3D visualization in booking systems

Erik Okfors

Computer Game Programming, bachelor's level
2019

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
3D visualization in booking systems.

Erik Okfors
erik.okfors@gmail.com

June 2019
Acknowledgements

This project was developed in collaboration with Explizit Solution AB in Skellefteå, Sweden. I want to thank them for giving me the opportunity to develop this application and for giving me their support. I also want to thank Luleå University of Technology for supporting me through my studies.

- Erik Okfors
Abstract

The use of 3D in web applications became available in the mid 1990’s with the release of VRML. Today the use of 3D is very common even with web application, and three.js and Babylon.js has become some of the most popular 3D libraries to support the creation of 3D within web services. This project was created together with Explizit Solution AB to test if 3D can improve the user experience within booking systems. The project was to be integrated with Explizit’s booking system Adoxa, and make comparisons with a version called Ånghästen which exists in Adoxa. This project was a front end development as the back end already existed within Adoxa, and a survey was made with 15 participants where they tested the original Ånghästen and the new 3D version, and filled out a question form based on that test. The result of the application was a front end 3D visualization of the original Ånghästen. The result of the survey showed that most participants preferred the new 3D version over the original one, and also stated that they would most likely continue to use a 3D application if it was designed well. Many bugs and fixes were discovered during the testing and survey and future work includes continuing to fix these and to make it more dynamic for multiple activities.

Sammanfattning

Användningen av 3D inom webbapplikationer blev möjliga i mitten av 1990-talet då VRML släpptes till marknaden. Idag är användningen av 3D välighet vanlig, även inom webbapplikationer, och three.js och Babylon.js har blivit några av de mest populära 3D-biblioteken som stöder skapandet av 3D inom webbtjänster. Det här projektet skapades tillsammans med Explizit Solutions AB för att testa om 3D kan förbättra användarupplevelsen vid användning av bokningssystem. Projektet skulle integreras med Explizit’s bokningssystem Adoxa, och göra jämförelser med en version som kallas Ånghästen som redan fanns i Adoxa. Detta projekt var en front end utveckling då back end redan fanns i Adoxa, och en undersökning gjordes med 15 deltagare som fick testa den ursprungliga versionen Ånghästen och den nya 3D versionen, och fick sedan fylla i ett frågeformulär baserat på dessa tester. Resultatet av applikationen var en front end 3D-visualisering av den originella versionen Ånghästen. Resultatet av undersökningen med frågeformuläret visade att de flesta av deltagarna föredrog den nya 3D versionen över den originella, och att de flesta deltagare skulle högst troligt fortsätta använda 3D applikationer om de har en bra design. Många buggar hittades under testning och under undersökningen, och fortsatt arbete i framtiden kommer att inkludera att fixa dessa och att göra applikationen mer dynamisk för att kunna hantera flera verksamheter.
Terms and Abbreviations

- **Angular JS** - A javaScript-based open-source front-end web framework.
- **API** - Application Programming Interface.
- **C#** - C Sharp. A general-purpose, multi-paradigm programming language.
- **CSS** - Cascading Style Sheets. A file format for style files.
- **DOM** - Document Object Model.
- **FOV** - Field of View.
- **glb** - Binary Binary GL Transmission Format.
- **gltf** - GL Transmission Format.
- **GPU** - Graphics Processing Unit.
- **GUI** - Graphics User Interface.
- **HTML** - Hyper Text Markup Language.
- **JavaScript** - A dynamic scripting language often used in developing web applications.
- **Markup Language** - A system for annotating a document in a way that is syntactically different from flowing text.
- **MVC** - Model-View-Controller. A design archetype for designing user interfaces.
- **PBR** - Physics Based Rendering.
- **Static document** - A document type that does not change, typically a web document meant to have a specific context every time it is called.
- **Three.js** - A cross-browser JavaScript library used to create and display 3D graphics.
- **UX** - User Experience.
## Contents

Acknowledgements ............................................ i

Abstract ....................................................... ii

Sammanfattning ................................................ ii

Terms and Abbreviations ......................................... iii

1 Introduction .................................................. 1
   1.1 Background ................................................. 1
   1.2 Goals and purpose .......................................... 1
   1.3 Limitations .................................................. 1
   1.4 Explizit ..................................................... 2
   1.5 Agile Software Development ................................. 2
   1.6 Related Work ................................................ 3
   1.7 Method ...................................................... 3
   1.8 Social, Ethical, and Environmental Considerations .......... 4

