Research on the development of a local area map of NRNU MEPhI University with a "Useful labels" and Augmented Reality navigation.

Daniil Shenyagin
Department of Cybernetics
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Moscow, Russian Federation
mr.shenyagin.1@gmail.com

Maria-Anastasia Kurochkina
Department of Computer Systems and Technologies
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Moscow, Russian Federation
kiwa190@mail.ru

***Abstract*—This article devotes to research on the development of a multi-user, dynamic, mobile application created in the Unity development environment on the C# language. The application allows to find the shortest and optimal routes on the local map of NRNU MEPhI university (hereinafter the NRNU MEPhI); create, visualize and attach "useful notes" the corpuses and audiences; help students and teachers easy way to find free classrooms. It will outline the main stages of this labor, and it will be thorough disassembly details and subtleties of each development step.**

The research is based on the combination of various styles and programming languages: C#, Python, JS; development environments are: Unity, Blender, VisualStudio; and API: GoogleCloudVisoin, GoogleMaps, YandexWeather. The main difficulty of course became modeling the interaction of the application and the implementation of all system's module.

Keywords—3D-modeling, Theory of Graphs, Navigation, client–server architecture(REST), API, .NET C# - Android App developing, networks.

I. Introduction

Before beginning to the essence of the research, it is necessary to describe its relevance and social significance. There is no secret that for a new flow of first-year students at first it is difficult to get used not only to the educational process, but also with the variety of buildings, classrooms and paths connecting them. Thus, an idea for research was born: it is necessary to create a resource that would help new students quickly get used to new places for them. But in order not to lose most of the potential students, staff and teachers of our institute, an addition was proposed: to implement the creation and visualization of "useful notes" which will help to quickly receive and transmit information about all objects of the NRNU MEPhI. Local map introduce augmented reality technology (next in the text – AR) into the project. Reaching as many students and staff of NRNU MEPhI as possible, it was necessary to provide the best environment or even practically a social network. Consequently, the relevance of this research is quite significant within the framework of modern possibilities. Social meaning, as mentioned above, covers all students, researchers and teachers, who are the overwhelming majority.

Essence of research

Research consists in the implementation of a 3d model based on manually collected data about corpuses and classrooms, the creation of a multi-user, dynamic mobile application based on this data, supported on the Android platform and the implementation of navigation functions on constructed maps. In the course of the study, it turned out that is quite new and has great potential for development. We have studied a large amount of resources on related topics, it became known that nobody has yet been involved in combining them — this is the uniqueness and novelty of the study, although separately such technologies are not new in themselves. For example, such cross-platform applications as Google Maps, Yandex Maps and 2gis were taken as the basis for the implementation of map models. From the received data from the satellite, a terrain map model is constructed, as well as a graph for further use of the coordination and navigation function [1]. In general, which will be done, but, unfortunately, without access to the nearest supersensitive satellite, data collection was carried out manually. Later this method turned out it was quite accurate.

All the above boils down to setting the main goal of this study is how to create a centralized synchronous application with a variety of useful functions.

II. Research Materials And Methods

To achieve the described goal, it was decided to divide the entire study into 6 main stages:

1) *Direct data collection and its methods*

2) *Implementation of a 3d model based on the collected data*

a) *Prototyping the sketches in a 3D computer graphics program — Blender*

b) *Importing the resulting model into the Unity development environment, using the solution to the problem related to rebuilding the coordinate axes*

3) *Creation of a centralized system for management and synchronous data transmission as a server*

4) *Development of an android application with all available coordination and navigation functions*

 978-0-7381-4275-3/21/$31.00 ©2021 IEEE

5) Implementation in an AR application with the definition of key objects

6) Testing

A. Beginning of study

In order to begin the description in details of each stage of the study, we should explain how it will look in general. The plan of interaction of application modules is as shown on the Fig.1:

