GNG1103 - Client Feedback and First Prototype for a VR Climate Change Simulation

Mohammad Odeh, Malcolm Ross, Krithivaas Vyas, James Bugeaud

March 2, 2025

Abstract

This document describes the objectives and results from testing the first prototype for a VR climate change simulation and sets out the changes and plans made for the second prototype. Client feedback on the design concepts considerations from initial planning are documented and used to guide development of systems critical for early testing. Analysis of these systems is conducted and observation of the prototypes found most are functional as proof-of-concept models. The bill of materials has been updated to reflect the budget constraints, while further additions and revisions have been made to the design criteria and test plans for the next prototype.

Table of Contents

1 Introduction

Following the budgeting and planning stage and feedback from client, development has moved forward onto making the first prototype of the simulation, guided by new understandings from these earlier phases. Using the prototyping and testing plan from this previous stage, the team took on roles to create proof-of-concepts for the unique characteristic elements of the simulation namely the dialogue and environmental scanning systems, as well the necessary surrounding systems, such as terrain design, sound and lighting, or user control. Working these together into a rough vertical slice of the simulation, the prototype was shown to prospective users to gather further feedback on the direction of the design. Taking this into consideration alongside the early testing results, revisions were made to the previous design specifications to adjust priorities and understand new criteria, as well as the bill of materials to account for additional assets to facilitate development.

2 Client Feedback on Design Concepts

From the design concept ideation stage, three potential stories were formed for the simulation: in the first, the user embodies a resident of Lytton, B.C., gathering what they can and evacuating from the 2021 wildfires that destroyed the town; in the second, the user takes the role of a firefighter rescuing a civilian from a forest fire; and in the third, the user is a fire inspector working with a non-player companion to scan the state of a forest and gauge the threat of an ongoing wildfire.

One of the main takeaways from the client meeting which altered the course of the project was the renewed emphasis on educational value and its priority over interactivity and personalization, the latter two of which had previously been considered the focal points of the design. While the design criteria before this had informed the conclusion that the firefighter simulation would be the best concept, the new information and opinions from the client indicated that the fire inspector story had greater potential to meet their needs. This design concept relies on the coupled subsystems of an environmental scanner tool and companion dialogue; the first provides a diegetic means of gathering and viewing data on climate change, as well as serving as a context-appropriate trigger for dialogue, while the second provides a believable way to communicate the personal consequences of climate change. A limitation of this concept suggested by the client was that it provides limited means of interactivity, insofar as the user's control is limited primarily to the pace and order of content while the ability to affect the state or outcome of the story is limited. Plans are made to augment the currently planned subsystems to accommodate for this, with the addition of further subsystems also being a potential consideration.

3 First Prototype of the Simulation

The first prototype was focused mainly on familiarizing group members with the basics of the Unity engine and developing proof-of-concept builds for the key features of the simulation and their particular area of content development.

3.1 Analysis of Critical Components

The scanner tool serves as the primary means of user interaction in the simulation and may be broken down into two components: the ray detector and the virtual screen. The ray fires straight forward from the center of the device and is locked to the user's virtual hand. The ray will track the nearest object it hits in any given frame and can use that to store a reference or pull data directly from the object or an attached script. The object tag system built into Unity provides a simple means of distinguishing what can and cannot be scanned, which should make it simple to avoid bugs from trying to call certain functions on ineligible objects. The ergonomics of the ray handling present a potential failure point if objects are too closely spaced or too small to get a good aim with, though the detection precision has proven good in early tests and could be later improved with upscaled collision meshes on scannable objects.

The scanner's UI is operated solely with controller inputs and the scanning ray, which avoids the issue of forcing the user to stabilize the scanner with one hand while trying to interact with it using the other. The main concern with the UI is how to tell it how to update after the scanner has tagged a new object. One element of making this simpler to implement and more user friendly is to develop a one consistent UI layout for the text, tabs, and images, with potentially a second should there come another sufficiently distinct category of content to display, which minimizes the number of extraneous or moving components to keep track of. To get the content for this UI, there are two potential avenues that have been identified. The first is to store all text and image references to be used inside the script for UI, which centralizes the everything so that there are no issues with having to get data from other scripts, but this would likely bloat the script and make it harder to debug, as well as requiring more public variables than desirable for each and every image reference. The alternative is to attach each scannable object with a container script that holds a standardized set of information and has getter functions that the scanner calls on scanning the object. While this raises the risk of an element being forgotten as the scripts are scattered across the objects and must be filled in after programming the generic container, it preserves the simplicity of the scanning script.

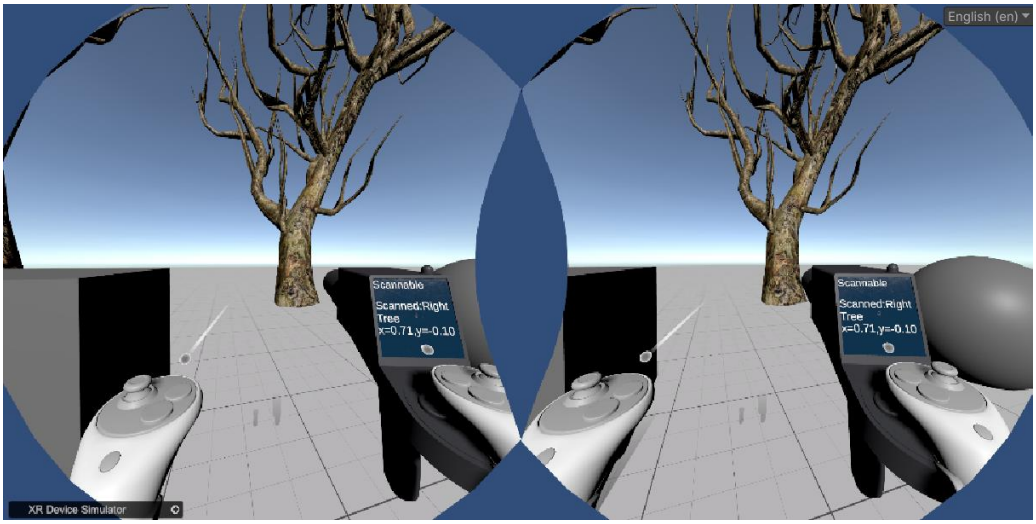
The construction of the setting involves all of the lighting, sound, and environmental design and so may be considered as a singular aggregate system. The purpose of this system is twofold; first, the environment itself and its evolution tell a story that the user subconsciously engages with alongside the more interactive elements to fully and continuously immerse them in the narrative. This is particularly important to this project as the world itself is subject to climate change and thus ought to be represented as changing with it. Secondly, the setting defines and communicates the bounds of what the user can and cannot do, as well as provide feedback on their actions to make them feel like a part of the virtual world and abide by fundamental UX principles. The bounds and collisions are key to making the space easy to engage with the interaction tools. The sound and lighting elements have been separated from the environmental design for role assignment as the former involves a fair amount of editing outside of Unity and can be handily tested with point sources in a small, isolated scene, while

