City of Markham
3D Modelling Project
Project Summary Report
June 2023

Prepared By

WSP

Executive Summary

The City of Markham retained WSP in November of 2022 to develop a 3D model to assist Planning staff with visualization and analysis, and to help streamline the City's development review process (the "project").

The City requested that the model be created using ESRI's CityEngine software to take advantage of its procedural rules and efficient modelling approach. The following objectives were identified by the City during project initiation:

- Develop a 3D model of the Markham Centre and Markham Road - Mount Joy Secondary Plan areas;
- Ability to visualize existing conditions and proposed developments;
- Provide visual aid and site statistics for use during development review;
- Improve efficiency in development approval process; and,
- Utilize visualization capabilities to benefit stakeholder engagement in the planning process.

Based on the needs identified by the City, the project scope includes:

- Geographic coverage area of Markham Centre and Markham Road – Mount Joy Secondary Plan areas;
- Compatibility with both CityEngine and ArcGIS Urban;
- Modelling of both existing buildings and proposed (or conceptual) developments, including building footprints and height at minimum;
- Proposed developments be based on a "snapshot" in time;
- Inclusion of common "base layers," such as aerial imagery, roads, and parcels;
- Site statistics, such as building height, density, Official Plan designation, in-effect zoning, zoning for proposed developments, building uses, site setbacks, Gross Floor Area, Floor Space Index, and lot coverage (where available); and,
- An ArcGIS Urban "Experiment," to explore the tools and capabilities of ArcGIS Urban, and its compatibility with CityEngine.

CityEngine is a desktop 3D modelling program developed by ESRI which allows users to quickly create large virtual urban environments. CityEngine differs from other more traditional 3D modelling programs due to it's use of "procedural" rules and data interoperability.

Instead of users manually modeling individual buildings within the software, CityEngine allows users to create code-based procedures that provide a list of modeling commands to the program. The program then follows these commands to produce models automatically. Rules are written in a native programming language called Computer Generated Architecture (CGA).

CityEngine also supports many industry standard file types and allows for importing and exporting of various data types, including shapefiles, line data and 3D geometry. This allows users to make use of existing data and use it alongside procedural rules.

Models created or imported into CityEngine can then be used for further analysis and support for development review, or can be exported to a wide range of formats for use in other platforms (i.e. renders, graphics, etc.).

ArcGIS Urban is a browser-based 3D modelling software offered by ESRI that allows users to model, visualize and analyze 3D urban environments. While CityEngine is used more broadly in other industries, ArcGIS Urban puts focus on land use planning tools and collaboration between City staff and other stakeholders. ArcGIS Urban has similar (but simplified) capabilities to CityEngine and is intended to be used by Planners as a decision-making tool. The software is accessed and administered using ESRI's ArcGIS Online platform.

This report provides a detailed methodology of how the 3D model was created in CityEngine. It also includes an overview of ArcGIS Urban and the findings from the ArcGIS Urban Experiment. Potential workflow options for using CityEngine or using both CityEngine and ArcGIS Urban are also provided.

Based on the work completed while developing the CityEngine model and our experience undertaking the ArcGIS Urban Experiment, we identified the following conclusions:

- Generally, CityEngine provides more functionality than ArcGIS Urban but requires specialized training.

- ArcGIS Urban is more accessible than CityEngine but is more limited in terms of functionality.
- Interaction between CityEngine and ArcGIS Urban is possible but limited. There are potential workflows options where this setup could work for the City, but this would depend on the approach that the City takes in maintaining both models.
- ArcGIS Urban does contain similar features to CityEngine (i.e. development scenarios, generating models based on rules, reporting and site statistics, analysis tools) but in a more simplified and limited capacity.
- ArcGIS Urban is not a direct replacement for CityEngine.

We also identified the following high-level recommendations for the City to consider when moving forward with the 3D model:

- CityEngine is required at minimum to use the 3D models prepared for this project. We recommend that the City acquire at least a single CityEngine license in order to make use of the model.
- Consider the intended users and the preferred workflows for maintaining the model(s) in the future. This should inform whether to proceed with only CityEngine or incorporate both CityEngine and ArcGIS Urban into the City's operations.
- The City should look to develop procedural rules based on zoning or land use regulations for the CityEngine model to make the most of the software capabilities.
- If the City chooses to use ArcGIS Urban, we recommend that it be used by Planning and Urban Design staff to create design concepts, assist with visualization and analysis, and track development applications. Any more complex tasks are best handled in CityEngine by technical staff.
- ArcGIS Urban provides additional public engagement tools that are not offered in CityEngine. If the City wishes to use the model for public engagement purposes, ArcGIS Urban is a possible consideration.
- If the City is unsure on the viability of using ArcGIS Urban or the number of licenses required to meet the City's needs, it may be worth undertaking a pilot project using a single license of ArcGIS Urban before purchasing additional licenses.

Table of Contents

1.0 Introduction

1.1 Project Background

The City of Markham retained WSP in November 2022 to develop a 3D model to assist Planning staff with visualization and analysis, and to help streamline the City's development review process (the "project"). The City requested that the model be created using ESRI's CityEngine software to take advantage of its procedural rules and efficient modelling approach. The following objectives were identified by the City during project initiation:

- Develop a 3D model of the Markham Centre and Markham Road - Mount Joy Secondary Plan areas;
- Ability to visualize existing conditions and proposed developments;
- Provide visual aid and site statistics for use during development review;
- Improve efficiency in development approval process; and,

The project is being undertaken as part of the Province's Streamline Development Approval Fund, which is intended to help Ontario municipalities modernize processes for managing and approving development applications.

1.2 Project Scope

Based on the needs identified by the City, the project scope includes:

- Geographic coverage area of Markham Centre and Markham Road – Mount Joy Secondary Plan areas;
- Modelling of both existing buildings and proposed (or conceptual) developments, including building footprints and height at minimum;
- Proposed developments be based on a "snapshot" in time;
- Inclusion of common "base layers," such as aerial imagery, roads, and parcels;
- Site statistics, such as building height, density, Official Plan designation, in-effect zoning, zoning for proposed developments, building uses, site setbacks, Gross Floor Area, Floor Space Index, and lot coverage (where available); and,
- An ArcGIS Urban "Experiment," to explore the tools and capabilities of ArcGIS Urban, and its compatibility with CityEngine.

In addition to the creation of the model, the project was conducted using an "exploratory" approach, where the WSP team undertook various experiments to better understand the capabilities and limitations of the software. Potential workflow options for the City are identified in Section 5.0 based on WSP's experience in preparing the model and use of both CityEngine and ArcGIS Urban. The original project scope did not include the creation of procedural rules based on existing or proposed zoning regulations.

WSP also provided two staff workshops to help familiarize staff with the finalized CityEngine model and share the results of the ArcGIS Urban experiment. The first workshop was held for technical staff and was intended for those who will be using and maintaining the CityEngine model. The second staff workshop was held for Planning and Urban Design staff, and focused on the content of the model, as well as features, capabilities, and limitations of ArcGIS Urban.

1.3 Software Overview

1.3.1 CityEngine

CityEngine is a desktop 3D modelling program developed by ESRI which allows users to quickly create large virtual urban environments. CityEngine differs from other more traditional 3D modelling programs due to it's use of "procedural" rules and data interoperability. Instead of users manually modeling individual buildings within the software, CityEngine allows users to create code-based procedures that provide a list of modeling commands to the program. The program then follows these commands to generate models automatically. Rules are written in a native programming language called Computer Generated Architecture (CGA).

CityEngine also supports many industry-standard file types and allows for importing and exporting of various data types, including shapefiles, line data and 3D geometry. This allows users to utilize existing data and alongside procedural rules. Models created or imported into CityEngine can then be used for further analysis and support for development review, or can be exported to a wide range of formats for use in other platforms (i.e. renders, graphics, etc.).

1.3.2 ArcGIS Urban

ArcGIS Urban is a browser-based 3D modelling software offered by ESRI that allows users to model, visualize and analyze 3D urban environments. While CityEngine is used more broadly in other industries, ArcGIS Urban puts focus on land use planning tools and collaboration between

City staff and other stakeholders. ArcGIS Urban has similar (but simplified) capabilities to CityEngine and is intended to be used by Planners as a decision-making tool. The software is accessed and administered using ESRI's ArcGIS Online platform.

Generally speaking, ArcGIS Urban has a more user-friendly interface and does not require as much technical training to use the software when compared to CityEngine. However, ArcGIS Urban models can be synced with CityEngine to allow more technical staff to update or maintain a more complex model in CityEngine, which can then be uploaded to ArcGIS Urban for other users to use. This also allows models or features from CityEngine to be imported to ArcGIS Urban, such as more detailed models. While this functionality does exist, it is somewhat limited in its implementation (see Section 4.5).

ArcGIS Urban models can represent a large geographic area and are generally intended to provide a comprehensive overview of the various projects and initiatives occurring within a municipality.

Section 4.0 covers ArcGIS Urban features and discusses the results of the ArcGIS Experiment in further detail.

2.0 CityEngine Concepts

CityEngine relies upon several key concepts that are essential for using the software to its full capability. The following section outlines some of these concepts to help understand the components, capabilities, and management of the software.

2.1 Workspaces

Workspaces are dedicated file locations on a user's computer which are created by default the first time a user opens CityEngine. This workspace will be the default file location for all CityEngine projects unless another workspace is established by the user. Each CityEngine user will be required to establish a workspace.

Multiple workspaces can be created on a single machine, allowing the user to switch between workspaces as needed. CityEngine will create versions of the default ESRI asset libraries within any new workspace. Any CityEngine project can be imported into a workspace, and multiple projects can be stored within a single workspace. However, only one project can be opened at a time.

Projects can either be copied into a workspace, or linked to a workspace. When a project is copied into a workspace, all files from the project are copied to the user's local machine. When linking a project to a workspace, the project files continue to be saved in their original location but can be viewed and edited by the user. This allows projects to be saved in a central location and edited by multiple users. It should be noted however, that issues can arise when multiple users make edits to a project simultaneously. It is recommended that users to do not edit the models simultaneously to avoid errors or corruption of the project files.

2.2 Projects

CityEngine projects consist of a series of files, including scenes, rules, and data (i.e. GIS or 3D assets) that are required for the project. CityEngine's proprietary default project space includes the folders shown in Figure 1:

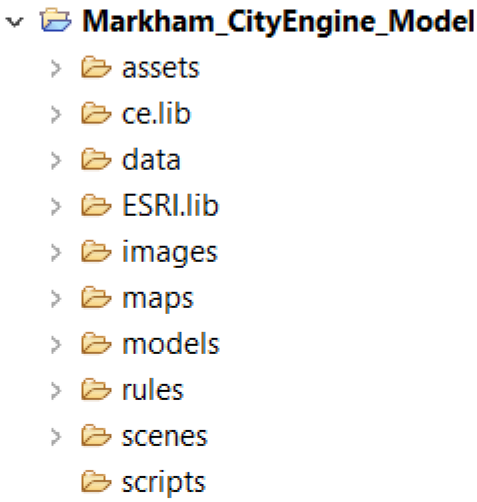


Figure 1: Default Project Folders in CityEngine

A description of each default folder is provided below:

- **Assets:** Default location for 3D models and textures that are used by CGA.
- **Ce.lib: A CityEngine library containing assets, textures, rule templates and other items that can be imported and modified by the user.**
- **Data:** Location to store data that is imported into the model such as GIS datasets, 3D features or other data.
- **ESRI.lib:** Default ESRI library containing assets, textures, rule templates and other items that can be imported and modified by the user.
- **Images:** Imagery such as snapshots or other items can be stored here.
- **Maps:** Stores information used by the map layers. Any data downloaded using the Get Map Data function in CityEngine is stored here.
- **Models:** Default file location for any 3D models that are exported from CityEngine (i.e. for use in other software).
- **Rules:** File location for CGA rule files (.cga).
- **Scenes:** File location for CityEngine Scene files (.cej).
- **Scripts:** A file location to store python scripts (.py).

2.3 Scenes

Scenes are used to display, manage, and interact with the model and its various components. Scenes display the data, models, and other components of the project. Users can interact and navigate the scene using the viewport. Figure 2 below provides an example of a Scene.

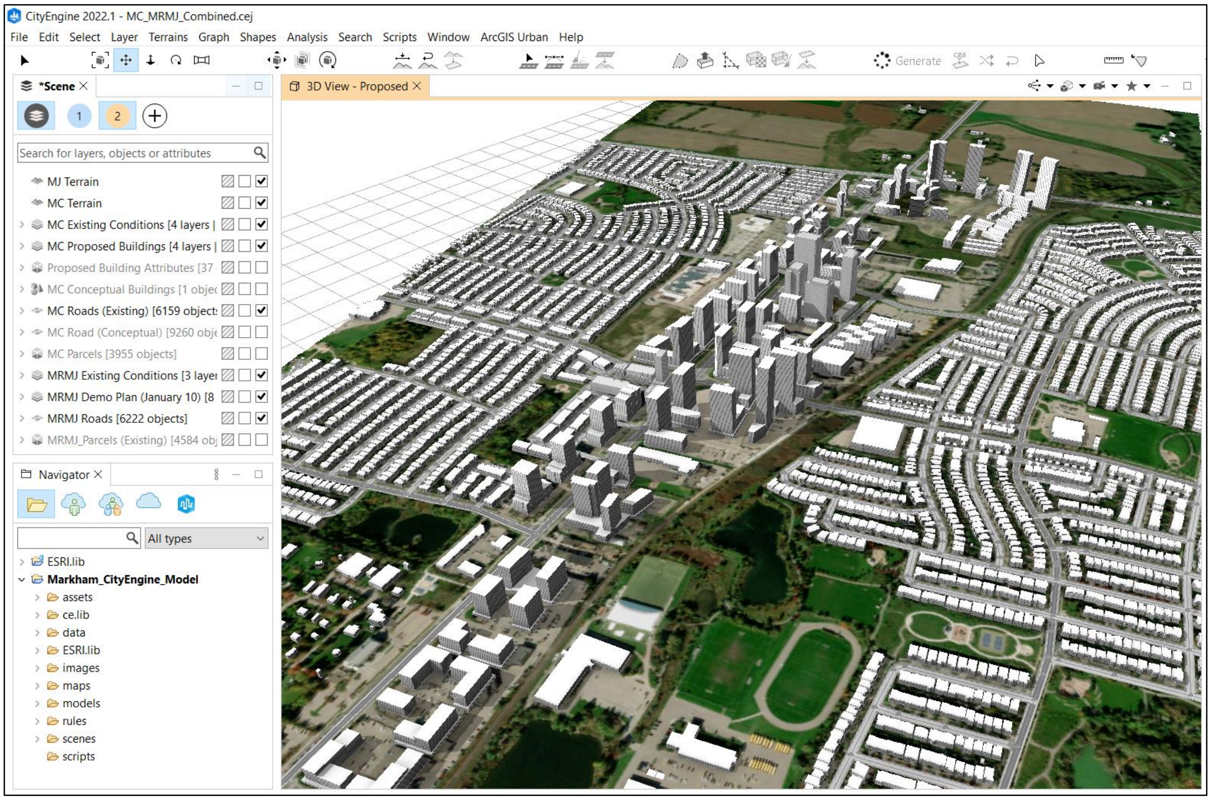


Figure 2: Example of a Scene (and Viewport) in CityEngine

Scenes are saved as .CEJ files in CityEngine and can be imported into a CityEngine project using the import tool. Importing a CEJ file will import the data from the imported scene into the current project. Users can also select which datasets they would like to import into the project.

CEJ files can be susceptible to corruption, especially when being edited by multiple users. It is recommended that users create backup versions of all CEJ files so data is not lost in the event of file corruption or other errors.

2.4 Scenarios

Scenes can be split into multiple development "scenarios", allowing the user to organize a scene into separate model elements. This means a scene can display both the existing conditions and proposed (or alternate) development scenarios. This also allows the user to compare and contrast between existing and proposed scenarios. 3D features can be included in different Scenarios or maintained between multiple scenarios based on the preference of the user. The models created for this project contain an existing scenario and a proposed scenario for both Secondary Plan areas. Figure 3 below provides an example of the Existing and Proposed scenario within the Markham Road – Mount Joy model.

Figure 3: Example of Existing vs. Proposed Scenario in CityEngine

2.5 Data Types

CityEngine also has data types that are specific to the software. The following sections describe these key data types.

2.5.1 Shapes

Shapes are 2D and 3D features that represent buildings or other objects that are generated within CityEngine, typically from building footprint data or those that are procedurally generated from CGA rules. Shapes can be modified by CGA rules within CityEngine.

While the term “Shape” is typically used to described building footprints or 3D buildings in CityEngine, this datatype is also used to represent other objects in CityEngine such as lots (i.e. parcels) and traffic lanes. Generally, when polygon data is imported or created in CityEngine, it is represented by the shape data type.

2.5.2 Static Models

Static models represent existing 3D features that that are imported into CityEngine “as-is.” Static models are not affected by CGA rules (i.e. will not be modified by CGA rules applied in CityEngine) and their geometry cannot be modified within CityEngine. However, static models can be moved, scaled, and rotated within the software. Static models can be useful when the user would like to display an existing or approved building within the model but does not want the building to be modified by CGA rules.

2.5.3 Graphs

Graphs consist of lines or "edges" and points or "nodes" that are used to represent roads or other linear features in CityEngine. When modelling a road network, edges represent road segments while nodes represent road intersections. Graphs can be drawn manually within CityEngine or generated from road centreline data that is imported into the software. Shapes can also be generated within a Graph dataset and are used to represent road lanes and sidewalks within the road right-of-way. Each edge and node in a Graph dataset can contain up to three shapes as "child" elements, where one shape represents the road lane(s) and the other two shapes represent sidewalks. Textures and symbologies can also be applied to these shapes using CGA rules to display road features such as road lines, parking spaces or bike lanes. Figure 4 below provides an example of the data structure of a Graph dataset.

Figure 4: Example of Graph Data Structure

Graphs can be modified by CGA rules, allowing road attributes such as lane number and widths to be changed. When doing so, the shape elements described above will react to any CGA rules applied to the Graph.

Graphs are also used to properly set parcel edges when establishing a model's parcel base. Proper parcel orientation ensures that setbacks and stepbacks will be correctly interpreted by CityEngine when applied through procedural rules.

2.5.3.1 Blocks and Lots

When a road network is created in CityEngine, CityEngine also recognizes the areas between the road right-of-ways as areas where buildings can be generated (i.e. land for development). When a closed loop of roads is created in CityEngine (either from drawing a roadway manually, generating a road using CGA rules or importing street line data), a "Block" is created which represents the area between the roadways, as shown in Figure 5.

A Block is tied to the surrounding streets, and will adjust automatically when the street network is moved or modified. Blocks can be subdivided into "Lots" (i.e. parcels) which are represented as shapes, allowing CGA rules to be applied. Therefore, buildings can be generated on lots using CGA rules.

