D3.4 Final prototype of the 3D-LIVE Platform

Deliverable data

Deliverable no & name	***D3.4 Final Prototype of the 3D-LIVE platform***
Main Contributors	*ARTS*
Other Contributors	*CERTH, CYBER, IT-INNOV*
Deliverable Nature	
Dissemination level	**PU** – Public – **X** **~~PP~~** – Restricted to other programme participants (including the Commission Services) **~~RE~~** – Restricted to a group specified by the consortium (including the Commission Services) **~~CO~~** – Confidential, only for members of the consortium (including the Commission Services)
Date	*23/03/2015*
Status	*Final Version*

Document history

Version	**Date**	**Author /Reviewer**	**Description**
V0.1	09/02/2015	B. Poussard	Initial Structure
V0.2	15/02/2015	B. Poussard	Golfing scenario section
V0.3	16/02/2015	B. Poussard	Coverage matrix additions
V0.4	18/02/2015	B. Poussard	Skiing scenario description
V0.5	23/02/2015	B. Poussard	Refinement of scenarios description.
V0.6	25/02/2015	B. Poussard	Conclusion + Architecture diagrams and appendices + scope + summary + structure of deliverable ° Introduction
V0.7	26/02/2015	B. Poussard	Reviewing paragraphs
V0.8	27/02/2015	J. Vatjus-Anttila	Jogging related descriptions
V1.0	09/03/2015	B. Poussard D.Zarpalas	Review tables and fix typos

Table of Contents

1 Executive Summary

The 3D-LIVE platform has been co-designed with end-users in an iterative way. 3D-LIVE partners have continuously worked on improving the components of this platform and their interoperability. We are presenting in this document the final version of the platform that was experimented with end users.

Basically the platform can be described as follows: Indoor users and outdoor users are running a set of technologies allowing them to be tele-immersed in one shared virtual environment. There are three main groups of components for both indoor and outdoor setups which are Acquisition, User Applications and Rendering. In the acquisition group, sensors track the activity of the users and transmit it to the user applications in order to generate a consistent representation of the user in the game. Once processed into the virtual scene, applications are capable of rendering it to different rendering devices depending on the setup. Those will be common devices like computer displays, Smartphones as well as immersive devices such as CAVE or Head Mounted Displays or smart goggles. The user applications, running on different platforms (Smartphones or computers) communicate through a server like an online multiplayer game. External data exchange has finally been set up in order to monitor two types of data through two different tools. One tool handles weather information aggregation and queries (including Environment Observation Service and Environment Reconstruction Service). The second tool, called ExperiMonitor, handles experimental data, collecting and monitoring measurements from the different user applications.

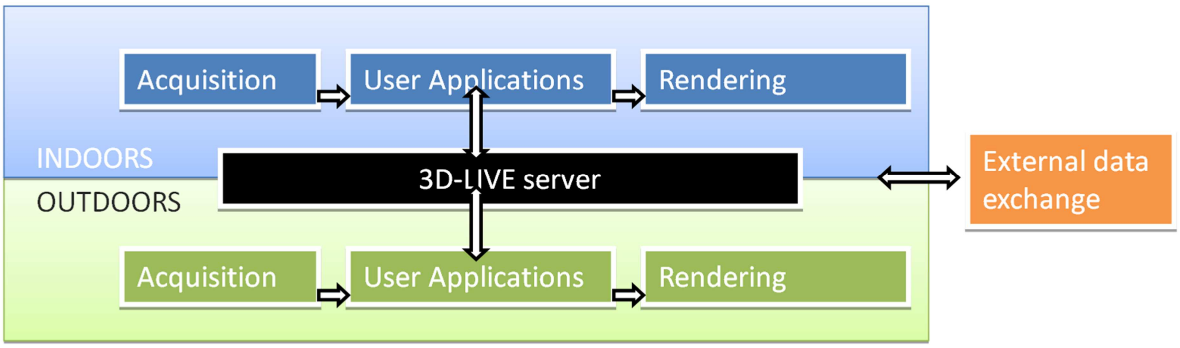

Figure 1: High level description of the 3D-LIVE platform

In the next subsections, we are presenting the list of all the components taking part of the platform, and a short description for each. Then the specific architecture deployed for each use-case is illustrated with diagrams detailing what is really inside the modules of the platform. Sections 3, 4 and 5 describe what experience users can live thanks to this platform, while tele-immersed into jogging, skiing and golfing scenarios. The entire project long, the consortium refined the platform architecture and scenario description based on users feedbacks. Indeed after each round of co-design iteration, including workshops and experiments for the three scenarios, users' feedbacks generated new requirements in terms of design and implementation. Those requirements have been aggregated in a traceability matrix, presented in section 6, and present the work we carried on towards the finalisation of the 3D-Live platform prototype.

1.1 Architecture of the second prototype

In the table below one can find the list of the main components found in the architecture of the platform, described on the next diagrams. Short descriptions are made available for each component, but for more technical information please refer to the D3.3 3D-LIVE Platform Modules.

Main components deployed in the platform	
Designation	**Description**
3D-LIVE indoor app	The indoor user application deployed to live the 3D-Live experience. It has been developed with Unity3D for the Golf and Ski scenarios, and with realXtend for the Jogging scenario. It aggregates data coming from other modules to properly run the gameplay.
3D-LIVE Smartphone app	The outdoor user application deployed to live the 3D-LIVE experience. It has been developed with Unity3D for the Skiing scenarios, and with Eclipse for the Jogging scenario.
Android tablet App	The outdoor user application deployed to live the 3D-LIVE Golfing experience. It has been developed with Unity3D.
Environment Reconstruction Service	The service collecting weather data from different sources and calculating the best interpolated information at any given GPS location. It allows then the applications to request weather and update the virtual environment accordingly.
EOS App	The Environment Observation Service app, is an Android app that collects local weather data from the Sensordone, to feed the Environment Reconstruction Service.

ExperiMonitor	The interface to monitor the experiments. It connects to a RabbitMQ server to allow multiple measurement sources to share live metrics evolution, and export the data afterwards.
Kinect Capturer app	The application that extracts data from the Kinect sensors, in order to create and send over the network either an animation frame of the user and a full reconstructed live 3D mesh of the user.
Main Screen app	The indoor application deployed for the Golfing scenario, used to render only the 3D content of the indoor 3D scene. It is made with Unity3D
Monitor Screen app	The indoor application deployed for the golfing scenario, used to render the game state and enable interactions in the game for the indoor user.
Mumble app	The Voice application service used in the Jogging scenario
RabbitMQ server	The server deployed to handle most of data flows not related to the game state itself. It uses the RabbitMQ protocol.
realXtend Server	Game server for the Jogging scenario. It handles all the data flows between indoor joggers, and by the intermediate web service, the data flows of the outdoor jogger. It uses the kNet network protocol.
Unity3D Server	The Game server for the Golfing and Skiing scenarios, it uses a custom game synchronisation API based on Network Views, Remote Procedure Calls and StateSynchronisation. The RakNet library is the back-end solution to that API.
Oculus Rift	The Oculus Rift is a Head Mounted Display (HMD) displaying an image with a wide field of view to a user to let him immersed in a virtual environment. Moreover it can track the orientation and the position of the head of user in order to make his point of view always consistent in the virtual environment.
CAVE	Cave Automatic Virtual Environment. (CAVE) is a system composed of a set of stereoscopic screens surrounding a user. The added value of a CAVE is that it can track the point of view of the user (orientation and position tracking of the head) to adapt the image rendered on each screen. The user has then the feeling to be inside the virtual environment but can still see his own body inside this virtual world.
Recon Snow 2 HUD	The Recon Snow 2 Heads' Up Display is a ski goggles including an Android OS, sensors and a little screen in the right corner. This allows connectivity to a Smartphone for instance to display some data. In 3D-LIVE we diverted the original use of these goggles to render in it a contextualised view of the virtual environment.

