



DV21222

## Interactive Simulation Workflows

Jose Elizardo  
Autodesk

### Learning Objectives

- Learn how to import real-world content into 3ds Max
- Learn how to create simulations using Autodesk CFD
- Learn how to interpret and visualize simulation data
- Learn how to create interactive experiences using Stingray

### Description

3ds Max software has a unique ability in that it can consume and output more data than any other application within the Autodesk, Inc., portfolio. In this class, we will explore how we can use 3ds Max software as a design hub. We will import real-world content from applications such as InfraWorks software and others. Then we will create physically accurate wind simulations using Autodesk CFD software (formerly known as Simulation CFD software). We will send these simulations to 3ds Max software for visualization purposes. Lastly, we will send everything to Stingray game engine to create an interactive and immersive experience. This session features 3ds Max and Autodesk Stingray.

### Your AU Expert(s)

Jose Manuel Elizardo is a 3ds Max software technical specialist for Media & Entertainment at Autodesk, Inc. With over 12 years of product experience, Jose Elizardo focuses mainly on promoting the value of visualization throughout various industries.



## 1.0 Introduction

---

Autodesk offers world class solutions that can be used to achieve the most elaborate and sophisticated designs in each of the industries it serves. But, what if we combined those solutions and blurred the industry lines? What if 3ds Max was not just an entertainment product? What if Stingray was more than a game engine? What if Simulation CFD's usage could be extended? What if Infraworks could be added to the mix? What could one hope to achieve in exploring these workflows? The following class handout goes down that rabbit hole and holistically demystifies much of the Autodesk portfolio.

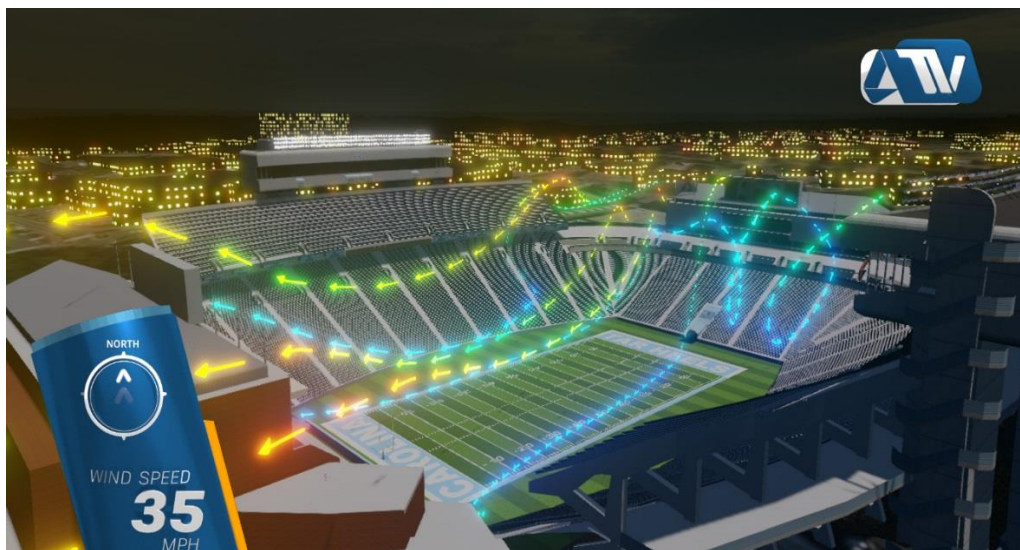
3ds Max can natively consume SKP files. The great thing about SKP files is that they are geo-located. Meaning, when imported into 3ds Max, they are correctly aligned in 3d space, as their real location in the world. So if for example, one would model the Kenan Memorial Stadium in Sketchup, once imported into 3ds Max, that model would be correctly oriented. And orientation becomes a critical factor if say, we wanted to perform a wind study of the stadium. See where we're going with this?

Simulation CFD is a powerful package used to generate many different kinds of simulation studies. The latest release of 3ds Max offers functionality that enables users to interpret and visualize this CFD data (Computational Fluid Dynamics) in ways that were never before possible. Once inside 3ds Max, we have an arsenal of tools available to "tell that story".

Infraworks is another great tool that can be used, among other things, to quickly and easily generate surrounding landscape based on GIS data.

The Stingray game engine offers tight interop integration with 3ds Max. Sending models back and forth between both products can be done with the press of a button. Furthermore, 3ds Max and Stingray share the same shader library: shader FX. So one could create shader networks in 3ds Max and have those interpreted one-to-one inside Stingray.

Lastly, 3ds Max can consume and output more file formats than any other application within the Autodesk portfolio. Leveraging this and the industry leading file format FBX, transferring data between all these applications becomes not only a possibility, but the cornerstone of this entire workflow.







## 2.0 Wind Simulation

---

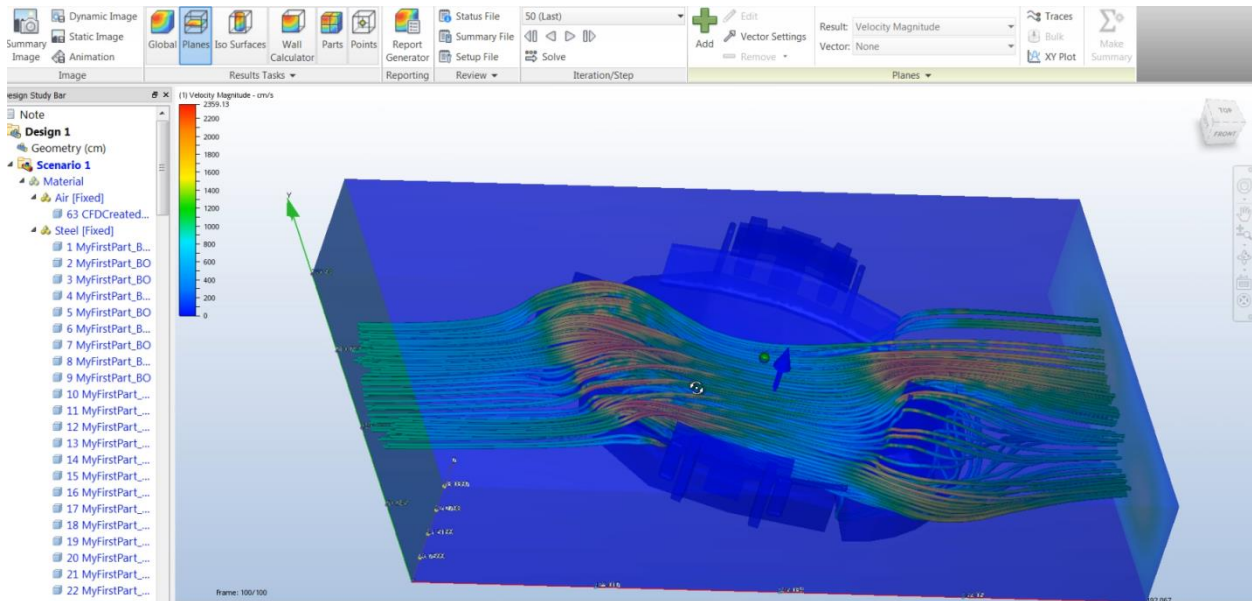
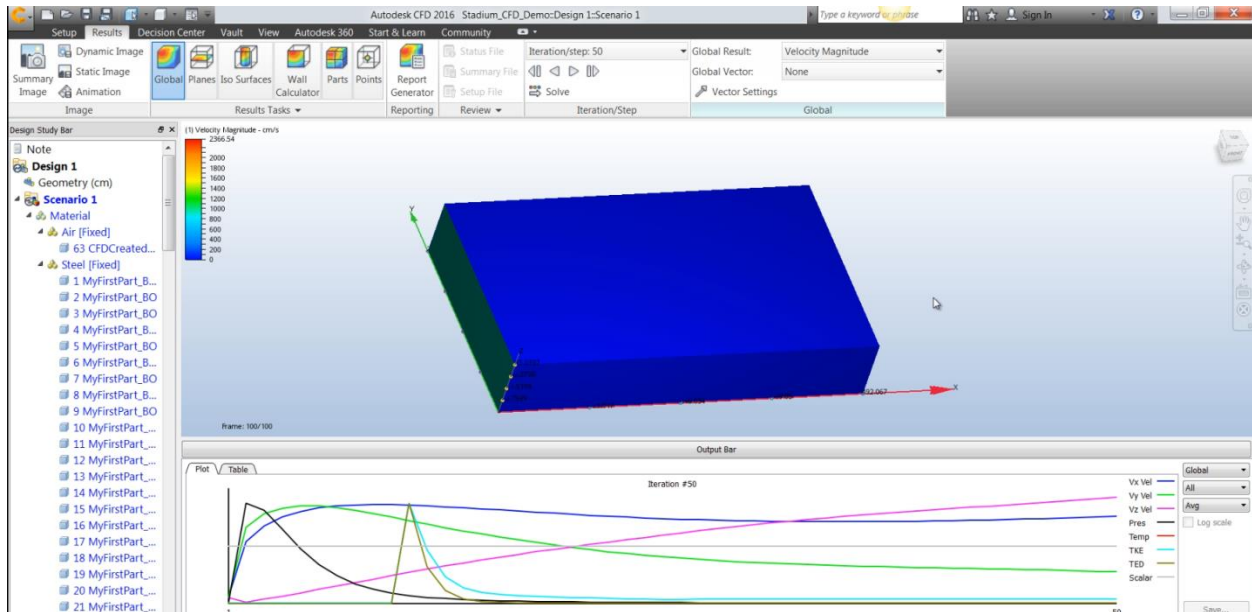
3ds Max has a unique ability in that it can import and interpret data from almost any 3d software package and maintain any real world information that may come along. After doing some research online, we found a nice model of the Kenan Memorial Stadium in North Carolina on Turbosquid (modeled by Trevor Whitlock). This model was generated in Sketchup. Once imported, we maintain the correct world orientation. See image below comparing the model once inside 3ds Max to its equivalent representation in Google Maps.

Also, notice the compass in 3ds Max that is generated automatically on import shows the correct North, South, East and West direction. This will be a critical factor when wanting to generate accurate wind simulations later on in the workflow.



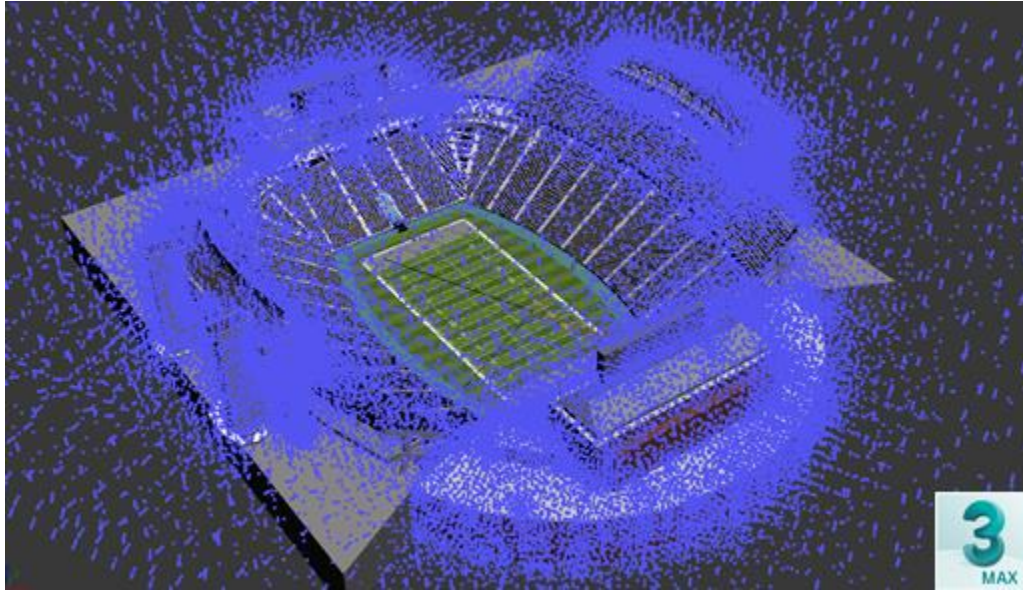


To generate a wind simulation of this model we can leverage Autodesk CFD. However, before doing so, we needed to create a proxy (simplified) version of this model. In its current state, it is a little too complex geometry-wise to be handled correctly in CFD. 3ds Max has powerful modeling tools that make this task a breeze. Once generated, we sent this proxy model to Autodesk CFD in order to perform various wind simulation studies. The idea was to have a wind simulation coming from all 8 possible wind directions. Simulation CFD makes this process very accessible, given the complex nature of such a study.





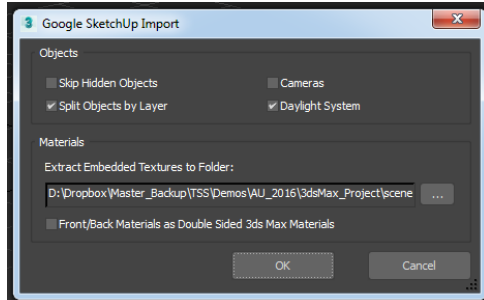
Once the study complete, what can we do with this data? In 3ds Max 2016, we introduced a revolutionary new platform called the Max Creation Graph (MCG). This platform allows users to extend the capabilities of 3ds Max via a node based, user friendly visual scripting environment. Since then, we've integrated tools into 3ds Max which leverage MCG. One of those tools, found in 3ds Max 2017, allows users to consume and accurately interpret CFD data from Simulation CFD.





## 2.1 Workflow - Importing SketchUp File

- Accessing the SKP importer can be done by clicking on the 3ds Max Application button>Import and selecting Google SketchUp from the File of Type dropdown. The following UI appears:





- In order to have a tidy 3ds Max project, enable Split by Layers. This will separate out the various parts of the model into separate layers. Also, enable daylight System; this will create the compass which will become critical later on when we generate wind simulations in Autodesk CFD.

