A TOOLSET FOR PHYSICAL INTERACTION IN AUGMENTED REALITY ENVIRONMENTS

by

ILIAS LOGOTHETIS

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2021

Approved by:

Dr. Papadourakis George

Abstract

Augmented reality nowadays is constantly expanding, and its applications become widely known. This technology combines the real world with the virtual, allowing virtual content to be placed on top of the real world in real time, enhancing or even altering the real world. As this technology evolves, new methods of interaction are currently researched to provide a better and more organic behavior in such environments. One of these methods is the ability to track and identify the pose of the hands of the user in the environment. Using such techniques allows to simulate the hand behavior which describes the interaction with real objects into the virtual world. As the research on these technologies progresses, tools are required to provide the means for building applications taking advantage of such technologies. Most of these of tools are targeted towards a specific device or method and therefore cannot work if those are changed. Another limitation of these tools is that it is hard to expand their functionality if it is required. In this master thesis a toolset is developed to enable the interaction between the physical hand and virtual objects in augmented reality environments using the Unity game engine. This toolset provides basic functionalities of interactions and it is made in such a way that allows for the developers to expand the capabilities and the range of interactions as they see fit. Furthermore, the toolset has the ability to adapt to different technologies for hand tracking, making it more flexible on what methods will be used each time. It is also important to mention that given the technology is always moving forward, the ability to continuous integrate the latest methods into this toolset is a significant feature. The resulting toolset enables the development of applications using the aforementioned technologies with a few basic steps.

Acknowledgements

I would like to thank my supervisor Prof. George Papadourakis for his support and guidance all these years in the ISCA lab, the knowledge he imparted and the trust he put in me for carrying out this thesis. A tool can be considered helpful only if it is used. For this reason, I would like to thank Prof. Nikolas Vidakis for the opportunity for collaboration he provided, making the creation of an application using the tool of this thesis possible. I would also like to thank my family for their continuous support throughout my studies. Finally, special thanks to Apostolis, Alexis, Michalis, Paris and Elpida for putting up with me, and for their valuable input on this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

Augmented reality is a technology vastly growing nowadays. It is a fascinating technology as it allows for such possibilities that years before people believed was only applied on science fiction movies. Some of those possibilities are the ability to display information on top of real-world objects, or 3D representations of objects with the ability to be processed in real-time [1]–[4]. The appliance on mobile phones has helped on the growth of this technology as many applications were introduced to users and new possibilities for companies to get advantage on.

The advantage that the mobile phones have on augmented reality applications are that they are not bound on external hardware and are a device that every person owns. The only requirement from a mobile phone to be able to run augmented reality applications are a camera. There can be a depth camera to provide the maximum capabilities that augmented reality can offer, or a simple camera that can also display limited information on screen in a similar way.

Not limited on mobile phones, augmented reality is a technology available in a variety of devices such as computers using a web camera, head mounted devices and more. Those devices have their own advancements in augmented reality as well. An example of such advancements constitutes the Microsoft HoloLens [5] that is a head mounted device, that allows the users to augment their environment with variety of virtual components as to control the light switches in the house or display the temperature.

Although the technology in augmented reality is improving rapidly, the focus leans towards advancements on technologies to track the environment, a better understanding of depth and light factors. Focus as well has been put on the display of such environments on the devices screen in addition on the way that the augmentations such as virtual objects will be displayed on top of the captured image, and how those objects will affect the displayed environment and vice versa. With the focus on display and tracking the interaction in such environments is neglected.

The interaction between user and virtual objects is a complicated topic leaving this task on the usual ways of interaction between human and a digital device (touchscreen, mouse, keyboard, controllers, etc.) that also are studied to provide a better experience[6]–[8]. There is a need to provide new ways of interaction in augmented and virtual reality that will allow a more natural interaction between the user and the environment. Such ways of interaction could be by using the user's hand as on the physical environment. Also, external wearables can be invoked such as a glove that can contain sensors and actuators that will allow for haptic feedback and provide a complete experience to the user.

Research has been made for better ways of interaction in augmented reality environments allowing the users to interact using their hands [9], [10]. This way of interaction is the best for the user as it is the most natural way of interaction that humans are used to, interacting with objects with their hands. Hands is a complicated mechanism that allows for many flexible and precise movements, that renders it difficult to be adapted and transferred on a virtual scene. There are many studies that tries to precisely track the hand and extract the pose of it and on the later years the results are extremely positive.

Still, despite the accurate results there are limitations on the usage of these techniques, as most of them are computational heavy or require the invocation of special sensors. Mobile devices do not possess that powerful hardware to handle the processing of such techniques in real-time effectively at present time. In the near future the hardware on mobile devices has the potential to reach the computational power required to run such techniques in real-time, so in the next few years the execution of such techniques in real-time should be a standard.

Until the upgrade of hardware on devices, other methods have been invoked for the hand tracking task. One famous method is to build a web service that will be hosted on a powerful enough machine to handle the computational heavy processes. In this way mobile devices would be able to connect to this service and use it to process the data required and send back the results to be utilized. The drawback of such methods is the requirement of an internet connection and in most cases a good one, as the streaming of video is not a light process for the internet connection.

1.1 Augmented reality applications

Augmented reality has appliances to a variety of fields as it has present potential to solve some big problems and facilitate many functionalities. Next, will be described some of the fields that augmented reality is applied and has contribute on the evolution of each field [11].

- **Medicine & Healthcare**

  The possibilities on medical applications are many. It has been suggested from universities that the students learn topics such as anatomy using augmented reality headsets so that they will be able to interact with a 3D model of the human body. It will allow doctors to study the patients in a holographic view acquiring data from CT, MRI and ultrasound [11], [12]. One more use could be the assistance on doctors with providing all the needed data when performing a surgery so that doctors will not have to turn around to access those data from other machines. An extra functionality of augmented reality systems is the ability to integrate haptic devices so that will allow doctors and surgeons to have feedback such as feeling a tumor without the need to perform an open surgery [13].

- **Retail & Marketing**

  In the field of marketing augmented reality has possibly the most uses from every other field. That is because companies have found a way to be presenting their products in 3D and in real-world and all that from the display screen of a device. This allows customers to have a precise understanding of how the product looks in real-world. Augmented reality also provides the opportunity for the customers to "try on" the product they are interested in before they purchase it [14]. Many companies have adapted this model of advertisement and promotion of their products as customers have started to ask to use augmented reality when they shop. This is observed mainly on the furniture market, that the customers want to check how the furniture looks in their place.

- **Construction & Maintenance**

  On construction and maintenance field augmented reality has to offer also great visualization techniques allowing plans and facilities to be visualized in a real-world place [15]. It also allows for representations of underground constructions and help with the planning and further development. Another great way that augmented reality is used in this field is using glasses or some other device that provide assistance on workers on repairing a damage or on maintenance of a system, providing extra information about the system such as where the damage is, what procedure should be followed for a successful result etc. Designers can easily have their models on real world and be able to have it next to their workstation machines so every design change can be visualized in 3D and not on a computer screen. This ability to have always a model representing the final product to inspect is a great asset on the creative process.

- **Travel & Navigation**

  The ability to navigate on a different place in the world using an application is mind blowing. Travel agencies taking advantage on augmented reality to present places from locations around the world to their customers before they book a ticket to visit the actual destination. Except of that, when traveling augmented reality can provide nice interfaces that will be used as tourist guides providing information about the visiting places [16]. This can be achieved using the GPS of the smart phone that it constitutes a basic device of daily use. Augmented reality applications can be used on cars to provide additional information about the real-world locations such as nearby gas stations guide directions for a place and more. Augmented reality is used also on route-optimization showing for example the shortest route to locate and reach items stored on a warehouse.

- **Entertainment**

Many of the classic board games have started to be developed on augmented reality such as Monopoly. Books also that incorporate animated images and sound can create a deeper bond between the reader and the characters in the book [17]. Video games also making substantial progress using augmented reality creating content that can interact with real-world objects. Also, through mobile devices it is possible to control drones which are equipped with a camera, that enables augmented reality games to be created like avoiding obstacles appearing on the screen etc.

- **Education**

On education augmented reality promises many of beneficial activities and functionalities for teaching [18], [19]. It can offer the ability to simulate real-world environments and situations that it would be hard for students to experience otherwise [20], that alone consist an extra motivation for students to explore [21]. The collaboration between students and instructors can be increased [22] as well as the imagination and creativity of students for the subject [23]. Studies have been conducted on many fields that uses AR applications related to education. Such fields are on medical applications [24], on mechanical engineering [25], architecture [26] and more [27][28].

A great application on education is the augmented reality Books that allows for interactive experiences with the students helping to make the subject of study more appealing. Another way to apply augmented reality on education is through games as a large of educators using video games to assist on the teaching of a subject. This way makes easier for students to understand the concepts of the class as allows teachers to better show and explain the subject. Using mobile devices enables the interactions with virtual objects placed on real world on specific locations that players will be able to interact when they are within reach.

Other ways to utilize augmented reality is by making virtual scavenger hunts that encourage students to search and find clues on an area assigned by the educator. This application is great for educators as they can prepare content for a place that a field trip will take place, in this way student education will become more interactive and interesting. Augmented reality applications could become a basic tool for students to build skills on a topic, as many scenarios can be simulated on an augmented environment that will give the opportunity to the learner to practice skills on a topic, providing on the same time instructions on how to perform the tasks in the right way. The ability to simulate tasks and guide through solution even if haptic feedback will not be possible is a great opportunity to enhance muscle memory for specific tasks. Such skill building techniques could be of great use on mechanics, engineers, architects.

1.2 Objective

Many frameworks that are addressed in augmented reality does not provide a means for interaction in such environments or focuses only on the interaction. In both cases, it is hard to expand the functionality of such frameworks if needed, having to wait for an update. Also, it is commonly hard to use a set of the functionalities provided by a framework combined with a different one, and in this case, it will result on almost writing a completely new framework.

In this thesis a toolset developed for the Unity game engine will be presented, that supports physical hand interaction with virtual objects in an augmented reality environment. This toolset aims to facilitate the production of augmented reality applications with physical hand interaction providing tools for the developers to use for this purpose. Developers can also expand the functionality of those tools in a clean and easy way, creating their own behavior of a tool by extending its basic interface. The hand tracking functionality is implemented as a different component of the toolset as it would had limited the options for the hand tracking techniques and the ability of the user to choose between those techniques.

This toolset has been created with the concept to provide basic tools, that can be used together to create an augmented reality application with basic interaction fast. Also, the toolset will be open for expansion and able to be used in single units with other available tools and frameworks. The provided tools of this toolset for Unity game engine are the following: communication with the external part for the hand tracking, hand visualization on the augmented environment, ways to retrieve each finger or part of the finger of the recognized hand. It also offers a number of predefined gestures such as grab, point and more. Also, there is an implementation of actions that describes what the virtual objects will do upon interaction. Actions mostly serve as a guide for the developers to code their own actions that will correspond to the needs of their application. The interaction between the hand and the virtual objects is performed by ray casting, as it is a lightweight method to identify intersecting objects, so it does not affect performance, while it allows to reach the goals that are primarily this toolset is focused.

The hand tracking part of the toolset acts as a standalone application that can be installed on a mobile device or run as a web service. It is written in python as most of the hand tracking open-source projects use python as their programming language. It provides a module that receives the images from the mobile device, then the images are processed and then it passes through the hand tracking algorithms. Those algorithms constitute a separate module that can be changed at any time. The reason for this is to offer the opportunity for experimentation with new algorithms in a way that will not break the application, or the necessity for the development of a new one will occur.

1.3 Outline

The thesis is structured as follows: In Chapter 2 the background of the literature for the corresponding technologies is outlined. In Chapter 3 an overview of state of the art of hand interaction in augmented reality is reported. In Chapter 4 the design and the main goals of the toolset is described. In Chapter 5 the implementation of the toolset is explained. In Chapter 6 the results of this thesis are addressed. In Chapter7 conclusions, the final thoughts and directions for future work are reflected.

Chapter 2 - Literature review

2.1 Augmented Reality

Augmented reality can be defined as a technology that enhances the real world with virtual information [29]. This information can be images, sounds, 3D virtual objects and more. An Augmented reality system as stated [30] should provide the following characteristics: 1) combine real and virtual imagery, 2) be interactive in real-time, 3) register the virtual imagery with the real world in 3 dimensions. The above statements are extracted from previous works [1]–[4]. Augmented reality has mostly adapted for visual augmentations, but it is not limited to the vision sense. Augmented reality can be applied to other senses as well, like hearing or touching [31].

A key feature of Augmented reality is the ability of feature tracking in the environment. This feature is important because based on this tracking ability of the environment, virtual objects can be placed in the real world [29]. Tracking has been separated into three categories [32]–[34] as listed ahead:

- **Sensor-based tracking**

  This approach is based on the usage of sensors for tracking and localization. The core advantage of these techniques is that offer high processing speed, but they require plenteous calibration and matching. The types of sensors commonly used are the following:

  - *Optical tracking*

    It is a technique that utilizes at least two cameras placed to an environment with known dimensions and can track many objects simultaneously [35].

  - *Magnetic tracking*

    Magnetic tracking is used in augmented reality by incorporating a transmitter and a receiver. The transmitter acts as the origin point and the receiver is placed to the viewer. A coordinate system can be defined for the points of the transmitter and the receiver, to assist on the extraction of the position and orientation of the device [36].

- *Acoustic tracking*

  The acoustic method acts similarly as the magnetic, in a sense that requires a transmitter that is the source of a sound and the receivers that are the entities the sounds reach to. The time that the sound takes to reach those receivers is measured to extract information about the position [36].

- *Inertial tracking*

  This type of tracking utilizes an accelerometer to measure the acceleration and a gyroscope to measure the angular velocity [36].

- **Vision-based**

  Vision based techniques utilizes optical sensors to extract the position. These optical sensors can be either infrared sensors, visible light sensors or 3D structure sensors [29]. The two main categories of vision-based tracking are the marker-based tracking and the markerless tracking. The first was used in earlier days but recently the markerless techniques are becoming more popular. Both methods have a similar workflow that is to capture an image of the real-world using a device with camera, then it processes this image to identify feature points or markers that helps in the understanding of the environment. After this step a virtual object can be placed in the points detected, finally the image is displayed on the device as background and the virtual objects are displayed on top.

- *Marker-based tracking*

  Marker-based tracking is based on markers in an environment that has to be recognized through certain algorithms and possibly match it with a product. These markers act as reference points on the world to allow for display desired information on screen. Marker-based approaches offers high position accuracy but has shown low stability in terms of camera movement that might hide the marker [37].

- *Markerless tracking*

  Markerless tracking is achieved by tracking features without artificial markers as on marker-based techniques [38]. Markerless approaches uses algorithms for feature detection and representation to acquire the understanding of the scene. Such algorithms are SIFT [39], SURF [40], FAST [41], ORB [42] and more [29]. Another technique is the simultaneous localization and mapping (SLAM) [43], that constructs an understanding of the environment by tracking the location of recognized features as the device moves in this environment, this is extremely effective on small areas. By detecting and recognizing features on the environment markerless approaches has the advantage of the ability to extract information as lighting changes, overlapping objects, rotation and other useful information about the environment. The computational power required to perform these tasks are extremely high to guarantee real-time performance on devices that have inadequate computational power.

- **Hybrid**

  Hybrid approaches for tracking exist that combine sensors of the device and the images captured to build a more robust result. These methods surpass the limitations of each technique faces alone but it is more complex to implement. An example of such techniques is the usage of GPS [44] and gyroscope with markerless-based tracking that can build a better understanding of the environment based on the world coordinates and the rotation of the device resulting to better localization.

The hardware used for an augmented reality experience include a computer with a web camera, smartphones, glasses, head mounted devices and more. Mostly the smartphones are used because of their portability that makes it appealing for developers to create applications to. Another commonly used device is the computer, as the processing power and the fixed position that makes it perfect for a marker based augmented environment with high accuracy. However, the drawback of a desktop computer is that it requires many external hardware components to enable some functionalities. An example is the sensor-based techniques most desktop computers do not poses and should be obtained from an external device.

Interaction in augmented reality environments can be performed in many ways. Depending on the device that is used, these ways can be from standard controllers to touch screens, using bare hands or the device itself as means of interaction. The focus in this thesis is on mobile devices thus smartphones mostly. On smartphones the leading means of interaction in augmented reality environments is through the touchscreen. The other most used means are using the device itself and by using bare hands. Each of these techniques comes with advantages and disadvantages that will briefly described in the following paragraphs.

- **Touch-based interaction**

  Touch-based interaction is highly used technique of interaction in augmented reality applications. Thou it is easy for the users to interact through the touch screen, the touch screen allows for 2D manipulation of the object while 3D manipulation is required in augmented reality. To address this problem researches proposed a two-finger interaction to calculate the translation or rotation that the virtual object will take on the z-axis [8].

- **Bare hands interaction**

  Bare hands interaction enables the user to interact in a natural way with the objects in the augmented reality environment. Research has been conducted for this technique that will be described later. Usually on these techniques there is an interaction with one hand rather than both hands [45]–[47]. This interaction is achieved by tracking the hand on the scene and estimating its pose, or by recognizing predefined postures. These techniques usually require a huge amount of computational power that is hard to be found on a mobile device and the battery life is drained fast through those computations.

- **Device-based interaction**

  Device-based interaction provides the advantage of using the mobile device like the smartphone as the means of interaction, utilizing the inertial sensors it possesses. An example is to start moving an object as the smartphone moves in the environment, or to rotate the smartphone while the virtual object rotates by the same degrees [48], [49]. This type of interaction was used to enable the 3D object manipulation [50]–[52]. Device-based interaction is presented as the most stable way of 3D interaction as it is free from occlusions and many studies have been conducted exploring how to take advantage of this method.

More interaction techniques exist in addition to the previously presented as the real object-based interaction [53], [54]. In this technique a virtual object is bound to the translation and rotation of a real object that acts as a controller for the virtual object. In this technique attention should be paid to the range of the trackable real objects to successfully operate. For the previously mentioned interaction techniques certain issues has been observed. Namely these issues include occlusion problems, fatigue phenomenon, prior knowledge needed, intuitiveness/ naturality, low precision, position mismatch in 3D, position and orientation in 3D object manipulation, large-range translation and rotation and the degrees of freedom allowed for 3D rotation and translation.

The issues mentioned previously constitutes the main problems that are encountered upon interaction in augmented reality applications. Each technique has a different approach on every issue and some of those issues are not addressed in certain techniques. Future directions on improving those issues include also realistic augmented reality experience and wireless networking.

2.2 Hand Tracking & Pose Estimation

Hand tracking and pose estimation of the hand are the keys for a natural interaction between human and computer in augmented or virtual environments. This brings substantial challenges as the hand has a complicated anatomy that supports a variety of movements, enabling the completion of accurate and precise tasks with speed. Techniques to track and retrieve the pose of the hand have been tested, like the invocation of a variety of sensor types as colored sensors or markers on the hand or fingers.

However, these techniques required plenteous calibration and measurements to be precise, so the efforts to solve this problem were set on vision-based techniques that would allow for a more natural way of interaction. The vision-based techniques use computer vision algorithms which aim to reproduce in a digital way the ability of human vision to sense moving objects [55]. These techniques acquire an image from one or more RGB cameras, stereo cameras, or RGB-D cameras. RGB-D cameras have shown the best results on the retrieval of the hand pose estimation.

There are many challenges in the ability to successfully track the hand, both technical and general. Low-quality cameras and low computational power are two bottlenecks in capturing and processing the hand data [55]. Depth ambiguities, occlusions, complicated backgrounds, different shapes of hands are just some of the general difficulties in the field. In the sense of the pose estimation there are also problems that need to be addressed, such as the unrealistic collision with the virtual objects and gimbal lock that is met when two rotational axes are aligned. As far as the interaction is concerned, two basic problems arise: the hand-object interpenetration and the object sticking.

Figure 1. Representation of the hand keypoints *[55]*.

The hand can be modeled in a vast variety of ways. The most used way of representation of the hand is as a kinematic tree with each finger represented as a kinematic chain [55]. Each finger chain has three segments which are connected hierarchically with one degree of rotational freedom (this means that it can perform rotations only on one axis) (Figure 1). The overall movement of the hand is managed from the wrist that is commonly modeled as a part of the hand and possesses complete freedom of movement (positional and rotational). In such models, the rotation is represented by a quaternion to handle the possible problem of gimbal lock when using Euler anglers.

Each constraint on the fingers is local, leaving the global state of the hand to the wrist, thus minimizing the risk of gimbal lock mentioned as stated earlier. Except for the positional and rotational constraints, more constraints are applied on the hand, including the maximum range of an angle the hand can rotate. Finally, this representation is displayed on the screen by simple shapes like spheres, lines and cylinders [56]–[58]. This is due to the performance issues that would occur if a more synthetic model of the hand were to be displayed.

Tracking of the hand can be separated into two categories based on the process followed to track the hand in the environment. These categories are the following:

- **Discriminative**

  These methods are commonly treated as a pattern recognition problem that tries to extract visual features to make a prediction. Discriminative methods provide the probability of a target being of a certain class, thus it may present multiple valid outcomes for the target. Among models of discriminative techniques are the Decision Trees, Support Vector Machines, and Neural networks.

- **Generative**

  Generative methods use a priori shapes and kinematics of the hand commonly combined with information from previous frames that describe the pose estimated. In these methods, the recognized points of the hand are checked against the predefined model as well as the poses from previous frames and through minimization techniques the best fitting pose is extracted. In generative models the probability of an observation to match a target is modeled. Generative models include Hidden Markov models, Gaussian mixture models, Bayesian networks and more.

- **Hybrid**

  As on many areas the combination of two techniques helps to overcome the difficulties that each technique might pose. This is also the case here, the discriminative and model-based techniques combine to take advantage of the discriminative power of the first and the explanatory power of the second [55].

When interacting with objects in a virtual or mixed environment, a large number of misconceptions can be introduced to the user, as the adopted interaction behavior is in real world environment. Those misconceptions triggered by a variety of reasons such as the pose of the hand in the virtual environment can look a little unnatural, or the hand will overlap with an object that normally should not happen. Other reasons are the latency and the delay of a machine, that will result on distortion on the user's hand movement. Except from the visual difficulties there are the physical, for example the weight of an object or the friction. These concepts cannot be solved without external hardware such as force gloves.

Chapter 3 - Related Work

As the physical interaction with virtual objects in Augmented and Virtual reality environments has become an interesting topic in the field and the world has shown that would like that capability, several studies have been conducted.

Handytool [59] is a method that tries to simulate tools with hand gestures and aims for a better interaction with virtual objects in this way. Inspired by the interaction of hand on a musical instrument called theremin [60], a method is suggested to use the hands in a way that will minimize the possible errors that occur when the hands occlude (one top of the other/ near etc.). They have developed a gesture-based interaction system which have implemented interactions such as menu selection, object manipulation (position, rotation), deformations that can be combined as well.

Aiming for a realistic grasp gesture [61], simulations of grasp actions of the hand have been modeled to predefined rules to overcome the difficulties of a pure physics simulation. For an object to be considered as grasped, the particles of a virtual hand (simulating the real hand) are checked, if two or more are interacting with the same object from opposite directions, then the object can be moved as together with the hand. A similar method is proposed [62] by using a glove-based system, that will enable the user to experience haptic feedback as well.

A virtual grasping algorithm using machine learning and particle swarm optimization is proposed [63] to automatically pre-compute stable grasp configurations. This method invokes a hidden virtual hand that performs the interactions with the virtual objects by checking if it overlaps with them. A visible hand is also implemented that will perform the realistic representation of the grasp action. This is achieved by tracking when the hand is near an object to start generating pre grasp postures, then searching from the learned grasp space the physically plausible grasps to match the pose of the hand.

Using Convolutional Neural networks and Convolutional pose machines to recognize hand gestures and hand tracking techniques, [64] a method is proposed for controlling TV functionality. It provides 19 gestures than consists of numbers 0-9, menu, direction (up, down, left, right), go back, mute/unmute and nothing. To better filter the random movements, two states are provided (active & inactive) and the outcome of the predicted gesture type must be the same for a prefixed number of frames to be considered as correct.

A method for collaborating remotely on an augmented reality scene [65] is described enabling users to have shared workspaces. The method obtains coordinates from one camera by simultaneous localization and mapping method that are shared and adjusted between the devices. Users can select the initial points of the scene in their environment to help for a better fit of the shared workspace. For hand tracking Convolutional Neural Networks and extra sensors are used and for the pose estimation Inverse Kinematics optimizers or particle swarm optimization are applied.

