

WITNESS - Unity3D Connectivity

PROOF-OF-CONCEPT

26/02/2018

1. Introduction

1.1. Purpose

The purpose of this document is to outline functionality required to create a proof-of-concept Unity application that demonstrates how a WITNESS simulation model can be visualized in a 3D Unity environment.

1.2. Goal

The goal is to be able to demonstrate that the capability described by this document is possible to create using Unity. The existing WITNESS 3D capability can be used as a reference.

A full fidelity solution is not required, just a good indication that the desired functionality is possible within an acceptable performance.

1.3. Background

WITNESS is a desktop process simulation modelling studio. It enables modellers to develop simulation models to provide process insight using a discrete event simulation method (DES).

Models are created by connecting a series of modelling elements on a 2D design surface. Each 2D modelling element has a 3D object association. At any time, the modeller can generate a 3D representation of the model, using the Quick 3D function.

The Quick 3D function sends a series of CRUD (create, update & delete) commands to a 3D engine which creates the 3D representation of the model.

A TCP socket connection is used to send these commands between WITNESS and the 3D engine. These commands are described in the “3D Command Reference 21.docx”¹ document. These commands are serialized as xml and sent over the socket connection.

1.4. Terminology

WITNESS	Lanner’s simulation product
3D engine	3 rd party rendering
Machine	A WITNESS element that represents a real machine in a manufacturing environment, sometimes called an Activity
Part	A WITNESS element that represents a part that moves through the simulation model from one element to the next, sometimes called an Entity
Buffer	A WITNESS element that represents a queue of items
Path	A WITNESS element that are made up of a series of line segments and arcs that also have may have gradients
Extrude	The ability to take a small geometry section and extend it along a Path
Labor	A WITNESS element that is used to represent a shared constraining resource that is required to carry out an Activity or Machine process step

2. Features for Proof-of-Concept

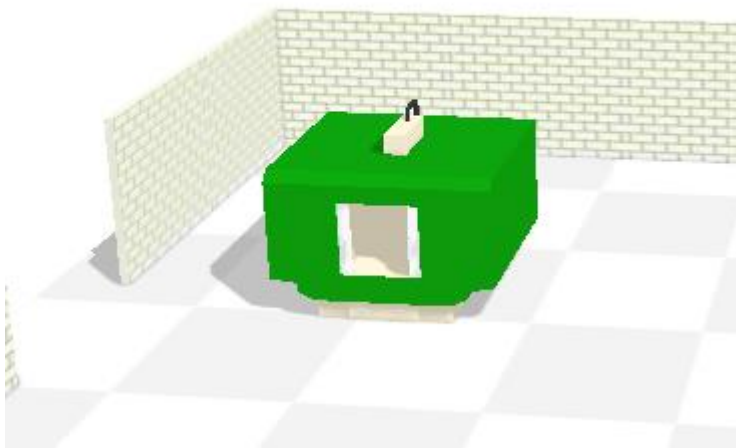
2.1. Create and Position objects in a 3D world

Read the xml commands and position geometry instances in the 3D world, in response to a <load>, <create> and <update> command. The <delete> command should destroy them.

Ensure that the coordinate system can be adapted to work with Unity. For simplicity, these can be read from a xml file rather than directly linking to WITNESS.

Ensure that geometries can be loaded from an external library.

For example: from the Quick3D-1.mod positions a Machine instance in 3D space



Note the current implementation of the <load> command returns a <geometryInfo> command containing the size of the 3D geometry. In the command sequence this is used to scale the size and orientation of the 3D geometry instance in-line with the 2D footprint size, using an <update> command. Dealing with this <geometryInfo> can be ignored for the proof-of-concept.

