FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme / specialisation:

Master's in engineering / Computer/Data Science

The spring semester, 2022

Open / Confidential

Authors:

Ove Kristian Jørgensen

Ove Jør

....
(signature author)

Marius Sørensen

Marius Sørensen

....
(signature author)

Supervisor:

Antorweep Chakravorty

Thesis title:

ARDENT – An Augmented Reality Framework for Education

Credits (ECTS): 30

Keywords:

- Mobile, web and cloud development
- Cross Reality(XR)/Augmented Reality(AR)
- Algorithms and data structures

Pages: 13

\+ enclosure: poster.pdf

Stavanger, 15th of June 2022

ARDENT - An Augmented Reality Framework for Education

Ove Kristian Jørgensen*
Dpt. of Electrical Engineering and Computer Science
University of Stavanger
Stavanger, Norway
ove.kristian.jorgensen@gmail.com

Marius Sørensen
Dpt. of Electrical Engineering and Computer Science
University of Stavanger
Norway, Stavanger
marius@s0rensen.no

Abstract

Virtual Reality (VR) in education and medicine has been a widely researched area. Increase in motivation and learning gain is demonstrated to be directly related to introduction of VR applications. However, the necessity for expensive and bulky hardware limits the usability of VR. This problem can be partially solved by using Augmented Reality (AR), which is readily available in most modern smart devices. In later years the same kind of research has been increasing for AR, but to a lesser extent. This increase might be attributed to improved affordability and the rapid technological advances of smart devices. ARDENT is a framework for the standard development of an AR application and its swift deployment across diverse smart device platforms. The framework seeks to make it as effortless as possible for a user to deploy their models before a multitude of other users can see these models through their smart devices. Finally, this framework is showcased in a one-to-many teacher-student relationship for a seamless in-class AR experience.

Keywords: Augmented Reality, Cross Reality, Application Development

1 Introduction

Augmented Reality (AR) is a technologically enhanced version of the real world, and an exponentially expanding area of research and development [9]. AR has an endless potential in a myriad of different use cases, be it for leisure, to socialize or to learn in entirely new ways.

However, as technology advances, so does the level of complexity. Despite the fact that AR has been implemented and experimented with in thousands of different applications, the field is still not particularly developer-friendly. The APIs and frameworks used for AR development have lackluster documentation and the community is still rather small. This leads to a more complex development procedure and limits motivation for further AR development. When new technologies emerge, the complexity of development will grow as well. For laypeople the learning curve when exploring AR technologies with no prior experience may be steep.

Through this project we intend to utilize AR to make the learning experience more interactive and enjoyable for students within any field. The application that has been created is simple to use for both teachers and students, and it can be used for any subject that may benefit from visual graphics. Displaying graphics in AR has been proven to increase learning motivation [1] and outcomes [10].

We decided to conduct a user evaluation as outlined in section 7 with a demonstration of a functional prototypic implementation of the framework to gain insight into how the framework would be perceived. This type of study is especially useful when a minimum viable product is available to pinpoint potential future development, and to locate realistic end-product functionality.

This paper firstly explains the background for choosing augmented reality and the technologies used when implementing a prototype of the ARDENT framework. Following the background we present our research of related works, and confirm that there is a gap in previous research which we intend to fill. After justifying the framework we explain the outline and functionality of the framework in sections 4 and 5. In section 6 we present our own implementation of the framework. Finally, a demo and evaluation was performed in section 7 which led to a positive conclusion with a plethora of possibilities for further work in section 8.

2 Background

Augmented Reality is a growing field of technological research with endless potential and exiting new ventures. As the name suggests, AR augments our reality by injecting virtual objects into a view of our world. This view can be experienced in a multitude of different ways, for example through a smartphone or a more sophisticated virtual reality headset which can put through a feed of the real world using external cameras [28]. This technology can be used for making every day tasks easier, like fitting new furniture into your house before you buy it, using the IKEA Place application, [1] or trying on new makeup from L'Oreal [2]. It can also be used for entertainment purposes, like when Pokemon Go [3] took the world by storm, leading to increased physical exercise amongst users [3].

*Both authors contributed equally to this research.

[1]https://www.youtube.com/watch?v=UudV1VdFtuQ
[2]https://www.loreal.com/en/articles/science-and-technology/makeup-virtual-try-on-maybelline/
[3]https://pokemongolive.com/en/

Unity [4] is a free, publicly available game development engine. It focuses on making game development as easy as possible for both new and experienced game designers. Unity AR Foundation is a cross-platform framework for creating AR applications for both major mobile platforms, Android and iOS. Unity AR Foundation works as an API which can be used to communicate with ARCore and ARKit from a single codebase. ARCore [5] and ARKit [6] are the respective AR APIs for Android and iOS.

Flutter [7] is a fast, productive and flexible framework for creating cross-platform applications. Hot reloading, an extensible UI framework using the Material UI [8] styling, and simple compilation to both Android and iOS provides the developer with a seamless development experience.

.NET [9] is a free, open-source framework for creating applications for Windows, macOS and Linux. .NET supports multiple different programming languages and remains popular for developing APIs for any application connected to the internet. Some advantages of .NET are strong typing through C#, a large and active community and excellent development tools from Microsoft [10] and JetBrains [11].

Microsoft Azure [12] is a cloud computing service offered by Microsoft. Azure provides more than 200 products, including hosting of server side applications and web-apps. Additionally, multiple different deployment possibilities reduces the development time lost to continuous operation of an application in a production environment.

React [13] is a JavaScript library for creating declarative, efficient and flexible web-based user interfaces. By utilizing the power of components and a vast selection of libraries and frameworks, React allows simple development of complex web applications.

3 Related Work

VR for use in education is an exhaustively researched and developed field. Multiple studies have shown that by using various types of VR applications, students of all ages may benefit [4, 23, 24].

AlAwadhi et al. developed a VR application in which students can "perform experiments, attend online live 360 degree lectures, watch pre-recorded lectures, have a campus tour, and visit informative labs virtually" [2]. This allows students who are impaired to have a more genuine student experience while also raising motivation for typical students.

Bryan et al. has developed a hybrid AR/VR application allowing the user to virtually travel to multiple places on earth [5]. While the user sees the area around them through Google Street View [14] they are able to read facts which are mapped on top of the VR view using AR technology.

CityViewAR developed by Lee et al. provides much the same functionality as Bryan et al. but through a smartphone camera and without virtual placement of the user anywhere on earth [16]. The user can find out facts about the city they are currently in by launching the application and pointing the camera at nearby landmarks. Information about said landmark will be shown in one of many data visualization options such as 2D maps, 3D models of current buildings or historic landmarks, or panorama photographs.

Colpani et al. [6] observes that multiple VR and AR applications have been made for users with motor disabilities, and far fewer for users with intellectual disabilities. Therefore, they propose an AR framework which should assist the learning procedure for children with intellectual disabilities.

An application that is especially relevant to education is GeoGebra AR[15] which lets users experience complicated mathematical graphs in AR to further understand mathematical concepts. Widada et al. proves that this way to visualize complex concepts increases learning gain for high school students [25].

SMART is "a SysteM of Augmented Reality Training" which was developed mainly for teaching second grade students [8]. The application superimposes 3D models into the real time video feed of a school class. Freitas and Campos further their research by presenting results from experiments with three local primary schools. These results suggest that SMART is effective in maintaining high levels of motivation between students and has a positive impact on the students' learning experience, especially amongst the weaker students.

