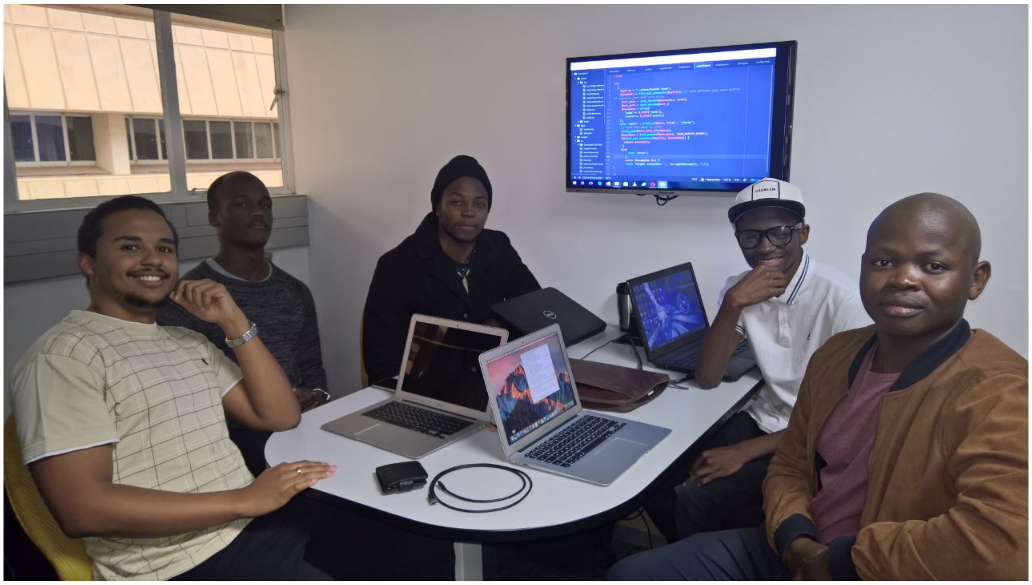
V3D Digraph Visualizer Architecture and Design

Team: App-Synth

Author(s):