2 Design and Implementation ...................................... 5
   2.1 Designing GUI ............................................... 5
   2.2 HTML and Angular JS ....................................... 6
      2.2.1 HTML .................................................... 6
      2.2.2 MVC ..................................................... 7
      2.2.3 Angular JS .............................................. 8
   2.3 Web 3D ...................................................... 8
      2.3.1 3D Objects .............................................. 9
      2.3.2 Animations ............................................. 9
   2.4 Adoxa ...................................................... 10

3 Survey .................................................................. 11

4 Result ............................................................ 12
   4.1 Application ................................................... 12
   4.2 Survey Result ................................................ 14

5 Discussion ...................................................... 15

6 Conclusion ...................................................... 17
   6.1 Future Work ................................................ 17

7 Appendix ........................................................ 20
   7.1 A - Tablet Specifications .................................... 20
   7.2 B - Question Form .......................................... 21
   7.3 C - Survey Results .......................................... 23
1 Introduction

This paper discusses the beginning of web 3D, its progress through time, and where it is now. It also discusses what can be done with web 3D, what the most common occurrences are, and other more unusual areas. This project was created to explore the different uses for web 3D and if it can be used to improve user experience in applications that already exist. Explizit AB in Skellefteå, Sweden has taken on this question and this project is in collaboration with them.

1.1 Background

In 1993, the first version of the now standard web language HTML was released[1]. Not long after its release in 1995, the first versions of 3D graphics for the web was introduced with VRML or Virtual Reality Modeling Language[2]. VRML was suspended and replaced with X3D in 1999, by the Web3D Consortium[3]. The Web3D Consortium is an international member-founded, non-profit industry founded 1997. They work towards promoting X3D standards when it comes to communication between 3D scenes across scenes in multiple applications, platforms, and more.

In 2007 WebGL[4] was released, which is based on the popular library OpenGL[5]. Open Graphical Library first released in 1992. OpenGL is used to render 2D and 3D vector graphics using the GPU. WebGL is a JavaScript[6] API for rendering 2D and 3D graphics without the use of plug-ins in any compatible web-browser. It is fully integrated with other web standards which allows GPU-accelerated usage of physics and image processing.

In 2010, the cross-browser, JavaScript based Library and API, Three.js[7] was introduced to the world, which uses WebGL and is fully open Source and can be found on GitHub[8].


1.2 Goals and purpose

The goal of this project was to investigate if 3D visualization can improve the user experience in a booking system.

The purpose with this web application was to enable Explizit an opportunity to improve user experience with visual aid, and to enable future development of new services.

1.3 Limitations

A decision was taken, together with Explizit Solution AB, that the development focus on one of their products called Anghästen, in their system Adoxa, see figure 1. One of the functions of Anghästen is a system for booking laundry rooms and washing machines. This project was to implement a 3D version of that system.

A limitation was that the back-end system already existed. Therefore the implementation was done with front end development.
1.4 Explizit

Explizit AB[10] is a Swedish group of companies providing IT services, environmental solutions, services in the public sector, and more.

Explizit Solutions AB[11], one of the companies in the Explizit Group, handles all IT services, development, and maintenance for the group. Today they develop a booking- and resource management system used by many clients like Skellefteå Kraft[12], Uppsala Academic Hospital[13], and more. The system is developed in Microsoft[14] .NET[15] framework with an architecture and design in accordance to Microsoft’s recommendations.

Explizit always looks for new technology when developing software. Creating a user-friendly, appealing system is crucial in designing a product for users and administrators.

Adoxa is a product that has its origin from when Explizit developed booking and scheduling systems for a company called Bilprovningen, and the product handled over 5 million bookings per year. Explizit saw the possibilities to create a more dynamic system which would fit with other activities that were similar to Bilprovningen, but the system would also be built so that small adjustments would allow it to be used in other types of companies. Today, Adoxa exists in six branch versions with customers like Arbetsförmedlingen, and BRF Änghästen. Adoxa is developed in .NET environment with HTML5, C#, Angular JS, Bootstrap[16], Ionic-platform[17], and more.

