International Journal of User-System Interaction 12(3) 2019, 189-206

Designing a Miniature Unmaned Aerial Vehicle Augmented Reality Application

Bianca-Cerasela-Zelia Blaga, Dorian Gorgan

Technical University of Cluj-Napoca
Computer Science Department
Cluj-Napoca, Romania
E-mail: zelia.blaga@cs.utcluj.ro, dorian.gorgan@cs.utcluj.ro

Abstract. Augmented reality video games combine the elements of a virtual application with that 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 the 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 stage.

Keywords: augmented reality, heuristic evaluation, video game design and implementation.

1. Introduction

Augmented reality (AR) is a field of computer science that deals with the combination between the real and virtual worlds as explained by Billinghurst et al. (2015). It was situated at the top of the Gartner technology hype cycle because there is a growing interest for this field, and because it can bring improvements in various areas, like gaming, medicine, teaching, travel, marketing, and research like Kim et al. (2016) describe. 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.

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. Michalos et al.

(2016) believe that with the increased popularity of AR glasses, 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 user, in which fantastic creatures exist, and the goal is to solve the quests. The player uses the GPS on the smartphone to move around the world and performs specific actions that bring him closer to achieving the goals. Some examples of successful AR games are Pokemon GO – LeBlanc& Chaput (2017), Ingress – Lee et al. (2017), Temple Treasure Hunt – Eishita & Stanley (2018), and Zombie GO – Soares et al. (2018).

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 as stated by Ruano et al. (2017). 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 level of excitement increases compared to a simple virtual game.

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

2. Related work

Encyclopaedia Britannica 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 as presented by Silvennoinen (2017). In 1990, the term "Augmented Reality" was coined by Tom Caudell as described by Arth et al. (2015), a Boeing researcher, followed by the first AR theater production by Julie Martin that featured performances around virtual objects as explained by Gandolfi et al. (2018), in 1994.

An open source development toolkit named ARToolKit was introduced in 2009 by Hirokazu Kato and presented in detail by Wang & Zhang (2015), which was succeeded by Apple's ARKit – Wang (2018) and Google's ARCore – Linowes & Babilinski (2017) – for the creation of iOS and Android applications. Three main industries that can benefit from AR technology stand out eCommerce, marketing, and gaming.

The appeal of 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 in bigger and the enjoyment is higher. For example, Disney and Lenovo teamed up to create the "Star Wars: Jedi Challenges" (2018) 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 has build an AR application using Apple's ARKit. 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, presented by Arnhem (2018), is Merge's 6DoF Blaster that uses a smartphone to create digital targets that the users can shoot at. Even though at first glance this device is a plastic gun which 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. A more complex application is presented by Pinzariu & Iftene (2016) in the form of a multiplayer game where players control a Sphero robotic ball to complete tasks like collecting diamonds, teleportation or finding a treasure.

Studies have shown that these types of games 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, as presented by Das et al. (2017). Chițaniuc & Iftene (2018) created an AR application with the aim of helping students in secondary school learn the geography of Europe. Păduraru & Iftene (2017) present an implementation of an AR tower defense video game which uses elements like keyboard and mouse interaction together with cardboard drawings for the game map, with the end goal of teaching students the A* search algorithm.

The Augmented Reality domain has some limitations that researchers strive to surpass. Work is carried oout for creating a flawless virtual world in which orientation and direction adjust synchronously with no errors or in which the player's movements are mapped in real time. We aim to present the available technology with a hands-on experiment and present some of the available tools for developers.

3. AR game development

To create an AR digital game, Rampolla & Kipper (2012) consider that researchers use the following components for the hardware:

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

As for the software used, this consists of:

- an application or program running locally,
- Web services,
- 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.

4. Analysis

In this section, the main ideas and goals for the application are established, together with the video game tasks, an initial prototyping, the 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 a cyclic work process.

4.1. Topic selection

The proposed game started 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, and can see through the camera 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, simulating a real-life interaction with a miniature UAV (Unmaned Aerial Vehicle). To make the game more difficult, enemies that are controlled by Artificial Intelligence (AI) are inserted in the scene. They will try to disrupt the gamer by attacking the drone when it gets close to them. Therefore the player has to efficiently navigate the environment, successfully avoiding the collision with objects from the scene and steering clear from enemies.