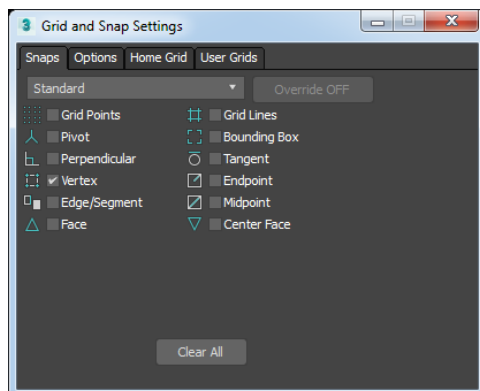
## 2.2 Workflow - Modeling a Proxy Stadium

Once we have our stadium model imported, the next step in the process is to model a low resolution, or proxy version of the model. This is because, in its current form, the model is too complex to be used in a CFD computation. 3ds Max offers a wealth of tools, ranging from easy to more sophisticated, that make this process a breeze. The next section outlines just a few of these techniques.

- First we will leverage 2d spline drawing and snapping to quickly re-create the bleachers.

- Orient view accordingly, in the modify panel select Shapes > Line , to enter Spline creation mode.

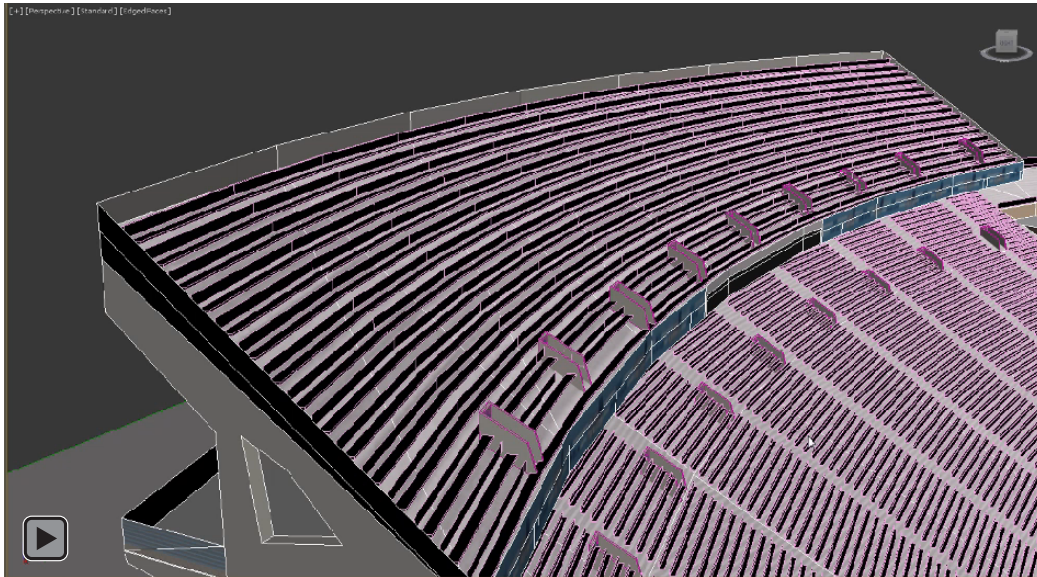
- Next, in the main toolbar, enable Snaps  by left clicking then right click on the Snaps button to access its options. Under the Snaps tab disable all options and enable Vertex.



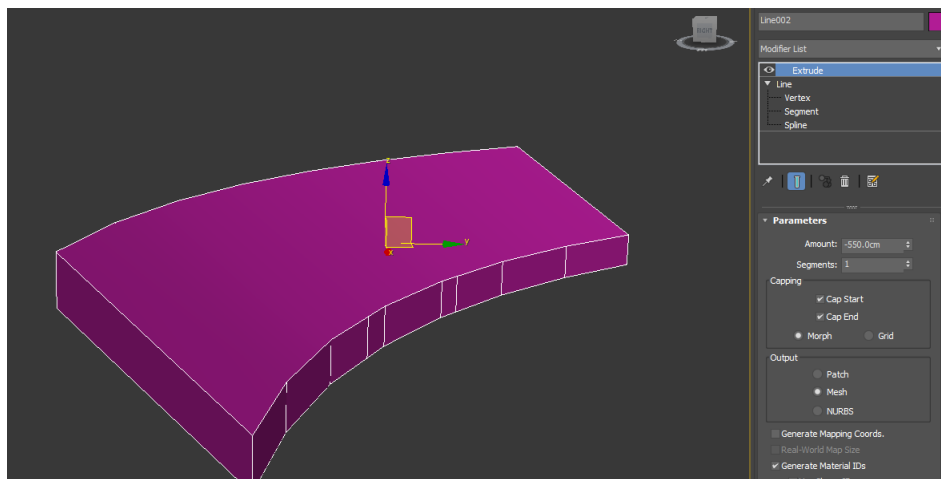





- Start drawing a spline by clicking on each vertex of the bleacher model, following along its shape. You can orbit the view while drawing and snapping.

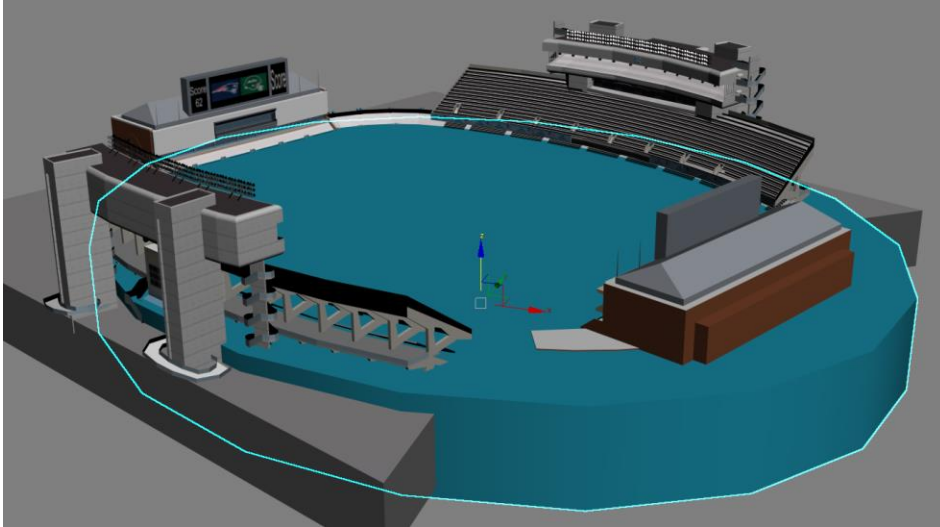


- One spline has been created and close, apply an Extrude modifier to the spline in order to give it 3d volume. Don't forget to adjust the Amount slide appropriately.

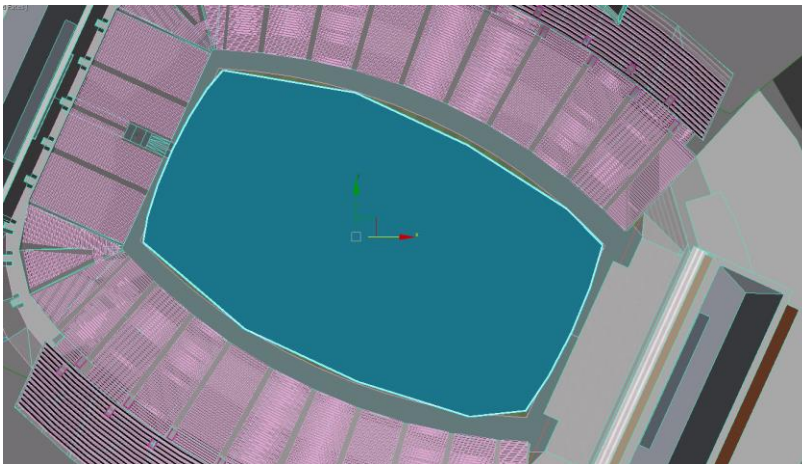


- Next, let's create the stadium itself. Create a Cylinder roughly of the same size as the actual stadium. Move, Rotate and Scale  the cylinder so it occupies roughly the same shape as the original stadium. This will act as the outer shell of the stadium.

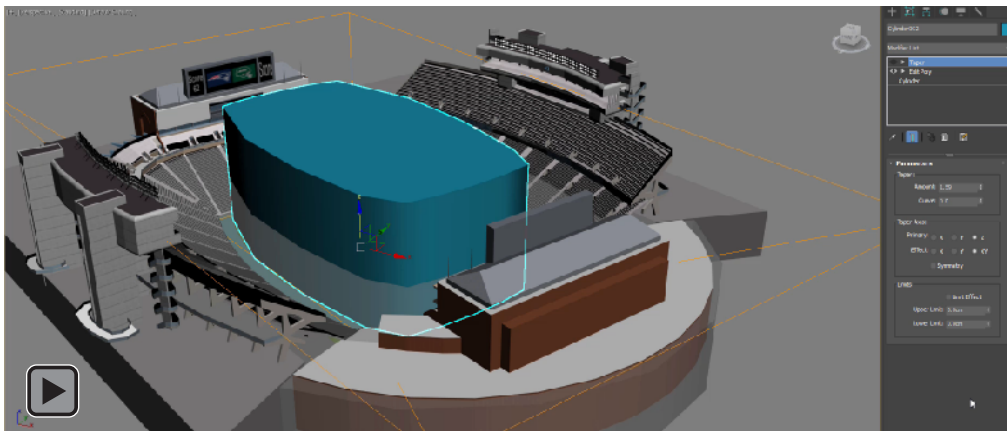




- Next, create a clone of the cylinder. Move, Rotate and Scale this new cylinder from Top view until it fills the grass area of the stadium. You may need to apply an Edit Poly modifier in order to manipulate some vertices.

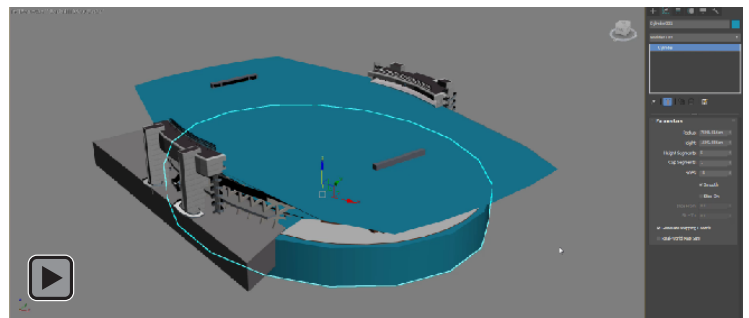
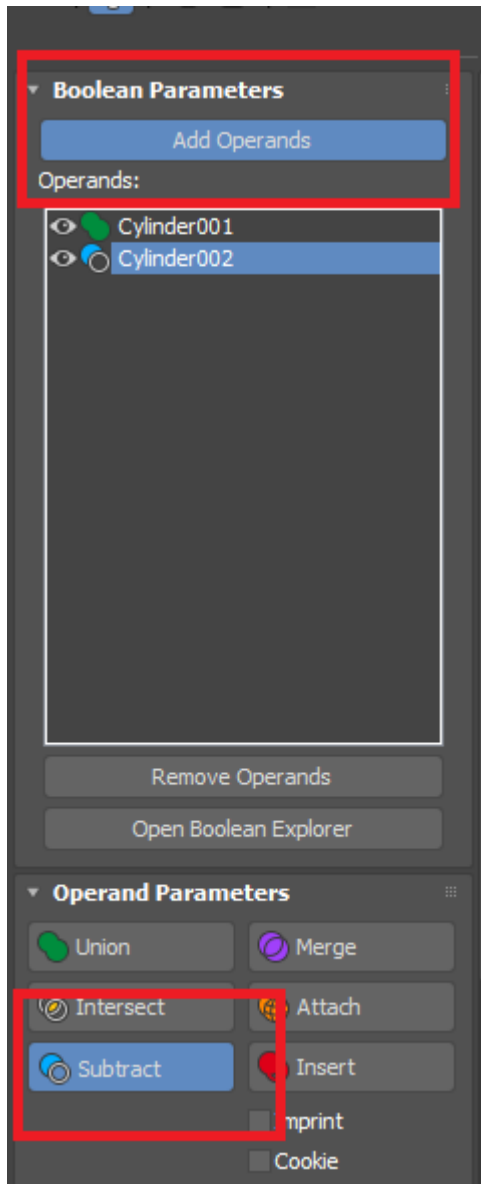


- Apply a Taper modifier to this new cylinder and adjust the settings until the shape remaining is that of the interior “empty” volume of the stadium. We will “cut” this volume from the first cylinder. What will remain will be only the stadium volume itself.





- The last step is the “cut” process, or Booleans. Select the outer cylinder, expand the Geometry tab and select Compound Objects. Next select Boolean. Click on Add Operands and select the inner cylinder. Set its operation type to Subtract.

