V3D Digraph Visualizer Documentation

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 Introduction

1.1 Project Background

Graph visualizations allow for humans to perceive the information contained within graph structures in a presentable manner. There exists a number of tools which can be utilised to create 2D visualisations of graph from a single specification. An example of such a tool is GraphViz (insert citation). The aim of this project is to provide the means to generate 3D graphs from a single specification. The graph will then be visualised in a 3D space, which will allow the viewer to immerse themselves within the graph and interact with it.

1.2 Project Vision

The increase in performance of graphics engines in this day and age has given rise to many innovative uses of graphics and the development of various graphical applications. The problem which is to be addressed by this project is the design and development of a graphing tool which will visualise a digraph on a plane or a sphere. The viewer of the graph should be able to interact with the digraph and manipulate various aspects of the visualisation.

1.3 Project Scope

The system must allow for the visualisation of digraphs which will defined using the set of triples notations as specified by Barla-Szabo (insert citation). The user should be able to change their vantage point while viewing the graph on a hyperplane or hypersphere. The edges of the graph are to be labels using tuples which contain metadata about the edges. This metadata will provide the necessary information required to produce a force

directed graph. The users will be immersed in the 3D environment by means of virtual reality 3D glasses. The user should be able to use 3D glasses such as Google Cardboard (insert citation) which they can build themselves in order to make the system more accessible. Besides viewing the graph, the viewer should be able to interact with the graph by manipulating the metadata related to certain vertices and edges. The system should allow for observers to view the 3D environment on an external display.

2   Overall Description

2.1   Product Perspective

The entire system will consist of 2 different applications, namely the desktop application and the mobile application. The mobile application will be deployed on a mobile phone running an android operating system. This application will contain a graphical user interface through which the user can make use of the system. A pair of virtual reality glasses will be required by the user in order to visualise a graph in a 3D space. The mobile application itself will interact with various components of the mobile device, including the camera and the gyroscope. The desktop application will be compatible with Windows, Linux and macOS operating systems. The desktop application will provide observers with a visual representation of the graph being visualized on the mobile device by the user.

2.2   Product Function

The system should allow for a predefined graph to be read into the graph visualizer. Once the graph has been loaded, it will be visualised either on a hyperplane or a hypersphere. When the user views the graph, he/she should be able to navigate through the graph by zooming in and out, following the edges between vertices and selecting various topics of interest on the graph. The viewer should be able to set various criteria which will be used for the

generation of meta-graphs, such as specifying force direction. The viewer should also be able to edit the graph visualization through the use of hand gestures and the system should provide functionality for the edited graph to be saved as a file in which the graph will be represented as a set of triples. The system should allow for observers to view the graph which is being viewed by the user from the users perspective and the visuals which the observers see should be updated accordingly as the users viewport changes.

2.3   User Characteristics

The typical user groups of the system can be categorised into two different user groups, namely educational professionals such as lecturers and researchers and undergraduate students. Educational professionals may use the graph visualizations to assist with educational activities and the visualizer will also aid researchers in their research activities. For undergraduate students, the graph visualizer will be a useful tool which will assist them in gaining a better understanding of graphs. All of the user groups must have a general understanding of graphs and their various applications. The user must also possess an understanding of certain graph-related concepts such as force direction. The user should possess basic computer literacy skills in order for them to be able to configure and operate the system and they should also be familiar with the environment on which the different components of the system will be deployed eg. Users who intend to make use of the desktop application on a Windows computer should have a general understanding of the Windows OS environment.

2.4   Constraints

The desktop component of the system should be available on multiple platforms, namely Windows, macOS and Linux. The mobile component of the system should be compatible with Android devices. The system itself should be modular to allow for additional functionality to be added as it becomes necessary.

2.5   Assumptions and Dependencies

The devices on which the system is deployed should contain the necessary hardware which is needed for the system to function. Specifically, the mobile device should have a mobile camera, a gyroscope and networking capabilities and the desktop should have networking capabilities and an adequate graphical processing unit. The user will also need to have access to a pair of virtual reality glasses in order to visualize the graphs in a 3D environment.

3   User Stories

The user requirements are derived from a number of user stories. This section serves as a reference to the constructed user stories and under which module they fall under.

