Test Document for 3D Annotation Tool

1. Introduction

This document talks about the testing strategy and approach for the 3D model Annotation tool project, which will focus on user acceptance, performance, compatibility and continuous feedback to optimise this project and meet the client's requirements. This document will focus on the core functions like loading and displaying 3D models, using hotkeys and annotation tools like spray, keypoint and the shortest path.

2. Testing

2.1 User Acceptance Testing

Objective: This will used to validate the usability and user-friendliness of the application to ensure the application meets clients' requirements and is user-friendly.
Procedure:

- Let clients and teammates use this application
- Set specific user goals (e.g., load an STL file, do the spray annotation on the model, use a hotkey to rotate this model, and so on)
- Observe and record the problems and let users give their feedback.

Criteria for success: The user can easily finish those user goals and have some positive feedback toward this application.

2.2 Performance Testing

Objective: To assess the response times of applications and try to handle capabilities with large and complex models.
Procedure:

- Upload large and complex 3D models and do the complex annotations
- Check the system response times and the accuracy in rendering and manipulation when doing the annotation.

Performance Metrics: response time, accuracy of the intersection between raycaster and model face.

2.3 Compatibility Testing

Objective: Ensure the application can be installed and run successfully on different operating systems and hardware.
Procedure:

- Test this application on various devices and operating systems.
- Record the issues and problems that show when running applications from different systems.

Acceptance Criteria: This application can run smoothly and does not appear to have any errors across tested platforms.

2.4 Continuous Feedback Integration

Objective: Having a mechanism to get ongoing feedback to iteratively improve this application.
Procedure:

- Having the weekly demonstrations to clients
- Collect and analyse the feedback from clients and improve the application according to that feedback.

Feedback Criteria: The team can get feedback frequently and solve those problems immediately.

3. Test environment

To run this application some requirements about operating systems: Windows 10 or higher, MacOS Sierra or higher, or Linux OS with a graphical user interface. It also needs at least 500MB of free space, 8GM RAM or higher.

4. Risks and Mitigation

4.1 Identical risks

Resource constraints, e.g. insufficient time, budget, or humans, may affect the comprehensive testing.
Technical constraints: The development of complex model processing and annotation may lead to technical constraints.
Software compatibility: the stability of the application may have some issues when running the application across different operating systems.

4.2 Mitigation ways

Resource allocation: This application will test the core function at first and ensure the core function does not have problems.
Technical pre-study: for the core function, do the pre-study and prototype test first to reduce the possibility of failure.
Compatibility test: At the beginning stage of the project, start the compatibility test to find out the potential problems.