Table 1: Description of the main components deployed in the platform

1.1.1 *Architecture of the Jogging prototype*

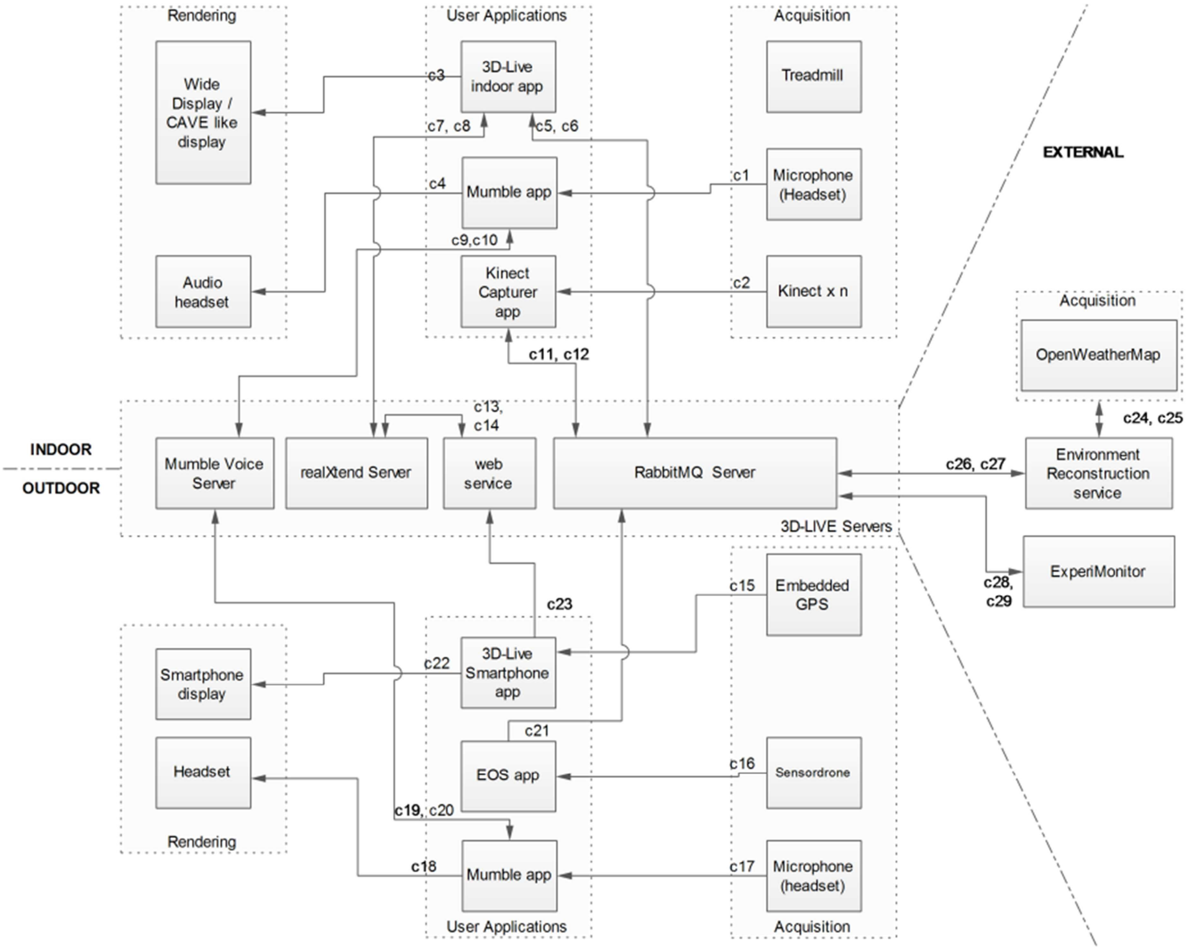

Figure 2: Architecture of the Jogging use-case deployment

1.1.2 *Architecture of the Golfing prototype*

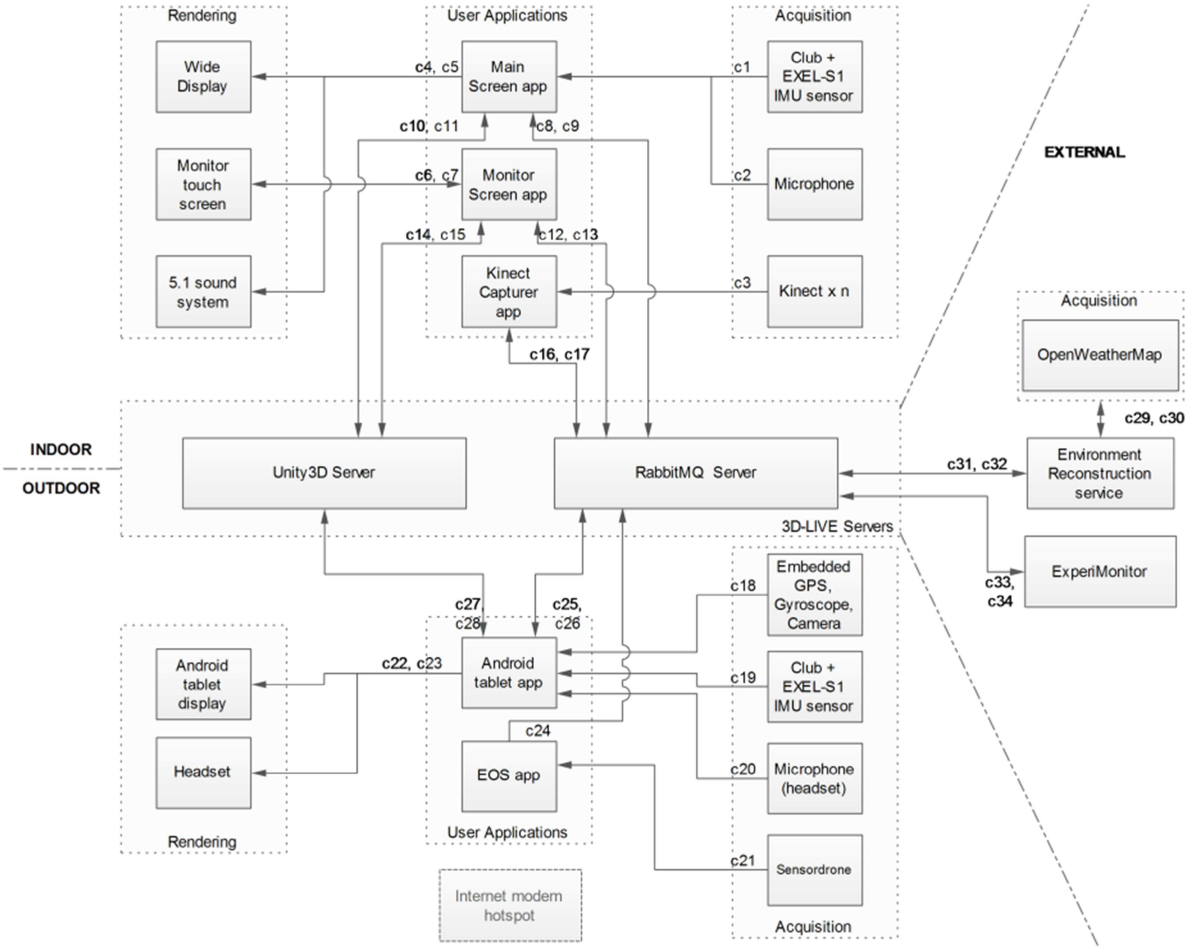

Figure 3: Architecture of the Golfing use-case deployment

1.1.3 *Architecture of the Skiing prototype*

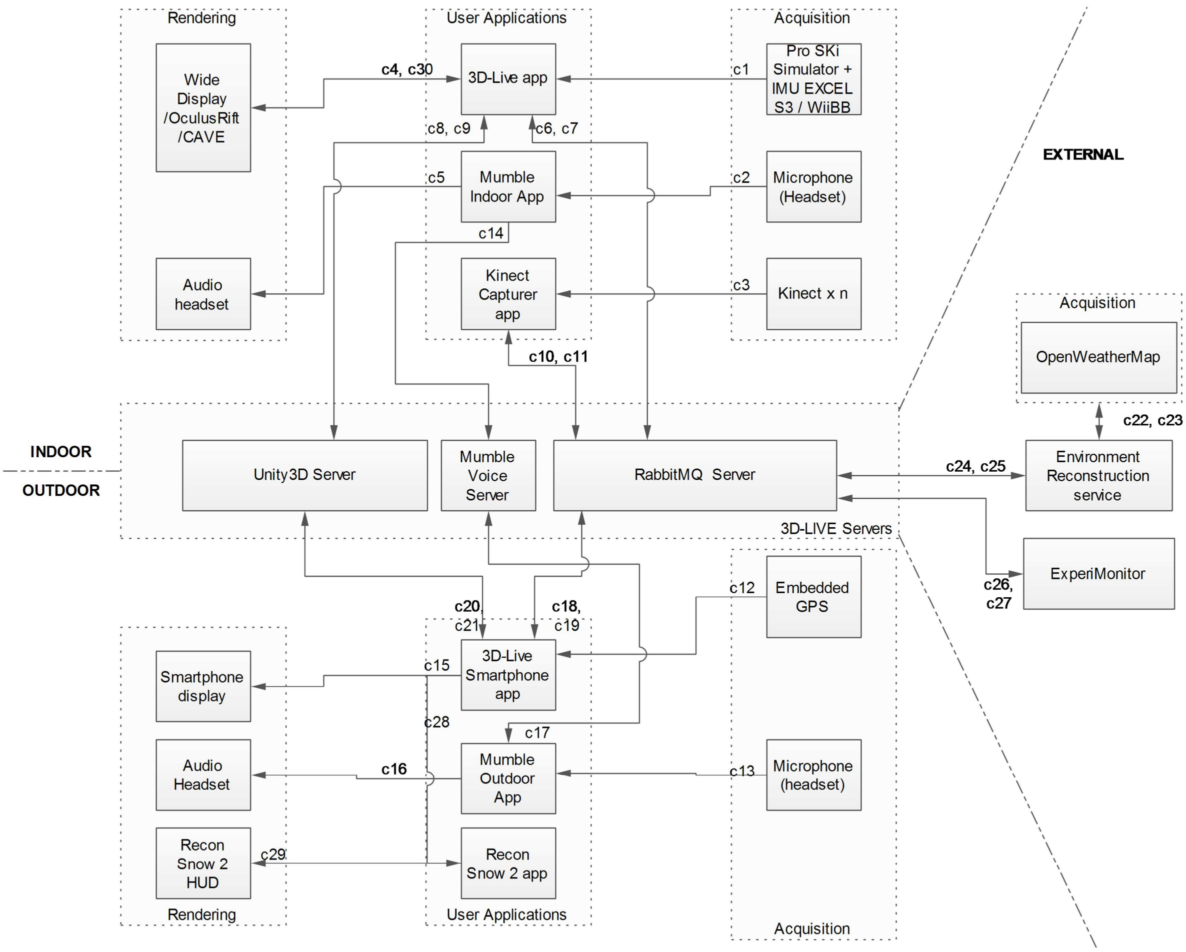

Figure 4: Architecture of the Skiing use-case deployment

2 Introduction

2.1 Summary of 3D LIVE Project

The 3D-LIVE project aimed at developing and experimenting a User Driven Mixed Reality and Immersive platform connected to EXPERIMEDIA facilities (FIRE testbeds) in order to investigate the Future Internet (FI) requirements to support Real-Time immersive situations as well as to evaluate both the Quality of Experience (QoE) and Quality of Services when users are fully immersed into Future Internet (IoS and IoT) based live (sport) mixed environments. The main objective of 3D LIVE was to explore 3D/Media technologies and Internet of Things (IoT) in order to construct applications bringing together real and virtual environments, among distant users with varying requirements. The combination of FIRE testbeds and Living Labs enabled both researchers and users to explore Future Internet capabilities to enter the Tele-Immersive application market and to establish new requirements for Internet technology and infrastructure. It is expected that, combining both FI technology and Tele-Immersion market, will promote and accelerate the creation and adoption of innovative sportive events based on FI Services, by user communities (e.g. sport practitioners). The consortium built and evaluated the final prototype of the 3D-LIVE Tele-Immersive platform. A scenario for three sporting use-cases (Golf, Ski, Jogging) has been experimented with end-users where they were fully immersed in one shared experience regardless they were indoors or outdoors. The modules taking part of the platform were validated and the resulting User Experience assessed by our UX evaluation methodology.

2.2 Partners in 3D LIVE project

- CENG – Collaborative Engineering, Italy (Project Coordinator)
- CERTH-ITI – Centre for Research and Technology Hellas, Informatics and Telematics Institute, Greece
- IT Innovation – University of Southampton, UK
- ARTS – Arts & Metiers ParisTech, France

- SportsC – SportsCurve, Germany
- CYBER – Cyberlightning Ltd, Finland

2.3 Purpose, Intended Audience and Scope

This deliverable presents in a very high level way the architecture of the 3D-LIVE Tele-Immersive platform and the content inside each scenario. It explains what technologies were deployed in terms of software and hardware to allow tele-immersion of distant users, and the storyline of each scenario, illustrated with game phases and pictures.

This document does not describe the technical details of the presented features, hardware, and algorithms that were used in the 3D-LIVE project. For more information about the technical aspects of this platform, please refer to the "D3.3 3D-LIVE platform modules" deliverable.

The following description of the prototype intends to help designers of Tel-Immersive platforms conceptualising the required technologies to build a Tele-Immersive environment, as well as understanding what are users' expectations and requirements in this kind of augmented sports applications.

2.4 Related 3D LIVE Documents

AD(1). 3D LIVE DoW

AD(2). 3D LIVE D1.1 Investigate and Formalise an Experiential Design Process

AD(3). 3D LIVE D1.1 Study and Create the Holistic User Experience Model

AD(4). 3D LIVE D2.1 Report on the Conceptual Design of the 3D-Live Platform

AD(5). 3D LIVE D2.2 Report on the Needs and Requirements of the 3D-LIVE Platform

AD(6). 3D LIVE D3.3 3D-LIVE platform modules

AD(7). 3D LIVE D4.2 Final Report on the experimentation & evaluation of the 3D-LIVE Tele-Immersive Environment

2.5 Brief work packages description

The work packages of the 3D-LIVE project group the tasks over the time considering 3 steps: i) Concept Modelling, ii) Exploration and Prototyping and iii) Experimentation and Evaluation.

WP6 and WP5 are continuously active during the entire project as they consist in management and dissemination, respectively.

WP1 is related to the design of the UX Model & Methodology, correlated to the WP2 that is the design of the 3D-LIVE platform. In parallel with the Design phase, WP3 starts the prototyping of some components of the platform, while WP4 prepares the experimental sessions. Once the first prototype is validated, there is an iterative process that consists in experimenting with and evaluating the platform and, then, a new design phase is necessary to deliver an improved prototype. 3D-LIVE includes three iterations allowing to deliver the third version of the prototype evaluated by users, which is introduced in this document.

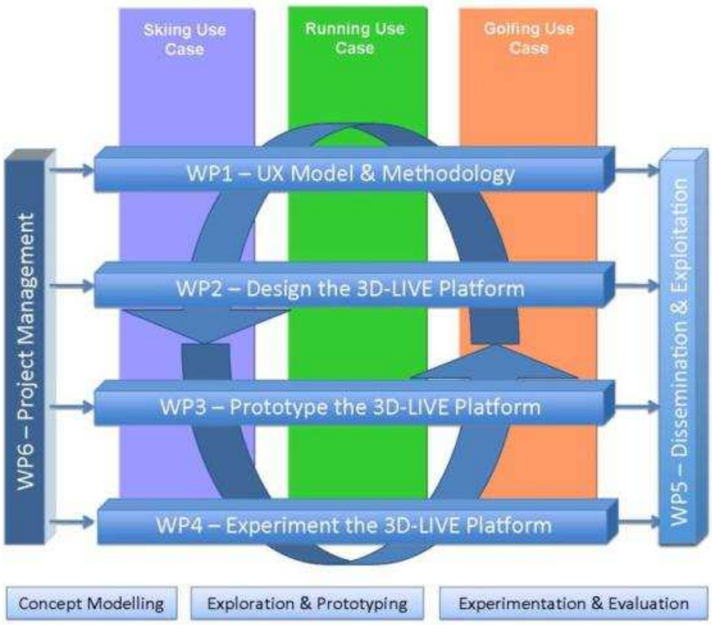

Figure 5: Functional description of work packages

2.5.1 *Positioning of current deliverable*

"D3.4 Final prototype of the 3D-LIVE platform" is the result of the work done across the workpackages 1 to 4. The iterative design, prototyping and evaluation through experimentations have led us to a final version of a tele-immersive platform applied to three augmented sport scenarios.

The deliverable is linked to the Workpackage WP3 – Prototype the 3D-LIVE platform, as it describes the prototype's components and what users can experience with it; but it represents the culmination of a consortium's effort into each conceptual task of the project.

2.6 Structure of the Deliverable

This document is built in three main sections presenting the high level description of the 3D-LIVE final prototype. First of all, we introduce in section 3 the architecture of the platform, what the main components and technologies are in 3D-LIVE, and how these technologies do connect with one another. For the three use-cases part of 3D-LIVE, we describe the specific components providing the tele-immersion to joggers, golfers and skiers.

In sections 4, 5 and 6 we detail the story users experience with the 3D-LIVE across the three sporting scenarios. We cover the overall gameplay, the rules, the actions users need to achieve and how they are engaged in the game, and illustrate that using pictures taken from the latest experiments.

In sections 7 and 8, the 3D-LIVE consortium recapitulates how the users' requirements were covered in terms of features and scenario elements, and opens a discussion about areas for improvement in case of a product development.

3 Jogging scenario

3.1 Overview of Prototypes 1 and 2

The 3D-LIVE jogging prototype has seen gradual improvement over the course of the successive prototypes. Prototype 1 already had all major software components implemented, including 3D human reconstruction on indoor low-cost scenario, and prototype 2 further refined and polished all the components. Early concept of the UI was introduced in prototype 1 which saw major improvements for prototype 2.

Both of the early prototypes allowed two indoor users and one outdoor user to connect to same game server and interact with each other on the jogging venue located in Oulu, Finland. Indoor users joined the game from Oulu university lab located in Center of Internet of Excellence premises, and from Thessaloniki Greece CERTH labs. Outdoor user naturally located physically in the actual venue in Oulu. Users could talk to each other through VOIP protocol and their movements were replicated accurately in the virtual world. Actual goal of the scenario was to run through the 1km course collaboratively. Only downside was the lack of visual feedback for the outdoor user, because he could only see the map of the route and talk to other users.

Figure 6: Testing the system in jogging LIVE1 and LIVE2

3.2 Description of Prototype 3

The final prototype of the 3D-LIVE jogging application brings some minor technical improvements in addition to requirements which were lacking in LIVE1 and LIVE2 prototypes.

In prototype 3 three users enter the scenario; two indoor and one outdoor jogger. Number of the indoor joggers is not restricted to two joggers but on the other hand outdoor jogger amount is fixed to one jogger at a time. Gameplay focuses entirely on collaborative running and immersion of the individual joggers over the duration of the course. Outdoor jogger carries Sensordrone-sensor which enables local weather capture in micro scale. This can be combined to macro scale weather data which is then used to replicate weather conditions and data to the indoor joggers.

Starting the scenario outdoors

Outdoor user enters the scenario by locating himself to the agreed starting location of the jogging scenario. Jogger is provided with an android smartphone and a Sensordrone-sensor, which connects to the android smartphone through Bluetooth protocol. User is also provided with Bluetooth headset which is used for voice communication with the joggers indoors. Outdoor user starts the Android application which connects the user to the Mumble VOIP –server and also enables the GPS-location capturing. This location data is sent over the 3G-connection to the indoor clients who use the data to replicate outdoor jogger avatar representation in their virtual version of the venue. Outdoor user can see the map of the venue, course highlight and current location on the android application. Voice connection is immediately enabled to the indoor joggers when applications are fully operational. After the initial setup jogger can choose to lock the smartphone screen and put the device in his pocket for better convenience while running, or keep the device at his hand if he is unsure what direction he should be going at. Voice communication is used to check if he is far ahead or behind of the other joggers. This is also the greatest drawback on outdoor jogging scenario because jogger does not have direct visual feedback where the other joggers are currently running.

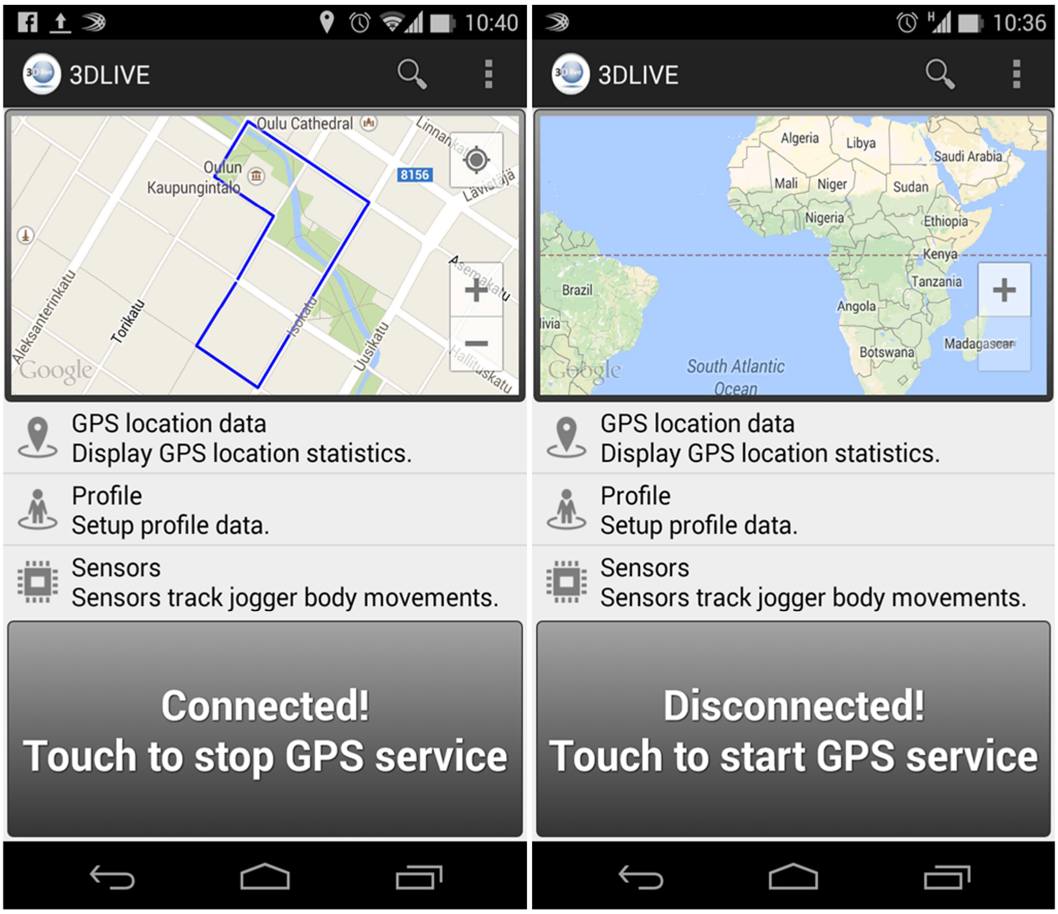

Figure 7: Outdoor application on connected and disconnected states.