Other studies [66] tries to improve user's immersion in the production process of virtual reality content for mobiles using a gaze-based method. An augmented reality hand interaction method is proposed [67] for the interaction with objects in museum. Both those methods use the Leap Motion for the hand tracking.

Different methods for the hand tracking have been proposed in the literature, from image processing and computer vision algorithms that extract features to recognize the hand and the fingertips [10], to external devices such as Kinect or Leap Motion that uses sensors to track the pose of the hand. Other methods are gloves with sensors that also gives the ability to provide haptic feedback. Recently with the rise of artificial intelligence and machine learning, many studies use artificial neural networks to recognize and track the hand.

Chapter 4 - Software Design

In this chapter the software design of the toolset will be described. Requirements of the toolset will be discussed as well as the actions needs to be taken to fulfill them. The overall architecture of the toolset will be presented indicating the key components and functionality. Furthermore, the structure and design decisions of the proposed toolset will be analyzed.

4.1 Problem Description

Most frameworks and toolsets focus to cover a certain problem on user interaction with virtual objects or provide the tools to build applications using specific already implemented functionalities. Such a framework or toolset most often might cover just the interaction part, or the hand tracking part, or a scenario of usage of hand tracking and interaction with virtual objects. While those frameworks are good to exist and helps many people and developers to create content, there are some throwbacks that occur from the usage of a different framework for every task. Such bottlenecks can be the compatibility between frameworks, the inability to add further functionality or extend its functionality without needing to implement everything from the beginning. Furthermore, usually it is not possible to change a part of the code or workflow with another that better suit the needs of the desired application.

With that in mind, a toolset is proposed that is free enough to let the developer to change or add functionality providing the tools to make this easy and without breaking or spending much time trying to make the new code to be compatible with the toolset. It also provides enough functionality that makes it easy for a designer or creator to implement such an environment without the need of coding knowledge.

4.2 Software Analysis

In this chapter the overall process of creating the toolset will be described. How the goals and main functionality was set, as well as how it decomposed into several tasks to facilitate the implementation. Requirements of the toolset will be discussed also concerning the performance, maintainability, and more. Design decisions will be explained, and diagrams of the overall toolset will be presented.

4.2.1 User Stories

To better understand the goals of the toolset and organize the tasks needed to be achieved user stories were created. User stories are separated to three categories as the persons that will likely use the toolset. These persons are the end-user that will use an application created with the toolset, the developer that will create an application and probably will want to add or customize existing functionality and finally the designer that will want to create an application without writing code.

Tables 1-3 present the fundamental functionality that each person category require from the toolset.

Table 1. Developer User-stories

Developer
As a developer I want to detect the pose of the hand so I can know when an interaction might occur.
As a developer I want to display a virtual hand on screen to let the end-user know how the tracking is performed.
As a developer I want to check if an interaction is possible so I can perform an action.
As a developer I want to capture the image from the device's camera so I can perform a hand tracking algorithm.
As a developer I want a hand tracking service to recognize the hand pose.
As a developer I want to add new functionality on the toolset so I can cope with the needs of a project.

Table 1 lists the developer user-stories. These six user-stories describe the core functionality that a developer will need from the toolset to implement desired applications. Each user-story will lead to the creation of one or more components for the toolset to satisfy the needs of the guidelines set. The most important as the most difficult to implement from these user stories is the last one that asks for expendability of the toolset from the developer at the time. From this final user-story the implementation techniques of the toolset will be decided.

Table 2. End-user User-stories

End-User
As an end-user I want to be able to interact with virtual objects so I can perform tasks with these objects
As an end-user I want to perform different gestures so I can interact with virtual objects
As an end-user I wat to perform different actions so I can interact with various ways
As an end-user I want to know if I am about to interact with an object so I can focus on that action

In Table 2 the end-user stories are listed, reporting the needs of the person that will use applications created by this toolset. There, four stories were extracted as those were enough to describe the basic functionality required by the end-user. The satisfaction of these user-stories is crucial as it involves the end-user and the overall experience that the application will provide. As so, the features in these user-stories comprise the focus of the toolset. It must be mentioned also that taking into consideration the further needs of the end-user the final task on the developer user-stories exists. As to be able to cover future needs that might arise that has not yet been exposed.

Table 3. Designer User-stories

Designer
As a designer I want to create an environment without writing code so I can make an application on my own
As a designer I want to configure basic information on tools so I can customize them to my needs

The final user-stories are those that involves the designer as indicated in Table 3. As designer is described a person that wants to create an application with the toolset but does not have a good programming knowledge to use the toolset by code to create one. The designer is a person that have an idea of an application, can describe, and organize that idea in several steps, or levels for example, and visualize the entire flow of the application to be created. So as, two user-stories are indicated for designers to allow them to create or structure an application in a way that suits them also.

4.2.2 Requirements

The user stories in section 4.2.1 can be decomposed into smaller tasks so the required functionality and specifications of the toolset will be exposed. In this way it makes clear what needs to be implemented and in what way so the desired functionality will be achieved. Breaking the project into smaller parts allows for better testing of the completed functionalities. Furthermore, those tasks can also categorize based of the importance of each one or the type of the task. For example, all the tasks that concerns the connection of the two modules should be of high importance in terms of the implementation order as it is required for the rest of the toolset to work. Both those tasks also can be included in a type named network connection as their role is to connect the two modules over a network.

In Table 4 the two user-stories about the designer have been broken into smaller tasks as written on the right, and on the left the processes that needs to be done to fulfill the user-story. As shown, the user-stories has exposed specific tasks that each of those tasks reflects to a specific way that the components of the toolset must be implemented to allow the ease of usage to the designers.

Table 4. Designer User-stories requirements

As a designer I want to create an environment without writing code so I can make an application on my own	
1. Create complete components to be used. 2. Provide the components on a user-friendly way.	Make specific components available to “drag and drop” so no need of code required. Separate those components on folders that are easy to access.
As a designer I want to configure basic information on tools so I can customize them to my needs	
1. Give access to key variables and objects of the components. 2. UI to edit the variables.	Tag the variables and objects as serialized field or make them public. Display the variables on the editor with information on how to use them.

It is very helpful to expose those requirements in this way as it makes clear the way of the implementation process and it also helps to remember the goals to be reached.

The developer user-stories Table 5.1 and Table 5.2 are the most demanding in terms of requirements as there is all the functionality of the toolset. The six user-stories presented with the required tasks on the right and the processes to complete them on the left. As one can see the processes to complete the tasks are numerous, and from Table 5.1 and Table 5.2 the basic concept and functions to be implemented can be extracted. How those process will be implemented and organized inside the project will be discussed later in this chapter and will be further analyzed in components and classes that will make the toolset reusable and easy to understand. As said before the last task is the most important as it annotates the importance of complying to certain rules of implementation to allow expandability of the toolset without the risk of breaking the code dependencies or the need to write everything completely from the beginning. These methods of implementation will be mentioned in the next chapter.

Table 5.1. Developer User-stories requirements

As a developer I want to detect the pose of the hand so I can know when an interaction might occur	
1. Capture the image. 2. Send the image to the hand tracking service. 3. Receive hand joint positions.	Implement an image capture function. Implement a socket client to send captured images. Serialize image to be sent. Deserialize the received message.
As a developer I want to display a virtual hand on screen to let the end-user know how the tracking is performed	
1. Create materials for the virtual hand. 2. Create starting pose of virtual hand. 3. Implement update function to display the changes of hand position.	Create materials and 3D shapes to represent the virtual hand and store them as prefabs. Define the starting points of the hand as a list of 3D points. Implement a function that will receive the new detected points of the hand and place the existing virtual hand on the new position.

Table 5.2 *Developer User-stories requirements (cont.)*

As a developer I want to check if an interaction is possible so I can perform an action	
1. Check if the hand intersects with a virtual object.	Implement ray casters on the hand. Check if ray casters hit an object.
As a developer I want to capture the image from the device's camera so I can perform a hand tracking algorithm.	
1. Get the data from device's camera. 2. Process data to extract the image.	Get the bytes from camera data. Convert obtained data to RGB format. Rescale the image. Encode image to minimize the size. Release engaged resources.
As a developer I want a hand tracking service to recognize the hand pose	
1. Create a socket server. 2. Create function to receive streaming images. 3. Hand tracking algorithm. 4. Send detected points.	Accept user packages. Deserialize messages. Process received images to make them compatible with the hand tracking algorithm input. Serialize detected points.
As a developer I want to add new functionality on the toolset so I can cope with the needs of a project	
1. Make code expandable.	Export an interface from every basic class so a developer can use it to add new functionality.

Finally, the end-user user-stories in Table 6. End-user User-stories requirements Table 6 are broken down on the requirements on the right and the processes on the left again. From this analysis of the user-stories the processes to be implemented here concerns mostly the way that the end-user will interact in the scene with the virtual objects. Many of those processes describe visual representations, so that leads to the implementation of methods that will allow each time the manipulation of such visuals that will suit the needs of the application to be created.

Table 6. End-user User-stories requirements

As an end-user I want to be able to interact with virtual objects so I can perform tasks with these objects	
1. Check what gesture the hand has. 2. Check if there is an available object. 3. Select an action to perform.	Implement a generic gesture interface. Implement method that returns the hand points of a selected gesture. Implement a method that checks if the hand points of the gesture hit an object. Implement Interfaces for hand points acquisition (points of each finger etc.). Implement a generic action interface.
As an end-user I want to perform different gestures so I can interact with virtual objects	
1. Implement gestures. 2. Access different gestures.	Implement gesture classes. Implement an Enumerated variable to set a type for each gesture. Implement a gesture manager to facilitate the access on the gestures.
As an end-user I wat to perform different actions so I can interact with various ways	
1. Implement actions.	Implement action classes. Provide methods to customize a specific action if needed.
As an end-user I want to know if I am about to interact with an object so I can focus on that action	
1. Create UI to highlight the object. 2. Get the interacting object.	Create materials for the UI highlight. Create methods to apply and remove highlight on the object depending on whether the hand is interacting or not.

The tasks from user-stories in Table 4 through Table 6 describe the core functionality of the toolset and a starting trail on how to implement it. From those tasks, certain specifications are implied for the successful execution of the resulting product. More specifications need to be included to ensure the maintainability, scalability, usability, portability, and compatibility of the toolset. Such specifications are namely the following:

- The application should be able to run in real-time. This means that at least 13 frames per second should be guaranteed.
- A machine capable of running the hand tracking algorithm without delays (For example a desktop computer with at least a GTX1050 GPU).
- Follow programming principles to ensure the maintainability and scalability of the toolset code.
- It requires a good internet connection to achieve high rates of data transfer (at least 10Mbps).
- The toolset should be able support devices that are not augmented reality enabled also in the future.

4.2.3 Architecture

In Figure 2 the main architecture of the toolset is displayed. It is separated into two different applications. The left part describes a service responsible to recognize and provide the hand joints from an image received through a socket client. The right side describes the front-end of the toolset with the main functionality that is composed of. The main idea is to separate the hand tracking module to a web service and keep all the interaction functionality and representation on the toolset. This way the final application running on the device will be lighter to execute, so the overall performance of the application will be better. Adapting this approach of a web service both modules should have a component for communication between them as it shows in Figure 2. The other components that are needed in the hand tracking module are the hand tracking algorithm and an image processing component.

The toolset on the other hand consists of several components most of them for the interaction of the hand with the virtual objects. Those are pictured in a sub package of the toolset named interaction. In this package the virtual representation of the hand as well as the virtual objects are included as they are the two objects that will interact with each other. Also, a gesture manager the desired actions are also included in the same package as those offer the means of the interaction of the previously mentioned objects. The *Frame Capture* as the name suggests is a functionality required to capture on each frame of the application the input of the device’s camera. Finally, the *RayCast Selector* describes the means of checking whether an interaction is available or not.

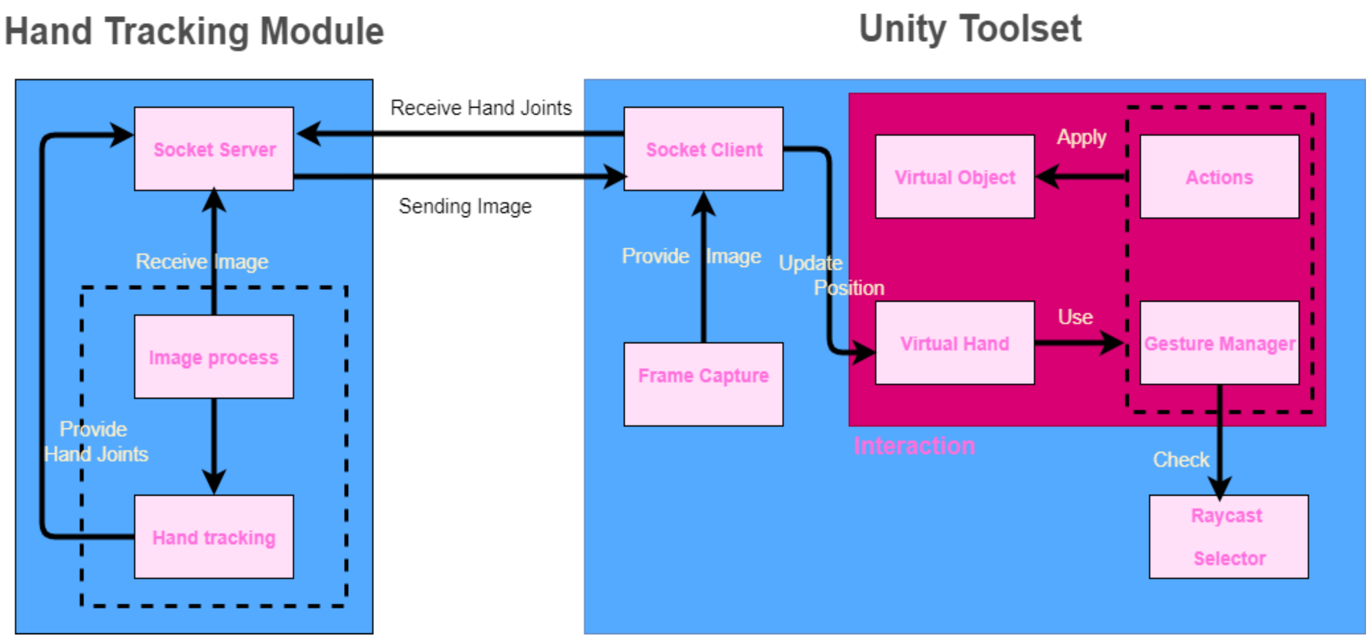


Figure 2. Architecture Diagram

4.2.4 Components

In this section are presented the components implemented to provide the required functionality. In these diagrams also is described the relationship between those components and how to use them to create an application. The *Player* and the *Service* components represents the main script that runs the application. From these diagrams is deduced the necessary creation and usage of these components for the application. This type of diagrams helps to break the system on its basic functionality, assist on the path of what it needs to be implemented and enhance the decision making about the system implementation. Both are color coded as reported in each of them, to better categorize the type of each component based on the task it is meant to perform. By coloring each functionality type allows the better understanding of the components and the connection between them.

4.2.4.1 Toolset Components

The diagram in Figure 3 presents the components of the Unity toolset. Those components are extracted from the requirements in the previous section and aims to the better usage of the toolset. The colors in each of the components describes a certain category of components in the toolset. Those categories are assigned based on the purpose of the component in the toolset. For example, the components that are presented with blue color are responsible to provide functionality about the way that an interaction will be detected. The *SocketClient* and the *ARFrameCapture* are both in green color as they both act as helper components that are required for the implementation and the correct execution of the toolset on the desired device.

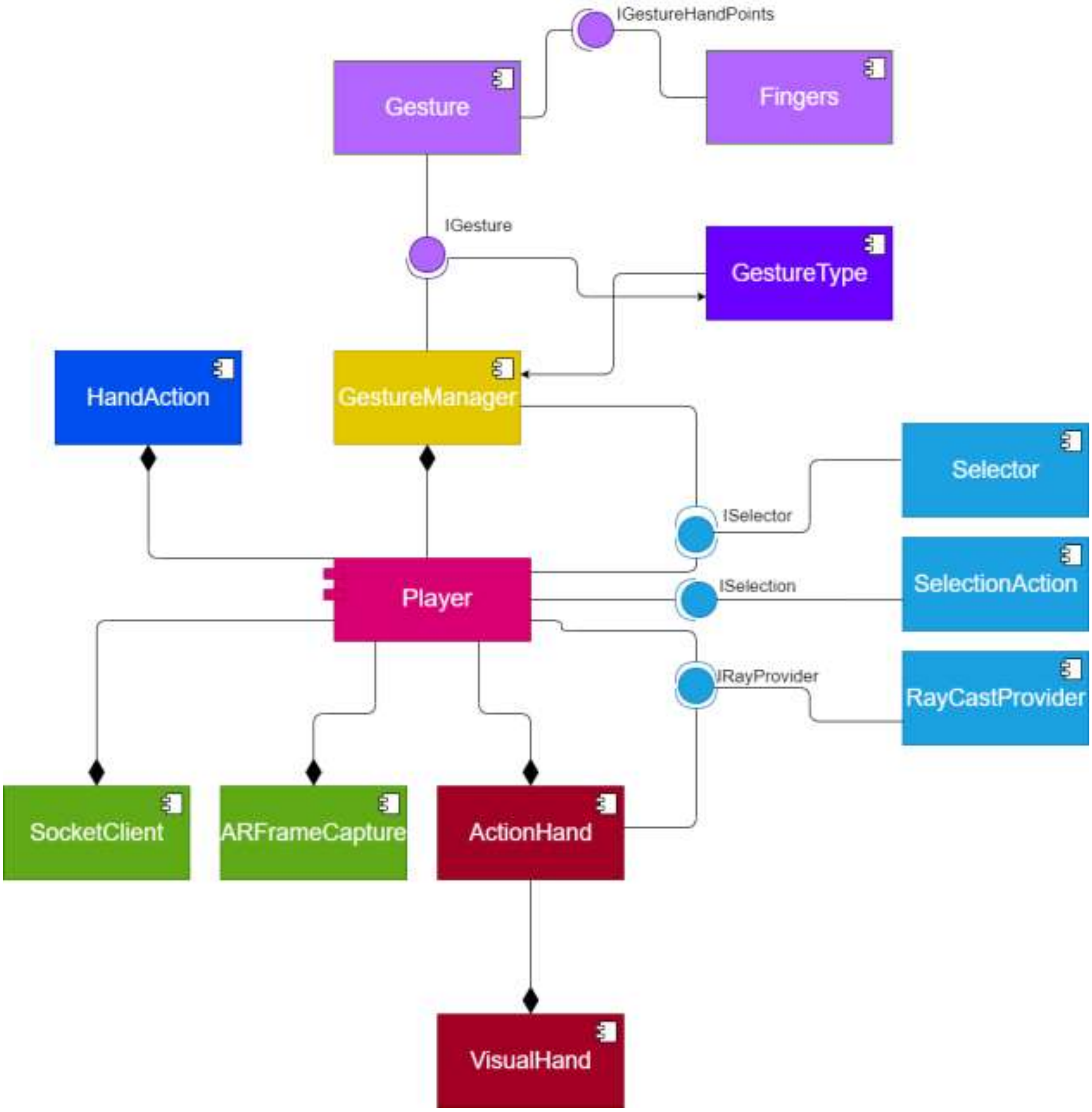


Figure 3. Toolset Components.

4.2.4.2 Service Components

Next, the components that compose the service for the hand tracking is presented in the form of a component diagram in Figure 4. To better understand the relation between those components, a color has assigned to each one based on the type of the task that it is assigned to perform. As shown the “Image Process” and the “Hand Tracking Algorithm” has the same color as they are thematically connected. The “Socket Server” with the “Message Receiver” also have the same color as the “Message Receiver” is connected to the “Socket Server”. The responsibilities of each component will be described in the next chapter.

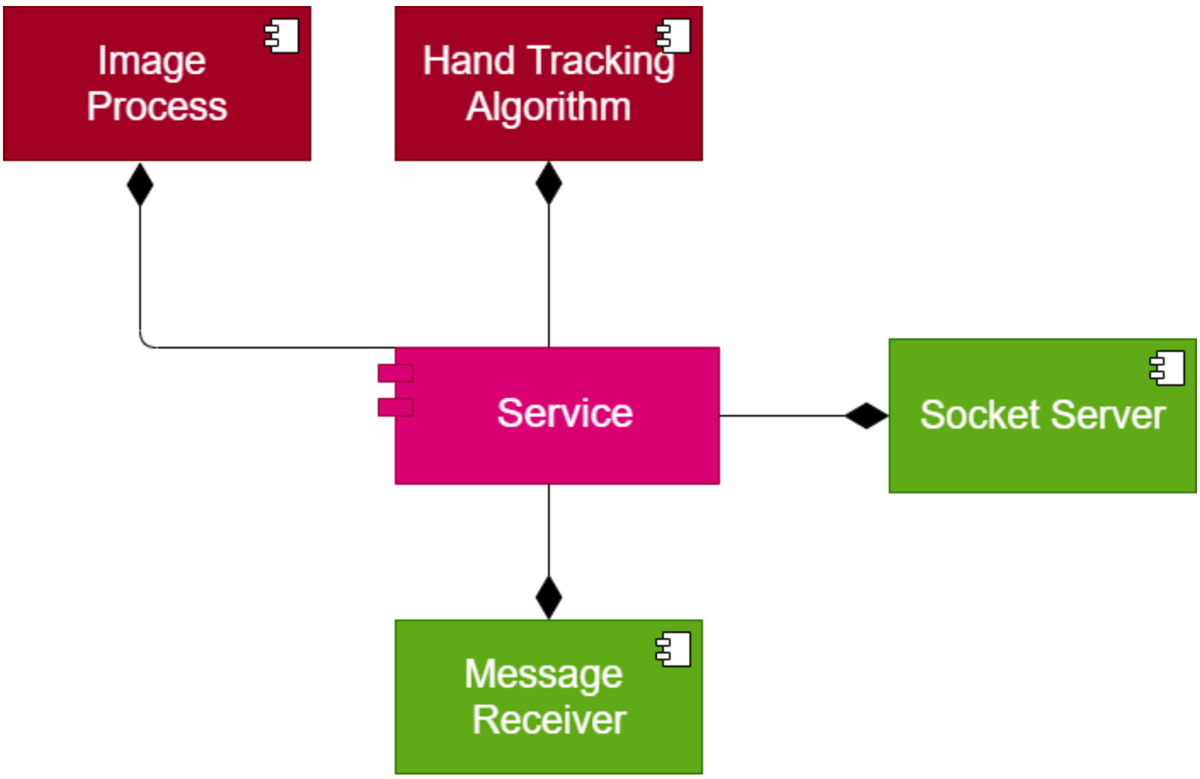


Figure 4. Hand tracking Components.

From the component diagrams all the information about the interfaces, classes, and the overall functionality that the system will require is organized into components and ready for the next step that is the implementation of those components. As it can be observed from the Figure 3 and Figure 4 the toolset components have a more complex design as the components are more, for this reason more demanding techniques required to structure the system complying to the object-oriented practices. For the service module the components are less so it makes easier to organize. The components on the hand tracking service also, does not require extracting interfaces from them as the assigned tasks does not change. The only thing required is to ensure that on this module the hand tracking algorithm should be able to change at the will of the developer. In the next section those components will be further analyzed on class diagrams that will include a more detailed representation of the final system, Those diagrams will act as a guide on exactly how a specific class should be implemented including all the methods and fields each class require.

4.2.5 Class Diagram

The packages and classes that were implemented on the Unity toolset is pictured in Figure 5. Each package contains the classes required for the tasks to be performed from it as described in the previous sections. These packages are color coded as the previous figures, so it is easy to find the connection between those and how the implementation progress. Later in this chapter an attempt to explicate each one of those packages and classes on design, implementation, and usage matters. There are six main packages and two sub packages under the *Gestures* package. There are also 2 classes that are outside of packages, these classes are thematically standalone therefore they are not categorized in a specific package. Also, the creation of a package to contain just one class would not be sufficient to justify it. In the future if more classes that fall under the same category will be created, they could be organized in a package.

As stated before, an object-oriented approach is followed by the toolset while the hand tracking module is structured in a different way engaging mostly functional programming. The hand tracking module has encapsulated some functionality into different files and folders to render some blocks easily removed from the module. One of those blocks is the hand tracking algorithm that should not in any way be directly connected in the rest of the module as this will make it difficult to replace it with a new one. This is the reason that a class diagram for the hand tracking module is not included here as the functionality of the components included will be described in the next chapter.