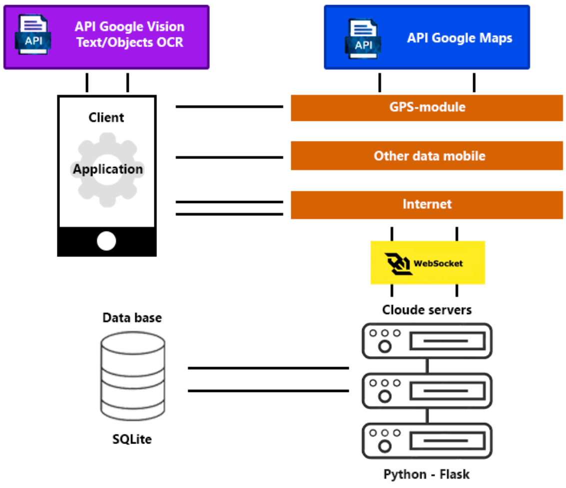

Fig. 1. Circuit of interaction of application components

It was decided to implement stage (6) throughout the entire study, because of two reasons. Firstly, the terms of the study are compressed. Secondly, there are a lot of details and work in general for one developer, therefore, after the implementation of each stage, it must be tested in parallel.

B. Data collection

We should start with the most paramount and important thing - the collection of data on all objects that are located on the map of the local area of NRNU MEPhI. As already emphasized, the main objects on this map are corpuses, paths connecting corpuses with others, and classrooms with corridors.

The main characteristics for enclosures are:

1) Length and width in the mathematical plane of the Earth

To start searching for data, its needs to make a request to the administration about the engineering drawings of the buildings for the construction of all buildings of NRNU MEPhI. But having received a refusal to provide that drawings, it was necessary to find a way out of this situationю. It was decided to collect data manually.

The length and width data are much easier to calculate relatively than the other component — the height: in order to find the exact dimensions of the bodies, we used the known software – Yandex Maps (tool – "Ruler", the scale is maximum for the greatest accuracy). By this way, the table of results for building 3d maps was compiled.

2) Height (measured from the base of foundation to building roof)

Since all familiar applications and solutions don't have height values, it had to get using some non-standard methods. The following methods have been invented, implemented and mostly discarded (some of them):

a) Astronomical method — use the height of the Sun above the horizon, its declination (δ) at a given time, latitude (φ) and shadow length (L):

$$H = L * \tan\left(\frac{\pi}{2} - \varphi + \delta\right) \quad (1)$$

The length of the shadow can be calculated with using hand-held measuring devices, but when we switch the map display mode "from satellite", we can clearly see shadow traces. Declination was calculated based on the date of images taken by Yandex. This method didn't show the maximum accuracy compared to the following.

b) Method of photographing and determining the height at the stage of processing the photograph relative to the object, the height of which is known in advance. Any object can serve as such, but the smaller it is, the greater detection accuracy. The choice fell on a primitive object – brick, the width of which is h = 15 (cm). The formula for overall height is intuitive:

$$H = (n \pm 2) * h \quad (2)$$

Amount of data isn't limited here, since navigation will be used inside buildings as well. It was more convenient here to use technical materials, namely fire cards, which are located on safety measures on each floor. After receiving all the maps, it was implemented 2D maps arranged parallel to the main mathematical plane, so when the user switches visibility mode, it is convenient to change the tilt of the camera angle and its position relative to the map itself.

The result table contains two data points:

1) Coordinates center of the building and its alignment relative to other buildings

2) The relative coordinates of its inputs-outputs — the same empirical method

Next, it would be sketched out the approximate coordinates of all other secondary objects and their approximate size. We will do all this just to try for exactly complement model to the real level.

C. Implementation of the 3d model based on the collected data

Immediately after collecting the necessary data, we were in need to try to implement the corresponding 3d model, exactly similar to the real MEPhI map. It was decided to implement all this in a special graphical environment — Blender. But in the process of using Blender, there was found a problem of importing the resulting model into Unity. Solution of this problem has been described in the work [2].