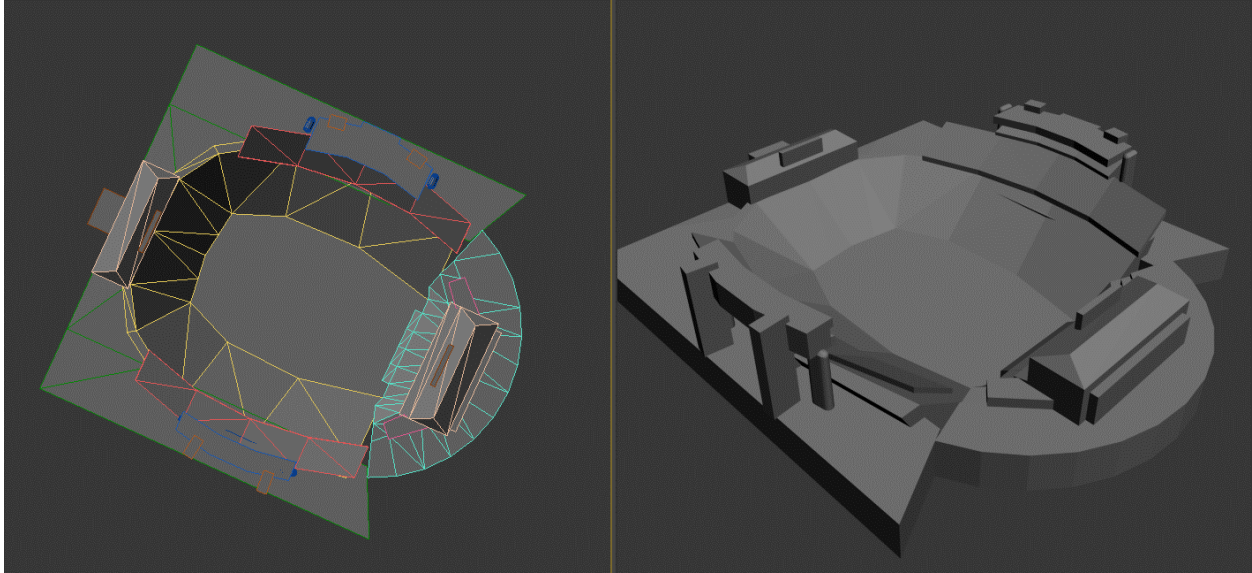




Once the majority of the stadium volumes have been re-modeled as proxy objects, the last thing we need to do is convert these volumes into Body Objects. This is so that we can export the proxy model into IGES format, for CFD computation.

- Select all the proxy volumes and in the Control Panel, expand the Geometry rollout and select Body Objects.
- Lastly, select all the volumes and go to the 3ds Max Application button and select Export Selected> ATF IGES format.

Your proxy model should resemble the image below.

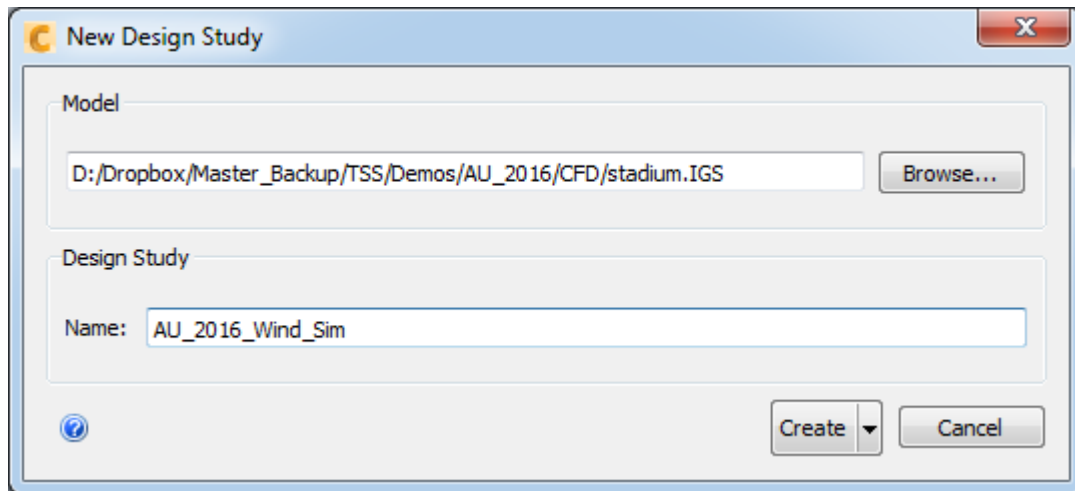




## 2.3 Workflow - Wind Simulation

To create a simulation of wind flow we will use Autodesk CFD (Computational Fluid Dynamics).

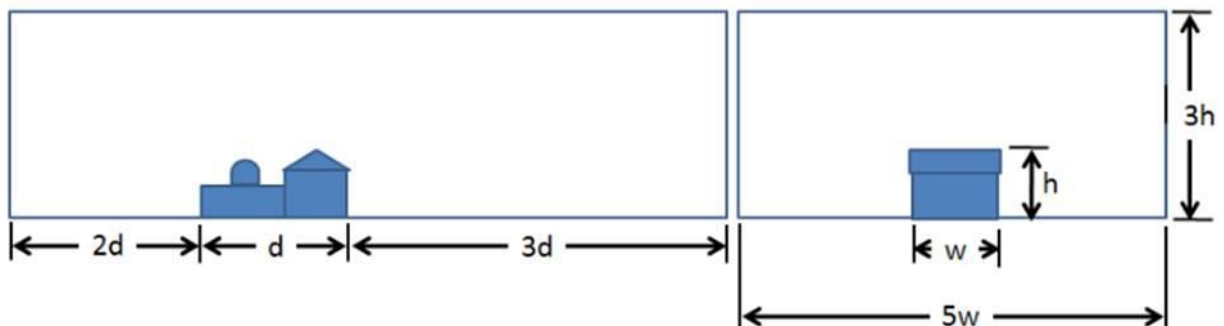
- Open Autodesk CFD and create a New Design Study.
- Under Model select the IGES file exported in the last section and give your study a name. The project folder will be placed next to the IGES file on disk.



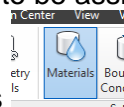
The Geometry Tool dialog appears. Here you are prompted to fix issues with the model. This is a relatively automatic process.

- Under Edge Merge select Merge with default settings.
- Under Small Object select Remove with default settings.
- Exterior Volume tab allows you to create wind tunnel. Use the handles in the viewport to adjust the size of the wind tunnel until satisfied.

The graphic below shows recommended size for wind tunnels. Don't forget to re-orient the wind tunnel depending on which direction the simulation is for (North, South, East, West). Use the view cube to help with view orientation.



- Materials now need to be assigned both to the wind tunnel and to the stadium model.

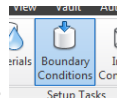


- Make sure Materials is selected in main toolbar.

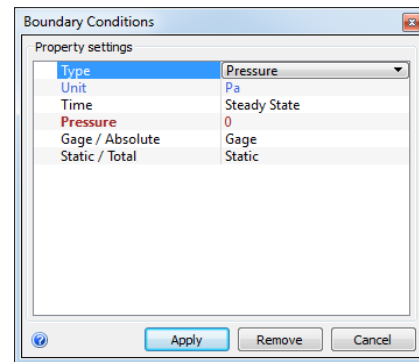
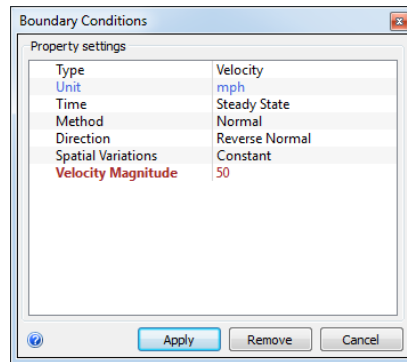


- Select the wind tunnel, right click and select Edit. Here you access the material properties. The default settings (Fluid and Air) are ok for a wind tunnel.
- To access the stadium geometry, CTRL+middle mouse button click on the wind tunnel to hide it (CTRL+middle mouse button in any empty space to bring everything back into display).
- Select all stadium volumes and in the material settings, set Type to Solid and Name to Steel.
- Bring everything back into view.

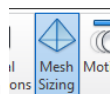
We now need to define some boundary conditions, in our case, where is the wind entering the tunnel and where is it exiting.



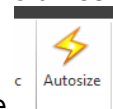
- Make sure Boundary Conditions in the main menu is selected.
- Select one extremity of the tunnel (where the wind tunnel starts – relative to the wind direction we chose to simulate) and in the Edit menu set the Unit to MPH and Velocity Magnitude to which ever wind speed you want to simulate; in our case we are simulating at 50 mph.
- Select the opposing extremity of the tunnel and set Type to Pressure and Unit to Pa.



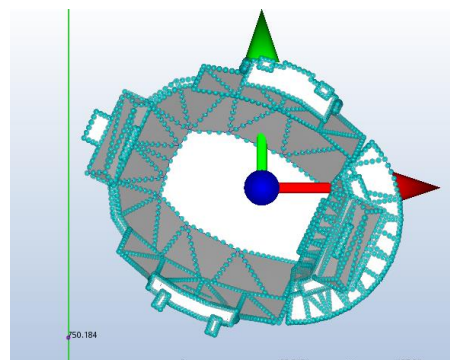
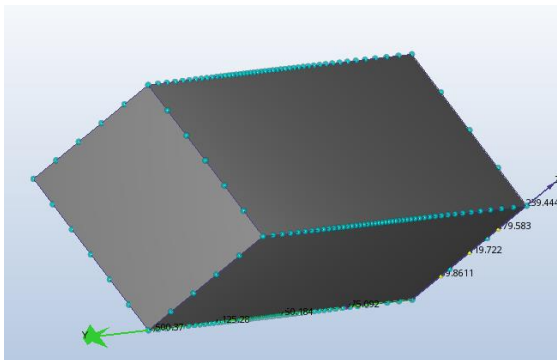
- The last step in the Simulation settings requires us to mesh these volumes.



- Select Mesh Sizing in the main menu and enable Autosize.



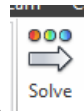
You should have results similar to these:



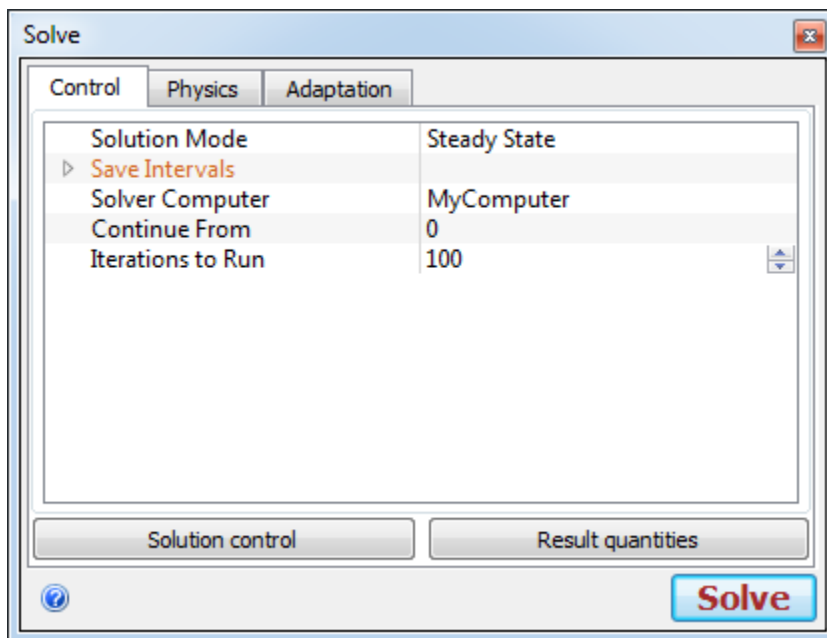




We're now ready to simulate.



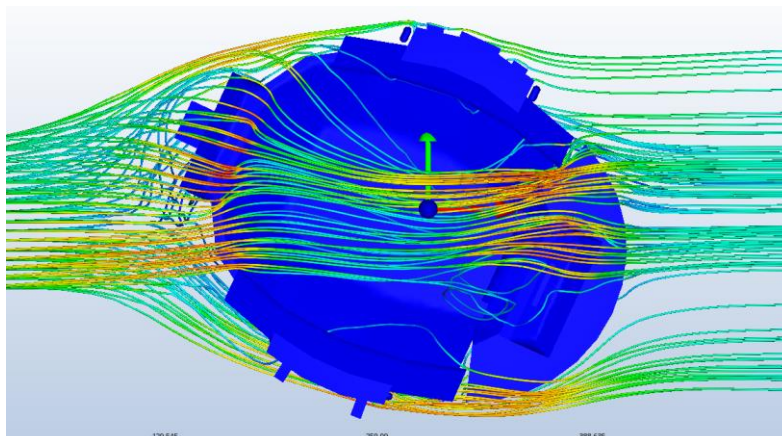
- Enable Solve in the main toolbar. The following dialog appears. There are many settings you can adjust here but as a first attempt at wind simulation, the iterations value is the most critical. For accurate results a value between 200 and 300 is usually recommended.

