

# Grobdesign

## TextureSync

|                   |   |
|-------------------|---|
| Version           | 1.2.0   |
| Datum             | 20.04.2019  |
| Autor             | Lukas Fürderer,<br>Hendrik Schutter (Client Diagramm(e))<br>Robin Willmann (Server Diagramm(e)) |
| Projektmitglieder | Hendrik Schutter,<br>Lukas Fürderer,<br>Robin Willmann,<br>Jannik Seiler                        |

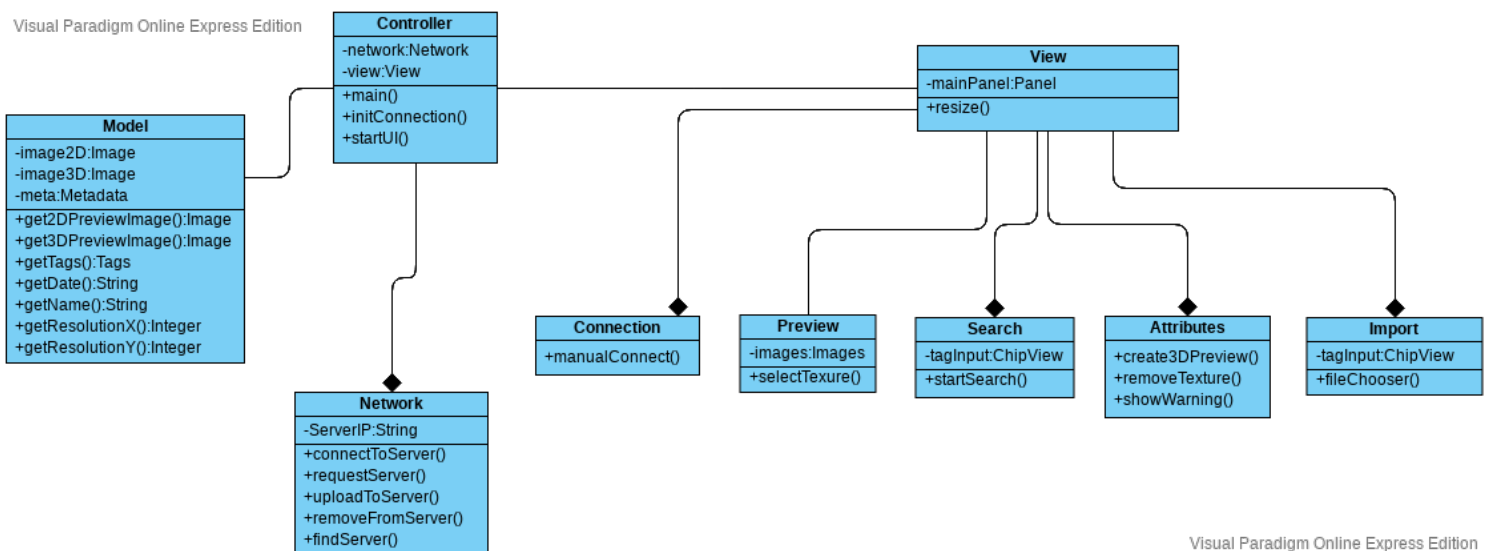
# Inhaltsverzeichnis

|                      |   |
|----------------------|---|
| 1 Einleitung.....    | 3 |
| 2 Client-Module..... | 3 |
| 2.1 Model.....       | 3 |
| 2.2 View.....        | 3 |
| 2.3 Controller.....  | 5 |
| 3 Server-Module..... | 6 |
| 3.1 Model.....       | 6 |
| 3.2 Main.....        | 6 |
| 3.3 Protocol.....    | 7 |
| 3.4 Persistency..... | 7 |
| 4 Changelog.....     | 8 |

# 1 Einleitung

Im Grobdesign wird festgelegt, aus welchen Modulen Client und Server jeweils bestehen und welche Aufgaben diese Module haben.

## 2 Client-Module



### 2.1 Model

Im Modul „Model“ ist eine Datenstruktur zum Repräsentieren einer Textur definiert. Dies dient als Austauschformat zwischen dem Netzwerk-Modul und dem View-Modul.