Starting the scenario indoors

Indoor jogger starts the client application and is greeted by the main menu of the application. User can choose which 3D-LIVE server he is about to connect to and change the configurations for Kinect capturing or ECC reporting, but this is not necessary because the test setup is already automated and in working order for the scenario. User selects the indoor jogger configuration from the application which initiates the connection to the remote 3D-LIVE server which handles the game state synchronization between clients. Kinect capturing starts automatically after the connection is ready and virtual jogging venue is loaded. All users spawn at the starting location and indoor users see outdoor user avatar locating himself to the reported GPS-location of the actual outdoor jogger. Similarly as in outdoor case, indoor joggers can immediately talk to each other through voice communication connection. When everyone is present and ready to start running, users talk and agree to start the course together. When everyone has reached the midpoint of the course, parameters of the Kinect capturing is altered without users knowing this and people are encouraged to finish the course together. After the course has been finished users are greeted with their performance analysis which can be used as comparison to other runners.

Figure 8: Indoor runners compilation and main menu.

3.3 Areas for improvement

Being the final prototype of the 3D-LIVE application does not mean it goes without flaws. While much of the application has improved from the user trials through prototype 1 and 2 there are few problematic areas that need more study and better implementation on the software side.

- Tracking the skeletal motion of the outdoor jogger. Multiple IMU sensors are needed to track outdoor jogger body movements but lots of technical obstacles remain. Combining the predefined avatar movements with real jogger movements for smooth visual animation. Accuracy of the sensors is also one area of problem with possible drift and keeping the sensors glued to the initial calibrated position on joggers arm or leg. Also transmitting all the data multiple times per second to the indoor clients through 3G might cause lag or other unwanted stutters.

- Visual representation of the indoor joggers for the outdoor jogger without disturbing the running experience from the safety and convenience point of view, meaning it would not obstruct real world or does not cause confusion and possible disorientation.
- More realistic rendering of the indoor graphical view is required. Many areas graphically are not on the par with currently best computer games to date. Therefore, visual representation could be made a lot higher quality to immerse people better on the scenario.

3.4 Final jogging configuration

Indoor Low-Cost	Indoor High End
• Treadmill • Monitor • Kinect reconstruction of the user for indoor high-end • Bluetooth Headset • Desktop Windows PC	• Treadmill • CAVE-system • Bluetooth Headset • Kinect motion capturing (Skeleton only) • Desktop Windows PC
Outdoor Low-Cost	**Outdoor High End**
• Android Smartphone • Bluetooth Headset • Sensordrone-sensor	• Android Smartphone • Bluetooth Headset • Sensordrone sensor • 4 units of wearable motion sensors for motion capturing • Heart rate monitoring

4 Skiing scenario

4.1 Overview of Prototypes 1 and 2

The 3D-LIVE ski prototype has been significantly improved since the first user trials of the prototype.

Prototypes 1 and 2 were early versions which allowed us to check the technical choices we made and refine them and refine the scenario to improve the User Experience. In LIVE 2 the application allowed three users to connect and join a ski game on the Schladming Hopsilift slope – a very short slope easy to ski down. The indoor users were able to use a Pro Ski Simulator or a Wii Fit Board to control their avatars, while the outdoor user had only his Smartphone in his pocket to join the game. They were able to communicate through voice service, but the outdoor skier had no screen and no visual feedback to understand the actual status of the game.

Figure 9: First tests of the skiing prototype on the Hopsilift ski slope

4.2 Description of Prototype 3

The third prototype of the 3D-LIVE platform included most of the User Requirements in addition to technical improvements.

The scenario is now much more immersive and engages skiers in a challenging mixed reality ski race: in this scenario the number of indoor players is not limited by the game, (it is actually limited by the network capabilities to send full human reconstruction data to the connected clients). One outdoor only can join the game, and the gameplay focuses more on his experience whether an outdoor skier is connected. The game automatically adapts the content depending on where the outdoor is and what his current activity is. All the players do have visual and audio feedbacks of the remote racers, and several configuration deployments are available, providing different levels of immersion in the game. If the setting named “ERS” is enabled, the weather information on site will be collected by the game in order to render the visual weather effects like cloud coverage, rain, snow, temperature and wind. If one of the indoor is using Full Human Reconstruction, he will see himself represented as he is instead of an avatar, and his representation will be sent to the other indoor players.

One outdoor user joins the game:

Once all the expected users are connected, the game triggers a meeting phase. The outdoor skier sees in his Recon Snow2 Head’s Up Display the different avatars of the remote players animated live, and their profile information. On the other side the indoor players are teleported next to the outdoor user’s location in the virtual environment. They can meet, talk and get ready for the race. The goal of this phase is to start engaging the users in the game, and virtually minimize the physical distance between them. As they feel they are close to one another they feel more challenged by the race.

Figure 10: Intro and Meeting screen in the Recon Snow2 HUD, with animated avatars of remote players

Figure 11: The indoor skier at the meeting point, wearing an Oculus Rift and tracked with a Kinect sensor.

If the meeting is done at the bottom of the slope, the outdoor user has to take the lift. This automatically ends the meeting phase in the game. The indoor users are then teleported at the starting gate at the top of the slope. They can race just together to practice with and without gates, or just wait for the outdoor user. In the meantime, the outdoor skier is in a "lift phase", where he is able to have a look at the virtual mountains with the pins of the different players on it, so he can follow the live race of the indoors. Once he reaches the top of the slope, the application automatically triggers a new phase in which the indoor players are automatically teleported at the starting gate waiting for the outdoor skier, while the outdoor player can see a virtual starting gate and the other players in his mask. He just has to ski down towards them and settle under the starting gate, and the indoor players are able to see him going down to the start line.

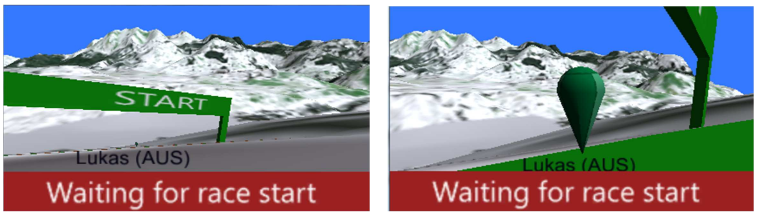

Figure 12: Two different views of the virtual start line and racers pins in the HUD

Figure 13: The four screens representing a ski competition between three players live from remote places.

Then they can chat and as soon as they are ready, the outdoor user can press a button on his wrist controller in order to trigger the countdown. At the end of the countdown, all the players can start racing when they see and hear the GO message.

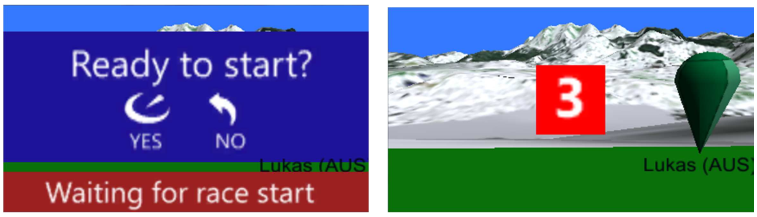

Figure 14: Views of the HUD at the starting line: press main button to trigger the countdown

Figure 15: An outdoor skier ready to start. He needs to press a button on his wrist band to start the countdown.

The game triggers now the race phase. While in race, the indoor players can use their input controller (Wii Fit Board, Ski Simulator, gamepad or keyboard) to control their avatar on the slope. Activity Recognition allows them to accelerate by pushing on their ski poles and gain speed by bending to hurtle down the slope, thanks to Microsoft Kinect sensors. The outdoor skier can race taking care of his safety if the slope is crowded. In order to see the actual positions of the different players a bird's eye view of the mountain with pins updated in real time is available, but a feature exists to hide the screen in the HUD if the speed is too high because looking at the screen at high speed is dangerous.

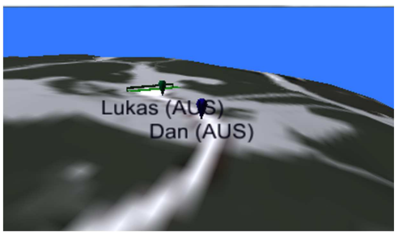

Figure 16: The bird's eye view in the HUD, showing pins of the different players

Figure 17: User Interface of the indoor application depending on the setup

If the real slope conditions are too bad, the outdoor skiers will require more time to go down the slope so a slalom mode with gates is enabled to balance the difficulty of the game.

Once the players reach the finish line, a gong resounds and a congratulations message pops up. The time of the race is made available for every single player. Once all the players reach the finish line the scores are displayed on the screen. The outdoor skier can see a podium with the avatars of the different players while the indoors can look at the scores panel popping over the virtual environment. The outdoor can then continue to the lift to trigger the end of the session and start again.

Figure 18: The finish line, podium and results displayed in the HUD

If no Outdoor joined the game

The normal game between two indoor users just runs smoothly the same phases. The only difference is that the gameplay is not driven by the activity of the outdoor skier.

4.3 Final ski configuration

Indoor Low-Cost	Indoor High End
Wii Balance Board Kinect Motion Capture + Full Reconstruction Wireless Audio Headset Wide Screen or HMD 3DLive Ski indoor app	Pro Ski Simulator + IMU sensor EXEL S3 Kinect Motion Capture (Skeleton only) 5.1 Sound system + microphone CAVE 3DLive Ski indoor app
Outdoor Low-Cost = Outdoor High End	
Smartphone + Recon glasses Earphones 3DLive Ski outdoor app	

4.1 Areas for improvement

In the skiing scenario, the setup has been improved a lot and most of the requirements were covered. However, we are willing to explore the possible improvements in terms of spatial representation of the outdoor skier, and graphics quality of the overall scenario.

In order to achieve a more accurate representation of the outdoor skier in the virtual environment, two parts need some experimentations and validation:

- The first one is the improvement of the position prediction of the skier. The 3D-LIVE team has actually integrated a tool to predict the skier's trajectory based on a fusion of GPS data filtered with inertial sensors' data attached to the torso of the player. Unfortunately, even if the technical approach has been tested in our laboratories this has not been experimented on the slope and the algorithm needs to be validated in real conditions. This improvement could provide a better consistency in terms of spatial and temporal positioning of the outdoor skier.

- The animation of the outdoor skeleton seems to be an interesting missing component in the platform. The 3D-LIVE consortium has been investigating the ways to get this animation on the real person, but the actual solution would be to wear a specific suit including sensors for each bone connected to the Smartphone, like the one proposed by the Heddoko company[1].

More than these features, immersion can be improved by the quality of the graphical elements in the virtual scene and the way it is rendered to the user. For instance the 3D modeling and texturing of the virtual environments could be made more realistic and complex.

For the outdoor skier, one of the areas that need to be investigated is the rendering of the game into the ski mask. Augmented Reality was abandoned due to users' requirements, but the recent availability of new Augmented Reality ski products like the RideOn[2] ski mask could open to new opportunities in terms of gaming scenarios.

[1] Heddoko company: http://www.heddoko.com/

[2] RideOn product : http://www.rideonvision.com/

5 Golfing scenario

5.1 Overview of Prototypes 1 and 2

The initial version of the 3D-LIVE Golfing prototype did include almost all the technical components designed to play a scramble Golf game, but did not include all the features and scenario elements required by golfers.

Two users were basically able to play a game remotely being in the simulator or on the actual Laval Golf course in France. The goal of the game was to play as a team to put the ball in the hole in a minimum of strokes, like in a normal scramble mode of play. The indoor user was playing in front of a wide screen displaying the virtual environment, on which he was striking the specific indoor golf balls. Next to him a touch screen was showing the map of the course, the summary of the scores and different events. Indoor users were using real golf clubs to strike balls since the second version of the prototype. A Bluetooth Inertial Measurement Unit was attached to his club so the shots parameters were calculated though a very first version of activity recognition algorithms. The outdoor users were using an Android tablet attached to a golf cart to play the game and display the activity of the remote indoor golfer thanks to Augmented Reality. The module to detect the parameters of shots was not available outdoors, so the balls positions were only validated by GPS location by the player after each stroke.

5.2 Description of Prototype 3

In the third prototype, basically the same technical configuration of the system has been deployed but the scenario has been reviewed in order to match all the users' requirements from the previous experiments.

Now both indoor and outdoor golfers do have the ability to play a scramble with the same equipment: 1 golf club with a sensor attached.

The two players must start their app and press GO in order to join a game. Once they are connected they can meet and talk while the application finishes loading on the Android tablet of the outdoor user. The loading time for the outdoor application can be around one minute or so, in order to get all the virtual elements in the scene with physics ready, and other external modules connected to the application (Environment Reconstruction Service, Skeleton data server, ExperiMonitor). The first round can start after all these modules are loaded and users will have to play one after the other and strike the ball towards the hole, like in a real scramble mode of play.

Figure 19: The Laval Golf course, Hole 1 - Par 4

Once the applications are fully loaded the players can get ready for the first strike. The indoor golfer must take the first stroke while the outdoor golfer is keeping an eye on him, and then the outdoor can take his shot while the indoor golfer can look at him doing.

Figure 20: An indoor pro golfer striking the ball

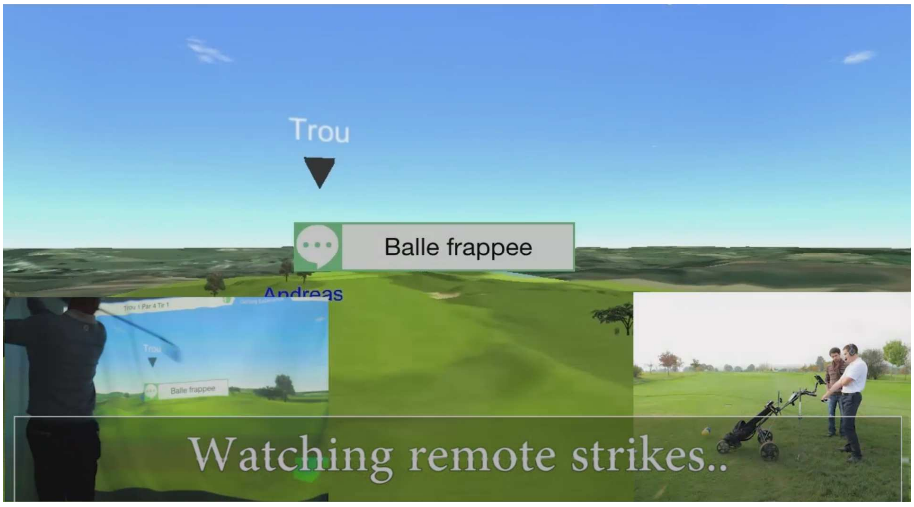

Figure 21: The outdoor golfer watching the indoor striking the ball through Augmented Reality on a tablet

To look at the indoor golfer taking his strike, the outdoor player can stand a few meters away from the ball position, and target the virtual indoor player thanks to augmented reality on his tablet. He can then see the indoor avatar animated striking the ball, then the distance travelled by the ball and the remaining distance for this stroke popup on both applications.