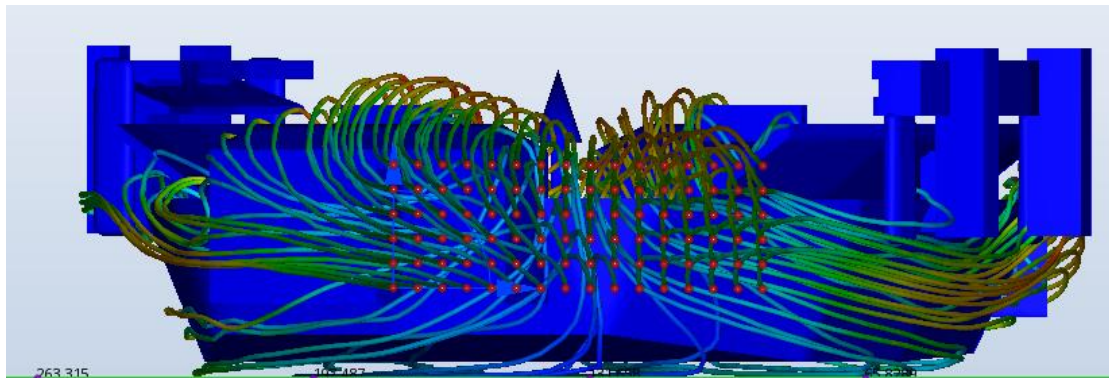
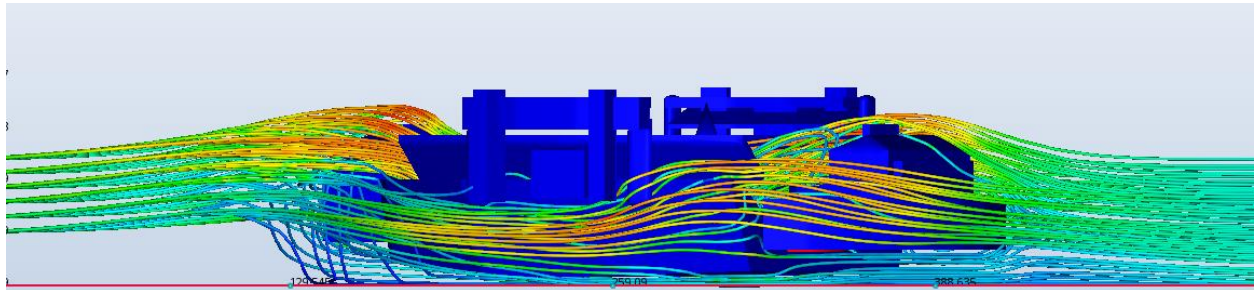


To view the simulation results, navigate to Results in the main toolbar. In here you can



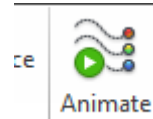
create Trace lines by selecting Traces and drawing the trace patterns based on what is selected in the Seed Type dropdown. For instance, a rectangle trace patterns would look like this:



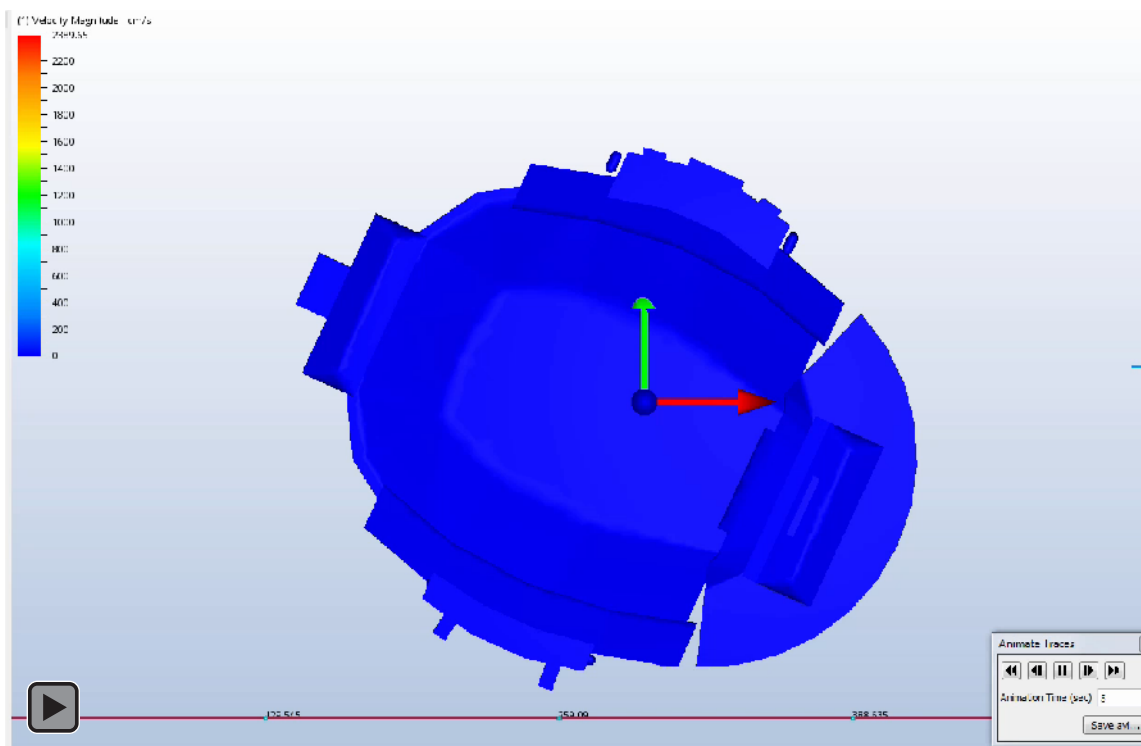


You can even animate and appearance of these trace lines as the wind flows through the

stadium. Accessed via the Animation



button on the main toolbar.





In order to bring this simulation data into 3ds Max we now need to export into a CSV format.

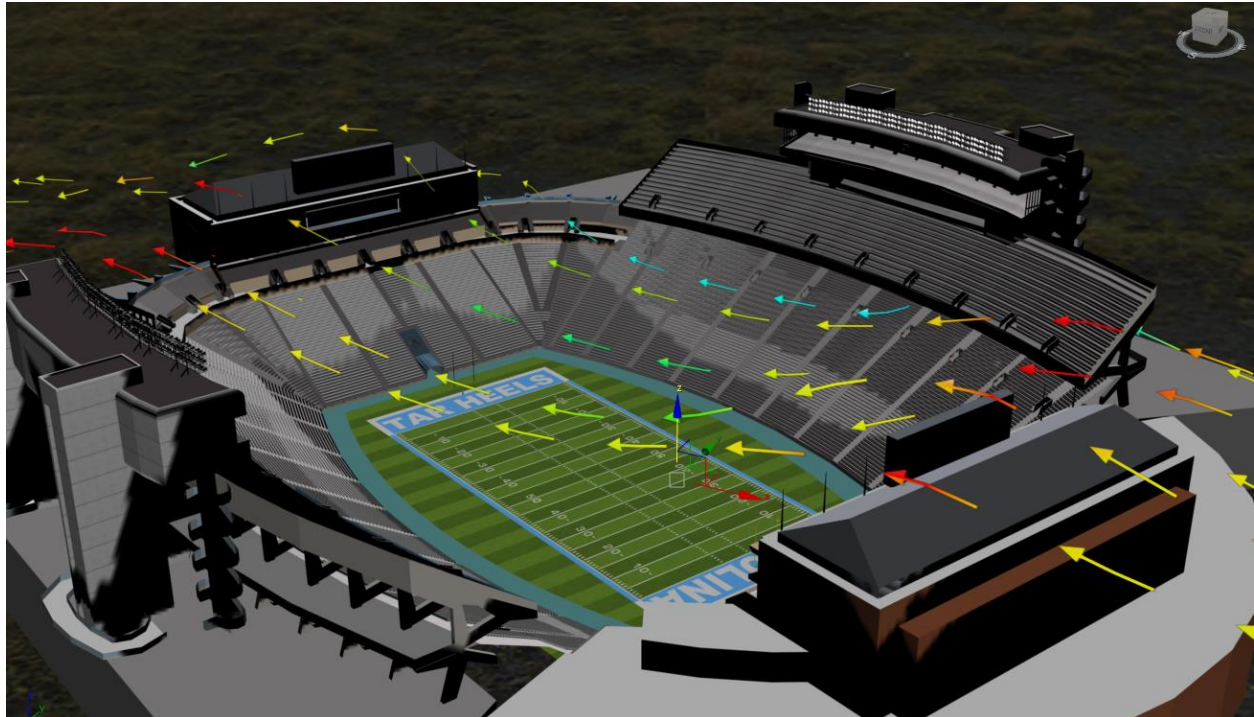
- Access the CFD Application button, Export> Export Nodal Result (this will save a CSV file on disk). There are no options to save to a specific location.
- To find the CSV, locate the project on disk and the following file within in: ***[project name]\ Design 1\Scenario 1\solver\Scenario 1\_nodal.csv***



### 3.0 Interpreting CFD data

---

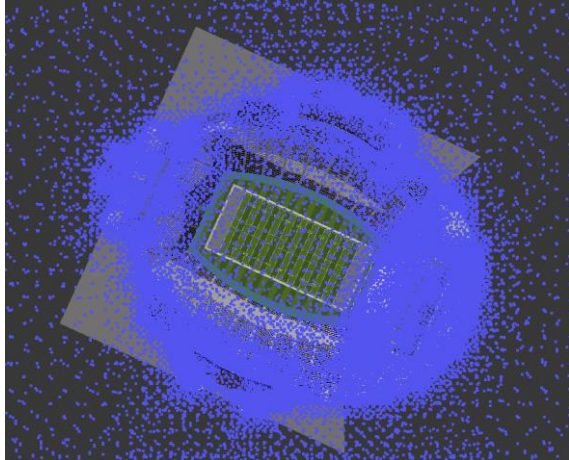
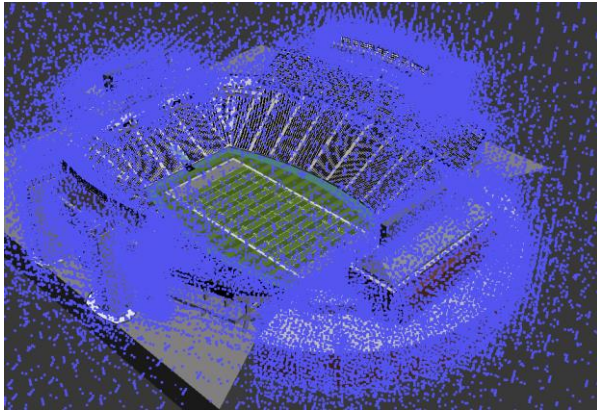
Next we needed to determine and assign a visual style to this data. Furthermore, we want to ultimately send this data to the Stingray game engine in order to create an engaging and interactive experience. To accomplish this, we will leverage the Shader FX platform found in both 3ds Max and Stingray. Shader FX essentially allows users to create physically based game shaders (PBS) without having to write a single line of code. This is accomplished through a user friendly, node based visual editor where users connect nodes together in order to create given shaders. In order to create a visually understandable representation of the CFD data, we will generate animated shaders of arrows flowing through the stadium. These arrows can also be shaded from red to blue where red represents higher velocity and blue lower velocity.



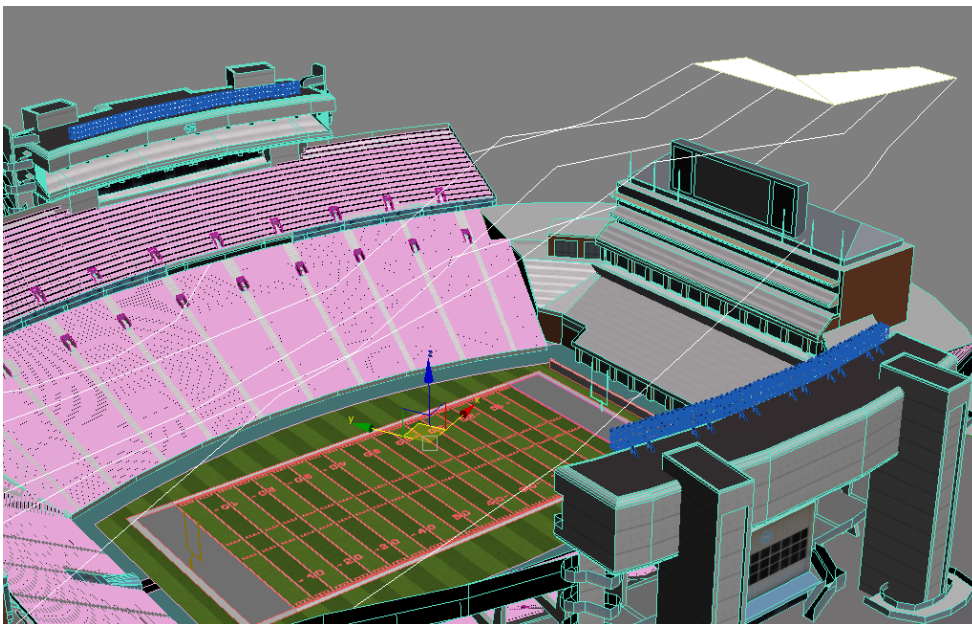
#### 3.1 Workflow - Importing CFD data

- In 3ds Max modify panel, navigate to Geometry> CFD dropdown and create a CFDImportData object.
- Add it to the scene and center it to 0 0 0.
- In its properties in the modify panel, connect the CSV file exported in the previous section. This takes a few seconds to load.
- Once loaded, right click to access the Object Properties dialog. In here, make sure to enable Vertex Ticks.
- You will see a blue point cloud, otherwise known as a velocity field, surrounding the stadium. This velocity field contains all the simulated data; in our case wind direction and wind velocity.





- Once imported, the velocity field can be hidden in order to remove visual clutter.
- Next, in the Shapes> CFD dropdown, create a CFDSplinePath object in the scene, center to 0 0 0. This object will allow us to create splines which will traverse the velocity field and accurately represent wind flow direction. This object requires us to first connect it to the velocity field. Go ahead and do that.
- Then, we need to assign a geometric object to it. This geometric object will generate a spline for every vertex found on the geometry. In my case, I created an organic looking plane object with 6 vertices.
- Assign your geometry to the Path Source button found on the CFDSplinePath object.



The default settings do not usually give satisfactory results. The CFDSplinePath object has settings in order to create nicer looking, more organic splines. Adjust these settings depending on your project. Here is a brief description of each parameter:





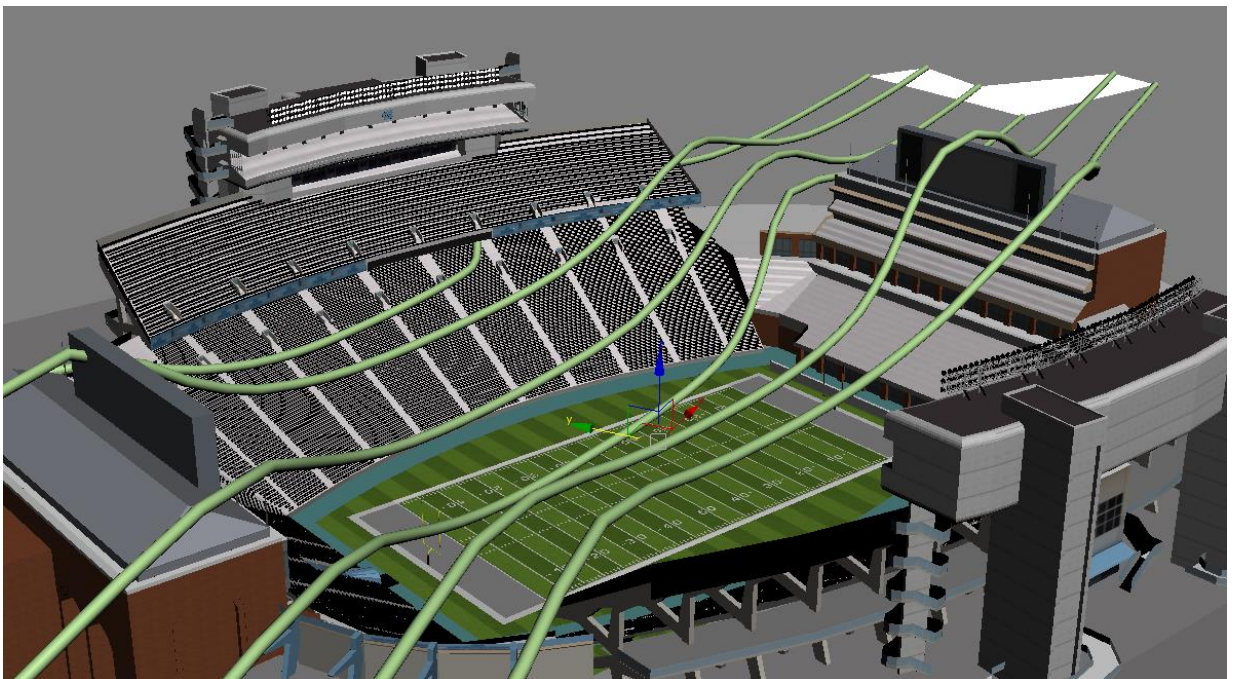
**Num Steps** is the total number of segments your splines will have. You can also imagine it as number of steps the vertices take as they walk through the velocity field.