Figure 5. Unity toolset packages.

4.2.5.1 Gesture Package

The Gestures package is described from two sub packages an enumerated field and a manager class as shown in Figure 6.

- The *HandPoints* package contains an interface to be implemented from every class that will describe the points of a finger or points from several fingers. The interface provides a method that the hand points specified on the class can be accessed.
- The *GestureDefinitions* package has an interface for the implementation of gesture classes. Those classes as shown on the interface should have one field that describes the hand

fingers that the gesture interested in. The *GestureType* enumerated value indicates the type of the specific class. Finally, the *Check* method that the logic of the gesture will be implemented to check whether the gesture is performed on an object or not. This method takes as arguments the rays from the model of the hand as well as a type of ray cast selection. The return type of this method is a transform of object that the hand has triggered the gesture. If the hand does not trigger the specific gesture the method will return null.

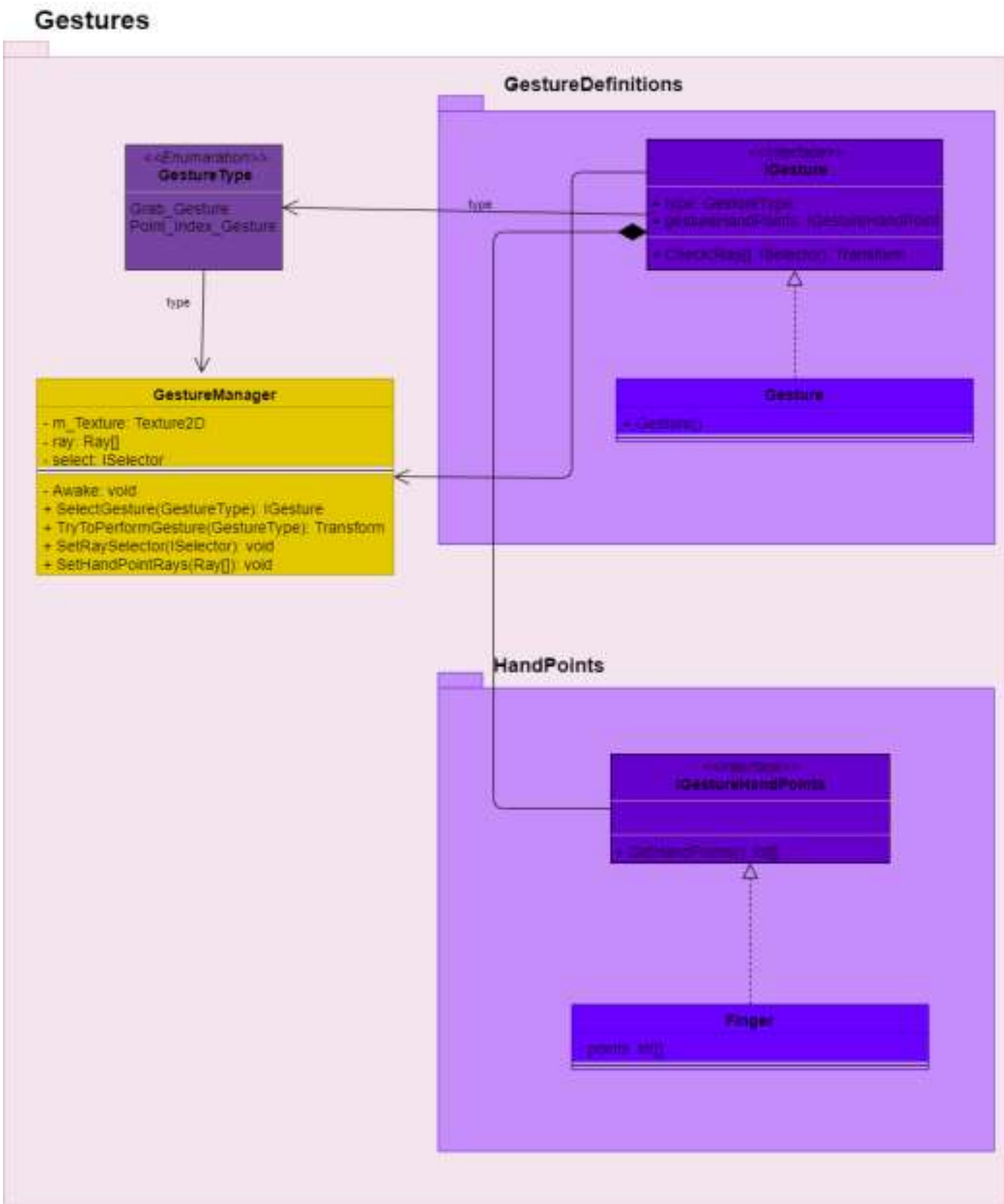


Figure 6. Gesture Package Class Diagram.

4.2.5.2 Hand Package

In the hand package are located implementations for the hand as shown in Figure 7. There is a visual representation of the hand with core functionalities and a more action based.

- The *VisualizedHand* contain all the functionality for the hand to be displayed on the screen and its position to be updated on every frame. Furthermore, there are implemented functionalities to allow for the retrieval of certain hand points from the model. For this class to work properly a *Material* and a *GameObject* object needed to be defined. There are predefined objects for this purpose, but the user can also create and use new customized objects.
- The *ActionHand* class wraps the functionality of the *VisualizedHand* and it also provides an initialization of ray casters that will be used as the mean of the interaction with the virtual objects.

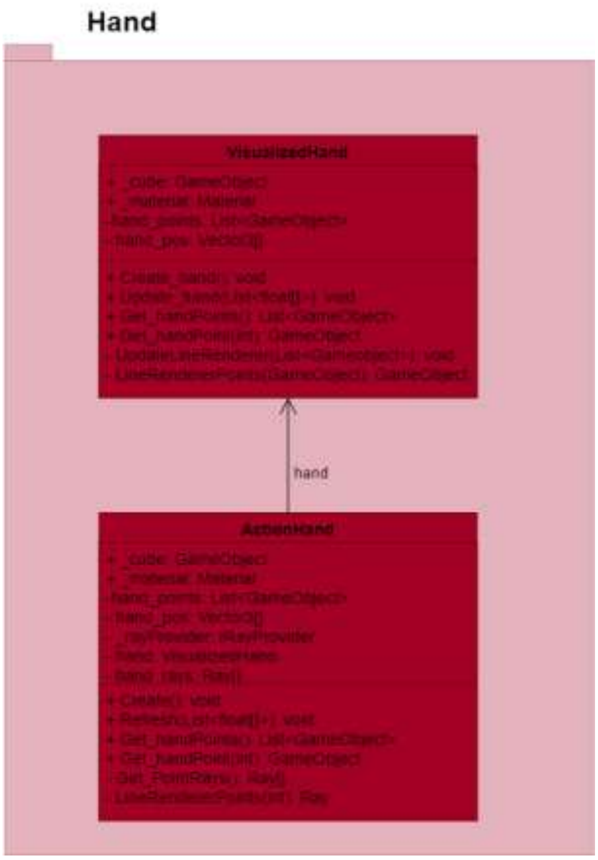


Figure 7. Hand Package Class Diagram.

4.2.5.3 Hand action Package

The HandAction package in Figure 8 has been created to provide a starting point on how the actions can be implemented. The package provides some actions already implemented such as movement, but it is mainly up to the user to create new actions that will suit the needs of the application being created.

- The first interface named *IHandAction* consists the base for all the other classes in this package. It offers the method *PerformAction* that requires a Transform as input, that describe the object that the action should be performed to, and a *GameObject* array that represents and points on the hand that will perform that action. An overload to this method is provided also with one more argument that is a generic type of <T> object. This is to give the opportunity for further control over the way that the action will be performed. For example, on the movement action implementation the third argument is implemented as an enumerated value to give the ability to move the object on one specific axis. This interface is not meant to be inherited directly from a class.
- The second interface is the *IHandAction<T>* which inherits from the base interface, but this indicates that the class which will implement this interface have to define the type of <T> on the declaration of the class. This interface can be used from a class that needs to implement the generic type right away.
- *HandAction* is an abstract class that its purpose is to be inherited from the action classes. It has a default implementation of the *PerformAction* method with the generic type of <T> argument. This allows the classes that inherit from that class to not implement the generic type of <T> method if it is not needed.

The design of this package is created to allow the access to all actions through a common interface, but also be versatile enough for the needs of a new action that might be needed. The generic type of <T> interface is a problem to be accessed as is because the type should always be defined when accessing the object. This interface is not meant to be used as the main guide, but only on specific and individual cases. To create new functionality the common way to be implemented is through the abstract class from the package and if necessary, override the already implemented with the generic type of <T> method.

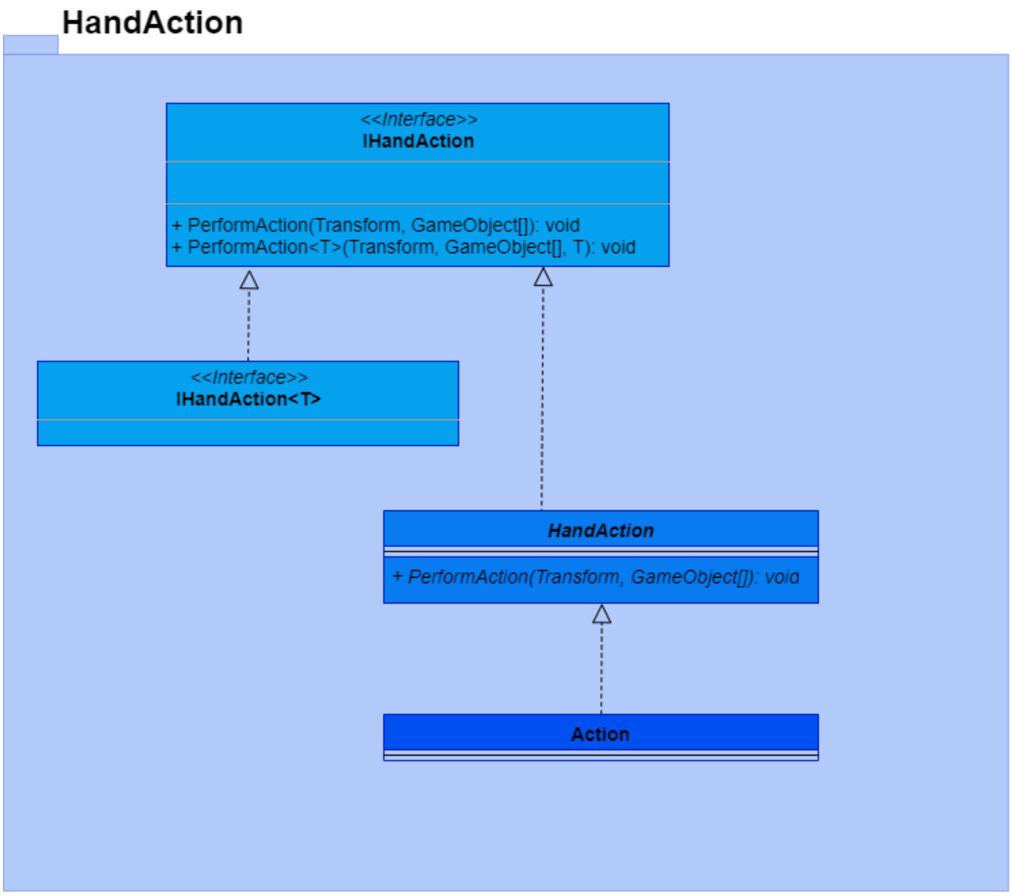


Figure 8. HandAction Package Class Diagram.

4.2.5.4 Selector Package

The selector package in Figure 9, is a package that determines the way of interaction between hand and virtual object. The *ISelector* interface provides the core functionality in an abstract way, and then the classes described as *Selector* will implement the necessary methods inherited from the *ISelector* interface. The logic of the decisions whether the hand has the ability to start interacting with an object or not is derived from this class. Each implementation of the *ISelector* interface can be different and as a user sees fit for the application under development. The new *Selector* classes have only to implement the *Check* method and comply to the requirements of input and return values of the same method.

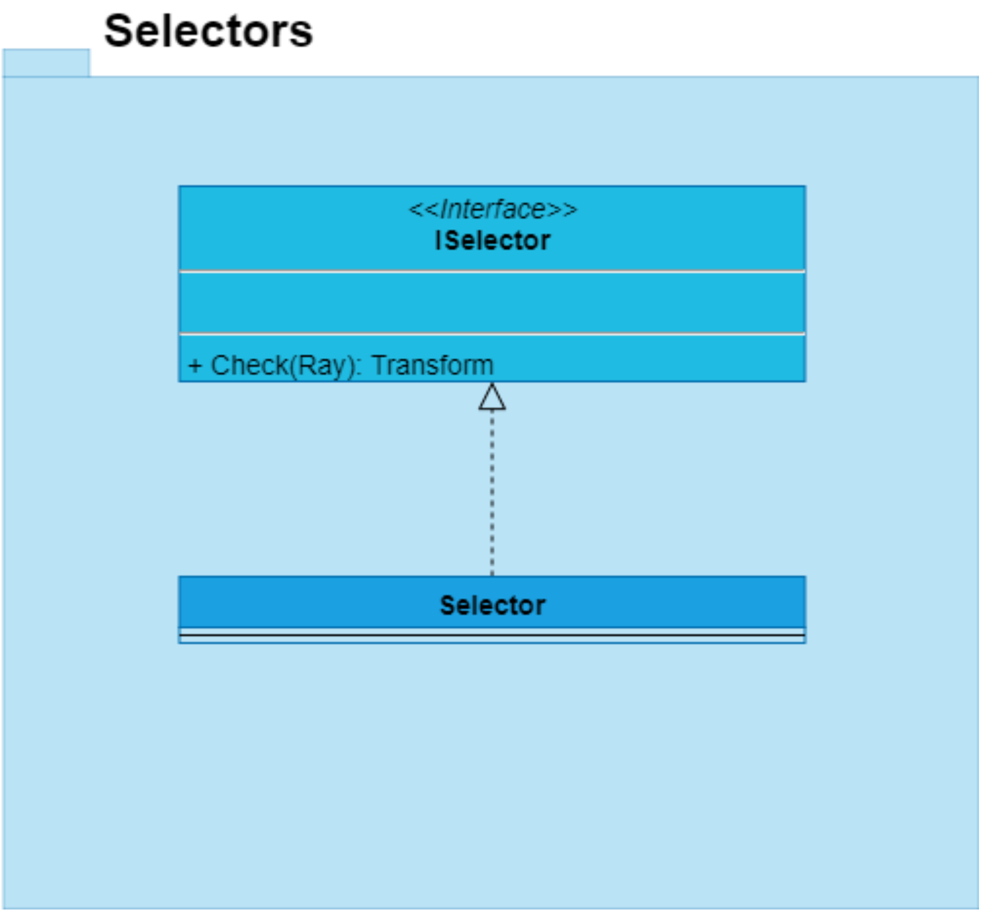


Figure 9. *Selector* Package Class Diagram.

4.2.5.5 Selection action Package

The *SelectionActions* in Figure 10 package provides the means of representation of the object that an interaction is about to begin. Such representations might be described as a highlighted outline on an object or as the usually met blur as the mouse on a button when it passes above it. There are two methods for this interaction. The *OnSelect* and *OnDeselect* which implements what should happen when an object is valid for interaction and when the object will stop interacting, receive the transform of the interacting object and manipulates it accordingly. Each of those methods can be called from the main glow of the code of the application by the developer.

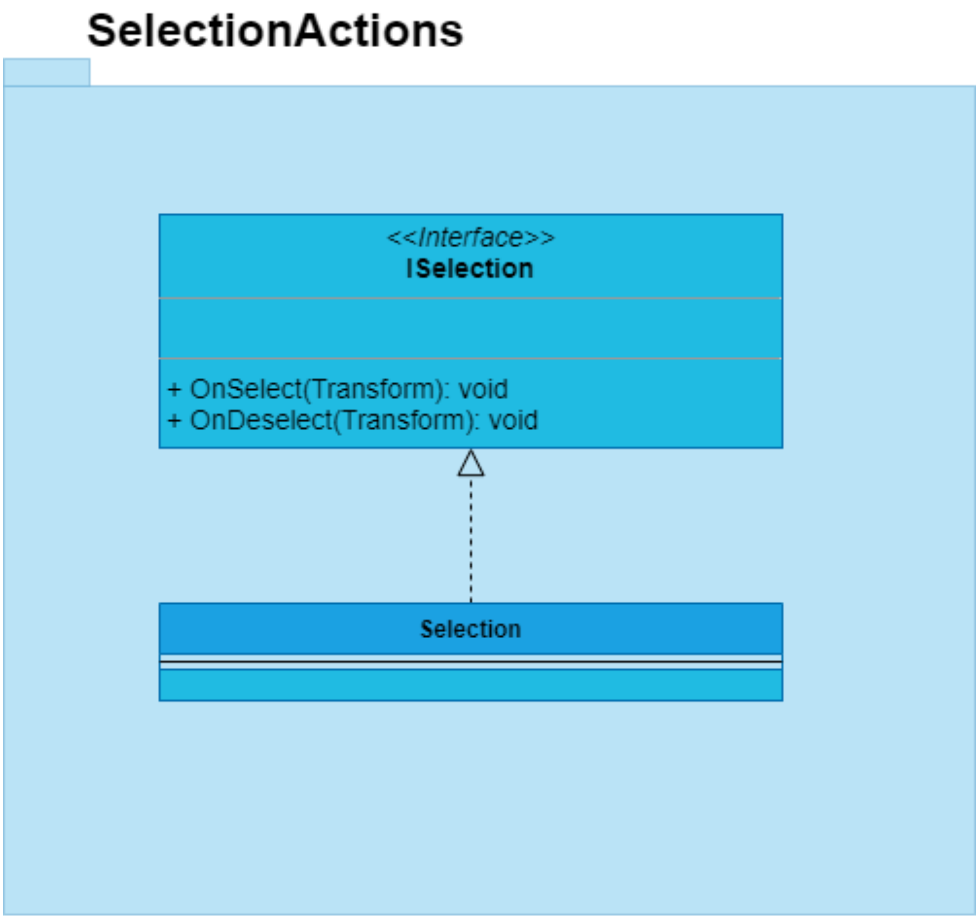


Figure 10. *SelectionActions* Package Class Diagram.

4.2.5.6 Ray caster Package

The *RayCasters* package in Figure 11 is the package that different types of ray casters are implemented. There is a basic interface with one method. This method is responsible for the creation of the ray caster that each class will implement. *Raycasters* already implemented on this toolset are the following: casting from a *GameObject* position and up, forward, and finally a *RayCaster* that convert the position of a *GameObject* to screen coordinates and create a ray from the screen to the scene.

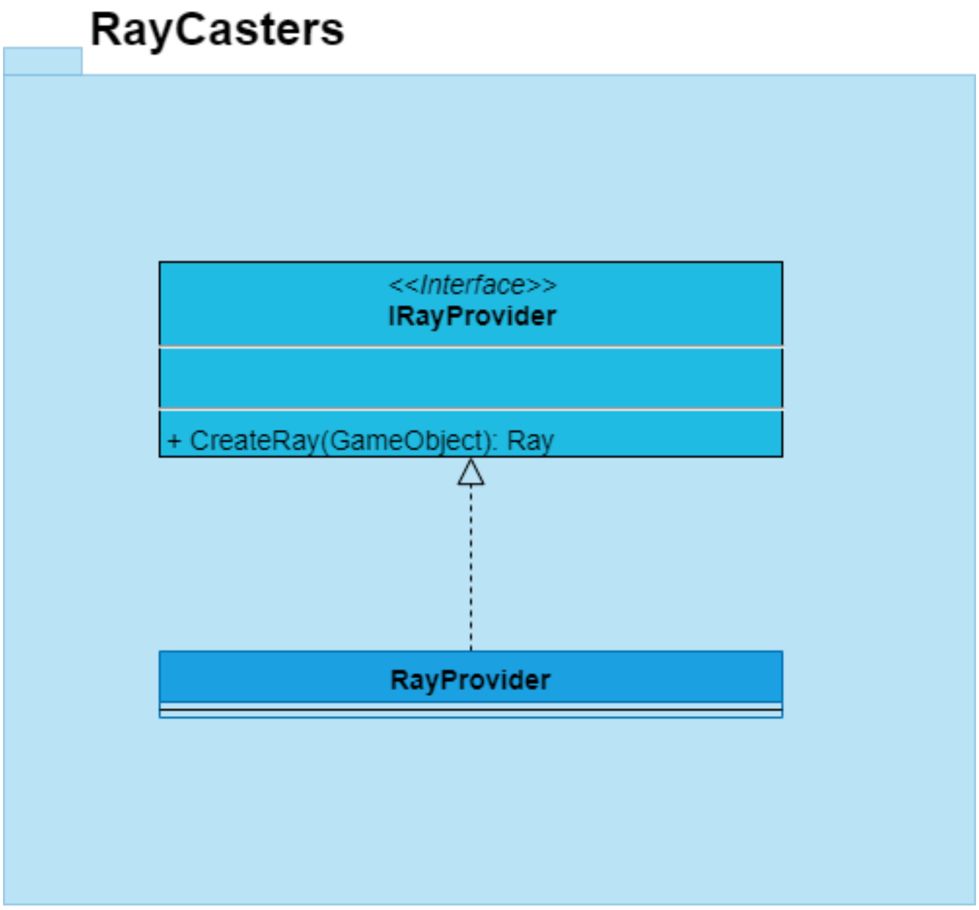


Figure 11. *Raycasters* Package Class Diagram.

4.2.5.7 Socket client class

In Figure 12 the *SocketClient* is outside of a package and it is responsible to provide the connection with the hand tracking part of the toolset. This class as the name suggests is the implementation of a socket client that allows the creation of a stream of data through the application on the mobile device to the hand tracking service. It is a singleton class, meaning that only an instance of this class can exist throughout the lifecycle of the application. This class connects to the hand tracking service and sends images obtained from the camera of the mobile device. As a response it receives the recognized points on the image so that they can be processed by the application and then displayed on the screen. As the only thing that this class is responsible to receive is hand points on a specific form (a list of float arrays) it has a deserialize method responsible for the conversion of the received data to the desired format.

Figure 12. *SocketClient* Class Diagram.

4.2.5.8 Frame capture class

The *ARFrameCapture* class in Figure 13 is responsible to acquire the frames from the camera and convert them on a desired format so that it will be able to be send through the *SocketClient* class to the hand tracking service to be processed. This class receives frames directly from the augmented reality enabled camera device. Each frame is processed accordingly to be shaped on a common image format that can be used later by other programs and services, in this case the hand tracking service. This method as the name suggests needs a device that is compatible with the augmented reality technology, which means that it has to be equipped with an augmented reality camera. The implementation of such a method for devices that are not augmented reality enabled is under development for future versions of the tooolset.

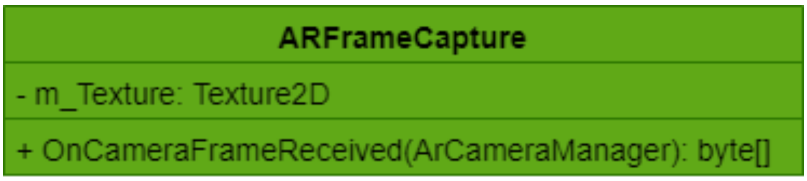


Figure 13. *ARFrameCapture* Class Diagram.

4.2.6 Flow Diagram

4.2.6.1 Toolset usage Flow Diagram

In Figure 14 the workflow of the program on Unity3D is represented. This workflow is just a demonstration of the logical order of functionalities to be executed. This workflow depending on the needs of the application to be created might change. Such a change could be the execution of an action right after the update position of hand.

