Immersive Learning using a virtual reality program

Table of Contents

1. Introduction

1.1 Non-linear learning

Adults learn most effectively when they form a connection between newly and previously acquired knowledge. This new knowledge can be acquired through linear or nonlinear learning. An example of linear learning is a traditional textbook, with which one is expected to first learn chapter one, before continuing to chapter two. Examples of non-linear learning are the mind map technique and hypermedia.

Mind mapping is a learning technique developed by Tony Buzan. [Buzan 2013] A mind map contains a central picture, representing the theme of the mind map. Different colored branches spring from the central picture, each containing a single word. Main branches can be divided in sub-branches with their own keywords. Finally, connections between different branches can be represented via arrows. The benefit of this technique is that it focuses on the relationship between various similar subjects, instead of describing the individual subjects in some arbitrary linear order. It is believed that this method is a more innate form of learning. [Eppler 2006]

Hypermedia is an extension of multimedia (such as text, pictures and video) on a computer, but with hyperlinks. Hyperlinks are references that can be clicked to visit other pages. This allows the user to travel from one subject to another based on his or her interests, instead of reading or viewing subjects in ordered chapters.

1.2 Project setup

Ordina is a Dutch ICT company interested in developing a learning tool that uses new psychological understanding about learning. The basic idea is to create a Virtual Reality learning environment in which users can interact with the subject manner. The goal is that this tool can complement or perhaps even replace standard textbook learning.

In order to achieve its goal, Ordina has partnered with the University of Utrecht to offer research internships for students of computer science, psychology and other studies. These interns are Maurits Barendrecht, Job Bergsma, Tycho Meyer and myself, Freek Kuethe.

Each of the interns had his own project with its own research goal, but all projects worked together to develop the same learning program. Maurits Barendrecht, a psychology student, helped to design the learning tool in its early stages and conducted a test study to test its effectiveness in improving learning. Job Bergsma improved the visual design of the learning tool for its later iterations. Tycho Meyer co-development the gamification part of the learning program (scoring, achievements etc.) and developed knowledge challenges.

The research goals of my project are presented in section 1.3.

1.3 Research goals

The research goal of this project was to investigate whether traditional textbook-learning could be replaced or complemented by a virtual reality learning program. Can any text, regardless of subject, be translated to an interactive virtual reality environment and what benefits does this provide?

Previous research shows that people learn best through a combination of seeing, hearing and doing. [Moore D. 1996] Based on this research, and the research on non-linear learning, we developed the idea for a new learning program. Users can learn information by exploring a graph, where each node is a subject that is related to other nodes via edges in the graph. Users can see the graph and select nodes to get further information.

Subsequently, we added some gamification elements, such as points and achievements. This allows users to see what they have already learned and test their memory by using challenges. Many textbooks have a question section at the end of each chapter, and we wanted a similar function. We theorized that awarding points and achievements for completing these challenges would help motivate our users to carry on learning.

Three research questions were formulated, that had to be answered to complete the main research goal and implement the proposed learning program.

1. How do you convert text into a format that can be used by our learning program?
2. How should the learning program visualize this information in a way that the user can easily comprehend? Does multimedia integration (sound, pictures, movies) significantly improve the user experience?
3. What kind of interaction works best for such a learning program? Can aspects of gamification be used to add to the learning experience?

1.4 Structure of this work

In chapter 2, related works are discussed. These related works provided guidelines and inspiration for the design of the learning program.

Chapter 3, 4 and 5 each answer one of the three research questions respectively. In chapter 3, we describe why we chose linked data as the data model for the learning program. In chapter 4 we present how this data is visualized. Multiple iterations of the program are shown. chapter 5 discusses the gamification elements that were added to the learning program.

While the design of the program is described in chapter 3, 4 and 5, chapter 6 goes into detail about the implementation of the program in Unity 3d.

Chapter 7 described the user test performed to test the design of the learning program and receive feedback and suggestions by users.

Finally, chapter 8 provides a discussion of the project; How the research goal was answered and what future work is suggested.

2. Related work

Most studies on virtual reality learning focus on “Virtual Reality Learning Environments” (VRLE). 3D virtual locations where the user can walk around. This project is unique in that it combines virtual reality learning with the concepts of hypermedia, graph visualization and gamification.

Because of this unique aspect, it is difficult to say how relevant previous findings on just one of these subjects are to this project. A literature study was done in all four of these subjects, to find the best practices within each.

2.1 Virtual Reality learning

Virtual Reality (VR) has been used for training purposes for quite some time, especially in the fields of aviation and medicine.[Lee and Wong, 2008] The benefits of VR are immersion and interactivity. Users can explore and interact with realistic environments or objects, such as the cockpit of an aircraft, a molecule or a virtual representation of a city.

We can distinguish two major types of VR; Non immersive and immersive VR. Non immersive VR is usually displayed on a screen and controlled via a mouse and keyboard. Immersive VR is usually displayed on multiple large screens or a head mounted display, such as the “Oculus Rift”. The goal of immersive VR is to put the user in the virtual environment, by shutting out outside stimuli. [Lee and Wong, 2008]

With both types of VR, the user is placed in a “Virtual Reality Learning Environment” (VRLE). Studies suggest that these VRLE's help education by doing three things: They allow users to see problems from multiple perspectives, they are a cost effective way of simulating situated learning and finally, they allow users to practice tasks in virtual reality which can help them perform tasks in real life. The effectiveness of a VRLE can be measured by the amount of skill or knowledge a user acquired, and how well that skill or knowledge can be transferred to real world settings. Flight- and surgical simulators have proven to be very successful in this regard. [Dede C, 2009].

Case studies by Huang H. et al and Freitas S. et al both performed user tests or VR learning tools. These studies aimed to help develop design principles for such VR applications. The case studies-included serious games, such as a triage training game and an infection prevention game for medical students.[Huang H. et al, 2010][Freitas S and Neumann T. 2009] In the study of Freitas S. et al, a model is used that they call the Exploratory Learning Model. In this model, learning happens through a cycle with five steps: Experience, Exploration, Reflection, Forming of Abstract Concepts and Testing. The authors suggest that VR can be used to allow instructors to create learning experiences for their students. [Freitas S and Neumann T. 2009]

They suggest a goal based approach for creating these learning experiences, that contains 7 components:

1. The learning goals – should be intrinsically motivating.
2. The mission – which can only be accomplished by using specific skills and knowledge.
3. The cover story – creates the need for the mission to be accomplished.
4. The role – the player as protagonist.
5. The scenario operations – the level design.
6. Resources (tools and resources available).
7. Feedback. Both negative and positive feedback is inherent and automatic

[Freitas S and Neumann T. 2009]

2.2 Hypermedia

Hypermedia is defined as an extension of multimedia, but with hyperlinks. It is a non-linear medium of graphics, audio, video and text.

Non-linear learning allows users to explore subjects on their own pace, starting with subjects in which they are most interested. However, non-linear learning may not be effective for everyone. A study by Sherry Chen, “A cognitive model for non–linear learning in hypermedia programs.” states that the effectiveness of non-linear learning is based heavily on the user's “cognitive style”. To put it simply, some people can get lost in hypermedia navigation. Sherry Chen suggests that designers of hypermedia should take this into account. [Chen S. 2002]
Suggestions include:

- give the user guidance, such as suggesting the next page to visit,
- Hide links to irrelevant pages until they become relevant.
- Annotating links, indicating to where a link leads before it has been clicked and whether that page has been visited before.

Another important aspect of hypermedia is obviously the media itself. For this project, we use a combination of text, audio and images, with the idea that video could be added in the future. This is called “Cue summation” or “Multi-channel communications”. According to a survey by Moore et al, studies show that stimulating multiple senses at once can improve learning. However, if used incorrectly, they can also lead to interference. To avoid this, the audio information must match with the visual information. [Moor D. 1996]

2.3 Linked data

“Linked data” is a method for storing and publishing information. It is related to the idea of a semantic web. In linked data, entities are stored along with their relationship to other entities. When viewing linked data in a browser, a user can jump from one entity to another entity using these relationships as hyperlinks. This is why linked data might be a usefull method for storing data for a hypermedia application. [Bizer et al 2009]

Linked data is explained in more detail in section 3.1. This section focuses on other applications that use linked data.

In “Linked data-the story so far”, Christian Bizer provides a small survey of developments with Linked data up until 2009. These include:

- The Linking Open data project, which is a grass-roots community, that published existing data on the web in the linked data format.
- Dbpedia, which is a project to translate wikipedia into the linked data format
- The development of linked data browsers, such as Tabulator and Marbles.
- Linked data publishing tools such as:
    - D2R server
    - Virtuoso Universal Server
    - Talis Platform
    - Pubby
    - Triplify
    - SparqPlug
    - QAI2LOD server
    - SOIC Exporters

One project that looks very similar to our project, is “Towards a Semantic Wiki Experience-Desktop Integration and Interactivity in WikSAR”, by David Aumueller and Sören Auer. WikSAR is a browser based semantic wiki. One unique feature was that it contained a graph view, which users could use to navigate. This is presented in figure 1.