the latter requires more large-scale planning and direct consideration for how the user will be able to navigate the scene.

3.2 Objectives and Results of the Prototyping Test Plan

Note carefully that this section references prototype tests in the format of Test #, which refers to the respective test ID in the appendix (section 11). Full details on the tests are not featured in the document body in the interest of brevity.

Test 1 from the original prototyping plan was to build a simple scanner that could accurately cast a ray to an object tagged as scannable and then use that to change the scanner's display to show the corresponding data. For this model, the display was made as just a canvas with three text elements: the first updates every frame to state whether the device is pointed at a scannable object, the second changes on click to display the name of the tagged object if it was scannable as a basic data-pulling proof, and the third updated every frame to show the device's direction, in case that becomes useful. Each of these was successful in differentiating scannable and non-scannable elements for each of the 3D meshes built into Unity. It remains to be tested, however, how this will pull more complex data and images from scripts.



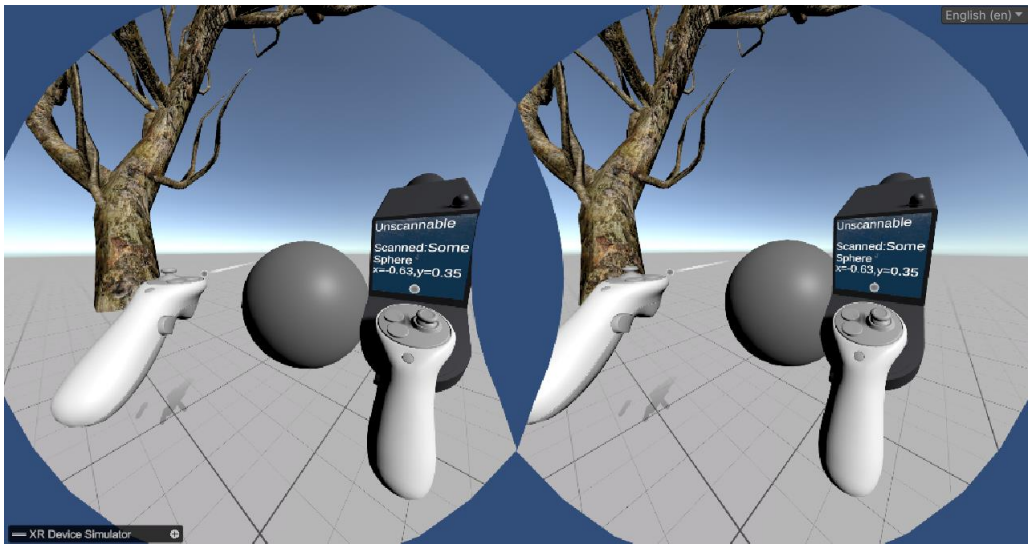


Figure 1. Scanner picking out scannable and non-scannable targets.

Test 9 was the implementation of localization support, which was successfully done with the localization package. Each translatable text object is attached with a localization string event which holds a key to the global localization table. This key allows it to automatically set the text to match the global language setting, which remains as set between scenes and keeps everything centralized and easier to track. In addition to this, the ability to cycle through menus was implemented to be able to build the survey into the simulation. This used sliders as radio buttons with labels to allow simple numerical input.

