sciendo

Polish Cartographical Review
Vol. 52, 2020, no. 1, pp. 16–26
DOI: 2478/pcr-2020-0003
Received: 26.03.2020
Accepted: 20.05.2020

BEATA KONOPSKA, MATEUSZ KAMIŃSKI
Maria Curie-Skłodowska University
Faculty of Earth Sciences and Spatial Menagement
Lublin, Poland
ORCID 0000-0002-9302-5164
beata.konopska@poczta.umcs.lublin.pl

City guide as a multi-sensory mobile phone application

Abstract. The authors analyses the issues inherent in implementation of a multi-sensory mobile application which uses a map as an interface for an edutainment-style city guide for tourists. Two models – the triad of tourist experiences (3E), i.e. education, entertainment, and excitement, and the hierarchy of needs of Abraham Maslow – were used as the basis for identifying what conditions should be met by such an application to encourage its use by both local residents and tourists, in equal measure. It was decided that only open source software would be used to achieve the goal of the application.

Keywords: mobile software, multisensory, city map, city guide, needs hierarchy

1. Introduction

The idea of developing a city guide in the form of an educational app game for smartphones that could be used to learn the topography and history of the city was born in response to an analysis of the market for this type of software and the fact that the sense of direction of contemporary people, especially younger ones, is worsening. The most widely available mobile phone apps do not require much of intellectual effort from their users, and are even less demanding in terms of physical activity, despite the fact that modern phones are equipped with many sensors and receivers, such as GPS, gyroscope, compass, accelerometer or camera, which make it possible to explore one's surroundings in a completely different way than a traditional guide.

The idea to give the city guide the form of a multi-sensory[1] mobile application, making it an educational outdoor game, places it within the growing field of edutainment tourism publications[2], i.e. ones which combine entertainment and knowledge (A. Stasiak 2013). The application was created for two groups of users: tourists who are not opposed to active ways of learning about new places, and city residents who like to spend their free time in an active way. These two groups of users – tourists and residents – differ in their level of interest in the city. After all, local residents will be less intrigued by this type of application, as contact with places that they see every day will provoke weaker emotional reactions in them than in tourists who are additionally affected by novelty. Therefore, the main research question posed for this project was as follows: what features would the application have to offer to encourage both local residents and visiting tourists to use it in an equal measure? Naturally, other more complex questions followed: what elements should the game scenario contain, and what mechanics should be used in the application for the

[1] The concept of the application assumes involvement and activation of several sensory channels to process information acquired during the game, which involves visual, auditory and kinaesthetic modalities.

[2] Edutainment is a term used to describe a skilful combination of knowledge and entertainment, the nature of which, according to A. Stasiak (2013), is best expressed by the Polish saying: "to teach by entertaining, and entertain by teaching".

final product to entertain but also fulfil its educational role, and, in turn, for the educational element to engage the recipient not only intellectually but also physically?

Such fields as psychology, theory of information transfer, and computer science were mined for solutions for stimulating the thought processes of tourists and local residents, and maintaining their level of engagement (both mental and physical). Two following models proved to be helpful: the triad of tourist experiences (3E) – education, entertainment, excitement (A. Stasiak 2015) and the hierarchy of needs of Abraham Maslow (2006).

Achievement of the cognitive goal means acquisition or deepening the app users' knowledge of the history of the place, as well as improving their spatial and analytical skills. The set goal of the app involved implementation of a universal prototype, which verified the feasibility of the employed technological solutions.

Due to the social (non-commercial) nature of the application, the project assumed the widest possible use of non-commercial IT tools. Tests of various substantive and technological solutions were conducted on the basis of the city of Lublin, the biggest university city of Eastern Poland, with population of over 300,000 people and more than one million tourists every year.

2. Technology and spatial data

The project's purpose was defined as recreational and educational, stimulating physical and mental activity of selected groups of recipients, and Lublin was indicated as a test area, which allowed to not only develop the game's idea and scenario, but also determine the selection of technological open source software solutions. A detailed analysis of solutions available on the market, taking into account their prices and popularity among recipients, had been conducted before the final selection of technological solutions was made.

2.1. The operating system of the prototype

Over 84% of all mobile devices support Android system, which has many compatible versions (https://www.idc.com/promo/smartphone-marketshare/os). Android is a continuously developed product, and its subsequent versions introduce additional features. The popularity of devices which use this system was the decisive factor for selection of Android for implementation of the project.

