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

Mohammad Odeh, Malcolm Ross, Krithivaas Vyas, James Bugeaud

March 23, 2025

Abstract

This document describes the objectives and results from testing the third prototype for a VR climate change simulation. User feedback on the previous and latest prototype is documented and used to guide the remaining development. File sharing and compatibility has been the most critical and frequent issue encountered, resolved later with physical file exchange. Scene transitions presented difficulty with VR compatibility but were ultimately successful. Recommendations are made to raise the priority of file sharing for future projects and record technical solutions in a shared document.

Table of Contents

1 Introduction

While the second prototype focused on completing individual systems, this third prototype focuses on combining the work of each group member together and ensuring that the solution is usable in virtual reality. Core functionalities have been successfully implemented in the interactive and narrative tools as well as the setting design and non-player character (NPC) movement. Some features have had to be scaled back, such as the number of interactable environment objects, length of dialogue segments, and mapping character model animations to the user's controls. We anticipate the quantity of content developed and its quality at this time will be sufficient to meet the client's identified needs.

2 Objectives and Results of the Prototype

Scene transitions are integral to the narrative structure of this simulation. Given the time constraints, the mechanic chosen for this was a simple fade-in-fade-out after a set time limit within each scene. Just throwing the user into the next scene, though, would be jarring and likely frustrating for stealing their agency. Instead, the timer ending will queue up and play a final piece of companion dialogue that indicates to the user that the current scene is over. The difficulty in this was that the timer has to override the dialogue box's function, but only after the current dialogue has finished playing so as to not tear the player from what they were engaging with. After creating a new script for this and working to integrate it alongside the existing dialogue and scene change programs, the transition successfully changed from start menu to first scene to end menu.

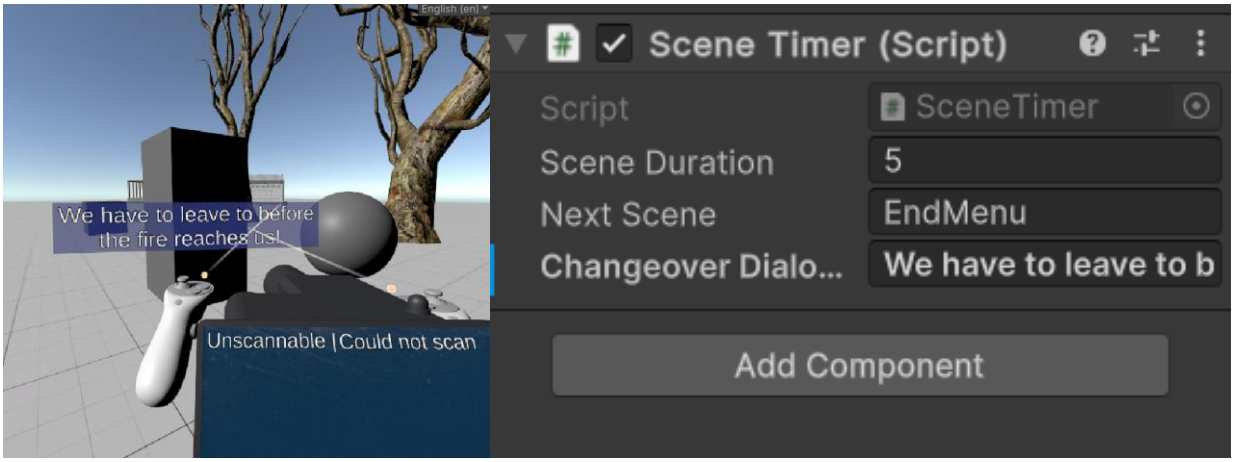
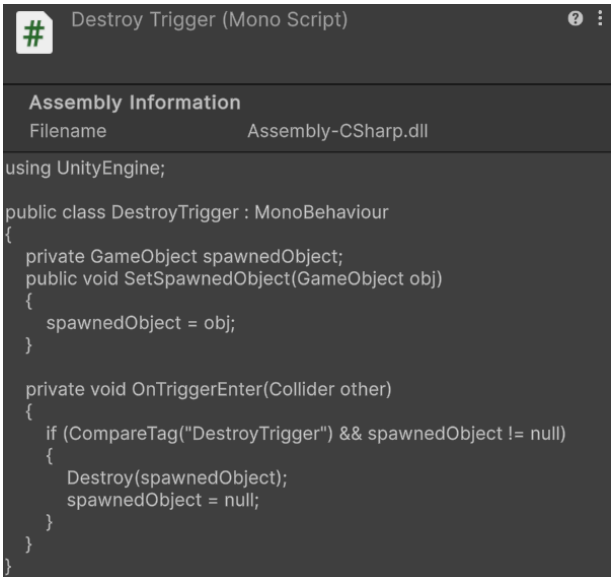