1.5 Agile Software Development

Agile Software Development[18] is an approach used in software development to help the development team and the client to achieve the most out of the development time.
general idea is to break down the work into priorities and small manageable tasks. The production time is often broken down into smaller time frames and the development team decides with the client what tasks should be developed under the current time frame. This results in a dynamic workflow and the client is always in the loop and can change the project according to their needs.

Another practice is that the team has short meetings at a regular basis and discuss the work that has been done and what is to come. This gives the whole team a clear vision of what is going on and can adapt to any changes. At the end of the time frame the team can look back on the work and plan for adjustments to the coming work ahead.

There are many types of agile workflow with some of the more common practices like Scrum and Kanban.[19]

1.6 Related Work

3D graphics on web pages and web applications started appearing on the market around the middle of the 1990s, but the 3D visualizations of booking systems is younger and less common.

InstaTable[20] is an application that allows the user to book tables at a restaurant. It uses a 3D visual layout of the floor plan to show what tables are free and where the table is in the restaurant. The application can then remind the user of upcoming bookings via text messages to their phone. The general structure of the application is close to what this project strives for and can learn from how they designed it.

A lot of game engines on the market right now, like Unity 3D[21] and Unreal Engine 4[22], support the option to build projects to Html 5. This enables an easy way to create web based games directly in the editor, the same way you would create a game for any other platform today.

1.7 Method

This work was divided into four major parts, design, development, integration, and survey. The design phase involves trying to understand what the user needs to see and how it will be organized so it will be easy to understand and use.

The development was done with Angular JS[23] together with Three.js. Angular JS was chosen because it is well known for its use of the MVC framework[24]. Moreover, the underlying architecture is using it as well. Three.js was chosen since it is a JavaScript library and could easily be included in the project.

The integration phase includes connecting the application with the already existing system, and was needed for enabling the use of the back-end logic.

When the integration was done, the application was tested with people that were given the task of doing a survey to evaluate the project.
1.8 Social, Ethical, and Environmental Considerations

The environment and server were all test versions. No user information could be altered or shared during the development, testing, and survey. User information like passwords are stored as encrypted versions so the users password cannot be distributed. The survey was made anonymous and the individual data has only been seen by the author of this paper, other than the tester.

This project was design to improve user experience in hopes of improving their social well-being.

This project was a digital development, which led to the following conclusion. If successful, this project could lead to more people using it, which could lead to less paper usage. On the other hand, if more people, companies, and housing societies decide to use this application, more devices would be bought and used, which would lead to a higher electricity consumption.
2 Design and Implementation

2.1 Designing GUI

The applications target audience was mostly middle-aged and older, and would use the application far apart in time, around once every one to two weeks. This meant that the GUI had to be intuitive, easy to use, and appealing to look at. The original system uses a straightforward, blocky style that is simple in itself but can be hard to read, and together with drop down menus it can be hard to understand at a first glance what the user is supposed to do. The idea with the new application was an attempt to create a design that the user could quickly look at and understand the layout and flow with the aid of visualizations and 3D graphics. It was important to understand the flow of the application so that it could be created with user friendliness in mind[25].

The design was made for a Samsung Galaxy Tab A, which is a tablet. The project had to take this into consideration since the screen layout and tablet hardware is different to standard browsers like Google Chrome[26]. Specifications for the tablet can be found in appendix A.

Firstly, the design started with understanding the flow of the already existing system and determining what the core part were. For example, what steps does the user take to accomplish the goal of booking a time, and how much control does the user have. After these parts were determined, it was time to figure out what parts could most feasible be represented with 3D visuals. This needed a lot of consideration as all part could be represented with 3D but that would most likely end in being to much for the user and end up making the application harder to use.

Secondly, the design was sketched on paper, to see the general look and visual of the application with all the parts in it. This was done to get an overview of the core application, what would be shown, and if something was missing or unnecessary. It was then built in blocky forms to get a feel for measurements and scale. It showed if some part needed to be moved, bigger or smaller, and most importantly, functionality and flow could be tested.

