Project Report

Augmented Flight

by Nadine Kupitza & Laura Forster

Augmented and Virtual Reality

University of Applied Sciences Ingolstadt
Faculty of Computer Science
User Experience Design Master

Date of Submission: 14.01.2022
Lecturers: Prof. Dr. Munir Georges, Andreas Riegler, Tamara von Sawitzky

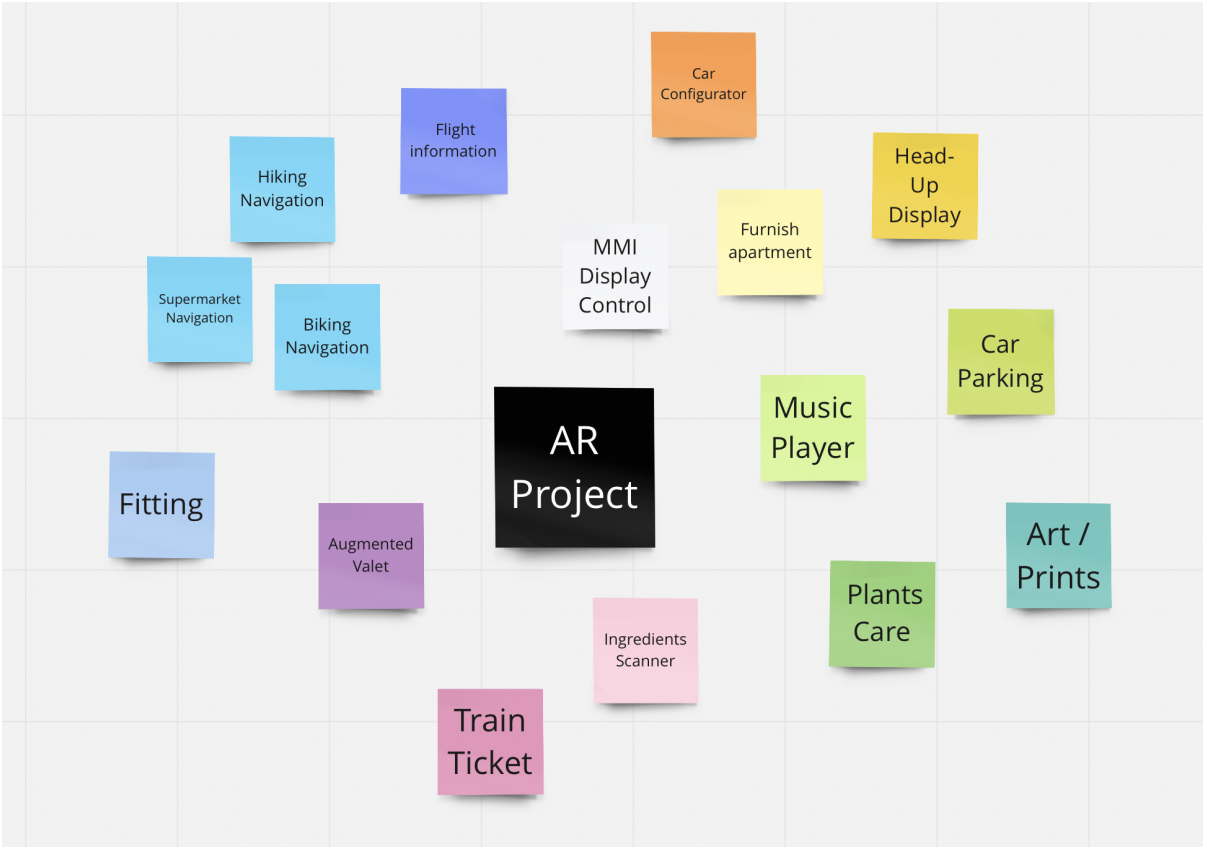
Table of Contents

1. Ideation

Ideation is the creative process of generating, developing and communicating new ideas. As such, it is an essential part of the design process. During this process we started with a brainstorming workshop to generate ideas and create a first project outline.

a. Brainstorming

During our brainstorming workshop, we collected our ideas on a Miro board. As we both already had experience with VR Projects in Unreal Engine, we focused mainly on ideas based on AR technology. We wanted to get deeper into its affordances and learn its creation process.
Since we both work in the automotive industry, our first thought was to develop something within this context. But in order to extend our portfolio and skills, we decided to open our minds also to ideas within other directions. We both agreed quickly on the idea of augmented airplane windows as it offers lots of opportunities within this project. During flights, usually lots of passengers, including us, like to sit at the window to see the world from another point of view. Unfortunately sometimes it is really hard to guess what country or city is exactly underneath the aircraft. Therefore Augmented Flight offers a nice solution which enhances the experience during a flight.



b. First Project Outline

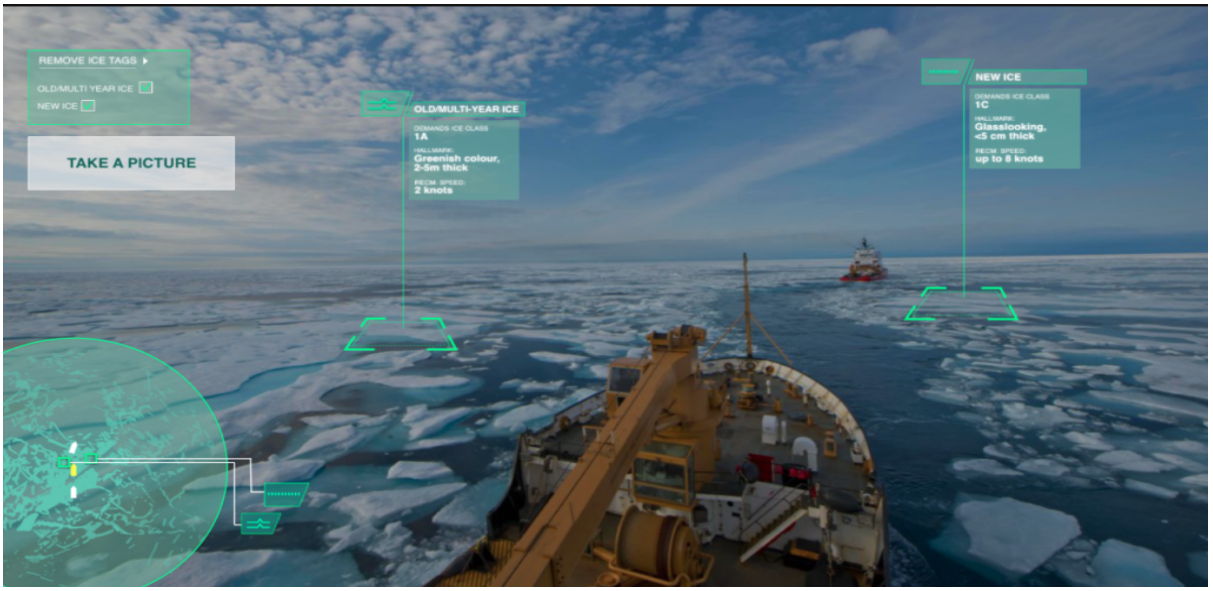
After we decided to use the Augmented Flight application, we started to define the concrete idea, motivation and proposed outcome. We also started to outline some requirements for the project and the Hardware and Software we want to use.