- Kulani Bamuza
- Keanan Jones
- Munyaradzi Mpofu
- Neo Thokoa
- Takalani Sigama

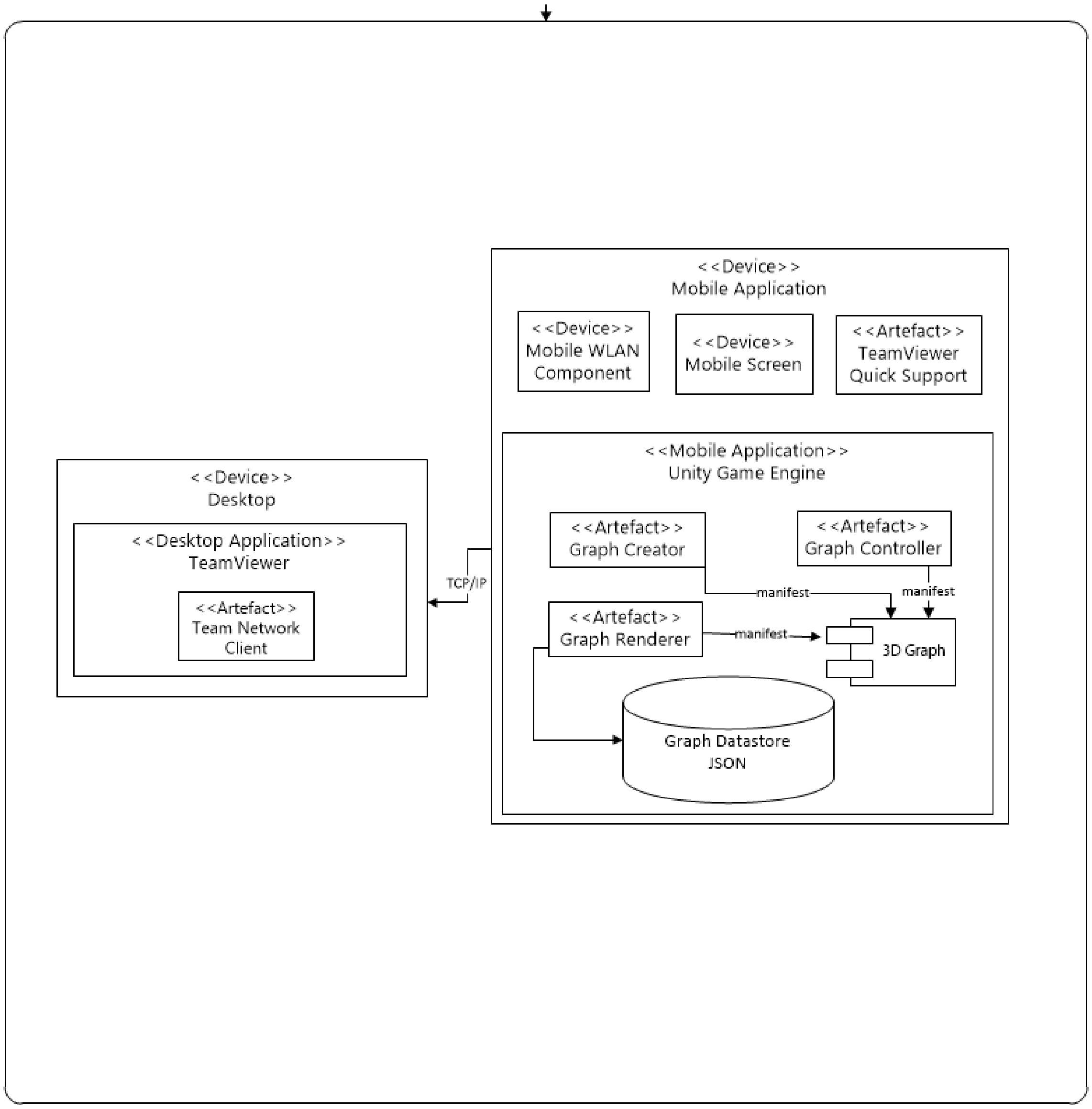


University of Pretoria, Department of Computer Science
15 May 2017

Contents

1 Architecture

For the V3D Digraph Visualiser, the Master-slave architecture is employed. This is due to the downstream flow of information from the ”master” node(V3D app on phone) to the external display (”slave” node).
All control is on the V3D mobile app, therefore this will be the master node of the system. The mobile application will stream instructions to the desktop application. We do this to achieve near real time synchronization between the two displays. The slave, ie. the desktop application, will receive the instructions, decode them and update the view accordingly.


<<Device>>
Mobile Application
<<Device>>
Mobile WLAN Component
<<Device>>
Mobile Screen
<<Artefact>>
TeamViewer Quick Support
<<Mobile Application>>
Unity Game Engine
<<Artefact>>
Graph Creator
<<Artefact>>
Graph Controller
manifest
manifest
<<Artefact>>
Graph Renderer
manifest
3D Graph
Graph Datastore
JSON
<<Device>>
Desktop
<<Desktop Application>>
TeamViewer
<<Artefact>>
Team Network Client
TCP/IP

2 Quality Requirements

The V3D Digraph Visualiser has a number of quality requirements which will have a major impact on the overall impression of the system. These include:

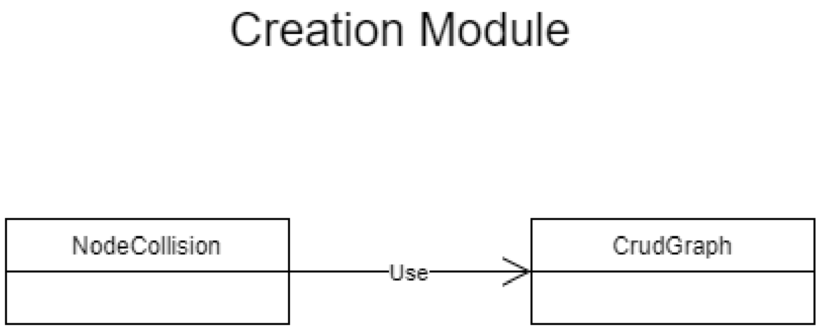
- Extensibility
- Reliability
- Usability

3 Creation Module

3.1 Scope

The purpose of this module is to handle the creation of new graphs in the virtual environment. This module is responsible for the creation of graph nodes and the creation of edges between these nodes. The module is also responsible for the removal of nodes from the graph and saving the created graph to a graph specification file in order for the graph to be visualised again at a later stage.