### 2.2 View

Das View-Modul und seine Untermodule steuern die gesamte Anzeige der GUI und deren Verhalten. Es ist weiter unterteilt in die verschiedenen Einzelkomponenten der GUI. Im View-Modul selbst werden nur das Hauptfenster verwaltet und die Anordnung der Komponenten darin bestimmt.

### Connection

Das Connection-Modul wird beim Programmstart benötigt und stellt ein Fenster zum Herstellen der Serververbindung bereit.

## **Search**

Das Search-Modul stellt ein Panel bereit, über das der Nutzer die gewünschten Stichworte, Tags und Auflösung einstellen sowie eine Suchanfrage abschicken kann.

## **Preview**

Das Preview-Modul hat die Aufgabe, alle gefundenen Texturen einer Anfrage jeweils mit einer 2D-Vorschau anzuzeigen. Der Nutzer muss einzelne dieser Texturen auswählen können, um im Attributes-Panel Änderungen daran vornehmen zu können.

## **Attributes**

Das Attributes-Modul stellt ein Panel bereit, das eine 3D-Vorschau der Textur zeigt. Außerdem werden der Name und die gesetzten Tags angezeigt und sind änderbar.

Zusätzlich gibt es einen Lösch-Button, über den nach einer Sicherheitsabfrage die gewählte Textur vom Server gelöscht wird. Der dafür notwendige Dialog liegt ebenfalls in dem Attributes-Modul.

## **Import**

Das Import-Modul enthält den Import-Dialog, um neue Texturen vom lokalen Dateisystem auf den Server hochzuladen. In diesem Dialog lässt sich eine Datei auswählen, bei Bedarf der Dateiname ändern und die gewünschten Tags setzen.