On the other side to look at the outdoor player striking the ball the indoor player just has to look at the wide screen in front of him where the view will focus on the outdoor virtual avatar animated in real time while doing the swing gesture. To validate the actual position of his ball the outdoor user must go find his ball after each stroke, and confirm in the application the real position in case shot parameters or outdoor weather conditions are not accurate enough. To do so, the augmented reality feature let him target his ball with a sight on the screen and the GPS location will be directly retrieved and shared with the remote player.

The avatars are animated thanks to predefined animations during the swing phases, and the value of the sensor attached to the golf club tells the application how to animate them.

Figure 22: The outdoor golfer just struck his ball. The ball took off in the indoor app.

After each round, the two players must talk and agree on which ball to restart from. On their different screens, a button is made available to select the ball they want to keep. The system validates the ball only if the two players have selected the same ball in the application. If the ball of the indoor player is chosen, the outdoor user must replace his ball at the proper location on the course. To help him an Augmented Reality feature in the tablet guides him towards that location and display an area where it is acceptable to throw the ball (this area corresponds to the radius of GPS accuracy). The outdoor player confirms to the app he correctly replaced his ball and the next round can start. In the meantime in the indoor simulator, the indoor player's avatar automatically follows the outdoor player and they can chat while walking to the target location. If the ball of the outdoor player is selected, the outdoor player has nothing to do as he will start from the current position of his ball, and the indoor player's ball will be teleported to that same location.

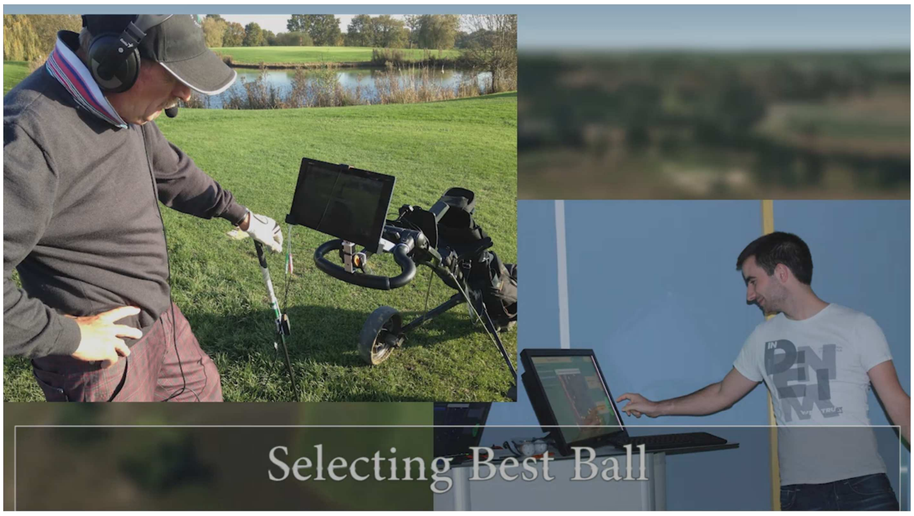

Figure 23: The two users must chat and select the best ball

The two players iterate until they put the ball in the hole. If the indoor player manages to put the ball in the hole, then the application automatically detects the end of this successful game, and triggers the screen summarizing the final results. If the outdoor player put the ball in the hole, the accuracy of our sensors here is not high enough to detect the event. Consequently the outdoor player must push a button in his app whether he finishes the job to tell the game is over.

Figure 24: The outdoor golfer looking at the final stroke

The same screen with the final results appears on both applications inside and outside. On this final view, users can look at the overall performance (Eagle, Birdy, Par, Bogey, Double Bogey,etc.). It displays also who put the ball in the hole, what was the longest stroke, the shortest stroke and which balls were selected during the game. A map next to these scores displays also the trajectory of the different balls chosen during the game in order to optimize the gestures for a future game.

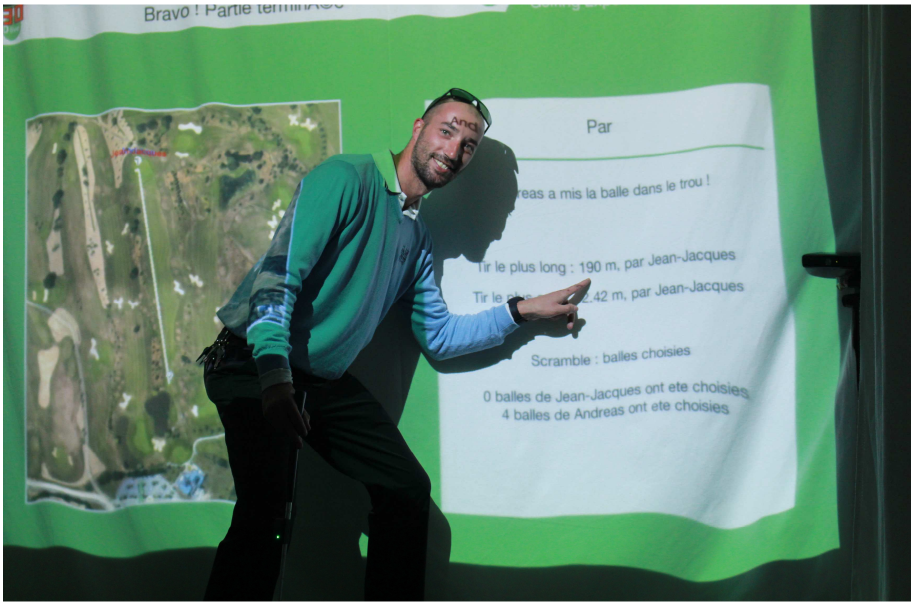

Figure 25: The final screen, the pro golfer and his best record.

5.3 Areas for improvement

In the golfing scenario, several aspects of the platform have been evaluated by the users and the 3D-LIVE consortium. Below is a summary of areas for improvement regarding the current prototype.

The gaming scenario

First of all in terms of the scenario, users were willing to see what those technologies could bring to the golfing practice, but it seems that it is too complicated for only a gaming purpose. Actually golfers really enjoyed the concept of playing on a golf course with remote people, but the Augmented reality and the tracking of the skeleton information is not a required feature anymore. The main areas of improvements proposed by our users aim to optimise the flow of the game and simplify the interactivity between players in order to create a robust and valuable product. For instance, we proposed a fully synchronised game here which was disturbed by game phases constraints and network failures. After our experiments users realised that due to the preparation phase of the players in order to detect their strokes, the time required for each round is definitely too long. Moreover the golfers are not really interested in watching the skeleton animation of the remote players. What they finally want is a easy to setup, quick social golf game between users really on the field and users in simulators. Users in simulators must play as they usually play while users on the field must have access to an app allowing us to connect to a game, talk to the players and have access to asynchronous evolution of the game. Practically golfers don't care of seeing the activity of the remote player when he strikes the ball (in the scope of a gaming scenario) but they want the results of the strokes and they want to be able to play at the same time to save time while playing and consequently save money.

To conclude on this first area of improvement, the appropriate gaming scenario for the Golf would not be a mixed reality platform, but more a social multiplatform game based on existing technologies and with no hug amount of data to transmit in real time. It would result in a web service connecting:

- Any existing indoor simulator

- A dedicated new mobile application linked to radars to have the best accuracy in ball trajectories.

The training scenario

Considering a training scenario, the professional golfers were really interested in a solution that could track the full skeleton of the golfer. Unfortunately the existing wearable sensors performances are too weak to deploy them on an outdoor golfer (Bluetooth bandwidth, update frequency of IMU for really fast movements). They could be usable on an indoor golfer but the cost of such solutions is too high for the moment to provide enough value for golf courses.

Current existing modules

In this final version of the prototype, users liked the different modules of the game but mainly two of them need some improvement:

- Voice Communications: The embedded plugin for voice communications was not efficient in the latest round of experiments. A robust and silent service must run in the application, like it was for the skiing and jogging final prototypes.

- Environment Reconstruction: The only weakness of this module is that it is referring to open free weather data which is not available in every city. Unfortunately the closest weather station was kilometres away from he golf course and the weather in the virtual environment was consequently not accurate.

5.4 Final golf configuration

Indoor
Wide Screen 5.1 Sound system + microphone Club motion capture (1 IMU sensor) Kinect Motion Capture 3DLive Golf indoor App
Outdoor
Android tablet Audio headset Club Motion Capture 3DLive Golf outdoor App

6 Updates based on Co-creation requirements

3D-LIVE has arranged a number of co-creation events, from which valuable feedback from the users has been obtained and has been used to re-shape the 3D-LIVE prototype. The first prototype was made available and tested during LIVE 1 experimentation, which allowed us to evaluate and refine our scenario implementation. The initial design that was based on co-creation activities internal to the consortium, was then refined as a result of the new co-creation activities with end users for all three use-cases. Their feedback provided us with a very positive input in terms of scenario description, technologies and usability of the platform, related to the anticipated experience. The Second prototype of the 3D-LIVE platform gathers most of the designed features for all of the three scenarios.

The purpose of this section is to provide insights on how the 3D-LIVE consortium has taken this feedback into consideration, transformed it into new requirements and the actions that were taken in order to appropriately address them. The table below summarises how 3D-LIVE addressed the new requirements that arise from the co-creation activities.

6.1 Requirements and Coverage Matrix

The 3D-LIVE platform design has been carried in an iterative way as described in the Experiential Design Methodology of the project. Each iteration, from the early workshops with end-users until the third iteration of users trials provided feedbacks and requirements from the users' point of view. We tracked the requirements which needed to be addressed to make the 3D-LIVE platform a High Level Mixed Reality Tele-Immersive platform suitable for several sporting activities.

Below is the summary of the different requirements that helped us shape the design, the setup and the scenarios of use within the 3D-LIVE platform.

Requirements Ids are defined as follows: RX-Y with X referring to the iteration of co-creation activities and experiments (0=co-creation workshops, 1=Feedbacks from LIVE 1 experiments, 2=Feedbacks from LIVE 2 experiments, 3=Feedbacks from LIVE 3 experiments). Please refer to the deliverable "D4.2 Final Report on the experimentation & evaluation of the 3D-LIVE Tele-Immersive Environment" for more information about the feedbacks from experiments. Y is the number identifying the prioritised item, keeping in mind that the lower the value is, the higher the priority was set on this requirement.

ID	Feedbacks	Requirement	addressed for	Action taken	Evaluation
Requirements extracted from Co-Creation workshops					
R0-1	Voice communications very important, like a phone call. A common language must be spoken	Integrate a voice communications system transparent to users	LIVE3	ARTS worked on the improvement of The Voice Chat Unity3D plugin, trying to optimise the compression/decompression and transmission in background using Unity3D. Finally this plugin has been removed for the Golf and Ski scenarios as no performances improvements were reached. The use of the external Voice Client Mumble has been chosen. Easy to setup and very efficient.	Mumble voice client validated for all the scenarios. Everything went well
R0-2	Kinect skeleton quality is poor occluded areas for indoor users	Enhance the quality of the skeleton in occluded areas for indoor users	LIVE1	Developed a skeleton merging algorithm from multiple Kinects. Published a paper in a conference. Integrated the Skeleton Merging algorithm from multiple Kinects in 3DLive Capturer.	Skeleton tracking is improved in occluded cases. Furthermore, the performance of the module was improved due to addressing R1-3
R0-3	Calibration only once, and quick if entertainment purpose	Calibration time must be < 1s and easy to users	LIVE2	CERTH has put in place an automatic calibration process in the Kinect capturer. Once the user is detected in T-Pose, the capturer calibrates the skeleton and add a calibration information to skeleton message queried by the 3Dlive applications.	Calibration of Kinect sensors automatically detected when user is in T-Pose
R0-4	Low latency in voice : most important data for users, communications	Voice quality : low latency < 500ms	LIVE3	Same as R0-1	Mumble voice client validated for all the scenarios. Everything went well

	must be like a phone call				
R0-5	Predefined animations as an outdoor player satisfaying	Integrate predefined animations for the outdoor player (three scenarios)	LIVE1	A set of Motion Capture data has been recorded by CERTH for the three scenarios, including animations for: walk, run, rest pose, golf swing, ski acceleration, ski turns, ski normal pose. A reader was created in order to keep a uniform way of animating the avatars whether they use live skeleton animation or pre-recorded animation data.	The animations are not very good. But they are sufficient to understand what the outdoor avatars are actually doing
R0-6	Graphics should be detailed.The virtual golf course must be realist, and look like the real one. Golfers must be able to recognize each part of the field.	Graphics: 3D environments realist enough to be recognizable	LIVE1	Jogging Scenario: A 3Dmodel of the city centre of Oulu was already available with enough details for users to recognize where they are in the streets. Golf scenario: A basic topology of the Laval golf course (Hole n°1) was available, materials, trees, grass, water and simple 3Dmodels were added to make the Unity3D terrain more representative of the reality. Skiing scenario: Based on a high resolution mesh provided by IT Innovation, ARTS created the Unity3D terrain of the Hopsilift ski slope in Schladming. Simple trees and 3D elements have been created in order to make the model representative of the reality.	The main topology of the course and important buildings/details have been modelled. Enough details were generated

R0-7	The weather must be the same for outdoors and indoors. However in a solo mode, the weather should be configurable	Module: Weather reconstruction system required	LIVE1	A first version of the ERS has been developed by IT Innov. It allows the collection of Open Weather Map data based on GPS locations inputs, on a RabbitMQ server. On the rendering side (both Unity3D and RealXtend) weather effects in the virtual environment have been integrated by Cyber and ARTS.	The ERS is able to propose a full representation of the weather. The Game engine is capable yet of reconstructing virtually the basic weather effects (sun, temp, wind, cloud coverage, rain)
R0-8	No text, prefer pics and graphics	Graphics: feedbacks with graphical elements more than text	LIVE2	ARTS worked with a graphics designer to create the overall 3DLive GUI based on our storyboards. The different GUI elements for the required data to display have been created, and integrated in the applications.	The 3DLive UI was able to display feedbacks with graphical elements rather than text.
R0-9	Graphical content consistent : avatars realistic = all game environment realistic	Module: Human Reconstruction with high quality required	LIVE2		Human Reconstruction was integrated successfully in the app.