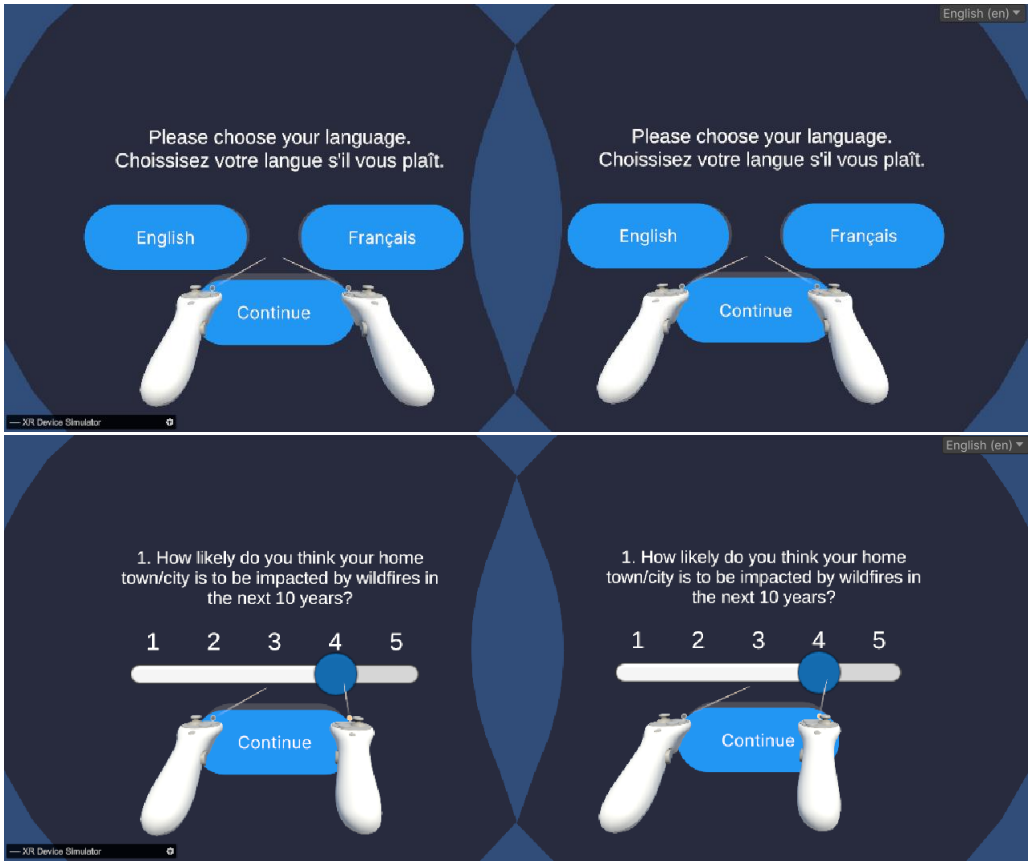


Figure 2. Localization, menu navigation, and rating system. Displayed with the occlusion mesh effect to better visualize how it would appear to the user.

Test 4 was to implement multiple fire particle effects from the two free packs carried over from Deliverable E; Particle Pack (PP) and Free Fire VFX - URP (FFV, part #8). The goal was to see how well they visually look, along with seeing how the light and shadows they create would interact with the environment. The fires for PP looked very simple but also had lots of movement that made it look quite distracting. FFV had a lot more to offer, as all the fires come with sound built into them, and an option that adds smoke to each individual fire if needed. As there needs to be a consistent style throughout the simulation, PP was dropped from the Bill of Materials. A flickering effect was added to the light using a script and that really helps to make the fire look more immersive. The light effect is made using a point light source, which is a feature built into Unity without need of additional assets.

Figure 3. Fire from FFV to show off how the light interacts with the surrounding environment

The objective of test 8 was to listen to the various sound effects of Casual Game SFX Pack (CG SFX) and Minimal UI Sounds (MUS), along with the song Sad Hope (part #4). This was done to see how well they all fit within the tone of the simulation and to gauge their quality. MUS's sounds seemed to be non-functional or at best low quality. On top of that, MUS only had around 8 sounds, and none of them were good enough to warrant keeping the pack. CG SFX had many more sounds to offer, with most of them being of high production and sound quality. Sad Hope sounds good, though where it should be implemented remains to be determined.

Figure 4. The list of sound files in the Casual Game SFX Pack.

Tests 3 and 4 were focused on importing and integrating environmental assets, including terrain, trees, ground textures, fire, and lighting. The lighting was adjusted to enhance depth and realism, ensuring fire sources interacted naturally with the scene. The ground textures

were evaluated for consistency with the dry-forest setting, with minor refinements made where necessary. Fog and rain assets were also added to the environment to have a more immersive feeling.

Character scaling was not possible at this stage as the issue of Unity file sharing remains in the process of being resolved. Future tests will address this issue to ensure proper proportion and interaction within the environment.

Figure 5. Bird's-eye view of the burnt forest scene showing fire effects and particle sources

Figure 6. Ground level view of the forest scene, closely modelling the intended environment

Test 6 is based on choosing, designing, and improving character design to suit the image visualized by the group of the fire inspector. The image in Figure 7 is the main character that will be used for the fire inspector scenario. The design is suitable to the role and can be perceived as an expert of the field due to the safety equipment displayed in the design. The design also avoids stereotyping due to the face being covered. Throughout the design process, there were problems in many of the tests, the first being the disappearing eye color that often occurred. This was fixed within the first test by switching the eye terrain to another terrain. After multiple tests, there seemed to be no issues with the character, however, this design would later be very difficult to use in Test 7, which was the reason it had to be switched with a placeholder character (Figure 8) in the assembly of the first prototype.

Figure 7. Fire Inspector Character.

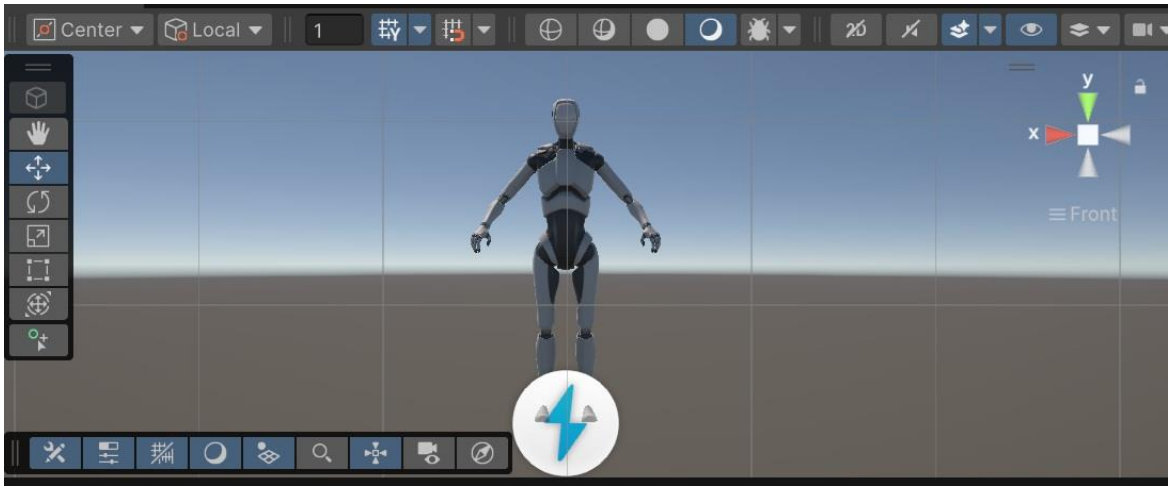


Figure 8. Placeholder game character originating from starter Pack.

