Title
The 3D SAQQARA project: Technical Workflow for Creating 3D Environments from 2D Archaeological Data

Authors
Sullivan, Elaine
Cole, Aaron

Publication Date
2017-04-24

Peer reviewed

White Paper

Report ID: 2879637

Application Number: HD-228890-15

Project Director: Elaine A. Sullivan

Institution: University of California, Santa Cruz

Reporting Period: 6/1/2015-12/31/2016

Report Due: 3/31/2017

Date Submitted: 4/24/2017

The 3D SAQQARA project:
Technical Workflow for Creating 3D Environments from 2D Archaeological Data

Author: Elaine A. Sullivan and Aaron Cole

Contact: easulliv@ucsc.edu

Final Project White Paper, NEH Digital Humanities Start-Up Grant

Grant Number: HD- 228890-15

Project Title: 3D Saqqara: reconstructing landscape and meaning at an ancient Egyptian site

Project Director: Elaine A. Sullivan

Grantee Institution: University of California, Santa Cruz

Date Submitted: April 24, 2017

Table of Contents

Abstract

The *3D Saqqara* project addresses ancient ritual landscape from a unique perspective, utilizing emerging 3D technologies to examine development at the complex, multi-period archaeological site of Saqqara, Egypt. Using a 3D + Geographical Information System (GIS) reconstruction model of the ancient Egyptian necropolis of Saqqara (covering the Pharaonic period, circa 2950-350 BCE), the project 'peels away' layers of later construction and environmental modification at the site, re-imagining the ancient site a series of time-slices. Harnessing the temporal layering abilities of digital environments, it demonstrates how 3D modeling allows archaeologists to approach questions of meaning and human experience in now-disappeared landscapes in new ways, visualizing change over time from a human point-of-view.

Project Contributors

Project Director

Elaine A Sullivan, Assistant Professor, UC Santa Cruz

VSim Technologist
Lisa Snyder, GIS/Visualization specialist, Institute for Digital Research and Education, UCLA

GIS Specialist Support
Eric Fries, (former) graduate student, UCLA
Max Van Rensselaer, (former) undergraduate, UC Santa Cruz
Aaron Cole, Senior GIS Analyst, Center for Integrated Spatial Research, UC Santa Cruz
Barry Nickel, Director of the Center for Integrated Spatial Research, UC Santa Cruz

CityEngine Workflow developers and Assistance
Marie Saldana, (former) graduate student, UCLA
Maxime Israel, University of Massachusetts Amherst

3D Modelers
Aria Klucewicz, (former) undergraduate, UCLA
Cori Hoover, (former) undergraduate, UC Santa Cruz
Sophie Short, (former) undergraduate, UC Santa Cruz
Victoria Schniedewind, (former) undergraduate, UC Berkeley

Data Contribution
Digital monument footprint files of the area of North and Central Saqqara were created by the University of Pisa (Italy) and updated (2013) by Dr. Emanuele Brienza (GIS Specialist, University of Enna, Sicily).

1m topographic line files of the area of Giza, 5m topographic line files of the area of South Saqqara and the section of Memphis between Abusir and Giza, and the original model for the pyramid of Djoser were provided by Dr. Mark Lehner, Director of AERA.

Key topographic maps of the Memphite area were shared by Dr. Ana Tavares.

Image and Data Credits
The superstructure of the Dynasty 1 *mastaba* tomb #3505 was textured using images from: Emery, W. (1958). Great tombs of the First Dynasty: excavations at Saqqara. vol. III, London, Egypt Exploration Society. Plates 6,7,8 (color plates). Images courtesy of the Egypt Exploration Society.

Satellite images courtesy of the DigitalGlobe Foundation.

Project Funding

The *3D Saqqara* project has been supported by:

NEH Digital Humanities Start-Up Grant, 2015-2016

ACLS Digital Innovation Fellowship award, 2012-2013

W.M. Keck Foundation, as part of UCLA's Digital Cultural Mapping program, 2011-2012

University of California Santa Cruz, Faculty Research Grants 2014-2015; 2016-2017

Project Goals

In 2015, the *3D Saqqara* project received NEH Digital Humanities Start-Up Level II funding to complete the construction of a 3D + Geographic Information Systems (GIS) model of the archaeological site of Saqqara, Egypt (Latitude 29.872513; Longitude 31.205370). The goal of this funded project is to create, publish, and distribute a geo-temporal reconstruction model of the site for interactive use by scholars and students.

The ancient Egyptian cemetery of Saqqara is a designated UNESCO World Heritage Site, the location of the first monumental stone structure in the world, and one of Egypt's most popular tourist destinations, visited by some twenty-thousand visitors a week before the Egyptian Revolution in 2011. The cemetery served as a burial place and cult center for kings, administrators, royal family members, artists, and non-elites over more than 3000 years of almost continual use. Covering some 700 hectares (7 square kilometers), pyramids, *mastaba* tombs, and huge funerary enclosures are still visible at the site today.

Figure 1. Satellite view of north, central and south Saqqara
Imagery courtesy Digital Globe Foundation

Despite almost two centuries of intense archaeological excavation, few synthetic investigations have considered visibility in the sacred landscape of this vast and important site across space and time. Modern archaeological field interpretation on questions concerning visibility are hampered by the significant temporal change at the site, as well as the degraded nature of many of the original monuments. Such investigations are especially difficult for the earliest phases of the site, whose monuments deflated or were intentionally re-used or disassembled in ancient times. As well, dramatic environmental change in the Memphite region (including the shifting of the Nile river, the rising of the alluvial plain, and the influx of desert sands from the west) has altered basic elements of the larger landscape since ancient times. Visual and spatial relationships today in the region thus do not accurately or fully reflect those relationships in the ancient past. Examining visuality and visual links within and across these ritual spaces can thus only be accomplished through a reimagining of these lost landscapes.

Figure 2. The deflated pyramid of Dynasty 5 king Userkaf at Saqqara in modern times, photo by the author, looking south

The goal of the project therefore focuses on the integration of archaeological and environmental data in order to visualize aspects of the historic site no longer observable in the field today. With the aim of digital 'reconstruction,' the model incorporates scholarly hypotheses on the original form and appearance of ancient monumental structures, along with archaeological and geographical knowledge about the contemporary natural and physical environment. The creation of the Saqqara model to-date has encompassed six major steps:

1. Establishing monument footprints for Central, North, and South Saqqara, as well as major monuments at the neighboring archaeological sites of Abusir, Abu Ghurab, Helwan, Dahshur, Giza, Abu Roash, Heliopolis and Memphis in a GIS system with appropriate attribute information

2. Creating terrains and elevation models to reflect ancient 'ground horizon' at Central and North Saqqara; building low-resolution terrains representing the entire Memphite ritual zone based on existing topographic maps and satellite

imagery

3. Modeling the superstructures of archaeologically-documented monuments at Saqqara and select monuments at neighboring sites to-scale using a 3D software program, approximating ancient height and materials via texture mapping

4. Combining 2D GIS information, 3D models, and terrain data in an urban simulation software program; generating procedural-based models from 2D GIS information in that program

5. Importing the resulting data model back into GIS software for further spatial analysis with GIS toolkits

6. Importing the resulting data model of Central, North and South Saqqara into the VSim real-time navigator for first-person navigation within the 3D space

Technical Description

This White Paper presents a workflow for how the *3D Saqqara* project transformed 2D and analog archaeological data into 3D representations of ancient built and natural environments, maintaining the geo-spatial coordinate system of GIS and allowing for both quantitative and qualitative visual analysis. A basic outline of the procedure followed is listed here to guide others interested in utilizing these programs. Specific technological issues encountered or overcome are highlighted (**) to help others avoid potential pitfalls in the process.

I. Geographic Information Systems (GIS)

The project used ESRI's ArcGIS suite of software programs (ArcMap, ArcScene, ArcCatalog) for GIS processing and analysis. Working in a GIS allowed for the integration of many disparate datasets, including those from different sources, different time periods, and at different scales, while retaining accurate relative geographic coordinates. The data was then moved into ESRI's urban simulation program, CityEngine, for 3D visualization capabilities.[1] The planned migration of the GIS data into CityEngine had impact on the form of the GIS data from the very beginning stages of the project.

Monument footprints

2D GIS data (in shapefile/SHP format) of the area of North and Central Saqqara was shared with the project (see Data Contribution section above). Information on the initial collection of the majority of that data was published by the University of Pisa.[2] The project created its own GIS data for additional monuments in the Saqqara and Memphis zone. Primarily, we accomplished this by geo-referencing published archaeological plans on high-resolution satellite imagery, then digitizing monument ground plans or 'footprints' in the GIS. Plans were at multiple scales, and locating monuments often required the layering of multiple plans together. 2D polygons were designed (or modified in the case of the original Pisa data) to represent the full footprint of the structure, including enclosure walls or additional buildings that would be built in 3D. This was critical for later aligning the 3D models onto the 2D footprints, and for creating clean visual joins between the constructed terrain layers and the monuments (discussed below).

The attribute structure of the original data (published by the University of Pisa) was initially maintained and expanded to include additional information relevant to this project. We added attributes to document the source of our 2D data, including references to the published plans or basemaps. Eventually, we stripped out much of the original published attributes, as many of those data elements focused on conservation and were unrelated to this project. We maintained the Pisa system and data for dating (period,

---

[1] http://www.esri.com/software/cityengine. During the full course of the project, 2012-2017, we utilized CityEngine 2012, 2014, and 2015. The latest version, 2016, has recently launched.

[2] The University of Pisa, ed. 2003. *The North Saqqara archaeological site handbook for the environmental risk analysis*, *Progetti 1*. Pisa: Plus.

Dynasty, reign of king), monument naming/numbering, and original discoverer (or publisher) of the monument.

**All GIS data was created in (or converted to) the projected coordinate system WGS 1984 UTM zone 36N.

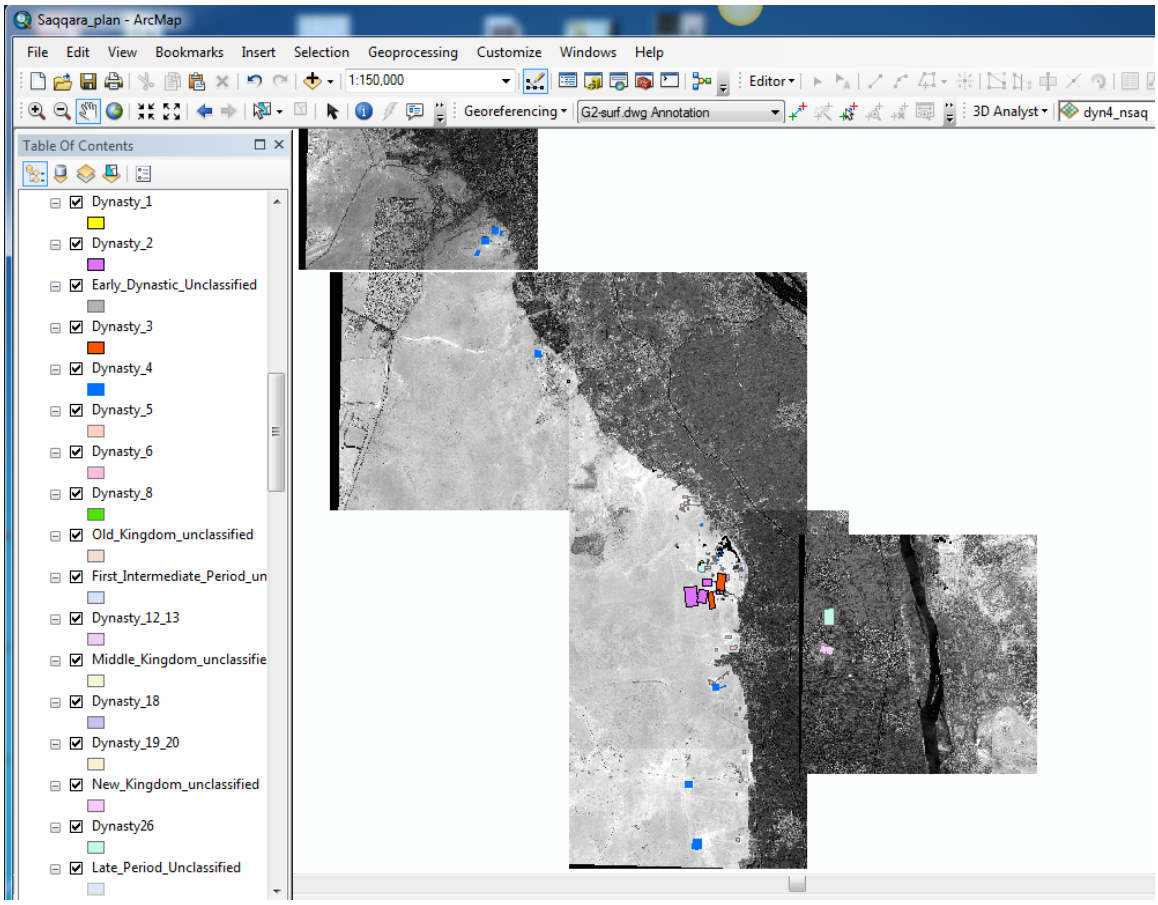

Figure 3. GIS map of Memphite necropolis, spanning Giza (N) to Dahshur (S) overlaid on Digital Globe satellite imagery (courtesy Digital Globe Foundation), in ESRI ArcMap

Terrains

The project utilized the Egyptian Ministry of Housing and Reconstruction (MHR) Survey of Egypt series ('Le Caire') topographic line maps to build our terrain layers for CityEngine. The 1:5000 resolution maps were created in 1977 from aerial photographs. In ArcMap, we hand-digitized the 1m topographic lines into the GIS. We experimented with using automated systems (ArcScan) to scan and build these lines, but found the resulting products inadequate, as they needed a great deal of repairing and did not save time through the automation. We next removed modern features from the data, as well as contour lines that clearly represented the height of monument superstructures, as we were attempting to recreate ground level around the monuments (monument height would be represented through the 3D models). A master copy of this final cleaned dataset was saved. We created separate datasets for the areas that each MHR map covered, as these

regions were very large and we wanted the ability to create smaller terrain zones that could be turned on and off, as opposed to one very large terrain layer. Central Saqqara/Abusir was included in one dataset, South Saqqara, Dahshur, Memphis/Helwan, etc. in separate datasets.