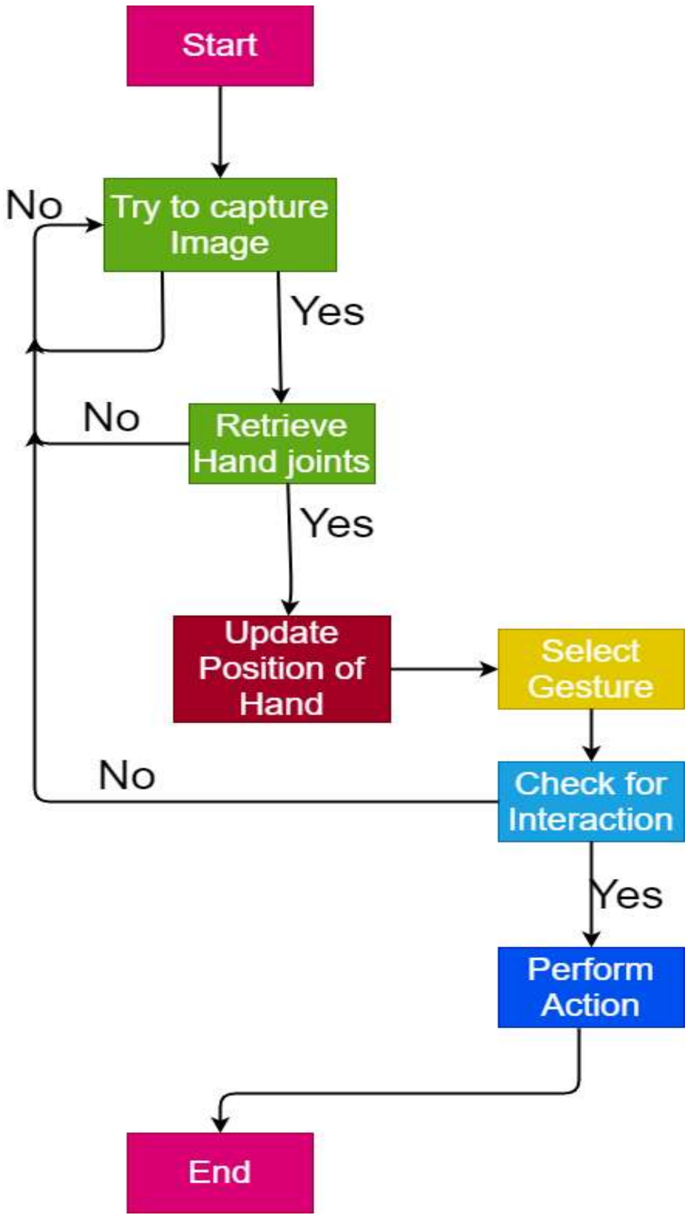


Figure 14. Toolset Flow Diagram.

4.2.6.1 Hand tracking service Flow Diagram

In Figure 15 the workflow of the hand tracking service is described. It starts when a connection is attempted, and the main responsibility of the service is to receive the message correctly and bring the received image in the desired format so it can be used by the hand tracking algorithm. Then the hand tracking algorithm extracts the hand joints and sends a response to the client with the recognized joints as a list of floating-point values

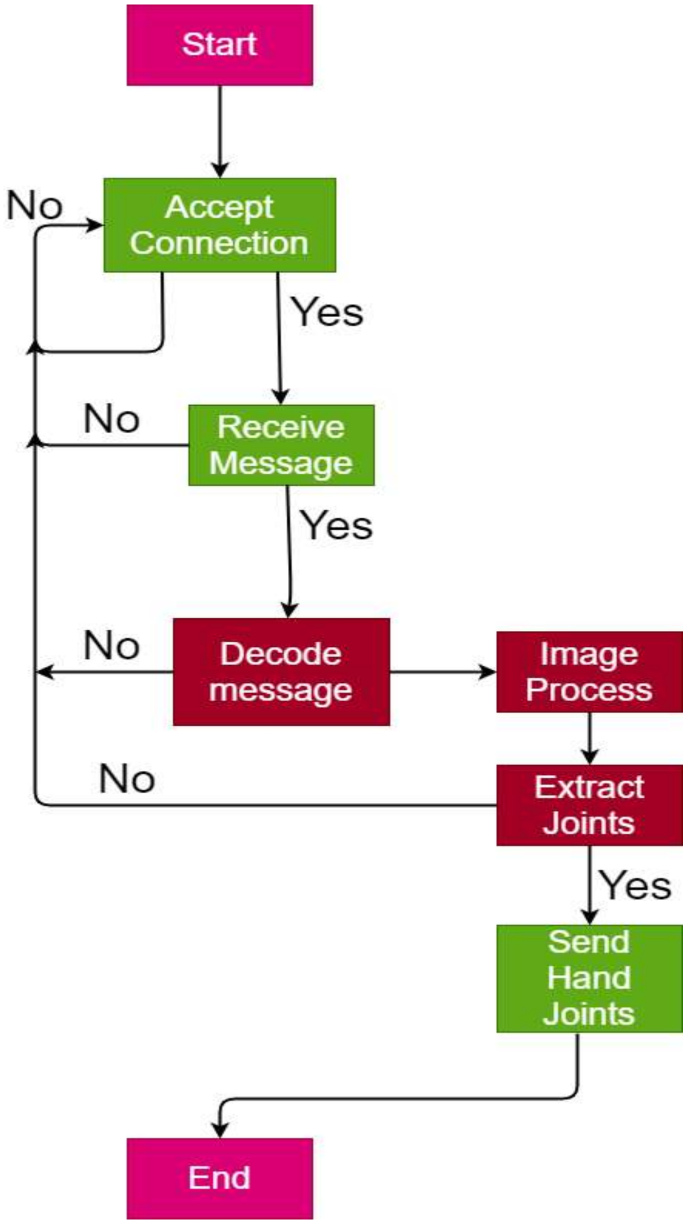


Figure 15. Hand tracking service Flow Diagram.

Chapter 5 - System Implementation

5.1 Technologies and Frameworks

Unity:

Unity game engine is a "software framework" that facilitates the creation of 2D, 3D, virtual reality, and augmented reality applications. Provides built-in functionality to support features such as rendering and physics that are difficult concepts to be implemented by any user. The selection of this game engine for the implementation of this project was based on the reasons described next.

Unity is easy to understand and start creating content with. This is because it offers a vast number of tutorials, from entry level to advanced. Unity also provides a good documentation about the functionalities, tools and components that it provides for development. In contrast with other game engines, Unity's editor is more user friendly. Another great asset for beginners and non-programmers is the drag and drop functionality that allows to create content without writing a line of code. Unity offers a free license and only bills the developers after a certain amount of revenue. Many developers use Unity for their applications, it is preferred by 45% of independent developers, making Unity the most popular among the other game engines. It makes easy to share projects with others, offering a store for unity projects and assets that can be downloaded and imported on other projects. The process to import and export a project is made simple by using the built-in tools for it.

One of the assets Unity has to offer is that it supports build tools for different platforms including Android, Windows, Linux, Xbox and more. A key factor for the selection of Unity is the support on augmented reality and virtual reality technologies, providing tools, libraries, and frameworks to build such applications with many ways and a large number of those ways do not require a specific hardware device.

AR Foundation:

AR Foundation is a package in Unity for working with augmented reality concepts. It supports concepts such as world tracking, plane detection, light estimation, object tracking and more. AR Foundation is dealing with its functionality through subsystems. Each subsystem is responsible for a specific functionality, for example the plane detection is handled from its own subsystem. Because AR Foundation allows working in a multi-platform way, providing a common

API both for Android and iOS mobile phones it needs the installation of a platform specific augmented reality package (ARKit on iOS and ARCore on Android). Those platform specific packages gather the implementation of each subsystem (AR Foundation providing only the interfaces). AR Foundation wraps the platform specific implementation and provides it on the subsystem form that is a common API on Unity.

ARCore:

ARCore is a platform made by Google that enables the building of augmented reality applications on mobile phones. ARCore consists of three basic functionalities, these are the following [68]:

- Motion Tracking: The ability to understand the device's position in the real world.
- Environmental Understanding: the ability to understand surfaces in the environment like the floor, walls etc.
- Light estimation: The ability to understand the lighting on the environment.

ARCore uses a process called simultaneous localization and mapping (SLAM) to track the position and the orientation of the mobile device. This is achieved by using the device's camera to identify interesting points in the environment and track them as they move, combined with the internal sensors (e.g. the accelerometer) of the device. In this way it can also build an understanding of the real-world environment and can detect flat surfaces which helps to estimate the lighting in the area.

TensorFlow:

TensorFlow is a machine learning software library created by Google and it is used by most of the developers for applications that require neural networks. It can run on many operating systems such as Windows, Linux and MacOS and it also provides versions for mobile operating systems as Android and iOS. TensorFlow offers its API on many programming languages, but mainly the Python library is used. TensorFlow can use the device's CPU to run the required models or Nvidia GPUs [69].

With TensorFlow problems as image recognition, natural language processing, handwritten digits classification and more can be easily addressed by training such models to running them. It offers a tool for model visualization called TensorBoard to provide information during the training process. TensorFlow releases new versions on a regular basis that always improving the library and fixing bugs that exists. Although is a good thing to always improve and update a library it comes with problems and these are the deprecation issues that occur. This means that each version might work completely differently from another so it should be treated carefully when updating to a new version or when a deployment ready model is going to be used.

Finally, Tensorflow offers high-level APIs that are easy to be used without a deep knowledge of TensorFlow. It also has low-level APIs that can be used when already familiarized with the framework.

OpenCV:

OpenCV as its name suggests is an open-source computer vision software library. It offers more than 2500 optimized algorithms that can be used in many areas of computer vision applications, such as face detection, track camera movement and more. The main API is written and optimized in C/C++ but also offers interfaces in Java and Python despite C++ and C. Interfaces exist in other languages also such as Matlab and C# and can run on the widely known operating systems (Windows, Linux, Mac OS, Android, iOS).

OpenCV aims on the real-time applications that is why it take advantage of multicore processing and has become very popular in the scientific community as it allows to quickly run demos or research projects [70]. It separates the functionalities it offers into two levels, one low-level that contains the functionalities addressing image processing. Such functionalities are noise reduction, filtering, edge detection, color conversion and more. The second level contains computer vision modules that are addressed as high level. In this level modules for optical flow and camera calibration are included.

Sockets:

Sockets are software structures that serves as endpoints for sending and receiving data from the network. They are defined by an IP address and a port. There are types of sockets such as connectionless sockets that uses the UDP where packets are sent without a specific order so many packages sent from one machine to other packages will arrive in a random order or maybe not arrive at all. There are sockets using TCP or other protocols which ensure that the packages will arrive in the correct order to their destination. Additionally, raw sockets exist that are not bound to specific transfer protocols and do not require mandatory headers for the packages to be sent.

Socket programming used in a client server model usually creates a socket on the listening state on the machine that describes the server and waiting for a client to request a service. For many clients, a server creates a socket for each client connection. The transportation of the data is through a stream of bytes.

5.2 Component Implementation

In this section the implementation of the main components and functionality of the toolset are presented in depth. It is separated into two sections, the first describing the Unity components and their functionality and the second section the service side that was implemented in python programming language. Code snippets will be presented for both modules to give an insight of how exactly those modules were implemented.

The implementation of the toolset was developed taking always into consideration the SOLID [71] principles. This was because those principles provide the means to keep the code clean and easy to maintain. Using those principles requires careful thought as on how a class or a set of classes will be implemented but when the core implementations are made, the addition of classes and functionality becomes quick and almost effortless. SOLID is the acronym of five principles that help on design and implementation of software applications. Those principles are used on agile development.

The five principles of the SOLID acronym are the following:

- *Single-Responsibility Principle: "A class should have one, and only one, reason to change".*

  This principle describes that a class should have only one responsibility, meaning that a class should be responsible for one thing only. Following this principle is beneficial as it reduces unexpected side-effects, makes code easier to expand and maintain. It also reduces the possibility of bugs due to a change and reduces the risk of a change to affect other parts of the code. Furthermore, if a bug arises it is easier to be found as each class has a clear task to perform and a malfunction on this task will be clear if it is not coupled together with other tasks. To summarize, Single Responsibility Principle advantages are on testing the code, lower coupling as classes will have less dependencies and better organization of the project as it is easier to search for classes responsible for a task rather than searching a specific task inside of a large class.

- *Open-Closed Principle: "Software entities (classes, modules, functions, etc.) should be able for extension, but closed for modification".*

  Aiming on the avoidance of making changes to an entity that will possibly change its behavior that will lead to the requirement of the depending entities to adapt to those changes. This principle uses interfaces to describe the general guidelines of the classes that will implement it, this way interfaces remain closed to modifications. The implementations of those interfaces are independent and the only code they might share is what is inherited from the interface. This principle introduces an extra level of abstraction on the code that offers loose coupling. Using this principle allows to provide a new implementation of the interface to an existing block of code and maybe the method that instantiates this implementation. This principle can be used with Dependency Injection or Reflection to replace the instantiation methods of the specified implementation.

- *Liskov Substitution Principle: "Derived classes must be substitutable for their base classes".*

  Based on the principle of substitutability in object-oriented programming that states that if S is a subtype of T, then the objects of type T can be replaced with objects of type S without altering any properties of the program. In code this means that each subclass that overrides methods of a superclass should have the same inputs and return types. This principle is difficult to be implemented as a lot of checks have to take place to ensure the correct functionality of the code and that the involved classes really comply to the principle. This principle has received a lot of criticism as it contradicts the case in object-oriented programming where a superclass is abstract without implementation. By that, the comparison of the implementations of supertype and subtype is introduced, which should not be a case. It has stated that the substitution terminology for this principle was used only as an informal rule and not as a standard definition for the principle. Aside from the criticism of the strict definition of the principle, the advantage of practicing it is that offers consistency on using parent classes or its subclasses without any errors.

- *Interface Segregation Principle: "Make fine grained interfaces that are client specific".*

  The goal of this principle is to detach the dependencies that a class or a method does not use. It is similar to the Single Responsibility Principle as both try to reduce the required changes and unexpected bugs on the code. The concept is to split interfaces and classes into smaller ones so a client will know about the methods they are interested in. It adds to the prevention of coupling and simplifies the code. It is easy to violate this rule as a program progresses and extra functionality is added so it needs attention when an extra feature is added so that it would be addressed accordingly. Each feature added should be implemented in a different interface if it is of a different responsibility to ensure that coupling dependencies will be avoided.

- *Dependency Inversion Principle: "Depend on abstractions, not on concretions".*

  This principle states that high-level modules (modules that execute an action with a tool) and low-level modules (the tool to execute an action) should depend on an abstraction and not be coupled with each other. High-level modules should be reusable and changes on the low-level modules should not affect them. Dependency inversion principle has two parts:

  1. High-level modules should not depend on low-level modules. Both should depend on abstraction.
  2. Abstractions should not depend on details. Details should depend on abstractions.

This principle does not change the direction of the dependency, but it separates it between high-level and low-level modules by the abstraction between them. Also, both modules should not know how the abstraction between them works. However, modules should know the specifications of the abstractions and comply to them. This allows to change components without affecting other classes in the code as long as the abstraction remain the same. Adapting the Open/ Closed Principle and the Liskov Substitution Principle also leads to adapting the Dependency Inversion Principle. A way to implement the Dependency Inversion Principle is through the Dependency Injection technique.

5.2.1 Unity

- **Frame Capture Algorithm**

To capture the image from the device's camera and adjust it on a format that makes it capable to process certain actions should be made. Such actions are the attempt to capture the latest image from the camera, define the conversion parameters of that image to the desired ones, allocate a buffer with the size of the final desired image. Then using the conversion parameters load the captured image on the buffer created before and discard the captured image. Now the buffer can be loaded on a texture2D Unity class and encode it to JPG with a given quality. Finally dispose of the buffer that was holding the image data and return the encoded JPG image. This process is depicted in Figure 16.

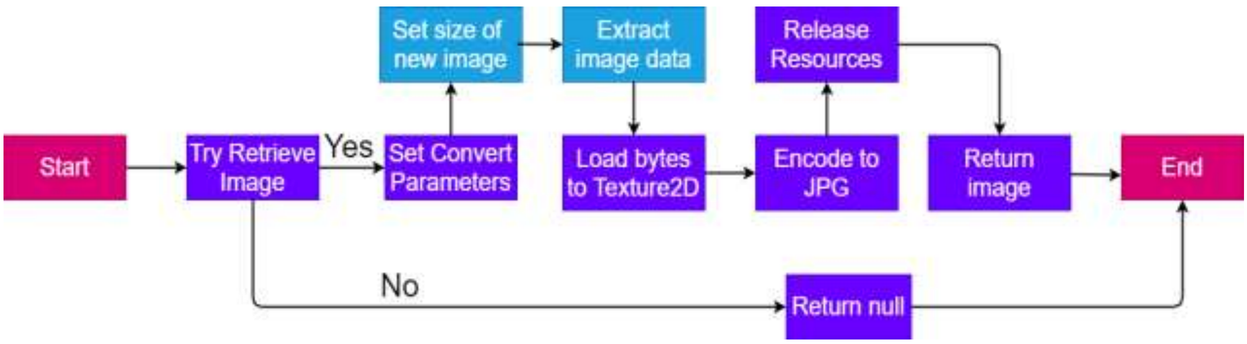


Figure 16. Frame capture process.

- **Gesture Manager Implementation**

The *GestureManager* class allows to access the defined gestures in a more automated way. It contains two functions that compose the functionality of the class, a list that contain all the gesture types and a ray selector interface. On initialization, *GestureManager* will try to retrieve an *ISelector* component from the *GameObject* that is attached to. If there is no *ISelector* to be retrieved the *GestureManager* will create a default component to be set.

The user can call one of those functions and enter as an argument an enumerated value that represents the gesture that the user wants to call. This value will get checked against the type of each gesture on the list and it will return the first gesture that matches with the specific type that the value contains. This function is presented in Figure 17.

```
public IGesture SelectGesture(GestureType type)
{
    IGesture[] runner = { new GrabThumbIndexGesture(), new PointIndexGesture(), new PointMiddleGesture(),
    new PointPinkyGesture(), new PointRingGesture(), new PointThumbGesture()};

    return runner.FirstOrDefault(key => type == key.type);

}
```

Figure 17. *SelectGesture* method code snippet.

The second function will additionally perform a check for interaction. After the gesture is retrieved instead of returning it as is, it will call the check function of the retrieved gesture and it will return the object that an interaction will occur or the null value. This function is shown in detail in Figure 18.

```
public Transform TryToPerformGesture(GestureType type)
{
    IGesture[] runner = { new GrabThumbIndexGesture(), new PointIndexGesture(), new PointMiddleGesture(),
    new PointPinkyGesture(), new PointRingGesture(), new PointThumbGesture()};

    var gesture = runner.FirstOrDefault(key => type == key.type);
    if (gesture == null)
        return null;

    ray = gameObject.GetComponent<ActionHand>().Get_pointRays();

    return gesture.Check(ray, select);
}
```

Figure 18. *TryToPerformGesture* method code snippet.

In Figure 19 the virtual hand is depicted interacting with a cube object. This interaction shows that the index finger is pointing at the cube object. The green line on the picture represents a ray cast from the index fingertip and in the forward direction.

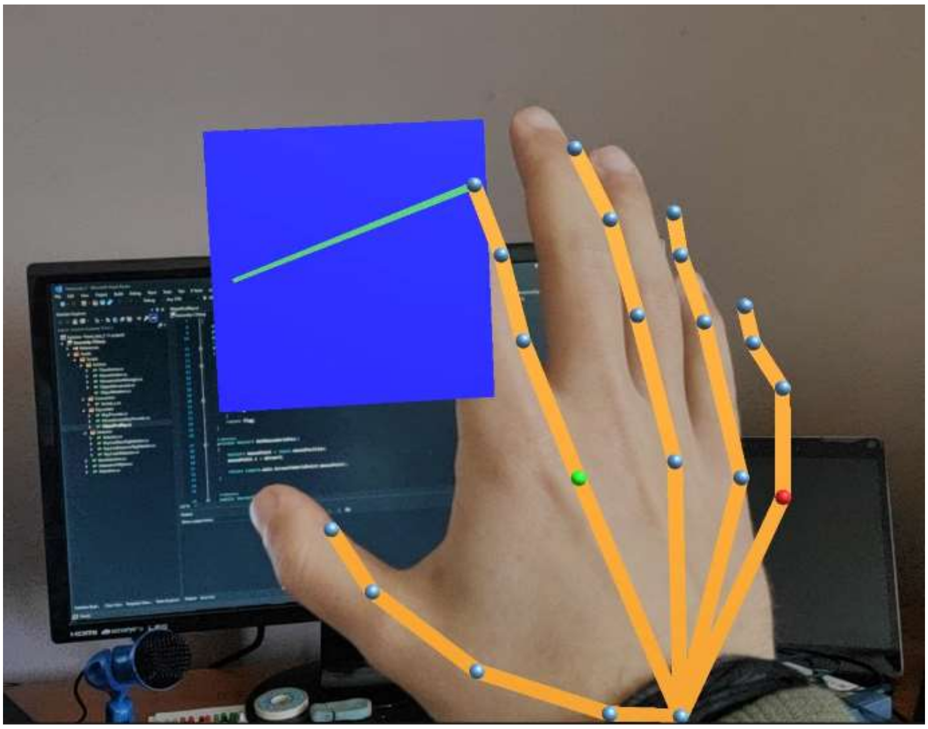


Figure 19. Virtual-hand interaction.

- **Virtual Hand implementation**

The hand is represented by a sphere *GameObject* and the bones are represented by *LineRenderers*. The hand contains a list with 21 *Vector3* variables. When the program starts it initializes those sphere *GameObjects* in predefined starting points to place the hand in the middle of the screen. *LineRenderers* are also attached to the sphere *GameObjects*. Functions are implemented to create and update the position of the hand. The function to update the hand receives the recognized hand points as screen coordinates and converts them to *Vector3* objects, then those *Vector3* objects are used to update the position of the hand by assigning those values on the corresponding sphere *GameObject*. In Figure 20 and Figure 21 is presented the virtual hand in two states. In Figure 21 the retrieved points from the service are presented in the scene as *GameObjects* and in Figure 20 those points are presented connected with the *LineRenderers* that form the complete visualized hand for the toolset.

Beside the visual part of the hand, there should be implemented features that will render the hand capable of interacting. For this purpose, on each *GameObject* representing a hand point a ray caster is assigned to provide the ability to check against it if there is an object that is capable of interaction. This functionality is implemented in a different class that also wraps all the functionality from the visual representation class of the hand.

Figure 21. Hand keypoints.

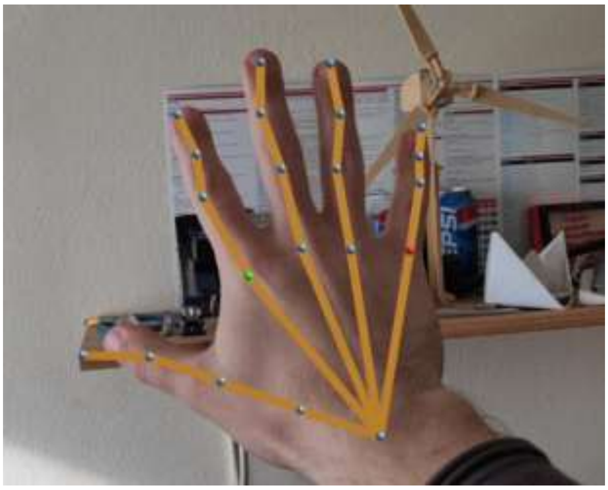


Figure 20. Virtual hand.

- **Actions Implementation**

The actions are implemented to provide some basic interaction and ready to be expanded. For this functionality two interfaces were created. The first interface contains the definition functions that need to be implemented by the classes that inherit from it. The second interface inherit from the first, but it has a generic type of argument that needs to be defined on the class that will inherit from it. An abstract class is implemented so that every action would inherit from that class, except for the classes that require an explicit implementation of the generic type argument. The abstract class is shown in Figure 22, it provides a default implementation for the function that requires a generic type as argument so if one action does not require the usage of such a function it will not have to implement it.

```
public abstract class HandAction : IHandAction
{
    9 references
    public abstract void PerformAction(Transform transform, GameObject[] obj);

    2 references
    public virtual void PerformAction<T>(Transform transform, GameObject[] obj, T type)
    {
        PerformAction(transform, obj);
    }
}
```

Figure 22. *HandAction* Abstract Class.

There are two actions implemented on the toolset. The first is the move action that is responsible for moving an object based on the hand movement. To perform the action a function named *PerformAction* should be called. This function has as arguments the position of the object that is about to get moved and the array the contains hand points. On the move action the second function provided by the abstract class is also overridden. The generic type is defined as an enumerated value that will limit the movement of the object to a specific axis. An example is presented in Figure 23.

```
public override void PerformAction<T>(Transform transform, GameObject[] obj, T type)
{


    if (typeof(T) == typeof(MoveAction.ActionType))
    {
        ActionType t = (ActionType)Convert.ChangeType(type, typeof(ActionType));
        var move_point = new IndexFingerTip();
        switch (t)
        {
            case ActionType.X:
                transform.position = new Vector3(
                    obj[move_point.GetHandPoints()[0]].transform.position.x * 1.2f,
                    transform.position.y,
                    transform.position.z);
                break;
            case ActionType.Y:
                transform.position = new Vector3(
                    transform.position.x,
                    obj[move_point.GetHandPoints()[0]].transform.position.y * 1.2f,
                    transform.position.z);
                break;
```

Figure 23. *MoveAction* Class method code snippet.

