04/22/2021

Nikki Jackson-Hebson
Executive Summary – Automation of 3D Model Utilizing Python and Tkinter

Overview – Throughout my career, I have been frequently tasked to create a 3D elevation visualization from a Digital Elevation Model (DEM). The current process of visualizing a geographic area in 3D through ESRI software, specifically ArcMap, is manually intensive, time consuming, and requires running multiple geoprocessing tools. The 3D Python Modeler application streamlines the manual process with automation and allows multiple analysts to create 3D visualizations through one Python application in a matter of seconds as opposed to an hour. The application was built in Python utilizing a variety of ArcPy and Tkinter libraries and implemented in a graphical user interface (GUI) that produces and exports a 3D model to optionally include reference GIS data sets of contours and a slope in degrees raster. The exported files are viewable in a variety of formats to include ArcMap, ArcScene, and CityEngine Web Viewer. The infographic in *Figure 1* depicts the geoprocessing tools required to create a 3D visualization and combines them into one application written in Python which minimizes the amount to time processing data and maximizes time providing surface information analysis to respective clients.

Audience – The intended audience is for any analyst interested in producing 3D visualizations for their respective clients, i.e., geospatial analysts in state, federal, and commercial positions, landscape and design artists, insurance companies, home and/or property owners.

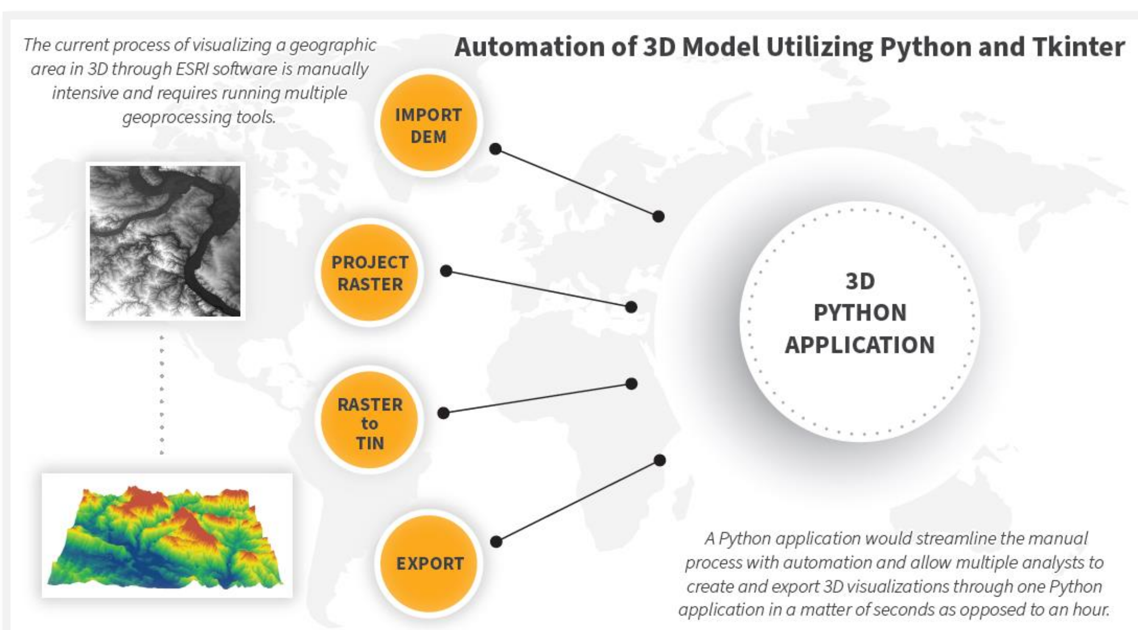


Figure 1: Infographic to visually represent the intended development of a 3D Python Modeler application

Technical Components

The following software is required to run the application:

- ArcGIS Desktop 10.06.1 or higher – Software license required
- Python 2.7 – Included and installed with ArcGIS
- Integrated Development Environment (IDE) – PyCharm is recommended.
- Online or offline version of ArcGIS CityEngine Web Viewer.
- All technical components will be hosted locally.

3D Python Modeler

The following is an overview of the application's standard operating procedures:

"This application converts a projected 2D digital elevation model (DEM) to a triangulated irregular network (TIN) and exports a 3D model and surface information viewable in ArcMap, ArcScene, and CityEngine."