4.2. 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 enemies controlled by AI which attack the drone or destroy 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 be precise. The enemies will have a limited field of view and will attack only when the UAV is at a certain distance from them.

There are three options for the building in which materials are placed. The

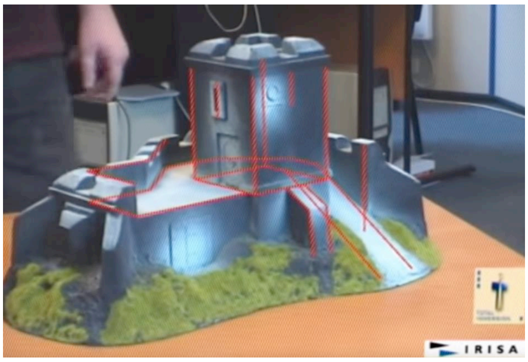

Figure 1. Example of wireframe for a building – Marin (2006)

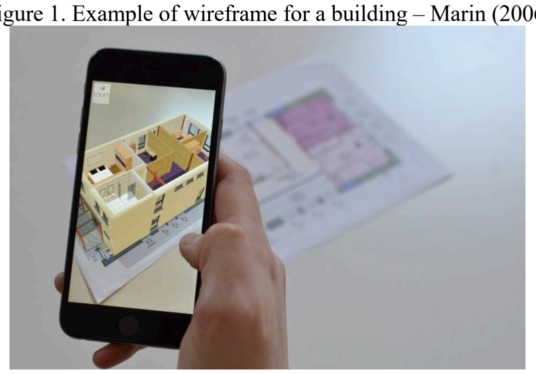

Figure 2. 3D model of building with the 2D image – Busta (2015)

first one, as seen in Figure 1, 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 2, which will display a 3D building when looking through the phone's camera. The third option would be to have the whole room as 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.

4.3. Prototyping

For the beginning, the player will have a simple user interface to control the drone, which will look like in Figure 3, together with a game menu seen in Figure 4. 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. Since the option of using the whole room as 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 5.

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,

Figure 3. Mobile prototype; the scene is composed of the room (captured using the camera), while the drone is a virtual object.

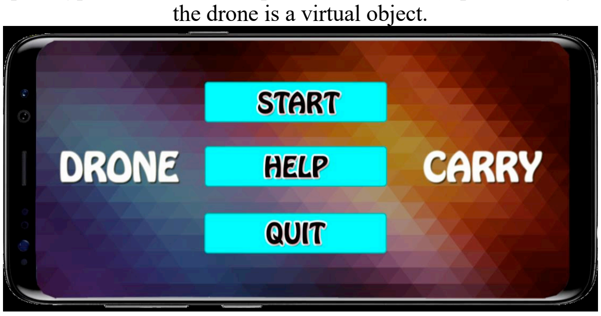

Figure 4. Game menu prototype.

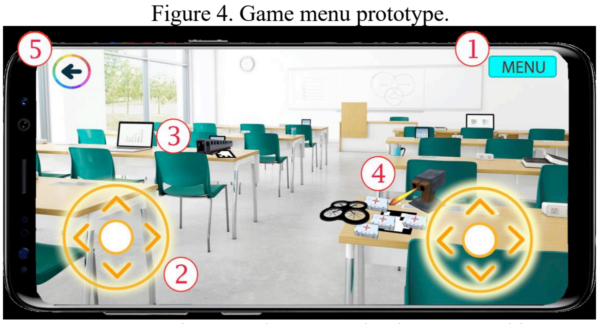

Figure 5. In-game prototype – 1. the menu button, 2. the drone control buttons, 3. marker with virtual container, 4. marker for materials and AI enemy attacking the drone, 5. back button.

the idea of using markers was put aside, and the option of tapping a surface to place an object has been chosen.

4.4. Scenarios and task description

We can divide the objectives into scenarios – how the user interacts with the system, and tasks – the actions of the user or the system. The following scenarios, together with their tasks, for the 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 control 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 cannon ball.

