DAR: Implementation of a Drone Augmented Reality Video Game

Bianca-Cerasela-Zelia Blaga, Dorian Gorgan
Technical University of Cluj-Napoca
Computer Science Department
Cluj-Napoca, Romania
{Zelia.Blaga, Dorian.Gorgan}@cs.utcluj.ro

ABSTRACT
Augmented reality video games combine the elements of a virtual application with those of the real world. With the recent advances in technology, this type of digital content is easily available to users. The player experience is enhanced by bringing the virtual experience to the world around us, and new interaction techniques emerge. We propose a video game in which the user controls a virtual drone which interacts with objects situated in a real environment. We present the development methodology that we followed in order to bring this project to life, from the early stages of topic selection to implementation details, and finally to the evaluation step.

Author Keywords
Augmented reality, heuristic evaluation, video game design and implementation.

ACM Classification Keywords
H.5.2 User Interfaces.

INTRODUCTION
Augmented reality (AR) is a field of computer science that deals with the combination between the real and virtual worlds [4]. It is situated at the top of the Gartner technology hype cycle [16] because there is a growing interest for this field, and it can bring improvements in various areas, like gaming, medicine, teaching, travel, marketing, and research (Figure 1). The recent advances in the domain have made it more accessible and removed the expensive special devices needed. The computational power of consumer goods makes this technology available to everyone, and thus more applications are developed. Users become more accustomed to AR as a form of digital media, as the usage of phones, laptops, and social media has increased.

Figure 1. Market predictions for the augmented reality domain [1]

One special field for the development of AR applications is the video game domain. The most common interaction method is through tangible user interfaces, where real objects are represented as virtual ones in the application, and thus the user can interact naturally with them. With the increased popularity of AR glasses [21], the gaming world is being redefined. Another attribute of AR digital games is geolocation, which enables the usage of real-world maps and locations. The game environment is the one around the player, in which fantastic creatures exist, and the goal is to solve the quests. The participant uses the GPS on the smartphone to move around the world and performs specific actions that bring him closer to achieving the game goals. Some examples of successful AR games are Pokemon GO[17], Ingress [18], Temple Treasure Hunt [9], and Zombie GO [29]. Moreover, classic video games such as World of Tanks consider expanding themselves with AR capabilities [35].

In this paper, we take on the quest to create an AR video game that is centered around a drone. The need for such an application arises from the recent growth of the Unmanned Aerial Vehicles (UAV) business [27]. Having such an application, the developer learns the physics specific to such a device, and the users can interact and learn the controls of the drone. By making the player complete tasks and navigate in a real environment through AR capabilities, the levels of excitement increase compared to a simple virtual game.

In this paper, in section Related Work we present the recent advances in the domains of augmented reality and digital games. We then present the AR Game development method we followed. First, we talk about the Analysis stage which is composed of Topic selection, Game specifications, Prototyping, and Scenario and task description. Then we explain the Implementation steps in detail and assess the usability of the application in the Evaluation section. We summarize the results and come with ideas for future directions in Conclusions.

RELATED WORK

Encyclopaedia Britannica [10] gives the following definition for AR: “Augmented reality, in computer programming, is a process of combining or ‘augmenting’ video or photographic displays by overlaying the images with useful computer-generated data.” The start of this domain came in 1968 with Microsoft’s wireframe drawings that showcased the prototype for Hololens [28]. In 1990, the term “Augmented Reality” was coined by Tom Caudell [2], a Boeing researcher, followed in 1994 by the first AR theater production by Julie Martin which featured performances around virtual objects [11]. An open-source development toolkit named ARToolKit was introduced in 2009 by Hirokazu Kato [33], which was succeeded by Apple’s ARKit [34] and Google’s ARCore [20] for the creation of iOS and Android applications. Three main industries stand out eCommerce, marketing, and gaming.

The appeal to games that employ Augmented Reality technology is that the player is actually the character, he/she does not just play as the character. Therefore, the level of immersion is bigger and enjoyment – higher. For example, Disney and Lenovo teamed up to create the “Star Wars: Jedi Challenges” [30] that brings the fans of this franchise closer to their beloved characters by allowing them to maneuver lightsabers and fight Darth Vader. Lego is another company that built an AR application using Apple’s ARKit [19]. In it, the players can combine Lego bricks with virtual elements for a game of dragons versus robots for example.