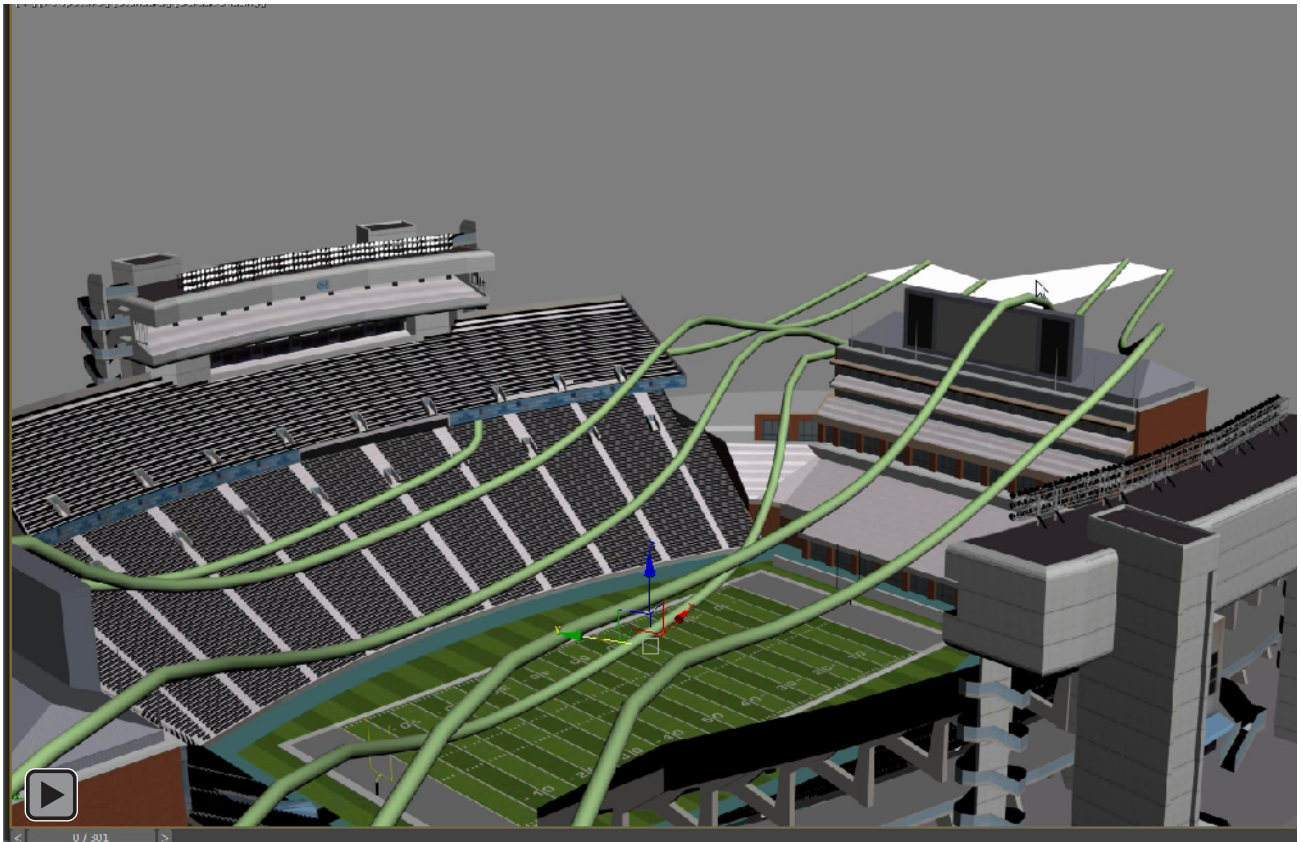
**Num Samples** is how many samples to look at to determine my next velocity. If you increase it, you'll consider more samples at each step, so you'll probably be going faster in your walk through the velocity field.

**Step Size** acts as a multiplier, and is responsible for the resolution of your splines. A lower step size means you'll be walking through in smaller increments through the velocity field so your splines will look smoother. If you increase it, you might traverse the velocity field faster, but your splines will look jagged.

There are lots of things you can modify now to adjust the shape of the splines themselves.

- First, give it volume by enabling spline rendering in the viewport found in the CFDSplinePaths object's rendering rollout. You can set this to either radial or rectangle and adjust the shape size directly in here.
- Make sure enable Generate Mapping Coords; this will allow us to add animated textures later on.
- You can modify both the position of the generating geometry and its actual sub object shape. Doing so will continuously update the spline objects to accurately reflect their new position within the velocity field.
- You can also experiment by adding modifiers such as Sweep to create custom shapes.
- Try further tessellating the generating geometry but adding a Tessellate modifier, this will create additional splines for each additional vertex.







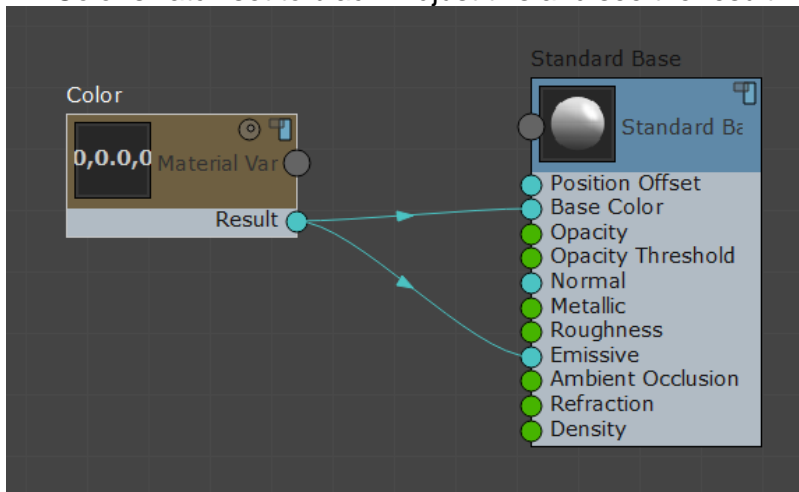
### 3.2 Workflow - Applying A Visual Style

Next we need to assign a material to these splines in order to have a visually compelling and pleasing effect. Since this will all eventually go into Stingray, let's use shader FX to achieve this. In the next section we will create a custom Stingray shader used to animate textures along splines.

- Open the material editor and create a new DirectX Shader. Assign it to the spline object.
- Make sure to set the shader dropdown to Stingray. This shader platform allows us to build and customize the UI of the shader using a friendly node graph. The UI that you see here is a default graph. Let's go ahead and modify this in order to include the controls that we need.

Open ShaderFX

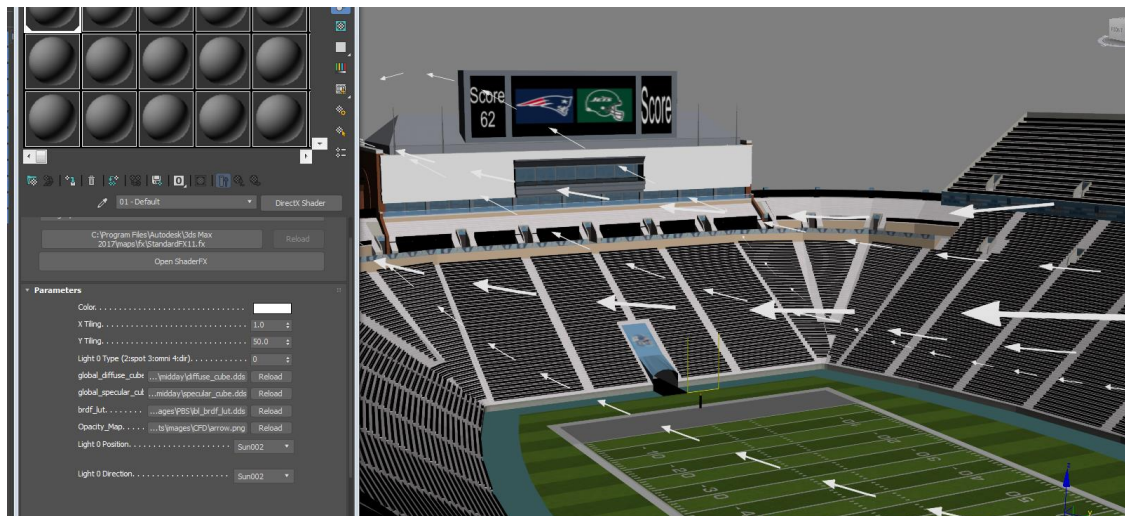
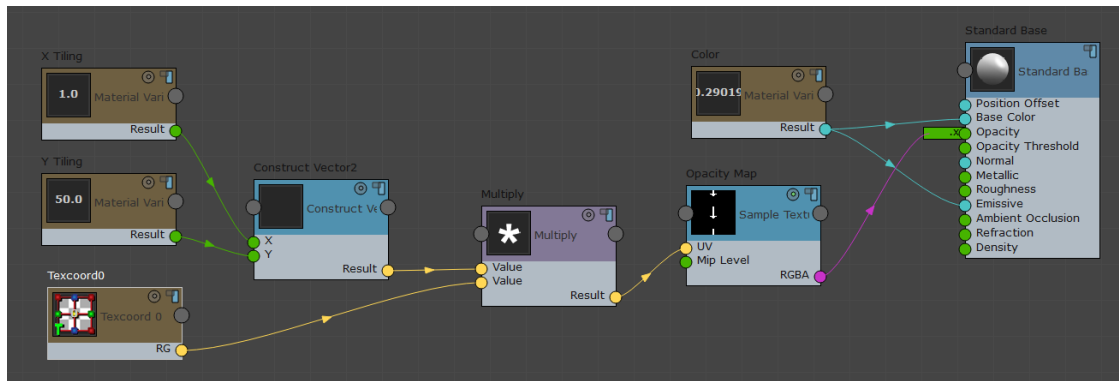
- Open the Shader FX graph. Select the Standard base node in the right panel, press on Make Unique. This will disconnect this shader from its parent shader which it is currently locked to. This will essentially create a new parent material and allow us to modify the graph. In here we're going to delete everything except for the Standard base node and the textcoord0 node.
- Let's give the shader some diffuse and emissivity. Add a Material Variable node. Rename it to Color. Set its Type to Vector 3 (color swatch). Connect its output to both Base Color and Emissive on Standard Base. Notice in the material editor we now have a Color swatch set to black. Adjust this and see the result in the viewport.



- Next we want to add opacity map and tiling controls to our shader. Let's go ahead and build that out. Add a Texture Sample node and rename it to Opacity\_Map. Connect its RGBA output into Standard Base Opacity.
- Add a Construct Vector 2 node.
- Add a Material Variable node and rename it to X\_Tiling. Set its type to Scalar (spinner) and its UI Slider Max to 100.
- Copy and paste this node into the graph to create a copy. Rename it to Y\_Tiling.
- Connect both these nodes into each input of the Construct Vector 2 node.
- Next we need to multiply this result by that of the objects UV.

