

Evaluate Opportunities to Provide Training Simulation for ODOT Snow and Ice Drivers - Phase 2



Prepared by:

John E. Ash, Jiaqi Ma, Mehdi Norouzi, Ming Tang, and Xuefu Zhou

Prepared for:

The Ohio Department of Transportation,
Office of Statewide Planning & Research

Project ID # 107316

February 2022

Final Report



Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/OH-2022-06		
4. Title and Subtitle		5. Report Date
Evaluate Opportunities to Provide Training Simulation for ODOT Snow and Ice Drivers - Phase 2		February 2022
		6. Performing Organization Code
7. Author(s)		8. Performing Organization Report No.
John E. Ash, Jiaqi Ma, Mehdi Norouzi, Ming Tang, and Xuefu Zhou		
9. Performing Organization Name and Address		10. Work Unit No. (TRAVIS)
University of Cincinnati 765 Baldwin Hall, Cincinnati, Ohio 45221-0071		
		11. Contract or Grant No.
		32391
12. Sponsoring Agency Name and Address		13. Type of Report and Period Covered
Ohio Department of Transportation 1980 West Broad Street Columbus, Ohio 43223		Final Report
		14. Sponsoring Agency Code
15. Supplementary Notes		
N/A		
16. Abstract		
<p>Winter maintenance poses great concerns both during and after the snow and ice season. During winter, snowplow drivers must use equipment (i.e., plow trucks), each valuing up to \$200,000, sometimes in severe snowstorms and unfavorable traffic conditions. This poses great safety concerns to both the plow operators and the road users. The complexity of and dangers attached to plowing operations during winter makes it important to ensure plow drivers are equipped with adequate skills to navigate in their environment and be effective while on the job. Recently, the Ohio Department of Transportation (ODOT) noted a desire to enhance the current training offered to snowplow drivers by integrating driving simulation into the overall curriculum. As such, ODOT purchased a snowplow driving simulator to use for this purpose. In this project, the research team helped ODOT set up the simulator, developed and implemented custom training environments and scenarios, and developed training materials to be used in snow and ice driver education training programs by both drivers and instructors. Virtual environments were modeled directly after actual ODOT facilities in order to allow drivers to gain familiarity with the details of their future work environment. Training scenarios were created to allow for teaching and evaluation of specific skills under varying environments in the simulator (e.g., how to plow an intersection). It is the hope of this project that the use of the simulator and developed materials can help ODOT drivers gain more experience before going out onto the road, thus improving their efficiency, as well as safety for both themselves and the motoring public.</p>		
17. Keywords		18. Distribution Statement
Snowplow driver training, driving simulation, safety, winter maintenance		No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	82	

Form DOT F 1700.7 (8-72)

Reproduction of completed pages authorized

Credits and Acknowledgments

Prepared in cooperation with the Ohio Department of Transportation
and the U.S. Department of Transportation, Federal Highway
Administration

The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Ohio Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

The authors would like to convey their sincere appreciation and gratitude to all who provided guidance and support on this project. Notably, we would like to thank Mr. Joe Phillips, Mr. Shawn Flannery, Mr. Andrew Bolzenius, Mr. Steve Jacobcik, and Mr. Doug McLaughlin for their inputs, feedback, and time spent testing the numerous scenarios developed for the simulator. Their comments helped make the virtual environments as realistic as possible. Additionally, we appreciate the inputs Mr. Doug McLaughlin and Ms. Daveen Goodman provided on the guidebook and training materials that helped us align the materials with ODOT's broader training vision. Further, we would like to thank Ms. Kimberly Conklin of the Technical Advisory Committee for her valuable inputs on the project throughout its duration. Finally, we would like to extend our gratitude to Ms. Jacquelin Martindale for her role in managing the progress and keeping everything on track. Her comments throughout the project, particular during the project update meetings, were invaluable.

Table of Contents

1	Executive Summary	1
2	Problem Statement	2
3	Research Background.....	3
4	Research Approach	5
4.1	Snowplow Simulator Setup	5
4.2	Simulation Scenario Development	8
4.2.1	Development of Virtual Environments Modeled from ODOT Facilities	9
4.2.2	Development of Customized Ohio Snow Maps	19
4.2.3	Development of Simulation Scenarios for use in Training Exercises	49
4.3	Training Material Development and Evaluation.....	55
5	Findings and Conclusions	59
6	Recommendations for Implementation.....	61
7	Bibliography	63
8	Appendix.....	64
8.1	Summary Literature Review on Practices of Agencies using Snowplow Simulators.....	64
8.2	Attributes of Doron City	65
8.3	Source files	65
8.4	Maps of Six ODOT Road Databases	66
8.4.1	D3-A South. Royalton, River Road, Lorrain County.....	66
8.4.2	D3B: North. I -90.....	66
8.4.3	D10 : OH-664	67
8.4.4	D6: Columbus.....	67
8.5	DoronIG Project Source Code.....	72
8.6	C++ source code for parsing VDB Grid information	81
8.7	Debug Mode Keys for Simulator	82
8.8	VDB and IVE files for Doron City.....	82
8.9	VDB Cache for snow.vdb	82
8.10	Doron City Source Geometry for Snow.....	83
8.11	How to compile and build DoronIG and OpenVDBCompiler from source code	83

List of Figures

Figure 1. SP660 Snowplow Driving Simulator.....	6
Figure 2. Trailer for Housing Simulator Components	6
Figure 3. Force America Patrol Controller TM (MPJC-4).....	7
Figure 4. Control Room in Trailer with Instructor Workstation	8
Figure 5. General Workflow.....	10
Figure 6. Reference Images.....	11
Figure 7. Diagram Drawings of Road	11
Figure 8. Diagram Drawings of Intersection	12
Figure 9. Comparison of Aerial Photo with Final 3D Model	12
Figure 10. Comparison of Aerial Photo with Final 3D Model of Intersection.....	13
Figure 11. Examples of Interactions in the Final Model.....	13
Figure 12. Highway Features	14
Figure 13. Natural Landscape at D10.....	14
Figure 14. Buildings, Signs, and Poles.....	15
Figure 15. Trees Modeled as Billboard Objects	15
Figure 16. Bridge Modeling	16
Figure 17. Workflow for Exporting both Visual and Physics ive Files	16
Figure 18. Visual Model including Material and All Landscape Features	17
Figure 19. Barrier as a Collision Object.....	17
Figure 20. Optimization Process in 3dsMax to Reduce Number of Vertices	18
Figure 21. If Vehicle Deviates from Training Route, Collisions may not be Detected....	18
Figure 22. File Integration	19
Figure 23. Instructor Workstation (IWS)	20
Figure 24. System Network Overview	21
Figure 25. 3D Geometry Preprocessing and Export Pipeline Flowchart.....	23
Figure 26. STAGE Actors and their Logical Relationship to the 3D Models	24
Figure 27. A Section of District 6c's Terrain Visualized in ObjectViewer	26
Figure 28. Same Section of District 6c's LevelCompiled Terrain Visualized in ObjectViewer.....	26
Figure 29. Section of Physical Terrain IVE for District 6C Visualized in ObjectViewer..	27
Figure 30. Contents of dtPhysics Folder Generated for District 6C using PhysicsCompiler	28
Figure 31. City_Road.ive Visualized in ObjectViewer (Top View). (Obtained after Physics OSGExport).....	30
Figure 32. VDB Visualization in Blender Software for City_Road.vdb	30
Figure 33. Creating New Map for Simulator	33
Figure 34. Terrain Actor	33
Figure 35. Terrain Parameters (I).....	34
Figure 36. Terrain Parameters (II).....	35
Figure 37. Player Start Actor	35
Figure 38. Player Start Actor Parameters.....	35
Figure 39. TBBInitComponent Actor.....	36
Figure 40. TRBInitComponent Actor Parameters.....	36
Figure 41. PhysicsMaterial Actor.....	36

Figure 42. PhysicsMaterial Actor Parameters	36
Figure 43. Voxel Actor for Snow VDB	37
Figure 44. How to Edit and Add Actor Components	37
Figure 45. Voxel Actor Parameters (I)	38
Figure 46. Voxel Actor Parameters (II)	39
Figure 47. File Contents of .SCN File	40
Figure 48. Snow.vdb's Grid Parameters (this is the VDB for pre-installed Doron Map).	42
Figure 49. Debug Mode Example (the wireframes represent the physics meshes)	47
Figure 50. Doron City Map	52
Figure 51. Routes of Snowplow Scenarios Represented on the Doron City Map	57
Figure 52. Lorrain County South. D3-A (20 miles)	66
Figure 53. Lorrain county North. D3-B (10 miles)	66
Figure 54. Hocking County (10 miles)	67
Figure 55. From south (left) to north (right), D6A, D6B, D6C	67
Figure 56. Reference image at Google D6A	68
Figure 57. Screenshot of the site D6A from 3D modeling program	68
Figure 58. Reference image at Google D6B	69
Figure 59. Screenshot of the site D6B from 3D modeling program	70
Figure 60. Reference image at Google D6C	71
Figure 61. Screenshot of the site D6C from 3D modeling program	71

List of Tables

Table 1. ODOT Facilities Modeled in Simulator	9
Table 2. OpenVDBCompiler Parameters	29
Table 3. Delta3D / DoronIG Actors Defined for a Snow-Enabled Map on DoronIG (as configured in STAGE)	31
Table 4. OpenVDBCompiler Parameters	41
Table 5. snow.vdb, VDB shipped on the system, City_Road_Edited_P.vdb is a recompilation of Doron's road geometry created by the research team (Appendix E contains links to the files)	43
Table 6. Voxel Actor Settings and Default Values in STAGE for Snow Map	44
Table 7. Scenarios Pre-installed in the DoronIG Simulator	49
Table 8. New Scenarios Created and Installed in DoronIG Simulator	53
Table 9. Training Scenarios Delivered for Project Organized as a Curriculum	58
Table 10. Popular Simulators with Snowplow Scenarios	64

1 Executive Summary

Maintenance is an incredibly important aspect of state DOTs' services. In areas with snowfall, winter maintenance is a critical issue for DOTs in terms of both its cost and potential to improve safety. Plow trucks needed for snow and ice removal operations are typically quite expensive; couple that with the fact that they are often driven under adverse weather and traffic conditions, and one can see the potential for safety concerns. As such, training of snow and ice drivers is critical to ensure they are prepared to handle the varied conditions they will face on their routes. Typically, training programs have focused on classroom sessions and on-the-road lessons in the plow truck. In wanting to enhance their training curriculum, ODOT funded a project to examine the potential benefits of snowplow driving simulators as a tool for inclusion in their program. In this project, also conducted by the research team, the team surveyed numerous options available from different vendors and conducted driver behavior experiments with existing plow drivers. Upon completion of this phase of the project, ODOT purchased the Doron SP660 snowplow simulator. This simulator is similar to typical driving simulators in the sense that it enables participants to drive through dynamic virtual environments using actual vehicle controls (e.g., steering wheel, pedals, etc.), but different in the sense that it is larger in size like an actual plow truck, and it also includes controls to operate and allow simulation of specific snowplow truck elements (e.g., plow blade, wing blade, etc.). Essentially the simulator consists of a partial truck cabin with driver controls located inside a trailer so it can be transferred to different locations. The cab is surrounded by four screens enabling a 240° field of view of the environment outside the vehicle. Additionally, the trailer houses the computers needed to run the simulator hardware and software, and a workstation with a PC to allow an instructor to set up training scenarios and observe the driver.

In this project, the research team worked to get the aforementioned simulator up and running for ODOT, as well as develop scenarios for use in the simulator and supporting training materials. To begin, the team setup and tested all hardware and software components of the simulator and found things to work as expected, with the exception of the motion base on occasion. The motion base is a platform supported by hydraulic cylinders on which the cab rests, and it moves to simulate acceleration, the dropping of the plow blade, etc. In some cases, the base did not move, but this issue could usually be resolved with a system restart. In order to ensure smooth operations for ODOT, the team has documented startup and usage procedures and demonstrated them to ODOT training staff.

Following the simulator setup and verification, the team worked to develop simulation scenarios for use in the training program. This scenario development phase had two components: development of virtual environments (maps) based on ODOT facilities and development of training exercises in which specific skills can be practiced in a virtual environment. With regard to the former, the team used 3D modeling software to create 70 miles of ODOT routes that can be driven in the simulator based upon maps and photos of the real-world facilities in ODOT Districts 3, 6, and 10. These maps were created in order to allow drivers to gain familiarity with driving a plow in their specific work locales, and each was verified to work in the simulator. The team also got feedback from drivers in each district as to the realism of the routes and adjusted them to enhance their quality.

The second component of the scenario development phase involved development of exercises in which skills could be tested/demonstrated. Each exercise essentially consisted of an objective to be practiced in a virtual environment (e.g., plow an intersection). For this part, the team first investigated and documented the built-in training scenarios with the simulator. After consultation with ODOT training staff, it was determined these scenarios were of value for use in the final training program. Additionally, the team developed custom scenarios to enable teaching of different skills not addressed with the built-in scenarios. In the end, the scenarios covered a range of topics from simple vehicle operations to complex plowing maneuvers. Originally, the team also intended to deliver custom scenarios that took place in the ODOT-specific virtual environments. While driving in such environments is possible, implementation of plowable snow in them was not. This was a non-trivial issue related to physics models in simulator the software, among other things, and it could not be resolved even after thorough discussion with the vendor.

For the final component of this project, the team developed training manuals in the form of guidebooks to be used during a training program by both drivers (trainees) and instructors. These materials were prepared based on discussion with ODOT training staff, and existing ODOT training materials, and covered exercises to take place based upon the developed scenarios within the simulator. Ultimately, it is believed that this project will lead to safer and more efficient plowing operations in the state of Ohio, as drivers will be able to take part in more thorough and more realistic training prior to getting behind the wheel of an actual plow truck.

2 Problem Statement

Winter maintenance poses great concerns both during and after the snow and ice season. During winter, snowplow drivers must use equipment (i.e., plow trucks), each valuing up to \$200,000, sometimes in severe snowstorms and unfavorable traffic conditions. This poses great safety concerns to both the plow operators and the road users. Cases where plow drivers do not react accurately and swiftly to stimuli during their operation could result in crashes and other severe consequences. The complexity of and dangers attached to plowing operations during winter makes it important to ensure plow drivers are equipped with adequate skills to navigate in their environment and be effective while on the job. Recently, the Ohio Department of Transportation (ODOT) noted a desire to enhance the current training offered to snowplow drivers by integrating driving simulation into the overall curriculum. Though driving simulation is often used as a means for conducting human factors and driver behavior-related experiments, driving simulators have been and are still used in applications for training drivers of vehicles such as tractor trailers, transit buses, fire trucks, etc. The idea behind this addition to the training curriculum was to use simulation to let drivers experience the diverse driving conditions they would encounter on the road, in a safe and controlled environment, before taking an actual plow truck out onto the road. While simulation has its benefits, primarily with regard to safety/control, it is still not a substitute for on the road training. As such, this simulation component is to be used to supplement and enhance the current training which involves both verbal instruction and time on the road.

The overall goal of the project is to enable plow drivers to gain experience in a safe and controlled environment (i.e., the snowplow driving simulator) before they plow their first snow event, as well as to improve safety for both the drivers and the motoring public. This will be accomplished by using a commercially available snowplow driving simulator and developing supporting scenarios and curriculum to be used in training applications for ODOT's plow drivers.

3 Research Background

As indicated by the title, this is the second phase of this project. In Phase 1 of the project, the research team conducted an extensive analysis of the current practices of different Departments of Transportation (DOTs), including ODOT's current process for training snow and ice drivers. Then a driver behavior study was conducted which involved data collection from snowplow drivers during operation, and also through the administering of a driver behavior questionnaire. This was done to determine the most important aspects of snowplow operation that drivers felt were critical to include in the training curriculum with the simulator.

Following this driver behavior study, the research team identified different state-of-the-art driving simulators on the market that could address the findings from the data collected from drivers. Once they had data on various options, the team conducted a benefit-cost analysis to find the most cost-effective option, and the recommendation was provided to ODOT research and planning staff with suggestions as to how to enhance snowplow operator training with a driving simulator structure to enable more effective plow operations during winter.

As a final component of the Phase I study, the research team identified some factors that were believed to have potential to reduce the effectiveness of the training simulation. These factors, outlined as follows, were addressed as part of/motivated the Phase 2 portion of this project.

1. The ability to reproduce the same types of plow trucks being used by ODOT using the driving simulator is a major factor in determining training realism. The simulator should be able to model all components of the ODOT snowplows such as a wing, underbelly or front plow as required. A single axle or tandem axle should also be easily reproducible. Various hydraulic systems (penguin, force, etc.) and different types of materials being used have a direct impact on snowplow operation and would definitely have a direct impact on driver training. These factors have to be accounted for to have a realistic training simulation.
2. Scenario (i.e., a virtual environment used in a simulation exercise) development issues have been pointed out in Phase 1 of this project. From the research team's recommendations, ODOT-specific routes need to be developed to enable drivers to become familiar with the actual ODOT roadways they may plow. With this, drivers can have a feel of the amount of traffic on Ohio roadways, the usual type of weather at different times of the day, the effect of terrain, roadway conditions, etc. This has a direct effect on the safety and effectiveness of plowing operations. In essence, it also reduces the potential cost of equipment damage and crashes.

This is a major gap that needs to be filled, the simulation exercise needs to capture this important aspect.

3. There are different ways to evaluate driver behavior in a driving simulator study. From the result of Phase 1 of the project, no advanced technology has been utilized to collect driver data which could further reveal the lags of a driving simulation training exercise. With advanced data collection technology, the training simulation could be further improved.
4. For a stable training program, a guidebook needs to be developed which takes the simulation aspect into account. An effective platform for training the snow drivers will inform following an actual curriculum based on observations and driver evaluation.

In order to address the important issues uncovered in Phase 1 about making the simulation environment as effective as possible, the research team proposed the research team proposed the following tasks for this phase of the project.

- Simulator Set up
 - The simulator setup involves the configuration and calibration of the hardware and software components of the simulator as obtained from the vendor, Doron Precision Systems, Inc.
 - The research team will work with the vendor on deploying the driving simulator on the UC campus for configuration, development, and testing purposes.
 - Upon project completion, the simulator will be transferred to a location selected by ODOT (e.g., a garage in District 03). This transfer of equipment will include both the driving simulator itself and the trailer compartment.
- Simulation scenario development
 - In order to enable trainees to become familiar with the roadway facilities in the districts in which they work, the research team will develop specific simulation scenarios that model roadways in ODOT's jurisdiction.
 - The research team will develop models of ODOT specific routes in the simulation environment. Routes will be drawn from District 3 (Lorain County), District 6 (Columbus), and District 10 (Hocking Hills).
 - A brief review of practices from other agencies that use snowplow simulators was conducted, and the results are shown in the appendix.
- Driver behavior evaluation
 - The research team will use real-time sensing and monitoring systems in order to evaluate driver behavior in the simulator as a means to guide development of the training materials. Monitoring the drivers during the following winter season will also be a consideration.
- Guidebook development
 - For this component of the project, the research team will develop a training component on operations in the driving simulator. This content will be split into two different manuals, one for trainees/drivers and the other for instructors.
 - A key part of the training will focus on a series of exercises within the simulator (e.g., plowing at an intersection), each of which has some specific

objective to teach and evaluate by the instructor and to practice by the driver.

- These elements will be based on ODOT's current snow plow driver training materials, discussion with plow drivers and the ODOT project team, as well as results of the Phase I study. Some of the key elements covered in the training manuals and training scenarios in the simulator will be vehicle operation, speed management, space management, crew communication, and plowing effectiveness.

As aforementioned, the overall goal of this project is to enhance training available for plow drivers with the use of a driving simulator component in their training exercises, prior to them plowing an actual road. It is believed that this additional experience will help drivers improve their skills, as well as improve safety for both themselves and the motoring public. Specific objectives of this project are as follows:

- Increase safety (for both plow drivers & motoring public) ;
- Increase "time behind the wheel" for plow drivers via simulated training for a variety of conditions;
- Increase driver awareness of winter weather conditions;
- Increase driver awareness of surrounding traffic and objects in their environment which may pose hazards;
- Reduce time needed for roadway condition recovery via more efficient plowing operations;
- Enhance training already provided by seasoned employees; and
- Reduce the amount of time needed from full-time personnel to train seasonal employees.

4 Research Approach

The following section outlines the major tasks in the project and the research/work conducted to carry out each one. The section begins with a brief discussion of the simulator set up, followed by a discussion of the simulator scenario development. Next, Finally, a discussion of the training/evaluation materials is presented.

4.1 Snowplow Simulator Setup

The snowplow simulator selected for use by ODOT from Phase I of the project was the SP660 driving simulator from Doron Precision Systems, Inc. (see Figure 1). The simulator and supporting components were delivered to the UC campus and all were housed in a trailer (similar to that shown in Figure 2) with hook-ups for electricity, Internet, etc. Since the simulator components are all stored in a trailer, the idea is that ODOT training staff can move the simulator around to different sites/training sessions across the state with relative ease. The key components of the simulator that were set up are as follows:

- The driving simulator cab;
- A TrueSteer™ digital servo steering system;
- Four (4) 55" LED flat screen displays (240° field-of-view);

- One (1) instructor console;
- Dispatch radio system;
- Remote controlled side-view mirror adjustment;
- Pre-recorded audio scenario description feature;
- Snowplow scenario package/Truck/CDL training scenario package;
- Self-contained – air ride adjustable seat;
- Second seat located within the cab for the instructor;
- Auto/Manual transmissions; and
- Surround sound Dolby stereo system.



Figure 1. SP660 Snowplow Driving Simulator



Figure 2. Trailer for Housing Simulator Components

The simulator itself is comprised of three major components: the Visual System, the Seat and Motion System, and the Transmission and Controls. The Visual System is comprised of with Four (4) 55" LED flat screen displays allowing for a 240° field-of-view and remote-controlled side view mirrors. Two (2) of the monitors (those in the middle) represent the windscreen, while the side monitors will have the side view mirrors and will capture the usual range of vision similar to a snowplow truck. The monitors were set up and calibrated in an effort to make the driving experience as realistic as possible in terms of field of view, continuity of field of view, etc. The Seat and Motion System is itself comprised of two components: two seats with seatbelts for a driver and passenger (instructor) and the motion base. The motion base is comprised of various hydraulic cylinders used to simulate vibrations in the cab (such as those due to dropping the plow) and enhance the realism of the driving experience. As with most driving simulators, especially those with moving components, there is a possibility of users getting temporary vertigo (also known as simulation sickness); this usually passes in several minutes and is not known to have any long-term effects. Finally, the Transmission and Controls include options for both manual and automatic transmission in the simulator, as well as the joystick controls to operate various components of the vehicle. Currently, ODOT uses the Force America Patrol Controller™ MPJC-4 controls (shown in Figure 3). This system consists of a four-joystick controller that controls the dump box, the forward plow, the right-wing plow, and the left-wing plow, as well as the lights. While the simulator did not have the exact same controller as used by ODOT plow drivers, the controller it did have was quite similar in terms of setup (i.e., four joysticks), location, and usage.