Several studies have shown that increasing the number of hands-on, interactive activities improves learning [20, 22, 27]. Yammine, for example, compiled a study based on 134 papers to determine whether using 3D models of bodily parts increases information gain when teaching anatomy [29]. They documented significant increases in information gained and knowledge outcome after long-term retention. In several of the examples cited by Yammine, the 3D models are physical models that each learning institution must obtain. For example, physical models of the heart or the human skeleton. For this learning strategy to be effective in large courses, multiples of the same costly model must often be purchased.

Zhu et al. identified a wide spectrum of research papers on the specific subject of AR 3D models applied in healthcare.

[4] https://unity.com/
[5] https://developers.google.com/ar
[6] https://developer.apple.com/augmented-reality/arkit/
[7] https://flutter.dev/
[8] https://mui.com/
[9] https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet
[10] https://visualstudio.microsoft.com/
[11] https://www.jetbrains.com/rider/
[12] https://azure.microsoft.com/en-us/overview/what-is-azure/
[13] https://reactjs.org/
[14] https://www.google.com/streetview/
[15] https://www.geogebra.org/m/R8Qd7U8y

Their findings indicate that as acceptance of these AR applications grows, so does the potential for boosting various sorts of healthcare competence [31].

The field of augmented reality develops and matures [28], which begs the question of whether AR can be used effectively in other educational environments. Saidin et al. examined AR research in education in a variety of subjects, including medicine, chemistry, mathematics, physics, geography, biology, astronomy, and history [21]. They discovered that AR-assisted learning increases students' motivation and allows them to swiftly enhance their visualizing skills. Furthermore, teachers felt that it was easier for them to explain and ensure that their students understood what they were taught. Finally, Saidin et al. believe that the fundamental limits of AR are technological in nature. They add that if this topic is further researched and these issues are resolved, the efficiency of both teaching and learning using AR will improve even more.

Augmented Reality might prove to be a key component in learning environments in the future. There is a limited number of studies with a significant enough number of participants to evaluate the effectiveness of AR as an educational tool. However, we can compose a generally positive view from several of the previously mentioned related works. This is further supported by multiple composite studies of AR in education. For example, Yuen et al. and Hsin-Kai et al. both concluded that the future for AR in education is bright[26, 30]. Additionally, the Horizon reports of 2010[12] and 2011[13] predict that AR applications will be a vital part of college education.

Finally, it becomes obvious that it is important to be critical of related work within the field, when Kerawalla et al. presents clear results that show that in some cases AR can be detrimental to the learning experience [15]. Luckily, they provide four guidelines for creating a successful AR interface design for motivational and educational gain.

Existing literature highlights the benefits from incorporating AR into education as it enhances the learning experience and increases learning gain. However, these applications do not let the content of the AR application be readily updated by a teacher or professor. Existing approaches to this problem do not provide scalability for the teacher as creating models is a complex procedure. Unnecessary complexity can reduce the motivation for a teacher or an institute to incorporate this technology in their day-to-day educational settings.

4 Framework Requirements

The ARDENT framework requires two parts, one for students to view AR models on their smart device and a second for a teacher or professor to create, edit and delete AR models which can be viewed in the mobile application.

For this framework we propose two types of AR-asset; image and sequence. An image is a single 2-dimensional asset that can be shown as a flat object in a 3-dimensional AR environment. A sequence is essentially multiple image-frames that can be shown in an animated fashion much like a video. In an AR environment, sequences can be displayed exactly how an image would be shown, whereas stepping through the sequence would swap the image currently being displayed in the environment.

4.1 Mobile Application

The mobile application should consist of two main parts. The first is a simple way for users to gain access to an AR model created by a teacher or professor. This can be achieved in multiple different ways according to the use case of the application. For example, consider the situation where a teacher is presenting some work to a class room and all students should be able to see the same AR model at the same time. The easiest way to achieve this would be some direct link to the immediately relevant model, for example a code one can enter or a QR code one can scan. However, if it is important that the students can bring the application home and further study on their own, it will be impractical to demand a code or QR code to view the AR model. In this case it would be more prudent to simply show a list of all available AR models the students can choose from.

The second part of the mobile app is the AR view. This view should appear once the student has chosen a specific AR model. The app should then identify planes in the surrounding environment that the AR model can be placed upon. A marker should be shown to clarify for the student that the model is ready to be placed in an acceptable position. The student should then be able to press a *place* button to choose where to position the AR model. Once the model has been generated in the environment, it should be viewable from different angles in the room while remaining in the same position relative to the surface it was originally placed on. This means the student should be able to walk around the room while inspecting the model from different angles.

When viewing a sequence in the AR view, the student should be able to move between the different images in a user friendly manner. This should let the students easily examine different stages of a process or differences between structures without having to leave the AR view.

4.2 Admin Panel

The admin panel should help teachers and professors to create models their students can view in the mobile application. The core part of this web-application is a canvas in which the teacher can drag and drop different building blocks to make models for their students. The canvas should be easy to use and have access to blocks that are useful for the teacher to create intuitive AR models. For the minimum viable product (MVP) only 2D models need to be supported. Each block should be scalable and rotatable. Additionally, teachers should be able to change colors and borders of the building blocks. In case the teacher needs more specific blocks, they should be able to upload their own images and use these in the same way.

Furthermore, the teacher should be able to create multiple of these images and submit them as a sequence. For example, if a professor creates a computer sorting algorithm as a sequence, a student may visualize each step of the algorithm in AR, and in turn see how the algorithm unfolds over time.

After creating the AR model a teacher should be able to give it a name and save it. With no further work the model should be available in the mobile application. Either by showing up in the list of all available AR models or via a code the teacher can distribute to students.

5 Methodology

The goal of this project is to produce a framework that may be useful in educational contexts. Because there has been little direct research on what is required for a successful framework of this sort, determining the needs during development is a component of the development process. Therefore, we adopted the *design science* [7, 17] methodology for our task.

When developing using the design science methodology one follows a pattern of implementation followed by evaluation. Various approaches are tested and evaluated. If they do not work, changes are discussed, implemented and reevaluated until an acceptable solution is found. This worked particularly well for us as the broader idea of the framework was well thought out, but no plan for the implementation had been created.

First, a rudimentary mobile application was created with basic AR functionality. This created a *learning by creating* [19] environment. In other words, we were able to learn about the technology used to construct the mobile application while tracking down poor design decisions and redesigning the mobile app accordingly.

As we are not trying to construct entirely new technology, but rather combine existing technologies to fill an area of research that has not been filled, design science is a good methodology. Rather, current technology must be integrated, repurposed, and recombined in innovative ways to address challenges that have yet to be solved. This helps us focus on the goal for the framework - increasing learning gain by using new, innovative technology in the class room. The current technology then becomes a facilitator for the construction of this new framework.

5.1 Architecture

The main contribution this paper provides is an architecture for an AR application with an accompanying drag and drop web interface. This section suggests such an architecture.

5.1.1 Mobile application. This architecture will show a list of all available AR models, rather than using a code or a QR code to access each individual model. Therefore, the mobile application should contain two views. A view that lists all of the available AR models, and a view containing the primary AR-environment showing a particular model.

