Verification and Validation Report: Digital Twin Forest

Team # 8, Forest Mirror
Bowen Zhang
Tingyu Shi
Jiacheng Wu
Junhong Chen
Yichen Jiang

April 3, 2023

1 Revision History

Date	Version	Notes
March 6th	1.0	First Draft
April 2	2.0	Final Version

2 Symbols, Abbreviations and Acronyms

The following are some naming conventions and definitions from SRS:

- LiDAR: Light Detection and Ranging(Scanning Technology)
- Plot: A square-shaped area in the forest. There are 14 plots in total.
- LAI: Leaf Area Index
- DBH: Diameter at Breast Height
- Target Forest: Target forest refers to the natural forest located at Turkey Point in Ontario.
- Digital Twin Forest: The virtual representation of the target forest.
- Forest Data: Forest Data include environmental data and tree parameters.
- Environmental Data:
  - Humidity
  - Temperature
  - Soil Carbon Content
  - Soil Nitrogen Content
  - LAI
- Tree parameters:
  - Density
  - Height
  - Age
  - DBH
- Tree types. There are 7 different types of trees, the following are the details:
  - Red Pine
  - Oak
  - Birch
  - Beech
  - White Pine
  - Red Maple
  - Red Oak

Contents

List of Tables

List of Figures

3 SRS Verification

- All the functional and non-functional requirements have been achieved.
- You can check the latest SRS here.

4 Design Documentation Verification

According to the SRS verification documented above, Design documentation is revised and modified. The major modifications are listed below:

- We rearranged the data categories and hierarchy, and removed the overall forest view. For environmental data, keep data `Humidity`, `Soil Carbon Content`, `Soil Nitrogen Content`, `Temperature`, and `LAI`. We added a pie chart to show the percentages of each tree type. For tree parameters, we kept data `Density`, `DBH`, `Height`, `Age`. Overall forest view is combined with the models for each plot instead, and the related information can still be reached.
- Change the stored models to the instantly generated models. Models for 14 plots are both space-consuming and redundant for our product. Store the constraints and methods to generate models instantly, instead of the models themselves.
- Remove the information for each tree. Instead, show the information about each kind of trees for each plot. This modification is for better accomplishing the scientific purpose, as the scientific work of our stakeholders concerns about the overall information within a certain area, instead of specific individuals. Also, due to limited resources and schedule, precisely reappearing every detail of the target forest in the model is not realistic.

5 VnV Plan Verification

During the test processes, the Forest Mirror team performed to verify the implementations of functional requirements, team members found that the Automated Testing and Verification Tool section is not correct because automated testing is not applicable to the application. The back-end code of the application is bound to the game components in Unity, which can not be tested by solely using automated tools and running it so the team used manual tests on Unity to verify the implementations of requirements. After the mistake was discovered, the team decided to delete this section from the VnV plan. The Forest Mirror team gathered some feedback and problems occurring in the VnV plan from the other team members and TAs and modified some contents in the presenting version of the VnV plan according to that feedback.

VnV plan has been modified according to the above result.

6 Implementation Verification

Our team deployed both dynamic and static testing to verify the implementation of our project. For dynamic testing, since we implement the software with Unity, we used integration testing and system testing most of the time to monitor each component to operate normally. Also, the tests cases are evaluated to make sure all lines of code are covered. All the dynamic testing was done

without any error.
During the development of the project, we constantly communicated with our stakeholder —- Dr.Gonsamo to ensure that we are doing the right product. Also, by demonstrating the proof of concept to stakeholders, we can ensure that our clients will be satisfied with our product.
For static testing, our team used code inspection and document inspection to make sure that the design is following our expectations (high cohesion and low coupling, easy to modify) and there is no big issue in the code. Our purpose for static testing is to make sure the code have good quality and complete. After our testing, we evaluated our code with good quality and completeness.

7 Automated Testing and Verification Tools

All the tests were conducted manually. So this part is not applicable.