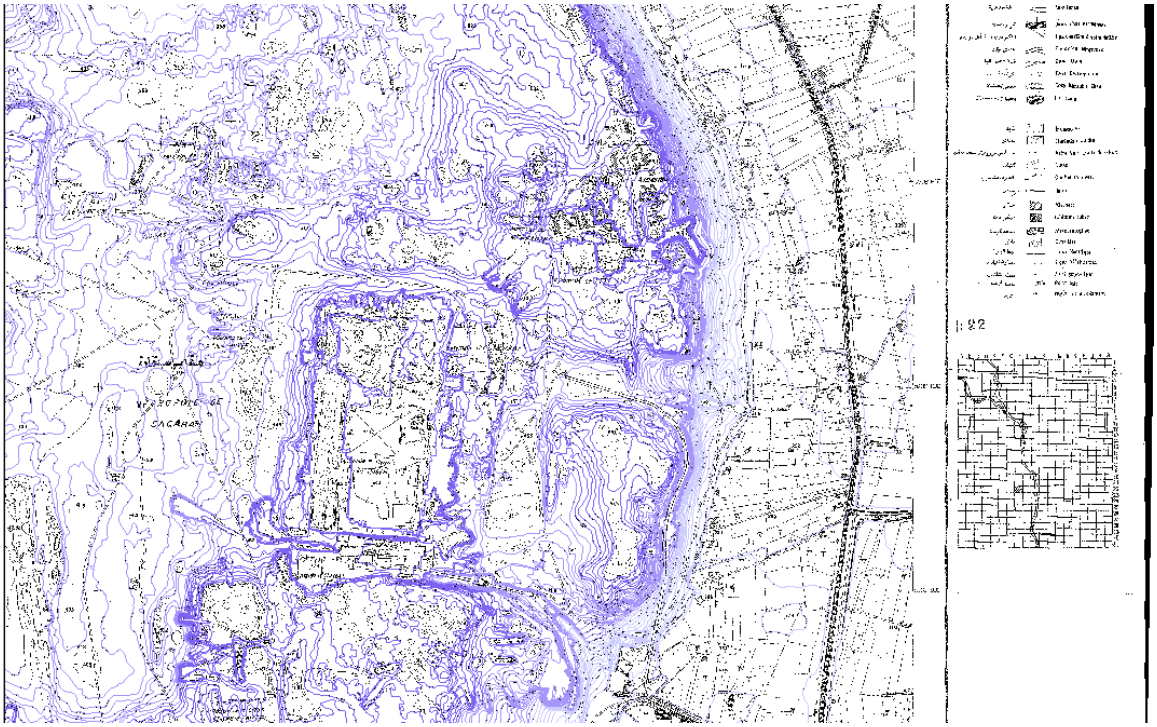

Figure 4. Digitizing topographic lines (purple) over the MHR maps (black & white) in ArcMap

We then created four copies of the master topographic line datasets for the core Saqqara area (Central Saqqara/Abusir, South Saqqara, Memphis/Helwan),[3] and assigned each copy a temporal span (Dynasty 1-4, Dynasty 5-17, Dynasty 18-25, Dynasty 26-30). Each of these was individually modified to match known heights above sea level (ASL) for 'ground horizon' in that period. These historic heights were gleaned from published materials, which often record information like building floor levels, doorway thresholds, or height of pathways between structures. When such ancient 'ground horizons' could be determined, topographic lines that in modern times are much higher were lowered to reflect ancient levels.

The topographic line datasets were then converted into raster datasets using the ArcGIS 'Topo-to-Raster' tool, which creates 'hydrologically correct' Digital Elevation Models (DEMs) from contour lines (http://pro.arcgis.com/en/pro-app/tool-reference/3d-analyst/how-topo-to-raster-works.htmn) For best visualization in CityEngine, we then converted these raster terrains into TIFF format.

[3] Note that the project made lower resolution terrains (usually 5-10m contour lines) for other areas in the Memphite zone, including around Zawiyet el-Aryan, Giza, Abu Roash, Heliopolis, and Dahshur, but these were not adjusted to show temporal change over time. The project has focused on making a higher resolution model with temporal change only for the core Saqqara area.

II. 3D Modeling of Custom Monuments (SketchUp Pro)

Modeling Process

The team (including Sullivan and student modelers) constructed individual to-scale models of more than 100 of the well-documented monuments at Saqqara and the surrounding sites. Models were limited to exterior superstructure features, as interior spaces were not relevant to the research question of visibility in the larger landscape. Archaeological plans scanned from excavation reports were imported into the program Trimble SketchUp Pro[4] and scaled, and models were designed directly on these plans. Published architectural drawings or textual descriptions of suggested original heights were utilized to reconstruct the Z (height) coordinate value. Additional details were based on modern photographs of the site.

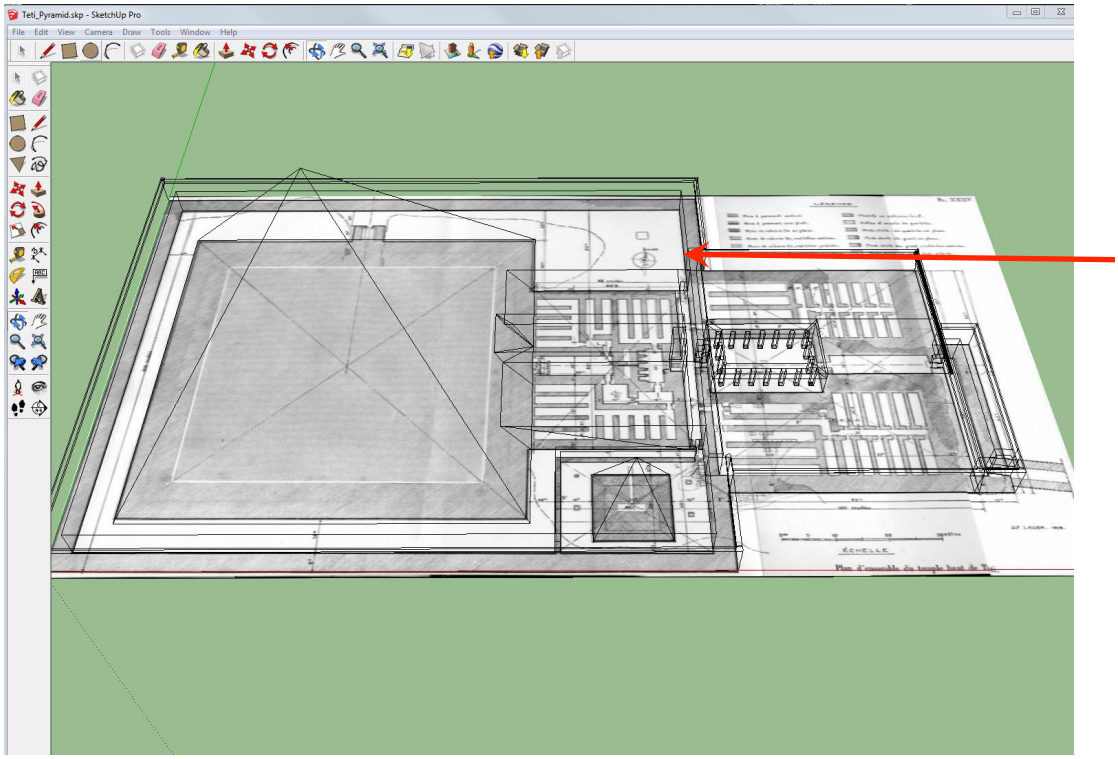

Figure 5. Modeling 3D monuments on scaled archaeological plans, SketchUp Pro

[4] https://www.sketchup.com/

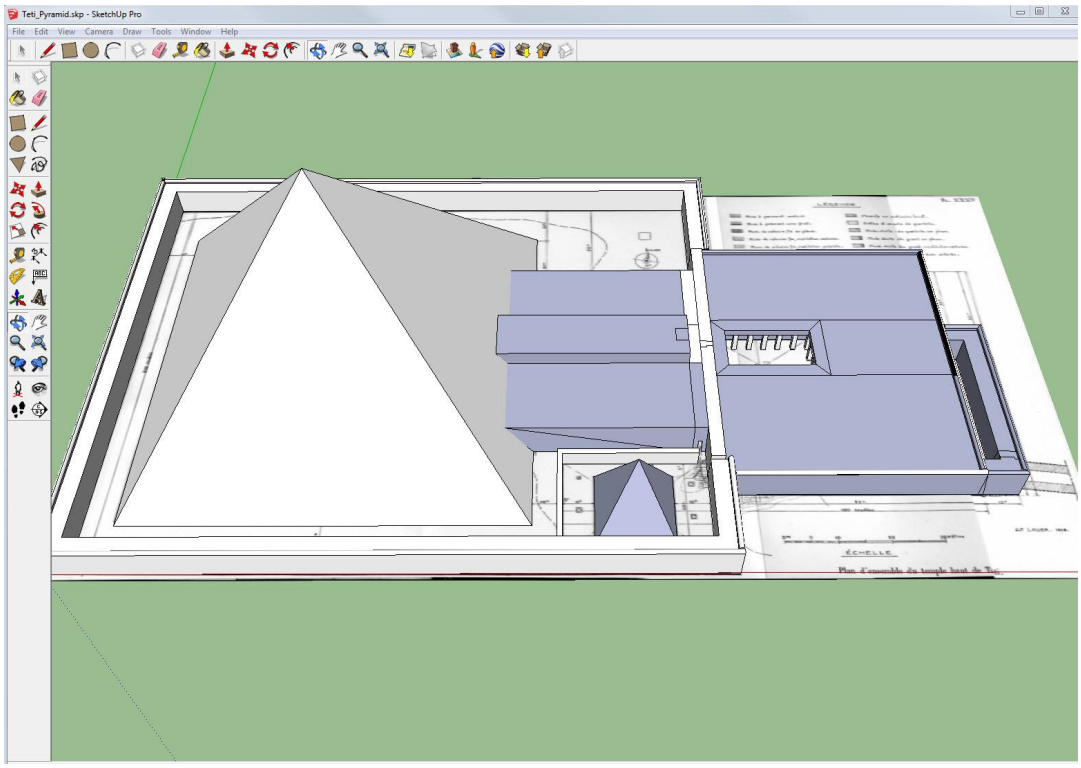

Figure 6. Modeling a scaled 3D massing model, SketchUp Pro

Metadata and Paradata

Metadata and paradata on all decisions made in the 3D modeling process were kept in a group collaborative document using Google Drive. This document was structured as a spreadsheet so that the modeler had a series of clear categories to fill out for each custom model (drop-down boxes for some sections helped to keep naming standardized). Modelers filled out paradata *during* the modeling process, as post-modeling documentation proved to be problematic and resulted in incomplete documentation. Close documentation of metadata and paradata for 3D model construction was especially important, as none of this information is embedded in the 3D SketchUp files themselves (unlike in GIS files, where such information can be directly linked to individual files using attributes and/or metadata in ArcCatalog).

Documentation categories included the following:

- Monument # (excavation or publication number) or Name (commonly accepted monument 'names' in Egyptology literature, i.e. the 'step pyramid of Djoser')
- Temporal period (following the Egyptian Dynastic system, by *Phase* and *Dynasty*, following the chronology guidelines of the UCLA online Encyclopedia of Egyptology, UEE: http://uee.ucla.edu/chronology/)
- Orientation
- Exterior material (original)
- Architectural superstructure type (*mastaba*, pyramid, tomb chapel, palace (in one case refined as 'royal resthouse' for this project), pylon, temple (building),

enclosure (wall), ramp, balustrade, catacomb, obelisk, based on the Getty Research Institute Art & Architecture thesaurus vocabulary: http://www.getty.edu/research/tools/vocabularies/aat/)

- Height in meters (superstructure original/reconstructed maximum)
- Length in meters (superstructure original/reconstructed maximum)
- Width in meters (superstructure original/reconstructed maximum)
- Exterior decoration (original)
- Sources used as basis for model and texturing (short reference to the published source by author last name and date, later linked to our full bibliography)
- Additional information on model construction (specific information from published sources related to model build)
- Enclosure wall dimensions in meters (superstructure original/reconstructed at maximum)
- Name of 3D modeler
- Interpretive choices in reconstruction (the 'paradata' explaining interpretation made for reconstruction)
- Model file name
- Software used for model build
- Date of model construction or update
- Copyright
- Site (Saqqara, Abusir, Dahshur, Helwan, South Saqqara, Zawiyet el-Aryan, Giza, Abu Ghorab)

Once all the base 3D modeling was complete, the spreadsheet with this data was saved and kept to later link with the original modeled files (see discussion below). At the end of the project, the OBJ files of each custom modeled 3D monument will be individually available for download by the public, and the above information will be attached to each file to provide full documentation.

An abridged version of this data was intended to be included in online publication of the model, so a copy of selected model metadata was also exported as CSV files, and the data was joined to the 2D GIS files in ArcMap (using the Excel to Table conversion tool: http://pro.arcgis.com/en/pro-app/tool-reference/conversion/excel-to-table.htm). For this project, 2D GIS files were separated into temporal groups (Dynasty 1, Dynasty 2, Dynasty 3, etc.) so the spreadsheet was separated accordingly into individual CSV files and joined with the correct temporal GIS data. This process allowed us to join important aspects of the metadata and paradata from the modeling process with the original attributes in the GIS. The GIS 2D shape footprints at this point then included all the information on the monument that we decided would be most critical for a user to identify and understand the model. See below for how we connected the 3D models to these files.

Texture Mapping

Texture mapping was based on published excavation descriptions and modern photographs of the site. A series of custom schematic textures (ie. 'limestone casing

blocks,' 'mud-brick walls,' 'red granite') were created using modern color photographs taken by the author at ancient Egyptian sites (primarily in the Memphis archaeological area, but some also in Luxor). These textures were utilized at multiple monuments to create a consistent visual 'vocabulary' for the model viewer/user. When original brick or stone size was able to be determined, modelers intentionally stretched textures to visually approximate those original dimensions. Exterior surfaces textures were selected to best represent the original appearance of the monument at time of construction/original use.

The use of schematic textures was especially important, as many of the monuments at Saqqara today are extremely deflated or greatly altered in appearance from ancient times. The white limestone casing stones that originally enclosed the monumental pyramids, for example, were later removed from the site, exposing the much darker interior stone cores, and thus texturing these structures with imagery taken from their modern state would not replicate original appearance.