Another possible application for AR in gaming is represented by Merge’s 6DoF Blaster [13] that uses the smartphone to create digital targets that the users can shoot at. Even though at first glance this device is a plastic gun that uses a phone with AR capabilities, the freedom of movement that allows the player to move unrestricted in the real world is what makes players enjoy it a lot.

Studies have shown that these types of game promote healthy behaviors and improve children’s emotional health. AR technology has also proven to be effective in the educational field by teaching in school subjects such as archaeology, anthropology, and geography [6].

The Augmented Reality domain has some limitations that researchers strive to surpass. Work is carried on for creating a flawless virtual world in which orientation and direction adjust rapidly, with no errors or in which the player’s movements are mapped in real time.

AR GAME DEVELOPMENT

To create an AR digital game [26], researchers use the following components for the hardware:

- a computer, either a PC or a mobile device,

Figure 2. Example of wireframe for a building [25]

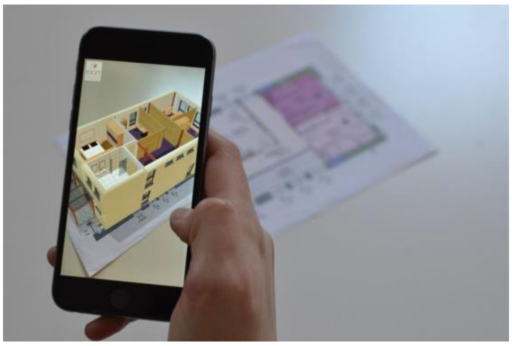

Figure 3. 3D model of building with the 2D image [31]

- a monitor or display screen,
- a camera,
- tracking and sensing systems (GPS, compass, accelerometer),
- a network infrastructure,
- a marker – a physical object where the real and virtual environments are fused together.

As for the software used, this is:

- an application or program running locally,
- Web services,
- a content server.

We used these components to create our own digital game with AR capabilities and followed an iterative development methodology that starts by analyzing the project – deciding on the topic, goals, specifications, and prototype, followed by the implementation step in which the application is created, and which is further evaluated in order to identify problems that need to be improved in the next iteration.

ANALYSIS

In this section, the main ideas and goals for the application are established, together with the video game tasks, initial interface prototyping, controls, and some gameplay scenarios. These represent just an outline for the application which is developed and can be changed in future stages since the game development methodology is an iterative process that strives to improve any detected flaws.

Topic selection

The proposed game came from the idea of having drones that collect materials which are carried to the necessary location by means of a train. The player can interact with the game using a smartphone, by seeing the building in which materials and enemies are located.

The player can control the drone's position using the buttons which are on the screen of the phone. To make the game more difficult, enemies that are controlled by Artificial Intelligence (AI) are inserted in the scene.

The game can be extended by integrating a real-life drone with which the player can directly interact as shown in [3].

Game specifications

The main functionalities of the application are:

- the player can observe the game scene from multiple perspectives,
- the building in which the materials are located is populated with AI which attacks the drone or destroys the materials,
- the player can select the desired drone,
- the player controls the drone's position using the buttons on the application's interface,
- the player performs the grab/drop action with buttons placed on the screen of the smartphone,
- the collected materials are placed in containers that are located in the game scene.

The drone will not be affected by the movement of the phone, and its physics will have to be precise. The enemies will have a limited field of view and will attack only when the UAV is at a certain distance to them.

There are three options for the building in which materials are placed. The first one, as seen in Figure 2, is to have an object of type wireframe, with digitally mapped shape. The second one is to have a 2D image with a map, as seen in Figure 3, which will display a 3D building when looking through the phone's camera. The third option would be to have the whole room as a playing field. The first two options are very limiting in terms of size for the game objects, while the third one offers a bigger area and freedom of movement.

Prototyping

For the beginning, the player will have a simple user interface to control the drone, which will look like in Figure 4, together with a game menu seen in Figure 5. With the help of the camera, the player can see the game scene, and with the help of the joystick buttons from the left and the right side of the screen, the position of the drone can be changed. The buttons from the middle are used to pick and drop objects, while the arrows control the translations and rotation on the 3 axes. Since the option of using the whole room as a playground, we will implement an object collision algorithm that detects horizontal and vertical planar surfaces. The player can place objects in the scene with the use of markers. These are for two types of game objects: materials and train. The player has to safely place the boxes in the corresponding containers. The final prototype of the application can be seen in Figure 6.