In the next section, we present the implementation steps in order to avhieve all the desired tasks. Focus will be placed on developing an enhjoyable real-time application to resemble an actual case scenario of controlling a drone.

5. Implementation

In the following sections, we present the most important steps we took to create the video game and how we incorporated Augmented Reality technology into it. The most important components to consider are the tools needed to build the application, together with the 3D virtual objects that are present in the game, and the AR technology which has to be realistic and to work seamlessly.

The user interface is a very important component since the player interacts with it all throughout the usage of the application. This has to be simple, yet effective in order to display with a minimum number of elements a large amount of representative information. We revisit this implementation step after performing intermediate evaluations, which tell us what issues arise, how the users feel about the game and what problems hinder the usability. We continuously address these, make the necessary adjustments and change elements accordingly.

5.2. Game objects

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

- real objects: the room,
- virtual objects: the cannon (Figure 6), the joystick buttons, the planar meshes (Figure 7), and the drone (Figure 8),.

5.3. Implementation details

For the game implementation, we have chosen Unity – Okita (2014) – 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

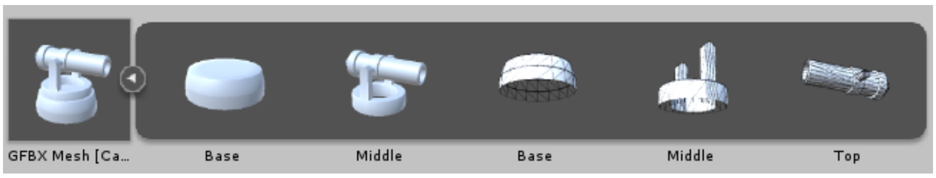

Figure 6. The components of the enemy cannon.

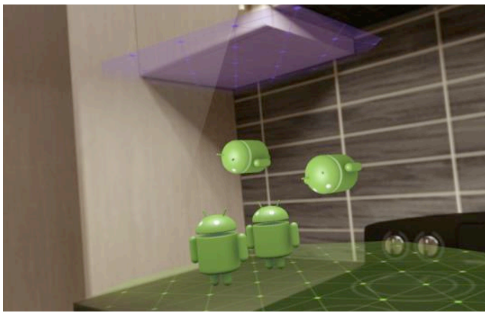

Figure 7. Real environment with virtual planar meshes and virtual objects – Unity (2019).

Figure 8. 3D virtual object representing a drone. © ProfessionalAssets

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 – Craig & Gerber (2014), and enabling developer mode on the smartphone. For 3D modeling we used Blender, Warmann (2000), and for prototyping and textures – Adobe Photoshop CS6, Onstott (2012). 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, and for game testing, a Samsung S8 smartphone with Android version 8. In this game, the following interactions between objects exist: 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 basic shape, which has rotors that act as wings. Two adjacent rotos spin in opposite directions, while two opposite rotos 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. We have created a script that controls the movement of the drone for the following motions:

- 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 contains the user interface elements, and have assigned the methods defined in the drone motion script to each button.

For the enemy cannon, a sphere was used as cannon ball, 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 9, 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 the objects which are placed in the scene suffer random transformations at small luminosity changes in the environment. Therefore, we have switched to ARCore – Linowes et Babilinski (2017), 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

Figure 9. Drive forward action performed by the user, together with turret attack and collision of the cannon ball with the drone.

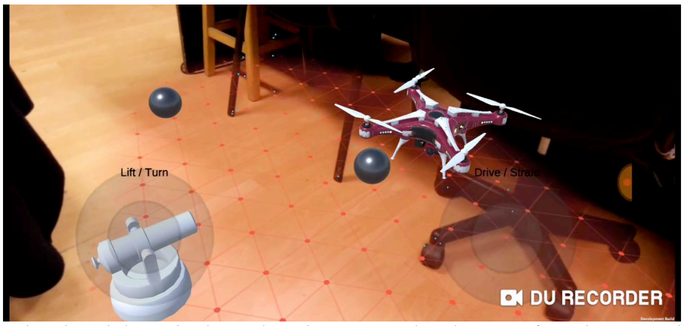

Figure 10. The virtual drone in the real environment. The planar surfaces have attached virtual meshes. The enemy cannon is placed on the floor and attacks the drone.

