Development of a Generic Mine Visualization Tool Using Unity

Timothy J. Orr[1], Brendan D. Macdonald, Stephen R. Iverson, & William R. Hammond
HHS/CDC/NIOSH, Office of Mine Safety and Health Research, Pittsburgh, PA, USA, Mechanical Engineer, 412.386.4403, BMacdonald@cdc.gov.
HHS/CDC/NIOSH, Office of Mine Safety and Health Research, Spokane, WA, USA, Mining Engineer, 509.354.8089, SIverson@cdc.gov.
HHS/CDC/NIOSH, Office of Mine Safety and Health Research, Spokane, WA, USA, Physical Scientist, 509.354.8076, WHammond@cdc.gov.

ABSTRACT

Researchers at NIOSH's Office of Mine Safety and Health Research developed an easy-to-use mine visualization application using the Unity game engine to provide features not available in other software solutions. The tool allows the researchers to quickly build these visualizations and include development and stope geometry, geologic structures, and other spatially-related data. Additionally, the software displays current microseismic events extracted at runtime. Users can walk through the mine from a 1st person perspective and use translucency adjustments to see features beyond the floor, back, and ribs. The software also provides a 'fly' mode to inspect the exterior of the workings and datasets from any perspective.

Because the Unity development environment supports multiplatform deployment, the authors were able to rapidly create variations of the visualization that take advantage of different display and interaction technologies. To date, the software has been deployed via standard desktop-executable, embedded web-browser, Oculus Rift VR headset, and fully immersive multi-screen virtual reality simulators. Future versions of the software will pull all data in a similar manner to allow the end user to rebuild the visualization on demand with the most up-to-date information across all data types. This paper will discuss the methods used to create this visualization tool and plans for future developments.

INTRODUCTION

Mining companies have a wide variety of computer software available to assist their operations in developing mine plans, tracking reserves, and the general visualization of operations. While these software solutions have import/export functions to provide cross-compatibility, few solutions have the capability of integrating supplemental infrastructure location and instrumentation data. Those that do may be limited in the scope and scale of their visualization, or suffer from a lack of viable deployment options. Researchers in other fields have also determined that a sufficient unified visualization tool to combine and display disparate datasets does not exist and opted to create a custom tool to meet their specific needs (Ryabinin and Chuprina 2013).

[1] Corresponding author: HHS/NIOSH/CDC, Office of Mine Safety and Health Research, Virtual Immersive Simulation Laboratory, Computer Engineer, 626 Cochrans Mill Road, Pittsburgh, PA 15236. Phone: +1 412-386-6651. Email: TOrr@cdc.gov

The continued improvements in the visual and simulation fidelity of 3D game engines make these an attractive platform for general visualization development. The emergence of 3D computer games in the late 1980s to early 1990s created a competitive marketplace and strong development community resulting in the creation and support of commercial off-the-shelf game engines (COTS) (Lilley 2009). With the strong economic base of the game industry, developers soon began modifying the technology for content outside of entertainment, such as training and education as well as architectural walkthroughs and virtual reality research (Riddel 1997; Miliano 1999; Jacobson et al. 2005).COTS 3D game engines offer a unique and refined method of 1st person and 3rd person visualization of many different data types. The use of COTS game engines also allows developers to take advantage of the most current computer technology in terms of graphics rendering, physics, and AIs while ensuring hardware and OS compatibility. Therefore, development time can be focused on improving the core functionality of the application rather than reinventing and maintaining basic features, thus providing an elegant solution to the scope and scale problem in mine data visualization.

The objective of this paper is to describe the creation of a generic visualization tool that enables the co-visualization of multi scale geometry, geology, and instrumentation locations with an easy-to-use interface that is conducive for mining research.

METHODS

Researchers at the Office of Mine Safety and Health Research (OMSHR) developed a visualization tool and the workflows required to pull data from several sources to create a data visualization application. A deep vein mine with complex geologic structure was chosen for the initial development. However, the software framework and workflow is generic enough to allow for rapid development of unified data visualizations of other surface or underground mines. In the sections below, the development framework, workflow, and future work is discussed.

Engine Selection

The team selected the commercially available Unity (Unity Technologies; San Francisco, CA) game engine to create this interactive data visualization solution. Unity is one of the current leading engines with respect to lighting, physics, and ease of development, but what makes it most attractive for mining visualization is the current licensing structure. Another key feature of Unity is the ability to create builds with numerous options for deployment including web player and standalone execution on PC, MAC, or Linux OS. With some modification of the UI, builds can be created for mobile devices running Android, iOS, or Windows Phone 8. The Oculus Rift development kit (Oculus VR; Irvine, CA) provides a plugin for Unity to ease the development of builds for this display/interaction device. The free version provides all of the features required to create and deploy such visualizations for non-commercial, in-house purposes.