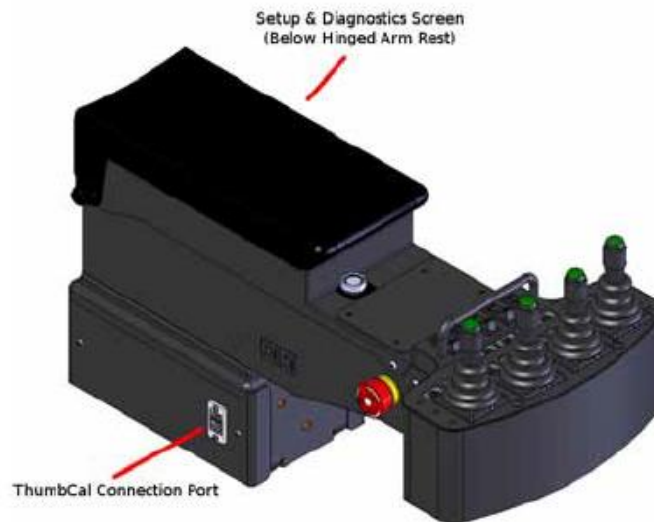


Figure 3. Force America Patrol Controller™ (MPJC-4)

A final important component of the simulator system is the control room (see Figure 4). Essentially, the control room is all of the space behind the simulator cab in the trailer. Inside the control room are many items including a desk with a PC and chairs (i.e., the

instructor workstation) to enable the instructor to operate the simulator and change between scenarios for the trainee. Additionally, the desk has a microphone attachment that can be used to communicate with the driver in the simulator via the vehicle radio. Besides these components, the control room also has an air-cooled cabinet to house the four PCs that are used to control the driving simulator itself; each of the PCs is connected to one of the simulator's four screens to provide inputs. Other than these components, the control room has a bench to allow a few people to sit and observe, enough space for a few additional people to stand, and some additional storage space.



Figure 4. Control Room in Trailer with Instructor Workstation

For this portion of the project, the research team worked with Doron staff in order to set up the driving simulator on the UC campus for the duration of this project. The research team got the system up and running, verified that all of the aforementioned components of the simulator and trailer setup work properly, and has documented procedures on how to use/interact with components in the simulator (e.g., how might one turn the screens on and start a training scenario in the simulator?). Additionally, the team demonstrated these procedures to ODOT training staff. It is important to note that in some cases, the research team did observe the motion base failing to work; in such cases, a restart of the system usually solved the problem.

4.2 Simulation Scenario Development

This section details the process of developing simulation scenarios to be used by training staff in their lessons. Here, a scenario is defined as a specific exercise and corresponding virtual environment in the simulator to be driven through by a trainee as part of the learning process. As such, the scenario development process involved two main parts. The first part of this task was the development of the virtual environments using 3D modeling software. As aforementioned, virtual environments were created and modeled after actual ODOT facilities were created in order to allow drivers to become familiar with the roads they will plow in a safe and controlled environment prior to getting on the road.

It is also important to note that the vendor provided a series of scenarios with the simulator itself. These scenarios take place in the virtual environment of Doron City, a fictional locale with urban and rural portions, roadways of varying functional classification (locals to arterials), and numerous roadside/environmental objects from buildings to people (both drivers and pedestrians), among many more. The second portion of this task involved the development of a series of scenarios for use in training applications by ODOT. For this part, the research team built several scenarios that take place in the Doron City environment as well as in the other customized ODOT environments. With regard to the latter, the research team experienced numerous problems with implementing plowable snow in the custom virtual environments. This software issue was non-trivial and could not be resolved after thorough consultation from both Doron and former software developers at Doron (i.e., the vendor did not have a solution), as well as tremendous time spent debugging the snow rendering code.

4.2.1 Development of Virtual Environments Modeled from ODOT Facilities

The objective for this task was to create customized Ohio virtual environments (referred to interchangeably as maps) based on the routes from three different counties. Each map, also called “database” in the Doron system, will generate scenarios for various training purposes (see Table 1).

Table 1. ODOT Facilities Modeled in Simulator

Map/Database	Length (miles)	County	Road #	reference
D3-A	20	Lorain	South Route. 82, 83, 57	Google map
D3-B	10	Lorain	North Route. I-90	Google map
D10	10	Hocking	664	Google map
D6-A	10	Columbus	270	See attached google earth file
D6-B	10	Columbus	270	See attached google earth file
D6 -C	10	Columbus	270	See attached google earth file

4.2.1.1 Content in the Ohio maps

Similar to the default Doron City map, which the Doron provides as a part of the default database included with the snowplow simulator, these six Ohio maps contain 3D models of each area and include features such as terrain, roads, trees, buildings, traffic signs, unity poles, mailboxes, etc. The content is based on the actual site data collected through various open-source databases such as Geographic Information System (GIS) maps prepared by state and national agencies, USGS terrain data, Bing Maps aerial photos, Google Maps, Google Earth, and Google Street View; typically, the map data is stored in

a file that can be examined with Geographic Information Systems (GIS) software. Each of the Ohio maps was constructed using software including Autodesk Civil 3D, AutoCAD, Infracore, 3dsMax, Maya, and Rhino. In order to ensure the highest degree of realism, each map was examined/driven on in the simulator by ODOT workers from each respective district.

4.2.1.2 Workflow

The general workflow for this task (i.e., creating custom maps) starts with preparing 3D models of each site in the 3D modeling software Autodesk 3dsMax. Each site model includes 3D geometric data and material and textures. Upon preparing the 3D model of the site, it is exported to the compiler to produce the files in a suitable format that can be fed into the IG engine in the Doron Simulator (see Figure 5).

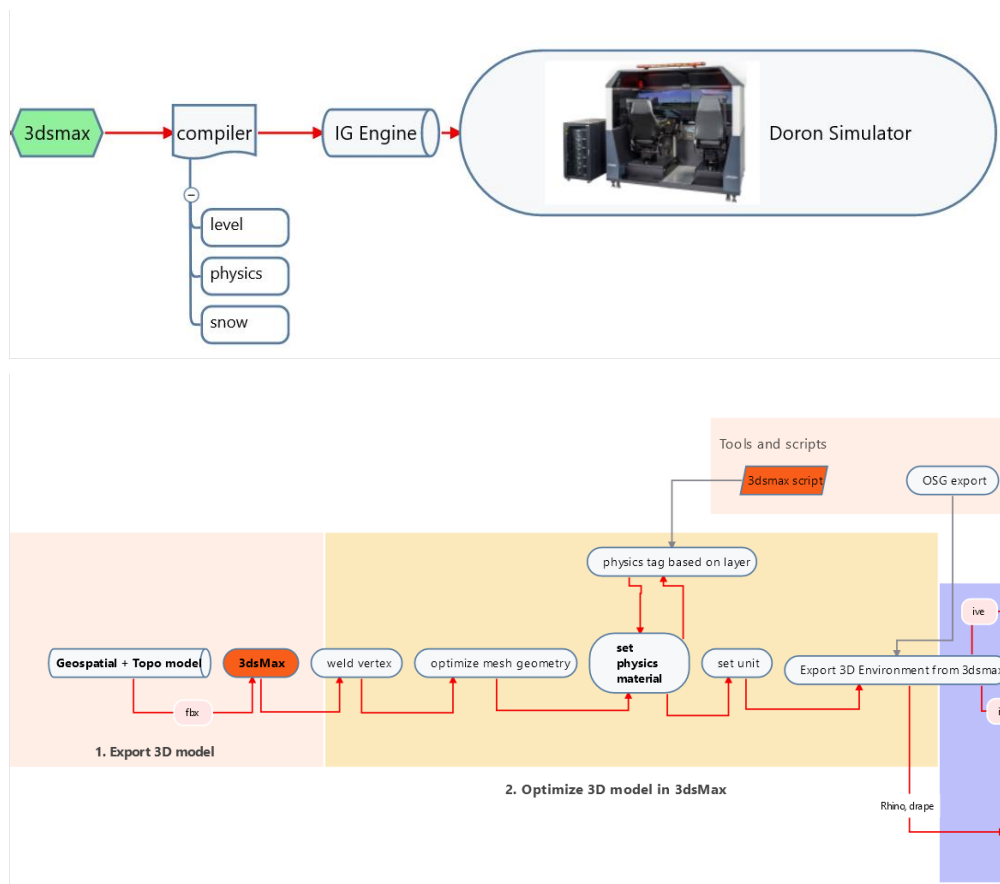


Figure 5. General Workflow

The following sections present a more specific guide to the steps in the workflow for creating custom 3D map files.

4.2.1.3 Modeling

4.2.1.3.1 Roadway Geometry

Using reference images (Figure 6), the modeling team measured various geometric elements of each site including the lane width, curb/corner radii, and dimensions of each intersection; from these inputs, they created diagram drawings (Figure 7, Figure 8). Then, the team used modeling tools to manipulate points, edges, and faces of the 3D geometry to match the elements to those in the reference photos as closely as possible (Figure 9, Figure 10, Figure 11).



Figure 6. Reference Images



Figure 7. Diagram Drawings of Road



Figure 8. Diagram Drawings of Intersection

W. River Rd. and
Royalton Rd.



Figure 9. Comparison of Aerial Photo with Final 3D Model

Fixed Left-turn only lane of
E. River Rd. + Royalton Rd.



Figure 10. Comparison of Aerial Photo with Final 3D Model of Intersection

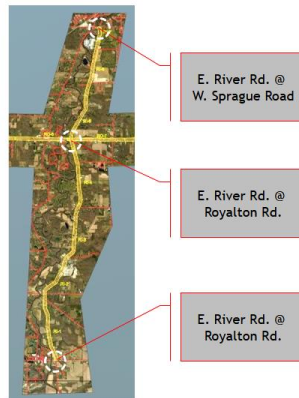


Figure 11. Examples of Interactions in the Final Model

4.2.1.3.2 Creation of Highways and Roadside Devices

Highways were constructed based on GIS data (e.g., shape files of the state's roadway inventory). For each highway, the roadbed itself served as a base on which to add the appropriate barriers, poles, signs, and other utility objects as viewed in the reference photos (Figure 12).

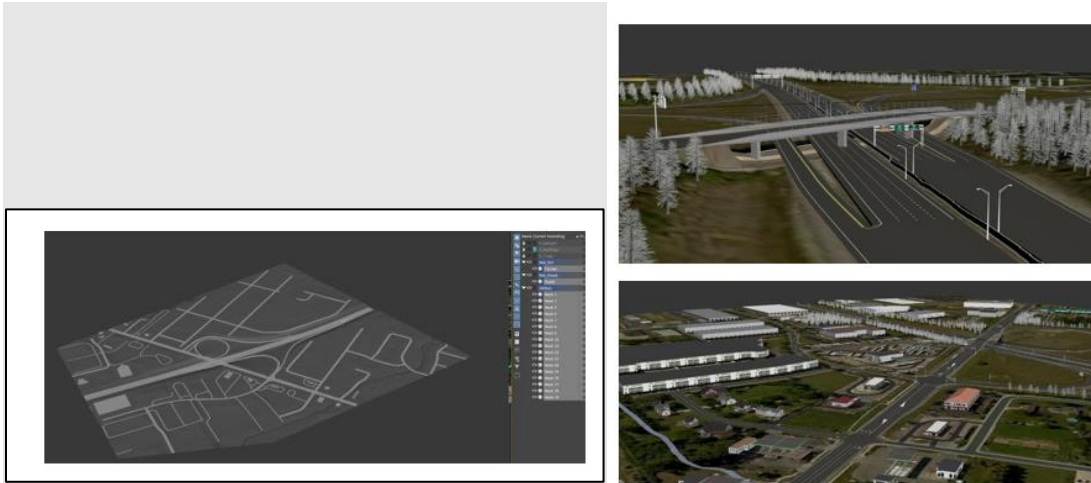


Figure 12. Highway Features

4.2.1.3.3 Terrain and landscape features

The modeling team then used the GIS and USGS databases to create the topographic ground model (i.e., the model of the surrounding environment), which included the basic terrain, bodies of water such as rivers, etc. (Figure 13).



Figure 13. Natural Landscape at D10

4.2.1.3.4 Buildings, Trees, signs, poles, mailbox

After modeling the terrain, other elements of the environment including buildings, trees, signs, utility poles, mailboxes, etc. were added to the 3D model. These 3D features were modeled manually, based on photo reference materials. Due to the volume of objects in each environment, these ancillary items were modeled as low polygon meshes or clipped billboards in order to make sure the final models can be loaded by and run smoothly (i.e., without lag) in the simulator (Figure 14 and Figure 15).



Figure 14. Buildings, Signs, and Poles



Figure 15. Trees Modeled as Billboard Objects

4.2.1.3.5 Bridges and tunnels

Additional features, including bridges and tunnels, were modeled using GIS data and manipulated manually in 3dsMax to match the dimension of their counterparts at the actual site (see Figure 16).



Figure 16. Bridge Modeling

4.2.1.4 Compiling

Upon completion of creating the 3D model of each site, the team exported the completed 3dsMax model into a format composed of two OSG IVE files. One of the files is used for visual rendering, and the other is required to enable physics simulation (as shown in Figure 17).

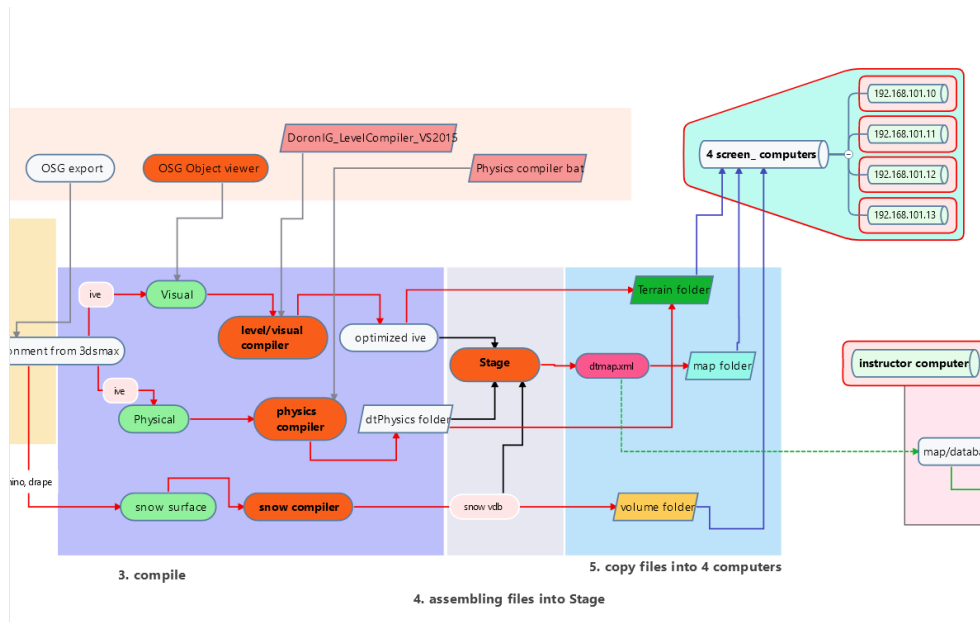


Figure 17. Workflow for Exporting both Visual and Physics ive Files

4.2.1.4.1 Visual model

More specifically, the visual model file described previously includes all the map features, with both environmental objects (e.g., vehicles, trees, buildings, etc.) and texture mapping (Figure 18).



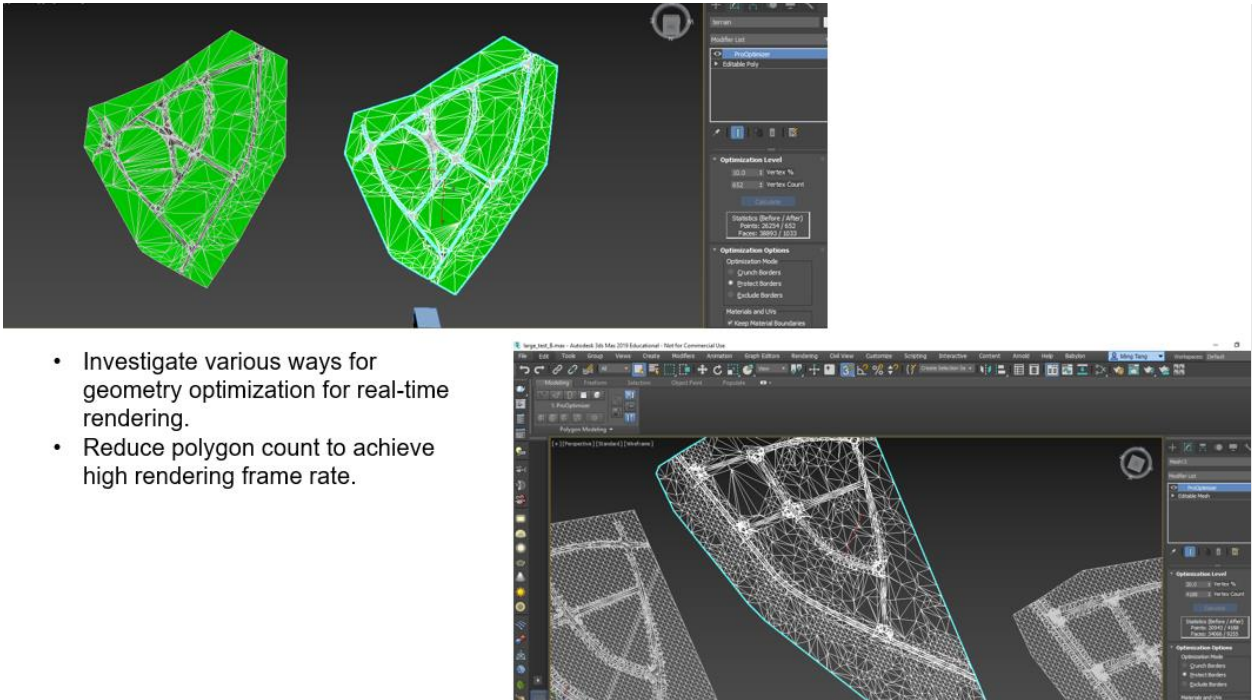
Figure 18. Visual Model including Material and All Landscape Features

4.2.1.4.2 Physics model

Unlike the visual model file, the physics model file only includes terrain, the roadway, and barrier geometry elements, each of which will interact with a vehicle to create simulated physics (Figure 19). Due to the limits of Doron IG engine, the physics model can only contain 0.5 million vertices for each map. As a result, the model must be optimized over multiple rounds to fit into the small vertex budget (Figure 20). In doing so, the team had to exclude some highly detailed models such as trees, poles, and mailboxes from the collision calculation. Ultimately, the team narrowed the focus area for the physics calculations to an area of approximately 100' on each side of the route. As a result, it is advised that the driver not stray too far from this route in order to ensure the highest degree of realism (Figure 21).



Figure 19. Barrier as a Collision Object



- Investigate various ways for geometry optimization for real-time rendering.
- Reduce polygon count to achieve high rendering frame rate.

Figure 20. Optimization Process in 3dsMax to Reduce Number of Vertices



Figure 21. If Vehicle Deviates from Training Route, Collisions may not be Detected

4.2.1.5 Integration

The compiled OSG files (i.e., the visual model and the physics model) are next assembled in the Stage program and fed into the Doron IG engine (for a more detailed discussion of this component, please see Figure 22. At this point, they can be used in the driving simulator (i.e., a participant can drive on one of the custom maps).

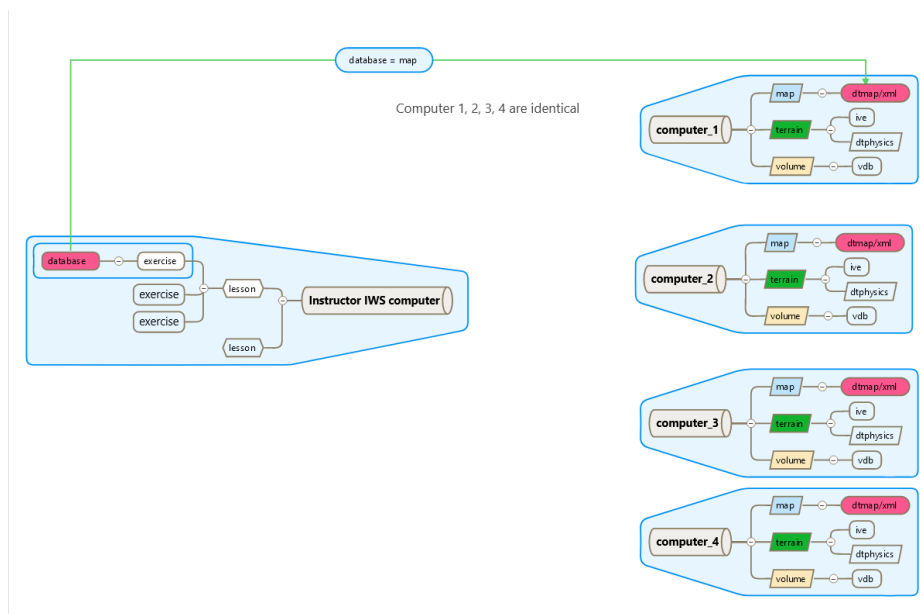
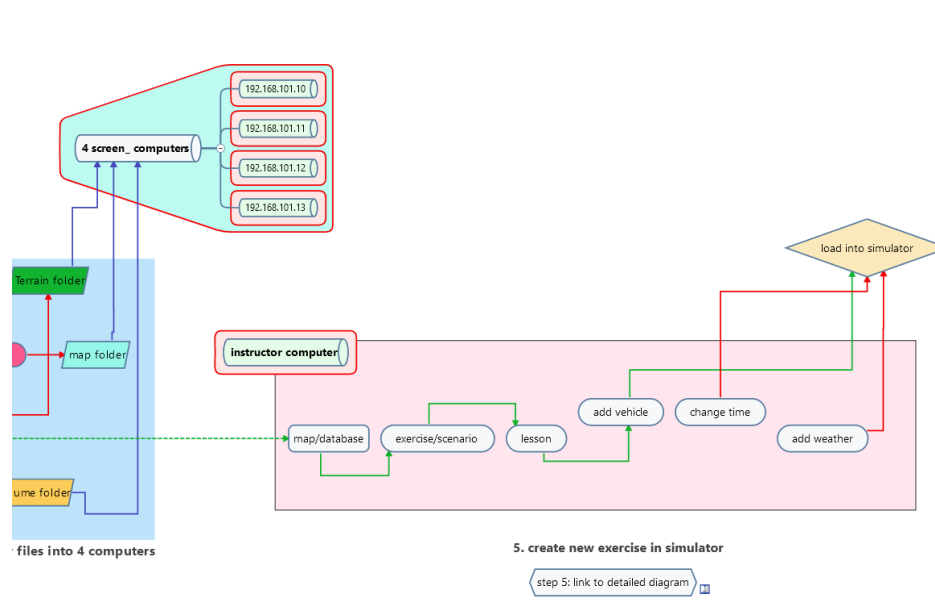


Figure 22. File Integration

4.2.2 Development of Customized Ohio Snow Maps

Following the general procedure described in Section 4.2.1, detailed procedures for the development of customized Ohio snow maps are described in the following.