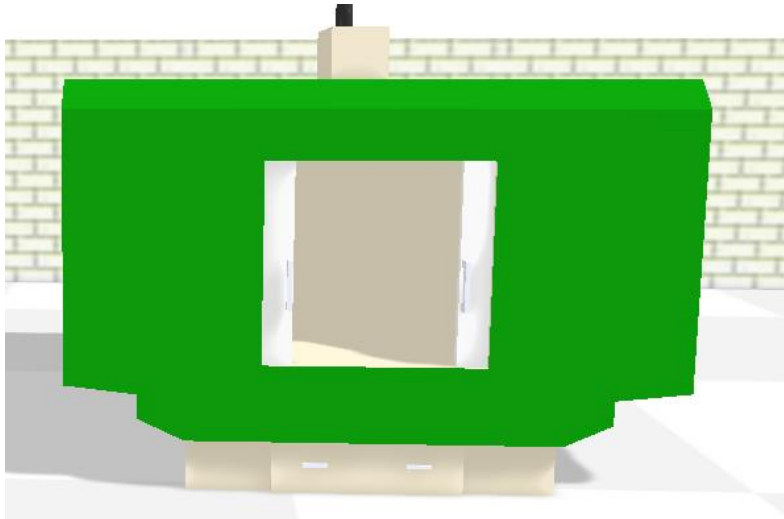
2.2. Place a Part at a Machine

Place a WITNESS Part at a Machine element in 3D. As the simulation model runs a Part is moved from element to element. Each modelling element defines a queuing position and direction in 3D. This information is passed with the <create> command when creating the 3D geometry instance. The document “Witness W3D Queue Implementation.docx”³ details the part queuing behaviours.

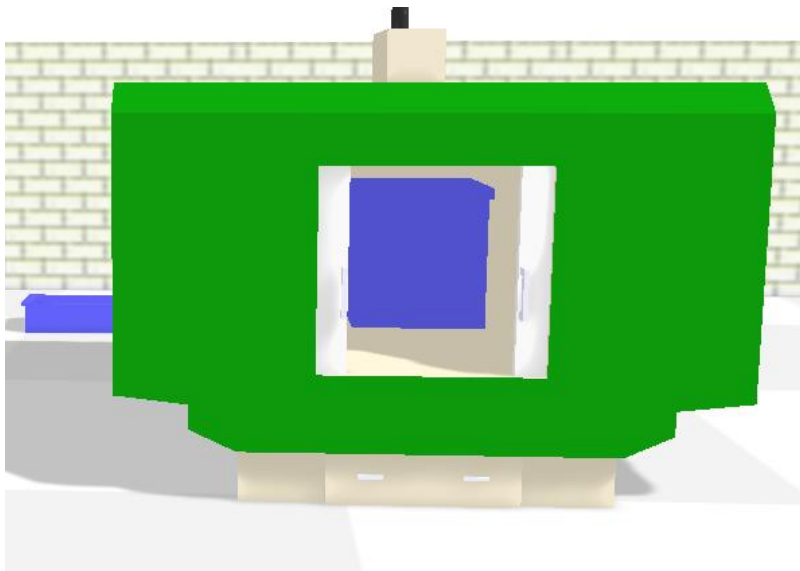
Ensures that the Part queuing position at an element can be replicated.

For example: from the Quick3D-1.mod

- Load Quick3D-1.mod
- Run the Quick3D command
- This is a view of the model without a Part, at time 0.0



- Run the model until time 1.0
- This is a view of the model with a Part located at the Machine, at time 1.0