3.1   Data Module

- As a User, I want to Save Graph for later use

- As a User, I want to Update Graph during interaction

- As a User, I want to Create Graph from a text file to create a model

3.2   Interaction Module

- As a User, I want to Select Vertex in order to move/edit

- As a User, I want to Create Vertex to add to the model

- As a User, I want to Update Vertex to a new value

- As a User, I want to Remove Vertex from the model

- As a User, I want to Create Edge add to the model

- As a User, I want to Update Edge to modify forces between vertices to observe structural changes

- As a User, I want to Remove Edge from the model

3.3   Rendering Module

- As a User, I want to Render VR Graph for viewing

- As a User, I want to Reload Graph to diaplay updated version

- As a User, I want to Undo Changes made to graph

- As a User, I want to Render Graph on External Display for 3rd party user to view

3.4   Synchronization Module

- As a User, I want to Publish State Changes to sync with external display

- As a User, I want to Receive State Changes to sync with VR display

4   Functional Product Requirements

Taking the the user stories in to consideration we will now create a set of product requirements. Each requirement we aim to meet within a sprint (a specified number of days).

4.1   Data Module

- Read graph data in all specified formats

- Format the data in such a way that it is easily interpreted for graph construction

- Take graph information and export it to a specified file format

4.2   Interaction Module

- Update graph information through graph interaction

- Provide undo functionality on graph

4.3   Rendering Module

- Take formatted graph information and construct a graph

- Provide real time rendering from user interactions

4.4   Synchronization Module

- Provide real time rendering on viewer based on VR view

- Push relevant changes to external viewers

5 Non-functional Product Requirements

The system should be able to respond to user actions in a timely manner to allow for smooth user interaction. This requirement is especially vital for user interactions with the graph visualization. There should be real time communication between the mobile application and the desktop application to allow observers to witness the users interactions with the graph in real time.

Extensibility: The system must be robust and modular, allowing for simple alterations to the current system and to cater for the future growth of the system by allowing additional features to be implemented and incorporated with ease.

Reliability: The system should be "failure-free" by able to process user input accurately and respond in the appropriate manner at all times.

Usability: The system should be user friendly.

6 Use Cases

6.1 Data Module

UC1.1 Save Graph

   i Description: The system must be able to save any changes which have been made to the graph by the user. Once the graph has been saved, the graph specification should be updated accordingly.

   ii Pre-condition: The graph visualization has been edited.

  iii Post-condition: The changes to the graph specification have been saved.

UC1.2 Update Graph

   i Description: The system should allow for the user to edit an existing graph specification.

  ii Pre-condition: The required graph specification is selected.

 iii Post-condition: The graph specification has been updated.

UC1.3 Create Graph

   i Description: The system should allow for the user to create a new graph specification to be rendered by the visualizer.

  ii Pre-condition: The option to create a new graph specification is selected.

 iii Post-condition: The graph specification has been created.
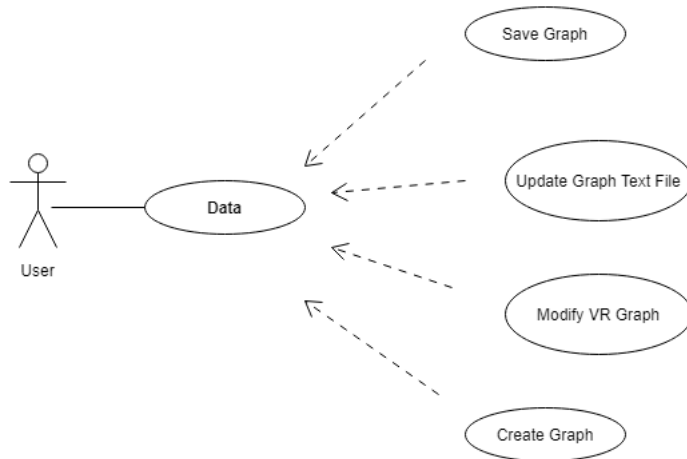
6.2   Interaction Module

UC2.1 Select Vertex

   i Description: The system should allow for the user to select a vertex to interact with through the graph visualization.

ii Pre-condition: The user must be viewing the graph in a 3D environment.

iii Post-condition: The desired vertex has been selected by the user.