Data Requirements

Although this software is being developed generically, the data selected for inclusion here are related to the specific ground control problem of the case study mine. In this case, the visualization is being used to relate the interaction of mining development with complex geology on a wide range of scales and across time.

Mine Geometry Data

The mine geometry from the GEOVIA Surpac (Dassault Systemes; Paris, France) data was used as a reference frame for the rest of the data. Stopes, development workings, historic workings, faults, borehole extensometer locations, and topography were selected to be included as they related directly to the ground control problem. Supplemental

information such as ore location could be included if needed. The geometric framework was also enhanced with survey map planes to provide a grounded reference and additional geologic information that would not otherwise be captured in the data.

Photogrammetry Data

Visually relating fine-resolution photogrammetry data to basic mine structures was a key feature driving this project. 3D surface meshes based on photographed stereo pairs were used to create photorealistic models of various fault intersection points in the ramp system using Shapemetrix3D (3GSM GmbH; Graz, Austria). This data was collected from eight sites in the ramp system near major faults on a quarterly basis over the course of a year to document movement in rock structures over time from mining induced changes in stress.

Microseismic Data

Displaying rock burst data from the case study mine's microseismic geophone array was another feature requirement of the visualization tool. SeisVis (ESG Solutions; Kingston, Canada) was used to extract date and time, magnitude, and location information for seismic events that were imported into the visualization to relate these events with stoping and structural movement over time.

Workflow Development

The data sources all use the mine coordinate system which is right-handed, orthogonal, z-up with units of feet. Unity uses a left-handed, orthogonal, y-up coordinate system with units of meters. Workflows to document export/import settings to scale and transform each data type were needed to ease the development process, ensure accuracy, and provide a framework for future automation of data import.

The mine geometry was exported in components from GEOVIA Surpac using the drawing exchange format (DXF) without transformations or altering scale. These components were imported into 3ds Max to unify polygon normals, weld vertices, and apply texture maps. Each component was exported from 3ds Max via Filmbox (FBX) with settings to transform and scale the data prior to import to match the Unity scheme (Figure 1).

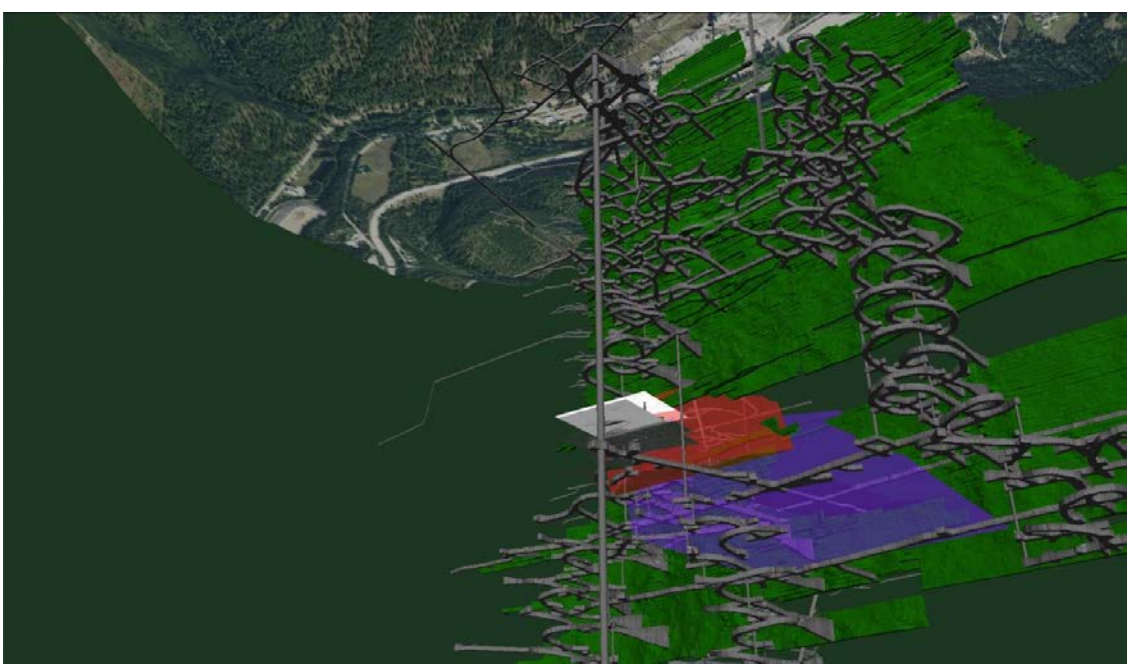

Figure 1. Data from GEOVIA Surpac assembled into the Unity scene viewed from below