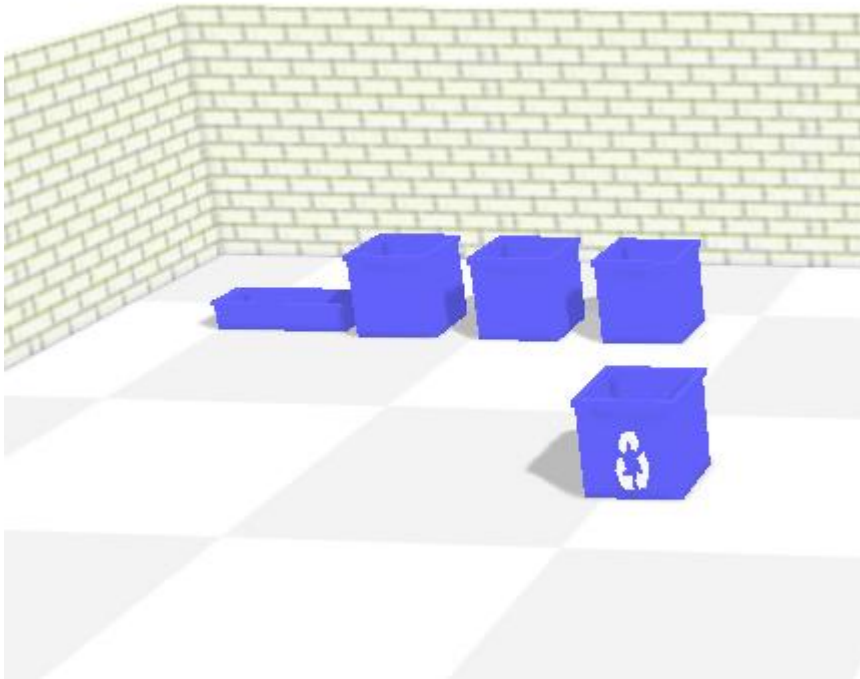
2.3. Create a simple Path and Position a Part on it

A simple WITNESS Path is a path that is composed of a series of line segments.

Ensures that a Path can be recreated and Parts placed on that Path.

For example: from the Quick3D-4.mod

- Load the Quick3D-4.mod model
- Run the Quick3D command
- Run the model to time 1.0
- This is a view of the model without a Part, at time 1.0. Notice the 4 parts along the path. Not all the Parts are the same size. In this case the Path is invisible as it does not have an associated 3D geometry.



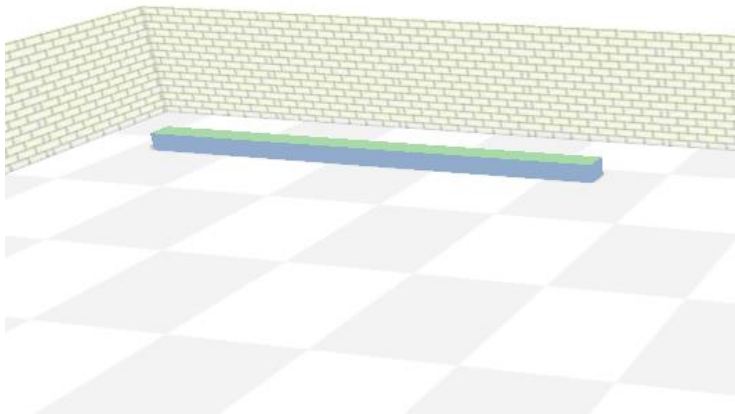
Note: when a Path is created, using the <create> command the <queueInfo> is used to specify the behaviour of the Part positioning. The options are partOver, partUnder, partCentre. These commands are described in the “3D Command Reference 21.docx”¹ document.

2.4. Apply an Extrusion to a simple Path

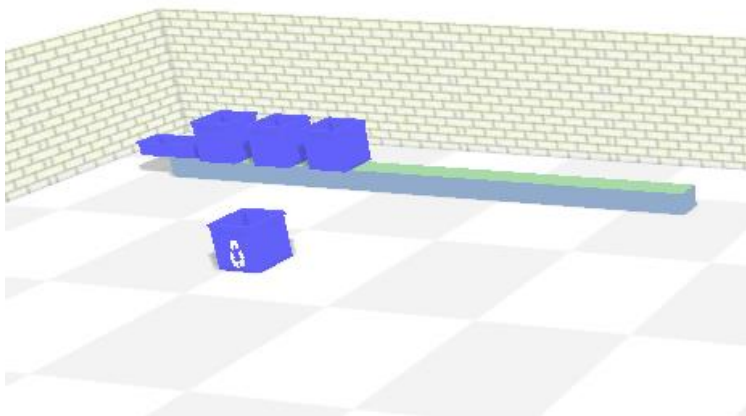
Ensures that a Path can be using a geometry that is “extruded” along the length of the Path.