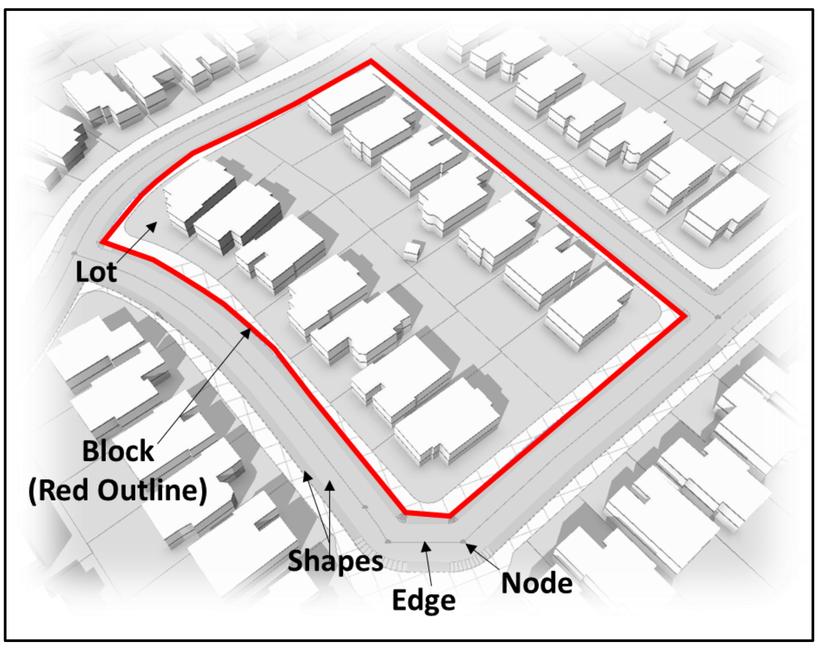


Figure 5: Components of a Graph dataset - Nodes, Edges, Shapes, Blocks & Lots

2.6 Procedural Rules

CityEngine allows users to write procedural rules to generate or modify buildings (shapes), roads (graphs) and lots (shapes) automatically. This means that buildings (or other features) can be quickly generated based on a defined procedure, rather than having to model features manually as one would using traditional modelling software (i.e. Sketchup, 3ds Max, etc.). This can save a great deal of time and resources when modelling large areas and allows users to program zoning regulations directly into the software and generate plausible buildings based on those regulations. These capabilities make CityEngine particularly useful for land use planning and urban design applications.

Procedural rules are written in Computer Generated Architecture (CGA), CityEngine's native programming language. Rules are highly customizable and can be used to perform a wide variety of operations, including extruding building footprints, applying zoning regulations, modelling roads, mapping textures, and generating metrics.

Users can also create procedural rules within ArcGIS Urban using a graphical interface. This means that no programming knowledge is required to create rules in ArcGIS Urban. However, the rules established in ArcGIS Urban are simpler in nature, and mainly pertain to applying zoning rules (i.e. setbacks, stepbacks, max height, etc.) to buildings. Creating rules in ArcGIS Urban is further discussed in Section 4.0.

3.0 Methodology

The following sections explain the methodology used to develop the models for both the Markham Centre and Markham Road – Mount Joy Secondary Plan areas.

3.1 Data Provided

The City provided a variety of GIS datasets for both the Markham Centre and Markham Road – Mount Joy Secondary Plan areas. The initial datasets provided by the City included the following:

- **Aerial Photo 2022:** Aerial imagery prepared by the City for use as the model "terrain."
- **Address_Points:** Point dataset with address information for all properties within the Secondary Plan areas. Used to assign addresses to parcels within the model.
- **DTM_100CM:** Digital Terrain Model containing elevation information for both Secondary Plan areas. Used for data processing when determining heights (see sections 3.3 and 3.4 below).
- **Exist_Buildings_York_Region:** Building footprint data with heights for existing buildings, prepared by York Region and provided in GIS format. This dataset was not comprehensive and did not include all existing buildings within the Secondary Plan areas.
- **Exist_Buildings_Markham:** Building footprint data for existing buildings with heights, prepared by City of Markham and provided in GIS format. This dataset was provided to fill gaps within the York Region dataset.
- **Model_Boundary:** Boundary prepared by the City showing the preferred extent of the model areas. Used for reference when setting up the model.
- **OP_2014_Map_3_Land_Use:** Official Plan land use designations for use in model site statistics.
- **Parcel:** Land parcel information for both Secondary Plan areas. Used as a base layer within the model and a starting point for generating buildings based on procedural rules.
- **Road_Asset:** Road polygon layer depicting road right-of-ways. Used initially in the model to display roadways but later replaced by Single_Line_Road_Network dataset (see sections 3.3 below).

- **Secondary_Plan_Boundary:** Boundary layer depicting the boundaries for both Secondary Plan areas. Used for reference when setting up the model.
- **Single_Line_Road_Network:** Road centreline data depicting roadways. Includes road names and road class attributes. Used to as road dataset within both models.
- **Zoning:** Zoning layer showing zone polygons and in-effect zoning for all properties within each Secondary Plan area. Used for site statistics within the model.

In addition to the GIS data identified above, additional datasets, 3D features and other materials were provided throughout the course of the project. The data and materials provided varied between the two Secondary Plan areas based on specific area requirements and data availability. The following items were provided in addition to the initial GIS data:

Markham Centre

- Sketchup model for the entire Secondary Plan area, including existing, proposed, and conceptual buildings.
- Proposed development concept overlay with future road network in AutoCAD (.dwg) format.
- Development application data for proposed developments, for use in preparing site statistics for proposed buildings.
- Digital Surface Model (DSM) containing height information for all of the Markham Centre Secondary Plan area. This data was used to correct heights and massing for existing buildings within the model.

Markham Road – Mount Joy

- Markham Road – Mount Joy Demonstration Plan depicting the conceptual buildout of the Secondary Plan area. Multiple versions of the Demonstration Plan were provided, including versions from April 2021, July 2022, December 2022 and January 2023. The proposed/conceptual buildings in the final model were based on the January 2023 version of the Demonstration Plan.
- **Demo_Plan_Buildings:** Building footprint data for the Markham Road – Mount Joy Secondary Plan area based on the April 2021 version of the Demonstration Plan. Used initially for modelling proposed buildings (later replaced by footprints from the January 10th version of the Demonstration Plan).

- **Demo_Plan_Parcels:** Conceptual parcel fabric for the Markham Road – Mount Joy Secondary Plan area based on the April 2021 version of the Demonstration Plan. Included as a reference layer in the proposed scenario for the Markham Road – Mount Joy model.
- **Demo_Plan_Parks:** GIS layer depicting proposed parks based on the April 2021 version of the Markham Road – Mount Joy Demonstration Plan. Included as a visual reference in the Markham Road – Mount Joy model.
- 3D models for three proposed buildings within the Markham Road – Mount Joy Secondary Plan area in COLLADA (.dae) file format (1709 Bur Oak Ave, 9781 Markham Rd & 9900 Markham Rd). These models were provided to the City by the applicant and provided to the project team for inclusion in the model.
- Digital Surface Model (DSM) containing height information for all of the Markham Road – Mount Joy Secondary Plan area. This data was used to correct heights and massing for existing buildings within the model.

3.2 Approach Overview

The general approach for developing both the Markham Centre and Markham Road – Mount Joy models is outlined below. Sections 3.3 and 3.4 provide a more detailed overview of the steps and processes that were applied for each Secondary Plan area. The approach involved the following steps:

1. **Data processing & preparation:** The initial step involved reviewing, gathering, and preparing the data provided by the City. Certain datasets required additional processing before being imported into CityEngine. Any processing of GIS data provided by the City was completed in ArcGIS or other GIS/database software.
2. **Experimentally importing City data:** After processing the data, the datasets were imported to CityEngine using the program's import tool. In certain cases, some experimentation was required to achieve desired results.
3. **Modelling existing conditions:** Once imported, the terrain, roads and parcel data were used to form the "base layers" of the model. The existing building footprints were then aligned to the terrain and extruded based on height attributes within the GIS datasets. Any 3D features representing existing buildings were also imported and compared against the building footprint data to identify which provided a higher level of detail. Final

QC and cleanup steps were applied to the model to ensure features were placed and displaying properly.

4. **Modelling proposed buildings:** The approach for modelling proposed buildings differed between the Secondary Plan areas. For Markham Centre, the proposed and conceptual buildings from the provided Sketchup model were imported and placed within the model. For Markham Road – Mount Joy, building footprints were derived from the December 2022 version of the Demonstration Plan and imported into CityEngine. The building footprints were then extruded based on heights defined in the Demonstration Plan. Once the January 10th, 2023 version of the Demonstration Plan was provided, the model was updated to reflect the changes. The 3D models of the three proposed properties were also directly imported into CityEngine. Final QC and cleanup steps were also applied to the proposed building models to ensure features were placed and displaying properly.
5. **Deriving key site statistics:** Key site statics were derived primarily from the GIS datasets provided by the City. Site statistics were assigned to the parcel data, including address, existing zoning, official plan designation, lot coverage and setbacks. Rules were created to automatically estimate Gross Floor Area (GFA) and Floor Space Index (FSI).
6. **Experimentally exporting models:** Once finalized, the project team experimented with exporting the models to various formats, including KML, shapefile/geodatabase, .dwg, COLLADA (.dae), ESRI Scene Layer Package and CityEngine Webscene. The results of this experimentation are outlined in Section 3.7.

In certain cases, the project team revisited previous steps (i.e. data processing, importing) to address issues discovered in the original source data, update aspects of the modelling process and to provide additional value-added enhancements requested by the City. Section 3.3 and 3.4 below provide a detailed explanation of the methodology and processes undertaken for both Secondary Plan areas as part of the project.

3.3 Markham Centre Secondary Plan Area

The following subsections describe the approach and processes which were used to generate the Markham Centre model.

3.3.1 Data Processing

Before importing the data provided, some initial processing was required to ensure that the models would display properly when imported into CityEngine.

The building footprint dataset provided by the City (Exist_Buildings_Markham) expressed building heights above sea level rather than from ground level. Extruding the building footprints based on this initial sea level-based value would have result in exaggerated heights relative to the majority of other data provided which is encoded relative to ground level. To correct this, the project team subtracted the elevation value in the building footprint data from the ground elevation provided in the DTM (DTM_100CM). A new height field was then created within the building footprint layer and updated with the resulting height values. The dataset was then imported into CityEngine.

Some of the GIS datasets provided also extended beyond the Secondary Plan boundaries and needed to be clipped for proper display within the model. This was completed in ArcGIS before importing the datasets into CityEngine.

3.3.2 Import Steps

Once the required data processing was completed, the various datasets were imported into CityEngine using the import tool. This involved importing the terrain, parcels, roads, building footprints and road asset layers. In addition, the 3D features from the Sketchup model were imported.

3.3.2.1 Aerial Imagery

We initially encountered issues when attempting to import the aerial imagery provided by the City. We experienced several program crashes when attempting to import the original TIFF image. In an attempt to resolve the issue, we converted the TIFF image to PNG and tried reimporting the image to see if CityEngine would respond differently to a different file type. We experienced similar errors and crashes when attempting to import the PNG file as well.

Additional research indicates that CityEngine has a maximum image size limit of 8192 x 8192 pixels. As a final workaround to establish the terrain within the model, we used the “Get Map Data” tool in CityEngine to retrieve aerial imagery and contour information directly from ESRI servers. We completed a comparison between ESRI’s imagery and the imagery provided by the City and found that while the ESRI imagery had a lower resolution than the City’s imagery, it was still quite high. After confirming that the ESRI imagery was suitable for project purposes, we proceeded with using the ESRI imagery to ensure the terrain could be properly generated and adequate performance would be maintained.

3.3.2.2 GIS Data

The GIS data (feature classes) provided by the City were imported directly via CityEngine’s import tool. This included the parcel, road (Road_Asset & Single_Line_Road_Network) and building footprint (Exist_Buildings_York_Region & Exist_Buildings_Markham) datasets.

3.3.2.3 3D Features

To import the Sketchup model into CityEngine, we first exported the model to COLLADA (.dae) format from Sketchup so it could be imported into CItyEngine. CityEngine does not support the import of Sketchup files (.skp) directly, so this is a necessary step when working between Sketchup and CityEngine.

Our first attempt at importing the model into CityEngine resulted in errors stating that the models were “too far from their origin.” To correct this issue, we georeferenced the models in Sketchup and tried reimporting them into CityEngine.

Our next attempt resulted in an error stating that the “data extent was too large.” After reviewing the Sketchup model in further detail, we identified that the original Sketchup model had been scaled in millimetres but the units were set to metres within the Sketchup document. This resulted in the model being much larger than what was intended. After notifying the City, a rescaled model was provided, which was converted to COLLADA and imported into CityEngine as static buildings. 3D features imported into CityEngine will maintain the scale that they were created in. This means it is important to ensure that proper scaling is applied within the original model before importing any existing 3D features into CityEngine. CityEngine does allow models to be rescaled within the software, but the exercise is less precise than properly scaling a model before importing it.

After importing the Sketchup model as static buildings, we noted that all buildings were treated as one object in CityEngine. To ensure that the buildings were properly separated, we then exported them in groups based on their current development status as indicated in the Sketchup model (i.e. existing, proposed, or conceptual). Each grouping was then re-imported into CityEngine and further separated within CityEngine afterwards (see next section).

3.3.3 Modelling Existing Conditions

Once the data was imported into CityEngine, we proceeded with modelling the existing conditions for the Markham Centre area. This involved establishing the terrain, road and parcels layers, followed by modelling the existing buildings. An “Existing” development scenario was created to contain all of the elements of the existing conditions.

3.3.3.1 Establishing Terrain

As mentioned in section 3.3.2.1, we utilized CityEngine’s “Get Map Data” tool to retrieve ESRI aerial imagery and generate the terrain for the model due to size limitations with the aerial imagery provided. The “Get Map Data” tool allows the user to select a location and CityEngine will then retrieve the aerial imagery and elevation data for the specified location. This allows the user to quickly establish “base data” when creating a new model.

3.3.3.2 Assigning Parcel Edges & Generating Graphs

We used the “Align Shapes to Terrain” tool in CityEngine to align the parcel data to the terrain. For display purposes, we aligned the parcels so that they were offset slightly above the terrain to ensure they would be visible to the user when the layer is turned on. This was due to some of the parcels being partially hidden underneath the terrain in areas with varying elevation.

Similarly, we used the “Align Streets to Terrain” tool in CityEngine to properly align the road data to the terrain.

We then used CityEngine's "Compute Edge Attributes" tool to assign the appropriate edge orientation to the parcel data. The "Compute Edge Attributes" tool assigns an edge orientation to each side of a parcel based on its relationship to the road (i.e. a parcel edge fronting a street will be assigned a "front" orientation. Based on the assignment of the "front" edge, the side edges of the parcel will be assigned a "side" orientation, and the rear edge of the parcel will be assigned a "rear" orientation). The parcel edges need to be properly set to ensure that any setbacks assigned using procedural rules are interpreted correctly by CityEngine (i.e. the front setback distance is correctly applied to the "front" edge of the parcel, the side setback distance is correctly applied to the side edge of the parcel, etc.).

We used the Road_Asset dataset in our first attempt at setting the parcel edges, which did not successfully assign the edge orientation due to it being a polygon layer rather than a graph dataset (see section 2.5.3 for information on graphs). As a result, we used the Single_Line_Road_Network road centreline data to generate a graph dataset. Once the graph dataset was generated, we re-ran the "Compute Edge Attributes" tool on the parcel dataset, which successfully assigned the proper edge orientations to the parcel dataset.

3.3.3.3 Extruding Building Footprints

After importing the York Region and City of Markham building footprint datasets, we aligned them to the terrain using CityEngine's "Align Shapes to Terrain" tool. We set the "Align Function" parameter in the Align Shapes to Terrain tool to "Project All" and set the offset amount to zero to ensure that each building footprint would be properly aligned to the correct elevation relative to its location.

Once properly aligned, we proceeded with extruding the building footprints based on the height field (in metres) included within the data's attribute table. As described in Section 3.3.1, the corrected building heights within the Exist_Buildings_Markham dataset were used instead of the elevation data that was originally provided.

To extrude the building footprints, we created a procedural rule which instructs CityEngine to retrieve the height information from the height field within the building footprint data and generate the building based on the height information. To use the procedural rule, it must be assigned by selecting all features within a dataset (right click and select "Select Objects") and selecting the “Assign” button in the “Rules” tab within the Inspector window. The user can then navigate to the rule they wish to assign and select it.

The proper height field must also be assigned by clicking the dropdown menu beside the “BUILDINGHE” parameter and selecting “Connect Attribute.” The “BuildingHe” field can then be selected from the list provided (Note, that the rule has already been configured in the existing model and should not need to be re-connected). The “Rules” tab and “Connect Attributes” option are shown in Figure 6 below.

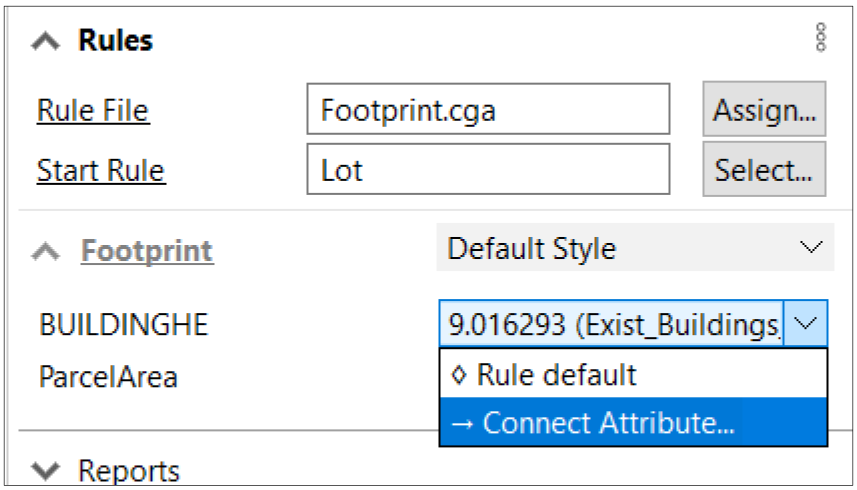


Figure 6: Assigning a Rule and Connecting Attributes in CityEngine

3.3.3.4 3D Features

As mentioned in Section 3.3.2.3, the existing 3D features from the Sketchup model were imported into CityEngine as static models in groups based on their status (i.e. existing, proposed, and conceptual). However, each of these groups were still treated as a single object once imported into CityEngine. To correct this, we first converted all of the buildings imported from the Sketchup model from static models to shapes using the “Convert Models to Shapes” tool in CityEngine. While this did successfully convert the buildings to shapes, they were still being treated as one object in CityEngine.

CityEngine does not have a native “explode” tool that can be used to separate groupings of features into individual items. Due to this limitation, we identified a workaround which would allow us to separate the group features into individual features. This was completed by first using CityEngine’s “Separate Faces” tool to separate all the buildings into individual components or “faces” (i.e. split the features into individual surfaces rather than individual buildings), then using CityEngine’s “Combine Shapes” tool to group the individual “faces” back together for each individual building. This was done by selecting all components or “faces” of each building and running the “Combine Shapes” tool. While this method was successful in separating the grouped buildings into individual buildings, it was highly labour intensive. It is recommended that the City import existing 3D features into CityEngine individually to avoid this approach in the future.

Once the 3D features had been separated, we used CityEngine’s “Cleanup Shapes” tool to smooth out building surfaces and improve the overall appearance of the models.

Since we both generated buildings from the building footprint data and imported existing buildings from the Sketchup model, the model contained some overlapping datasets that needed to be consolidated. Where the datasets overlapped, we undertook a manual visual comparison between the buildings generated from building footprint data versus those that were imported from the Sketchup model to determine which looked best in appearance and provided the higher level of detail. In many cases the buildings imported from the Sketchup model provided a higher level of detail (i.e. variation in heights) than those based on the building footprint data and were therefore kept in the final version of the model.

3.3.4 Modelling Proposed Development

The terrain, parcels and roads established for the Existing scenario were used as the base layers for the Proposed scenario.

The proposed buildings within the Markham Centre model were also imported from the Sketchup model provided. These buildings were imported into CityEngine as static models and were grouped into “proposed” and “conceptual” buildings based on the information provided in the Sketchup model. Once imported, we moved the grouped buildings into their correct positions within the model and aligned them to the terrain. Since the buildings are treated as one object when grouped, the buildings did not properly align and needed to be re-aligned to the terrain once the buildings were separated.

The same approach described in Section 3.3.3.4 was used to convert the two groups of static models to shapes, and then separate them into individual buildings. Similar to the Existing scenario, we used CityEngine's "Cleanup Shapes" tool to correct any discrepancies and improve the appearance of the models. We re-aligned the buildings to the terrain using CityEngine's "Align Shapes to Terrain" tool to ensure each building was aligned appropriately to the ground elevation.