Figure 1. Scene transition dialogue (left) and inspector window controls (right)

The ability to share work done by each member to assemble the full simulation is critical, but most methods attempted proved largely infeasible. GitHub was suggested and tried, but it proved unexpectedly difficult for simply uploading files and folders. Even when functioning, it was not suitable for many of our needs due to the 25 mb limit per upload instance, which forbade sharing of larger assets such as character or terrain models. Exchanging entire project folders was, by extension, impossible. File sharing thus had to be done through physical

transfer over USB drives, which carried its own constraints due to team members needing to physically meet up, but at least met the objective of being able to move files.

Another concern for the late-stage prototyping was VR integration. Using the Unity VR template, most of the VR functionalities are already implemented; playing the simulation in the editor while a VR headset is connected will play it on the headset, with head and hand tracking built-in. In terms of coding, the one issue was being able to get the controller trigger input for the scanner to fire, which was handled earlier in development. Getting the simulation to work on the Oculus system, however, was quite difficult. The headset had to be wired to the computer to get it to work, and even then it was unreliable and demanded a Meta account to connect at all. By the time these issues were handled, there was no time to actually test our simulation in VR. The prototype failed in this regard. More efforts will have to be diverted to focus on getting this to work.



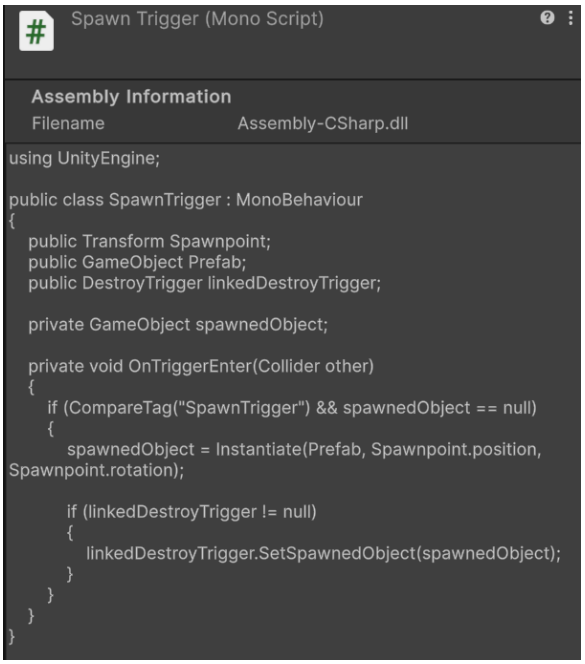


Figure 2 & 3. Destroy trigger (right) and the spawn trigger (left)

Creating the fire triggers was especially needed for the simulation. There had to be a way to have the fires always be “present” per say without always being rendered into the final product. In the second prototype that was done, the spawning trigger was created, but the only problem was that that’s all it did. The trigger only spawned in the fire and could never be removed or changed. This was eventually thought to be fixed with the next trigger, one that would have two points that when entered, one would spawn the fire and another would remove it. This seemed to work for a bit, but unfortunately it only worked with a single pair of triggers. The final and current solution was to create two separate triggers, one for spawning and the other for destroying. The way it works is by attaching a specific destroy trigger to a

specific spawn trigger, i.e. linking Destroy A to Spawn A. This allowed there to be multiple different pairs that can all spawn and destroy their own respective effects.