For example: from the Quick3D-5.mod

- Load the Quick3D-5.mod model
- Run the Quick3D command
- This is a view of the model without a Parts on the Path, at time 0.0



- Run the model to time 1.0 and notice the Parts are placed on top (partOver) of the Path.



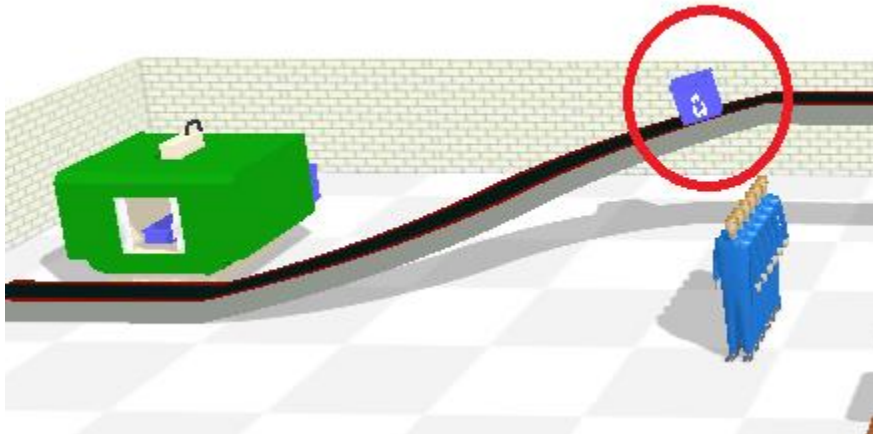
2.5. Move a Part along a simple Path

As the model runs a series of <update> commands change the position of a Part along a Path. It does this by changing a Part's percentage position along the Path.

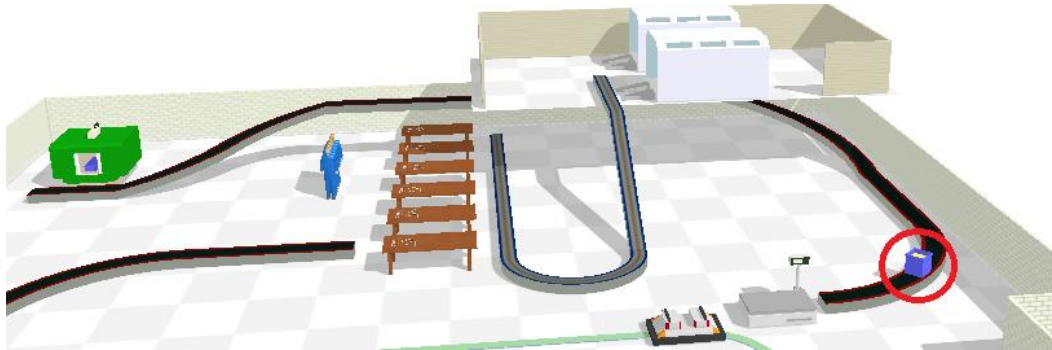
Ensures that the Part can be moved along a Path in response to changing its percentage position.

For example: from the Quick3D.mod

- Load the Quick3D.mod model
- Run the Quick3D command
- Run the model to time 4.5 and notice a Part has moved along the conveyor named *Conv1*.



- Run it again to time 6.8 and notice that the Part has almost reached the end of the Path.