R0-10	A race / pursuit is required by the users	Gameplay: Race between the users, gamification to increase sense of immersion	LIVE3	ARTS, IT-Innovation and CERTH worked on the integration of a complete racing gameplay and new Activity Recognition features to get engaged in the race. The flow is the following: The indoor users can practice and race with or without gates while the outdoor is taking the lift. A countdown is triggered; the indoor users have the ability to accelerate by pushing on their virtual ski poles. Once they are hurtling down the slope, the capturer tells the app how much the indoor user is bending. This lower down the drag coefficient of the virtual skier to increase his speed. Consequently indoor users have the opportunity to race down from 70km/h to 110km/h depending on their actions. During the race each user can see the position of other players, the chrono, their speed and the delay compared to the first in race when crossing checkpoints gates. At the end a scores panel pops up to display the times of all the players and the best times of aggregated races. When the outdoor user is ready to race, indoors is automatically teleported at the starting gate, the outdoor user sees the virtual gate through his Heads Up Display and reaches the indoor racers. He can trigger the start of a new race with his wrist controller once he is ready. During the race, he can see the mountain view with pins of the players. At the end he can even	The Ski application provides now a rich gameplay with feedbacks about other users. Users are free to race straight downhill, to have a slalom or a free run. The competition spirit is very clear for both sides of the scenario and the users wanted to race each time. Adding gates for a slalom or a simple race downhill was a really good thing, as he balanced the difficulty of the game indoors depending on the outdoor slope conditions (ridges, quality of snow in the afternoon)

				see the podium with avatars animated live on it.	
R0-11	Information of speed is important	Gameplay: Display speed information	LIVE3	Skiing scenario: ARTS worked much on the physics of the indoor skier. New algorithms have been set up based on the literature to obtain a realistic physics on the virtual slope. Tests have been performed and outdoor/indoor users got similar race times while going full speed on the World Cup slope.	Speed consistent with the activity of the players in Skiing and Jogging Scenario

R0-12	Comparative Stats must be displayed after a game	Gameplay: Comparative Performance feedbacks post experience	LIVE3	Jogging Scenario: Scaling of the speed indoors and outdoors worked very well after few calibration rounds. People matched their running speed with each other fairly easily.	At the end of a game, the users for all scenarios are able to compare their statistics of the race
R0-13	Face realism important on 3D representation of indoor users	Improve Microsoft Kinect sensor placement	LIVE1	One Microsoft Kinect sensor placed horizontally and closer to the user front in order to improve face capturing	The face capturing quality showed a significant improvement when compared to initial trials.
R0-14	Users free to choose position to start from	Gameplay: ability to define starting points in the app	LIVE1	Before starting a game, the position of the starting point and the position of the hole (which changes over time) must be configured. In the settings of the outdoor app we implemented a feature allowing a user to record these positions based on GPS signal. This position will be automatically shared with all connected players in the game, to update their scenario and the position of the appropriate 3Dmodels. For the position of the hole, a dynamic hole has been created based on enabling/disabling mesh colliders + custom occlusion shader on the main virtual course.	The option allows users to define the pin and starting position on the real golf course before starting the game

R0-15	Include rules for "lost balls" (golf)	Gameplay: Implement a consistent scenario in case of lost ball	LIVE3	In the Golfing scenario, indoor golfers can strike the ball out of virtual boundaries. We included a new game phase in order to virtually consider the ball as "lost" and replace it at the proper position based on golfers' requirements. On the other side, in LIVE 2 we learnt that outdoor golfers can just lose their ball in the rough.. There is now a button on the tablet to tell the app a user lost his ball. Consequently the game will consider that users must start again from the previous position.	Lots of new game phases implemented for the golf scenario including the "lost balls" feature, which is used sometimes, so it is was really necessary
R0-16	Include stats 'number of strikes, length and 3D trajectories'	Gameplay: display the 3D trajectories, length and number of strikes	LIVE3	ARTS worked on the Activity Recognition. Golf strikes activity recognition has been reviewed in order be used in every direction outdoors. Tests on the practice course were performed in order to assess and adjust some parameters of our model in an empirical way.	No more problems with carbon shafts. But still some problems of drifting after strong strikes. The EXEL-S1 sensors seem not appropriate for the golf purpose
R0-17	Device: sporting equipment in a fitness club	Input: A sporting equipment from fitness club is required	LIVE1	A first version of the Pro ski simulator controls has been implemented by ARTS thanks to a setup using Arduino board and standard IMU sensor. The data is sent to the port COM via Bluetooth or USB. Then the orientation of the sensor is read to get the range of the turns	The inertial sensor attached to the ski simulator allows us to control the direction of the skier on the slope in a natural way.

R0-18	The outdoor must not wait for others, once he is ready the game must start when he wants	Gameplay: Outdoor triggers the start of the shared experience	LIVE3	ARTS and IT-Innov worked on the integration of the Recon Snow 2 goggles. In the 3DLive Smartphone application, ARTS integrated more game phases to identify better the exact status of the game for indoors and outdoors separately. In addition to the game phases, GPS areas have been defined in order to trigger automatically events to display in the HUD. Based on our specifications, IT Innov developed the content in the HUD including pins animation, avatar representation, and scores management. Communications between phone and Glasses are handled through UDP sockets communications. The wrist controller of the goggles can send instructions to the phone in order to start the race, pair or un-pair from the phone for instance.	The Recon Snow2 goggles are now integrated to provide the outdoor user an immersive experience with his remote friends. He has also the ability to start the race when ready at the top with the wrist controller paired to the goggles
R0-19	simple gaming devices (like WiiBB) not appropriate for sports purpose, but for gaming only	Input: A wii fit board controller must be integrated for gaming purpose	LIVE1	The Wii Fit board has been integrated by ARTS. The distribution of the weight of the user was retrieved by the GlovePie middleware then sent locally through OpenSoundControl (UDP) messages to the ski application. The weight distribution allows the user to control his skier like in a video game.	The weight balance allows users to control the direction of the indoor skier. The control is still a bit hard.

R0-20	A HMD must be considered for indoor users (lower price with interesting immersion)	Rendering: different visualisation devices are required. Users need lower prices with highest sense of immersion	LIVE3	The 3Dlive platform has been designed to be able to connect various external VR devices. ARTS used MiddleVR middleware to configure the indoor ski application to Oculus Rift and Clarté SAS3+ CAVE. A specific 3D User Interface handler has been designed in order to manage graphical UIs with 3D models instead of overlay that cannot be rendered in those immersive devices.	HMDs and CAVEs are now part of the 3D-LIVE system, which can be deployed on different devices. They provide a great User Experience and satisfaction. The sense of immersion was assessed to be even higher in the HMDs with the small user group which tested both solutions.
R0-21	The outdoor mask must not be dangerous for the activity. AR Glasses very important to rationalise the point of wearing sensors	Rendering: Outdoor visualisation required, but AR not appriopriate	LIVE3	See R0-16	The Recon Snow2 goggles are now integrated to provide the outdoor user an immersive experience with his remote friends. He has also the ability to start the race when ready at the top with the wrist controller paired to the goggles
R0-22	The outdoor mask must be protected from moisture, falls and weather conditions	Safety: mask but be protected	LIVE3		The Recon Snow2 goggles is a product on the market, including moisture protection, comfort and safety regulations
R0-23	The indoor user must feel the sensation of steepness and speed	Rendering: The immersion must be high enough to let the users feel the steepness	LIVE3	ARTS performed a work on shaders, camera frustum distortion and visual effects but the result was not satisfying with simple 2D screens. The integration of immersive rendering devices (CAVE and HMD), created the needed feeling of speed and steepness of the ski slope.	On simple displays, the feeling of speed is not great. Even visual effects did not improve much the feeling of speed for the users. But the addition of immersive rendering devices like HMDs and CAVEs and sound feedbacks increased much the feeling of speed and immersion on the actual slope. When the head of a user is looking down, he can now feel the steepness.

R0-24	Predefined routes are necessary	Gameplay: Predefined routes for the jogging	LIVE1		The indoor jogger automatically follows a predefined route
R0-25	Include a way to socialize after a game	Gameplay: Comparative Performance feedbacks post experience	LIVE3	Ski scenario: We included race performance handler, in order to save the different race positions and times of races done during a game. Indoor and outdoor users can now see the race times of other racers. The outdoor user can even see a podium with animated avatars on it, thanks to the communications between the Smartphone and the Recon Snow2 goggles Golf scenario: All shots, distances and performances are saved during a game, Once the ball is in the hole, the two players have access to a summary of their performance: who did the best/worst strike, how many times their ball was selected..etc.	All the users, for the three scenarios have now access to the performance feedbacks of other players after a game. Users were satisfied by the information displayed.
R0-26	Sensory data is important (heart rate)	Input: Heart rate and other sensory data required for the jogging activity	LIVE3	Bluetooth Low Energy HR protocol was implemented to the outdoor application and tested using Viiiiva HRM accessory successfully. IMU sensors provided by Sports Curve were implemented partially to outdoor jogger application but late arrival of the sensors just before LIVE3 testing did not allow working implementation of the jogger avatar animation.	

R0-27	Runners should be close to one another even if they run at different speeds	Gameplay: The indoor player speed must be adjusted	LIVE3	No action taken. Adjustment of the indoor jogger speed outside of the treadmill would have undermined the purpose of Kinect speed estimation.	
R0-28	Ability to replay shots	Gameplay: ability to replay shots	LIVE3	No action taken, as replay in Golfing scenario was important only for a training purpose	Not integrated. But no user finally asked for that during the live tests
R0-29	Users should not be able to talk to each other during strikes	Gameplay: disable voice comms during shots	LIVE2	We added a feature in order to force mute the different players during strikes while a user requires some focus and preparation during his strike.	Outdoors were muted when indoors were taking strikes. Indoors were muted while outdoors were taking strikes
R0-30	Outdoor can ski down without indoors	Gameplay: outdoor user able to run without indoors	LIVE3	A mode solo is available for both indoors or outdoors if they want to run	Tested internally. Not experimented with users because no interest in our research purpose.
R0-31	A replay must be available for the players. Ghost runner from previous run through AR glasses	Gameplay: A replay must be available for users	LIVE3	During the ski race, the locations of all the players are stored in order to be replayed at the end. ARTS included this replay only on a 2D map considering the effort required for a 3D replay that would not add great value for this game (save positions + animations of avatars and replay it.)	A replay for the skiing scenario has been integrated. Finally users don't care and don't even look at it. They want to go at the top faster to take a revenge on the next race.
R0-32	Zero configuration on startup	Gameplay: Ability to start and go without changing settings	LIVE3		The settings were well configured in the LIVE3, decreasing considerably the waiting time for users to start a game

R0-33	Motion sensors only for training purpose, else too much disturbing.	Human Motion sensors accuracy very high for training purpose, Accuracy not needed for gaming purpose	LIVE3	ARTS integrated 4 IMU sensors (EXEL S3) in the 3D-LIVE platform. We created a new motion handler in order to perform avatar animation on a part of the body. One sensor was attached to the Torso, one to the head, one to the upper arm and the last one on the arm. A new model of animation has been created in order to fuse predefined animations with live IMU animations, in the local references of the Torso. It was working well, but the process of attaching the sensor needs much accuracy, sensors can move. So for sure they would have moved during a race. An existing MoCap suit would be more convenient here.	Investigation has been carried on. 4 sensors on the body allowing local bones animations with the torso as a reference, has been tested. The problem is attaching the sensors on the body requires much accuracy to get consistent animation during the whole race. To do so, suit with sensors already attached (xSense/Heddokko) would be necessary.
R0-34	Voice Comms optional	Voice: The component must be easily enabled/disabled by the user	LIVE2		The setting to enable/disable voice is straightforward to use
R0-35	Option to be able to see others (birds eye view, map with markers)	Gameplay: Birds eye view required	LIVE3	A key on a keyboard can simply switch between views.	Key can switch the view into first/third person view or bird's eye view
Requirements extracted from LIVE 1					

R1-1	Poor Outdoor UX Skiing LIVE 1	Create the user engagement of outdoor users in the game	LIVE3	IT Innov and ARTS reshaped the skiing scenario with the support of Schladming 2030. IT Innovation provided a new high accuracy 3D model of the slope, and several 3D models of the buildings at the bottom of the Planai station. ARTS integrated this model in the rendering engine and added the required content to feel on the real slope: trees around, missing buildings, lift. In addition to that a shader displaying procedural snow on this new terrain was used. See responses for the R0-9 and R0-19 for more information on the gameplay and Heads Up Display	The race mode is happening on the World Cup black slope of Schladming. The new 3D model was created, the physics to control skier has been totally reviewed for this purpose. Gamification and sense of competition is important now. The HUD allows even outdoor users to "give it all to win the race".
R1-2	The movements of the users where mapped in reverse to avatars' movements. E.g. User's left hand movements were mapped to avatar's right hand movements.	Map the user's movement to avatar's movement appropriately for left and right hand/feet	LIVE2	Skeleton mirroring support was added in 3DLive Capturer	Avatars' movements correctly resemble the movements of the users
R1-3	It is difficult for the users to improve their Ski activity score.	The ski activity evaluation algorithm should map scores in a more natural way	LIVE2	The ski activity recognition & evaluation algorithm was reviewed and updated to better map users' activities in to the score interval [0.0-1.0]	Users find it easier to adapt their movements to the evaluation mechanism and improve their scores

R1-4	Poor voice driven voice communications in Golf and Ski scenarios	Create the user verbal engagement with one another	LIVE3	See R0-1 action taken	Mumble voice client validated for all the scenarios. Everything went well
R1-5	Unexpected avatar postures/actions	Guarantee correct avatar animations of remote players	LIVE3	After reviewing again the skeleton acquisition and skeleton merging algorithm, it was observed that at certain circumstances OpenNI would falsely detect non-existent users in the background of the capture site and the skeleton merging algorithm would try to fuse all the skeletons together. Solution was provided by adding filtering to remove non-existent, miss-detected user skeletons from OpenNI.	No more complain about weird avatar animation
R1-6	Incorrect response to physical efforts in Jogging LIVE 1	Make consistent the physical action of runners with virtual avatar motion	LIVE3	See R2-2, R2-10	For outdoor avatar motion tracking only GPS location was used therefore it was more of an approximation of the real current location, but combining this data with predefined idle, walking and running animations with speed interpolation, indoor joggers perceived the outdoor jogger avatar to be moving really smoothly and realistically.

R1-7	Not enough information of the game status	Integrate all the required information for the users	LIVE3		Users are now able to follow their activity during the game. In golf, they can have all required information about their shots and performances. In Jogging the speed and the overall performance of the activity is made available, with advices to users. In Ski, the speed, race chrono and delay compared to the faster are made available for instance, keeping our users focused on the game objectives
R1-8	Software crashes sometimes, but the game status is lost if tentative reconnection	Integrate a game content saver in case of network failure/disconnection/crash of the app in order to reconnect to the game and keep the overall progression	LIVE2 and LIVE3	The work has been pursued on the server and client side in order to make the re-connection more robust and retrieve the overall game status of all the connected/disconnected players.	Validated in LIVE 2 and made more robust for LIVE 3. Rare disconnections and re-connections went well at different locations in the game. Users did not complain about it.
R1-9	Bluetooth connectivity problem while using IMU sensors and bluetooth headset at the same time	Allow a headset to work in parallel with bluetooth IMU sensors	LIVE2	The wireless headphones have been removed from the deployment setup. A wired headphone has been selected in order to let the IMU sensors work without disconnecting every two minutes.	The Wired headset and the IMU sensors worked concurrently. Users were not disturbed by using a wired headphones
Requirements extracted from LIVE 2					

R2-1	Integrate a HUD for outdoor skiers	Integrate a HUD for outdoor skiers	LIVE3	See R0-19	The Recon Ski goggles (Snow2) was integrated, allowing users to see avatars, race positions, performances of players and even the virtual mountain. This UX was really satisfactory for outdoor users
R2-2	Improve skeleton capturing quality and motion mapping	Improve skeleton capturing quality and motion mapping	1	Replaced OpenNI with Microsoft Kinect SDK in 3DLive Capturer. Additionally. developed skeleton filters. This allows smoother animations and more natural postures, even when the user is not properly facing the Kinect (hands in the back for instance). This also allows us to include robust Head tracking to improve the immersion while using HMDs.	The animation quality of the avatars was substantially improved. Users reported accurate animation of the avatars according to their movements
R2-3	The jogging activity evaluation algorithm should map scores in a more natural way. Speed recognition should be improved.	The jogging activity evaluation algorithm should map scores in a more natural way. Speed recognition should be improved.	1	Reviewed and updated the Jogging evaluation algorithm. Additionally, added support in 3DLive Capturer for Kinect v2.0 for improved body tracking and thus superior activity recognition results.	The users found the updated evaluation algorithm robust. They reported ease of adaptation of their movements in order to improve their score. Jogging speed estimation also dramatically improved.
R2-4	Race on a challenging slope like World Cup slopes	Race on a challenging slope like World Cup slopes	LIVE3	See R1-1	The new slope has been integrated. A raw model from the GISS in Austria was provided and transformed into a Unity3D terrain. Trees, buildings and necessary details in the environment were modelled to provide enough realism for the Schladming users