UC2.2 Create Vertex

i Description: The system should allow for the user to create a new vertex and add it to the graph visualization.

ii Pre-condition: The user must be viewing the graph in a 3D environment.

iii Post-condition: A vertex has been added to the graph.

UC2.3 Update Vertex

i Description: The system should allow for the user to update any metadata related to a particular vertex.

ii Pre-condition: The user must be viewing the graph in a 3D environment.

iii Post-condition: The desired vertex has been updated.

UC2.4 Remove Vertex

i Description: The system should allow for the user to remove a vertex from the graph visualization.

ii Pre-condition: The user must be viewing the graph in a 3D environment.

iii Post-condition: A vertex has been removed from the graph.

UC2.5 Create Edge
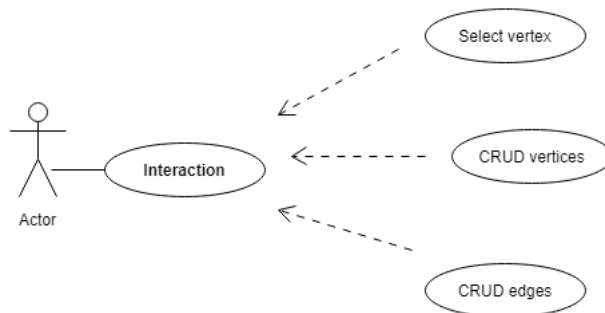
    i Description: The system should allow for the user to create a new edge and add it to the graph visualization.

    ii Pre-condition: The user must be viewing the graph in a 3D environment.

    iii Post-condition: A vertex has been added to the graph.

UC2.6 Update Edge

    i Description: The system should allow for the user to update any metadata related to a particular edge.

    ii Pre-condition: The user must be viewing the graph in a 3D environment.

    iii Post-condition: The desired vertex has been updated.

UC2.7 Remove Edge

    i Description: The system should allow for the user to remove an edge from the graph visualization.

    ii Pre-condition: The user must be viewing the graph in a 3D environment.

    iii Post-condition: A vertex has been removed from the graph.

6.3   Rendering Module

UC3.1 Render VR Graph

 i Description: The system should provide the functionality to render a graph visualization. The visualization will be constructed based on a graph specification provided by the user.

 ii Pre-condition: A graph specification must be provided by the user.

 iii Post-condition: A graph visualisation has been rendered by the system.

UC3.2 Reload Graph

 i Description: The system should allow for a graph visualisation to be re-rendered. The graph visualisation will be reconstructed based on the latest available graph specification.

 ii Pre-condition: The graph specification must be valid.

 iii Post-condition: The graph has been re-rendered.

UC3.3 Undo Changes

 i Description: The system should allow for the user to undo any changes which have been made to the graph visualization.

 ii Pre-condition: The user must be viewing the graph in a 3D environment.

 iii Post-condition: The graph has been reverted to a valid previous state.

UC3.4 Render Graph on External Display

 i Description: The system should provide the functionality to render a graph on an external display. The graph rendered on the external display will be constructed based on the graph which is rendered by the graph visualizer on the mobile device.

ii Pre-condition: A graph visualization should be rendered on the mobile device.

iii Post-condition: A graph has been rendered on the external display.
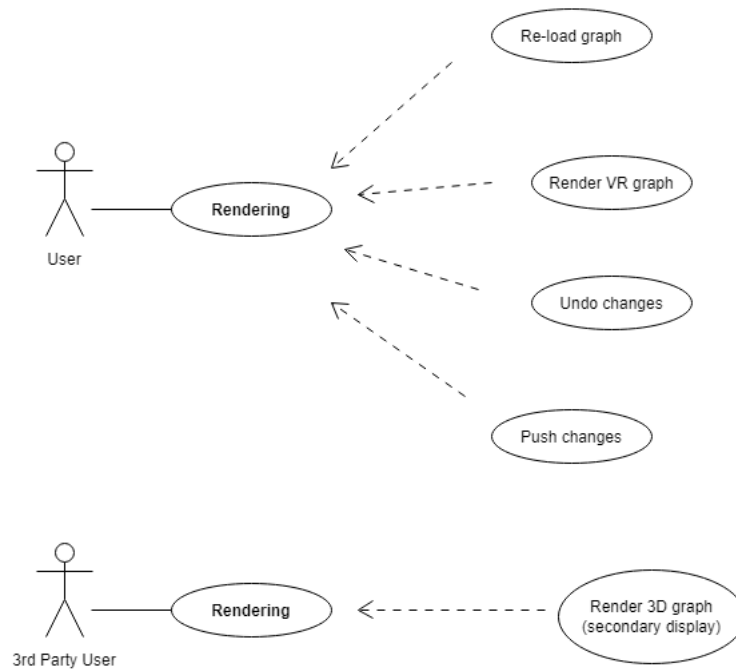
6.4 Synchronization Module

