POLITECNICO MILANO 1863

INNOVATIVE INTERACTIVE INTERFACES Laboratory

DHolophin: Rainbow

DESIGN AND TECHNOLOGY DOCUMENTATION

Year 2018/2019

Andrea Mafessoni

Alessandro Magni

Alejandra Imaz

Gianmarco Dello Preite Castro

ABSTRACT: Nowadays many people in the world suffer from NDD, a complex gamma of neurodevelopmental disorders that may cause deficit in attention, focus and memory. *DHolophin : Rainbow* is a memory game in mixed reality intended for these people, that provides them with a funny and useful tool to keep their abilities trained. The game consists in an MR experience with real world-tangible interaction, that makes use of both HoloLens technology and DolphinSAM smart toy to offer an innovative experience in a familiar physical environment. This document explains the process of design of the application, from the scratch to the final actual implementation.

THE TEAM:

 ANDREA MAFESSONI andrea1.mafessoni@mail.polimi.it +39 345 609 8669	 ALESSANDRO MAGNI alessandro2.magni@mail.polimi.it +39 331 222 6866
 GIANMARCO DELLO PREITE CASTRO gianmarco.dello@polimi.mail.it +39 331 788 1959	 ALEJANDRA IMAZ alejandra.imaz@polimi.mail.it +34 687 786 413

TABLE OF CONTENT:

1 INTRODUCTION

DHolophin: Rainbow is a memory game in mixed reality, developed on HoloLens and intended for people affected by Intellectual Disability. The game has been developed as project of *Advanced User Interfaces course*, in collaboration with therapists from the cooperative non-profit organization *Fraternità e Amicizia*.

The project brought us to face two main challenges:

- from the Design point of view, manage to devise an application that could provide useful content to people who suffer from NDD.
- from the Technology point of view, manage to build a Mixed Reality experience that could also offer a Tangible Interaction with real-life smart objects.

This document explains the whole process of design and further implementation of the application, starting from the analysis of our stakeholders' needs and then proceeding with the definition of the requirements, the design of the user experience, and finally the details about the effective implementation. The document also provides an analysis of the current State of the Art and some critical consideration on the problematics faced during the process of the app creation. The last section contains some advices and indications about how to proceed in the next future, highlighting some possible improvements and enrichments for the application

.

In details, these are the section covered:

- Section 2: Elicitation, Analysis and Specification; understanding of stakeholders' needs and definition of goals and requirements of the application.
- Section 3: design of the solution; game rules, design choices, user interfaces and user experience.
- Section 4: State of the Art; analysis of technology used and similar projects in the field of MR/VR therapies for NDD patients.

- Section 5: details about the effective implementation; system components and their interaction, software architecture, application code.
- Section 6: critical reflections on the work made and possible improvements to be implemented in the next future.

2 NEEDS & REQUIREMENTS:

2.1 STAKEHOLDERS:

As first step of our design process, we identify the classes of users involved in the use of the applications and/or potentially interested in the interactive product, who constitute the category of stakeholders of the project.

We highlighted two main stakeholders for *DHolophin: Rainbow*:

- The patients;
- The therapists.

Patients represent the main end-user of the application and the effective players of the game, while therapists are the professional medical figures that follow patients in their activities and supervise their progresses.

Of course, there is a third class of stakeholders:

- The developers (actual and future)

To whom section 5 (*Implementation*) and 6 (*Propositions and Improvements*) are dedicated.

2.2 NEEDS:

The next step of our process is the elicitation. We then proceed to recognize the needs of our stakeholders, which represent the expressions of desire of our users.

Of course, the primal need for a patient to be fulfilled is having fun and being consequently entertained in an enjoyable way during the therapies. Also, a patient needs his/her personal abilities (such as memory, focus and attention) to be improved. In order to full enjoy the therapeutic activities, then, patients need tools that can be played with low or none external support.

Recall:

- Have fun during therapies;
- Have a tool to train his/her skills;

- Have a tool that does not require constant external support.

By considering therapists, instead, we highlighted the need of being able to provide patients with training exercises suitable for their level of skill and to easily follow them during the activities to eventually give advices. Also, therapist needs safe and familiar tools for training which can be played by patient without a constant surveillance. Which such considerations, we identify three main needs for the therapist:

- Being able to customize complexity of activities depending on the patient;
- Being able to easily supervise patients during the activities;
- Being able to leave patients play without constant attention.

We then obtain the following table:

STAKEHOLDER:	NEEDS:	
PATIENT	Have fun	N1
	Improve personal abilities	N2
	Play with low/no external support	N3
THERAPIST	Customizable exercises	N4
	Easily supervise patients	N5
	No constant attention	N6

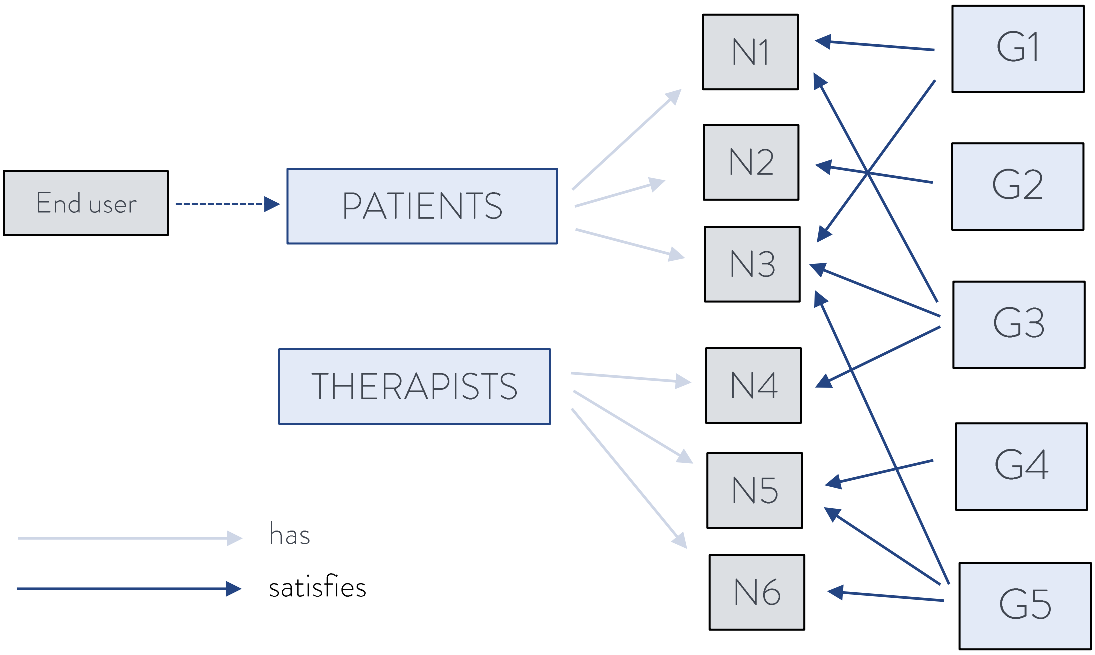
2.3 GOALS:

From the previous analysis, we identified a potential main goal for our application:

G: provide a simple & funny game, customizable by therapists, that allows patients to improve their memory, focus and attention skills in a safe and supervised environment.

from which we obtain the subsequent goals:

GOAL:	Description:
G1	Simple and funny game
G2	Improvement of memory, focus & attention
G3	Customizable content
G4	Possibility to assist patients
G5	Safe and familiar environment



2.4 CONTEXT & CONSTRAINTS:

The application is intended to be used in a therapeutic context, as training method for patients to improve their capacities. Considering the possible scenario of a therapeutic session, we set the following constraints:

- The application must be playable even in small physical spaces;
- Since the application is intended to be played even by children, the content must be appropriate to minors;
- Since the users may not have advanced knowledge in IT field, the application must be usable even by non-expert in technology.

2.5 REQUIREMENTS:

We now proceed with the specification. By translating the considerations made until now into effective possible properties of our system, we obtain the following set of requirements:

- The application must provide a game to train memory and attention; (R1)
- The game must potentially be playable by the patient alone; (R2)
- The game must provide therapist with a way to set a suitable level of difficulty before launch; (R3)
- The application must provide therapists with a system to follow the patient during the gameplay; (R4)
- The application must allow communication between therapist and patient for the whole duration of activities; (R5)
- The content and style of the application must result enjoyable and entertaining, even for children; (R6)

	G1	G2	G3	G4	G5
R1	X	X			
R2	X				X
R3		X	X		
R4				X	X
R5				X	X
R6	X				X

Now that the requirements for the application are set, we can effectively proceed to the next step and start designing the user experience at a more precise level.

3 UX DESIGN:

3.1 GENERALITY:

As stated in the previous section, the final decision was to design and develop a memory game, which could constitute a helpful tool for NDD patients to keep their focus and attention skills trained. The main challenge for us has been trying to turn a potentially boring and complex a memory game into an experience that could result attractive and funny for the users, especially if children.

This section of the document provides info about our solution, in terms of game design and user experience. For additional details about the technical implementation, go to Section 5: *Implementation*.

The final name of the application is DHolophin: Rainbow.

3.2 TECHNOLOGY:

During the process of user experience design, we made the following choices about which technologies could better fit our purposes:

- *HoloLens:* this headset (see Section 4.2: *Microsoft HoloLens*) allows users to experiment *Mixed Reality*, which is a powerful tool to provide players with an immersive experience, while at the same time keeping them in a safe and familiar environment.
- *Dolphin SAM*: this innovative smart toy (see Section 5.3: *Dolphin SAM*) will be used in the game as main controller, providing users with a smart and friendly assistant that will also act as tangible and interactive link with the physical reality.

3.3 THE GAME:

DHolophin: Rainbow provides a game in which the player must remember a combination of colors that will be shown on a virtual reality panel made of different circles. The circles are initially switched off, and will switch on when the game starts, showing each one a color. The player must remember the global combination and re-insert it when the circles switch off, using Dolphin SAM as controller.

In MANUAL game mode, by pressing either the left or the right fin of the dolphin, the player can move between the different color of the dolphin leds. Once that the desired color is found, it is possible to color circles with it by looking at them using HoloLens and pressing the central fin of Dolphin SAM.

In AUTOMATIC game mode, the player does not need to switch between the different colors, because the game does it automatically after a fixed period of time. This game has the goal of encouraging players to be fast and reactive.

If all the circles are colored in the corret way, the game will notify the player's success with lights and happy animations. Dolphin SAM will behave accordingly, emitting happy sounds and moving its body.

The game is playable at different levels of complexity, given by the number of circles and color used. Both the parameters can be set in the main menu, before launching the game.

3.4 GRAPHICS AND AUDIO:

Given that the game has the purpose of resulting funny and entertaining, we opted for a cartoon-style graphic, with bright colors and friendly interfaces.

For the same reason, we chose children's music as soundtrack and added funny audio effects during the gameplay.

3.5 USER INTERFACES:

The two actors involved in the possible scenarios are the patient, which is the effective player of the game and end user of the application, and the therapist, whose goal is to set the game difficulty at launch and supervise the patient during the play.

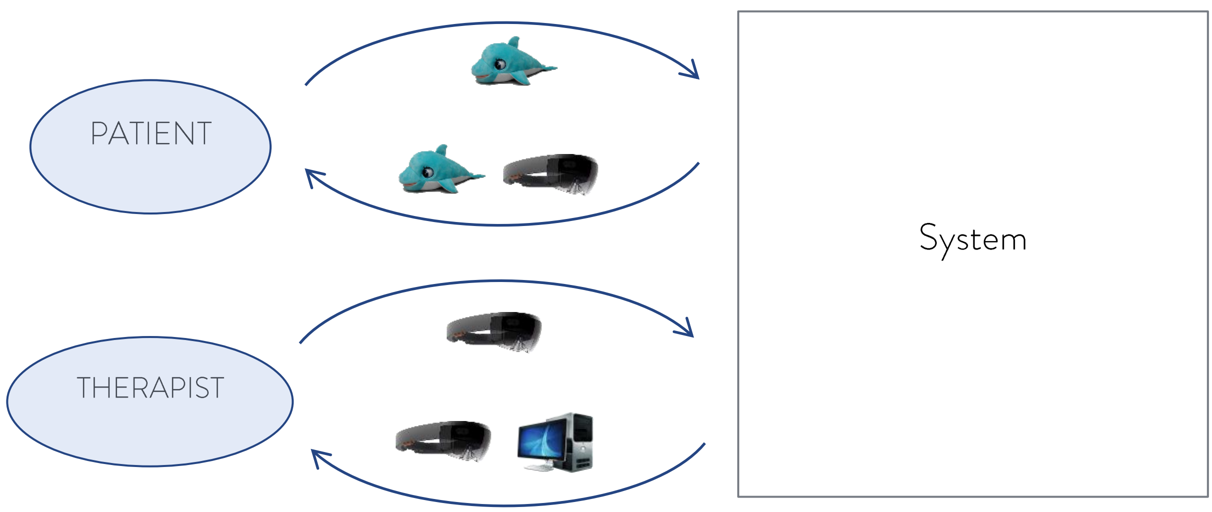
Since the game is primarily designed for NDD patients, we choose to use Dolphin SAM as controller of the game in such a way that:

- the player does not have to learn the complicated HoloLens gesture;
- the player keeps a constant familiar link to the physical world.

The patient will then interact with the game by sending inputs using Dolphin SAM and visualizing outputs both in the virtual world (inside the application scene) and in the real one (on the dolphin).

The therapist is instead assumed to master HoloLens gesture, so he/she will compile the game settings wearing the headset and using air ap gestures. When the game is launched, then, the headset will be passed to the patient, and the therapist will follow his/her movement on a pc desktop.

The final global schema of the interactions with the system is he following:



3.5 SCENARIOS:

5.4.1 *Game launch*:

Actors: Therapist, Patient

Locations: Therapy room

Description: The therapist wears HoloLens headset and proceeds to switch on the device and launch the application. He faces the main menu of the game in MR, that requires to set the number of circles and colors to be used in the game session. Since the target patient has a sufficient level of attention and memory skill to be challenged on a medium difficulty, he sets 5 as number of circles and *4* as number of colors. He then taps on the *Start!* button, helps the patient wearing HoloLens and starts following his moves on the PC.

5.4.2 *First gameplay*:

Actors: Therapist, Patient

Locations: Therapy room

Description: The patient is wearing HoloLens and watching the panel with three circles displayed in front of him. The circles get colored in blue-red-yellow, and the therapist encourages the patient to memorize the combination, while she follows his moves on the pc. The circles switch off, and Dolphin SAM shines in red. The therapist suggests the patient to squeeze the dolphin's fin, and, as matter of example, she presses the right fin making the dolphin switch to green. The patient understands the game mechanic and starts having fun by continuously making the dolphin change color. The therapist explains him that he needs to re-insert the combination and suggests him to look at a circle and press the central fin of the dolphin. The patient follows her suggestion and watches the focused circle getting colored with the color of Dolphin SAM. The therapist compliments him and encourages him to complete the combination with blue, red and yellow. In this way, the patient understands the main gameplay.

5.4.3 *Correct guess*:

Actors: Patient

Locations: Therapy room

Description: The patient is wearing HoloLens and playing the game. He has successfully memorized the combination and understood the game mechanics. The patients focus on the last circle and presses the central fin, coloring the circle in green. The combination is correct, and the game confirms the guess by playing happy sound effects and shooting fireworks in the scene. The dolphin shows happiness too, by making movements and emitting funny sounds. The patient is amused, and when the game restarts, he keeps playing to beat the game again.

5.4.4 *Automatic mode*:

Actors: Patient

Locations: Therapy room

Description: The patient is wearing HoloLens and playing the game. He has already played with the game before, but only in Manual mode. This time, the game is launched in Automatic mode, and the patient sees Dolphin SAM changing colors by itself. He then tries to press left and right fin to test if it is still possible to switch color in the same way as the previous sessions, but nothing happens. The patient then understands he cannot change the color by himself and tries to color one circle in red after that Dolphin SAM shows red lights. He squeezes the central fin, but since he took a few seconds to focus on the action, the dolphin switches to green and the circle gets colored in green. The patient then understands that he needs to be more reactive and starts focusing on the game to beat it.

4 STATE OF THE ART:

4.1 VR, AR AND MR FOR NDD PATIENTS:

Several studies analyze the use of Virtual Reality (VR) with Neurodevelopmental Disorder (NDD) patients with the main purpose of allowing people to be able to improve their social skills so that they can fit better in society and in daily life activities. Most of the work started being developed using VR [5][7], besides the expansion to Augmented Reality (AR) and Mixed Reality (MR) is starting to be notable. Some interesting related projects in the research and market field address similar problems with different approaches. In the following section we won't only focus on game-based activities which work specifically with HoloLens but will also focus on a wider spectrum. As far as we know, most of the study in this field is developed by research labs rather than companies.

As explained in [3], virtual reality exposure therapy (VRET) is being used for flying phobia, fear of heights or spiders, but also to post-traumatic stress disorder (PTSD). However, VR applications to medicine are not limited to psychiatric and behavioral healthcare, they are also been used in neuropsychology, for the assessment and rehabilitation of disabilities that result from brain injury, memory deterioration or attention deficits. More specifically, in [4] it is documented the use of VR in brain damage rehabilitation. Many examples are provided in which the use of VR was successful in helping patients. One of the sections covers memory impairments, which is strictly related to our work. A very basic approach for VR is Google Cardboard, which is used in [5] with Android smartphones. In their project, two different storytelling scenarios are shown which are focused on language and communication skills improvement.

Compared to the potential drawbacks of VR, which are associated to physical and cognitive side-effects such as motion sickness and isolation effect [8] [9], AR has the advantage to allow subjects to get immersed in the virtual world but keeping the awareness to be in the real and familiar world, which is thought to be more beneficial for these specific purposes. Many studies regarding the use of AR for NDD patients have been developed at Politecnico di Milano. These studies could be referred as a major baseline for any research in this field. Specifically, in HoloLearn [2] project, it was intended to improve autonomy in everyday life with simple application for NDD patients. The use of a virtual assistant is especially interesting as a positive feedback for the patients. Also, the MemHolo project [8] is intended for people affected by AD (Alzheimer's Disease), Suite of games, mainly focused on training memory skill using HoloLens.

Many of the projects in the related field are focused on patients with Autism. The project Hol'Autisme [1], developed by the company Actimage, consists of a suit of games intended for improving autonomy and communication using HoloLens. In [6], their main challenge is to improve the patient's ability to stay focused thanks to the interaction with a real object.