4.2.2.1 DoronIG Simulator System Overview

The DoronIG snowplow simulator system (also called the Doron 660Snowplowplus) is run by a system of five PCs in a local area network. The instructor workstation (IWS, or Instructor Operating Station IOS, iss shown in Figure 23) has the IOS application installed on it, and this application is essentially a control panel for the simulation program running

on the four PCs connected to the four screens of the simulator cab (snowplow interior where driver and one passenger may sit as shown in Figure 1). The IWS provides a central area for instructors to work before, during, and after training sessions. The overview of the system in terms of how the various PCs are connected with each other is shown in **Error! Reference source not found..** Each of the four PCs runs an instance of the DoronIG Delta3D game engine (i.e., the program used for the driving simulator exercises) where the instances exchange Delta3D game messages over the network. The IOS sends control messages to the runtimes for common operations including starting/stopping the simulation, changing vehicles, changing the map, etc. The DoronIG project has a service which converts these IOS messages to the respective Delta3D messages (so they can be processed by the game engine).



Figure 23. Instructor Workstation (IWS)

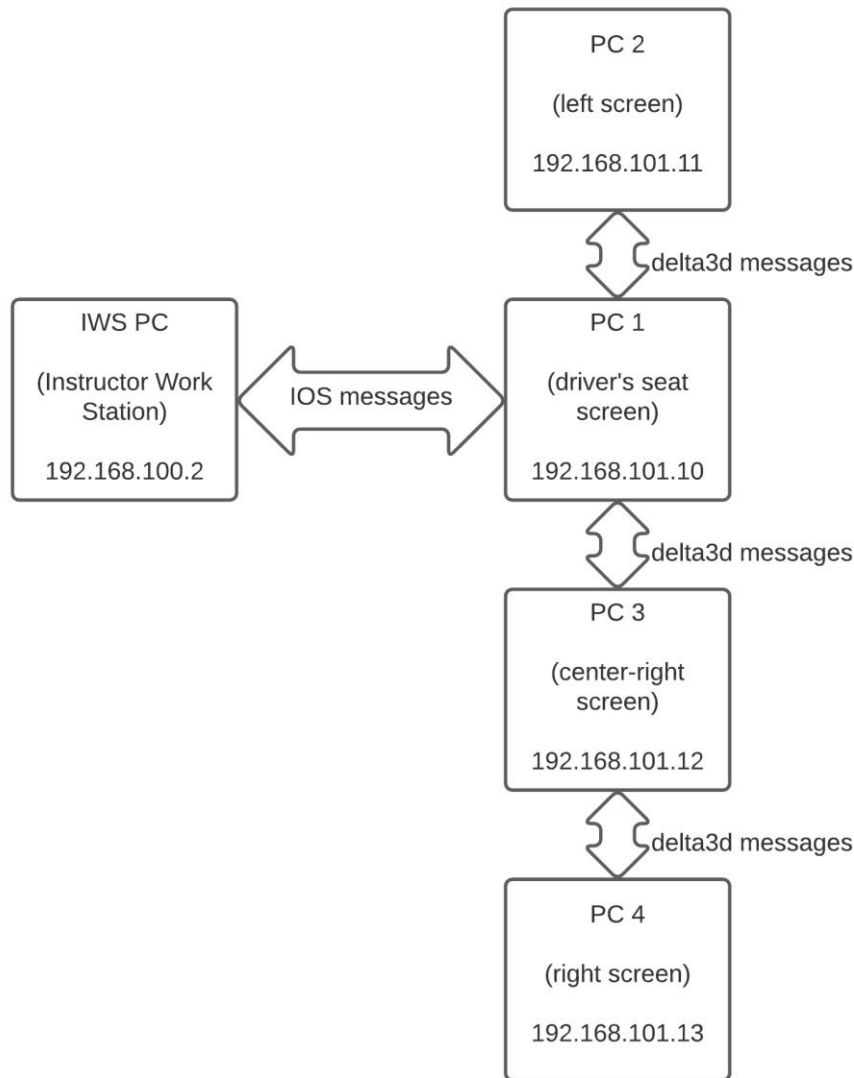


Figure 24. System Network Overview

On a software level, DoronIG is a virtual reality simulation based on the Delta3D game engine. Many tools the research team used to prepare maps for the simulator are a part of the Delta3D game engine's toolkit including the PhysicsCompiler, LevelCompiler, OpenVDBCompiler, STAGE editor, and ObjectViewer to name several of the most important ones. Details about the game engine are published in (Darken et al., 2005).

The source codes for the DoronIG project are included in Appendix 8.5 and organized as follows:

- /DoronIG
 - ↳ /DeltaEXT
 - ↳ /Project

↳ /Resources

On the top level, there is a folder named DoronIG. In this folder, there are three subfolders including DeltaEXT, Project and Resources, respectively.

The subfolder named DeltaEXT contains dependencies (i.e., dynamic-linked libraries) needed to run Delta3d game engine, the subfolder named Project contains Doron's source codes for the simulator, and the subfolder named Resources contains Delta3D engine and Simcore's source code. Software tools from Delta3D for such as the LevelCompiler, PhysicsCompiler, OpenVDBCompiler, ObjectViewer and STAGE etc. can be generated by following source code build process described in Appendix 8.11.

4.2.2.2 DoronIG Simulator Snow Map Implementation

The procedure to develop new maps in DoronIG with plowable snow on the road after a 3D model of the environment is developed (as discussed previously in Section 4.2.1 of this report) can be summarized as the following:

1. Preprocessing and Exporting the requisite geometries or 3D models.
2. Compilation and Optimization of the geometries.
3. Map creation in STAGE.
4. Deploying the Map and geometries to the Simulator.
5. Configuring a Scenario to bootstrap the Map into an Exercise.

The following sections detail these steps. Please refer to Figure 25 and Figure 26 to visualize the role of these steps in the entire procedure.

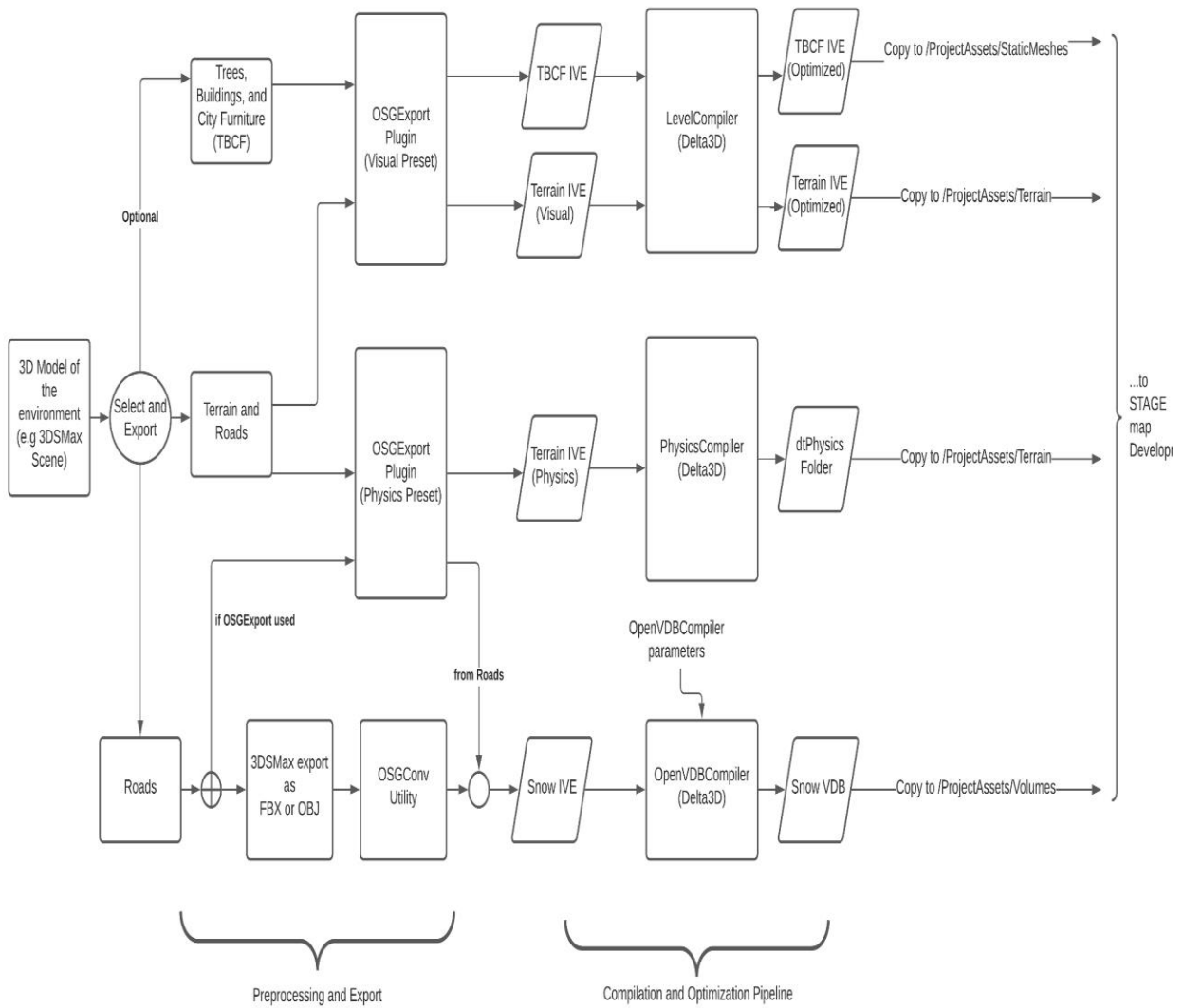


Figure 25. 3D Geometry Preprocessing and Export Pipeline Flowchart

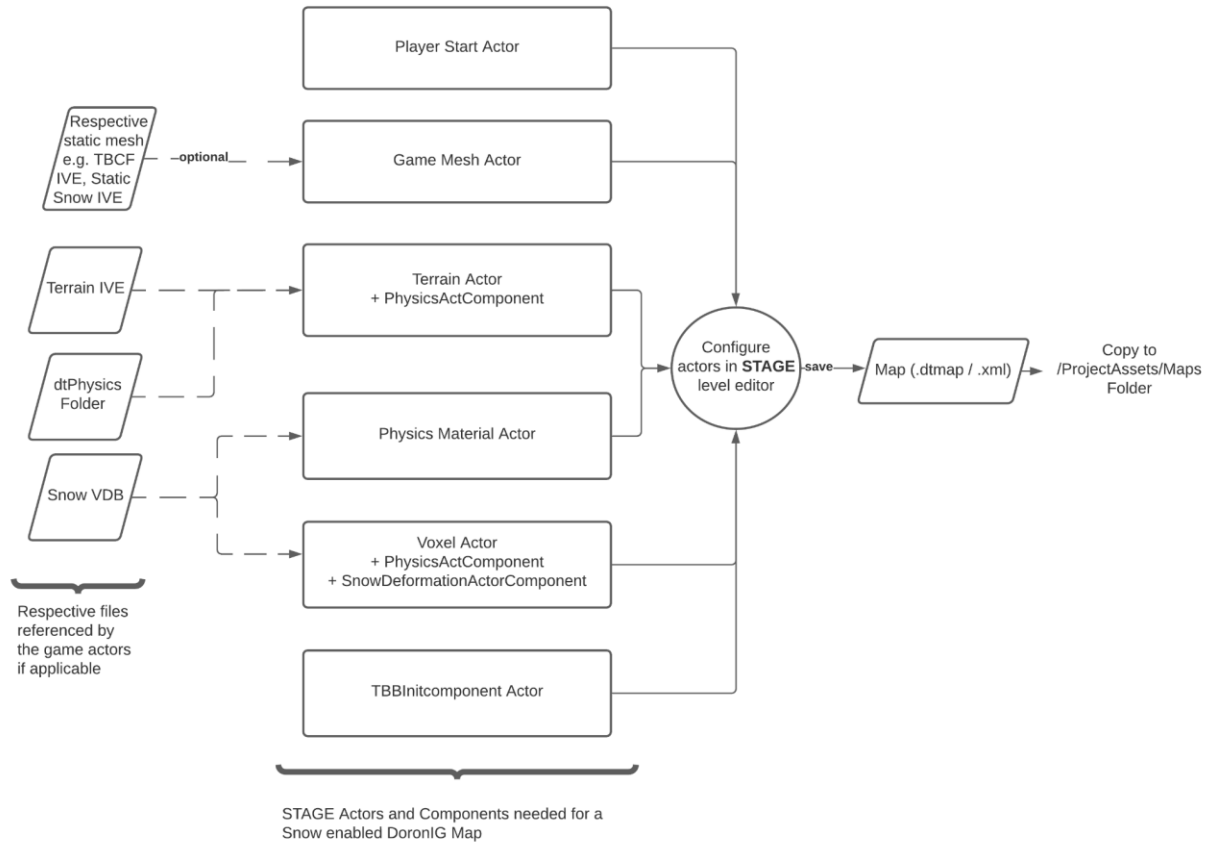


Figure 26. STAGE Actors and their Logical Relationship to the 3D Models

4.2.2.2.1 Step 1: Preprocessing and Exporting the requisite geometries or 3D models

The 3D snow environment for the simulator is designed using sophisticated 3D graphics software like Autodesk 3DSMax and InfraWorks which enables designers to be highly productive with the process. The designed 3D environment can include the terrain or ground of a site, roads, trees, buildings, and rest of the city furniture. However, the Delta3D engine for the simulator is built to be compatible with 3D file formats native to OpenSceneGraph (eg. .IVE, .OSG, .OSGB, .OSGT) for 3D models and VDB format for its voxel features. Hence, a suite of tools and plugins are used to obtain the respective compatible geometries before they are processed further for compilation, optimization and subsequently STAGE development. STAGE stands for Simulation, Training and Game Editor. It is a GUI tool to create and edit ‘Maps’ for the DoronIG simulations.

First, terrain and roads are selectively exported twice from the 3D scene by:

1. Using OSGExport (OpenSceneGraph Exporter) plugin with a ‘visual’ preset giving one a ‘Visual’ Terrain IVE; and
2. Using OSGExport plugin with a ‘Physics’ preset giving us our ‘physical’ Terrain IVE.

The visual terrain IVE represents is the virtual environment's ground and roads in the simulation, it has texture information present. The "physical" terrain IVE omits texture information from the model, but retains the wireframe and "material" information in the geometry. This is relevant for Delta3D's physics engine processing as discussed in the later sections.

Since, the plowable snow needs to be on the road of a site, the road geometry itself also represents the snow "volume." This is similarly exported using OSGExport (Physics preset). As an alternative approach, the road is exported natively from Max in a common format like FBX or OBJ (no texture, re-tessellated) and then converted to a compatible IVE using the OSGConv command line utility.

Optionally, when one does not need aspects of the 3D environment to have physics properties during the simulation (e.g., trees, city furniture, buildings, and static snow on terrain); runtime performance of the simulation can be optimized if one excludes them from terrain. Hence, one can selectively export these from the 3D scene using the visual OSGExport plugin to be used as a "Static" mesh actor in STAGE as discussed later.

4.2.2.2.2 Step 2: Compilation and Optimization of the geometries

Visual IVE Optimizations using the LevelCompiler

The visual IVEs generated in the preceding procedure are very detailed and can be optimized further to keep the simulation running at a performant frame rate. The LevelCompiler (also referred to as Visual Compiler) is a command line utility in Delta3D that takes an IVE and exports an optimized version of the file. It is specifically useful for large terrain databases as utilized in this project. It restructures the terrain model for faster display in the Delta3D runtime. This file generated is also smaller in size and reduces GPU bandwidth during the simulation. In DoronIG, it can be invoked as follows (with a terrain IVE present in the same directory)

```
>>> RunLevelCompiler.bat district_6C_terrain_visual.ive
```

The preceding example command takes a visual terrain IVE for district 6C and saves an optimized version named '_COMPILED.ive' in the same directory. In this particular example it is noted that the size of the terrain decreased from 89,291 KB to 48,683 KB, while retaining the visual fidelity of the terrain. One can preview how these IVE files look by opening them in Delta3D's 'ObjectViewer' utility (refer to Figure 27 and Figure 28).

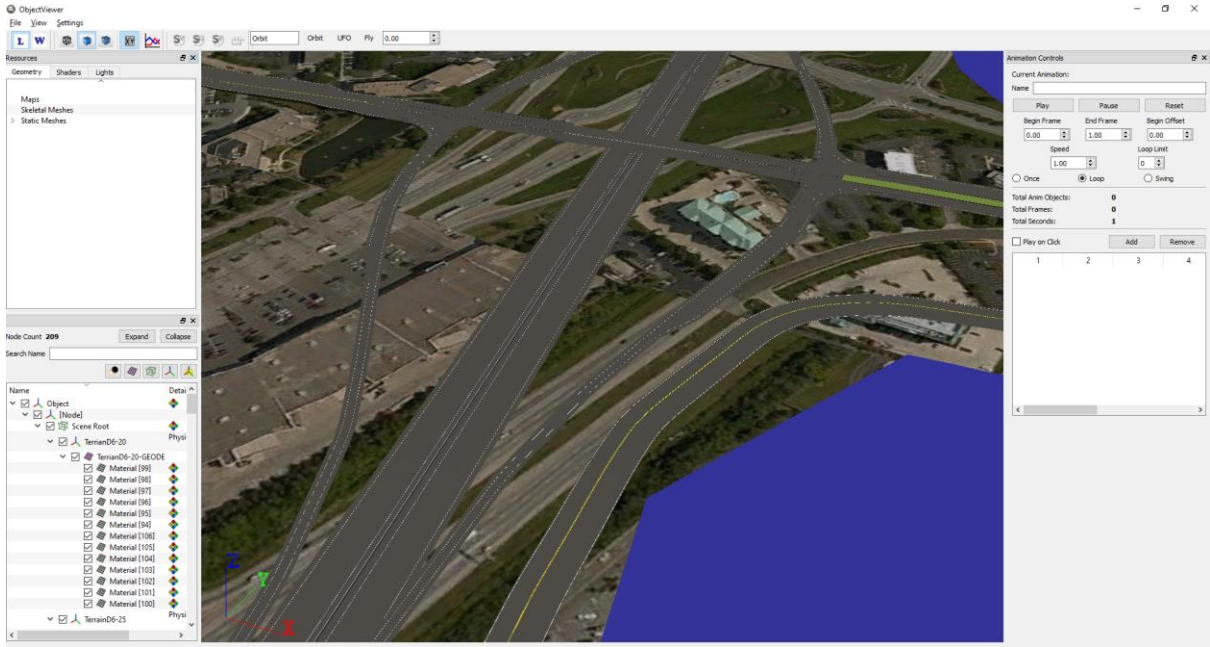


Figure 27. A Section of District 6c's Terrain Visualized in ObjectViewer

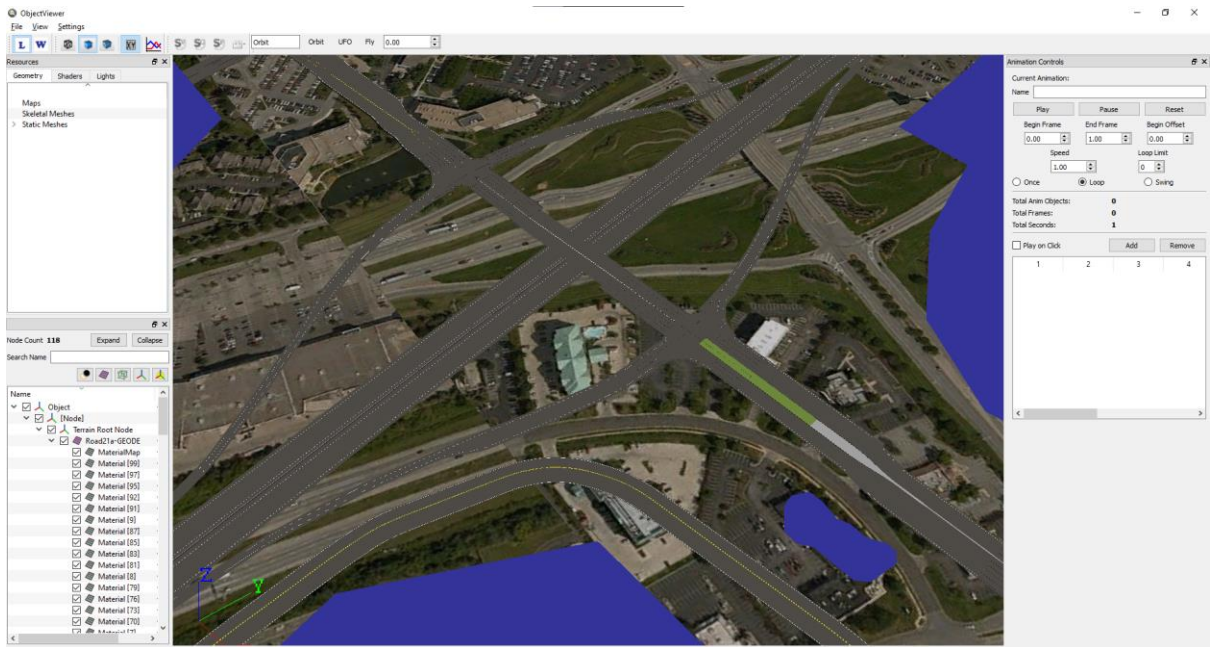


Figure 28. Same Section of District 6c's LevelCompiled Terrain Visualized in ObjectViewer

A similar process is repeated for any visual IVE's one might use in a DoronIG map for example for trees, buildings and City furniture or Static snow portions.

Physics mesh generation using PhysicsCompiler (dtPhysics folder)

Delta3D's terrains require a form of pre-processed physics meshes associated with them so that physics properties like collision, friction for different surfaces etc., can be configured. Note that the visual terrain IVE is not responsible for interacting with the physics engine of Delta3D and does not contain texture information (see Figure 29). For example, the research team wanted a vehicle's wheels to interact realistically to the terrain and behave respectively as it is driven on dirt, road or snow. These meshes are essentially a set of polygons with material tags embedded in them. This is stored in the /dtPhysics folder of the respective Terrain. One can use the physics IVE for the terrain (and roads) generated in the previous section and obtain the compatible dtPhysics folder. This command line utility in Delta3D is called PhysicsCompiler and can be used as follows:

```
>>> runPhysicsCompiler.bat district_6C_terrain_physical.ive
```

The command will create the respective dtPhysics directory for the physics terrain provided (Figure 30).