Photogrammetry datasets were exported from Shapemetrix using the Object file (OBJ) format for each site and collection date. This format retains the material assignments for texture application and imports directly into Unity. The exported data retains the mine coordinates and scale, and therefore was transformed within Unity. Polygon meshes from this data can be quite dense, so export settings were limited to 100,000 vertices per model. Texture map data was set to the Unity maximum of 4096x4096 pixels to preserve the highest image quality in the final product. With these settings, the quantity of photogrammetry data overloaded the system memory. To avoid this problem without sacrificing data quality, the tool was programmed to only display specific photogrammetry data when directly requested via user interface controls describe below. When the user shifts focus to another dataset the previous data is cleared from memory.

Microseismic data was exported to CSV format from SeisVis for specific time periods. Because these events can occur at an hourly time scale, this data required near real-time updates, so a custom script was developed to import this data into the Unity scene at runtime. The visualized data is extracted from all CSV files placed by the user in a data folder prior to launching the application. Coordinate transformation of this data occurs as it is read from the file(s) each time the visualization tool is launched.

User Interface Development

The user interface (UI) allows the end user to control the perspective and the data displayed. The user can toggle camera controls between 1st person and 3rd person viewpoints. The 1st person view creates the illusion of walking through the mine workings (Figure 2). The 3rd person view enables to user to zoom in and out and 'fly' around the scene to view the exterior of the mine workings (Figure 3). The view controls are located in the upper right hand corner menu; in addition to the 'Fly Camera', a set of onscreen buttons also provides shortcuts to switch the view to one of eight photogrammetry data collection sites.

Figure 2. 1st person view of photogrammetry data and UI

Data visibility can be manipulated through toggling on and off of data, changing translucency, and advancing time. The visibility controls are located in the upper left hand corner menu, where the buttons toggle the visibility of each element and the sliders adjust the translucency. Time can be manipulated by the temporal slider located at the bottom of the screen, where users can also see the name of the currently selected photogrammetry model and date it was collected.