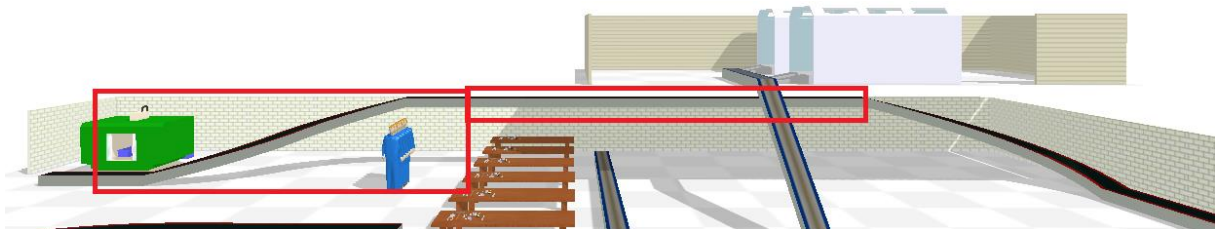
2.6. Position a Part on curved Paths

Paths are a WITNESS element that are made up of a series of line segments and arcs that also may have a gradient.

Ensures that a curved Path with a gradient can be created from the set of segments and arcs supplied by the <create> command.

For example: from the Quick3D.mod

- Load the Quick3D.mod model
- Run the Quick3D command
- Notice the conveyor named *Conv1* is drawn as a Path whose vertical height changes. It also has an associated geometry that is “extruded” along the Path.



For the proof-of-concept it is not necessary to extrude a geometry along the curved path, just an indication that this is possible.

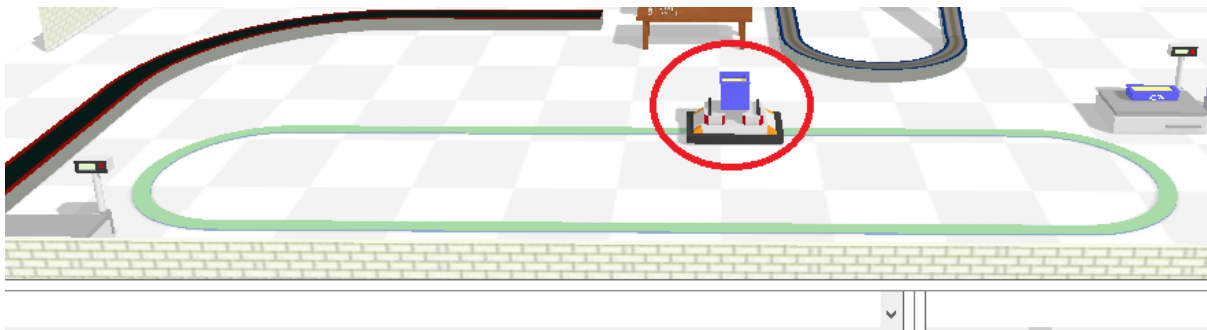
2.7. Move a Part that is contained by another moving Carrier

WITNESS has modelling elements that model tracks and vehicles. Tracks are displayed as a Path and Vehicles move along a Path and carry a Part.

Ensure that a vehicle, that has a queuing position containing a Part, can move along a Path.

For Example: from the Quick3D.mod

- Load the Quick3D.mod model
- Run the Quick3D command
- Run the model until time 15. Notice that the carrier travels along the Path and contains a Part (box).



2.8. Text

WITNESS has a number of variables / properties whose values update as the model runs. These are displayed on the design surface in 2D. WITNESS supports updating these values, using the <update> command but currently this is not implemented on the existing 3D engine.

Ensure that text, from the simulation mode, can be displayed in the 3D scene, on the floor.

For example: from the Quick3D-6.mod

- Load the Quick3D-6.mod model
- Run the Quick3D command
- Run the model to time 15 and observe the trace output

The command used to create the text item is as follows:

```
<create time="0.000000"
      instanceName="[124] Speed(0) - Value"
      x="25.750000" y="0.000000" z="21.312500">
  <text>
    <![CDATA[0.000]]>
  </text>
</create>
```