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 tracing the position of the phone in relation to the surrounding objects.

To detect horizontal and vertical object planes, we used the method provided by ARCore for planar surface detection. Using it, we first detect feature points and then cluster them into a point cloud. We attach a collision mesh to the detected planes, which enables the collision between them and the drone. The final look of the Drone AR game can be seen in Figure 10.

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.

6. Evaluation

Since game development is a user-driven process as explained by Duenser & Billinghurst (2011), 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 evaluation stages that have the same goal in mind: evaluating the usability of the video game.

In a paper written by Duenser et al. (2008), the authors show an overview of methods that have been used to evaluate AR applications, while in 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 evaluations aim to define the current state of software in order to improve it, as noted by Pausch et al. (1994). When applying such an evaluation to the iterative game development methodology, it can refine the final product by using the obtained feedback to correct the detected problems. Nielsen (1993) has defined such heuristics that were associated with specific objectives.

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 by Guimaraes & Martins (1993). Additionally, we used Severity Levels to assign the problem 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 – minor problem,

3 – major problem,

4 – the issue must be solved immediately.

In Table 1 and Table 2, we have synthesized the results of the evaluation. A number of four evaluators in the age group of 20 to 30 years old, with experience in video game development, answered a total of 17 representative questions about the game, during and after playing it. In the final tables, we have recorded the mean SL value, as well as the most important observations. Based on these, the game developer can make changes to obtain a higher usability level of the AR video game.

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.

Table 1. Summary of answers for questions 1 – 7, given by the game evaluators.

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.

Table 2. Summary of answers for questions 8 - 17, given by the game evaluators.

Nb.	Heuristic	Question	SL	Observations
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.

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 and response time of the application.

7. Conclusions

In this paper, we present the development of an Augmented Reality video game centered around a drone. We started from an iterative development methodology and incorporated Augmented Reality elements. We present all the necessary steps for creating a video game from scratch. We chose the topic of drone control, established the desired use case scenarios with their respective tasks, and created a protoype for the application. Then we implemented the AR video game using a variety of tools, from software to hardware components, from 2D objects to 3D ones, and incorporated the AR technology with the help of ARCore. By continuous evaluation, we identified problems and worked towards solving them. In the end, we obtained a highly usable video game, which presents an enjoyable activity to its users. As future developments, we consider adding more elements like markers and other interaction techniques, and improving the issues discovered in the evalution stage related to latency. One interesting extention would be the development and analysis of the game strategies by using visual techniques, such as presented by Catana & Gorgan (2015). the optimal strategic solution is sought by exploring massive multidimensional data.

References

Arnhem, J. P. J. V. (2018) *Mobile Apps and Gear for Libraries: Merge Cube's Handiness with Holograms Makes it a Good Place to Start with Augmented Realit*y. The Charleston Advisor, 20(1), pp.56-58.

Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A. and Wagner, D., (2015) *The history of mobile augmented reality*, arXiv preprint arXiv:1505.01319.

Billinghurst, M., Clark, A. and Lee, G., (2015) A survey of augmented reality. *Foundations and Trends in Human–Computer Interaction, 8(2-3)*, pp.73-272.

Busta, H., (2015) Three Augmented and Virtual Reality Apps for Design and Construction, *Architect Magazine*.

Catana M.C, Gorgan D., (2015) Analyzing Computer Game Strategies through Visual Techniques. *Proceedings of the 12th Romanian Conference on Human-Computer Interaction*, ISSN 2344-1690, pp.41-48.

Chițaniuc, M., Iftene, A., (2018) GeoAR-An Augmented Reality Application to Learn Geography. In *Romanian Journal of Human-Computer Interaction, vol. 11, issue 2*, pp. 93-108.

Craig, C., Gerber, A., *Learn Android Studio: Build Android Apps Quickly and Effectively*, Apress, 2014.

Das, P., Zhu, M.O., McLaughlin, L., Bilgrami, Z. and Milanaik, R., (2017) Augmented reality video games: new possibilities and implications for children and adolescents. *Multimodal Technologies and Interaction, 1(2)*, p.8.

Duenser, A., and Bilinghurst, M., Evaluating Augmented Reality Systems, *Handbook of Augmented Reality*, 2011.