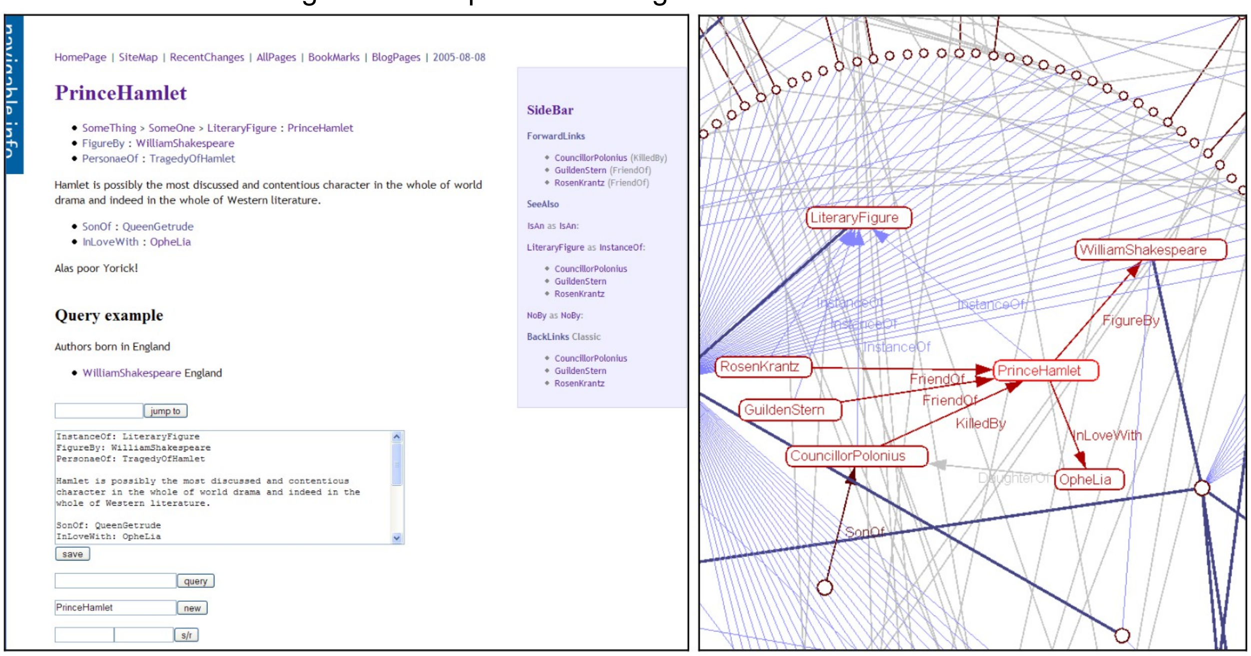


Figure 1: Interface of the WikSAR prototype with interactive graph [Aumueller, 2005]

2.4 Graph visualisation

We examined studies about “graph visualization” to answer part of the second research question: “how to best visualize information”. These studies were specifically relevant to the design of the overview screen, designed in 4.2.

Herman et al wrote a survey about various graph visualization tools, called "Graph Visualization and Navigation in Information Visualization: A Survey". [Herman et al, 2000] This survey inspired us strongly in the development of the overview screen, by discussing a wide variety of ways to display a graph. We chose an approach similar to the approaches of Peter Eader and Guy Melancon. Both of their approaches involved radially displaying a tree structure, while avoiding overlay between the branches. [Eades 1991] [Melancon 1998]

In “Applying graphical design techniques to graph visualization” Martyn Taylor and Peter Rodgers describe a tool they developed to automate graph visualization based on specific design principles. These principles have been shown to improve the readability of graphs. [Taylor and Rodgers 2005] Because these design principles are also useful for this project, they have been taken into consideration while designing the overview.
The principles are as follows:

- Avoid edge crossings
- Straight or at least uniformly bended edges
- (Create graphs within a good) Aspect Ratio
- Minimize total edge length
- Uniform edge length (minimize variance in edge length)
- Angular resolution (avoid small angles between edges)
- Symmetry
- Node separation (nodes should not be to close together to distinguish)
- Node clustering for close relationships

These principles were used in the overview screen, although some were better followed than others. Edge crossings do occur in the overview graph, but they are hidden, because only the edges connected to a highlighter node are shown. The nodes are shown in a circle, which provides both a good aspect ratio and symmetry. Finally, the algorithm in the last iteration of the overview groups nodes close together based on their relationship to one another, with the exception of some rare edge cases. More about the overview screen can be read in section 4.2.

2.5 Gamification in learning

Gamification is the use of gameplay elements for non game applications, to encourage behavior or make mundane activities more appealing. It is mostly used in marketing, but there is a push to promote its utility in educational settings as well. [Muntean 2011]

Gamification aims to combine intrinsic motivation with extrinsic motivation in order to raise supplementary motivation and engagement. Intrinsic motivation comes from within, when the user decides to perform an activity or not. Some examples are: altruism, competition, cooperation, sense of belonging, love or aggression. Extrinsic motivation, on the other hand, is when something or someone convinces the user to take action. [Muntean 2011]

Brian J. Fog created a model for persuasive design. With this model he suggests that in order to be persuaded to take action, one needs three factors: motivation, ability and a trigger. Motivation refers to the users motivation to do the task. This motivation can be intrinsic or extrinsic. Ability is the users ability to accomplish the task. Users will be less likely to perform a task they find difficult. However, a high amount of motivation might convince a user to try regardless. Finally, A trigger is needed, to press the user to do a task at a specific time. [Fog 2009]

Examples of motivational game mechanics include levels, points, badges and awards. These game mechanics work by triggering the intrinsic motivations of the user. This is illustrated in figure 2. [Muntean 2011]

Figure 2: The relationship between game mechanics and human desires [Bunchball.com]

“Gamification for Learning”, by Tu et al, describes a model for constructing gamification. They construct their model based on the fact that many users have different play and learning styles. Some prefer to learn through social interaction in a group, while others prefer to test their own skill by attempting difficult tasks alone or competing with others. The model suggests as follows: [Tu et al 2015]

- Goal Setting: Set clear learning goals
- Player Engagement: Adapt games based on the users play-style
- Progressive Design: Make the system easy to learn but hard to master

- Environment Building: Create a community where users can work together and show of their awards.

Additionally, they identify three possible pitfalls with gamification. Extrinsic rewards may replace intrinsic motivation, gamification in education can be time consuming for teachers and gamification might still be perceived as less serious by adult learners. [Tu et al 2015]

2.6 Relevance to this work

The literature on Virtual Reality Learning Environments suggests that VR can be a usefull tool for learning programs. Considering however that the virtual reality program that Ordina wishes to make differs drastically from the VRLE studied in most of the literature, it is unclear how relevant the findings of these studies are in this project.

The literature on Hypermedia is more clearly related to this project. The study by Sherry Chen suggests that it is important to annotate links and create a good overview, to avoid users getting lost. This is certainly something that we aimed to do in this project. Additionally, voice over was added for all the text in the program, based on the literature on cue summation, which suggested to us that a combination of reading and hearing text might improve learning.

Linked data was chosen as the content storage for the learning program. This was stored using the free version of Virtuoso Universal Server. Dbpedia and WikSAR were used as work examples.

The principles for graph visualisation were used in the overview screen of this project. The overview screen also uses a radial tree model as can be found in [Eades 1991] and [Melancon 1998].

A gamification mechanic was implemented in the learning program that rewards users with points for reading and completing challenges. Users can level up by achieving certain scores. The idea of creating a community where users can share rewards and can help each other with learning is something that could not be implemented in a prototype, but it is a feature that was kept in mind as an eventual goal.

3. Data model

In order to visualize information, the information must be stored in a way that allows it to be easily parsed or queried by the program. This requires a data model for storing the information in a database or a text file. It was decided to use the linked data method for this project. This method and its implementation in this project is described in this chapter.

We will explain why we chose linked data as the answer to the first research question: ““How do you convert text into something that can be used by our learning program?””.

3.1 Linked data

Linked data is method to structure and publish data, making it easier to link multiple databases together. It makes use of the Resource Description Framework (RDF), an entity-relationship data model. [Bizer et al 2009][duCharme 2013]

There are a number of reasons for the use of Linked Data in this project. It is generic enough to be applied on almost any type of information and structured enough to allow it to be parsed easily when generating a visualization. Additionally, the very structure of Linked Data resembles that of a mind map.

Linked data is an idea credited to Tim Berners-Lee, the inventor of the internet. The idea is that instead of a web of pages connected with hyperlinks, concepts themselves are connected via relations. Linked data has three different rules:

1. Use uniform resource identifiers (URI’s) to denote subjects, not just documents, such as persons, locations etc.
2. These URI’s can be visited to get some basic information about that subject.
3. These URI’s are connected via relationships, and when you visit one subject, you can see and link to all related URI’s.

The benefit of this is that multiple databases can easily be linked together and accessed via semantic queries. [duCharme 2013]

To store linked data, the RDF data model is used. In the RDF, data are stored in the form of triples. A triple in the RDF contains a subject, a predicate and an object, just like a simple sentence. “John - Married to - Jane” “John - Age - 31 “. In these triples, the subject and predicates are always URI’s. The object can be a URI or it can be a literal, such as a number or a string.

Triples with literals are often used to give some basic information about the subject, such as its name, birthday or a description. “John - Age - 31” would be a good example of this, where 31 is simply a numerical value.

Triples without literals denote a relationship between two subjects, such as “John - Married to - Jane”. In that example, both John and Jane are URI’s each with other relationships of their own. If there is also a triple: “Jane - Age - 29” we could then query for the age of the person John is married to. Querying is done using SPARQL (**S**imple **P**rotocol **A**nd **R**DF **Q**uery **L**anguage); a language similar to SQL except written for RDF data. [duCharme 2013]

A good example of linked data is DBpedia, a project carried out by the Free University of Berlin and the University of Leipzig, in collaboration with OpenLink Software, to extract raw data from Wikipedia and make it available in the form of linked data.

In this project, a large amount of information is subdivided into various smaller subjects the user can access and learn in any order. Linked data allows each of the subjects to be stored as a node in a graph. Each node could then be connected to related subject nodes and potentially even URI's from the web. This is why linked data was chosen.

3.2 File based system

Initially, the prototype used a simple file parser for data loading. A .net library called dotnetRDF was used to read and interpret RDF data that was stored in a .ttl file. This file was written manually in a format called n-triple, using notepad or a similar program. The dotnetRDF library allows the program to parse the .ttl file storing a graph at runtime in the program, which can quickly be accessed when needed.

The limitation of this system is that it relies on a person to manually write such a text file. While n-triple is a fairly simple to read format, this can become very labor intensive and error sensitive, especially with large collections of data. However, it an easy way to test the prototype. For later versions, a database was used to store the data.

3.3 Virtuoso database

Virtuoso universal server (shortened to virtuoso) is a database management system by Openlink Software, with support for RDF data. It has an open-source version that was used in this project. Using a virtuoso database has the advantage of allowing the data to be easily edited and for the client program to access multiple databases. This is far more practical than using text files.

The dotnetRDF library was used to communicate with the server. This library has classes to download entire graphs from the server, in a similar way as the parser used in the file based system. It also allows the program to query the database directly using SPARQL, loading only the result set.

Virtuoso was preferred over other database systems because it has an open source version and the dotNETRDF library has specific support for virtuoso. Most other databases do not support RDF data.

3.4 Alternative models

The decision to choose linked data as the data model was made early in the project. To explain why, we must first examine the alternative models that could have been chosen. We will discuss hypertext and relational database models.