UC4.1 Publish State Changes

i Description: The system should allow for the mobile application to publish any graph state changes to the desktop application.

ii Pre-condition: The mobile application and the desktop applications should have a means of communication.

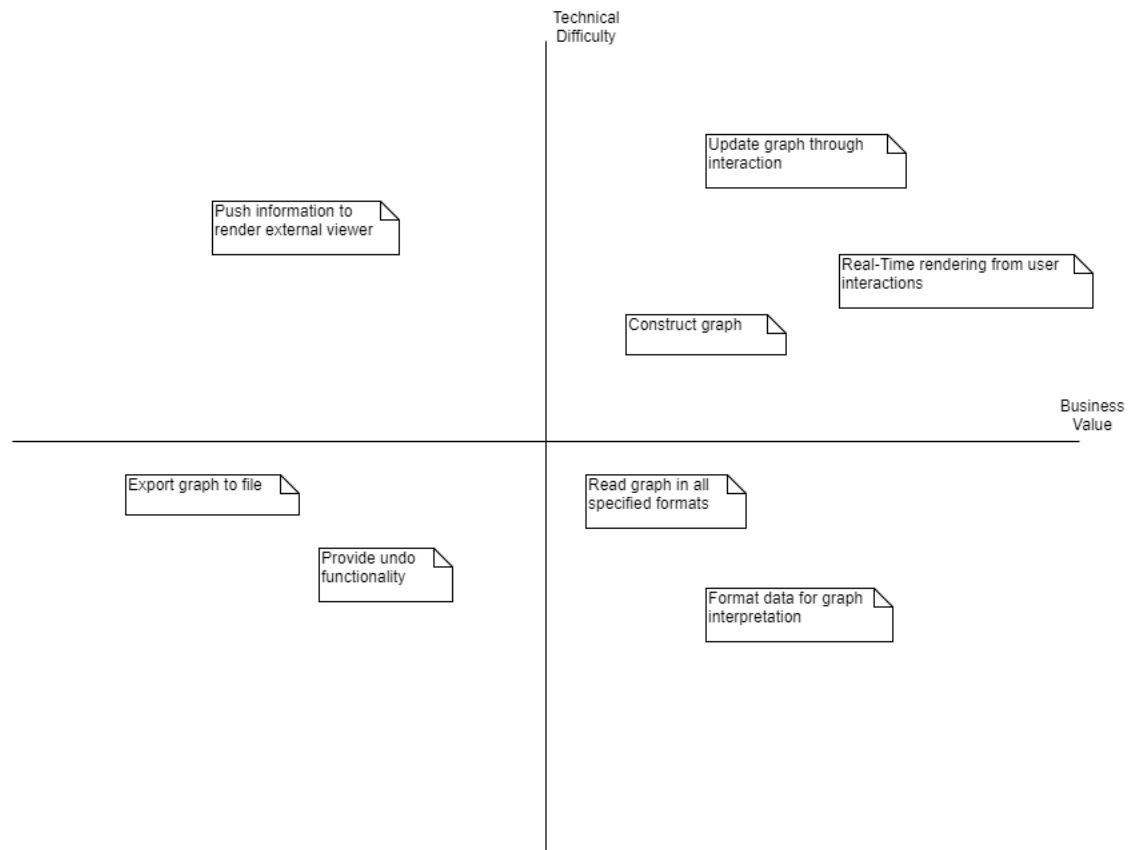iii Post-condition: The state changes have been pushed to the desktop application.