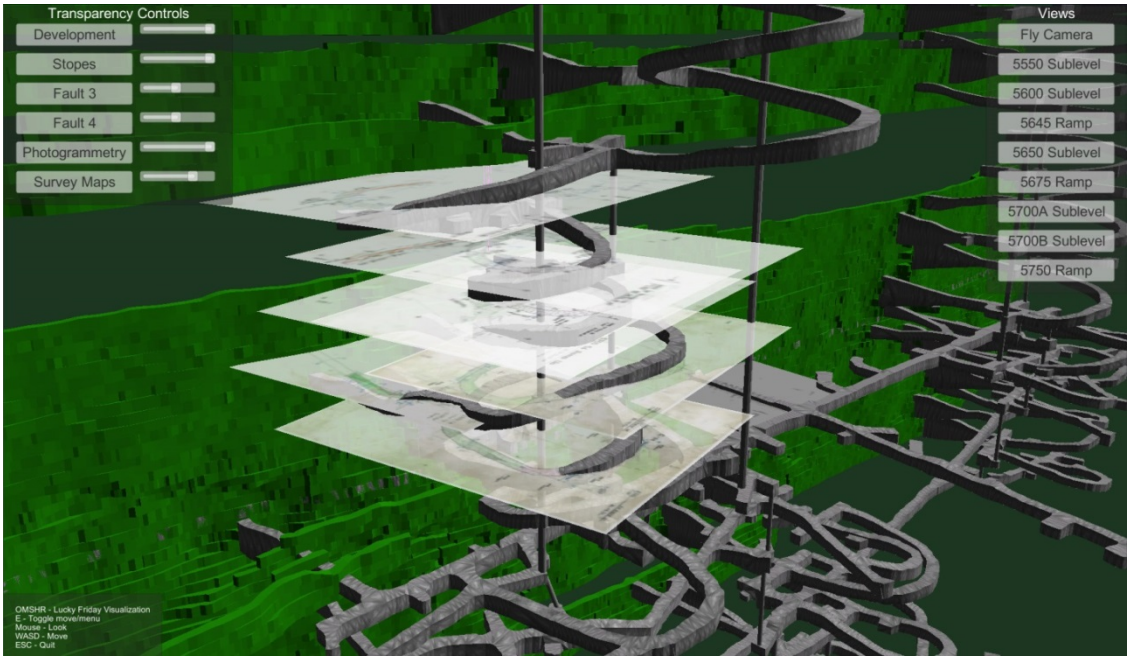

Figure 3. Geologic maps showing the translucency feature

Navigation through the virtual space is accomplished with mouse and keyboard controls using standard key mappings from PC-based video games (e.g., WASD). An Xbox gamepad can also be used for this purpose and serves as the primary controller for the Oculus Rift head-mounted display (HMD) version. Because the HMD has limited screen resolution, this version removes the on-screen menu, but it provides basic functions via controller buttons. An earlier version allowed users to navigate the environment based on gestures mapped to an Xbox 360 Kinect motion sensor, but was abandoned because (1) few end users would have this device, (2) it required significant space to be useful, and (3) it only added limited additional functionality.

Software Builds

The final step in the workflow was to create software builds of the visualization from the assembled data components and UI scripts. As described above, a variety of build options are available native to the Unity game engine. For demonstration purposes, standalone PC and Oculus versions were built and tested. The PC version was also demonstrated using various stereoscopic systems; active 120 Hz laptop display, single stereoscopic projection, and the immersive 360 theater at OMSHR's Virtual Immersive Simulation Laboratory in Pittsburgh, PA.

FUTURE WORK

This workflow and development framework provides engineers and researchers with a simple, inexpensive software solution that fills a gap in the visualization software market. During the development of the example project, additional features and potential improvements were identified. For example, with each update, new data

was added to the visualization which exposed limitations to system memory. This was circumvented with the photogrammetry data using the work-around described above, but other solutions should be explored. One possibility is the optimization of the photogrammetry meshes with 3ds Max using normal maps to represent some of the complexity of the rock structure instead of vertices. However, further exploration should be completed, because it will result in loss of data that could impact engineering analysis. Loading data into memory as the user approaches an area of interest might be a way of making the current data handling scheme more seamless and require less user input, while still maintaining the original fidelity of the data. These techniques could prove valuable as additional data sources are identified for future versions, and new methods for handling large datasets will no doubt be required.

Future work will also be required to handle new data input. Already, researchers and engineers would like to include additional data types such as mineable reserve data, numerical model data, and geologic formations. Assembling the data within Unity manually can become burdensome and prone to error. Therefore, as the list of data sources grows, the need to automate the importation process becomes a priority feature for the next iteration. Ideally, the end user would collect the data to be visualized into folders and select which files to include in the visualization when the application launches.

The HMD version as tested on the Oculus Rift development kit 1 proved to be a popular display option with mine engineers because it provided an immersive, intuitive view. Newer HMDs with higher resolution displays might afford an opportunity to include a menu or overlay that could provide the user with context-sensitive data about the current view. This would offer considerable improvement by providing additional information and functionality to the user. Furthermore, more formal user testing would likely yield numerous interface improvements across all deployment platforms. Some options could include voice controls, non-gamepad hardware controllers, or even multi-user options.

CONCLUSION

The generic visualization tool described here has proven valuable for researchers and mine engineers in understanding and communicating the complex interactions of spatial and temporal data in underground mining operations. Using the groundwork described here, adaptation of the visualization framework should allow users to easily create custom visualizations to include other mine operations, commodities, or facilities.

REFERENCES

Jacobson, J., Le Renard, M., Lugrin, J. L., and Cavazza, M. (2005, June). The CaveUT system: immersive entertainment based on a game engine. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology* (pp. 184-187). ACM.

Lilley, P. (2009, July 21). Doom to Dunia: A Visual History of 3D Game Engines. Retrieved from http://www.maximumpc.com/article/features/doom_dunia_visual_history_3d_game_engines

Miliano, V. (1999, September). Unreality: application of a 3D game engine to enhance the design, visualization and presentation of commercial real estate. In *Proceedings of 1999 International Conference on Virtual Systems and MultiMedia (VSMM'99)* (pp. 508-513).

Riddell, R. (1997, April 1). Doom Goes to War. Retrieved from http://archive.wired.com/wired/archive/5.04/ff_doom.html

Ryabinin, K., and Chuprina, S. (2013). Adaptive Scientific Visualization System for Desktop Computers and Mobile Devices. Procedia Computer Science, 18, 722-731.