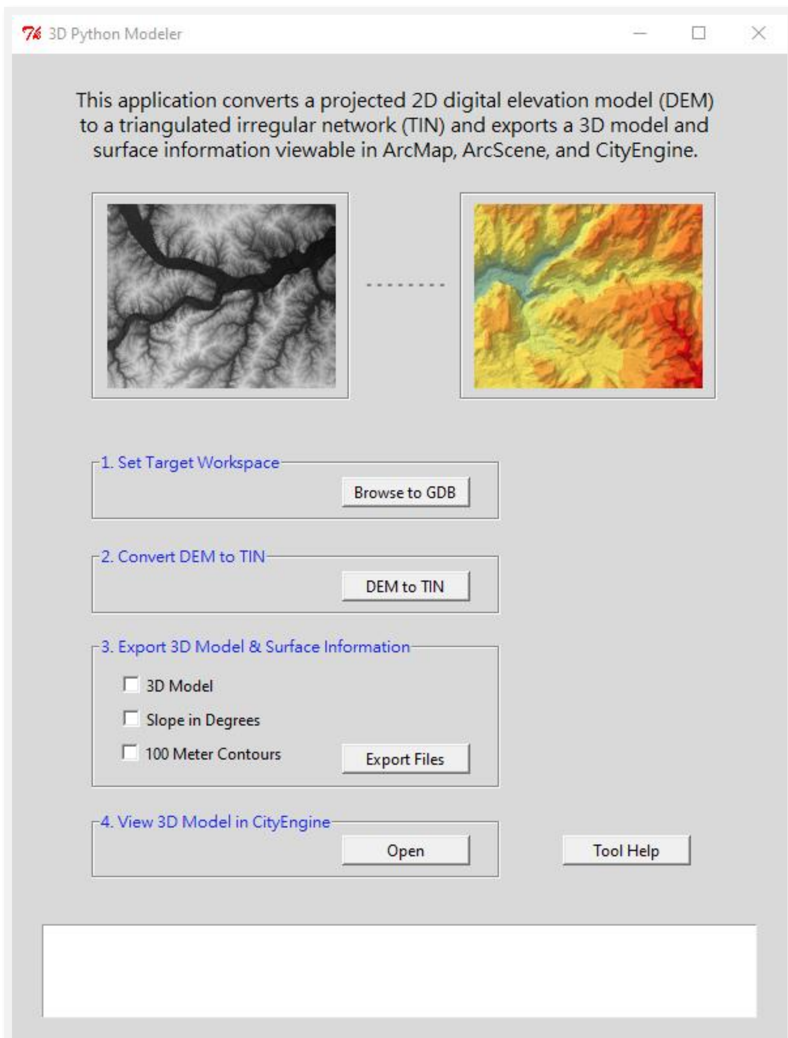


Figure 2: 3D Python Modeler application

Application Steps

A clipped and projected DEM of your area of interest will need to be imported into a geodatabase. *Figure 3* depicts a DEM of Yosemite National Park projected to WGS 1984 UTM Zone 11S and clipped to the area of interest. *Please note – the tool will experience memory issues if the extent is too large.*

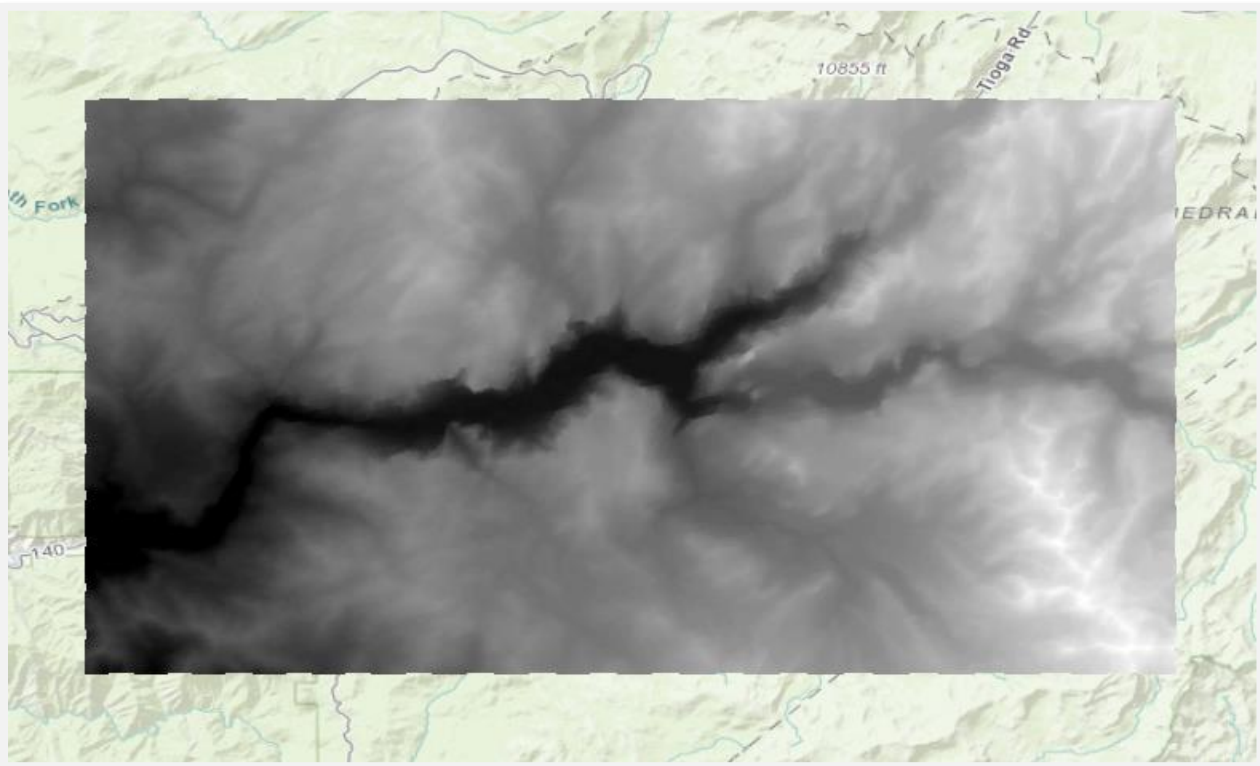
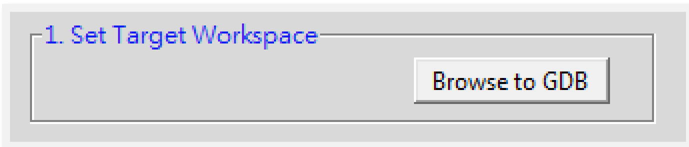