R2-5	EOS service silent or not used	EOS application considered as optional service.	LIVE3	For a low-end use the wide area weather data is enough and including EOS set-up added additional complexity/set-up time for end users	EOS not used in LIVE 3 Ski
R2-6	Avoid unexpected app exit	Avoid unexpected app exit	LIVE3	The Unity3D java activity has been updated in order to force native functions not to happen when users touch the return button on Android. Moreover a case has been attached to the Smartphone to solve the problem of "pressing the Home button"	No way to improve this part of the Golfing scenario in Laval, there is not sufficient data resolution (with specific reference to low-level atmosphere) covered by Open Weather Map data.
R2-7	ERS to match the local weather of the golf course	ERS to match the local weather of the golf course	X	Unfortunately, the Open Weather Map closest station from the golf course in not in Laval, so the weather interpolation can sometimes be wrong.	No way to improve this part of the Golfing scenario in Laval, not covered by Open Weather Map data.
R2-8	Avoid inconsistencies and misdetections of the shots outdoors	Avoid inconsistencies and misdetections of the shots outdoors	LIVE3	Same as R0-14	Carbon shafts were used. Detections worked better at the beginning of the day. But after a few strikes some problems appeared : sensors drifting / sensors not providing accurate data. EXEL S1 not appropriate for this purpose
R2-9	The Overall setup must be easy and quick	The Overall setup must be easy and quick	LIVE3		The deployment configuration has been decided and pre-configured in the apps. A simple start and GO allowed users to join the game without waiting time

R2-10	Voice latency must be very low, and not disturb the framerate of the apps	Voice latency must be very low, and not disturb the framerate of the apps	LIVE3	See R0-1	Mumble was finally used, as we did not manage to implement background tasks Unity3D to handle voice compression and transmission on multiple platforms. Mumble was a good option, it is stable, efficient and do not disturb the game at all. We recovered a high framerate in our apps.
R2-11	Make a clear 2D map of the golf course for the golfers to understand where their balls are on the course.	Make a clear 2D map of the golf course for the golfers to understand where their balls are on the course.	LIVE3	The map view was updated by ARTS, and rotated 90°	The users did not complain anymore and understood easily where their balls were on the course, by reading this 2D map.
R2-12	Harmonise the 3D reconstruction and activity recognition/speed estimation module.	Harmonise the 3D reconstruction and activity recognition/speed estimation module.	LIVE3	The capture frame-rates between the trained model of the activity recognition module and the actual subject in-game were inconsistent when 3d reconstruction was enabled. Make them consistent. When 3d reconstruction was enabled, capturing frame rates would drop to nearly ~12 fps instead of ~30fps. Moving the reconstruction process in a separate thread enabled the skeleton capturing to be performed in 30fps and be consistent with the activity recognition trained model.	The speed estimation worked flawlessly in Jogging LIVE 3

R2-13	Minimize delays of the 3d reconstructed user's transmission to a level that allows real-time interaction	Minimize delays of the 3d reconstructed user's transmission to a level that allows real-time interaction	LIVE3	In this typical issue of a producer-consumer problem where the producer produces packets at higher rates than the consumer can consume, there was a need to update the 3D Reconstruction Network receiver to use a RabbitMQ queue with network characteristics suitable for real-time tele-immersion. Thus, 3D Reconstruction Network receiver was updated to use a RabbitMQ queue that drops all received frames but the last one.	No more delays were reported from users during LIVE 3 experiments

7 Discussion - Conclusions

This document describes the final 3D-LIVE prototype. Recruited external users engaged in each use-case of the 3D-LIVE project experimented the platform and reported new feedbacks to validate the scenario elements and provide areas for improvement whether a product could be developed.

The final user requirements were reported and mapped with the implemented features for evaluation. Most of them were covered, nevertheless some users identified new requirements once the full prototype was implemented, in terms of scenario element or functionality.

For instance, in the golfing scenario, some features were finally considered useless by the golfers. For a gaming purpose they would rather have a fully connected platform that does not disturb the waiting time between strokes, so an asynchronous game between players without huge amount of game data transmitted real time, just information on phases and game status. Probably the technology is not mature enough to satisfy the users or mixed reality does not make sense here. One thing is certain, the potential of such a product proposing to golfers a game between golfers outdoors and golfers indoors is very important.

In the case of the skiing scenario, the scenario was really challenging and satisfying for the users, it generated interest from skiers and gamers. Feedbacks were positive and the improvements could be made with graphics improvement and adding some features for the pros.

In the case of the Jogging scenario, the scenario was very well accepted and people liked the collaborative jogging experience. Generally feedback was very positive and major complaints were on the animation of the avatars which did show fair amount of noise and awkward poses occasionally. With this improvement and inclusion of proper race mode for pro runners could spice up the scenario even more.

The technical achievements of the final prototype conducted us towards a strength platform allowing three different sporting scenarios. The 3D-LIVE consortium

- Validated a version for each designed module of the platform and reported inputs for the improvement of these modules in case of the development of a product. (Rendering and Visualization, Activity Recognition, Environment Reconstruction, Human Reconstruction and Animation, Voice Communications, Data Compression and Transmission).

- Validated the interconnectivity of the different modules
- Managed to integrate all these components into a very modular platform. A game can be run with or without the modules, including different kinds of input devices and output rendering devices by a simple setting change.

The technical lessons we learnt from the experiments of the platform helped us shape a list of recommendations for the development of such a platform, some are listed below:

- The different applications and services are suitable for real-time tele-immersive experience.
- The hardware running the applications must be powerful enough to process real time network communications, computer graphics and acquisition of data from multiple sources in parallel.
- The higher the network performances are, the stronger the 3D-LIVE experience is (in terms of bandwidth and quality of communication channels). Bottlenecks can generate inconsistencies, with negative impacts on the overall experience.
- The more accurate the sensors embedded in the mobile devices are (GPS, Gyroscope), the stronger the 3D-LIVE experience is.

In terms of User eXperience, the lessons learnt are numerous and, in case, vary among different use cases. An overview of conclusions that will drive implementation activities for the remaining of the project is presented below:

- Skiing Scenario: The improvements of the skiing scenario in the LIVE 3 completely transformed the UX of both outdoor and indoor skiers. The integration of a new World Cup slope and the Recon Snow2 goggles connected to the game (rendering the virtual mountain, the avatars animated live, the scores of the races, the virtual starting and finish gates) just brought the outdoor skier into the game. This enhanced the immersion of the outdoor player and the way he was challenged by other players.

And finally the same applies for the other players, the fact the outdoor skier felt better immersed in the game just bring them closer to the real slope. In terms of immersion of the indoor skier, the integration of the Oculus Rift and the CAVE provided a totally better sense of belonging to the virtual environment.

- Golfing scenario: The improvements required from LIVE 2 were all covered. People realised that the status of the game and the activity of their peers was clear. Unfortunately now that the features are integrated, they thought about a different way to deploy such a platform. It seems that the concept of augmented reality on the field is not really interesting for real golfers. The UX of this platform must focus on the freedom of each user to play when they want. In the current scenario the UX was decreased by the fact users were waiting for the other to strike his ball – which is the concept of the scramble – but which generates a delay in the overall process of the game. Golfers would love a social network on which to play games with remote friends indoors, and to communicate vocally. But the flow must remain smooth, and not slow down a game.
- Jogging Scenario: LIVE3 for jogging was all about refinement and polishing the final jogging application for both indoors and outdoors. Scenario was changed to be three phases where users run half of the course, stop at the midpoint and finally finish the course together. This pacing brought more interaction between people and it was generally well accepted. Indoor joggers thought outdoor avatar representation really belonged to the game but on the flip side of the count outdoor jogger felt his experience was not much different from when using a tracking applications popular to mobile market. Unfortunately IMU sensors for outdoor jogger motion tracking did not get produced in time for LIVE3 test, but in the end it would only improve indoor jogger experience without affecting the outdoor jogger lack of immersion. Outdoors definitely require augmented reality device to really feel part of the scenario as the indoor joggers do. Finally proper graphical representation for jogging performance was finalized and implemented for the jogging LIVE3 which caused positive reactions among joggers.

All scenarios: The main improvement common to all scenarios was the choice of only one voice service (Mumble) for the users. It reduced the delays and the quality and eased the communications amongst players. Despite our efforts in terms of robustness, one of the issues that need much care in the design of this kind of platform is the network conditions available on site. Network failures caused stops in the overall gameplay, due to the need of a continuous real time transmission of data, decreasing the quality of the UX sometimes.

The platform has been significantly improved for the LIVE 3 round of experiments and all the main required features have been implemented since then. The user-centred design process including end-users and stakeholders we adopted allowed us to understand what really matters for them. Their main feedbacks were collected for each iteration of co-creation activity, workshops and experiments in the three deployed scenarios. Working concurrently on three scenarios permitted to maximize the number of new requirements to address for the overall development of the platform. All these requirements, the solutions proposed by the consortium addressing them and the assessment of the actual implementations have been concatenated in one traceability matrix. This methodology lead us to take into account almost all the actual needs reported by users to finalize the 3D-LIVE prototype. The platform now focuses on the easiness of use and the sense of tele-immersion for the different setups of the players.

8 Appendix

8.1 Appendix A: Jogging use-case architecture connections

Link	From		To		Data	Description
	Type	Designation	Type	Designation		
c1	H	Microphone	S	3D-LIVE indoor app	Voice	Wireless Bluetooth connexion of the headset. Voice is transmitted to the PC, through the default input. The app then reads the incoming audio flow.
c2	H	Kinect x n	S	Kinect Capturer app	Depth map + RGB map images	Wired connexion / USB ports. The Kinect Capturer retrieves depth and RGB maps, in order to extract skeleton data/3D Full reconstructed time varying meshes of humans
c3	S	3D-LIVE indoor app	H	Wide Display / CAVE like display	Rendered image	The 3D-Live app render the image on the wide display / CAVE like display made of three wide wall screens.
c4	S	Mumble app	H	Audio Headset	audio	The Mumble desktop application send the incoming voice to the wireless audio headset.
c5	S	3D-LIVE indoor app	S	RabbitMQ server	- ERS Requests - ExperiMonitor metrics data	The 3D-LIVE indoor app connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction service) and the ExperiMonitor. To the ERS, it will send requests for the weather information. To the ExperiMonitor it will send the measurements it takes during experiments. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.

c6	S	RabbitMQ Server	S	3D-LIVE indoor app	- ERS Results - ExperiMonitor Instructions - Skeleton information - Full 3D Human reconstruction	Through the same connexion than c5, the app will receive results from the ERS about the actual weather on site, and some instructions from the ExperiMonitor relative to the experiment process. More than that, the app subscribes to RabbitMQ queues containing the skeleton animation of remote users and 3D mesh of their full reconstruction.
c7	S	3D-LIVE indoor app	S	realXtend Server	- Position of the user - Orientation of the user - Game phase triggers - Game Events	The 3D-LIVE indoor app knows everything about the behaviour of the user in the 3D world. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the kNet network open source library.
c8	S	realXtend Server	S	3D-LIVE indoor app	- Position of the remote users - Orientation of the remote users - Game phases triggers by remote events - Game remote events	As in the c7 link, The Main Screen app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API.
c9	S	Mumble app	S	Mumble Voice server	Voice data	The Mumble app communicates with the Mumble server to transmit the voice of the user.
c10	S	Mumble Voice server	S	Mumble app	Voice data	The Mumble server sends to the Mumble app the incoming voice of remote users.

c11	S	Kinect Capturer app	S	RabbitMQ server	- Skeleton information - Measurements about skeleton information - Activity Recognition labels - Full 3D reconstructed time varying mesh	The Kinect capturer connects to the RAbbitMQ server and updates a queue containing live skeleton data of the user as well as Activity Recognition labels, evaluating the posture of the skier. It also fully reconstruct the user in a 3D time varying mesh and transmit it to the server. Moreover as it measures metrics evolution over the time, it sends it to the appropriate exchange created for the ExperiMonitor. It uses the RabbitMQ protocol.
c12	S	RabbitMQ Server	S	Kinect Capturer app	- ExperiMonitor Instructions	On the same connection than the c9 link, the ExperiMonitor will send instructions regarding the experiment process to the Kinect capturer.
c13	S	realXtend Server	S	Web Service	GPS requests	The 3realXtend server requests a webservice to know at any moment the position of the outdoor user, in order to synchronize with the indoors.
c14	S	Web Service	S	realXtend Server	GPS data	The Webservice push the GPS data of one user to the realXtend server in order to render the outdoor avatar in the indoor clients' 3D scenes.
c15	H	Embedded GPS	S	3D-LIVE Smartphone app	- GPS data from the Smartphone	The app connects to the Android API to retrieve the data coming from the embedded hardware: GPS for positioning the user in the virtual world.
c16	H	Sensordrone	S	EOS app	- Temperature - Capcitance - Illuminance - Humidity - Pressure	Bluetooth connectivity between the sensor "Sensordrone" and the Smartphone. Local weather information is sent to the smartphone in order to then feed the ERS (Environment Reconstruction Service). The sensor must be paired to the Android OS.
c17	H	Microphone (headset)	S	Mumble app	Voice	Wired connexion / Jack 3.5 stereo input. The voice is acquired by the microphone is retrieved by the Mumble app

c18	H	Mumble app	H	Headset	Voice	Voice data sent by the Mumble voice service in the headset worn by the outdoor user.
c19	S	Mumble app	S	Mumble Voice server	Voice data	The Mumble app communicates with the Mumble server to transmit the voice of the user.
c20	S	Mumble Voice server	S	Mumble app	Voice data	The Mumble server sends to the Mumble app the incoming voice of remote users.
c21	S	EOS App	S	RabbitMQ server	- Temperature - Capcitance - Illuminance - Humidity - Pressure	The Android EOS app connects to the RabbitMQ server, to reach the ERS (Environment Reconstruction Service) in order to send to it the local weather data captured by the Sensordrone. It uses the RabbitMQ protocol.
c22	S	3D-LIVE Smartphone app	H	Smarpthone display	Rendered image	The image / GUI + camera flow is rendered on the display of the Smartphone
c23	S	3D-LIVE Smartphone app	S	Web Service	GPS data	The Smartphone App pushes GPS data from the Smartphone to the webservice, to allow indoor users to know the smartphone position at any given time.
c24	S	Environment Reconstruction Service	S	OpenWeatherMap	Weather requests	The ERS(Environment Reconstruction Service) uses the OpenWeatherMap open API to retrieve weather information at the GPS coordinates of the venue.
c25	S	OpenWeatherMap	S	Environment Reconstruction Service	Weather Results (Temperature, Humidity, Cloud coverage, Rain fall rate, Snow fall rate,)	The Weather effects are sent to the Environment Reconstruction Service (which will interpolate the weather data coming from different sources like Sensordrone to generate a weather information at any GPS location of the venue)

c26	S	RabbitMQ Server	S	Environment Reconstruction Service	- ERS Requests - Experimonitor instructions	The ERS(Environment Reconstruction Service) connects to the RabbitMQ server in order to listen to weather requests at any provided GPS location that belongs to the venue. Requests will be sent to it through the RabbitMQ protocol. Moreover the ExperiMonitor will send instructions to the ERS about the experiment process, through the RabbitMQ protocol as well.
c27	S	Environment Reconstruction Service	S	RabbitMQ server	- ERS results - Experimonitor measurements	Using the same connection than c31, the ERS (Environment Reconstruction Service) send weather results to RabbitMQ clients requesting weather information. Moreover the ERS send measurements taken like the rate and the nature of the requests.
c28	S	ExperiMonitor	S	RabbitMQ server	- Experiment instructions	The ExperiMonitor is the service that handles the experiment monitoring process and live visualization of metrics. It connects to the RabbitMQ server to allow remote applications to send their metrics using a RabbitMQ protocol.
c29	S	RabbitMQ Server	S	ExperiMonitor	- Experiment metrics	Using the same connection than c33, the ExperiMonitor will get from the different clients experiment metrics, store them and visualize them in real time.