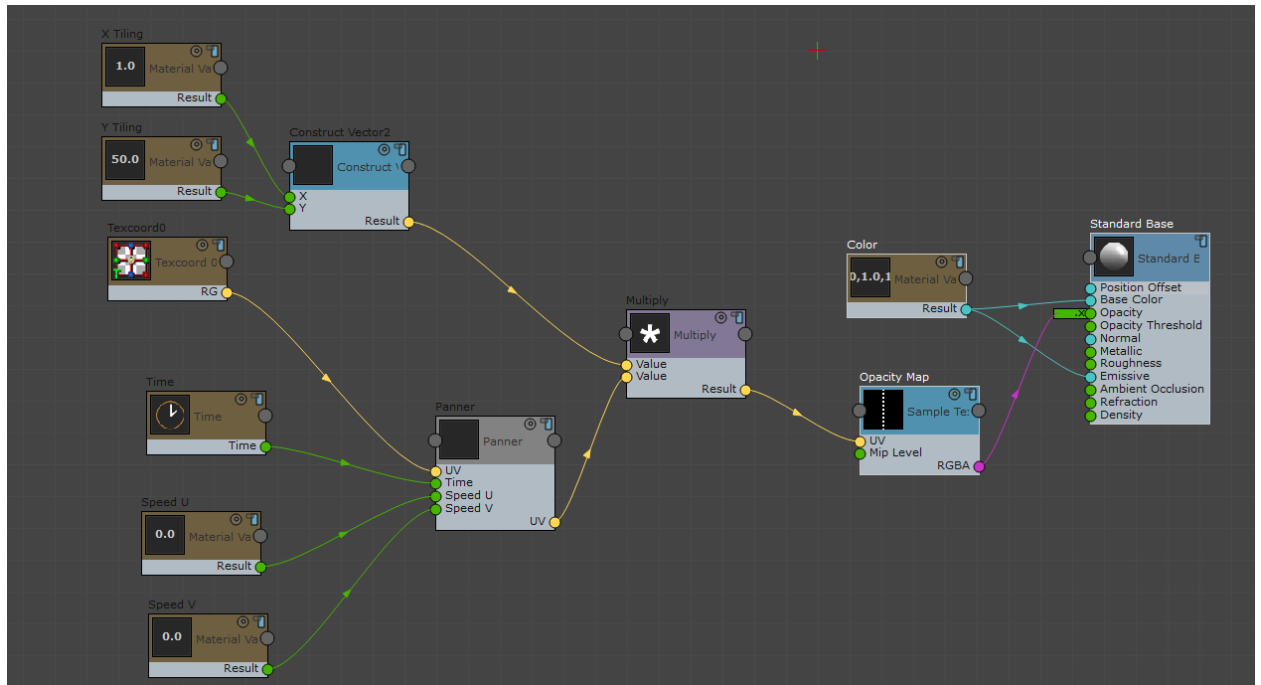


- Add a Multiply node. In one Value input connect the out of Construct Vector 2. In the other Value input connect TextCoord0 output.
- Connect the Multiply node's output into the UV input of the Opacity node we added earlier.
- In the material editor a black and white map to the Opacity map (I used an arrow texture). Your graph and result should look like this:

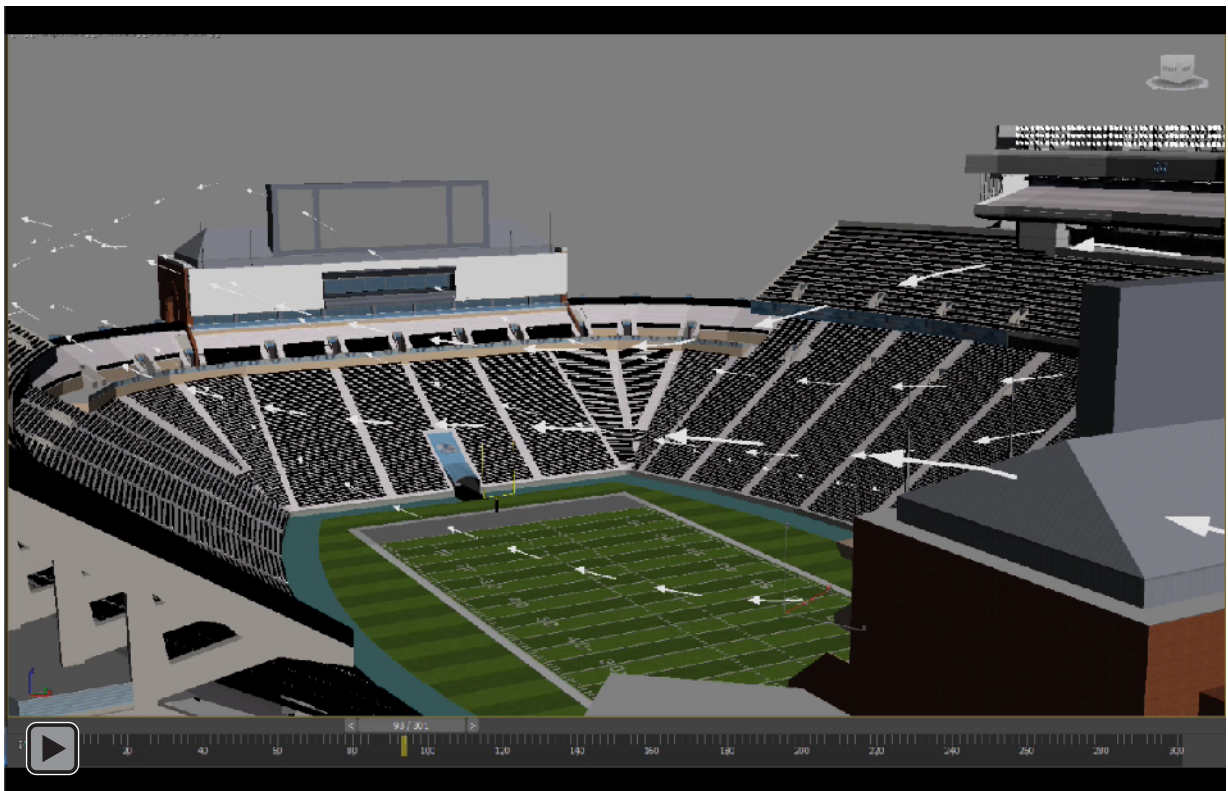


- Say we want to animate these arrows? We can easily achieve this by adding a few more nodes to our graph. Add a Panner and Time node to the graph.
- Connect Time to Panner time input.
- Connect Textcoord0 output to Panner UV input.
- Add a Material Variable node, rename it to Speed\_U and set its Type to Scalar (spinner). Clone it and rename to Speed\_V.
- Connect both these nodes into the corresponding Panner Speed U and V inputs.
- Finally, connect the Panner output into the Multiply node value input where Textcoord0 is currently connected, thus overriding that connection.
- Your graph should look like this:





In the material editor UI, you have new controls. Try setting the Speed V value to 0.05 and pressing play on the timeline.

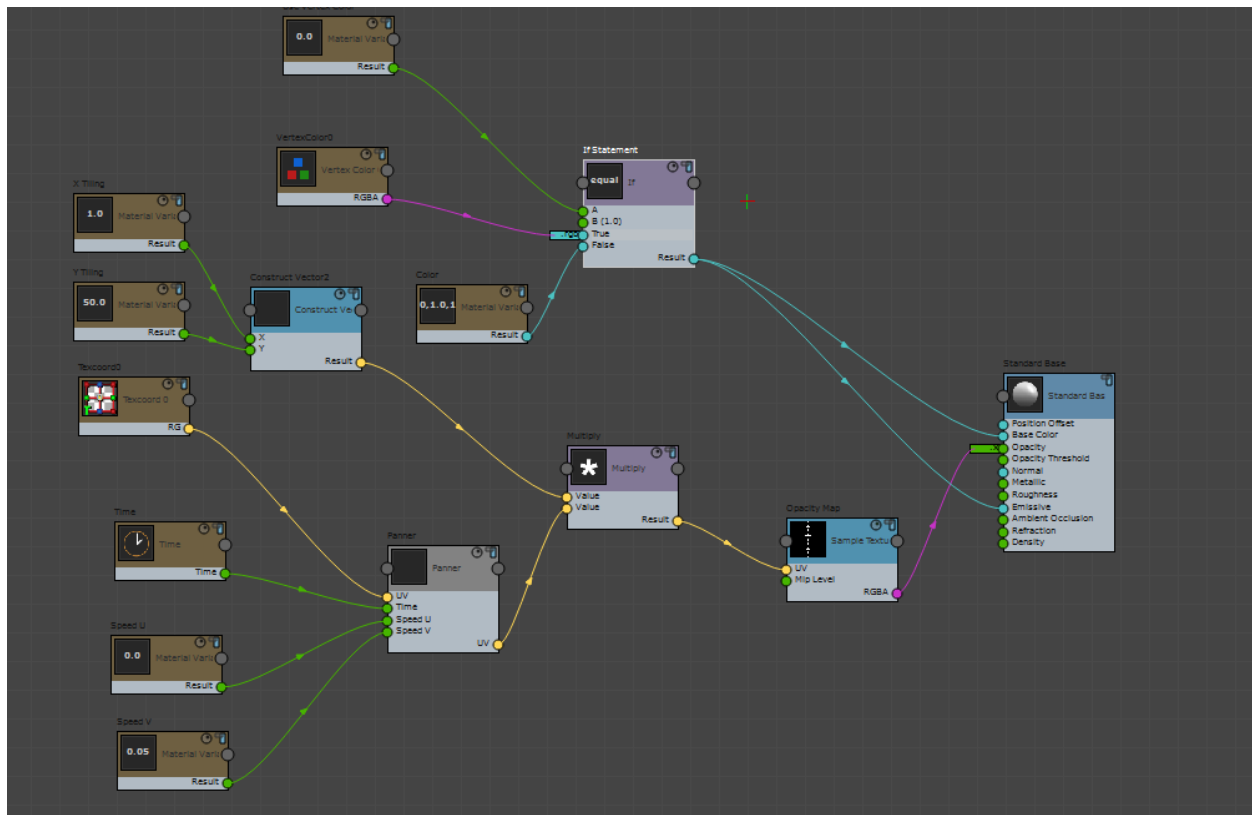




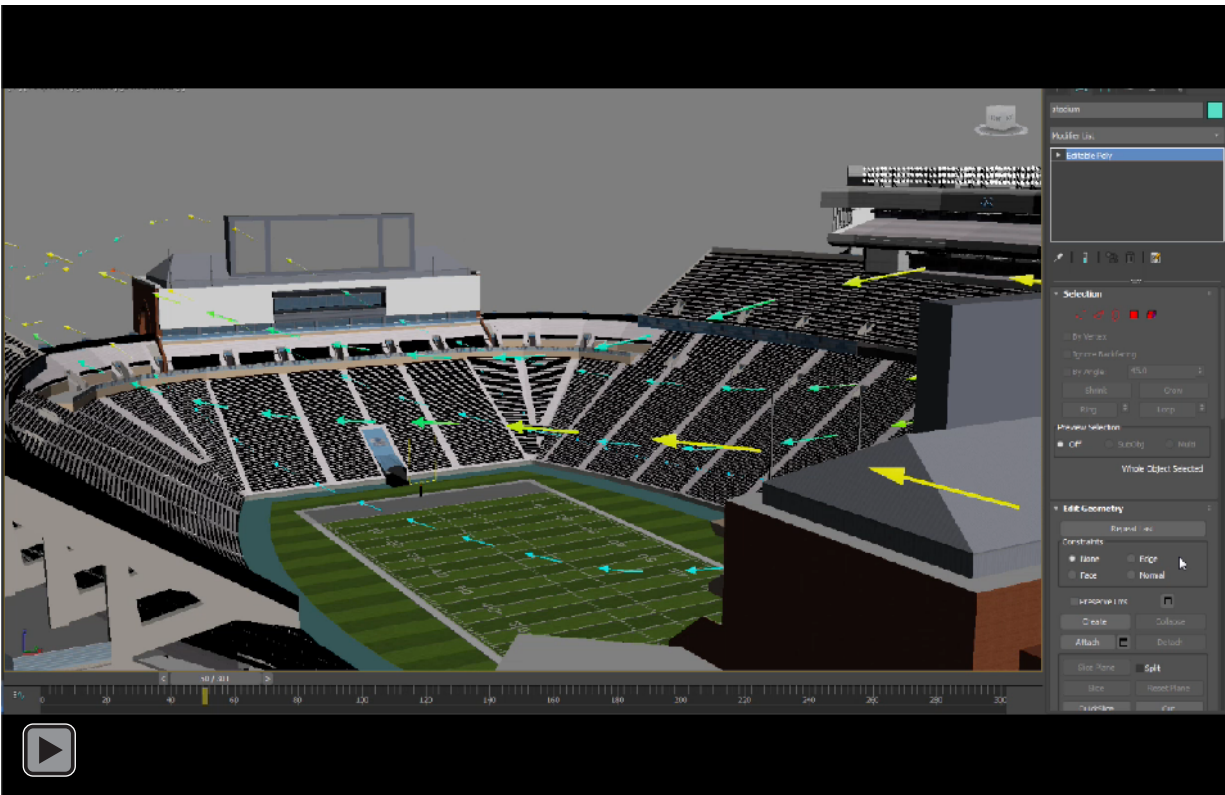


Let's now see how we can read the velocity information in the velocity field and display that as colored arrows on our splines.

- In our shader FX graph, add an IF node and a Material Variable node.
- Rename the material variable node to Use\_Vertex\_Color. Set its Type to Scalar and its UI Type to Checkbox.
- Connect its output to the A input on the IF node. Select the B input on the IF node and set its default value to 1.0 in the properties pane.
- Add a Vertex Color 0 node to the graph. Connect it to the True input on the IF node.
- Select the IF node and set its component swizzle to rgb (lower case).
- Connect the original Color node result to the False input on the IF node.
- Finally, connect the If output to both the Base Color and Emissive inputs in the Standard Base (Overriding the Color node connection).
- Your graph should look like this:



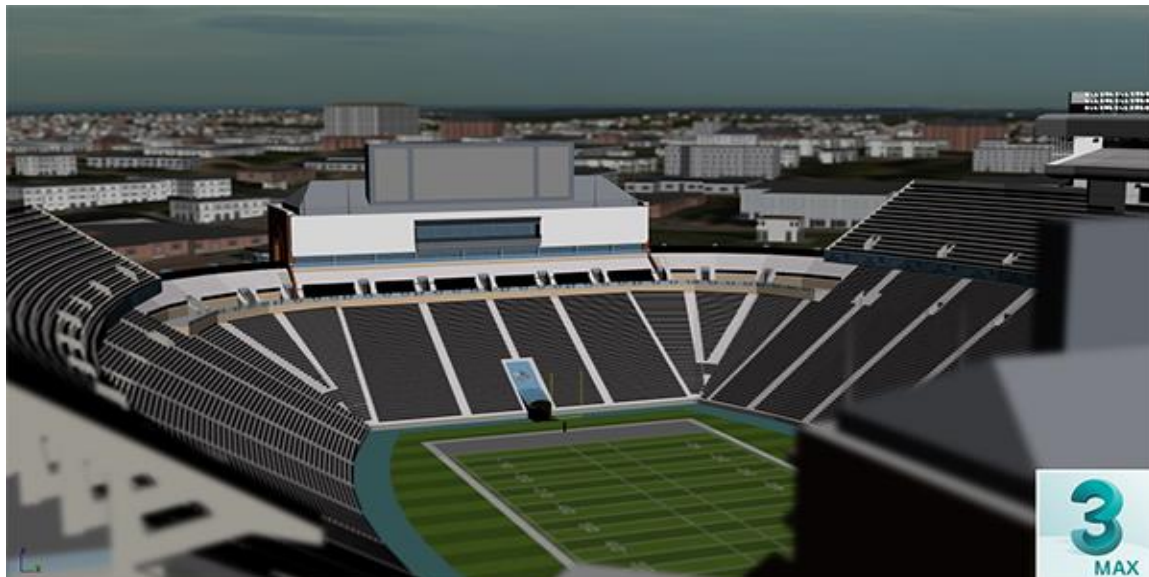
- The material UI now has a new Use Vertex Color spinner. This spinner can be seen as a checkbox. Where 0 does not use vertex colors and 1 uses vertex colors.
- Next, select CFDSplinePath object and add a CFDCOLORVerticesMod modifier to it. Assign the velocity field to it.
- Turn on vertex colors to see velocity information on the splines displayed as colored arrows. Where the arrows are bluer the wind is moving more slowly and where the arrows are more red the wind is moving faster.