The high level of availability of the Android system is important not only for the user, but also for the programmer, because of the need to test prototype IT solutions on at least several different devices[3].

2.2. The prototype engine

When choosing a game engine[4], it is crucial to provide the optimal tool to achieve the designated goals. The two most popular programmes offering platforms with game development components are Unity and Unreal Engine. Unity is easier to use than Unreal Engine and offers a more extensive base of components to programmers with limited experience in creating models and soundtracks (https://docs.unity3d.com/Manual/index.html). Unreal Engine has a more complicated user interface and requires the programmer to know the harder programming language, C++, whereas Unity requires knowledge of C Sharp, which is an easier language. Therefore, Unity was chosen to achieve the goals set in the introduction, as a more appropriate tool for the inexperienced programmer.

The Unity engine allows for creation of 2D and 3D games, and it can be used with many operating systems, such as Microsoft Windows, macOS, and Linux. It makes it possible to create games on more than 20 different hardware platforms, including Android, Linux, Oculus Rift, or web browsers. The engine offers a platform (Asset Store) which includes such components as sounds, graphics, and ready-made programming solutions.

2.3. Graphic design

Graphic design of the game is responsible for the user's first impression of the product

[3] The tests can also be performed on virtual emulators, i.e. programmes that simulate the operation of selected devices and allow to choose the appropriate one. However, emulators have many imperfections, e.g. it is impossible to test cameras and other sensors using emulators, therefore, in many cases, the use of a real device is still necessary.

[4] The game engine is the main part of the computer game's code, which is responsible for such elements as graphics, interactions between objects, and sounds.

and is the decisive factor for the user's interest in the game. Beginner game developers usually use websites that provide ready-made graphics solutions.

Flaticon is one of the websites that support people interested in creating mobile applications. It collects, stores, and shares Flat Design style graphics. The site has approximately 1,200,000 icons, divided into almost 24,000 groups, representing various thematic categories. The website is free to use in its basic, standard scope, but if you want to customise any graphics to fit your specific needs, you have to pay for it, as it constitutes a non-standard service. The application should include a tab listing the authors of the used graphics (https://www.flaticon.com/). Flat Design is one of the most commonly used styles in web design and is widely used in touch screen mobile devices. The three main reasons for the robust development of this graphic design style include its simplicity and intuitiveness, the fact that it is an ideal fit for responsive[5] and adaptive[6] design, not to mention, its self-perpetuating popularity. This design style reflects the main trends in the design of Google web services and Android applications (J. Cao et al. 2015).

2.4 Augmented reality tools

ArCore is software that allows for implementation of augmented reality (AR) technology in applications designed for mobile platforms that analyse the camera recordings taken by the device for the purpose of recognising images and planes in real time. According to Ronald T. Azuma (1997), the Augmented Reality System combines real and virtual elements, is interactive in real time, and indexed in three dimensions (cf. H.K. Skórska 2017). This functionality allows to position virtual objects in relation to the real world. Thanks to an algorithm which operates in real time, the virtual object remains in relation to the real object, seen on the user's screen, creating an impression that it is part of the real world.

ARCore supports many types of 2D graphics, and 3D models, and offers an option of positioning virtual models without a prior detection of an image or object. It allows for attaching a virtual object to a plane or marker. The built-in algorithm corrects the position of the virtual object on the basis of data received from the device's sensors (gyroscope, accelerometer), ensuring that the user's movements do not change the location of the object in the real world. ARCore has APIs (Application Programming Interfaces) for C++, Java, and C#, as well as a special extension for the Unity engine. This solution allows the programmer to create applications that offer a chance to experience augmented reality on the Android platform. ARCore's strength is the undemanding procedure for placing new 2D and 3D objects in the real world. Any application created in Unity in the above-described way can be launched on all devices with Android operating system, version 7.0 or newer (https://developers.google.com/ar/discover).

2.5. Spatial data

In the last several years, the amount of spatial data available via the Internet has been significantly increasing. The phenomena is linked to availability of various location services and computer tools that allow for creation of one's own thematic maps (P.J. Kowalski 2007, D. Gotlib 2008, K. Król 2015, K. Król, L. Szomorova 2015). The choice of an appropriate spatial data source for the application is determined by the software used for creation of the game. There are many services providing spatial data, but only the following were taken into account for the project: Google Maps, Open Street Map, Geoportal 2, and Mapbox.