8 Software Validation

After the Rev0 demo, we had a meeting with Dr. Gonsamo, and he gave us the following advice:

- We need to adjust the initial camera position when users just enter the forest model. The camera position should allow users to have a full view of the plot.
- We should add another GUI for displaying leaf information.
- We should implement seasonal change.
- Tree distribution should follow the real forest tree distribution.

9 Functional Requirements Evaluation

9.1 Presentation

The following testing results of test cases related to the presentation of our products.

9.1.1 Test-FR1

- Inputs: Testers clicked `Instruction` button for 10 times and reset the system to the initial state(Main page) after each click.
- Excepted Value: Instruction page should appear on the screen 10 times.
- Actual Value: Instruction page appeared on the screen 10 times.
- Result: PASS

9.1.2 Test-FR2

- Inputs: Testers clicked `Contact Us` button for 10 times and reset the system to the initial state(Main page) after each click.
- Excepted Value: Contact Us page should appear on the screen 10 times.
- Actual Value: Contact Us page appeared on the screen 10 times.
- Result: PASS

9.1.3 Test-FR3

- Inputs: Testers clicked `Start` button for 10 times and reset the system to the initial state(Main page) after each click.
- Excepted Value: Full view of forest plot 1 should appear on the screen 10 times.
- Actual Value: Full view of forest plot 1 appeared on the screen 10 times.
- Result: PASS

9.1.4 Test-FR7.1

- Inputs: Testers clicked `Environmental Data` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but environmental data are not displayed.
- Excepted Value: Environmental data should appear on the left side of the screen 10 times
- Actual Value: Environmental data appeared on the left side of the screen 10 times.
- Result: PASS

9.1.5 Test-FR7.2

- Inputs: Testers click `Tree Parameters` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but tree parameters are not displayed.
- Excepted Value: Tree parameters should appear on the right side of the screen 10 times
- Actual Value: Tree parameters appeared on the right side of the screen 10 times.
- Result: PASS

9.1.6 Test-FR10.1

- Inputs: Testers recorded all the environmental data from the GUI. This includes environmental data for 14 plots and the overall forest.
- Expected Value: All the environmental data recorded from the GUI should match the JSON files stored here.
- Actual Value: All the environmental data recorded from the GUI matches the JSON files.
- Result: PASS

9.1.7 Test-FR10.2

- Inputs: Testers record all the tree parameters from the GUI. This includes tree parameters of 7 types of trees from 14 plots and the overall forest.
- Expected Value: All the tree parameters recorded from the GUI should match the JSON files stored here.
- Actual Value: All the tree parameters recorded from the GUI matches the JSON files.
- Result: PASS

9.1.8 Test-FR13

- Inputs: Testers updated tree parameters(including tree density, tree height, tree DBH) for all 7 types of trees in 14 plots.
- Expected Value: Tree models should change according to the data update.
- Actual Value: Tree models changed according to the data update.
- Result: PASS

9.1.9 Test-FR14

- Inputs: Testers clicked on `Seasonal Change` button 10 times in succession.
- Expected Value: Summer season and winter season should appear 5 times, respectively.
- Actual Value: Summer season and winter season appeared 5 times, respectively.
- Result: PASS

9.1.10 Test-FR15

- Inputs: Testers clicked on `Switch` button 10 times and reset to the initial state after each click event.
- Expected Value: The pie chart should appear on the left side of the screen 10 times.
- Actual Value: The pie chart appeared on the left side of the screen 10 times.
- Result: PASS

9.1.11 Test-FR16

- Inputs: Testers got the top view of each plot.
- Expected Value: The virtual tree model distributions should be consistent with satellite pictures.
- Actual Value: The virtual tree model distributions are consistent with satellite pictures.
- Result: PASS