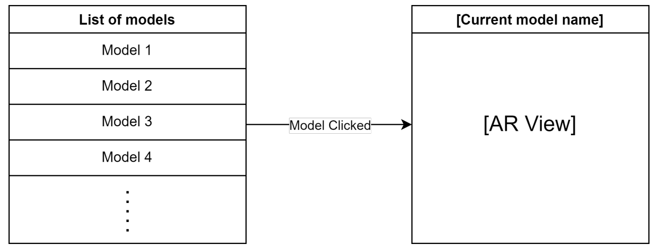


Figure 1. Mobile application user flow

The list should display all AR models that are available. These should be intuitively named by the professor who created them so that it is easy for the user to find the model they are looking for. Additionally, sequences should be split from regular image models. Either by using two lists in the same view or by adding an icon to the relevant list entries. In this architecture two separate lists are created.

By tapping a model in the list the user should be sent to the AR view. This screen should let the user perform the actions presented in the requirements part of this paper. Including, seeing the model from different angles, moving back and forth in image-sequences and placing the model on different surfaces.

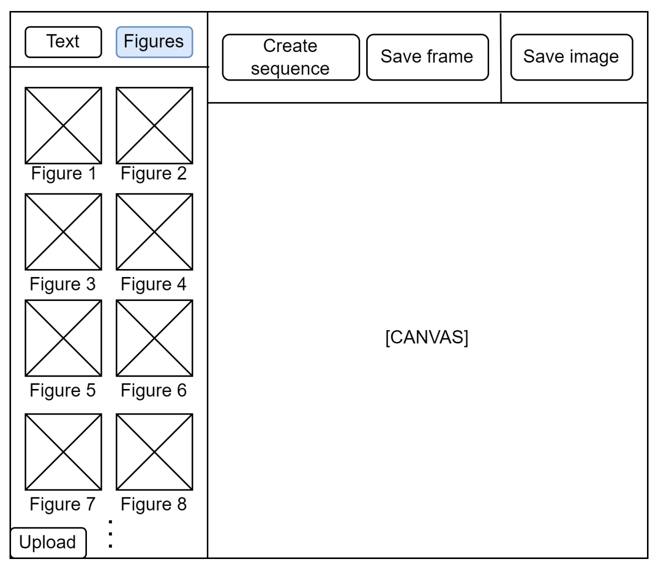


Figure 2. Admin panel low fidelity prototype

5.1.2 Admin panel. To enable users to easily and efficiently create graphics to be displayed in AR, an intuitive admin panel had to be created. In the admin panel, users should be presented with two options, creating a sequence of images or creating a single image to be displayed in AR.

Figure 3 depicts our proposed user flow for creating either an image or a sequence.

To build elements on the canvas, users should be able to select relevant generic figures or text-elements from a sidebar. An image-upload function will be necessary should a user desire more advanced or personalized graphics. A low fidelity prototype is presented of such an admin panel, as portrayed in figure 2. A sidebar should be included where a user can toggle between either text or figures along with an upload button users can press to open a file-upload menu. A top-bar should be included displaying buttons toggling the actions shown in figure 3. The canvas should reside in the main portion of the page, allowing plenty of space for users to build graphics.

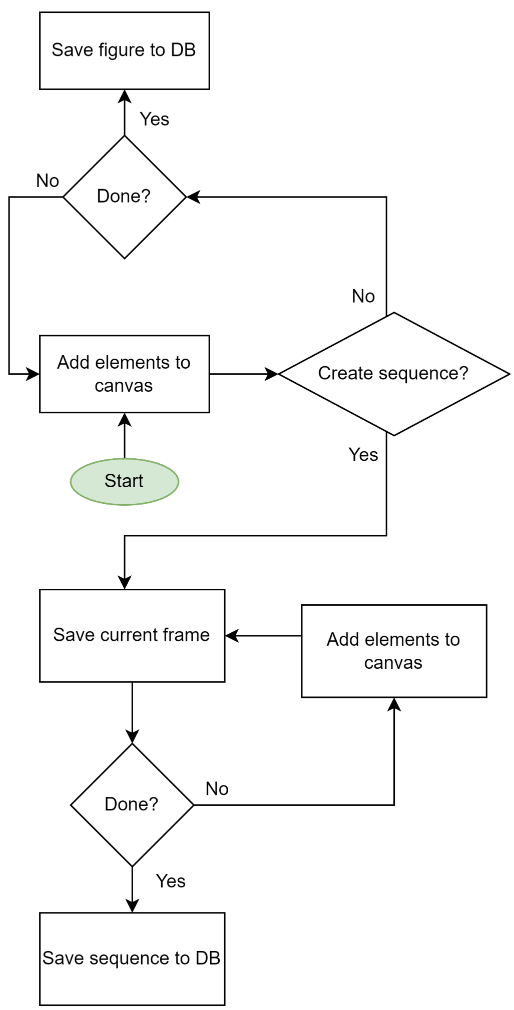


Figure 3. Admin panel user flow

5.1.3 Web API.

For seamless communication between the mobile app and the admin panel, a web API should be created. Our proposal for the flow of data between the three parts of the framework is as follows:

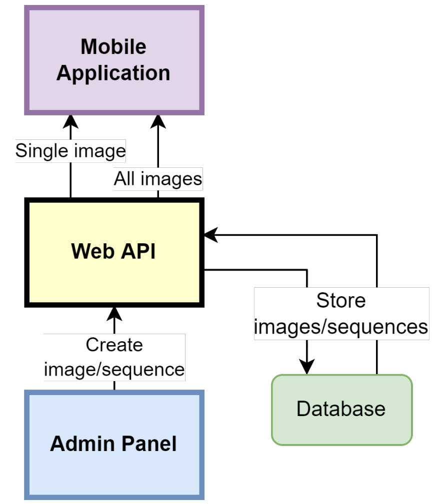


Figure 4. Flow of images and sequences between the three parts of the framework

The web API connects the other pieces of the framework by getting the requested data from the database or saving new or changed images and sequences to the same database. The web API will need at least three endpoints. Firstly, a teacher should be able to send data from the admin panel which the web API receives and saves. Secondly, the mobile application needs to be able to display a list of all available images and sequences. Once a user has selected one of the images from the list, the rest of the data should be fetched from the final endpoint, which returns a single image or sequence to the mobile application.

The database is connected to the web API and stores meta-data and image data for each image and sequence. For this type of data storage multiple database paradigms are viable. For example, MSSQL with Azure [16], NO-SQL with MongoDB [17] or PostgreSQL [18].

This is the minimum viable product for a framework for education using AR. There are, however, several ways to extend this structure. For example, adding a caching service like Redis [19] between the web API and the mobile application might reduce loading times in the mobile application when there is a significant amount of concurrent users.

6 Prototypic Implementation

This paper specifies a framework for using augmented reality in education based settings. Additionally, in this part of the paper, our own implementation of said framework is explained. The code for the implementation is located on

[16] https://azure.microsoft.com/en-us/products/azure-sql/database/#overview
[17] https://www.mongodb.com/
[18] https://www.postgresql.org/
[19] https://redis.io/

our organizations GitHub [20] and is open for use. Community contributions are greatly appreciated. Furthermore, a video demonstration can be found on YouTube [21].