Ever since flying around the world has been open to the public, passengers always seeked to get one of those popular and rare window seats. Passengers enjoy looking outside the window and guessing where they are currently at. With the introduction of screens in the aircraft, on the back of the seats, it has become easier for passengers to determine where they currently are, which city it is, that they see at the horizon.
Even if the screens already offer more information about the flight and location, the experience still does not vary much from what we had in the past, and moreover, did not significantly develop further. Comparing it to the automotive industry, vehicle brands always try to integrate the newest technology and seek for the best experience for the users. In order to emphasize this, we would like to list the example of head-up displays, that offer the driver an extended reality and provide more information about the vehicle status and surroundings. So why not think about integrating the same technology and a similar use case into the context of aircrafts, to enhance the experience of passengers? With a head-up display, the windshield of the car serves as the final mirror, which casts the digital image onto the road. This content can be both static and augmented. In an airplane, the window would be the alternative to the windshield.

In our project we want to explore the opportunity of projection technology in aircraft windows, to display detailed information about the flight and location. We therefore use Augmented Reality as a prototyping tool, to explore and also test the opportunity to improve the user experience during flights. In the following weeks we will discuss our prototyping opportunities, like creating an app for tablets or using Microsoft Hololens. In reality, the use of projection will have advantages over AR technology implemented with glasses. For instance, the use and interaction will be more natural, when passengers can just look out of the window, instead of putting glasses on. Especially for people who wear glasses (about 64% of the German population) it will be easier and more comfortable to experience the information without any additional devices. Another argument is that providing AR glasses to passengers will certainly be expensive in procurement and cleaning, and economically, just makes no sense. Nevertheless we might consider AR Glasses for our project, in order to prototype the experience and create a very close result.

In order to exhaust all possibilities and create a natural and mind blowing experience, we will not only try to focus on an aesthetic and informative visualization, but also make it interactive. For this project, we use direct and natural interaction

oriented to the everyday experience of the users, which is also intuitive and an easy hand-eye-coordination. For the users, it appears as if they can interact directly with a virtual object. As an example, the user should be able to navigate through the information and decide what he or she wants to see. Talking about information, we consider adding information about countries and cities in general, but also informative facts about certain landmarks and buildings. So far, we have not found any similar project in the context of airplanes. Instead we found a very nice visualization in the context of ships, which we take as an inspiration and reference for our information visualization.



Source: Pinterest (https://www.pinterest.de/pin/536772849347252455/)

For creating the AR content for our "Augmented Flight" project we will use Unity. As already stated before, we will consider either a tablet or a hololens as a hardware device. Depending on the hardware, the interaction input will take place through gestures, controllers or touch.

2. User Research

User research is the methodic study of target users, including their needs and frustrations.

a. Personas

A Persona is a fictional character created to represent the user type that might use our product. By answering the following questions we clarified the target audience for our Augmented Flight Application:

1. Who is our ideal user?
2. What are the goals and needs of our users?
3. What are the frustrations of our users?

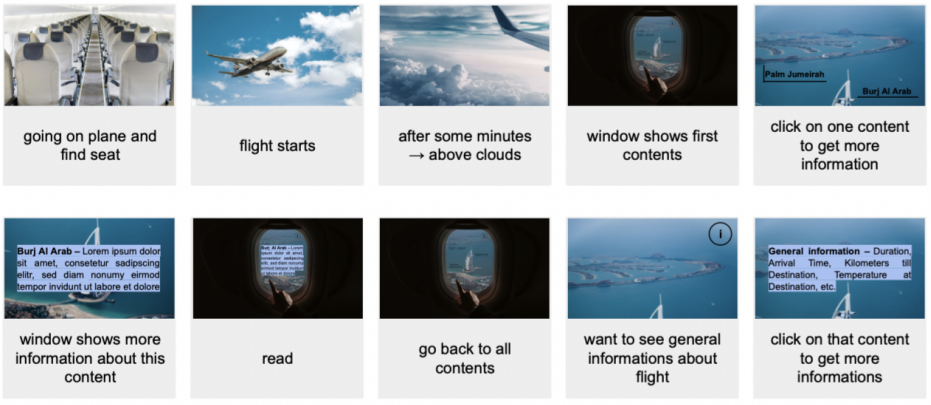
b. User Story

A user story is a simple and short description of a feature of your product. It is formulated in everyday language and told from the perspective of our personas Amanda and Brian.

“As a passenger I want to experience the flight in all its details. I want to understand how far from the ground I am currently at, which cities I am flying over and which sights I can see at the horizon. Augmented Flight extends the overall flight experience. It helps to use the time to destination efficiently, while making flights more pleasurable, educating and entertaining.”

c. Storyboard

A storyboard is a visualization of a concept or idea. It can help to visually predict and explore the user experience with a product.



d. User Requirements

User requirements are requirements set by the end users. These are written early in the validation process, typically before the system is created.

Functional requirements:

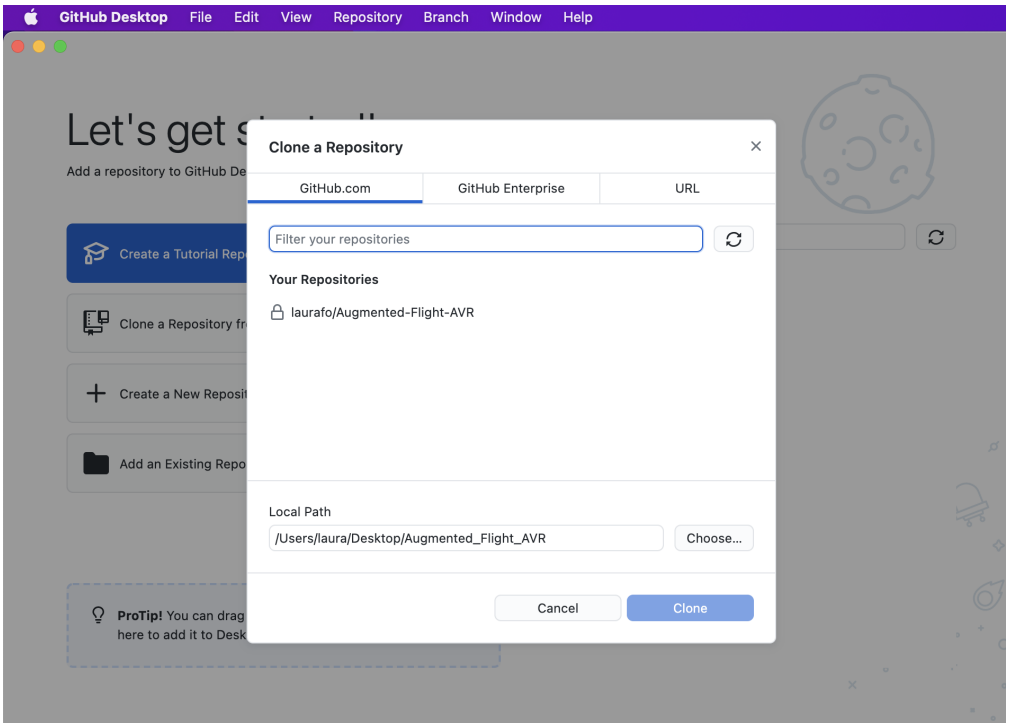
- Easy touch interaction
- Tracking cities, sights, etc.
- Filtering by different categories e.g. sights
- Showing general information (time to destination, altitude)
- Showing short information of the tracked items
- Showing detailed information when clicking on a item

Non-functional requirements:

- Colorful
- Entertaining
- Simple textes
- Satisfying

3. GitHub

In order to both have shared access to our project, we created a new repository on GitHub and included a .gitignore file so we do not waste space and upload folders that are automatically generated at startup either way. To share our work in GitHub, we both installed GitHub Desktop. After we installed GitHub Desktop, we were able to authenticate the application with our accounts on GitHub. Following this, we cloned our repository to our GitHub desktop. To do this, we selected "Clone Repository" under the File menu item in GitHub Desktop. Then a window with three tabs opened and under the first tab "GutHub.com" our repository was already displayed. We selected it and cloned it using the "Clone" button at the bottom right.