4.1 MICROSOFT HOLOLENS:

Microsoft HoloLens [10] is a self-contained, holographic computer that is mounted on a headset and is equipped with sensors, advanced optics and a custom holographic processing unit. HoloLens allows the user to interact with digital contents rendered by holograms that are placed in the virtual view of real environment.

Thanks to the spatial mapping techniques [11], the device captures the information about the physical space, identifies planes and generates a realistic rendering of the environment where interactive holograms can be placed. Hand gestures [12] are used to interact with holograms or other digital contents placed in the digital rendering of the surrounding physical space. The main gestures are Air tap and Bloom. Fast Air tap, done with thumb and index fingers, selects and activates the gazed interactive element. Maintained air tap performs drag and drop. In order to move the selected item to the desired position, both fingers must be kept closed and opened to drop the element there. Bloom, done with all fingers, is the "home" gesture and is reserved for that alone. It is a special system action that is used to go back to the Start Menu.

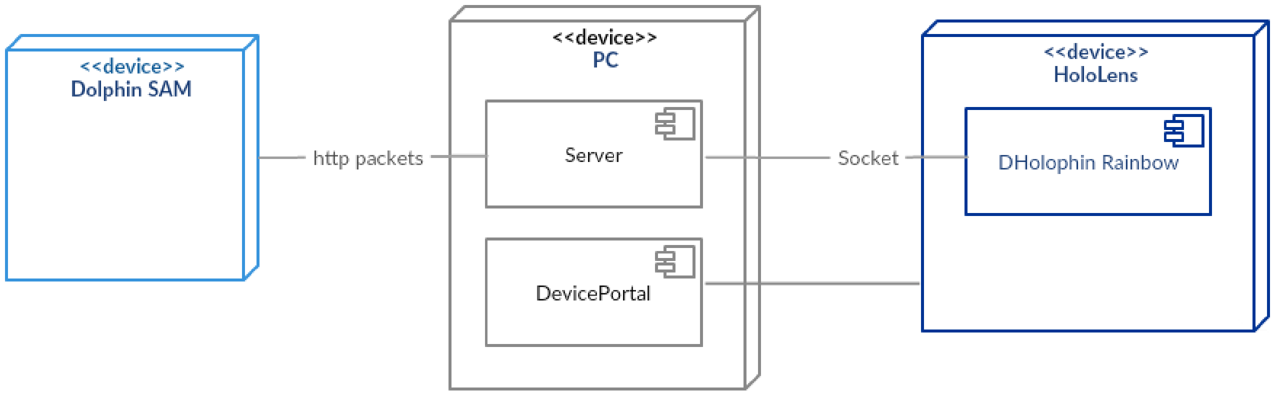
Regarding the gaze movement [13], the user view of the environment is updated. More specifically, the tracking is made only by head movements rather than eye tracking. The user's gaze focus is determined by head orientation, the changes of the pointer position and the space view correspond to head movements.

5 IMPLEMENTATION:

5.1 OVERVIEW:

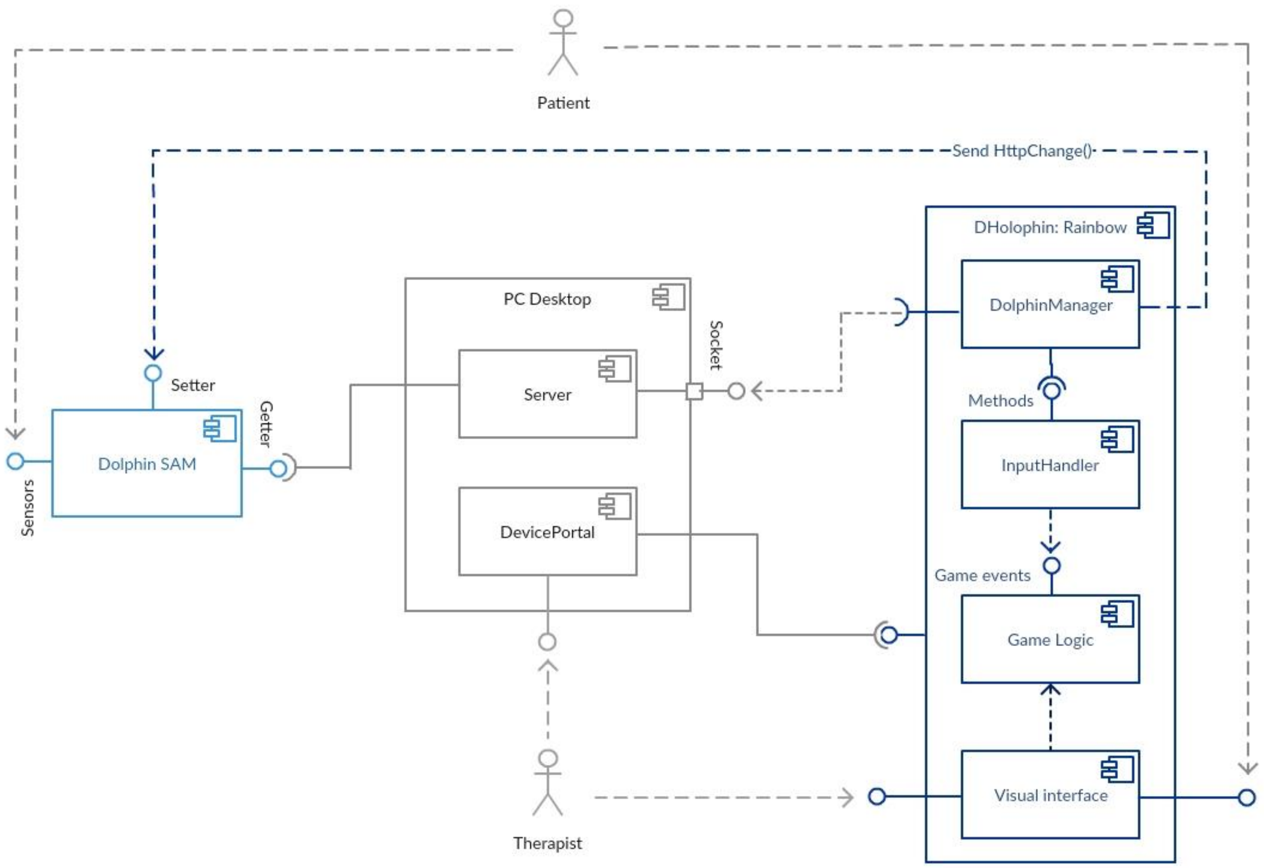
DHolophin: Rainbow is a UWP application developed in Unity and playable on HoloLens, that makes use of DolphinSAM smart toy as physical controller to interact with the game. A third component, a local server written in Python and running on PC, has been added to the system to interconnect the two devices, but it is a temporary solution and will probably be removed in the final version of the application (see Section 6: *Propositions and Improvements* for further details).