Test 7 focused on testing the character movement. The first test implemented a simple script for character movement, which successfully enabled the character to move in four directions.

The second test was after improving the code to implement more movement, this test was also successful in being able to implement jumping motion and sprinting. To complement Test 3, hand and leg animations were planned to be implemented, however, after editing the script, the character started sinking through the ground despite the character and terrain having active colliders. For Test 4, I had edited the animator and my code, however, I had a problem that my character would float horizontally, the hand and foot animations were being achieved but the character was not upright and it looks like swimming motion rather than walking, Hence, this was a failed test. A character movement asset was imported from the Unity asset store to continue Test 7, and while it was easy to use and implement, the character would randomly float or sink into the ground despite all scripts and colliders appearing to be in order. Movement for the firefighter was not successfully implemented, so a placeholder model has been used instead for the current prototype, with plans to fix the rigging issues in the next prototype.

Figure 9. Cinemation and character movement implemented, displays jumping motion and hand-movement.

Figure 10. Character moving on higher terrain.

4 Feedback on the Prototype

Both the client and potential users were consulted for feedback on the current state and direction of the prototype. For the UI, the discrepancy in visual fidelity between the headset display and desktop screen raised an issue with the readability of text, especially for HUD elements which the user cannot enlarge or focus on. This was troubling especially for users with visual impairment as general use VR headsets do not physically accommodate the use of glasses. To better account for the limitations of the headset when one is not immediately available, the play test in the Unity editor may be set to show in ocular occlusion mode as the "live feed" of what gets displayed on a headset.

5 Updated Target Specifications

With new understandings of the limitations and abilities of the Unity engine, as well the constructive feedback provided by the client, some of the original design specifications are revised to better reflect what can and should be done:

- The simulation time was at first given as a constraint of 1 to 3 minutes, which has now been revised to a target duration of 1.5 to 2 minutes for a realistic window that retains some safety margin. The first scene should last for 45 to 60 seconds, followed potentially by a 10 to 15 second transition cutscene, then another interactive segment lasting approximately 30 seconds. This intentionally does not factor time spent in the

pre- or post-simulation survey into the total, as the user should have as much time as they wish to think over their answers.

- The financial budget to be eligible for reimbursement has been altered to $25 to reflect new constraints, down from the previously considered $50. Some assets have been purchased out-of-pocket where free counterparts do not suffice, but the cost of these has been appropriately tabulated in the bill of materials (Section 4).
- After testing the simulation with a headset, the accessibility criteria are broadly worth updating. Setting a low base move speed largely removes the concerns with motion sickness, especially when the avatar is not frequently moving, and the VR template in Unity features both joystick and teleportation movement built in, meaning there is no compromise in development resources to include both.
- One previously unanticipated concern was the lack of support for visual impairment. The headset displays with fairly low resolution and cannot be used while wearing glasses, meaning text objects must scale or be physically approachable to remain legible for all users.

The other design specifications from deliverables C and D are largely qualitative and have been chosen to remain as they are.

6 Updated Bill of Materials

The original bill of materials proposed a list of assets that would have cost $65. Following notice that the project budget could be no greater than $25, adjustments are made to which assets are planned for use. Further, additional plugins required for some of the desired functions have been included. The MockHMD plugin is only used for prototyping and will not be enabled in the end product.

Table 2. Updated bill of materials

Part #	Count	Cost	Item Name	Link
1	4	$0	Unity 6 Game Engine	https://unity.com/download
2	1	$452	Meta Quest 3S VR Headset	https://www.meta.com/ca/quest/quest-3s/
3	2	$50	Meta Quest 2 Remote Controller	https://www.meta.com/ca/quest/accessories/quest-2-controllers-refurbished/
4	4	$0	Music - Sad Hope	https://assetstore.unity.com/packages/audio/music/music-sad-hope-157746
5	4	$0	Dry Tree Asset	https://assetstore.unity.com/packages/3d/vegetation/trees/dry-trees-86967
6	4	$0	FREE Casual Game SFX Pack	https://assetstore.unity.com/packages/audio/sound-fx/free-casual-game-sfx-pack-54116

7	4	$0	Survivalist character	https://assetstore.unity.com/packages/3d/characters/survivalist-character-181470
8	4	$0	Free Fire VFX - URP	https://assetstore.unity.com/packages/vfx/particles/fire-explosions/free-fire-vfx-urp-266226
9	4	$0	Animated Sun Skybox	https://assetstore.unity.com/packages/2d/textures-materials/sky/animated-sun-skybox-98447
10	4	$0	Haptic Feedback	https://assetstore.unity.com/packages/tools/integration/haptic-feedback-308330
11	4	$0	UK Terraced Houses Pack FREE	https://assetstore.unity.com/packages/3d/environments/urban/uk-terraced-houses-pack-free-63481
12	4	$0	Town Houses Pack	https://assetstore.unity.com/packages/3d/environments/urban/town-houses-pack-42717
13	4	$0	Dry Trees	https://assetstore.unity.com/packages/3d/vegetation/trees/dry-trees-86967
14	4	$0	Localization Plugin 1.5.4	(Built in to Unity, requires install from package manager in the engine) About Localization | Localization | 1.5.4
15	4	$0	MockHMD XR Plugin 1.4.0	(Built in to Unity, requires install from package manager in the engine) About the Mock HMD XR Plugin | MockHMD XR Plugin | 1.4.0-preview.2
16	4	$0	OpenXR Plugin 1.14.0	(Built in to Unity, requires install from package manager in the engine) OpenXR Plugin | OpenXR Plugin | 1.14.1
17	4	$0	Starter Assets: Character controllers|URP	https://assetstore.unity.com/packages/essentials/starter-assets-character-controllers-urp-267961
18	1	$5	Floating Dust & Falling Ash	https://assetstore.unity.com/packages/vfx/particles/environment/floating-dust-falling-ash-175524
19	1	$10	Dead forest environment	https://assetstore.unity.com/packages/3d/environments/dead-forest-190976
20	1	$15	FireFighter	https://assetstore.unity.com/packages/3d/characters/humanoids/humans/firefighter-171169
21	1	$35	3D Rural Houses Set	https://assetstore.unity.com/packages/3d/rural-houses-set-95289