9.2 Users' interaction with the product

9.2.1 Test-FR4.1

- Inputs: Testers clicked `Environmental Data` button 10 times and reset the system to the initial state after each click. The initial state is defined as "environmental data are displayed"
- Expected Value: Environmental data display should disappear 10 times.
- Actual Value: Environmental data display disappeared 10 times.
- Result: PASS

9.2.2 Test-FR4.2

- Inputs: Testers clicked `Tree Parameters` button 10 times and reset the system to the initial state after each click. The initial state is defined as "tree parameters are displayed"
- Expected Value: Tree parameters display should disappear 10 times.
- Actual Value: Tree parameters display disappeared 10 times.
- Result: PASS

9.2.3 Test-FR5

- Inputs: Testers clicked `Environmental Data` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but environmental data are not displayed.
- Excepted Value: Environmental data should appear on screen 10 times.
- Actual Value: Environmental data appeared on the screen 10 times.
- Result: PASS

9.2.4 Test-FR6

- Inputs: Testers clicked `Tree Parameters` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but tree parameters are not displayed.
- Excepted Value: Tree parameters should appear on screen 10 times.
- Actual Value: Tree parameters appeared on the screen 10 times.
- Result: PASS

9.2.5 Test-FR8

- Inputs: Testers pressed `W` key, `A` key, , `S` key, and `D` key.
- Expected value:
  - Users' first view should move forward after pressing `W` key.
  - Users' first view should move left after pressing `A` key.
  - Users' first view should move backward after pressing `S` key.
  - Users' first view should move right after pressing `D` key.
- Actual value:
  - Users' first view moved forward after pressing `W` key.
  - Users' first view moved left after pressing `A` key.
  - Users' first view moved backward after pressing `S` key.
  - Users' first view moved right after pressing `D` key.
- Result: PASS

9.2.6 Test-FR9

- Inputs: Testers moved the mouse in 4 different directions, which are left, right, forward, and backward.
- Expected value: The user's point of view should rotate according to the input.
  - Move the mouse forward: The user's first point of view should rotate up.
  - Move the mouse backward: The user's first point of view should rotate down.
  - Move the mouse left: The user's first point of view should rotate left.
  - Move the mouse right: The user's first point of view should rotate right.
- Actual value: The user's point of view rotated according to the input.
  - Move the mouse forward: The user's first point of view rotated up.
  - Move the mouse backward: The user's first point of view rotated down.
  - Move the mouse left: The user's first point of view rotated left.
  - Move the mouse right: The user's first point of view rotated right.
- Result: PASS

9.2.7 Test-FR11

- Inputs: Testers clicked `Quit` button located in the main page.
- Expected value: The product should terminate.
- Actual value: The product terminated.
- Result: PASS

9.2.8 Test-FR12

- Inputs: Testers entered new data in the input box and clicked `Update` button.
- Expected value: A text should pop up indicating if the update was successful. And environmental data display and tree parameters display should reflect the newly entered data.
- Actual value: A text popped up indicating if the update was successful. And environmental data display and tree parameters display reflected the newly entered data.
- Result: PASS

NOTE: The above test was conducted for 14 plots and 7 tree types. All tests passed.

10 Nonfunctional Requirements Evaluation

10.1 Look and Feel Requirements Evaluation

10.1.1 Test-NFR-LF1.1

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the first question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 10 chose A and 7 chose B in this question.(17/20)
- Result Analysis: PASS

10.1.2 Test-NFR-LF2.1

- Testing Process: Testers compared the forest data stored in the software with physical measurements and calculate the relative errors of the data
- Expected Result: All the forest data have errors less than 0.1 compared to the actual data our clients provide us
- Actual Result: The data has relative errors less than 0.1.
- Result Analysis: PASS

10.1.3 Test-NFR-LF2.2

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the second question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 9 chose A and 9 chose B in this question.(18/20)
- Result Analysis: PASS

10.2 Usability and Humanity Requirements Evaluation

10.2.1 Test-NFR-UH1.1

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the third question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 5 chose A and 12 chose B in this question.(17/20)
- Result Analysis: PASS