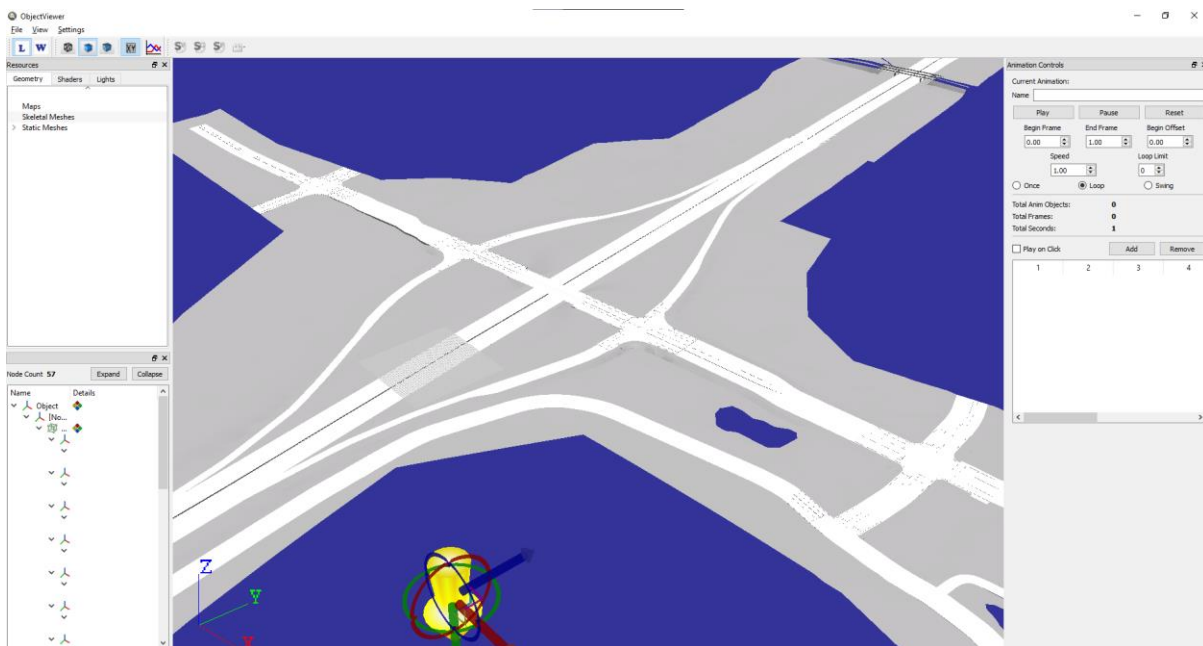


Figure 29. Section of Physical Terrain IVE for District 6C Visualized in ObjectViewer

dtPhysics_district_6C_terrain_physical	
Name	Size
DiffuseUV.dtpphys	1,143 KB
Mat_Dirt_split1.dtpphys	4,218 KB
Mat_Dirt_split2.dtpphys	2,926 KB
Mat_Dirt_split3.dtpphys	2,271 KB
Mat_Dirt_split4.dtpphys	1,875 KB
Mat_Road.dtpphys	48 KB
Mat_Road_split1.dtpphys	3,320 KB
Mat_Road_split2.dtpphys	2,472 KB
Mat_Road_split3.dtpphys	857 KB

Figure 30. Contents of dtPhysics Folder Generated for District 6C using PhysicsCompiler

Compiling Snow VDB using OpenVDBCompiler

OpenVDBCompiler is a command line utility bundled with the Delta3D engine responsible for converting IVE meshes to VDB volumes; exploration of this utility was crucial in the research team’s testing process. A 3D model of the snow shape in IVE format can be given as input to the OpenVDBCompiler, and a VDB for that file is hence generated. On a high level the compiler (1) takes an IVE, (2) extracts primitives (Triangles and Points) using Delta3D and OSG’s library methods, (3) sends them to an OpenVDB class method to calculate LevelSet, and finally (4) exports the levelset as a VDB grid to a VDB file; the compilation parameters available for the utility are listed in.

Table 2. OpenVDBCompiler Parameters

Parameter	Description
Bool Flag -b	If present, a boolean valued VDB grid is generated
Float Flag -f	If present, a float valued VDB grid is generated
Voxel Size	Voxel Size between 0 and 1. (for all x,y,z directions)
Voxel Size in Z	Override for Voxel Size in z direction
Thickness External	How thick to make mesh walls externally
Thickness Internal	How thick to make mesh walls internally
Subdivisions	Number of subdivisions to break the VDB into if needed, reduces memory usage

As an example the following command compiles City_Road.ive to City_Road.vdb.

```
>>> OpenVDBCompiler.exe -f -r 0.15 -z 0.05 City_Road.ive
```

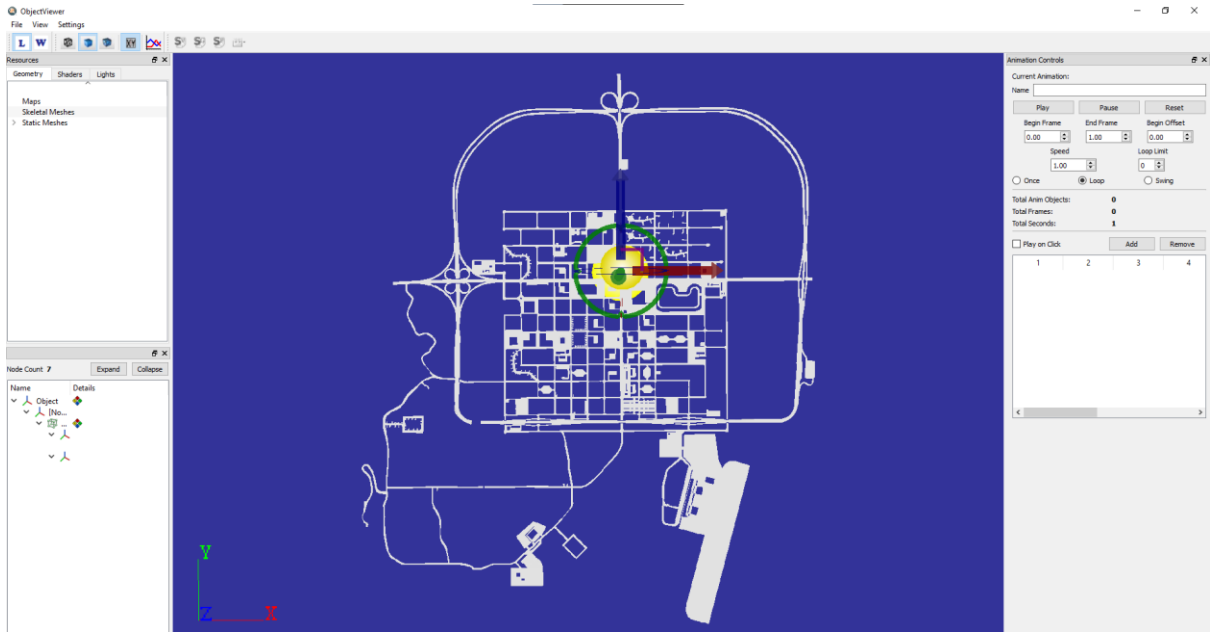


Figure 31. City_Road.ive Visualized in ObjectViewer (Top View). (Obtained after Physics OSGExport)

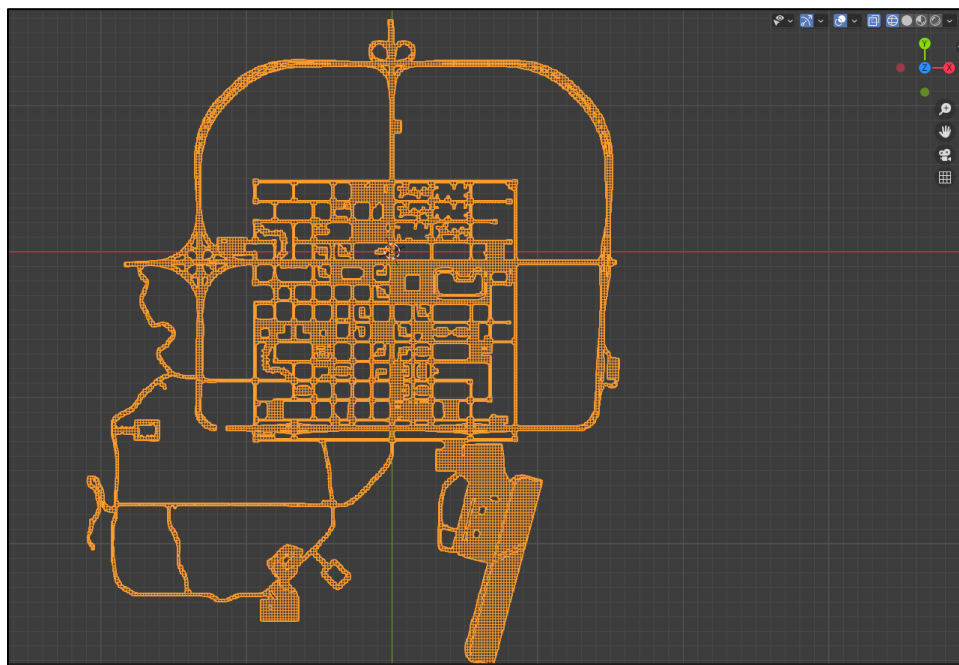


Figure 32. VDB Visualization in Blender Software for City_Road.vdb

Preparation for STAGE development

Once all the aforementioned 3D optimized geometries are ready, they need to be present in the correct directory inside DoronIG before we use them to make our 'Map' in STAGE.

The assets need to be transferred to the folders as listed in the following:-

- ProjectAssets/Terrains/ should have the visual (IVE) and physical terrains (dtPhysics folder)
- ProjectAssets/Volumes/ should have the VDB compiled.
- ProjectAssets/StaticMeshes/ should have any static IVEs if compiled (eg. TBCF, static snow)

4.2.2.2.3 Step 3: Map creation in STAGE

The output of this process is a .dtmap or .xml file which is stored in the /ProjectAssets/Maps directory (see Figure 26).

In the context of DoronIG maps, a map would have the following actors defined and configured as shown in Table 3:

Table 3. Delta3D / DoronIG Actors Defined for a Snow-Enabled Map on DoronIG (as configured in STAGE)

Actor	Delta3D Class / Category	Description
PlayerStart	dtcore	Defines where a player (driver) starts in a Map.
Snow Physics Material	dtPhysics	Defines physical properties of snow material, mainly its frictional properties.
Snow Database	dtVoxel	This Voxel Actor represents parameters for loading the snow VDB grids. PhysicsActComponent is added which sends collision messages to the runtime during plowing.

Static Snow Mesh	dtcore.Game.Actor	Static 3D mesh of static snow off the road. This is not plowable, present for cosmetic purposes. Refers to the Static snow IVE.
TBBInitComponent	GMComponent.DoronIG	Settings for configuring the number of processing threads the runtime should utilize.
Terrain	DVTETerrain	Terrain basically refers to the ground surface; material properties of road portions and roadside portions are defined. Refers to Terrain IVE and its dtPhysics directory.

See Appendix 8.10 for the complete listing of the STAGE configuration for the built in Doron City map.

STAGE Map creation walkthrough

1. This section describes the procedure for developing a "Map" for DoronIG in STAGE. In the guided example the research team recreated the virtual world, Doron City, as it is deployed on the simulator with dynamic snow. Open STAGE program.
2. Click on File -> New to create a new Map for the simulator. Fill in the name and click OK, See Figure 33 as follows.
3. Add Terrain
 - a. From the Actors panel on the left, create a Terrain Actor (Figure 34).
 - b. Configure it on the Property Editor panel according to the parameters given in Figure 35 and Figure 36.
4. Similarly, add a Player Start actor from the Actor Panel, set parameters in Property Editor as in Figure 37 and Figure 38.
5. Add TBBInitcomponent (Figure 39 and Figure 40) with the given parameters.
6. Add and configure snow PhysicsMaterial actor (Figure 41 and Figure 42).
7. Create Voxel actor (Figure 43).

- a. With the Voxel Actor selected in the Actor Search Panel, on the top bar click on Edit -> Add Actor Components (Figure 44).
 - i. Add SnowDeformationActorComponent
 - ii. Add PhysicsActComponent
8. Configure the Voxel actor's Properties as shown in Figure 45 and Figure 46.
9. Note the VoxelActor database being = Volumes:<path-to-our-vdb>. Press *Ctrl + S* to save the map. The .dtmap file will be saved in the ProjectAssets/Maps directory.

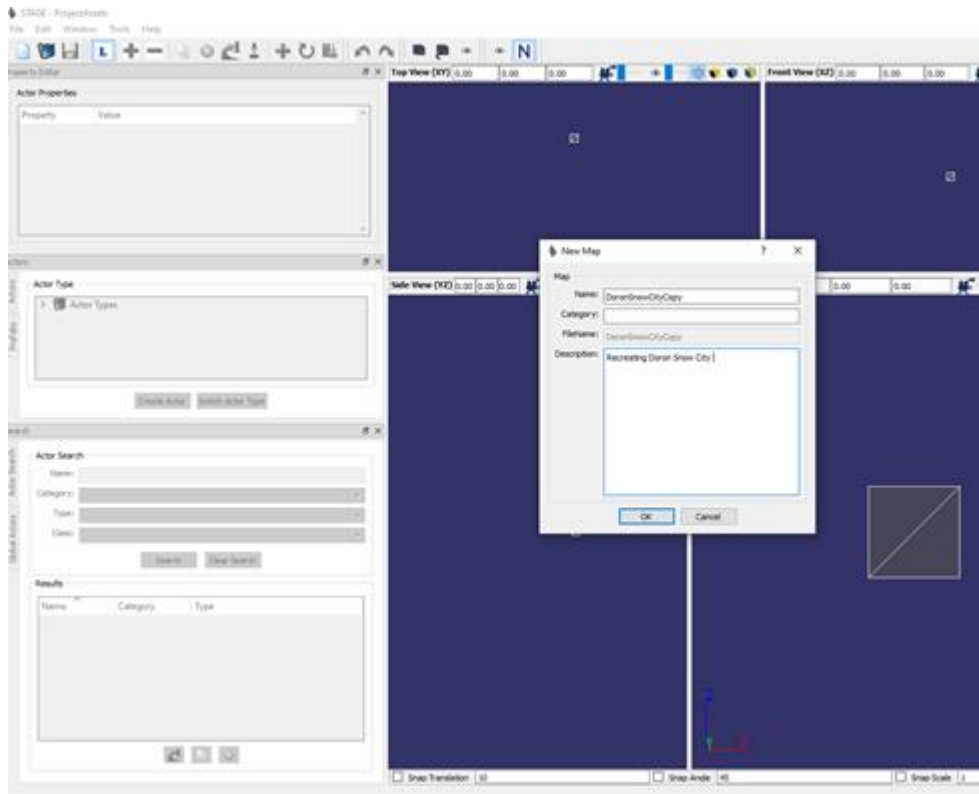


Figure 33. Creating New Map for Simulator

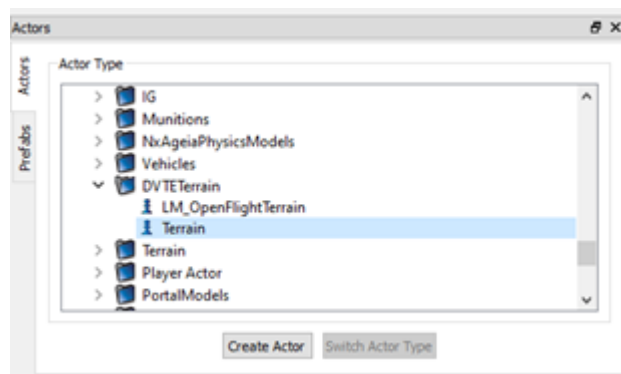


Figure 34. Terrain Actor

Property Editor

Actor Properties ('Terrain' selected)

Property	Value
Base	
Is Game Actor	True
Is Remote	False
Is Published	False
Initial Ownership	Client and Server Local
Local Actor Update Policy	ACCEPT_ALL
DeltaDrawable	
Description	
IsActive	True
PhysicsActComp	
Material Actor	<None>
Physics Name	
Physics Mass	0
Default Collision Group	0
Auto-Create Physics Objects	False
Physics Objects	1 Elements (0-*)
[0]	Physics object properties [Type: PROPERTY_CONTAINER]
PhysicsObject	
Name	Terrain
Collision Group	0
Mechanics Type	Static
Primitive Type	Terrain Mesh
> Translation	(X=0, Y=0, Z=0)
> Rotation	(X=0, Y=0, Z=0)
> Origin Offset	(X=0, Y=0, Z=0)
> Dimensions	(X=-1, Y=-1, Z=-1)
Mass	1
Skin Thickness	0.02
Notify Collisions	False
> Moment Of Inertia	(X=-1, Y=-1, Z=-1)
ActivationLinearV...	-1
ActivationAngular...	-1
ActivationTimeTh...	-1
LinearDamping	0.01
AngularDamping	0.01
PhysicsMesh	<None>
> PhysicsMeshScale	(X=1, Y=1, Z=1)
CollisionResponse...	True
Material	<None>
Joints	0 Elements (0-*)
ShaderParams	
ShaderGroup	TerrainShaderGroup
Taxonomy	
Name	Terrain
Type Category	DVTErrain
Type Name	Terrain
Actor Class	
Terrain	
Terrain Physics Mode	IMMEDIATE
TerrainMesh	Terrains:City.ive

Figure 35. Terrain Parameters (I)

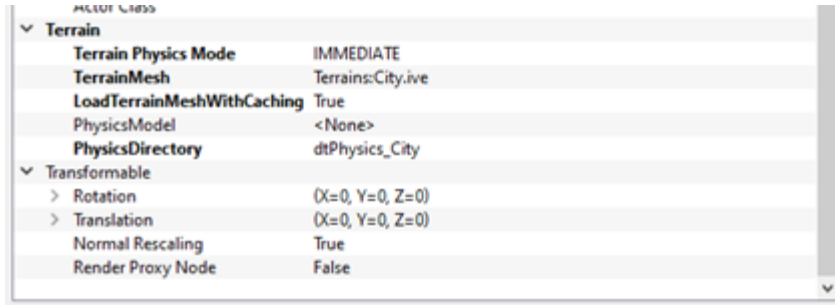


Figure 36. Terrain Parameters (II)