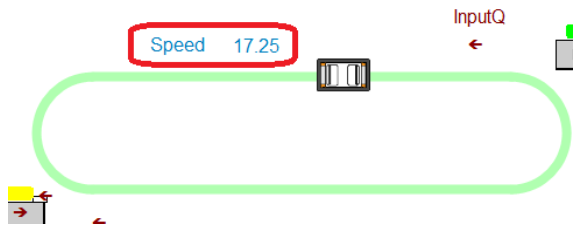
The command used to update the text item with a value is as follows:

```
<update time="0.000000" instanceName="[124] Speed(0) - Value">
  <text>
```



```
<![CDATA[17.250]]>
</text>
</update>
```

On the 2D surface it is displayed as follows:



2.9. Animations

WITNESS can start and stop animations contained by a geometry instance, using the animation start, update and stop commands described in the “3D Command Reference 21.docx”¹ document.

For an example see [Walking people](#).

2.10. Walking people

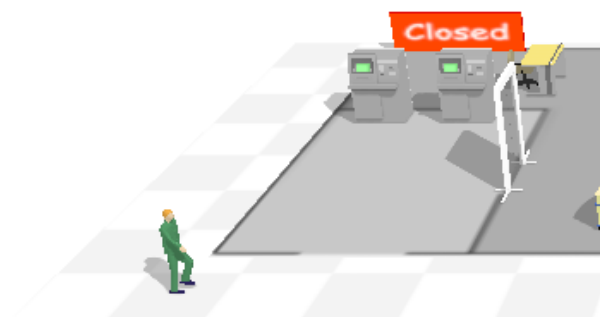
WITNESS has a Labor element that is used to represent a resource that is required to carry out an Activity. A Labor instance “walks” in 3D along a Path.

Ensure that an animation can be started on a geometry instance. This example uses a walking person.

The existing 3D people are fixed geometries with animations, suggested alternatives would be welcome – humanoids, deformable mesh, bone structure etc.

For example: from the BankQuick3D.mod

- Load the Quick3D-6.mod model
- Run the Quick3D command
- Run the model until time 1.0. Notice that the “Green Person” starts to walk.



The <animationStart> command is used to start an animation.

2.11. Robot Arm

Currently, in WITNESS 3D, it is challenging to model a robot that picks up a Part from one location and moves it to another.

In the simulation a robot is modelled as a WITNESS Machine which will have an associated 3D geometry. The 3D geometry will have a queuing position for a Part i.e. the end of the arm as illustrated below.

When a Part arrives at the Machine, WITNESS triggers an animation to start in the associated geometry instance, giving a duration (the cycle time for the Machine). When the Machine has completed its cycle, it triggers an animation stop.

Ensure that a robot arm behaviour can be demonstrated.

For example: from the Robot Animation.mp4

- Open the Robot Animation.mp4
- Run the video
- Note the robot picks up the Part from one location



- And moves it to another





The difficulty in the current solution is linked to scaling of the 3D geometry instance during creation. Scaling the size of the robot, during creation, also scales the part queue, therefore the Part changes in size when placed in the robot geometry hierarchy.

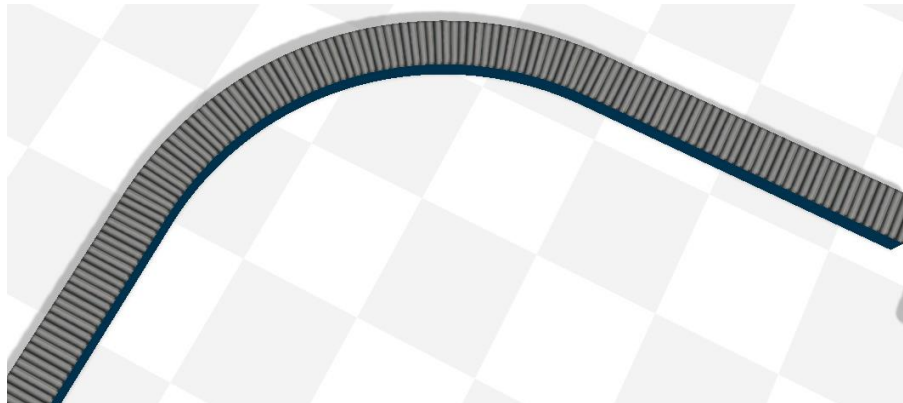
2.12. Performance

WITNESS Quick3D when running simulation model needs to deliver a reasonable run speed on a 'normal' laptop such that the view is a viable way to watch the model. By 'normal' it is accepted that it should run as well as the current 3D engine solution.