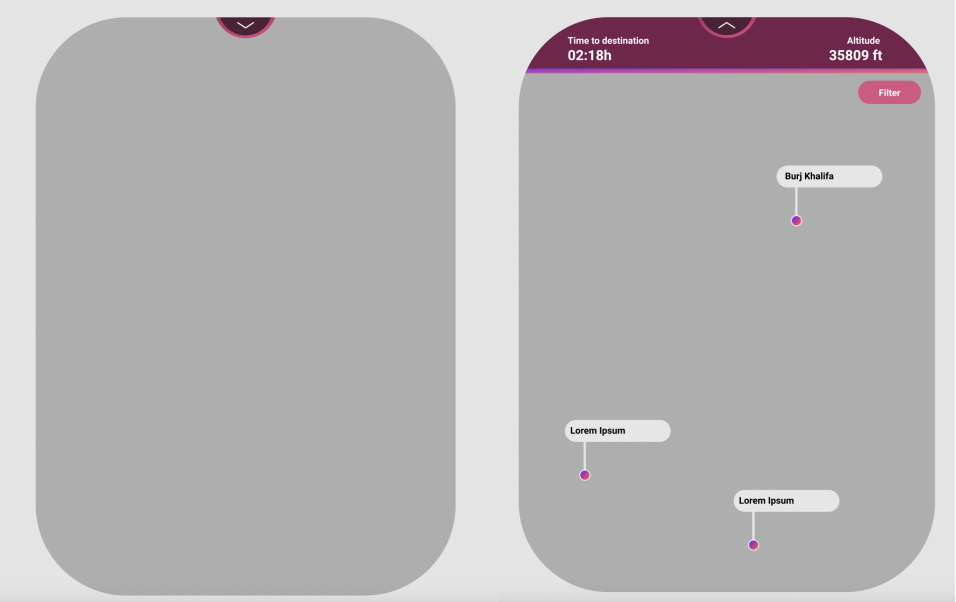
After that, GitHub created a folder, with our project, on our desktop and we could open the files locally and put new files in this folder. After we made changes to our project locally, we were able to review them in GitHub Desktop and make a commit. To share these changes within our team, we pushed our commit to GitHub. After that, the other team member was able to pull the changes to their own desktop via GitHub Desktop, allowing them to continue working on the latest version of the project.

4. Implementation

Since both of us have almost no experience with Unity, we did several tutorials at the beginning to get familiar with the program. For this we installed Unity, Unity Hub and Visual Studio Code. In this phase we also tested the transfer of the projects with GitHub Desktop, so that everything would work later with our augmented flight project. After this phase, we started to conceptualize a prototype of our idea in Figma.

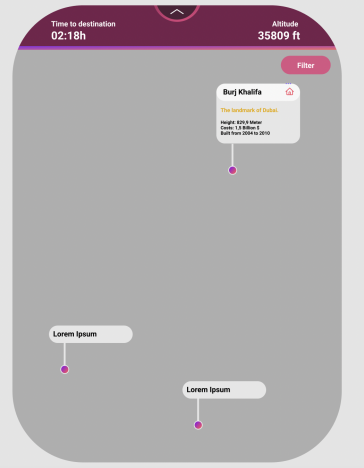
a. First Prototype in Figma

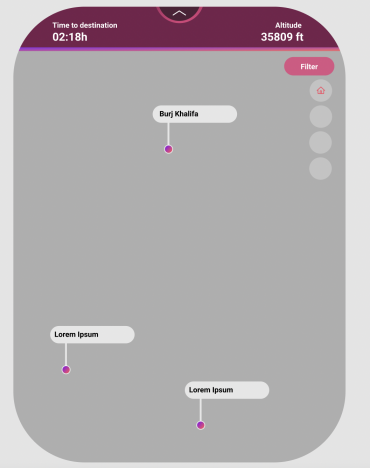
It was especially important to us that you can also hide the application, e.g. for people who just want to enjoy the view. For this purpose, there is a button with an arrow at the top center, which can be used to show or hide the application.



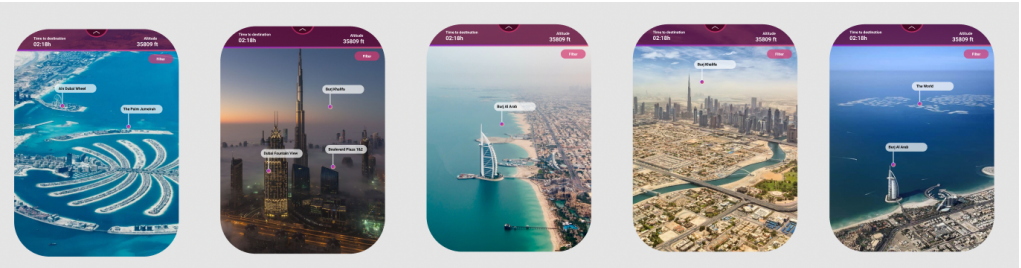
In the faded state the general information "Time to destination" and "Altitude" should always be shown, for this the place around the hide/show button is perfectly suitable.
However, the main indicators of our applications are the pins that show what is underneath the plane. These depend on what height you are at. For example, if you are at a normal flying altitude, cities, rivers, etc. will be shown. But if you are in a landing approach, then detailed information of the destination city is shown, e.g. places of interest, beaches, etc.
Points of interest, which can be seen further in the field of view, are already shown with their names. A point of interest that is at the edge of the field of view is marked with a pin, but without a name.
To provide passengers with even more information, our application is interactive. This means that the airplane window is no longer just a normal window, it is a touchscreen. If a passenger is more interested in a pin, he or she can click on it and a small window with more information will open.

By showing different categories (cities, rivers, sights, etc.), these can also be filtered. The filter can be operated like the pins via touch.