3.4.1 Hypertext

Hypertext is text displayed on a computer with references called hyperlinks that allows the user to travel immediately to other pages. Since the goal of Ordina is to create a hypermedia learning program, using a hypertext application such as MediaWiki might be considered an easy and suitable choice. It would not require much work to convert text into hypertext.

The downside of this approach is that existing hypertext applications are mainly focussed on text, and not on other forms of multi-media. Because Ordina wishes to create a VR program, a text focus does not suite. Creating a more customizable data storage, that can handle multimedia and games, allows much more flexibility.

3.4.2 Relational database models

The most common type of database is the relational database, whereby data are stored in tables. Each entity type has its own table in which each entity is a row within that table. The entities have a primary key and multiple attributes. Entities can be related to one another using these keys.

The problem with relational databases is the rigidity of the design. Once a database has been designed, it is very difficult to add new attributes to entities or change the way entities are related. For example. if you want to add a picture to a specific entity, such as a subject within the learning program, all entities must be redesigned to be able to have pictures. In comparison, linked data simply allows you to make a new relationship between the subject entity and the picture entity. It is much more flexible.

3.5 Conclusion

One of the research questions was: “How do you convert text into something that can be used by our learning program?”.
We have chosen for the use of linked data, because the concept of linked data, matched well with our idea for the learning program. Instead of reading linearly through a book, a learner should be able to explore subjects and their relationship to other subjects. Because linked data, works by storing relationships in graph structures, this was quite an obvious choice.

The other advantage of linked data over relational databases, is that it is very flexible in the information that it can store. It is very easy to add a new attribute to a specific node or to create new types of relations between nodes. This makes modifying the database simple. In relational databases, you first need to decide what types of attributes and relationships you want to store, before you can store any content. This makes linked data easier to use when adding different datasets and more future proof.

Converting text into RDF data that could be used by our program still proved to be rather labour intensive, but when Ordina develops better database editor tools with a release version, it will be well realizable.

4. Visualisation & interaction

Once the information is loaded into the program, it is important for users to be able to examine it properly. In this chapter, the choices made when visualizing the data are explained. This answers the research question: “How should the learning program visualize this information in a way that is easy to understand for the user?”

First, the user must be able to view and read everything in virtual reality. This has both benefits as well as disadvantages. In a virtual 3d environment, the user is completely immersed, and is unlikely to be distracted from outside sources. However, making things readable through a head mounted display requires large text fonts, because a majority of such displays has a low resolution.

We decided to display information on a series of virtual screens in front of the user. To make all these screens readably in a 3d environment, the screens where placed at a fixed distance, directly around the viewpoint of the user, forming a virtual sphere. The screens are always rotated towards the user, so that the user never has to look at a screen from an angle.

Secondly, it is probably bad practice to present the user with too much information at once, especially because of the large text font required. The main idea behind this project is that the user learns, by navigating the linked data graph, instead of reading large amounts of text. Therefore, each node should only give a small and memorable piece of information on a single topic.

Finally, there has to be a way for the user to get an overview of all the topics. This allows the user to see what he has learned, to plan his further reading and to quickly navigate between topics.

With these considerations in mind, it was decided to create two different views: A Topic view would give the user all relevant information for a single Topic node. An overview view would only show the names of each node, and allow the user to make a selection, zooming into a topic view for a specific topic node.
More views were later added for the gamification aspect of this project, but these will be described in part 5.

This chapter will give an overview of the visualization aspect of this project. Each section will discuss one aspect and how it evolved through the different iterations of the program.

4.1 Topic view

As already discussed, the learning program displays information as a series of interlinked nodes. The Topic view is the view that the user sees when a specific node is selected. This view has to teach the user about a particular subject and also show all related nodes.

In the first iterations of the node view, the node displayed the subjects name, a description and, where applicable, an image. The values of these fields are stored as literals in the linked data network. This information was displayed on a simple blue screen in front of the user. Around this screen, the relationships with other nodes, were displayed on smaller “relationship screens”. For example, if the database contains triple “John - Married to - Jane” and John is the selected topic, a relationship screen will show “Married to Jane”. Clicking on this screens brings you to the “Jane” topic screen. Figure 3 shows a concept and screen-shot of this topic view with the main screen, surrounded by relationship screens.

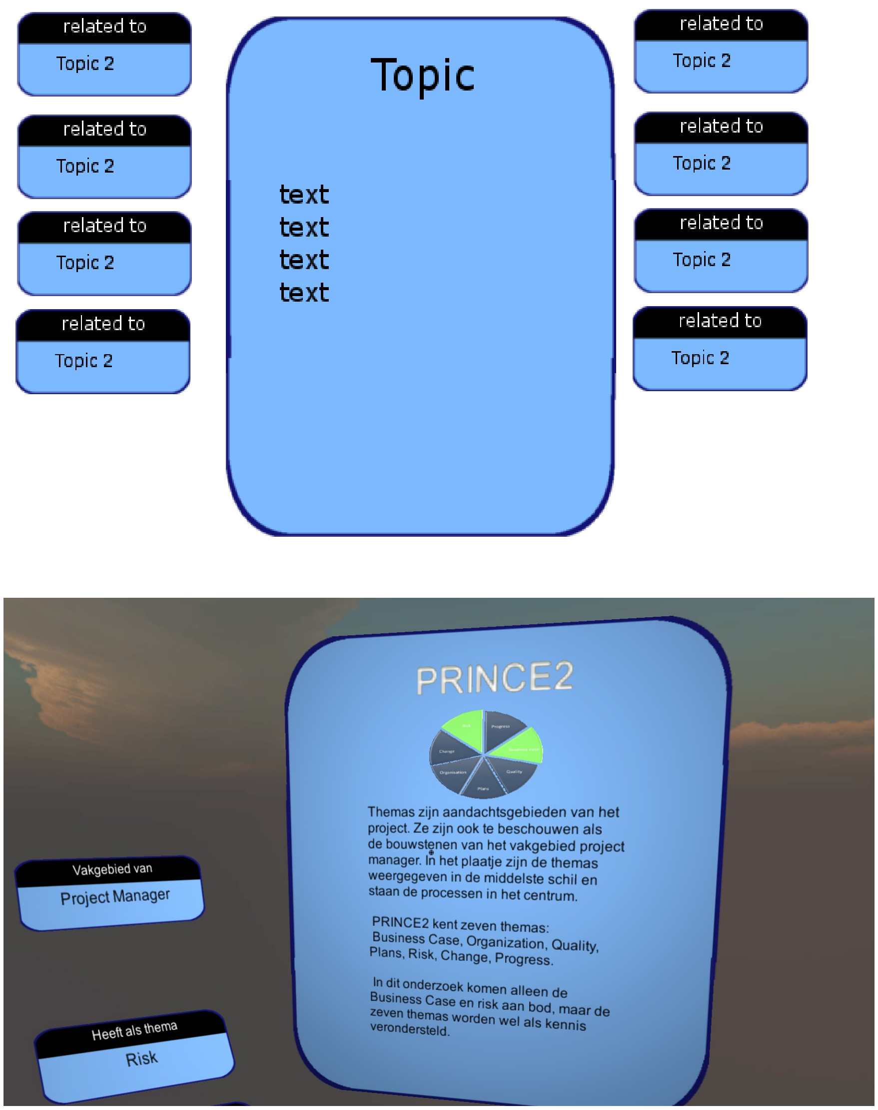


Figure 3: Concept and Screenshot of early iteration of the topic screen. On the sides of the topic screen, the relationships are shown in separate screens.

In these early iterations, user tests carrier out by Maurits Barendrecht revealed that users sometimes have difficulties reading the text or understanding that the relationship screens represented other nodes that they could visit. This revealed two possible limitations to the design. One limitation was that, due to the low resolution of VR headsets, texts needed to be short and displayed in a large font to be readable. Increasing the font size of the text is fairly easy. However, this means that whoever writes content for this application will need to create many nodes with little text per node, due to a small text limit.

The second limitation concerned the fact that this is a new idea for a learning program, many people need time to learn how to use the application. The application is not immediately intuitive.

To try and address these issues, later iterations included fewer texts, a cleaner design, the possibility to zoom in on pictures and a shared circular design between the relationship screens and the overview nodes. This cleaner look was designed by Job Bergsma, who also provided the new textures, which were used to implement the design in “Unity 3d”. Figure 4 shows this renewed design, which also includes some gamification elements that will be explained in part 5.

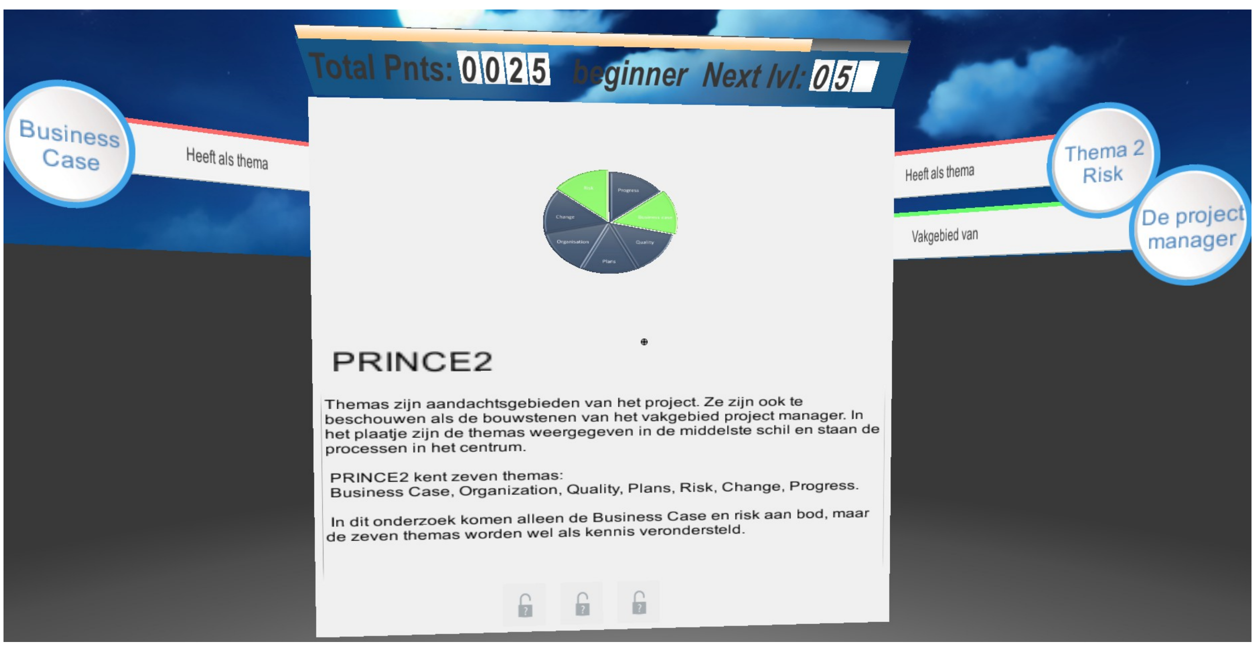


Figure 4: Later iteration of the topic screen. This also shows some gamefication elements, such as the score bar at the top.