Google Maps is one of the most visited map services in the world, and Google search engine has been at the top of popularity charts for years, according to the Alexa website (https://www.alexa.com/topsites). The service and its possibilities have been widely discussed in the literature of the subject (Ł. Halik 2011, K. Król 2015). In turn, Open Street Map (OSM, openstreetmap.org) is the most popular social networking site that allows for creating and using

[5] The use of Responsive Web Design ensures that the prepared website will adapt to the browser and device conditions on which it is displayed, will look good regardless of the browser used by particular users, by changing the size and arrangement of individual elements of the site, along with its resolution.

[6] Adaptive Web Design ensures that the website is displayed in the best possible quality. It requires preparation of several (usually 5 or 6) versions of the site in different resolutions.

spatial data resources. OSM is a global project and its goal is to create a publically available map of the world. Its open nature is a specific feature of the project's design. The literature of the subject includes many a publication on the quality of its data, the vast majority of which concern urban areas (M. Haklay 2010, J.-F. Girres et al. 2010, S.S. Sehra et al. 2013, J. Nowak Da Costa et al., 2016, S. Marczak 2015, 2017). The Geoportal 2 project (geoportal.gov.pl) is the result of implementation of the European initiative INSPIRE in Poland and it is being created in accordance with the Act on Spatial Information Infrastructure. It allows for combining spatial data sets coming from various government resources into an integrated set, which is made available to the public. As in the case of Google Maps and OSM, the literature of the subject includes also a thorough assessment of the resources and functionality of Geoportal 2 (D. Dukaczewski, E. Bielecka 2009, M. Ślusarski 2012, A. Dawidowicz et al. 2014)

Ultimately, it was decided that the right choice for the project is Mapbox (mapbox.com/maps/). It is an Internet platform for creating on-line maps, equipped with tools for designing and publishing maps based on open source software and open data, e.g. from Open Street Map resources. In this case, no additional software has to be downloaded to view maps, and all the needed tools are available from the level of Mapbox (C. Cadenas 2014). The platform makes it possible to display data with full adaptation of graphic elements (symbolization modification, and generalization) to the user's needs (https://www.mapbox.com). The Unity engine has a dedicated MapBox plug-in that allows to add a world map to the developed application.

3. Game scenario, and course of the game

The plot of the game consists of several phases. Its main character is the user who is introduced to the game process after creating an avatar that represents them[7]. At the beginning of the game, the user determines the appearance of their virtual representative, selecting individual elements from the prepared base.

After creating the avatar, the virtual partner explains the goal and course of the game to the user with the help of short messages, and, afterwards, moves them to the main menu. Here, the user decides to start the game (or checks the status of the interrupted game) and, after moving to the next screen, selects one of the proposed tasks. The task is to find a historical object located in the city.

At the start of the game, a city map is displayed and the avatar provides information about the place where the game begins. For example, when the route starts before the entrance to one of the shopping malls, the message reads: "You will never want for anything. Clothes, food, entertainment. The slender columns will show you the way." The user learns the route to their destination during the game – an indicator on the screen shows the user the estimated distance from their goal and monitors whether they are moving in the right direction. While performing the task, the user takes advantage of hints displayed on the screen, using the features of field objects.

When the player reaches their destination, the screen displays a message confirming that the object has been found. A moment later, the user's device's camera activates and the augmented reality screen appears. To complete this stage, the user needs to find an indicated graphic target, e.g. a road sign, store logo, clock, etc. Information about the target is displayed on the screen. When the user finds the target and points their camera at it, the screen will display the virtual mechanism of the logical puzzle in relation to the real world (fig. 1).

To complete the task, the user has to solve the puzzle related to the found object. The puzzle can consist in arranging blocks to match their colours, connecting dots in the correct way, or answering a quiz question. As a reward, the player receives points and a brief historical description of the object. If the user has any difficulties with solving the puzzle, they can consult the avatar. However, this operation reduces the number of points awarded for completing the entire route. After solving the puzzle and receiving the points, the user gets a hint on how to get to the next destination, which is e.g. the main city hall. The procedure is then repeated until the entire route is completed. The reward screen with a summary of the results is displayed at the end of the game. Then, the user moves to the map view on which the already

[7] Avatar is a virtual character that symbolises the player.

located objects are marked, and can start a new round.

The objects found during the game create educational trails with historical themes, e.g. “World War II” (objects and places referring to this event), “The Authorities” (objects referring to the history of the city’s authorities), “Culture”, “Belle Époque”, “The Interwar period,” etc. Some objects are included in more than one trail, but and adding one’s own graphic elements in the indicated coordinates.