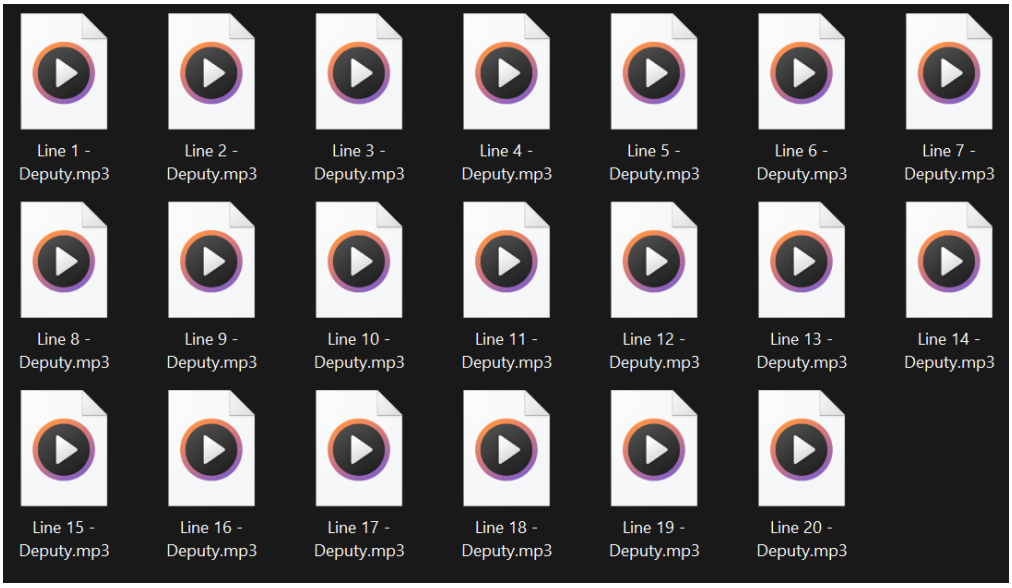


Figure 4. All voice lines prepared from the script

Since the user is accompanied by a deputy character throughout the simulation, it made sense to give them a voice to better guide and inform the player. Trying to voice the deputy ourselves was considered, but our group does not have anyone with voice acting experience. Bad voice acting and sound design can detract the immersive experience of the simulation, sohe solution to this small issue was to just have a text to speech voice do it. This ended up sounding quite good as the voice used, the Eleven Labs Adam voice, has a lot of customization to how it can pronounce things along with the speed it talks. All the voice lines were written into this TTS program, and they all came out sounding nice in the end without being monotonous.

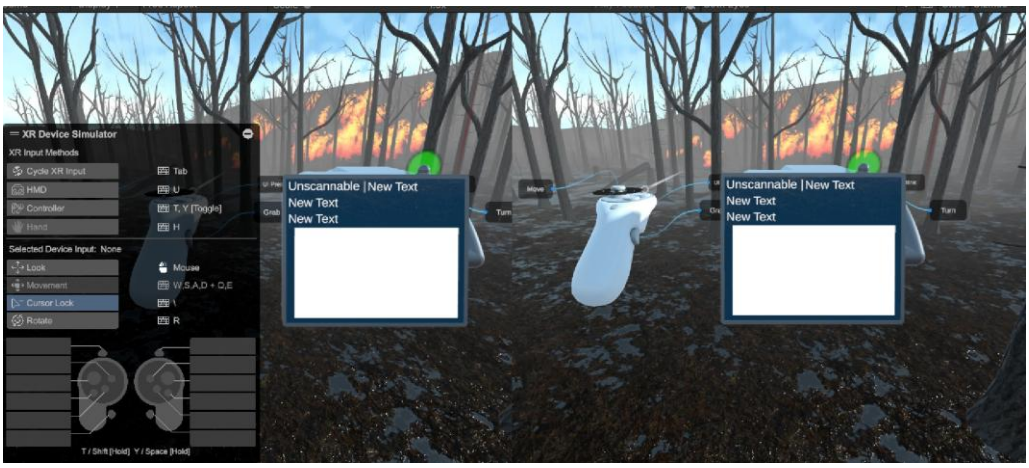


Figure 5. Dead forest scene

As we managed to transfer our files to each other, we successfully integrated the post-fire environment, the probing tool, and the characte. The character was placed into the burnt forest with XR controls for movement and interaction. We placed colliders on key objects of this scene to prevent unintended movement and scaled the trees appropriately relative to the character to keep the simulation realistic.

Figure 6. Old town scene

The pre-fire environment was completed in the third prototype with the town being made ready for character interaction. Certain trees are interactable via the scanner tool. Character movement in the town was tuned to speeds that felt realistic but not frustratingly slow.