In addition to modelling proposed buildings, we also modelled the proposed road network identified by the City for use in the Proposed scenario of the model. The City provided an AutoCAD (.dwg) file containing the conceptual road network for a Markham Centre concept plan. We imported the AutoCAD file into ArcGIS and separated the road right-of-way from the rest of the drawing components. We then converted the right-of-way to a polygon within ArcGIS, and then derived the centreline from the road polygon. We then imported the resulting road centreline data into CityEngine and converted it to a graph so it could be represented in the model and react appropriately to procedural rules. The proposed road network was included as a separate layer in the Proposed scenario (i.e. is separate from the primary road layer).

We then undertook a quality control exercise to ensure that buildings were properly aligned, scaled, and displayed properly within the Proposed scenario.

3.4 Markham Road-Mount Joy Secondary Plan Area

The following sections describe the steps and processing used to develop the Markham Road – Mount Joy CityEngine model.

3.4.1 Data Processing

Some data processing was required to implement building footprints from the later versions of the Demonstration Plan and to introduce varied heights into the building footprint data. These processes are outline in Sections 3.4.4.1 and 3.4.4.2 below.

3.4.2 Import Steps

3.4.2.1 GIS Data

The GIS data provided by the City was imported directly via CityEngine's import tool. This included the parcel, road (Road_Asset & Single_Line_Road_Network) and building footprint (Exist_Buildings_York_Region & Exist_Buildings_Markham) datasets. The datasets related to the

Markham Road – Mount Joy Demonstration Plan (Demo_Plan_Buildings, Demo_Plan_Parcels & Demo_Plan_Parks) were also imported directly into CityEngine.

3.4.2.2 3D Features

Three 3D models for proposed developments (1709 Bur Oak Ave, 9781 Markham Rd & 9900 Markham Rd) were provided in COLLADA (.dae) file format for the Proposed scenario in the Markham Road – Mount Joy model. These models were imported as static models directly using CityEngine's import tool.

3.4.3 Modelling Existing Conditions

3.4.3.1 Terrain, Roads, Parcels & Building Footprints

Once the data was imported into CityEngine, we followed many of the same steps as what was completed for the Markham Centre Model. This included using CityEngine's "Get Map Data" tool to retrieve ESRI aerial imagery and generate the terrain for the model. We then aligned the parcel data and road data to the terrain and established the parcel edges using CityEngine's "Compute Edge Attributes" tool.

We then aligned the building footprint data (Exist_Buildings_York_Region & Exist_Buildings_Markham) to the terrain and used the same procedural rule and method identified in section 3.3.3.3 to extrude the building footprints based on the heights included in their attribute tables.

3.4.3.2 3D Features

For the 3D features that were provided in COLLADA format, we first aligned the buildings with the terrain and placed them in their proper location. These buildings were imported individually, so we did not experience the same "grouping" issues as we did when importing the buildings from the Sketchup model in Markham Centre. However, we did convert the three static models to shapes using CityEngine's "Convert Models to Shapes" tool.

The City identified that two buildings for the 1709 Bur Oak Ave development were already built, with the final building yet to be constructed. For this reason, the two existing buildings were included in the Existing scenario with the proposed building being included in the Proposed scenario.

3.4.3.3 Editing Terrain

Certain areas within the model contained uneven terrain due to the current elevation information included in the terrain data. An example of this was on the proposed GO Station site at the southeast corner of Markham Road and Major Mackenzie Dr. Dirt mounds from the current earthworks being undertaken in the area were captured within the terrain data, meaning the grade was not level (as it would be in the future after the lands are developed). Modelling buildings on this section of terrain resulted in uneven or tilted buildings, which did not reflect real life conditions. To address this issue, we used CityEngine's "Terrain Edit" tool to flatten the terrain within this area to a common elevation, reflecting what the eventual graded site will look like once constructed.

3.4.4 Modelling Proposed Development

The terrain and road data established for the Existing scenario were used as the base layers for the Proposed scenario in the Markham Road – Mount Joy Secondary Plan area. Both the existing parcels and the parcels from the April 2021 version of Demonstration Plan were included as reference layers in the Proposed scenario as well.

Initially, we modelled the proposed buildings based on the building footprints provided in the Demo_Plan_Buildings dataset. Similar to the approach identified for the existing conditions, we extruded the building footprints based on the height attributes included in the Demo_Plan_Buildings dataset. However, this dataset was based on the April 2021 version of the Demonstration Plan and therefore did not reflect the most recent version of the development concept.

3.4.4.1 Deriving Building Footprints from the Demonstration Plan

The December 9th version of the Demonstration Plan provided by the City was prepared in Adobe Illustrator and provided in PDF format. To extract the building footprints from the PDF document, we first opened the PDF in Bluebeam Revu and turned off all layers in the PDF besides those containing the building footprints, and exported the isolated building footprints to a separate PDF (the "December 9th building footprints"). We them imported this separate PDF containing only the December 9th building footprints into AutoCAD. We then imported the Demo_Plan_Buildings and Demo_Plan_Parcels datasets provided by the City into AutoCAD as well to provide existing georeferenced features that could be used to georeference the December 9th building footprints from the PDF. We then scaled and aligned the December 9th building footprints to the Demo_Plan_Buildings and Demo_Plan_Parcels datasets, creating a georeferenced version of the December 9th building footprints.

Once georeferenced in AutoCAD, we saved the December 9th building footprints as a DWG file, imported them into ArcGIS and converted them to a feature class. We also georeferenced the full December 9th Demonstration Plan (with all PDF layers) to serve as a reference in ArcGIS when making any adjustments to the December 9th building footprint feature class. We then manually adjusted the building footprints in ArcGIS as required based on the georeferenced Demonstration Plan and assigned the various heights from the December 9th Demonstration Plan to the newly created building footprint dataset. Once the new December 9th building footprint dataset was prepared, we imported it into CityEngine and extruded the footprints based on the defined heights.

The City later provided an updated Demonstration Plan from January 10th, 2023 with some minor changes to the development concept. We then updated the model to reflect the changes shown in the January 10th Demonstration Plan. These changes mainly consisted of assigning certain buildings to the Existing development scenario instead of the Proposed scenario within the model.

As mentioned in section 3.4.3.2, we imported COLLADA files for three proposed developments which were originally provided by the applicant. These existing 3D features were imported as static models, then placed in their appropriate location and scaled to fit within the surrounding context. Those buildings identified as proposed were then assigned to the Proposed development scenario.

We identified that the 3D model provided for 9900 Markham Road did not reflect the most recent design shown in the project's development application and notified the City of the discrepancy. The City confirmed that the design shown in the 3D model was outdated, and directed us to model the development based on the building footprints shown in the Demonstration Plan. We proceeded with modelling the buildings for 9900 Markham Road based on the footprints in the Demonstration Plan.

3.4.4.2 Adjusting for Height Variations

After a review session with the City, it was identified that the massing for certain buildings was not reflective of their real-life counterparts, and there was some detail lacking for buildings with height variations. This was due the building footprint data having only a single height attribute versus having sections with different heights delineated within the building footprint data.

To address these issues and introduce an additional level of detail to the building models, we requested that the City provide a Digital Surface Model (DSM) so we could identify building height variations and integrate them within the building footprint data. However, both data sources used metres above sea level to express their elevation, meaning some processing was required to derive the true building height expressed in metres above grade. This result can be achieved by subtracting the DSM from the DTM to normalize the height in terms of height above grade.

This was done by using the “Raster Calculator” tool in ArcGIS to subtract the elevation information in the DSM from the DTM. A DSM provides elevation information for all objects above grade (i.e. buildings, trees, other structures), and the ground surface (i.e. contours on the ground). A DTM provides elevation information for only the ground surface itself (i.e. no buildings, trees or structures are reflected in a DTM). By subtracting the elevation information of the DSM from the DTM, we were left with the height of only the objects which were higher than the ground surface (i.e. buildings), expressed in metres. This allowed us to identify building heights, including height variations for any given building.

Once the normalized DSM was created, we used the new height information to identify those buildings with height variations and separated the building footprints into different sections where the height changed. This resulted in certain building footprints being separated into multiple features. We then assigned different height attributes to each section of the building based on the normalized DSM. Once complete, we imported the updated building footprint data into CityEngine and extruded the footprints based on the heights. This resulted in the building models having varying heights where the sections were identified. The Image on the left in Figure 7 below shows an example of the buildings based on a single height attribute, while the image on the right shows the buildings with the different heights sections delineated.

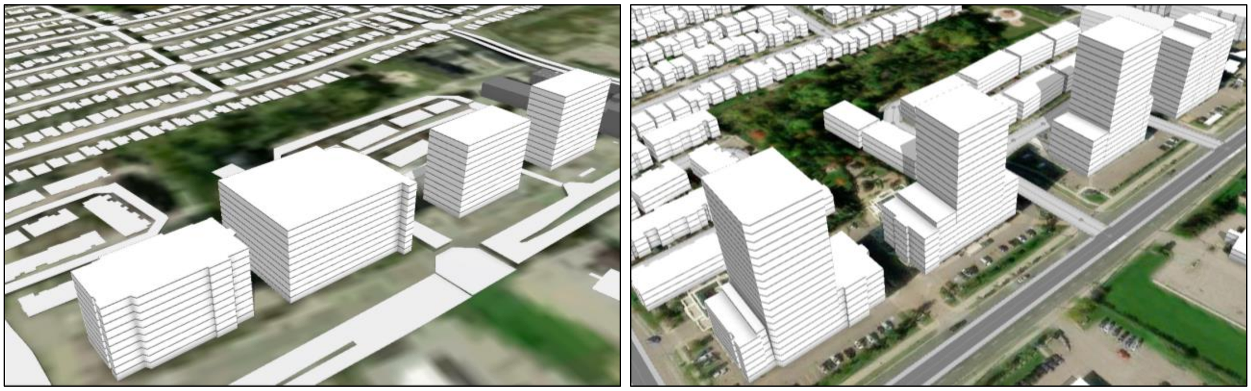


Figure 7: Level of Detail Comparison - Single Height Attribute vs, Delineated Heights

As mentioned in section 3.4.3.3, we needed to correct some of the terrain elevation for the existing conditions in order for the building models to display properly. We used the same updated terrain data within the Proposed scenario to ensure that the conceptual buildings from the Demonstration Plan would be displayed properly.

3.5 Deriving Site Statistics

As part of the project scope, the City requested that WSP derive a number of site statistics to allow staff to quickly view site information when working in the model. We prepared sites statistics for both existing and proposed sites, which are outlined further below:

3.5.1.1 Existing

For existing sites, much of the processing and calculations used to derive the statistics were completed outside of CityEngine, in ArcGIS or other GIS software. The GIS datasets provided by the City were the primary data sources for these statistics. The resulting site statistics were incorporated into the model's parcel dataset.

We incorporated the sites statistics within the parcel data for the following reasons:

1. We wanted to keep all statistics assigned to one dataset to avoid confusion (i.e. rather than having statistics tied to multiple datasets) and for ease-of-use when maintaining the data moving forward.

2. Attribute information is not easily assigned to models that were imported from existing 3D features. Models generated from GIS data (i.e. building footprint data) already have attributes associated with them, making it easier to integrate the site statistics to the associated attribute information. It is possible to add attributes to shape features in CityEngine but the process requires manual input of all attribute information, which can be very time consuming.
3. String or other GIS attribute data cannot be displayed within CityEngine's "Reports" window (described further below). The Reports window is primarily used to calculate certain metrics automatically based on selected model features (i.e. GFA and FSI).

For these reasons, we determined that the best approach would be to store and manage the site statistic information within the parcel data since it provided full coverage of the secondary plan area and is independent from the building features (i.e. can still provide stats for buildings that were imported from the Sketchup model).

The site statistics that we derived for the existing conditions include:

- Address;
- Official Plan designation;
- In-effect zoning;
- Lot coverage; and,
- Setbacks;

The address information was applied to the parcel data by completing a spatial join between the Address_Points dataset and the parcel dataset in ArcGIS. A new field called FULL_STREE was added to the data containing the full address for each parcel.

Similarly, the Official Plan and in-effect zoning information was applied using a spatial join in ArcGIS between the parcel data and the OP_2014_Map_3_Land_Use and Zoning datasets. When completing this process, we noticed that in certain cases the parcels did not perfectly line up with the zoning polygons contained within the Zoning dataset, resulting in some parcels having more than one zone assigned to it. After discussing with the City, it was determined that the best method to address this issue was to complete the spatial join based on the centroid of the Official Plan and zoning data. We proceeded with this approach and added the resulting information to the OP_Landuse and Zone fields in the parcel data.

The lot coverage was calculated in ArcGIS by dividing the area of the building footprint by the parcel area and converting to a percentage. This was applied to all parcels containing a building and populated in the LotCover field within the parcel data.

Setbacks were derived for existing buildings based on distances between each parcel edge and the building footprint on the parcel. We first attempted to create a procedural rule to determine the setbacks based on this concept but were unsuccessful. We then developed an alternative approach using an SQL script to determine the setback distances.

The process involved first separating all parcels into individual line segments so that each parcel edge could be assigned an orientation (i.e. front, side & rear). We then created a buffer around the road centreline which was used to select the parcel edge located closest to the road. We assigned this parcel edge as the “front” edge. After determining the front edge, we then assigned the “side” edge orientation to those edges which touched the front edge. The “rear” edge orientation was then assigned to the edge that was opposite the front edge.

Once the edge orientations were assigned, we calculated the distance from each parcel edge to the closest edge of the building footprint for each parcel. The setback distances were then consolidated based on their associated parcel and joined to the main parcel dataset based on the parcel ID number. The setback information was then populated in the main parcel dataset in the Front, Side and Rear fields.

In addition to the site statistics incorporated in the parcel data, a procedural rule was created to calculate gross floor area (GFA) and floor space index (FSI) for any buildings that are currently selected within the model. The resulting calculations are displayed in CityEngine’s Reports tab (in the Inspector window) and is automatically updated whenever one or more buildings are selected within the model. Figure 8 below shows an example of the Reports tab displaying metrics.

Reports

Report	N	%	Sum	%	Avg	Min	Max	NaNs
FSI	1	0.00	980.08	0.00	980.08	980.08	980.08	0
GFA	1	0.00	53904.14	0.00	53904.14	53904.14	53904.14	0

Figure 8: Example of the Reports tab in CityEngine

This rule only applies to buildings that were generated within CityEngine (i.e. those based on building footprint data or any buildings generated from procedural rules). We attempted to modify the rule so that the GFA and FSI would be calculated for those buildings that were imported from the Sketchup model or COLLADA files but were unsuccessful.

3.5.1.2 Proposed

The site statistics for the Proposed scenario were primarily derived from development applications that were provided by the City. This was done because development applications were the only available sources of information containing site statistics for the proposed developments. We populated the site statistics for the proposed buildings based on what was available within the applications. In certain cases not all information was provided, which resulted in some gaps. The information we attempted to populate includes:

- Address;
- Applicant;
- Owner;
- File Number;
- Height (m);
- Height in Storeys;
- Residential GFA;
- Non-Residential GFA;
- Units; and,
- Zoning.

We created a “pseudo” parcel layer to store and manage the information for the proposed buildings. This was done for two reasons:

1. Similar to the site statistics for the existing conditions, attribute information is not easily assigned to features that were imported from existing 3D features because they do not have associated attribute information (as opposed to features based on GIS data, which do contain attribute information). Most of the proposed buildings are based on the Sketchup model or the COLLADA files provided, meaning they do not have attribute

information associated with them. While attributes can be added within CityEngine, the information has to be input manually which can be very time consuming.

2. The existing parcel data that we received from the City does not reflect any proposed buildings at this time. Therefore, the parcels are not yet subdivided for each individual building within a development. This means that an existing parcel may currently "contain" multiple proposed buildings, making it difficult to provide site statistics on an individual building basis using the existing parcel fabric.

The "pseudo" parcel layer was created to solve the issues mentioned above. Since the attributes could not easily be added to the models of the proposed buildings, the information is stored in the "pseudo" parcel layer so it can be viewed within the model and managed more easily by City staff. This data could also be easily copied to a different dataset in the future if the City would like to manage this information elsewhere.

The "pseudo" parcel layer was created from the existing parcel fabric, and then subdivided based on the building location. This way the data could be separated for each individual building within a proposed development rather than having one parcel for the entire development. However, the parcels were subdivided for storage and information management purposes only and should not be interpreted as existing or proposed property lines.

As mentioned in section 3.5.1.1, a CGA rule was created to automatically calculate the GFA and FSI for buildings generated within CityEngine. Since the rule was not able to calculate GFA and FSI for buildings imported from existing 3D features, it is unable to work on any of the proposed buildings that were imported from the Sketchup or COLLADA files. However, the rule does work for the buildings based on the Demonstration Plan building footprint data in the Markham Road – Mount Joy model.

3.6 CGA Rules

There were two primary CGA rules that were created for the model. The first is the "Footprint" rule which extrudes building footprints based on the height attribute within the building footprint data and creates line delineations for each floor in a building. In addition, the Footprint rule also calculates the GFA and FSI for any building models that are selected (as mentioned above, this calculation only applies to models created within CityEngine and does not apply to any buildings that were imported from existing 3D features). The CGA code for the "Footprint" rule is shown in Figure 9 below:

```
version "2022.1"

attr BUILDINGHE = 1
attr ParcelArea = 1

@StartRule
Lot --> extrude(BUILDINGHE)
comp(f) { side : Facade | top   : Roof |
bottom : BaseInfo}

Facade -->
   split(y) { ~3   : Floor
            | ~0.5 : Ledge }*

Ledge -->
color("#7A7A7A")

BaseInfo -->
report("GFA",geometry.area*BUILDINGHE/3)
report("FSI",geometry.area/ParcelArea)
```

Figure 9: Example of the "Footprint" CGA rule

The second rule that was created for the model is the RoadAttributes rule. This rule changes the texture of a roadway based on the road class specified within the road data. The preferred textures can be modified within this rule by changing the hex code identified in the "streetcolour" constant, as outlined in red in Figure 10 below:

```
version "2022.1"

attr CLASS = ""


const streetcolor =
    case CLASS =="Local Road" || CLASS =="Laneway" || CLASS =="Private Road" : "#bfbfbf"
    case CLASS =="Major Arterial" || CLASS =="Major Collector" || CLASS =="Minor Collector" : "#737373"

    else : "#4d4d4d"

@StartRule
Street --> color(streetcolor)
Sidewalk --> color("#b3b3b3")
Joint --> color(streetcolor)
```

Figure 10: Example of the "RoadAttributes" CGA rule

3.7 Exporting from CityEngine

Once the model was completed, we experimented with exporting the models to various formats to confirm CityEngine's overall compatibility with other pieces of software. CityEngine's export function can be accessed by selecting File, then Export Models as shown in Figure 11 below.

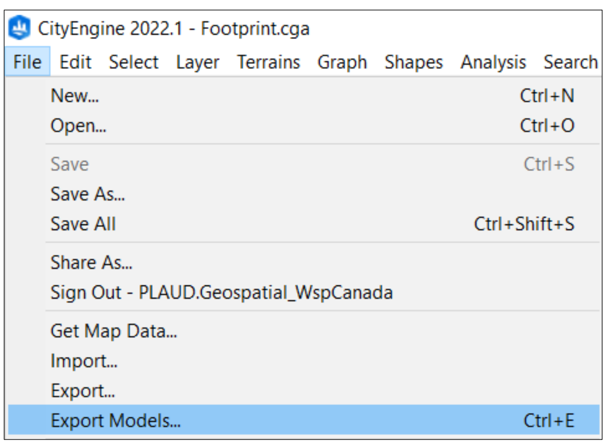


Figure 11: Navigating to the "Export Models" Function

CityEngine can export to various other file formats and packages natively, including (but not limited to):

- ESRI File Geodatabase;
- ESRI Scene Layer Package;
- COLLADA (.DAE);
- Autodesk FBX;

- Autodesk DWG;
- KML; and,
- CityEngine WebScene

In our experience, the export function generally worked well, with some limitations or discrepancies identified when exporting to certain file types. The following sections go into further detail on the results of exporting to each file type identified above.

3.7.1 ESRI File Geodatabase

When exporting to File Geodatabase, models can either be exported as polygon or multipatch features. Exporting to multipatch will allow the model to still be viewed in 3D within ArcGIS Pro or ArcScene, while exporting to polygon will result in just a 2D footprint of the building model.