The coverage of the generated map is limited to the area where the game is played. The content of the map and its graphic design scheme were adapted to the purpose and functionality planned for the application. The basic elements of the map are objects of historical significance and pedestrian routes (streets, squares, pro-

Fig. 1. Virtual view of the logical puzzle in relation to the real world

the information provided to the user is always focused only on one context. Each route is linked to a set of logic puzzles, whose graphics and content are adapted to match the route’s theme. All information is grouped and can be displayed in the map view (fig. 2).

3.1. Preparation of a city map

When preparing the application, development of a new city map was also initially considered. However, the use of a self-made map would create the need to perform complicated conversions of the actual position of the player in the WGS-84 coordinate system into the simulated Cartesian space used in Unity, which could potentially generate erroneous results. The issue was solve by using the Mapbox tool, which has been developed for the Unity engine, and allows for dynamic generation of the map menades, alleys, passages between buildings, paths, and courtyards, etc.), because during the game the user will search foremost for spatial objects and will have to determine the shortest path to reach their destination. If users are to be able to determine the shortest route, a detailed picture of pedestrian routes has to be provided on the map, because the player will use every opportunity to get through to their destination. As the application not only emphasises historical education, but also aims to improve the user’s sense of direction, when developing the content of the map, great care was taken with representation of topographic elements that can help the user figure out where they are in urban space, such as monuments, squares, characteristic shops, churches, etc.

The application uses a mechanism of dynamically adjusting symbols which adapt to the changing scale of the map. Symbols on a larger-

Fig. 2. A sample map view

-scale map should show the features of the object described in the hints, e.g. by colour differentiating objects on the basis of their height expressed in a number of floors, or, in the case of a symbol representing a specific object, the sign object should refer to its characteristic features, e.g. its architectural elements.

3.2. Adaptation of the application to the perception and physical capabilities of users

Determination of the characteristics of the app's recipients and adapting it to their perceptual and physical abilities was one of the project's more detailed objectives. As the goal is to reach users from fairly broadly selected target groups (tourists and local residents of different ages), it is necessary to anticipate and solve many substantive and technological problems arising from such issues as different levels of general knowledge, perceptions of the environment, skills in mobile phone usage, and physical fitness, etc. For example, not everyone will understand references to contemporary popular culture or local traditions. The same applies to the stimuli or gestures related to information technology, such as sliding one's finger from the right to the left edge of the screen, spreading two fingers on the screen to zoom in, shaking the phone, or tilting it. If the user is unfamiliar with such phone features as the gyroscope and accelerometer, they may feel lost. Lack of knowledge of the principles of augmented reality technology or navigation can significantly hinder or even prevent the user from enjoying the game. Hence the implementation of optional tutorials to help understand gameplay-related issues.

The problem resulting from different levels of general knowledge about the city was solved by making sure that logic puzzles refer only to the knowledge provided during the game. Furthermore, any potential problems related to moving within urban space were solved by formulating hints for the sought locations by referring to them only with the official names that are identified in the field.

In turn, the problem related to physical fitness was solved by limiting the distance between successive points of interest (POI). In the im-

plemented project, the distances covered by the user do not exceed two kilometres, because if greater distances were used, the application would have to provide information about urban transport and include additional stimuli, keeping players engaged in the game while waiting for or travelling in a chosen means of transport.

3.3. Maintaining a high level of engagement in the game

The hierarchy of needs proposed by A. Maslow (2006) prioritises the basic needs that every human being tries to meet in their life. The concept of the game and application is based on the knowledge of some of these needs – the need for belonging, recognition, respect and self-fulfilment. A well-designed game is received in a multi-sensory manner, as it contains a number of programmed mechanisms that directly or indirectly affect the user, encouraging them to continue or even restart the game. In accordance with modern leisure time trends, the most important result of the game should be the user's satisfaction with the entertainment experience, the resultant of which is education (A. Stasiak 2011, 2015). The solutions adopted in the application, together with an interesting gameplay, should meet the most important needs of the user who will have a good time while at the same time acquiring historical knowledge and improving their skills in map reading and their sense of direction.

Interactions, creation of groups, and mutual interests are the cornerstones of the lives of a large part of society. Cyberspace is also the place where groups are formed to share thoughts, conduct informal discussions, or technical dialogue (A. Naruszewicz-Duchlińska 2014). When creating a single-player game, the player's need to belong to a group can be satisfied by adding an on-line forum that could bring users together in one on-line location, allowing them, for example, to solve puzzles or arrange a joint game. Maintaining interest in the game is facilitated by the section allowing for communication with the programmers. Its purpose is to allow for sharing opinions on the functioning of the application and logic puzzles.