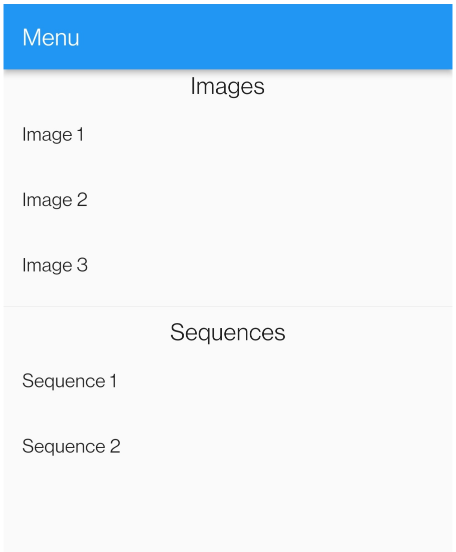
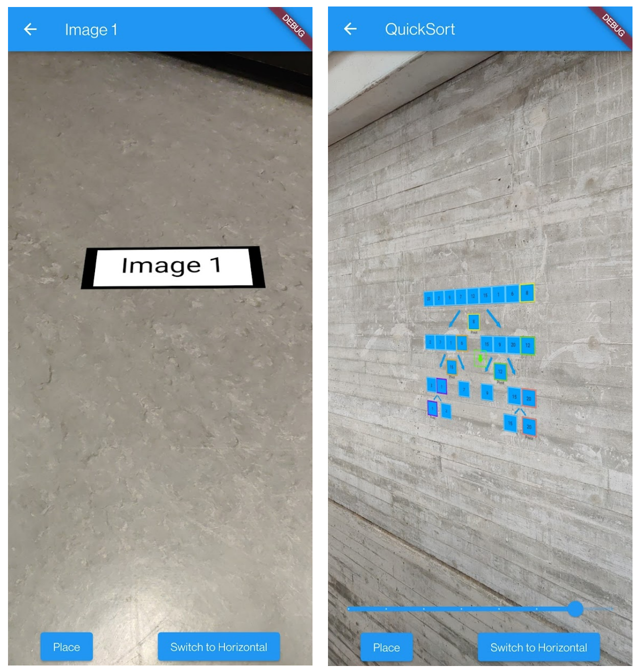


Figure 5. Selecting an image or sequence from the menu screen in the mobile application

6.1 Mobile Application

The mobile application needs to be able to handle both major mobile device platforms and fulfill the previously proposed architecture. Due to limited time and resources, native application development for both platforms was discarded. Rather, the multi-platform framework Flutter was chosen to implement and compile the application to both mobile operating systems from a single code base. Additionally, an AR framework was needed to handle the main feature of the application. Namely visualizing images and sequences of images in an augmented reality view using the smart devices camera. ARKit and ARCore are the two main AR drivers for iOS and Android, respectively. These engines are not directly integrated with the cross-platform nature of Flutter. Therefore, if they were to be used directly, the development would have to be split between the two to achieve full support on both mobile operating systems. To solve this problem the Unity AR Foundation [22] API was chosen to handle multi-platform AR tasks with a single code base.

The mobile application can then be split into two parts. The Flutter part of the application serves to show the user a list over all available images and sequences and lets the user click the one they wish to examine. When an image or sequence has been chosen, Flutter communicates the id for the current model to Unity. The model is loaded in Unity and the AR view is instantiated. On top of the AR view the user can press a button hosted by Flutter which communicates to the Unity application that it should place the image in the current position. When the image has been placed the Unity AR Foundation application handles the heavy lifting needed to produce a realistic AR view. AR Foundation exposes simple APIs for detecting surfaces upon which the models can be placed, moving them around the room and updating the model in the case that a new model is chosen.
Figure 5 shows the landing page of the mobile application. The user is presented will all available images and sequences in two separate lists. They can then tap one of these to get access to the AR view presented in Figure 6.



(a) Images

(b) Sequences

Figure 6. The AR view after an image or sequence has been chosen from the menu in Figure 5

Figure 6 depicts the two different views used for inspecting AR elements using a smart device camera. In both cases the placement indicator is presented as a green rectangle on the surface. In Figure 6a the indicator is hidden behind the model. In Figure 6b the indicator can be seen on the wall in the middle of the image. This indicator is only visible when a viable surface has been detected and the model can be placed. Using the *place* button located near the bottom of the screen, the model may be placed such that it sticks to the detected surface.

In Figure 6b there is an additional bar above the place button. This lets the user navigate between the images in a sequence. In this figure the user is displaying the second to

[20] https://github.com/Augmented-Reality-for-Education
[21] https://youtu.be/GXYz2oXtxzo
[22] https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1

last image in the current sequence. Sliding the bar updates the image automatically, eliminating the need for the user to re-place the model.

As the *place* buttons, the sequence slider and the menu are hosted in Flutter it is vital that this information can be communicated to Unity AR Foundation effectively. This is done using *flutter_unity_widget* [23]. This widget lets Unity be hosted within Flutter and allows for messages to be sent between the two. Methods are defined in Unity and can then be called by method name from the Flutter code. Figure 7 depicts how data is exchanged between Flutter and Unity.

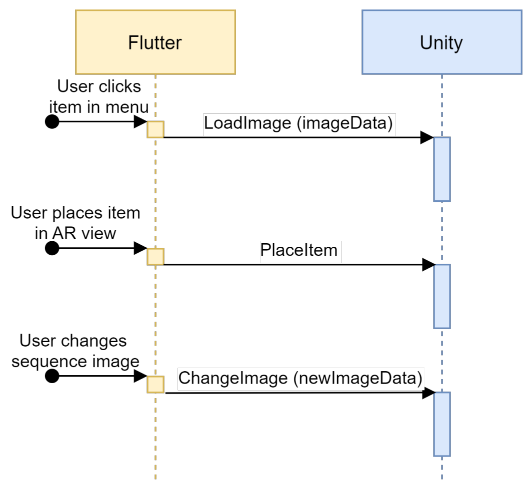


Figure 7. Sending data from Flutter to Unity

Firstly, the user selects an image or a sequence from the menu in Figure 5. The Unity instance is immediately loaded and the *LoadImage* [24] method is called with the image data. For sequences the image data for the first image in the sequence is communicated. Now that the Unity instance is loaded the user is able to see the view in Figure 6. When the user presses the place button to place the item in the AR view a new function in the embedded Unity application is called; *PlaceItem*. This time there is no need for extra data to be communicated as the Unity application has the image data from the *LoadImage* step. If the user selected a sequence from the menu they are also able to change the image by sliding the bar close to the bottom of the screen in Figure 6b. This calls the *ChangeImage* function in Unity, which communicates the new image data. The image in the AR view is instantly replaced with the new image, and the user does not have to place the object again to see the change.

Unity AR Foundation handles the AR view of the application. It incorporates extensive functionality for AR applications while taking advantage of the sophisticated game development engine on which it is based. The first Unity AR Foundation functionality this implementation of ARDENT utilizes is the ability to detecting planes. To place the AR model is has to be connected to a single plane at any angle. To detect if the camera is currently pointing at a valid plane the application uses the Unity *Raycasting* [25] functionality. A Raycast is an invisible ray which is sent from an origin in a specific direction. This can be used to detect collisions with other objects in the scene and is important when creating interaction between objects. In this case Raycasting is used to find the point directly in front of the camera which is part of a valid plane the model can be placed on.