During the implementation phase, we have noticed that the latency of the application is high, and it takes a long time to load virtual objects. Therefore, the idea of using markers was put aside, and the option of tapping a surface to place an object has been chosen.

Figure 4. Mobile prototype; the scene is composed of the room (captured using the camera), while the drone is a virtual object.

Figure 5. Game menu prototype

Figure 6. In-game prototype – 1. the menu button, 2. the drone control buttons, 3. marker with virtual container, 4. marker for materials and AI enemy attacking the drone, 5. back button

Scenario and task description

The following scenarios for an Augmented Reality video game have been established:

1. Creating the game scene
    - the system displays the control buttons,
    - the system displays the virtual objects (drone, planar surfaces, cannons) on the images captured from the camera.
2. Game scene navigation
    - the player presses the left and right joystick buttons to control the position of the drone,
    - the player moves the phone and the system detects and displays the planar surfaces.
3. Real object collision
    - the player controls the movement of the drone,
    - the AR tool detects the planar surfaces of the real environment and creates collision meshes,
    - at a collision between the drone and the mesh, the first object will stop advancing.
4. Interaction with the enemies
    - the player taps a surface mesh to place a cannon enemy,
    - the system controls the movement and attack of the turret,
    - the drone will change its position (fly away or fall) when hit by a cannonball.

IMPLEMENTATION

In this section, we present the most important steps we took to create the video game and how we incorporated Augmented Reality technology into it.

Tools

For the development of the application, the software resources that were used are:

- game development: Unity [22],
- augmented reality: ARCore [20],
- backend: C#,

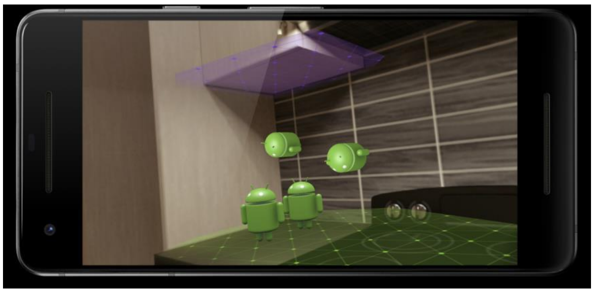

Figure 7. Real environment with virtual planar meshes and virtual objects [32]

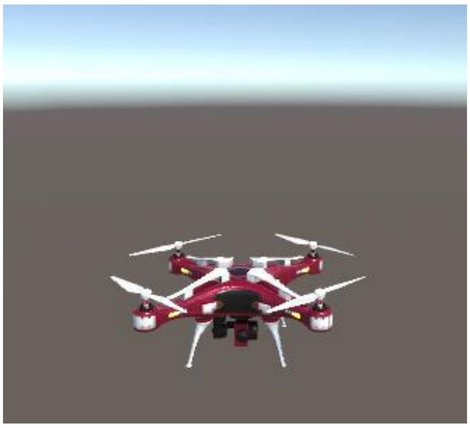

Figure 8. 3D virtual object representing a drone

- 3D modeling: Blender [36],
- prototyping and textures: Adobe Photoshop CS6 [23],
- Android development: Android Studio [5].

The hardware resources used are:

- for game development: a laptop with Intel(R) Core(TM) i7-6700HQ CPU, 2.60GHz, 8.0GB RAM, 1TB memory, NVIDIA GeForce GTX 960M,
- for game testing: a Samsung S8 smartphone with Android version 8.