The final product should then look something like this:

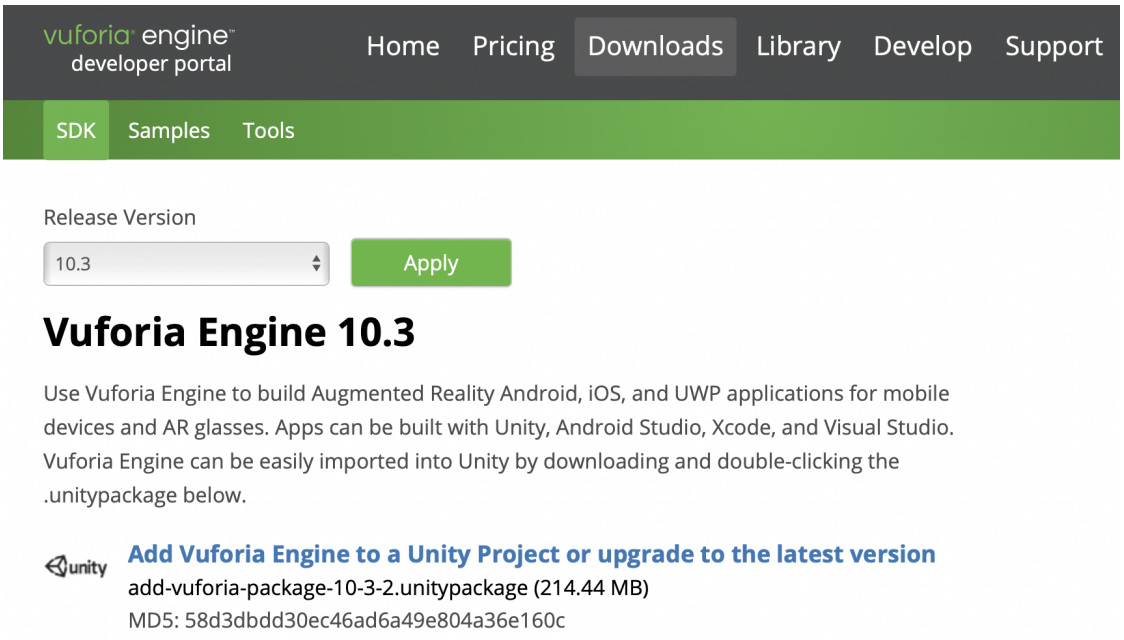


For our project, we decided to use Dubai as an example city and Qatar Airways as an example airline. That's why we picked up Qatar Airways' purple color scheme and branding in our interface.

b. Vuforia / Unity / Xcode

Vuforia

To get started with our augmented flight project, we created a new 3D project in Unity. Since we are creating an iOS application for the tablet, we also use Vuforia. For this we downloaded the Vuforia Engine 10.3 under https://developer.vuforia.com and added it to our Unity file.



Furthermore, we have created a Basic license in the Vuforia Developer Portal under the License Manager. We also created a database under the Target manager and uploaded different images there. Based on the rating, we were able to see very nicely which images are more suitable for tracking. We therefore chose one target at the beginning with four out of five stars. After that, we downloaded the database for the Unity Editor and added it to our Unity file.

Unity

Next, we created an AR camera in Unity (Hierarchy → right click → Vuforia Engine → AR Camera). We then added our license from the Vuforia Developer Portal in the Inspector of the AR Camera under Vuforia Behaviour (script). To do this, copy this license from the Vuforia Developer Portal and paste it into Unity under the "Open Vuforia Engine configuration" button in the App License Key category.

In the next step, we recreated the canvas from our Figma prototype. For this we created a canvas in the Hierarchy (Hierarchy → right click → UI → Canvas). For the canvas we chose the portrait format of an iPad. With a further right click on the canvas we created a panel under which the individual elements of the menu are located. Since the panel is the same size as the canvas, we made it narrower and

moved it to the top of the screen, and also changed the color and transparency to match our prototype. The decorative strip at the bottom of our panel also consists of a UI panel to which we assigned a graphic. For this, we created a line with a gradient in Adobe Photoshop and saved it as a PNG. In Unity, we created an "Images" folder under Assets, into which we then dragged and dropped this PNG. Then we changed the "Texture Type" value in the Inspector of the image to "Sprite (2D and UI)" and applied this setting by clicking on the scene. Now we could assign the PNG to the panel (Inspector → Image → Source Image → Panel_Decor (our PNG)). After that we adjusted the size and the position of the decorative strip.

Then we created the texts "Time to destination", "Altitude" and the respective values for them. We adjusted the color, size and position as in our prototype. In order to ensure that the individual values do not remain static, but also change accordingly, we created two scripts (Time To Destination and Altitude Value) and wrote the appropriate code for them. We then assigned these scripts to the respective text (Add component → Time to destination (script) / Altitude Value (script).

In order to provide the best experience for passengers, we included a filter button. This allows passengers to filter by the five different categories - Activity, Hotel, Shopping, Sights and Buildings. To activate the filter, the filter button must first be pressed and then the categories appear. If you select a category now, only the corresponding pins are shown. For this we have written two scripts - Filter Button Click Event and Filter Pins. The Filter Button Click Event is responsible for what happens when you click on the filter button. It also makes sure that when the filter is closed, the previous selection is cancelled. Lastly, the WhenFilterSelected() method causes the icon of the respective category to color to indicate the selection. The second script, Filter Pins, is responsible for correctly displaying the pins of the chosen category.

In the Inspector of the Filter button, we have added, under On Click (), the three click events described above and the corresponding methods. We also assigned the corresponding objects to the different filter options.
We then added the Filter Pins script to the individual objects and assigned the corresponding pins to the Pin deactivated variables. In addition, the WhenFilterSelected() method from the Filter Button Click Event script is also needed here. Here the assignment of the corresponding sprites for the activated and deactivated status takes place.

However, with filters there is still the possibility to filter not only by one category, but also by several at the same time. Since both have advantages and disadvantages, we included this idea in our study. There we asked the subjects what they like better and 40% of them liked the selection of only one filter better, the other 60% preferred multiple selection. Based on these results, we decided to implement a multiple filter selection.

The first step was to allow multiple categories to be selected at the same time. For this we made a small change in the filter button click event script. The correct display of the matching pins has proven to be more difficult. The last four days we worked on our project, we worked intensively on a solution. We tried different approaches and adapted or changed the code accordingly. Unfortunately, we did not have enough time to find a satisfactory solution.

The last element of our canvas is the Menu Button, which can be used to show and hide the Augmented Flight application. For this we created a button in our canvas and assigned a graphic to it. Then, in the button's Inspector, we changed the button transition to Sprite Swap. We also wrote a script that changes the button's graphic when clicked and fades our panel in and out. We assigned this script to our Menu Button and selected the various methods under On Click () and added the appropriate graphics under Button Sprites.

To illustrate the tracking, we decided to use four images of Dubai. Dubai is a very diverse city and therefore offers the perfect example. The images are already in our Vuforia Database and can now be embedded in Unity via Image Targets. For this we created four Image Targets in the Hierarchy (Hierarchy → right click → Vuforia Engine → Image Target). Then we changed the type in the Inspector to "From Database", selected our database and the corresponding image. For a better overview we moved the images all next to each other.

The following description was made for all four Image Targets:
In the Hierarchy, we created a canvas under the Image Target. We rotated this by 90 degrees on the X axis and dragged it over the image with some distance, we also made it slightly larger than the Image Target. In the Inspector of the canvas we changed the Render Mode from Screen Space - Overlay to World Space and made the AR Camera the Event Camera.
Then we picked out different buildings, landmarks and more on each picture and created the pins for them. A pin consists of a pin circle, a pin line and a pin label. We created a button for each, inserted the corresponding graphic and adjusted the size and position. The graphic for the pin label has a different icon depending on the category and shows the name of the hotel, sight, etc. To provide more information about each pin, we made the pin label interactive. This means that when you click on the Pin Label button, it will pop up and show more information. For this we wrote the script "Pin_OpenClose" with the method whenPinClicked(). This method makes the pin show more or less information. In the inspector of the pin label we have added this script as a component and selected the corresponding method under On Click ().
If you start the game now in the Game view and hold one of the four images in front of it, the pins will be displayed in the right places and move when you change the position of the image.