Lastly, the design was finalized by picking colors, beautifying, animating, and matching Anghästens style. The steps of this design went a lot back and forth because changes happened frequently. These changes ranged from layout changes to application flow restrictions. Most of the times this meant going back to the sketch phase.
2.2 HTML and Angular JS

2.2.1 HTML

HTML or Hyper Text Markup Language is the standard language for creating web pages and web applications. HTML works on the principle of processing events, parses HTML code, builds DOM, and renders the page[27].

Events are generally something that the browser or the user does. This can be, for example, when the page finishes loading or when a button or link is clicked. This can be linked with JavaScript to have the browser execute certain code when events are detected.

The HTML-parser goes through the code and parses it, by resolving sentences and describes their syntactic roles.

DOM, or Document Object Model, is an API that is cross-platform and language independent. It treats HTML documents, and other markup documents like XML, as a logical tree structure, where every branch ends in a node, and every node contains objects. This way, the structure, style, or content of the document can be changed through DOM methods, which allows programmatic access to the tree.

To render, included .css files, style files, and DOM are combined to create a render tree. This tree contains all the information of the visible DOM content that are to be present on the page. Elements that are hidden through style options are not represented in the render tree. It then calculates the layout, size and position of elements, and finally paints them on the screen[28].
2.2.2 MVC

MVC is an architectural pattern commonly used when developing user interfaces. It structures the application into three interconnected parts, Model, View, and Controller, hence MVC. The Model is where all logic and data is stored and managed and it is independent to the user interface. The View is everything that the application displays and the user sees. The user does however not interact with the View, that is handled by the Controller. It takes input from the user, or calls from the Model and translates them to commands for the Model or the View.

A common approach with the MVC is along the lines of this. The user makes an input which is handled by the Controller. The Controller translates the input to a command that it sends to the Model. The Model takes the command, handles data, and sends back instructions to the Controller or directly to the View. The Controller then translates that instruction to a command for the View. The View is then updated for the user to see.
2.2.3 Angular JS

Angular JS is a JavaScript-based, open-source, front-end web framework that works like an extension to the browser. HTML is great at developing static document so Angular JS was developed to extend it to be able to create applications. Angular JS’s dependency injection and data binding simplifies and eliminates much of the code that otherwise had to been written. Angular JS is good in combination with server technology since everything happens within the web browser.

2.3 Web 3D

Three.js was chosen to handle the web 3D for this project, due to its simplicity and since it is written in JavaScript. The components required to create this 3D environment were a camera, a scene, objects, a light source, and a renderer.

The camera defines what the user sees in the 3D world. This camera was created as an orthographic camera, which is a camera without perspective, as opposed to a perspective camera. This choice was made based on the design. The camera requires the width and height of the ”screen” that it is supposed to render to, though it does not have to be the whole physical screen. It can be a part of a site in which case the screen size is the size of that particular window. The camera also needs a near and far plane. These determine the minimum and maximum depth that the camera can see. Anything behind the near, or further away than the far plane won’t be captured by the camera. The camera usually needs a field of view, but since this project used a orthographic camera without perspective, it was not needed.

The scene defines all the components that are in the 3D world, all 3D objects, lights, animations, and so on are part of the scene.

The objects for this project were all the 3D elements that we desired, the washing machines, the background, and signs. Every one of these objects were setup using geometry and
materials. The geometry is all the data for the shape of the object, like a cube, or a sphere. The material describes the surface properties like color, roughness, and shininess. These combined together made a mesh that could be added to the scene. Geometry and materials can be made with code, or in this case, imported from external files.

The scene now had objects to populate the 3D world but if it were to render now, nothing would be seen, or would be seen as black. The world needs a light source to distinguish the separate objects. The light source used in this project were two directional lights, and an ambient light. A directional light is a light source that illuminates all objects from one specific direction. An Ambient light is a light that globally illuminates the whole world from every direction. The first directional light was used as the main light source, the second as a fill light to simulate bounce light. Lastly the ambient light was used to brighten the whole scene, which also made pitch black shadows more transparent.

The renderer is where the magic happens. It takes the camera and the scene, and uses WebGL to render all the scenes content through the camera so that it can be visible to the user.

2.3.1 3D Objects

Objects in the scene for this project were the washing machines, the background, and props. They were all created using Autodesk Maya 2018[29]. The object were optimized as much as possible without loosing detail, this due to the fact that the application was a web application and was run on a tablet. The specification for the tablet can be found in appendix A.

