

## Report Contents

### 1. INTRODUCTION

*General description of the project and brief summary of the report.*

In this project, I created a software (middleware) that can recognize a person's skeleton using a 3D motion capture camera (either the Kinect for Xbox One or Kinect Azure, both by Microsoft) and can send this information to Unity in real-time. To do that, I had to learn how to access and work with the data provided by the Kinect SDKs (Software Development Kits). During programming, it was necessary to create a code that could access the skeleton data provided by the Kinect SDK and convert it into a Unity compatible format, which required a knowledge of both the Kinect SDKs and Unity engine.

The project I was working on is being used in the process of creating Unity games for children in rehabilitation process, for example: "Phiby's Adventures". My project was based on the K2UM (Kinect to Unity Middleware), a program created by a former student as a final degree project. I had to create a new middleware based on that project to make my program compatible with the data receiving asset created in Unity.

I rewrote the code of the K2UM middleware, in order to use an existing implementation of a User Datagram Protocol (UDP) connection in the application to transmit the skeleton data in real-time. This required refactoring the code to make it easier to understand and make it more readable. The UDP connection implementation was a crucial part of the project because it allowed me to transmit the skeleton data in real-time, which was necessary to achieve the project's goals.

I ran into a lot of issues and challenges throughout the project, like problems with the Kinect SDKs or difficulties debugging the code. Making a translation layer for the joints that would enable me to translate the joint data from the Kinect into a format that Unity could understand was one of the biggest challenges I had to face. This was a time-consuming and difficult task that required me to write code to convert the joint data and map it to the proper Unity joint names. However, with careful planning and creative problem-solving, I was able to overcome this obstacle and discover a workable solution that enabled me to use the skeleton data in Unity.

Overall, working on this project was rewarding but challenging, I gained a lot of knowledge about the Kinect SDKs. I learn how to incorporate it into software, and how to send real-time motion data over a UDP connection. These abilities and knowledge could be really useful in my future career.

### 2. CONTEXT

*UPM centre in which the Project has been carried out, main responsibilities, interaction with other people (students, researchers, etc.), lab description (if appropriate), etc.*

The project described in this report was carried out at the UPM center CITSEM (Software and Multimedia Systems Technologies for Sustainability Research Centre). This center aims to strengthen and promote R&D&I and attract research talent at the UPM South Campus.

As a student working on this project, my main responsibilities included writing and debugging code, and integrating the Azure Kinect SDKs. I also had to keep track of my progress and report regularly to my supervisor with which I had weekly meetings.

The lab in which the project was carried out was well-equipped with a variety of tools and resources for software development, including computers, software licenses, and other necessary equipment

like Xbox One Kinect, Azure Kinect and Xbox 360 Kinect, the latter I did not need to work with. The lab also had a supportive and collaborative atmosphere, which was helpful for working on the project.

Overall, the UPM Centre provided an ideal environment for carrying out this project, with excellent resources and support for students and researchers.

### 3. PROJECT OBJECTIVES AND TASKS

*Completed tasks, with indication on how they have contributed to the fulfilment of the project objectives, problems and difficulties, proposed solutions, etc.*

The global objective at the beginning of the project, was the upgrading of a middleware called K2UM (Kinect to Unity Middleware), created by a former final project student, that transmits skeleton data from a Kinect camera (® Microsoft) to Unity. The version of that middleware was based on the Kinect v2 belonging to Xbox One and had to be adapted to work with the newest Kinect release called Kinect Azure.

The main objective of this project was to implement compatibility with the Kinect Azure camera into the existing project, unfortunately this project does not support the new library required to achieve this goal. So, I needed to create the project from scratch. I chose WPF with use XAML for creating user interfaces and C# for logic. C# was used in the previous project, so I was able to base my work on it. After determining what technologies, I had to use, the next goals were to rewrite the existing UDP layer to use it in the new project, add support for Kinect Azure, create a layer translating joints from Kinect to Unity, and add support for Kinect One that was already created in the previous project. Unfortunately, due to the use of different technologies to create interfaces, the work required a lot of adaptation to the new project.

While working on this project, I finished several tasks, including:

- Integrating the Kinect SDKs: I had to use the SDKs of two different Kinect cameras in my program in order to detect a person's skeleton, the one for Kinect One and the one for Kinect Azure. As both versions are very different (no compatibility provided by Microsoft), this required writing two separate classes that acquire the information that the Kinects produce and to parse it, such that one can make use of it in Unity.
- Developing a class to translate Kinect joints to be compatible with Unity: I had to make a class that would convert the joint data from the Kinect into a format that Unity could understand in order to use the skeleton data. It was necessary to map the joint data to the relevant Unity joint names and convert it to the correct orientation.
- Design the user interface: to make the application user-friendly and easy to use, I have created a modern user interface that allows users to start, stop, and chose which Kinect is used for skeleton detection, then the data is transferred to Unity using UDP. The user interface was an important aspect of the project as it gave the user a way to interact with the application and control skeleton recognition and data transfer.
- I rewrote the code of the existing K2UM project to make it more readable: To improve the maintainability and readability of the code base, I rewrote some of the existing code, which consisted of splitting one file that was in the previous project into many smaller components. This requires refactoring the code to follow best practices and include informative variable names and comments. This work was essential to maintaining and improving the application as it made the code easier to understand and edit.

