Mobile application for AR navigation

Author: Repina Anastasia, MSE2018

Moscow, 2019

Abstract

Every day people navigate from one place to another in order to achieve their tasks and goals. One of the most common types of navigation is on foot. Walking through the streets people do not have to pay for tickets and stack into the traffic jams. According to **the …….,** people spend half of their life on going somewhere. So, inclusion of an educational or the cultural part to this way can broaden the people’s horizons and help not to get bored while walking. Although, it is the 21 century already, most of the citizens prefer to use paper maps or their online analogs in order to find the exact way. However, this case involves a lot of problems: from dependency on the stable internet connection up to the difficulties connected with understanding is it the right place to turn left of right. The existing applications on the market are devoted mainly to the usual maps and do not pay much attention to the improvement of the navigational process, which can positively influence of the user’s experience. So, that is why trying to make the everyday navigation easier seems a good idea. The only questions is how to do it?

The main purpose of the work is to create a mobile application which helps the user to faster and easier navigate to the destination point as well as stay informed of the nearby excursion places. The augmented reality is one of the options of how this goal can be achieved. Having an augmented reality projection together with a usual map can help to solve the mentioned difficulties, so the user’s experience will become even better. Monitoring the geolocation in a real time using the in-device modules and creation an augmented reality projection together with changes on the common map do not require the expenditure of personal time. In case of any problems the application will signal the user. It stores information about the excursion objects: theaters and museums, so while navigating to the destination location it is possible to learn more about the city where you are now. Furthermore, an opportunity to load the saved and frequently used routes can save time used for searching. The application is implemented using technologies: ARKit, Google Firebase, MKMap and others.

The proposed project is of great importance since it contributes to the improvement of contemporary complex and far from ideal approaches to human navigation. The application is available for download for more all the iOS devices which have the AR module inside on GitHub via the link **…….**

***Index Terms*** *— mobile application, iOS, geolocation, navigation, augmented reality, map.*

Definitions and acronyms

GitHub – the largest web service for hosting IT-projects and their joint development

Firebase – is a mobile and web application development platform developed by Google

Augmented reality – the result of blending of interactive digital elements together with the real world

MKMap – framework which if used for setting a map to the application

ARKit – a set of tools for the iOS platform which are used the augmented reality creation

iOS – mobile operating platform

Geolocation – the position coordinates of the object on the Earth

Swift – programming language which is used for the iOS applications development

Splash-screen – an introduction screen inside the application

Walkthrough-screen – a several page screen used for helping with understanding of curtain tasks

ZenHub – independent project management tool natively integrated with GitHub

Contents

[Abstract 2](#_Toc4666665)

[Definitions and acronyms 3](#_Toc4666666)

[1. Introduction 6](#_Toc4666667)

[2. Existing solutions 8](#_Toc4666668)

[Conclusion 8](#_Toc4666669)

[3. Information model and algorithms 9](#_Toc4666670)

[3.1. Information model 9](#_Toc4666671)

[3.2. Navigation algorithm 9](#_Toc4666672)

[Conclusion 9](#_Toc4666673)

[4. Requirements 10](#_Toc4666674)

[4.1. Functional requirements 10](#_Toc4666675)

[4.2. Reliability and security requirements 10](#_Toc4666676)

[4.3. Technical and software requirements 10](#_Toc4666677)

[Conclusion 10](#_Toc4666678)

[5. Implementation details 11](#_Toc4666679)

[6. Architecture and design 12](#_Toc4666680)

[6.1. MVC Description 12](#_Toc4666681)

[6.2. Design and interface 12](#_Toc4666682)

[Conclusion 12](#_Toc4666683)

[7. Technology stack 13](#_Toc4666684)

[7.1. Development tools 13](#_Toc4666685)

[7.2. Main libraries 13](#_Toc4666686)

[7.3. Main third party services 13](#_Toc4666687)

[7.3.1. ARKit 13](#_Toc4666688)

[7.3.2. Firebase 13](#_Toc4666689)

[7.3.3. GitHub 13](#_Toc4666690)

[7.3.4. MKMap 13](#_Toc4666691)

[Conclusion 13](#_Toc4666692)

[8. Usability evaluation 14](#_Toc4666693)

[9. Conclusion 15](#_Toc4666694)

[10. References 16](#_Toc4666695)

[Appendix A. Use case diagram 18](#_Toc4666696)

[Appendix B. Database scheme 19](#_Toc4666697)

1. Introduction

Over the past few years, the mobile applications market has seen a significant increase in the number of new programs that solve various tasks [1]. Now the most familiar things can be done online using a phone or tablet: ordering the food, playing games or just talking with friends and relatives, without wasting time moving from one place to another. This success can be explained by the emergence of new special tools and devices that can be associated with the mobile phones, as well as with the emergence of new features within the gadgets themselves.