Total Cost - All Items	$567	-	-
Total Cost - Planned Items	$25	-	-

Of these, only part #19 and part #20 are being considered for the final budget. The other paid assets either do not need to be purchased or are being considered for out-of-pocket procurement.

7 Second Prototype Test Plan

The details of the second prototyping test plan are given in full in the appendix, section {}. Tests for elements not implemented in prototype 1 have been carried over, as well as the stress tests which must be repeated as the project is built further to catch new bugs. The roles of each member in development of the simulation are tabulated as follows:

Table 3. Content specialization of each group member

Name	Content Roles
Mohammad	Character models and animation
Malcolm	Lighting, sound, and particle effects
Krithivaas	Environmental and setting design
James	UI, dialogue, and interactable elements

8 Conclusions and Recommendations

The primary purpose of this first prototype was to learn the skills necessary to develop the simulation and debug issues anticipated to reoccur, as well as implement the core functions for user interaction. The user interface and scanning tool system were successfully built and tested with good results, though the companion system remains in development. Initial tests for audiovisual effects are promising, though further tests will be necessary as the effects are integrated into a full scene. Difficulties in manipulating the avatar rigs and collision detection have forced delays in testing and implementing the planned character models. Target specifications remain largely unchanged beyond new considerations for accessibility. With the initially considered budget curtailed, changes were made to the bill of materials to use free or less expensive assets or make purchases out-of-pocket without expectation of reimbursement.

9 Future Work

With most of the core functionalities implemented, the next prototype aims to complete the remaining core features, namely working the proof-of-concept lighting and environment designs into a well-laid-out scene, as well as developing the system by which the companion character displays dialogue and follows the player. Beyond this, there is work to be done in scaling up the existing scenes to ensure there is sufficient content to engage with and provide an environment more reflective of the final product in which to stress test the interactive features.