A chance to observe the process of gradually improving one's result and climbing higher and higher on the scoreboard is an effective motivator for further engagement with the game. The related need for recognition becomes even more important when the application implements many rankings or shows results of many players, e.g. one scoreboard highlights the leaders with the largest number of objects found, the other, those who spent the most time playing the game, while a different one presents only the achievements of local community members, etc.

The need for self-realisation is the last type of need that has been taken into account when developing the concept of the game and application. By playing the game, each player gains historical knowledge about the city, improves their sense of direction, and logical thinking skills, as well as their ability to analyse and combine facts by solving logical puzzles.

4. Programming

Work on the application involves the development of detailed documentation (J. Schell 2014), planning of its mechanics, optimisation, unified nomenclature of classes, methods, and variables in the code. The names used must be unambiguous so that a reference to the method or variable does not require searching through the entire code (J. Schreier 2018).

The programming architecture is inspired by the Model-View-Controller (MVC) design pattern, with the additional implementation of a double state machine. The main controller, which is also a state machine, controls the entire logic of the game. The user interface controller is responsible for the appropriate setting, while the virtual controller is responsible for the data. The gameplay is controlled by an internal state machine that performs the logic of the appropriate stage of the game. Figures which record e.g. the player's progress are saved locally on the user's device in a file encoded into JSON format. Because players do not compete directly with each other, no Internet access is required. After the game is over (before leaving it), all data is saved in JSON format on the device and loaded again once the game is restarted. The list of all achievements is an important addition, because, as already noted, it is the basis for maintaining a high level of interest in the game (E. Freeman et al. 2004). Because of the above, it was ensured that, as recommended, the latest record was automatically loaded when starting the application, and, the

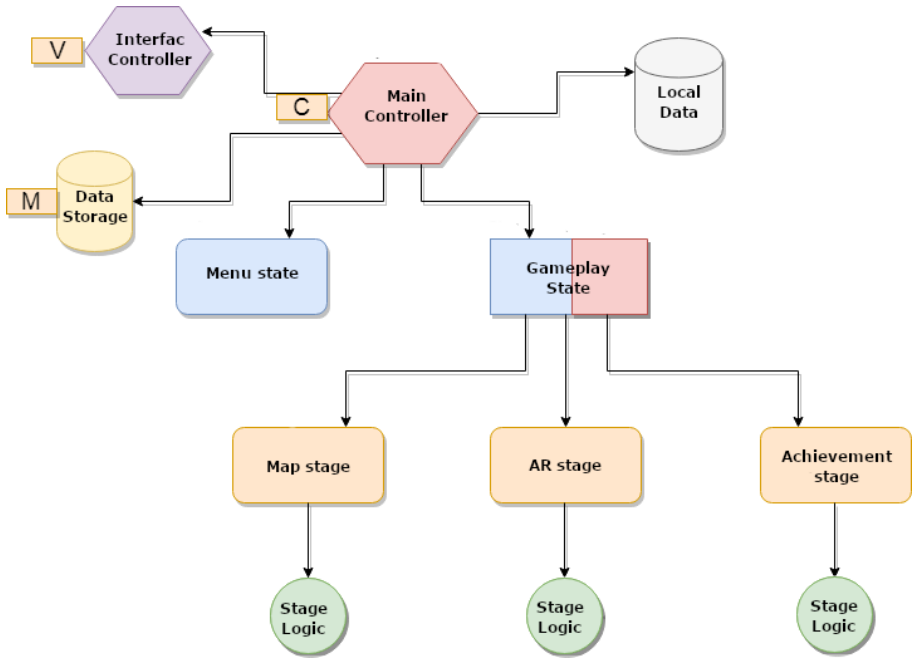

Fig. 3. System architecture diagram

just-made progress was loaded when the device was turned off (R. Martin 2014).

The game consists of a menu screen, city map screen, showing the location of the found objects and other elements of the game, while the gameplay is based on searching for the target and the AR state in which the user solves the puzzle. In this case, it is beneficial to nest one state machine in another and thus gain control over the gameplay process. Still, if an error occurs, it is easy to locate. This solution encapsulates the mechanics and helps maintain order in the design. When a state machine is used, each state has its own object implementing the logic of the segment.