**…..**

**The goal of this work is** thecreation of the mobile IOS application for outdoor AR navigation.

The created application will be used for the route-finding purposes. Using this application user will be able to navigate to the destination point from the current location using both usual map and the AR mode. There are no specific restrictions of whom can use the application. However, the person should not have any visual or motor disabilities which can spoil the user experience. Also, it is necessary to have an iOS platform device with an AR module inside.

The developing application will be used

**In order to achieve the goal several tasks should be done:**

1. Study of the subject area, analysis of existing solutions;
2. Development of application functionality;
3. Selection of technical means of implementation and platform;
4. Development of models and algorithms for data transmission and processing;
5. Study of technical means of implementation and platform;
6. Choose of modules and application architectures;
7. Development of application design;
8. Application development and testing;
9. Development of technical documentation.

In the further chapters, the existing analogues of the AR navigation applications are analyzed, the platform is selected, functional requirements are developed (Chapter 1), the information model and data transmission, as well as the AR navigation algorithms are established (Chapter 2), the chosen libraries and the technology stack, as well as the details of the software implementation are mentioned (Chapter 3). In conclusion, the main results of the work are presented and the ways of its continuation are considered. The Appendices to the work contain the data base schema and the use case diagram.

1. Existing solutions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Price** | **AR mode** | **Stability** | **Favorite places** | **Search** |
| Yandex maps | free | + | + | + | + |
| Google Maps | free | - | + | +/- | + |
| Apple Maps | free | - | + | + | + |
| Welc map | free | +/- | - | - | - |
| Eye maps | $4.99 | + | + | - | + |
| Wikitude | free | +/- | + | + | +/- |
| AR Maps | free | +/- | - | + | + |
| Mosgorpass | free | + | +/- | + | + |
| ARCity | free | + | +/- | - | + |
| My application | free | + | + | + | + |

Conclusion

1. Information model and algorithms
   1. Information model
   2. Navigation algorithm

Conclusion

1. Requirements
   1. Functional requirements
   2. Reliability and security requirements
   3. Technical and software requirements

Conclusion

1. Implementation details

The main application algorithm is divided into several steps. Each of them is responsible for their own mathematical calculations.

During the development process several technologies were used. First of all, it is the ARKit itself. None of the augmented reality applications for the iOS can be done without this framework. For the navigational purpose CoreLocation framework was chosen as the most frequently used one.

The article will be divided into two parts. The first part will explain the basics of ARKit, whereas the second one will concentrate on the navigation algorithm itself.

Together with the iOS 11 release the Apple company has unleashed the power of ARKit onto the iOS development community.

Before going into details it is necessary to mention that there are two types if steps: main and intermediary. Main steps are those in which the pedestrian will change the direction, for example, turn left or right. Each pair of the main steps is connected with a set of intermediary steps, so it is possible to easily navigate from one main point to another. The calculation of the main points is given to the integral maps API, created by the iOS map application developers. In order to find these steps the developer needs to pass the start and finish coordinates as well as the type of transport which is supposed to be used. In our case, this type is named ‘walking’. Although, it is possible to see the main routes for a ‘car’ type route.

//todo add code here

After that steps is finished we move to the calculation of the intermediary points. For that purpose there is a special algorithm.

//todo add code here

After all the preparation steps are done it is necessary to pass to the method the geological coordinates of the start and finish points, and get the AR nodes for each of the route steps.

//todo add code here

So, now we have a set of the AR nodes. However, it is also important to mention, how the AR node is created itself. For the visualization purposes the bean form was chosen for each of the nodes.

//todo give an example of AR node class

1. Architecture and design
   1. MVC Description
   2. Design and interface

Conclusion

1. Technology stack
   1. Development tools
   2. Main libraries
   3. Main third party services
      1. ARKit
      2. Firebase
      3. GitHub
      4. MKMap