Following the table of measurement results, create inline 3d objects and apply the Position and Scale properties. We got an intermediate result:

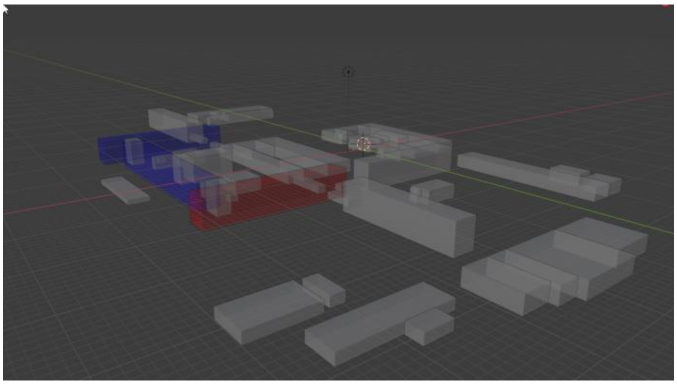

Fig. 2. Example of prototype 3d-model in Blender where all corpuses are basic and one is main

After import according to the instructions of the used work [2] and with some detail we got almost the final result:

Fig. 3. Result of importing model from Blender to Unity

D. Creation of a centralized system for managing and collecting data. Implementing HTTP and Web-Socket Servers.

This application requires a centralized system for processing, storing and managing data. This system in the context of interaction, proposed on image Fig. 1.

In the course of research, server was developed that processes requests from clients using Web-Sockets technology, the main functions of which are:

1) *Storing data in SQLite databases*
2) *Opening the port on which the paths will be presented (HTTP GET)*
3) *Synchronous operation of data forwarding*
4) *Work of previously written modules*
5) *Providing an interface for working with data structures*
6) *Notification of connected users about any data changes*

There was a problem of choosing a programming language suitable for the criterion of performance and containing many libraries for web development. Therefore, two programming languages were considered: NodeJS and Python. Due to availability of algorithms for some modules and in no way inferior in performance to others, as the basis was chosen Python. Since the server were accepting and returning requests, it was wise to use a "light" micro-framework like Flask. In work [3] the authors have described in detail the documentation for all the necessary functions of this framework.

Before implementing the server, modules were written for the schedule parser, filtering text for obscene words, as well as an interface for working with databases in SQLite [4].

About Web-Sockets

For the research, technology was being needed to implement synchronous data transfer in a client-server system. This technology is a protocol overlaid with HTTP – Web-Sockets. To implement a synchronous application that sends data to all users at once (*broadcast* = *True*), it was needed in the SocketIO library. In work [5] the mechanism of Web-Sockets operation as a synchronous protocol is considered.

The HTTP-server accepts GET and POST requests and outputs the result to the client. In it, as in a server on Web-Sockets, the paths along which client can navigate and receive information available to him are described, since for security purposes data protection is implemented by blocking some functions. The server also describes an interface that falls under the CRUD rule – [6].

E. Creating an application in the Unity development environment

Application's functions:

1) *Displaying 3d model and main UI*
2) *User interaction with the application - moving around the map, implementing events (clicks and toches)*
3) *Maintain constant communication with the server*
4) *Navigation functions*
5) *Implementation of AR (OCR text and labels)*

User interaction with the application.

Already at the prototyping stage, the application was conceived as a project with voxel graphics, which positions itself as 2.5D – this means that all objects look like three-dimensional, but they can be observed from one plane, moving at a certain constant angle to it (in the case of Unity in the XOZ axis plane). It is canonized by the aforementioned navigation apps as well as many mobile games.

Navigation functions

The Unity IDE contains many Maps that contain the building models, and many object Paths that keeps all the route models. To navigate "through the buildings", it needs to implement a function that would allow to lay all possible routes along the paths connecting conditional buildings A and B. Further, for all possible routes, using the necessary data, it needs to sort them and display the optimal and shortest paths to the user. It was done by knowing the weather data, user preferences and initial application settings. Each of such criterion will have a conventional unit of weight. In other words, ranking of the paths was based on the principle of choosing the maximum weight obtained by summing up many of the above criteria.

To find all possible routes between two points on the map, based on the existing objects in Unity, it is necessary to implement such a mathematical model as a weighted (separately weighted) undirected graph. In the resource [7] it was founded all the necessary algorithms to realize this technology. Next, an adjacency matrix is created for the paths so that it can later know which paths are connected.