3. Features Not for Proof-of-Concept

List of items that are not required for a proof-of-concept but listed as they would be required for an any end application / solution.

- Socket connectivity between WITNESS and the Unity app
- The ability to return geometry information following a load command. i.e. return 3D bounding box information for a geometry
- The ability to be able to target the Unity application for WebGL (browser friendly)
- Toggle between full screen resizable window
- Host the Unity app in a WITNESS window (if possible)
- The ability to be able to turn on / off shadows
- The ability to implement the commands outlined by “3D Command Reference 21.docx”¹ document
- Work over a remote desktop connection
- The ability to tile the floor with a texture. This is often used to create a CAD backdrop for the floor of a factory building
- The ability to apply an “extrusion” to a shape. WITNESS uses this feature to create walls and windows where textures are applied
- The ability to change the material of a 3D geometry instance
- The ability to display Japanese characters in the 3D scene as text
- The ability to respond to a <query> command that gives the position, rotation and scaling of a geometry instance (transform). See the <query> command in the “3D Command Reference 21.docx”¹ document
- The ability to apply an extrusion to a curved Path. More than just a box section. The material on curved sections are altered to take account of the curved path. For example:
 - Load Quick3D-7.mod
 - Run the Quick3D command
 - Note the material on the curved section is dynamically adapted during the <create> command



- The ability to edit 3D shape queuing positions
- Ability to have the 3D scene lit correctly. Currently we use a mix of directional light and ambient lighting
- Ability to easily switch views - Front, back, left, right, top & reset to default view with a some form of command
- Ability to use AR/VR headsets
- Import user shapes (.fbx, .obj, .dae)

- Understand the features that the Unity environment offer
 - Choice from a range of camera positions – ideal if these could be defined for a model
 - Ability to preview 3D geometries when creating a mapping between 2D and 3D shapes. Perhaps needs to be a separate viewer for support file formats
 - Ability to record the 3D scene as a video while running the model
 - Collision detection, what is supported?
- Ability to be able to move a 3D shape in Unity and update the WITNESS 2D layout. Currently there is no support in the 3D command protocol for this capability (for future)