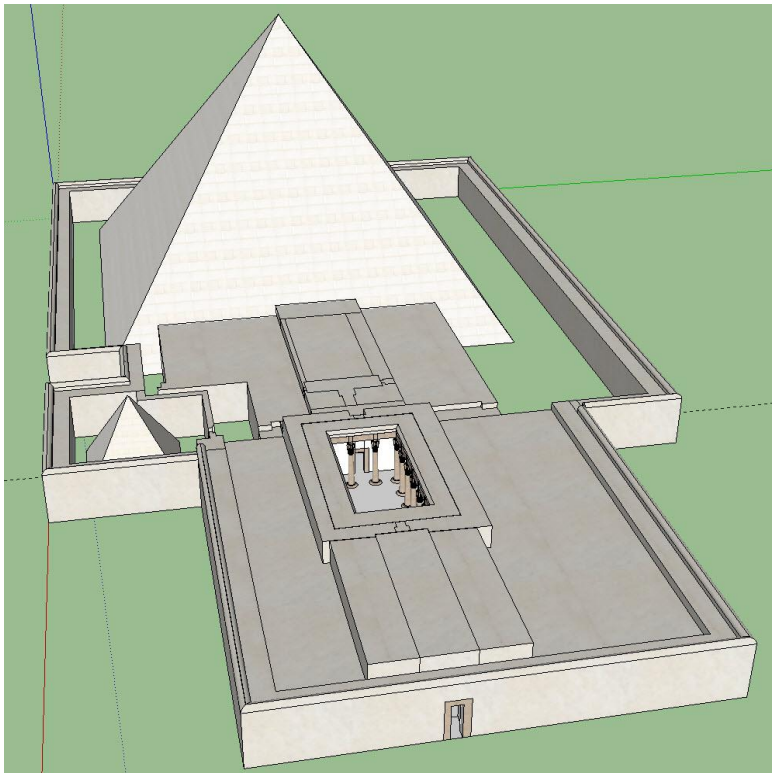

Figure 7. 3D massing model with textures added, SketchUp Pro

**It was critical to texture all potentially visible surfaces of a monument with custom textures (usually JPG or TIF files), as the default SketchUp texture colors are not included in the export process (see below) and thus 'disappear' from the models when imported and converted into certain file types in the CityEngine program. The default color textures initially appear successfully when 3D models are imported to CityEngine as OBJ files, but once converted into shapes to join with 2D GIS data (see section below on *Joining 2D & 3D datasets)*, the default SketchUp textures disappeared and the models turned dark grey.

Export

Completed models with identifying file names (stripped of original plans) were exported as OBJ files and then placed in the appropriate file structure to be imported into the CityEngine software program (see below).

**Monuments with multiple structures that covered large areas, such as the pyramid complex of king Unas (including a pyramid, memorial temple, causeway, and valley temple) were split into a series of separate files to be exported individually. This was crucial to placement in the CityEngine software program, as adjusting each piece separately to the terrains was critical for a close visual join with the underlying landscape.

III. Combining Data into 3D Simulator (CityEngine)

Importing 2D GIS Data

The CityEngine software platform allows for simple import of ESRI ArcGIS shapefiles. Following the strict file data structure laid out by CityEngine, GIS shapefiles were copied and pasted into the appropriate folder ('assets') and then individually imported into the program. All GIS files must be converted into projected coordinate systems before import.

Upon loading into CityEngine, all 2D data maintains its horizontal position based on the coordinate system referenced by the data, but is placed on the 0 elevation plane (in CityEngine, this is the 'Y' vertex). In order to reflect elevation ASL, each polygon was moved individually or as a group (using the 'Move' tool) to a designated height/elevation meters above-sea-level (ASL). For this project, polygons were assigned ASL height based on known 'ground horizon' for each structure as documented in archaeological publications. Note that in many cases, this differed dramatically (sometimes many meters) from current site elevation in modern times.

Importing 3D Custom Models

Following the strict file data structure laid out by CityEngine, the OBJ custom model files (along with their accompanying MTL and corresponding folder with texture files) were copied and pasted into the appropriate folder ('assets') and then individually imported into the program. Since these models are not yet georeferenced, they must be individually moved and raised to their correct elevation ASL in the 3D viewer. We carefully positioned each one based on the 2D monument footprint brought in as part of the previous steps.

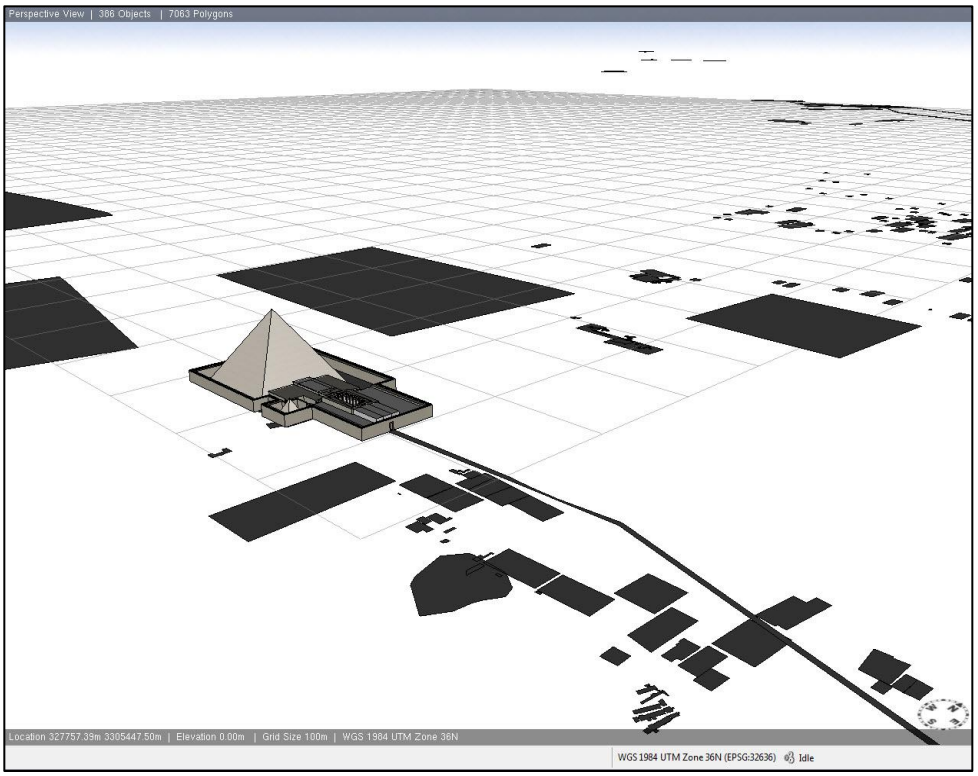

Figure 8. Importing 3D monuments and raising them to correct height ASL, CityEngine

Procedural Modeling

Hundreds of *mastaba* tombs dating to the Early Dynastic Period and Old Kingdom at Saqqara and the neighboring site of Helwan were identified and mapped by excavators in the late 19th and early- to mid- 20th century. Many of these monuments are no long visible at the sites and/or have not been studied in any detail. In order to include these important monuments (as well as others with poor documentation) in the model, we used the procedural modeling functions of CityEngine. Using 'Computer Generated Architecture' (CGA) rule based modeling, basic 3D massing models were created from the 2D data. We modeled a series of simple, 'generic' building forms that represented typical tomb types at different periods as OBJ files, and then used CGA rules to customize them to each footprint, based on the length and width of that polygon.

Procedurally modeled structures included round-topped *mastabas*, flat-topped *mastabas* (also with niches), mud-brick *mastabas*, *mastabas* with elaborated niched facades (the 'palace facade'), collapsing pyramids, and rectangular enclosures that protected shaft tombs. The height limits used for the rule files were based on extant examples of tombs with better documentation or preservation, or estimates by archaeologists. It is important that users understand then that all procedural models were a generalized form that does not reflect unique aspects of a monument's original appearance; these create approximations of original forms that accurately reflect length and width, but only estimate original height.

An example of our the CityEngine CGA rule script, for *mastabas* with sloped sides, a flat top and a niche on one side used in the model for Dynasty 4-5 tombs:

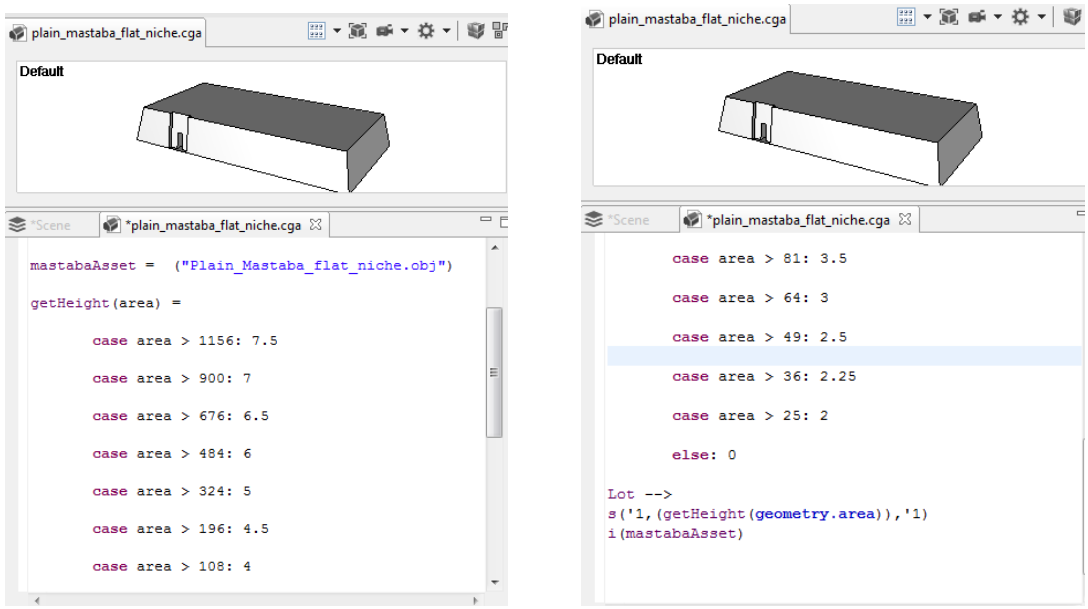

Figure 9. Procedurally modeled *mastaba* tomb in CityEngine, showing rule file (CGA) and linked massing model (OBJ)

Importing Terrain Layers

The four DEM tiffs were placed in the appropriate file structure to be imported into the CityEngine program ('maps'). They were then imported and displayed as terrain layers in the 3D viewer using CityEngine's 'Terrain Import' function. For high detail, we adjusted terrain layer resolution to 2049x2049. The adjusted 3D models and 2D polygons lay at or near the height of the terrain layer at this point, demonstrating the success of our integration of the individual data elements.

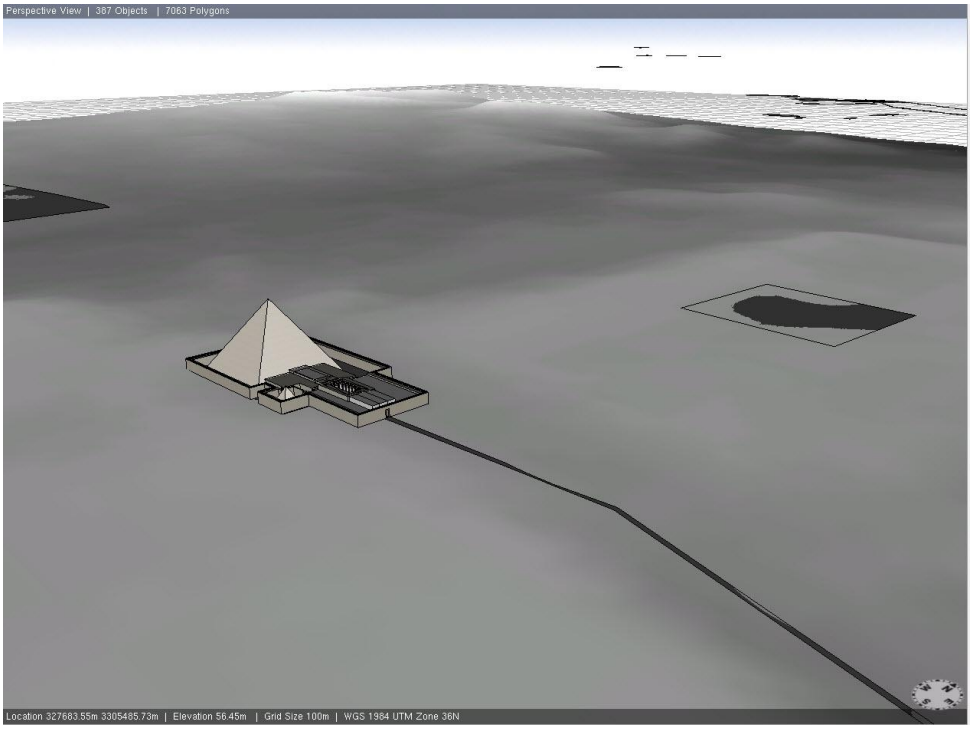

Figure 10. Terrain imported into CityEngine

Adjusting Terrain Layers

CityEngine allows the 2D polygons (as well as the procedural models resulting from 2D polygons) and the terrain layer data to interact with each other, in order to create a close visual connection between polygons and terrain (to prevent 'floating' architecture). Turning on both the terrain layers and the 2D footprints, we selected each footprint and used the 'Align Terrain to Shapes' tool to slightly lower or raise the terrain layers around the footprint to create a close visual connection. For monuments where historic heights were known, terrain layers could be adjusted to reflect that knowledge. When original monument height was unknown, we instead could use the 'Align Shapes to Terrain' tool, which shifted the 2D polygon up or down to match the terrain layer. This feature of the program allowed us to utilize known 'ground horizons' when we had them, but terrain elevation data when we such information was lacking.