## 4.0 Real-World Contextual Surrounding

---

The stadium now needs some contextual surrounding. Infraworks is an Autodesk solution that allows, amongst other things, to very quickly and accurately generate city models based on GIS data. Generating these models is as simple as literally drawing a region on an Infraworks map. Infraworks leverages the cloud to quickly generate a fully textured model of a given region. This model can then be imported into 3ds Max via FBX. We will cover this workflow in the next section.



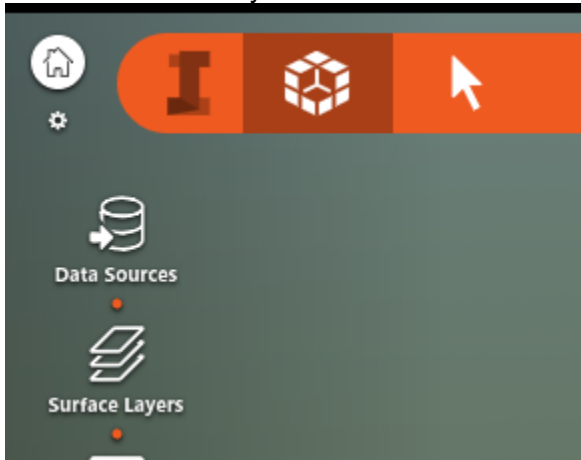


## 4.1 Workflow - Infraworks 360

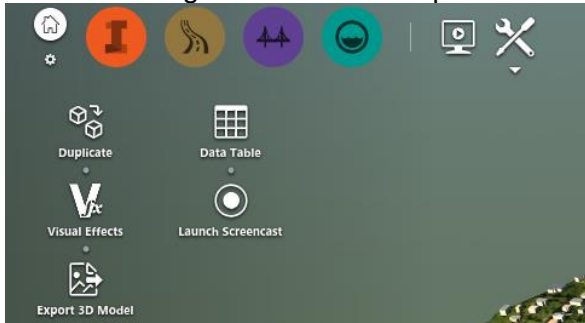
- Open Infraworks 360 and access Model Builder from the landing page. A Bing map opens allowing you draw a selected area to generate a city model of. Search for your location of choice, in my case the Kenan Memorial Stadium. Next select an area by selecting one of the selection methods



- Give it a name and press Create Model. This will be processed on the cloud and once the model is ready you will receive an email.
- Back in the landing page, the model now appears under My Models. Open it.
- Access Surface Layers and make sure roads and railways are hidden.



- Access Settings and Utilities > Export 3d model



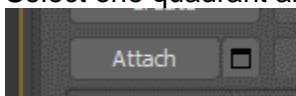
- Enable Use Entire Model
- Enable Multiple Files. This will create separate FBX files for both the ground and the buildings
- Give it a path on disk to save the files and hit Export



## 4.2 Workflow - 3ds Max Import and Clean Up

Once the model has been exported from Infracore we are going to import it into 3ds Max. The FBX files as is cannot be consumed by Stingray just yet. There are a few things that need to be fixed in Max first.

- Let's start by dealing with the ground.
- Back in 3ds Max, let's import our Ground.fbx from the section above. Once imported, notice the ground is split into various quadrants and are all named mesh 1. They are also extremely tessellated. Let's fix all this.
- First, delete the water object. Select it in the viewport and hit the delete key
- Let's optimize the model. In top view, select all of the ground objects (it helps to have isolated this geometry on import.) Apply a ProOptimizer modifier to the objects. In the modify panel, set the following parameters:
  - o Exclude Borders
  - o Keep Textures
  - o Keep UV Boundaries
- Hit Calculate. Now you have access to the Vertex % spinner. Adjust this value to something very low. This is a subjective value of optimization so adjust this until you are satisfied with the quality of the mesh versus the density. I used a value of 10% and got great results.
- Collapse to editable poly
  - Next let's attach these objects into one single entity.
  - Select one quadrant and in the modify panel, access the Attach parameters

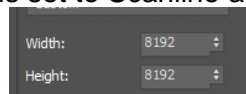


and select all quadrants from the scene explorer labelled Mesh 1, rename it to Ground.

Now we have one single mesh to worry about. We still have an issue with the textures, there's just too many and they all have "." in their names which is not supported by Stingray. Let's simplify this.



- Open the render dialog , make sure the renderer is set to Scanline and set the



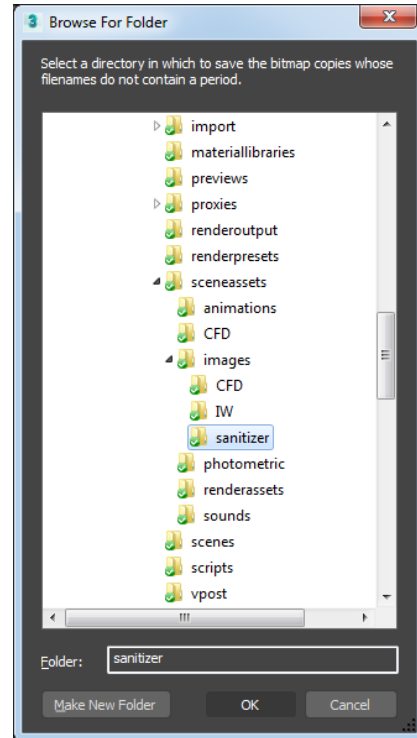
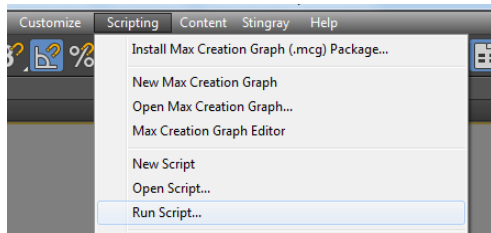
output resolution in the Common tab set to 8192 x 8192. Stingray prefers textures that are to the power of 2.

- Next go into Top view and render. Save this render to disk as a PNG file and open it in Photoshop.
- In PS, CTRL-click the layer thumbnail to select its content. Go to Image> Crop to remove the unnecessary excess in the texture save as a new image.
- Next in 3ds Max, create a new standard material and apply the newly saved texture as a diffuse map; apply to ground object.
- Apply a UV map modifier to the ground object and set Alignment to Bitmap Fit. A windows browser will appear, select the image saved from Photoshop.

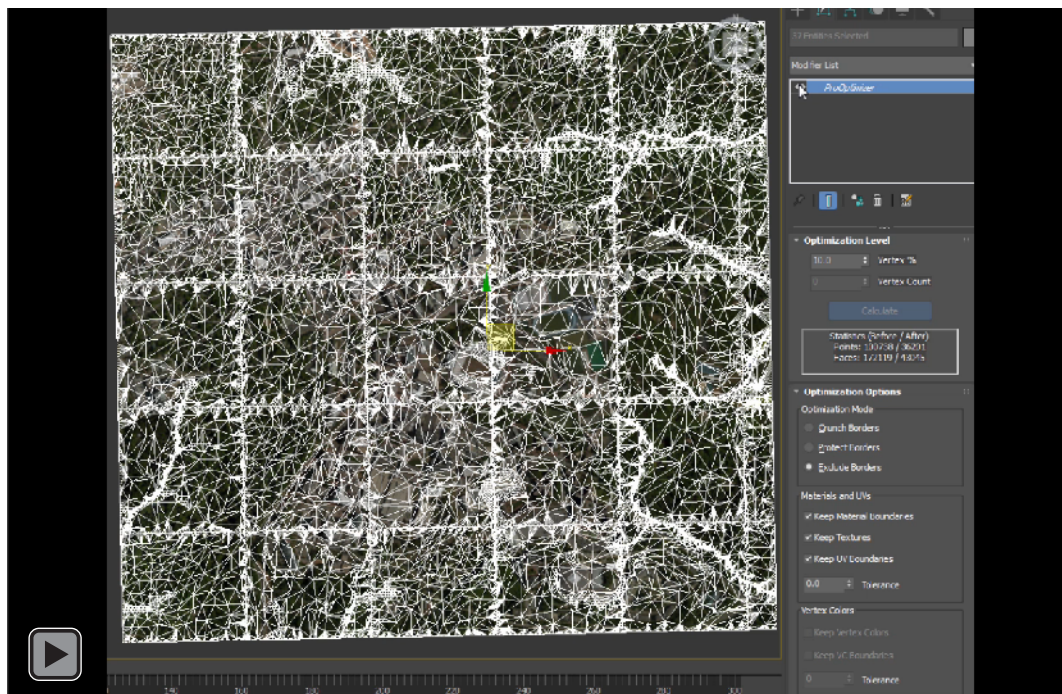


Now we'll deal with the buildings. The texture maps assigned to the buildings all have "." in their names well. This time we are going to batch rename these maps.

- Import Buildings.FBX. Go to Scripting> Run Script and run the stingray\_bitmap\_sanitizer.ms provided with this class. It will prompt you to for a location to save the renamed texture maps.



Now your Infraworks model is ready to be consumed by Stingray.

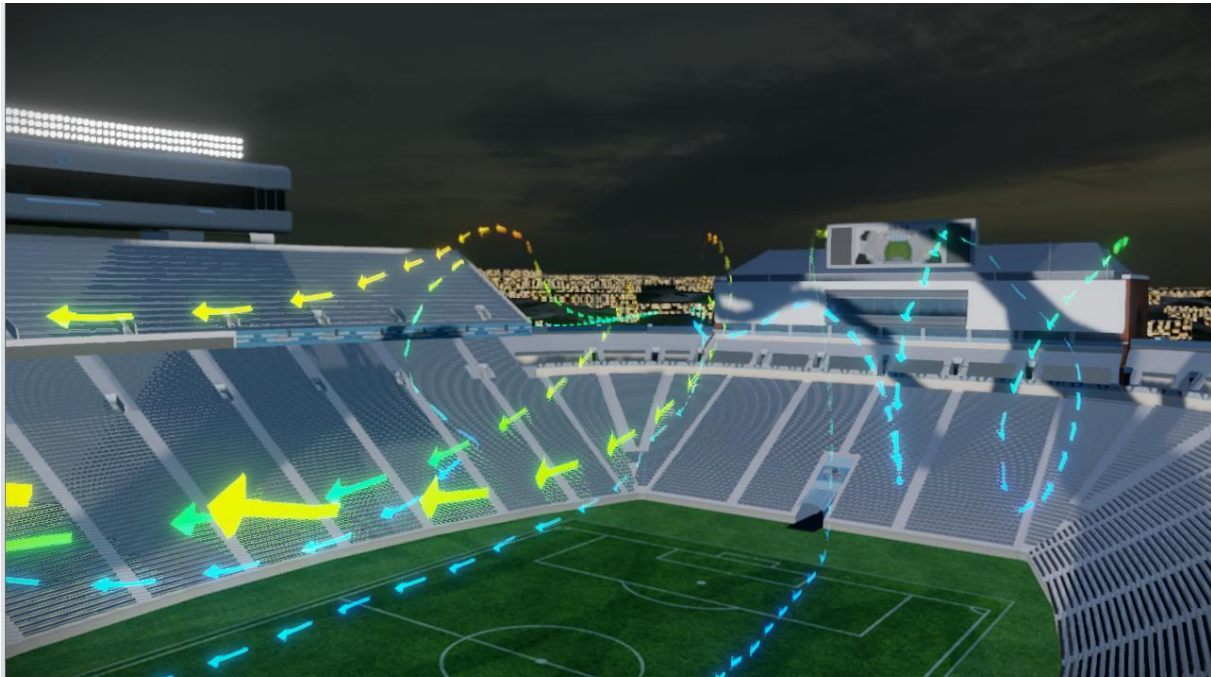




## 5.0 Real-Time Interactivity

---

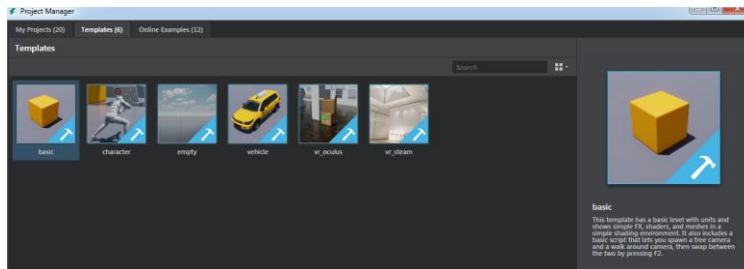
The last step in the process involved preparing and packaging all this content and data for the Stingray game engine. In 3ds Max 2017 we have one click workflows for accomplishing this. Stingray has a wonderful node based system called Flow. Flow is used to determine how things behave and interact inside the game environment. We setup triggers to perform certain actions based on pre-determined events. In our situation, we want to have the various wind simulations appear by clicking a given key on the keyboard. Once setup, we are able to be inside the stadium and experience the wind simulations as if we were there, from any given view and any given angle.