Figure 3: Clipped DEM of Yosemite National Park projected to WGS 1984 UTM Zone 11S

Step 1: Set Target Workspace

The targeted workspace is the geodatabase that stores the clipped and projected DEM. In *Figure 4*, the target workspace is DEM.gdb as it stores the projected yosemite_11s DEM.



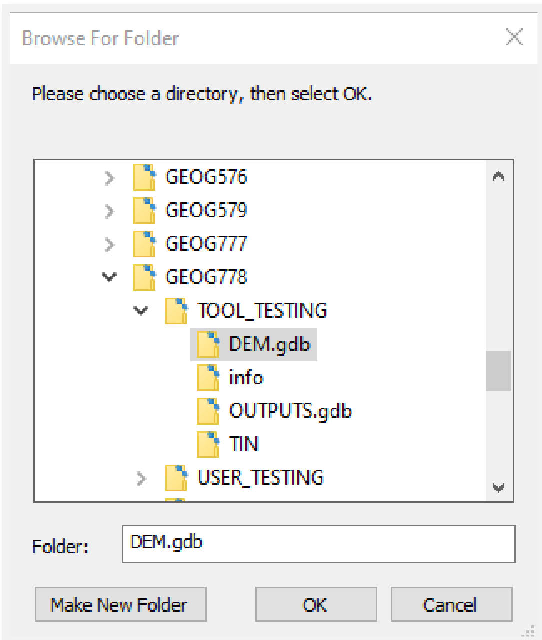
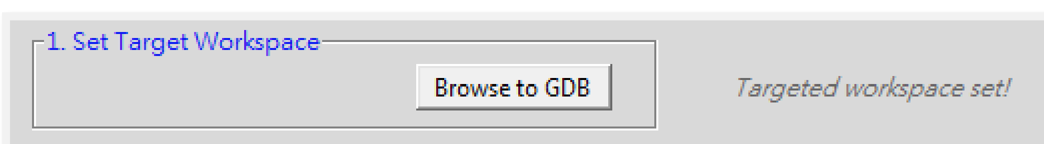
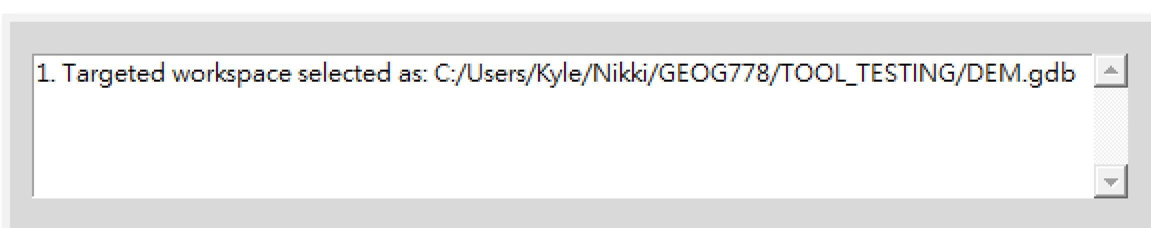


Figure 4: Example of Browse for Folder navigation

Once the targeted workspace is set, notifications will pop up in the application.