8.2 Appendix B: Golf use-case architecture connections

Link	From		To		Data	Description
	Type	Designation	Type	Designation		
c1	H	Club + EXEL-S1 IMU sensor	S	Main Screen app	Acceleration, Orientation (quaternion)	Bluetooth 2.1 connectivity to transmit Acceleration, Quaternion data to the game application. The sensor must be first paired to the Windows operating system on which the app is running
c2	H	Microphone	S	Main Screen app	Voice	Wired connexion / Jack 3.5 stereo input. The voice is acquired by the microphone is retrieved by the app
c3	H	Kinect x n	S	Kinect Capturer app	Depth map + RGB map images	Wired connexion / USB ports. The Kinect Capturer retrieve depth and rgb map, in order to extract skeleton data
c4	S	Main Screen app	H	Wide Display	Rendered image	The main screen app renders the image on the wide display.
c5	S	Main Screen app	H	5.1 sound system	3D sound	The main screen app process the incoming voice and virtual sounds as 3D sounds, then render it in the 5.1 sound system, wired by 5 jack mono outputs.
c6	S	Monitor Screen app	H	Monitor touch screen	Rendered image	The Monitor app render on the touch screen the "monitor" golf application
c7	H	Monitor touch screen	S	Monitor Screen app	Touch interactions	the monitor touch screen retrieves the touch information from the user to interact with the app GUI
c8	S	Main Screen app	S	RabbitMQ server	- ERS Requests - ExperiMonitor metrics data	The main screen application connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction service) and the ExperiMonitor. To the ERS, it will send requests for the weather information. To the ExperiMonitor it will send the measurements it takes during experiments. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.

c9	S	RabbitMQ Server	S	Main Screen app	- ERS Results - ExperiMonitor Instructions - Skeleton information	Through the same connexion than c8, the app will receive results from the ERS about the actual weather on site, and some instructions from the ExperiMonitor relative to the experiment process. More than that, the app subscribes to RabbitMQ queues containing the skeleton animation of remote users.
c10	S	Main Screen app	S	Unity3D Server	- Position of the user - Orientation of the user - Position of the ball - Game phase triggers - Game Events - Swing animation - Voice of user	The main screen app knows everything about the behaviour of the user in the 3D world. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the standard Unity3D network API based on RPC (remote procedure calls).
c11	S	Unity3D Server	S	Main Screen app	- Position of the remote users - Orientation of the remote users - Position of the remote balls - Game phases triggers by remote events - Game remote events - Swing animation - Voice of remote users	As in the c10 link, The Main Screen app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API.
c12	S	Monitor Screen app	S	RabbitMQ server	- ERS Requests	The Monitor screen application connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction service) to display weather information on screen. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.

c13	S	RabbitMQ Server	S	Monitor Screen app	- ERS Results	Through the same connexion than i5, the app will receive results from the ERS about the actual weather on site to display it on GUI.
c14	S	Monitor Screen app	S	Unity3D Server	- Position of the user - Position of the ball - Game phase triggers - Game Events	The monitor screen app knows the same data than the main app but the orientation of users. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the standard Unity3D network API based on RPC (remote procedure calls).
c15	S	Unity3D Server	S	Monitor Screen app	- Position of the remote users- Position of the remote balls- Game phases triggers by remote events- Game remote events	As in the i7 link, The Main Screen app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API, in order to display the information (maps, dynamic icons, text..) to the user.
c16	S	Kinect Capturer app	S	RabbitMQ server	- Skeleton information - Measurements about skeleton information	The Kinect capturer connects to the RAbbitMQ server and updates a queue containing live skeleton data of the user. Moreover as it measures metrics evolution over the time, it sends it to the appropriate exchange created for the ExperiMonitor. It uses the RabbitMQ protocol.
c17	S	RabbitMQ Server	S	Kinect Capturer app	- ExperiMonitor Instructions	On the same connection than the i9 link, the ExperiMonitor will send instructions regarding the experiment process to the Kinect capturer.
c18	H	Embedded GPs, Gyroscope, Camera	S	Android Tablet app	- GPS data from the tablet - Orientation of the tablet (Quaternion) - Image buffer of the tablet camera	The app connects to the Android API to retrieve the data coming from the embedded hardware: GPS for positioning the tablet in the virtual world, Orientation (quaternion) to know at any given time where is looking the tablet in the virtual world, and camera to add the reality layer in the app.

c19	H	Club + EXEL-S1 IMU sensor	S	Android Tablet app	Acceleration, Orientation (quaternion)	Bluetooth 2.1 connectivity to transmit Acceleration, Quaternion data to the game application. The sensor must be first paired to the Android operating system on which the app is running.
c20	H	Microphone (headset)	S	Android Tablet app	Voice	Wired connexion / Jack 3.5 stereo input. The voice is acquired by the microphone is retrievd by the app
c21	H	Sensordrone	S	EOS app	- Temperature - Capcitance - Illuminance - Humidity - Pressure	Bluetooth connectivity between the sensor "Sensordrone" and the Smartphone. Local weather information is sent to the smartphone in order to then feed the ERS (Environment Reconstruction Service). The sensor must be paired to the Android OS.
c22	S	Android tablet App	H	Android Tablet Display	Rendered image	The image / GUI + camera flow is rendered on the display of the tablet
c23	S	Android tablet App	H	Headset	Stereo audio	The Android tablet App processes the incoming voice and virtual sounds as 3D sounds, then render it in the stereo system, wired by a jack stereo output.
c24	S	EOS App	S	RabbitMQ server	- Temperature - Capcitance - Illuminance - Humidity - Pressure	The Android EOS app connects to the RabbitMQ server, to reach the ERS (Environment Reconstruction Service) in order to send to it the local weather data captured by the Sensordrone. It uses the RabbitMQ protocol.
c25	S	Android tablet App	S	RabbitMQ server	- ERS Requests - ExperiMonitor metrics data	The Android tablet application connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction service) and the ExperiMonitor. To the ERS, it will send requests for the weather information. To the ExperiMonitor it will send the measurements it takes during experiments. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.

c26	S	RabbitMQ server	S	Android tablet App	- ERS Results - ExperiMonitor Instructions - Skeleton information	Through the same connexion than c25, the app will receive results from the ERS about the actual weather on site, and some instructions from the ExperiMonitor relative to the experiment process. More than that, the app subscribes to RabbitMQ queues containing the skeleton animation of remote users displayed in Augmented Reality.
c27	S	Android tablet App	S	Unity3D Server	- Position of the user - Orientation of the user - Position of the ball - Game phase triggers - Game Events - Swing animation	The Android tablet app knows everything about the behaviour of the user in the 3D world. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the standard Unity3D network API based on RPC (remote procedure calls).
c28	S	Unity3D Server	S	Android tablet App	- Position of the remote users- Orientation of the remote users- Position of the remote balls- Game phases triggers by remote events- Game remote events- Swing animation	As in the c27 link, The Android tablet app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API.
c29	S	Environment Reconstruction Service	S	OpenWeatherMap	Weather requests	The ERS(Environment Reconstruction Service) uses the OpenWeatherMap open API to retrieve weather information at the GPS coordinates of the venue.
c30	S	OpenWeatherMap	S	Environment Reconstruction Service	Weather Results (Temperature, Humidity, Cloud coverage, Rain fall rate, Snow fall rate,)	The Weather effects are sent to the Environment Reconstruction Service (which will interpolate the weather data coming from different sources like Sensordrone to generate a weather information at any GPS location of the venue)

c31	S	RabbitMQ Server	S	Environment Reconstruction Service	- ERS Requests - Experimonitor instructions	The ERS(Environment Reconstruction Service) connects to the RabbitMQ server in order to listen to weather requests at any provided GPS location that belongs to the venue. Requests will be sent to it through the RabbitMQ protocol. Moreover the ExperiMonitor will send instructions to the ERS about the experiment process, through the RabbitMQ protocol as well.
c32	S	Environment Reconstruction Service	S	RabbitMQ server	- ERS results - Experimonitor measurements	Using the same connection than c31, the ERS (Environment Reconstruction Service) send weather results to RabbitMQ clients requesting weather information. Moreover the ERS send measurements taken like the rate and the nature of the requests.
c33	S	ExperiMonitor	S	RabbitMQ server	- Experiment instructions	The ExperiMonitor is the service that handles the experiment monitoring process and live visualization of metrics. It connects to the RabbitMQ server to allow remote applications to send their metrics using a RabbitMQ protocol.
c34	S	RabbitMQ Server	S	ExperiMonitor	- Experiment metrics	Using the same connection than c33, the ExperiMonitor will get from the different clients experiment metrics, store them and visualize them in real time.

8.3 Appendix C: Ski use-case architecture connections

Link	From		To		Data	Description
	Type	Designation	Type	Designation		
c1	H	Pro Ski Simulator + IMU / WiiBB	S	3D-LIVE indoor app	Orientation (quaternion) / Weight distribution	Bluetooth connectivity to transmit Quaternion data to the game application (thanks to EXCEL S3 IMU sensor), corresponding to the movements of the Pro Ski Simulator platform displacements. The data is then available on a COM port of the PC. Bluetooth connectivity between the WiiBB and the PC, transmission of the weight distribution of the user on the Wii balance board through the GlovePie middleware. Those data are used for low-end/high-end setups to control avatar turns on the slope.
c2	H	Microphone	S	Mumble indoor app	Voice	Wireless Bluetooth connection of the headset. Voice is transmitted to the PC, through the default input. The app then processes the incoming audio flow.
c3	H	Kinect x n	S	Kinect Capturer app	Depth map + RGB map images	Wired connexion / USB ports. The Kinect connection retrieve depth and rgb map, in order to extract skeleton data
c4	S	3D-LIVE indoor app	H	Wide Display / Oculus / CAVE	Rendered image	The 3D-Live app renders the image on the wide display or immersive devices. In the case of Oculus Rift or CAVE, the MiddleVR middleware allows the adaptive tracking of the user and rendering the proper way.
c5	S	Mumble Indoor app	H	Audio Headset	Stereo audio	The Mumble app renders in the audi headset the decoded incoming voice data.
c6	S	3D-LIVE indoor app	S	RabbitMQ server	- ERS Requests - ExperiMonitor metrics data	The 3D-LIVE indoor app connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction service) and the ExperiMonitor. To the ERS, it will send requests for the weather information. To the ExperiMonitor it will send the measurements it takes during experiments. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.
c7	S	RabbitMQ Server	S	3D-LIVE indoor app	- ERS Results - ExperiMonitor Instructions - Skeleton information - Full body Reconstruction data	Through the same connection than c6, the app will receive results from the ERS about the actual weather on site, and some instructions from the ExperiMonitor relative to the experiment process. More than that, the app subscribes to RabbitMQ queues containing

						the skeleton animation or full human reconstruction of remote users.
c8	S	3D-LIVE indoor app	S	Unity3D Server	- Position of the user - Orientation of the user - Game phase triggers - Game Events	The 3D-LIVE indoor app knows everything about the behaviour of the user in the 3D world. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the standard Unity3D network API based on RPC (remote procedure calls).
c9	S	Unity3D Server	S	3D-LIVE indoor app	- Position of the remote users - Orientation of the remote users - Game phases triggers by remote events - Game remote events	As in the c8 link, The Main Screen app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API.
c10	S	Kinect Capturer app	S	RabbitMQ server	- Skeleton information - Full body reconstruction data - Measurements about skeleton information - Activity Recognition labels	The Kinect capturer connects to the RAbbitMQ server and updates a queue containing live skeleton data and full body reconstruction data of the user as well as Activity Recognition labels, evaluating the posture of the skier. Moreover as it measures metrics evolution over the time, it sends it to the appropriate exchange created for the ExperiMonitor. It uses the RabbitMQ protocol.
c11	S	RabbitMQ Server	S	Kinect Capturer app	- ExperiMonitor Instructions	On the same connection than the c10 link, the ExperiMonitor will send instructions regarding the experiment process to the Kinect capturer.
c12	H	Embedded GPS	S	3D-LIVE Smartphone app	- GPS data from the Smartphone	The app connects to the Android API to retrieve the data coming from the embedded hardware: GPS for positioning the user in the virtual world.
c13	H	Microphone (headset)	S	Mumble outdoor app	Voice	Wired connexion / Jack 3.5 stereo input. The voice is acquired by the microphone is retrieved by the app
c14	S	Mumble Indoor app	S	Mumble server app	Voice encoded data	The Mumble app compresses the voice data and transmits it to the Mumble server. Reversely it retrieves voice from the remote users.
c15	S	3D-LIVE Smartphone app	H	Smarphone display	Rendered image	The image / GUI + camera flow is rendered on the display of the Smartphone
c16	S	Mumble outdoor app	H	Headset	Stereo audio	The app renders the audio in the stereo system, wired by a jack stereo output.
c17	S	Mumble Outdoor app	S	Mumble server app	Voice encoded data	The Mumble app compresses the voice data and transmit it to the Mumble server. Reversely it retrieves voice from the remote users.
c18	S	3D-LIVE Smartphone app	S	RabbitMQ server	- ERS Requests - ExperiMonitor metrics data - Skeleton animation requests	The Smartphone application connects to the RabbitMQ server in order to reach the ERS (Environment Reconstruction

						service) and the ExperiMonitor. To the ERS, it will send requests for the weather information. To the ExperiMonitor it will send the measurements it takes during experiments. It will also query for skeleton animations of remote players. It uses the RabbitMQ protocol which is a advanced implementation of AMQP.
c19	S	RabbitMQ server	S	3D-LIVE Smartphone app	- ERS Results - ExperiMonitor Instructions - Skeleton information	Through the same connection than c18, the app will receive results from the ERS about the actual weather on site, and some instructions from the ExperiMonitor relative to the experiment process. More than that, the app subscribes to RabbitMQ queues containing the skeleton animation of remote users to display in the ski mask.
c20	S	3D-LIVE Smartphone app	S	Unity3D Server	- Position of the user - Orientation of the user - Game phase triggers - Game zones triggers - Game Events	The Smartphone app knows everything about the behaviour of the user in the 3D world. It will send it to the game server, in order to synchronize the game state with all remote users. It uses the standard Unity3D network API based on RPC (remote procedure calls).
c21	S	Unity3D Server	S	3D-LIVE Smartphone app	- Position of the remote users - Orientation of the remote users - Game phases triggers by remote events - Game remote events	As in the c20 link, The Smartphone app needs to by synced with the remote users. In that way the game server sends to the app the current game state through the Unity3D network API.
c22	S	Environment Reconstruction Service	S	OpenWeatherMap	Weather requests	The ERS(Environment Reconstruction Service) uses the OpenWeatherMap open API to retrieve weather information at the GPS coordinates of the venue.
c23	S	OpenWeatherMap	S	Environment Reconstruction Service	Weather Results (Temperature, Humidity, Cloud coverage, Rain fall rate, Snow fall rate,)	The Weather effects are sent to the Environment Reconstruction Service (which will interpolate the weather data coming from different sources like Sensordrone to generate a weather information at any GPS location of the venue)
c24	S	RabbitMQ Server	S	Environment Reconstruction Service	- ERS Requests - Experimonitor instructions	The ERS(Environment Reconstruction Service) connects to the RabbitMQ server in order to listen to weather requests at any provided GPS location that belongs to the venue. Requests will be sent to it through the RabbitMQ protocol. Moreover the ExperiMonitor will send instructions to the ERS about the experiment process, through the RabbitMQ protocol as well.
c25	S	Environment Reconstruction Service	S	RabbitMQ server	- ERS results - Experimonitor measurements	Using the same connection than c31, the ERS (Environment Reconstruction Service) send weather results to RabbitMQ