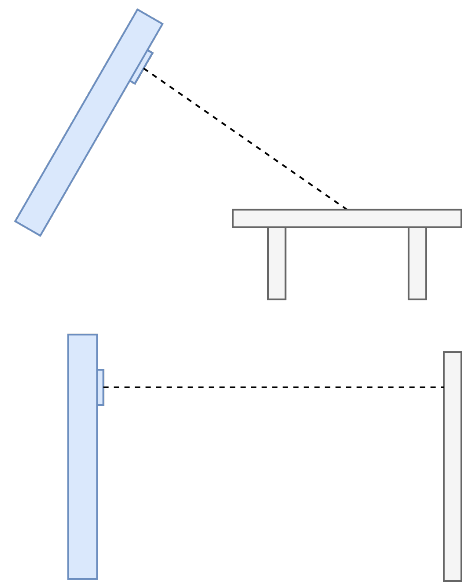


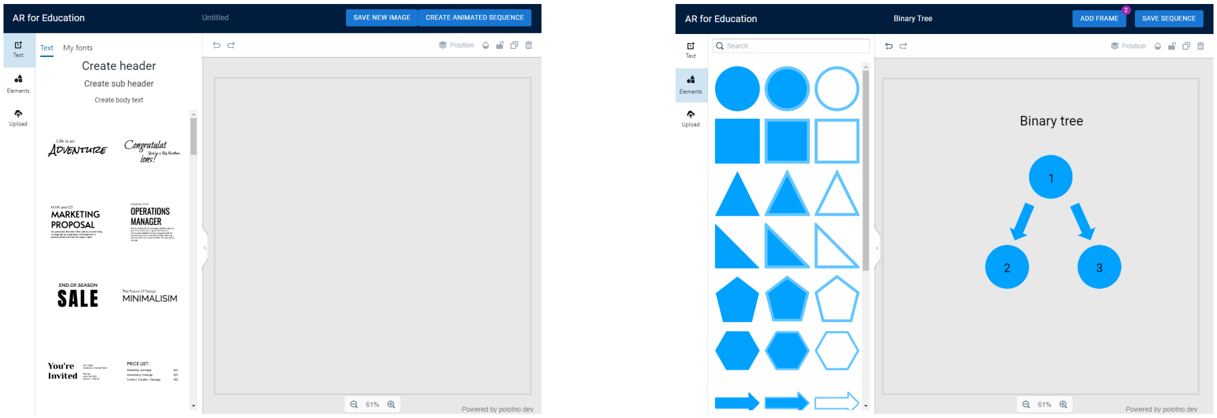
Figure 8. Raycasting from mobile camera onto different surfaces

Figure 8 displays how the blue mobile phone casts a ray directly forward onto the first surface it hits. If this surface has been deemed a valid plane by the Unity API, the green placement indicator in Figure 6 will appear. When the place button is pressed the application will record the position and rotation of the plane the ray has hit and instantiate the AR model on the surface.

6.2 Admin Panel

Creating graphical assets to be shown in AR through the mobile application has to be easy and intuitive. To enable easy access to an editor of this sort, a web-based graphical interface was decided upon. By utilizing the power of the modern front-end framework React.js, we were able to quickly build an interface with the help of the canvas-builder library Polotno [26]. Polotno is by design created to let developers quickly and seamlessly build canvas editor applications

[23] https://github.com/juicycleff/flutter-unity-view-widget
[24] Some function names have been simplified for this paper and do not directly correspond to the names of the functions in the code.
[25] https://docs.unity3d.com/ScriptReference/Physics.Raycast.html
[26] https://polotno.com/



(a) Initial canvas

(b) Binary tree created in the admin panel

Figure 9. Admin Panel Prototype

directly in a browser. Polotno provides an initial boilerplate editor, which in turn could be altered to suit our use case.

When opening the admin panel, as shown in figure 9a the user will be greeted with an empty canvas and the sidebar showing a default menu of the available text types. If the users adds elements and decides to save it as a single image, simply pressing the *SAVE NEW IMAGE* button will deploy the current canvas to the database, making it available in the mobile application.

Figure 9b shows the prototype of the admin panel, where in this case a simple example of a binary tree has been built to illustrate a potential use-case. In this case, the user has decided to save the binary tree as a sequence. This is initialized by first pressing the *CREATE ANIMATED SEQUENCE* button and thereafter saving each frame of the sequence individually.

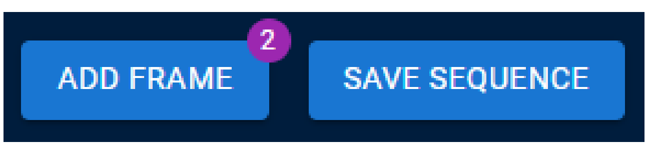


Figure 10. Adding frames to a sequence

Figure 10 shows how a counter is incremented each time the user adds a new frame to the sequence. When the user decides that the sequence is done, pressing the *SAVE SEQUENCE* button will save the sequence to the database, and makes it instantly available on the mobile application, similarly to the process of saving as an image.

6.2.1 Libraries and tools. Instead of using plain JavaScript, for this project we utilized TypeScript[27]. TypeScript is a strongly typed programming language that is a superset of JavaScript, adding additional syntax and provides a tighter integration with the IDE.

To connect to the .NET Web API, *Axios*[28], a promise based HTTP client for the browser was implemented. Axios provides a simple interface for making HTTP requests from the browser, working hand-in-hand with our back-end API.

To create a visually pleasing website, the Material Design component library (MUI)[29] was used. MUI offers a vast library of prebuilt components allowing rapid development of new features, shifting the focus from design to building the actual logic. MUI was used in conjunction with Styled-Components[30]. Styled-components lets you use component-level styling in the application and enables using JavaScript to define CSS (Commonly refered to as CSS-in-JS).

6.3 Web API

The web API works as a connector between the mobile application and the editor in the admin panel. Additionally, it utilizes an SQL database to store images, sequences and meta data. For this web API we opted for .NET 6[31]. The connection with the database is handled by Entity Framework Core[32], a data access technology which simplifies communication between .NET applications and databases.

6.3.1 Domain. The API has to define the domain of the application. For this implementation there are two classes, *Image* and *Sequence*, which are represented in Figure 11. The Image entity is the core of the domain. It stores some metadata about each image. While only *name* is depicted here it also stores time of creation and time of last modification. As these are not imperative to the application they are not further explained. The most important field of this

[27] https://www.typescriptlang.org/

[28] https://github.com/axios/axios

[29] https://mui.com/

[30] https://styled-components.com/

[31] https://dotnet.microsoft.com/en-us/

[32] https://docs.microsoft.com/en-us/ef/core/

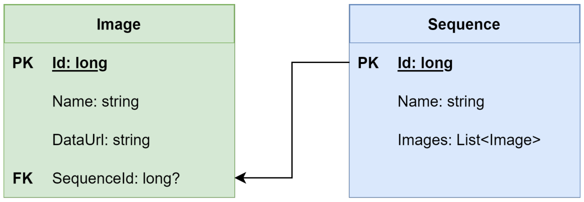


Figure 11. Entities and their relationship

class, however, is the *DataUrl*. DataUrls were specified by L. Masinter in 1998 [18]. Their main functionality is storing binary data in a simple string. The DataUrl string is made up of four fields that are presented at the start of the string and define what type of data the DataUrl stores. Data types are defined using the same *mimetypes* as in regular HTTP requests[33]. Examples are *text/plain*, *video/webm* or *image/png*[34]. Storing image data using DataUrls removes a layer of complexity. Production ready applications should use more robust storage options like a cloud providers blob storage or local file storage. However, for this simple implementation the DataUrl provides the exact functionality needed to store images without further complexity.