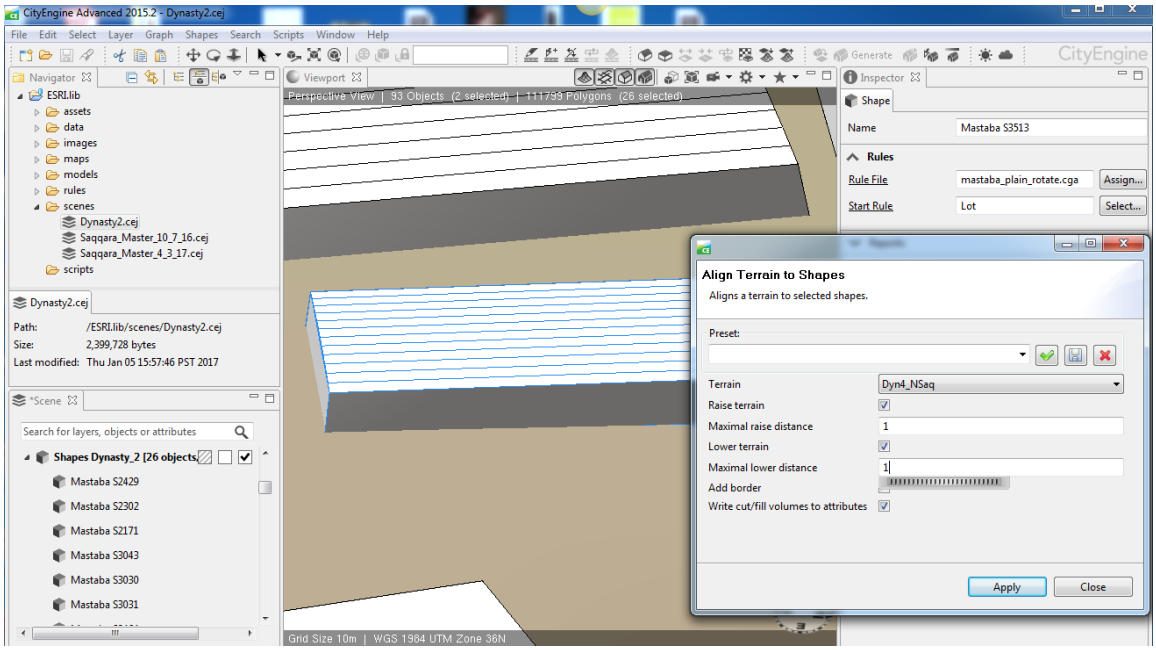

Figure 11. Aligning terrain to the procedural model in CityEngine

Joining 2D & 3D datasets

The 2D polygon footprints imported from the GIS contained all the original attribute information for each monument. Additional metadata and paradata from the modeling process (see above) were amended onto the attributes through a join process. The resulting 2D data then was joined with the 3D custom models (created in SketchUp and lacking metadata) imported separately (as OBJ files) into the program.

**Before beginning the join process, it is critical to copy and create a duplicate shape of each of the adjusted 2D polygon files within the CityEngine scene (we renamed them to make sure it was clear these were different shapes). Once the 2D shapes and custom-created 3D OBJ files are joined, the CityEngine terrain tools (such as 'Adjust Terrain to Shapes') no longer recognize the 2D monument 'footprints,' as all the points and lines (from the 2D and 3D data) are instead recognized by the program as part of that shape. Duplicates of the correctly adjusted 2D shapes allow the creator to continue to modify the visual connection between monuments and terrains, which is one of the most difficult and time-consuming aspects of the process.

**Before converting 3D OBJ files into CityEngine 'Shapes' and joining 2D and 3D data, we exported the 3D monument files as collada/DAE files for use in visibility analysis in ArcScene (see below for a description of this process). We kept a duplicate of the pre-joined CityEngine Scene in case joining the monuments later created problems for that workflow.

For this project, we had multiple terrain layers based on chronology, and wanted to make different adjustments between each terrain layer and shapes/monuments depending on

time-period, so the duplication of the original 2D shapes as ‘footprints’ of the monuments was critical. We could turn on the duplicate shapes when we wanted to re-adjust the visual relationship between monuments and terrains, without having to re-import and adjust 2D ‘footprints.’

Once that was accomplished, we joined the 2D monument footprints with the custom 3D models. In CityEngine, we selected the appropriate 3D monument and chose ‘Shapes > Convert Models to Shapes.’ Locating the newly created ‘Shape’ from the original model, we selected it, then selected the corresponding 2D shape ‘footprint’ and selected it as well. We joined these two files together using ‘Shapes > Combine Shapes.’ The result was a 3D model then contained all the appropriate metadata attached from the GIS attributes. If desired, the monument footprints could now be deleted out (as long as the duplicate 2D footprints are saved somewhere else in the scene).

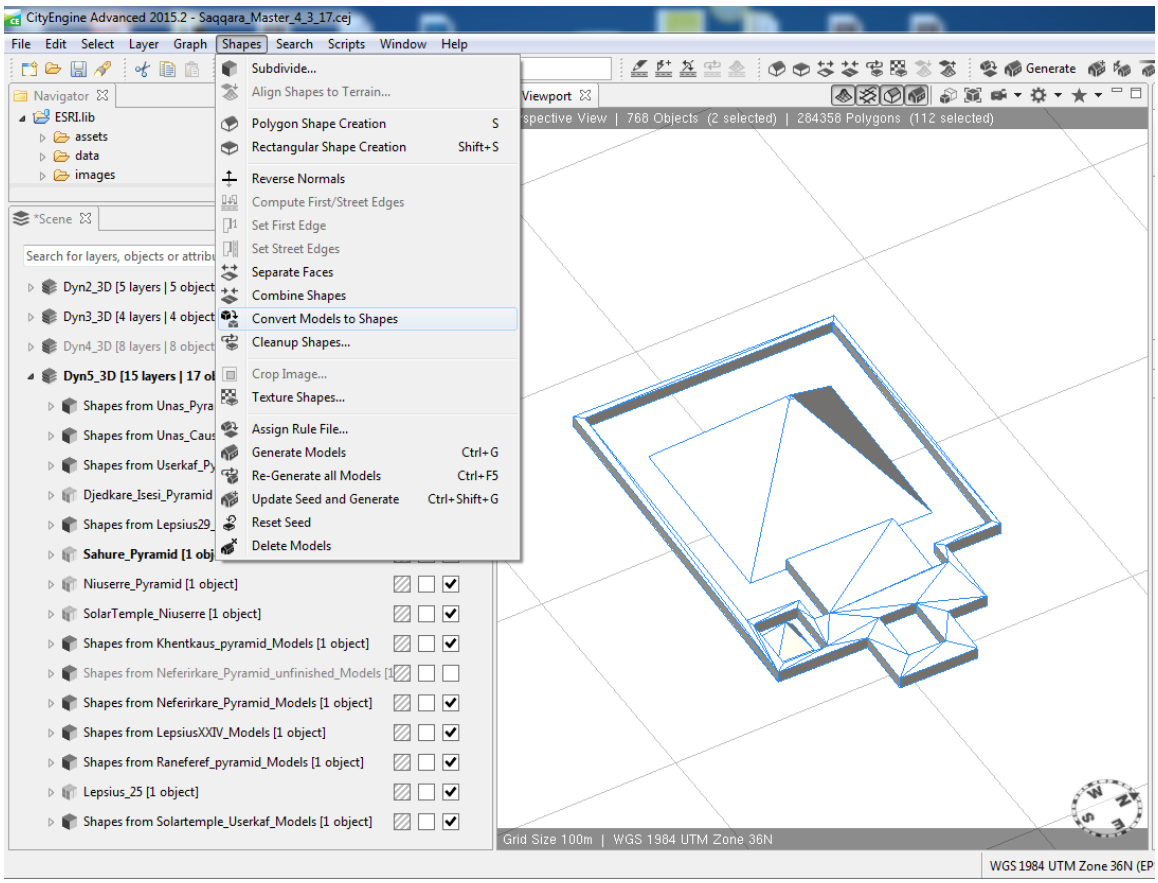

Figure 12. Converting 3D models into Shapes to join with 2D polygons, CityEngine

As noted above, if default color textures were utilized in SketchUp, or colors were not correctly exported as image files in the export process, colors ‘disappeared’ in the join process and sections of the new 3D Shape turned grey. Missing textures following the join process can be dealt with in the CityEngine platform with varying degrees of success.

For monochrome monuments (like white plastered *mastaba* tombs), entire models could be assigned a color using a CGA rule. Online RGB color code charts provide information on the codes for specific colors.

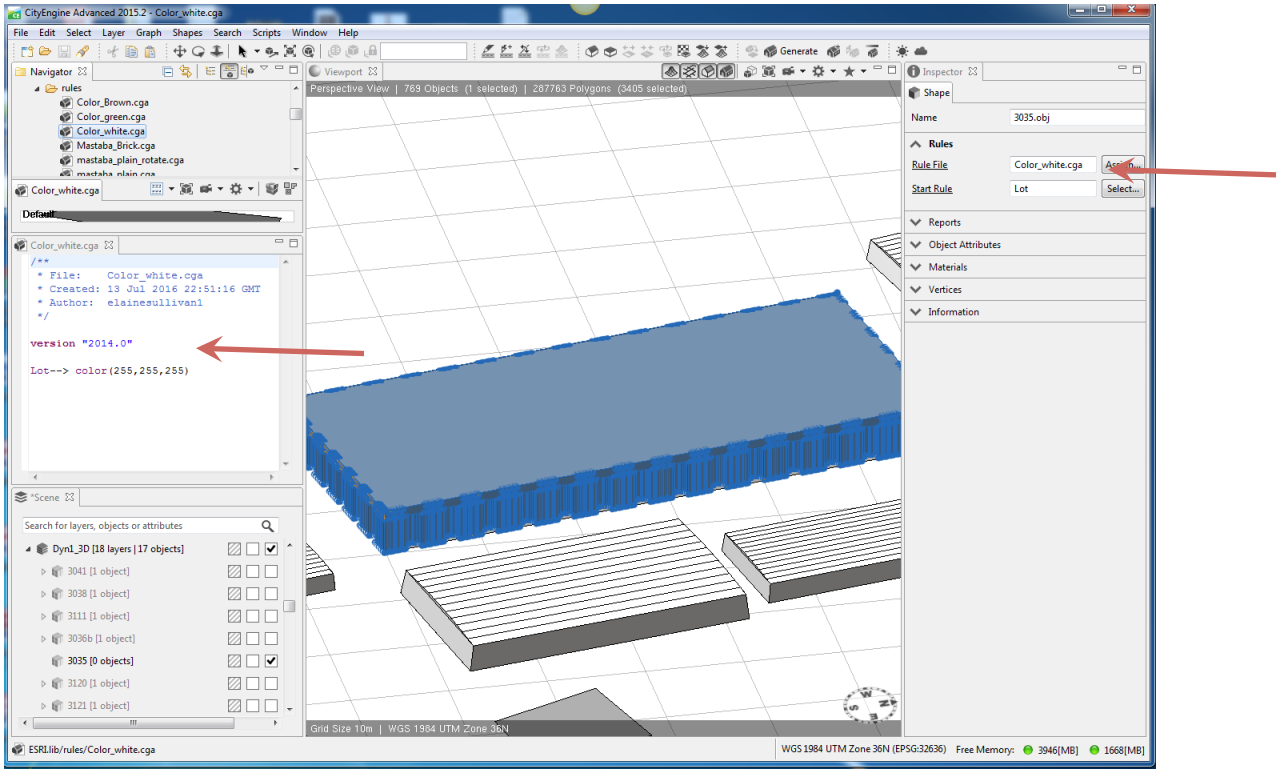

Figure 13. Re-coloring an entire 3D shape with a rule file

If only a few areas of a structure lost a texture, these areas were individually selected, and the ‘Shapes> Texture Shapes’ tool was used to assign those faces a texture. The tool also allows for the resizing of textures. It is important to create an image file with the desired color swatch and place that in the texture folder associated with that Shape in the 'assets' folder. For this project, we created these image files by taking screen shots of the desired color in the SketchUp program.

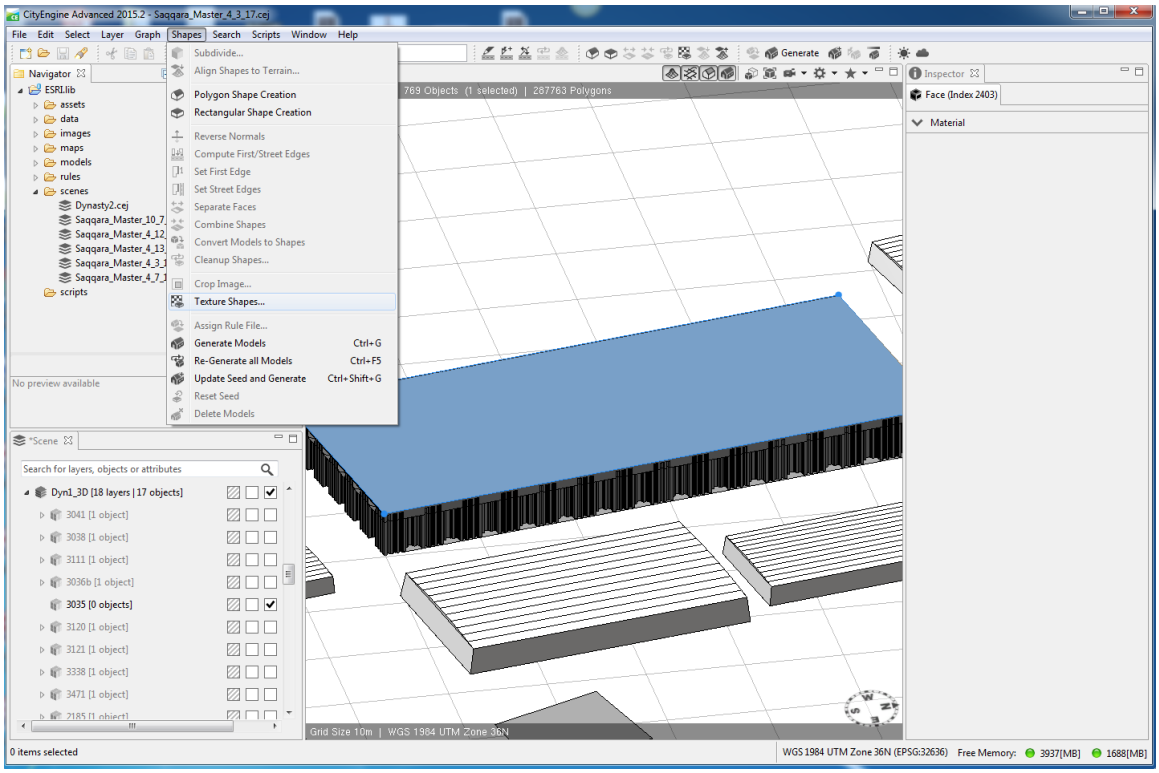

Figure 14. Selecting a single polygon and re-texturing in CityEngine

For very complex texturing, we found it easier to re-texture the models in SketchUp and re-import them into the CityEngine Scene.

IV. Import for Line of Sight Analysis in 2.5D GIS System (ArcScene)

Because of monument deflation, destruction, and modern construction in the Cairo area, ancient sight lines and the visibility of individual monuments at different moments in the history of Saqqara is difficult to gage in modern times. This project hoped to use the reconstruction model to examine questions in the digital environment that cannot be answered in the field today. In order to run GIS-based visibility analysis on our resulting 3D model, which is not possible within the CityEngine program, we moved it into ESRI's 2.5D program ArcScene.[5]

In CityEngine, we selected the 3D content of interest, selected the export option ('Export Selected Shapes and Terrain Layers') and exported to a collada file type (DAE). We did NOT export the terrain layer, as doing so continually caused CityEngine to crash.