Figure 7. Forest trail

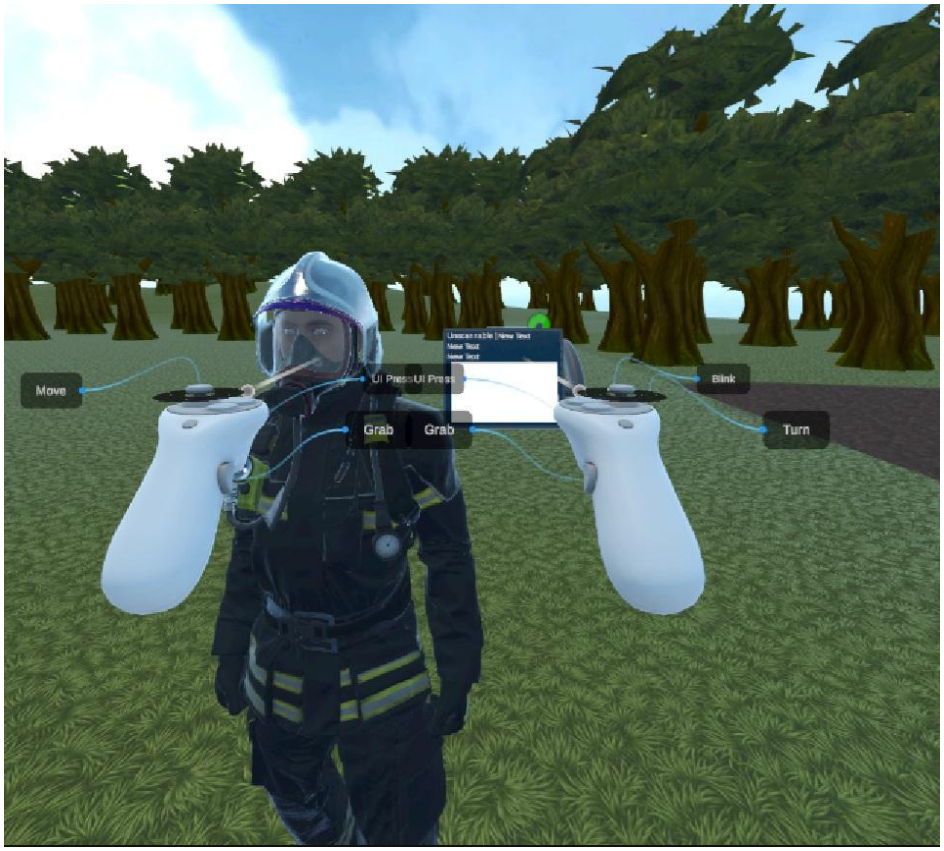


Figure 8. User and companion movement setup

As shown in figure 8 the character and the companion character are both fully set up in the scene, the user has the scanner attached to his right controller, which gives him the ability to scan assigned objects. The companion character was set up with a third person controller that allows him to move alongside the user, which is planned to be revised with an automated follower script.

Figure 9. Townspeople ready to evacuate

Townspeople from an old town setting. They are equipped with an automatic movement script that allows for them to appear to run in one direction. Sound will be attached to individual characters to elevate the sense of panic in the scene.

3 User Feedback on the Prototype

The scale of the scanner tool felt too small and toy-like when testing with the headset, and was thus scaled up by 50%, making it feel more like a real piece of equipment. Using the Oculus headset instead of the Valve Index, the resolution of text was not found to be an issue, though concerns were raised regarding the position of text, with elements of the UI being too spread out to be comfortably legible at the edges. This has since been adjusted with minor repositioning of the menus. Much concern was raised regarding the style of buildings and characters, with a large discrepancy between them and the supposed late 1900's Canadian setting, but there was not time remaining in development to adjust.

4 Updated Target Specifications and Bill of Materials

At this stage, no changes have been made to the target specifications. The final bill of materials is listed below.

Table 1. 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	Free Fire VFX - URP	https://assetstore.unity.com/packages/vfx/particles/fire-explosions/free-fire-vfx-urp-266226
8	4	$0	Animated Sun Skybox	https://assetstore.unity.com/packages/2d/textures-materials/sky/animated-sun-skybox-98447
9	4	$0	Haptic Feedback	https://assetstore.unity.com/packages/tools/integration/haptic-feedback-308330
10	4	$0	UK Terraced Houses Pack FREE	https://assetstore.unity.com/packages/3d/environments/urban/uk-terraced-houses-pack-free-63481
11	4	$0	Town Houses Pack	https://assetstore.unity.com/packages/3d/environments/urban/town-houses-pack-42717
12	4	$0	Dry Trees	https://assetstore.unity.com/packages/3d/vegetation/trees/dry-trees-86967
13	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
14	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
15	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
16	4	$0	AI Navigation Plugin 2.0.6	(Built in to Unity, requires install from package manager in the engine) AI Navigation | AI Navigation | 2.0.6
17	4	$0	Cinemachine Plugin 3.1.3	(Built in to Unity, requires install from package manager in the engine) Cinemachine package | Cinemachine | 3.1.3

18	1	$0	Starter Assets: Character controllers|URP	https://assetstore.unity.com/packages/essentials/starter-assets-character-controllers-urp-267961
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	$0	Village Houses Pack	https://assetstore.unity.com/packages/3d/characters/village-houses-pack-63695
22	1	$0	Town Creator kit Lite	https://assetstore.unity.com/packages/3d/environments/fantasy/town-creator-kit-lite-25574
23	4	$0	Fantasy Forest Environment	https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361
24	4	$0	Yughues Free Bushes	https://assetstore.unity.com/packages/3d/vegetation/plants/yughues-free-bushes-13168
25	4	$0	Low Poly StreetPack	https://assetstore.unity.com/packages/3d/environments/urban/low-poly-street-pack-67475
Total Cost - All Items		$500	-	-
Total Cost - Planned Items		$25	-	-

The Planned Items cost only covers the costs of parts 19 and 20, with other paid items either not requiring purchase or being purchased out of pocket.