4.1. Technological description of the game

To start the game, you need a city map, rules of the game, and an avatar. When the user clicks “start”, the state machine will use the state of the game and provide a reference to the model. After receiving the reference, the state will download the appropriate objects and carry out the initialisation. Based on the MVC pattern, three basic design elements will be created in the first phase (Unity does not offer ready-made design patterns).

The role of the view is played by the user interface controller, which is responsible for displaying, turning off and animating interface elements. The key issue is to completely eliminate game logic elements from the controller. In addition, the interface control class provides instructions on how to activate game appearance elements. At the start of the game, the main state machine instructs the interface controller to turn on the gameplay view, and the view, in turn, activates the gameplay using the reference (UIGame) and turns off the menu view (UIMenu).

The main state machine of the gameplay can operate on the basis of two states. Each state is a class implementing an interface. For proper functioning, the state requires the use of three basic instructions (initialisation, update, and deinitialisation) (fig. 3).

The graphic elements of the menu are activated during initialisation, and during the update, the programme waits for the player to start interacting with the application. Using the option button, the main state machine sends a command to the interface controller to activate the

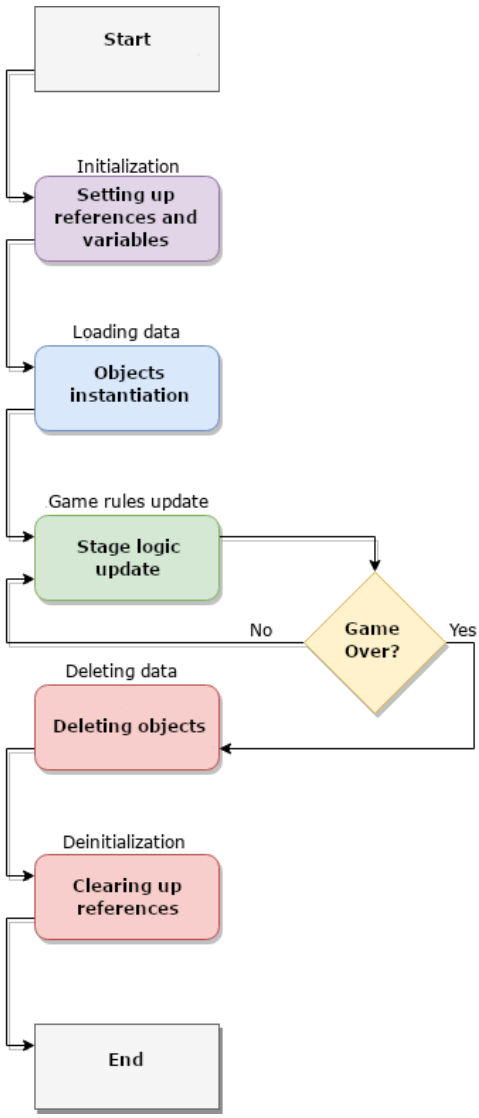

Fig. 4. Gameplay diagram

graphics elements responsible for the game settings. When the game starts, the menu status is de-initialised, all graphics are turned off and a second state is activated after selecting the option to start the gameplay. The gameplay state has two functions – that of a state and that of a state machine. The state machine has two necessary instructions. When starting the second state machine, once the route is selected, the programme activates the gameplay stage, in which the player sees the city map and their location. Once the task is completed, the augmented reality state will activate automatically. When the player returns to the menu, the internal state machine will de-initialise all states and send notification to the main state machine that the gameplay is over. After receiving this notification, the main state machine deactivates the gameplay state and activates the menu state. In order to avoid infecting the gameplay code with the remaining parts of the programme, the logic responsible for the gameplay will be stored in a special separate object. The rules of the gameplay in force at the stage of using the city map, are different from those which are for the AR segment.

The gameplay process consists of initialisation, loading data, executing the rules of the gameplay, deleting data, and deinitialisation (fig. 4). This division guarantees the transparency of the programme's operations. The proper functioning is ensured by creating the class of base rules that all other classes of rules are to inherit from.

The transfer of information between classes is an important element of the software's operations. In the adopted schema, methods have been added to the UnityAction-type variable, which itself is based on the delegate, which are called upon when it is launched. For the purposes of the planned project, this system is used to record player statistics, but it required better adaptation to the project's needs. The "observer" pattern was used for this purpose, as, according to the literature of the subject, it constitutes an effective solution (J. Albahari, B. Albahari 2018). A listener variable has to be identified in one class, and its inheritances in the second class to make it possible to use the system. Once the reference is provided, calling the method in this type of variable immediately results in calling the method in the interface inheriting class. Both systems can be used interchangeably in the game, depending on current needs.

