

Acceptance Tests

Note: please install the software first before doing these, see the installation guide

Sprint 1

Story 1

User story 1 As a consumer, I want to be able to take up to 6 images and generate it into a 3D model, so that I can work with it in a 3D environment.

- Open 3D-Model-Generator.exe in GitHub\3D-Model-Generator\dist,
- Open the editor by typing in "editor" to the console and hitting enter
- Once the UI is open click on one of the empty right buttons in the CCW turned t-formation, these 6 buttons each represent an image that can be put onto one of the sides.
- To add an image to that side click on the load image button after clicking on one of the 6 buttons, this will open a browser window where you can select an image
- The software is currently limited to images on a white background, if you do not have any images, feel free to use the simple shapes in the resources folder
- The formation of pictures is as follows
 - Top
 - Left front back right
 - Bottom
- Click on update to see a preview of your object
- next click on view model and finally render to create a model

Story 2 As a consumer, I want export my generated 3D model as a .obj file, so that I can use it across multiple platform.

- Moved to next sprint

Story 3

Sprint 2

Story 1 As a developer, I want to be able to clean up and test the work done from the previous sprint, so that we have a working prototype to continue developing off of:

- Open the TEST Folder
- Open the project in QT (<https://www.qt.io/>) or look at the source code in the src folder
- In QT open project file (3D-Model-Generator(.pro)), compile and run to see unit tests
- Open folders in sidebar to see source code commenting

Story 2 As a consumer, I want export my generated 3D model as a .obj file, so that I can use it across multiple platform.

- Currently unimplemented
- If working, it could be checked by clicking export in the model editor window (view model).

Story 3 As a consumer, I would like to be able to upload my own images through a browser window so I don't have to manually name the files or put them in a folder.

- Open 3D-Model-Generator.exe in GitHub\3D-Model-Generator\dist,
- Open the editor by typing in "editor" to the console and hitting enter
- Click on one of the push buttons in the first section of the right-hand bar.
- Click on "load image", located near the bottom of the right-hand bar.
- Find an image in the file browser and open it.
- The image should be loaded into the slot represented by the push button earlier selected.

Sprint 3

Story 1 As a consumer, I would like an interface to tweak the settings of the model generator so that I can have more control over the output/outcome.

- When in model rendering window before hitting preview, change the settings of the model renderer with the two dropdown menus

Story 2 As a consumer, I would like a tweak on the shape algorithm to improve the smoothing and faces of the shape to make it look a lot more like the desired image.

- in the src folder there are 2 classes, VertexLinker and FaceMaker.
- VertexLinker's structure and algorithm was replaced multiple times with an improved algorithm until the final algorithm which is stored in FaceMaker which took up a good chunk of the manhours in this project
- To see these old changes (requires knowledge of git):
 - Instal some form of git/github
 - Open a git or github console
 - Clone from directory <https://github.com/cricketspike/3D-Model-Generator/> into git
 - Checkout the main branch with git checkout
 - git log -p -- src/vertexlinker.cpp to see all the changes done over time
 - Up and down arrows scroll, q to quit
 - Omit the -p to just see commit names over time

Story 3 As a consumer, I would like options and tools to be able to change the input images so that I have more control over how the output model will end up looking in the end.

- Open 3D-Model-Generator.exe in GitHub\3D-Model-Generator\dist,
- Open the editor by typing in "editor" to the console and hitting enter
- Once the UI is open click on one of the empty right buttons in the CCW turned t-formation, these 6 buttons each represent an image that can be put onto one of the sides.
- While one of these buttons is selected, click