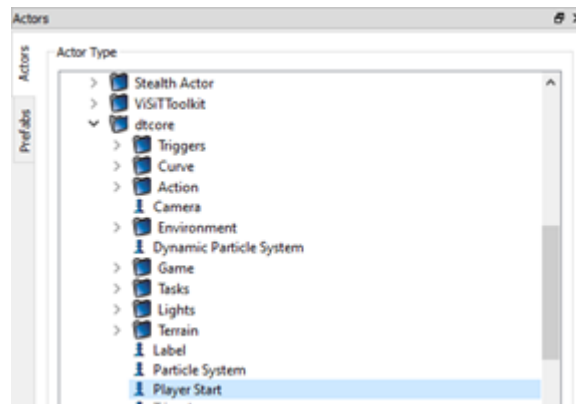


Figure 37. Player Start Actor

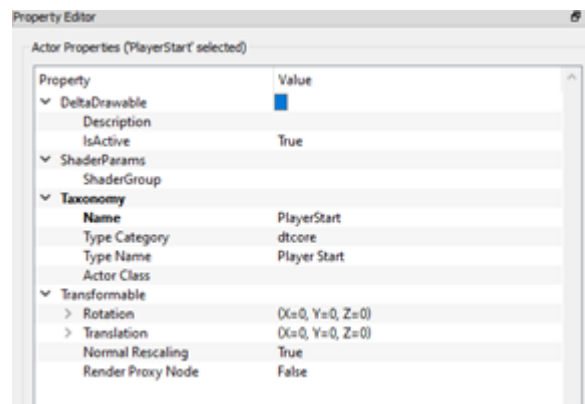


Figure 38. Player Start Actor Parameters

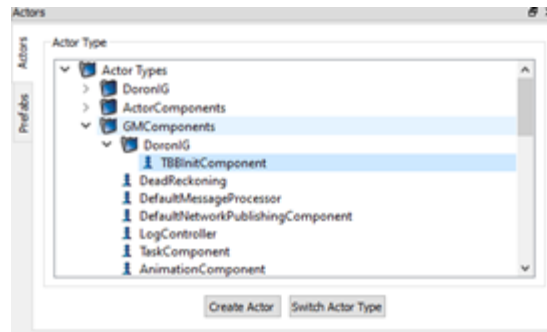


Figure 39. TBBInitComponent Actor

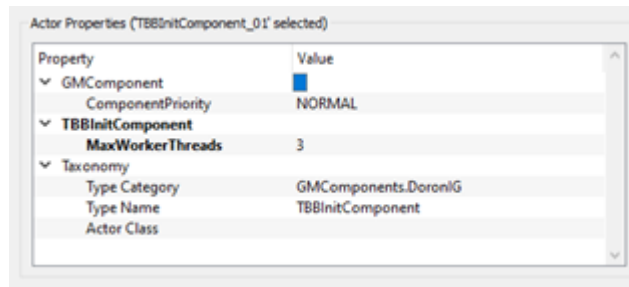


Figure 40. TRBInitComponent Actor Parameters

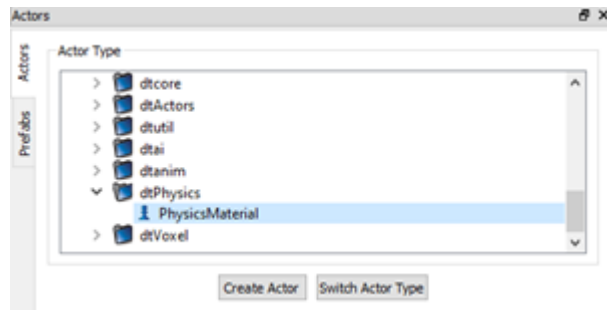


Figure 41. PhysicsMaterial Actor

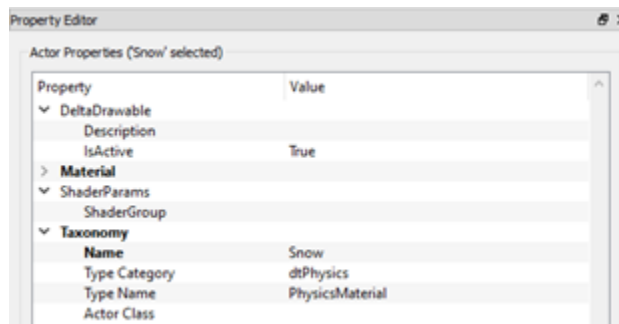


Figure 42. PhysicsMaterial Actor Parameters

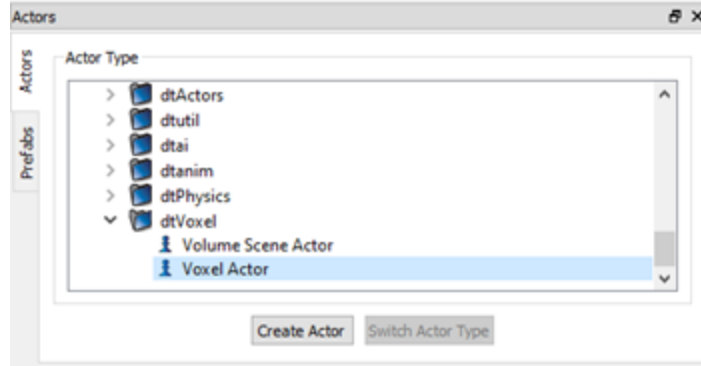


Figure 43. Voxel Actor for Snow VDB

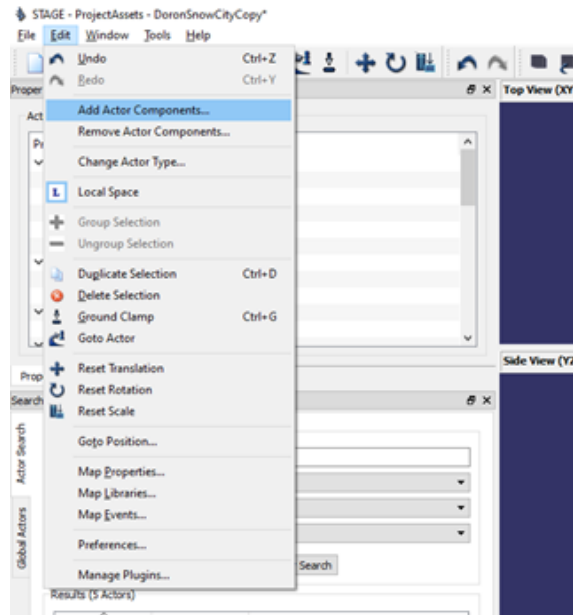


Figure 44. How to Edit and Add Actor Components

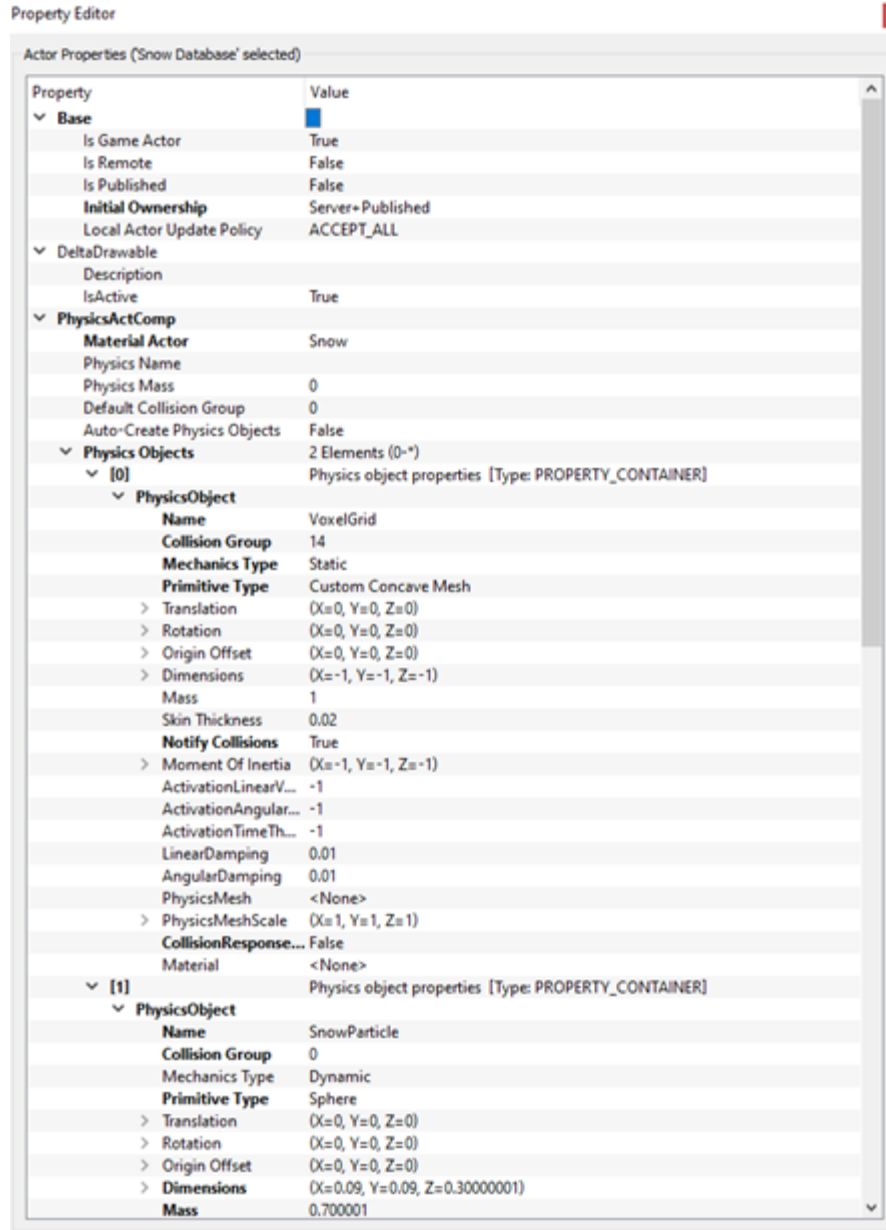


Figure 45. Voxel Actor Parameters (I)

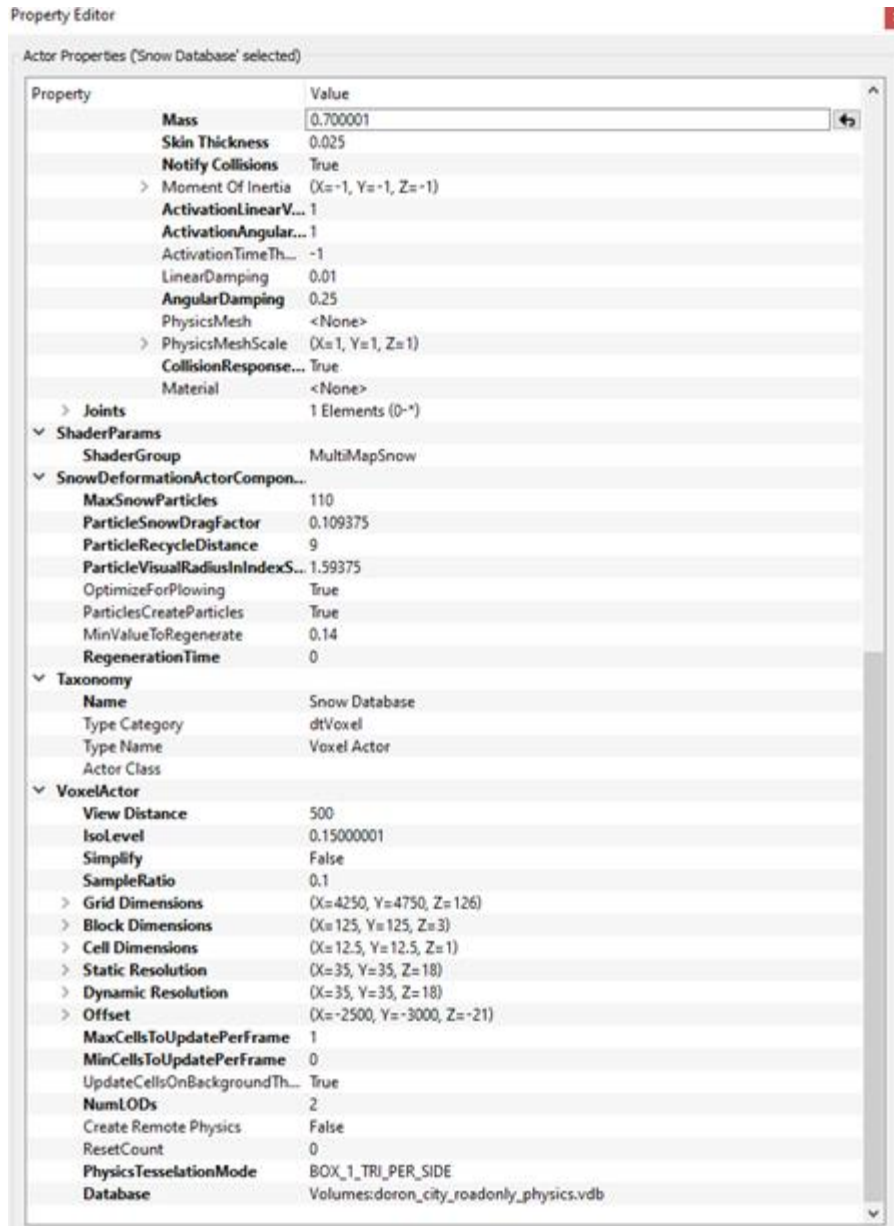


Figure 46. Voxel Actor Parameters (II)

4.2.2.2.4 Step 4: Deploying the Map and geometries to the Simulator

Once the research team successfully created a .dtmap file, all the files in /ProjectAssets needed to be transferred to each of the /ProjectAssets directory of all 4 PCs of the Cab. This included: the Map file (.dtmap), visual terrain IVE, dtPhysics folder, vdb's and static meshes into their respective folders as discussed above during the development procedure.

4.2.2.2.5 Step 5: Configuring a Scenario to bootstrap the Map into an Exercise

For the simulator to recognize and run a simulation on a newly deployed Map, the Instructor Work Station requires a scenario to be added for the new Map. This process needs to be done on the IWS PC by completing the following steps:

1. Create a .SCN file at C:/DoronIG/Scenarios. Name the file same as what one would want to name the exercise as; for eg: ExampleSnowMap.scn. Note that the Database specified should be the Map name.
 - a. The File contents (derived from existing .SCN files) would be as shown in Figure 47.
2. Add the Scenario file name to /DoronIG/data/exercises file by editing the in any text editor for e.g., Notepad

```
AMOS_2_FMT_1.1.0_SCENARIO ExampleSnowMap.scn
Database: NewSnowMap
{ SCENARIO
  { ENVIRONMENT
    17.50 0.00 90.00
  }
}
...
```

Figure 47. File Contents of .SCN File

One can now load this scenario which initiates the simulation in the new map created via the IWS application.

4.2.2.3 Summary of Tests Performed

In an effort to replicate the snow functionality of Doron City for new maps (i.e., to have plowable snow on custom maps such as those of various ODOT facilities), the research team experimented with numerous options at multiple steps of the development process to determine the root cause of the missing snow texture problem.

4.2.2.3.1 3D Export of snow geometry

As discussed earlier in the document, the research team has experimented the following methods to explore the snow geometry IVE:-

1. Using OSGExport plugins for 3dsmax
2. Converting the 3dsmax to a common format like OBJ or FBX and using OSGConv to get the IVE.

4.2.2.3.2 OpenVDBCompiler Parameters

Upon the successful building the OpenVDB compiler from open source codes, the research team performed various tests on the OpenVDBCompiler, and experimented the following parameters as listed in Table 4.

Table 4. OpenVDBCompiler Parameters

Parameter	Description
Bool Flag -b	If present, a boolean valued VDB grid is generated
Float Flag -f	If present, a float valued VDB grid is generated
Voxel Size	Voxel Size between 0 and 1. (for all x,y,z directions)
Voxel Size in Z	Override for Voxel Size in z direction
Thickness External	How thick to make mesh walls externally
Thickness Internal	How thick to make mesh walls internally
Subdivisions	Number of subdivisions to break the VDB into if needed, reduces memory usage

While any set of parameters here should be supported for VDB rendering in DoronIG, and the research team has tested them across their respective ranges, in order to rule out these variables the research team determined the actual parameters Doron used themselves, by analyzing their snow.vdb using the OpenVDB C++ API (see Figure 48). See Appendix 8.6 for the source code used to generate this data.


```

C:\>
Printing human-readable data for Grid in file: D:\development\3dassets\snow.vdb
Information about Tree:
Type: Tree_float_5_4_3
Configuration:
  Root(1 x 68), Internal(68 x 32^3), Internal(11,677 x 16^3), Leaf(1,981,100 x 8^3)
Background value: 0.150024
Min value: -0.141846
Max value: 0.150024
Number of active voxels:      224,957,726
Number of active tiles:      0
Bounding box of active voxels: [-13958, -18951, -98] -> [10393, 10598, 758]
Dimensions of active voxels: 24352 x 29550 x 857
Percentage of active voxels: 0.0365%
Average leaf node fill ratio: 22.2%
Number of unallocated nodes: 0 (0%)
Memory footprint:
Actual:      4.341 GB
Active leaf voxels: 858.146 MB
Dense equivalent: 2.244 TB
Actual footprint is 0.189% of an equivalent dense volume
Leaf voxel footprint is 19.3% of actual footprint
Additional metadata:
class: level set
file_bbox_max: [10393, 10598, 758]
file_bbox_min: [-13958, -18951, -98]
file_compression: active values
file_mem_bytes: 4660592564
file_voxel_count: 224957726
is_saved_as_half_float: true
Transform:
voxel dimensions: [0.15, 0.15, 0.05]
index to world:
  [0.15, 0, 0, 0]
  [0, 0.15, 0, 0]
  [0, 0, 0.05, 0]
  [0, 0, 0, 1]
Done...

```

Figure 48. Snow.vdb's Grid Parameters (this is the VDB for pre-installed Doron Map)

By looking at the details the research team determined that the Bash command to use for exporting snow geometry in the utility was as follows:

```
>>> OpenVDBCompiler.exe -f -r 0.15 -z 0.05 snow.ive
```

Note that in Table 5, it was determined that the pre-installed snow.vdb differs in its values of some parameters from the re-compiled Doron City vdb. This difference was found to occur despite using the definitive geometry file that Doron shared as being believed to be the source geometry for the original snow.vdb. See Appendix 8.10 for a copy of the 3dsmax scene for the geometry.

Table 5. snow.vdb, VDB shipped on the system, City_Road_Edited_P.vdb is a recompilation of Doron's road geometry created by the research team (Appendix E contains links to the files)

Geometry	VDB file:	snow.vdb	City_Road_Edited_P.vdb
	Input IVE (3D mesh)	Not known	City_Road_Edited_P.ive
OpenVDB Compiler parameter	Voxel Size X	Not known	0.15
	Voxel Size Y	Not known	0.15
	Voxel Size Z	Not known	0.05
	float OR boolean	Not known	float
	internal mesh thickness	Not known	0.15
	external mesh thickness	Not known	0.15
	subdivisions to break into	Not known	1
Parameters observed in Houdini	File Size	684 MB	658 MB
	Voxel Size X	0.15	0.15
	Voxel Size Y	0.15	0.15
	Voxel Size Z	0.05	0.15
	Voxel Count	224957726	210118255
	Datatype	float	float
	BoundingBox for activeVoxels (in Voxels)	24352 x 29550 x 857	24371 x 29550 x 1060
	BoundingBox in 3D grid space (in Meters)	3652 x 4432 x 42	3655 x 4432 x 53
	Is SDF	TRUE	TRUE

	isEmpty	FALSE	FALSE
--	---------	-------	-------

4.2.2.3.3 STAGE Voxel Actor Parameters

The Voxel Actor configuration in STAGE has chosen the following parameters for Doron City's sample as shown in Table 6.

Table 6. Voxel Actor Settings and Default Values in STAGE for Snow Map

VoxelActor Parameter	Description	Value
View Distance	The distance at which voxels will be generated into groups of volumes and rendered	500
IsoLevel	The value of the database which marks the surface, use the same number as the thickness if you compiled a database with the OpenVDBCompiler	0.15
Simplify	Whether or not to simplify the result, this can take a long longer to generate	False
SampleRatio	The percentage (expressed on a scale of 0-1) at which to sample in each dimension when simplifying, make sure to turn on the Simplify option to use this property	0.1
Grid Dimensions	The size of the grid to allocate into blocks	X=4250, Y=4750, Z=126

Block Dimensions	The size of the blocks within the grid	X=125, Y=125, Z=3
Cell Dimensions	The size of the cells within the blocks	X=12.5, Y=12.5, Z=1
Static Resolution	The resolution to sample the VDB database for the static pregenerated data.	X=35, Y=35, Z=18
Dynamic Resolution	The resolution to sample the VDB database for the dynamic deformable data.	X=35, Y=35, Z=18
Offset	The offset of the database in world space.	X=-2500, Y=-3000, Z=-21
MaxCellsToUpdatePerFrame	The number of dirty cells to regenerate per frame.	1
MinCellsToUpdatePerFrame	The minimum number to regenerate. If this is less than max, it will decrease the number if simulation cannot simulate as fast as configured in fixed time step. See system.h	0
NumLODs	The number of LODs to generate, can be 0, 1, 2 or 3	2
Create Remote Physics	Create the voxel geometry for the physics if this actor is remote	False

ResetCount	A database reset counter that will increment when reset is called so remotes will update	0
PhysicsTesselationMode	The mode of tessellating the physics geometry.	BOX_1_TRI_PER_SIDE
Database	Voxel database file to use	Volumes:snow.vdb

An important parameter for this actor as described by previous DoronIG developers consulted during the project was the Offset setting. It was initially suggested that the VDB is not centered at origin (0, 0, 0) in its local coordinate space. This implied the center of the VDB having some positive coordinate value and the origin was a "corner" of the volume. This idea led to experimenting with a range of Offset values which would compensate for the file's displaced center. In other words, it was possible that the voxels were rendering in the runtime, however they may have been out of view, they have been too far, or they may have been rendered below the terrain. However, after experimenting with a range of values for simpler maps and volumes it was observed the issue did not resolve, and there was indeed a problem with the rendering of the voxels. One can further observe that the snow.vdb is in fact centered on the true origin and is also viewable in debug mode in the simulator in the proper locations. It is worth noting that with regard to the snow.vdb file that ships with the simulator, any deviation from its default offset value results in the failure of snow rendering in the system. All this seems to contradict the developers' intuition of how this offset parameter is supposed to behave, since any such deviation from the offset should at most cause a displacement of the snow rendering, not a complete failure of the rendering.

4.2.2.3.4 Debug Mode and the Simulator runtime

The simulator runtime refers to a set of configuration parameters from a file called '661Truck_c_config.xml'. The file lets developers toggle logging levels for different modules of the system like dtVoxel, GameManager, Collision etc. Thus, by adding entries for specific (CPP) class files, developers can enable the logs for that module and define which level's logs are enabled; namely Error, Info, Debug, Always. One can then enable Debug Mode keys for the simulator cab by enabling the entry for it like so:

```
<Property Name="Debug.EnableDebugKeys">true</Property>
```

This enables one to view additional information (e.g., additional logs) about the runtime which can be useful in debugging issues. The source code of DoronIG (see Appendix 8.5) has supporting documentation describing how to use this Debug mode; a listing of debugging features is also provided in Appendix D.

One important feature of this mode is visualization of physics meshes involved in the simulation. This refers to meshes stored in the dtPhysics folder as discussed in previous sections. These are the meshes that interface with the physics engine modules of the system and notify any collisions that take place. See Figure 19 for an example where the wireframe rendering of the physics meshes is enabled. One can see the terrain's mesh, the vehicle's mesh, and also a physics mesh for the dynamic snow being plowed.

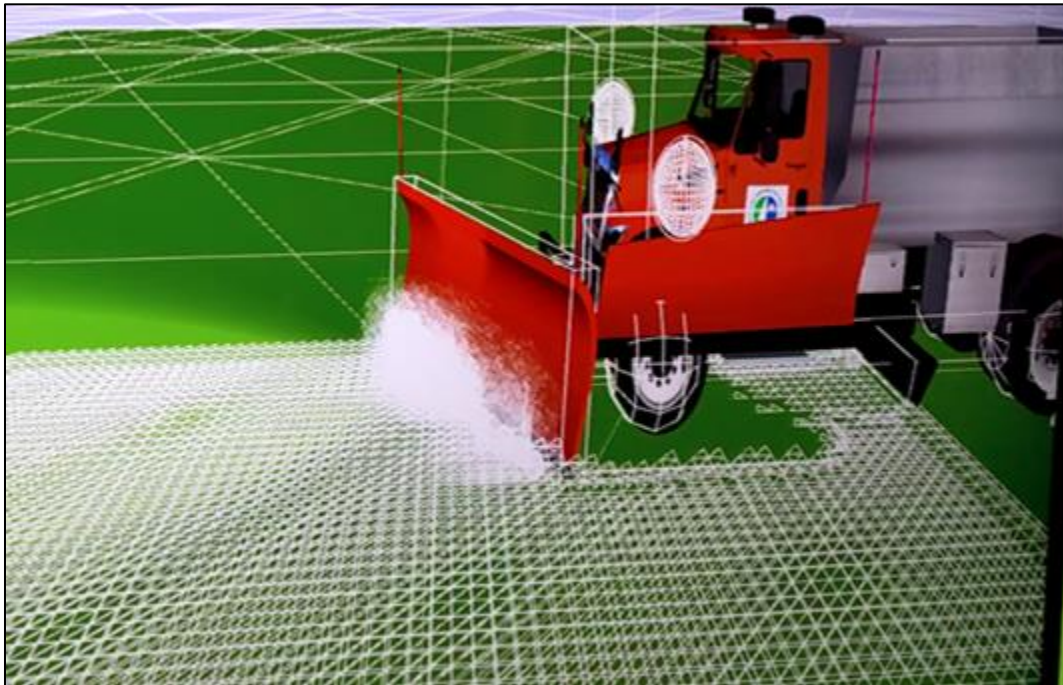


Figure 49. Debug Mode Example (the wireframes represent the physics meshes)

As shown in Figure 49. Debug Mode Example (the wireframes represent the physics meshes)Figure 49, it is possible to visualize the plowable physics mesh associated with the VDB in the debug mode. This is the case for all of the VDB tests. Any VDB the research team compiled worked as expected as visualized in this mode. Looking at the source code of the engine, particularly the module called dtVoxel, it was determined that the thread of execution for the physics mesh is different from that of the "Visua" VDB grid (has snow texture); the grid that should be rendered during runtime. Thus, this essentially verifies that the PhysicsActComponent of the Voxel actor is behaving as expected (only); however no general conclusions about the render functionality of the Voxel actor could be obtained. PhysicsActComponent is a STAGE configured component that is attached to the Voxel Actor which polygonizes the VDB into a physics mesh for notifying physics related changes like collision and volume updates. Appendix B contains STAGE actor settings PhysicsActComponent for the Voxel actor too.

4.2.2.3.5 Cache generation during VDB load

The delta3D engine powering DoronIG generates a cache directory for each VDB it tries to load in its file system for example, the working VDB's path is: C:/DoronIG/Project/ProjectAssets/Volumes/cache/snow.vdb/

See Appendix 8.9 for a copy of this directory from the simulator system.

This directory is utilized by the project to maintain a set of PagedLOD nodes of the snow vdb grid. The voxel actor's vdb grid is subdivided into blocks (and consequently into cells) which are processed in a parallel thread of execution. When one interacts with a block of snow, that block of geometry is essentially what receives volume update messages from the "Game Manager." Hence, only relevant sections of snow geometry are processed in the interest of processor and memory bandwidth.

LOD nodes refer to Level-of-Detail nodes, where a 3D geometry may have low resolution versions in cases when its rendering location is far from the viewer and the minute details of the geometry become irrelevant. This reduces memory usage because low fidelity models of far-away objects will occupy less running memory in the scene. However, one should note that the low LOD nodes still need to be present in the memory, and further optimizations need to be done when the number of LOD nodes are very high. Hence, a PagedLOD mechanism further tries to load-balance a scene's rendering pipeline by lazy loading the LOD nodes instead. All this decreases memory utilization because in a big map like Doron City in which one only interacts with VDB/snow nodes in proximity to the camera/vehicle/player.

Upon the initialization of the VDB grid in the runtime, this cache is probed to see if a PagedLODGrid was already created. If it is present rendering logic proceeds as usual, otherwise dtVoxel tries to recreate and populate this cache directory with PagedLOD nodes for blocks of snow VDB.

In numerous tests, the research team observed that cache for Doron's snow.vdb is present in the cache folder already, and upon it being deleted the regeneration of cache fails and the engine fails to render the (visual) snow.vdb. Note that this is the same snow VDB that ships with the system and works fine with all the existing scenarios and maps (when the cache is intact). Similarly, it was determined that the cache directory for the new recompilations of VDB also generate nodes which are empty, and this points to the potential root cause of the snow rendering issues. As an experiment, the research team tried to force the working cache to be referenced by the recompiled VDBs, snow rendering is observed successfully; however as expected, the behavior is undefined upon plowing, since there is a mismatch with respect to which block is getting volume updates upon being plowed.