All the objects were exported as .glb files, which is a binary version of WebGL’s gltf[30] file extension. Autodesk Maya does not have an exporter for that file format per default so an exporter called Babylon JS[31] was used. It can export the selected objects, or the whole Autodesk Maya scene, and transforms the materials in Autodesk Maya to PBR materials that is supported by .glb files. The files where then loaded by using a gltf loader in Three.js, which is one on the standard loaders for the version that was used in this project.

Figure 5: 3D Objects in Autodesk Maya. From left to right. Background raw mesh, background with material, machine raw mesh, machine with material, sign raw mesh

2.3.2 Animations

Animations were added to give the application more life and enhance the user experience. Listed below are the different methods and libraries that were used in the creation of the animations for this project.

Tween[32] is an open source library that works together with three.js to create animations
with object in the 3D scene used in three.js. Tween was used to create simple animations, like translating or rotating an object from state a to state b, were created using tween, since it could all be done through code and be used on dynamically created objects like billboards and other pop ups.

Not everything in the application is 3D, instead they are HTML and Angular JS elements. Tween cannot animate these, so Angular JS’s animation library was used here instead. It works in a similar way to .css and are written in style files like .css. With this, regular HTML elements could be animated and made to fit the style with the 3D elements.

More complex animations were created in Autodesk Maya. It was done using keyframe animation and then creating animation sets using the Babylon JS .glb exporter. The animation sets were setup by assigning a start and end keyframe, and a name of that animation. These animations could then be called inside the application by calling the the specific animation by name.

![Figure 6: Babylon JS exporter for Autodesk Maya](image.png)

### 2.4 Adoxa

The setup between this projects application and the Adoxa’s back end system was relatively straight forward. By adding the View and Controller to Adoxa’s system setup, the 3D application was integrated in the system. A service module could then be passed to the Controller so that function could be called from the 3D application Controller. These function calls were:

- Getting user information, for example id.
- Getting the available times for the machines on a specific date
- Reserve chosen times
• Booking reserved times

Since this application used libraries that was not already in Adoxa, a bundle was created for this project that contained all the necessary includes to make this project work.

Lastly, a setting had to be changed so that the web server allowed the .glb files to be loaded. This was done by telling the web server to allow a specific file extension and declaring what kind of format the file has, like plain text or binary. The web server allows per default many of the standard files but .glb appeared to be not one of them so the web server would deny access to the files.

\section{Survey}

The survey was done using a tablet that this project was designed for and sitting down with the user, explaining the steps, and letting them do the test. They were instructed to ask any question if they felt the need to but would otherwise be left to their own device.

The survey was made up of three parts.

• Book a washing machine using the regular system, Ånghästen.

• Do the same task as before but by using the new 3D version.

• Fill out a question form.

Since this project was to evaluate if the use of 3D could improve the experience of booking systems, it was deemed necessary to use the regular system, which the 3D application was based on, as a reference. First the user used the original version and booked a time and washing machine. The user would then do the same task again using the new 3D application created in this project. After that was done, the user would fill out a question form, see appendix B. The questions were designed to determine the users experience with mobile devices, web services, 3D, and lastly what they thought of the 3D application. Many of the questions were designed so that the question would relate to the difference between the two versions.
4 Result

4.1 Application

The result of the design and integration was an application that was comprised of a large 3D view and a few HTML elements like a date selector and the buttons, see figure 7 and 8. The general approach is that the user chooses a date and then picks the desired machine they wish to book. The application will then zoom in to the machine, displaying the available times that can be booked. Different times can be viewed by swiping left and right when in the zoomed in view. When on a desired machine and time, the user can choose to add that selection to their bookings, see figure 9. The user can see the current selections by pressing the lower-mid button marked "Bokningar". The button also show the current number of selections. When the desired selections has been selected, the user clicks the "Boka" button which will reserve the times and bring up a final confirmation pop-up that the user can agree or decline to. If agreed, the times are booked and the application clears all selections. If declined, the application reverts the reservation of the times, and the user is brought back to the application without the loss of their selections. The design, choice of colors, and the shapes of elements were made to match the original Ånghästen application, see figure 1.