4.2 Overview

The node overview screen must give the user a good overview of all the nodes. This overview is important for the user in order to remember what he or she has already learned and still needs to learn. Because the nodes are linked in a graph, it makes sense to depict this structure in the overview, allowing the user to see the relationship between all nodes. However, it is important to keep the overview clutter free. This graph structure must be visualized in a way that does not obstruct the view with a web of connecting lines.

4.2.1 First iteration

Because a quick prototype was needed for testing, the first iteration of the overview had little notice for the structure of the graph. We only looked at the amount of relations each node had. The reasoning was that nodes with many relations were more important. Using the dotNetRDF library, it was easy to query the amount of relations for each node. We sorted the nodes on the amount of relations. Next, we placed the nodes in concentric circles, with the most connected nodes in the centre and the least connected nodes in the outer circles.

Nodes with only one relation were not displayed at all. We decided that their singular relationship made them a sub-node to a "parent" node and that therefore these nodes were not worth displaying by default. Instead, these nodes where only shown once a user hovered over the "parent" node. The sub-nodes were then placed in a small circle around the "parent" node.

Permanently showing all connections as lines would result in an incomprehensible web of lines criss-crossing across the overview. Instead, connecting lines were only shown when selecting a node. When a user clicked once on a node, it showed the connecting lines of that node to other nodes, including any sub-nodes. When the selected node was clicked again, or when a node was double clicked, the node would be shown in the topic view.

The last node visited is coloured red, so that it is easy to return to that node. This iteration is presented in figure 5.

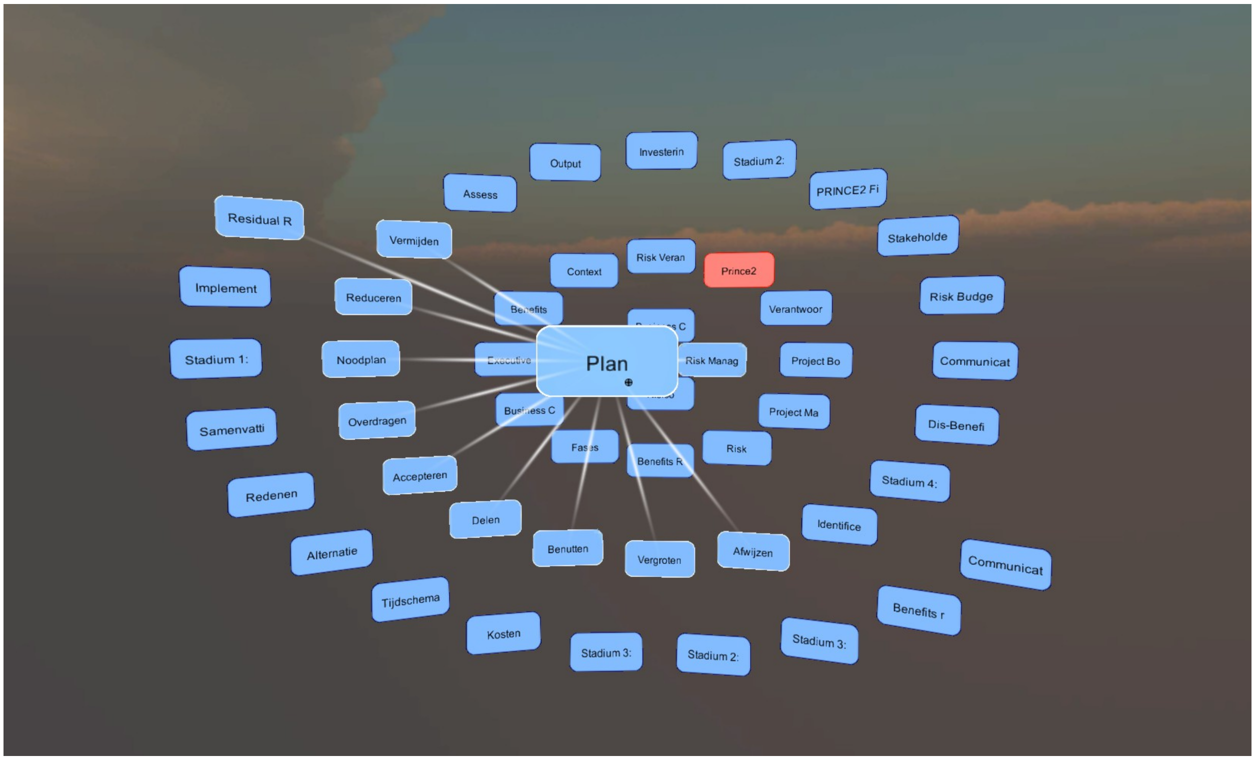


Figure 5: An early iteration of the overview screen, with the "Plan" topic selected. The white lines show the relationships with the "Plan" node.

In the first user test, the first iteration overview was not well received. Users did not understand how the nodes where ordered and placed, and hence could not easily find what nodes they needed to select next.

4.2.1 Second iteration

For the second iteration, we wanted to display the relationships between nodes in our overview, in a better way. This meant that nodes that are related in the graph, should be grouped closer together in the overview. Our reasoning was that this would make it easier for users in order to remember what subjects were connected, because they could remember their position in the overview screen.

We took inspiration from a mind map. A mind map is a diagram, with a single central topic drawn as an image in the middle. From this central topic, there can be several branches that can branch out further in a hierarchical way. [Eppler 2006]

We decided to put a central topic node in the middle of our overview and to place the other nodes around the central topic based on their distance from the central node in the graph. For this we used the Dijkstra path planning algorithm [Dijkstra 1959], with the minor exception of having no destination node. This created a radial tree, similar to the graph visualization described in "*Drawing free trees.*"by Peter Eades.

Our version of the Dijkstra algorithm first assigns the central node with distance value 0 and all other nodes with distance value infinity. It then creates a set of visited and unvisited nodes. Starting with the central node, each node is visited once. At each visit, all connected neighbouring nodes have their distance updated. If their distance value is higher than the current node's distance value plus one, the value is lowered to this new value. After all neighbouring nodes have been checked, the next unvisited node with the lowest distance is chosen to be the next current node.

With the Dijkstra algorithm, we have, for each node, the distance as well as the shortest route to the centre. By exclusively selecting the edges on these shortest paths, we create a tree graph, instead of the cycled graph that the linked data provides us.

In the overview, the main topic is placed in the centre. The location of the other nodes is then determined by a theta and a radius. To group related nodes together, we want sub-nodes in the tree graph, to be placed close to their roots. To accomplish this, we need to subdivide the space in the overview for each branch of the tree graph. This is done by reserving certain values of

Figure 6: Two screenshots of the later iteration of the overview view. At the top, "De project Manager" is selected. In the bottom "Business Case" is selected.

theta for each branch.
For the nodes with a direct relation to the central topic, meaning they have a distance value of 1, 360 degrees are divided into arc segments for each node. This division is done based on the number of total sub-nodes a particular branch has, which is calculated by using the Dijkstra algorithm to see how many paths from any node to the central node, run through that node. We call this value the “thickness” of the node, and it can be imagined as the thickness of a branch in the mind map. Each node on distance 1 gets an arc proportional to their thickness, with all the arcs totalling 360 degrees.

Subsequently, for nodes on farther distances, these arcs are subdivided again. Each node has a root or "parent" node, that is their shortest path to the centre. The arc of this root node is divided amongst the "child" nodes, proportional again to their thickness. This process gives every node an arc in which the node can be placed, which is always a part of the arc of the "parent" node. As a result, "child" and "parent" nodes will always have similar theta values.

As a result, we can place the nodes from low distance to high distance, making sure there is little wasted space and no overlap by adjusting the radius where needed. We use a simple collision check system, build into Unity 3d, to detect any overlap and then increment the radius.

To make clear what nodes have more relations, and hence might be more important, we adjust the size of the nodes. Nodes can have three different sizes, with the largest size meaning more relationships and the smallest size meaning only one relationship. The so called sub-nodes are now always shown, but are much smaller. We also colour the last node visited with a darker colour blue, instead of the colour red. This was a choice made by Job Bergsma.

Figure 6,clearly shows that related nodes are grouped close together. In the top image, the node “Project Manager” is selected. One can observe that the node is connected to the central “Prince2” node as well as to some more distant nodes at the right side of the overview. All other nodes have their opacity lowered, to make the connected nodes appear more clearly.

In the lower image, “Business case” is selected, and all its nodes are grouped together at the bottom of the overview. You can observe that some overlap exists, due to the cyclical structure of the linked data grapth. For example, “De Executive” is connected to both the “Business case” and the “Project Manager” node.

4.3 Voice cues

Early in the development, it was decided to add a voice-over to the text, to make the program more accessible. This meant that users would hear the text while reading it. The idea behind this is that multi-channel communication would improve a user's ability to learn and remember a text. [Moore, 1996] It also alleviates readability issues.

Voice cues were tied to where the user is looking. This meant that in the topic view, when looking towards the text, the text will start to be narrated. When looking away, the narration will pause until the user looks back. Additionally, when looking at the relationship screen, the relationship will be read like a sentence. For example: “The Prince2” subject node has a relationship called “has theme” with the subject “Risk”. When looking at this relation, the user hears: “Prince2 has theme Risk”.

We experimented with using a text to speech program, but we experienced difficulties in getting that to work on Dutch texts. Therefore the voice cues were voiced by Maurits Barendrecht.