Game objects

There are two types of game objects for this AR application:

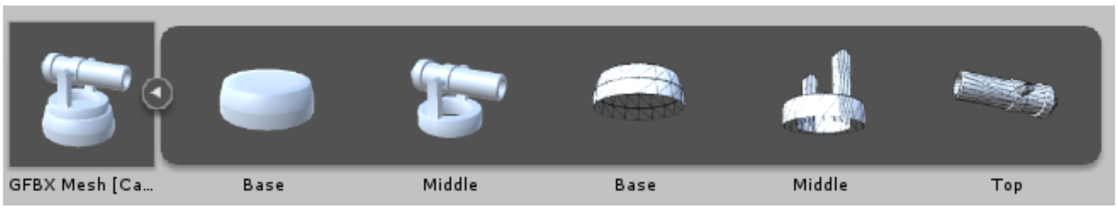

Figure 9. The components of the enemy cannon

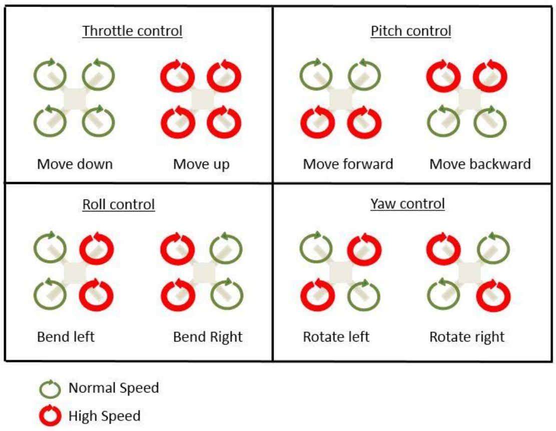

Figure 10. Motion of a quadcopter [15]

- real objects: the room,
- virtual objects: the drone (Figure 8), the cannon(Figure 9), the joystick buttons (Figure 11), and the planar meshes (Figure 7).

Implementation details

For the game implementation, we have chosen Unity which is a tool for creating games on different platforms, e.g. Windows, Linux, Android, iOS, Oculus Rift, etc. The graphics can be both 2D and 3D, and the coding is done with the help of C# scripts. For development, a large set of tools are available, from textures, mini-maps, terrain, shadows, special effects and particles, physics and user controls, all packed in a user-friendly interface.

For creating Android applications, a series of tools have been installed: Java Development Kit, Android Studio, and enabling developer mode on the smartphone.

In this game, the following interactions between object exist, with their corresponding logically modeled interactions: drone – room, AI enemy – drone, and player – game.

To create the drone, we first understood the basic physics it uses to move. We used the quadrotor as the basic shape, which has rotors that act as wings. Two adjacent rotors spin in opposite directions, while two opposite rotors spin in the same direction. By rotating at fast speeds, these generate thrust which pulls the air downwards and keeps the drone in the air. When the thrust cancels out the weight then the quadrotor hovers. There are three types of motion: roll – moving to the left or right, pitch – moving forward or backward, and yaw – rotation around the center. For example, to roll towards the right, the thrust is increased in the rotors on the left and decreased on the rotors on the right. To drive forward, the speed of the rear rotors is increased. These movements are illustrated in Figure 10. We have created a script that controls the movement of the drone for the following motions:

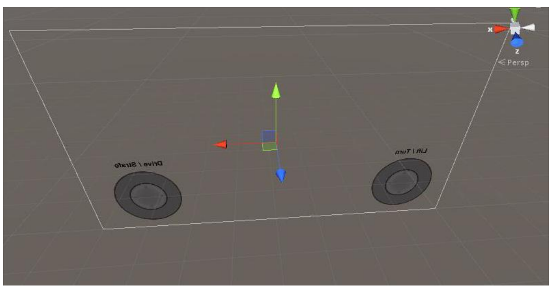

Figure 11. The 2D canvas containing the two joystick buttons

- hovering,
- lift up,
- lower down,
- forward and backward movement,
- left and right swerving,
- rotation around the z-axis.

To control the motion of the drone, we created the joystick buttons. The user inputs are handled in Unity by the Input Manager, where we set up axis and button inputs for each controller. We created the 2D canvas that can be seen in Figure 11, and have assigned the methods defined in the drone motion script to each button.

Figure 12. Drive forward action performed by the user, together with turret attack and collision of the cannonball with the drone

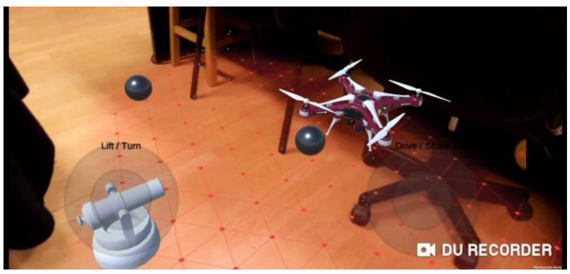

Figure 13. The virtual drone in the real environment. The planar surfaces have attached virtual meshes. The enemy cannon is placed on the floor and attacks the drone.