The sequence class is an extension of the image class. It provides meta data for each sequence, just like the image class does for images. The core part of the sequence class is an ordered list of images. As illustrated in Figure 11 the SQL primary key (PK)[35] links images to sequences using an SQL foreign key (FK)[36]. Each image also has the field *SequenceId*. If this field is filled with a number we can identify which sequence the image belongs to. If the SequenceId is undefined, however, the image is considered a standalone model and not part of a sequence. In this way multiple images can store the same SequenceId which links them all to the same sequence. Finally, the order of the images in a sequence is determined by the time the image was created at. The first image in a sequence when a new sequence is created will be created first, then the second and so on. This will keep the images in every sequence in the right order in the database.

6.3.2 API structure. A popular approach to structuring web APIs is by using the *controller-service-repository* pattern. This pattern aims to split the application into three different layers which the data flows through. The parts are:

The *controller*, the part that external resources can communicate with, ensures that each entry point has the correct URL and returns the correct HTTP response code[37]. It also ensures that each request executes the correct method on the correct *service*.

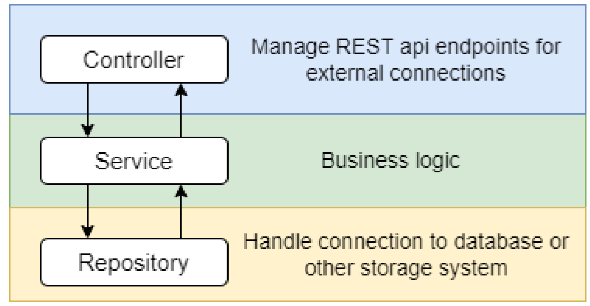


Figure 12. Visualization of the controller-service-repository pattern

Services handle all the business logic of the application. In this case the business logic is rather simple and only handles creation and fetching of images and sequences. To be able to create, read, update or delete information in the database, the service has to utilize the *repository* layer.

Repositories solely handle the connection the application has to any databases or storage systems. This implementation of the ARDENT framework uses Entity Framework Core (EFCore) to query the database. EFCore provides data structures and functions which utilize the strong typing in C# while keeping the code more semantic than pure SQL. This speeds up the development process as well as reducing the likelihood of falling into the many pitfalls of plain SQL with larger databases [14].

6.3.3 API endpoints. Web APIs define *endpoints*, which are the gateways external applications can communicate through to access the functionality of the API. As depicted in Figure 4 this specific web API needs at least three endpoints; one for getting a single image, one for getting the list of all images and sequences and one for creating an image or sequence. To follow the structural pattern presented in section 6.3.2 the application is split into separate logic for sequences and images. This results in the following API endpoints.

Image Controller

Get By inputting an image id a single image is returned. The purpose of this endpoint is to load a single image with all data into the AR view.

GetAll This endpoint returns all images, but without the binary image data. No input is needed. The purpose of this endpoint is to swiftly load all available images into the menu so users can choose the model they want to inspect. This endpoints does not return the images that are connected to a sequence.

Create The create endpoint needs an image name and the binary image data as input and saves the image to the database. This endpoint is used from the editor/admin panel.

[33] developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URLs
[34] developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types
[35] https://www.w3schools.com/sql/sql_primarykey.ASP
[36] https://www.w3schools.com/sql/sql_foreignkey.asp
[37] https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Sequence Controller
GetAll This endpoint returns all sequences. Similarly to the same endpoint in the image controller no binary image data is included and no input is needed. This endpoint is used for displaying all sequences in the menu view.
Create The create sequence endpoint needs a sequence name and multiple images as input. Each image is saved separately, in the same manner as for individual images, and linked to the sequence. This endpoint is used from the editor/admin panel.

Firstly, it is noticeable that the sequence controller has no ***Get*** endpoint. This is because in the current implementation there is no need for one. When the metadata for all sequences are loaded into the menu the user can choose a sequence. When the AR view loads, the Flutter application uses the image ids stored in the sequence to fetch one image at a time via the ***Get*** endpoint on the image controller. This reduces loading times and the user will never load images they are not going to view.

Secondly, it is also noteworthy that the ***GetAll*** endpoints does not include the binary image data. For this implementation of the ARDENT framework we discovered that more than 99.95% of storage needed for each image was a result of the DataUrl string. Therefore, loading times can be vastly reduced by only loading the data for the selected images when it is needed. As of now this is not a problem, considering that the current implementation has only been tested by a handful of users. However, this might very well have become a huge hurdle in later development, should the consecutive user count increase manyfold.

6.4 Integration

Sections 6.1, 6.2 and 6.3 present our individual implementation of the three parts needed to run this framework. The API controls the domain and business logic, as well as the storage of images and sequences. The mobile application lets users place these images and sequences on a myriad different surfaces and allows for inspection from all angles. The admin panel enables intuitive creation of images and sequences with multiple different shapes, fonts, colors and the ability to upload your own image files.

For this framework to provide any value, however, these parts need to be interconnected and work together. The most popular and robust way to handle communication between applications over the internet is the REST web standard, which is built on the HTTP protocol. REST allows the three applications to exchange data in two-way communication. All the following data-exchanges utilize the REST web standard. Figure 13 visualizes the simple flow of data when a sequence is created in the admin panel.
The first procedure starts when a user in the admin panel creates a new sequence. When all the frames of the sequence have been designed and the user is happy, they press the save

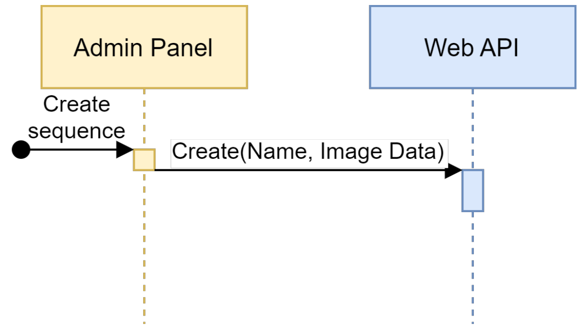


Figure 13. Sending data to the web API to create and save a sequence

sequence button. This starts the communication between the admin panel and the Web API. The sequence name and all the images are communicated via REST and handled by the Web API, as described in section 6.3.3. Once one or more sequences have been created these will be available in the menu in the mobile app. Figure 14 describes the flow of data between the Web API and the mobile application to display a sequence in the AR view.