The current configuration of the implementation is shown in the following UML deployment diagram:



UML Deployment Diagram

More in details, the interactions between the physical and logical components of the system is represented in the next page component diagram:



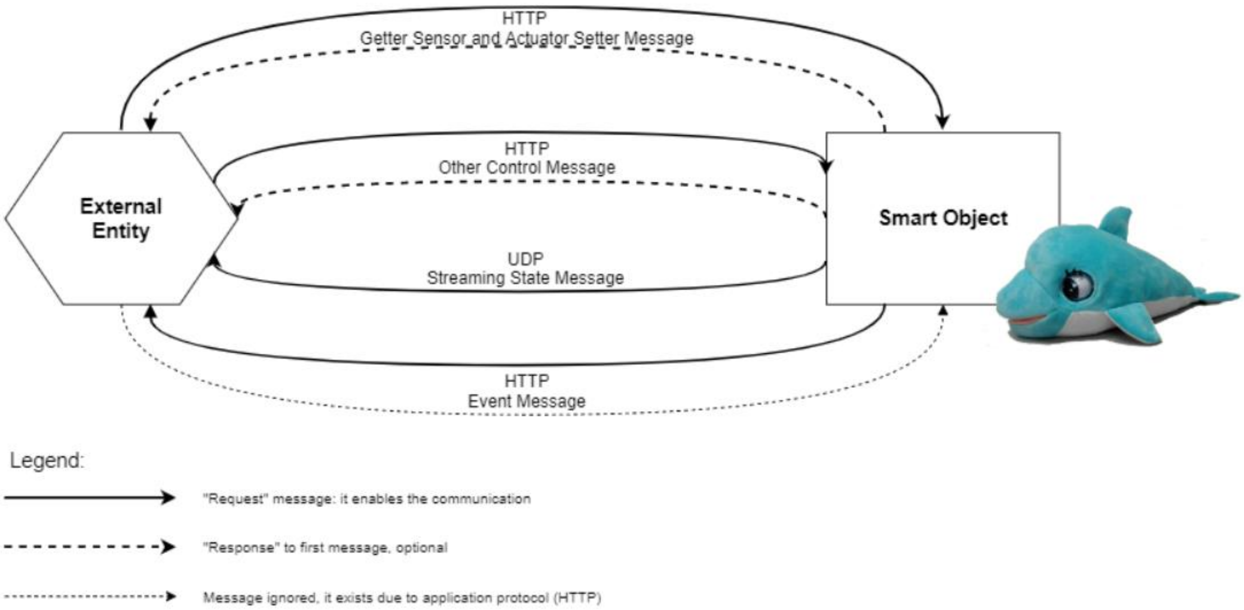
UML Component Diagram

As shown is the above figure, the three main components of the system are Dolphin SAM, the server and *DHolophin: Rainbow* application. We then proceed to analyze the different functionalities of these components.

5.2 DOLPHIN SAM:

Dolphin SAM is a smart toy for children with intellectual disabilities developed by i3Lab (Politecnico di Milano) in collaboration with *L'Abilita*, a therapeutic center in Milan that takes care of children affected by NDD. Dolphin SAM is equipped with touch sensors and RGB leds on body, belly and fins and it is able to move eyes and fins and emit sounds and colors as requested.

More technically, the API structure of the smart object is explained in the following image:



The smart object is able to send HTTP messages to a specific target in the local network each time an event detected by sensors occurs, and continuously provide a stream of UDP state message that update the target with the current state of the toy. Plus, it is possible to send HTTP message from any device in the network to the dolphin to make it move, change color of leds, emit sounds, ecc.

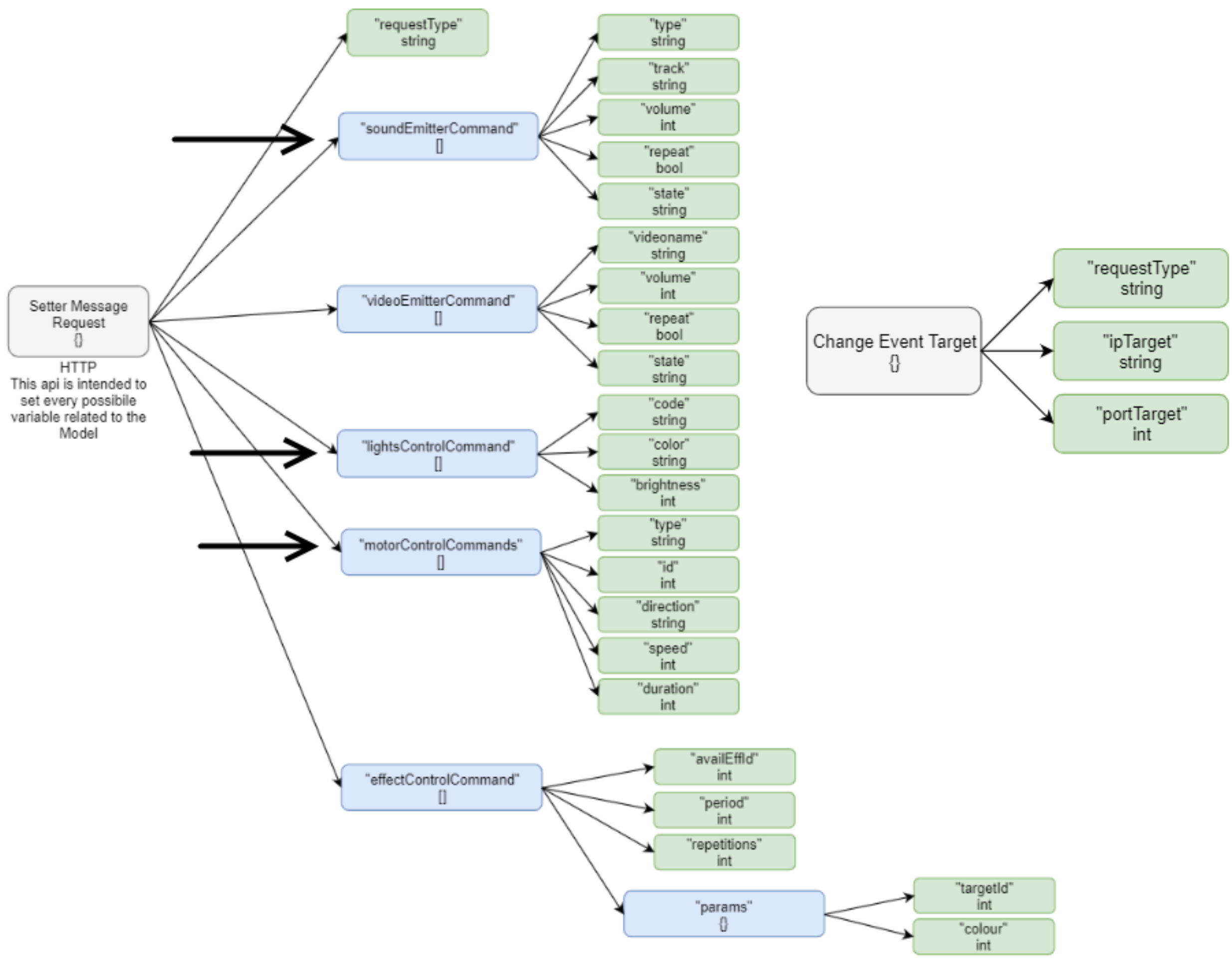
For the purpose of our application, we are interested in the following interactions with Dolphin SAM:

