing development of the virtual setting was a vital component of the development phase. Early on, the setting was intended to be less abstract, imitating a real-life setting such as Qatar's Old Port or West Bay.

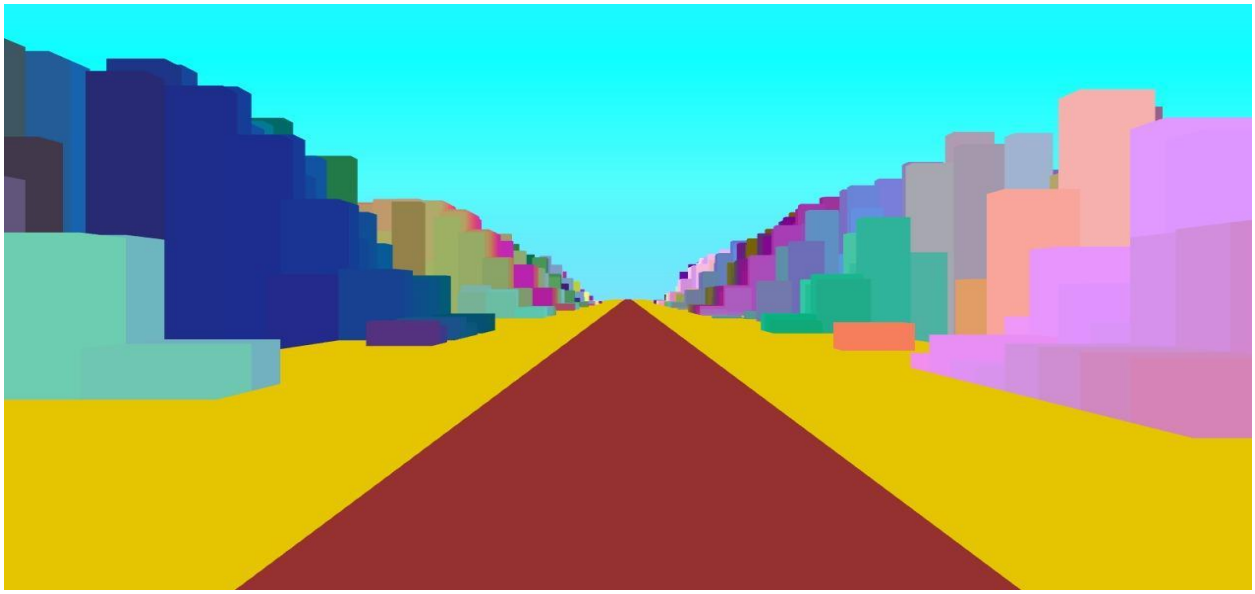


Figure 1: *First iteration of environment Design*

As the concept evolved, the team concluded that a more abstract setting would make users have a sense of attachment to the other world as discussed in the findings of the research conducted. The environment design evolved into the following:

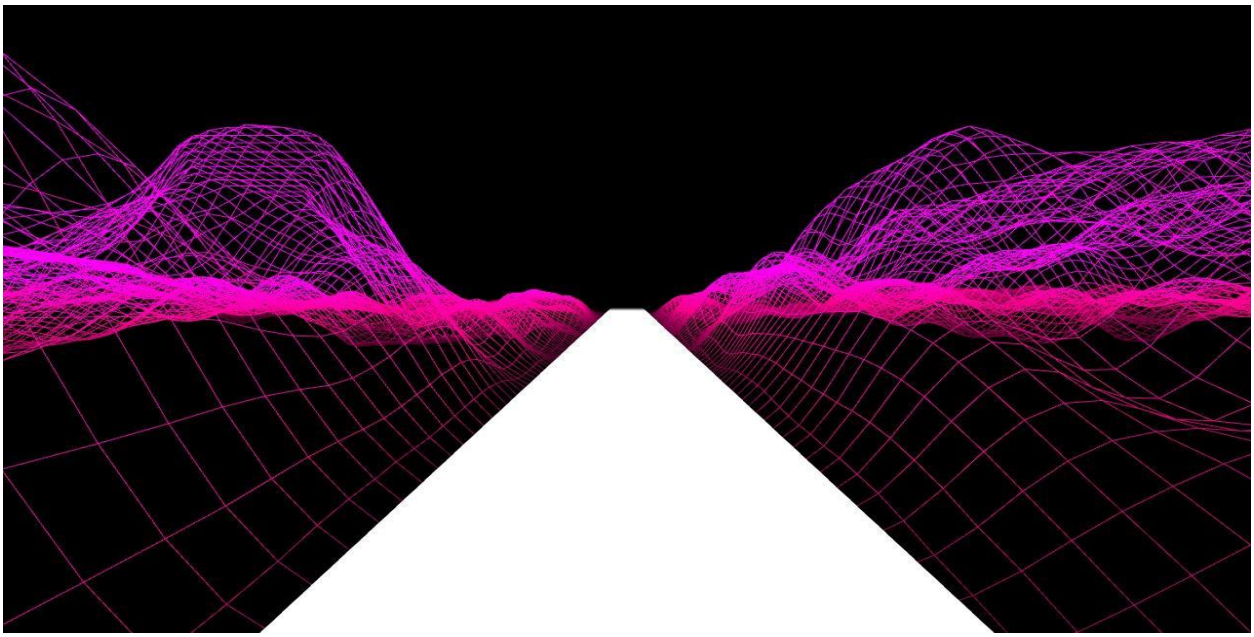
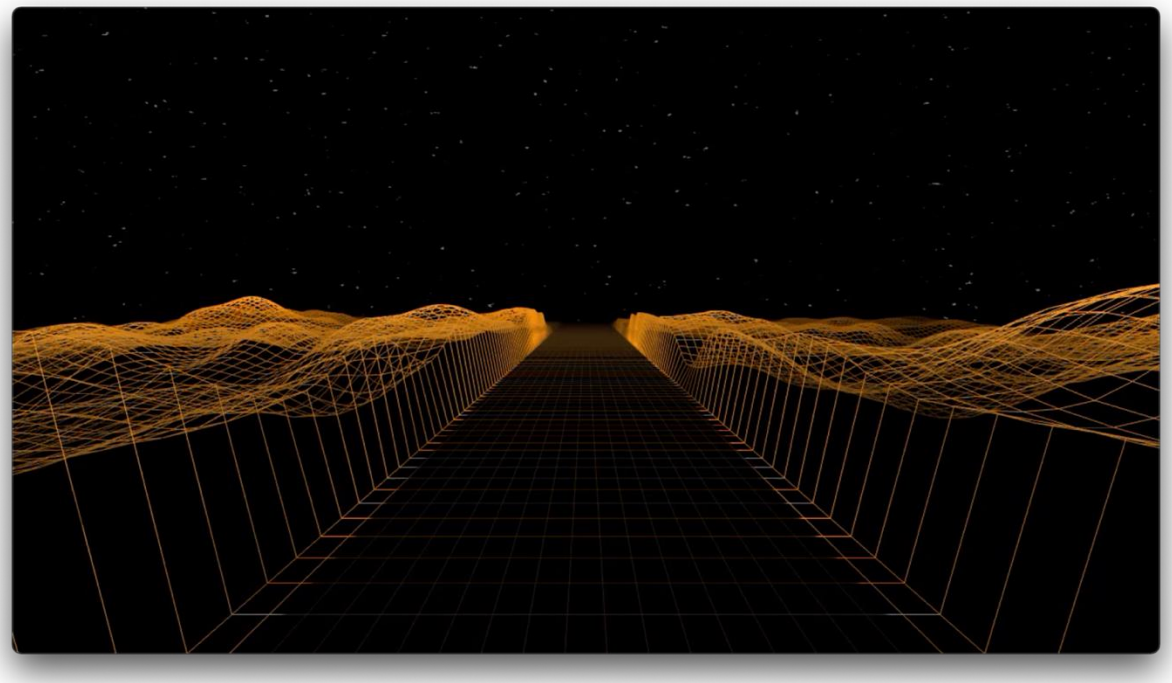