An example of the second action process is presented in Figure 24, that shows the hand to hold a pencil. This happens through an action class named *BindAction*. With this action the pencil is bound to the hand on the base hand point of the index finger and the second to the base hand point of the thumb finger. Furthermore, the pick of the pencil is bound to always "look" at the fingertips of thumb and index fingers for movement and rotation.

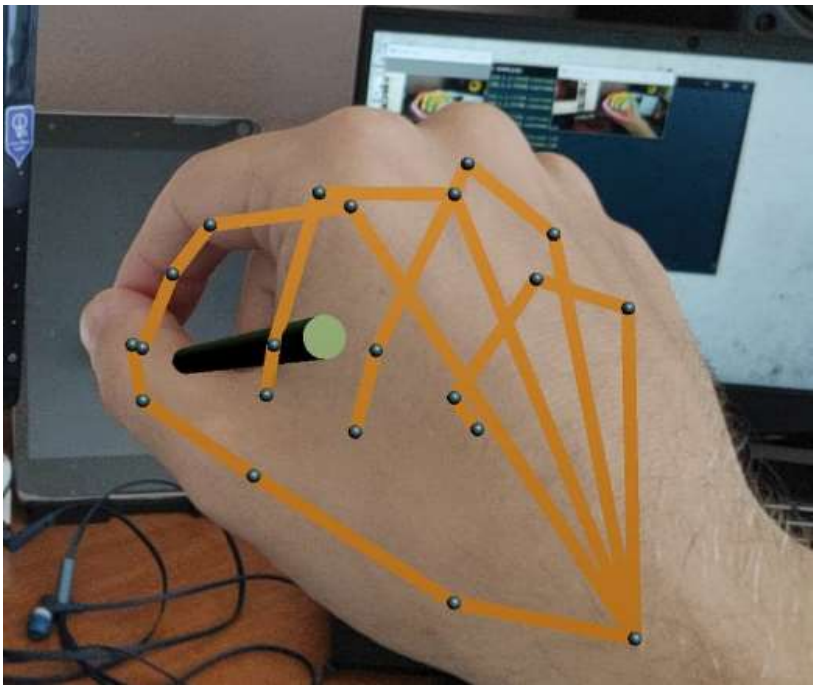


Figure 24. *BindAction* interaction.

- **Connection implementation**

A simple socket client was created to send the captured image to the service. The instance is unique in the lifetime of the application, meaning that no other instance of the same client can be created. The header size is set to 1024 bytes as this is the largest amount of bytes that can be transferred in a single packet. On the first 1024 bytes there is information about the image that will be sent. The most important information transferred in this first packet is that the payload contains an image and the size in bytes of that image. To send the image a function was implemented called *SendMessage*, it takes as an argument the captured frame from the camera and sends it to the service. To use the client on the application is achieved through a single line of code presented in Figure 25.

```
response = SocketClient.GetInstance.SendMessage(msg);
```

Figure 25. *SocketClient* method call code snippet.

As a response to the images sent by the client, the service sends a list of floats formatted in a string. This string has to be deserialized and stored in a list of float arrays so that each item in the list represents the (x, y) coordinates of a hand point. The function responsible to deserialize the received message from the service performs the subsequent steps. Firstly, it makes a check that the values received are not null. After the check, using the text functionalities provided by C#, separation on the received string based on the defined characters is performed. Finally, it takes every separated string and converts it to a float number and stores its value on a new created list in the right format for the system to work. This is presented in Figure 26.

```
public static List<float[]> Deserialize(string data)
{
    if (data.Equals("null"))
    {
        return null;
    }
    var list = new List<float[]>();

    char[] separators = { '[', ']', ',', ' ' };
    string[] float_str = data.Split(separators, StringSplitOptions.RemoveEmptyEntries);
    for (int i = 0; i < float_str.Length; i += 2)
    {
        float[] tmp = new float[2] {float.Parse(float_str[i], CultureInfo.InvariantCulture.NumberFormat),
          float.Parse(float_str[i + 1], CultureInfo.InvariantCulture.NumberFormat )};

        list.Add(tmp);
    }
    return list;
}
```

Figure 26. Message deserializer code snippet.

- **Menu button implementation**

  Every application is customary to provide a menu. The problem with augmented reality is that the provided UI on unity is working in a specific way that is not available for the augmented reality app we build. We made a button prefab that can be used by the virtual hand of the toolset. It consists of a cube object and a UI TEXT component to display the functionality the specific button will provide. A sample UI button is presented in Figure 27.

Figure 27. Sample Menu Button icon.

The rotation of the button is constraint to the rotation of the main camera of the application. Usually, the menu buttons are on the top left corner of the screen. To keep the button in this place we add some lines of code that will help to achieve this task. There, the position of the button relative to the screen is given and converted in world coordinates, then this position is applied to the menu button object. Then, the direction of the device is computed and used to apply it on the object as a Quaternion rotation. This process is showed in Figure 28.

```
Vector3 worldPoint = Camera.main.ScreenToWorldPoint(new Vector3(250, Screen.height - 150, 17));
MenuUI.transform.position = worldPoint;
Vector3 direction = Camera.main.transform.position - MenuUI.transform.position;
Quaternion rotation = Quaternion.LookRotation(-direction);
MenuUI.transform.rotation = rotation;
```

Figure 28. UI components screen follow code snippet.

- **Ray casters**

Ray casters implementing the *IRayProvider* interface provide the means for the interaction with virtual objects. In the classes, a method is implemented that returns a ray pointing in a direction. Three rays were implemented so far for the toolset: rays pointing forward and up from the hand point into the world space and a ray that takes the hand point from the screen coordinates. Code snippet presenting the forward ray is shown in Figure 29.

```
public Ray CreateRay(GameObject obj)
{
    return new Ray(obj.transform.position, obj.transform.forward);
}
```

Figure 29. Creation Forward Ray code snippet.

In Figure 30 the ray from the screen points is presented, where it shows the conversion of the object's conversion from world space coordinates to screen point and then into a ray.

```
public Ray CreateRay(GameObject obj)
{
    var pos = Camera.main.WorldToScreenPoint(obj.transform.position);
    return Camera.main.ScreenPointToRay(pos);
}
```

Figure 30. Create Screen point Ray code snippet.

5.2.2 Python

On the service part of this toolset written in Python, a simple socket server was implemented. The server is responsible for the image acquisition from the Unity application. For this purpose, the standard scenario of how a message should be send is through sockets. Sending and receiving messages through sockets is in bytes format, so the incoming message that contains the image is in bytes format. Using the OpenCV and NumPy libraries the image is converted back from bytes to a standard image format that can be processed and manipulated.

This image is further processed as of rotation, size, and aspect ratio. The image has to be flipped as it gets received in a reflected way. The aspect ratio of the image acquired directly from the camera of the mobile phone is not the same as the one on the displayed screen of the mobile phone, so the image should be matched on the mobile phones' viewport to have the desired outcome. Finally, the size of the image is processed to match the required size that the neural network requires as input. Then the image is sent to the hand tracking module to detect the hand points and store them on a list. The points of this list are resized on the size of the mobile device's viewport. This list will be sent back as a response to the mobile device.

The initialization of the socket server is done by the lines of code presented in Figure 31. There are declarations of the socket type and the address family on the first line. Then the socket layer is given and allowing to reuse a local socket before timeout. This prevents errors that may occur if there is not enough delay between calls. On the third line there is a bind to the IP of the machine with a predefined port to act as the endpoint for the clients. Finally, the server is put on listening state waiting for clients to request services.

```
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

server_socket.bind(('', PORT))
server_socket.listen()
```

Figure 31. Server Initialization code snippet.

In Figure 32 is presented the process of receiving a message from a client. Firstly, the header is retrieved to get the overall length in bytes of the message. Then in a *while* loop it starts receiving the payload until it has received the same number of bytes as specified in the header. After this, the image should have been received correctly and an attempt to convert from bytes back to a JPG image is initialized. In the end, when the image has been successfully converted to a desired format, it is returned to the main flow of the program to further be processed.

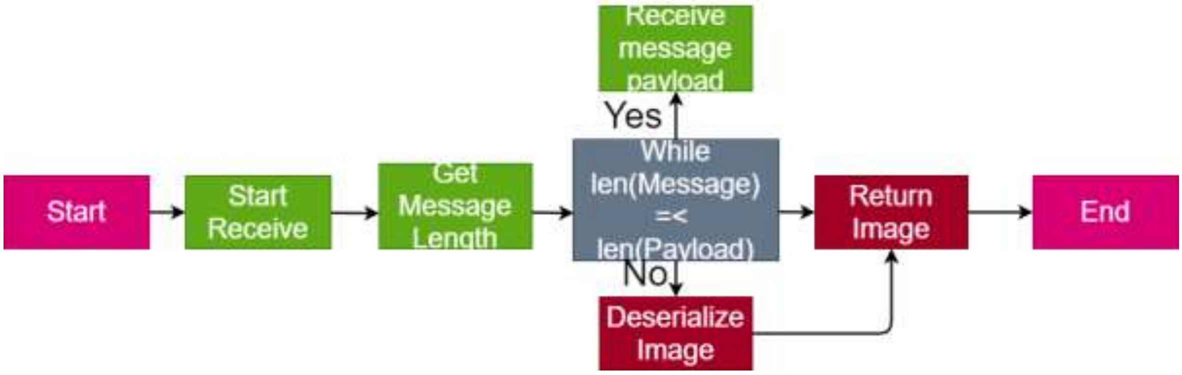


Figure 32. Message receive Flow Diagram.

In Figure 33 is presented a code snippet that describes how the points recognized from the hand tracking model are resized to the size of the mobile phone screen. The *coords* array contains the (X, Y) coordinates of each hand point. For each hand point the recognized pixel from the processed image is converted back to the actual pixel that needs to be displayed on the mobile phone.

```
send_coords = np.zeros(shape=(21, 2))
for joint_num in range(FLAGS.num_of_joints):
        send_coords[joint_num][0] = int(coords[joint_num][0] * ( Mobile_Width / full_img.shape[1]))
        send_coords[joint_num][1] = int(coords[joint_num][1] * ( Mobile_Height / full_img.shape[0]))
return send_coords
```

Figure 33. Hand keypoints resize code snippet.

The hand tracking algorithm should be slightly modified to be used with the rest of the toolset. This means that a method should be used to retrieve the recognized points. There is no need to change the implementation of the already existing code. The creation of a function to wrap the main functionality of the algorithm with a return value of the outcome is enough. The reason for the need of this addition is to allow the service to easily call the functionality of the hand tracking algorithm while keeping the different functionalities separated that results in a more clean and readable code.

Chapter 6 - Results

The outcome of the toolset has shown potential on future development and expansion of it. To prove that the toolset can be used as a development tool for the interaction on augmented reality applications some simple applications were created to test how the toolset can be used, how it performs, the behavior of the toolset at runtime and the overall impression of the toolset throughout the development, test, maintenance, and usage processes.

6.1 Case studies

- **Puzzle application**

  To test the toolset the first application that was created was a kind of a puzzle game that consists of two pieces that have to be placed in the right place. The terrain of the puzzle is as shown in Figure 34, the cube has to be placed on the square on the floor and the sphere on the circle.

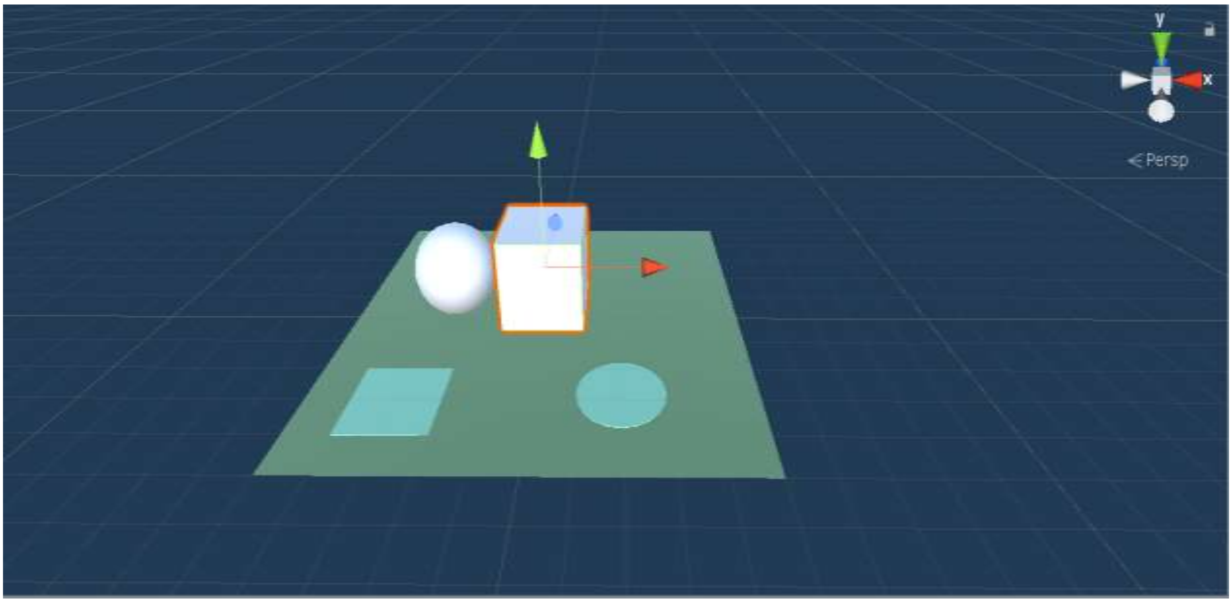


Figure 34. Puzzle example Terrain.

The detection of the cube and sphere, whether they are placed correctly or not, is achieved with ray casters on the square and circle objects on the terrain as shown in Figure 35.

Through this demo application the result revealed that it is not that easy to move an object to a specific location with high accuracy. The action to place the object on the right place was harder than with physical objects. Another difficulty on this application was on the usage of the plane detection provided by ARCore library. The difficulty was on the delay of the plane detection and in some cases the height of the detected plane was not as it should be.

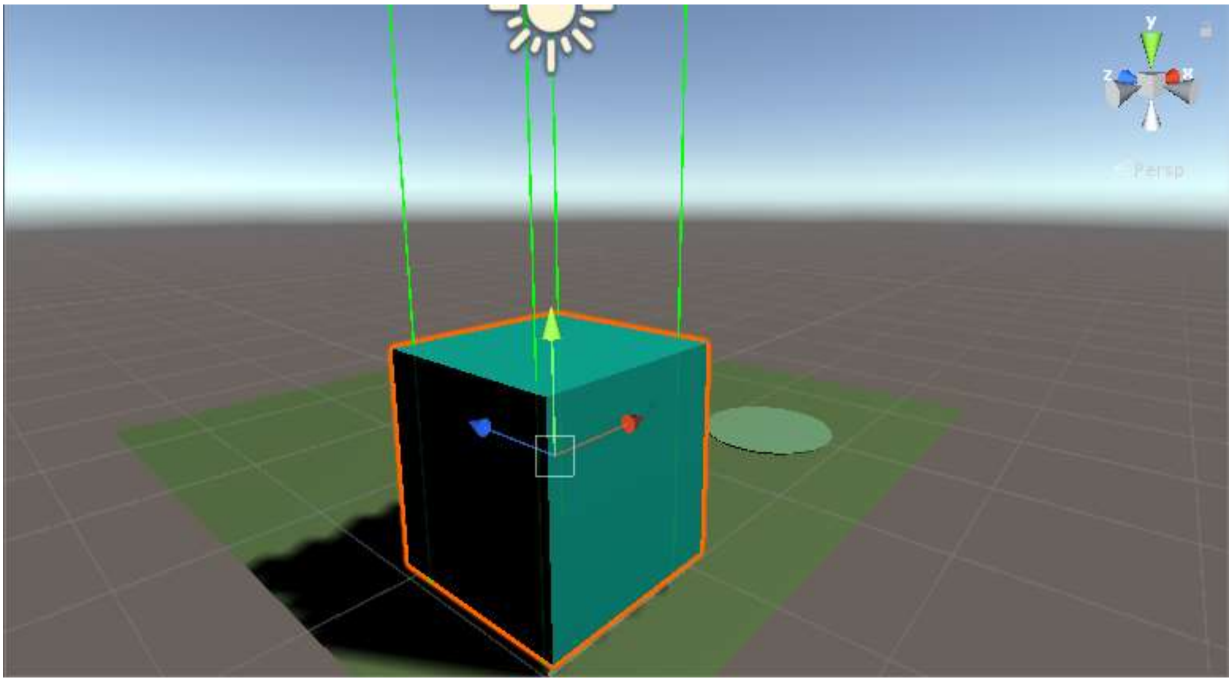


Figure 35. Puzzle example completion

- **Keyboard application.**

  A second application was created representing a virtual keyboard. There is only one octave as presented in Figure 36, but it was enough to prove that despite the performance on a virtual piano mid-air is a hard task to be achieved it was possible to be accomplished in a fundamental level. It is hard to play the piano on a high level of performance, but this application was created just to show that this concept can be achieved if more details are considered from the development team of the application.

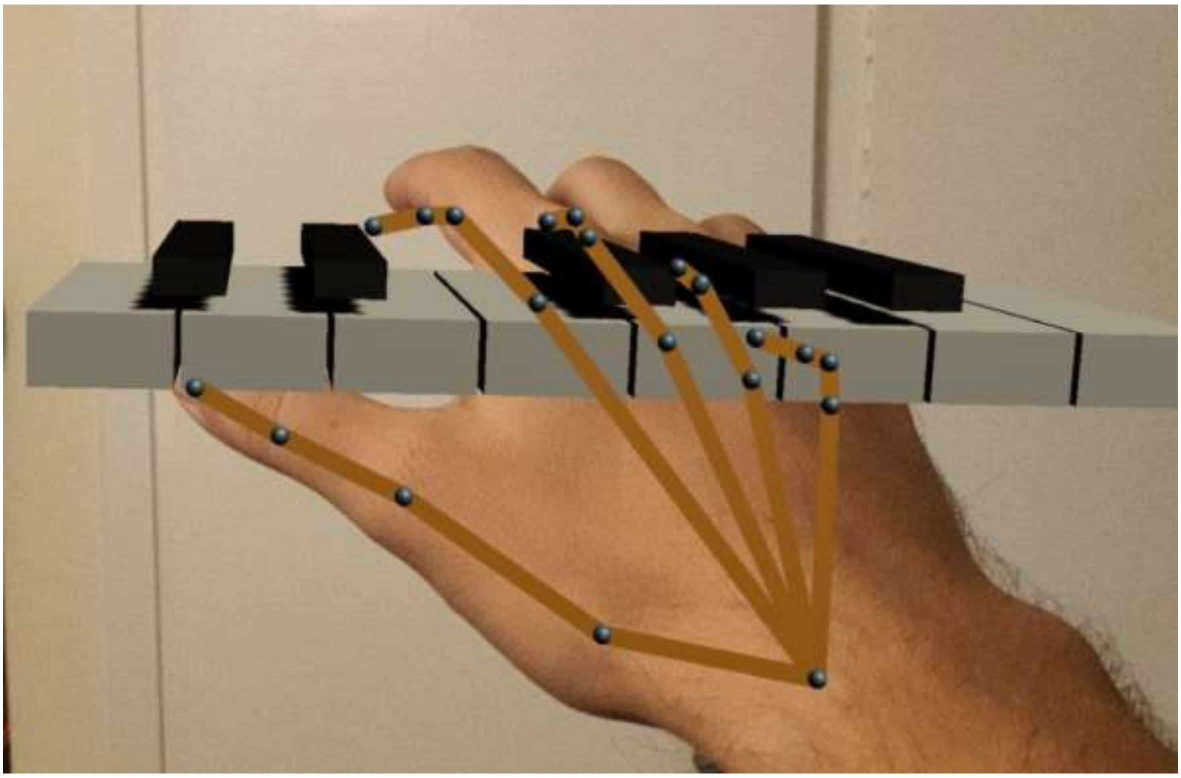


Figure 36. Keyboard example.

- **Painting application**

A third application has to do with arts also. Here providing a pencil and a canvas a user can draw shapes on the canvas using the virtual pencil. The pencil is bound on the virtual hand and moving with it. When the pencil is within the borders of the canvas it starts to paint with a predefined color. To stop painting there is a gesture that allows the user to move the pencil freely without painting. This application is just a demo again but there are many simple additions that can transform it to a full-scale painting application. In Figure 37 is shown the painting demo in the early stages. It has great potential and also a possible 3D painting in the environment of the end-user.

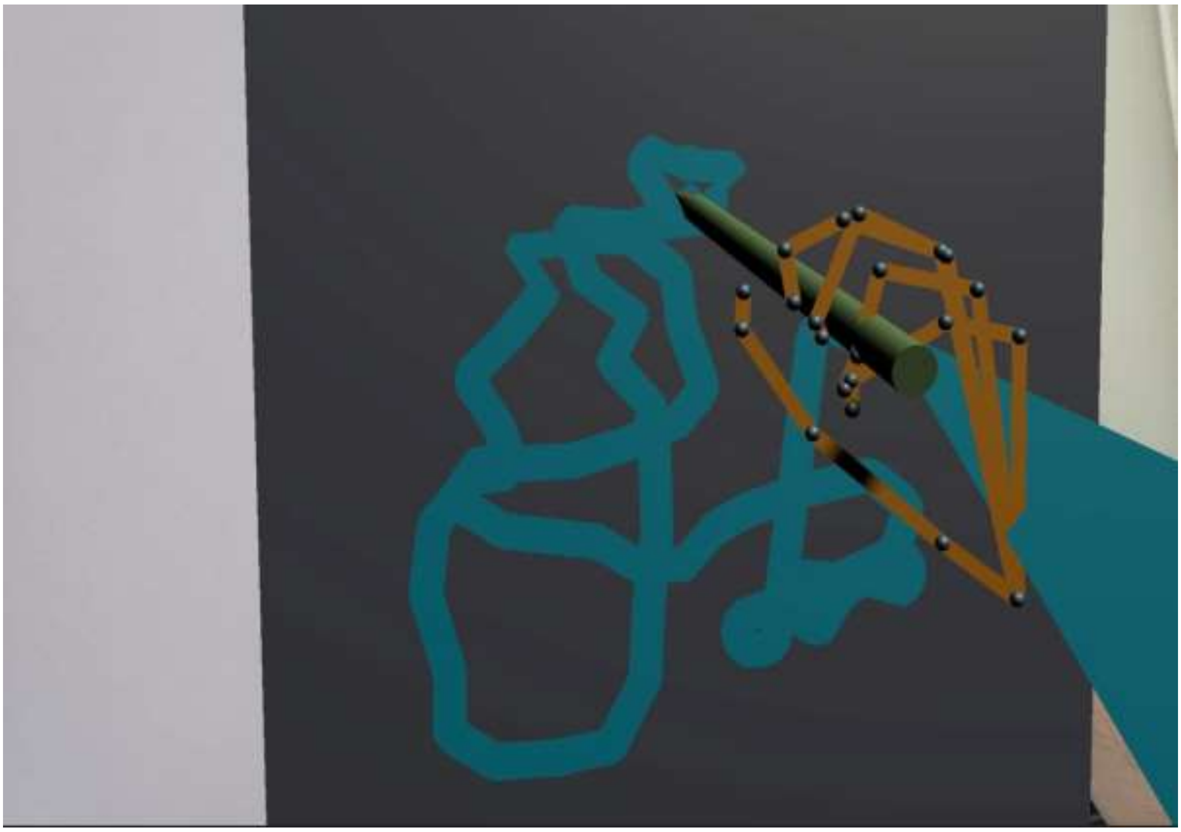


Figure 37. Painting example.

- **Sample application**

The last application in Figure 38 can be described as the standard application that is commonly used as a sample application in this field of study. Here the task is to grab and freely move a cube in the scene. This task can be achieved with ease as it is not a task that requires high accuracy on the movement.

- **Word Hangman Game**

A game was developed also using this toolset as a more motivational way for students to learn a foreign language [72]. This work has been accepted for presentation at the 23rd Human Computer Interaction International Conference (HCII 2021) a virtual event to be held July 24-29, 2021 and the full paper is reported in the Appendix.