- The dolphin must change color when left and right fins are pressed;
- The dolphin must "send" the current color when the central fin is pressed;
- The dolphin must move and emit sounds when the user submits a guess.

So, the messages of our interest are:

TYPE:	FROM:	TO:	Description:
Setter	HoloLens	Dolphin SAM	Set server as target of event msg
Setter	HoloLens	Dolphin SAM	Change color of dolphin leds
Setter	HoloLens	Dolphin SAM	Make dolphin emit sounds
Setter	HoloLens	Dolphin SAM	Make dolphin move
Event	Dolphin SAM	Server	Left fin touched
Event	Dolphin SAM	Server	Right fin touched
Event	Dolphin SAM	Server	Central fin touched

which correspond to the following APIs of Dolphin SAM:



5.3 SERVER:

Given the difficulties of implementing a listening server on HoloLens (see Section 6: *Propositions and Improvements*) which could directly receive Dolphin SAM HTTP packets, we decided to temporary implement a simple desktop server written in Python that could act as mediator between the two devices.

The server firstly opens a socket, which is used by HoloLens application to connect as client, and then proceeds to listen to the HTTP event messages sent by the dolphin and immediately redirect them to HoloLens when they are received.

The code is the following:

```python
import socket
import logging
import json
import datetime
import sys, os
from colorama import init, Fore, Back, Style

from threading import Thread
from time import sleep

from flask import Flask
from flask import request

from subprocess import check_output

localPortDefault = 60001

localPort = 0
localIp = "0.0.0.0"

init()

log = logging.getLogger('werkzeug')
log.setLevel(logging.ERROR)

app = Flask(__name__)
app.config['ENV']="development"
holoClient = "client"
stringMessage = "a"

@app.route("/", methods=['GET', 'POST'])
def index():
    global stringMessage
    if request.method == 'POST':
        # print ("content lenght: " + str(request.headers.get('Content-Length')) + " " + str(request.data))
        if request.is_json:
            # try-catch to check json format
            try:
                # Get Json and convert to pure string, no u'
                dataStr = json.dumps(request.get_json())
                responseStr = "{\"status\":\"ok\"}"
            except:
                dataStr = "A request with no-json format is arrived"
                responseStr = dataStr
        else:
            dataStr = "A request with no \"Content-Type: application/json\" header is arrived"
            responseStr = dataStr


        # This complex and heavy functions to generate a date formatted like:
        # 2018-11-15 12:53:51.606
        (datetimeStr, micro) = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f').split('.')
        datetimeStr = "%s.%03d" % (datetimeStr, round(float(micro) / 1000.0))
        if '"act": 0' in dataStr and len(dataStr) == 65:
            print (datetimeStr + " " + dataStr)
            stringMessage = dataStr
        return responseStr
    else:
        return "You shouldn't use get method!"
```

```python
# Get real and actual IP address
def getIp():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        # doesn't even have to be reachable
        s.connect(("10.255.255.255", 1))
        IP = s.getsockname()[0]
    except:
        IP = "127.0.0.1"
    finally:
        s.close()
    return IP

#Send Message to Hololens
def check_messages():
    global stringMessage
    global holoClient
    while 1:
        sleep(0.05)
        if stringMessage != "a":
            holoClient.send(stringMessage.encode())
            stringMessage = "a"

if __name__ == "__main__":
    print(Style.BRIGHT + Fore.GREEN + "Smartifier x Esp - Event Receiver" + Style.RESET_ALL)
    print("")

    localPort = localPortDefault

    # open a port through windows firewall
    #command = "netsh advfirewall firewall add rule name=\"Open Smartifier Event\" protocol=TCP dir=in localport=" + str(localPort) + " action=allow"
    #print(command)
    #check_output(command, shell=True)

    localIp = getIp()
    print("Listening at " + Style.NORMAL + Fore.GREEN + str(localIp) + ":" + str(localPort) + Style.RESET_ALL)

    #Starting Connection with Hololens
    serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = localIp
    port = 60000
    print (host)
    print (port)
    serversocket.bind((host, port))
    serversocket.listen(1)
    print ('server started and listening')
    connected = None
    while connected == None:
        (clientsocket, address) = serversocket.accept()
        print ("connection found!")
        holoClient = clientsocket
        connected = True

    #Thread sending message to Hololens
    thread = Thread(target = check_messages)
    thread.start()

    #Server Flask start running
    app.run(host = "0.0.0.0", port = localPort)
```

5.4 *DHolophin: Rainbow* (HOLOLENS APPLICATION)

DHolophin: Rainbow is a UWP mixed reality application developed with Unity 2017.2.1f1, the represents the core and focus of the whole project. The application consists in a memory game of customizable complexity, where the player is required to memorize a sequence of colors displayed on a virtual panel and then re-insert the same combination when the color switch off.