4.2.3 Development of Simulation Scenarios for use in Training Exercises

4.2.2.4 Scenario Development Overview

A scenario is defined as a combination of a specific map, truck begin/end points, static objects (e.g. signs) and dynamic objects (e.g. moving cars). Throughout this task, the research team reviewed methodologies on scenario development and customization and created a set of scenarios to be used as ODOT training package. Scenario Development Overview

In the DoronIG simulator a “Scenario” represents an exercise that you can choose to practice on. A scenario can be as simple as spawning a vehicle at a particular location in the city to drive from (e.g., starting with the vehicle at a certain point), to more complex events involving scripted traffic lights, pedestrians, traffic, and weather control. Whenever one starts an exercise in DoronIG, a scenario file (with extension of the form .scn) is loaded, so the terms scenario and exercise can be used interchangeably. Hence, a scenario aims to simulate driving conditions which the instructor intends for the trainee/driver to practice on. Instructions should be given to the driver before each scenario in terms of objectives, general guidance, expected time, etc. Having this information will assist the driver in having the proper mindset for engaging in the situations confronted in the scenario and making appropriate decisions. Note that one can also combine a set of related scenarios into a “Lesson.”

4.2.2.5 Existing Snowplow Scenarios in DoronIG Simulator

The DoronIG simulator comes with 11 pre-installed snowplow scenarios which are listed as follows in Table 7. Please refer to the “Snowplow Scenarios” document packaged with the simulator operations manual for a detailed description of the scenarios. It is important to note that these scenarios are only designed to work on the default virtual world in the simulator called Doron City (as shown on the map in Figure 50. Doron City Map).

Table 7. Scenarios Pre-installed in the DoronIG Simulator

S. No.	Scenario	Category	Time (min:sec)	Short Description	Misc. information for Instructors	Location features
1	SP-5004	Exercise / Simulator Orientation	N.A.	Clear snow from the right cones.	Smooth and proper operation of plow.	Skills course area
2	SP-5005	-	3:00	Clear snow from the	Bridge overpass &	Urban streets

				right curb.	railroad tracks encountered.	
3	SP-5006	-	3:50	Enter gas station after plow exercise.	Choose driveway with less conflict.	Urban streets
4	SP-5007L	Tandem plowing (Designed to be used with 2 simulators)	N.A.	Plow interstate.	Appropriately deal with bridge overpass and on ramp.	Interstate, Tunnels, Tollbooth, Overpasses
5	SP-5008	-	4:30	Clear snow from cloverleaf.	Appropriately deal with bridge overpass and ramps.	Cloverleaf Interchange, On-off ramps, Overpass
6	SP-5009	-	6:50	Clear snow from cloverleaf.	Appropriately deal with overpass and 90° ramps.	Cloverleaf Interchange, On-off ramps, Overpass
7	SP-5010L	Tandem plowing (Designed to be used with 2 simulators)	4:00	Plow interstate.	Appropriately deal with tunnel and tollbooth.	Interstate, Tollbooth, Tunnels
8	SP-5011L	Tandem plowing (Designed to be used with 2	4:15	Refuel and plow interstate.	Appropriately deal with underpass and several stalled	Gas station, Interstate, Underpa

		simulators)			vehicles on the route	ss
9	SP-5012	-	3:08	Clear snow.	Appropriately deal with pedestrian overpass and a bridge.	Urban streets
10	SP-5013	Defensive Driving	5:50	Clear snow along country road	Yield to ambulance, sherrif's car; stop for school bus; pass white sedan only when line allows. Vehicle traffic present, animal crossing event involved.	Rural roads
11	SP-5014	-	2:20	Clear snow.	Appropriately deal with various vehicles and related issues faced. Traffic lights enabled, vehicular and pedestrian traffic present.	Urban streets

Doron 5.0 Map

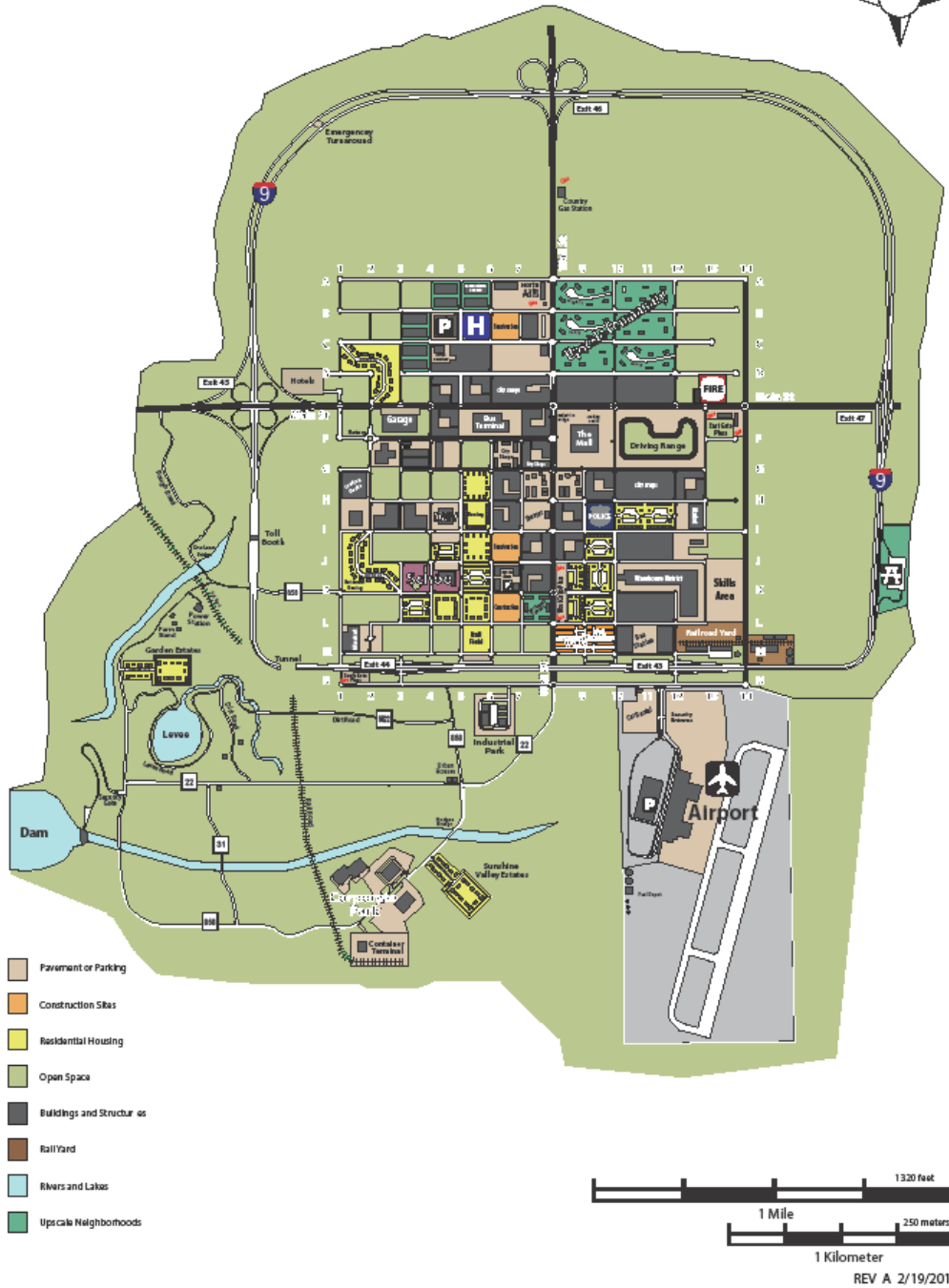


Figure 50. Doron City Map

The simulator also ships with non-snowplow scenarios for simulating relevant situations for law enforcement car chases, tractor trailers, buses, and mining trucks etc.

Interested readers can refer to the "Miscellaneous Scenarios," "Tractor Trailer Scenarios," and "Truck Scenarios" included in the simulator operations manual for more details.

4.2.2.6 Custom Scenario Development for DoronIG

The DoronIG simulator ships with a software utility that enables users/developers to create new scenarios or edit existing ones for Doron City (The scenario developer utility is made solely for Doron City). Following the procedures detailed in the "660Snowplowplus user guide" and the "Scenario Developer Manual" provided by Doron, UC team learned that the following features can change/edit in the existing scenarios:

- Configure custom Replay Target Areas (RTAs)
 - All exercises can be "replayed" by default for review/feedback purposes. However, by defining RTAs one can specify points of interest for recording the driving performance. This can be useful for saving time in very long scenarios.
- Configure environmental conditions
 - Allows one to adjust time of day (ambient lighting), fog density, and coefficient of friction.
- Add static objects
 - E.g., pedestrians, cones, vehicles etc.
- Add moving objects
 - E.g., pedestrians, vehicles, animals.
- Script Traffic Light Control (TLCs)
 - Allows user to program timing and location of traffic signals.

For this project, the research team explored the scenario development utility and used it to create several scenarios to supplement those packaged with the simulator and expand the exercises available for training. The snowplow scenarios developed for this project by the research team are shown in Table 8, and all scenarios were installed in the DoronIG simulator for use by ODOT. In the table, the "Scenario Code" is the name with which it is identified in the system.

Table 8. New Scenarios Created and Installed in DoronIG Simulator

S. No.	Scenario Code	Objectives	Description
1	SP-2-1	Plowing 2 lane 2 way route (Multipass)	Practice plowing a 2 lane 2 way route, might use multiple passes. Understand appropriate front plow usage, center line and shoulder estimation.

2	SP-2-2	Plowing 2 lane 2 way route (Singlepass)	Plow a 2 lane 2 way route in a single pass using wing-plow to plow greater width.
3	SP-2-3	Plowing Multilane/Interstate route (Wide median)	Plow an interstate route with wide median available for snow collection (along with shoulder).
4	SP-2-4	Plowing Multilane/Interstate route (Narrow median)	Plow an interstate route with narrow or no median available; 2 passes needed from center to shoulder.
5	SP-2-5	Plowing Intersections	Practice plowing a 4 way Intersection.
6	SP-3-1	Plowing bridges (type 1)	Practice best practices for plowing bridges.
7	SP-3-2	Plowing bridges (type 2)	Practice best practices for plowing bridges.
8	SP-3-3	Plowing over Rail-road crossings	Understand best practices for plowing through rail road crossings; appropriate plow and spreader adjustments.

4.3 Training Material Development and Evaluation

The final task of this project involved development of training materials for use in ODOT snow and ice driver education programs as well as information on evaluating driver performance in the training scenarios. The work for this task was itself comprised of two parts. The first portion of this task involved applying the development process discussed in the previous section to design specific scenarios to be used in ODOT's training curriculum. The second part of this task involved development of a guidebook focused on the training curriculum. Based on feedback from ODOT, the guidebook was split into two volumes to be used in tandem: one for drivers and one for instructors.

Combining the customized scenarios for Doron City, the research team created a training curriculum consisting of five modules. Each module consists of multiple scenarios (both pre-packaged and custom). The objectives of each module are briefly explained as follows:

- **Module 1:** This module focuses on the basics of snowplow vehicle operation and includes scenarios covering topics including but not limited to, pre-trip checks, backing up, parking, mirror use, etc. The focus here is on safely and properly driving the vehicle before learning more about plowing in the second module. This lesson provides the general introduction to participants and details components of the simulator system.
- **Module 2:** This module provides drivers with their first taste of operating the plow (i.e., basic plow techniques). Participants will complete scenarios plowing basic types of roadway facilities such as two-lane highways, interstate highways, intersections, and ramps. This lesson contains scenarios that deal with space management (distance behind other vehicles, using lanes properly, maintaining plow blade distance from curbs and adequate truck maneuver) and precision. Drivers will test/hone skills related to using plow controls for accurate and precise operations. The blades must be used effectively without going beyond the roadway segment or hitting standing objects used to delineate lanes. Trainers provide short insights on required skills for such operations before taking participants through applications in simulation.
- **Module 3:** The exercises in this module will allow the participants to practice operation on special areas. This includes bridges, rail crossings, uphill, and downhill terrains, etc. Instructor will observe the participant's maneuvers and plowing practices. Additionally, the participant will gain experience plowing at night, under low visibility conditions, and/or conditions with low surface friction. Here, drivers are tested on decision-making and speed management under various conditions. Skills such as SIPDE (Search, Identify, Predict, Decide, and Execute) will be briefly discussed by the instructors and demonstrated in the simulation environment using one of the exercises. Driver decisions under emergency conditions are also to be assessed here. Proper speed management techniques must be applied while observing road conditions and or performing certain snow and ice tasks in this

lesson. Instructors should provide insights to participants before the exercises. Instructors are expected to provide some insights to trainees on required skills for such operations before taking them through applications in simulation.

- **Module 4:** This module focuses on safe operations of a snowplow in the real-world, namely. Here, drivers must apply skills learned in the earlier modules to complete scenarios in which live traffic (both vehicle and pedestrian) is present. Defensive driving to avoid hazards and other general operations for safety (i.e., yielding to emergency vehicles, safe passing, etc.) are to be demonstrated and tested.
- **Module 5:** This lesson focuses on driver familiarity with ODOT routes. By training the participants on ODOT roadways, it is expected that this would provide enhanced training experiences and equip drivers with added skills during actual plowing seasons. Instructors may demonstrate useful skills to the participants based on their expertise and driving experience on these routes. The terrain of each route is a major part of this exercise intended to make drivers more aware with roadways in their jurisdiction before taking an actual plow out.

Figure 51 shows the configuration of the scenarios on the Doron City map. Scenarios in the last module replicate driving along actual ODOT routes in counties across the state.



Figure 51. Routes of Snowplow Scenarios Represented on the Doron City Map

Table 9 details the description of modules and their corresponding scenarios; the research team made an effort to ensure that the curriculum has a logical storyline and also covers the following challenges:

- Driving on facilities of different functional classification (differing numbers of lanes, different speed limits, different roadside and median design, etc.);
- Plowing roadway segments vs. intersections;

- Driving in conditions with differing ambient lighting (i.e., day vs. night);
- Driving along various terrain (e.g., level vs. rolling)
- Need to use different types of plows (i.e., wing, underbelly, and front)

Table 9. Training Scenarios Delivered for Project Organized as a Curriculum

Module Name	Prerequisite	Objectives	Description	Development Status	Scenario Code (revised)	
Module 1: Basics of Simulator Operation	1.1	NA	Familiarize with Cab controls, Pre-drive checks	Practice important pre-drive checks, understanding plow controls, radio operation; practice in training area; change types of vehicles as needed.	SP-5004 was used	SP-1-1
	1.2	NA	Precision driving and backing	Practice mirror use, understand pivot points and blind spots while practicing parking exercise.	Various (non Snowplow) Trucks scenarios can be used	TK-5001 to TK-5019

Module Name	Prerequisite	Objectives	Description	Development Status	Scenario Code (revised)	
Module 2: Plowing Techniques (basic)	2.1	1.1, 1.2	Plowing 2 lane 2 way route (Multipass)	Practice plowing a 2 lane 2 way route, might use multiple passes. Understand appropriate front plow usage, center line and shoulder estimation.	New Scenario Created	SP-2-1
	2.2	1.1, 1.2	Plowing 2 lane 2 way route (Singlepass)	Plow a 2 lane 2 way route in a single pass using wing-plow to plow greater width.	New Scenario Created	SP-2-2
	2.3	1.1, 1.2	Plowing Multilane/Interstate route (Wide median)	Plow an interstate route with wide median available for snow collection (along with shoulder).	New Scenario Created	SP-2-3
	2.4	1.1, 1.2	Plowing Multilane/Interstate route (Narrow median)	Plow an interstate route with narrow or no median available; 2 passes needed from center to shoulder.	New Scenario Created	SP-2-4
	2.5	1.1, 1.2	Plowing Intersections	Practice plowing a 4 way Intersection.	New Scenario Created	SP-2-5
	2.6	1.1, 1.2	Plowing Ramps	Practice plowing snow on ramps.	SP-5007 was used	SP-2-6
	2.7	1.1, 1.2	Plowing Cloverleafs	Practice plowing cloverleaf intersections.	SP-5009 was used	SP-2-7

Module Name	Prerequisite	Objectives	Description	Development Status	Scenario Code (revised)	
Module 3: Plowing Techniques (advanced)	3.1	2.1 to 2.4	Plowing bridges (type 1)	Practice best practices for plowing bridges.	New Scenario Created	SP-3-1
	3.2	2.1 to 2.4	Plowing bridges (type 2)	Practice best practices for plowing bridges.	New Scenario Created	SP-3-2
	3.3	2.1 to 2.4	Plowing over Rail-road crossings	Understand best practices for plowing through rail road crossings; appropriate plow and spreader adjustments.	New Scenario Created	SP-3-3
	3.4	2.1 to 2.7	Plowing in low visibility and precipitation	Practice plowing in low traction and low visibility conditions and/or night time.	SP-5005 was used	SP-3-4
	3.5	2.1 to 2.7	Night time plowing (1)	Night time plowing.	SP-5006 was used	SP-3-5
	3.6	2.1 to 2.7	Night time plowing (2)	Night time plowing.	SP-5008 was used	SP-3-6

Module Name	Prerequisite	Objectives	Description	Development Status	Scenario Code (revised)	
Module 4: Safe operation in the real world	4.1	3.1 to 3.3	Plowing in Traffic (1)	Reacting to stalled vehicles, stationary hazards.	SP-5011 was used	SP-4-1
	4.2	3.1 to 3.3	Plowing in Traffic (2)	Practice with Traffic enabled, pedestrian hazard.	SP-5012 was used	SP-4-2
	4.3	3.1 to 3.3	Plowing in Traffic (3)	Defensive driving, yielding to emergency vehicles, safe passing, animal crossing hazard.	SP-5013 was used	SP-4-3
	4.4	3.1 to 3.3	Plowing in Traffic (4)	Driving in traffic congestion exercise	SP-5014 was used	SP-4-4

Module Name	Prerequisite	Objectives	Description	Development Status	Scenario Code (revised)	
Module 5: Drive on Ohio Routes	5.1	N.A.	Practice plowing on real Ohio Route	Lorain county, South Route. 82, 83, 57 (20 Miles)	Deployed	D3-A-L1
	5.2	N.A.	Practice plowing on real Ohio Route	Lorain county, South Route. 82, 83, 57 (20 Miles) Different vehicle starting point from D3-A-L1	Deployed	D3-A-L2
	5.3	N.A.	Practice plowing on real Ohio Route	Lorain county, North Route. I-90 (10 Miles)	Deployed	D3-B
	5.4	N.A.	Practice plowing on real Ohio Route	Hocking county, Road 664 (10 Miles)	Deployed	D- 10
	5.5	N.A.	Practice plowing on real Ohio Route	Columbus, Road 270 (10 Miles)	Deployed	D6-A
	5.6	N.A.	Practice plowing on real Ohio Route	Columbus, Road 270 (10 Miles)	Deployed	D6-B
	5.7	N.A.	Practice plowing on real Ohio Route	Columbus, Road 270 (10 Miles)	Deployed	D6-C

The second part of this task involved development of a guidebook to be used as part of the training curriculum for ODOT. Again, the book was split into two volumes, one for students/drivers and one for instructors. The guidebooks discuss numerous aspects of the training process including simulator operation (e.g., how to start up and use the simulator system, how to load scenario exercises, etc.), instructive material on snowplow operations (e.g., space management, speed management, plow operation, etc.), discussion of specific ODOT facilities and gaining familiarity with a driver's future routes, and finally information on how an instructor may evaluate a student's performance in various scenarios of the training process. Both guidebooks have similar material, the main difference, however, being that the guide for the students focuses on instruction, while that for the instructor has components on higher-level learning objectives and evaluation of students.

5 Findings and Conclusions

To recap, the goal of this project was to prepare a snowplow simulation component for ODOT's snow and ice driver training program in order to allow drivers to gain experience with plow operations before driving an actual vehicle on the road. The benefits of adding this component to the training program are to improve safety for both drivers and the motoring public, as well as to improve efficiency of plowing operations. In the first phase

of this project, the research team collected data on snowplow driver behavior and used this to make a recommendation to ODOT on a snowplow simulator system to purchase, the Doron SP660.

With the SP660, the research team was able to help set the simulator up for use by ODOT. This process included simple aspects such as determining how to start up the system and load exercises, to more complicated aspects such as minor hardware upgrades to the system's computers. Different aspects of the simulator ranging from components to be used in the training (e.g., the gas pedal, steering wheel, and control joysticks), to components of the simulator system itself (e.g., the motion base), to the PC system used to control the simulator were tested and verified to work properly for the most part. Occasionally, there were issues with the motion base failing to function, however, as noted earlier, a simple restart of the system usually solved this problem. The research team has documented necessary setup/usage procedures for the simulator as part of this project's deliverables.

The bulk of the work in this project involved development of training scenarios for use by ODOT in its snow and ice driver training program. This scenario development component was two-fold. First, the research team worked to develop detailed 3D models of ODOT facilities in three different districts across the state. These models were based on maps and images of the real sites, and they were verified for authenticity by drivers of each region. Each model was further optimized in order to ensure smooth running in the simulator system; typical optimization efforts included things such as adjusting the resolution of objects in the environment to adjusting the length of the segment to be driven in each scenario. The team determined that using commonly available modeling software, it was possible to develop such maps to enable more realistic simulation environments where drivers can drive on virtual versions of roads they would drive on in the real world.

The second component of the scenario development involved the creation of scenarios in which drivers could practice specific skills as a part of their training (e.g., how to plow an intersection). The research team worked with ODOT training staff and snowplow drivers and determined that both scenarios included with the simulator system, as well as custom scenarios developed to allow for practicing additional skills would be used in the training curriculum. The scenarios for practice of specific skills were developed to take place in the built-in simulator system map called Doron City. Originally, the goal of the project was to develop training scenarios to take place in the ODOT maps discussed earlier. In the end, the research team was able to deliver ODOT maps (i.e., the models of specific ODOT facilities) that can be driven on in the simulator system, but these maps were not able to include plowable snow. This was a major issue, determined to be related to both physics models within the software as well as other issues related to generation of the virtual environments. The research team investigated this issue as thoroughly as possible, and even consulted Doron and former software engineers at Doron for help. Neither group was able to provide a solution, hence, the issue remains open.

The final portion of the project involved the development of training manuals in the form of guidebooks to be used during a training program by both drivers (trainees) and instructors. These materials were prepared based on discussion with ODOT training staff,

existing ODOT training materials, and the developed scenarios within the simulator. Ultimately, the combined efforts in this project will lead to a stronger training program for ODOT snow and ice drivers, which has obvious benefits in terms of safety and efficiency. Based on numerous discussions and demonstrations of simulation scenarios with ODOT snowplow drivers and training staff, the research team believes that this system has the potential to provide drivers with a relatively realistic experience enabling them to build up their skills before getting on the road.

6 Recommendations for Implementation

Based upon the work conducted in the Phase 2 portion of this project, the research team believes that the SP660 driving simulator is a tool of potential value for use in ODOT's snow and ice driver training program. The simulator functions as expected and is ready for use by a broader, non-research audience. Upon collecting the simulator and trailer from the research team's facilities, ODOT will be able to use the simulator around the state to train snow and ice drivers. Further, the training staff will be able to use the following components in their teaching program, all of which were developed under this project:

- A series of training scenarios for the driving simulator that teach/test different components of snowplow operations across varying environments;
- Customized virtual environments modeled to mirror actual ODOT facilities; and
- Training materials in the form of guidebooks for drivers and instructors.