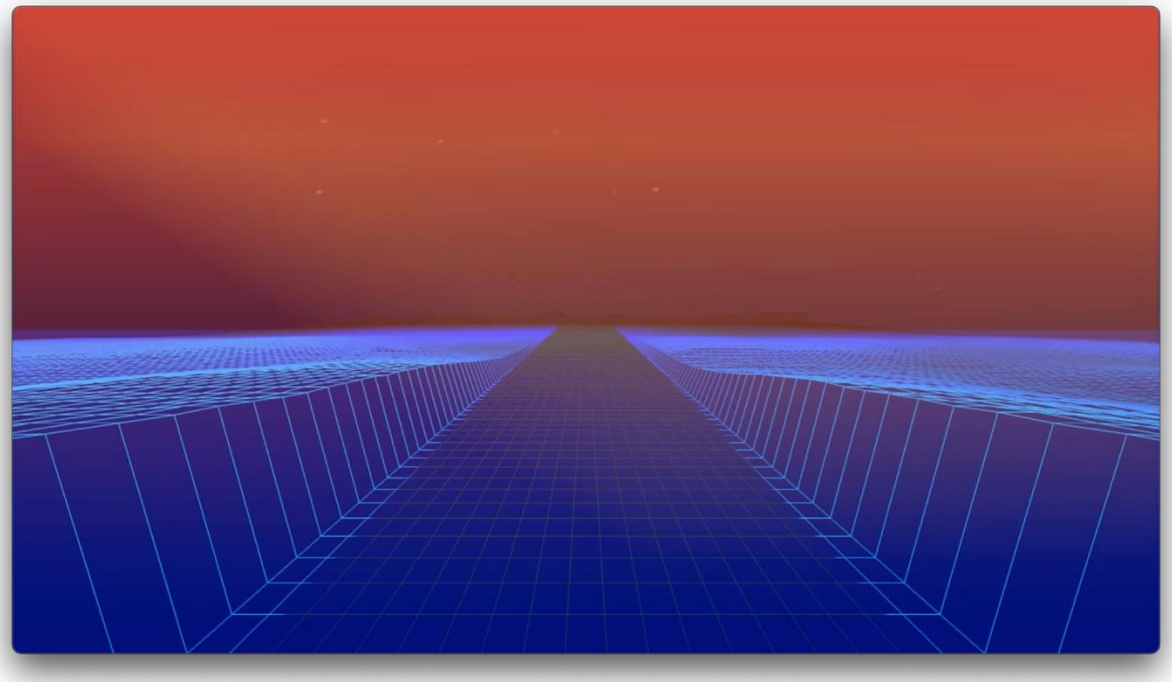
Figure 2: *Second iteration of environment design to be more abstract.*

The team investigated the different methods to build an abstract world. However, it became evident during client sessions and user feedback that the technique did not give the intriguing look and texture as anticipated by the client and users. The atmosphere was drab and unappealing to users, thus degrading the experience.