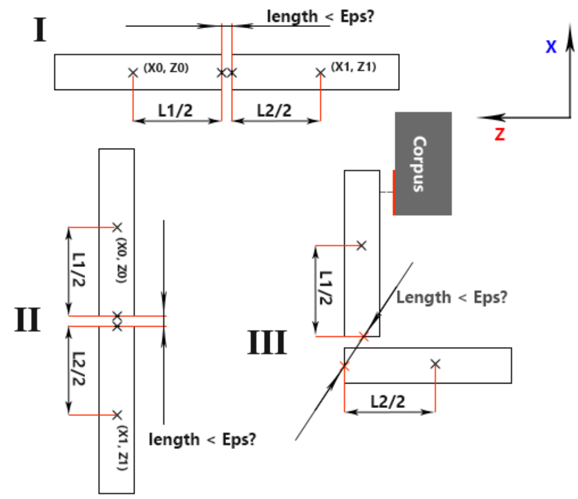

Fig. 4. Algorithm of how works the automatic generation of the adjacency matrix. Here: Roman numerals indicate variations in the location of paths as objects.

Since the element of the a_{ij} matrix took on the values True or False, it was necessary to enter in parallel an array with path lengths. By this way, the navigation module had has all necessary data in order find optimal routes on the map. Further, an additional set was introduced, which indicated which of the paths was connected to the entrances of the buildings. The entrances to buildings are already contained on the map in the Entries folder, where each building has at least one entrance. Therefore, the algorithm contained next steps: initialization of corpuses and their inputs as GameObjects and launching the breadth-first traversal algorithm. If such a path passing through intermediate objects exists, then Dijkstra's algorithm is launched. In work [8] matches the adjacency matrix of paths as well as their lengths**.** Result will contain many structures containing chains of passing paths, the length of the entire route, as well as the approximate travel time.

Implementation of AR and object detector with text

This study was at least viewed as integrating AR technology into an application that would allow using the NRNU MEPhI map navigation functions. This technology will be used as a route building on navigation maps. To plot visual steps along the navigation paths, it was necessary to connect libraries aimed at implementing AR — Vuforia [9]. This library made it possible to visualize three-dimensional objects in augmented reality, in our case, on a smartphone screen.

One of the main functions of the application is the identification of the audience by its image of the door and the index plate on which information about where classrooms are located (data: classroom's number and additional information). Definition of the types of objects and text was done by Google Cloud Vision algorithms, which are connected via the REST API. The request and response are characterized by JSON-Objects input, which are parsed at the application input. In other words, cyclically receiving individual frames from the camera, they are translated into byte[] and sent via a GET request to the Google servers, after processing the image, the API returns a JSON-Object, which contains detailed information about the image (label, text).

F. *Putting together a test setup to check the data transfer and the functionality of the application already on the smartphone.*

This installation will be represented by a local network, where through a router (in this case, Huawei HG8245H), the server was configured as follows: it was founded out of the mac-address of the computer inside this local network: CMD> ipconfig> MAC-address and IP. In router settings, it was set to DHCP as static, specifying the received Mac-address. For this, there is an article[10] on the topic of local networks.

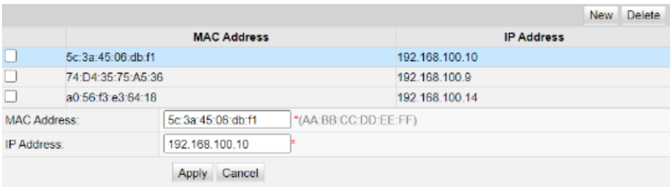

Fig. 5. Example of setup DHCP to StaticHCP

Then in "Port Forwarding" it was created an HTTP application, specifying the interval of used ports.

