**User Story Object Replacement:**

As a software developer

I want to create an object replacement python script.

So, that I can lower 3d model poly-counts.

**Tasks:**

* Pull object mesh data from blender.
* Research methodologies to create a training set for mesh recognition.
* Establish a training set for recognizing screws.
* Use TDD approach to successfully recognize screw meshes in Blender.
* Establish a low poly repository for model replacement.
* Create script that implements the above. Successfully replace screws using machine learning.
* Test for errors, issues and edge cases.
* Expand to include other mesh types of small size/high density.

**Notes:**

The initial idea is that we can use machine learning with libraries such as sci-kit learn to create a mesh recognition script that can be used within blender. Once, the mesh is recognized as a valid mesh type belonging to a, for example, “screw family” we will replace the mesh in scene with a low poly version. This will require that we either generate the mesh on recognition, replacing the high poly version OR that we do a mesh swap from an establish low poly repository. Generating the model will allow us to save memory but will require additional computation time.

**User Story Transformation Baking:**

As a software developer.

I want to bake world transformations on meshes.

So, that model manipulation is consistent across all import types.

**Tasks:**

* Research/Document how FBX applies transformations on import.
* Review how RMIT flatten script works and how world transformations conflict.
* Develop a way to apply world transformations on all import types besides FBX.
* Use TDD approach to develop solution.
* After successfully applying solution to a single input type, extend to all file types.
* Test several model types and file types for issues.

**Notes:**

FBX is currently the only native importer that supports baking in world transformations. We have a program feature that will flatten model hierarchy trees into a 1D list. The reason we do this to is simplify models while preserving the transformations in the mesh tree. We currently can not do this with file types other than FBX without losing transformation relationships. It is possible that the current flatten script logic could be the issue. It is more likely we need to implement logic to do the baking like in the FBX importer. Both must be looked into.

**User Story Model Simplification:**

As a software developer.

I want to upgrade our decimation/model simplification.

So, that I can better preserve model integrity while reducing poly counts.

**Tasks:**

* Read white papers on QEM4vr.
* Build and test code based on the above white papers.
* Do a comparison of current decimation to QEM4vr.
* Use TDD to implement QEM4vr as an alternative to Blender decimation.
* Update current scripts to utilize QEM4vr
* Run analysis and create metrics comparing before and after

**Notes:**

Blender currently uses a decimation algorithm to reduce mesh polycount. The blender method uses an edge collapse algorithm. This allows you to set a target triangle and vertex count. While effective there is reason to believe that the QEM4vr model simplification algorithm we yield better results. We have the white paper, code and an point of contact to assist us with implementing it into our project.

**User Story Hole Closure:**

As a software developer.

I want to create a mesh hold closure solution.

So, that I can ensure model integrity.

**Tasks:**

* Identify and document the different mesh issues that can occur.
* Find ways to identify holes and geometric issues in meshes.
* Create a solution that attempts to fix geometric issues that are found.
* Use TDD to implement a code solution that can be ran as a script or add-on.
* Test on a sample set of meshes.

**Notes:**

We currently have some tools that allow us to ensure model integrity and attempts to correct and errors that are found. We need to further this and ensure we have a robust solution that identifies and fixes and mesh issues. We then need to ensure the mesh checks/fixes are well documented and understood.

**User Story STP Importer:**

As a software developer.

I want to create an importer for stp files

So, that I can import stp files into blender

**Tasks:**

* Research if there are already any solutions that exists.
* Determine the difficulty of developing a solution.
* Begin to develop a solution that can feasibly be implemented.

**Notes:**

Initial research into this seems to suggest that creating such a solution is very difficult. Further investigate and determine the feasibility of developing a solution.