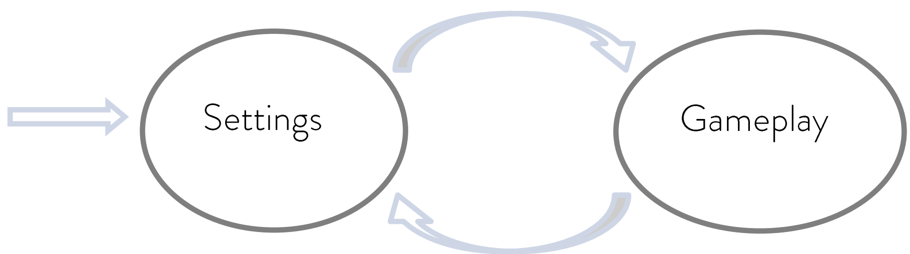
Dholophin: Rainbow is developed on Microsoft HoloLens and makes large use of the functionalities provided by MixedRealityToolkit, a collection of scripts and components intented for MR app development available at: https://github.com/Microsoft/MixedRealityToolkit-Unity .

The source code of the application is entirely available on GitHub on page https://github.com/AndreaMaffe/DHolophin-Rainbow .

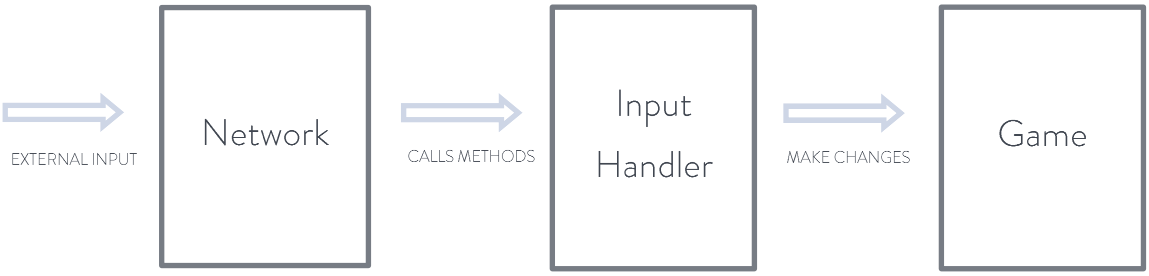
5.4.1 Application Logic:

The application consists in a first phase of settings, where the player has the possibility of configuring the complexity of the game by choosing the number of circles and colors that will be used later, and a second phase in which the game is effectively played.

In every moment of the game, it's possible to switch back to the settings and restart the game with an improved or reduced level of difficulty.

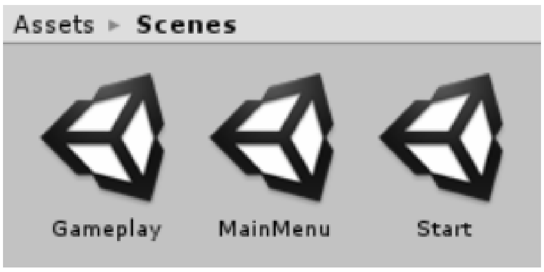


Game logic, networking logic and input handling have instead been treated as separated logic components, in such a way that that it could be easier to work on the separate parts without compromising the stability of the application. Furthermore, this modularity allows future changes in the structure of the application to be easily performed without the risk of possible conflicts.



5.4.2 Application Scenes:

The Unity project is made up of three Scenes: Start, MainMenu and Gameplay:

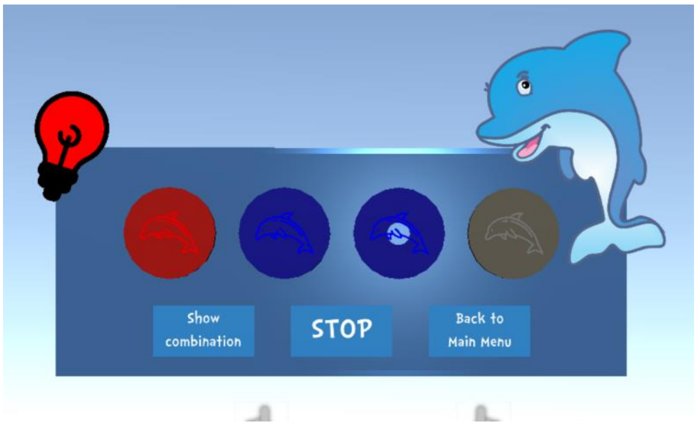


Unity Scenes

The first Unity Scene is called *MainMenu* and it is composed by a big panel where the user (eventually, the therapist) can select the number of circles and colors that will be used in the *Gameplay* Scene. The third choice to be made is between the two possible game modalities (see Section 3.3: *The game* for details).

When the *StartButton* is pressed, the application switches to the *Gameplay* Scene and the parameters set by the user are saved in the *GameManager*.

Main Menu



Gameplay

The third Scene, *Start*, is used as default launch Scene and it is needed to set some starting parameters for the game.

5.4.3 Application scripts:

The *GameManager* script is the main manager of the game logic. It contains all the main data about the current user play and provides methods to initialize the game, generate a new combination of colors, check whether the player guess is correct or not and modify the GameObjects in the current Scene.

The *AudioManager* script provides methods to manage the soundtrack of the game and to play audio effects during the gameplay.

The *DolphinHandler* script encapsulates the networking logic, allowing the application to connect via Socket to the Python server and sending HTTP messages to Dolphin SAM. The script also makes uses of the functions provided by *HttpMessage* class, which is a class built to hide the process of build the HTTP strings which provides high level APIs to send setter messages to Dolphin SAM.

The *InputHandler* script provides the effective interface on the game, defining methods that allows external entities to submit and change the current color of the game. Its methods are called by *DolphinManager* every time a new input is received by Dolphin SAM.

Other scripts are less important and will not be analyzed in this document.