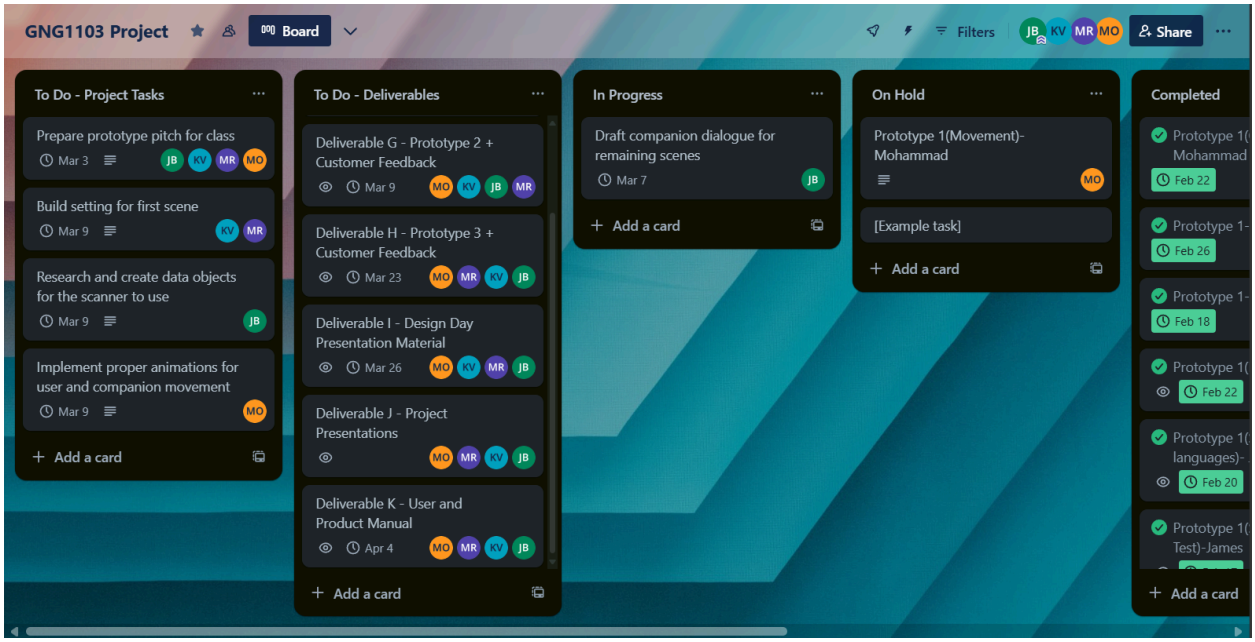
10 References

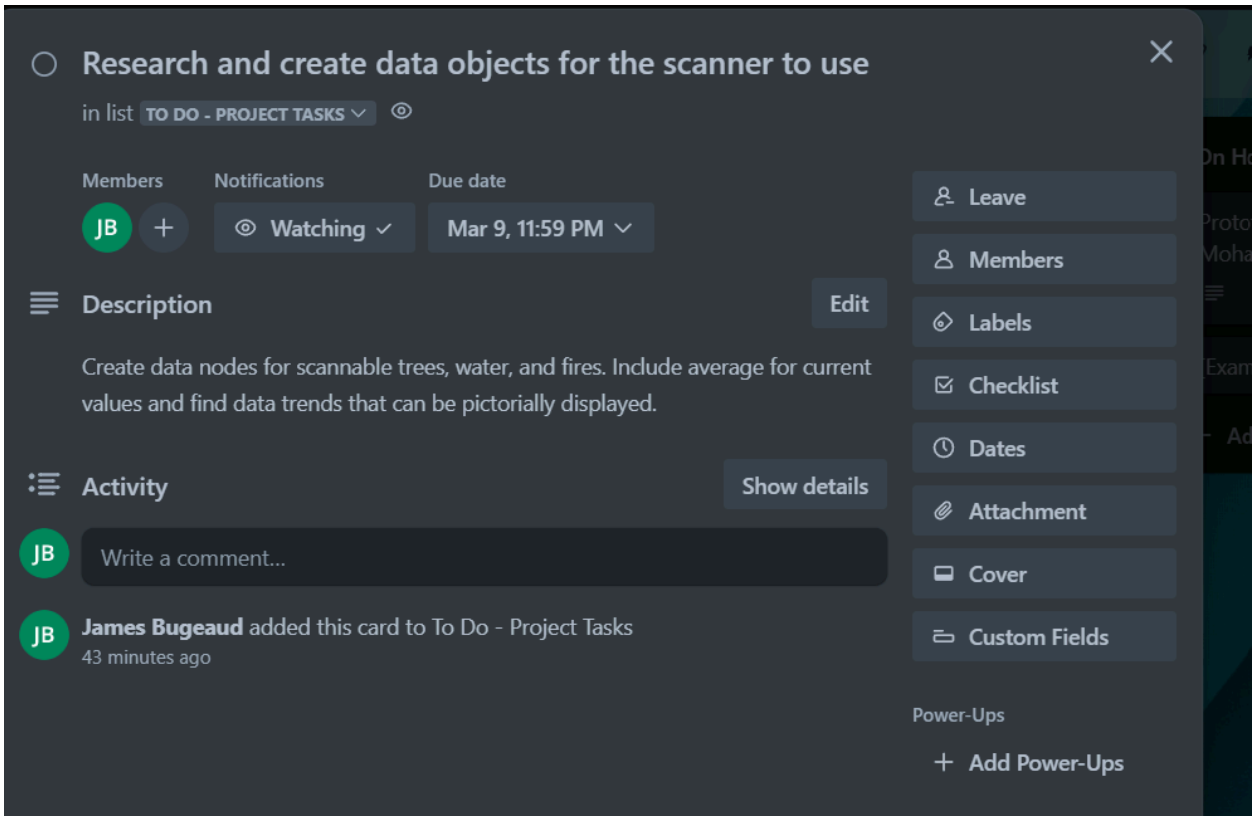
No external references were cited for this document.