When exporting to File Geodatabase, we found that some additional processing and cleanup may be required depending on the models that are being exported. The building models we tested resulted in polygons that had unnecessary features and linework that required additional cleanup within ArcGIS to simplify the geometry, as shown in Figure 12 below:

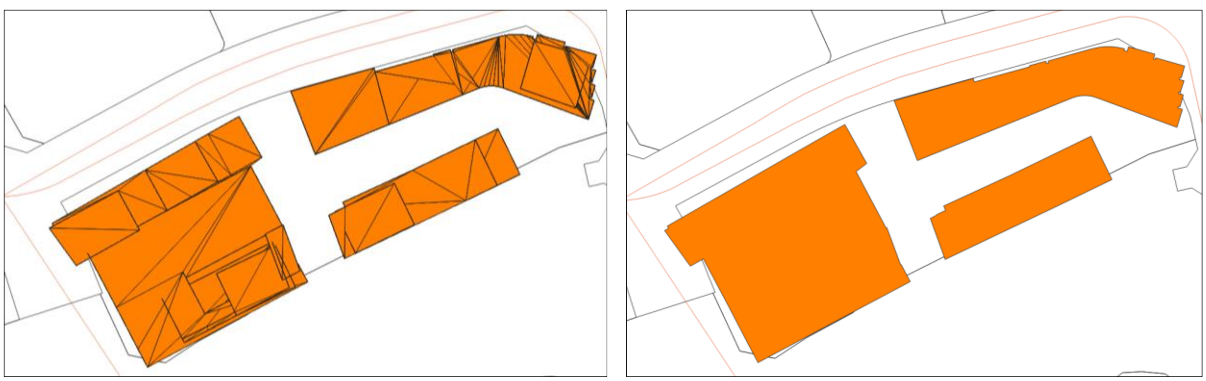


Figure 12: FGB Export - Exported Footprints Before and After Cleanup

However, this can be an effective way to create building footprint data from a CityEngine model, if required.

3.7.2 ESRI Scene Layer Package

An ESRI Scene Layer Package can be used to store and display 3D models in ArcGIS Online or ArcGIS Urban. Models can be selected and exported from CityEngine to an ESRI Scene Layer Package (.slpk) and uploaded to ArcGIS Online.

Based on our testing, exporting to ESRI Scene Layer Package from CityEngine works well and is the primary way of uploading models to ArcGIS Online. There is the option to connect directly to an ArcGIS Online account within CityEngine and upload Scene Layer Packages from within the software as well.

3.7.3 COLLADA

Exporting to COLLADA (.dae) from CityEngine was also very straight forward. The user can select any models they wish and select the COLLADA option from the Export window. We tested the results by importing the COLLADA file into Sketchup and found that the models displayed correctly.

3.7.4 Autodesk DWG

CityEngine does have the ability to export to DWG natively. However, when exporting to DWG only the wireframe of the models is exported. During our testing we were able to open the DWG in AutoCAD and view the wireframe of the models within the software. The wireframe can be viewed in 2D or in 3D when AutoCAD's 3D view is enabled.

3.7.5 KML

Models can be exported to KML from CityEngine for use in Google Earth or other software programs. When undertaking our testing, we imported the resulting KML file into Google Earth and found that some of the models did not fully display properly. This mainly applied to buildings that had varying heights or higher levels of detail, where certain sections of the building would not display. An example of this is shown in Figure 13 below:

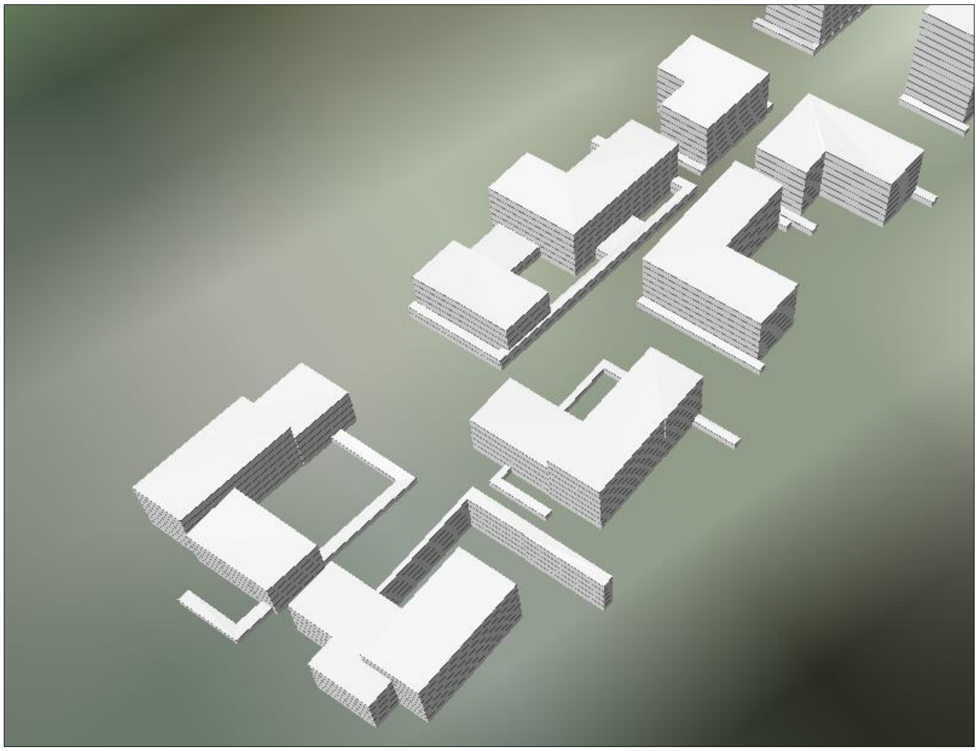


Figure 13: Results from Exported to KML from CityEngine

Due to these results, there may be better options to display the models on an online platform, such as through ArcGIS Online, ArcGIS Urban or a CityEngine WebScene.

3.7.6 CityEngine WebScene

CityEngine also allows the user to export models to a CityEngine WebScene (.3ws). A CityEngine WebScene is a format which allows users to upload models to a CityEngine Web Viewer, which will display the model information and allow other users to view and interact with the model components within a web browser. The Web Viewer also has some basic tools included, such as a search function, commenting, sunlight adjustment and the ability to turn layers on and off.

Based on our experience, this option works well and is a great option to quickly create web scenes that can be shared with others. In addition, any bookmarks created in CityEngine will also be maintained within the web scene and can be used to create basic model flythroughs. It should be noted however, that the CityEngine Webscene (.3ws) format has been deprecated by ESRI and is being replaced by Scene Layer Packages. This means this format will be removed from CityEngine in the future.

4.0 ArcGIS Urban Experiment

As part of the project scope, the City requested that WSP explore ESRI's ArcGIS Urban software to better understand its capabilities, use cases, workflow options and integration with CityEngine. To do this, we created a test model that was used to explore the functionality of ArcGIS Urban and its relationship with CityEngine.

The following section provides an overview of ArcGIS Urban and its capabilities, along with insight on software setup, integration with CityEngine, as well as findings and limitations.

4.1 Overview

ArcGIS Urban is a browser-based 3D modelling program offered by ESRI that is focused on land use planning applications and is geared towards planners as users. It includes several core elements of CityEngine that have been simplified for ease of use, including:

- Creation of development scenarios;
- Input and creation of site statistics;
- Some (but limited) data import capabilities;
- Land use designation and zoning boundaries;
- Procedural land use or zoning rules (can be created directly in ArcGIS Urban);
- Ability to generate conceptual buildings based on land use or zoning rules;
- Manual editing of buildings; and,
- Analysis tools including shadow analysis, view shed analysis and measuring distances.

ArcGIS Urban is administered through ESRI's ArcGIS Online platform, meaning an ArcGIS Online organizational account is required to use the software. This also allows for models to be shared via ArcGIS Online and the ability to embed models directly into webpages. ArcGIS Urban has additional public engagement features including commenting, model interaction and the option for members of the public to generate their own models.

ArcGIS Urban is more accessible than CityEngine due to its user-friendly interface and browser-based approach. This allows staff that may be less technical or those without specialized training to also make use of the software.

ArcGIS Urban models can cover a large geographic area and are generally intended to provide a comprehensive overview of the various projects and initiatives occurring within a municipality.

4.2 ArcGIS Urban Concepts

ArcGIS Urban has a number of key concepts that should be understood to make best use of the software. ArcGIS Urban models are generally organized into two categories: Plans and Projects.

Plans represent longer-term City-led planning initiatives, while Projects are used to track shorter term developments projects from initiation to construction.

In addition to Plans and Projects, ArcGIS Urban offers tools to support analysis and decision-making using Indicators and Dashboards. The following sections describe these concepts in further detail and provide additional insight and considerations based on our experience working with ArcGIS Urban.

4.2.1 Plans

Plans represent longer-term City-led planning initiatives such as Secondary Plans, block plans or other planning studies. Plans are intended to cover a defined study area and can incorporate land use or zoning regulations to generate conceptual buildings (similar to CityEngine's procedural rules). Plans can be used to conceptualize the build out of the entire plan area based on the defined parameters (i.e. land use restrictions, zoning, etc). Figure 14 below shows an example of a Plan in ArcGIS Urban.

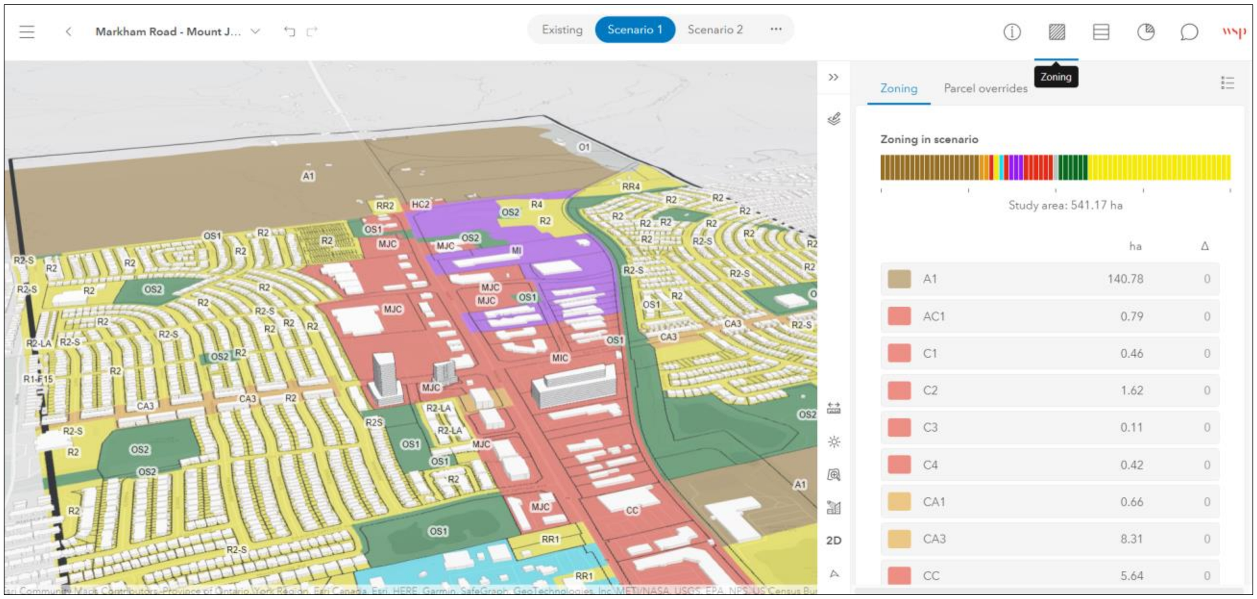


Figure 14: Example of a Plan in ArcGIS Urban

Users can create two types of Plans within ArcGIS Urban: Land Use Plans and Zoning Plans.

Land Use Plans are intended to represent higher-level planning initiatives such as Official Plan or Secondary Plans. Land use designations and regulations such as maximum height, maximum density or coverage area can be defined within the software and can then be used to generate building concepts. Permitted uses can also be defined for specific land use designations, allowing for land use plans or policies to be better represented within the model. Metrics can also be configured and automatically calculated for entire Land Use Plans, allowing users to track and monitor planning targets directly in ArcGIS Urban.

Zoning Plans represent more detailed planning initiatives or concepts. Zones and zoning regulations can be defined within ArcGIS Urban, allowing for more detailed concepts to be developed using Zoning Plans. In addition to defining max heights, densities or lot coverage as in Land Use Plans, Zoning Plans also allow users to define zoning regulations such as setbacks and stepbacks for each zone. This allows users to quickly generate building concepts based on real-world zoning regulations. Figure 15 below provides an example of generating buildings or developments concepts based on zoning regulations in ArcGIS Urban.

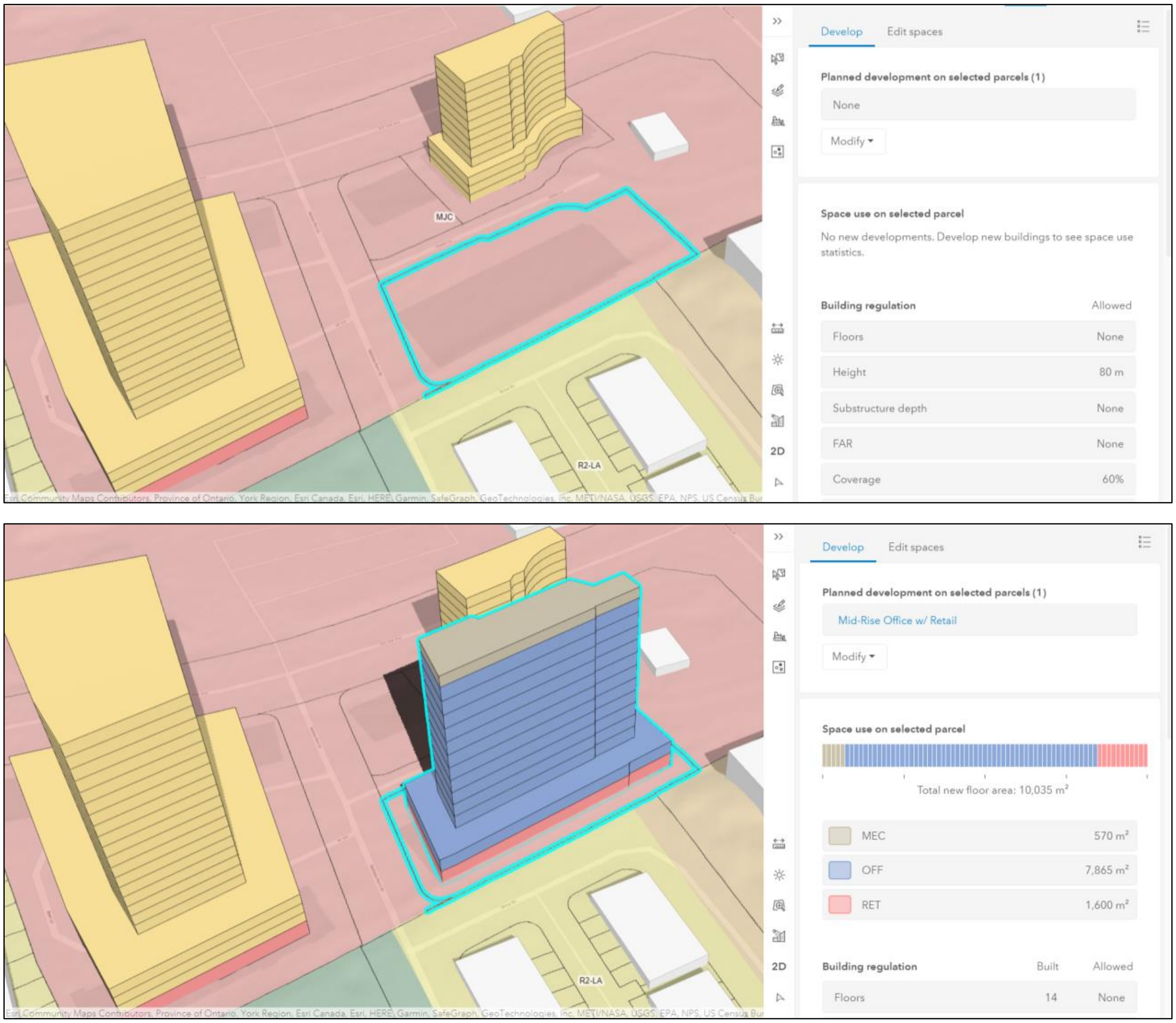


Figure 15: Generating Development Concepts in ArcGIS Urban

Similar to Land Use Plans, permitted uses can be defined for each zone. Permitted uses within individual mixed-use buildings can also be defined. Metrics can also be setup and calculated for Zoning Plans, allowing for site statistics and plan targets to be generated for entire Zoning Plans and displayed in the Plan's dashboard.

The primary difference between Land Use Plans and Zoning Plans is the level of detail that can be represented. Building setbacks or stepbacks can be defined within a Zoning Plan but cannot be defined in a Land Use Plan. Therefore, if a user wishes to create building concepts with more detailed massing, Zoning Plans would be the preferred option.

Both Land Use Plans and Zoning Plans can have multiple development scenarios, similar to CityEngine. This allows users to view and compare between what is currently existing and what is proposed.

In addition to generating conceptual buildings from defined land use or zoning regulations, existing 3D features can be imported into ArcGIS Urban. This is done by uploading the 3D features as an ESRI Scene Layer Package to ArcGIS Online and importing them into ArcGIS Urban (this can also be done directly from CityEngine). However, the 3D features that are added to a Plan as "Scenario design context." This means that the 3D features can be viewed alongside the conceptual buildings a user has created within a Plan but the land use or zoning parameters that have been defined will not apply to the imported 3D features. Therefore, the imported features act as a visual reference but do not react to the "rules" created in ArcGIS Urban.

This is a current limitation of the software as building concepts must be created in ArcGIS Urban directly to make most of the software's capabilities. This means that any models created outside of ArcGIS Urban (i.e. those created in CityEngine or other programs) must be recreated in ArcGIS Urban to ensure that the zoning regulations and other parameters will be applied.

Within the test model, we created both Land Use Plans and Zoning Plans for the each of the Secondary Plan areas to better understand the differences and functionality of both types of Plans. While Land Use Plans can be useful in situations where higher-level planning initiatives with fewer design regulations need to be represented (i.e. an Official Plan), Zoning Plans offer a finer level of control over building concepts due to the inclusion of setbacks and stepbacks. Based on our experience, if the City wishes to incorporate this finer level of control into their Urban models, it is recommended that Zoning Plans be used to model the Secondary Plan areas.

4.2.2 Projects

Projects are intended to represent shorter term development projects or applications. Projects are managed on an individual basis (i.e. a project can be created for each development application currently active within the City) and can include application information or statistics such as the development name, proposed zoning, number of units or gross floor area. These development metrics are manually added to a Project's dashboard by the user, unlike with Plans where metrics are automatically calculated based on the development concept. This results in some additional manual effort when creating Projects.

Projects can be used to track development status and updates throughout a development's life cycle. The user can create a custom development "Status Indicator" which can be used to track a Project's progress over time.

Similar to Plans, Projects can have multiple development scenarios, allowing users to compare existing conditions to the proposed development project or alternate designs. Existing 3D features can be uploaded to ArcGIS Urban and displayed within a Project scenario. The approach is similar to uploading 3D features to a Plan (see section 4.2.1), where the 3D feature is uploaded to ArcGIS Online and imported into ArcGIS Urban. The imported features are added as "Scenario design context" to a Project and provide a visual reference of the development project. Figure 16 below provides an example of Project in ArcGIS Urban.