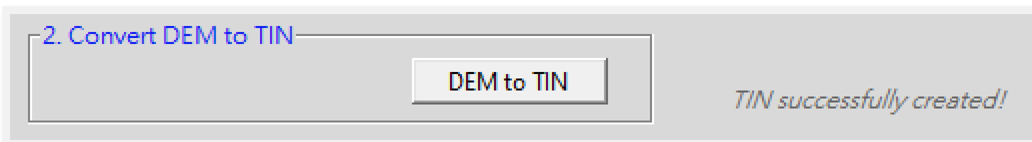


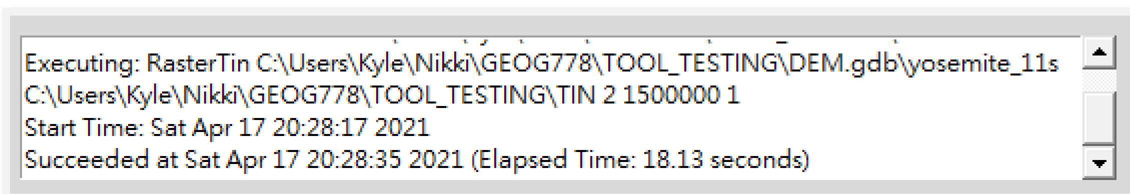


Step 2: Convert DEM to TIN

The second step is to convert the DEM to TIN by clicking the DEM to TIN button in the application. This will run a geoprocessing script that converts the previously selected DEM to a TIN. The TIN will automatically export to an OUTPUTS.gdb folder. *Please note, after clicking DEM to TIN the application is working to convert to a TIN.*

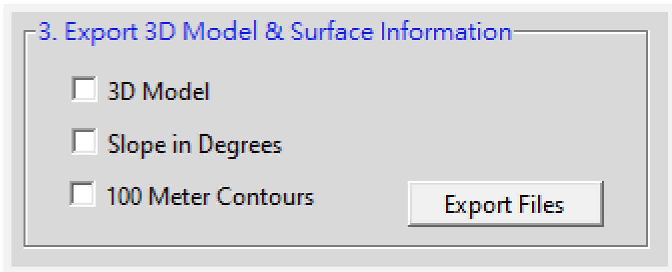
After the conversion successfully completes, notifications will pop up in the application.



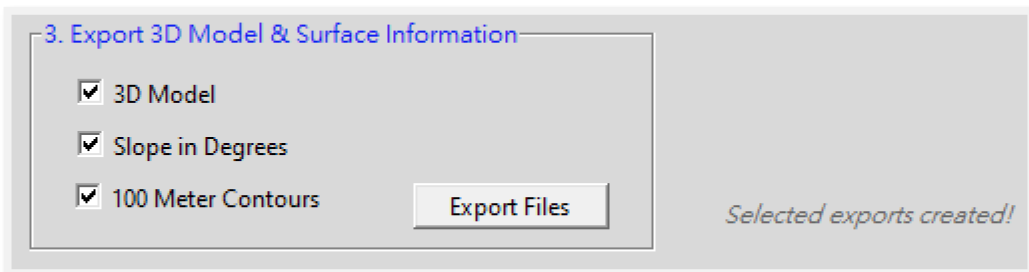


Step 3: Export 3D Model and Surface Information

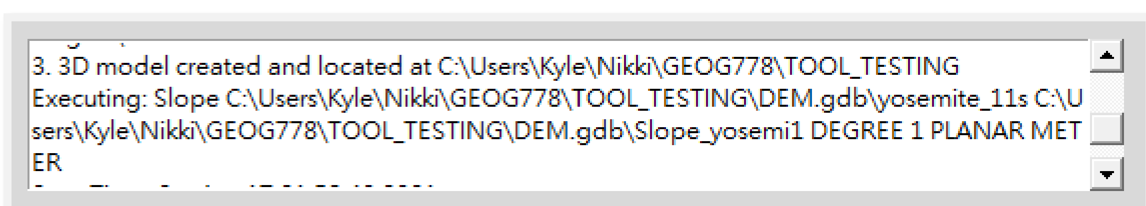
The third step is to select which surface information file/s to export by checking their respective box and clicking the Export Files button within the application. Depending on the file/s selected, the application will export 1, 2, or 3 surface information files.



After the exports successfully run, notification will pop up in the application.



Notifications of export locations are found in the scrolling textbox shown below.



The exports are viewable in the following locations:

3. Contour shapefile created and located at:
C:\Users\Kyle\Nikki\GEOG778\TOOL_TESTING\OUTPUTS.gdb\CONTOURS

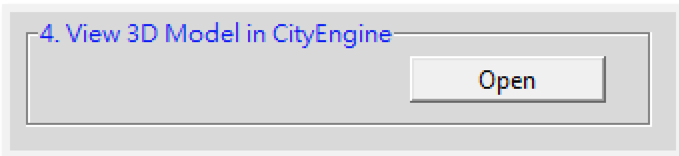
3. Slope raster created and located at
C:\Users\Kyle\Nikki\GEOG778\TOOL_TESTING\OUTPUTS.gdb\SLOPE