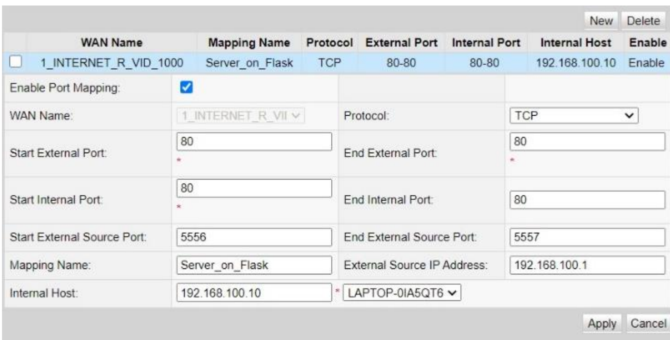

Fig. 6. Example of creating port forwarding for ports mapping in app

This installation allows to test the application and creates a virtual environment, exactly similar to the real one, where artificially created users will be real ones, and a cloud server will act as a computer.

III. Results

As the result there is a finished product [11] in the form of an android-application for ordinary users — students, employees, teachers. The resulting model is presented as a web-server for programmers who want to connect it to other projects, because project was written with the REST API provided. So far, the server is running inside a local network based on a test setup. After some testing, the following results were obtained: when connecting more than 5 users, the system works stable. After virtual connection by running 50 clients in parallel (another mini-test installation on NodeJS Express) showed almost the same result: stability, but slow web servers (the health indicators of the HTTP-server slowed down slightly relative to the indicators of the Web-Socket server).

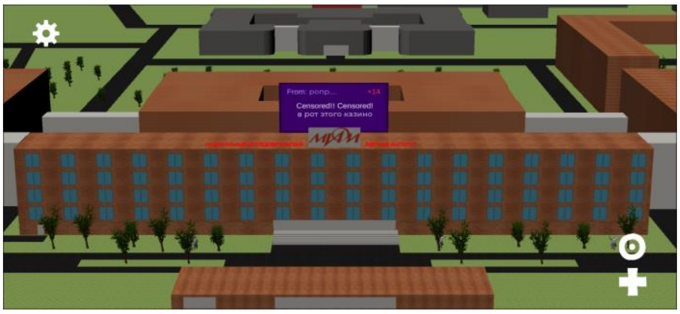

Fig. 7. This is how this application looks like

IV. Discussion And Conclusions

In current study most of the resources were analyzed on related topics with this project. Based on the works of all authors who were mentioned, from this we can conclude that these topics will develop and integrate into many areas of activity.

Client-server interaction system was designed, an android-application and web-server were developed for members who are included in the NRNU MEPhI. Also, a system for testing web applications in a local network was modeled and implemented.

The potential of this application is enormous, but there are some disadvantages: for example, non-autonomous work without the Internet and a GPS-module, incompleteness of objects located on the map.

In the future we will plan to research:

1) Analyze the existing resource base for social networks and implement a real network based on this application.

2) Conduct a global research and testing of the resulting project for students on their interaction with this application.

References

[1] H. Li and L. Zhijian, “The study and implementation of mobile GPS navigation system based on Google Maps,” 2010, doi: 10.1109/ICCIA.2010.6141544.

[2] W. McDermott, *Creating 3D Game Art for the iPhone with Unity iOS ● About Book*. 2011.

[3] A. Ronacher, “Flask (A Python Microframework),” *http://flask.pocoo.org/*, 2013. .

[4] M. Owens, *The definitive guide to SQLite*. 2010.

[5] D. G. Puranik, D. C. Feiock, and J. H. Hill, “Real-time monitoring using AJAX and WebSockets,” 2013, doi: 10.1109/ECBS.2013.10.

[6] C. O. Truica, F. Radulescu, A. Boicea, and I. Bucur, “Performance evaluation for CRUD operations in asynchronously replicated document oriented database,” 2015, doi: 10.1109/CSCS.2015.32.

[7] E. R. Scheinerman and D. H. Ullman, “Fractional Graph Theory A Rational Approach to the Theory of Graphs,” *Notes*, 1997.

[8] Y. Deng, Y. Chen, Y. Zhang, and S. Mahadevan, “Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment,” *Appl. Soft Comput. J.*, 2012, doi: 10.1016/j.asoc.2011.11.011.

[9] Vuforia, “SDK Download | Vuforia Developer Portal,” *developer.vuforia.com*, 2018.

[10] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2014.