- Testing the application: I ran a series of tests to ensure that the skeleton data was correctly identified and sent. I made it to confirm that the application was working correctly. These tests were very important to verify the functionality of the application and ensure that it met the project goals. I put a lot of effort into these tests and code tweaking to make sure everything works.

After finishing all the objectives of the project, the final architecture consists of two classes “AzureKinect” and “OneKinect” that receive the information from the Kinect SDKs. Those classes used “MJoint” to translate received data into one format that is later used inside of the “KinectMiddleware” class to create a string that is send to Unity using the “UDPSend” class. In a great simplification, this is how the system works, a more detailed description is provided of the “Figure 1”.

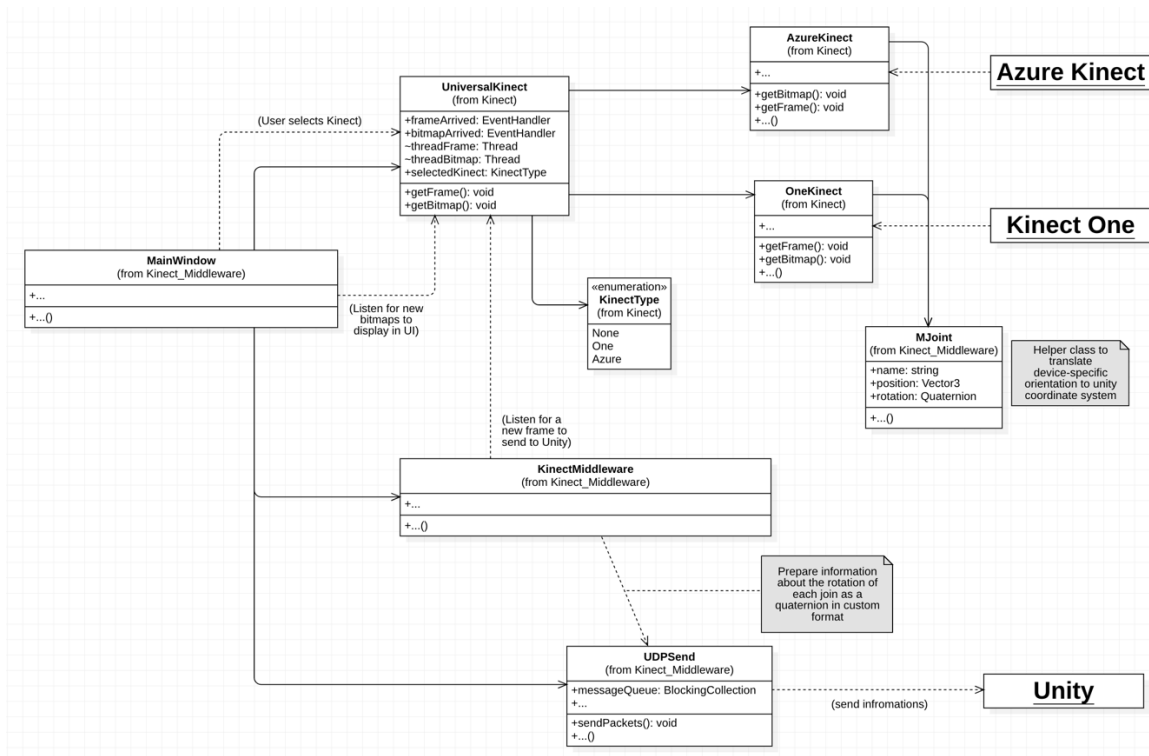


Figure 1: Architecture of the system

While working on the project I encountered several problems and difficulties, those includes issues with the Kinect SDKs or challenges with debugging the code. Although, I manage to overcome these issues by careful planning and problem-solving. I was able to find solutions to the problems that were not immediately obvious. I searched for solutions by analysing the previous project to understand how the problem was solved. I also used documentation provided by the Microsoft. Unfortunately, it was mainly about C and C++, the documentation about C# was quite poor, but fortunately, in places where Microsoft did not deliver, there were people on the forums who faced the same problems as me and shared their solutions. The biggest problems I faced making the translations layer because very few people describe their process of translating joints from both Kinects to Unity.

Overall, I was able to design an application that satisfied the project's goals and showed the practicality of our system for capturing and delivering real-time motion data by accomplishing these tasks and addressing the problems and challenges that arose. The project was a difficult but

gratifying experience, and I am sure that the knowledge and abilities I acquired will be useful in my future career.