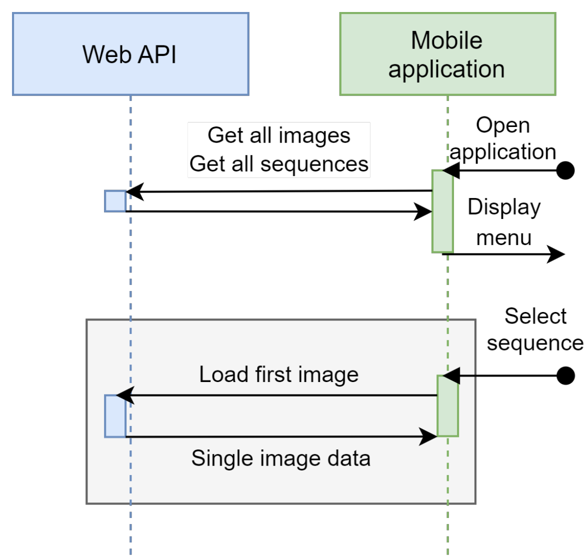


Figure 14. Communication between the mobile application and the web API to view a sequence in the AR view

Firstly, when the user opens the application on their mobile smart device, they are presented with a skeleton menu view with no data. At this moment two requests are sent from the mobile application to the web API requesting the metadata for all images and all sequences. When this data has returned, the menu view can be updated and the available images and sequences are displayed to the user.

The user is now able to select the image or sequence they would like to inspect in AR. When a sequence is clicked the

Unity player starts loading, as described in 6.1. While the Unity player is loading another request is sent to the Web API. At this time the image data for the first image in the sequence is fully loaded onto the smart device, as described in 6.3.3. This procedure is encapsulated in gray in Figure 14. When the user is done examining the first image in the sequence, they can move back and forth between all images that are a part of said sequence. Whenever the user moves between these frames the encapsulated procedure in the figure is repeated. The mobile application requests a new image from the server and the old image is replaced.

7 Evaluation and Assessment

To evaluate not only the framework, but also the prototypic implementation, a demo was prepared and showcased at the University of Stavanger. At the ARDENT stand, students and professors were able to test both the admin panel and the mobile application. Multiple students and professors provided valuable praise and constructive criticism.

Most commonly among the constructive criticism was a desire for a better user experience and user interface. It should be easier and more intuitive for professors to create models in the admin panel. Similarly, the mobile app should be simpler and more accessible for students. Finding your way from opening the app to the content you are looking for when studying should be trivial. In the AR view, it should be intuitive for the user to know when and how the object can be placed. These problems can be solved with more intuitive design, following well researched industry standards or by implementing a tutorial for users so they learn how the app is supposed to be used when they first try it.

Multiple professors found the idea of using the admin panel exciting and intuitive. The same professors, however, failed to see the inherit value in the product in its current state. Nevertheless, these same respondents all agreed that there is massive potential if the application is further developed. Some of their responses are further discussed in section 8.1. In its current state these professors questioned how this view will be more beneficial to the students than a simple image in a textbook or an animation on a PowerPoint slide. To this some of the questioned students answered that by using such an app, users might be motivated to play with the model for longer, and as such could end up learning more than they would by reading traditional PowerPoint slides and textbook examples.

In the demo presented at the University of Stavanger, a simple multi-step sequence of the QuickSort[11] algorithm was presented. Multiple students responded that this specific example, although excellent for showing off the technology, might not be the best way of representing the potential of the application. This was further argued by there being better examples of this algorithm on other platforms; namely YouTube, in textbooks and on multiple educational algorithm websites [38][39]. However, these students were sure that if the correct content was presented they would be able to learn more efficiently using AR than with ordinary 2D images. Either way, the exact domain this application is useful in in its current state remains to be researched.

Finally, surprisingly few participants suggested extending the application to using 3D models. When asked about this every single student agreed that this would vastly increase the usability of the app. Most of the students from the study are in the field of computer science and electrical engineering. Some of these people could not instantly recognize any fields within their own studies which this would be inherently useful. However, all of them were able to mention other fields of study that could take advantage of this technology. Amongst others were medicine, constructional engineering and mathematics. When the professors were asked whether they would use the application if 3D models were supported, the response was positive, but two-sided. All of them agreed that extending to 3D might increase learning gain. Unlike the students they also aired concerns about the complexity of creating these 3D models, compared to the simple admin panel currently used for 2D models.

In summary, the views of like minded students and professors from our university line up well with our own internal evaluations of the application during development. The responses suggest that this implementation of ARDENT might be useful in the right scenarios. The same respondents agree that an expanded implementation of ARDENT with an improved upon user interface and better models would be embraced by students and utilized by professors.

8 Conclusion

Throughout this paper we have presented the architecture and a prototypic implementation of the ARDENT framework, an Augmented Reality Framework for Education. The aim of the project has been to improve academic results through the utilization of a quick and easy-to-use framework for showing educational content in AR. The project has been focused around two primary interfaces; the mobile application students will use to display content in AR, and an admin panel where professors easily can build 2D graphics to be instantly shown in the mobile application. Through our study of related works, we discovered a notable absence of AR applications of this sort as opposed to similar applications using VR. Furthermore, while there exists applications utilizing AR for educational purposes, we identified no projects or frameworks taking advantage of building a related admin panel for creating graphical assets.

With a demonstration of the framework and its prototypic implementation, we performed a user evaluation with students and professors at the University of Stavanger. The

[38] https://www.geeksforgeeks.org/quick-sort/

[39] https://www.programiz.com/dsa/quick-sort

study's general consensus was that with additional improvement and further development, the framework might become a very helpful educational tool.

We may conclude that this framework is a tremendous step in the right direction, with significant potential for additional development and research. The prototypic implementation is built on the standardized development architecture with a .NET Web API on the back-end and the modern front-end frameworks Flutter and React.js for the mobile application and admin panel. This implementation serves as the foundation for further development, whereas building additional features should be seamless provided this robust setup.

8.1 Further Work

The value from this framework is currently limited by the amount of dimensions the AR model can be created in. Increasing the model dimensions to 3D increases the learning possibilities manyfold. For example, medical students studying internal organs, constructional engineers observing schemas which display load balancing, or mathematics students inspecting 3D matrix operations and complex visualizations of three dimensional graphs. However, increasing the dimensions also vastly increases the complexity of the admin panel for the teacher. Creating 3D models in the browser is much harder than 2D models and one might have to hire experts to be able to create what is needed. Either way, the advantages outweigh the disadvantages and increasing the number of dimensions by adding 3D models is the obvious next step for this framework.

Furthermore, adding multiple models in the same view and having them interact with each other can in many cases be helpful. Medical students could be able to compare lungs of smokers with those of a non-smoker or constructional engineers could inspect the differences between multiple bridge designs. To take this even further, one could use the Unity physics engine to have objects interact with each other. Biology students could then inspect every single instance of what happens to the human body during a car crash while physics students could use sophisticated 3D models of complex solar systems to understand how planets pull on each other.

9 Acknowledgements

Firstly, we would like to extend our thanks to Antorweep Chakravorty, our guide, counselor and supervisor from the University of Stavanger. He helped us tremendously with the language and structure throughout the project, as well as the finishing touches when approaching publication.

We have to quickly thank Brandon Sanderson for providing inspiration when we chose the name for the framework.

Finally, than you to the participants of the study we completed to discover what this framework needs to be the future of immersive education; Odin Bjørnebo, Tor-Fredrik Torgersen, Brynjar Ulriksen, Trym Ullestad, Håvard Godal and multiple unnamed students and professors at the University of Stavanger.