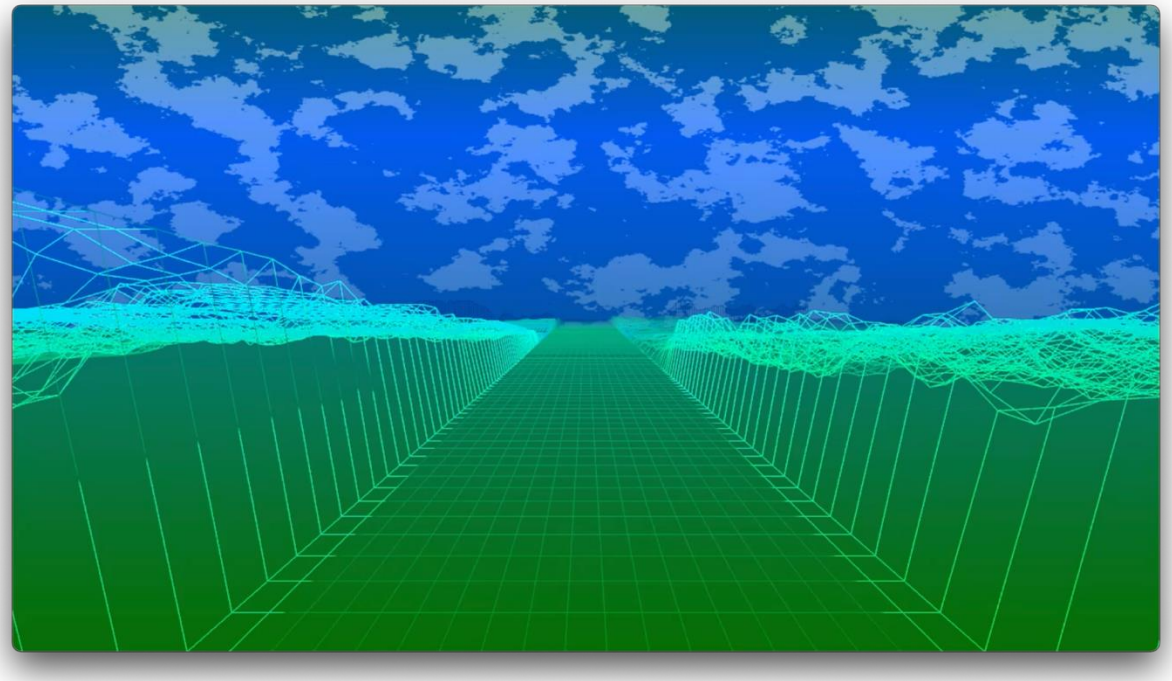
In response to the feedback received from the client and users, the team adopted a more abstract and conceptual technique for landscape layout, emphasizing on delivering an exceptional and deep experience. The team utilized conceptual elements including patterns of geometric designs, vibrant shades, and illuminating effects to establish an exotic and vivid scene. The team succeeded in managing to create an environment that made users get more attached to the other side, the experience, through the development of more abstract settings that improved the experience as a whole. In addition, the abstract setting allowed users to relax and leave their daily challenges and fully immerse themselves in the experience, making it more interesting and effective. Thus, we have finally developed successful environments that were seen as attractive by both the client and users as shown below.



1) Desert Dunes during the night



2) Ocean Landscape during sunset



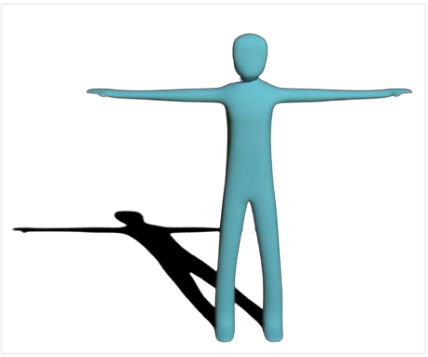
3) Mountain Landscape during the day

These three environments act as the selling point for the experience as they allow the user to experience different regions with increasing difficulty level.

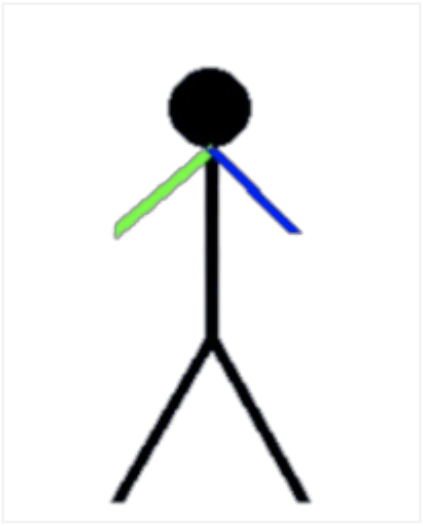
II. Character Design

The character's evolution can potentially be noticed as a succession of modifications influenced by feedback from both users and STA. For cleanliness and ease of use, the shape of the persona was first depicted as a rectangle. However, as the design evolved and the user experience was polished, a running avatar was constructed using sprite graphics as discussed in the initial requirements which was then embedded into the experience. Following feedback from STA, the team concluded that the initial avatar layout was too precise, possibly leaving our certain demographics. As a result, we streamlined the design by developing a stick figure designed to appeal to a broader audience. Further feedback suggested that users prefer a figure that mirrors their actions as it makes the experience more interactive and captivating. The following pictures show the evolution of our character:

1)



2)



3)

4)

III. Obstacle Design

The obstacle design is expected to be adapted to the unique needs of the experience, presenting obstacles suited for participants' level of endurance and objectives. The team intends to make the obstacle design more three-dimensional and blended with the surrounding environment in order to improve the immersive experience. This is mainly through employing various gradients and forms that are more representative of the environment, which is expected to make the experience more enjoyable for users. We have used different obstacles for the different movements that suit the different environments' context as shown below.

Environment Integration Manual

Purpose

This document serves as an encompassing manual on the integration between the environments and the remaining mechanisms of the system that makes up the immersive fitness experience. From this manual, you will be able to understand the relationship between the environment background and the Python-created aspects of the system, how to change the background of the experience through the Python code, and how to alter the environment's presentation.

Description of the relationship between Python & Environment Background

Currently, the system behind the experience is split into two parts. the environments are created on TouchDesigner and the remaining mechanics are created on Python, mainly through a Pygame base. The three designs created on TouchDesigner have the same dimensions as the screen presented on Python (1280x720), and are exported as separate mp4 videos that are stored in the same file directory as the Python file. The videos can be of varying lengths in time and this does not and should not affect experience presentation (unless this is something that one would like to purposely achieve in future development, visit the alteration section).

The videos of the environment, which we will refer to as the environment or the videos interchangeably, are then integrated into the Python code as follows.

Firstly, the video, or the file name of the video, is captured to store information on its size and length.

```
video = cv2.VideoCapture("background2.mp4")
```

Then, a window the same size as the video is created - this goes on top of the screen that is presented and allows both to be the same size. Here, the file name of the video is all that needs to be identified.

```
window = pygame.display.set_mode(video_image.shape[1::-1])
```

Additionally, a numerical variable, ‘frame_counter’, is created and set to equal 0. This is created for the purpose of looping the video that will be playing in the background. This variable increases by one with each event while the camera is opened. So, constantly.