#### **4. TECHNOLOGIES AND EQUIPMENT**

*Description of algorithms, equipment, software, and other tools used during the project.*

For this project, I used a variety of tools and technologies to develop an application for detecting and transmitting real-time motion data from Kinect SDKs to Unity. These included:

- Visual Studio: This integrated development environment (IDE) was used for writing and debugging the code for our application.
- Kinect v2 (Xbox One): This motion-sensing device was used to detect the skeleton of a person. It uses a combination of cameras and infrared sensors to track the movements of a person's body.
- Kinect Azure: Like the Kinect v2, this motion-sensing device was also used to detect the skeleton of a person. It is a more advanced version of the Kinect v1 (Xbox 360), with higher resolution cameras and more powerful processing capabilities.
- An existing project: As a starting point, I used an old project as the base for my application. This provided a foundation for me to build upon and allowed me to focus on the specific tasks of integrating the Kinect Azure.

With these tools, I was able to develop an application that can detect a person's skeleton using either the Kinect v2 or the Kinect Azure and transmit this data in real-time over a UDP connection to Unity game.

#### **5. DEVELOPED COMPETENCES AND SKILLS**

*Explanation on how the tasks performed have helped the student in developing the competences and skills defined in Annex I.*

Developing this application and working on the tasks involved in this project has helped me in several ways by allowing me to develop a range of competences and skills. Some examples of these include:

- Programming skills: By working with Visual Studio and writing C# code for the application, I have had the opportunity to practice and improve mine programming skills. This includes skills such as problem-solving, debugging, and code organization.
- Integration skills: I have had to integrate the Xbox and Azure Kinect SDKs into the application, which required me to understand how these hardware's work and how to use them effectively. This has helped me to develop skills in integrating different technologies and making them work together.
- Communication skills: I have had to communicate effectively with the supervisor throughout the project, including presenting mine progress and results. This has helped me develop mine communication skills, including the ability to clearly and concisely explain technical concepts to a person which does not work on the project.
- Time management skills: By working on a project with specific goals and deadlines, I have had to manage mine time effectively and prioritize tasks. This has helped me to develop mine time management skills and the ability to work efficiently.

Overall, working on this project has provided me with a wealth of learning opportunities and has helped me to develop a range of valuable competences and skills.

## 6. CONCLUSIONS

*Evaluation on how the project has contributed to your training, and how the supervisor has influenced this process.*

The completion of this project has been a valuable learning experience for me, and has contributed significantly to my training and development as a software developer. Throughout the project, I had the opportunity to develop a range of technical skills, including programming in C#, working with SDKs, debugging and troubleshooting code. I also had the chance to apply my knowledge of object-oriented programming principles and design patterns, which helped me to structure and organize my code in a clear and maintainable way.

In addition to these technical skills, I also had the opportunity to develop my problem-solving and critical thinking abilities, as I encountered a number of challenges and had to find creative solutions to overcome them. This required me to think critically about the problems at hand and to use my knowledge and skills in a practical and effective way. It also helped me to learn how to manage my time and resources effectively, as I had to prioritize tasks and work efficiently in order to meet deadlines.

Throughout the project, I received guidance and support from my supervisor, who provided valuable feedback and suggestions on my work. This helped me to stay on track and make progress on the project, and I am grateful for the time and effort that my supervisor put into helping me to succeed. This was a valuable experience, as it gave me the chance to see different perspectives and approaches to problem-solving, and to learn from the successes and challenges of others.

Overall, the completion of this project has been a valuable and rewarding experience, and I am confident that the skills and knowledge that I have gained will be valuable in my future work. I am grateful for the opportunity to have worked on this project, and I believe that it has helped me to become a better software developer and a more skilled individual.

## 7. PROJECT LOG

*List and brief description of the weekly tasks in chronological order.*

Schedule of activities from the beginning of the project:

3 weeks:

During this period, I spent a significant amount of time studying the existing codebase and identifying areas that could be improved or updated. This involved reading through the code, understanding how it worked, and making notes on areas that could be improved or optimized. I also conducted research on the latest developments in the field of Kinect and motion capture, in order to identify technologies or approaches that could be used to enhance the project.

4 weeks:

During this period, I worked on creating the initial version of the project based on the previous version. This involved writing new code and integrating it with the existing codebase, as well as refactoring the code to make it more readable and easier to understand. I also spent a significant amount of time testing and debugging the code, in order to ensure that it was functioning correctly.

3 weeks:

In order to make the project compatible with the Azure Kinect, I had to write code to access the data provided by this device and interpret it in a way that could be used by the application. This involved learning about the features of the Kinect SDK and how to access and interpret the data it provided. I also had to write code to translate the data from the specific format of the Azure Kinect into a format that could be used in Unity.

2 weeks:

One of the key challenges I faced during this project was creating a translation layer for the joints that would allow me to convert the joint data from the Kinect into a format that could be understood by Unity. This was a time-consuming and challenging task, as the documentation provided

2 weeks:

In order to add support for the Xbox One Kinect to the project, I had to write code to access the data provided by this device and interpret it in a way that could be used by the application. I also had to write an abstract Kinect class that could be used in the rest of the project to extract data from both of Kinects in a consistent and efficient manner.

1 week:

During the final week of the project, I focused on completing the final report which present my overall work over the semester.