4.2. Gameplay mechanism

Once the game opens, a menu screen appears which allows the user to start the gameplay. When the "start" button is used, it sends information about the gameplay's start to the main game machine. Then, the game state is

initialised, i.e. the gameplay rules are loaded. The user sees a screen with a map image, information about their location, and the first hint. The object storing hints and information about the route start and end points is also loaded at the same time. Then, the application enters the mode of tracking the player's position and representing it on the map. The location is continuously checked in the WGS-84 coordinate system and converted by Mapbox tool to the coordinates of the Cartesian system used in the engine. The exact distance between points and information on the angle is required to calculate the position (https://demoman.net/?a=trig-for-games). With accurate location information, it is expect that the user's distance from the destination will be smaller than the assumed margin of error. When the player approaches the target location at a sufficient distance, a method is called that decides to activate the puzzle (or redirect the player to a different destination). Switching to the AR mode activates the augmented reality camera and launches a logic puzzle. The map view returns when the puzzle is solved or the user gives up on solving the problem. The process is repeated until the entire route is completed, or the user gives up on the game.

5. Conclusions

Planning and creation of a multi-sensory mobile application which is meant to be used in free time, but at the same time offers high cognitive values, combining mental and physical effort, is a very complex process. It is impossible to create an edutainment-type multi-sensory application without an in-depth multi-faceted analysis of intended recipients' perception capabilities and needs, as well as some knowledge of and familiarity with the use of open source programming tools. The decision to emphasise the importance of the 3E model (education, entertainment, excitement) and Maslow's hierarchy of needs for the project significantly influenced both the game scenario and the very concept of the application. The user's engagement with the game, which is maintained by various IT-controlled stimuli, results in acquisition of knowledge about the history of the selected places and improving the user's sense of direction. It should be emphasised that this educational effect is a result of entertainment, and not the other way round.

Development of the game prototype as universal base for similar implementations for other cities and testing it on the example of Lublin allowed for successfully completion of the project. Furthermore, the fact that the game was developed primarily with open source tools and in its cognitive aspect emphasises knowledge of local history and improvement of one of the basic skills, i.e. sense of direction, places the game among products with special significance for society.

Literature

Albahari J., Albahari B., 2018, *C Sharp 7.0. Leksykon kieszonkowy*. Gliwice: Helion.

Azuma R.T., 1997, *A survey of augmented reality*. "Presence: Teleoperators and Virtual Environments" Vol. 6, no. 4, pp. 355–385.

Cadenas C., 2014, *Geovisualization: Integration and visualization of multiple datasets using Mapbox*. "Computer Engineering" https://digitalcommons.calpoly.edu/cpesp/137 (access 20.02.2020)

Cao J., Zieba K., Stryjewski K., Ellis M., 2015, *Elegant Web UI design techniques: flat design & colors*, UXPin.

Dawidowicz A., Sońska M., 2014, *Analiza porównawcza wybranych geoportali europejskich*. „Acta Scientiarum Polonorum. Administratio Locorum" 13 (2), pp. 59–76.

Dukaczewski D., Bielecka E., 2009, *Analiza porównawcza krajowych geoportali w Europie*. „Roczniki Geomatyki" T. 7, nr 6, pp. 35–60.

Freeman E., Freeman E., Sierra K., Bates B., 2004, *Head First Design Patterns*. Polish ed. (*Rusz głową*!). Gliwice: Helion.

Girres J.-F., Touya G., 2010, *Quality assessment of the French OpenStreetMap dataset*. "Transaction in GIS" Vol. 14, no. 4, pp. 435–459. DOI: 10.1111/j.1467-9671.2010.01203.x

Gotlib D. 2008, *Nowe oblicza kartografii – Internet a kartografia*. „Polski Przegląd Kartograficzny" T. 40, nr 3, pp. 237–246.

Haklay M., 2010, *How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets.* "Environment and Planning B: Planning and Design" Vol. 37, pp. 682–703. DOI:10.1068/b35097

Halik Ł., 2011, *Zastosowanie Google Maps API do multimedialnej wizualizacji informacji o obiektach przestrzennych w Internecie*. „Badania Fizjograficzne" R. 2, Seria A, Geografia Fizyczna, pp. 17–26.