4.4 Selection

The selection of nodes and objects has been mentioned several times during this article. Some clarification may be required on how this is done in virtual reality, when access to a mouse is not available.

A small cursor was added to the centre of the screen. Because the “Oculus Rift” uses stereoscopic 3d, this cursor has an actual location in 3d space, between the virtual camera point and the scene objects. This allows the user to use head movements to hover over various objects. To select an object, the user must simply look at an object and click a button. During tests, a wireless slide-show clicker was used.

The cursor icon in the centre of the screen can be clearly seen in figure 4, 5 and 6.

4.5 Conclusion

One of the goals of this project was to explore how to visualize information, so that users can easily browse and learn it. This project proposes one possible solution. The Topic view gives a small amount of text for each subject and the overview shows a graph structure which clearly shows the relationships between the subjects.

We discovered during development that reading text in virtual reality is difficult, especially when lower quality HMD's are used. To alleviate this problem, we implemented voice overs and a clear symbology. (The circular design for nodes, both in the topic and overview screens.)

The project also uses link annotation (the circles change colour when already visited and clearly show the subject name) and a clear overview to avoid users losing track. This is a risk factor for some users of hypermedia programs. [Chen S. 2002]

5. Gamification

This chapter focusses on the research question: “What kind of interaction works best for such a learning program? Do aspects of gamification add to the learning experience?”

As mentioned in the introduction, one element of this project was the idea of using gamification to stimulate motivation of learners. This resulted in two ideas: Knowledge challenge mini-games, that help the user to test his knowledge, and a progression system, that shows the user what he has learned so far and how much work is still left.

This progression system was the focus within this project, because designing good mini-games could be a whole project by itself. It was a project for another student intern called Tycho Meyer. Nevertheless, we will briefly discuss these mini-games, before moving on to the progression system.

5.1 Knowledge challenges

The mini-games, which we called knowledge challenges, are a similar concept as test sections in textbooks. They allow students to test what they have learned. Obviously, a VR environment might allow for more engaging and exciting alternatives then just a simple list of questions. However, the creation of complex mini-games was not included in the scope of this project. The project only focuses on the overall implementation of the scoring and progression system.

Tycho Meyer helped design the scoring system for the program and developed the first knowledge challenges, starting with a simple quiz game. After that, he focused on creating challenges that used multimedia and the virtual environment.

The challenges were tied to the topic nodes and made accessible in the topic view, so that users could play them directly after studying a certain topic. However, if there was more then one challenge on a certain topic, only the first challenge would be unlocked. The other challenge would be unlocked later. The reason for this was that users should have a motivation to return to previously studied topics to review them.

For each knowledge challenge, the user receives a grade based on his performance. Users can improve their grades by replaying the knowledge challenges.

5.2 Progression system

At first, the progression system only consisted of a score board. Reading each topic grants the user a small amount of points. Completing the knowledge challenges, grants even more points.

Later on, a levelling system was added to encourage users to progress. This would reward them with a new title (for example: beginner, advanced, expert etc.) every time they crossed a certain score threshold. The scoreboard was updated to show the users progress to the next level. This is shown in figure 7. On the left, the user has earned 25 points. In the middle his level is “beginner”. On the right, the user needs 5 points to progress to the next level. The orange bar at the top also shows his progress to the next level.

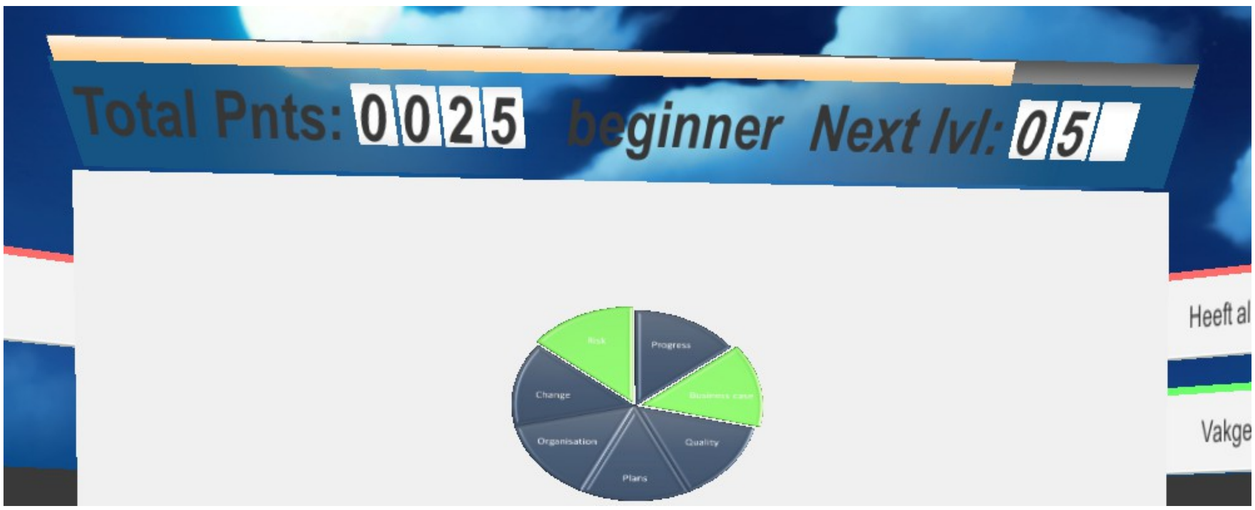


Figure 7: Score bar with orange progress meter.

Every time a new level is gained, locked knowledge challenges of previously visited nodes are unlocked. If a topic node has three or more challenges, the user will need to gain one level to unlock the second challenge, two levels to unlock the third etc. Once the user has gained the maximum level or the program detects that the user has no more points to gain, all challenges are unlocked immediately.

5.2.1 Showing progress in the overview screen

To encourage users to continue learning, it is important for them to see their progress in each topic and to see where new knowledge challenges have become available. That is why the overview screen, on which users can see all the topics, must also show the progress within each topic.

To accomplish this, a circular progress bar per topic was added to the design of the nodes in the overview screen. This bar changes from blue to orange, based on the percentage of the points earned within that nodes topic.

Figure 8: Overview Node with normal view on the left and blinking star to the right.

Additionally, to show whether a new knowledge challenge is unlocked, nodes can blink, alternating between their normal appearance and the appearance of a star. This blinking is meant to attract the attention of the user. The progress circle and blinking is represented in figure 8.

5.2.2 The progress screen

To show the progress of a user in more detail, a progress screen was created. This screen shows the users score, progress, the number of games he played and the average grade for all the games he played. It also shows the last topic nodes he visited and the three topic nodes with the highest average grade for the knowledge challenges.

Figure 9: The Progress Screen

Three other screens were envisioned alongside the progress screen, but were not fully implemented in the prototype. The other screens would be a player screen, that showed some information about the user, a friends screen, where the user could compare his score with the score of his friends, and a settings screen. Since these screens would require the implementation of an on-line account based system, which is outside the scope of this project, the screens were merely placeholders. The four screens can be accessed by the user at any time, by selecting the score bar.

5.3 Conclusion

The third research question of this project was: “What kind of interaction works best for the learning program that we developed? Do aspects of gamification add to the learning experience?” These questions cannot be fully answered until the prototype is properly tested, but the literature study suggests that gamification can improve the motivation of learners.

Based on the literature, as explained in section 2.5, a system of challenges, points and levels was implemented. This system also allowed the user to see his or her overall progress in the learning program. We think this will help motivate the learner.

Further development could encompass social tools that allow multiple students to work together in learning and show of their virtual rewards. This would elicit a student's desire for status, competition and socialization, and allow students to help each other.

6. Implementation

This chapter describes in more detail on how the program was implemented in Unity.

6.1 Unity 3d

The learning program was created in “Unity 3d” with “C#” as the programming language. “Unity 3d” is a game engine, developed by Unity Technologies, that is used to make games for PC, mobile devices, several game consoles and websites. Because there is a free personal use version available, it is a widely used game engine, with a large community around it. The choice for “Unity 3d” and “C#” allowed the convenient use of third party libraries and utilities; namely the “dotNetRDF” library and the “Oculus VR” utilities for “Unity 3d”.

A “Unity” game consists of different scenes, with each scene containing different game objects. Each game object has a position within the scene and can have multiple child game objects. These child game objects can also have child game objects, creating a hierarchy. Game objects can also have different components, such as text meshes and sprite renderers.
C# scripts can be attached to game objects to give them unique behaviour.

In the learning program, the content of each scene is dependent on the information the program receives from the database. This means that instead of using fully complete scenes, a method had to be found to create these scenes procedurally. In both the Overview and Topic scenes, a single main game object was used, that had a script that could create and place child game objects where needed. In this section, we will describe each scene in detail.

6.2 Start scene (Global game objects)

By default, game objects are tied to only one scene. To keep track of the users progress, there have to be game objects that exist throughout a session, regardless of the current scene. To load these objects, a start scene was created. Unity 3d contains a script that can be added to a game object to prevent it from being destroyed when a new scene is loaded. Using this script, the game objects in the start scene keep existing in the entire session. The start scene only has to be visited once, at the beginning of the session, to load these game objects.

These global game objects are: the viewHistory object, the Event System object, the main camera and the OVR camera rig. The OVR camera rig is part of the “Oculus VR” utility, that allows users to see the program via the “Oculus Rift”. The main camera is a replacement virtual camera that was used for testing, when the “Oculus Rift” was not available. This allows the program to be displayed on a standard monitor. The Event System was used for the game scenes.

The viewHistory object is an object used to contain all the scripts that need to exist throughout the session. It also contains, as a child game object, the score bar. In figure 10, the game objects can be seen in their hierarchy.

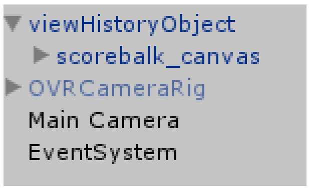


Figure 10: Start Scene Objects

The scripts contained in the "viewHistory" object are: The "viewHistory" script, which stores the users progress, the "ScoreBarSelect" script, that allows the user to select the scorebar and go to the progress screen and the "DBreader" script, which reads the virtuoso database and subsequently loads and stores the data.

The "ScoreBarSelect" script is a simple script that checks when the user selects the score bar and then loads the scene containing the progress screen.