The potential benefits from this project are numerous and range from providing a realistic, yet safe and controlled environment for training snow and ice drivers to enabling the use of a standardized tool for driver training. It is expected that drivers will be able to better learn vehicle operations before going out onto the actual roadway, thus improving their efficiency, as well as improving both their safety and that of the motoring public. Risks to implementation appear relatively small overall. There is potential for drivers training in the simulator to get short-term vertigo (e.g., brief dizziness and nausea), but as noted previously, this typically subsides in minutes. Additionally, as with all computer and mechanical equipment (the primary components of the simulator), potential failures of elements will necessitate replacement, a task which may not always be easy within a proprietary system (e.g., how might one fix the motion base if it fails?). This issue can ideally be resolved by maintaining a relationship with the vendor and servicing the system as necessary.

The simulator system is ready for implementation and use in training programs throughout the state, following final training sessions for ODOT staff on the system's operations as well as overall project developments to be conducted by the research team. Upon transfer of the system to ODOT from the research team, it is believed that the system can begin to be used immediately. As the introduction of the simulator system itself brings about new elements for the ODOT training program on snow and ice operations, training staff will need to be briefed on the new curriculum and learn to use the simulator system. Further, it would be beneficial to monitor the performance (e.g., accident record, etc.) of drivers who take part in the new training program with the driving

simulator compared to that of drivers who did not use the simulator in their training. This has the potential to help demonstrate the value of the system over time.

7 Bibliography

Camden, M. C., Hickman, J. S., Tidwell, S., Soccolich, S. A., Hammond, R., Hanowski, R. J., & Virginia Tech Transportation Institute. (2020). Defensive Driving for Snowplow Operators. <https://rosap.ntl.bts.gov/view/dot/53985>

Darken, R., McDowell, P., & Johnson, E. (2005). rojects in VR: the Delta3D open source game engine. *IEEE Computer Graphics and Applications*, 25(3), 10–12. <https://doi.org/10.1109/MCG.2005.67>

Iowa Department of Transportation. (2020). Winter Operations Training Series. <https://www.youtube.com/playlist?list=PLurY2WfsVWKn9ismDC4Uz3lbRivAnf0Ld>

Kihl, M., & Arizona State University. College of Architecture and Environmental Design. Dept. of Planning. (2006, 11 1). Snowplow simulator training evaluation. (FHWA-AZ-06-585). <https://rosap.ntl.bts.gov/view/dot/16219>

Masciocchi, C. M. (2007). Evaluation of virtual reality snowplow simulation training. *Retrospective Theses and Dissertations 14528*. <https://doi.org/10.31274/rtd-180813-15717>

Minnesota Department of Transportation. (n.d.). MnDOT Mobile Snowplow Simulator Training. MnDOT Mobile Snowplow Simulator Training. Retrieved 2021, from <http://www.mnltap.umn.edu/training/topic/maintenance/simulator/index.html>

Strayer, D. L., Drews, F. A., & Burns, S. (2004). The Development and Evaluation of a High-Fidelity Simulator Training Program for Snowplow Operators. <https://rosap.ntl.bts.gov/view/dot/38137>

8 Appendix

8.1 Summary Literature Review on Practices of Agencies using Snowplow Simulators

Many agencies across the country have utilized simulator training for training their snowplow drivers, or at a minimum investigated the option as part of the research to develop their training programs. With the application of scenarios in mind, we have reviewed these works and summarized them below. First, however, Table 10 shows a variety of simulators in use by agencies today. The simulators often come bundled with pre-programmed scenarios, and the scenario library features are mentioned on their respective websites.

Table 10. Popular Simulators with Snowplow Scenarios

Simulator	Manufacturer	Website
660 Snowplowplus	Doron Precision Systems	https://doronprecision.com/snowplow-driving-simulators/
VS600M	Virage Simulation	https://viragesimulation.com/snowplow-simulator-vs600m/
FAAC SP	FAAC	https://www.faac.com/simulation-training/transportation/snow-plow-driver-training/

MnDOT offers a snowplow training program which is a combination of Classroom and Simulator Training (MnDOT, n.d.). They use the SP660 simulator provided by Doron Precision Systems, the same as is being used by the Ohio DOT for this project (herein referred to as 'DoronIG'). A study on effectiveness of snowplow simulators for training was conducted by Iowa State University (Masciocchi, 2007). Training involved three modules: (1) Powerpoint session on S.I.P.D.E. method; (2) 5-10 minute simulator driving scenario for employing techniques learnt; and (3) Short video training clips about passing vehicles, speed management, and space management. Performance was measured in an experimental scenario which involved merging onto an Interstate with traffic and plowing for 10 minutes. The simulator used was the TranSim VS III, from L3 Communications. The Iowa DOT also has a video series for winter operations training which covers essential concepts and techniques used for snow plow operations (Iowa Department of Transportation, 2020). Utah DOT's (UDOT) training program focused on a series of 18 customized scenarios (9 in urban interstate conditions and 9 in rural mountain settings) that provided training on the critical issues of a) space management, b) speed management, c) crew communication, and d) fuel

management; were developed. The simulators used UDOT were the MARK II high-fidelity motion-based simulator and the TranSim VS sold by L3Harris (Strayer et al., 2004). The Arizona DOT (ADOT) Technical Training Group (ITD-Tech) contracted with L-3 Communications - MPRI Ship Analytics to give third-party simulator training to snowplow operators in rural Arizona (Kihl, 2006). The program consisted of classroom portions on SIPDE and simulator scenarios based on: snow-covered freeways, mountain passes with tunnels, parking lots, high country driving, and in-town driving. The Virginia Tech Transportation Institute focused on developing training modules based upon the goal of improving safety. They noted that the five most common preventable snowplow crash types were as follows: (1) fixed-object strikes, (2) run-off-road crashes, (3) backing crashes, (4) wing-plow strikes and (5) crashes where another vehicle rear-ended the snowplow. In turn, they developed two comprehensive and engaging snowplow operator training modules on safe and defensive driving (Camden et al., 2020).

Based on the review of existing agency practice for use of simulation in training of snow and ice drivers, the following suggestions for improving the scenario library were documented. Scenarios suggested included: (1) cars passing on both sides of the plow truck; (2) predictable road hazards (railroad tracks, expansion joints, cattle guards, etc.) that are generally known on familiar routes are potentially dangerous on unfamiliar routes; (3) unexpected road hazards (e.g., snow-covered rock in a curve); and (4) various weather and road conditions, other than snow (e.g., rain, sleet, hail, black ice, etc.). Some drivers suggested that it would be useful to begin simulator scenarios in a very simple form and add more distractions as the trainee becomes more comfortable and gains more experience.

8.2 Attributes of Doron City

Recall the map for Doron City can be seen in Figure 50. This simulated city where the scenarios occur has 97 blocks of varying shapes and sizes. Features include:

- 97 urban blocks & rural areas;
- Highways/freeways/interstate;
- Traffic lights;
- River, railroad tracks, overpass and hills;
- Hospital and School zones;
- Industrial and Residential areas;
- Truck station; and
- Skills course & Training area.

8.3 Source files

Besides the six databases (Table 1) and one initial scenario developed for each database, the team has provided the source files to ODOT to allow further development. These source files include:

- Complete 3dsMax files for six databases

- Textures in the “3dsMax/asset/images” folder
- IVE files in the “export” folder

8.4 Maps of Six ODOT Road Databases

8.4.1 D3-A South. Royalton, River Road, Lorrain County



Figure 52. Lorrain County South. D3-A (20 miles)

[Google map](#)

8.4.2 D3B: North. I-90

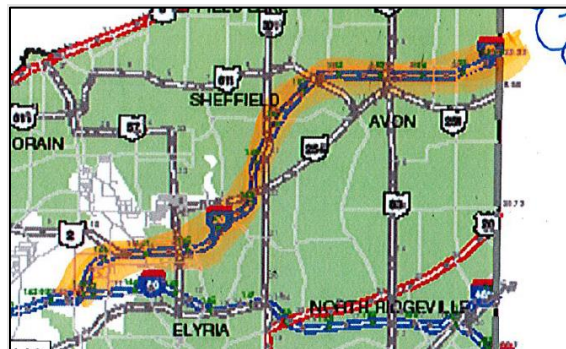


Figure 53. Lorrain county North. D3-B (10 miles)

[Google map](#)

8.4.3 D10 : OH-664

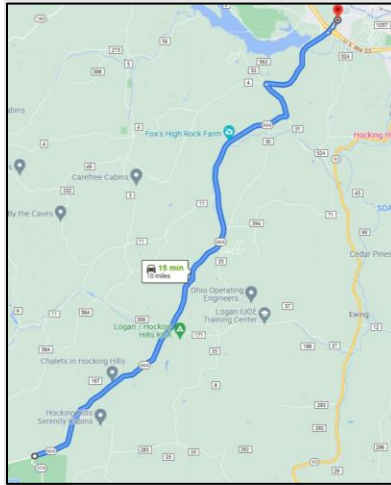


Figure 54. Hocking County (10 miles)

[Google map link](#)

8.4.4 D6: Columbus

Due to the complexity of this route, the site was organized into three databases as D6A, D6B, D6C

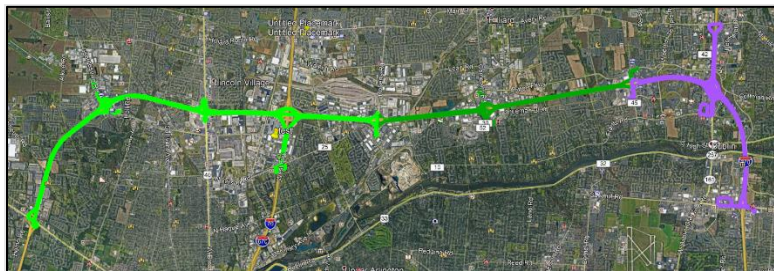


Figure 55. From south (left) to north (right), D6A, D6B, D6C

8.4.4.1 D6A



Figure 56. Reference image at Google D6A

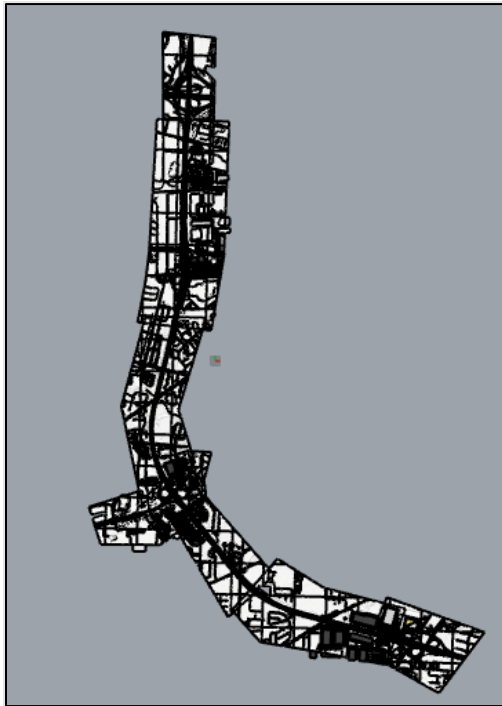


Figure 57. Screenshot of the site D6A from 3D modeling program

8.4.4.2 D6-B

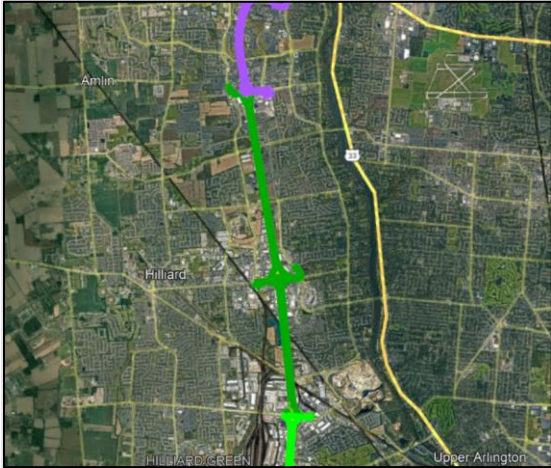


Figure 58. Reference image at Google D6B

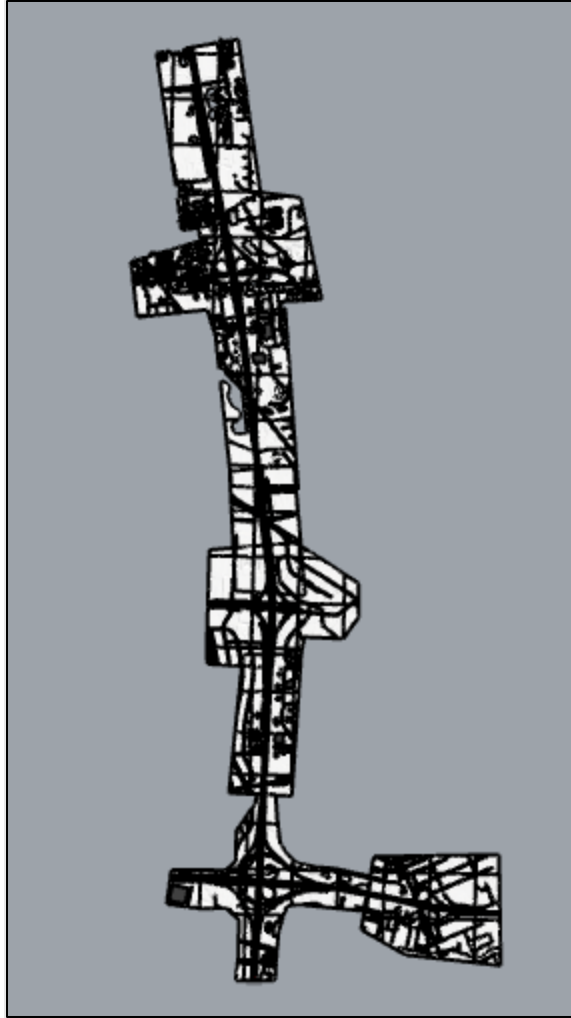


Figure 59. Screenshot of the site D6B from 3D modeling program

8.4.4.3 D6-C

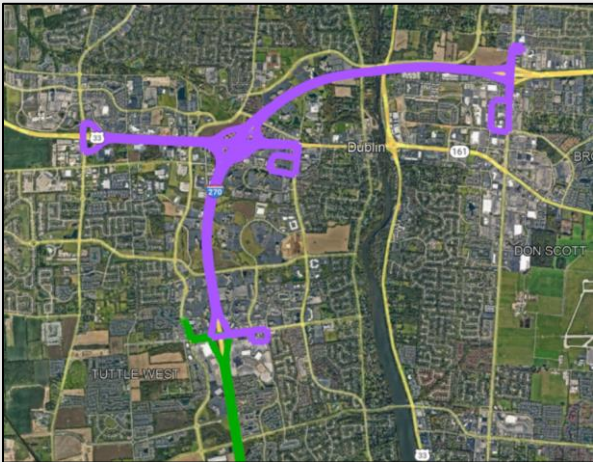


Figure 60. Reference image at Google D6C

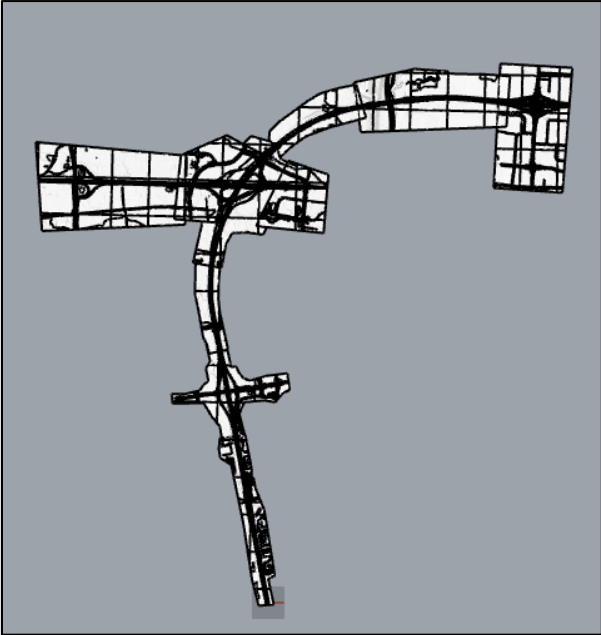


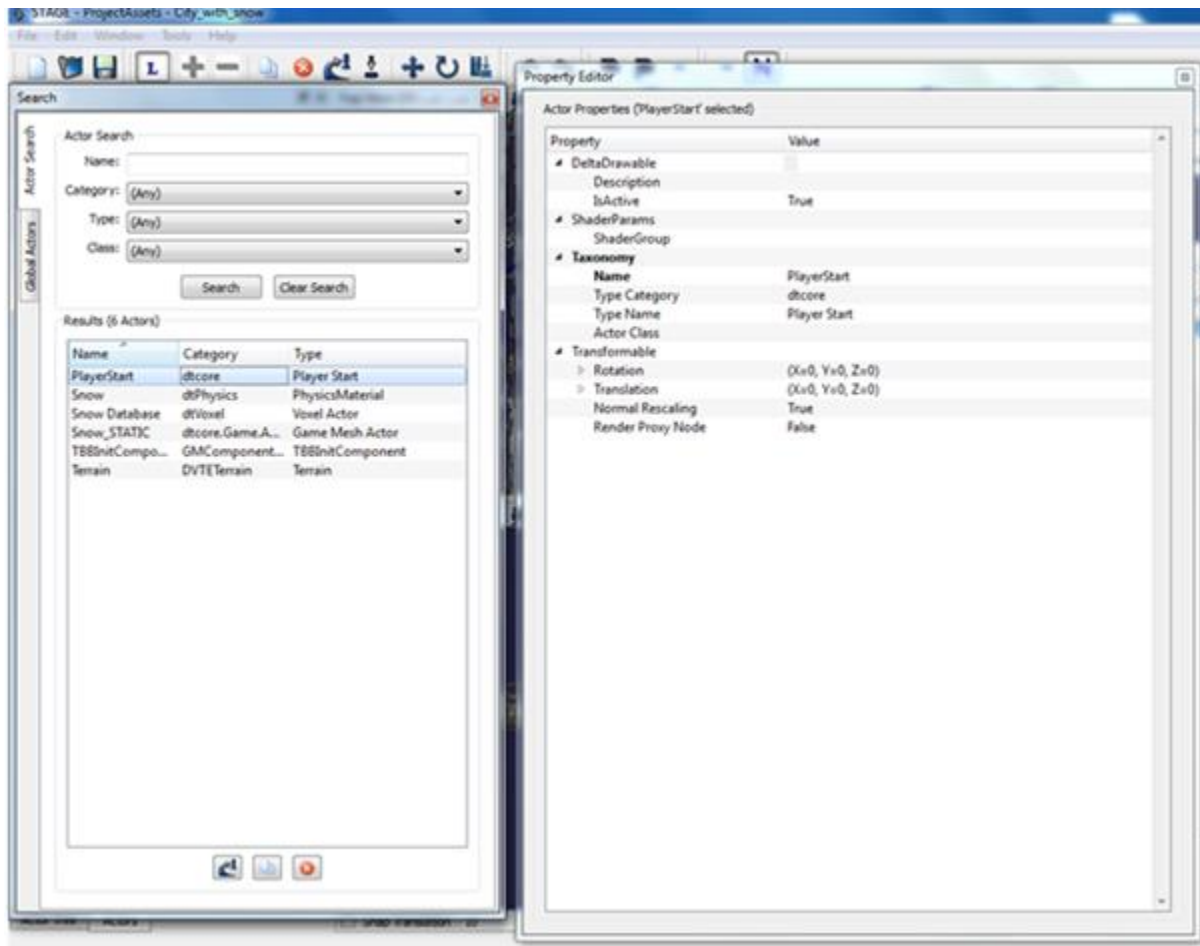
Figure 61. Screenshot of the site D6C from 3D modeling program

8.5 DoronIG Project Source Code

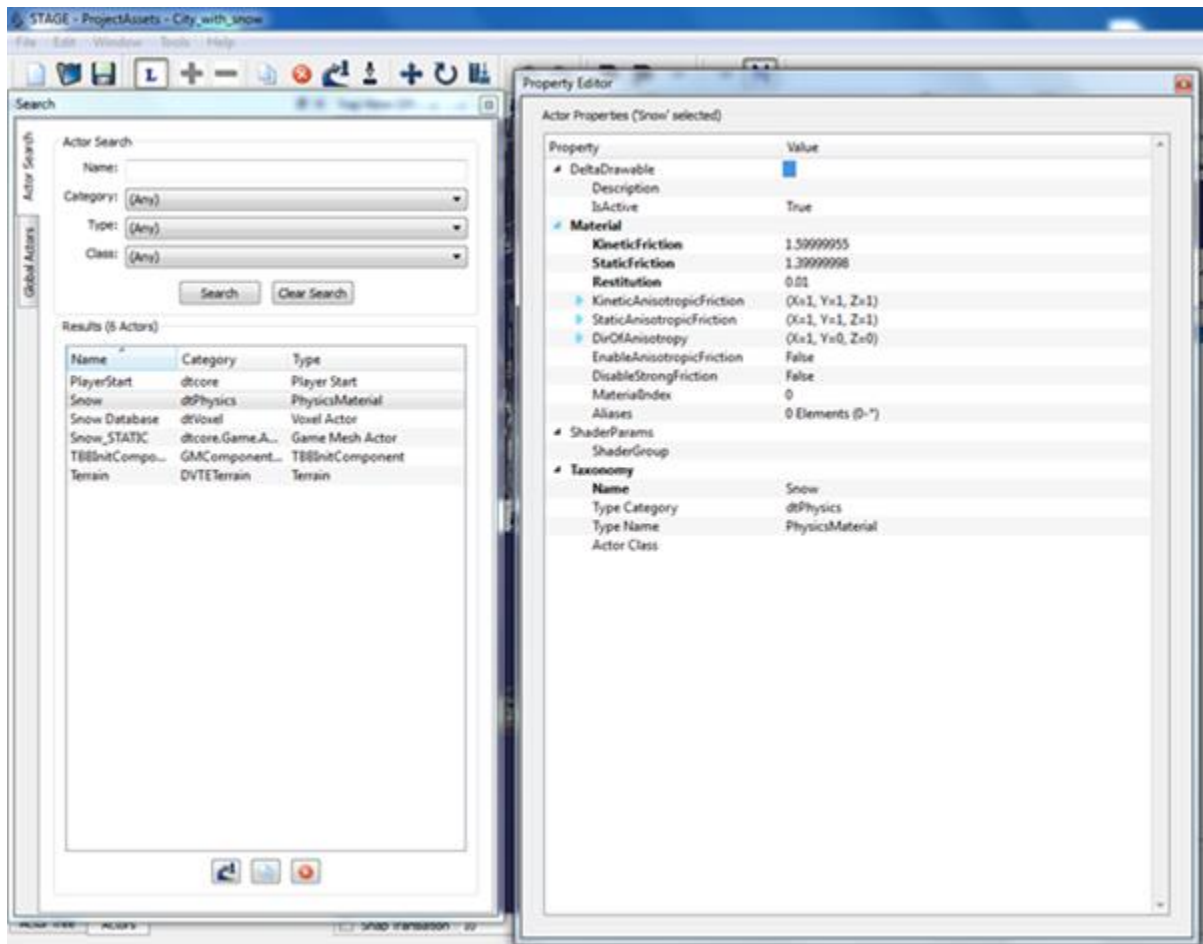
[DoronIG Source Code Repository Hyperlink](#)