The actual code that plays the video is as below. Firstly, check if the system was able to locate and read the video. In the case that it does, we turn that video into a pygame surface object titled ‘video_surf’ to allow us to present that video alongside the rest of the screen and objects on the screen.

```
success, video_image = video.read()

if success:

    video_surf = pygame.image.frombuffer(video_image.tobytes(), video_image.shape[1::-1],
"BGR")
```

After this, we simply present this video (now a pygame surface) on the window, starting from the top left corner, with (0,0) coordinates.

```
window.blit(video_surf, (0, 0))
```

The video is looped through the following code. It first is triggered when it has been identified that the number of frames passed is the number of frames in the video itself. If this is the case, we simply reset the frame counter and re-capture the video which loops it by playing it from the beginning.

```
if frame_counter == video.get(cv2.CAP_PROP_FRAME_COUNT):

    frame_counter = 0

    video = cv2.VideoCapture("background2.mp4")
```

Additional information and alterations will be made accordingly upon implementation of the three environments and the systems random choice between the three environments. In this case, a string variable with the file name will be stored in the beginning of the code and a randomizer will assign the variable one of the three file names. The rest of the aforementioned code will remain the same.

Alterations to video presentation

Speed

To change the speed of the environment in the background, one simply has to change the framerate, which is set at 120. This is found at the bottom of the code and is as follows. Increasing the framerate will speed the video up and decreasing the framerate will slow the video down.

```
clock.tick(120)
```

Looping

To modify the looping aspect of the video, for example, to loop once we reach the penultimate frame, one simply has to change the if statement that resets and recaptures the frame_counter and video, respectively.

```
if frame_counter == video.get(cv2.CAP_PROP_FRAME_COUNT):

    frame_counter = 0

    video = cv2.VideoCapture("background2.mp4")
```

Randomization

Additionally, to alter the randomness of the environment presentation, one can alter the code presented in the looping section above to reset the file name variable to be a different file name. Thus, once one video ends, it will switch it another (or the same one, depending on the random choice).

Source Code Documentation

Purpose

The purpose of this document is to document and explain all the code about the immersive fitness experience created for STA. From this document, one can learn what the code does, how to install, configure, and maintain it, opportunities for updating and future development, and the testing conducted to ensure a working, functional, and smooth experience. In addition to the aforementioned information, the author's (initial developers) information is provided at the end, and the Python file contains consistent comments throughout, describing and explaining the purpose of code snippets and their relation to the experiences' functionality and presentation.

Description of code's purpose and functionality

The code's purpose and functionality is to launch an endless runner game, in which the user can manipulate the character's actions through physical gestures.

Softwares

The code behind the experience (excluding the environment video background) is created solely on Python 3. It has been developed on multiple versions of Python 3 including 3.1, 3.7, 3.9, and are functional on all three versions. Visual Studio Code was used to develop the python files.

Python Libraries Required

The following libraries (and sub-libraries) were utilized during the development of the final code:

Pygame - 2.1.3

Pygame is the base of the code and is used for game mechanics, obstacle creation, collision detection, font presentation, and more.

OpenCV - 4.7.0.68

OpenCV is a computer vision library that specializes in real-time computer vision. It is used in combination with other libraries to capture the background video and connect with the camera.

Numpy - 1.24.1

Numpy is a mathematical-based library and is utilized in array usage and data transformation of the motion-detected landmarks to a format that is accepted by pygame.

Mediapipe - 0.9.1.0

Mediapipe is an open-source framework that uses machine learning to read data from a video or audio source and detect poses from it. It is utilized in detecting the user, their motion, creating the landmarks, the character, and more.

Pandas - 1.5.3

Pandas is a python library that is focused on data manipulation and analysis. It it utilized in updating the leaderboard file and identifying the top 10 scores.

Instructions on how to install the required libraries.

To install the required packages, navigate to the directory containing the requirements.txt file in your terminal by using the "cd" command. Once you're in the directory, execute the following command: "pip install -r requirements.txt"

Code Breakdown

I. Motion detection

```
# Initialize Mediapipe Pose Detection
mp_pose = mp.solutions.pose

# Start video capture
cap = cv2.VideoCapture(0)

def checkHandsJoined(landmarks):
    # Get the left wrist landmark x and y coordinates.
    left_wrist_landmark = (landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].x * WIDTH,
                           landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].y * HEIGHT)

    # Get the right wrist landmark x and y coordinates.
    right_wrist_landmark = (landmarks.landmark[mp_pose.PoseLandmark.RIGHT_WRIST].x * WIDTH,
                            landmarks.landmark[mp_pose.PoseLandmark.RIGHT_WRIST].y * HEIGHT)

    # Calculate the euclidean distance between the left and right wrist.
    euclidean_distance = int(math.hypot(left_wrist_landmark[0] - right_wrist_landmark[0],
                                        left_wrist_landmark[1] - right_wrist_landmark[1]))
    # Compare the distance between the wrists with a appropriate threshold to check if both hands are joined
    if euclidean_distance < 130:
        return True
    return False
```

This Python script uses the Mediapipe and OpenCV libraries to detect the pose of a person in a video capture and determine if their hands are joined or not. First, the script initializes the Mediapipe Pose Detection model and starts capturing video from the default camera. Then, it defines a function called checkHandsJoined that takes in the pose landmarks detected by the Mediapipe model as a parameter. Within this function, the x and y coordinates of the left and right wrist landmarks are retrieved, and their euclidean distance is calculated. If the distance between the wrists is less than a certain threshold value of 130, it is considered that the hands are joined, and the function returns True. Otherwise, it returns False. This function can be used in a larger

application to detect and track the pose of a person in real-time video and determine if their hands are joined or not, for example, as a gesture control for a computer or a game.