References

[1] Ismo Alakärppä, Elisa Jaakkola, Jani Väyrynen, and Jonna Häkkilä. 2017. Using Nature Elements in Mobile AR for Education with Children. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, Article 41, 13 pages. https://doi.org/10.1145/3098279.3098547

[2] S. AlAwadhi, N. AlHabib, D. Murad, F. AlDeei, M. AlHouti, T. Beyrouthy, and S. Al-Kork. 2017. Virtual reality application for interactive and informative learning. In *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*. 1–4. https://doi.org/10.1109/BIOSMART.2017.8095336

[3] Tim Althoff, Ryen W White, and Eric Horvitz. 2016. Influence of Pokémon Go on Physical Activity: Study and Implications. *Journal of Medical Internet Research* 18, 12 (Dec. 2016), e315. https://doi.org/10.2196/jmir.6759

[4] Hadi Ardiny and Esmaeel Khanmirza. 2018. The Role of AR and VR Technologies in Education Developments: Opportunities and Challenges. In *2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM)*. 482–487. https://doi.org/10.1109/ICRoM.2018.8657615

[5] Samuel J. Bryan, Abraham Campbell, and Eleni Mangina. 2018. Scenic Spheres - An AR/VR Educational Game. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*. 1–9. https://doi.org/10.1109/GEM.2018.8516456

[6] Rogério Colpani and Murillo Rodrigo Petrucelli Homem. 2015. An innovative augmented reality educational framework with gamification to assist the learning process of children with intellectual disabilities. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 1–6. https://doi.org/10.1109/IISA.2015.7387964

[7] Aline Dresch, Daniel Pacheco Lacerda, and José Antônio Valle Antunes. 2015. *Design Science Research*. Springer International Publishing, Cham, 67–102. https://doi.org/10.1007/978-3-319-07374-3_4

[8] Rubina Freitas and Pedro Campos. 2008. SMART: A SysteM of Augmented Reality for Teaching 2nd Grade Students. 27–30. https://doi.org/10.1145/1531826.1531834

[9] Juan Garzón. 2021. An overview of twenty-five years of augmented reality in Education. *Multimodal Technologies and Interaction* 5, 7 (2021), 37. https://doi.org/10.3390/mti5070037

[10] Junjie He, Jiali Ren, Gaoxia Zhu, Su Cai, and Guang Chen. 2014. Mobile-Based AR Application Helps to Promote EFL Children's Vocabulary Study. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*. 431–433. https://doi.org/10.1109/ICALT.2014.129

[11] C. A. R. Hoare. 1962. Quicksort. *Comput. J.* 5, 1 (01 1962), 10–16. https://doi.org/10.1093/comjnl/5.1.10 arXiv:https://academic.oup.com/comjnl/article-pdf/5/1/10/1111445/050010.pdf

[12] L. Johnson, A. Levine, R. Smith, and S. Stone. 2010. The 2010 Horizon Report. (2010).

[13] R. Johnson, L.and Smith, H. Willis, A. Levine, and K. Haywood. 2011–. The 2011 Horizon Report. (2011–).

[14] Bill Karwin. 2010. *SQL antipatterns: avoiding the pitfalls of database programming*. Pragmatic Bookshelf.

[15] Lucinda Kerawalla, Rosemary Luckin, Simon Seljeflot, and Adrian Woolard. 2006. "Making it real": exploring the potential of augmented reality for teaching primary school science. *Virtual Reality* 10, 3 (01 Dec 2006), 163–174. https://doi.org/10.1007/s10055-006-0036-4

[16] Gun A. Lee, Andreas Dünser, Seungwon Kim, and Mark Billinghurst. 2012. CityViewAR: A mobile outdoor AR application for city visualization. In *2012 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH)*. 57–64. https://doi.org/10.1109/ISMAR-AMH.2012.6483989

[17] Salvatore T. March and Gerald F. Smith. 1995. Design and natural science research on information technology. *Decision Support Systems* 15, 4 (1995), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

[18] L. Masinter. 1998. *The "data" URL scheme*. https://doi.org/10.17487/rfc2397

[19] Briony J Oates. 2006. *Researching Information Systems and Computing*. Sage Publications Ltd.

[20] Melissa Regan and Sheri Sheppard. 1996. Interactive Multimedia Courseware and the Hands-on Learning Experience: An Assessment Study. *Journal of Engineering Education* 85, 2 (1996), 123–132. https://doi.org/10.1002/j.2168-9830.1996.tb00221.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2168-9830.1996.tb00221.x

[21] Nor Saidin, Noor Abd halim, and Noraffandy Yahaya. 2015. A Review of Research on Augmented Reality in Education: Advantages and Applications. *International Education Studies* 8 (06 2015). https://doi.org/10.5539/ies.v8n13p1

[22] Donna Satterthwait. 2010. *Teaching Science* 56, 2 (2010), 7–10. https://search.informit.org/doi/10.3316/aeipt.182048

[23] Yoana Slavova and Mu Mu. 2018. A Comparative Study of the Learning Outcomes and Experience of VR in Education. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 685–686. https://doi.org/10.1109/VR.2018.8446486

[24] Maged Soliman, Apostolos Pesyridis, Damon Dalaymani-Zad, Mohammed Gronfula, and Miltiadis Kourmpetis. 2021. The Application of Virtual Reality in Engineering Education. *Applied Sciences* 11, 6 (2021). https://doi.org/10.3390/app11062879

[25] Wahyu Widada, D Herawaty, Khathibul Umam Nugroho, and A Anggoro. 2021. Augmented Reality assisted by GeoGebra 3-D for geometry learning. *Journal of Physics: Conference Series* 1731 (01 2021), 012034. https://doi.org/10.1088/1742-6596/1731/1/012034

[26] Hsin-Kai Wu, Silvia Wen-Yu Lee, Hsin-Yi Chang, and Jyh-Chong Liang. 2013. Current status, opportunities and challenges of augmented reality in education. *Computers & Education* 62 (2013), 41–49. https://doi.org/10.1016/j.compedu.2012.10.024

[27] Yun Wu and Chetan S. Sankar. 2013. Impact of Hands-On Research Experience on Students' Learning in an Introductory Management Information System Course. *Decision Sciences Journal of Innovative Education* 11, 4 (2013), 335–358. https://doi.org/10.1111/dsji.12017 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/dsji.12017

[28] Jianghao Xiong, En-Lin Hsiang, Ziqian He, Tao Zhan, and Shin-Tson Wu. 2021. Augmented reality and virtual reality displays: emerging technologies and future perspectives. *Light Sci. Appl.* 10, 1 (Oct. 2021), 216.

[29] Kaissar Yammine and Claudio Violato. 2016. The effectiveness of physical models in teaching anatomy: a meta-analysis of comparative studies. *Advances in Health Sciences Education* 21, 4 (01 Oct 2016), 883–895. https://doi.org/10.1007/s10459-015-9644-7

[30] Steve Chi-Yin Yuen, Gallayanee Yaoyuneyong, and Erik Johnson. 2011. Augmented reality: An overview and five directions for AR in education. *Journal of Educational Technology Development and Exchange (JETDE)* 4, 1 (2011), 11.

[31] Egui Zhu, Arash Hadadgar, Italo Masiello, and Nabil Zary. 2014. Augmented reality in healthcare education: An integrative review. *PeerJ* 2 (2014). https://doi.org/10.7717/peerj.469