3.2 Domain Models

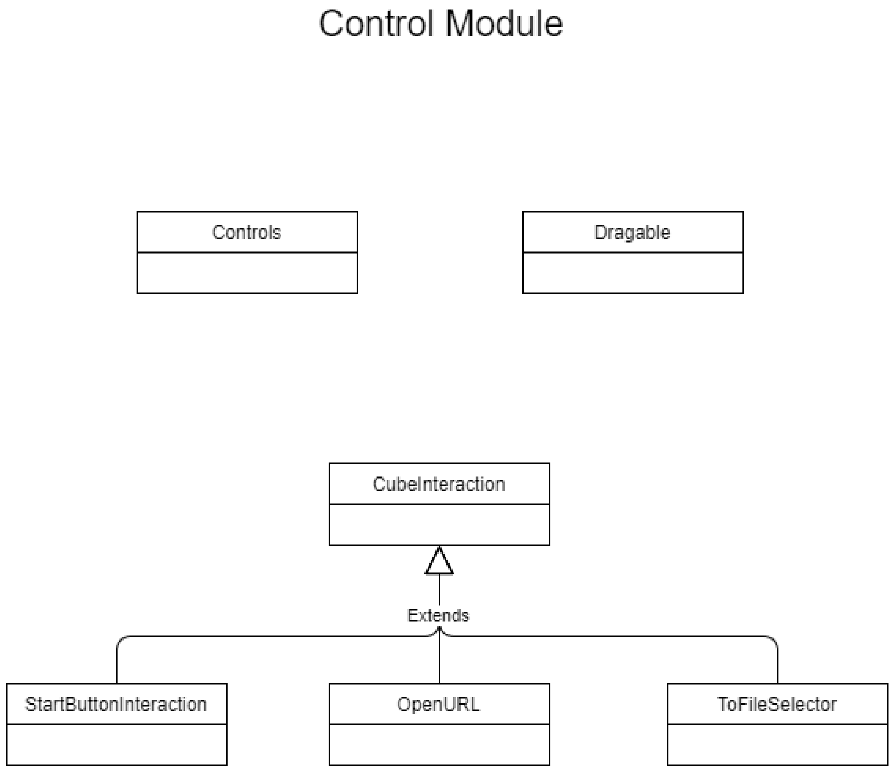


4 Control Module

4.1 Scope

The purpose of this module is to handle all of the controls within the system. These include menu controls, graph controls and node controls. Menu controls are used to navigate between different scenes in the virtual environment, graph controls which are used to rotate and reposition the graph and node controls are used to interact with graph nodes by performing actions such as picking up, dropping and repositioning nodes.

4.2 Domain Models

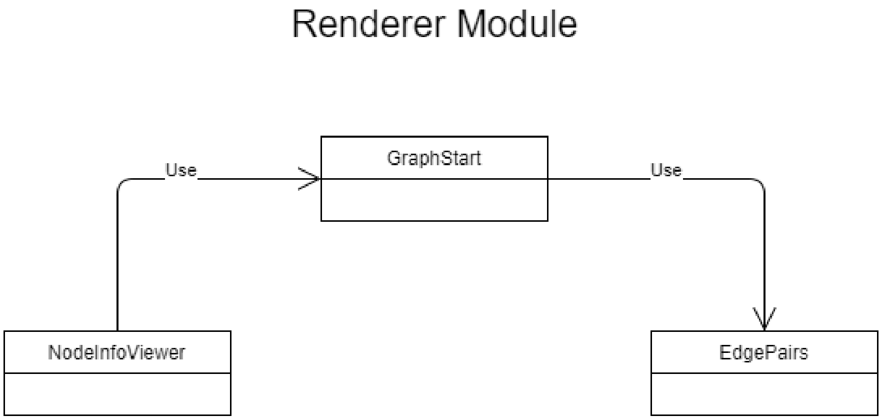


5 Renderer Module

5.1 Scope

The renderer module is responsible for creating the digraph model in the virtual environment. This module renders all of the nodes and relationships between all the nodes and clusters nodes of a particular type. In addition, this module will also display the attributes of a selected node and display the name of each node directly above the node.

5.2 Domain Models

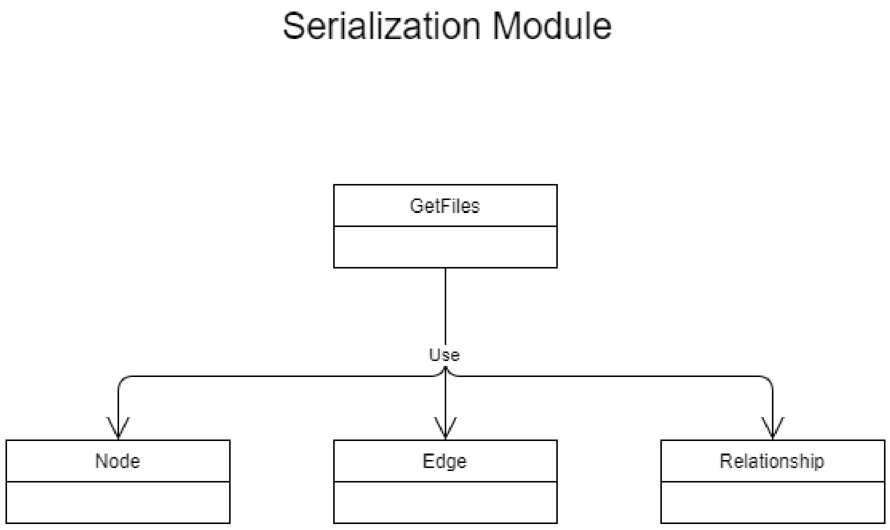


6 Serialization Module

6.1 Scope

The serialization module is responsible for the serialization of JSON data into objects. These JSON objects will contain the necessary information for the graph model to be created in the virtual environment. This module will also be responsible for serializing the objects to JSON format for the purposes of creating a graph specification file.

6.2 Domain Models



Serialization Module

7 Technologies

For the purposes of the development of this application, the Unity Game Engine is used. Unity is used to create and render all of the graphical components of the system. A 3rd party 3D modelling software called Blender is also used for the creation of the graphical components which are rendered within the application. The Newtonsoft C-Sharp library is used for the purpose of serializing and de-serializing of the graph specification files which are stored in JSON format. The TeamViewer mobile and desktop applications were used to display the mobile screen on an external display, where the mobile device took on the role of a server sending screen data and the desktop application took on the role of a client receiving screen data.