From the applications tested, observations were made as on how the interaction happens. One observation is that the absence of haptic feedback creates a misconception regarding the positioning and the force that will be put on the hand. Another observation is that because of hardware restrictions resulting on short delays of the application the virtual hand or the interacting objects will not move alongside the real hand and that is another reason for thew user to be confused. The tasks that required high accuracy were found hard to complete and further research is required on this topic so a better result will be accomplished in the future.

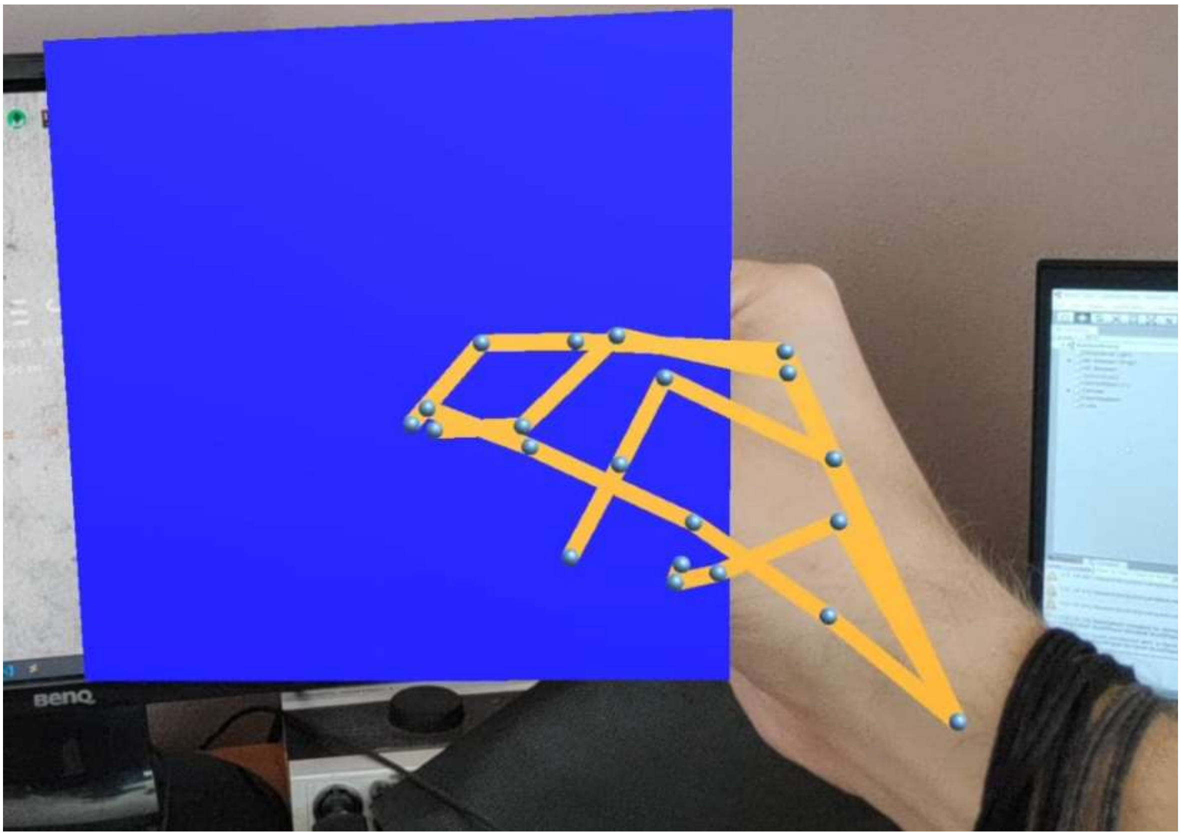


Figure 38. Basic movement example.

Other tests for the toolset were conducted on the service side. There, the ability to change the hand tracking algorithm with a simple and easy way was tested as this feature is one of the core features that this toolset provides. Various hand tracking algorithms checked how the service corresponds to the changes and what actions it requires to be taken to ensure that such algorithms will be able to change in a simple way.

Most of the algorithms that were tested were using a neural network to provide the detected hand points. Some of the neural network models tested were just for the hand tracking without the ability to retrieve the pose of the hand. These neural network models were used to improve the performance of the models that could retrieve the pose of the hand, or alongside computer vision techniques that retrieves hand points. Some of the models tested are the following:

- Convolutional pose machines [73].

  This was the most used model as this project offered a great implementation and with many extra functionalities as the Kalman filters that helps for a better prediction of the movement of the hand. The drawback of the model is that it is computationally heavy to run. Running this model in real time requires a desktop computer or a laptop with at least a mid-range gaming graphics card.
- Accurate Hand Keypoint Localization on Mobile Devices [74].

  This model was light enough and could be executed on a mid-range laptop with a good performance. The drawback of this model was that extra functionality had to be implemented to achieve better results as it would easily be influenced from the environment.
- HandTrack [75].

  This model was used to assist the other models to perform better but numerous computations were required, and a large amount of delay was introduced. This model was also used alongside computer vision algorithms to provide a square indicator on the complete image that the hand is located in it, to make the algorithms easier to recognize and subtract the hand from the background.

Some of the computer vision algorithms that used together with the models aforementioned are as already stated the Kalman filters that use measurements obtained over time to produce estimates of unknown variables. With this technique the outcome of the hand points was more accurate than running the neural networks without it. Other algorithms used for the extraction of hand points given a sub image containing the hand only provided using the Handtrack model described previously. One of such algorithms was the Optical flow that tries to observe the motion of objects in a scene. This technique required a static background and many configurations that made it hard to be adapted for this toolset. Another technique tested was subtracting the remaining background from the image provided by the Handtrack model and then calculating distances from a center point to the farthest points of the resulting image that contains only the hand. In this way the fingertips of the hand were extracted and can be used to perform tasks. This method also was not very stable and as it did not offer better optimization on the computational power required for the hand tracking, this algorithm was also not preferred.

6.2 Performance

The overall performance of the application as fathomed, was linked to the performance of the hand tracking algorithms. So, the performance of the toolset is highly related to the selection of the hand tracking algorithms. This performance restriction can be surpassed by using a desktop computer with more computational power to support such models or even more demanding ones.

A second performance issue is this of the network connection, it requires a good network connection that in most cases everyone has access to nowadays. Of course, there are still some cases where a network connection is not available and from the mobile phone network access there is usually a limitation on the data a user can spend, or an extra payment for the data is required. This problem is also going to be solved by the adaptation of 5G networks that promises even faster and more accessible internet from everywhere.

Another performance limitation occurs from the ARCore library. This does not cause an actual delay on the application, but some tasks such as the plane detection that requires several resources and some extra actions to be taken, can detune the user from the starting goal. Most of these limitations occur due to the real-time understanding of the environment from the augmented reality libraries that run those tasks in parallel with the rest of the tasks of the application.

All tests were performed on a laptop with an Intel 9^{th} Gen i7 CPU, Nvidia GTX 1050 4GB GPU, and 32 GB RAM and on Xiaomi Pocophone F1 smartphone. The requirements of those devices do not consist of the best hardware available in the market but rather a standard specifications of a common desktop computer and a mobile phone. The above specifications shows that the toolset does not require high-end hardware to run and can be applied in the majority of the devices available in the market.

Chapter 7 - Future Work

The aim of this thesis was to provide a toolset that facilitates the process of interaction in augmented reality applications while the toolset remains open to expand and be versatile on the selection of which components will be used. A great deal for the toolset was to provide the ability to the user to change the hand tracking algorithm without breaking the rest of the toolset. This task was important because such algorithms everyday become better and require less resources to run. So, the opportunity given to the user to change such algorithms provides a great advantage as the user can update the toolset without relying on an update of the toolset for a new hand tracking algorithm.

The way that the toolset designed is addressed to a variety of users and with a variety of skills. It is not a necessity to be a professional programmer to use the toolset, but a little bit of programming knowledge is required at this stage. A designer can use the components provided to create an application, but it will be limited to some default functionality. Furthermore, it is a nice way for hand tracking researchers to test the models they develop on a real-world application.

As the outcome suggests, the toolset does not require extreme equipment and hardware devices to be able to run in a satisfying rate. This makes it accessible to a variety of users as the required hardware can be easily found.

Further development of the toolset is required so that it may become a more complete development tool able to perform in the most demanding situations. The toolset can expand even more to provide extra functionalities beyond the aim of this thesis. Additions and improvements of the toolset are listed next, presenting simple modifications to the already implemented functionalities but also new functionalities that will enhance this toolset.

- *Support many users simultaneously on the service.*

  Provide the means to allow many users to connect to the service and use it in many devices at the same time. This will require the implementation of registering each device that registers to the service so that the service will know where to send the corresponding data. It also needs a better handling of the processes running to optimize the performance when used heavily. The support of many users will lead to the need for a better hardware to host the service as the computations will be increased.

- *Support remote collaboration of two or more players.*

  A great expansion of this toolset will be the ability to allow two or more players to collaborate in the same scene. To achieve this, the support of many users from the service is required, but also extra data must be exchanged, such as what action each player does and how it affects the scene. Those extra data must also be organized in classes to be clear each time what data is retrieved.
- *Support more hardware devices and software libraries.*

  Right now, the toolset supports mobile devices that are augmented reality enabled. The goal is to provide the means for more mobile devices to use the toolset as well as newly emerged augmented reality glasses, simple cameras and more.
- *Create a better interface for users without any knowledge of code.*

  It is a very important step to provide a better user interface for the creation of a game or an application with the toolset without the need for righting code. This needs a well thought out design to be practical for the user and expose the right variables on checkboxes, sliders, and other user interface components. Also, attention is required on the implementation of such components to guarantee that the values available on those components will not lead to a conflict with the acceptable values on the code.
- *Expand the actions that the toolset provides.*

  The toolset provides some basic actions in its current state, such as the grab action. In the future more actions are about to be implemented to give more options on the interaction means.
- *Create a better-looking hand model for the virtual hand.*

  The model of the hand can be described as a very basic model just to act as a guide on where the application has detected the real hand. A better model for the hand, possibly a 3D model, is about to be introduced in next versions of this toolset. The improvement of the hand model will help the user to understand better how the hand moves and will be more pleasing to the eye.
- *Implement an inverse kinematic movement on the virtual hand to be more realistic.*

  With the improvement of the hand model, the movement of that model also has to be improved. This can be achieved by applying inverse kinematics for the movement of that model. In this way, some computations can be relieved from the service and introduced on the device. For the implementation of this technique attention must be focused on not over-loading the device with heavy tasks.

- *Implement extension methods to directly access functionality from components.*

  Using extension methods is a way of structuring functionality for a class type without modifying that class specifically. In this way of implementation, the extra functionality will benefit the programmers to access the methods in a more natural way and also allow them to expand functionality without directly modifying the class but rather just by introducing a new method of their liking.

- *Use inversion of control.*

  Inversion of control is a design principle in software engineering that allows the separation of the functionality of the program into independent modules and the extension of the programming language. By inversion of control, abstractions can be created that allow to control how a class will act by injecting the dependencies of that class and not initializing those dependencies inside the class.

References

[1] P. Milgram and F. Kishino, “A Taxonomy of Mixed Reality Visual Displays,” *IEICE Trans. Inf. Syst.*, vol. E77-D, pp. 1321–1329, 1994.

[2] T. Ronald, “Azuma A Survey of Augmented Reality.”

[3] D. W. F. van Krevelen and R. Poelman, “A Survey of Augmented Reality Technologies, Applications and Limitations,” *Int. J. Virtual Real.*, vol. 9, no. 2, pp. 1–20, 2010.

[4] M. Mekni and A. Lemieux, “Augmented Reality : Applications , Challenges and Future Trends,” 2014.

[5] “MICROSOFT HOLOLENS.” https://www.microsoft.com/en-us/hololens/ (accessed Oct. 20, 2020).

[6] A. Martinet, G. Casiez, and L. Grisoni, “Integrality and Separability of Multitouch Interaction Techniques in 3D Manipulation Tasks,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, pp. 369–380, 2011, doi: 10.1109/TVCG.2011.129.

[7] J. Polvi, T. Taketomi, G. Yamamoto, A. Dey, C. Sandor, and H. Kato, “SlidAR: A 3D positioning method for SLAM-based handheld augmented reality,” *Comput. Graph.*, vol. 55, pp. 33–43, 2016, doi: 10.1016/j.cag.2015.10.013.

[8] M. S. Hancock, F. D. Vernier, D. Wigdor, S. Carpendale, and Chia Shen, “Rotation and translation mechanisms for tabletop interaction,” in *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP ’06)*, 2006, pp. 8 pp.-, doi: 10.1109/TABLETOP.2006.26.

[9] D. Datcu, S. Lukosch, and F. Brazier, “On the Usability and Effectiveness of Different Interaction Types in Augmented Reality,” *Int. J. Hum. Comput. Interact.*, vol. 31, no. 3, pp. 193–209, 2015, doi: 10.1080/10447318.2014.994193.

[10] C. S. Yusof, H. Bai, M. Billinghurst, and M. S. Sunar, “A review of 3D gesture interaction for handheld augmented reality,” *J. Teknol.*, vol. 78, no. 2–2, pp. 15–20, 2016, doi: 10.11113/jt.v78.6923.

[11] S. C.-Y. Yuen, G. Yaoyuneyong, and E. Johnson, “Augmented Reality: An Overview and Five Directions for AR in Education,” *J. Educ. Technol. Dev. Exch.*, vol. 4, no. 1, 2011, doi: 10.18785/jetde.0401.10.

[12] E. Samset *et al.*, “Augmented reality in surgical procedures,” in *Proc.SPIE*, Feb. 2008, vol. 6806, doi: 10.1117/12.784155.

[13] E. Samset *et al.*, “Augmented reality in surgical procedures,” in *Proc.SPIE*, Feb. 2008, vol. 6806, doi: 10.1117/12.784155.

[14] B. Marr, “The 10+ Best Real-World Examples Of Augmented Reality,” 2020. https://bernardmarr.com/default.asp?contentID=1841&fbclid=IwAR2LWzesyF7_lb1u4qHJ49ek6KFT41dXmaC3M5bd6aAueddKPqrSQs6ZBYs.

[15] A. H. Behzadan, “ARVISCOPE: Georeferenced Visualization of Dynamic Construction Processes in Three-Dimensional Outdoor Augmented Reality,” p. 282, 2008.

[16] J. Paine, “10 Real Use Cases for Augmented Reality,” *Inc.*, 2018. https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html.

[17] M. Billinghurst, H. Kato, and I. Poupyrev, “The MagicBook: A transitional AR interface,” *Comput. Graph.*, vol. 25, no. 5, pp. 745–753, 2001, doi: 10.1016/S0097-8493(01)00117-0.

[18] J. R. Cooperstock, "The Classroom of the Future : Enhancing Education through Augmented Reality," *Proc HCI Inter 2001 Conf HumanComputer Interact.*, pp. 688–692, 2001, doi: 10.1.1.15.4355.

[19] E. Klopfer and K. Squire, "Environmental detectives-the development of an augmented reality platform for environmental simulations," *Educ. Technol. Res. Dev.*, vol. 56, no. 2, pp. 203–228, 2008, doi: 10.1007/s11423-007-9037-6.

[20] L. Kerawalla, R. Luckin, S. Seljeflot, and A. Woolard, "'Making it real': Exploring the potential of augmented reality for teaching primary school science," *Virtual Real.*, vol. 10, no. 3–4, pp. 163–174, 2006, doi: 10.1007/s10055-006-0036-4.

[21] B. E. Shelton and N. R. Hedley, "Using augmented reality for teaching Earth-Sun relationships to undergraduate geography students," in *The First IEEE International Workshop Agumented Reality Toolkit,* 2002, pp. 8 pp.-, doi: 10.1109/ART.2002.1106948.

[22] H. K. Mark Billinghurst, "Collaborative augmented reality," *Communications of the ACM Volume 45, Number 7*, pp. 64–70, 2002.

[23] E. Klopfer and S. Yoon, "Developing games and simulations for today and tomorrow's tech savvy youth," *TechTrends*, vol. 49, no. 3, pp. 33–41, 2005, doi: 10.1007/bf02763645.

[24] D. Liu, S. A. Jenkins, P. M. Sanderson, P. Fabian, and W. J. Russell, "Monitoring with head-mounted displays in general anesthesia: A clinical evaluation in the operating room," *Anesth. Analg.*, vol. 110, no. 4, pp. 1032–1038, 2010, doi: 10.1213/ANE.0b013e3181d3e647.

[25] F. Liarokapis *et al.*, "Web3D and Augmented Reality to support Engineering Education," *UNESCO Int. Cent. Eng. Educ. J. World Trans. Eng. Technol. Educ.*, vol. 3(1), no. 1, pp. 11–14, 2004.

[26] M. Billinghurst and A. Henrysson, "Mobile Architectural Augmented Reality," in *Mixed Reality In Architecture, Design And Construction*, X. Wang and M. A. Schnabel, Eds. Dordrecht: Springer Netherlands, 2009, pp. 93–104.

[27] H. S. Yang, "Realistic e-learning system based on mixed reality," p. 10, 2009, doi: 10.1145/1670252.1670255.

[28] V. T. Phan and S. Y. Choo, "Interior Design in Augmented Reality Environment," *Int. J. Comput. Appl.*, vol. 5, no. 5, pp. 16–21, 2010, doi: 10.5120/912-1290.

[29] S. K. Kim, S. J. Kang, Y. J. Choi, M. H. Choi, and M. Hong, "Augmented-reality survey: From concept to application," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 2, pp. 982–1004, 2017, doi: 10.3837/tiis.2017.02.019.

[30] E. S. Goh, M. S. Sunar, and A. W. Ismail, "3D object manipulation techniques in handheld mobile augmented reality interface: A review," *IEEE Access*, vol. 7, pp. 40581–40601, 2019, doi: 10.1109/ACCESS.2019.2906394.

[31] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, 2001, doi: 10.1109/38.963459.

[32] F. Zhou, H. Duh, and M. Billinghurst, "Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR," *2008 7th IEEE/ACM Int. Symp. Mix. Augment. Real.*, vol. 2, pp. 193–202, Sep. 2008, doi: 10.1109/ISMAR.2008.4637362.

[33] I. Rabbi and S. Ullah, "A Survey on Augmented Reality Challenges and Tracking Authors

Abstract : Keywords :," vol. 24, pp. 29–46, 2013, [Online]. Available: http://hrcak.srce.hr/file/150828.

[34] X. Wang, S. K. Ong, and A. Y. C. Nee, "A comprehensive survey of augmented reality assembly research," *Adv. Manuf.*, vol. 4, no. 1, pp. 1–22, 2016, doi: 10.1007/s40436-015-0131-4.

[35] "Optical Tracking Explained." http://www.ps-tech.com/optical-tracking-explained/ (accessed Oct. 26, 2020).

[36] "Positional tracking." https://xinreality.com/wiki/Positional_tracking (accessed Oct. 26, 2020).

[37] J. C. P. Cheng, K. Chen, and W. Chen, "Comparison of Marker-Based and Markerless AR: A Case Study of An Indoor Decoration System," no. July, pp. 483–490, 2017, doi: 10.24928/jc3-2017/0231.

[38] S. W. Jang, J. Ko, H. J. Lee, and Y. S. Kim, "A Study on Tracking and Augmentation in Mobile AR for e-Leisure," *Mob. Inf. Syst.*, vol. 2018, no. Figure 1, 2018, doi: 10.1155/2018/4265352.

[39] D. G. Low, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, pp. 91–110, 2004, [Online]. Available: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf.

[40] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Computer Vision -- ECCV 2006*, 2006, pp. 404–417.

[41] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Computer Vision -- ECCV 2006*, 2006, pp. 430–443.

[42] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

[43] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," *2013 IEEE Int. Symp. Mix. Augment. Reality, ISMAR 2013*, pp. 209–218, 2013, doi: 10.1109/ISMAR.2013.6671781.

[44] A. L. S. Ferreira, S. R. Dos Santos, and L. C. De Miranda, "TrueSight: Sistema de navegação para pedestres baseado na detecção e extração automática de landmark em smartphone android," *Proc. - 2012 14th Symp. Virtual Augment. Reality, SVR 2012*, pp. 91–99, 2012, doi: 10.1109/SVR.2012.14.

[45] J. Segen and S. Kumar, "Gesture VR," pp. 455–464, 1998, doi: 10.1145/290747.290822.

[46] R. G. O'Hagan, A. Zelinsky, and S. Rougeaux, "Visual gesture interfaces for virtual environments," *Interact. Comput.*, vol. 14, no. 3, pp. 231–250, 2002, doi: 10.1016/S0953-5438(01)00050-9.

[47] M. Baldauf, S. Zambanini, P. Fröhlich, and P. Reichl, "Markerless visual fingertip detection for natural mobile device interaction," *Mob. HCI 2011 - 13th Int. Conf. Human-Computer Interact. with Mob. Devices Serv.*, pp. 539–544, 2011, doi: 10.1145/2037373.2037457.

[48] J. Rekimoto, "Tilting operations for small screen interfaces," *UIST (User Interface Softw. Technol. Proc. ACM Symp.*, pp. 167–168, 1996, doi: 10.1145/237091.237115.

[49] S. J. Cho, C. Choi, Y. Sung, K. Lee, Y. B. Kim, and R. Murray-Smith, "Dynamics of tilt-based browsing on mobile devices," *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 1947–

1952, 2007, doi: 10.1145/1240866.1240930.

[50] A. Henrysson, M. Billinghurst, and M. Ollila, "Face to face collaborative AR on mobile phones," *Proc. - Fourth IEEE ACM Int. Symp. Symp. Mix. Augment. Reality, ISMAR 2005*, vol. 2005, pp. 80–89, 2005, doi: 10.1109/ISMAR.2005.32.

[51] A. Henrysson, M. Billinghurst, and M. Ollila, "Virtual object manipulation using a mobile phone," *ACM Int. Conf. Proceeding Ser.*, vol. 157, pp. 164–171, 2005, doi: 10.1145/1152399.1152430.

[52] A. Henrysson, M. Ollila, and M. Billinghurst, "Mobile phone based AR scene assembly," *Proc. 4th Int. Conf. Mob. Ubiquitous Multimedia, MUM '05*, vol. 154, pp. 95–102, 2005, doi: 10.1145/1149488.1149504.

[53] D. N. Ta Huynh, K. Raveendran, Y. Xu, K. Spreen, and B. MacIntyre, "Art of defense: A collaborative handheld augmented reality board game," *Proc. 2009 ACM SIGGRAPH Symp. Video Games, Sandbox '09*, vol. 1, no. 212, pp. 135–142, 2009.

[54] T. Ha and W. Woo, "An empirical evaluation of virtual hand techniques for 3D object manipulation in a tangible augmented reality environment," *3DUI 2010 - IEEE Symp. 3D User Interfaces 2010, Proc.*, pp. 91–98, 2010, doi: 10.1109/3DUI.2010.5444713.

[55] A. Ahmad, C. Migniot, and A. Dipanda, "Hand pose estimation and tracking in real and virtual interaction:A review," *Image Vis. Comput.*, vol. 89, pp. 35–49, 2019, doi: 10.1016/j.imavis.2019.06.003.

[56] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1862–1869, 2012, doi: 10.1109/CVPR.2012.6247885.

[57] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," *Procs. BMVC, Dundee, UK (August 29–September 10 2011)[547]*, 2011.

[58] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1106–1113, 2014, doi: 10.1109/CVPR.2014.145.

[59] E. Seol and G. Kim, "HandyTool: Object Manipulation Through Metaphorical Hand/Fingers-to-Tool Mapping," 2019, pp. 432–439.

[60] J. Cui and A. Sourin, "Mid-air interaction with optical tracking for 3D modeling," *Comput. Graph.*, vol. 74, pp. 1–11, 2018, doi: 10.1016/j.cag.2018.04.004.

[61] J. S. Kim and J. M. Park, "Direct and realistic handover of a virtual object," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016-Novem, pp. 994–999, 2016, doi: 10.1109/IROS.2016.7759170.

[62] H. Liu *et al.*, "High-fidelity grasping in virtual reality using a glove-based system," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 5180–5186, 2019, doi: 10.1109/ICRA.2019.8794230.

[63] H. Tian, C. Wang, D. Manocha, and X. Zhang, "Realtime Hand-Object Interaction Using Learned Grasp Space for Virtual Environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2623–2635, 2019, doi: 10.1109/TVCG.2018.2849381.

