**CS 560 – FALL 2017**

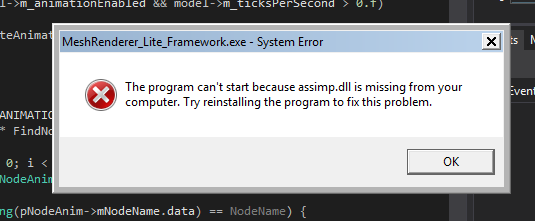
**Mesh Renderer User Guide**

**By:**

**Esteban Enrique Maldonado Cabán**

**Before running the renderer:**

* **I’ve tested in school computers with both Win10 and Win 7 so it should work when being graded. If it doesn’t feel free to email me(**[**esteban.maldonado@digipen.edu**](mailto:esteban.maldonado@digipen.edu)**)**
* When you run the solution for the first time, you’ll probably get this screen:

****

Like the screenshot shows, the assimp.dll is missing from the executable’s directory.

Copy the dll located at:

**MeshRenderer\_Lite\MeshRenderer\_Lite\_Framework\MeshRenderer\_Lite\_Framework\Lib\Assimp\assimp.dll**

Copy that file to:

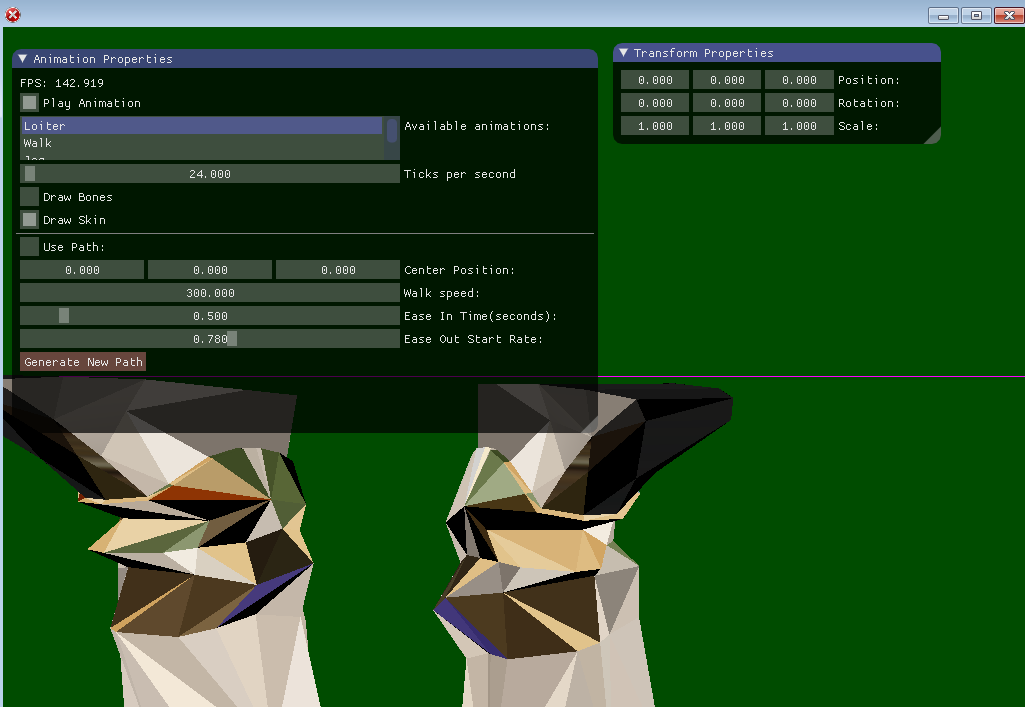
**MeshRenderer\_Lite\MeshRenderer\_Lite\_Framework\x64\[Debug | Release]**

*\* These directories are relative to where the framework is downloaded \**

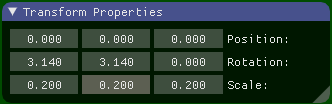
**Project 1:**

* I’ve chosen to start this with the model “tiny\_4anim.x” because it has multiple animations, and I’ve added functionality to change between those animations.