[5] http://desktop.arcgis.com/en/arcmap/10.3/guide-books/extensions/3d-analyst/3d-analyst-and-arcscene.htm. Note that this workflow was developed in 2013 using ArcGIS 10.2 and thus we used the current version of ArcScene at that time. At this time of the publishing of this paper (2017), ArcGIS 10.5 is current.

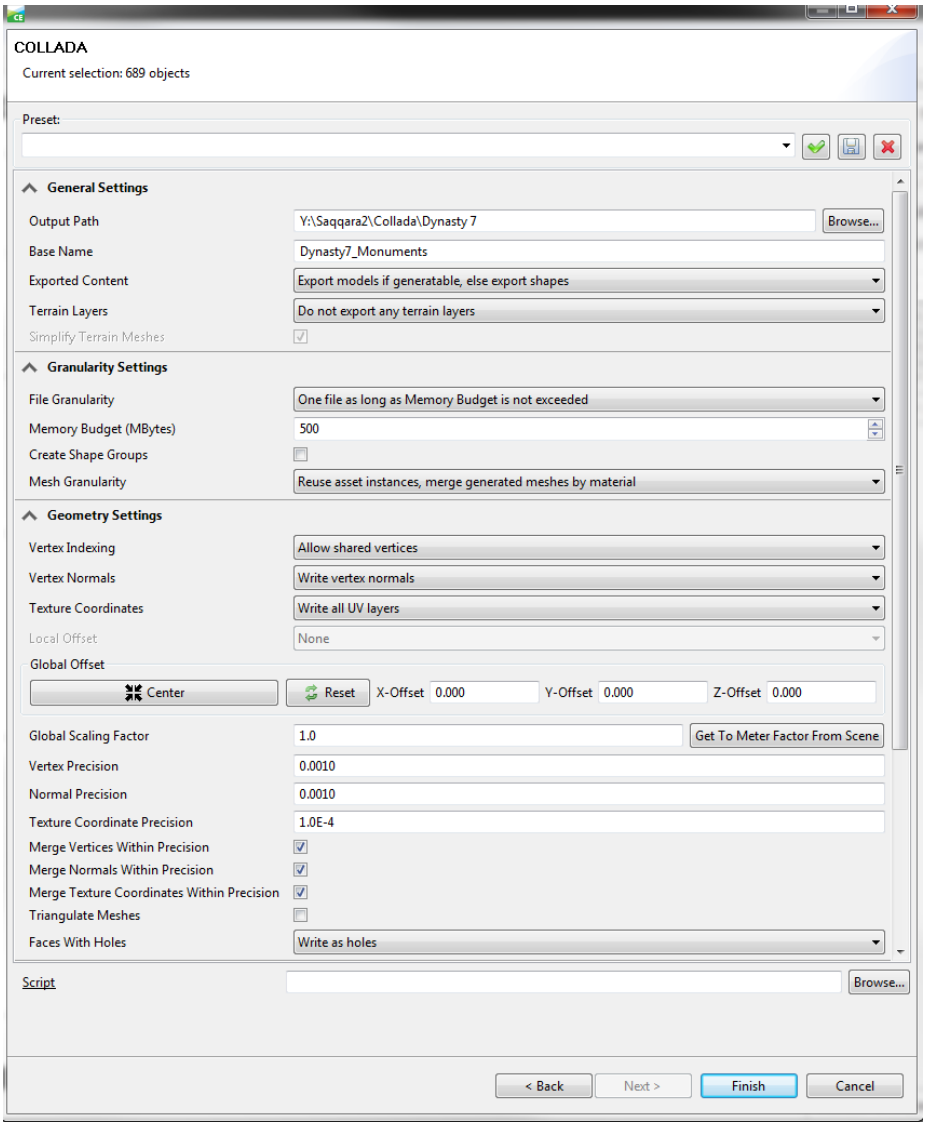

Figure 15. Exporting collada (DAE files) from CityEngine

ArcScene Import

For use in ArcScene, we converted the collada files into multipatch feature classes ('3D Analyst'> 'Conversion'>'From file'> 'Import 3D files'). Note that like CityEngine, ArcScene necessitates data to have a projected coordinate system.

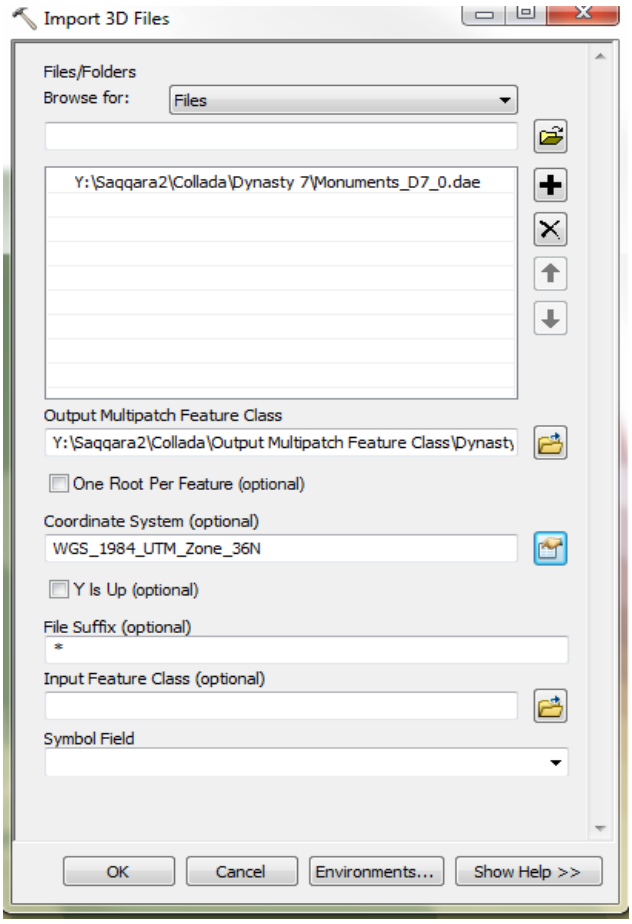

Figure 16. Importing 3D files into ArcScene

Importing Terrain Surface Layer

To visualize the contour lines in ArcScene as a 3D surface, we converted the contour line features into TINs (Triangulated Irregular Networks). To convert our original temporal contour line files (see above section on *Terrain Layers*) into TIN files, we used the 'Create Tin' tool in the 3D Analyst toolbox. Note the 'Height Field' box must be changed to designate the elevations of the contour lines.

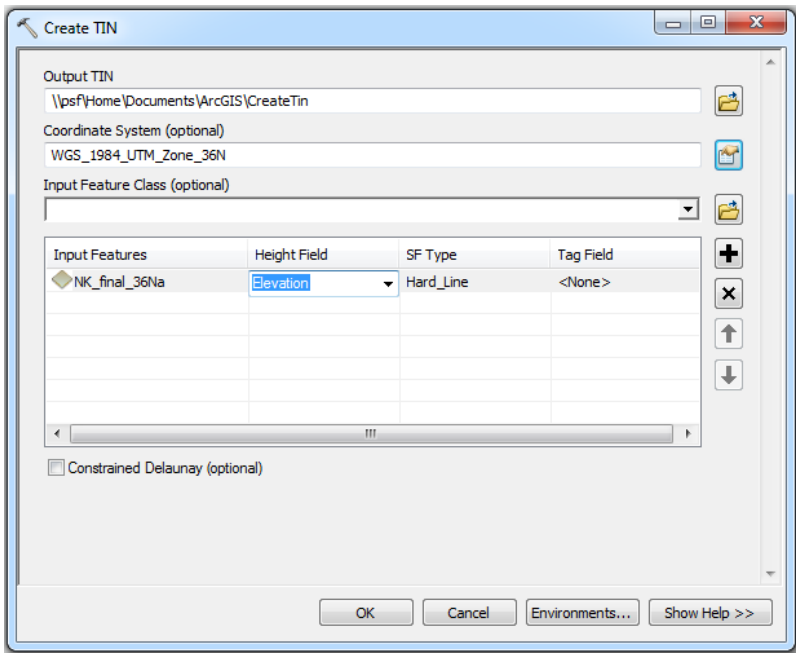

Figure 17. Creating a TIN for viewing surfaces in ArcScene

After producing the TIN, we changed its symbology to display a color ramp that reflected our contour elevations. The TIN was then added to the Scene.

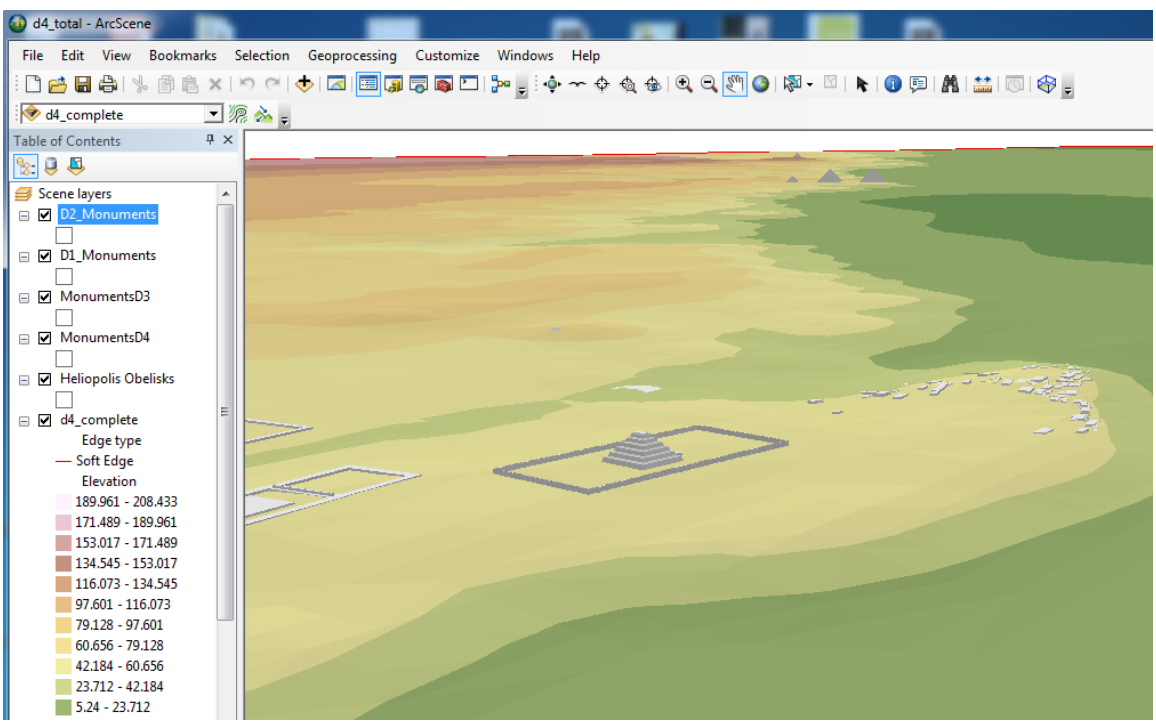

Figure 18. The 3D monuments and TIN file visualized in ArcScene

Line-of-Sight Analysis

Performing long-distance visibility studies in the model must take into account curvature of the earth and visual refraction. Note that these options in the 'Line-of-Sight' tool are only selectable options if the input surface raster has a defined vertical coordinate system.

Creating Observer and Target Features

To keep the data organized, we created a geodatabase for each time period (in our case, based on Egyptian Dynasties) in which to store separate feature classes and outputs of tools. Within each geodatabase we created a feature class for observer (3D Point feature) and target features (3D Point, Polygon, or Line features) to be used in the analysis. Each individual observer or target feature class can store a single point or polygon, or many different ones, depending on the features it is representing. To create target features that encompass an entire side of a 3D shape (as we did for this project), polygons must be selected for the target feature type.

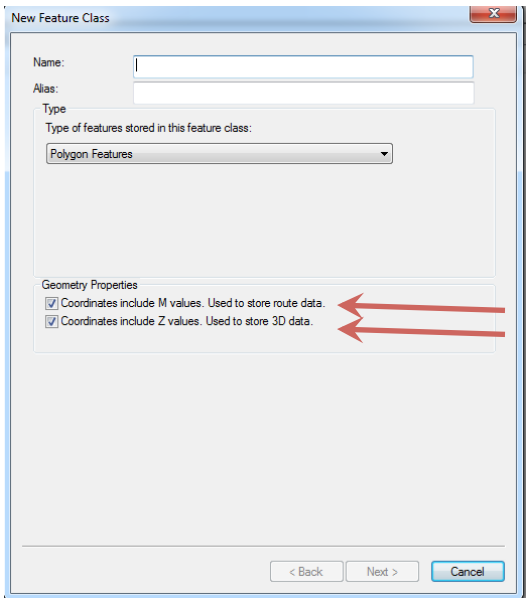

Figure 19. Creating new 'feature classes'

After adding the observer point feature class to an ArcScene document, a 3D edit session was used to manually add the observer points in specific locations of interest. To create the elevation values for the observer points, we populated a field with the appropriate height for each temporal terrain, plus an additional 1.5m to approximate the eye level of someone standing in each position. To embed the elevation value for each observer point into the feature geometry, we used the 'Feature To 3D By Attribute tool' and created a new 3D feature class with Z-values pulled from the elevation values of the attribute table. We appended '_withZ' to the newly created 3D features to distinguish them from the original observer point features.