10.2.2 Test-NFR-UH2.1

- Testing Process: Every member of our team inspected all the UI components that contain text in the software to see if there are any non-English texts.
- Expected Result: 100 percent of the texts should be used in English except for numbers.
- Actual Result: 100 percent of the texts are written in English except for numbers.
- Result Analysis: PASS

10.2.3 Test-NFR-UH3.1

- Testing Process: Testers launched the system, went to the main page and clicked the instructions option.
- Expected Result: Instructions are displayed on the screen.
- Actual Result: Instructions are displayed on the screen.
- Result Analysis: PASS

10.2.4 Test-NFR-UH4.1

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the fourth question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 12 chose A and 7 chose B for this question.(19/20)
- Result Analysis: PASS

10.2.5 Test-NFR-UH4.2

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the fifth question in the questionnaire
- Actual Result: Out of 20 interviewed users, 11 chose A and 9 chose B for this question.(20/20)
- Result Analysis: PASS

10.2.6 Test-NFR-UH5.1

- Testing Process: We asked five users to navigate through our system using a mouse and keyboard.
- Expected Result: All of them will go through the system without any trouble.
- Actual Result: All of them can go through the system without any trouble.
- Result Analysis: PASS

10.2.7 Test-NFR-UH5.2

- Testing Process: We conducted interviews with random people to gather feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the sixth question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 15 of them chose A and 1 of them chose B(16/20)
- Result Analysis: PASS

10.3 Performance Requirements Evaluation

10.3.1 Test-NFR-PR1.1

- Testing Process: Testers launched the software, set a timer, and recorded the time it took to respond to the events from the mouse and the keyboard.
- Expected Result: The system should respond to any requests within 2 seconds.
- Actual Result: The average response time is less than 2 seconds.
- Result Analysis: PASS

10.3.2 Test-NFR-PR1.2

- Testing Process: Testers launched the software, moved the mouse, clicked it, and pressed the keyboard. At the same time, they recorded the FPS displayed on the screen and the time.
- Expected Result: 80 percent of the time, the software should have at least 30 FPS.
- Actual Result: The software has over 30 FPS all the time when it is running.
- Result Analysis: PASS

10.3.3 Test-NFR-PR1.3

- Testing Process: Testers clicked on the start option on the main page and recorded the time the software took to show all the models.
- Expected Result: The system should load all the models within 10 seconds.
- Actual Result: The system loads all the models in less than 10 seconds.
- Result Analysis: PASS

10.3.4 Test-NFR-PR3.1

- Testing Process: Testers recorded all the data and parameters displayed in the application and compared the data with the actual data collected by Dr.Gonsamo and his lab members, then testers computed the relative error for each data.
- Expected Result: All data should have relative errors less than 0.05.
- Actual Result: All the forest data have relative errors less than 0.05.
- Result Analysis: PASS

10.3.5 Test-NFR-PR4.1

- Testing Process: Testers clicked on the application icon and waited for the application to open.
- Expected Result: The application should open normally every time when testers click on the application icon.
- Actual Result: The application open normally every time when testers click on the application icon.
- Result Analysis: PASS

10.3.6 Test-NFR-PR4.2

- Testing Process: Testers launched the software and ran the application in the background for 24 hours.
- Expected Result: The application should not crash in 24 hours.
- Actual Result: The application runs normally and does not crash in 24 hours.
- Result Analysis: PASS

10.3.7 Test-NFR-PR5.1

- Testing Process: Testers ran the application without an internet connection, then clicked all the buttons in the system, pressed the keyboard keys, moved the mouse on the forest page, and checked if the system responded correctly to the mouse and the keyboard inputs.
- Expected Result: the system should respond correctly to the mouse and the keyboard inputs.
- Actual Result: the system responds correctly to the mouse and the keyboard inputs.
- Result Analysis: PASS