All 3D assets were created using Autodesk Maya 2018 and Adobe Photoshop[33], which were then exported to a single .glb file using the Babylon JS Exporter for Autodesk Maya. The 3D view was rendered using Three JS render library, and the HTML code was created with HTML, JavaScript, css, and Angular JS, in Visual Studio 2017[34].

Figure 7: Ånghästen 3D Visualization, start view
Figure 8: Ånghästen 3D Visualization, machine selected

Figure 9: Ånghästen 3D Visualization, current selections
4.2 Survey Result

The number of participants in the survey was 15 with an average age of 28.2 years old. The majority of them were male, with only 13.3 % female. When asked how much they used their mobile devices, 100 % of them answered that they use them every day, and the majority, 86.7 %, had experienced with modern web applications. The participants thought that the application was easy to understand, with an average result of 4.27 out of 5. When asked if the applications improved their experience, the results were 3.47 out of 5, above middle. Though the result showed that they did not think it made it easier to accomplish the goal of booking a time, with a result of 2.4 of 5.

The results for the questions "First impression", "Overall design", "User friendliness", and "The use of 3D" had almost the same result, with the values ranging from 3.54 to 3.87 out of 5. Every one of them were above the midpoint resulting in a more positive outcome.

When asked if they would prefer the original or the new 3D application they answered that 2/3 would use the 3D version instead of the regular one. Lastly, 66.7 % of the participants would use a 3D booking application again if presented with one, but only if it was designed well. 26.7 % said that they would use one without question, and 6.6 % answered that they would probably not use one.

See appendix C for the full documentation of the survey result.
5 Discussion

I tried to keep the application as simple as I could to reflect the original version that this 3D version was based upon. The goal for me was to see if the users would prefer the new 3D version over the original one, not to create a universally perfect 3D booking application. And with that, I think I succeeded. Over half of the testers did in fact choose the 3D application over the original one. There were mixed reasons for it but one of the more common one was that it was more interesting to look at. On the other hand, majority of the testers that choose to stay with the original one stated that they liked the simplicity of it. This resulted in a quite clear split in the target audience, those who like the application to be simple and to the point, and those who like the visual aesthetics and does not mind a less direct approach. This comes back to the question of "Can we make both happen?", which brings us back to "Is it necessary?". My first thought of this project was just that, "This is not necessary." but was happily surprised that the end result, though not perfect in any way, did come through with mostly positive results.

That said, the decision of creating this project in the form of a laundry booking system gave me mixed feelings. On one hand, from a marketing and user experience point of view, this was a really good thing. Many people know what the goal is and can quick and easily get the purpose of the application. This gave this project a wide consumer audience to develop towards. On the other hand, from the perspective of an artist and 3D designer, this was going to be challenging. The use of 3D can improve a lot of visual design if done right, but can at the same time just complicate things. Going from a 2D perspective to 3D with no reason or goal can lead to a complicated user experience, and that users with less experience with the technology will not use it. That is not to say that the 3D did not make the application more appealing to look at, but if the majority of the users can’t, or is not willing to use it there is not point in doing this transition.

There was a bit of a struggle when it came to the creation of the 3D assets. At the beginning of development I used an older version of Three JS, thinking that I could use the integrated json-loader to import 3D assets. I soon realized that Autodesk Maya does not have any way to export to that format, and that the plugin that Three JS supplies is not updated for new versions of Autodesk Maya. This lead to the discovery of the gltf file type, which supported all areas I needed, mesh, animation, materials, and textures, stored in one file. After that I realized that the current version of Three JS had a problem with the gltf-loader and could only load the binary version of the gltf file, which was .glb. In the end this was the best I could have done since this sped up the pipeline from Autodesk Maya to Three JS.

When optimizing the assets for the tablet, I may have underestimated its potential. When starting the testing on the tablet, I noticed that the application was running a little slow, thinking that it was due to the 3D scene. I optimized it by reducing texture sizes, polygon count, and the resolution on shadow maps. These optimizations can be noticed in figure 7, especially on the floor and the shadows. In the end it turned out not to be the problem but instead it was the environment that I ran the application on. I had been given the system test environment and work was constantly being done in the background which I had no control over. So the optimization ended up being unnecessary but was kept in the final version since I had already started the survey and changing the application meant I had to do it over.
The integration with the back end went really well. There were just a few complications like telling the web server that the glb file extension was needed and should not deny access to it. The rest was adding a reference to the back end to my Controller in the MVC so that I could call functions to for example book times and reverse times. Also since I integrated my application into theirs, I did not have to handle user log in or passwords.