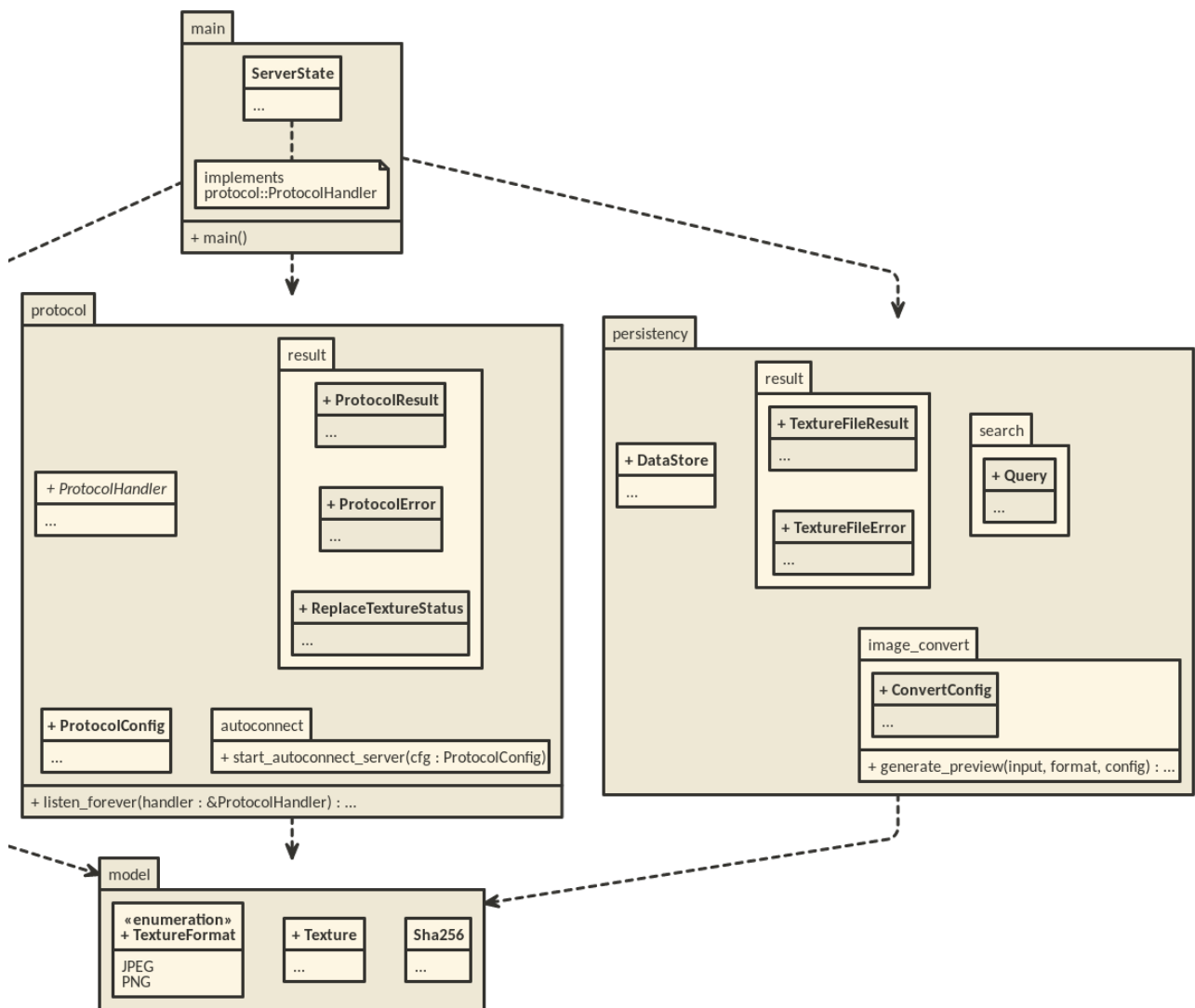
## **2.3 Controller**

Das Controller-Modul enthält den Einsprungspunkt für das Programm. Es steuert den grundlegenden Ablauf (Verbindungsaufbau, Erstellen der GUI) und stellt die Verbindung zwischen dem View-Modul und dem Network-Modul her.

## **Network**

Im Network-Submodul des Controllers befindet sich eine Routine zum automatischen Finden eines Servers im lokalen Netzwerk. Außerdem ist hier die Client-Seite des Netzwerkprotokolls implementiert.

## 3 Server-Module



### 3.1 Model

Im Modul „Model“ wird der Datenbestand der Texturesammlung, d.h. die Namen aller Texturen, deren Größe und gesetzte Tags abgebildet. Das Modul speichert auch die Vorschaubilder der Texturen im Arbeitsspeicher, sofern sie bereits generiert sind. Es speichert jedoch nicht die Bilddatei selbst, dafür ist das Persistence-Modul zuständig.

## 3.2 Main

Das Main-Modul enthält den Einsprungspunkt des Server-Prozesses und übernimmt koordinierende Aufgaben. Es verwaltet die Anfragen verbundener Clients und sorgt für ein sauberes Beenden des gesamten Server-Prozesses.

## 3.3 Protocol

Im Protocol-Modul ist die Server-Seite des Netzwerkprotokolls implementiert. Nur die Kommunikation per TCP ist Teil dieses Moduls, für die Abarbeitung eingehender Anfragen ist das Main-Modul zuständig.

### Autoconnect

Implementiert den Autoconnect-Server: Arbeitung und Beantwortung von Autoconnect-Anfragen per IPv6-Multicast.

## 3.4 Persistency

Das Persistency-Modul hat die Aufgabe, eintreffende Bilddateien auf dem Dateisystem des Servers zu sichern. Darüber hinaus sichert es in regelmäßigen Abständen sowie vor dem Beenden des Serverprozesses eine Abbildung des Datenbestands im Dateisystem. Darin sind jedoch keine Vorschaubilder enthalten, da diese nach einem Neustart neu generiert werden können.

### ImageConverter

Das ImageConverter-Modul ist vorgesehen, um für gewünschte Texturdateien Vorschaubilder einer festen Größe (256x256 Pixel) im JPG-Format zu erzeugen.

### Search

Hat die Aufgabe Suchen die Persistenten Daten zu durchsuchen. Diese ist Teil von Persistency, weil sie eng an das Daten Format gekoppelt ist.

Search hat auch die Aufgabe die textuelle Suchanfrage zu interpretieren.

## 4 Changelog

| Version | Änderung  |
|---------|---|
| 1.0     | Grobdesign erstellt   |
| 1.0.1   | Client Klassendiagramm hinzufügt  |
| 1.1.0   | <ul style="list-style-type: none"><li>• Server Packetdiagramm,</li><li>• Network in Protocol umbenamt</li><li>• ImageConverter ist nun Teil von Persistency</li><li>• Neu: Search</li></ul> |
| 1.1.1   | Nur eine Textur ist zur gleichen Zeit auswählbar.   |
| 1.2.0   | Autoconnect nun im Server Diagramm ergänzt.   |