For the enemy cannon, a sphere was used as a cannonball, which directly collides with the drone, as both have attached materials that mimic real-life physics. The cannon will emit a warning sound before firing so that the player can know if the drone is close to the enemy, even when the turret is not seen on the image captured by the camera. The turret follows the movement of the drone using the Lerp function. The results can be seen in Figure 12, where the player controls the movement of the drone by applying the drive forward action. The enemy cannon senses that the UAV is close enough to it and performs the attack action.

The most important requirement for the application was the AR functionality – for this, we first considered using Vuforia. However, after a few tests, we have noticed that

Table 1. Results of the heuristic evaluation of the AR video game

Nb.	Heuristic	Question	SL	Observations
1.	Visibility of system status	Do you know what is going on during all of the interactions?	2	If the camera does not capture when the enemy turret appears in the scene, it may attack the drone and the user has a hard time adapting to the situation.
2.	User control and freedom	If the camera or sensors detect more than one plane in the scene, is it possible to identify which one?	0	Yes, all detected planes have different colors.
3.	Satisfaction	Does the application achieve the goal?	0	Yes, the main objectives are met.
4.	Aesthetic and minimalist design	Is the number of virtual objects in the scene appropriate?	0	There are enough objects.
5.	Aesthetic and minimalistic design	Is the number of interaction options satisfactory (marker, keyboard, mouse, joystick)?	2	The joystick buttons are small, yet the occupy too much of the screen (they can be made transparent).
6.	Help and documentation	Is the user guide satisfactory (video, text, audio)?	1	There is no straightforward information on placing the cannon on the game scene.
7.	Satisfaction	Are you satisfied with the interaction solution?	2	The movement speed of the drone is too high. The same is available for the speed and the force of the cannon ball.
8.	Satisfaction	Are you satisfied with the freedom to move around during interactions (e.g. you don't need to look directly at the camera constantly)?	0	The drone, turret and detected planes do not change their position even when they are not seen by the camera.
9.	Visibility of system status	Is the loading time of virtual objects in the scene satisfactory?	3	The problem appears when loading the enemy turret – it takes a long time to appear in the game scene.
10.	Match between system and the real world	Are the virtual objects merged correctly with the real world (position, texture, scale)?	1	The enemy cannon is too small.
11.	Match between system and the real world	Is the virtual object animation coherent with the real world?	0	The drone looks realistic.
12.	Consistency and standards	Are actions/feedback standardized?	2	There is no text feedback from the system.
13.	Recognition rather than recall	Is it easy to remember the application's functionalities (i.e. is it easy to memorize the functionalities of each object)?	0	There are no issues regarding the interaction with the objects from the scene – drone and buttons.
14.	Flexibility and efficiency of use	Is it easy to detect planes? How about colliding the drone with them?	2	Very rarely, the detected planes seem to float and therefore the collision is not intuitive.
15.	Environment configuration	Are there specific requirements (camera, marker, mobile, GPS, user position, lighting, print, calibration)?	1	The game only works on Android and needs only the camera.
16.	Accuracy	Is the tracker system stable?	0	The tracker is very stable.
17.	Accuracy	If the tracker system detects more than one object in the scene, does the application continue to function correctly?	0	All the detected planes are memorized. The virtual objects do not change their position in the real world.

the objects which are placed in the scene suffer random transformations at small luminosity changes in the environment. Therefore, we have switched to ARCore, which proved to be more stable in terms of detections and object placement. The AR SDKs enable the developer to use tools that detect planes, images, or objects. The environment is recorded using the camera from the computer or phone, and on the recorded images, virtual objects can be positioned, making it possible for the user to interact with them in the real world. Additionally, the motion tracking capability enables tracking the position of the phone in relation to the objects around it.

