Réalité virtuelle et Augmentée pour la maintenance d'usines

Guide Développeur

BONHOURE Gilles
DEREYMEZ Maxime
LACHARTRE Denis
LESAGE Lucas
ZENNOUCHE Douria

Table des matières

1. Introduction

CyberHoloCampus2055 est une application permettant la localisation indoor d'un individu et la maintenance de bâtiment à l'aide d'un casque de réalité augmenté Hololens. Il est possible de visualiser sa position directement sur un plan 3D du bâtiment et de recevoir des notifications. Un étudiant peut également demander sa salle de cours, cette dernière sera indiquée en surbrillance sur le modèle 3D. Connaissant donc sa position par rapport à la salle vers laquelle il doit se rendre. il peut facilement trouver son chemin.

2. Fonctionnalités

CyberHoloCampus présente plusieurs fonctionnalités :

- Affichage d'un modèle 3D
- Localiser indoor d'un utilisateur
- Isolation d'un emplacement par effet de surbrillance
- Affichage de notification et de tâches assignées (maintenance)
- Afficher d'informations concernant un bâtiment (ex : menu d'un restaurant universitaire)

3. Technologies utilisées

a. Unity 3D

Unity3D est un logiciel développé par Unity Technologies, utilisé principalement pour son moteur de jeu qui permet de facilement et rapidement créer des prototypes de jeux, jeux sérieux, ou même des outils. Les applications développées peuvent être déployées sur la plupart des plateformes actuelles (Pc, smartphone, Hololens), et cela grâce à un développement cross-platform. La facilité d'accès, sa haute modularité et l'utilisation du langage C# ,créé par Microsoft, les ont poussés à choisir Unity pour le développement de l'Hololens.
pour plus d'informations : https://unity3d.com/fr

b. JHipster

JHipster est un générateur d'applications microservice utilisant le framework Spring Boot et Angular. JHipster génère en quelques lignes de commandes, une application dans son intégralité avec certains services intégrés tel que la création de compte ou encore l'authentification. À sa création, un projet est complètement configurable : il est possible de choisir le type de base de données à utiliser, ou encore quels méthodes seront utilisées pour l'authentification des utilisateurs...
pour plus d'informations : http://www.jhipster.tech/

4. Prérequis

- Maven : https://maven.apache.org/
- NodeJs / npm : https://nodejs.org/
- MySQL : https://www.mysql.com/
- Unity : https://unity3d.com/
- Prérequis Hololens : https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools

5. Configuration et utilisation

- Back end

  Application Java Spring censée tourner sur un serveur faisant le lien avec la base de données et donnant accès à ces données via une API RESTful.

  1. Dans le fichier **src/main/resources/config/application*.yml**
     indiquer les identifiants de la base de données
     note : * en fonction du mode de déploiement, dev ou autre
  2. Exécuter la commande ***./mvnw*** dans un terminal

     Note : il est possible de configurer d'autres fichiers selon si vous souhaitez des logs plus détaillés (ex : netflix eureka).

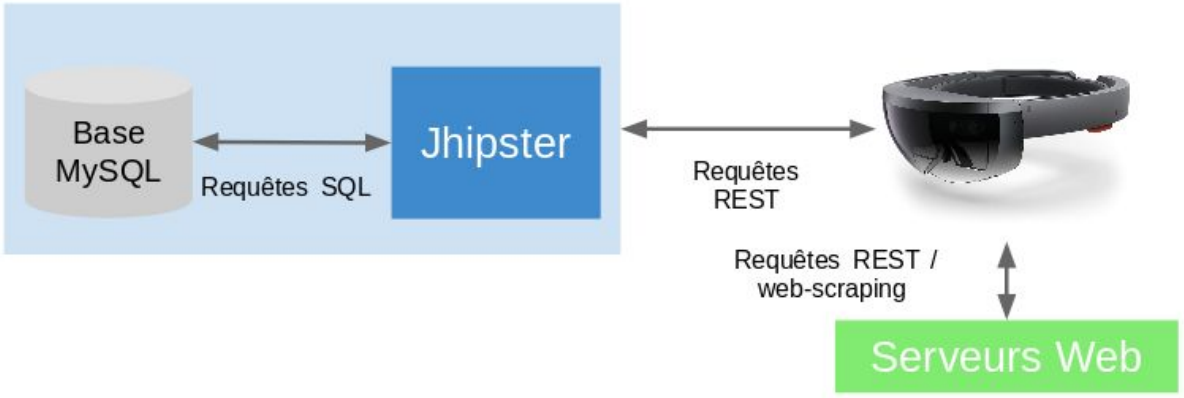
- Front end (JHipster)

  Si l'application n'a pas encore été lancée, installer les dépendances via la commande ***npm install***.
  Lancer l'application avec la commande ***npm start***.

- FrontEnd (Hololens)

L'application est installée sur le casque Hololens, elle peut donc être lancé directement depuis le casque Hololens.

6. Architecture générale



Plusieurs composantes qui interagissent. Le casque Hololens récupère des informations depuis la base de donnée en communiquant avec l'API via des requêtes REST.
Le casque Hololens communique également avec d'autres serveurs, par exemple pour la récupération des salles de cours depuis le site ADE. Ce genre de communication est également assuré grâce à des échanges REST.