10.3.8 Test-NFR-PR6.1

- Testing Process: Testers opened the property window of the application and checked its size.
- Expected Result: The size of the software should be less than 10 GB.
- Actual Result: The size of the software is around 250MB.
- Result Analysis: PASS

10.4 Operational and Environmental Requirements Evaluation

10.4.1 Test-NFR-OE1.1

- Testing Process: Testers ran the application on both desktops and laptops to check if it worked normally.
- Expected Result: The application should ran normally on desktops and laptops.
- Actual Result: The application runs normally on desktops and laptops.
- Result Analysis: PASS

10.4.2 Test-NFR-OE1.2

- Testing Process: Testers ran the software and performed all kinds of actions in the software using a mouse and keyboard.
- Expected Result: All actions should be accomplished without any problems.
- Actual Result: All actions are accomplished without any problems.
- Result Analysis: PASS

10.4.3 Test-NFR-OE2.1

- Testing Process: Testers ran the software on Windows 10 or later version or MacOS 12 or later version and performed operations on the software.
- Expected Result: All the operations should be accomplished normally.
- Actual Result: All the operations are accomplished normally
- Result Analysis: PASS

10.4.4 Test-NFR-OE3.1

- Testing Process: Testers downloaded and installed the application from GitHub and checked if it could run normally.
- Expected Result: The application should be downloaded successfully from GitHub to the computer, and it can run without any errors and bugs.

- Actual Result: The application is downloaded successfully from GitHub to the computer and runs without any errors or bugs.
- Result Analysis: PASS

10.4.5 Test-NFR-OE4.1

- Testing Process: Testers looked at the git commit history and checked if there were weekly updates.
- Expected Result: There should be weekly updates in the git commit history.
- Actual Result: There are weekly updates in the git commit history.
- Result Analysis: PASS

10.4.6 Test-NFR-OE4.2

- Testing Process: Testers performed tests on the previous functions after each update and recorded the test results.
- Expected Result: All the previous functions should be unaffected, and all the test cases should be passed.
- Actual Result: All the previous functions are unaffected, and all the test cases are passed.
- Result Analysis: PASS

10.5 Maintenance and Support Requirements Evaluation

10.5.1 Test-NFR-MS1.1

- Testing Process: Testers checked the documentation of the product to see if the new features, functions, and any modifications were added to the documentation.
- Expected Result: All features, functions and modifications should be on the documentation.
- Actual Result: New features, functions and modifications after February are not on the documentation.
- Result Analysis: FAIL

10.5.2 Test-NFR-MS1.2

- Testing Process: Testers read the documentation of the product to see if the documentation specifies the functions clearly.
- Expected Result: All functions on the documentation should be clearly documented.
- Actual Result: All functions on the documentation are clearly documented.
- Result Analysis: PASS

10.5.3 Test-NFR-MS1.3

- Testing Process: Testers recorded the time from the bug occurred to the time when the bug was fixed and Measured the time to see if it was within three days.
- Expected Result: All bugs should be fixed within three days.
- Actual Result: All bugs are fixed within three days.
- Result Analysis: PASS

10.5.4 Test-NFR-MS2.1

- Testing Process: Testers first looked for the contact method on the contact page, then used the contact method to contact the developers to see if they could contact the developers successfully or not.
- Expected Result: Testers should be able to reach the developers successfully.
- Actual Result: Testers are able to reach the developers successfully.
- Result Analysis: PASS

10.5.5 Test-NFR-MS3.1

- Testing Process: Testers tried to launch the application on more than one operating system to see if it could be run successfully on a different system or not.
- Expected Result: The application should be able to launch on more than one operating system.
- Actual Result: The application is able to launch on more than one operating system.
- Result Analysis: PASS

10.5.6 Test-NFR-MS3.2

- Testing Process: Testers tried to launch the application on the devices located indoors and outdoors respectively to see if the application could be used both indoors and outdoors.
- Expected Result: The application should be able to launch on devices located both indoors and outdoors.
- Actual Result: The application is able to launch on devices located both indoors and outdoors.
- Result Analysis: PASS