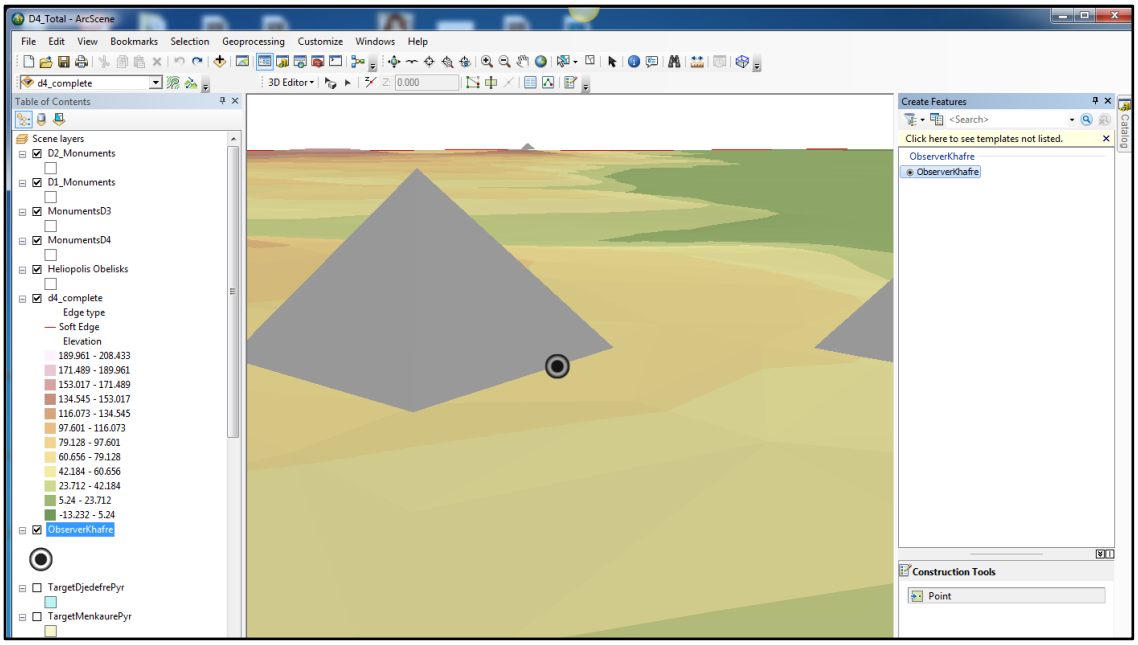

Figure 20. Creating observer points in the Scene

To create the target features we designated the specific forms/shapes/monuments of interest and digitized polygons while tracing whole sides or sections of a monument's shape. To do this, we selected vertices of the collada object. Small green dots showed where vertices were created and a purple area showed the outline of the created feature class. After edits were saved, a blue outline of the polygon appeared. The placement of any created vertex could be edited with the 'Edit Placement' tool.

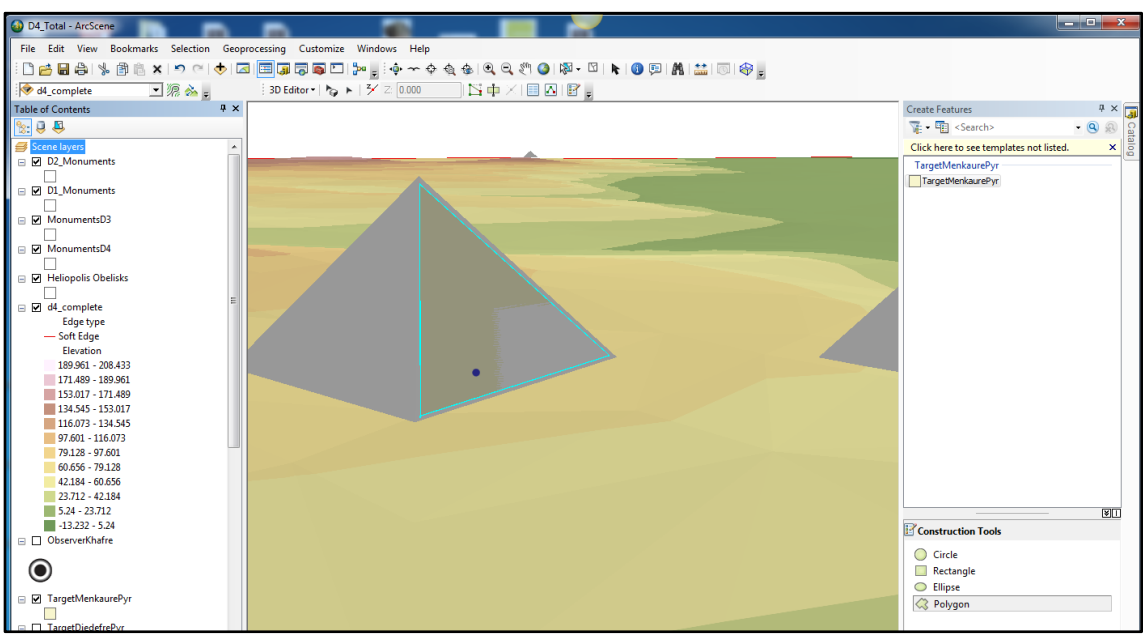

Figure 21. Creating Target Features as polygons in the Scene

To adjust the vertical height of target feature, we selected the target feature in the 'Create Features' dialogue box, then selected 'Move' from the dropdown box of the 3D Editor toolbar and edited the X, Y, and Z units of the selected feature. In this case, the last box (Z) allows for the vertical adjustment of the feature in meters.

Running Line of Sight Analysis

Line-of-sight analysis can be performed once all the observer point, target feature, and monument feature classes have been created. The first step in the workflow is to run the 'Construct Sight Lines' tool (note the 3D monuments did not need to be made into a feature class, as they could merely be selected as 'Input Features' with the 'Line of Sight' tool). Observer points and target feature classes were selected for their respective fields from the dropdown menu, and then a location was selected to save the output. Both observer and target heights should be to SHAPE.Z (this only works if you already added in the Z-values as specified above). If processed correctly, a series of sight lines are added to the Scene.

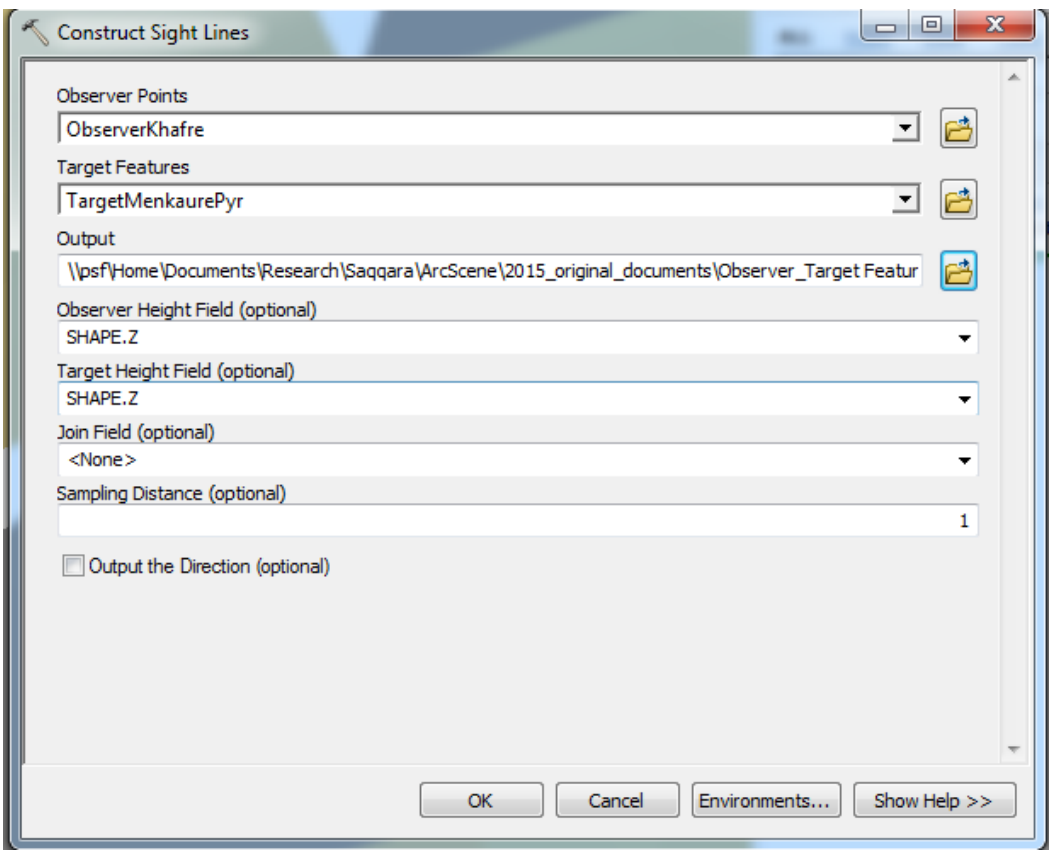

Figure 22. Constructing Sight Lines tool

The next step utilized the 'Line-of-Sight' tool. Under input surface, we selected a raster elevation model created from our contour lines. Note that rasters, not TINs, are used for 'Line-of-Sight' tool analysis. Under 'Input Line Feature,' we selected the output of the previous process. 'Input Features' is an optional dialogue, allowing for the selection of the monuments that are grouped together in a single feature class, or a single target

building if features are separate. In our process, we selected as an 'Input Feature' a collada file with all the monuments present at the site at the time period of interest (exported from CityEngine). Note that the more objects in an 'Input Feature' class, the more processing time this tool will take.

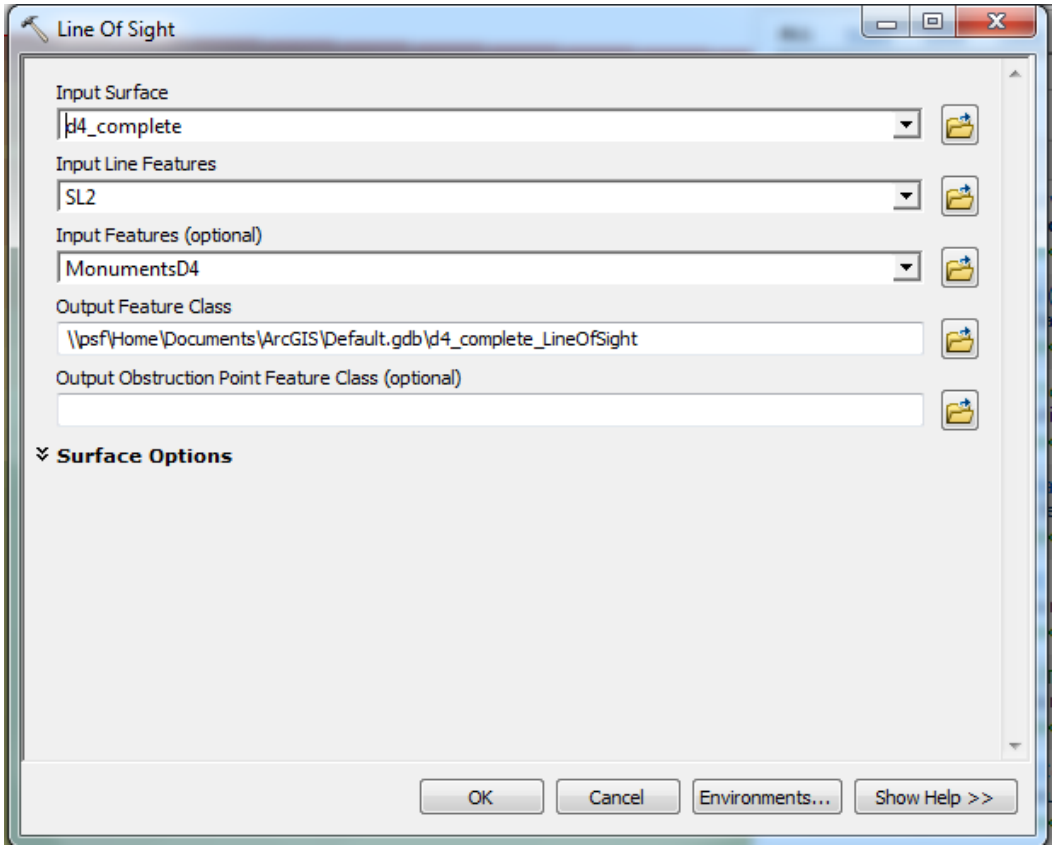

Figure 23. Running the Line of Sight tool

Symbolizing Line of Sight Outputs with Multipatch Features

The default symbology when 'Line-of-Sight' output is added to the scene does not usefully reflect feature obstruction by the 3D multipatch features in the scene (or in our case, visibility of the target feature).[6]

To display lines of sight that show what part of our target polygon was *visible* from the observer point, we had to alter the symbology display. In the symbology tab of the new layer we changed the 'Value Field' to 'Obstr_MPID.' This field takes into account the

[6] Note in the attribute table there are 3 different columns with codes that relate to visibility: VisCode, RarlsVis and OBSTR_MPID. The 'VisCode' and 'TarIsVis' fields do not take the multipatch features into account. The 'OBSTR_MPID' field is populated with the unique ID of the multipatch feature that obstructs line of sight. See ESRI's help for a discussion of the feature attribute tables: http://desktop.arcgis.com/en/arcmap/10.3/tools/3d-analyst-toolbox/line-of-sight.htm

3D multipatch features. [7] Next, we selected 'Add Values,' and chose '9999,' giving that label 'visible' (-9999 signifies no multipatch feature obscures visibility), and labeled 'all other values' 'not visible' ('1' indicates the multipatch feature obscures visibility). We then changed the color symbology to green and red. This changed the visualization in the map to highlight the result of our process. It is critical to note that the visibility symbology chosen here *only* indicates visibility of the target feature. It does not represent the visibility of the multipatch features that act as obstructions from the observer point, it only registers them as obstructions to the designated target.

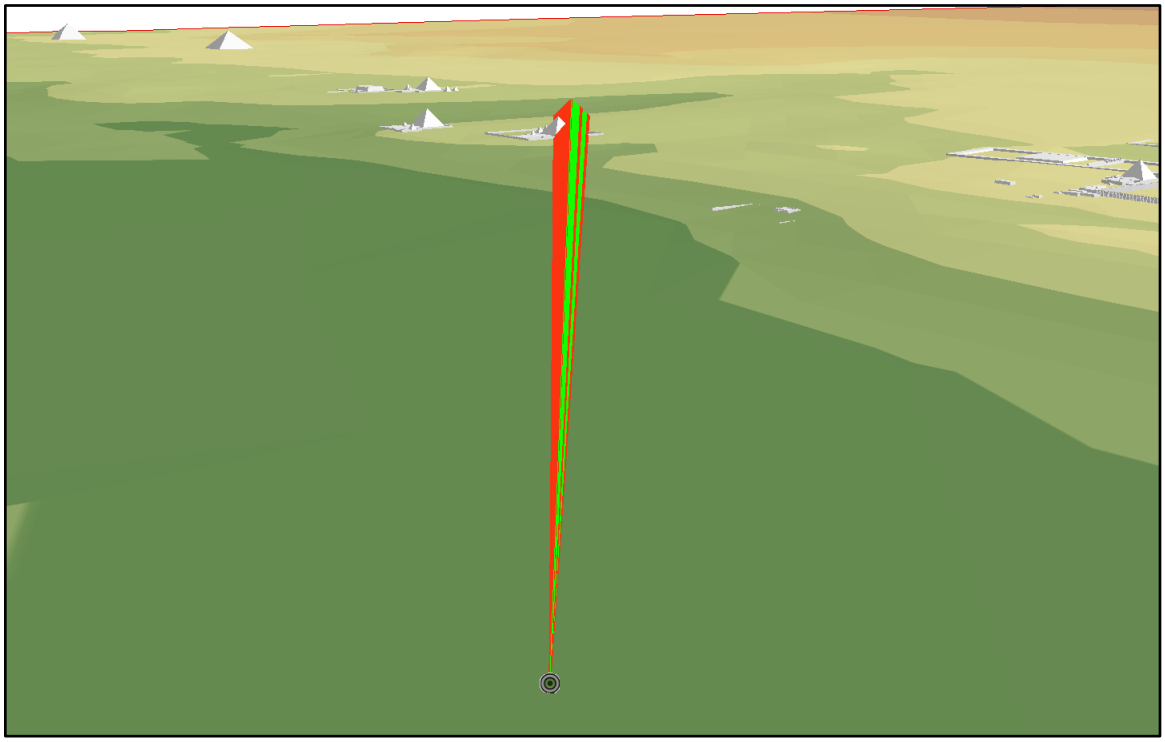

Figure 24. Observer point (foreground) and target feature (Pyramid of Merenre) with altered symbology, green lines indicated what parts of the target feature are 'visible' from observer

[7] "If no multipatch feature obstructs the line of sight, then the field contains a value of -1 or -9999. If the target is obstructed by the surface, the value will be -1. If the target is visible, the value will be -9999." Hence if it is obstructed by a multipatch feature, that number will be 1" : http://desktop.arcgis.com/en/arcmap/10.3/tools/3d-analyst-toolbox/line-of-sight.htm .