6.2.1 Dbreader script

During the development, this script sometimes had different names as different versions were created, such as "database" or "sparql". The task of the script is to make contact with the virtuoso database and load all the required information.

The script uses the "dotnetRDF" library to download and store the information in a graphs. There are two graphs; one containing all the information and content for the topic nodes, and one containing all the information for the knowledge challenges. The script also contains a list to triples, which contain all triples with the currently selected URI. A search function fills this list of triples when provided with a specific URI. This is used when creating the content of the topic and overview screen.

The script also creates a list of all knowledge challenges, and a sorted list of levels and their required score. Both are then stored in the viewHistory script.

6.2.2 viewHistory script

The "viewHistory" script is designed to keep track of the users progress. At first, this only meant keeping track of the nodes the user had viewed. In later iterations of the program the users game scores had to be saved as well.
To perform the first tasks, the "viewHistory" script contains a list of visited nodes, in the order by which they were visited. This list, which is updated every time a new node is selected, allows other scripts to see what node is currently selected and what nodes were previously selected. This information is used in the Progress Screen for example, to show the five last visited nodes.

The “viewHistory” also keeps track of the users score, how many games he has played, and what level he is on.

Finally, the “viewHistory” also keeps a dictionary of “TopicScore” objects. The “TopicScore” class is created to store the information about the knowledge challenge games for each topics, such as the number of knowledge challenges, the total score points earn-able for that topic and how many score points the user has actually earned. The TotalScore class also contains a list of ChallengeScore objects, which contain the information for each challenge score, such as how many times it has been played, what the last grade was and how many points the challenge score is worth.

6.3 Topic scene

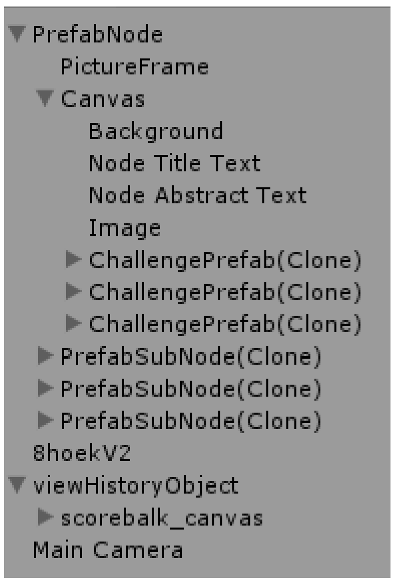


Figure 11: The Objects in the Topic Scene

After the start scene and the database has been loaded, a function within the viewHistory script automatically switches to the topic scene. The topic scene checks with the viewHistory script what scene is currently selected. After the start scene, this is always the central topic. This topic is also the centre of the overview scene.

As can be seen in Figure 11, the Topic scene contains two new objects: “PrefabNode” and “8HoekV2”. 8HoekV2 is merely a background for the Topic scene, so we will only discuss PrefabNode.
PrefabNode is the main object in the Topic scene, where all the information is displayed. It has a “canvas” object as a child object, which contains text and picture fields. Additionally, it can have multiple “prefabSubNode” objects, which represent the relationship of the topic with other topics. The canvas can also contain multiple “challengePrefab” objects; one for each knowledge challenge. These “challengePrefab” objects are buttons, that transition the program to a gamescene, when unlocked and pressed.

The prefabNode object has two important scripts: The NodeContent script and the MouseSelect script. We will also discuss the prefabSubNode.

6.3.1 NodeContent script

As mentioned in the beginning of this section, this program required scenes to be created dynamically, based on the content of the database. The NodeContent script is responsible for creating many of the objects within the TopicView scene and filling them with content.

After the viewHistory script has finished reading the database, the NodeContent script can check all the triples related to the currently selected Topic. These triples are kept in a list, by the viewHistory script. The NodeContent script reads these Triples one by one. It checks if the Tripples contain the topics title, its description or a filepath to its picture or audiofile. These are then stored and displayed in a corresponding Unity component. If the Tripple instead denotes a relationship with another topic, the NodeContent script will create a prefabSubNode object as a child of the PrefabNode.

When all Topic information and sub-nodes are completed, the NodeContent script looks at the TopicScore of the topic, which is also kept by the viewHistory object. With this information, it can create the knowledge challenge buttons and adjust these to be locked or unlocked.

6.3.2 MouseSelect script

The MouseSelect script is responsible for checking if the cursor in the centre of the screen is pointing towards a sub-node and for changing the selected Topic when a user selects a subNode.

When the user hovers over a sub-node, it slightly changes color and the audio voice over changes. The MouseSelect script interacts with a script attached to the sub-node called SubNodeColor to achieve the desired color change. It subsequently loads and plays the appropriate audiofiles as described in 4.3.

When a sub-node is selected, the MouseSelect script is responsible for telling the viewHistory object about the selection. Next, the MouseSelect script instantiates a new PrefabNode object. Finally, it destroys the old prefabNode object and its children. This will obviously also lead to the scripts own destruction.
When the viewHistory object is finished loading the new data for the selection, the new NodeContent script wil begin displaying the content in the new PrefabNode object and creating new sub-nodes.

6.3.3 Other scripts

Other scripts of note are:

- The subNodeColor script, attached to the sub-node objects, which handles the appearance of the subNodes. Amongst the tasks of this script is ensuring that the last visited node is coloured with a darker shade of blue.
- The start game script, that is attached to the challenge button objects, which transitions the program to a Game Scene when the button is pressed and unlocked.

6.4 Overview scene

The implementation of the Overview Scene is very similar to the Topic Scene. There is a single main object, called “Overview”, with an attached script called OverviewContent. This script creates “OverviewNode” objects as children of the “Overview” object.
There is also a script, similar to “MouseSelect”, called “OverviewSelect”. This script manages checks when the user is hovering over a certain overviewNode, and transitions back to the Topic Scene when a node is selected.

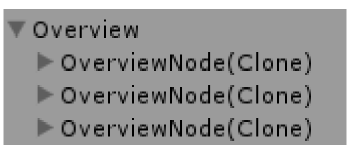


Figure 12: The Overview object, with OverviewNode objects as children

6.4.1 OverviewContent script

The way in which the Overview scene is created is already discussed in 4.2.1. All this happens in the OverviewContent script.
First, an OverviewNode is instantiated for each topic. Each OverviewNode has a script called “OverviewNodeScript”. This script has as one of its functions to store the distance, thickness and theta values.
Next, the Dijkstra algorithm is run, assigning each OverviewNode a distance value, from the central topic node. This is also when the nodes size is adjusted, based on how connected the node is.
Subsequently, each node is visited a second time to assign each node a root node and a thickness value, based on how many paths run through the node.
On a third pass, each node is assigned a theta value arc. The central node is given an arc from 0 to 360 degrees. Each other node gets a portion of the arc of its root node, proportional to its thickness. Each node is given a theta value in the centre of its arc.
Finally, the nodes are placed, in order of lowest distance value to highest distance value. They are placed in a circle based on their theta value and an incrementing radius, so that they do not overlap with previously placed nodes.

6.4.2 Interaction scripts

Once the overview is created, the interaction with the overview is facilitated by three scripts: The SubNodeColor script and the OverviewNodeScript script, both attached to the OverviewNode objects, and the OverviewSelect script, attached to the Overview object.

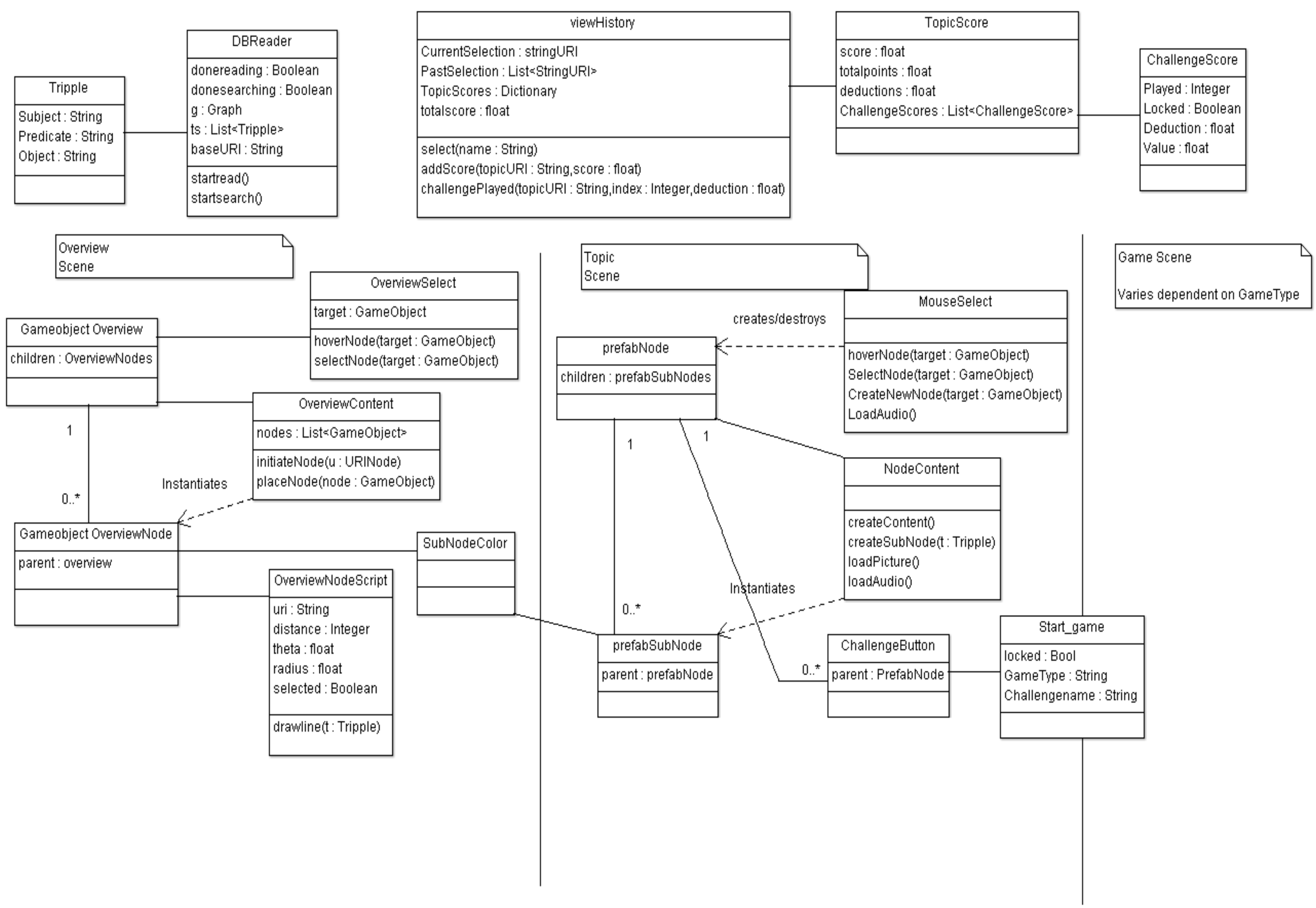
When a user hovers over an OverviewNode, the OverviewSelect script calls the SubNodeColor and OverviewNodeScript scripts. The OverviewNodeScript changes the way the name of the Node is displayed, placing it in a larger text-field. This makes smaller nodes, with large titles, readable to the user, by allowing the user to hover over them.
When a node is selected once, that node and all connected nodes have their titles displayed in the same way as when the user hovers over them. In addition, lines are drawn between the node and the connected nodes and all unconnected nodes are made slightly transparent. This is also done by the OverviewNodeScript, having been informed by the OverviewSelect script.
When a selected node is clicked or if a node is double clicked, the OverviewSelect will communicate this with the viewHistory script and the program transitions to the Topic Scene.

