

Master testplan

TextureSync

Version	1.0.0
Datum	08.04.19
Autor	Hendrik Schutter
Projektmitglieder	Hendrik Schutter, Lukas Fürderer, Robin Willmann, Jannik Seiler

Inhaltsverzeichnis

1 Einleitung.....	3
1.1 Ziel des Mastertestplans.....	3
2 Modultests.....	3
2.1 Testumgebung.....	3
2.2 Test Definitionen.....	3
3 Integrationstests.....	3
3.1 Testumgebung.....	3
3.2 Test Definitionen.....	4
4 Systemtests.....	4
4.1 Testumgebung.....	4
4.2 Test Definitionen.....	4
5 Akzeptanztests.....	4
5.1 Testumgebung.....	4
5.2 Test Definitionen.....	5
6 Changelog.....	7

1 Einleitung

1.1 Ziel des Mastertestplans

Das Ziel des Mastertestplans ist es die einzelnen Test-Phasen zu definieren und zu kontrollieren.

In den Tests des Mastertestplans werden die Voraussetzungen und die Durchführung der Test beschrieben.

Die einzelnen Testpläne (Modultest, Integrationstest, Systemtest und Akzeptanztest) werden aus dem Mastertestplan abgeleitet. In den einzelnen Testplänen werden auch die Ergebnisse festgehalten. Im Mastertestplan nur die erwarteten Ergebnisse.

2 Modultests

2.1 Testumgebung

Teile der Server bzw. Client Software werden nach dem entwickeln einzeln getestet. Als Testwerkzeuge dienen eigens geschriebene Test-Anwendungen welche die Module ansprechen und deren Ergebnisse darstellen. Da die Tests direkt am Source-Code in der IDE vorgenommen werden wird mittels White-Box Verfahren getestet.

2.2 Test Definitionen

Nummer	Titel	Erklärung
MT#1	blabla	blabla
MT#2	blabla	blabla

3 Integrationstests

3.1 Testumgebung

Mehrere komplette Arbeitsaufträge für die komplette Software werden getestet. Aus einem bestimmten Ausgangspunkt muss mit den gleichen Interaktionen das gleiche Ergebnis entstehen.

Als Ausgangspunkt werden speziell präparierte Situationen gewählt welche wiederholt werden können.

Die fehlerfreie Kommunikation innerhalb der Client und der Serversoftware wie auch die Kommunikation des Servers und des Clients werden dadurch verifiziert.

3.2 Test Definitionen

Nummer	Titel	Erklärung
IT#1	blabla	blabla
IT#2	blabla	blabla

4 Systemtests

4.1 Testumgebung

Die Musskriterien wie auch die implementierten Wunschkriterien aus dem Pflichtenheft werden einzeln getestet. Dabei werden auf Randbedingungen wie auch auf Fehlerfälle besonders geachtet. Mittels Black-Box Tests werden dabei der Output des kompletten Software verifiziert.

4.2 Test Definitionen

Nummer	Titel	Erklärung
ST#1	blabla	blabla
ST#2	blabla	blabla

5 Akzeptanztests

5.1 Testumgebung

Als Testumgebung gilt hier kontre wie im Lastenheft angegeben Ubuntu 18.04 (x64) für die Server wie auch Client Software. Die funktionalen wie auch nicht-funktionalen Anforderungen werde getrennt von einander getestet. Die User-Stories werden von Endbenutzern durchgeführt.

5.2 Test Definitionen

Nummer	Titel	Erklärung
AT#1	Texturen hinzufügen	Jeder Nutzer ist in der Lage, neue Texturen in die Sammlung aufzunehmen und auf den Server zu laden. Bei doppelten Namen schlägt das Hinzufügen fehl und informiert den Nutzer. Zum Hinzufügen wählt der Nutzer eine Texturdatei aus, die auf dem Computer des Nutzers abgespeichert ist.
AT#2	Tags verwalten	Jeder Nutzer kann Tags zu beliebigen Texturen der Sammlung hinzufügen oder löschen.
AT#3	Metadaten speichern	Jede Textur in der Sammlung hat einen eindeutigen Namen, ein Einpflegedatum, eine Auflösung und kein, ein oder mehrere Tags.
AT#4	Filter	Die Sammlung aller Texturen lässt sich zur besseren Übersicht nach Metadaten filtern. Mögliche Filterkriterien sind: <ul style="list-style-type: none">• Das Vorhandensein von Tags• Das Nichtvorhandensein von Tags• Mindestauflösung• Maximalauflösung• Filtern nach Einplegedatum• Namen
AT#5	Synchronisation	Alle in die Sammlung aufgenommenen Texturen sowie deren Metadaten sind für alle anderen Nutzer sichtbar.
AT#6	Texturen extrahieren	Jeder Nutzer kann beliebige Texturen aus der Sammlung extrahieren und in seinem lokalen Dateisystem abspeichern.
AT#7	Detailansicht	Jeder Nutzer kann von beliebigen Texturen eine 3D-Vorschau anzeigen lassen und die gesetzten Metadaten einsehen.
AT#8	Darstellung	Bei der Nutzung des Filters werden alle passenden Texturen mit einer Vorschau angezeigt.
AT#9	Anzahl von Clients	Es können mindestens 10 Clients gleichzeitig aktiv sein.
AT#10	Anzeigen von Änderungen	Neue Texturen oder geänderte Tags werden nach dem Hochladen bei einer neuen Suchanfrage angezeigt.
AT#11	Defekte Daten	Wenn es beim Übertragen von Texturen auf den Server zu einem Fehler kommt wird die Textur nicht in das System übernommen, und der Benutzer wird gewarnt.

AT#12	Konfiguration, Server	Die Server-Software kann auf einem Ubuntu 18.04 (x64) ausführbar gemacht werden.
AT#13	Konfiguration, Client	Die Client-Software kann auf einem Ubuntu 18.04 (x64) ausführbar gemacht werden.
AT#14	Sicherung von Daten	Alle Daten im System können gesichert werden durch das Kopieren eines einzelnen Ordners im Dateisystem des Servers. So gesicherte Daten können wiederhergestellt werden, indem sie zurückkopiert werden und danach die Server-Software neu gestartet wird. Hierbei wird vorausgesetzt, dass die Struktur beibehalten wird.
AT#15	Als Anwender möchte ich eine Holz-Textur finden. Ich gebe den Tag „Holz“ ein und es werden mir sofort entsprechende Texturen angezeigt.	User-Story
AT#16	Ich will einfach neue Texturen von einem Rechner in das System einfügen. Ich kennzeichne diese mit Tags, um sie für meine Kollegen auffindbar zu machen.	User-Story
AT#17	Wenn ich eine passende Textur gefunden habe will ich diese auf meinem Rechner im Projekt abspeichern.	User-Story
AT#18	Ich will beim Hinzufügen neuer Texturen nicht viele Eigenschaften, wie z.B. Auflösung, von Hand eintippen, das soll das System zum größten Teil tun.	User-Story
AT#19	Ich möchte nur Texturen mit einer Mindestauflösung von 1024 x 1024 Pixel angezeigt bekommen.	User-Story

6 Changelog

Version	Änderung
1.0.0	-