5 Third Prototype Test Plan

Tests for essential elements not previously implemented have been carried over, as well as the stress tests which must be repeated as the project is built further to catch new bugs. New tests run for the latest prototype are recorded below:

Table 2. New test plan and results for the third prototype

Test ID	Test Objective	Prototype Description and Basic Test Method	Metrics, Results, and Interpretation	Estimated Test Duration
1	Reliably change betweens menus	Play the simulation from the start menu to the test scene	Yes/No for dialogue loading with or without other	5 min. to test, 2

	and scenes with dialogue triggers. Necessary for narrative structure.	to the end menu (all carried over from previous prototypes). Check whether the scene-change dialogue loads and whether it changes the scene to the end menu.	dialogue already playing, and yes/no for each scene change button working. Initial issues with dialogue not loading after changing to test scene, later resolved. Continue use of script, update to work for all scenes.	hours to debug
2	Track animations of the player model to the controller movements.	Add the Firefighter model to the VR template sample scene. Attempt to get the hand to follow the VR controllers, and get the legs to walk in the direction of user movement. Check whether the limbs move and if they do so in the correct directions.	Yes/No for hand tracking following controllers, yes/no for legs playing walking animation when user moves. Tests for this were unsuccessful, with the character failing to match its position and lateral rotation to the camera. With the time remaining, this feature is not critical and will be put on hold for development.	10 min. to test, 4 hours to construct
3	Test that scene transition works for scenes made by all team members.	Aggregate all menu and simulation scenes, add them to the build profile, and set up the scene switcher for each scene, set to a 5 second timer with the intended trigger dialogue. Play the simulation from the starting menu, try to switch through each scene. Record if it fails and log the error message.	Yes/no for each scene switching correctly.	30 min. to fully test, 3 hours to construct
4	Ensure content is educational and/or raises awareness of climate change.	Deliver the completed prototype to early test users and ask them in-person how educational they found the different elements of the simulation to be. Record also their survey responses before and after the simulation.	Measure difference in response to each survey question before and after the simulation. Interpret tester feedback to evaluate which elements were effective at being educational.	5 min. per test run + 5 min. per feedback discussion, run at least 3 trials
5	Dialogue can at minimum reach the 1 minute	The model is the dialogue script outside of Unity. Read it out loud several times and	Measure time to read the entire script and how much of the script can be	20 min. to test all trials

	time constraint.	mentally several times, recording the time it takes. Stop the time and mark the current location if the timer reaches 1.5 minutes.	comfortably read in 1.5 minutes. Tests found that just half the script takes 1.5-2 minutes to read, passing the time requirement. Keep using the script.	

A more detailed test plan that includes results from past tests has been included in the appendix. The specific duties of each member in development of the latest simulation are tabulated as follows:

Table 3. Development roles for the latest prototype

Name	Content Roles
Mohammad	Character animations, project file sharing
Malcolm	Lighting and shader support, dialogue voiceovers
Krithivaas	Building the first and second scene settings, shader support
James	Scene transitions, scanner tool remodel

6 Conclusions and Recommendations

Having finished drafting all mechanical systems in the second prototype, this final prototype was concentrated on debugging and finalizing these systems and completing the scene designs. By collaborating more closely when file sharing, many of the issues with materials and shaders breaking on transfer were able to be fixed. The testing focused on this as well as proofing the VR compatibility and ensuring the scene transition works. These tests were broadly successful, but with how late they were run, there was not time allocated to run more comprehensive iterations. No changes were made to the design specifications, and some environmental elements were added to the bill of materials to improve the realistic aesthetic. It is recommended that project sharing be reprioritized as an early test that can be repeated as each group member makes major progress to patch anticipated bugs.

7 Future Work

Past this document, the opportunity for further content development is limited. The remaining work is focused on running more rigorous tests and making small refinements such as stylistic UI and asset model changes or adjusting the placements of scenery items.