[64] Y. Wu and C. M. Wang, "Applying hand gesture recognition and joint tracking to a TV controller using CNN and Convolutional Pose Machine," *Proc. - Int. Conf. Pattern Recognit.*, vol. 2018-Augus, pp. 3086–3091, 2018, doi: 10.1109/ICPR.2018.8546209.

[65] J. Yu, J. Jeon, G. Park, H.-I. Kim, and W. Woo, “A Unified Framework for Remote Collaboration Using Interactive AR Authoring and Hands Tracking,” 2016, vol. 9749, pp. 132–141, doi: 10.1007/978-3-319-39862-4_13.

[66] S. Han and J. Kim, “A study on immersion of hand interaction for mobile platform virtual reality contents,” *Symmetry (Basel).*, vol. 9, no. 2, 2017, doi: 10.3390/sym9020022.

[67] P. Kyriakou and S. Hermon, “Can I touch this? Using Natural Interaction in a Museum Augmented Reality System,” *Digit. Appl. Archaeol. Cult. Herit.*, vol. 12, no. November, pp. 1–9, 2019, doi: 10.1016/j.daach.2018.e00088.

[68] “ARCore overview.” https://developers.google.com/ar/discover (accessed Oct. 28, 2020).

[69] M. Heller, “TensorFlow review: The best deep learning library gets better.” https://www.infoworld.com/article/3250165/tensorflow-review-the-best-deep-learning-library-gets-better.html (accessed Oct. 28, 2020).

[70] M. Mohamad, *A Review on OpenCV*. 2015.

[71] R. Martin, “Design principles and design patterns,” *Object Mentor*, no. c, pp. 1–34, 2000, [Online]. Available: http://www.cogs.susx.ac.uk/users/ctf20/dphil_2005/Photos/Principles_and_Patterns.pdf%5Cnhttp://scm0329.googlecode.com/svn-history/r78/trunk/book/Principles_and_Patterns.pdf.

[72] I. Logothetis, G. Papadourakis, I. Katsaris, K. Katsios and N. Vidakis, “Transforming Classic Learning Games with the Use of AR: The Case of the Word Hangman Game,” in Learning and Collaboration Technologies: Games and Virtual Environments for Learning, 2021, doi: 10.1007/978-3-030-77943-6_4.

[73] T. Ho, “convolutional-pose-machines-tensorflow.” https://github.com/timctho/convolutional-pose-machines-tensorflow (accessed Oct. 28, 2020).

[74] F. Gouidis, P. Panteleris, I. Oikonomidis, and A. Argyros, “Accurate hand keypoint localization on mobile devices,” *Proc. 16th Int. Conf. Mach. Vis. Appl. MVA 2019*, 2019, doi: 10.23919/MVA.2019.8758059.

[75] D. Victor, “HandTrack: A Library For Prototyping Real-time Hand TrackingInterfaces using Convolutional Neural Networks,” *GitHub Repos.*, 2017, [Online]. Available: https://github.com/victordibia/handtracking/tree/master/docs/handtrack.pdf.

Appendix

Transforming Classic Learning Games with the Use of AR: The Case of the Word Hangman Game

Ilias Logothetis, George Papadourakis, Iraklis Katsaris, Konstantinos Katsios, and Nikolas Vidakis(✉)

Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Heraklion, Crete, Greece
mtp166@edu.hmu.gr, {papadour,katsarisir,nv}@hmu.gr

AQ1

AQ2

Abstract. Augmented Reality in combination with playful learning is used to enhance students' engagement in blended learning environments. This paper presents an AR game with hand tracking & hand interaction that transforms the classic Hangman word game. The thematic word areas of the game are recycling and Covid 19, two areas that are of great interest today.

The game offers alternative interaction through a 3D hand model and 3D objects/letters. The player moves the 3D letters to complete words and collect paper cards with the words' image. These paper cards are then placed, in any order, by the student to create a picture story and a virtual maquette with AR objects is generated. This maquette allows player to create different picture-stories by shuffling the order of found words' paper cards. A field study has been contacted with two groups of students, one group used classic learning techniques and the other group used our AR game. Preliminary results of the study show that all students improved their language skills but only the group with the AR game evolve into being more active, involved, participative and engaged in the educational process and developed their creativity skills.

Keywords: AR game-based learning · NUI · Gamification · Blended learning · Creativity development

1 Introduction

Technological advancements in software and hardware and specifically the major improvements of the performance of smart devices have diversified our everyday life. One of the areas in which these technological changes are gradually becoming apparent is that of education. Augmented Reality (AR) is expected to diversify teaching in the coming years as AR applications blend physical and virtual worlds through a touch screen, a camera, or a head-mounted device [1]. AR technology is compatible and can be used in conjunction with the Game Based Learning (GBL) methods to boost educational processes. It can also help students both in the cognitive part and in gaining motivation for learning, but also in developing basic 21st century skills [2, 3].

P. Zaphiris and A. Ioannou (Eds.): HCII 2021, LNCS 12785, pp. 1–18, 2021.
https://doi.org/10.1007/978-3-030-77943-6_4

Traditional teaching methods do not make much use of new technologies with the result that students often lose interest in education due to the way they learn. While in their daily lives, students are accustomed to function and interact within environments with multiple stimuli, the traditional school remains a place with limited options [4]. The change to a new model is imperative to make learning more creative and interactive by introducing new visual forms of learning [5].

Through play students can develop their thinking and concentration [6] around a topic via a fun and entertaining way of achieving specific learning goals [7]. Students' engagement helps to achieve multiple goals outside of learning, such as in-depth understanding of the topic and greater awareness of the topic [8].

Augmented reality allows information and 3D virtual objects to be displayed on top of real-world objects. Furthermore, AR offers the ability on such objects and information to be processed in real-time [9, 10]. As smart mobile devices become more powerful (computationally) they can support augmented reality applications and thus make AR reachable to more people.

In addition, software libraries and frameworks such as ARCore and ARToolkit have helped the software development community to have effortless access to AR technology and include it in their applications. Research about the interaction in such environments with bare hands are also evolving along with the improvements on the hand tracking field of study making possible, for mobile devices, to run such algorithms [11].

1.1 Game Based Learning

The educational community is shifting the goals, that had been set in previous decades, related to the acquisition of knowledge and the traditional model of teaching to a new approach. One of the main goals of this new approach is the utilization of knowledge but also the development of a pleasant environment for enhanced creativity in the classroom [4]. Discussion, research, and experimentation can help students activate their interest and make learning more interactive.

GBL creates a framework where students have additional motivation compared to the traditional teaching method. Their participation because of games is greater, while the willingness they show towards the lesson in general increases [2]. According to Fotaris et al. [12] learning using digital games in education reported significant improvements in subject understanding, diligence, and motivation. In this way the game becomes an educational tool to achieve the goals set by the teacher. In addition, as Sitzmann states [13], through play, students' performance can be improved and relationships between them can be strengthened.

According to Prensky [14] an educational game is designed not only to develop players' skills but also to maintain and apply this object in the real world. Thus, it is important to connect education with everyday life as in this way it will be possible to develop students' metacognitive skills and their awareness of important issues.

1.2 Augmented Reality in Education

AR is a relatively new technology that combines the physical with the virtual world through the addition and integration of virtual information [15]. In terms of education, it is important to connect the two worlds namely the physical and the virtual as opposed to VR where the user is completely immersed in a virtual environment [16]. The use of AR enables students to diversify and enrich the physical world through new technology and the use of smart devices rather than completely replace it [9]. The additional information, images and objects provided help students to effectively understand abstract concepts, thus helping to improve knowledge and learning [17, 18].

Mobile AR is considered one of the most up-to-date technologies in education for the coming years as it is easily accessible due to the high penetration of smart devices [19]. The reasons are that it is an inexpensive technology, easy to apply in the classroom and safe. With AR, students can simultaneously interact with both the real and virtual worlds in real time [20]. This turns AR into an exciting entertainment and learning tool for kids. In addition, the new features it offers help to develop their imagination and creativity [21] and at the same time helps to connect the game with learning.

AR can be used as a learning tool in the hands of the teacher to activate students' interest and gain their attention, so that they can focus more on the lesson. According to Oranç and Küntay [22], through AR applications, students delve deeper with this technology. In addition, they believe that to be effective there must be utilization of the knowledge and skills that students acquire in their daily lives. According to Hirsh [23], the principles that should guide the use of AR in education are essential: (a) to encourage students to be actively involved in the process, (b) to include additional material, (c) to provide important new experiences with which they can relate, and (d) to offer an environment of socialization and cooperation.

Moreover, the combination of AR with game-based learning (ARGBL) has gained momentum in the field of education as it creates a playful and enriching learning experience [24]. Utilization of this technology leads to the creation of new interactive visual learning environments [5] that allow new approaches to teaching and learning experience in various disciplines.

2 Background Work

Game Based Learning is an approach that is constantly gaining ground in the field of education and teaching due to the opportunity it gives students to learn through enjoyable and creative activities. The field of literature & language is no exception since this is one of the most frequent courses in the typical school curricula. Through the electronic or non-electronic games students can develop their vocabulary [50] both in their mother tongue and in the second foreign language [51-53].

2.1 Alternative Educational Tools for Literature and Language Education

The field of literature & language as mentioned is one of the most important in education. The pedagogical teaching methods used are many and range from traditional to modern and innovative. From creative writing and the use of everyday objects to the use of technology with computers, smart devices, applications, games, VR and AR, educational robotics.

Creative writing is a method that utilizes various techniques to spark students' imagination and help them create their own texts using language in a different way [28]. However, in addition to the classical methods, the use of technology is a new way to encourage students to improve their language and vocabulary skills [28]. Mobile games [54, 55] are increasingly being used in classrooms, presenting positive results in the areas of knowledge and skills [26, 29]. In fact, studies have shown that mobile games are an effective tool for both high school and younger ages as students developed their knowledge in repeating/reviewing, using multiple modalities and means, and having control over their own learning [30].

According to Hwang, W. et al. [31] game-based learning activities can significantly improve students' speaking skills if they work in combination with a mobile game-based learning system. The WhatsApp Social Networking Tool (WSNT) seems to work for older ages [25] as it helps students learn English interactively and collaboratively. Furthermore, Ghazisaedy [27] argues that educational robotics can help in learning English as a foreign language as the new way of learning helped them to develop additional motivation and to learn more effectively in the long run.

In their research, Cai, S. et al. [17], on 38 students of the eighth grade of an AR and motion-sensing learning technology that teaches magnetic fields in a junior high school physics course observed the effects of using natural interaction on students can improve students' learning attitude and learning outcome.

2.2 AR in Literature and Language Education

A significant number of AR applications, related to education, have use cases from language & literature education as shown by Parmaxi's et.al. [41] systematic review of 54 studies regarding language learning between the years 2014–2019. According the review, most AR studies dealt with vocabulary (23.9%), reading (12.7%), speaking (9.9%) writing (8.5%) or generic language skills (9.9%). However, the authors point out that no special connection was found with learning theories during the application of AR. This gap can be filled through the creation of applications in collaboration with stakeholders in the educational community.

AR-related language applications and research have shown positive results in terms of vocabulary and spelling development [4], improving student motivation and increasing memory. Similar results appeared in a study on the improvement of vocabulary and grammatical structures [42]. Also, positive results were found regarding the learning of English as a second foreign language through the application of AR [43] in relation to the traditional teaching methods.

The use of AR seems to help students significantly in the complex and demanding field of language as they seem to achieve higher performance, to retrieve knowledge from memory more easily [44], and also to improve their narrative ability, the size of their stories as well as their creativity and imagination [45]. Another important fact is that in most researches it seems that, at the same time with the cognitive objects increases the motivation, social collaboration, and interest of the students for the lesson [38, 46].

Collectively, we have researched more than 30 studies and the conclusions can be summarized to: (a) naturalistic approaches applied to AR technology help to foster positive attitudes towards AR, (b) AR facilitate collaboration and (c) AR-related language applications enhance the users' social collaboration, personal development and skills.

2.3 Hand Interaction with AR

Research on hand interaction with the use of AR suggests that one should use the hands like when interacting with the musical instrument called Theremin. In this way the errors are minimized when hands block or confuse. A gesture-based interaction has been developed which includes functionalities as menu selection, object manipulation and more [33]. A method for realistic grasp is proposed that uses predefined rules for the movements of the hands as a physics simulation would be computationally heavy. The system checks if an object is considered grabbed and if two or more points of the hand are interacting with the object from opposite sides [34]. A similar method to the above is proposed but with the use of a glove that will provide haptic feedback as well [35]. A system to control the tv functionality is developed using Convolutional neural networks and Convolutional pose machines for hand recognition [36]. Another system for interaction with 3D objects in museum is proposed using the Leap Motion device for the hand recognition [37]. Using the Leap Motion device, researchers developed an application for learning geometry, that students can draw 2D and 3D shapes of geometrical shapes on top of a marker [32].

Objectives
The purpose of our study is to raise awareness and develop students' vocabulary and creativity on the topics of recycling and Covid 19. The research utilized the approach of game-based learning and AR hand interaction technology. Our goals were for students to:

- Develop their vocabulary around the topics of recycling and Covid 19.
- Utilize AR hand interacting technology to become more active in class (motivation).
- Develop concentration, engagement, and enjoyment using an AR game.
- Develop their creativity by making a story of their own based on the objects they found.
- Be aware of recycling and Covid issues 19.

3 Field Study Methodology

Our field study was carried out at the Elementary School of Plakia, Crete, Greece, in October 2020. While further expansion of the research was originally planned, the field study was stopped due to the mandatory quarantine in respect with covid-19. The research sample was homogeneous in language and consisted of 36 students (16 boys, 20 girls). The age of the students was 10 years old. The requirements regarding information, consent, confidentiality and use of data were met carefully, both orally and in writing, by informing school staff, students, and parents about the purposes of the study and their right to deny their participation. The students were divided into two groups based on their classroom.

We also obtained the approval of the Primary School Directorate to conduct this study in schools in its area. Then, in respect to the protocol of the institutional review committee, we contacted the school principals to describe the study and ask for permission to meet with primary school teachers, to explain the study and determine their interest in participating.

The study was conducted in two levels, the first concerned the teachers and the viewpoint they had about the lesson and the second had to do with the students. In this way, the aim is for the study to capture a more comprehensive opinion of the process and how to use an AR game in the learning practice. For the teachers, the semi-structured interview [47] was used as a tool while for the students, the observation method [48] was used. The interviews were conducted shortly after the end of the process, so that the conclusions and thoughts of the teachers are fresh.

4 Our AR Hangman Game

In this section of our paper we present a game that we have implemented in Unity3D Game Engine using (AR foundation and ARCore) a custom tool built to facilitate the process of game development that requires interaction with physical hands in augmented reality environments.

4.1 Constructing the Game

The design of our application is separated in two main modules (see Fig. 1), the first module, namely the hand tracking module is a service that is implemented for hand recognition and tracking and the second module, namely the Unity toolset, constitutes the functionality and representation of the hand in the game. The separation in two different modules allows for higher frame rates of the game as the device running the game is relieved from the hand recognition process.

The main components of our application, in a high architectural level, are presented in Fig. 1 while Fig. 2 shows the usage of these components described in a workflow diagram of a common usage scenario created with our application.

The Unity toolset provide basic components such as a socket client for the communication with the hand tracking service and an image capture method that is required to send a stream of images on the service. The rest components concern the game creation and interaction process. These components are a virtual hand, hand actions, gestures, selectors, and raycasters (see Fig. 1 right part & see Fig. 2(a)).

To better use the hand points, we have separated them on their corresponding fingers, so when there is a need to retrieve specific hand points, they can be accessed by calling the involved fingers. Predefined gestures also exist that can be used as is without the need of accessing the finger classes or the hand points. To add on this and make it even easier and simpler to use a gesture type enumerated value is assigned to each gesture that is implemented and a gesture manager class has been created. Through the gesture manager class all gestures can be accessed by using the interrelated gesture type name.

The HandAction class (see Fig. 2(a)), is responsible for defining the behavior of the virtual hand when interaction occurs with another virtual object in the application. Such actions involve the movement of a virtual object. This class exists mostly to cover additional behaviors or different implementations of already existing behaviors.

Selection Interfaces (see Fig. 2(a)) include Ray cast provider that is a component that describes the type of a ray caster to be used. This ray caster is used on the selector classes that their responsibility is to check if the hand is about to interact with a virtual object or not. Finally, in this category of components the selection action classes are also included, which are used to visually inform that a virtual object is about to or interacting with the hand.

The hand is represented virtually within the visualHand class which is wrapped in the actionHand class. In the actionHand class further information about the hand is given such as the method to detect possible interactions.

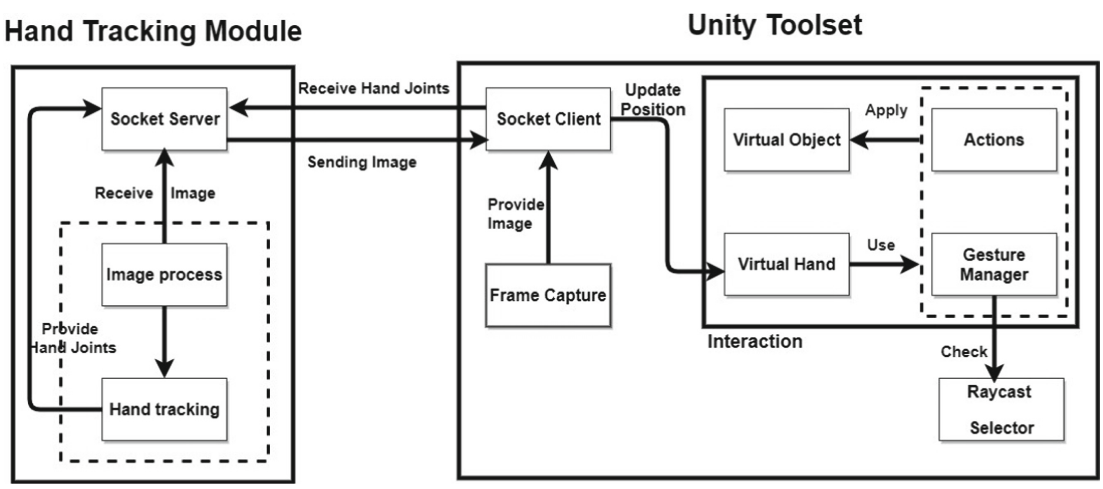
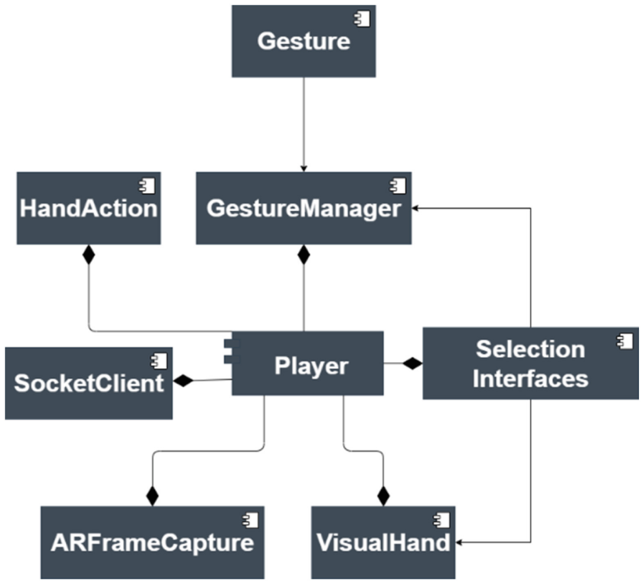


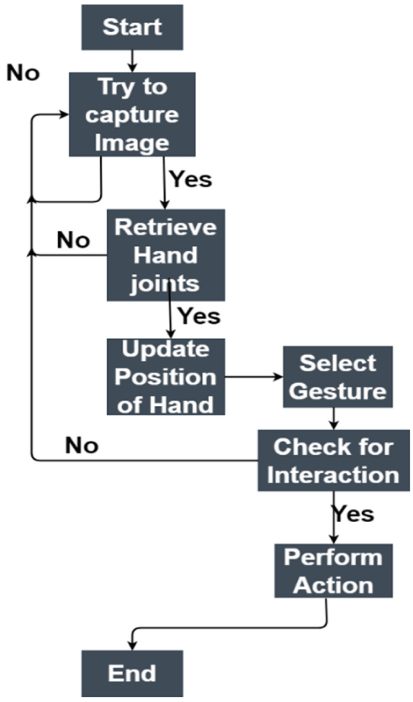
Fig. 1. Architecture diagram

The service which connects the Image process module and Socket Server (see Fig. 1), is responsible for receiving a stream of images from the device and processing the image to a form that is compatible with the hand recognition algorithm. The hand recognition algorithm can be changed easily as the rest of the service is not bound to a specific algorithm. This allows each developer to use their desired algorithms. For this project we use a modified version of for the handtracking [39].

The game was implemented following the Unity toolset flow Diagram (see while Fig. 2(b)). When the game starts it initializes a connection to the service and the virtual hand. After the connection is established it starts to update the position of the virtual hand to the corresponding position of the real hand on the screen. Each frame we check if a “grab” gesture is detected that is defined by the ray cast hits of the thumb and one more finger at least to a virtual object. If the gesture is detected, then a “performAction” method is fired that starts to move the grabbed object with the hand. To place the grabbed object to the user desired position, the application calculates the distance between the target position and the grabbed objects’ relative position. To help the player to put the object easier we have set a “wider” area for the desired position.



(a) Unity Toolset Component Diagram



(b) Unity Toolset flow diagram of main task

Fig. 2. Component and flow diagrams

The game has taken advantage of ARCore augmented images to link the 2D images that get acquired from the first part to their respective 3D objects.

4.2 Playing the Game: The Case of the Word Hangman

The game aims to (a) raise students' awareness about recycling and Covid 19, (b) enable students to develop their vocabulary in relation to the thematic areas and (c) develop creativity skills by making their own virtual picture stories and telling their own stories as stated previously at Sect. 2 where the objectives of our work were defined.

The game involves two main tasks, the first task is to support language learning and the second is about enhancing creativity. In specific, the first task is based on the classic word game of hangman and its goal is for the student to complete five words correctly to progress to the second task of the game which is to present a short story involving the previously completed five words.

The first task of the game has 5 levels, each of increasing difficulty. Each level contains a word that the student needs to fill and a set of letters that represents possible matches to the word's letters that needs to be completed. Additionally, to make it more challenging the student has five "lives" for each word. The given word is presented with the first letter on spot and the rest letters are presented by empty dashes. The student must grab a letter from the given set and place it on an empty dash. On a correct move the letter will be fit on top of the dash and stay there until the word is complete. On a wrong move the letter will change color as an indication of a wrong placement and after that it will disappear, and one player life will be lost.

For each level a pool of words has been created (see Table 1) for an indicative selection of words per game level. The selection of the word on each level is done randomly by the game. As levels advance the number of letters per word increases making it harder for the student to complete the word.

Each time a word is completed the student receives a card representing this word, then the game continues to the next level where the student must find a new word. The game completes when the student wins all levels or when the student ends the game earlier.

For every word the student has 5 wrong guesses. To help the student guess, if two wrong guesses are made, a picture representing the word is shown as help. If the student cannot find the word and loses all the guesses, then a “retry level” option is presented offering a restart of the level again with a new random word. When this option is selected the letters to choose from will be more straight forward as extra help to the student.

This help is given to the student as motivation and encouragement to not give up or have an unpleasant experience. The goal of the game is for the student to complete all the levels without skipping any. Also, it is important to increase students’ confidence and thus not lose interest on the game and complete all levels. The completion of the levels is also important for the next part of the game as it allows students to build on the imagination and storytelling when more words are involved.

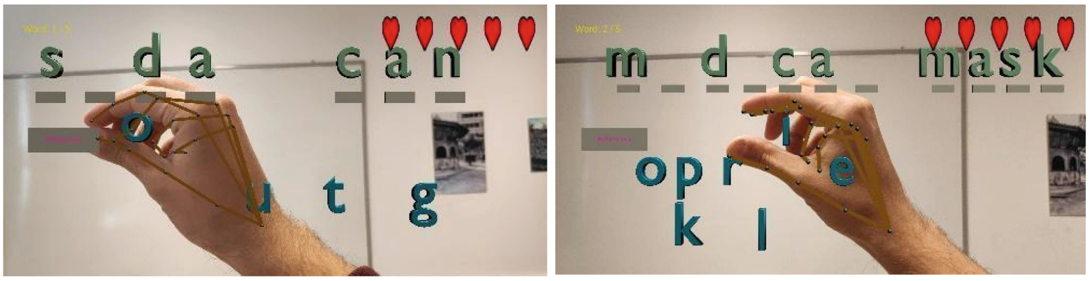
Table 1. Sample words per game level.