10.6 Security Requirements Evaluation

10.6.1 Test-NFR-SR1.1

- Testing Process: Testers tried to find and download the product from GitHub and other websites.
- Expected Result: Testers can not download the product in other websites other than GitHub.
- Actual Result: Testers can not download the product in other websites other than GitHub.
- Result Analysis: PASS

10.6.2 Test-NFR-SR1.2

- Testing Process: Testers tried to update the data of trees and forests via the interface and anywhere else to see if they could update the data successfully or not.
- Expected Result: Testers should not be able to update the data except through the interface.
- Actual Result: Testers can delete or modify the .json files to modify the data.
- Result Analysis: FAIL

10.6.3 Test-NFR-SR2.1

- Testing Process: Testers injected 100 errors on purpose in our application and checked if the scan results of the computer security application would detect the errors in the computer system or not.
- Expected Result: The number of errors that the product propagates should be less than 2.
- Actual Result: 0 error is propagated.
- Result Analysis: PASS

10.6.4 Test-NFR-SR2.2

- Testing Process: Testers used the software and performed some complicated tasks such as zooming in and zooming out for a long time to see if the application will crash or not.
- Expected Result: The application should not crash.
- Actual Result: The application does not crash.
- Result Analysis: PASS

10.6.5 Test-NFR-SR2.3

- Testing Process: Testers inputted some invalid data in the `Update Data` page.
- Expected Result: Fail to update the data. And a warning message should pop up indicating an invalid update operation.
- Actual Result: Fail to update the data. And a warning message pops up indicating an invalid update operation.
- Result Analysis: PASS

10.6.6 Test-NFR-SR2.4

- Testing Process: Testers manually updated some data into the product, took records, and compared it with the displayed data.
- Expected Result: The data should be consistent with the updated data.
- Actual Result: The data are consistent with the updated data.
- Result Analysis: PASS

10.6.7 Test-NFR-SR2.5

- Testing Process: Testers manually compared the GUI and data files.
- Expected Result: Each data should have a unique position to display in the GUI, and each GUI should correspond to unique data.
- Actual Result: Each data has one unique position to display in the GUI, and each GUI corresponds to unique data.
- Result Analysis: PASS

10.6.8 Test-NFR-SR3.1

- Testing Process: Testers used a questionnaire to check if the users have been asked to provide personal information when using the product.
- Expected Result: All the users choose B in the ninth question in the questionnaire.
- Actual Result: No user was asked to provide personal information.
- Result Analysis: PASS

10.6.9 Test-NFR-SR3.2

- Testing Process: Testers used a questionnaire to check if the product has sent a notification without permission.
- Expected Result: All the users choose B in the seventh question in the questionnaire.
- Actual Result: All the users chose B in the seventh question in the questionnaire.
- Result Analysis: PASS

10.7 Cultural and Political Requirements Evaluation

10.7.1 Test-NFR-CP1.1

- Testing Process: Testers used a questionnaire to check if any users have been offended by the product.
- Expected Result: All users chose B for the eighth question in the questionnaire.
- Actual Result: All users chose B for the eighth question in the questionnaire.
- Result Analysis: PASS

10.8 Legal Requirements Evaluation

10.8.1 Test-NFR-LR2.1

- Testing Process: Testers asked the users if any part appeared lawfully unreasonable when spreading the questionnaires.
- Expected Result: Users should not notice any part lawful unreasonable.
- Actual Result: No user notices any part lawful unreasonable.
- Result Analysis: PASS

11 Comparison to Existing Implementation

This section will not be appropriate for every project.

12 Unit Testing

Unit testing does not accommodate Unity's workflow. Each script is automatically compiled by Unity. If the team wants to test a particular module, they still have to run the whole system and check the Unity console. As a result, all unit testing is included in the system testing. Please check sections 3 and 4 instead.