3. 3D model created and located at C:\Users\Kyle\Nikki\GEOG778\TOOL_TESTING

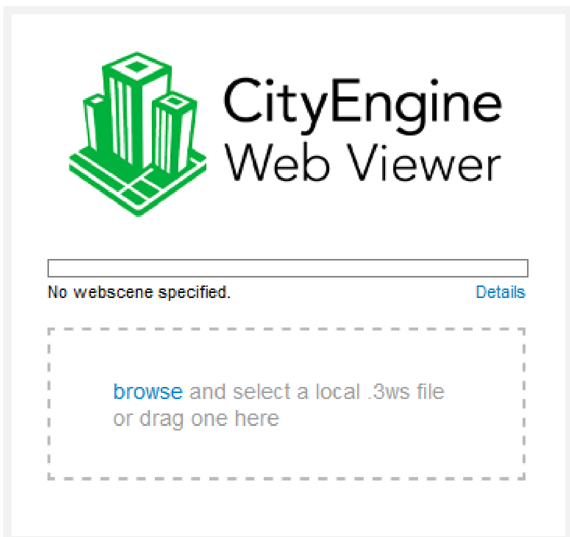
04/22/2021

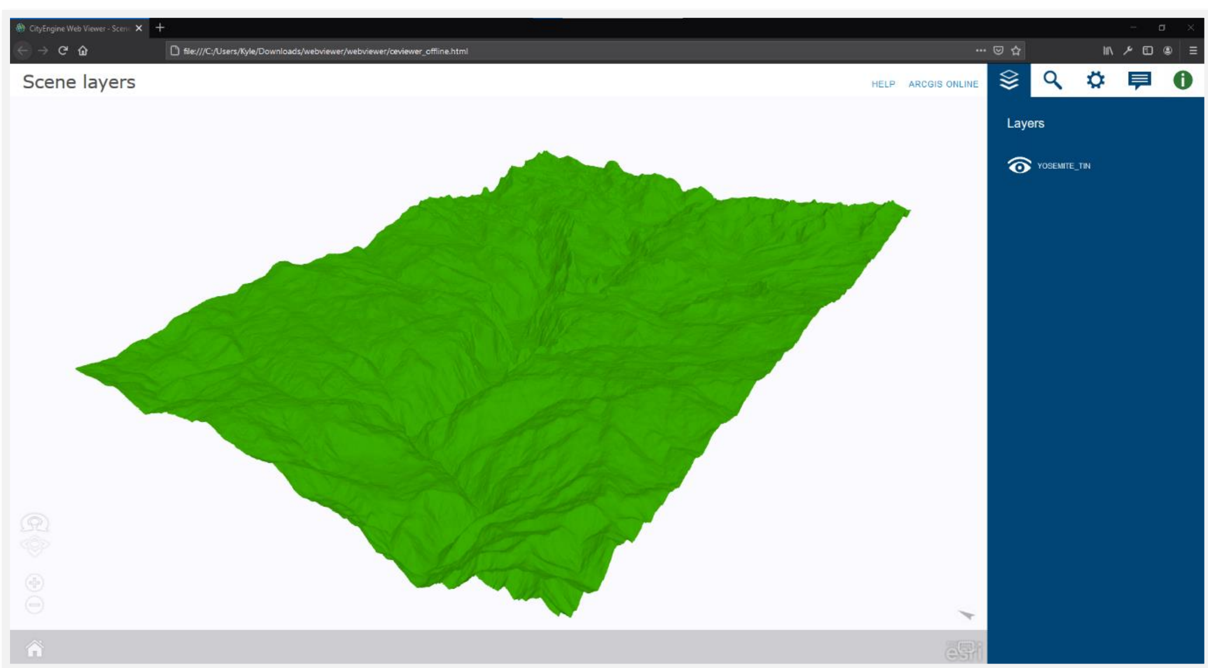
Step 4: View 3D Model in CityEngine

The fourth and final step is to view the 3D Model in an offline CityEngine viewer by clicking Open under step 4 in the application.



This opens the offline CityEngine viewer. Browse to the TIN and view in 3D.