Doron City with Snow STAGE actors settings:

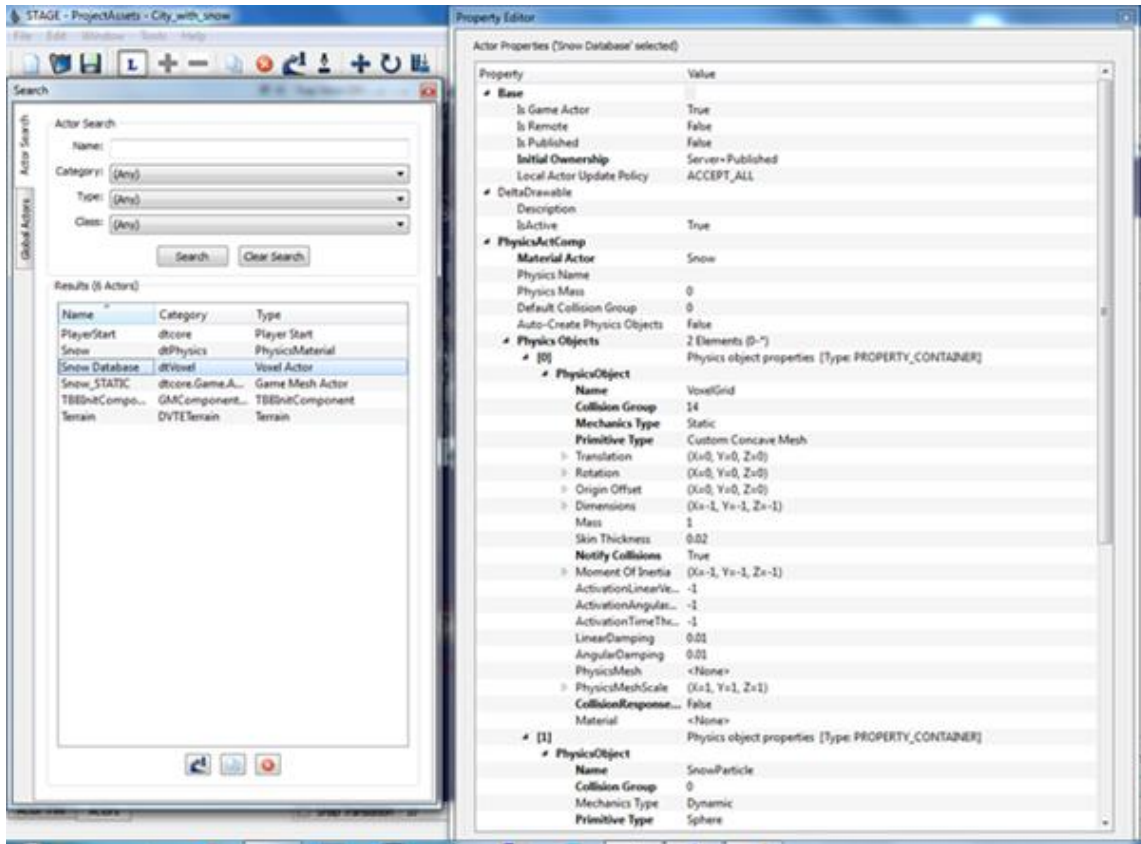
Dtcore:

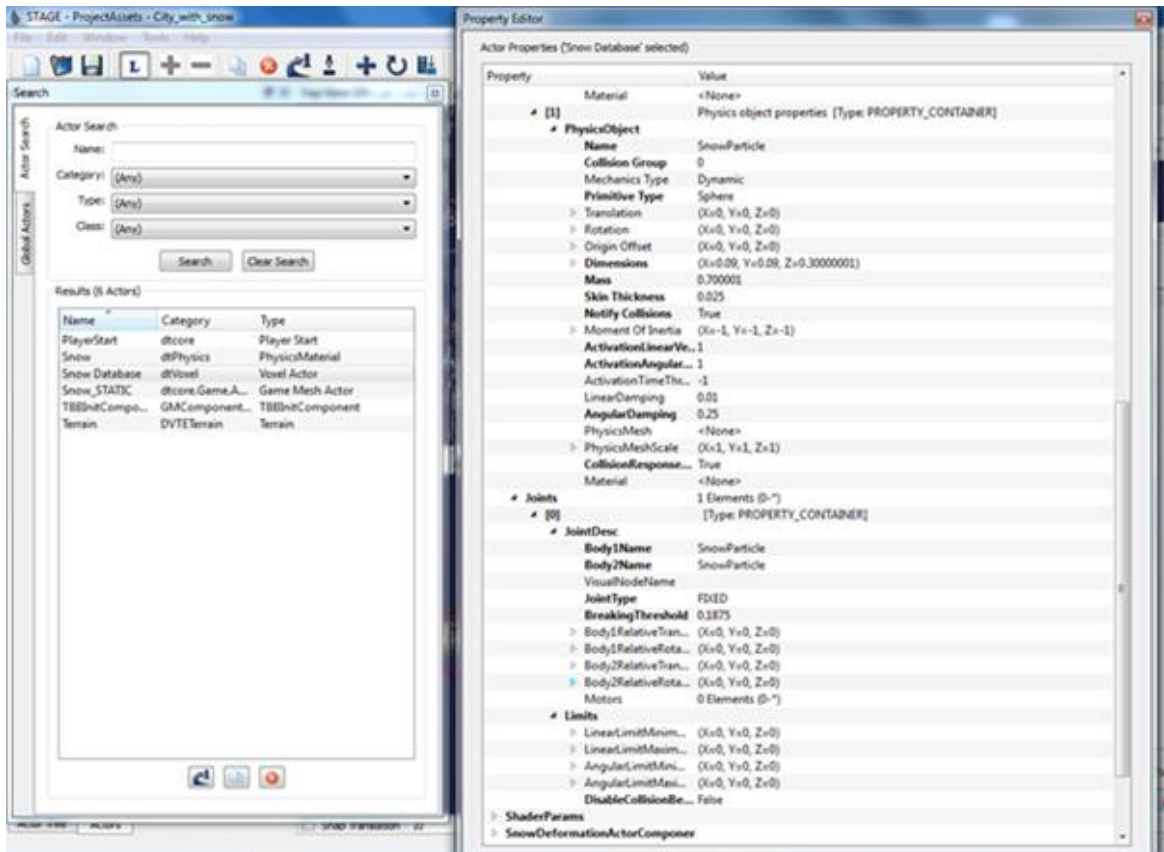


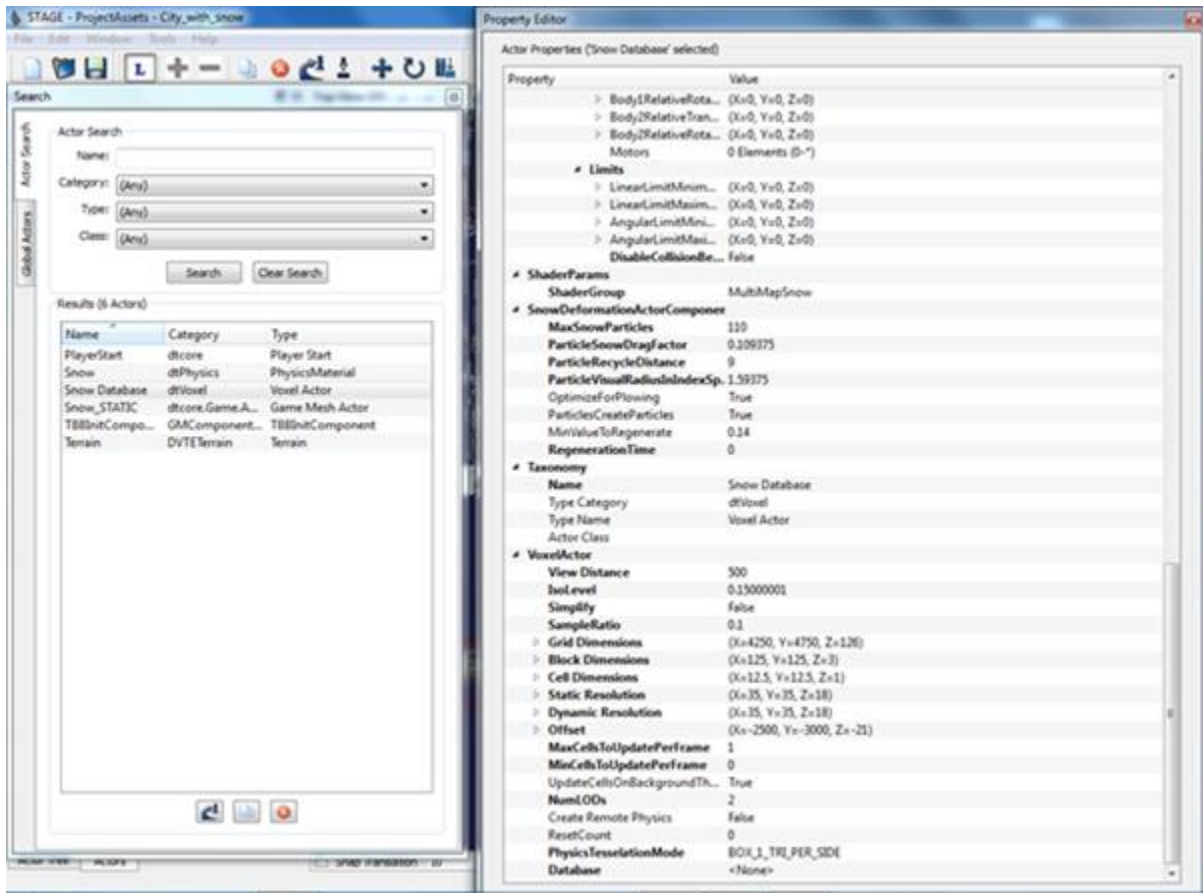
dtPhysics:



dtVoxel:







Dtcore.Game.actors:

The screenshot displays the Unreal Engine interface with two windows open: 'Actor Search' and 'Property Editor'.

Actor Search Window:

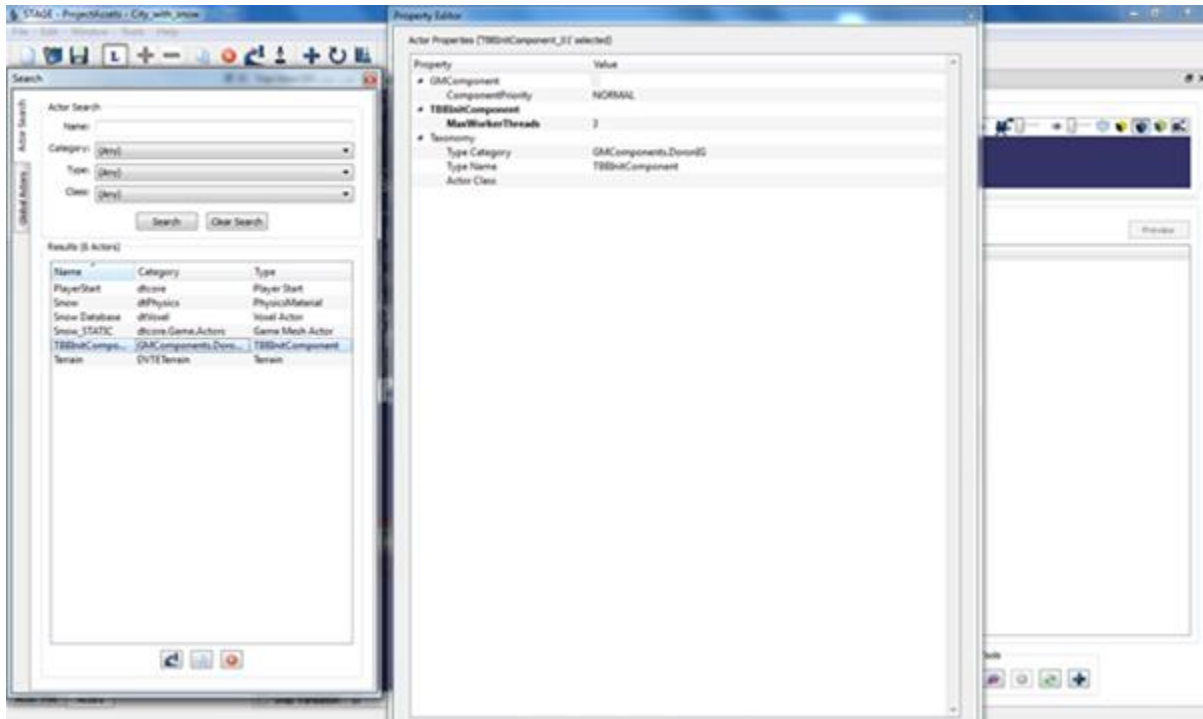
- Search:** Actor Search, Name: (Any), Category: (Any), Type: (Any), Class: (Any). Buttons: Search, Clear Search.
- Results (8 Actors):**

Name	Category	Type
PlayerStart	dtcore	Player Start
Snow	dtPhysics	PhysicsMaterial
Snow Database	dtLevel	Visual Actor
Snow_STATIC	dtcore.Game.actors	Game Mesh Actor
TBBintCompo...	GMComponents.Dorb...	TBBintComponent
Terrain	DVTErrain	Terrain

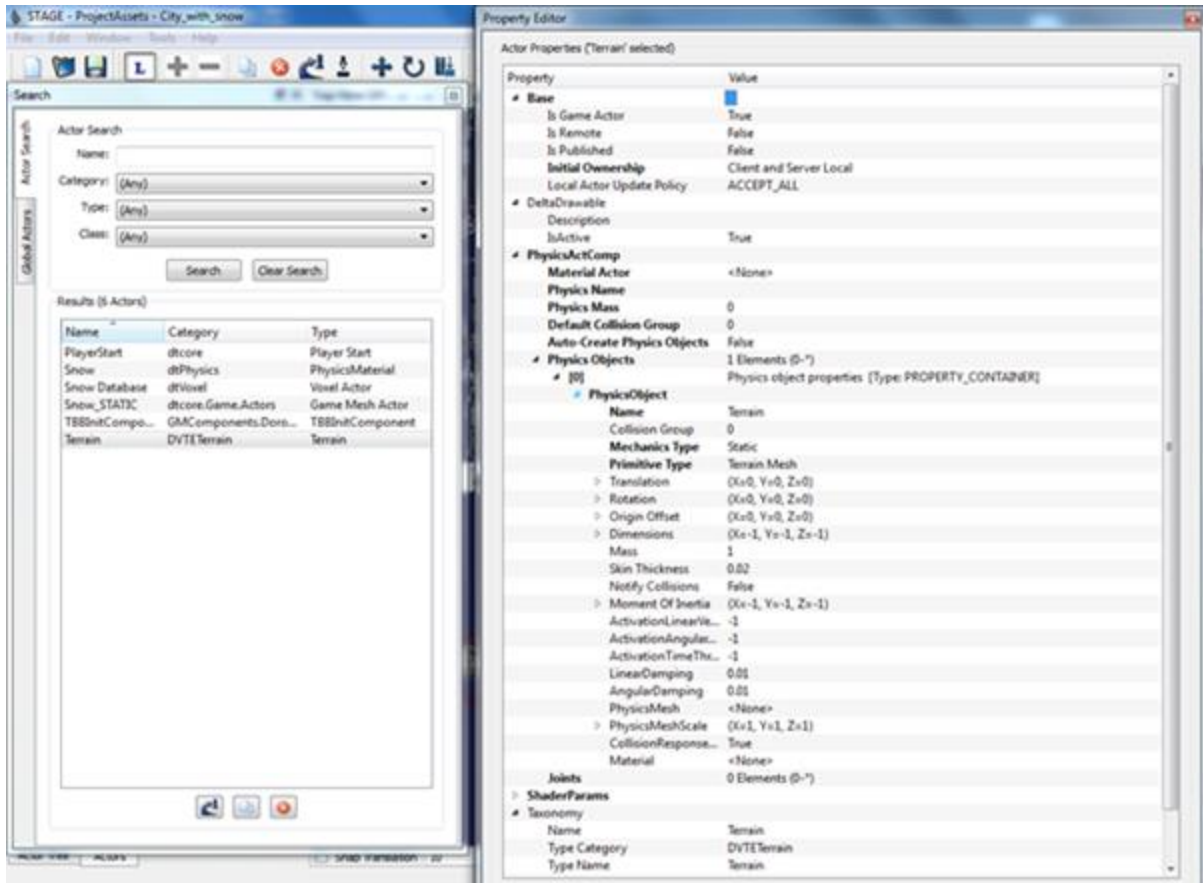
Property Editor Window (Actor Properties [Snow_STATIC selected]):

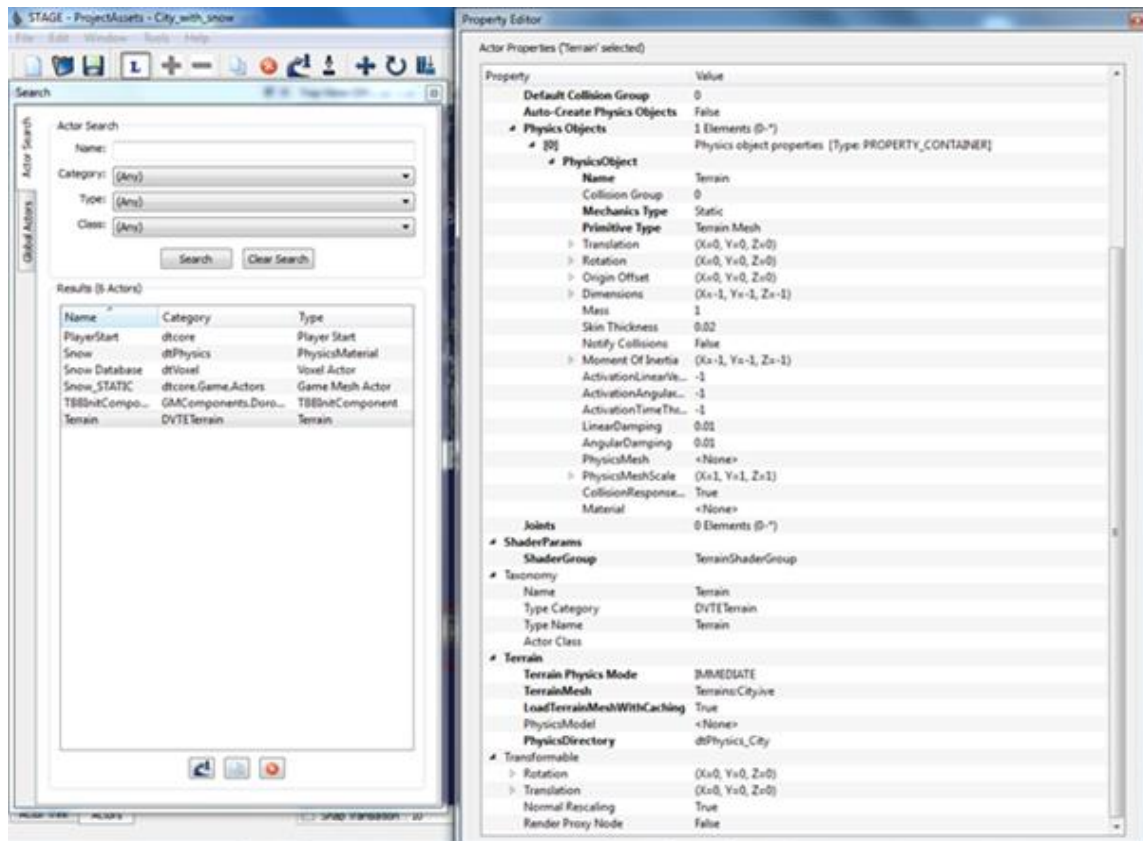
Property	Value
Base	
Is Game Actor	True
Is Remote	False
Is Published	False
Initial Ownership	Client and Server Local
Local Actor Update Policy	ACCEPT_ALL
DeltaDrawable	
Description	
IsActive	True
GameMesh	
Static Mesh	StaticMeshes/Snow/City/Snow_OffRoad.vae
Use Model Cache	True
RecenterGeometry	False
GenerateTangents	False
ShaderParams	
ShaderGroup	MultiMap/Snow_Static
Taxonomy	
Name	Snow_STATIC
Type Category	dtcore.Game.actors
Type Name	Game Mesh Actor
Actor Class	
Transformable	
Rotation	(X=0, Y=0, Z=0)
Translation	(X=0, Y=0, Z=0)
Normal Rescaling	True
Render Proxy Node	False
Scale	(X=1, Y=1, Z=1)
Model Rotation	(X=0, Y=0, Z=0)
Model Translation	(X=0, Y=0, Z=0)

GMComponents.DoronIG:



DVTETerrain:





8.6 C++ source code for parsing VDB Grid information

```
#include <openvdb/openvdb.h>
#include <iostream>
#include <openvdb/tools/ChangeBackground.h>
#include <openvdb/tools/Diagnostics.h>
#include <openvdb/tools/Statistics.h>
#include <openvdb/io/Stream.h>
int getHumanReadableData(std::string absPathStringToVdb) {

    openvdb::initialize();

    // Create a VDB file object.
    openvdb::io::File file(absPathStringToVdb);

    // Open the file. This reads the file header, but not any grids.
    file.open();

    // Loop over all grids in the file and retrieve a shared pointer
    // to the first one. ASSUMPTION
    openvdb::GridBase::Ptr baseGrid;
    for (openvdb::io::File::NameIterator nameIter = file.beginName();
        nameIter != file.endName(); ++nameIter)
    {
        // Read in only the grid we are interested in.
        std::cout << nameIter.gridName() << std::endl;
        baseGrid = file.readGrid(nameIter.gridName());

        // assumption is always of a single grid being present
        break;
    }
    file.close();
}
```



```

        // From the example above, [0] is known to be a FloatGrid,
        // so cast the generic grid pointer to a FloatGrid pointer.
        openvdb::FloatGrid::Ptr          grid          =
openvdb::gridPtrCast<openvdb::FloatGrid>(baseGrid);

        std::cout << " Printing human-readable data for Grid in file: " <<
file.filename() << std::endl;
        grid->print(std::cout, 11);

        std::cout << " Done..." << std::endl;
        return 0;
}

int main()
{
    std::string fpathsnow = "D:\\development\\3dassets\\snow.vdb";
    getHumanReadableData(fpathsnow);
}

```

8.7 Debug Mode Keys for Simulator

[DoronIG Debugging Keys Document Hyperlink](#)

8.8 VDB and IVE files for Doron City

[Doron City Geometries Hyperlink](#)

8.9 VDB Cache for snow.vdb

[VDB Cache Directory Copy](#)

8.10 Doron City Source Geometry for Snow

[Doron City Source Geometry for Snow](#)

8.11 How to compile and build DoronIG and OpenVDBCompiler from source code

[Source Code Build Instructions](#)