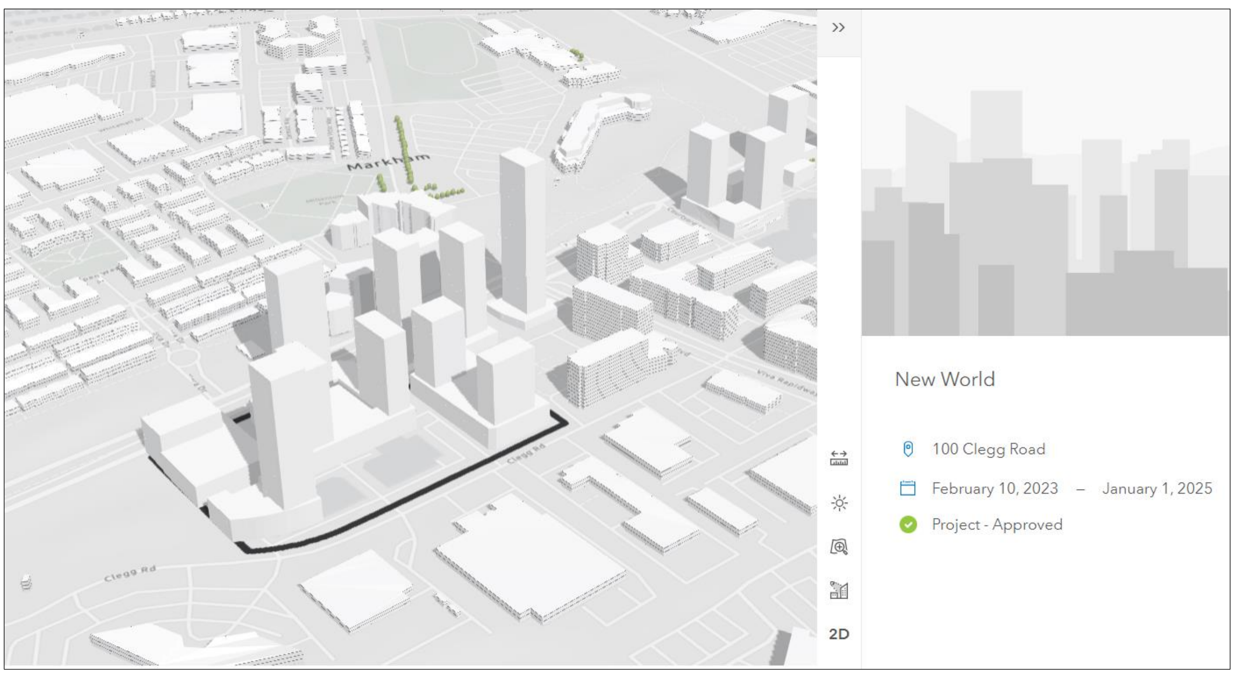


Figure 16: Viewing a Project in ArcGIS Urban

The main purpose of Projects is to provide a visual reference of a development within the surrounding context, allow for comparison between scenarios and to assist with managing project status over time. They are not intended to be used for creating concepts and do not make use of defined land use or zoning regulations. Plans should be used when the user wishes to create development concepts based on established zoning regulations.

Projects can be batch uploaded to ArcGIS Urban based on either GIS features layers imported from ArcGIS Online or using an Excel spreadsheet template. After uploading the project information based on these data sources, building footprints matching the project information can also be uploaded and displayed in ArcGIS Urban as “Future Buildings.”

When testing this feature, we found the Future Buildings would not display in ArcGIS Urban, despite the building footprints being uploaded successfully. In addition, uploading Projects in batches using this method will lock the individual Projects by default, meaning then cannot be manually edited from the main map interface once uploaded. Depending on the City’s preference, this may not be a desirable method for establishing Projects in ArcGIS Urban.

In the test model, we created Projects for each of the “proposed” buildings identified by the City for use in the CityEngine model. Since these buildings were all part of active development proposals that had been submitted to the City, they were logical candidates to be managed as Projects within the test model.

A separate Project was created for each development, and development information was populated based on the development applications provided. In addition, the models for each of these developments that were created in CityEngine were uploaded to ArcGIS Urban and assigned as “Scenario design context” to the Existing and Proposed scenarios for each Project. This was configured for each Project included in the test model.

While this did require a fair amount of manual effort, we found that Projects do offer a convenient way of managing and tracking active development projects all in one platform, while also providing a visual reference that can be compared to both existing and proposed development.

While the initial setup of the Projects may be somewhat time-consuming depending on the number of Projects that need to be entered, Projects only need to be updated periodically once the initial setup is complete. This means that on-going maintenance of the Projects is not likely to be as labour intensive as the initial setup.

4.2.3 Indicators

Indicators represent spatial datasets or other information that can be overlayed on top of ArcGIS Urban's overview map for analysis purposes. Items such as demographic information, floodplain areas or other datasets can be added to the model to assist with decision making. Custom Indicators can be created from scene layers or web scenes imported via ArcGIS Online. Dashboards created in ArcGIS Online can also be incorporated into custom Indicators.

The City may wish to identify potential indicators of interest and associated supporting datasets to enable a detailed exploration of Indicators to be undertaken in future.

4.2.4 Dashboards

Dashboards display metric values and statistics for a Plan or Project. Dashboards are handled differently for Plans and Projects within ArcGIS Urban. The following subsections provide more detail on how dashboards and metrics are handled for Plans and Projects.

4.2.4.1 Plans

When used in Plans, dashboards display and automatically calculate metrics based on the development concept within the current scenario. The metrics that are calculated for a Plan are defined within a Plan's configuration settings. ArcGIS Urban uses a graphical interface to define the metrics which will be displayed within a plan, as seen in Figure 17 below:

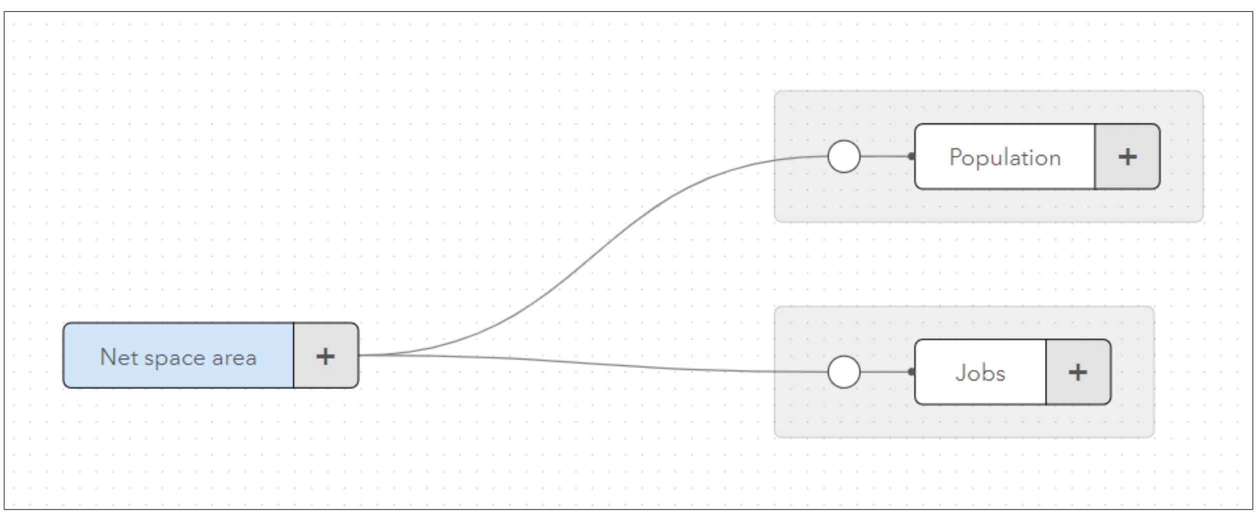


Figure 17: Metrics Interface for Plans in ArcGIS Urban

All metrics within a Plan are based on the "Net space area" as a starting point. The Net space area represents the total area for all buildings (or all currently selected buildings) within a Plan. From the Net space area, the user can define other metrics that can be generated based on a per m^2 area calculation. In addition, metrics can be calculated for specific uses (i.e. based on residential area, commercial area, etc.)

In the test model, we created metrics to estimate the population and number of jobs within the Plan. This is done by adding a metric to the interface and defining a formula, as shown in Figure 18 below:

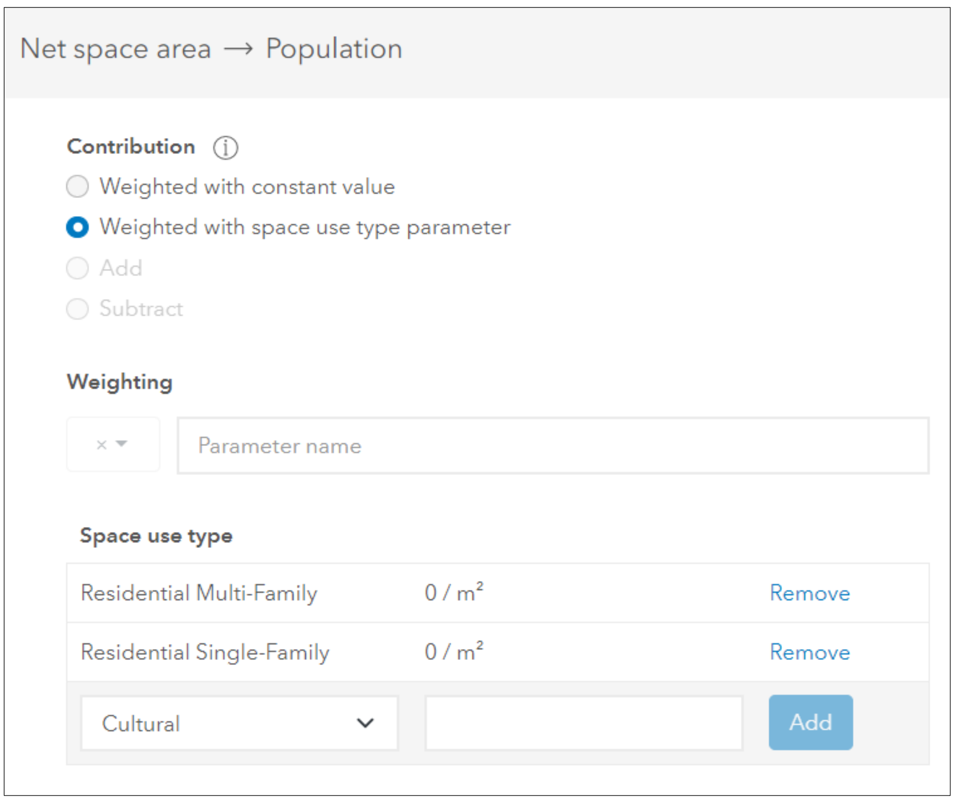


Figure 18: Calculating Metrics for Plans in ArcGIS Urban

For the population metric example, we arbitrarily defined a density assumption of 0.002 people/m^2 for Residential Single-Family uses and 0.0025 people/m^2 for Residential Multi-Family uses to generate a fictional population estimate. The Population metric within our Plan would then calculate a population estimate for any buildings that contained a Residential Single-Family or Residential Multi-Family use.

Similarly, we arbitrarily defined a job density assumption of 0.0007 jobs/m^2 for all commercial and industrial uses to generate a fictional job estimate for the Plan area. New metrics can then be generated based on previous metrics (i.e. generating a metric on required number of parking spots based on the population and jobs), allowing for wide variety of metrics to be generated within a Plan.

It should be noted that metrics are self-contained within a Plan and cannot be linked to other Plans (i.e. the metrics in a Zoning Plan cannot be fed into the dashboard of a larger Land Use Plan). This means that if the City chooses to use ArcGIS Urban in the future, it may be best to create Zoning Plans rather than Land Use Plans to represent both the Markham Centre and Markham Road – Mount Joy Secondary Plan areas. This would allow the City to generate metrics and monitor targets for the entire Plan area, while also being able to utilize the setback and stepback parameters within the Zoning Plans (as setbacks and stepbacks cannot be defined for Land Use Plans).

4.2.4.2 Projects

When used within Projects, dashboards are also used to display information and metrics related to the Project. However, information and metrics are manually entered into the dashboard by the user when working with Projects, rather than metrics being automatically calculated as they are with Plans.

Custom items can be added to the dashboard, allowing the user to include virtually any metrics related to the development. Since the information is manually entered into a text or number field, the information will remain static unless updated by the user as the Project progresses. Dashboards can also be configured for each scenario, allowing different information to be displayed for each scenario.

It should be noted that the dashboard for each Project in a model needs to be configured manually. While this approach requires more manual effort, it does allow the user to track information that is specific to the development and assists with overall organization and management of a Project.

4.3 ArcGIS Urban Setup

As part of the ArcGIS Urban experiment, we created a test model to explore the features and capabilities of the software, as well as its interaction with CityEngine. To use ArcGIS Urban to its full capability, a number of data inputs and settings must first be configured. Without providing these inputs, many of ArcGIS Urban's features will not be available or will not operate as intended.

The following section provides an overview of these inputs, along with the setup approach we used in the test model.

4.3.1 Initial Setup & General Settings

When creating a new Urban model, you will first be asked to import parcel and zoning data. These datasets can be imported into ArcGIS Urban from ArcGIS Online. Once these datasets are added successfully, a new Urban model will be created. The user can also skip this step and add the parcel and zoning data at a later time if desired.

Once the Urban model is created, the user can begin configuring and populating the model with information. The general settings for an Urban model can be accessed using the "Manage" button located in the top right corner of the main screen, as shown in Figure 19 below.

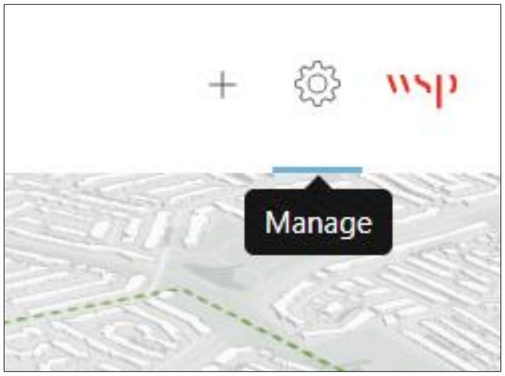


Figure 19: Navigating to the "Manage" Button in ArcGIS Urban

After entering the “Manage” menu, a number of tabs are located along the top of the screen (see Figure 20 below). These tabs provide access to the various settings and options for ArcGIS Urban’s key features.

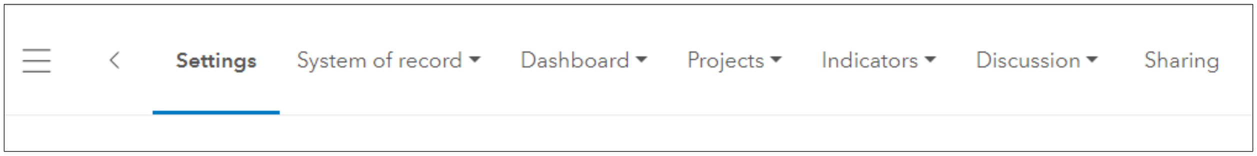


Figure 20: Available Tabs within the Manage Menu

The General settings menu (shown in Figure 21 below) is where the user can populate basic information about the model, such as the City name and City logo, and also define items such as the base map to be displayed in the overview map and the existing building dataset.

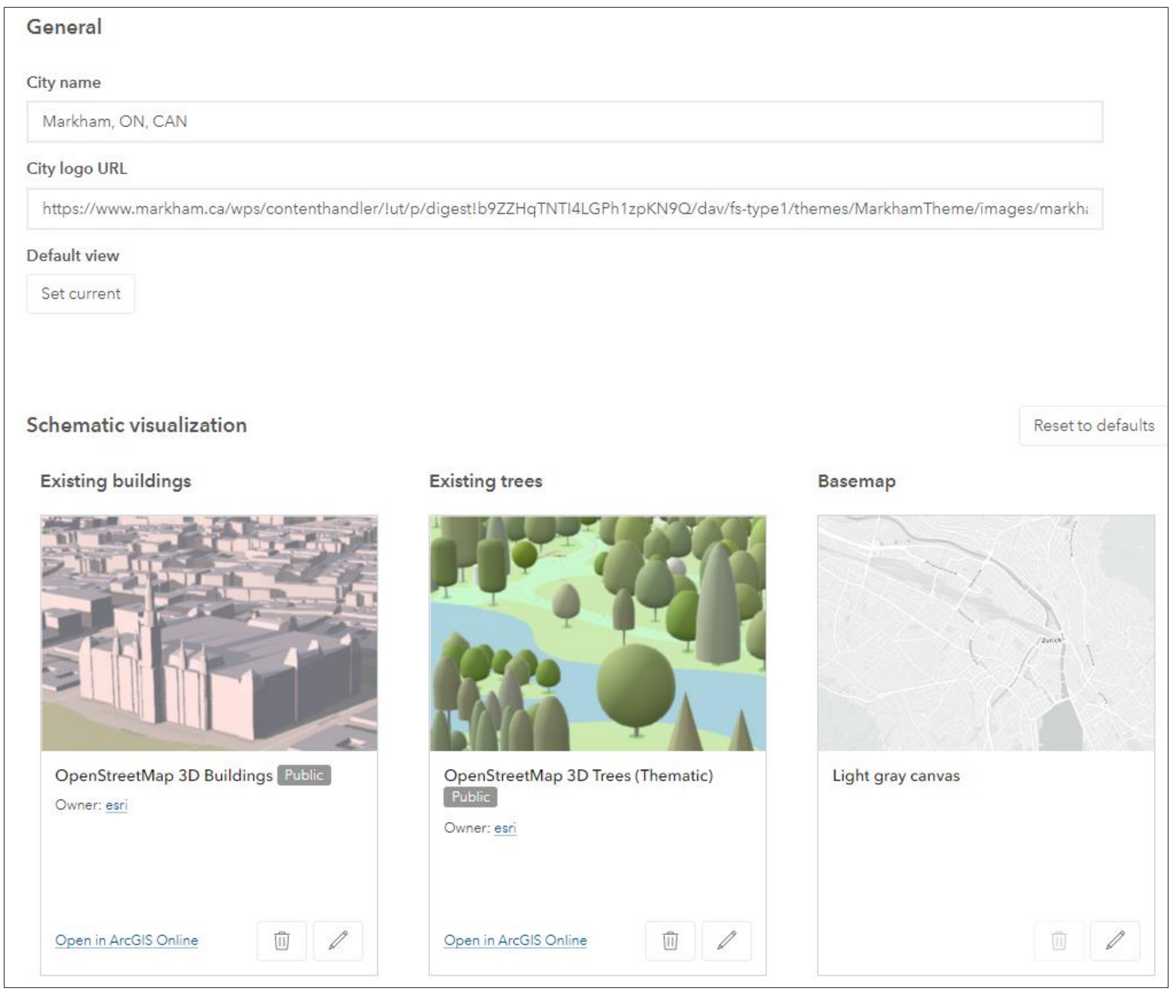


Figure 21: General Settings Menu in ArcGIS Urban

4.3.2 System of Record

ArcGIS Urban requires a number of data inputs and settings to be established in order for make best use of all of the software's features. These inputs are referred to as the "System of Record" within an Urban model.

The System of Record includes all base data that the model and its functions are be based upon. These datasets include parcels, zoning (Zoning types and Zoning boundaries), land use (Land Use types and Land Use boundaries), Space Use types and Building types.

4.3.2.1 Parcels

Parcel data can be added to an Urban model by selecting the dropdown on the System of Record tab and selecting Parcels. Parcel data can be added from a feature layer hosted on ArcGIS Online. This dataset will act as the main parcel base for the entire Urban model.

Once added, users can then select and view parcel information from within the Urban model. Projects and Plans can also be created by selecting a parcel and selecting the "Create here" option. The parcel base is also used in conjunction with the established zoning rules to generate conceptual buildings within a Plan. Figure 22 shows an example of the information that is available when selecting a parcel in ArcGIS Urban.