(All graphics from the canvas and the pins were created in Adobe Photoshop by ourselves.)

The Main_Interaction script was created during the process of trying to solve a problem with the AR camera. However, through several tutorials and research, we solved the problem in a different way. We didn't delete the script for now, though, as it might come in handy again later.

Xcode

To get our Augmented Flight applications onto the iPad afterwards, we first installed Xcode. Using the "File" menu item in Unity, we changed the platform to iOS under Build Settings. Under Player Settings, we then set the app logo and the start screen of the app. In addition, we set the target device to iPad and the target minimum iOS version to 12.0, otherwise we would get an error message in Xcode.

After that, we used the Build and Run button to submit our program to Xcode. In Xcode we could then select the iPad and under Unity-iPhone -- Suitability & Capabilities check the box Automatically manage signing and then select our own team. We then clicked the Play button and the app was added to the iPad.

5. Usability Testing

a. Preparation

For our usability test of the augmented flight application, we prepared an experiment guide and a subject code. Since the data and information we get from our subjects are treated anonymously, subjects can withdraw and delete their data via this code. The experiment guide contains demographic data to get a better understanding about our trial participants and what experience they have so far with augmented reality. It also contains several questions about our concept. Furthermore, we prepared two standardized questionnaires, the SUS (System Usability Scale) and the UEQ (User Experience Questionnaire).

b. Conduction

On January 3rd, 2022, we tested our application with 5 people. At the beginning, the demographic questions were answered and then each participant had five minutes to familiarize themselves with the AR application. During the five minutes, the participants were asked to think aloud so that we could take notes about it. Afterwards, several questions were answered about it. Lastly, after the two questionnaires were completed, the study was over.

c. Results

Experiment Guide

Five participants took part in our study, of which 40% were female and 60% male. The average age was 24.8 years and 80% have already tried an AR application, 20% use an AR app now and then.

The following questions, were all rated using a Likert scale of 1- 5, where 1 means "don't like it" and 5 means "really like it":

- How do you like the information that is shown to you? → Average score: 4.4
- How do you like the filter function? → Average score: 4.8
- If a pin interests you more, you can click on it to get more information. What do you think about this idea? → Average score: 5

Regarding the filter function, we asked another question: Would you rather filter by just one category or multiple categories at once and why?
→ 40% would like to be able to apply only one filter, 60% prefer more.

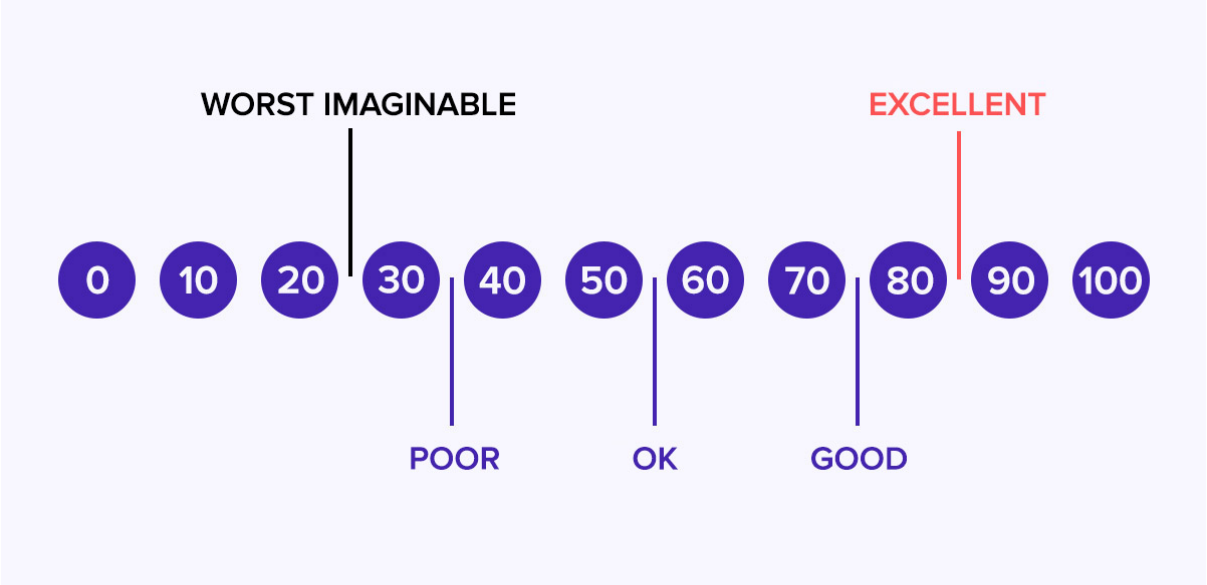
Another question we asked was what other information participants would like to see:

- Pictures/videos of the individual pins
- Ratings
- Booking the attractions and hotels directly through the application

The final question was "Now that you're a little more familiar with the concept of the app, we'd be interested to hear what you think of the idea and if you'd like to use it on a flight." All participants would like to use our application on a flight. The possibility to fold the applications and just enjoy the view was very well received. Two of the participants look at the cities and countries they fly over before each flight and would love to be able to use this concept in real life. In addition, everyone thinks the concept is very nicely laid out and doesn't interfere with the view out of the airplane window, as the pins are discreet, making it a perfect combination of view and information.

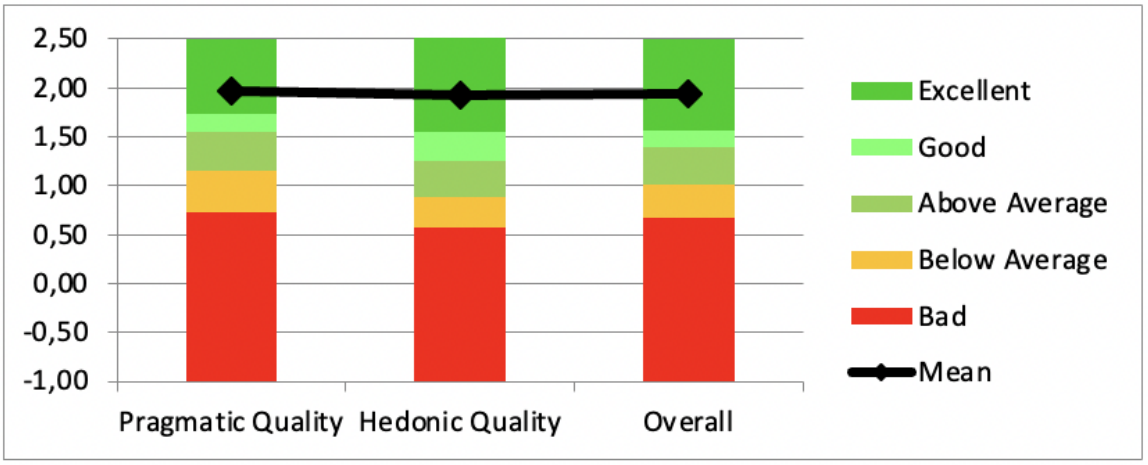
System Usability Scale (SUS)

The overall usability was defined by this standardized Questionnaire. The results can be transferred into a score that corresponds with a certain level of usability. Our application has achieved an SUS Score of 92, which means that the usability of Augmented Flight can be defined as "excellent".