13 Changes Due to Testing

- Feedback from the Rev 0 demo: It does not make sense that there is so much grass on the forest floor. The tree heights are unbalanced because no tree shall grow in the shade of other trees.
  The team change the ground texture to show more details of the soil and fallen leaves instead of grass for realism. The team measures the accurate height of the tree model, later the models are resized to fit the data collected by drones.

- Feedback from the Rev 0 demo: There are spelling and grammar errors in the user interface and codes.
  The team goes through the user interface and comments to fix the typo in the project.

- Feedback from the Rev o demo: The user interface is not straightforward. It is hard to learn how to update forest data.
  The team rearranges the user interface to make it easier to learn.

- Feedback from Dr. Gonsamo's lab: The species and tree distribution shall be consistent with the real forest.
  The team checked the species counted by Dr. Gonsamo's lab and updated the models. The team later writes a script to dynamically control the distribution of tree models based on satellite photos.

14 Automated Testing

Automated testing does not accommodate Unity's workflow, everything is tested manually in Unity, such as UI, scene manager, and terrain.etc.

15 Trace to Requirements

Function Requirement	Test Case ID
FR1	Test-FR1
FR2	Test-FR2
FR3	Test-FR3
FR4	Test-FR4.1, Test-FR4.2
FR5	Test-FR5
FR6	Test-FR6
FR7	Test-FR7.1, Test-FR7.2
FR8	Test-FR8
FR9	Test-FR9
FR10	Test-FR10.1, Test-FR10.2
FR11	Test-FR11
FR12	Test-FR12
FR13	Test-FR13.1
FR14	Test-FR14
FR15	Test-FR15
FR16	Test-FR16

Table 1: Traceability between Functional Requirements and Tests

Non-functional Requirement	Test Case ID
NFR-LF1.1	Test-NFR-LF1.1
NFR-LF2.1	Test-NFR-LF2.1
NFR-LF2.2	Test-NFR-LF2.2
NFR-UH1.1	Test-NFR-UH1.1
NFR-UH2.1	Test-NFR-UH2.1
NFR-UH3.1	Test-NFR-UH3.1
NFR-UH4.1	Test-NFR-UH4.1
NFR-UH4.2	Test-NFR-UH4.2
NFR-UH5.1	Test-NFR-UH5.1
NFR-UH5.2	Test-NFR-UH5.2
NFR-PR1.1	Test-NFR-PR1.1
NFR-PR1.2	Test-NFR-PR1.2
NFR-PR1.3	Test-NFR-PR1.3
NFR-PR3.1	Test-NFR-PR3.1
NFR-PR4.1	Test-NFR-PR4.1
NFR-PR4.2	Test-NFR-PR4.2
NFR-PR5.1	Test-NFR-PR5.1
NFR-PR6.1	Test-NFR-PR6.1

Table 2: Traceability between Non-Functional Requirements and Tests Part 1

Non-functional Requirement	Test Case ID
NFR-OE1.1	Test-NFR-OE1.1
NFR-OE1.2	Test-NFR-OE1.2
NFR-OE2.1	Test-NFR-OE2.1
NFR-OE3.1	Test-NFR-OE3.1
NFR-OE4.1	Test-NFR-OE4.1
NFR-OE4.2	Test-NFR-OE4.2
NFR-MS1.1	Test-NFR-MS1.1
NFR-MS1.2	Test-NFR-MS1.2
NFR-MS1.3	Test-NFR-MS1.3
NFR-MS2.1	Test-NFR-MS2.1
NFR-MS3.1	Test-NFR-MS3.1
NFR-MS3.2	Test-NFR-MS3.2
NFR-SR1.1	Test-NFR-SR1.1
NFR-SR1.2	Test-NFR-SR1.2
NFR-SR2.1	Test-NFR-SR2.1
NFR-SR2.2	Test-NFR-SR2.2
NFR-SR2.3	Test-NFR-SR2.3
NFR-SR2.4	Test-NFR-SR2.4
NFR-SR2.5	Test-NFR-SR2.5
NFR-SR3.1	Test-NFR-SR3.1
NFR-SR3.2	Test-NFR-SR3.2
NFR-CP1.1	Test-NFR-CP1.1
NFR-LR2.1	Test-NFR-LR2.1

Table 3: Traceability between Non-Functional Requirements and Tests Part 2

16 Code Coverage Metrics

Code coverage does not accommodate our project since we did not use automated testing.