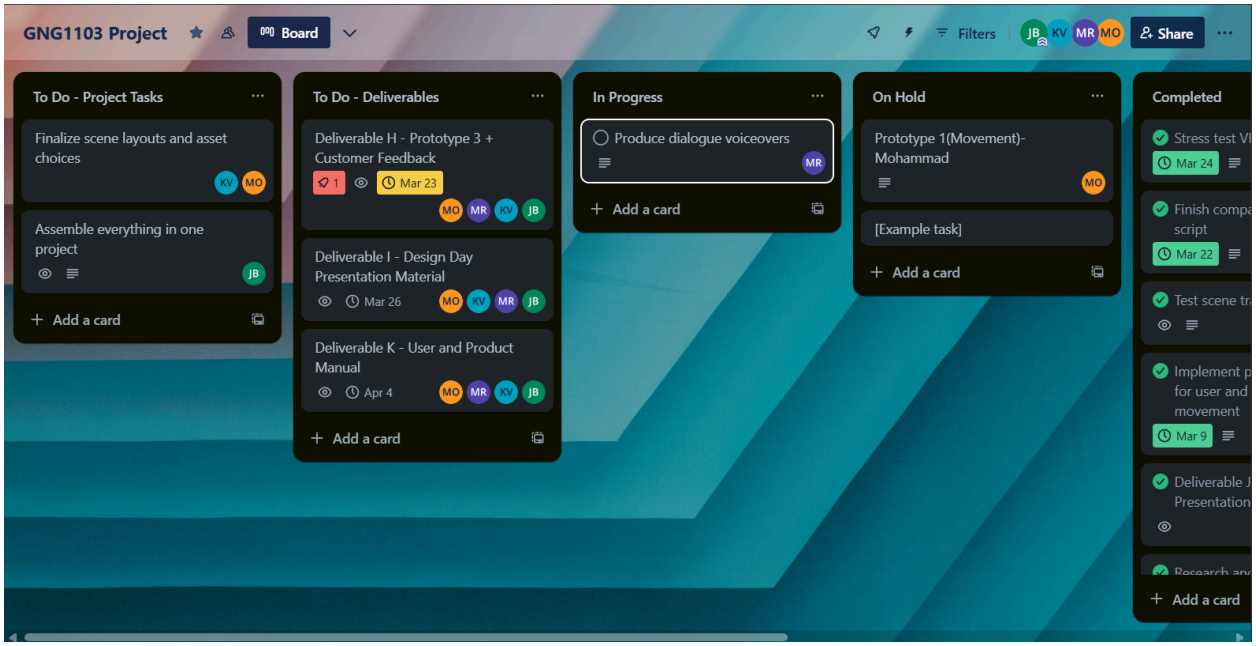
8 References

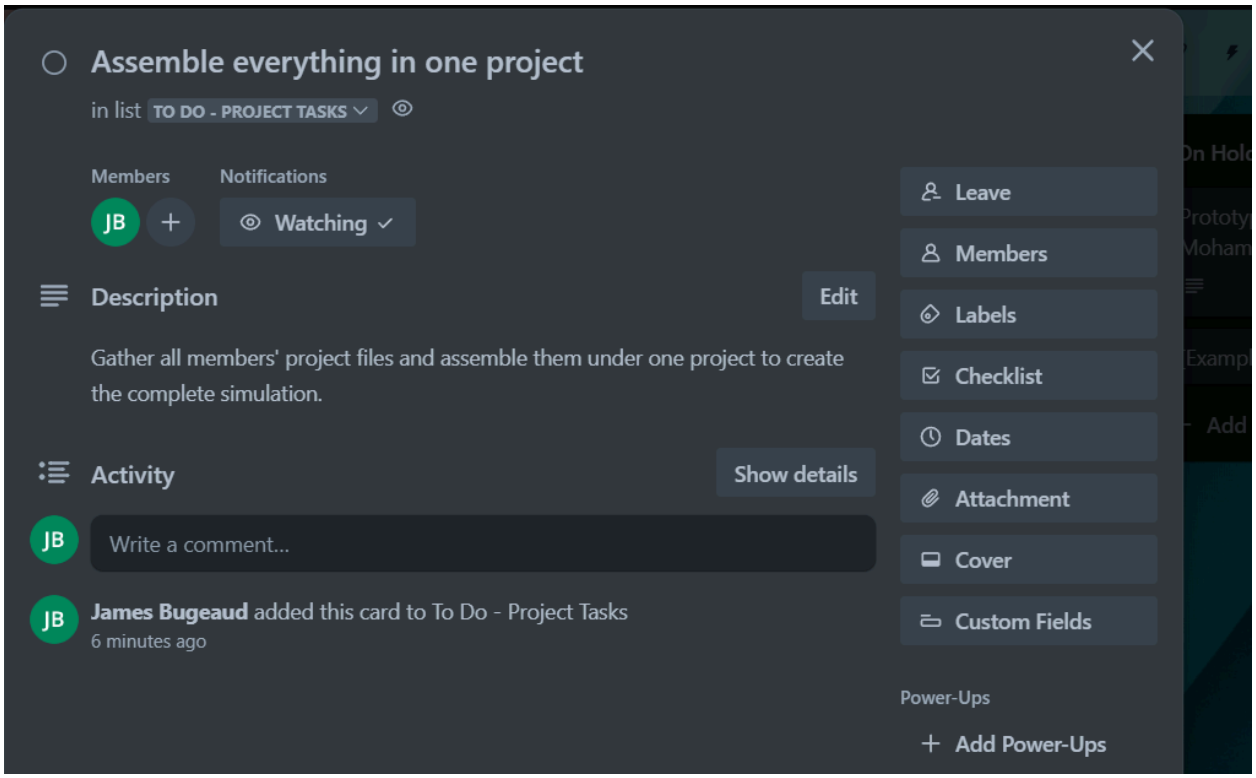
No external documents were referenced for this document outside the bill of materials.