Conclusion

1. Usability evaluation
2. Conclusion
3. References
4. Mata Rivera Miguel, Claramunt Christophe, Juarez Alberto. An experimental virtual museum based on augmented reality and navigation, 497-500, 10.1145/2093973.2094058, 2011;
5. Narzt Wolfgang, Pomberger Gustav, Ferscha Alois, Kolb Dieter, Maјller Reiner, Wieghardt Jan, Hartner Horst, Lindinger Christopher. Augmented reality navigation systems. Universal Access in the Information Society, 4, 177-187, 10.1007/s10209-005-0017-5, 2006;
6. Reitmayr Gerhard, Schmalstieg Dieter. Collaborative augmented reality for outdoor navigation and information browsing, Proceedings of the Symposium on Location Based Services and TeleCartography, 2004;
7. Raskar Ramesh, Low Kok-Lim. Interacting with Spatially Augmented Reality, ACM International Conference on Computer Graphics, Virtual Reality and Visualisation in Africa, 10.1145/513867.513889, 2001;
8. Alessandro Mulloni , Hartmut Seichter , Dieter Schmalstieg. User experiences with augmented reality aided navigation on phones, Mixed and Augmented Reality (ISMAR), 2011;
9. Kyle Roche. Pro iOS 5 Augmented Reality, 345, APRESS, 2011;
10. Randall Shumaker Stephanie Lackey (Eds.). Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality, 6th International Conference, VAMR 2014 Held as Part of HCI International 2014 Heraklion, Crete, Greece, June 22–27, 2014, Proceedings, Part II;
11. Swift developer blog, “Pass Information Back to the Previous View Controller” [Electronic resource]. URL: https://swiftdeveloperblog.com/pass-information-back-to-the-previous-view-controller/ (request date: 03.01.2019);
12. GitHub, Side Menu for IOS library site [Electronic resource]. URL: https://github.com/jonkykong/SideMenu (request date: 05.01.2019);
13. Apple Inc., ARKit [Electronic resource]. URL: https://developer.apple.com/arkit/ (request date: 05.01.2019);
14. Figma, Online UI design tool [Electronic resource]. URL: https://www.figma.com/ (request date: 04.12.2018);
15. LongList, IOS Popup View Using Separate View Controller [Electronic resource]. URL: http://longlist.org/ios+popup+view+using+separate+view+controller++xcode+8++swift+3 (request date: 05.01.2019);
16. Сайт Ray Wenderlich, Augmented Reality and ARKit Tutorial [Electronic resource]. URL: https://www.raywenderlich.com/378-augmented-reality-and-arkit-tutorial (request date: 09.02.2019);
17. Collective Idea, ARKit Wall and Plane Detection for iOS 11.3[Electronic resource]. URL: https://collectiveidea.com/blog/archives/2018/04/30/part-1-arkit-wall-and-plane-detection-for-ios-11.3 (request date: 05.01.2019);
18. Medium, ARKit and CoreLocation [Electronic resource]. URL: https://medium.com/journey-of-one-thousand-apps/arkit-and-corelocation-part-one-fc7cb2fa0150 (request date: 20.01.2019);
19. YouTube, GPS search [Electronic resource]. URL: https://www.youtube.com/watch?v=8-TDf\_7j59Y (request date: 05.01.2019);
20. GitHub, Swift Spinner for IOS library site [Electronic resource]. URL: https://github.com/icanzilb/SwiftSpinner (request date: 07.02.2019);
21. GitHub, Swift Auto Diagram for IOS tool site [Electronic resource]. URL: https://github.com/yoshimkd/swift-auto-diagram (request date: 17.01.2019);
22. Rochester Institute of Technology, Introduction to Augmented Reality [Electronic resource]. URL: http://www.se.rit.edu/~jrv/research/ar/introduction.html (request date: 23.01.2019);
23. Academia.edu, Scientific platform [Electronic resource]. URL: https://www.academia.edu/ (request date: 12.01.2019);
24. ZenHub extension for GitHub[Electronic resource]. URL: https://www.zenhub.com/ (request date: 21.11.2018);
25. GitHub Inc., Hosting service for version control using Git [Electronic resource]. URL: https://github.com/ (request date: 21.11.2018);
26. YouTube, Walkthrough screens creation [Electronic resource]. URL: https://www.youtube.com/watch?time\_continue=73&v=1F\_HVMHocdA (request date: 21.11.2018);
27. Movable Type, Calculate distance, bearing and more between Latitude/Longitude points [Electronic resource]. URL: http://www.movable-type.co.uk/scripts/latlong.html (request date: 10.12.2018);
28. SwiftBook, Marshrutisation with MapKit and Core Location [Electronic resource]. URL: https://swiftbook.ru/post/tutorials/marshrutizaciya-s-mapkit-i-core-location/ (request date: 07.12.2018);
29. HackerNoon, Building a location app with ARKit, CoreLocation and Pusher [Electronic resource]. URL: https://hackernoon.com/building-a-location-app-with-arkit-corelocation-and-pusher-bee44fdec44f (request date: 05.02.2019);
30. Russia Open Data [Electronic resource]. URL: https://data.gov.ru/ (request date: 25.01.2019).

Appendix A. Use case diagram

Appendix B. Database scheme