To detect the planar surfaces, we used the method provided by ARCore for planar surface detection. In it, we first detect feature points and then cluster them into a point cloud. We attach a collision mesh to the detected planes, to enable the interaction between them and the drone. The final look of the Drone AR application can be seen in Figure 13.

We have focused on creating a functional Augmented Reality application. The most important components are the drone and the enemies, which were successfully created. In a future implementation, the materials and their gathering operation will be implemented.

EVALUATION

Since game development is a user-driven process [7], an important aspect is evaluation. Traditional usability evaluation methods are able to discover some problems in AR applications. However, these types of application require specific testing stages with the same goal in mind: evaluating the usability of the video game.

In [8], the authors show an overview of methods that have been used to evaluate AR applications, while in [12] some guidelines that can be used to evaluate AR applications are explained. These are based on interviews, questionnaires and usability testing that aim to assess the strengths and weaknesses of the application. Usability is defined in ISO-09241-11 as the: "Extent to which the users of products are able to work effectively, efficiently and with satisfaction." Heuristic evaluation aims to define the current state of software in order to improve it, as noted in [24]. When applying such an evaluation to the iterative game development methodology, this can refine the final product by using the feedback obtained to correct the problems detected. Nielsen has defined such heuristics that were associated with specific objectives [14].

The components that we evaluate in order to assess the usability of an AR application are:

- goals – related to the aim of the video game,
- user – related to the ease of use, understanding of the application and satisfaction,
- tasks – the actions taken to achieve the goals,
- technology – AR applications need special devices and sensors,
- usability – how user-friendly is the interface.

A series of specific questions have been established for each of the 10 categories proposed by Nielsen, using the methodology presented in [12]. Additionally, we used Severity Levels to classify the type of problems that arise and decide their priority when we reiterate through the previous stages. These are:

- 0 – there is no problem,
- 1 – the issue must be solved if there is enough time,
- 2 – a minor problem,
- 3 – a major problem,
- 4 – the issue must be solved immediately.

In Table 1, we have synthesized the results of the evaluation. After this assessment, we have improved the application by changing the joystick buttons – we made them bigger and transparent, and we created a video showing to the users how to interact with the interface and how to place the cannon in the scene. Also, we have decreased the speed of the drone, so the player can have more control on its movements, and we decreased the attack speed of the turret so that the user can enjoy the game more.

An issue that is more difficult to correct is the loading time of the cannon. There is a big latency between the time the user taps the screen to place the object and the time it actually appears. This is conditioned by the AR development tool, which severely limits the future development of the game. If we desire to use markers in order to place materials and a train, we must consider a trade-off between the number of objects and interaction capabilities, and the loading time and response of the application.

CONCLUSIONS

In this paper we present the development of a Drone Augmented Reality video game. We started from an iterative development methodology and incorporated Augmented Reality elements. As future developments, we consider adding more elements to the game and improving the issues discovered in the evaluation stage.

REFERENCES

1. 6 VR and AR Statistics: Shaping the Future of Augmented Reality with Data, https://www.newgenapps.com/blog/6-vr-and-ar-statistics-shaping-the-future-of-augmented-reality-with-data, visited: 01.06.2019.
2. Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A. and Wagner, D., The history of mobile augmented reality, 2015.