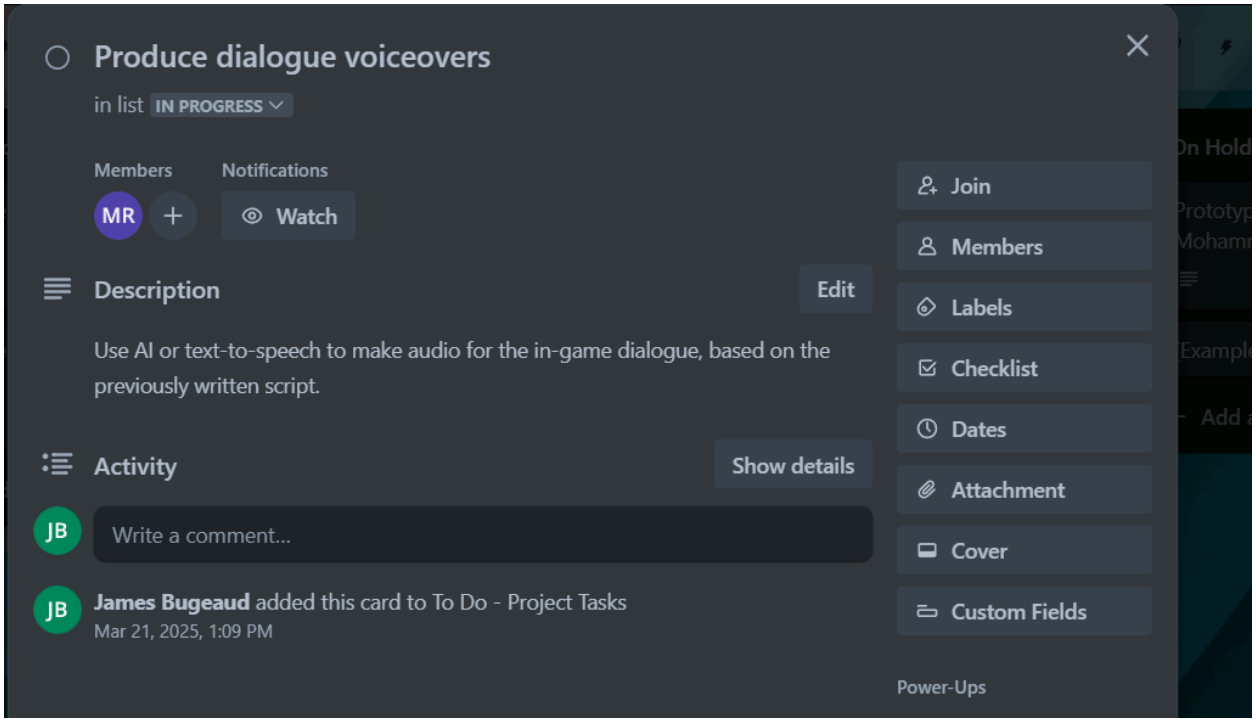
9 Appendix

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. Continue use of scanner script	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. Debug existing sound script.	
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	Setting elements were found to be appropriately proportioned for the player height.	Success. Continue to use the scale set in this test as reference for further development.	
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	Surveys were majority positive on the depiction of the fire inspector, but raised some concerns about the realism of the light and shadow effects on the model.	Pass. Continue using model and add a later stage low-priority test for fixing the lighting effects on the character.	
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	Sound was properly altered based on user's facing direction. Surveys were broadly positive on the ambient sound effects.	Pass. Continue to use the sound effects as they are.	
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. Contiue use of localization plugin	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. Redo for later prototypes	
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	Unable to test due to delays in assmebling the project fully.	Failed. Attempt to test in remaining time, use box colliders as invisible walls to patch any major issues.	
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. Continue using current scanner script.	
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	Companion stayed put when moving near it and navigated aroudn obstacles when walking or teleporting far away.	Pass as movement meets distance and reliability criteria. Continue use of companion follower script.	
14	Dialogue queues and plays lines when triggered	Performance Measurement (of dialogue system)	Dialogue test: Dialogue queue accepts and returns lines in the correct order when triggered	Write a dialogue script that can queue and write different sets of lines. Attach this to three different taggable objects, each with its own set of at least three dialogue lines, and scan each varying orders. For each tag, mark which dialogue lines do and don't play, tracking whether lines from certain objects are skipped or repeated in relation to the scanning order. Estimated 4 hours to build, 15 min. to test.	Consistency of the dialogue system	Yes/No for each line played no or less than once and in order	HiFi Focused	Physical	Initial issues with the first line not playing and the box not resizing. Later solved, so box will autoscale to fit the text and play all queued lines in order.	Pass. Continue to the dialogue script, update it to fuinction also with voiceover audio.	