Kowalski P.J., 2007, *Znaczenie integracji danych geograficznych w serwisach internetowych typu mashup*. „Archiwum Fotogrametrii, Kartografii i Teledetekcji" Vol. 17a, pp. 395–404.

Kowalski P.J., 2012, *Mapa jako praktyczny interfejs serwisu internetowego*. „Archiwum Fotogrametrii, Kartografii i Teledetekcji" Vol. 23, pp. 159–168.

Król K., Szomorova L., 2015, *The possibilities of using chosen jQuery JavaScript components in creating interactive maps*. "Geomat. Landmanag. Landscape" no. 2, pp. 45–54.

Król K., 2015, *Tworzenie statycznych map obiektów przestrzennych z wykorzystaniem metody parametryzacji adresu zasobu sieciowego – analiza porównawcza*. „Acta Scientiarum Polonorum. Formatio Circumiectus", 14(4), pp. 61–73. DOI: 10.15576/ASP.FC/2015.14.4.61

Marczak S., 2015, *Ocena zaangażowania społeczeństwa w tworzenie danych przestrzennych w Polsce na przykładzie projektu OpenStreetMap*. „Roczniki Geomatyki" T. 13, z. 3, pp. 239–253.

Marczak S., 2017, *Wyznaczenie cech społeczeństwa wpływających na zaangażowanie w tworzenie VGI w Polsce*. „Roczniki Geomatyki" T. 15, z. 2, pp. 233–248.

Martin R., 2014. *Czysty kod. Podręcznik dobrego programisty*. Gliwice: Helion.

Maslow A.H., 2006. *Motywacja i osobowość*. Warszawa: Wydawnictwo Naukowe PWN.

Naruszewicz-Duchlińska A., 2014, *Nowe media a nowe gatunki – wstępna charakterystyka genologiczna internetowego forum dyskusyjnego*. „Postscriptum Polonistyczne" T. 2 (14), pp. 27–38.

Nowak Da Costa J., Bielecka E., Całka B., 2016, *Jakość danych OpenStreetMap – analiza informacji o budynkach na terenie Siedlecczyzny*. „Roczniki Geomatyki" T. 14, z. 2, pp. 201–211.

Schell J., 2014. *The Art of Game Design: A Book of Lenses*. Second Edition, AK Peters/CRC Press.

Schreier J., 2018, *Krew, pot i piksele. Chwalebne i niepokojące opowieści o tym, jak robi się gry*. Kraków: Wydawnictwo SQN.

Skórska H.K., 2017, *Systemy i zastosowania rzeczywistości rozszerzonej*. „Przegląd Mechaniczny" T. 1 nr 7–8, pp. 48–52. DOI: 10.15199/148.2017.7-8.7

Sehra S.S., Singh J., Rai H.S., 2013, *Assessment of OpenStreetMap data – a review.* "International Journal of Computer Applications" Vol. 76, no. 16, pp. 17–20.

Stasiak A., 2011, *Turystyka rodzinna w świecie ponowoczesnym*. In: J. Śledzińska, B. Włodarczyk (eds.), *Turystyka rodzinna a zachowania prospołeczne*. Warszawa: Wydawnictwo PTTK „Kraj", pp. 27–38.

Stasiak A., 2013, *Produkt turystyczny w gospodarce doświadczeń*. "Turyzm" T. 23, nr 1, pp. 29–38.

Stasiak A., 2015, *The triad of tourist experiences and "wow". Effect as the basis for the creation of modern tourist offer*. "Research Papers of Wrocław University of Economics" no. 379, pp. 332–347. DOI: 10.15611/pn.2015.379.32

Ślusarski M. 2012, *Propozycja oceny geoportali internetowych poziomu lokalnego*. „Infrastruktura i Ekologia Terenów Wiejskich" Nr 1, pp. 109–115.

Internet sources

www.alexa.com

Dokumentacja ARCore, https://developers.google.com/ar/reference/unity (access 20.02.2020).

Dokumentacja Mapbox, https://docs.mapbox.com/ (access 20.02.2020).

Dokumentacja Unity, https://docs.unity3d.com/Manual/index.html (access 20.02.2020).

Funkcje trygonometryczne w grach, https://demoman.net/?a=trig-for-games (access 20.02.2020).

Strona Flaticon, https://www.flaticon.com/ (access 20.02.2020).

Udział systemów operacyjnych na rynku, https://www.idc.com/promo/smartphone-market-share/os (access 20.02.2020).