Duenser, A., Grasset, R., and Bilinghurst, M., (2018) *A Survey of Evaluation Techniques Used in Augmented Reality Studies*, Human Interface Technology Laboratory New Zealand.

Eishita, F.Z. and Stanley, K.G., (2018) The impact on player experience in augmented reality outdoor games of different noise models. *Entertainment Computing 27*, pp.137-149.

Marin, E., *Physics and Augmented Reality*, YouTube, 2006.

Gandolfi, E., Ferdig, R.E. and Immel, Z., (2018) *Educational opportunities for augmented reality*. Second Handbook of Information Technology in Primary and Secondary Education.

Guimaraes, M., and Martins, V. F., (2014) A Checklist to Evaluate Augmented Reality Applications, *Symposium on Virtual and Augmented Reality*, pp. 1-13.

Nielsen, J., *Usability Engineering*, Elsevier, 1993.

Kim, K., Hwang, J., Zo, H. and Lee, H., (2016) Understanding users' continuance intention toward smartphone augmented reality applications. *Information Development, 32(2)*, pp.161-174.

LeBlanc, A.G. and Chaput, J.P., (2017) Pokémon Go: A game changer for the physical inactivity crisis?. *Preventive medicine, 101*, pp.235-237.

Lee, J.H., Hall, M.G. and Keating, S., (2017) *Challenges in preserving augmented reality games: a case study of Ingress and Pokémon GO*.

Linowes, J. and Babilinski, K., *Augmented Reality for Developers: Build Practical Augmented Reality Applications with Unity, ARCore, ARKit, and Vuforia*, 2017.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö. and Chryssolouris, G., (2016) Augmented reality (AR) applications for supporting human-robot interactive cooperation, *Procedia CIRP*.

Okita, A., *Learning C# Programming with Unity 3D*, AK Peters/CRC Press, 2014.

Onstott, S., *Adobe Photoshop CS6 Essentials*, John Wiley & Sons, 2012.

Pausch, R., Gold, R., Skelly, T., and Thiel, D., (1994) What HCI designers can learn from video game designers, *Proceedings of Conference Companion on Human Factors in Computing Systems*, pp. 177-178.

Păduraru, B. M., Iftene, A., (2017) Tower Defense with Augmented Reality. In *Proceedings of the 14th Conference on Human Computer Interaction - RoCHI 2017*, ISSN 2501-9422, ISSN-L 2501-9422, pp. 113-118.

Pinzariu, M.N., Iftene, A., (2016) Sphero - Multiplayer Augmented Game (SMAUG). In *Proceedings of the 13th International Conference on Human-Computer Interaction RoCHI'2016*, Pp. 46-49.

Rampolla, J., Kipper, G., *Augmented Reality: An Emerging Technologies Guide to AR*, Elsevier, 2012.

Ruano, S., Cuevas, C., Gallego, G. and García, N., (2017) Augmented Reality Tool for the Situational Awareness Improvement of UAV Operators. *Sensors, 17(2),* p.297.

Silvennoinen, M., *Displaying Augmented Reality content on HoloLens environment*, 2017.

Soares, L.P., Musse, S.R., Pinho, M.S. and Boussu, J.B., (2018) *Evaluation of Selection Techniques on a Mobile Augmented Reality Game*. In 2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames).

Lenovo, (2018) *Star Wars: Jedi Challenges*, last visited: 18.12.2019, online: https://www.lenovo.com/gb/en/jedichallenges/,

Unity ARCore, (2019) *Quickstart for Android*, last visited: 18.12.2019, online: https://developers.google.com/ar/develop/unity/quickstart-android,

Wang, H., Qin, J. and Zhang, F., (2015) *A new interaction method for augmented reality based on ARToolKit*. In 2015 8th International Congress on Image and Signal Processing (CISP), pp. 578-583, IEEE.

Wang, W., (2018) Understanding Augmented Reality and ARKit. In *Beginning ARKit for iPhone and iPad*, pp. 1-17. Apress, Berkeley, CA.

Wartmann, C., *The Blender Book: Free 3d Graphics Software for the Web and Video*, No Starch Press, 2000.