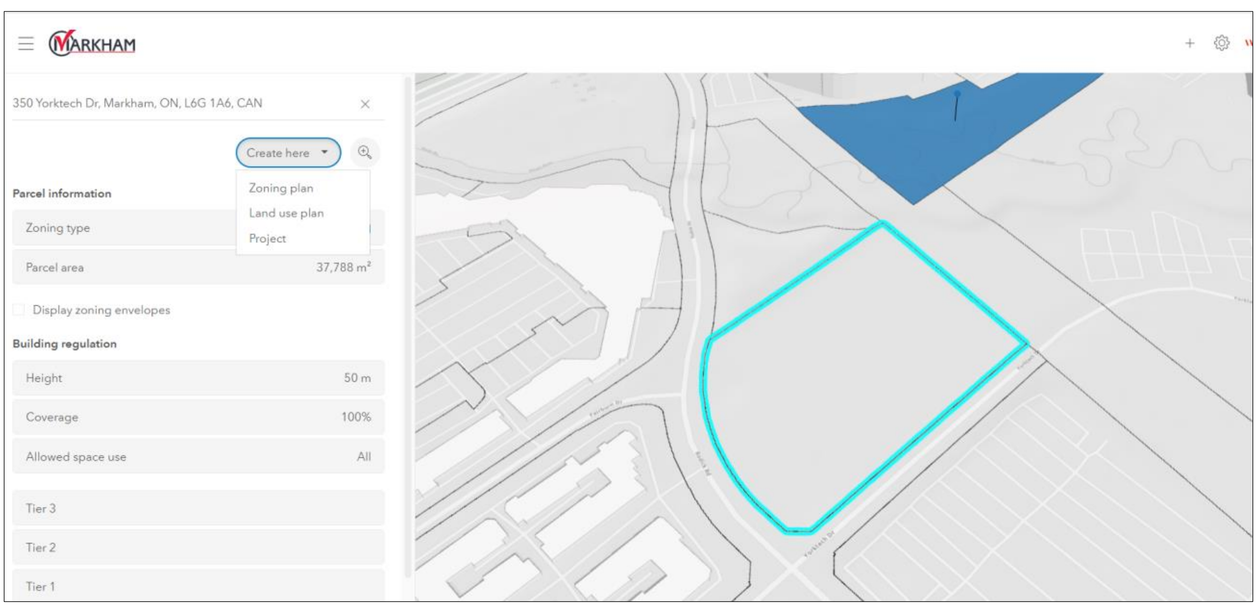


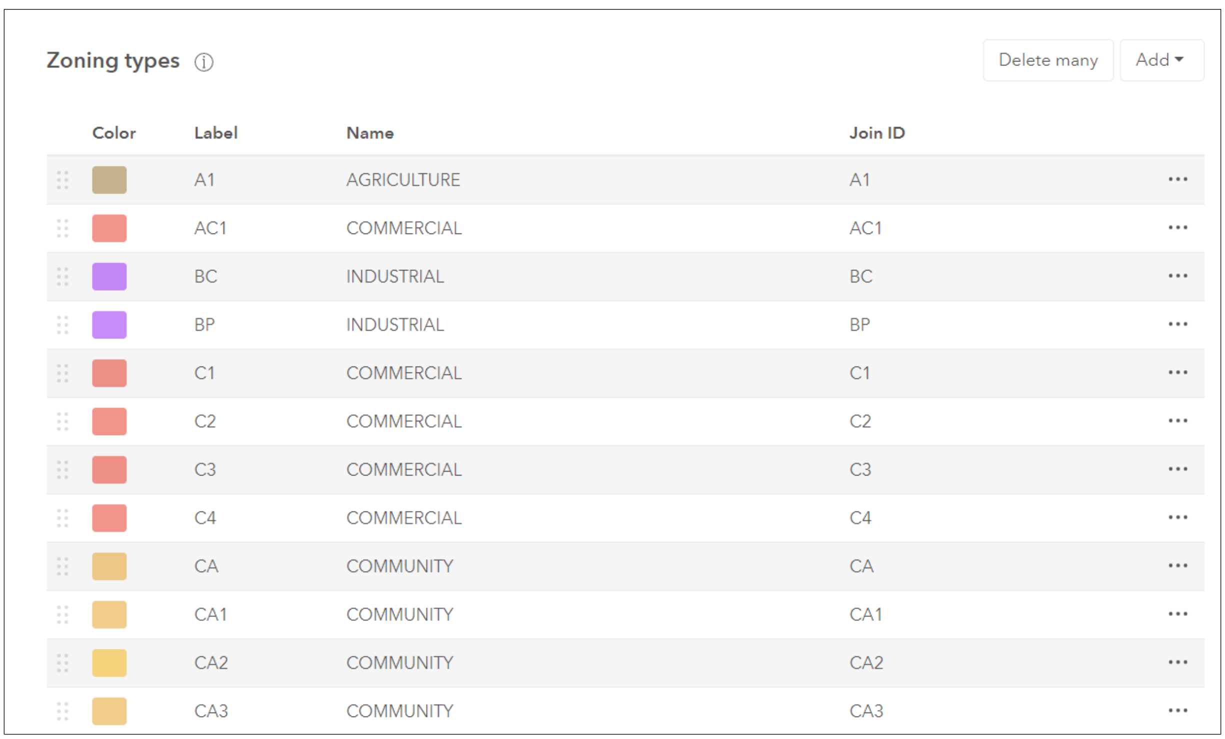
Figure 22: Viewing Parcel Information in ArcGIS Urban

While ArcGIS Urban is intended to allow users to create models for an entire City, we found that adding parcel data covering the entire City of Markham impacted the overall performance of the model quite significantly. To help improve performance, we only included the parcel base for the Secondary Plan areas within our test model.

4.3.2.2 Zoning Data

Zoning data is separated into two categories: Zoning types and Zoning boundaries. Zoning types include all the various zone categories that are included in a municipality's zoning by-law. Zoning boundaries are the geometric shapes representing the coverage area of each zone category. Both of these datasets need to be added and configured in ArcGIS Urban for any zoning related features to work properly. Zoning types need to be added first, followed by the zoning boundaries.

Zoning types can be added to ArcGIS Urban from an Excel spreadsheet, a feature table from an existing GIS dataset or from the symbology defined for a feature layer hosted in ArcGIS Online. Once the zoning types are added successfully, all zone categories are shown within the Zoning types menu. An example of Zoning types is shown in Figure 23 below.



Zoning types (i) Delete many Add

Color	Label	Name	Join ID
	A1	AGRICULTURE	A1
	AC1	COMMERCIAL	AC1
	BC	INDUSTRIAL	BC
	BP	INDUSTRIAL	BP
	C1	COMMERCIAL	C1
	C2	COMMERCIAL	C2
	C3	COMMERCIAL	C3
	C4	COMMERCIAL	C4
	CA	COMMUNITY	CA
	CA1	COMMUNITY	CA1
	CA2	COMMUNITY	CA2
	CA3	COMMUNITY	CA3

Figure 23: Example of Zoning Types in ArcGIS Urban

For the test model, we derived the zone categories from the Zoning dataset provided by the City. This was done by separating the parent zone from the other suffixes in the zoning code (i.e. site-specific exception numbers, etc.) and uploading the resulting zoning information in Excel format to ArcGIS Urban. ArcGIS Urban provides an Excel template that can be used to organize the zoning information in a structure that is readable by ArcGIS Urban. We then applied colour symbologies based on the type of zone (i.e. residential, commercial, etc.).

Each Zoning type can then be selected and modified within the Zone type menu. This is where specific zoning parameters such as maximum height, number of floors, coverage area and setbacks can be entered for each zone category. In addition, permitted uses (called Space Use types in ArcGIS Urban) can also be defined for each zone category. For the test model, generic zoning parameters were used for each Zoning type (i.e. the City's existing zoning regulations were not incorporated into the model). An example of the available zoning parameters is shown in Figure 24.

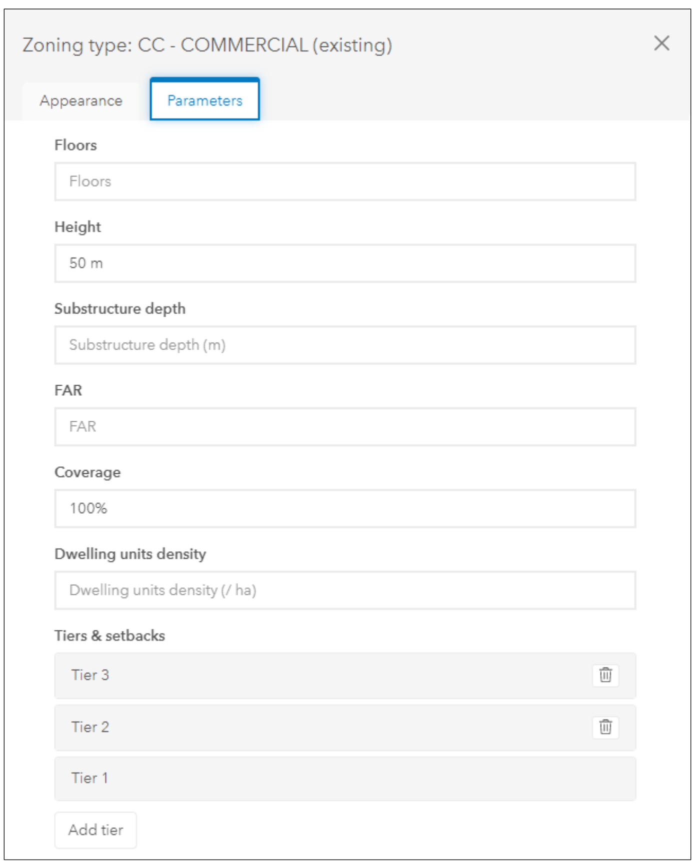


Figure 24: Accessing Zoning Parameters in ArcGIS Urban

The defined zoning parameters can also be overridden on a site-by-site basis within a Plan. This allows for users to visualize proposed development, test possible zoning scenarios, or represent site-specific exceptions within the model.

Once the Zoning types have been established, the Zoning boundaries can then be uploaded to ArcGIS Urban. Zoning boundaries can be imported from feature layers hosted in ArcGIS Online. All the Zone types within the Zoning boundary feature must match the Zoning types defined in the System of record. Once the Zoning boundaries are added successfully, the user should be able to view zoning information within the model when selecting a parcel or when viewing a Plan. The Zone boundaries will display on the map when a Plan is selected, as shown in Figure 25 below.

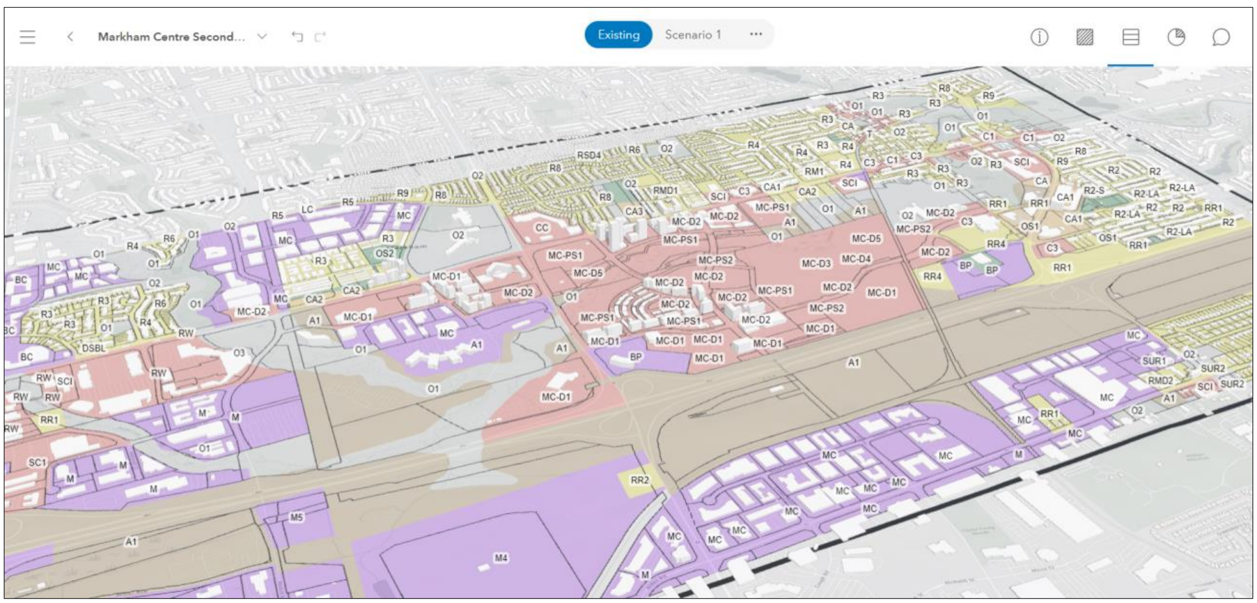


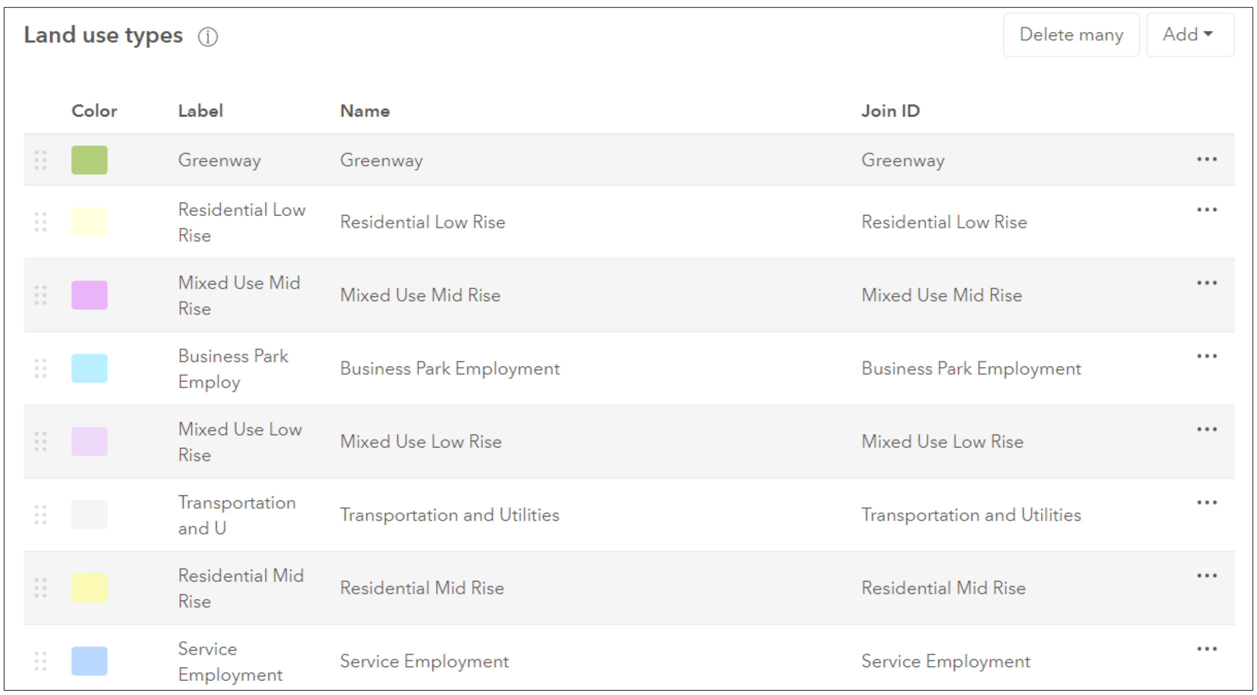
Figure 25: Zoning Boundaries shown in a Zoning Plan

4.3.2.3 Land Use Data

Land Use data in ArcGIS Urban is handled in a similar way to Zoning data. The user must first define the Land Use types within the System of Record, and then Land Use boundaries can be uploaded. Land Use types represent land use designations from a municipality's land use plan (i.e. Official Plan, Secondary Plan, etc.), while Land Use boundaries are the geographic coverage area for those land use designations.

Land Use types can be uploaded via Excel spreadsheet, feature table from an existing GIS dataset or from the symbology defined for a feature layer hosted in ArcGIS Online (similar to Zoning types). They can also be added individually directly within the Land Use types menu.

For the test model, we used the land use designations defined in the OP_2014_Map_3_Land_Use dataset provided by the City. We populated this information into the Excel template provided and uploaded the table to ArcGIS Urban via the Land Use types menu. We then selected layer symbologies for all of the land use designations. An example of Land Use types is shown in Figure 26.



Land use types ⓘ — Delete many — Add ▾

Color	Label	Name	Join ID
	Greenway	Greenway	Greenway
	Residential Low Rise	Residential Low Rise	Residential Low Rise
	Mixed Use Mid Rise	Mixed Use Mid Rise	Mixed Use Mid Rise
	Business Park Employ	Business Park Employment	Business Park Employment
	Mixed Use Low Rise	Mixed Use Low Rise	Mixed Use Low Rise
	Transportation and U	Transportation and Utilities	Transportation and Utilities
	Residential Mid Rise	Residential Mid Rise	Residential Mid Rise
	Service Employment	Service Employment	Service Employment

Figure 26: Example of Land Use Types in ArcGIS Urban

Once added, each Land Use type can be selected and further modified. Parameters such as maximum number of floors, maximum height, FSI and coverage area can be set within this menu. Similar to Zoning types, permitted uses (Space Use types) can also be defined for each of the Land Use types. For the test model, generic parameters were entered into the Land use types (i.e. the City's existing land use regulations were not entered into the model).

The defined land use parameters can also be overridden on a site-by-site basis within a Land Use Plan. An example of the available land use parameters is shown in Figure 27.

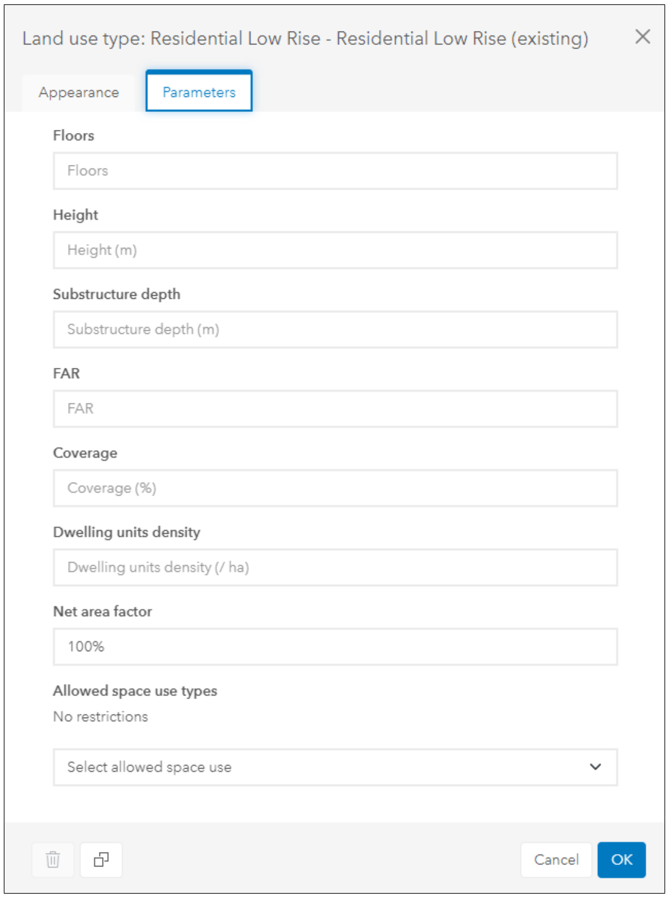


Figure 27: Accessing Land Use Parameters in ArcGIS Urban

Once all Land Use types are defined, the Land Use boundaries can be added to ArcGIS Urban. Land Use boundaries can be imported from a hosted feature layer in ArcGIS Online. For the test model, we used the boundaries included in the OP_2014_Map_3_Land_Use dataset provided by the City. Similar to Zoning data, the Land Use types (i.e. land use designations) within the Land Use boundary dataset must match the Land Use types defined in the System of Record in order for the boundaries to display.

Once both the Land Use types and Land Use boundaries are added to the System of Record, the land use information should display and set parameters should apply when viewing and developing concepts in a Land Use Plan, as shown in Figure 28.