6 PROPOSITIONS AND IMPROVEMENTS:

By dealing with the project, we faced different issues and problematics who brought us to better understand the used technology. Here is a summary of some technical observations pointed out during the development:

- HoloLens field of vision is quite poor, since the area of sight is imitated to a tight horizontal band outside which holograms are no longer displayed.
- Holograms provided by HoloLens tend to disappear when the user is too close, so every attempt to make a user "physically" interact with holograms may result in a failure.
- HoloLens only support UWP applications, so the only feasible libraries for coding are the ones admitted by Windows UWP apps. This may cause some issues, since traditional System libraries like *System.Net.Sockets* and *System.Threading* do not work anymore on HoloLens.
- HoloLens seems to have some some problems in receving messages via network. For still unknown reasons, all the attempts to implement a listening server on HoloLens using Windows *StreamSocketListener* libraries, resulted in a crash of the application when a message is receipt.
- Both setter and getter method of Dolphin SAM work via http messages; then, since after a request the dolphin is blocked until he receives the correspondent response, a high traffic of exchange messages may create conflicts.

At the current state of the implementation, *DHolophin: Rainbow* presents some critical issues that should be solved before effectively proceed with advanced expansions of the game. The main problem of the application is the lag between the input of the user and the effective response on the dolphin, which is caused by two factors:

- The message must be received by the server and forwarded to HoloLens, with a loss of time due to the dispatch.

- Dolphin SAM send two different messages each time a fin is squeezed: one for the press and one for the release. We make use of the second one because otherwise it's possible to have conflicts, since the dolphin receives the SetColor message when is in the middle of sending the release message.

We highlighted two possible solution for the problem:

- Replace the python server with a server running directly on HoloLens. A first trial has already made with the Microsoft *StreamSocket* libraries, but the code did not properly work. We advise to wait until the release of the new version of *HoloToolkit* to see if new network functionalities are added. This solution would reduce the lag in message exchanging and eliminated the need of the intermediate server.
- Modify the configuration of Dolphin SAM in such a way that the sensors do not send a message for release events.

We also suggest the possible two future improvements for the application:

- Add more animations inside the game, or an interactive environment to keep the player entertained during the gameplay. We mainly focused on making the system working efficiently, so such details in the game scene are missing.
- Possibly, add new game modes over the two already provided.

7 BIBLIOGRAPHY:

1. Actimage - Hol'Autisme https://www.actimage.com/en/credentials/holautisme-apprentissage-realite-mixte/. Accessed 5th of February 2019.
2. B. Aruanno, F. Garzotto, E. Torelli, F. Vona, "*HoloLearn: Wearable Mixed Reality for People with Neurodevelopmental Disorders (NDD)*".
3. A. Coppens, "*Merging real and virtual worlds: An analysis of the state of the art and practical evaluation of Microsoft Hololens*", pages 29 and 35.
4. F. David Rose, Ph.D., Barbara. M. Brooks, Ph.D., and Albert A. Rizzo, Ph.D., "*Virtual Reality in Brain Damage Rehabilitation: Review*", pages 242 to 249.
5. N. Petrovic, "*Aural Immersive Virtual Reality for Neuro-Developmental Disorder Therapy Based on Storytelling Approach*", 2017.
6. L. Escobedo, M. Tentori, E. Quintana, J. Favela and D. Garcia-Rosas, "*Using Augmented Reality to Help Children with Autism Stay Focused*," in IEEE Pervasive Computing, vol. 13, no. 1, pp. 38-46, Jan.-Mar. 2014.
7. L. Bozgeyikli, A. Raij, S. Katkoori, R. Alqasemi, "*A Survey on Virtual Reality for Individuals with Autism Spectrum Disorder: Design Considerations*".
8. B. Aruanno, F. Garzotto, "*MemHolo: Mixed Reality Experiences for Subjects with Alzheimer's Disease*".
9. F. Garzotto, M. Gelsomini, D. Occhiuto, V. Matarazzo, N. Messina, "*Wearable Immersive Virtual Reality for Children with Disability: a Case Study*".
10. Microsoft HoloLens. https://www.microsoft.com/en-us/hololens. Accessed 3 February 2019.
11. Spatial Mapping - Mixed Reality | Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixedreality/spatial-mapping. Accessed 3 February 2019.
12. Gestures - Mixed Reality | Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixed-reality/gestures. Accessed 3 February 2019.
13. Gaze - Mixed Reality | Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixed-reality/gaze. Accessed 3 February 2019.
14. What is mixed reality? - Mixed Reality. Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality. Accessed 31st of January 2019.
15. HoloLens hardware details - Mixed Reality. Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details. Accessed 31st of January 2019.

16. Unity development overview - Mixed Reality. Microsoft Docs https://docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview. Accessed 17th of December 2018.
17. Testing your app on HoloLens - Mixed Reality. Microsoft Docs https://docs.microsoft.com/en-us/windows/mixed-reality/testing-your-app-on-hololens. Accessed 31st of January 2019.
18. Using the HoloLens emulator - Mixed Reality. Microsoft Docs https://docs.microsoft.com/en-us/windows/mixed-reality/using-the-hololens-emulator. Accessed 15th of January 2019.
19. Microsoft Hololens https://www.microsoft.com/en-us/hololens/developers. Accessed 31st of January 2019.
20. Sockets - .NET. Microsoft Docs https://docs.microsoft.com/en-us/dotnet/framework/network-programming/sockets. Accessed 31st of January 2019.
21. Windows.Networking.Sockets Namespace – Windows Dev Center. Microsoft Docs https://docs.microsoft.com/en-us/uwp/api/windows.networking.sockets. Accessed 31st of January 2019.
22. Github - Windows Universal samples https://github.com/Microsoft/Windows-universal-samples/tree/master/Samples/StreamSocket. Accessed 5th of February 2019.
23. Github - Mixed Reality Toolkit-Unity https://github.com/Microsoft/MixedRealityToolkit-Unity. Accessed 3rd of February 2019.
24. Unity Documentation – Unity User Manual (2017.2) https://docs.unity3d.com/2017.2/Documentation/Manual/UnityManual.html. Accessed 31st of January 2019.
25. Kuzhagallyev T. FoxyPanda - TCP client in a UWP Unity app on HoloLens (2017) https://foxypanda.me/tcp-client-in-a-uwp-unity-app-on-hololens/. Accessed 7th of February 2019.