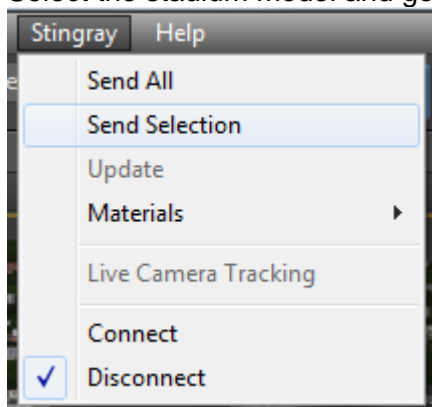


## 5.1 Workflow - Stingray Intro

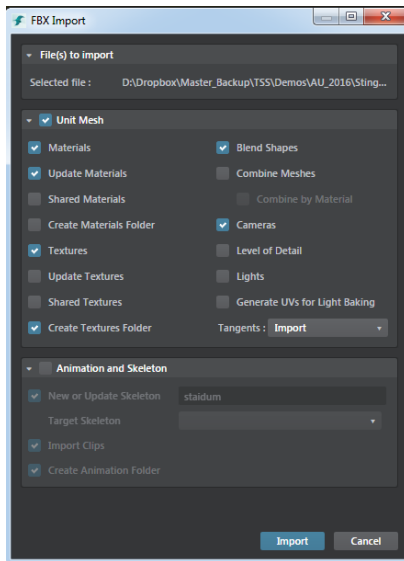
When opening Stingray, you will encounter the Project Manager window. Here you can either select a project in progress or start from a template. Templates are projects that are setup in a way that facilitate certain kinds of workflows. They have the necessary scripts already configured for the given template objective. In many cases, the Basic template is good starting point. It has both free camera and character modes available and switchable via the F2 button. Let's go ahead and start it up. Simply give it a path on Disk and a name and hit Create.



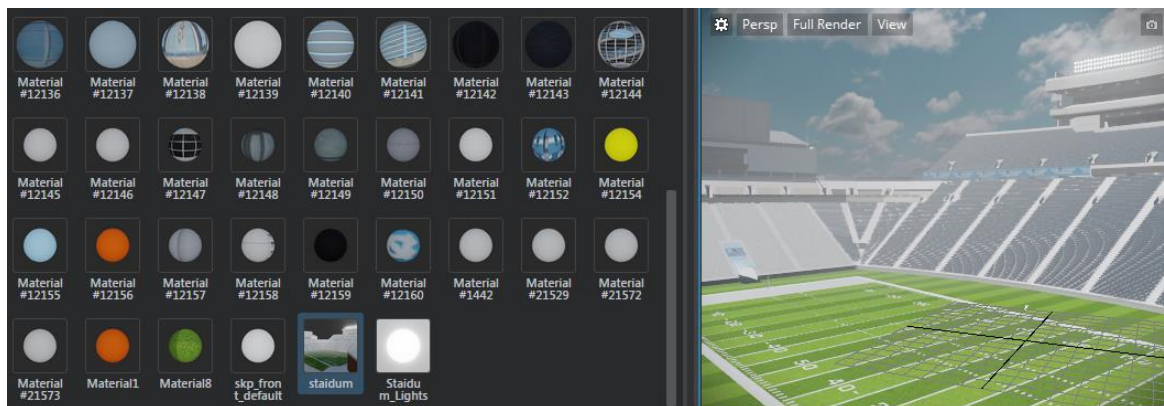
- Once Stingray has finished compiling the template, you can go ahead and save a new level under File> Save Level As.
- As you add contents to the Stingray project itself, you can aggregate these assets into a Level. The Level only aggregates the content, therefore deleting a level does not delete the assets it has aggregated from your project, those will remain in the project folder and can be aggregated into any level.
- Let's go ahead bring in some assets from 3ds Max.
- Select the stadium model and go to Stingray> Send Selection



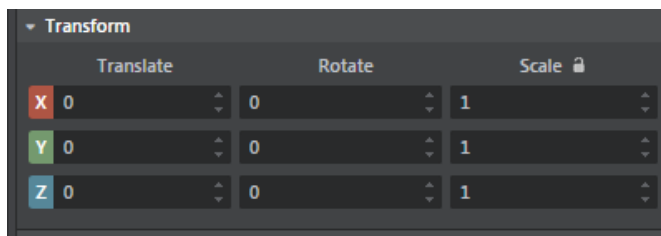
- This will open a browser pointing to the Stingray project currently open. Select a folder in which to place this asset. It helps to create one folder per imported asset. All the additional files that Stingray creates for the asset will be self-contained in this one folder. So if an asset is no longer needed in a given project, it is much easier to simply delete the parent asset folder than trying to find all the related files in a larger folder.



- Once the stadium has been imported into stingray, locate it in the Stingray project browser. In here, you will find the stadium model itself as well as all the shaders associated with the model.



- You can now drag and drop the model asset into the viewport. Make sure to center it to 0 0 0. This is so that it lines up to the same location as 3ds Max.



- Next, bring in all other assets following the same steps. The Infracore model (ground and buildings) should be one asset and the CFD splines another asset. You should have a result similar to this:

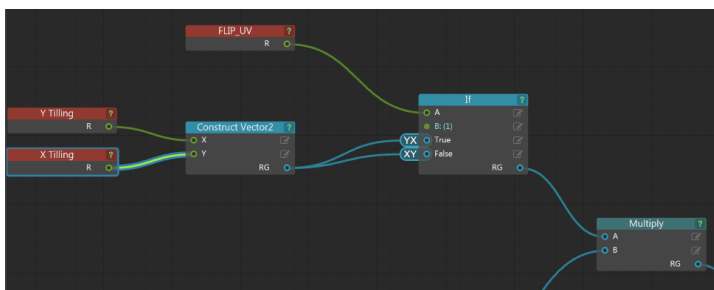




The first issue we encounter is that the CFD splines are not displaying the animated arrows. This is because Stingray handles UV direction differently than 3ds Max. To solve this, we need to invert the X Y mapping on the shader. There are several ways to solve this. One way is to simply invert the values on the material itself. Another more sophisticated way of solving this is to add a checkbox to the shader itself that inverts this for us. You can achieve this in both Stingray and 3ds Max since the shader FX platform is exposed identically in both products. Let's solve this directly in Stingray.

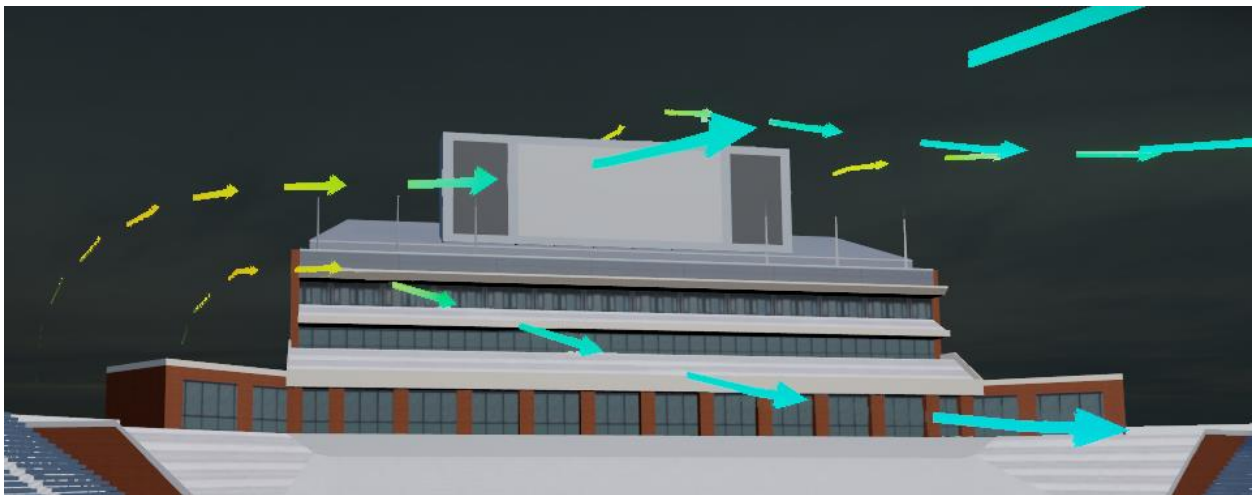
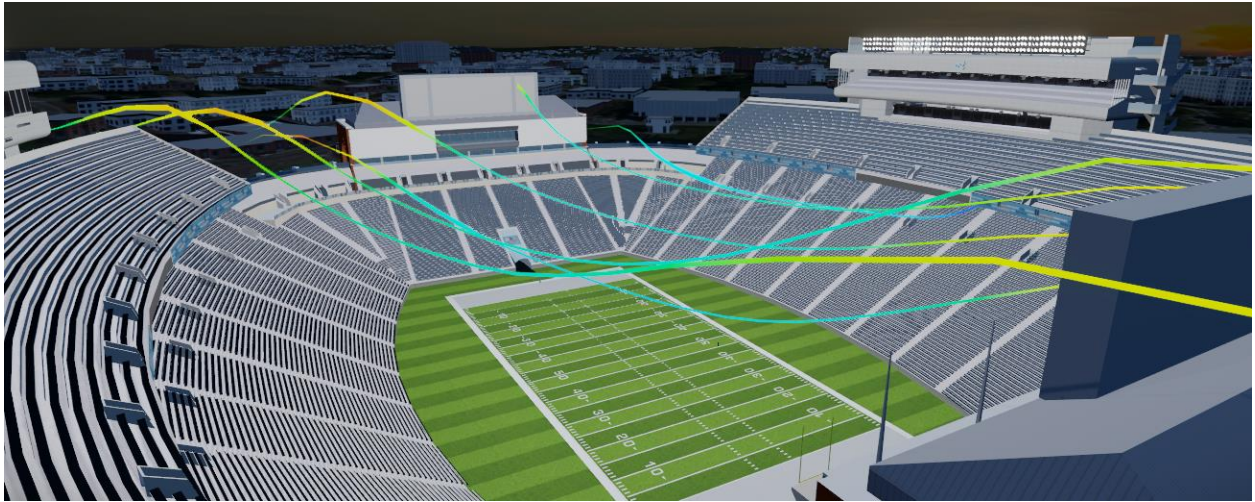
Locate the CFD shader. Double click on it to open the Shader FX node editor. Hit the TAB key to search:

- Add an IF node
- On the IF node, click on the B input option box, set it to 1.0
- Next add a Material Variable node, rename it to Flip UVs
- Set its Type to Scalar and UIType to checkbox
- Connect its output to the A input of the IF node
- Locate the Construct Vector2 node used for X and Y tiling. Connect its output to both- True and False inputs on the IF node
- Double click on the connection line going into True, add YX in the text field that appears
- Double click on the connection line going into False, add XY in the text field that appears
- Connect the IF node's output into the A input of the existing Multiply node
- Your graph should look like the following:





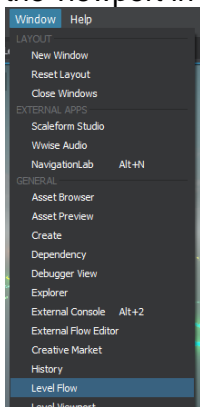
Now when you go back to the material UI in the properties editor, you should see a new checkbox. This will toggle on and off the Stingray UVs, give it a try. Try playing with the tiling and speed to create different looks and effects.



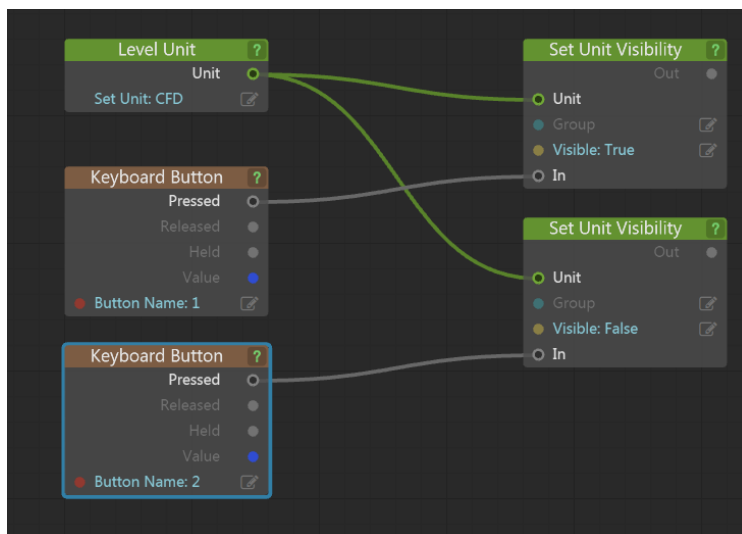
## 5.2 Workflow - Setting Up Triggers

In this section we will see how to use the Flow node editor in order to add triggers to our experience. In order to keep things simple, let's add a simple trigger to hide and unhide the CFD spline paths.

- First, we need to bring the CFD unit into Flow. The easiest way to do this is to first select the CFD unit itself in the viewport (doing so copies the unit to the clipboard).
- Now open the Flow editor by going to Window> Level Flow (it's the also the tab next to the viewport in a default UI layout).



- In here, right click and select Create Level Unit CFD. This is shortcut in where a Unit node is created already pointing to the selected object.
- Next hit the TAB key enter search mode and locate Set Unit Visibility
- Open the Visibility Option box and set it to True
- Copy\Paste this node and it to False
- Now connect the CFD unit node into both Unit inputs
- Next create a Keyboard Button node.
- Assign it to the key of your choice in the Option box
- Create a copy and assign a different key
- Connect both Keyboard Button nodes to both IN inputs on the Visibility nodes







As you can imagine, the keyboard keys assigned above will now control the visibility of the CFD unit. In a more elaborate setup you can imagine one key hiding many CFD simulations from various directions and only enabling the visibility of one. Or another possible scenario could be to have one single key cycle through various simulations. Whichever mechanism you prefer, Stingray and Flow allow you to achieve almost anything.

Thank you for reading,

Jose