						clients requesting weather information. Moreover the ERS send measurements taken like the rate and the nature of the requests.
c26	S	ExperiMonitor	S	RabbitMQ server	- Experiment instructions	The ExperiMonitor is the service that handles the experiment monitoring process and live visualization of metrics. It connects to the RabbitMQ server to allow remote applications to send their metrics using a RabbitMQ protocol.
c27	S	RabbitMQ Server	S	ExperiMonitor	- Experiment metrics	Using the same connection than c33, the ExperiMonitor will get from the different clients experiment metrics, store them and visualize them in real time.
c28	S	3D-LIVE Smartphone app	S	Recon Snow2 app	- Profile Info/Positions/Orientations/Animations of remote players - Game zones and game phases of the current player	The smartphone app is paired via a WiFi connection to the Recon Snow 2 Heads Up Display. This allows the Smartphone application to send an important set of game data in real time to the HUD application, that will display the avatars and game status on the virtual environment on the screen of this HUD.
c29	S	Recon Snow2 app	H	Recon Snow2 HUD	Virtual Environment to render	The HUD application processes the incoming data in order to render the continuously updated environment to the outdoor skier on the HUD
c30	S/H	Oculus / CAVE	S	3D-Live app	- Head orientation and position tracking	Thanks to the MiddleVR middleware, the different sensors provided by the CAVE and Oculus systems send data to the 3D-Live application, in order to update in real time the position and orientations of the cameras in the virtual world, adapting the point of view of the player

8.4 Appendix D: Description of the different components of the prototype 3

List of components in the diagrams / designation		
Designation	**Description**	**type (Hardware/ Software)**
3D-LIVE indoor app	The indoor user application deployed to live the 3D-Live experience. It has been developed with Unity3D for the Golf and Ski scenarios, and with realXtend for the Jogging scenario	S
3D-LIVE Smartphone app	The outdoor user application deployed to live the 3D-LIVE experience. It has been developed with Unity3D for the Skiing scenarios, and with Eclipse for the Jogging scenario	S
5.1 sound system	A sound system made of 5 speakers and 1 bass to render the sounds all around the indoor user. It brings the audio immersion to the user.	H
Android tablet App	The outdoor user application deployed to live the 3D-LIVE Golfing experience. It has been developed with Unity3D.	S
Android Tablet Display	The display attached to the tablet	H
Audio Headset	The audio headset used in 3D-LIVE. Depending on the scenario it can be wired or not.	H
CAVE like display	The Virtual Lab in CIE in Oulu used for Jogging indoor High-End setup	H
Club + EXEL-S1 IMU sensor	The device used for the Golfing scenario. Golfers use a normal club, which has an inertial sensor attached to it, to analyse strikes characteristics and move the virtual ball.	H
Embedded Camera	The camera in tablet	H
Embedded GPS	The GPS chip in the tablet/smartphone	H
Embedded Gyroscope	The gyroscope chip in the tablet	H

Environment Reconstruction Service	The service collecting weather data from different sources and calculating the best interpolated information at any given GPS location.	S
EOS App	The Environment Observation Service app, is an Android app that collects data from the Sensordone, to feed the Environment Reconstruction Service.	S
ExperiMonitor	The interface to monitor the experiments. It connects to RabbitMQ server to allow multiple measurement sources to share live metrics evolution, and export the data afterwards.	S
Kinect Capturer app	The application that extracts data from the Kinect sensors, in order to create and send over the network either an animation frame of the user and a full reconstructed live 3D mesh of the user.	S
Kinect x n	The Microsoft Kinect sensors. Providing depth maps and RGB images	H
Main Screen app	The indoor application deployed for the Golfing scenario, used to render only the 3D content of the indoor 3D scene. It is made with Unity3D	S
Microphone	The Microphone used for the Golfing scenario indoor application.	H
Microphone (headset)	The microphone attached to the headset of the user	H
Monitor Screen app	The indoor application deployed for the golfing scenario, used to render the game state and enable interactions in the game for the indoor user.	S
Monitor touch screen	The touch screen of the monitor, which allows the indoor golfer to interact with the application	H
Mumble app	The Voice application service used in the Jogging scenario	S
Mumble Voice server	The voice application server used in the Jogging scenario	S

OpenWeatherMap	The Open service providing weather information world wide. The Environment Reconstruction Service requests this service to interpolated weather data at any given GPS location of the venues.	S
Pro Ski Simulator + IMU	The fitness device Pro Ski Simulator has been equipped with a IMU (on Arduino board) to communicate the indoor ski application the orientation of its moving platform, to allow indoors high end to turn on the slope	H
RabbitMQ server	The server deployed to handle most of data flows not related to the game state itself. It uses the RabbitMQ protocol.	S
realXtend Server	Game server for the Jogging scenario. It handles all the data flows between indoor joggers, and by the intermediate webservice, the data flows of the outdoor jogger. It uses the kNet network protocol.	S
Sensordrone	The sensor capable of retrieving local weather information and send it to the EOS app, in order to feed the Environment Reconstruction Service.	H
Smarpthone display	The display of the Smartphone	H
Unity3D Server	The Game server for the Golfing and Skiing scenarios uses a custom game synchronisation API based on Network Views, Remote Procedure Calls and StateSynchronisation. The RakNet library is the backend solution to that API.	S
Web Service	A Web service designed to retrieve the incoming data from the Jogging outdoor app (GPS, metrics about the perofrmances) and transmit it to the realXtend server	S
Wide Display	A wide display to render the indoor application	H
Wii BB	Wii Balance Board from Nintendo allowing the retrieveal of weight distribution of an indoor low-end user, to interact with the virtual ski environment.	H

8.5 Appendix E: Report on sensing technologies investigations

Introduction

During the design of the 3D LIVE Tele-Immersive environment (WP2, in particular at Laval meeting) it became clear that an important sensor input is from motion sensors.

The motion sensors functionality is to retrieve the orientation of the body in space are based on inertial sensors. Micro-electro-mechanical-systems (MEMS) technology allows for implementing them in computer chips since a few years, which makes them small, low cost and able to run on small batteries, all necessary pre-requirements for consumer goods.

They are not suited to calculate inertial navigation (including the x,y,z coordinates) like in ships, airplanes, rockets or missiles, as their accuracy is several orders of magnitude lower (as well as cost, size and weight). The reason is that this needs a double integration of the accelerometer signal after subtracting earth gravity in the correct direction and even the smallest errors sum up very fast, therefore MEMS sensors run away within a few seconds. Other external data (GPS, Kinect, UWB) information would be necessary to correct for this drift.

Taking sensor data from all 3 axes, for example with the MTx–System of XSense, angular orientation can be derived. This unit consists of 3x 1-D gyroscopes, 2 of them are located on small sub-boards arranged vertically to the main board, normal to each other. There are 2x 2-D accelerometers used to cover the 3 axes in space, the magnetometer comes in a full 3D version already within a chip.

Fig. 1: construction of an

The gyroscopes capture all rotations in space, i.e. the change of the orientation. Calculating the orientation is a single step of integration, which will also drift over time, but this can be controlled. To avoid the drift, during resting time, 2 fixed directions in space are used to correct the calculated orientation data: the direction of earth gravity as measured by the accelerometers and the direction of the earth magnetic field ad detected by the magnetometer. The accuracy according to their documentation is about 0.5 degrees in rest and about 2 degrees when moving, respectively.

In very recent years the miniaturization got further and enabled 3 gyroscopes integrated into one 3-D package (a chip of a few mm in size), and very recently there are 2 or even all 3 of the 3-D sensors available within one package available or announced. There are even chips or modules available now which contain standard software to calculate the orientation out of the sensor data, in particular the iNEMO 9x9mm module by ST in Q4 2013.

Fig 2: iNEMO

Matching requirements for 3DLive

Fig 3: MTx + Xbus

The preferred solution would be internal calculation of orientation, which are expressed in quaternions. Quaternions improve with the frequency of sensor data collection (50 Hz min required).Once quaternions are calculated, a transmission rate of them in video frequency (25 Hz) is sufficient. Furthermore, quaternions reduce the amount of data transmitted, as they consist of 4 values of 2 bytes each, raw sensor data are 9 values of 2 bytes each.

Wireless transmission of data is essential. Cables on the body reduce movability and would put acceptance on a low rate.

Further important are low power wireless transmission protocols to keep battery size small. The good old X-sense Mtx doesn't fit these criteria.

SportsCurve performed a market survey of solutions for wireless transmission of orientation data by inertial sensors, which are listed in the table.

The results from the connectivity study by Benjamin Poussard & Guillaume Loup from July 22nd 2013, suggested that, in order to avoid time delays, a direct connection to the smartphone is the way to go. Very important is an open wireless communication standard, to be able to get data directly in applications in the smartphone. This, as well as the unit prices which are not suited for consumer products, rule out X-sense and APDM; moreover it is not clear how much support the companies can provide to integrate their products in the Tele-Immersive platform. If it is at all possible, it would take several months.

Furthermore, the most advanced technology, iNEMO by ST, would need to be integrated into a newly to be developed hardware project, which will not be available during the time frame of the 3DLive project.

Fig 4: EXL-S3

Initially the most promising sensor system was the one by TI (Texas Instruments). It is with 25$ the only really consumer prized solution, development time is on the shorter side. Xybermind sensors are better but more costly, both in units and development costs. It is the only other available option with reasonable development time. However, during some testing it turned out that Bluetooth 4.0 does not offer sufficient transmission capacity for this large amount of data to support such 3D LIVE related sporting activities. Hence, many of the available sensor solutions in the market could not be used and the natural choice was then the EXEL EXL-S3 sensor. This sensor is the successor of the EXL-S1, which was first considered for the project but proved to not meet requirements. The EXL-S3 sensor provides efficient transmission of accurate quaternion data and is considered especially suitable for body movement analysis. It has an open API, which is an important requirement, and is pricing wise at a reasonable level. The sensor is also considered to have a fairly low drift, which is discussed more in detail below.

Drift of the EXEL sensors

All inertial sensors typically suffer from accumulated error because the guidance system is continually adding detected changes to its previously calculated positions. Any errors in measurement are accumulated from point to point and can hence increase over time. This leads to drift, or an ever-increasing difference between where the system thinks it is located and the actual location. In the EXL-S3 model, the inertial sensing chip is Invensense MPU-9150. The manufacturer hasn't specified the drift over time but EXEL's tests indicate that the drift for these are low and accurate enough for body movement applications such as in the 3D LIVE project. The firmware of EXLs3 applies a precise compensation and the sensors are tuned by EXEL before shipment in order to have a nominal precision <1%. Depending on the operating conditions a drift might occur over time though. The values, which normally drift the most, are the gyroscope offset and the magnetometer offset. The latter is very sensitive to induced magnetic fields, which can cause permanent magnetization of the sensor, hence it should never be left close to magnets or big metallic objects which can "concentrate" the normal earth magnetic field. Recalibration of the sensors can easily be done via a PC software.

Comparison of wireless 9-D sensor systems

Company	X-sense	APDM	Texas Instruments	ST, to be integrated	Xybermind	EXEL
Product photo						
Product name	MTw	Opal	CC2541DK-SENSOR	iNEMO (integration + BTLE t.b.done)	aims + BTLE	EXL-S3
Unit price €	1920	1800	20	50 (module only), 300 integrated	250 + 50	300
Initial costs	13.900 for 6 Sensors + unknown integration	11.000 for 6 Sensors + unknown integration	0 (6 Sensors in house) + 7.500 for integration	1.800 + 24.500 for integration	1.800 + 17.500 for integration	1800 for 6 sensors + 3600 integration
Development time	tbd (a few months)	tbd (a few months)	a few weeks	about 1 year	a few weeks	a few weeks
Wireless protocol	proprietary	proprietary	Bluetooth 4.0	Bluetooth 4.0	Bluetooth 4.0	Bluetooth 2.1 class 1
Live data	yes	yes	yes	yes	yes	yes
Max. live data rate for 1/5 devices [Hz]	120/75	128/128	26/15 @iOS, 133/33 @PC, Android	26/15 @iOS, 133/33 @PC, Android	26/15 @iOS, 133/33 @PC, Android	200 Hz raw data, 100 hz with orientation
Orientation calculation	internal	external	external	internal on ST module	internal	internal
On data frequency	1600	128	25	800	400	200
Local Data storage [Gb]	no	8	no	8	8	1
Housing form, Size [mm]	Rectangle, 58x35x15	More Squared, 50x37x14	Oblique, 65x35x15	USB stick, 85 x 21 x 16	boards: 60x20x6 (board) + 35x17x4 (BTLE) + Battery	Rectangle, 54x33x14
Volume [cm³]	27	24	27	23	20	25
Weight [g]	29	22	25	25	20	22
Battery	internal, re-chargeable	internal, re-charchable	internal, non rechargeable CR2032 cell	internal, re-chargeable	external, re-chargeable	internal, re-chargeable LiPo
Battery time [h]	4	8	NA	1	4	4
Extendable with external batteries	no	no	NA	yes	always external	Yes using USB plug
Accelerometer max [g]	16	6	8	16	16	16
Gyroscope max [deg/sec]	1200	2000	2000	2000	2000	2000
Additional Sensors			air pressure, surface temperature, humidity	air pressure	air pressure	3 axis magnetometer (Full scale +-1000 uT, Temperature)
External Accelereometer for high g [max g]	no	no	no	24, 100, 200, 400	24, 100, 200, 400	no
External 9D minimal size (3,4cm³, <2g)	no	no	no	yes	yes	no
AD converter for other external sensors [bit]	no	no	no	12, 16	12, 16	no
Suited for gait analysis data recording	limited (data loss when wireless connection lost)	yes	limited (data loss when wireless connection lost)	yes	yes	yes
Suited for sports motion data recording	limited (data loss when wireless connection lost, frequency low)	limited (range of acc. low, frequency low)	limited (data loss when wireless connection lost, frequency low)	yes	yes	yes
Suited for sports live data transmission	yes	limited (range of acc. low, orientation external)	limited (range of acc. low, orientation external)	yes	yes	yes

Use in 3DLive scenarios

Outdoor: 4 sensors:

- For the Golfing scenario:1 placed on the back, 1 sensor placed on the elbow, 1 one the shoulder and 1 sensor on the golf club near the handle. Potentially the back sensor could be placed in a cap at the back of the head to recreate head movement instead. This will be evaluated during the testing.
- For Jogging and Ski scenario: 1 sensor placed on the back, 1 on the head, 1 on the shoulder and the 1 on the elbow of the user. To animate the movement of one arm.

Indoor: 1 sensor

- For the Golfing scenario, one sensor attached on the golf club.