QUIVRON Martin
EUTAMENE Adel

MyRobo

Supervised by: MASANOBU Koga
2024

SUMMARY

1) Technologies to improve in web Interface

2) Technologies to improve the in AR Interface

3) Technologies to improve the performance of control of robot (simulation)

4) How the simulation works

1. **Technologies to improve the user interface in web Interface :**

Having a good user interface (UI) in a web application that utilizes augmented reality (AR) is crucial for many reasons:

First it will enhance the user experience by making it more intuitive and enjoyable. The interactions have to be clear, offer feedback and be smooth for a more engaging experience.

Secondly it will reduce the cognitive load (minimizing the mental effort to use the app). It is crucial because users are already engaging with both the physical world and virtual elements simultaneously, which can be mentally taxing. A streamlined UI can help reduce this cognitive load by presenting information in a clear and organized manner, making it easier for users to understand and interact with AR content.

To finish with accessibility: A well-designed UI can improve accessibility for users with disabilities, ensuring that everyone can engage with the AR application effectively. We have to consider text size, color contrast, and navigation options that accommodate diverse user needs.

For this project we decided to use **Bootstrap**, a front-end framework that facilitates the development of responsive and mobile, web projects. **HTML, CSS,** and **JavaScript** frameworks for building modern and visually appealing UI.

https://getbootstrap.com/

Responsive Grid System: **Bootstrap** utilizes a responsive, mobile-first grid system. Allowing developers to create layouts that adapt to various screen sizes and devices.

Pre-styled Components: **Bootstrap** provides a rich collection of pre-styled UI components. There are buttons, forms, navigation bars, dropdowns, alerts, and modals. Customizable and can be easily integrated into web projects.

***JavaScript** Plugins:* **Bootstrap** comes with a variety of **JavaScript** plugins, like carousels, modals, tooltips, popovers, dropdowns, and scrollspy.

Customizable Themes: **Bootstrap** allows developers to customize the appearance of their projects.

Browser Compatibility: **Bootstrap** has cross-browser compatibility and consistent rendering across different web browsers, including Chrome, Firefox, Safari, Edge, and Internet Explorer.

2. **Technologie to improve the in Ar Interface:**

A well-designed AR interface can significantly enhance the user experience by providing immersive and engaging interactions. AR allows users to interact with digital content overlaid onto the real world, creating novel and exciting experiences that captivate users and keep them engaged. AR interfaces enable users to interact with digital objects and information in real-world contexts. The AR interfaces will convey information and instructions more effectively than traditional interfaces by overlaying digital content directly onto the user's surroundings. It has the potential to provide contextual information and personalized experiences based on the user's location, environment, and preferences.

For this project we decided to use **Babylon.js**, a powerful, open-source 3D engine used for creating complex and interactive 3D graphics in web applications. And we also used **WebXR**, a web standard that provides support for virtual reality (VR) and augmented reality (AR) devices on web browsers.

3. **Technologies to improve the performance of control of robot**

Improving the performance and control of robots with **Babylon.js**, **WebXR**, and **Oimo.js** involves leveraging each technology's strengths. **Babylon.js** allows for the creation of detailed 3D models of robots and their environments, making monitoring and control easier. It supports smooth animations for precise robot movements and includes physics engines like **Oimo.js** to simulate real-world physics, making robot actions more realistic.

WebXR enhances interaction by providing immersive experiences through VR (Virtual Reality) and AR (Augmented Reality) devices. This allows users to control robots in a more intuitive and natural way, with real-time feedback. For instance, you can use VR controllers or hand gestures to interact with the robot, seeing immediate responses to your actions.

Oimo.js, specifically, is a lightweight physics engine that integrates well with **Babylon.js**. It efficiently handles physical simulations, including collisions and dynamic interactions between objects. Using **Oimo.js** with **Babylon.js** allows for realistic physical simulations without compromising performance.

By integrating these technologies, you can achieve real-time visualization and control of robots. **Babylon.js**'s 3D capabilities combined with **WebXR**'s immersive interfaces and **Oimo.js**'s efficient physical simulations create a powerful system for monitoring and manipulating robots. To ensure smooth performance, it's essential to optimize 3D models and write efficient code. Leveraging GPU acceleration can further improve rendering and computation speeds, ensuring that the system remains responsive and efficient.

4. How the simulation works

When we launch the simulation, the robot obtains the coordinates of the end point. It will then try to head towards the end point, if there is no obstacle between the robot and the end point, then the robot will rotate to face the end point and move straight forward.

Now when obstacles are across the path, the robot detects it with its laser (raycast). First the coordinates of the laser detection point are stored in an array. Then the robot will search for the corner closest to the arrival point. When the robot reaches the nearest corner, it will move forward again until it encounters a new obstacle or goes directly to the end point. All of this is stored in an array to ultimately have a well-defined path.