UC4.2 Receive State Changes

i Description: The system should allow for the desktop application to receive the graph state changes which have been published by the mobile application.

ii Pre-condition: The mobile application and the desktop applications should have a means of communication.

iii Post-condition: The state changes have been received by the desktop application and the graph on the external display has been updated accordingly.
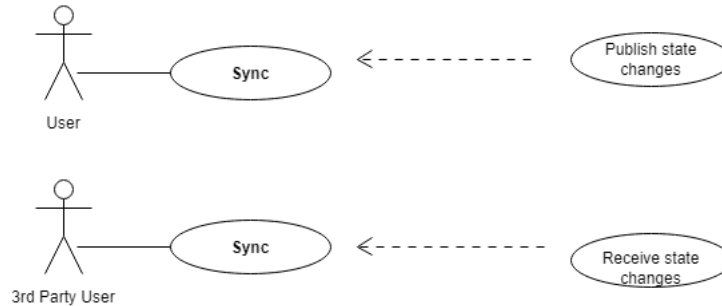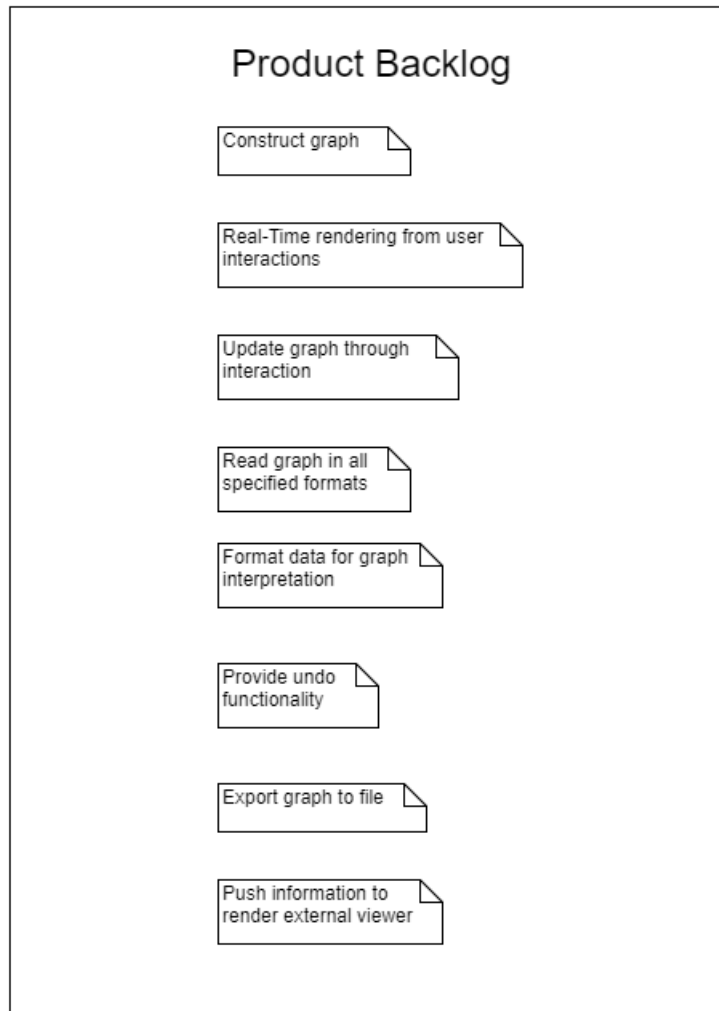

7 Business Value vs Technical Difficulty & Product Backlog

The product requirements are plotted on the following graph to illustrate how important they are for business success and how difficult each is to implement.

Technical Difficulty

Update graph through interaction

Push information to render external viewer

Real-Time rendering from user interactions

Construct graph

Business Value

Export graph to file

Read graph in all specified formats

Provide undo functionality

Format data for graph interpretation

According to the diagram and fulfilling as many prerequisites as possibly while maintaining concurrency we have the following implementation order on the product backlog.

Product Backlog

Construct graph

Real-Time rendering from user interactions

Update graph through interaction

Read graph in all specified formats

Format data for graph interpretation

Provide undo functionality

Export graph to file

Push information to render external viewer

We use the order of cards in the backlog as a guide to help us order our execution order. Taking into account other factors such as dependencies. The execution order are reflected in the burndown charts as well as how we conduct our sprints.