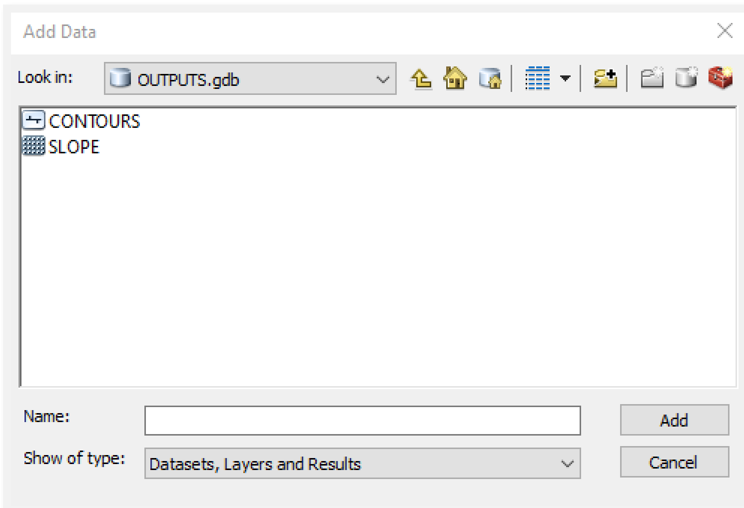




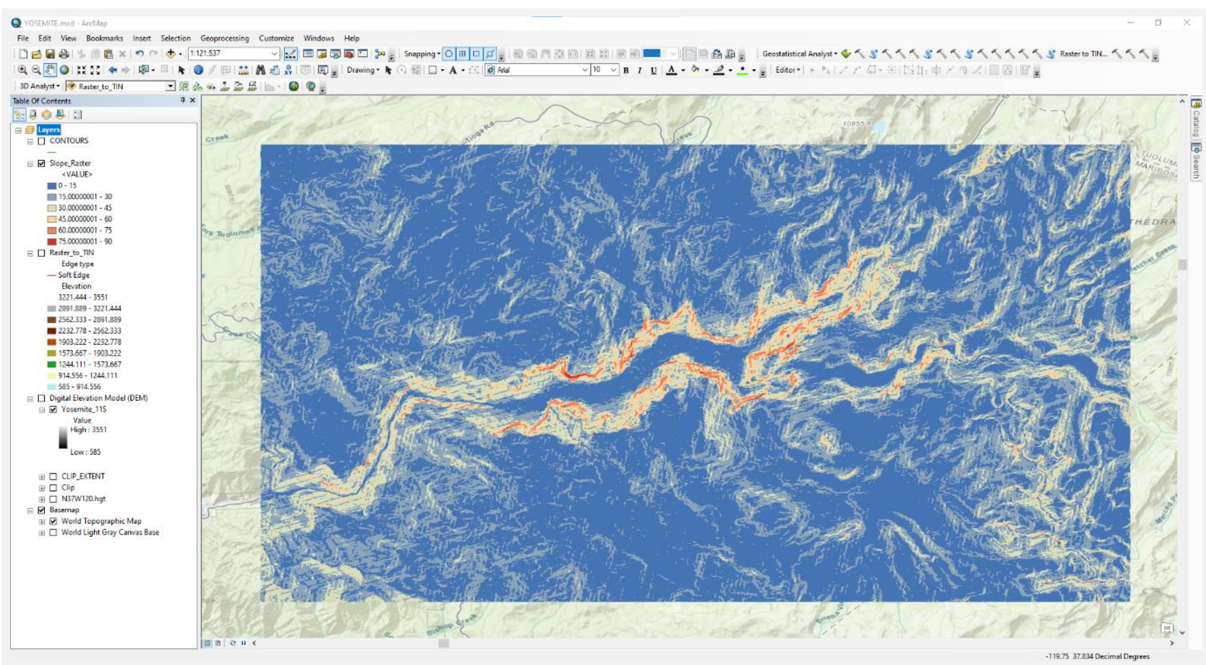
Step 5: View Surface Information in ArcMap and ArcScene (Optional)

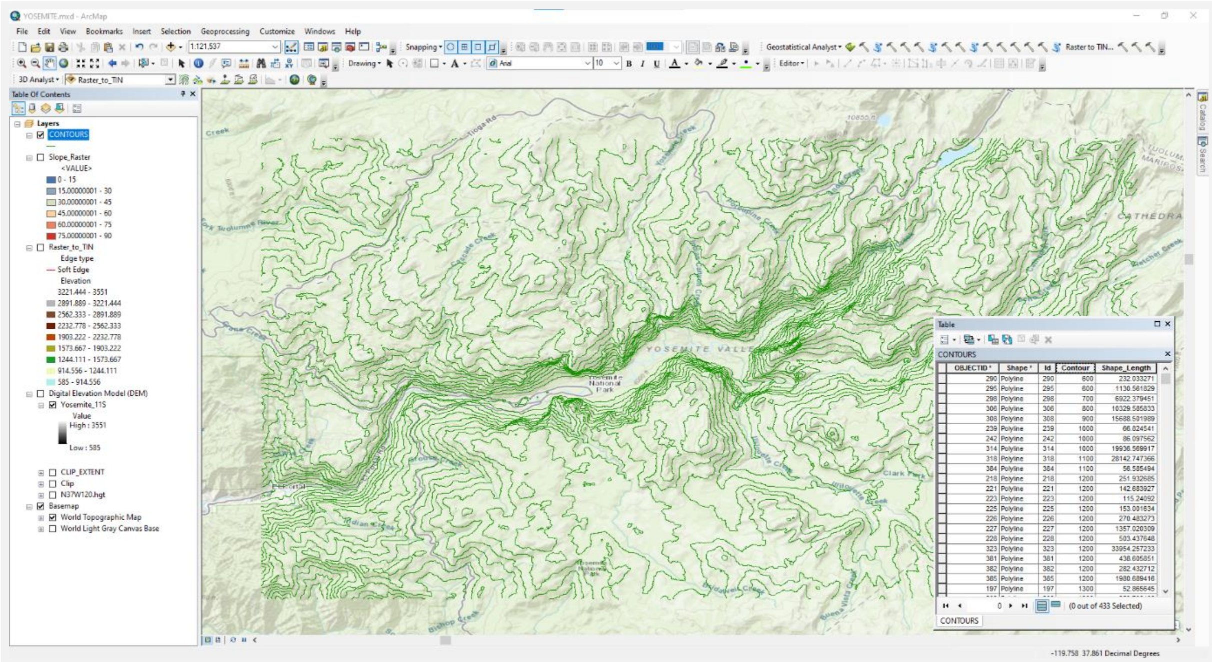
The exported contour shapefile and slope in degrees raster can be viewed in ArcMap while the 3D rendered TIN is also viewable in ArcScene.