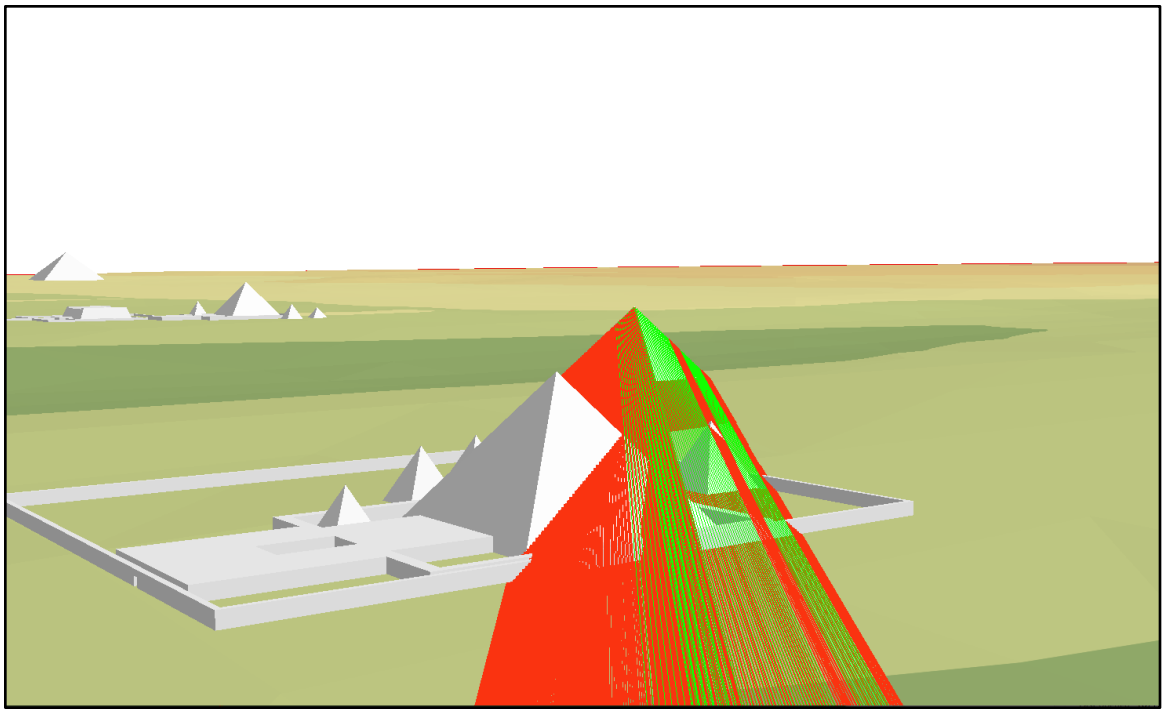

Figure 25. Close up of Target (Pyramid of Merenre) from previous figure, showing sections of the target polygon obstructed by another monument (therefore red/'not visible') from the observer point, note the multipatch feature acting as an obstruction is not indicated as 'visible,'as the symbology only relates to the visibility of the target feature

V. Import into Real-Time 3D Simulator (VSim)

While CityEngine and ArcScene offer certain useful types of visualizations, this project hoped also to interact within the 3D reconstructed landscape of Saqqara in a more experiential, human-centered format, virtually 'walking' around the necropolis, mimicking human movement and viewpoints. The model of North, Central, and South Saqqara (with the rest of the Memphite area excluded) was therefore exported into a real-time simulation platform designed for education called VSim. The VSim program is a free, open-source software player that allows for viewers to fully interact with modeled environments, following 'narratives' created by content specialists, but also navigate and explore on their own: https://idre.ucla.edu/research/active-research/vsim.

After consultation with UCLA GIS and Visualization Specialist (and VSim co-creator) Lisa Snyder, the individual temporal terrain layers and 3D models were exported separately from CityEngine as OBJ files, in temporal and spatial groups (all the 3D buildings for 'Dynasty 1' would thus be in one file export).

After much experimentation, we discovered that the high-resolution terrain layers utilized in CityEngine (with resolution as high as 2049x2049) were untenable in the VSim program (and likely many other programs) due to their large size (due to the large number of polygons) The terrain layers were thus reduced down to a resolution of 513x513 in CityEngine (from 30MB to between 7-9MB each) before export.

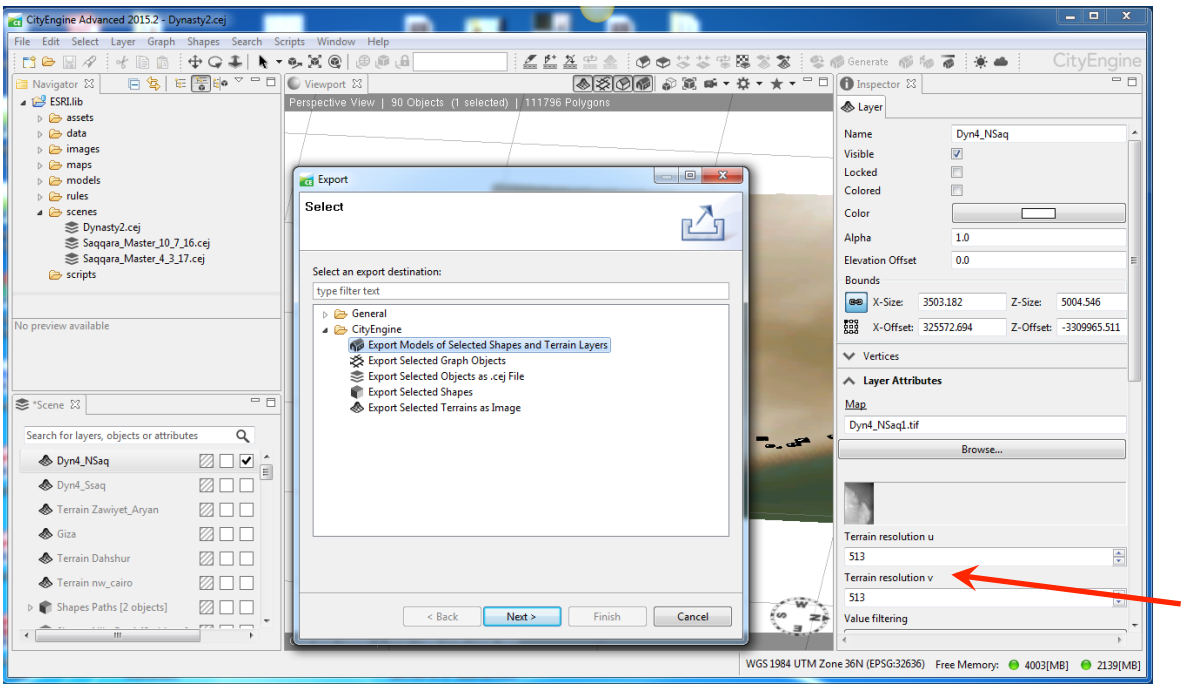

Figure 26. Exporting terrain files from CityEngine,
note terrain resolution at 513x513

We also found that specific terrain export options minimized the size of the files. We turned OFF all other terrain layers in the scene, to make sure these were not accidentally exported, and exported each terrain separately. We eventually settled on the following export choices for the lightest terrain files:

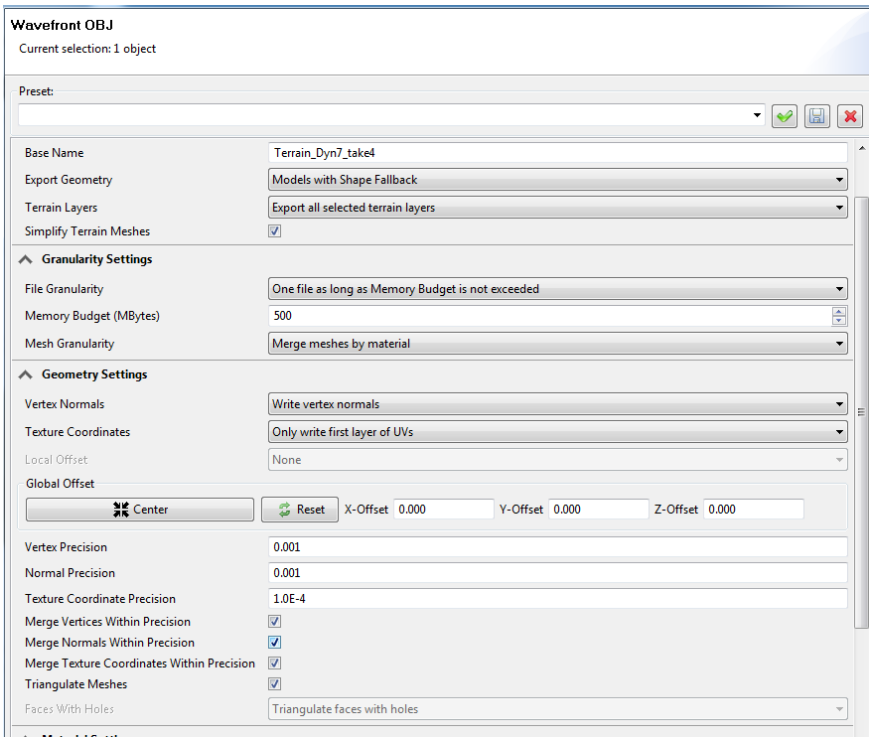

Figure 27. Export choices for OBJ files in CityEngine

This included ‘Simplify terrain meshes: ON’ and ‘Triangulate meshes: ON.’

Snyder individually imported each of these files into the 3D modeling program MultiGen Creator (http://www.presagis.com/). Once in Creator, she rotated the geometry to set the Z axis as up and repositioned it at the proper coordinates. The individual building and terrain files were then linked into a master file, and assigned construction and destruction dates (in this case, BCE). These dates allow the models and terrains to trigger a ‘time-slider’ in the VSim platform so that users can track the chronological development of the site. Snyder also examined and normalized each of the models in Creator to ensure a consistent visual tone. This work included minor geometry adjustments (e.g., reversing faces), adding textures stripped during the conversion process, removing inconsistent materials, and shading. Finally, the master 3D Saqqara Creator file was opened in VSim, basic metadata about the project was added, and the package exported as a distributable file.

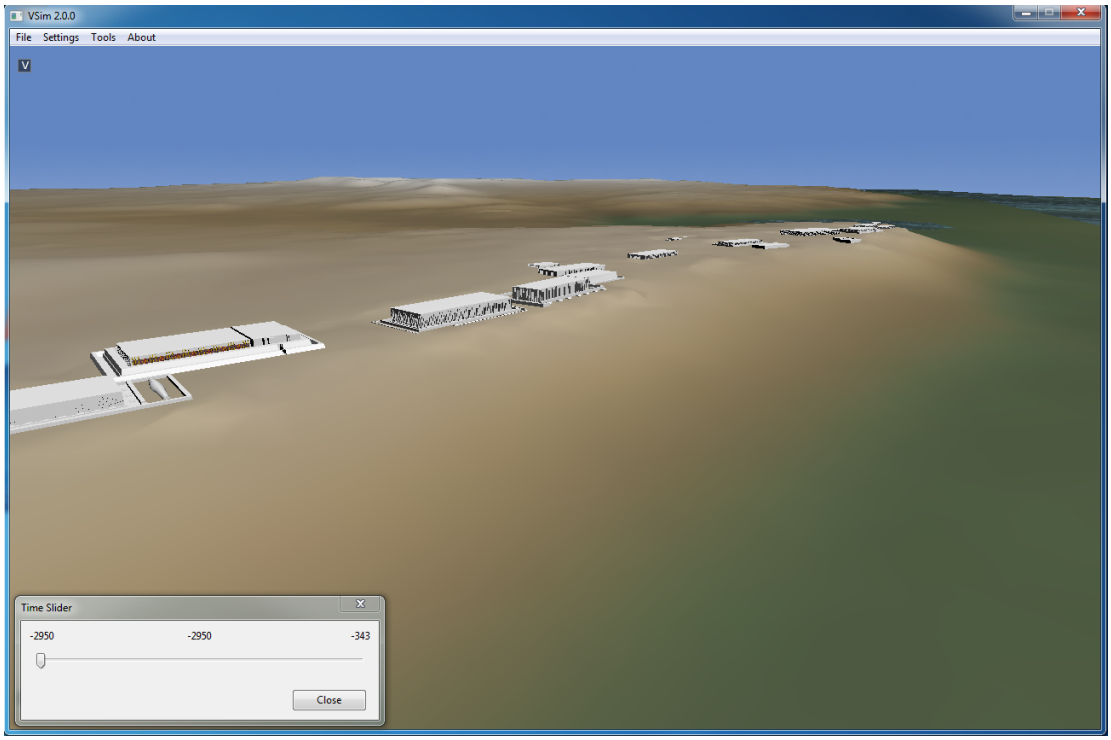

Figure 28. The resulting 3D Saqqara model in Vsim, with time-slider displaying the site in its first reconstructed phase in Dynasty 1