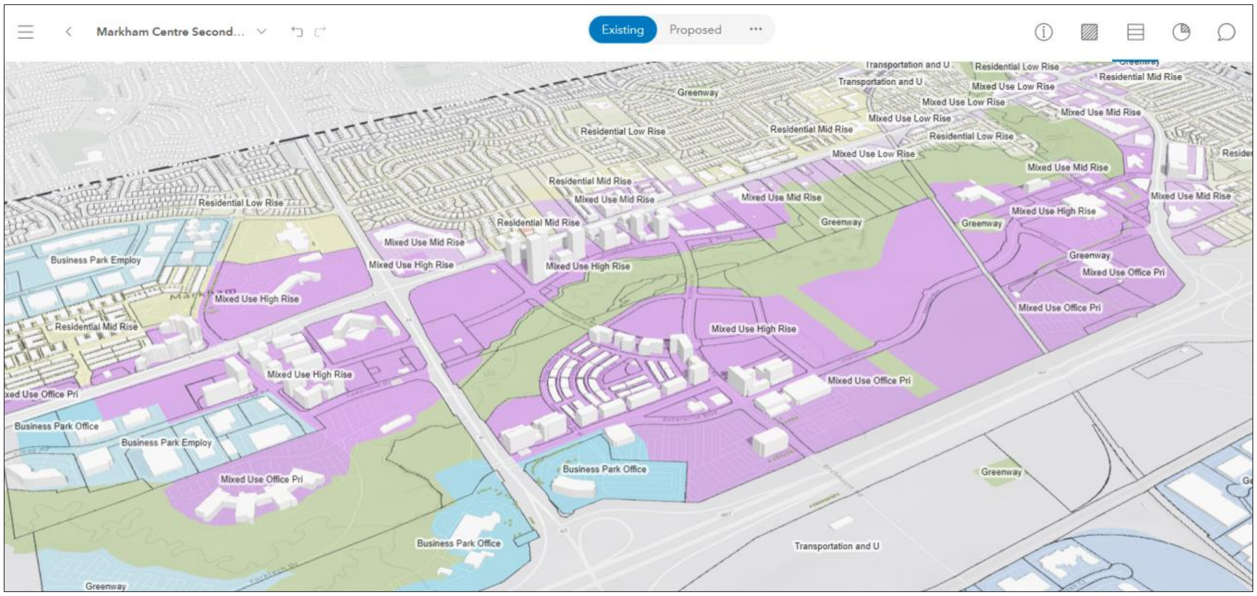


Figure 28: Land Use Boundaries shown in a Land Use Plan

4.3.2.4 Space Use Types

Space Use types are used to represent permitted uses within zoning by-laws or other land use plans (i.e. secondary plans). Space Use types can be assigned to specific zones or land uses within ArcGIS Urban. Space Use types can also be used to assign different uses within different parts of a building for mixed-use developments (i.e. retail on the first floor and residential on upper floors). Area percentages of each Space Use type can also be defined.

Space Use types can also be used to calculate metrics that are specific to certain uses. For example, residential Space Use types can be used to estimate population or commercial and employment Space Use types can be used to estimate the number of jobs within a Plan (see section 4.2.4.1).

For the test model, ArcGIS Urban's default Space Use types were used (i.e. no permitted uses from the City's land use plans or zoning by-law were entered into the model).

4.3.2.5 Building Types

Building types are templates which represent different building typologies within ArcGIS Urban. These building templates can be used to create conceptual buildings based on the defined zoning or land use parameters.

Building types are composed of one or more Space Use types which can be used to define the mix of uses within a building (i.e. retail in podium and a residential tower on top). In addition, the height range, massing and building dimensions can be defined for each Building Type. This allows users to quickly model buildings that fit a certain typology. However, the Building type will still respond to any zoning or land use parameters that are imposed upon it. An example of a Building type is shown in Figure 29.

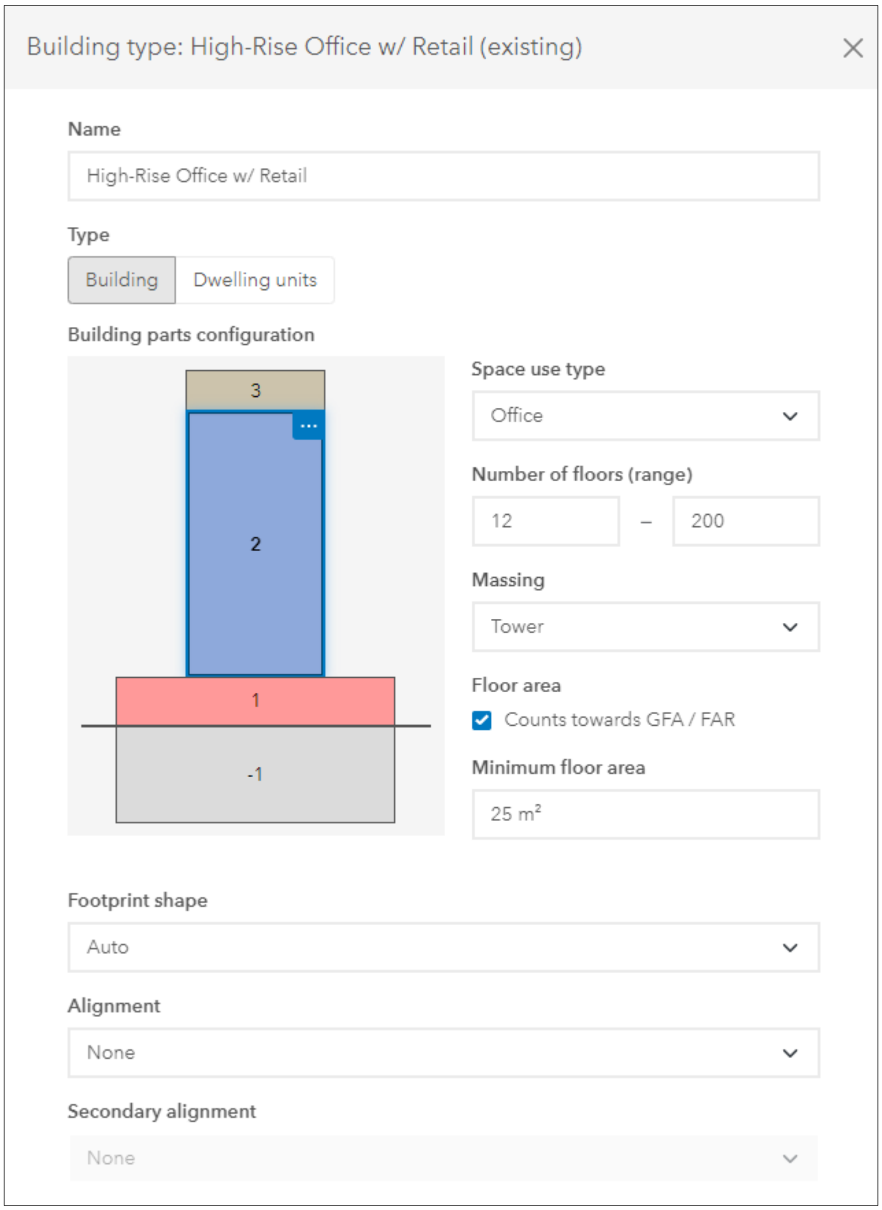


Figure 29: Example of a Building Type in ArcGIS Urban

Users can modify or create custom Building types to match existing definitions within their land use plans or zoning by-laws. Within the test model, only ArcGIS Urban's default Building types were used (i.e. we did not populate Building types based on the City's existing zoning by-law or land use plans).

4.4 Analysis Tools

ArcGIS Urban also has a number of basic tools available to assist users with analysis and decision making. Some of these key tools include Measure, Line of Sight, Profile and Shadow Cast Analysis.

4.4.1 Measure

The Measure tool allows users to measure both distance and area within the Urban model. This tool can be useful when determining distances between buildings, building heights, confirming setbacks, or checking building areas. An example of the Measure tool is shown in Figure 30.

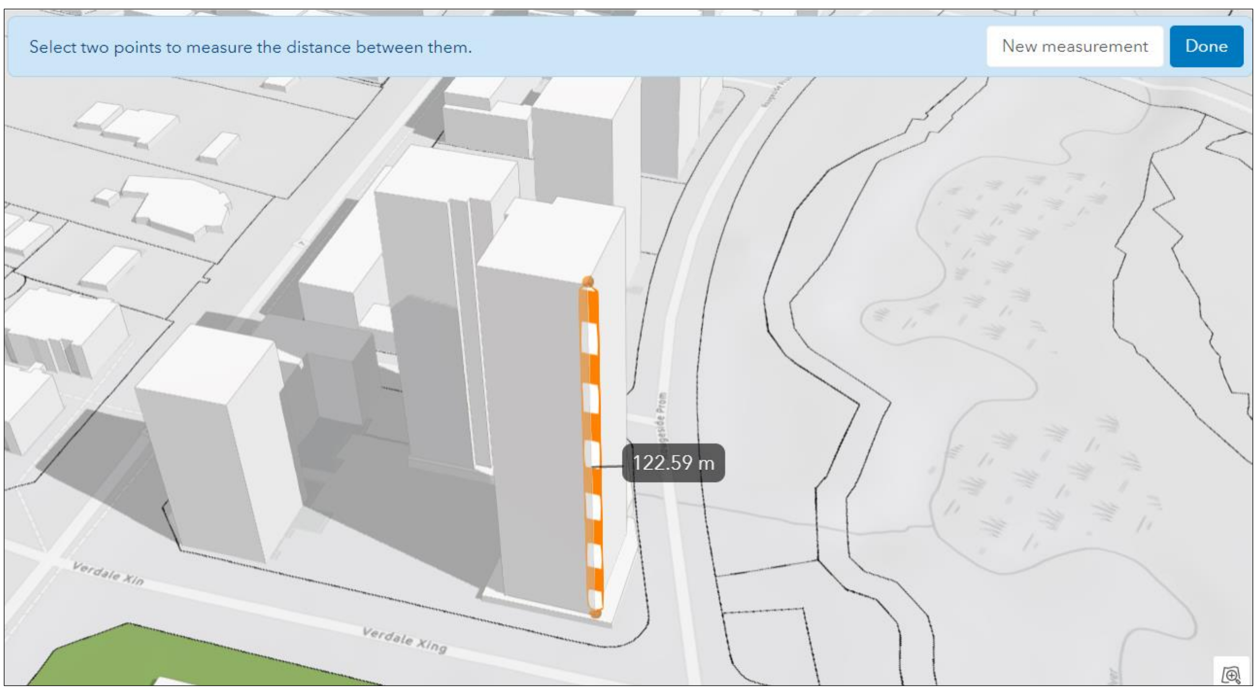


Figure 30: Example of the Measure Tool in ArcGIS Urban

4.4.2 Line of Sight

The Line of Sight tool allows users to complete quick and effective viewshed (view corridor) analysis directly within ArcGIS Urban. This is done by selecting an observer point and one or more targets. ArcGIS Urban will then determine whether the target can be seen from the observer point. If so, the line segment will be highlighted in green. If not, the line segment will be highlighted in red. Figure 31 below provides an example of a result from the Line of Sight tool.

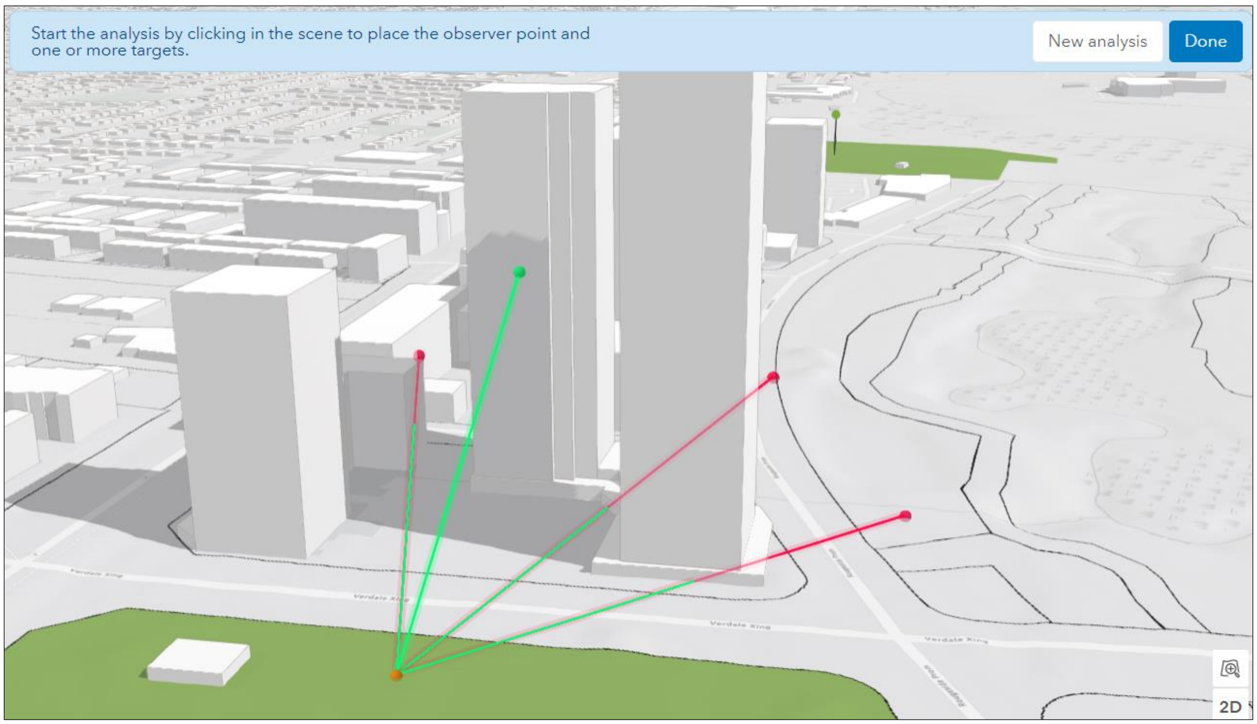


Figure 31: Example of the Line of Sight Tool in ArcGIS Urban

4.4.3 Profile

The Profile tool can be used to analyze the elevation within an area of interest. This is done by drawing a line segment along the ground in a desired area. ArcGIS Urban will then develop a profile of the ground and building elevations along the line segment. This tool offers a quick and easy way to assess the landscape and building elevation profiles within a given area. Figure 32 below provides an example of the Profile tool.

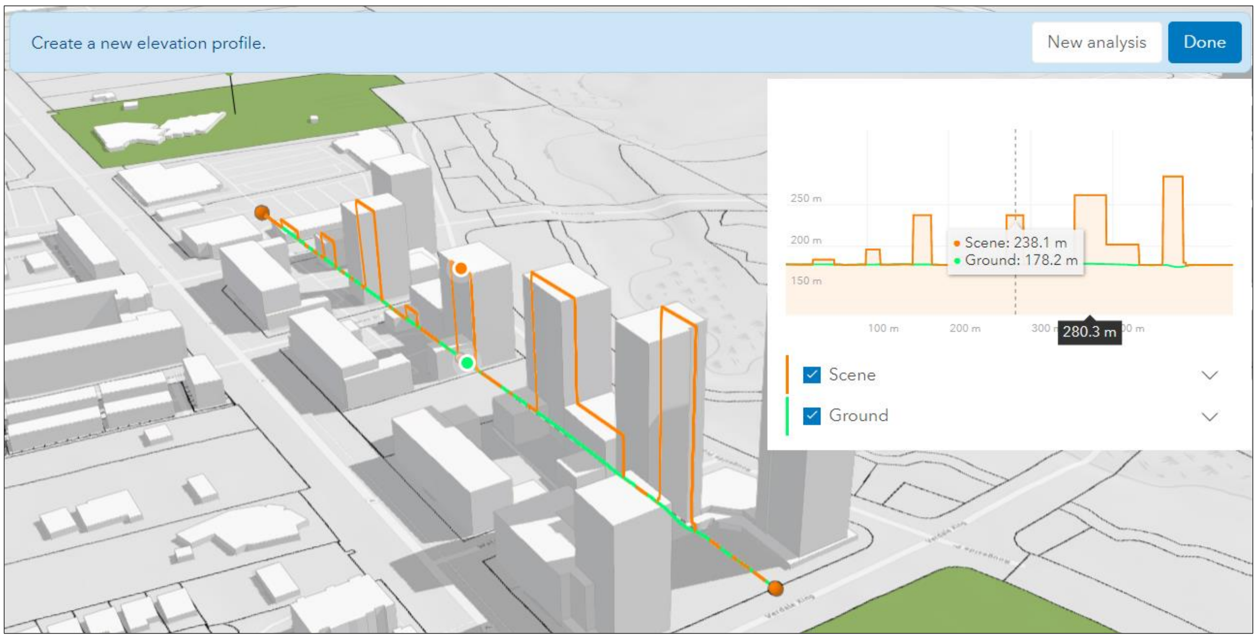


Figure 32: Example of the Profile Tool in ArcGIS Urban

4.4.4 Shadow Cast Analysis

ArcGIS Urban also allows users to complete Shadow Cast analysis easily within the software. The user simply needs to set their time zone, a time range and a specific date and ArcGIS Urban will generate shadows based on this information. The user can also specify whether they want to show the total shadow duration, or areas that are contained within shadow for a set number of hours during the day (as shown in the example below). Figure 33 below shows an example of the Shadow Cast Analysis tool.

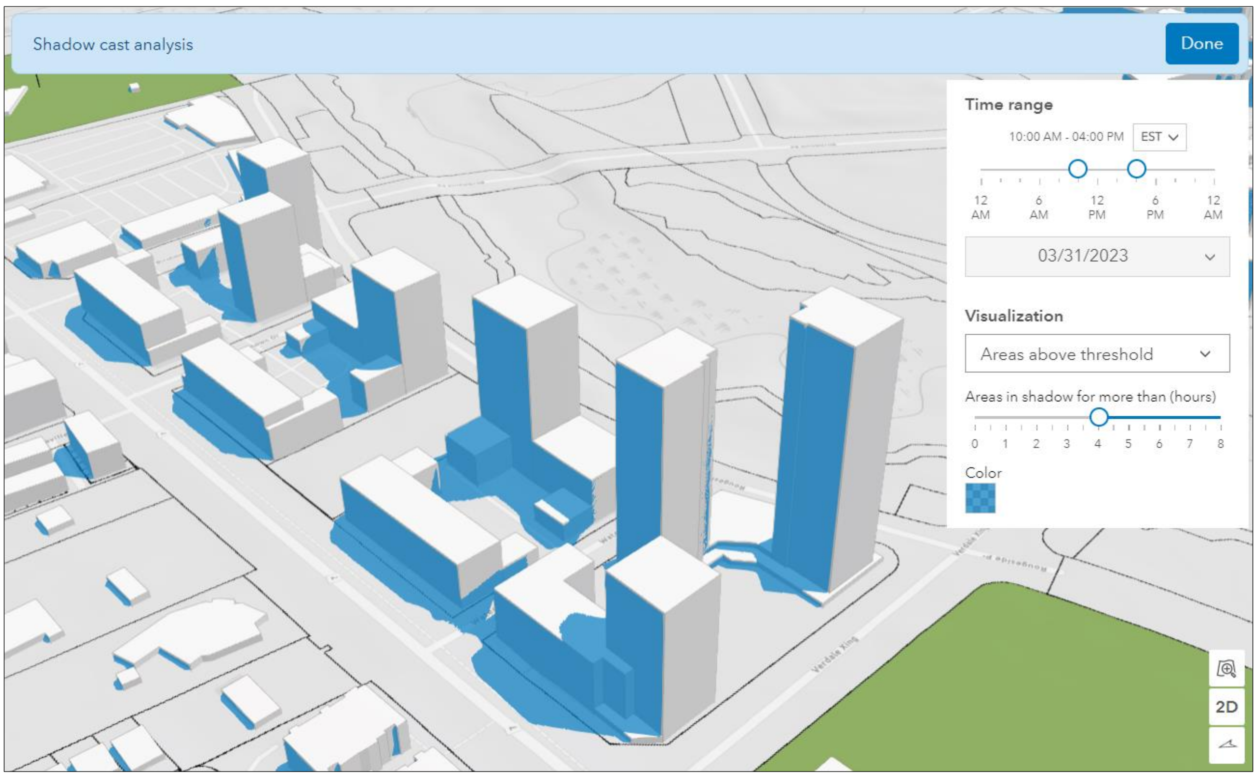


Figure 33: Example of the Shadow Cast Analysis Tool in ArcGIS Urban

4.5 Integration with CityEngine

ArcGIS Urban does have some integration with CityEngine, although the level of integration is somewhat limited. ArcGIS Urban models can be imported into to CityEngine, allowing users to edit ArcGIS Urban models from CityEngine and synchronize these changes back to ArcGIS Urban.