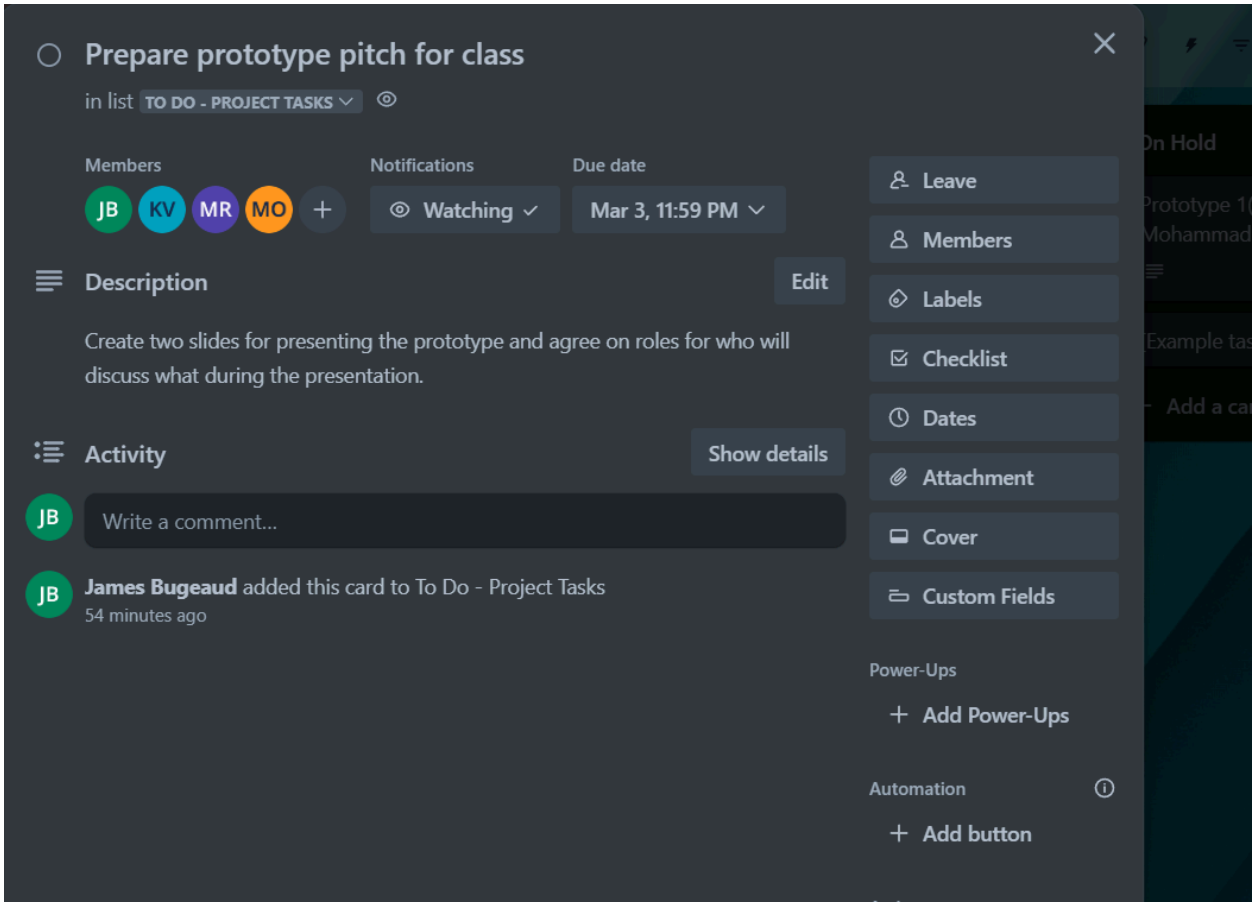
11 Appendix – Prototype 2 Testing Plan

Design Concept: "Fire Inspector" Simulation			User takes the role of a fire inspector with a companion to explore how wildifres can be caused by and propagate climate change								
Test Number	**Probable Critical Issue**	**Test Objective**	**Test Description**	**Analysis Method**	**Determine Measurables**	**Metrics**	**Level and Fidelity of Prototype**	**Kind of Prototype**	**Results**	**Interpretation and Feedback**	**Notes**
	What assumption are you testing?	*Communication, Performance Measurement, Risk Management, or Learning and Understanding*	*What specifically will you test? What is your hypothesis?*	*Specifics of how the test will be run, what it uses, pass/fail criteria, and how data is collected*	*What are you testing with the concept (focus on measurable attributes)*	*What metrics will you test and what are the associated units?*	*High or low fidelity? Focused or comprehensive?*	*Visual, Physical, or Analytical*	*Observe and record results*	*Pass or fail, why, and other feedback collected*	
1	Scanner correctly updates the display when targeting objects of interest	Performance Measurement (of scanner)	Collision test: Update text UI when a line intersects with different simple geometries	Make a crude scanner model and scene with two copies of each simple geometry mesh: one interacts with the scanner, one doesn't. For each object, tag it with the scanner while it is on the default text and try again when it has the reading from a different item. Estimated 30 min. to build scene, 10 min. to test all items.	Collision interaction with all simple geometries	Reading text on UI, Yes/No for correctness	LoFi Focused	Physical	Correctly gets and displays data attached to scanned objects, does not raise errors on tagging non-scannable objects.	Pass as each object scanned as expected	Consider using a player-facing PNG and for the light and glow texture overlay on the surface to simulate the indicator bulb instead of actual dynamic lighting.
2	Scanner plays correct sound effect when scanning and navigating UI	Performance Measurement (of scanner)	Sound test: Scan objects from Test 1 and listen for sound effects. Listen for correct sound effect on each of 4 menu navigation buttons	Update test 1 to include sound in the interaction functions. Repeat Test 1, marking which interactions do and don't play sound and any new fail conditions. Estimated 10 min. to test all items.	Presence and volume of sound effects from scanner	Yes/No for presence, test group feedback on volume	LoFi Focused	Physical	Plays the correct sound when the object is scannable, but plays no sound when not scannable. Requires debugging to figure out why.	Failed as the sound does not play consistently	
3	Simulation environment is realisitcally proportioned	Communication (of setting)	Visual test: Establish a frame of reference for further setpieces relative to avatar height and camera position	Place cars, houses, and tree assets in a scene along with user avatar that carries the camera. Scale objects to look appropriate to the avatar. Run the simulation and view objects from three different angles and distances. Estimated 10 min to build scene, 5 min. to test.	User approval of setpiece sizing	Yes/No for approval, collect and interpret other specific feedback	LoFi Focused	Visual			
4	Lighting and shadow effects from the fire appear believable	Communication (of setting)	Aesthetic test: Lighting from fire effects and the shadows cast around objects look realistic	Place a few fire assets in the scene from Test 3. Run the scene and observe the effects, checking whether for issues such as shadows not being cast away from the light source, or surfaces facing the source not being illuminated. Distribute to test users to gather feedback. Estimated 5 min. to build, 10 min. to test per user.	User approval of lighting effects	Yes/No for approval, collect and interpret other specific feedback	LoFi Focused	Visual			
5	Simulation runs smoothly with particle and lighting effects from the fire	Performance Measurement (of framerate)	Performance test: Check framerate does not fall to the point of looking choppy	Build a scene with several of each simple geometry, placed randomly in a loose cluster. Place instances of the chosen fire asset within this cluster 10 at a time. Run the simulation and move the avatar around the cluster for several seconds, then record the lowest stable framerate. Repeat until this falls below 30 fps. Estimated 30 min to build and render scene, 10 min. to test.	Average framerate and instances of severe frame loss	Observe and record framerate, units of frames per second (FPS)	LoFi Focused	Physical			
6	Character features suitably fit the image portrayed in the storyboard description	Communication (of characters)	Visual Test: The character is well designed with appropriate features and clothing	Build a simple character based off a real human figure, the figure will be adjusted to approproiate specifications in order for character to appear porportional comapred to the surrounding area. The character must have appropriate clothing that fits their role. Test will be visual and will be shared to the group to verify opinions and prespectives on potential improvements.	User approval of character asthetics	Yes/No for approval, collect and interpret other specific feedback	LoFi Focused	Visual			
7	Character movement is seamless, the character can move in all directions	Performance Measurement (of movement	Performance test: Check that the character has the freedom of movement in a space	Character movement setting will be set to move in all directions in the avaialble space. Run the simulation and attempt to have the character move around, record all movement pros and cons for future develpoment.	User approval of character movement	Observe and record movement precision and fludity	LoFi Focused	Physical			
8	Environment sound effects and music fit into the world	Communication (of setting)	Sound test: Interact with different objects to gauge how well they sound	Build a small scene with many little interactible objects, bushes, trees, doors, etc., and give them all the respective sound effects needed. From there, just try and play the simulation like normal and see how well they sound in fit into the game world. Add different music tracks in the background and see if they're too distracting or if they fit well. Estimated 30 min. to build, 20 min. to test.	User approval of audio quality	Yes/No for approval, get group feedback for what fits best	LoFi Focused	Visual			