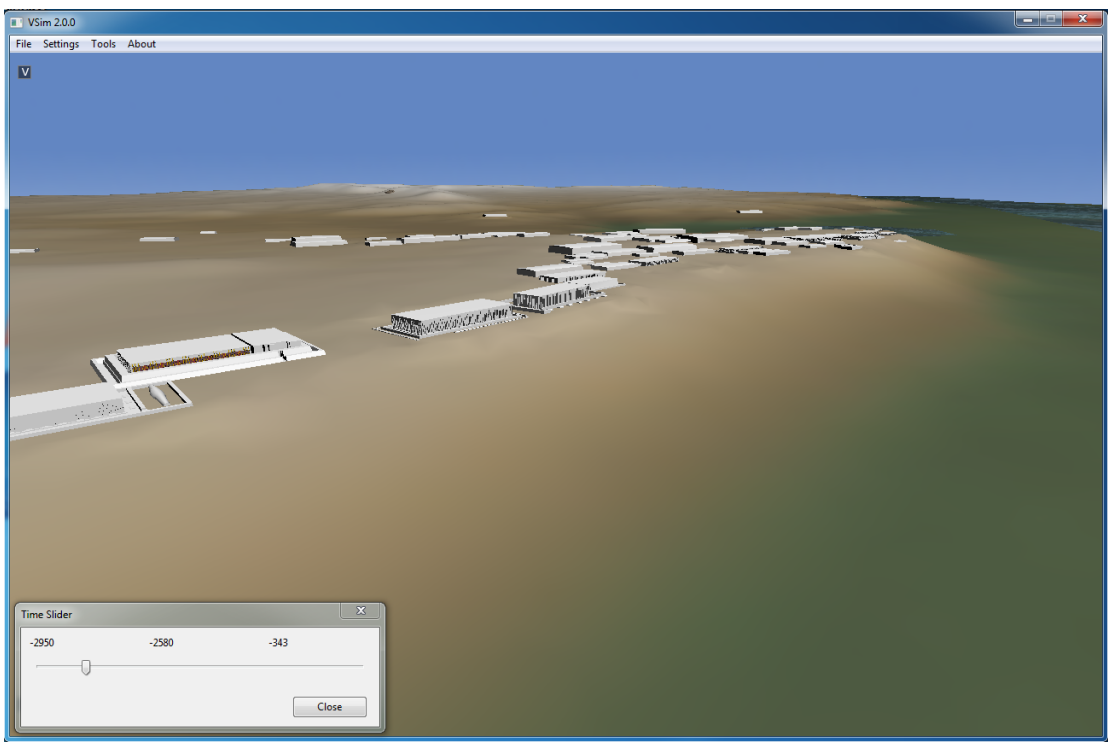

Figure 29. The 3D Saqqara model in Vsim, with time-slider displaying the site in Dynasty 3

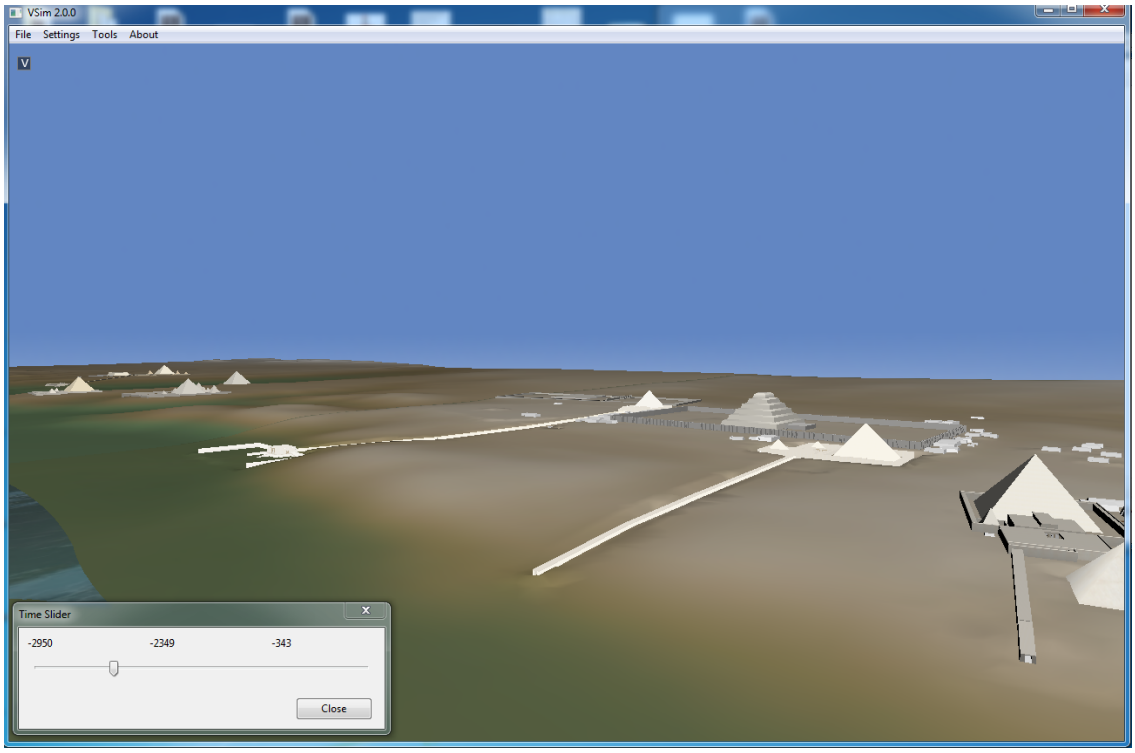

Figure 30. The 3D Saqqara model in Vsim, with time-slider displaying the site in Dynasty 6

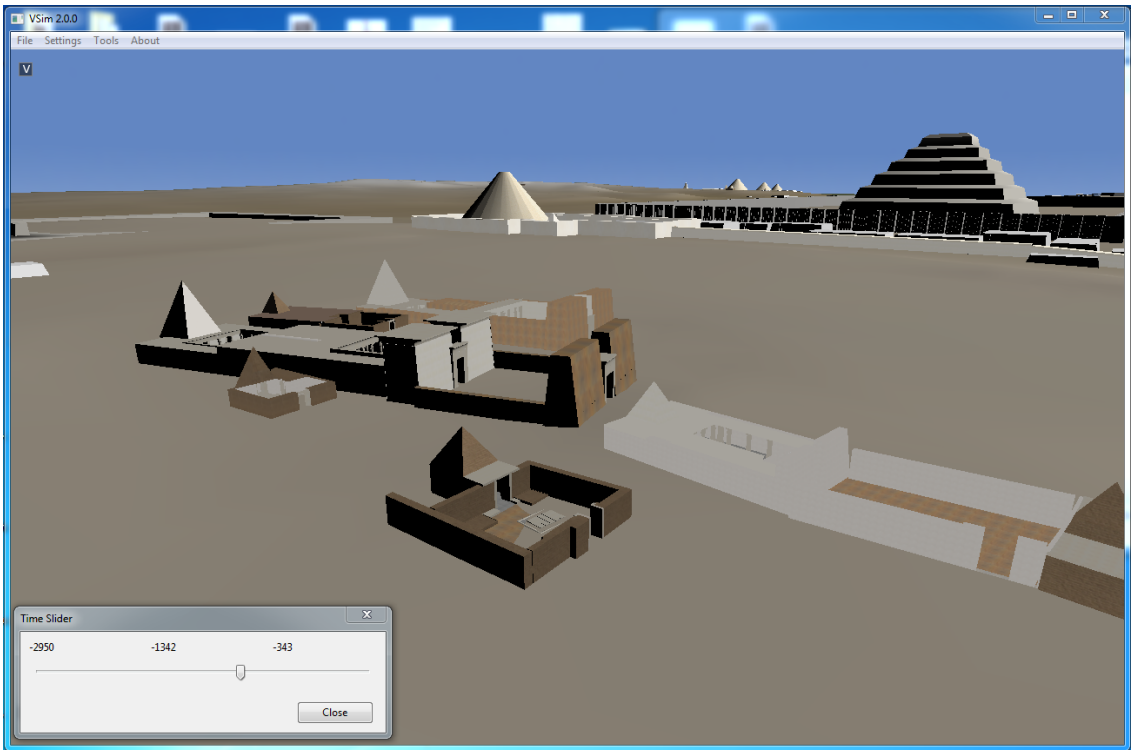

Figure 31. The 3D Saqqara model in Vsim, with time-slider displaying the site in Dynasty 18

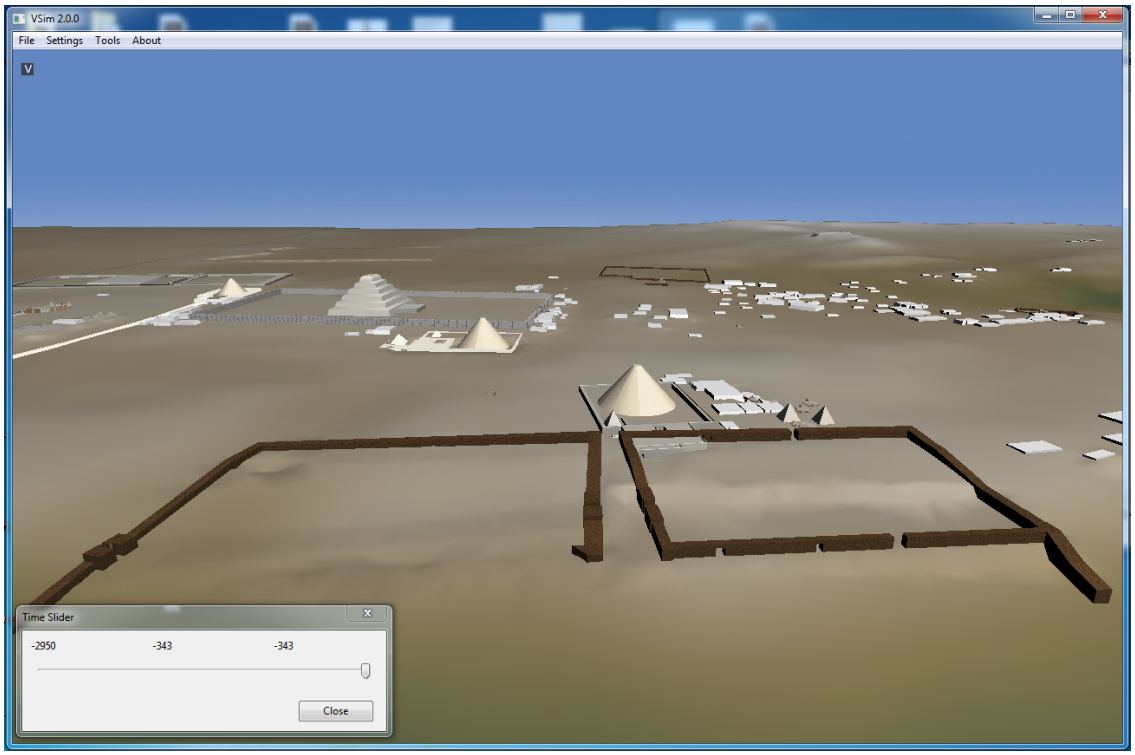

Figure 32. The 3D Saqqara model in Vsim, with time-slider displaying the site at the end of the Late Period (Dynasty 30)

VI. Lessons learned

The process described here was at every step guided by trial-and-error. We made many missteps and were forced to replicate work at a number of points in the process. We have flagged a number of these processes in this document, with the intention that others might avoid these pitfalls. With the exception of the CityEngine tutorials designed by Dr. Marie Saldana (discussed above), little formalized guidance exists for moving data from within the CityEngine program out to other formats. It is hoped that our discussion here will assist others wishing to use this program, and the data models it creates, in creative ways.

One challenging aspect of utilizing CityEngine is that its intended users are architects, urban planners and game designers (not archaeologists). The program was not designed for use by scholars working with precise (and fuzzy) data from the past. As with many digital tools used by archaeologists, the team has adapted CityEngine to align with our goal as scholars interested in better representing ancient places, but tools adopted from outside disciplines rarely address our specific needs. Despite its powerful capabilities to combine 2D GIS data with 3D models and terrains, the program has a number of crucial limitations. Like 2D GIS programs, change over time is difficult to incorporate or display (we only do this effectively by moving our models out of CityEngine into other programs like VSim). Many of the tools are designed for creating 'future' or modern cityscapes, where precise data is either easy to gather or not important. The emphasis in the program on procedural modeling for 3D buildings may be useful for some projects focusing on

very large scale cities (such as Rome), but many scholars want to design custom models focusing on the unique architecture of individual historic structures, not create a 'generalized' model of an ancient place. The most time-consuming aspect of our work in CityEngine was importing, aligning, and joining our custom 3D models with the 2D Shapes.

In addition, the technology learning curve for CityEngine (as well as traditional 2D GIS) is steep. This project (especially the export out from CityEngine into other programs) could not have been accomplished without the collaboration of many individuals with significant GIS and 3D modeling technology skills. At every stage, the project PI consulted with others (see Project Contributors) for assistance with technological hurdles. It cannot be stressed enough that such projects cannot succeed without sustained institutional support, through the intellectual and labor contributions of staff members and skilled students.

Project Outcomes and Future Plans

Publications

Two peer-reviewed journal publications were the direct result of the NEH support period. These publications discuss the research potential of the *3D Saqqara* model for Egyptology and for the larger field of Archaeology. Both publications utilized parts of the model completed during the first part of the NEH granting period. 'Seeking a Better View' documented the project methodology in detail, discussing data sources, processes utilized in the construction of the model, and the problems and potentials for these types of visualization projects. 'Potential Pasts' discusses in depth the Humanistic questions that can be approached with these types of human-centered 3D GIS models.

Sullivan, Elaine. "Potential Pasts: Taking a Humanistic Approach to Computer Visualization of Ancient Landscapes." *Bulletin of the Institute of Classical Studies* 59, no. 2 (2016): 71-88.
http://onlinelibrary.wiley.com/doi/10.1111/bics.2016.59.issue-2/issuetoc

Sullivan, Elaine. "Seeking a Better View: Using 3d to Investigate Visibility in Historic Landscapes." *Journal of Archaeological Method and Theory* (2017).
http://link.springer.com/article/10.1007/s10816-016-9311-1

Ongoing Work

VSim Model

In collaboration with UCLA technologist Dr. Lisa Snyder, the Central, North and South Saqqara model was transitioned into the open-source VSim 3D navigation program during the period of NEH support. Immediate plans for this model include robust annotation with a series of VSim 'narratives' guiding the viewer through the site. These, and the VSim version of the model, will be published and archived in UCLA's *VSim* online project repository for open distribution (that work is funded by current NEH grants). https://securegrants.neh.gov/publicquery/main.aspx?f=1&gn=HK-50164-14 Anticipated archive date is 2018.

Future Publication

Full results from the completed model will be incorporated into a large-scale born-digital monograph focusing on visibility and ritual landscape at the site of Saqqara. Plans for this publication include the import of the model of North, Central, and South Saqqara into a free, online webviewer for interactive use by the reader. This work will undergo peer-review summer 2018, with an anticipated publication date of 2019.