Level 1	Level 2	Level 3	Level 4	Level 5
Words concerning recycling				
Soda can	Waste	Cardboard	Efficiency	Greenhouse compost
Metal	Rubber	Paper bag	Glasshouse	Conservatory
Glass	Bucket	Waste bin	Garbage truck	Regenerate
Oil	Plastic	Trash bag	Disposable	Biowaste
Bin	Plant	Recycle	Ecosystem	Biodegradation
Words concerning Covid-19				
Flu	Medical mask	Hygiene	Quarantine	Social distancing
Virus	Vaccine	Anosmia	Dispenser	Stethoscope
Soap	Gloves	Pandemic	Epidemic	Asymptomatic
Nurse	Hospital	Sanitation	Infection	Contagiousness

As stated, before a word pool has been created for the needs of the game and an indicative set of words is presented on the above table. The difficulty of each level is based on the number of letters in each word. As the levels progress the student has not only more compound words to find but also has to think and understand better the thematic areas of the game.

In respect with the game-flow, players first choose the thematic area and then the game begins by presenting as many dashes as necessary for the word of the first level.

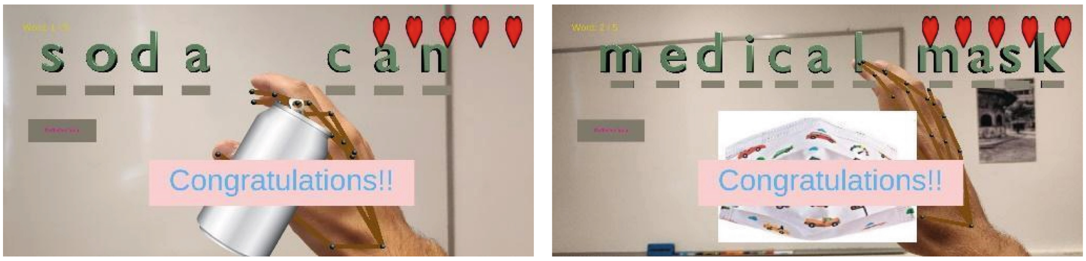
Players use their hand to grasp and move letters that appear on the screen (Fig. 3(a and b)). For each incorrect letter placement, the player loses one live. Player lives are denoted with red hart images.



(a) Thematic area: **Recycling** (b) Thematic area: **Covid 19**

Fig. 3. Virtual Hand Moving Letters

When all letters are placed in the correct position a 2D object representing the word is displayed (Fig. 4). Consequently, in the classroom the teacher gives the student a paper-card corresponding to the word completed in the AR game (Fig. 5). In this way the game materializes Blended learning as students play in both virtual and real worlds.



(a) Thematic area: **Recycling** (b) Thematic area: **Covid 19**

Fig. 4. Completed word and 2D word object

Through storytelling students have the opportunity to develop both their vocabulary and their imagination. They are able to consider new ideas and express them in words.

Once the last game level has been reached and the students have collected all their paper cards, representing the artefacts that describe the words found, they can place them on their desk in any order they want and create their picture story (Fig. 6). Once the paper-cards are in the desired sequence, students can place the mobile device that they use to play the game, e.g. a smart phone or a tablet, on top of the paper-cards to generate the 3D virtual objects (Fig. 6). The story can be presented to and shared with their classmates.

As explained at the Field study Methodology section above our students have been seperated into two groups according to their classroom. The students of the first classroom had at their disposal the teaching tools they use in their daily student life such as the blackboard and the projector. These students played the traditional hangman game and were then asked to write a story related to these words using as many as they could.

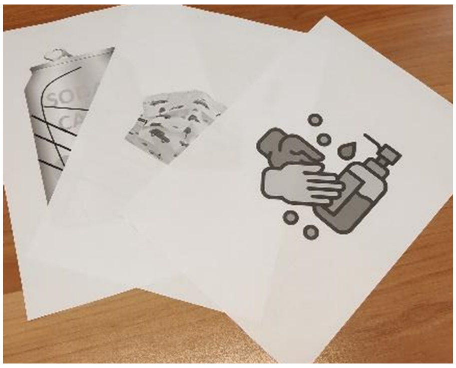
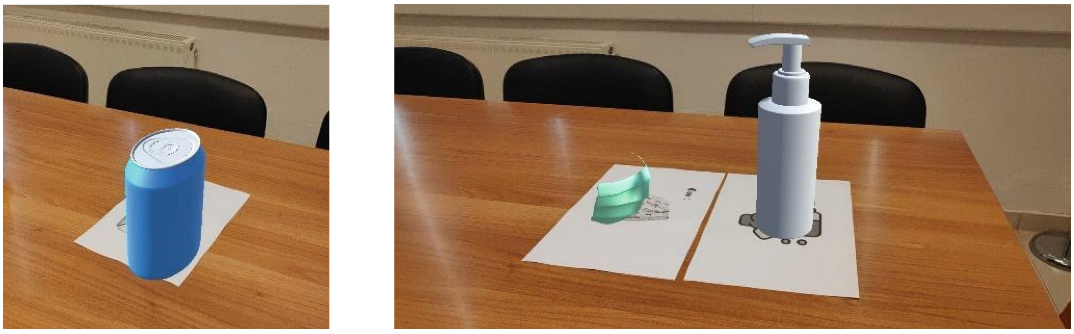


Fig. 5. Paper-card icons



(a) Thematic area: **Recycling** (b) Thematic area: **Covid 19**

Fig. 6. AR based story

The students of the second classroom were given smartphones with the application. During the use of the application they were given additional image paper cards after the successful completion of a word to place them anywhere in the room they wanted. Respectively, these students were asked to create a story. At the end of the process there was a general awareness of students about the issues of recycling and Covid 19.

5 Discussion - Research Outcomes - Results

Research has shown that using GBL in conjunction with AR can help elementary school students develop their vocabulary, become more active in class, and become more aware of important areas such as the areas recycling and Covid 19 that were investigated in our research. Both teachers and students converged in this direction and support the above statement. The table below (Table 2) shows the questions by topic and some typical answers of teachers.

5.1 Field Study Results' Analysis

The following results emerged through participatory observation and interviews with teachers. The students had a great interest and participation in the exploitation of the AR game. It piqued their interest as a new and different teaching tool through which they can learn by playing. At the same time, the teachers were positive with the use of this

Table 2. Questions & typical answers by topic.

Interdisciplinary unified curriculum framework	Questions	Typical answers
Vocabulary - learning tool interdisciplinary learning	Did they develop their vocabulary around the two topics? And how?	"The students learned some new words and understood them through play." "Through the game they began to use in their written word the words they learned"
	Has the spelling improved?	"I saw a slight improvement in their spelling." "Some weak students have significantly improved their spelling on the words in the game"
	How did playing with QR codes help them?	"Almost all students were excited about the game and would like it to be repeated"
Game based learning - alternative motivational training tool	Motivation	"Almost all students wanted to win the game to get the cards with the QR codes"
	Concentration	"While playing the game, the students were completely focused on the screens of their smartphones"
	Engagement	"I saw students with reduced participation be interested and play an important role in all activities"
	Enjoyment	"Rarely do all students rejoice and enjoy the lesson as with the use of the AR game"
Connection with society awareness	Have you noticed any changes in their behavior regarding the issue of recycling?	"I have noticed that most students are now more actively involved in recycling, either inside or outside the school" "They are much more careful about where they put garbage, and they have a lot of ideas for improving recycling in their school and in their area"

(*continued*)

Table 2. (*continued*)

Interdisciplinary unified curriculum framework	Questions	Typical answers
	Did you notice any changes in their behavior regarding Covid 19?	"The students understood the actions that are necessary to protect themselves and the means at their disposal." "Now they use the sanitary ware for Covid 19 with great comfort and ease"

technology as a teaching tool. They believe that with game involvement teaching goals can be achieved as a result of a pleasant experience. More specifically, the results per thematic unit were the following:

In the field of vocabulary, no significant differences were observed between the two classes. Both in the class that the hangman game was played in the traditional way and in the second class, students developed their vocabulary and their ability to spell. The difference observed between the two classes has to do with the way the students behave. In the first case the students had a similar behavior in relation to the daily lessons. However, in the second case, they considered that they were participating in a game and not that they were taking part in a school lesson.

Significant differences were observed in the part where they used the cards with the QR codes and created their own stories. Students who attended the lesson in the traditional way did not have many ideas as opposed to those who used AR game technology. The students became more active from the moment the teacher announced that they would receive a card when they found a word and seemed to be more enthusiastic. They liked to place the cards with the QR codes in different parts of the classroom, "decorating" it with their own style. In this way they started to have more ideas and to think differently about the stories that the teacher asked them to make. The collaboration between the students increased and they started discussing and proposing thoughts and ideas to each other.

In terms of awareness, in both cases an increase was observed with a slight difference in the part using the AR game. Students in both classes learned about the possibilities offered by recycling and how they can protect themselves and those around them from Covid 19. After the end of the lesson, most of them started coming up with ideas for ways that could improve the spirit of recycling both inside and outside the school. This showed that through various activities within the school, students can become more aware and acquire new habits in their daily lives. In addition, their activities helped them gain a better knowledge and argument to propose changes in the local community. An example is that the students suggested a meeting with the local mayor to raise their concerns and suggest ideas about the recycling framework. This showed that there was a connection between the school and the local community. About Covid 19 the activities helped the students to better understand the dangers and to change some of their daily habits. In addition, they learned to use disinfectants properly and to be vigilant so as not to be exposed to the virus.

5.2 Conclusions

The use of new technologies can help significantly in the learning process if the tools used to fulfill the goals are set by teachers. Using Augmented Reality games as an educational tool can help students in a variety of ways. Students become more active throughout the lesson and are more focused for longer. This helps them to gain more benefits from the course than traditional teaching. In addition, the new way of learning and the different stimuli help them to develop their thinking and their creativity. This was evident in the present study as the stories they created had great diversity, strong signs of imagination and new ideas that these students had not used in the past. This had showed that it is not only the application that matters but the general educational approach that can highlight the potential of technology and is in line with Juhee's findings in paper "Problem-based gaming via an augmented reality mobile game and a printed game in foreign language AQ3 education [40].

Using our AR game and the activities that followed, the students got to know better and in depth the areas of recycling and Covid 19. The AR game did not make much difference in terms of improving spelling, but students were more motivated for the lesson and were more positive in continuing until the game was over. Finally, addressing these two important issues of daily life helped students become more aware of recycling and Covid 19, take initiatives and develop active social action.

6 Future Work

The research was carried out during the first semester in only two classes. According to our original schedule, the goal was to expand to more students so that they could get safer and clearer results regarding the relationship between an AR game and the classic learning process. However, the field study was stopped due to the mandatory quarantine that last three months and up to this day. The goal for the future is to continue and add new modules to our AR game that can provide additional information such as the xAPI library which can create data for learning analytics. Through XAPI, the AR game will be able to note players' actions in more detail, as well as the points that made it difficult or aroused their interest to a greater extent. Another point that can be explored in the future is the connection of AR games with the STEM teaching method. In other words, students should combine their ideas with virtual models and then apply them in practice.

References

1. Radu, I.: Augmented reality in education: a meta-review and cross-media analysis. Pers. AQ4 Ubiquit. Comput. **18**(6), 1533–1543 (2014). https://doi.org/10.1007/s00779-013-0747-y
2. Hamari, J., Shernoff, D.J., Rowe, E., Coller, B., Asbell-Clarke, J., Edwards, T.: Challenging games help students learn: an empirical study on engagement, flow and immersion in game-based learning. Comput. Hum. Behav. **54**, 170–179 (2016)

3. Riemer, V., Schrader, C.: Learning with quizzes, simulations, and adventures: students' attitudes, perceptions and intentions to learn with different types of serious games. Comput. Educ. **88**, 160–168 (2015). https://doi.org/10.1016/j.compedu.2015.05.003
4. Mahayuddin, Z., Saif, A.F.M.: Augmented reality based AR alphabets towards improved learning process in primary education system. JCR **7**(19), 514–521 (2020)
5. Huang, T.C., Chen, C.C., Chou, Y.W.: Animating eco-education: to see, feel, and discover in an augmented reality-based experiential learning environment. Comput. Educ. **96**, 72–82 (2016). https://doi.org/10.1016/j.compedu.2016.02.008
6. Schechter, R., Macaruso, P., Kazakoff, E.R., Brooke, E.: Exploration of a blended learning approach to reading instruction for low SES students in early elementary grades. Comput. Sch. **32**(3–4), 183–200 (2015). https://doi.org/10.1080/07380569.2015.1100652
7. Alarcia, D.T., Barco, D.I.: Videojuegos y aprendizaje de la Historia: la saga Assasin's Creed. Contextos Educativos. Revista de Educación **17**, 145 (2014). https://doi.org/10.18172/con.2598
8. Chu, H.-C., Hung, C.-M.: Effects of the digital game-development approach on elementary school students' learning motivation, problem solving, and learning achievement. Int. J. Distance Educ. Technol., 472–487 (2015). https://doi.org/10.4018/978-1-4666-8200-9.ch023
9. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE Trans. Inf. Syst. **E77-D**, 1321–1329 (1994)
10. Ronald, T.A.: A survey of augmented reality. Presense **6**(4), 355–385 (1997)
11. Gouidis, F., Panteleris, P., Oikonomidis, I., Argyros, A.: Accurate hand keypoint localization on mobile devices. In: Proceedings of the 16th International Conference on Machine Vision and Applications (MVA) (2019). https://doi.org/10.23919/MVA.2019.8758059
12. Fotaris, P., Mastoras, T., Leinfellner, R., Rosunally, Y.: Climbing up the leaderboard; an empirical study of applying gamification techniques to a computer programming class. Electr. J. E-learn. **14**(2), 94–110 (2016)
13. Sitzmann, T.: A meta-analytic examination of the instruction effectiveness of computer-based simulation games. Pers. Psychol. **64**, 489–528 (2011)
14. Prensky, M.: Digital Game-Based Learning. McGraw-Hill, New York (2007). https://doi.org/10.1145/950566.950567
15. Enyedy, N., Danish, J. A., Delacruz, G., Kumar, M.: Learning physics through play in an augmented reality environment. Int. J. Comput. Support. Collab. Learn. **7**(3) (2012). https://doi.org/10.1007/s11412-012-9150-3
16. Bacca, J., Baldiris, S., Fabregat, R., Graf, S., Kinshuk: Augmented reality trends in education: a systematic review of research and applications. Educ. Technol. Soc. **17**(4), 133–149 (2014).
17. Cai, S., Chiang, F.K., Sun, Y., Lin, C., Lee, J.J.: Applications of augmented reality-based natural interactive learning in magnetic field instruction. Interact. Learn. Environ. **25**(6), 778–791 (2017). https://doi.org/10.1080/10494820.2016.1181094
18. Laine, T.H., Nygren, E., Dirin, A., Suk, H.J.: Science spots AR: a platform for science learning games with augmented reality. Educ. Technol. Res. Dev. **64**, 507–531 (2016)
19. Alakärppä, I.; Jaakkola, E.; Väyrynen, J.; Häkkilä, J.: Using nature elements in mobile AR for education with children. In: Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, Vienna, Austria, 4–7 September (2017)
20. Hockly, N.: Augmented reality. ELT J. **73**(3), 328–334 (2019)
21. Shabalina, O., Malliarakis, C., Tomos, F., Mozelius, P.: Game-based learning for learning to program: from learning through play to learning through game development. In: Pivec, M., Gründler, J. (eds.) Proceedings of the European Conference on Games Based Learning, vol. 11, pp. 571–576. Academic Conferences and Publishing International Limited, Reading, UK (2017)

22. Oranç̧, C., Küntay, A.C.: Learning from the real and the virtual worlds: Educational use of augmented reality in early childhood. Int. J. Child-Comput. Interact. **21**, 104–111 (2019). https://doi.org/10.1016/j.ijcci.2019.06.002
23. Hirsh-Pasek, K., Zosh, J.M., Golinkoff, R.M., Gray, J.H., Robb, M.B., Kaufman, J.: Putting education in educational apps: lessons from the science of learning. Psychol. Sci. Public Interest. **16**, 3–34 (2015). https://doi.org/10.1177/1529100615569721
24. Tobar-Muñoz, H., Baldiris, S., Fabregat, R.: Augmented reality game-based learning: enriching students' experience during reading comprehension activities. J. Educ. Comput. Res. **55**(7), 901–936 (2017)
25. Mwakapina, W., Mhandeni, J.S., Nyinondi, O.: WhatsApp mobile tool in second language learning: opportunities, potentials and challenges in higher education settings in Tanzania. Int. J. English Lang. Educ. **4**(2), 70 (2016). https://doi.org/10.5296/ijele.v4i2.9711
26. Önal, N., Çevik, K.K., Şenol, V., Önal, N.: The effect of SOS Table learning environment on mobile learning tools acceptance, motivation and mobile learning attitude in English language learning. Interact. Learn. Environ., 1–14. https://doi.org/10.1080/10494820.2019.1690529
27. Alemi, M., Meghdari, A., Ghazisaedy, M.: The impact of social robotics on L2 learners' anxiety and attitude in English vocabulary acquisition. Int. J. Soc. Robot. **7**(4), 523–535 (2015). https://doi.org/10.1007/s12369-015-0286-y
28. Gilakjani, A.P., Branch, L., Branch, L.: A review of the literature on the integration of technology into the learning and teaching of English language skills. Int. J. English Linguist. **7**(5), 95–106 (2017). https://doi.org/10.5539/ijel.v7n5p95
29. Berns, A., Isla-Montes, J.-L., Palomo-Duarte, M., Dodero, J.-M.: Motivation, students' needs and learning outcomes: a hybrid game-based app for enhanced language learning. Springerplus **5**(1), 1305 (2016). https://doi.org/10.1186/s40064-016-2971-1
30. Butler, Y.G.: The use of computer games as foreign language learning tasks for digital natives. System **54**, 91–102 (2015). https://doi.org/10.1016/j.system.2014.10.010
31. Hwang, W.-Y., Shih, T.K., Ma, Z.-H., Shadiev, R., Chen, S.-Y.: Evaluating listening and speaking skills in a mobile game-based learning environment with situational contexts. Comput. Assist. Lang. Learn. **29**(4), 639–657 (2016). https://doi.org/10.1080/09588221.2015.1016438
32. Le, H.Q., Kim, J.I.: An augmented reality application with hand gestures for learning 3D geometry. In: 2017 IEEE International Conference on Big Data and Smart Computing, BigComp, pp. 34–41 (2017). https://doi.org/10.1109/BIGCOMP.2017.7881712
33. Cui, C., Sourin, A.: Mid-air interaction with optical tracking for 3D modeling. Comput. Graph. **74**, 1–11 (2018). https://doi.org/10.1016/j.cag.2018.04.004
34. Kim, J.S., Park, J.M.: Direct and realistic handover of a virtual object. In: IEEE International Conference on Intelligent Robots and Systems, vol. 2016, pp. 994–999 (2016). https://doi.org/10.1109/IROS.2016.7759170
35. Liu, H., et al.: High-fidelity grasping in virtual reality using a glove-based system. In: Proceedings of the IEEE Conference on Robotics and Automation, vol. 2019, pp. 5180–5186 (2019). https://doi.org/10.1109/ICRA.2019.8794230
36. Wu, Y., Wang, C.M.: Applying hand gesture recognition and joint tracking to a TV controller using CNN and Convolutional Pose Machine. In: Proceedings of the International Conference on Pattern Recognition, vol. 2018, pp. 3086–3091 (2018). https://doi.org/10.1109/ICPR.2018.8546209
37. Kyriakou, P., Hermon, S.: Can i touch this? Using natural interaction in a museum augmented reality system. Digit. Appl. Archaeol. Cult. Herit. **12**, 1–9 (2019). https://doi.org/10.1016/j.daach.2018.e00088
38. Sdravopoulou, K., Castillo, J.J.G., González, J.M.M.: Naturalistic approaches applied to AR technology: an evaluation. Educ. Inf. Technol. **26**(1), 683–697 (2020). https://doi.org/10.1007/s10639-020-10283-4

39. Victor, D.: HandTrack: a library for prototyping real-time hand tracking interfaces using convolutional neural networks, GitHub Repos. (2017). https://github.com/victordibia/handtracking/tree/master/docs/handtrack.pdf
40. Juhee, L.: Problem-based gaming via an augmented reality mobile game and a printed game in foreign language education. Educ. Inf. Technol. (2020).https://doi.org/10.1007/s10639-020-10391-1
41. Parmaxi, A., Demetriou, A.A.: Augmented reality in language learning: a state-of-the-art review of 2014–2019. J. Comput. Assist. Learn. **36**(6), 861–875 (2020). https://doi.org/10.1111/jcal.12486
42. Martinez, A.A., Benito, J.R.L., Gonzalez, E.A., Ajuria, E.B.: An experience of the application of augmented reality to learn English in infant education. In: 2017 International Symposium on Computers in Education, SIIE 2017, vol. 2018, pp. 1–6 (2017). https://doi.org/10.1109/SIIE.2017.8259645
43. Dalim, C.S.C., Piumsomboon, T., Dey, A., Billinghurst, M., Sunar, S.: TeachAR: an interactive augmented reality tool for teaching basic English to non-native children. In: Adjunct Proceedings of the 2016 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2016, pp. 344–345 (2017). https://doi.org/10.1109/ISMAR-Adjunct.2016.0113
44. Solak, E., Cakır, R.: Investigating the role of augmented reality technology in the language classroom . Croatian J. Educ. **18**(4), 1067–1085 (2017). https://doi.org/10.15516/cje.v18i4.1729
45. Yilmaz, R.M., Goktas, Y.: Using augmented reality technology in storytelling activities: examining elementary students' narrative skill and creativity. Virtual Reality **21**(2), 75–89 (2016). https://doi.org/10.1007/s10055-016-0300-1
46. Tobar-Muñoz, H., Baldiris, S., Fabregat, R.: Augmented reality game-based learning: enriching students' experience during reading comprehension activities. J. Educ. Comput. Res. **55**(7), 901–936 (2017). https://doi.org/10.1177/0735633116689789
47. Bartholomew, K., Henderson, A.J.Z., Marcia, J.E.: Coding semi-structured interviews in social psychological research. In: Reis, H., Judd, C.M. (eds.) Handbook of Research Methods in Social and Personality Psychology, pp. 286–312. Cambridge University Press, Cambridge (2000)
48. Lapassade, G.: L'observation participante. In: Hess, R., Gabriele, W. (eds.) L'observation participante. Paris. ECONOMICA, Anthropos (2006)
49. Hall, G.: Recent developments in uses of literature in language teaching. In: Teranishi, M., Saito, Y., Wales, K. (eds.) Literature and Language Learning in the EFL Classroom, pp. 13–25. Palgrave Macmillan, London (2015). https://doi.org/10.1057/9781137443663_2
50. Calvo-Ferrer, J.R.: Educational games as stand-alone learning tools and their motivational effect on L2 vocabulary acquisition and perceived learning gains. Br. J. Edu. Technol. **48**(2), 264–278 (2017)
51. Ragatz, C. M.: Playing vocabulary games and learning academic language with gifted elementary students (Doctoral dissertation). Retrieved from ProQuest Dissertations & Theses Global. (Order No. 3689691) (2015)
52. Taheri, M.: The effect of using language games on vocabulary retention of Iranian elementary EFL learners. J. Lang. Teach. Res. **5**(3), 544–549 (2014). https://doi.org/10.4304/jltr.5.3.544-549
53. Casañ-Pitarch, R.: Gamifying content and language integrated learning with serious videogames. J. Lang. Educ. **3**(3), 107–114 (2017)

54. Vidakis, N., Barianos, A.K., Trampas, A.M., Papadakis, S., Kalogiannakis, M., Vassilakis, K.: Generating education in-game data: the case of an ancient theatre serious game. In: Proceedings of the 11th International Conference on Computer Supported Education, CSEDU 2019, vol. 1, pp. 36–43 (2019)
55. Vidakis, N., Syntychakis, E., Kalafatis, K., Christinaki, E., Triantafyllidis, G.: Ludic educational game creation tool: teaching schoolers road safety. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2015. LNCS, vol. 9177, pp. 565–576. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20684-4_55