6.5 Other scenes and UML diagram

There are some scenes not yet discussed. First is the “Avatar” Scene. This is the Scene that contains the Progress screen, described in section 5.2.2 as well as several unfinished screens. This scene consists entirely of UI elements, with scripts that collect and display information from the viewHistory script.

Additionally, the game scenes have not been discussed, since the knowledge challenge games fall outside the scope of this project.

The next page shows an UML diagram of the Overview and Topic Scenes, with the global scripts and their related C# objects displayed at the top.



Tripple
Subject : String
Predicate : String
Object : String
DBReader
donereading : Boolean
donesearching : Boolean
g : Graph
ts : List<Tripple>
baseURI : String
startread()
startsearch()
viewHistory
CurrentSelection : stringURI
PastSelection : List<StringURI>
TopicScores : Dictionary
totalscore : float
select(name : String)
addScore(topicURI : String,score : float)
challengePlayed(topicURI : String,index : Integer,deduction : float)
TopicScore
score : float
totalpoints : float
deductions : float
ChallengeScores : List<ChallengeScore>
ChallengeScore
Played : Integer
Locked : Boolean
Deduction : float
Value : float
Overview
Scene
OverviewSelect
target : GameObject
hoverNode(target : GameObject)
selectNode(target : GameObject)
Gameobject Overview
children : OverviewNodes
OverviewContent
nodes : List<GameObject>
initiateNode(u : URINode)
placeNode(node : GameObject)
1
0..*
Instantiates
Gameobject OverviewNode
parent : overview
SubNodeColor
OverviewNodeScript
uri : String
distance : Integer
theta : float
radius : float
selected : Boolean
drawline(t : Tripple)
Topic
Scene
MouseSelect
hoverNode(target : GameObject)
SelectNode(target : GameObject)
CreateNewNode(target : GameObject)
LoadAudio()
creates/destroys
prefabNode
children : prefabSubNodes
1
1
NodeContent
createContent()
createSubNode(t : Tripple)
loadPicture()
loadAudio()
Instantiates
0..*
prefabSubNode
parent : prefabNode
0..*
ChallengeButton
parent : PrefabNode
Start_game
locked : Bool
GameType : String
Challengename : String
Game Scene
Varies dependent on GameType

7. User Test

During the process of making the learning program, many design choices were made to improve the usability. A user-test was required to test whether or not these choices had the desired effect. This user-test also had the goal of testing the overall functionality of the program and to gather suggestions for future improvements.

The user-test had 20 participants; ten men and ten women with ages ranging from 26 to 68. The concept was that each user would test the learning program for approximately 10 minutes, before answering a questionnaire about their impressions.

7.1 Goals

The user-test had two goals. The first goal was to test the overall experience of users and to get suggestions for future improvements. The second goal was to verify several assumptions made during the design of the program. Namely:
1. The voice-over reading of the text will assist in learning, by presenting an additional mode of sensory input. This might also help with readability in VR, which was an issue we faced during development.
2. The score system helps motivate the users to continue learning.
3. The score system as it is implemented in our program also helps the user keep track of what he has visited and not yet visited, due to the score based orange circles filling up.

To test these assumptions, two versions of the learning program were made. In one version, which was called the control version, two features of the program were disabled; the voice-over and the score system. Half the participants were randomly assigned to test this control version. Men and women were assigned separately to ensure both groups had 5 of each. The participants were not told that there were two different versions and believed they were just giving their opinion about a single version of the program.

7.2 Questions

After both test-groups had tested their version of the program, they were given a questionnaire. This questionnaire had two sections. A Likert-scale [Likert 1932] section, in which participants were asked how much they agreed or disagreed with nine statements, and an open section where testers were asked to write down their impressions and suggestions.

The first section had the following statements for testers to agree or disagree with. The statements were translated from Dutch.

1. *The text was easy to read.*
2. *The program held my interest. During the test I was always motivated to continue learning.*
3. *It was easy to navigate from topic to topic.*
4. *In the "overview" screen, it was clearly displayed which topics were connected to one-another.*
5. *It was easy to keep track of which topics I had already visited and which I still had to visit.*
6. *During the test, I experienced nausea or headaches.*
7. *I can remember most of what I have read.*
8. *Visually, the program looks nice.*
9. *I think the program is a helpful learning aid.*

Testers had the following options for each question:
Agree completely, Agree, Unsure, Disagree, Disagree completely
To prevent bias in the questionnaire itself, a second version was created with the 5 options flipped in order. This alternate version was handed out to half the participants in each group.

A Likert-scale was chosen, because this allows a statistical comparison of both groups, using the Fisher's exact test. [Likert 1932][Fisher 1922] However, due to the small sample size of participants, significant results were unlikely. In 7.3 the results indeed show no significant variation between the groups. The Fisher's exact test was chosen because it is accurate even for small sample sizes.

The questions 1, 2 and 5 were specifically chosen to test the assumptions described in 7.1. All the other questions were included to test the basic functionality.

The open section of the questionnaire asked the participants to write about four things:
A. Your general impression of the program.
B. Specific problems during the test.
C. Suggestions to improve the program.
D. Other comments.

7.3 Results

In this section, the Likert-scale results are shown and discussed. The results for each question are shown in a result table, along with the calculated P-value. The p-value is a two-tailed p-

value. The p-value represents the chance that, given the null hypothesis, we happen to select two groups whose answers differ as much or or more as the observed differences.
The null hypothesis is that both groups answer the questions with the same distribution of probabilities, based on the observed totals. For example, in question one, as can be seen in result table 1, the totals are 3, 15,1,1,0. By dividing by 20 we get the null hypothesis that 15% of all participants will answer *Agree Completely,* 75% *Agree* etc. To reject the null hypothesis, a p-value of 0.05 or lower must be observed.

The results form question 1 (*The text was easy to read*), show no significant difference between both groups. The p-value is 1. We can speculate whether this means that the text is properly readable even without the voice-over. The results show only one tester disagreeing with this statement.

Question 1	**Agree C**	**Agree**	**Unsure**	**Disagree**	**Disagree C**
test	2	7		1	
control	1	8	1		
total	3	15	1	1	0

Result_Table 1: The text was easy to read. ***P-value****: 1*

The results for question 2 also show no significant difference between the two groups. 75% percent agree that the program held their interest during the test, with the remaining 25% being unsure.

Question 2	**Agree C**	**Agree**	**Unsure**	**Disagree**	**Disagree C**
test	3	4	3		
control	2	6	2		
total	5	10	5	0	0

Result_Table 2: The program held my interest. During the test I was always motivated to continue learning. ***P-value****: 0.727*

The results for question 3 (*It was easy to navigate from topic to topic.*) show a relatively large amount of variation between the answers, with 40% agreeing completely with the statement and 10% disagreeing. The features disabled in the control version do not seem to help with navigation, since both participants disagreeing with the statement tested the test version. It is possible that navigating in the program takes some time getting used to, with some people picking it up quicker than others. Overall 75% found it easy to navigate.

Question 3	**Agree C**	**Agree**	**Unsure**	**Disagree**	**Disagree C**
test	4	3	1	2	
control	4	4	2		
total	8	7	3	2	0

Result_Table 3: It was easy to navigate from topic to topic. ***P-value****: 0.714*

Question 4 is the first of three questions where we see a p-value that comes close to the standard of 0.05 or lower. The biggest difference is that 3 participants of the control group agreed completely, while most of the test-group participants merely agreed, with none agreeing completely. It is difficult to imagine the features enabled in the full test version hindering the way connectivity is displayed in the overview screen. A more likely explanation would be that the control group happened to be slightly more positive.

Overall, this results show that the overview screen could use more work, with 35% unsure and 10% disagreeing with the statement.

Question 4	Agree C	Agree	Unsure	Disagree	Disagree C
test		6	3	1	
control	3	2	4	1	
total	3	8	7	2	0

Result_Table 4: In the "overview" screen, it was clearly displayed which topics were connected to one-another. ***P-value****: 0.142*

Question 5 has the lowest p-value of all the questions with 0.086. This was one of the questions were a difference between test and control was expected, given the assumption that the score system would also aid in keeping track of visited nodes. 50% of the test participants found it easy to keep track of topics, while none of the control group did. While the p-value is not below 0.05 this does seem like a strong indication. A larger test could possibly provide a more significant result.

However, even assuming that the score system helps, it appears as if more can be done to help users keep track of visited nodes, with 50% of the test group unsure or disagreeing.

Question 5	Agree C	Agree	Unsure	Disagree	Disagree C
test	2	3	3	2	
control			7	3	
total	2	3	10	5	0

Result_Table 5: It was easy to keep track of which topics I had already visited and which I still had to visit. ***P-value****: 0.086*

The results for question 6 were unexpected. Overall, most people didn't experience nausea, with only one person agreeing, one person unsure, 2 disagreeing and the rest completely disagreeing. Strangely, all participants who didn't disagree completely are in the control group, giving us a p-value of 0.087 , similar to that of question 5.