II. Player class

```
class Player():
    def __init__(self):
        self.lives = 3
        self.landmarks = None
    def update(self,landmarks):
        self.landmarks = Player.pygame_landmarks(landmarks)
    def pygame_landmarks(landmarks):
        # If landmarks are detected, draw them on frame
        if landmarks is not None:
            old = landmarks
            # Scale landmarks to make character smaller

            landmarks = np.array([(landmark.x, landmark.y, landmark.z) for landmark in landmarks.landmark])
            landmarks = landmarks[:, :2]

            factor = 0.4
            center_x = screen.get_width() / 2
            center_y = screen.get_height() * (1.2-factor)

            # Map landmarks to Pygame coordinates
            x = landmarks[:, 0]
            y = landmarks[:, 1]
            x = np.interp(x, (0, 1), (0, screen.get_width()))
            y = np.interp(y, (0, 1), (0, screen.get_height()))
            dist_x = x - center_x
            dist_y = y - center_y
            # Apply the dilation factor to the distances
            dist_x *= factor
            dist_y *= factor
            # Add the dilated distances to the center of dilation coordinates
            x = dist_x + center_x
            y = dist_y + center_y
            x = [i for i in x]
            y = [i for i in y]
            landmarks_pygame = list(zip(x, y))

            # Add points to fill the character more
            left_shoulder = landmarks_pygame[12]
            left_elbow = landmarks_pygame[14]
            left_arm = interpolate_points(left_shoulder,left_elbow, int(10*factor))
            left_hip = landmarks_pygame[24]
            left_body = interpolate_points(left_shoulder,left_hip, int(10*factor))
            left_knee = landmarks_pygame[26]
            left_thigh = interpolate_points(left_knee,left_hip, int(10*factor))
            landmarks_pygame += left_arm + left_body + left_thigh
            right_shoulder = landmarks_pygame[11]
            right_elbow = landmarks_pygame[13]
            right_arm = interpolate_points(right_shoulder, right_elbow, int(10*factor))
            right_hip = landmarks_pygame[23]
            right_body = interpolate_points(right_shoulder, right_hip, int(10*factor))
            right_knee = landmarks_pygame[25]
            right_thigh = interpolate_points(right_knee, right_hip, int(10*factor))
            landmarks_pygame += right_arm + right_body + right_thigh
            return landmarks_pygame
        return None
```

This is a Python class called Player that is used to represent a character in a game. The class has two methods: __init__ and update. In the __init__ method, the Player object is initialized with three lives and a landmarks variable that is set to None. In the update method, the landmarks variable is updated with the new pose landmarks detected by the Mediapipe model passed in as an argument. This method calls the pygame_landmarks method to convert the Mediapipe landmarks to Pygame coordinates and scales the character based on the screen size. The pygame_landmarks method takes in the Mediapipe landmarks as an argument and converts them to Pygame

coordinates, which are used to draw the character on the screen. If no landmarks are detected, the method returns None. If landmarks are detected, the landmarks are scaled to make the character smaller and mapped to Pygame coordinates. Then, the distance of each landmark from a center point is calculated and multiplied by a scaling factor to dilate the distance. The dilated distances are added to the center point coordinates to get the new coordinates of each landmark. Additional points are added to fill the character more by interpolating points between some landmarks. The interpolated points are added to the list of landmarks, and the final Pygame coordinates of all the landmarks are returned. Overall, this code is used to create a Player object in a game and update its position based on the pose landmarks detected by the Mediapipe model. It also scales the character based on the screen size and adds additional points to make the character appear more filled.

III. Obstacle class

```
# define the Obstacle class
class Obstacle(pygame.sprite.Sprite):
    s_line = 34 / 2
    b_line = 752 / 2
    distance = 370
    side = (distance) / (1-(s_line/b_line))
    speed = 5
    def __init__(self, position, style):
        super().__init__()
        self.style = style
        self.position = position
        self.color = self.get_color()
        self.height = self.get_height()
        self.width = self.get_width()
        self.y = self.get_position_y()
        self.x = self.get_position_x()

    def get_colision_line(self,nose_y,ankle_y):
        if self.style == "crouch":
            return ankle_y - 20
        elif self.style == "jump":
            return nose_y
        else:
            return (nose_y + ankle_y) / 2

    def get_color(self):
        if self.style == "stand":
            return BLUE
        elif self.style == "crouch":
            return BLACK
        else:
            return GREEN
    def get_height(self):
        initial_size = HEIGHT * 0.01
        if self.style == "stand":
            return initial_size
        else:  # Crouching or Jumping
            return initial_size / 4

    def get_width(self):
        if self.style == "stand":
            return self.height / 2
        else:  # Crouching or Jumping
            return self.height * 14

    def get_position_y(self):
        self.start_point = HEIGHT - Obstacle.distance
        if  self.style == "jump":
            self.start_point =  HEIGHT / 2 - 100
            return HEIGHT / 2 - 100
        return self.start_point
```

```
def get_position_x(self):
    center_x = WIDTH / 2 - self.width / 2
    if self.style == "stand":
        if self.position == "left":
            return center_x - self.width * 3.5
        elif self.position == "right":
            return center_x + self.width * 3.5
        else:  # center
            return center_x
    else:
        return center_x

def update(self):
    # start_point = HEIGHT - Obstacle.distance - self.height
    # move the obstacle down the screen
    self.y += Obstacle.speed + self.height / 10
    d = (self.y - self.start_point) + (Obstacle.side - Obstacle.distance)
    rate = (size(Obstacle.side,Obstacle.b_line,d)) / (Obstacle.s_line)
    self.height = self.get_height() * rate
    self.width = self.get_width()
    self.x = self.get_position_x()

def get_rect(self):
    return pygame.Rect(self.x, self.y, self.width, self.height)
```

This code defines a class called Obstacle that inherits from the Pygame Sprite class. It represents obstacles that the player must avoid in the game. The class has several class-level variables:

- **s_line, b_line**: these are values used to calculate the side distance of the obstacle
- **distance**: the distance of the obstacle from the player.
- **side**: the calculated side distance.
- **speed**: the speed at which the obstacle moves.

The class has an init method that initializes various instance-level attributes of the object such as its style (stand, crouch, or jump), position (left, center, or right), color, height, width, and x, y position. The position is calculated based on the style and position of the obstacle. The height and width of the obstacle are determined based on its style. The class also has several methods:

- **get_colision_line**: returns the collision line of the obstacle based on the player's nose and ankle positions.
- **get_color**: returns the color of the obstacle based on its style.
- **get_height**: returns the height of the obstacle based on its style.