The survey was interesting. I only got 15 people to participate, which is a fairly small number for a survey but I think they made their point clear in that range of users. The test required me to be with the tester which complicated things since I had to be there when the participant had time. Luckily I only needed the tablet and the question form so it was easy to travel with it, but most of the participants in the survey are mostly comprised of the people in my surrounding, 30 year old people with daily experience with computers or smart phones. Young people today tend to be introduced to this kind of technology early, but I would have liked to test this on more people that are of an older generation, whom did not grow up with this access to computers and smart phones. That would have evened out the survey more and gotten a more realistic result over the broad group of users that would use Adoxa today.

The users seemed to like the test, even though they looked skeptical when I described what the test was going to be about. When testing the original one, many stated that it was good and asked how 3D was going to improve it. Then they tested the 3D version and the results were mixed. Many of them smiled at the start up and said it was more fun to look at, but the results shows that looks are not always the strong point. See the full result of the survey in appendix C.
6 Conclusion

According to the result of the survey, it is fully possible to improve the user experience with the aid of 3D visuals. It is important to know what the target audience is for it to have the most impact. The use of 3D might be more preferable for a younger user base that are more experienced with the technology and will most likely be more open to the new design. The design also needs to be refined and thought through thoroughly before starting up this kind of project, and think through if it really is necessary to make this change. If the target audience is not willing or does not find it easier to use the new version, the development is not needed.

The project needed to have a clearer design choice and direction earlier in the project, instead of in the middle of it. This could have been solved by more research at the start of the project to avoid setbacks like what happened with the exporting of the 3D assets. Same goes for the optimization. If researched more it would have been clear that the tablet could handle more detail in the 3D and possibly enhance the experience even more.

This project needed more tester that were older and/or less experienced with 3D and today’s technology to get a more accurate representation of the demand for this kind of application.

6.1 Future Work

The results of the survey gave a lot of valuable feedback on the application, which can be used to improve the application for the future. Some feedback was very specific for a participant and can be overlooked if the developer find it necessary, but other comments came from multiple participants which points to a recurring problem which should be solved. Other problems were found by the developer during testing and the survey which the participants did not notice but needs to be fixed if the application is to be taken to further development.

One request that Explizit Solution AB came with was that the application could be setup dynamically, so that the visual setup with the washing machines could be changed to fit a specific setup or activity, without the need for code change or new assets.
References


# Appendix

## A - Tablet Specifications

### GENERAL INFORMATION

<table>
<thead>
<tr>
<th>Model Name:</th>
<th>Samsung Galaxy Tab A 10.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Number:</td>
<td>SM-T580</td>
</tr>
<tr>
<td>Released:</td>
<td>2016</td>
</tr>
</tbody>
</table>

### PHYSICAL SPECIFICATIONS

<table>
<thead>
<tr>
<th>Dimensions:</th>
<th>254.3 x 164.2 x 8.2 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight:</td>
<td>554 g</td>
</tr>
</tbody>
</table>

### OPERATING SYSTEM

<table>
<thead>
<tr>
<th>Operating System:</th>
<th>Android v 6.0.1 (Marshmallow)</th>
</tr>
</thead>
</table>

### NETWORK

<table>
<thead>
<tr>
<th>Wifi Only:</th>
<th>Yes</th>
</tr>
</thead>
</table>

### DISPLAY

<table>
<thead>
<tr>
<th>Technology:</th>
<th>TFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size:</td>
<td>10.1” (255.4 mm)</td>
</tr>
<tr>
<td>Resolution:</td>
<td>WUXGA 1920 x 1200</td>
</tr>
<tr>
<td>Color Depth:</td>
<td>16 M</td>
</tr>
<tr>
<td>Pixel Density:</td>
<td>Around 224 PPI</td>
</tr>
</tbody>
</table>

### PROCESSOR

<table>
<thead>
<tr>
<th>Type:</th>
<th>Exynos 7870</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Speed:</td>
<td>1.6 GHz</td>
</tr>
<tr>
<td>Cores:</td>
<td>8 Cores (Octa-Core)</td>
</tr>
</tbody>
</table>