9	Language selection appropriately changes in-game text	Performance Measurement (of text objects and UI)	UI test: Text in each section of the simulation matches the language chosen by the user	Build two scenes, each with a text object containing a line in either English or French. Make a HUD button that changes the language. Run the first scene and choose English, check the displayed language matches the button. Check that text on the scanner language matches or all menus. Change to next scene while running and repeat. Run all again starting in French. Estimated 2 hours to program, 10 min. to test.	Right or wrong language on different elements consistent, i.e. not reset, between scenes	Mark which elements do and do not change language during test. Mark whether language changes itself on scene change.	LoFi Focused	Analytical	All text and buttons configured to be bilingual correctly updated text as chosen, which persisted between scenes	Pass as all bilingual text updtaed appropriately.	Completed using localization package, requires special configuration for all text objects
10	Scenes load in quickly and smoothly	Performance Measurement (of scene change functions)	Performance test: All scene elements load in within 3 seconds, tested by timing how long it takes to do this on starting the simulation and changing scenes	Using scenes built for the latest prototype, run the simulation in whole. Time how long it takes to change from one scene to the next. The scene is considered "loaded" once all objects are loaded in and the framerate stabilizes. Pass if the scene takes less than three seconds to load. Estimated 10 min. total to test 5 trials.	Load time between scenes	Load time (seconds)	LoFi Comprehensive for early prototypes, moves to HiFi for later versions	Physical	Quickly loaded between test scenes, requires further stress testing as scenes are built more	Pass since scenes loaded in less than one second	
11	User is appropriately confined to the story area	Learning and Understanding (of level boundaries)	Constraint test: The user can not take themselves out of the intended areas, nor do they clip through collision objects	Using scenes built for the latest prototype, run the simulation in whole. Move the avatar all along the bounding wall and into all collision objects. Observe which, if any, the character moves through. Make note of any other unexpected physics interactions with the avatar, e.g. slingshotting, falling through the ground. Estimated 20 min. total to test 3 trials.	Presence of non-functioning collisions	Yes/No for faulty collisions of each object	LoFi Comprehensive for early prototypes, moves to HiFi for later versions	Physical			
12	Scanner is stable for all variety of user inputs	Performance Measurement (of scanner stability)	Input test: The scanner UI follows its intended controls even when subjected to unintended user input patterns	Test "bad" input patterns of using unassigned keys, spamming the same key, spamming different keys, holding keys, using an assigned key while holding down another, and simultaneous inputs, each for 10-15 seconds. Between each trial, check that all scanner menus can be navigated to. Estimated 15 min. to test all patterns in each scene.	Presence of bugs or non-functional UI navigation on the scanner	Yes/No for marking a bug after each trial	HiFi Focused	Physical	Text did not fail to update during any of the stress test patterns.	Pass as the test did not encounter bugs	
13	Companion keeps appropriate distance to and from the user during free navigation	Performance Measurement (of companion movement)	Tracking test: Companion follows the user while giving them space to act	Write a tracking script to test, which makes the companion move to keep a distance of between 1 and 5 units to the player, turning appropriately when changing direction. Simulate companion with a stretched box. Test by running around the companion, walking in and out of the following range, and teleporting out of range. Record distance companion moves to after each. Estimated 2 hours to build, 10 min. to test.	Distance at which companion stops moving	Unity position transform units	LoFi Focused	Physical			


GNG1103 Project
Board
Filters
Share
To Do - Project Tasks
Prepare prototype pitch for class
Mar 3
Build setting for first scene
Mar 9
Research and create data objects for the scanner to use
Mar 9
Implement proper animations for user and companion movement
Mar 9
Add a card
To Do - Deliverables
Deliverable G - Prototype 2 + Customer Feedback
Mar 9
Deliverable H - Prototype 3 + Customer Feedback
Mar 23
Deliverable I - Design Day Presentation Material
Mar 26
Deliverable J - Project Presentations
Deliverable K - User and Product Manual
Apr 4
Add a card
In Progress
Draft companion dialogue for remaining scenes
Mar 7
Add a card
On Hold
Prototype 1(Movement)-Mohammad
[Example task]
Add a card
Completed
Prototype 1(Mohammad
Feb 22
Prototype 1-
Feb 26
Prototype 1-
Feb 18
Prototype 1(
Feb 22
Prototype 1(languages)-
Feb 20
Prototype 1(Test)-James
Add a card


Research and create data objects for the scanner to use
in list TO DO - PROJECT TASKS
Members
Notifications
Due date
JB
Watching
Mar 9, 11:59 PM
Description
Edit
Create data nodes for scannable trees, water, and fires. Include average for current values and find data trends that can be pictorially displayed.
Activity
Show details
JB
Write a comment...
JB
James Bugeaud added this card to To Do - Project Tasks
43 minutes ago
Leave
Members
Labels
Checklist
Dates
Attachment
Cover
Custom Fields
Power-Ups
Add Power-Ups


Prepare prototype pitch for class
in list TO DO - PROJECT TASKS
Members
JB KV MR MO
Notifications
Watching
Due date
Mar 3, 11:59 PM
Description
Edit
Create two slides for presenting the prototype and agree on roles for who will discuss what during the presentation.
Activity
Show details
JB
Write a comment...
JB
James Bugeaud added this card to To Do - Project Tasks
54 minutes ago
Leave
Members
Labels
Checklist
Dates
Attachment
Cover
Custom Fields
Power-Ups
Add Power-Ups
Automation
Add button