- **get_width**: returns the width of the obstacle based on its style.
- **get_position_y**: returns the y position of the obstacle based on its style and position. **get_position_x**: returns the x position of the obstacle based on its style and position.

IV. Tutorial / Warm-Up Session

The tutorial/warm-up session aspect of the Python script is achieved by creating variables and points that track the state of the experience and the tutorial.

```
tutorial_completed = False

tutorial_hands_joined = False

tutorial_point = "not started"



positions_moved = []



obstacles_avoided = 0
```

This is because the player is prompted to join their hands at three points in the experience - initially, to start the tutorial, secondly, to start the game, upon completing the tutorial, and finally, if they choose to restart the game, after losing. So, these variables help keep track.

The first screen that is shown to the user informs them the need to join their hands to start the experience.

```
if tutorial_hands_joined == False:

    welcome = font.render("welcome...", True, WHITE)

    text = font.render("join your hands to start", True, WHITE)

    screen.blit(welcome, (WIDTH / 2 - welcome.get_rect().width / 2, HEIGHT / 4))

    screen.blit(text, (WIDTH / 2 - text.get_rect().width / 2, HEIGHT / 2))

    player.draw()
```

If the code detects landmarks on the user, it checks the following.

```
if checkHandsJoined(frame, results) == 'Hands Joined':



        #if hands joined for the first time or after reset, set tutorial hands completed to actually start tutorial and move from welcome screen

        if tutorial_hands_joined == False:

            tutorial_hands_joined = True

            #change tutorial point to first point, the motion detection highlight

            tutorial_point = "motion detection highlight"
```

As mentioned, it changes the tutorial state and changes the tutorial point to the first part of the tutorial - the motion detection highlight. After this, the code is structured consistently in that, an if statement is set for each point in the tutorial, and different conditions are within it. After meeting the conditions specific to that tutorial point, the variable is changed and the player and the code move to the next stage or point in the tutorial.

```
# All the different points and screens of the tutorial, past welcome screen
if tutorial_completed == False:
        if tutorial_point=="motion detection highlight":
            #encourage user to move left and right and move arms around
            tutorial_text = font.render("highlight this experience is based on motion detection ", True, WHITE)
            screen.blit(tutorial_text, (WIDTH / 2 - tutorial_text.get_rect().width / 2, HEIGHT / 4))

            #check that they moved around screen
            if player.current_position not in positions_moved:
                positions_moved.append(player.current_position)

            #once done, change tutorial point
            if len(positions_moved) ==3:
                tutorial_point = "obstacles highlight"

        if tutorial_point == "obstacles highlight":
            tutorial_text = font.render("showing obstacles coming towards them", True, GREEN)
            screen.blit(tutorial_text, (WIDTH / 2 - tutorial_text.get_rect().width / 2, HEIGHT / 4))
            for obstacle in obstacles:
                if player.get_rect().colliderect(obstacle.get_rect()):
                    # the player has collided with an obstacle, so notify them
                    obstacles.empty()

                if obstacle.y > HEIGHT-30:
                    if not player.get_rect().colliderect(obstacle.get_rect()):
                        # print(obstacles_avoided)
                        obstacles_avoided+=1

                if obstacles_avoided == 3:
                    tutorial_completed = True
                    #move to next point in tutorial

            # add obstacles
            current_time = pygame.time.get_ticks()
            # multiply last spawn time by 1.5 to slow the obstacles down for the tutorial
            if current_time - last_spawn_time*1.5 > spawn_time:
                last_spawn_time = current_time
                position = random.choice(["center","left","right"])
                style = "stand"
                obstacle = Obstacle(position,style)
                obstacles.add(obstacle)
                print("adding obstacles")

            # remove obstacles that have gone off the bottom of the screen
            for obstacle in obstacles:
                if obstacle.y > HEIGHT:
                    obstacles.remove(obstacle)

        for obstacle in obstacles:
            pygame.draw.rect(screen, obstacle.color, [obstacle.x, obstacle.y, obstacle.width, obstacle.height])

        if tutorial_point == "movement for obstacles highlight":
            print(0)
        if tutorial_point == "different obstacle types highilght":
            print("tutorial completed, go to start screen")
```

Alterations and developments will be made to the code upon completion of the tutorial mechanisms. It remains true that, upon completion of the final point in the tutorial, the variable will be set to true.

When the tutorial is completed, the player will once again be prompted to join their hands and, in doing so, start the actual game.

```
if tutorial_completed == True:

    if game_started == False:

        game_join = font.render("join your hands to start", True, GREEN)

        screen.blit(game_join, (WIDTH / 2 - game_join.get_rect().width / 2, HEIGHT / 4))
```

```
        if results.pose_landmarks:

            if checkHandsJoined(frame,results) == "Hands Joined":

                game_started = True

    player.draw()
```