A Plan or Project must first be created in ArcGIS Urban before being imported into CityEngine. The Plan or Project can be imported as a CityEngine Scene. This allows CityEngine to link to the ArcGIS Urban model (but only at the individual Project or Plan level).

Once a Plan or Project has been created in ArcGIS Urban, it can be accessed in the Navigator window within CityEngine. The user will first need to login to ArcGIS Online from within CityEngine to be able to connect to any ArcGIS Urban models saved to their ArcGIS Online account.

Once the user has logged in, they will see a list of their ArcGIS Urban models, along with all Plans and Projects available to import into CityEngine. A Plan or Project can then be imported by right clicking on the Plan or Project and selecting “Import as new Scene.” An example of accessing ArcGIS Urban from the Navigator window in CityEngine is shown in Figure 34 below.

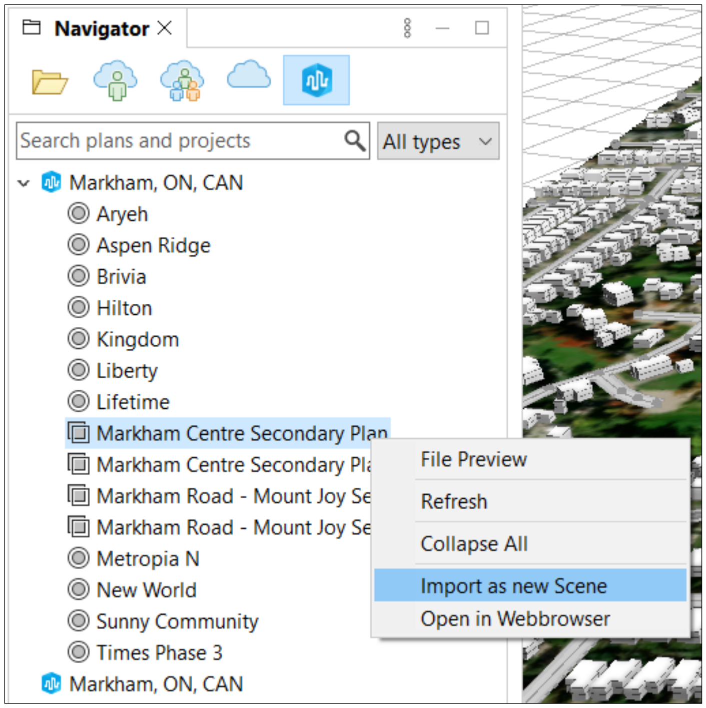


Figure 34: Importing an ArcGIS Urban Plan or Project into a Scene in CityEngine

The Plan or Project will then be imported into CityEngine as a new Scene.

When CityEngine is linked to an ArcGIS Urban model, any zoning or land use parameters, as well as items defined in the System of Record (i.e. Zoning types, Land use types, Space use types, Building types, etc.) established within ArcGIS Urban will be applied within CityEngine. This means that any conceptual buildings created within CityEngine will abide by the parameters established in the ArcGIS Urban model rather than any procedural rules created in CityEngine.

If any changes are made to the parcels, shapes, or scenarios in CityEngine, they can be synced with the ArcGIS Urban model. Plausible buildings generated in the Scene can also be synced to the ArcGIS Urban model, however these building will be generated based on the zoning or land use parameters established in ArcGIS Urban (i.e. not based on rules created in CityEngine). This can be done by navigating to the ArcGIS Urban tab along the top of the CityEngine window and selecting the “Save changes to Urban…” option, as shown in Figure 35.

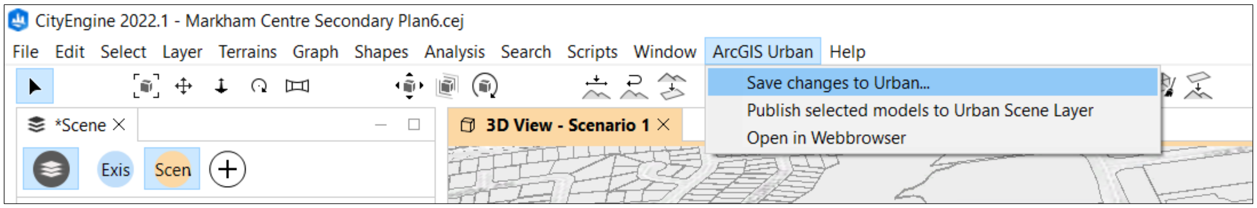


Figure 35: Saving Changes to ArcGIS Urban from CityEngine

CityEngine models or other 3D features can be published as Urban Scene Layers to ArcGIS Online and imported into ArcGIS Urban, however, these will be added as “Scenario design context” and will be used as a visual reference only (i.e. they do not respond to any zoning or land use parameters in ArcGIS Urban). This can be done within CityEngine by first selecting the models you would like to publish, then selecting the “Publish selected models to Urban Scene Layer” option under the ArcGIS Urban tab.

The integration method between ArcGIS Urban and CityEngine does impose some limitations since the parameters established in ArcGIS Urban take priority when the two programs are linked together. While this does seem necessary based on ArcGIS Urban’s structure and requirements, it does mean that some of the more sophisticated features found in CityEngine (i.e. custom procedural rules, building textures, etc.) become limited when used in conjunction with ArcGIS Urban. While more detailed models from CityEngine can be uploaded and displayed in ArcGIS Urban, they will only serve as a visual reference and will not respond to any of the land use or zoning parameters or contribute to any metrics within an Urban model.

4.6 Findings and Limitations

Key findings and limitations identified during the ArcGIS Urban experiment are summarized in the list below:

1. ArcGIS Urban does offer a more user-friendly experience than CityEngine and can be used more easily by non-technical staff.

2. ArcGIS Urban being browser-based makes the software more accessible to a wider variety of staff since specialized computer hardware is not required.

3. If all data inputs and land use or zoning parameters are properly defined, ArcGIS Urban does offer a quick and easy way to generate and modify concepts and track active development projects throughout the City.

4. ArcGIS Urban may offer a number of sharing options and interactive functionality that could be useful to the City for public engagement purposes.

5. ArcGIS Urban does offer a number of functions that are similar to CityEngine, but are more limited. This includes the ability to create different development scenarios (for both Plans and Projects), generate buildings from rules (i.e. land use or zoning parameters), displaying and calculating site statistics (although only Plans allow for automatic reporting of site statistics) and basic analysis tools. However, ArcGIS Urban as it stands currently is not a directly replacement for CityEngine.

6. Plans and Projects are managed independently in ArcGIS Urban. There is no interaction between the two items, meaning if a Project is within the same geographic area as a Plan, it will still not be recognized by the Plan. This also means that Projects will not contribute to a Plan's metrics, so if the City would like a development project to contribute to the metrics of a Plan (i.e. for measuring Secondary Plan targets) the development would have to be recreated within the Plan.

7. ArcGIS Urban is intended to maintain models for entire Cities, but we experienced a downgrade in performance speed and an increase in loading times when using parcel data for the whole City. The performance increased significantly when using parcel data that only covered the two Secondary Plan areas.

8. While Plans can be used for quickly creating development concepts, Projects are used more for asset management and tracking status of development projects.

9. Separate Plans do not interact with each other in a significant way. For example, Zoning Plans cannot currently react or feed into the metrics of a larger Land Use Plans if they cover the same geographic location.

10. Land Use Plans are intended for higher-level planning initiatives and do not have the ability to include setbacks or stepbacks in their land use parameters. However, Zoning Plans do allow users to define setbacks and stepbacks.

11. Because of the lack of interaction between Plans and the omission of setbacks from Land Use Plans, it may be best to manage the Secondary Plan areas (and any future Secondary Plan areas) as Zoning Plans rather than Land Use Plans.

12. Importing 3D existing features into Projects or Plans is currently limited. 3D features imported into ArcGIS Urban are treated as “Scenario design context” and essentially act as a visual reference. They do not react to the Land Use or Zoning parameters that have been defined in the model. However, they do provide some visualization benefits and can be used with ArcGIS Urban’s analysis tools.

13. There is a method to visually represent Projects within the overview map but this feature did not work during our testing. This method requires that the Project information be batch uploaded, and an accompanying building footprint layer be imported from ArcGIS Online. Despite both uploads being successfully uploaded during our testing, the Project models did not display.

14. ArcGIS Urban does offer the ability to import building footprints into a Plan but footprints must be imported as a certain Building type, and they abide by the maximum height of the defined Building type when imported (i.e. the height attribute included in the GIS data is ignored). This means that the user would need to manually edit the heights of the footprints once imported.

15. Generally, it seems that ArcGIS Urban is be better suited for creating conceptual buildings natively within the software rather than creating concepts in other software programs (i.e. Sketchup) and importing them into ArcGIS Urban.

16. Interactions between CityEngine and ArcGIS Urban are also somewhat limited since the parameters and structure of ArcGIS Urban is imposed on CityEngine when the two are linked together. While this is acceptable if only basic zoning parameters are required (i.e. max height, lot coverage, setbacks, etc) any more complex rules established in CityEngine will not be applied to the ArcGIS Urban model. Models generated in CityEngine can be uploaded to ArcGIS Urban but act only as “Scenario design context” layers.

17. Buildings created in ArcGIS Urban are not photo realistic (unless a third party “mesh” is purchased) and mainly represent the massing of a building. More realistic buildings can be imported from CityEngine but they will act as context layers that do not react to land use or zoning parameters. As such, CityEngine is more suited to creating models for use in renderings.

18. Due to certain limitations of ArcGIS Urban (i.e. importing existing 3D features into the model, manual entry of information and metrics for Projects, input of data requirements) a significant amount of re-work would be required to set up an ArcGIS Urban model initially. However, once this initial setup is complete, less effort will be required to maintain the model moving forward.

5.0 Workflow Options

Based on our experience working with both CityEngine and ArcGIS Urban, we have identified two general workflow options that could be suitable depending on the approach the City would like to take. Due to the City's needs and future goals for the 3D models (i.e. development of rules, applying textures, importing existing 3D features and other datasets) we recommend that the City use CityEngine at a minimum. As such, we have provided two possible workflow options below, one where only CityEngine is used, and the other where both CityEngine and ArcGIS Urban are used in combination.

5.1 Workflow Option 1: CityEngine Only

For Option 1 where only CityEngine is used to maintain the 3D model moving forward, it is likely that only technical staff or staff with specialized training would be interacting with the model directly. For this option, the City would require either named licenses for all staff who would be using CityEngine, or a pool of floating licenses that City staff can draw from as required.

Generally, the City's workflow for updating and maintaining the model under this option would resemble the following approach:

- Technical or specialized staff manage the 3D model in CityEngine;
- Any modelling, analysis or update requests are submitted to the technical team by non-technical staff;
- The technical team completes the request and provides the model deliverable (i.e. visualization, analysis result, model update, etc.) to the requester; and,
- Non-technical staff do not interact with the model directly.

The benefit to this approach is that the technical team only needs to maintain the CityEngine model(s) and does not need to maintain a model in ArcGIS Urban. The drawback of this option is that non-technical staff are not able to use the model directly, and all requests will need to go through the technical team. This could result in an increased workload on the technical team and increased turnaround time if many requests are received around the same time.

5.2 Workflow Option 2: CityEngine and ArcGIS Urban

In Option 2, both technical and non-technical staff would be able to interact with the model. For this option, the City will need to acquire both CityEngine licenses (named or floating) and licenses for ArcGIS Urban. Given that ArcGIS Urban is administered through ArcGIS Online, there is the ability to share models with staff members with varying degrees of permissions. Depending on the City's requirements, the City may only need to acquire full ArcGIS Urban licenses for certain key members of the team, rather than for each team member using ArcGIS Urban.

The City's workflow for updating, maintaining, and using the model for this option would resemble the following approach:

- Technical or specialized staff still maintain the model in CityEngine;
- Technical staff use CityEngine to import 3D assets, or complete any complex tasks outside the scope of ArcGIS Urban;
- Technical or specialized staff handle initial configuration and larger updates (i.e. adding new models, updating land use or zoning parameters, uploading new datasets, etc.) to ArcGIS Urban;
- Technical staff can use either CityEngine or ArcGIS Urban to make edits to ArcGIS Urban model, if requested;
- Planning staff use ArcGIS Urban to track development projects, visualize possible development scenarios, complete analysis tasks, create design concepts or monitor Secondary Plan targets; and,
- City staff also use ArcGIS Urban for public engagement (optional).

The benefit of this option is that both technical and non-technical staff can make use of the model. This may alleviate some of the more minor requests from the technical team as non-technical staff can complete these tasks in ArcGIS Urban themselves. There is also the option to use ArcGIS Urban as a public engagement tool in the future. The drawbacks to this approach is that the technical staff will need to maintain both the CityEngine model and the ArcGIS Urban model. Depending on the complexity of the models and the approach to maintain them, this could result in an increase in workload for technical staff.

In addition, there would be a significant amount of rework required initially to translate the models from CityEngine to ArcGIS Urban due to ArcGIS Urban's limitations with importing existing features. Once this initial work is complete, however, maintaining the ArcGIS Urban model moving forward would require less effort from technical staff. There would also be additional licensing costs associated with using both CityEngine and ArcGIS Urban since both pieces of software would require multiple licenses.

The preferred option will depend on the needs of City staff, the associated costs and the level of effort required to maintain the model(s) moving forward.

6.0 Conclusions and Recommendations

Based on the work completed while developing the CityEngine model and our experience undertaking the ArcGIS Urban Experiment, we have identified the following conclusions:

1. Generally, CityEngine provides more functionality than ArcGIS Urban but requires specialized training;

2. ArcGIS Urban is more accessible than CityEngine but is more limited in terms of functionality;

3. Interaction between CityEngine and ArcGIS Urban is possible but limited. There are potential workflows options where this setup could work, but it would depend on the approach that the City takes in maintaining both models;

4. ArcGIS Urban does contain similar features to CityEngine (i.e. development scenarios, generating models based on rules, reporting and site statistics, analysis tools) but in a more simplified and limited capacity;

5. ArcGIS Urban is not a direct replacement for CityEngine;

6. Developing procedural rules in CityEngine requires knowledge of Computer Generated Architecture (CGA) programming language. Similar rules can be established in ArcGIS Urban using a graphical interface (i.e. does not required programming knowledge). However, procedural rules created in CityEngine can be much more complex than those created in ArcGIS Urban;

7. Additional work would be required to maintain models in CityEngine and ArcGIS Urban. There would also be an initial setup process required to remodel the buildings in ArcGIS Urban if the City wishes to use it;

8. In ArcGIS Urban, Plans are used for creating development concepts for longer-term planning initiatives and monitoring the resulting metrics. Projects are used for asset management and tracking status of shorter-term development projects;

9. Importing external 3D models into ArcGIS Urban is limited. Models imported from CityEngine are used for visual context and do not react to land use or zoning parameters established in ArcGIS Urban;

10. ArcGIS Urban is currently more suited for creating models natively within the software rather than developing models in a separate program (i.e. CityEngine, Sketchup) and importing them into ArcGIS Urban;

11. Buildings created in ArcGIS Urban are not photo realistic (unless a third party “mesh” is purchased) and mainly represent the massing of a building; and,

12. CityEngine is more suited to creating models for use in renderings as textures and additional details can be applied.

6.1 General Recommendations

The following are general recommendations for the City to consider when moving forward with the 3D model:

1. CityEngine is required at minimum to use the 3D models prepared for this project. We recommend that the City acquire at least a single CityEngine license in order to make use of the model;

2. Consider the intended users and the preferred workflows for maintaining the model(s) in the future. This should inform whether to proceed with just CityEngine or incorporate both CityEngine and ArcGIS Urban into the City’s operations;

3. The City should look to develop procedural rules based on zoning or land use regulations for the CityEngine model to make the most of the software capabilities;

4. If the City chooses to use ArcGIS Urban, we recommend that it be used by Planning and Urban Design staff to create design concepts, assist with visualization and analysis, and track development applications. Any more complex tasks are best handled in CityEngine by technical staff;

5. ArcGIS Urban provides additional public engagement tools that are not offered in CityEngine. If the City wishes to use the model for public engagement purposes, ArcGIS Urban is a possible consideration; and,

6. If the City is unsure on the viability of using ArcGIS Urban or the number of licenses required to meet the City's needs, it may be worth undertaking a pilot project using a single license of ArcGIS Urban before purchasing additional licenses.

6.2 Technical Recommendations

The following are more specific technical recommendations to consider when maintaining the models in both CityEngine and ArcGIS Urban (assuming that the City uses ArcGIS Urban in the future):

6.2.1 CityEngine

1. When importing a CityEngine project into a workspace, use the Copy option to copy all files to a local machine and use the Link option to link a project to a workspace from its current file location. The Link option can be used when multiple users wish to use the same data sources or edit the same model;

2. It is recommended that users do not edit the same CEJ file simultaneously. CEJs are prone to corruption and editing them simultaneously may cause the file to become corrupted;

3. Ensure that backups are maintained for any CEJ files, especially when multiple users are editing the same Scene;

4. CityEngine has an image size limit of 8192 x 8192 pixels. Ensure that any aerial images or other terrain data imported into CityEngine is smaller than this limit to avoid errors or other issues;

5. Use static models only when you do not want the building to be modified by any procedural rules (i.e. for landmark buildings);

6. If adding new parcel data to the model, run the “Compute Edge Attributes” tool to assign the proper edge orientation to the parcel data. This will ensure that any procedural rules applied to the parcel data will accurately reflect building setbacks and stepbacks;

7. When importing external 3D models, ensure that buildings or features are imported one at a time, or in clusters representing a single development. 3D models imported into CityEngine will be treated as one feature and separating the features in CityEngine is labour intensive;

8. CityEngine does not import Sketchup files natively. Sketchup files must be exported to COLLADA (.dae) or FBX before being imported into CityEngine;

9. 3D features imported into CityEngine will maintain the scale that they were created in. It is important to ensure that proper scaling is applied within the original model before importing it into CityEngine. CityEngine does allow models to be rescaled within the software, but the exercise is less precise than properly scaling a model before importing it;

10. Use road centreline data to generate road datasets (graphs) in CityEngine rather than road polygon layers. CityEngine interprets road centreline data as roadways but does not interpret polygons as well. The road datasets (graphs) also play a part in assigning parcel edges, which are critical to generating accurate buildings based on procedural rules; and,

11. The “Reports” window in CityEngine can only display numerical values. It is recommended that any string values or non-numeric attributes be stored in a separate dataset within the model, such as the parcel layer.

6.2.2 ArcGIS Urban

1. Plans should be used for creating development concepts based on zoning and land use parameters and for calculating the resulting metrics;

2. Projects should be used for tracking the status of active development applications throughout the City;

3. Utilize Zoning Plans to represent the Secondary Plan areas to make sure that setbacks can be defined and Plan metrics can be calculated for the entire Plan area;

4. When establishing Zoning data or Land Use data in ArcGIS Urban, ensure that the Zoning types or Land Use types match those included in the zoning or land use boundaries;

5. There was a noticeable downgrade in performance when using parcel data for the entire City. It is recommended that the City explore ways to optimize performance or only include parcel data covering the two Secondary Plan areas (and any future models); and,

6. It is recommended that all of the data inputs and settings in ArcGIS Urban's "System of Record" (i.e. Zoning types, Land Use types, Space Use types, Building types) be configured in accordance with the City's land use policies and zoning by-law to ensure that generated buildings reflect real-world regulations. This will ensure that ArcGIS Urban's functions work properly and any design concepts will be accurately represented in the model.