1. Open ArcMap and import the contour shapefile and slope raster from OUTPUTS.gdb

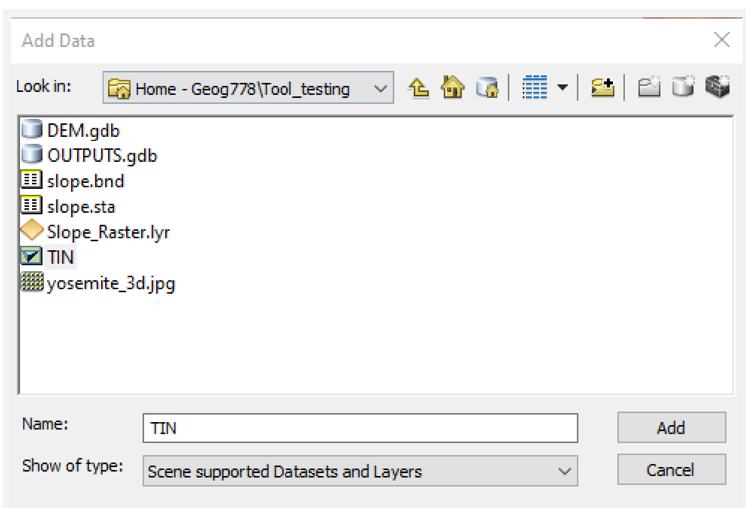


2. Symbolize the slope in degrees raster and contour shapefile as needed.

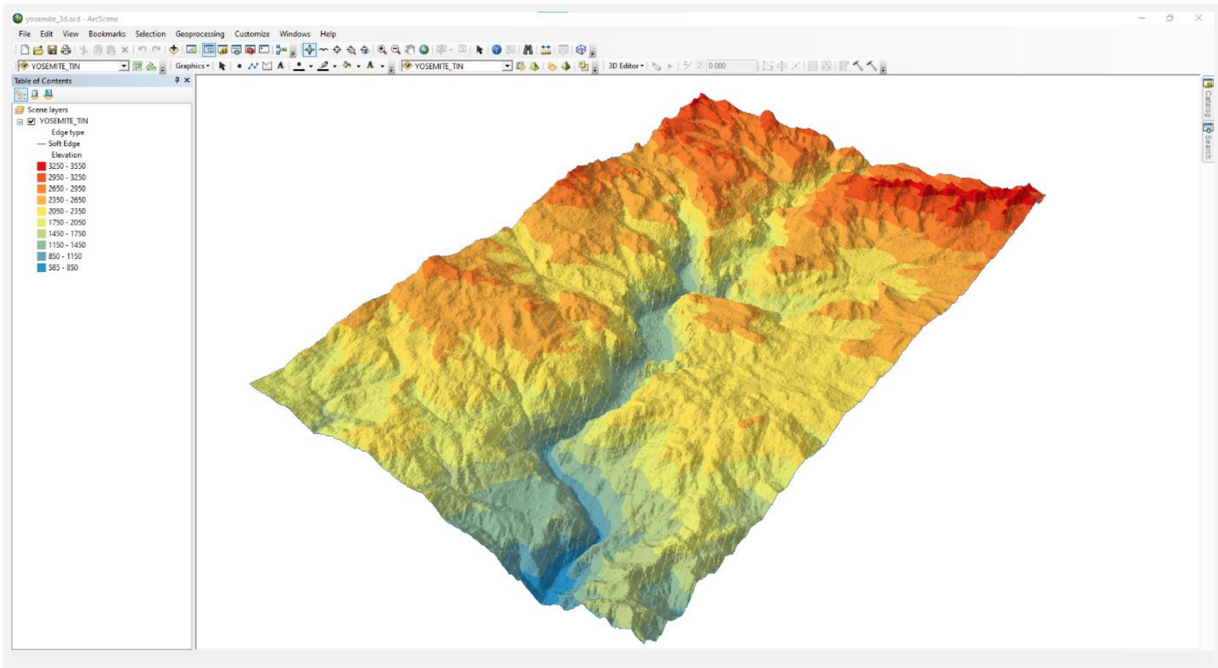




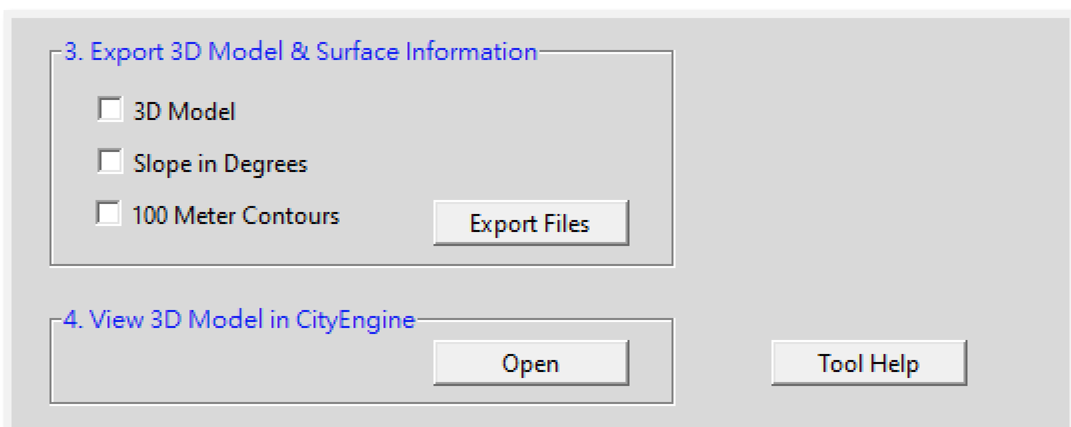
3. Open ArcScene and import the TIN located in the TOOL_TESTING folder.



4. The TIN will import rendered in 3D. Additional options include changing the color classification scheme and exaggerating the base heights.



Tool Help:
The application is equipped with a Help document that provides the standard operating procedures provided above by clicking the Tool Help button in the application.



Limitations within the application – The application performs as expected when providing the tool with a clipped and projected DEM. If the DEM is not clipped to a focused area of interest the tool may experience performance issues. Also, if the DEM is not projected, the tool will export inaccurate surface information. Please ensure the required actions are met prior to exporting surface information files.

In closing – The 3D Python Modeler is a valuable application for analysts in the geospatial field to reduce the amount of processing time to convert a 2D digital elevation model to a 3D rendering viewable in multiple platforms. *Figure 5* depicts the processing time saved is approximately 20 minutes as the tool converted a 2D image to a 3D model in 00:01:12 as opposed to manually processing in 00:21:46. This is an exceptional amount of time saved, especially within an operational environment where minutes and seconds impact mission success or failure.

Test Case	**3D Python Modeler**		
Task number	Task steps and details	Processing Time	Comments
1	Set Target Workspace: Click ***Browse to GDB*** button under ***1. Set Target Workspace*** to navigate to a clipped and projected DEM to import into the application	0:00:10	
2	Convert DEM to TIN: Click ***DEM to TIN*** button under ***2. Convert DEM to TIN*** to convert the imported raster to a TIN	0:00:19	