V. Game loop

```
while True:
    # Read frame from camera
    ret, frame = cap.read()
    frame = cv2.flip(frame, 1)

    # Convert frame to RGB
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    success, video_image = video.read()
    if success:
        video_surf = pygame.image.frombuffer(video_image.tobytes(), video_image.shape[1::-1], "BGR")
    screen.blit(video_surf, (0, 0))
    # Detect pose landmarks
    with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
        results = pose.process(rgb)
        landmarks = results.pose_landmarks
        player.update(landmarks)
        obstacles.update()

        # Draw the character through a combination of lines and shapes
        if player.landmarks:
            x1, y1 = player.landmarks[0]
            x2, y2 = player.landmarks[8]
            r = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
            color = (255, 0, 0)
            line_width = 15

            pygame.draw.polygon(screen, color, [player.landmarks[11],player.landmarks[12],player.landmarks[24],player.landmarks[23]])
            pygame.draw.circle(screen, color, player.landmarks[0], r * 1.3, width=0)
            pygame.draw.line(screen, color, player.landmarks[12], player.landmarks[14], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[14], player.landmarks[16], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[16], player.landmarks[22], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[16], player.landmarks[18], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[16], player.landmarks[20], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[18], player.landmarks[20], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[12], player.landmarks[24], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[24], player.landmarks[26], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[26], player.landmarks[28], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[28], player.landmarks[30], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[28], player.landmarks[32], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[30], player.landmarks[32], width=line_width)

            pygame.draw.line(screen, color, player.landmarks[11], player.landmarks[13], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[13], player.landmarks[15], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[15], player.landmarks[21], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[15], player.landmarks[17], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[15], player.landmarks[19], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[17], player.landmarks[19], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[11], player.landmarks[23], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[23], player.landmarks[25], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[25], player.landmarks[27], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[27], player.landmarks[29], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[27], player.landmarks[31], width=line_width)
            pygame.draw.line(screen, color, player.landmarks[29], player.landmarks[31], width=line_width)
```

```
if game_started:
    if len(obstacles) == 0:
        addObstacle()
    for obstacle in obstacles: # this loop is used to store the obstacle in a variable
        pass
    # seting up the colison zone
    y1 = y2 = obstacle.get_colision_line(nose_y,ankle_y)
    y3 = y4 = y1 + 30
    d1 = (y1 - HEIGHT + Obstacle.distance) + (Obstacle.side - Obstacle.distance)
    d2 = (y3 - HEIGHT + Obstacle.distance) + (Obstacle.side - Obstacle.distance)
    w1 = (d1 * Obstacle.b_line) / Obstacle.side
    w2 = (d2 * Obstacle.b_line) / Obstacle.side
    w1 *= 2
    w2 *= 2
    x1 = WIDTH / 2 - w1 / 2
    x2 = WIDTH / 2 + w1 / 2
    x3 = WIDTH / 2 - w2 / 2
    x4 = WIDTH / 2 + w2 / 2
    vertices = [(x1, y1), (x2, y2), (x4, y4), (x3, y3)]
    # colision handling
    if player.landmarks:
        if obstacle.y > y1 and obstacle.y < y3:
            for x, y in player.landmarks:
                temp_rect = pygame.Rect(x, y, 1, 1)
                if temp_rect.colliderect(obstacle.get_rect()):
                        # the player has collided with an obstacle, so lose a life
                        if player.lives !=0:
                            player.lives -= 1
                        elif player.lives == 0:
                            # player has no lives left
                            # Create randomly generated username for player
                            # generating random strings
                            if username ==None:
                                username = ''.join(random.choices(string.ascii_uppercase, k=5))
                                user_score = [username,int(score)]
                                #!! in the end, this needs to be modified for the leaderboard to store the top 10
                                with open('leaderboard.csv', 'a+', newline='\n') as write_obj:
                                    # Create a writer object from csv module
                                    csv_writer = writer(write_obj)
                                    # Add contents of list as last row in the csv file
                                    csv_writer.writerow(user_score)
                            #end the game
                            # to-do
                        obstacles.empty()
                        break

    # remove obstacles that have gone off the bottom of the screen and add new obstacles
    for obstacle in obstacles:
        if obstacle.y > HEIGHT:
            obstacles.remove(obstacle)
            addObstacle()
            score += 5

    score += 0.1
```

```
      # drawing the collision zone
       pygame.draw.polygon(screen, BLUE, vertices)

       # drawing the obstacles
       for obstacle in obstacles:
           pygame.draw.rect(screen, obstacle.color, [obstacle.x, obstacle.y, obstacle.width, obstacle.height])

   else:
       if landmarks:
           game_started = checkHandsJoined(landmarks)
           nose_y, ankle_y = player.landmarks[0][1], player.landmarks[28][1]
       text = font.render("join your hands to start", True, WHITE)
       screen.blit(text, (WIDTH / 2 - text.get_rect().width / 2, HEIGHT / 4))


   screen.blit(font.render(f"score: {int(score)}", True, WHITE), (10, 10))
   screen.blit(font.render(f"lives: {player.lives}", True, WHITE), (WIDTH - 150, 10))

   # Display frame in Pygame window
   pygame.display.update()
   clock.tick(160)

   # Handle Pygame events
   for event in pygame.event.get():
       if event.type == pygame.QUIT:
           cap.release()
           cv2.destroyAllWindows()
           pygame.quit()
           exit(0)
```

The main loop of the program is initiated with a while statement that continues to run indefinitely until manually stopped. Within this loop, the program first reads a frame from a camera connected to the computer, flips the image horizontally, and then converts the frame from the default BGR format to RGB format.

Next, the program attempts to read a video stream and if successful, converts the stream to an image that can be displayed on the screen using the Pygame library. The video image is then displayed on the screen.

The next section of the program uses the MediaPipe Pose library to detect pose landmarks from the converted RGB image. These landmarks are stored in a landmarks variable. The program then updates the position of the player's character based on the landmarks detected. The character is drawn on the screen using Pygame drawing functions, which are called in a specific order to create a polygonal shape that represents the character's body.

After drawing the character on the screen, the program checks if the game has started and if there are any obstacles present in the game. If there are no obstacles, a new obstacle is added to the game. The program then creates a collision zone around the obstacle, with four vertices that are defined based on the position of the obstacle and the dimensions of the collision zone. Finally, the program handles collisions between the player's character and any obstacles present in the game.

VI. Game restarting

The restart functionality is embedded within the main game loop and starts with declaring a number variable called countdown. This variable alongside the timer serve as how long the countdown is.

```
countdown, countdown_seconds = 15, '15'

pygame.time.set_timer(pygame.USEREVENT, 1500)
```

The idea is that, when a player loses all their lives, and subsequently the game, the screen will show two things 1) a countdown timer which allows the player to restart the game if they join their hands in time and 2) the leaderboard with the username and scores which we discuss below.

Restarting the game means that the player retains their username and does not have to undergo the tutorial/warm-up session again. The countdown works as follows:

```
for event in pygame.event.get():

    # countdown for restarting
```

```
    if player.lives == 0:
        if event.type == pygame.USEREVENT:
            countdown -= 1
            if countdown > 0:
                countdown_seconds = str(countdown)
```

In the code above, in every instance, the if statement checks if the player has lost all their lives and if the event is part of the timer. If both of those statements are true, the countdown gets reduced and the countdown seconds, its string equivalent, are reduced by one, as long as the countdown is not zero.