### MEMORY

<table>
<thead>
<tr>
<th>RAM Size (GB):</th>
<th>2 GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM Size (GB):</td>
<td>16 GB</td>
</tr>
<tr>
<td>External Memory Support:</td>
<td>Up to 200 GB</td>
</tr>
</tbody>
</table>

These specifications were found on Samsung’s website
7.2 B - Question Form

Age: _____

Gender:
- Male
- Female
- Other

How often do you use mobile devices? (List of choices)
- Every day
- Every other day
- Once a week
- Almost never

How much experience do you have with modern day web applications, for example Youtube, Facebook, Instagram, or news sites?
- Nothing
- Very little
- Moderate
- Very much

How would you rate your experience with 3D in general?
Not much: 1 2 3 4 5

Was the 3D application easy to understand?
Not much: 1 2 3 4 5

Did the 3D improve your experience? (1-5)
Not at all: 1 2 3 4 5

Did the 3D make it easier to accomplish your goal? (Book a time)
Not at all: 1 2 3 4 5
What did you think of:
1 = Bad ← → 5 = Excellent

First Impression
1 2 3 4 5

Overall Design
1 2 3 4 5

User friendliness
1 2 3 4 5

The use of 3D
1 2 3 4 5

Did you prefer the regular applications or the new 3D version?
Regular
3D

Would you use 3D booking applications again if available?
No
Probably not
If designed well
Yes, definitely

Open comments.
Result Survey

Number of participants: 15

Ages:

Average 28.2
Mean 26

Gender:

Male 86.7 %
Female 13.3 %

How much do you use mobile devices?

Every day 100 %
Every other day 0 %
Once a week 0 %
Almost never 0 %

How much experience do you have with modern day web applications, for example Youtube, Facebook, Instagram, or news sites?

Nothing 0 %
Very little 0 %
Moderate 13.3 %
Very much 86.7 %

How would you rate your experience with 3D in general?

Average 4.07
Mean 4

Was the 3D application easy to understand?

Average 4.27
Mean 5

Did the 3D improve your experience?

Average 3.47
Mean 4
<table>
<thead>
<tr>
<th>Question</th>
<th>Average</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did the 3D make it easier to accomplish your goal? (To book a time)</td>
<td>2.4</td>
<td>2</td>
</tr>
<tr>
<td>First impression</td>
<td>3.6</td>
<td>4</td>
</tr>
<tr>
<td>Overall design</td>
<td>3.54</td>
<td>4</td>
</tr>
<tr>
<td>User friendliness</td>
<td>3.87</td>
<td>4</td>
</tr>
<tr>
<td>The use of 3D</td>
<td>3.87</td>
<td>4</td>
</tr>
<tr>
<td>Did you prefer the regular application or the new 3D version?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regular</td>
<td>33.3 %</td>
<td></td>
</tr>
<tr>
<td>3D</td>
<td>66.7 %</td>
<td></td>
</tr>
<tr>
<td>Would you use 3D booking applications again if available?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>0 %</td>
<td></td>
</tr>
<tr>
<td>Probably not</td>
<td>6.6 %</td>
<td></td>
</tr>
<tr>
<td>If designed well</td>
<td>66.7 %</td>
<td></td>
</tr>
<tr>
<td>Yes, definitely</td>
<td>26.7 %</td>
<td></td>
</tr>
</tbody>
</table>
Open Comments

- Gui was ok, relatively pedagogy

- I will probably most often go for the version that is the fastest, least amount of clicks. When I book laundry I just want it to be quick.

- Liked the physical aspect of the 3D version. But it was still hard to understand what times were available and if they had been booked or not.

- The hardest part was to get an overview over the times that were available.

- It was hard to understand that the machines were buttons.

- Really nice that you could see what kind of machine or tool that you were booking with visual aid.

- Prefer the regular one cause you get a better overview of the times.

- Hard to understand at the end that the booking was done. Many buttons.

- Better graphics

- Improve responsiveness on the machines.

- Indicate which machine you have booked by making the sign above it green, like the date selection buttons.

- Maybe make those signs red instead of having the black machines when no time is available.

- You should not be able to select more dates than you are allowed.

- Read information about an entire week at a time, and make booking requests asynchronous maybe.