3. Augmented reality shooting Drone n Base, https://www.youtube.com/watch?v=AfUDYzFRxKs, visited: 01.06.2019.
4. Billinghurst, M., Clark, A. and Lee, G., A survey of augmented reality. Foundations and Trends in Human–Computer Interaction, 2015.
5. Craig, C., Gerber, A., Learn Android Studio: Build Android Apps Quickly and Effectively, 2014.
6. Das, P., Zhu, M.O., McLaughlin, L., Bilgrami, Z. and Milanaik, R., Augmented reality video games: new possibilities and implications for children and adolescents. Multimodal Technologies and Interaction, 2017.
7. Duenser, A., and Bilinghurst, M., Evaluating Augmented Reality Systems, Handbook of Augmented Reality, 2011.
8. Duenser, A., Grasset, R., and Bilinghurst, M., A Survey of Evaluation Techniques Used in Augmented Reality Studies, 2008.
9. Eishita, F.Z. and Stanley, K.G., The impact on player experience in augmented reality outdoor games of different noise models. Entertainment Computing, 2018.
10. Encyclopedia Britannica definition for augmented reality, https://www.britannica.com/technology/augmented-reality, visited: 01.06.2019.
11. Gandolfi, E., Ferdig, R.E. and Immel, Z., Educational opportunities for augmented reality. Second Handbook of Information Technology in Primary and Secondary Education, 2018.
12. Guimaraes, M., and Martins, V. F., A Checklist to Evaluate Augmented Reality Applications, Symposium on Virtual and Augmented Reality, 2014.
13. Hands on: Merge 6DoF Blaster review, https://www.techradar.com/reviews/merge-6dof-blaster, visited: 01.06.2019.
14. J. Nielsen, Usability Engineering, 1993.
15. Khuwaja, K, & Chowdhry, B. S., Khuwaja, K. F., Mihalca, V. O., and Tarca, R, Virtual Reality Based Visualization and Training of a Quadcopter by using RC Remote Control Transmitter. IOP Conference Series: Materials Science and Engineering, 2018.
16. Kim, K., Hwang, J., Zo, H. and Lee, H., Understanding users' continuance intention toward smartphone augmented reality applications. Information Development, 2016.
17. LeBlanc, A.G. and Chaput, J.P., Pokémon Go: A game changer for the physical inactivity crisis?. Preventive medicine, 2017.
18. Lee, J.H., Hall, M.G. and Keating, S., Challenges in preserving augmented reality games: a case study of Ingress and Pokémon GO, 2017.
19. Lego Playgrounds uses ARKit to bring your sets to life on iOS devices, https://www.pocket-lint.com/ar-vr/news/lego/146483-lego-playgrounds-uses-arkit-to-bring-your-sets-to-life-on-ios-devices, visited: 01.06.2019.
20. Linowes, J. and Babilinski, K., Augmented Reality for Developers: Build Practical Augmented Reality Applications with Unity, ARCore, ARKit, and Vuforia, 2017.
21. Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö. and Chryssolouris, G., Augmented reality (AR) applications for supporting human-robot interactive cooperation, Procedia CIRP, 2016.
22. Okita, A., Learning C# Programming with Unity 3D, 2014.
23. Onstott, S., Adobe Photoshop CS6 Essentials, 2012
24. Pausch, R., Gold, R., Skelly, T., and Thiel, D., What HCI designers can learn from video game designers, Proceedings of Conference Companion on Human Factors in Computing Systems, 1994.
25. Physics and Augmented Reality - Part 1, https://www.youtube.com/watch?v=enXTKvhE7yk&t=1s, visited: 01.06.2019.
26. Rampolla, J., Kipper, G., Augmented Reality: An Emerging Technologies Guide to AR, 2012
27. Ruano, S., Cuevas, C., Gallego, G. and García, N., Augmented Reality Tool for the Situational Awareness Improvement of UAV Operators. Sensors, 2017.
28. Silvennoinen, M., Displaying Augmented Reality content on HoloLens environment, 2017.
29. Soares, L.P., Musse, S.R., Pinho, M.S. and Boussu, J.B., Evaluation of Selection Techniques on a Mobile Augmented Reality Game. In 2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 2018.
30. Star Wars: Jedi Challenges, https://www.lenovo.com/gb/en/jedichallenges/, visited: 01.06.2019.
31. Three Augmented and Virtual Reality Apps for Design and Construction, https://www.architectmagazine.com/technology/products/three-augmented-and-virtual-reality-apps-for-design-and-construction_o, visited: 01.06.2019.
32. Unity ARCore: Quickstart for Android, https://developers.google.com/ar/develop/unity/quickstart-android, visited: 01.06.2019.
33. Wang, H., Qin, J. and Zhang, F., A new interaction method for augmented reality based on ARToolKit. In 2015 8th International Congress on Image and Signal Processing (CISP), 2015.
34. Wang, W., Understanding Augmented Reality and ARKit. In Beginning ARKit for iPhone and iPad, 2018.
35. Wargaming shows off World of Tanks spectating in augmented reality, https://venturebeat.com/2018/10/27/wargaming-shows-off-world-of-tanks-spectating-in-augmented-reality/, visited: 01.06.2019.
36. Wartmann, C., The Blender Book: Free 3d Graphics Software for the Web and Video, 2000.