Later on in the code, the countdown is presented on the screen when the player loses all their lives through the below mechanism.

```
  if player.lives == 0:
    #player lost, show leaderboard
    obstacles.empty()
    #restart option - 15 seconds
    if countdown > 0:
        game_over = font.render(f"game over.. join hands in {countdown_seconds} seconds to restart", True, GREEN)
        screen.blit(game_over, (WIDTH / 2 - game_over.get_rect().width / 2, HEIGHT / 4))
```

Here, as the text prompts them to join their hands to restart, we check if they did so, before the countdown reaches zero.

```
#Player chooses to restart experience
        if checkHandsJoined(frame,results) == "Hands Joined":
            player.lives+=3
            score = 0
```

```
#Player has not restarted, reset experience
    if countdown == 0:
        # if checkHandsJoined(frame,results) == "Hands Not Joined":
            #reset the players attributes
            player.username =  ''.join(random.choices(string.ascii_uppercase, k=5))
            player.lives +=3
            score = 0

            #reset the tutorial attributes
            tutorial_completed = False
            tutorial_hands_joined = False
            tutorial_point = "not started"
            game_started = False
            reset = True
            positions_moved = []
            obstacles_avoided = 0

            #reset countdown
            countdown, countdown_seconds = 15, '15'
            pygame.time.set_timer(pygame.USEREVENT, 1500)
```

As shown above and in the comments, if the player does not join their hands and the countdown reaches 0, the experience resets and all the relevant variables related to the restart, tutorial, and game mechanisms are reset to their initial state.

VII. Data Storage & Leaderboard

Data generated from the experience is stored in a CSV file in the directory that is constantly updated. This is the leaderboard of the top 10 scores and usernames that are presented at the end of each session where a player loses all their lives. The username is stored as a variable both in the Player class and outside it to account for cases where a user restarts the experience, allowing them to retain the same username they had the first time. This is because usernames are randomly generated.

```
if game_started & tutorial_completed:
    if player.lives !=0:
        screen.blit(font.render(f"score: {int(score)}", True, WHITE), (10, 10))
        screen.blit(font.render(f"lives: {player.lives}", True, WHITE), (WIDTH - 150, 10))
    player.draw()
    for obstacle in obstacles:
        if player.get_rect().colliderect(obstacle.get_rect()):
            # the player has collided with an obstacle, so lose a life
            if player.lives !=0:
                player.lives -= 1
            if player.lives == 0:
                # player has no lives left
                # Create randomly generated username for player
                # generating random strings
                username = player.username
                user_score = [username,int(score)]

                with open('leaderboard.csv', 'a+', newline='\n') as write_obj:
                    # Create a writer object from csv module
                    csv_writer = writer(write_obj)
                    # Add contents of list as last row in the csv file
                    csv_writer.writerow(user_score)
                # Add the new row to the dataframe
                data.loc[len(data)] = user_score
                #end the game
                break
            obstacles.empty()
```

The code above checks if and when the players lives reach 0, it stores their username and score. The randomly generated string for the username is created within the Player class. Next, using the writer module, we add a new line to the leaderboard CSV file with the users name and score.

As this is a leaderboard, the csv file should only contain the top 10 rows. So, the rest of the leaderboard updating mechanism is conducted when the player reaches 0 lives and in the same case where the countdown screen is presented, as mentioned above.

```
if countdown > 0:
    game_over = font.render(f"game over.. join hands in {countdown_seconds} seconds to restart", True, GREEN)
    screen.blit(game_over, (WIDTH / 2 - game_over.get_rect().width / 2, HEIGHT / 4))
        #order dataset based on highest scores
    data = data.sort_values(by=['Score'], ascending=False)
    # get the top 10 scores
    data = data.head(10)
    usernames = data.loc[:,"Username"]
    #offset so the rows show one after the other
    offset=0
    for row in usernames:
        score = data.loc[data['Username'] == row]['Score'].values[0]
        #Highlight the users score if they make it on the leaderboard
        if row == username:
            user_row = font.render(f"{row.lower()}.................{str(score)}", True, GREEN)
            screen.blit(user_row, (WIDTH / 2 - user_row.get_rect().width / 2, (HEIGHT / 3)+offset))
        else:
            leader_row = font.render(f"{row.lower()}.................{str(score)}", True, WHITE)
            screen.blit(leader_row, (WIDTH / 2 - leader_row.get_rect().width / 2, (HEIGHT / 3)+offset))
        offset+=30
    #save the leaderboard to the updated csv so there will always be only 10 rows stored
    csv_save = data
    csv_save.to_csv("leaderboard.csv", index=False)
```

Here, we first sort the csv file by the highest scores, and limit it to the top 10 scores by getting the head rows. Additionally, to show the leaderboard appropriately, we instantiate an offset for the presentation and store all the usernames to loop through.

In the loop, we check if the username we are on is the same as the user's username. This means that the user reached a high enough score to make it to the leaderboard, To distinguish this from the others, we present the text in green rather than white.

After the updating and presentation of the leaderboard, we save the CSV file as data - the variable that contains the sorted top 10 scores and usernames of all time. This ensures that we will always only have 10 usernames and scores in the CSV file.

Testing

All the testing for the code was constantly conducted manually throughout the development of the code both by the developers and through usability testing with various groups of users and scenarios.

Bugs

Currently, there are no bugs in the code that we are aware of but will update this section as we move forward with completing the project, conducting further testing, and finalizing the code.

Author Information

Ahmed Issaoui

E-mail: aissaoui@andrew.cmu.edu

Github: https://github.com/aissaoui-cmu

Mariam Al Thani

E-mail: mathani@andrew.cmu.edu

Github: https://github.com/mariamalth

Jewel Dsouza

E-mail: jdsouza@andrew.cmu.edu

Github: https://github.com/jeweldsouza

Raghad Abunabaa

E-mail: rmabunab@andrew.cmu.edu

Github: https://github.com/raghadabunabaa