4. References

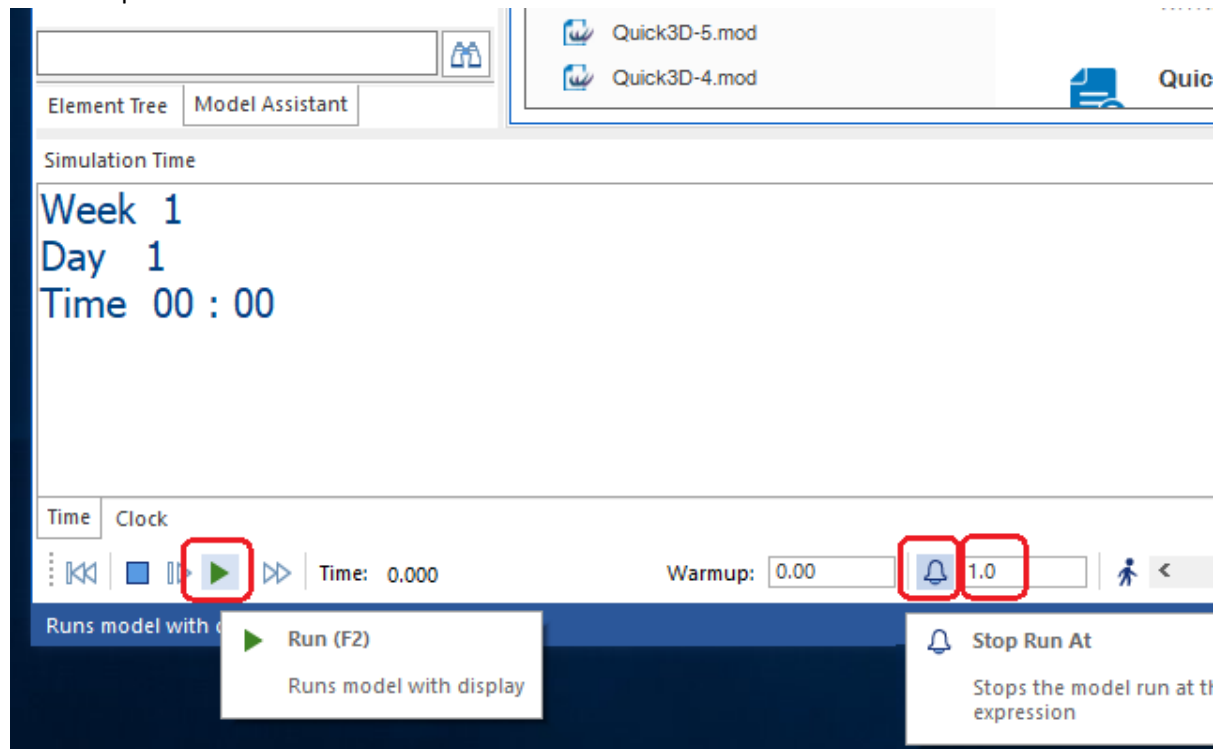
1. The “3D Command Reference 21.docx” that details the entire set of commands
2. “W3DServer.real.xml” – a sample xml command trace output captured from the Quick3D.mod
3. The “Witness W3D Queue Implementation.docx” gives a detail description of how Part queuing is implemented

5. Running WITNESS

5.1. Running the model until a time

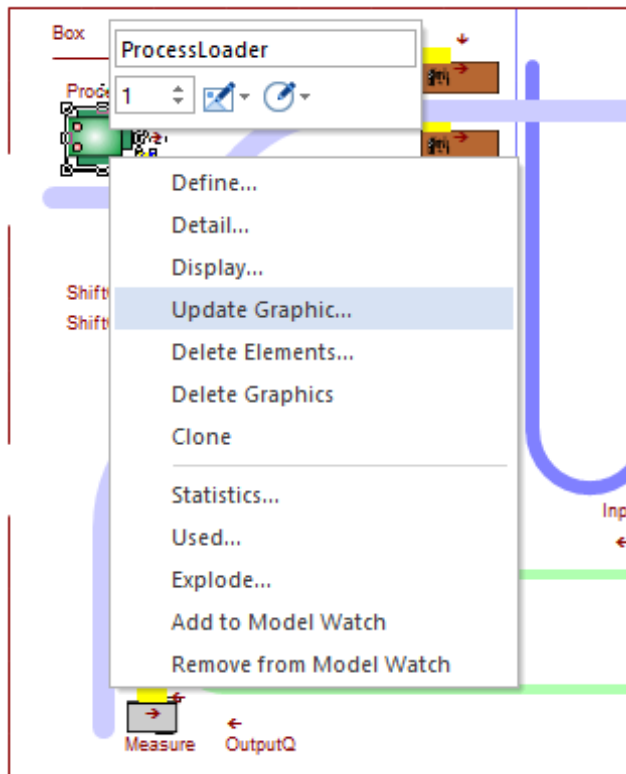
- Load the model using File | Open command
- Using the Run toolbar (lower left of the screen), enter a time to run until
- Ensure that the “Stop At Run” command is selected
- Use the “Run” command to run the model until the specified time

For example:



5.2. Changing the 3D geometry associated with a 2D shape

- Right click in a 2D shape or path and select the “Update Graphic...” command from the context menu



- Use the “3D” command and select a different geometry from the list