It was foreseen that nausea could be an issue in a VR application, but it was not expected that the features enabled in the test version could help with nausea. There are studies that suggest that sound (in particular music) can help with motion sickness. [Keshavarz, 2014] This means the voice over enabled in the test version could have had a positive effect. There is also a 9% chance that this is a random result.

Question 6	Agree C	Agree	Unsure	Disagree	Disagree C
test					10
control	1		1	2	6
total	1	0	1	2	16

Result_Table 6: During the test, I experienced nausea or headaches. ***P-value****: 0.087*

Question 7 shows that most people are unsure that they can remember what they have read, with nobody agreeing or disagreeing completely. The test features had no significant effect on this.

Question 7	Agree C	Agree	Unsure	Disagree	Disagree C
test		4	2	4	
control		4	4	2	
total	0	8	6	6	0

Result_Table 7: I can remember most of what I have read. ***P-value****: 0.485*

Question 8 and 9 show that most people were positive about the visual look of the program, as well as the potential as a learning program, regardless of which version they tested.

Question 8	Agree C	Agree	Unsure	Disagree	Disagree C
test	1	8	1		
control	2	7	1		
total	3	15	2	0	0

Result_Table 8: Visually, the program looks nice. ***P-value****: 1*

Question 9	Agree C	Agree	Unsure	Disagree	Disagree C
test	3	6	1		
control	2	6	2		
total	5	12	3	0	0

Result_Table 9: I think the program is a helpful learning aid. ***P-value****: 1*

7.4 Open questions

Not all participants responded to the open questions, but those that did, gave some valuable answers and suggestions.

All people give a positive general impression of the program, saying it was both informative and engaging, although some participants mention some issues with getting used to the program when starting. Several participants had no problem with the program once they figured out how the program worked, but said they would have liked to have better instructions before starting.

Another issue was the readability of the text. For some people this could be resolved by adjusting the Oculus Rift headset, but others experienced difficulty reading throughout the test.

Several participants mentioned some technical issues with the Oculus Rift headset. This was likely caused by the fact that not enough time was taken to properly configure the headset for each user. This resulted in several people experiencing blurriness in the beginning of the test and requiring assistance to resolve the issues.

There were also comments about the structure of the information. Several participants found the overview screen a bit chaotic and would have preferred fewer nodes or a recommended order in nodes. One particular issue is when a topic gives a list of sub-topics, these sub-topics are not ordered or numbered. Perhaps it would be a good idea to include a next topic function for such sub-topics. One participant also would have liked a “back” button.

Many participants mentioned the voice-over audio as monotone and redundant. They recommend making this voice-over optional.

Finally, many participants suggested more pictures, animations and sounds to make the program more engaging and informative without relying almost exclusively on text.

7.5 Conclusion

Overall, the small scale of the user-test meant that no results were statistically significant, when comparing the control- and test-groups. However, two results did come close. In combination with the open-question results, the results are still valuable for improving the program.

The first goal of the user-test was to test the overall impression of users and to get suggestions. The results show that the overall impression is positive. The open question section has also given several suggestions for improvement: more structure in the overview screen, the ability to turn voice-over off, more images instead of text.

The second goal of the user-test was to test certain assumptions made during the design and development of the program:
1. The voice-over reading of the text will assist in learning, by presenting an additional mode of sensory input. This might also help with readability in VR, which was an issue we faced during development.
2. The score system helps motivate the users to continue learning.
3. The score system as it is implemented in our program also helps the user keep track of what he has visited and not yet visited, due to the score based orange circles filling up.

The results from question 1 and the open question seem to prove the first assumption false. Most people did not find the voice-over helpful.

The second assumption is unclear. Most people find the program engaging, regardless of which version they tested. The small scope of the user-test and the limited implementation of the gamification aspect of the program may have contributed to the fact that no significant difference can be found between the control and test versions.

The third assumption does appear true, although again, due to the limited scope of the user-test the results do not meet the usual standards for significance. The result of question 5 has a p-value of 0.86, which is higher than 0.05.

Finally, we also have an unexpected result; audio may help with motion sickness. While this result is also unclear due to the small scale of the user-test, this result is similar to other studies. [Keshavarz, 2014]

8. Discussion

The main research goal was: “ to investigate whether traditional learning via a textbook could be replaced or complemented by a virtual reality learning program. Can any text, regardless of the subject, be translated to an interactive virtual reality environment and what benefits does this provide?”

While one single project cannot give a definitive answer to all these questions, our project focussed on creating a solution to visualize information in a virtual learning program.

We found that linked data is a good method to structure information for a generic learning program. The advantage of linked data is that it works well for any subject, where relational databases need to be specifically designed for a specific application. Additionally, linked data is a great choice for any hypermedia application, because hyperlinks can be modelled easily as relationships in the RDF.

Naturally, our solution to visualize the information is just one of many. We experienced that visualizing text in Virtual reality is difficult. We tried to solve this problem by displaying a lot of information through the relationships between nodes, instead of just text within the nodes. These relationships can then be clearly seen by users through the overview screen. In the topic screen, we added pictures and voice-over audio. It would further improve the program if more audio and video is used to replace the text in the topic view nodes. This is also something that was suggested in the user test.

At the conclusion of this project, the gamification elements of the learning program were not yet fully developed. Based on the literature study described in 2.5, we assume that these elements would benefit the learning program, by motivating the users.

A small user test was done to test our prototype. While the overall impressions were positive, more work can be done in creating a clear overview and replacing text with images and games.

The learning program that we created has the benefit that it can be modified, by providing custom RDF data, in order learn any subject. This is a benefit it has over other VRLE, that are specifically created for a single subject. This means that any text can be converted to work with our program. In this respect our project certainly succeeded in answering the question: “Can any text, regardless of subject, be translated to a interactive virtual reality environment?”

The learning program is only one possible solution and has not been fully tested. This is a limitation of our project. It is therefore not possible to claim whether it is the best solution to the research goal. The benefits of using this program as a substitute of textbook learning are also uncertain, pending proper user tests.

8.1 Future work

Improvements can be made to both the programs presentation as well as the gamification aspects. The usability of the program will probably improve when more audio and video is used to replace text. However, this could also make it more difficult to create content for the program. The overview screen could also use more structure.

The user test showed that certain participants wanted more guidance on in what order they should visit nodes. While this somewhat contradicts our goal of creating a non-linear learning experience, it may be an option to include a recommended topic system, that could guide users through topics in a logical order.

The gamification part of the program could benefit from a social component. This would make received awards more meaningful and it would allow users to learn together. A social component would require an online account based system. Within such a system, you could implement friends lists as well as leaderboards, permitting users to view the progress of other users, and compare their scores. This would add a competitive element to the learning program and help create a social environment around the program ("Environment building") as recommended in "Gamification for Learning", by Tu et al.

Finally, once the gamification aspects are better developed, a larger user test would be required to test its effectiveness.

9. References

Aumüller D, Auer S. "Towards a Semantic Wiki Experience-DesktopIntegration and Interactivity in WikSAR." *Semantic Desktop Workshop*. 2005.

Buzan T, "Mind map handbook: The ultimate thinking tool" *HarperCollins UK 2013*

Bizer C, Tom H, Tim BL. "Linked data-the story so far." *Semantic Services, Interoperability and Web Applications: Emerging Concepts* (2009): 205-227.

Chen S. "A cognitive model for non–linear learning in hypermedia programmes." *British journal of educational technology* 33.4 (2002): 449-460.

Dede C. "Immersive Interfaces for Engagement and Learning" *Science 323* (2009) p 66

DuCharme B.. *Learning Sparql*. " O'Reilly Media, Inc.", 2013.

Dijkstra, Edsger W. "A note on two problems in connexion with graphs." *Numerische mathematik* 1.1 (1959): 269-271.

Eades P.. *Drawing free trees*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991.

Eppler MJ. "A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing." *Information visualization* 5.3 (2006): 202-210.

Fogg BJ. (2009) A behavior model for persuasive design, Proceedings of the 4th International Conference on Persuasive Technology, April 26-29, 2009, Claremont, California

Freitas S, Neumann T. "The use of 'exploratory learning' for supporting immersive learning in virtual environments" *Computers & Education 52* (2009) p 343-352

Fisher, Ronald A. "On the interpretation of χ 2 from contingency tables, and the calculation of P." *Journal of the Royal Statistical Society* 85.1 (1922): 87-94.

Herman I, Melancon G, Marshall M, "Graph Visualization and Navigation in Information Visualization: A Survey," *IEEE Transactions on Visualization and Computer Graphics, vol. 6, no. 1* (2000) pp. 24-43,

Huang H, Rauch U, Liaw S "Investigating learners' attitudes toward virtual reality learning environments: based on a constructivist approach" *Computers & Education, 55 (3)* (2010), pp. 1171–1182 Elsevier Ltd

Keshavarz, Behrang, and Heiko Hecht. "Pleasant music as a countermeasure against visually induced motion sickness." *Applied ergonomics* 45.3 (2014): 521-527.

Lee EA, Kok WW. "A review of using virtual reality for learning." *Transactions on edutainment I*. Springer Berlin Heidelberg, 2008. 231-241.
Melancon H, Ivan H. "Circular drawings of rooted trees." *IN REPORTS OF THE CENTRE FOR MATHEMATICS AND COMPUTER SCIENCES*. 1998

Likert, Rensis. "A technique for the measurement of attitudes." *Archives of psychology* (1932).

Moore DM, John KB, Robert JM. "Multiple-channel communication: The theoretical and research foundations of multimedia." *Handbook of research for educational communications and technology* (1996): 851-875.

Squire K. "From content to context: Videogames as designed experience." *Educational researcher* 35.8 (2006): 19-29.

Taylor M, Peter R. "Applying graphical design techniques to graph visualisation." *Information Visualisation, 2005. Proceedings. Ninth International Conference on*. IEEE, 2005.

Tu C, Laura E. Sujo-Montes, Cherng-Jyh Y. "Gamification for learning." *Media Rich Instruction*. Springer International Publishing, 2015. 203-217.

9.1 Tools and libraries

Unity 3d
*https://**unity3d**.com/*

Virtuoso universal server
https://www.semantic-web.at/virtuoso-universal-server

dotNetRDF library
http://dotnetrdf.org/