User Experience Questionnaire (UEQ)

The UEQ shows the pragmatic and hedonic quality, as well as the overall user experience. Again, our application is described as "excellent".



6. Summary

Before we started our project, we took several days to think about different ideas. Many interesting thoughts came together, as you can see under the point 1a. Brainstorming. We finally agreed on the Augmented Flight idea, as this is something we would both like to use in the future. Afterwards we defined our first project outlines.

After the first thoughts about our project were completed, we started with the user research. For this, we created two personas to represent the potential users. We also created a user story and a storyboard to match our target group. Furthermore, we defined various user requirements to help us with the rest of the process.

In the following weeks, we worked extensively on a first prototype in Figma and familiarizing ourselves with Unity and Vuforia. After that we got to implement everything in Unity.

Later, after we implemented the most important features, we tested the application to get feedback from different people. The results of the trial were very good and therefore more than we expected. We got very interesting input to make the applications even better in the future. However, we were not able to implement these comments in our project in terms of time, but we have planned to add them after the exam period at the THI and continue the project.

7. Lessons Learned

During this project, we both learned an extreme amount that can be useful for us in the future. Among other things, we learned about GitHub and GitHub Desktop as a great tool for teamwork. Unfortunately, we could only use it at the beginning and at the end of the project, because sharing Unity via GitHub, due to the large amount of data, did not work. Furthermore, it was the first time for both of us to work with Unity, Vuforia and Xcode, so the project was very challenging and time-consuming for us. We watched a lot of different tutorials, did a lot of research and tried a lot of different ways, but through that we learned a huge amount of features in Unity. However, the effort was worth it, as we are more than satisfied with our result. Overall, we also enjoyed the project because we could celebrate many small successes, e.g. when a script we wrote worked.

8. Outlook

Since we see great potential in Augmented Flight, we want to continue to pursue and improve it. Since we both do not have a master thesis topic yet we are thinking about one of us even working on it further.

Unfortunately, there is one point we did not find a solution in time. The filter does not allow multiple selection of different categories yet. This was very frustrating to both of us and is the first point we want to implement after the exams. What we know so far is that there needs to be an extension of the "**Filter Button Click Event"** script. The pins need to be integrated into it, so we can access them with the Filter Buttons. Unfortunately we lack in programming skills here and did not manage to find the right method to integrate this functionality. But, we are very confident that we will be able to solve this issue, as we were also able to teach ourselves everything else that we achieved to this point.

We also want to implement the comments of our usability test participants, as we found them very helpful and will make our application even more pleasurable. Some of our participants suggested adding pictures and, if suitable, videos of the various pins. They would also like to read ratings and reviews from previous guests or visitors. Furthermore, another interesting comment, worth implementing, was a direct booking tool for e.g. attractions and sights.

We are excited about what the future holds for this project.

9. Appendix

a. Time to Destination Script
b. Altitude Value Script
c. Menu Button Event Click Script
d. Filter Button Click Event Script
e. Filter Pins Script
f. Pin Open Close Script
g. Experiment Guide
h. Subjects Code
i. System Usability Scale
j. User Experience Questionnaire

a. Time to Destination Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class TimeToDestination : MonoBehaviour
{

    public float timeValue = 300;
    public Text timeText;

    void Update()
    {

        if(timeValue > 0)
        {
            timeValue -= Time.deltaTime;
        }
        else
        {
            timeValue = 0;
        }

        DisplayTime(timeValue);

    }

    void DisplayTime (float timeToDisplay)
    {
        if (timeToDisplay < 0)
        {
            timeToDisplay = 0;
        }

        float minutes = Mathf.FloorToInt(timeToDisplay / 60);
        float seconds = Mathf.FloorToInt(timeToDisplay % 60);

        timeText.text = string.Format("{0:00}:{1:00} h", minutes, seconds);

    }
}

```

b. Altitude Value Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class AltitudeValue : MonoBehaviour
{

    public GameObject changingValue;
    public int feetLeft = 808;
    public bool takeAway = false;

    // Start is called before the first frame update
    void Start()
    {
        changingValue.GetComponent<Text>().text = "35" + feetLeft + "ft";
    }

    // Update is called once per frame
    void Update()
    {
        if (takeAway == false && feetLeft > 500)
        {
            StartCoroutine(ValueTake());
        }
    }

    IEnumerator ValueTake()
    {
        takeAway = true;
        yield return new WaitForSeconds (3);
        feetLeft -= 3;
        if (feetLeft <99)
        {
            changingValue.GetComponent<Text>().text = "350" + feetLeft + "ft";
        }

        else
        {
            changingValue.GetComponent<Text>().text = "35" + feetLeft + "ft";
        }

        takeAway = false;
    }
}
```

c. Menu Button Event Click Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Menu_Button_ClickEvent : MonoBehaviour
{
    public GameObject Panel;
    [SerializeField] private Sprite[] buttonSprites;
    [SerializeField] private Image targetButton;


    public void TaskOnClick(){

        if (Panel !=null)
        {
            bool isActive = Panel.activeSelf;

            Panel.SetActive (!isActive);
        }
    }

    public void ChangeSprite() {

      if (targetButton.sprite == buttonSprites[0])
      {
        targetButton.sprite = buttonSprites[1];
        return;
      }

      targetButton.sprite = buttonSprites[0];