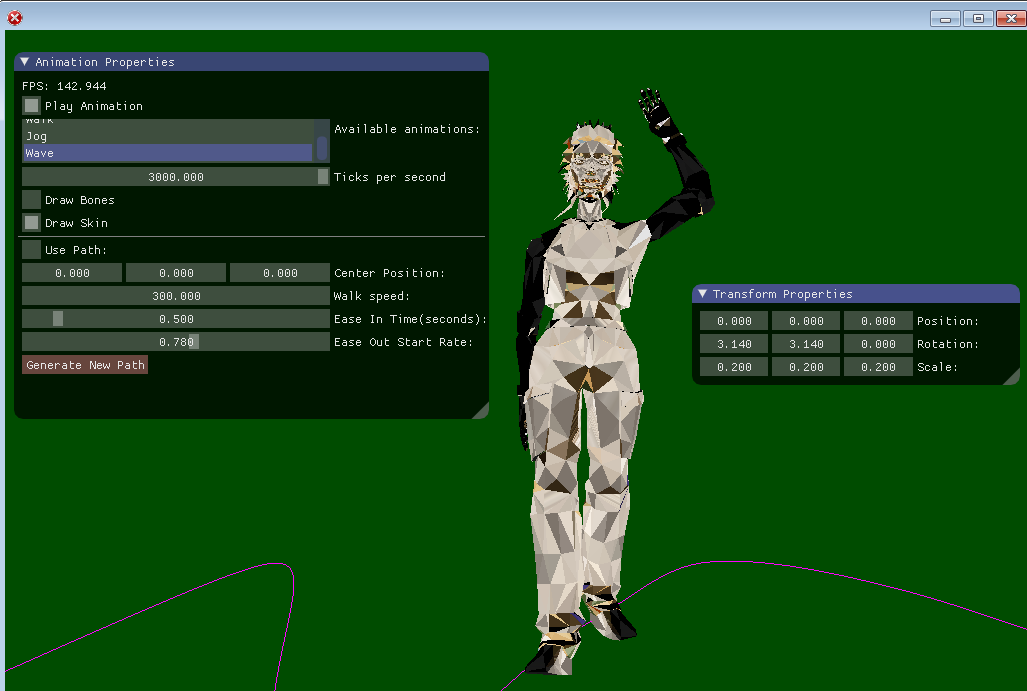
The only thing is that this model is super big and has a lot of frames:



On the **Transform Properties** window of the UI change the values to:



This will scale and rotate the model to face the screen. The different models that I’m using have different orientations so some adjustments are needed:



As a final adjustment, please drag the **Ticks per second(animation speed)** slider all the way up to 3000(current max value).

**Keyboard Camera Controls**

* **W & S:** Move in either the Z+ or Z-
* **A & D:** Move in either the X+ or X-
* **Q & E:** Move in either the Y+ or Y-

***Since I currently only support DX11, orientation is LH. Y axis is Up. –Z direction is towards the screen.***

**Project 2:**

**Framework controls for path following:**

* **Use Path:** enable or disable the use of the Curve Path component
* **Center Position:** The position of the center of the curve. If you modify these values the curve will change position and the model will adapt to this offset.
* **Walk Speed:** The speed that the model will use to follow the points along the path
* **Ease-In Time:** The time in seconds that the model will slowly accelerate until it reaches the full Walk Speed. I’ve capped this value to 0 and 5 seconds.
* **Ease-Out Rate:** A normalized value represents how much of the total distance has the model traveled. When it hits this factor, it will start to slow down until it reaches the end.
* **Generate New Path:** Click on this button and a new random curve will be generated

**Project 3:**

**Details of the scene you’ll see:**

**This is a very simplified version of the CCD algorithm. In order to achieve precise results I’ve taken the liberty of running the algorithm in 2D but rendered in a 3D world.**

**Also, please move the camera closer to the origin by using the camera controls explained for project 1. In this case just move it forward with the ‘W’ key.**

* **The Red cube** drawn on the scene is the end effector of the robotic arm.
* **The White cube** drawn on the scene is the target position.
* Use the **Target Position:** slider control and the algorithm will run again.
* Use the **Re-run CCD** button to set a new random position to the target.
* Whenever the algorithm is run the CCD properties window will show wether it found the target or not.

**Project 4:**

* **RUN THIS ON RELEASE MODE PLEASE!**
* **For this demo you cannot move the cloth’s position. So please move the camera closer to the cloth using the camera controls explained for project 1.**

**Framework controls for fan:**

* **Fan Position:** The position of the “fan”, the direction of the fan is always from fan to center of the cloth.
* **Fan Strength:** The scalar multiplier for the fan strength in that direction
* **Fan Radius:** A radius of “wind attack”, the closer the fan is to the cloth, the stronger reaction you will see.