15	Scene transitions function on both button and dialogue triggers	Risk Management (of simulation halting early)	Transition test: The scene switch and scene changer scripts consistently take the user to the prescribed next scene	Copy the existing starting menu scene and tweak it into an end menu. Write a new script that uses the dialogue and scene switcher to make a scene changing dialogue button. Put this in the simulation testing ground scene, then play the simulation from the start menu. Click through everything and ensure that each continue button triggers the fade effect and changes scenes. Enusre this occurs whether or not dialogue is queued or currently playing	Presence of bugs in the final scene switcher script	Yes/No for the scene switch working from menu to simulation, Yes/No for scene switch working in each dialogue state	HiFi Focused	Physical	Early models failed to fade when using the headset. Revised script fixed the fade issue after replacing the scene switcher with an XR canvas.	Pass as script eventually worked. Continue to use script, monitor for bugs when transitioning between narrative scenes.	*Always* use XR canvas for anything that will be displayed in VR.
16	The Unity XR rig properly maps the user's hand and head movements to the simulation character	Risk Management (of control scheme functionality)	VR test: The controls in VR are functionally identical to those tested with the MockHMD plugin	Load the prototype in its current state onto one of the MakerSpace laptops and play it in VR. Play through as much of the simulation as possible, stopping if the controls don't work. Take note of any bugs that occur during this that do not immediately break the playability.	Joystick movement, controller movement, head movement, and trigger movement all tracking and activating. on command	Yes/No for each of the previously listed control elements functioning.	HiFi Comprehensive	Physical	Initial failure with default settings. Fixed with some tweaks, but user's speed felt too slow.	Pass. Increased player walking speed, continue to use the XR rig.	If not working, go to project settings => XR interaction toolkit and turn off the XR device simulator


GNG1103 Project
Board
Filters
Share
To Do - Project Tasks
Finalize scene layouts and asset choices
Assemble everything in one project
Add a card
To Do - Deliverables
Deliverable H - Prototype 3 + Customer Feedback
Mar 23
Deliverable I - Design Day Presentation Material
Mar 26
Deliverable K - User and Product Manual
Apr 4
Add a card
In Progress
Produce dialogue voiceovers
Add a card
On Hold
Prototype 1(Movement)- Mohammad
[Example task]
Add a card
Completed
Stress test
Mar 24
Finish
script
Mar 22
Test scene
Implement
for user and
movement
Mar 9
Deliverable
Presentation
Add a card


Assemble everything in one project
in list TO DO - PROJECT TASKS
Members
Notifications
Watching
Description
Edit
Gather all members' project files and assemble them under one project to create the complete simulation.
Activity
Show details
Write a comment...
James Bugeaud added this card to To Do - Project Tasks
6 minutes ago
Leave
Members
Labels
Checklist
Dates
Attachment
Cover
Custom Fields
Power-Ups
Add Power-Ups


Produce dialogue voiceovers
in list IN PROGRESS
Members
MR
Notifications
Watch
Description
Edit
Use AI or text-to-speech to make audio for the in-game dialogue, based on the previously written script.
Activity
Show details
JB
Write a comment...
JB
James Bugeaud added this card to To Do - Project Tasks
Mar 21, 2025, 1:09 PM
Join
Members
Labels
Checklist
Dates
Attachment
Cover
Custom Fields
Power-Ups