    }
}
```

d. Filter Button Click Event Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Filter_Button_ClickEvent : MonoBehaviour
{

  public GameObject Filter_Option_1;
  public GameObject Filter_Option_2;
  public GameObject Filter_Option_3;
  public GameObject Filter_Option_4;
  public GameObject Filter_Option_5;
  public GameObject Filter_Option_1_pressed;
  public GameObject Filter_Option_2_pressed;
  public GameObject Filter_Option_3_pressed;
  public GameObject Filter_Option_4_pressed;
  public GameObject Filter_Option_5_pressed;

  public bool Filter_Option_1_active = false;
  public bool Filter_Option_2_active = false;
  public bool Filter_Option_3_active = false;
  public bool Filter_Option_4_active = false;
  public bool Filter_Option_5_active = false;

  public bool IsMenueOpen = false;

  public GameObject FilterSelectedDeactivated;
  public GameObject FilterSelectedActivated;

// Filter werden angezeigt wenn Filter Button geklickt wurde
    public void TaskOnClick()
    {
        Debug.Log ("click");
        IsMenueOpen = !IsMenueOpen;
```

```
35
36          if (Filter_Option_1 !=null)
37          {
38              bool isActive = Filter_Option_1.activeSelf && Filter_Option_1_active;
39
40              Filter_Option_1.SetActive (!isActive);
41
42              if (!IsMenueOpen)
43              {
44                  Filter_Option_1.SetActive(false);
45              }
46          }
47
48          if (Filter_Option_2 !=null)
49          {
50              bool isActive = Filter_Option_2.activeSelf && Filter_Option_2_active;
51
52              Filter_Option_2.SetActive (!isActive);
53
54              if (!IsMenueOpen)
55              {
56                  Filter_Option_2.SetActive(false);
57              }
58          }
59
60          if (Filter_Option_3 !=null)
61          {
62              bool isActive = Filter_Option_3.activeSelf && Filter_Option_3_active;
63
64              Filter_Option_3.SetActive (!isActive);
65
66              if (!IsMenueOpen)
67              {
68                  Filter_Option_3.SetActive(false);
69              }
70          }
```

```
71
72          if (Filter_Option_4 !=null)
73          {
74              bool isActive = Filter_Option_4.activeSelf && Filter_Option_4_active;
75
76              Filter_Option_4.SetActive (!isActive);
77
78              if (!IsMenueOpen)
79              {
80                  Filter_Option_4.SetActive(false);
81              }
82          }
83
84          if (Filter_Option_5 !=null)
85          {
86              bool isActive = Filter_Option_5.activeSelf && Filter_Option_5_active;
87
88              Filter_Option_5.SetActive (!isActive);
89
90              if (!IsMenueOpen)
91              {
92                  Filter_Option_5.SetActive(false);
93              }
94          }
95      }
96
97  // wenn filter pressed aktiv ist, dann soll er deaktiviert werden
98      public void DeactivatePressedFilters ()
99      {
100
101         if (Filter_Option_1_pressed.activeInHierarchy == true)
102         {
103             Filter_Option_1_pressed.SetActive(false);
104         }
105
```

```
106         if (Filter_Option_2_pressed.activeInHierarchy == true)
107         {
108             Filter_Option_2_pressed.SetActive(false);
109         }
110 
111         if (Filter_Option_3_pressed.activeInHierarchy == true)
112         {
113             Filter_Option_3_pressed.SetActive(false);
114         }
115 
116         if (Filter_Option_4_pressed.activeInHierarchy == true)
117         {
118             Filter_Option_4_pressed.SetActive(false);
119         }
120 
121         if (Filter_Option_5_pressed.activeInHierarchy == true)
122         {
123             Filter_Option_5_pressed.SetActive(false);
124         }
125     }
126 
127 // Filter selected, Filter mit anderem Sprite wird angezeigt
128     public void WhenFilterSelected()
129     {
130         if (FilterSelectedDeactivated.activeInHierarchy == true)
131         {
132             FilterSelectedDeactivated.SetActive(false);
133             FilterSelectedActivated.SetActive(true);
134         }
135         else
136         {
137             FilterSelectedDeactivated.SetActive(true);
138             FilterSelectedActivated.SetActive(false);
139         }
140 
```

```
        if (FilterSelectedActivated == Filter_Option_1)
        {
            Filter_Option_1_active = !Filter_Option_1_active;
        }

        if (FilterSelectedActivated == Filter_Option_2)
        {
            Filter_Option_2_active = !Filter_Option_2_active;
        }

        if (FilterSelectedActivated == Filter_Option_3)
        {
            Filter_Option_3_active = !Filter_Option_3_active;
        }

        if (FilterSelectedActivated == Filter_Option_4)
        {
            Filter_Option_4_active = !Filter_Option_4_active;
        }

        if (FilterSelectedActivated == Filter_Option_5)
        {
            Filter_Option_5_active = !Filter_Option_5_active;
        }

        if (FilterSelectedDeactivated == Filter_Option_1_pressed)
        {
            Filter_Option_1_active = !Filter_Option_1_active;
        }

        if (FilterSelectedDeactivated == Filter_Option_2_pressed)
        {
            Filter_Option_2_active = !Filter_Option_2_active;
        }
```

```

         if (FilterSelectedDeactivated == Filter_Option_3_pressed)
         {
             Filter_Option_3_active = !Filter_Option_3_active;
         }

         if (FilterSelectedDeactivated == Filter_Option_4_pressed)
         {
             Filter_Option_4_active = !Filter_Option_4_active;
         }

             if (FilterSelectedDeactivated == Filter_Option_5_pressed)
         {
             Filter_Option_5_active = !Filter_Option_5_active;
         }
    }
}
```

e. Filter Pins Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Filter_Pins : MonoBehaviour
{
    public GameObject Pin_deactivated_1;
    public GameObject Pin_deactivated_2;
    public GameObject Pin_deactivated_3;
    public GameObject Pin_deactivated_4;
    public GameObject Pin_deactivated_5;
    public GameObject Pin_deactivated_6;
    public GameObject Pin_deactivated_7;
    public GameObject Pin_deactivated_8;
    public GameObject Pin_deactivated_9;
    public GameObject Pin_deactivated_10;
    public GameObject Pin_deactivated_11;
    public GameObject Pin_deactivated_12;
    public GameObject Pin_deactivated_13;
    public GameObject Pin_deactivated_14;


    public void TaskOnClick()
    {

        if (Pin_deactivated_1 !=null)
        {
            bool isActive = Pin_deactivated_1.activeSelf;

            Pin_deactivated_1.SetActive (!isActive);
        }

        if (Pin_deactivated_2 !=null)
        {
            bool isActive = Pin_deactivated_2.activeSelf;
```

```
36
37              Pin_deactivated_2.SetActive (!isActive);
38          }
39
40          if (Pin_deactivated_3 !=null)
41          {
42              bool isActive = Pin_deactivated_3.activeSelf;
43
44              Pin_deactivated_3.SetActive (!isActive);
45          }
46
47          if (Pin_deactivated_4 !=null)
48          {
49              bool isActive = Pin_deactivated_4.activeSelf;
50
51              Pin_deactivated_4.SetActive (!isActive);
52          }
53
54          if (Pin_deactivated_5 !=null)
55          {
56              bool isActive = Pin_deactivated_5.activeSelf;
57
58              Pin_deactivated_5.SetActive (!isActive);
59          }
60
61          if (Pin_deactivated_6 !=null)
62          {
63              bool isActive = Pin_deactivated_6.activeSelf;
64
65              Pin_deactivated_6.SetActive (!isActive);
66          }
67
68           if (Pin_deactivated_7 !=null)
69          {
70              bool isActive = Pin_deactivated_7.activeSelf;
```

```
71
72              Pin_deactivated_7.SetActive (!isActive);
73          }
74
75          if (Pin_deactivated_8 !=null)
76          {
77              bool isActive = Pin_deactivated_8.activeSelf;
78
79              Pin_deactivated_8.SetActive (!isActive);
80          }
81
82          if (Pin_deactivated_9 !=null)
83          {
84              bool isActive = Pin_deactivated_9.activeSelf;
85
86              Pin_deactivated_9.SetActive (!isActive);
87          }
88
89          if (Pin_deactivated_10 !=null)
90          {
91              bool isActive = Pin_deactivated_10.activeSelf;
92
93              Pin_deactivated_10.SetActive (!isActive);
94          }
95
96          if (Pin_deactivated_11 !=null)
97          {
98              bool isActive = Pin_deactivated_11.activeSelf;
99
100             Pin_deactivated_11.SetActive (!isActive);
101         }
102
103         if (Pin_deactivated_12 !=null)
104         {
105             bool isActive = Pin_deactivated_12.activeSelf;
```

```
106
107             Pin_deactivated_12.SetActive (!isActive);
108         }
109
110         if (Pin_deactivated_13 !=null)
111         {
112             bool isActive = Pin_deactivated_13.activeSelf;
113
114             Pin_deactivated_13.SetActive (!isActive);
115         }
116
117         if (Pin_deactivated_14 !=null)
118         {
119             bool isActive = Pin_deactivated_14.activeSelf;
120
121             Pin_deactivated_14.SetActive (!isActive);
122         }
123     }
124 }
```

f. Pin Open Close Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Pin_OpenClose : MonoBehaviour
{

    public GameObject Pin_Unfold;

    public void whenPinClicked()
    {
        if (Pin_Unfold.activeInHierarchy == true)
        {
            Pin_Unfold.SetActive(false);
        }

        else
        {
            Pin_Unfold.SetActive(true);
        }
    }
}

```

g. Experiment Guide

Usability Test – Augmented Flight AVR

Participant number:

Welcome and introduction

Thank you for participating in our usability study for our Augmented Reality Application "Augmented Flight". This study is a part of the course "Augmented and Virtual Reality Applications" in the User Experience Design Master at the University of Technology Ingolstadt.