3	**Export 3D model, slope in degrees, & 100 meter contours simultaniously:** Check all boxes under ***3. Export 3D Model & Surface Information*** to export all 3 options simultaniously	0:00:25	
4	View TIN in 3D Model in CityEngine: Click ***Open*** under ***4. View 3D Model in CityEngine*** to open and view the TIN in 3D in CityEngine Web Viewer	0:00:18	
	Total processing time:	**0:01:12**	

Test Case	**Manual Geoprocessing with ESRI Software**		
Task number	Task steps and details	Processing Time	Comments
1	Open ArcGIS ArcMap: Import clipped and projected DEM	0:01:43	
2	Convert DEM to TIN: Use a geoprocessing tool to convert the DEM to a TIN	0:02:41	
3	Ensure TIN exported: Verify TIN exported	0:00:10	
4	Export slope in degrees in ArcGIS ArcMap: Use a geoprocessing tool to export slope in degrees raster	0:02:51	
5	Ensure slope raster exported: Verify slope exported	0:00:08	
6	Export 100 meter contours in ArcGIS ArcMap: Use a geoprocessing tool to export contours in 100 meter intervals	0:02:23	
7	Ensure 100 meter contours shapefile exported: Verify shapefile exported	0:00:05	
8	Open ArcGIS ArcScene: Export a 3D Model using a geoprocessing tool	0:07:40	
9	View 3D Model in CityEngine: Open CityEngine Webviewer and import 3D Model exported from ArcScene	0:04:05	
	Total processing time:	**0:21:46**	

Figure 5: Time trials - 3D Python Modeler vs ArcGIS manual geoprocessing

The tool not only provides a 3D model but also exports additional data to include contours and slope surface information. The application is easily packaged and delivered to clients who meet the *Technical Components (Page 2)* and is also accessible in a closed offline environment. Future iterations of the tool are to provide users the ability to enter contour intervals based on their project needs – as the tool currently exports contours in 100-meter intervals. Although limitations of the application are noted above, the limitations are avoidable when prior required actions are met.