17 Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:
In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

1. Bowen Zhang: In general, there can be differences between the VnV plan and the actual activities conducted for VnV due to the testing process of the Unity project. In the team's VnV plan, it is assumed that there will be unit testing and automated testing after completing MG and MIS. However, the team found that each script has been automatically tested by Unity during implementation. When a script is finished, Unity compiles it directly to see if there are any bugs. In addition, the tester needs to run the Unity project to use the script. As a result, unit testing has been combined with system testing. The team shall not use automated testing, everything could be tested manually in Unity, otherwise, it is just a duplication of work. In this way, the team decides to modify the VnV plan. The unit testing section and automated testing section are deleted. The team adds more details in the tests for functional and non-functional requirements to make them more comprehensive. In future projects, this problem can be anticipated by designing a flexible VnV plan that can accommodate Unity workflow. The team should regularly review and update the plan to reflect any changes that happened in Unity. In this way, the team can solve the difference between the VnV plan and the actual activities conducted for VnV and ensure project success.

2. Yichen Jiang: The major difference between the VnV plan and the activities that were actually conducted for VnV is brought by the change on the design. The VnV report was completed months ago, and within the time till now, our supervisor clarified more ideas that he would love to have in our product. There are many more features now in our product, so the verification and validation activities were adapted for the changes. Besides, the workflow is modified as well. We completed the VnV plan before we worked on MIS and MG, of course also before the real implementation. Our plan used to be to do the testing after all the implementation work's done, but when we implement it, we found that the testing is actually everywhere and anytime. We have to compile the scripts in Unity and check if everything's right all the time, and that is actually how our testing works. This method also influences the unit testing part of our testing plan. Any test has to be done with launching the whole system. Therefore you may notice that we put almost all efforts into system testing in the verification and validation report.

3. Junhong Chen: There are some differences between the VnV plan and the actual activities conducted for Vnv. In the VnV plan, we assumed that we can perform unit tests on a single function or a single module. However, in the actual test process, we can't perform unit tests for most of the functions because the implementations of the functions require launching the entire system. For example, it is hard to perform unit tests on the onClick() functions bound to some buttons which capture the click event from the mouse device and opens a new page just by simply running the code itself without launching the entire software. In addition,

after several meetings with Dr.Gonsamo and his lab members, a few new features were added to the product, which causes changes in some requirements, leading to the actual activities conducted for VnV differing from the VnV plan.

4. Jiacheng Wu: There are a few differences in the two V&V documents. The first one is that the V&V plan is more about designing the test case and V&V is about conducting the designed test cases. The second difference is the time. At the time we wrote the V&V plan, the software was native and we didn't have a clear view of our product but for the report, we have our product completed. Also, during our development, our requirements were changing constantly because we kept contacting with our clients. Due to the change in the requirements, we have to change the SRS as well as the test plan. Our anticipated changes are probably UI and seasonal changes to the model. During this process, I realized that we should constantly change our SRS and V&V documents every time we contact our clients because our requirements are changing constantly during communication.

5. Tingyu Shi: VnV Plan is different from the activities that we conducted for the VnV report. In the VnV plan, we documented that we will use automated testing. However, our product involves a lot of GUI and interactions between users and the software; therefore, manual testing is more feasible and efficient. So, we need to modify the automated testing part for the final document. For a future project, if it involves many GUI and user interactions, manual testing may be a better way to test.