In addition to socio-demographic data such as age and gender, we also collect subjective statements that you share with us about the concept. These will of course be evaluated anonymously and will not be passed on to third parties.

The study will take about 15 minutes.

We will ask you several questions and ask you to always think aloud so that we can take notes during the process. Afterwards, you will fill out two questionnaires. These will help us evaluate the user experience and usability of the app. Important is that we test the system and not you, which means you can't do anything wrong. At the end of our interviews, we have two questionnaires for you to fill out.

If you don't understand a question or anything is unclear, please let us know.

If at any point you want to stop the study, just let us know. You can stop it at any time without giving any reason.

Now we would like to ask you to fill in the subject code. If you want to delete your data afterwards, please send us an e-mail with the code. → Test person code

First, a few questions about you

1. May I ask how old you are?

2. Gender:

female	male	diverse

3. Do you have any experience with AR Applications?

don't know	no experience	already tried	use now and then	use regularly

Interview

Before we start, we ask you to get familiar with the app for five minutes.

Please remember to think aloud.

How do you like the information that is shown to you?

don't like it				really like it
1	2	3	4	5

How do you like the filter function?

don't like it				really like it
1	2	3	4	5

Would you rather filter by just one category or multiple categories at once and why?

1 Categories	>1 Categories

If a pin interests you more, you can click on it to get more information.

What do you think about this idea?

don't like it				really like it
1	2	3	4	5

Are there any information you would also like to see?

Now that you're a little more familiar with the concept of the app, we'd be interested to hear what you think of the idea and if you'd like to use it on a flight.

The study is now finished! Thank you for your participation! → after SUS and UEQ

If you have any suggestions, please feel free to express them now:

h. Subjects Code

Subjects Code for Subjects:

Please enter your code word in the boxes

The last two letters of your mother's maiden name: ☐☐

The number of letters of your mother's name: ☐☐

The last two letters of your father's name: ☐☐

Your own birthday (day only): ☐☐

Example: ER-09-ED-17

To delete your data, please send this code word to: laf3489@thi.de or nak7705@thi.de

Subjects Code for Interviewer

Participant number: ☐

Please enter your code word in the boxes

The last two letters of your mother's maiden name: ☐☐

The number of letters of your mother's name: ☐☐

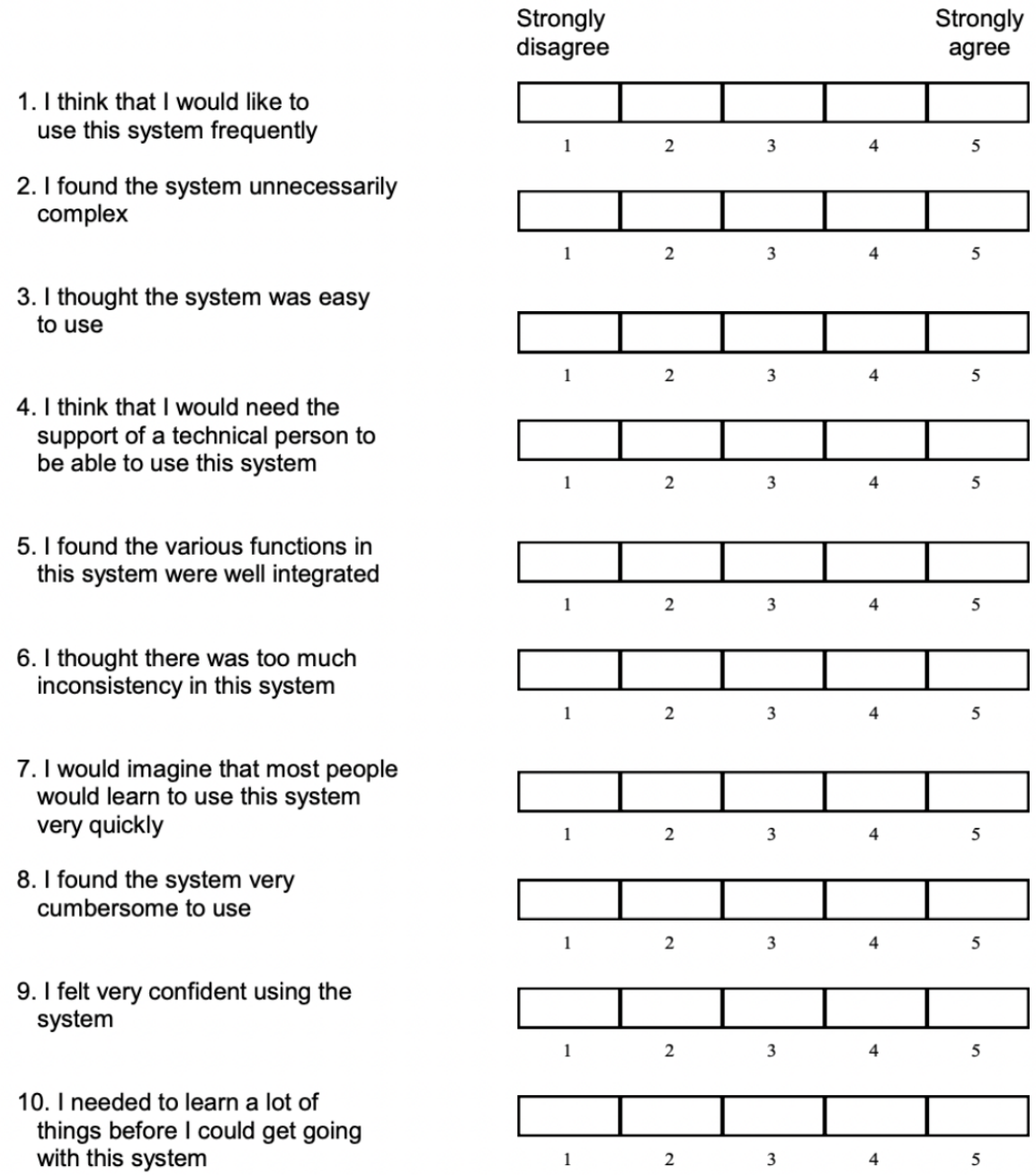
The last two letters of your father's name: ☐☐

Your own birthday (day only): ☐☐

Example: ER-09-ED-17

i. System Usability Scale

System Usability Scale



j. User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26