- Vue logicielle

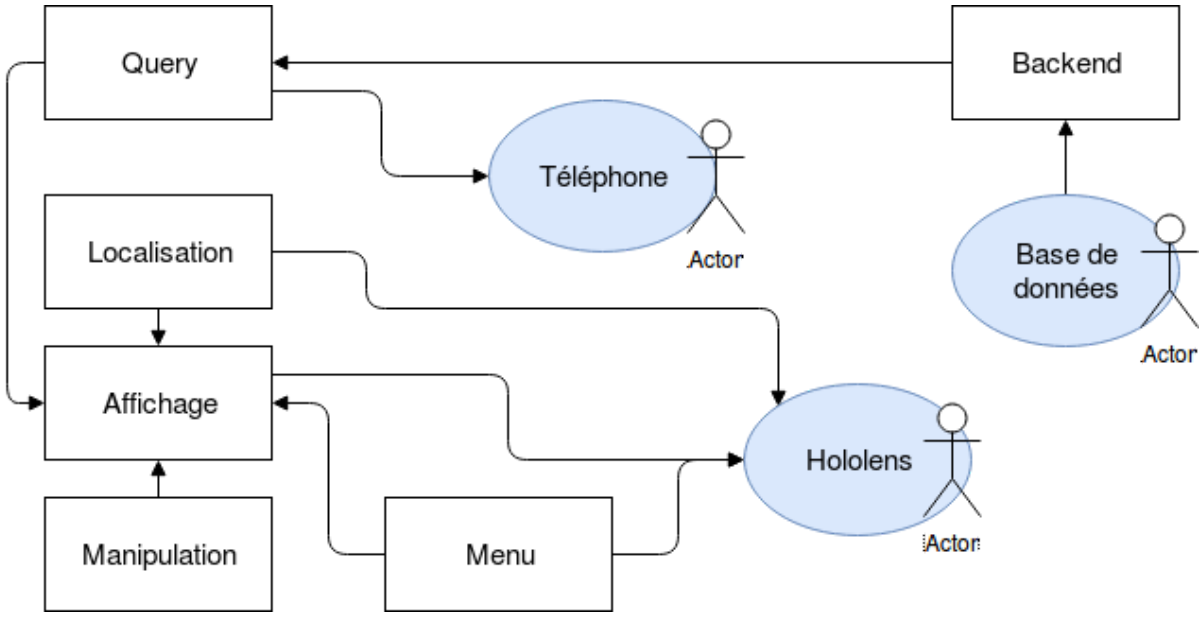
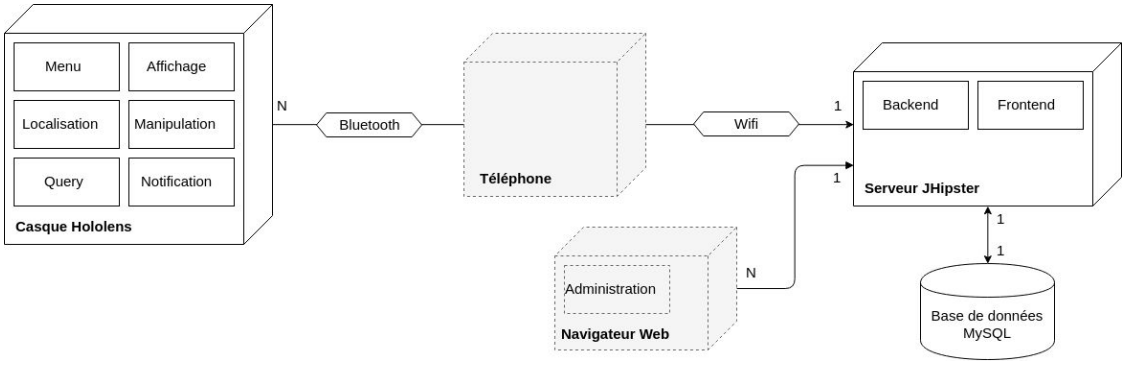


Diagramme de séquence pour une localisation indoor

- Vue physique



7. Gérer les entités de la base de données

Cette partie requiert en premier lieu que la base de données “cyberholocampus” (nom configurable dans le fichier **application*.yml**, cf. partie configuration) soit créée au préalable.
Les tables de la base de données sont créées et remplies au moment ou la commande ***./mvnw*** est exécutée. Cette commande va générer l’application dans son intégralité si ce n’est pas déjà le cas. Il est possible de manipuler la base de données en utilisant l’API ou bien via l’interface web.

8. Communication du casque hololens avec le serveur

Le casque hololens communique avec l’application JHipster via des requêtes REST. Ce générateur fournit en effet une API grâce à swagger ce qui rend tout le back-end facilement testable. Du côté de l'Hololens, il a fallu automatiser certaines de ces requêtes pour les rendre utilisable avec le core du système. Pour cela, nous utilisons des fonctions déjà présentes dans le sdk.

```
public IEnumerator Get(string ipServer, onComplete onComplete)
{
    string url = ipServer + "url/of/api";

    using (UnityWebRequest www = UnityWebRequest.Get(url))
    {
        yield return www.SendWebRequest();
        if (www.isNetworkError || www.isHttpError)
        {
            Debug.Log(www.error);
        }
        else
        {
            onComplete(www.downloadHandler.text);
        }
    }
}
```

Exemple de requête get

OnComplete est à définir autre part dans le code et représente le lien entre l'exécution de cette fonction et une fonction de callback.